

DM 14,-  
OS100,-Stf.14,-

SH 1

# HAPPY COMPUTER

**AKTUELLES  
SONDERHEFT**

*Alternativen zur Kassette*

**Vom Microdrive  
bis zur Floppy.**

*Grundlagen, Tips & Tricks*

**Alles über Interface 1**

*Basic-Programme an ZX 81  
und Spectrum anpassen*

**Programmierhilfen:  
So wird's gemacht**

**Einmalig:  
ZX 81-Autostart  
als Bausatz**

**Neue Top-Listings  
mit ausführlichen  
Beschreibungen für  
ZX 81 und Spectrum**

*Mit ausführlichen  
Anleitungen*

**Geld gespart:  
Spectrum-Hardware  
selbst gebaut**



# Drei Bücher für Sinclair-Besitzer:

Stephen Adams  
Ian Beardsmore  
John Gilbert

## Alles über Sinclair Computer

Software, Peripherie und Hintergrundstory

Auf dieses Buch haben Sie als Besitzer eines Sinclair-Computers schon lange gewartet. Hier finden Sie so ziemlich alles, was Sie über Ihren Computer wissen wollen: Welche Hard- und Software wo angeboten wird, wie teuer sie ist und was sie leistet, dies aber nicht bloss summarisch, sondern mit allen Anschriften der jeweiligen Produzenten. Eine neue Tastatur oder eine RS 232-Schnittstelle finden Sie genauso leicht wie RAM-Erweiterungen oder ein Sprachausgabe-Gerät. Und falls Ihnen das noch nicht reicht: Im dritten Teil des Buches dann die authentische Sinclair-Story, die einem Krimi in nichts nachsteht; und zu guter Letzt Tips und Kniffe zur Problembeseitigung bei Schwierigkeiten mit der Hardware. DAS Kompendium für ernsthafte Anwender!

1984. 172 Seiten, zahlreiche Abbildungen, Broschur.  
sFr. 26.80 / DM 29.80  
ISBN 3-7643-1625-X

Andrew Pennell

## ZX Microdrive-Buch

Programme, Maschinencode, Netzwerke

Das ZX-Microdrive, bei zahlreichen Spectrum-Besitzern bereits im Einsatz, bietet Möglichkeiten, die bisher auf wesentlich teureren Geräten mit Diskettenlaufwerk realisierbar waren. Das Buch zeigt Ihnen diese Möglichkeiten auf: Wie Sie Files organisieren und mittels BASIC- oder Maschinencode-Programmierung das Microdrive steuern können. Ebenfalls erläutert werden Begriffe wie «Strom» oder «Kanal»; Sie erfahren aber auch, wie Sie das im Buch enthaltene Datenbank-Programm anwenden und so das Microdrive optimal nutzen können. Besonderen Wert legt der Verfasser auf Details über den Gebrauch des RS 232-Netzwerks und des ZX Interface 1 auf Maschinensprachebene. Das Buch stellt somit sicher eine notwendige und nützliche Ergänzung zum originalen Sinclair-Benutzerhandbuch dar.

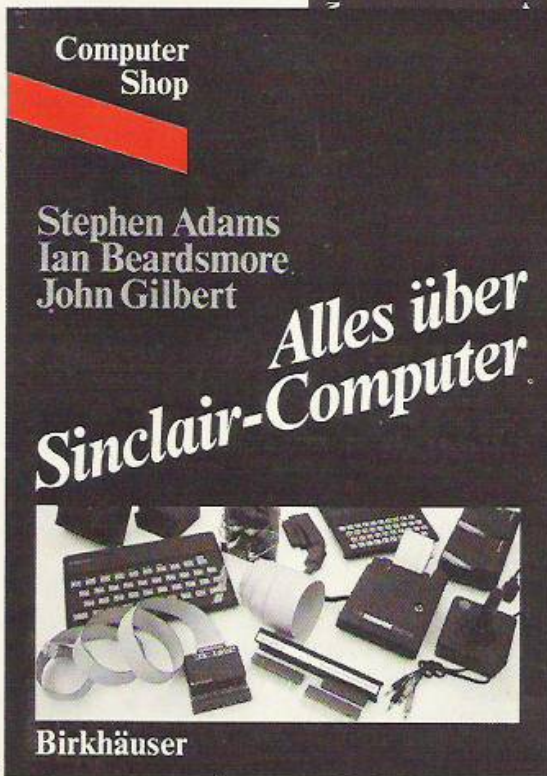
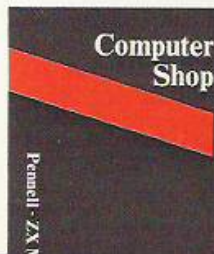
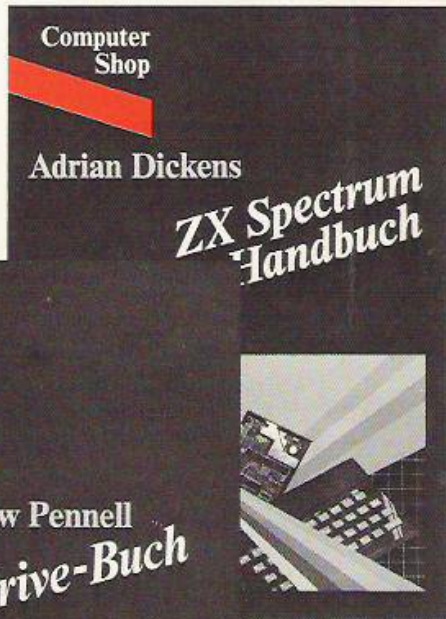
1984. 136 Seiten, Broschur.  
sFr. 26.- / DM 29.80  
ISBN 3-7643-1600-4

Adrian Dickens

## ZX Spectrum Hardware-Handbuch

Der Sinclair ZX Spectrum hat die Mikrocomputerwelt revolutioniert: Einen Computer mit derart raffinierten Farb- und Grafikmöglichkeiten zu einem solch tiefen Preis anzubieten wäre vor ein paar Jahren noch ein Ding der Unmöglichkeit gewesen. In diesem Buch nun erklärt Adrian Dickens etliche Besonderheiten, die im Original-Handbuch von Sinclair nicht zur Sprache kommen: Wie Sie zum Beispiel den Computer an Ihren Color-TV-Apparat anpassen können, wie der Ton des internen Lautsprechers verstärkt werden kann, usw. Ausserdem natürlich weitere Tips zum Betrieb mit dieser vielseitigen Maschine.

1984. 104 Seiten, Broschur.  
sFr. 15.- / DM 28.80  
ISBN 3-7643-1621-7



Ausschneiden und einsenden an:  
**Birkhäuser Verlag AG**  
Ringstrasse 39  
CH-4010 Therwil

**B**  
Birkhäuser  
Verlag  
Basel · Boston · Stuttgart

Bitte senden Sie mir den ausführlichen Prospekt zur Reihe «Computer Shop»!

Name: \_\_\_\_\_

Anschrift: \_\_\_\_\_

Stand September 1984.  
Bei allen Angaben Änderungen vorbehalten.



## Am Anfang...

war da das Problem, wie bringe ich die vielen guten und wichtigen Artikel, die uns angeboten wurden, in der Happy-Computer unter. Und dann waren da noch die Berge von tollen, aber langen Listings. Unsere Leser aus der Sinclair-Ecke sind besonders aktive Computer-Freaks. Und das muß belohnt werden: mit einem Sonderheft zum Beispiel.

Ich möchte die Gelegenheit nutzen und mich ganz herzlich bei Ihnen fürs Mitmachen bedanken, besonders natürlich bei den Autoren dieses Sonderheftes. Hierzu zählen auch zwei weibliche, was ich besonders gut finde. Die Damen melden sich bei dem Thema Computer noch viel zu wenig zu Wort, obwohl sie recht gut mitreden können.



James Hutchby so erfolgreiche Programme wie »Chess the Turk« und »Master Toolkit« fabriziert. Da kann man nur gespannt auf die angekündigten Spiel- und Anwenderprogramme warten.

Warten mußten die Spectrum-Benutzer auch recht lange auf ein funktionierendes Floppy-Disk-System. Seit August ist eines auf dem Markt, und weitere werden auch nach unseren Informationen ebenfalls noch vor Weihnachten zu kaufen sein. Darunter auch ein 3-Zoll-Laufwerk, welches von der Größe her wohl am ehesten zum Spectrum paßt.

## Auf Ihr spezielles...

Mit schöner Regelmäßigkeit erreichen uns Leserbriefe, wir sollten doch mehr über diesen Computer berichten oder mehr Programme für jenes System veröffentlichen. Da es zu viele verschiedene Computer gibt, können wir's natürlich nicht jeden Monat jedem recht machen. Zumindest den Benutzern von ZX81 und Spectrum wollen wir's einmal recht machen — mit diesem Sonderheft, das ausschließlich ihnen gewidmet ist. Dabei haben wir auch Gelegenheit, längere Programme zu veröffentlichen, die zwar interessant sind, aber normalerweise aus Platzmangel in Happy-Computer nicht ohne weiteres unterzubringen wären. Wir haben auch versucht, in diesem Heft einmal Erfahrungen zusammenzufassen — etwa, welches die schönsten Spiele, die nützlichsten Standardprogramme, die besten Bücher sind — und dem Sinclair-Benutzer eine Übersicht über die wichtigsten Peripherie-Bereiche (wie Massenspeicher) zu geben. Wir bieten Ihnen außerdem Berichte über einzelne Produkte, die unsere Mitarbeiter für besonders gut halten — weil wir glauben, daß das für die Benutzer letzten Endes sinnvoller ist als eine Sammlung von möglichst vielen Produktbeschreibungen. Wir hoffen, daß es Ihnen gefällt und freuen uns auf Kommentare. Zum Wohle!

(Michael Pauly, Chefredakteur)

## Sprachgenie

Zum Thema Mitreden fällt mir noch ein, daß es neben Basic, Pascal und Forth mittlerweile auch Logo und »C« — letzteres lieferbar ab Anfang 85 für knapp 100 Mark — für den Spectrum gibt. Er kann also recht vielsprachig mitreden.

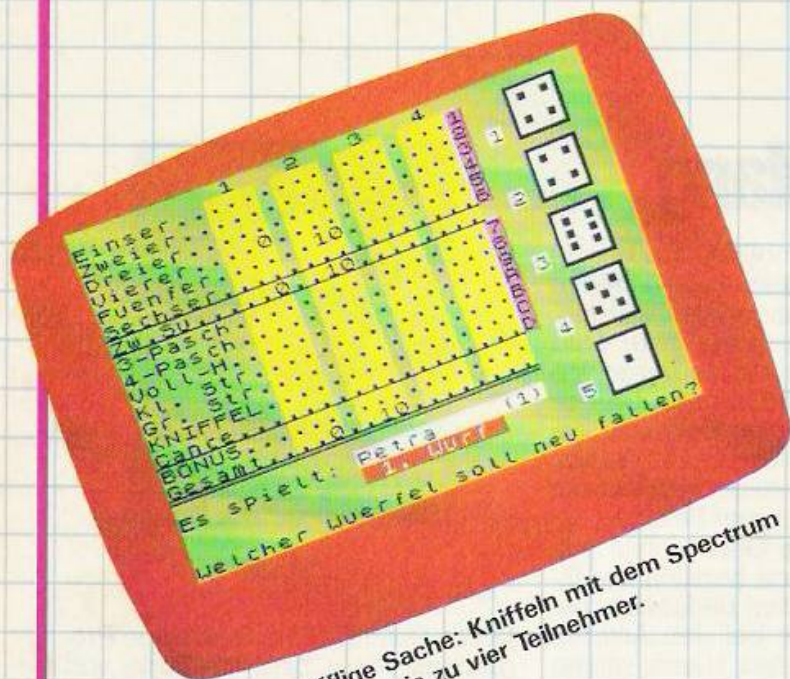
Schweigen muß ich leider zum Thema »QL in deutsch«. Uns lag bis Redaktionsschluß für dieses Heft kein »germanisierter« QL vor. Und das britische Modell mochten wir hier nicht ersatzweise anbieten, da die Tastatur nicht der deutschen Version entspricht und die mitgelieferte Software und auch das Handbuch eingedeutscht werden sollen. Übrigens ist dies ja wohl ein toller Fortschritt für einen englischen Hersteller. Aber auch Herr Sinclair kann nicht auf Dauer an den Wünschen der Kunden vorbei. Ebenfalls positiv zu vermerken ist, daß sich ein großes britisches Softwarehaus, die Firma OCP, entschlossen hat, Teile ihrer Programmpalette für den hiesigen Markt zu übersetzen. Immerhin hat die Oxford Computer Publishing mit ihrem Chefprogrammierer

## ZX81 am Ende?

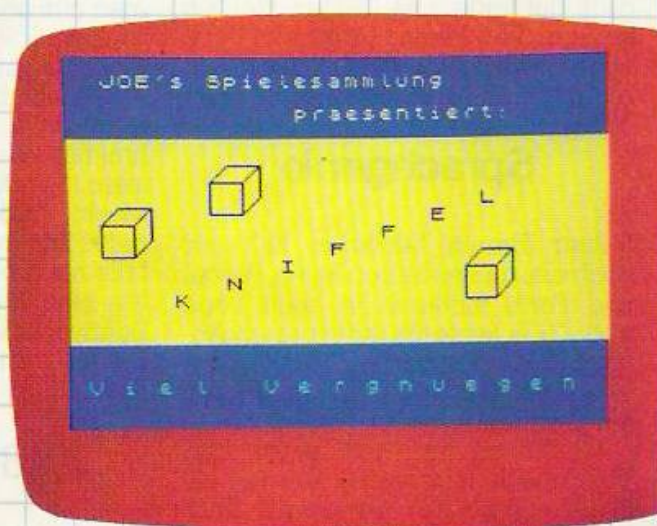
Was vielen nicht paßt, ist die Tatsache, daß es alle Neuheiten nur für den Spectrum gibt, der ZX81 also vernachlässigt wird. Diese Entwicklung ist aber nicht zu übersehen und wird wohl auch von Sinclair geplant sein. Indiz dafür: Der ZX81 ist aus der Sinclair-Werbung verschwunden.

Ich will Sie nicht länger auf die Folter spannen oder gar langweilen, auch dem ZX81 ist ein Teil dieses Heftes gewidmet, und ich wünsche Ihnen viel Spaß und noch mehr gute Tips und Informationen beim Lesen unseres oder besser gesagt Ihres Sonderheftes.

(Manfred-D. Kottling, Redakteur)



Eine knifflige Sache: Kniffeln mit dem Spectrum  
16 KByte für bis zu vier Teilnehmer.



Bringen Sie Ordnung in das farbige Chaos. Babylon ist ein Strategiespiel für einsame Stunden mit dem 16-KByte-Spectrum.



Rund um Sinclair  
Hintergründiges aus der Szene 8

Der schöne Koreaner  
Eine SW-Monitorfamilie stellt sich vor 12

Mit Nadel und Faden  
Fädelttechnik für Eigenbau-Schaltungen 15

Massenspeicher in Massen  
Welche Speichermedien gibt es für Sinclair-Computer? 30

Hexloader  
Hexerei für Maschinencode-Listings 35

Kompatibilität  
Programmanpassung an andere Basic-Dialekte 127

Die Zahlensysteme und ihre Umwandlung  
Mathe-Grundkurs für Programmierer 133

Programmieren in Basic  
Eine programmierte Unterweisung 148

Bücher für Sinclair-Fans  
Empfehlenswerte Lektüre für Lernwillige 152

## Spectrum

### Hardware

Ein RGB-Monitor  
Viel fürs Auge, nichts fürs Ohr 14

Centronics-Interface  
Schnittstelle im Selbstbau 16

Mini-Port  
Einfache Schaltmöglichkeiten 20

Soft-Schalter  
Schalten mit Basic-Befehlen 23

Joy 31  
Bastelanleitung für ein Joystick-Interface 25

Notbremse  
Notausgang für Endlos-  
schleifen 27

Interface 1  
Die Grundlagen 140

Interface 1  
Praktische Anwendung 144

**Zum besseren Verständnis**

OPEN/CLOSE  
Von Strömen und Kanälen 28

Tips und Tricks für den  
Spectrum 66

In/OUT  
Kommunikation mit der  
Peripherie 153

**Software-Tests**

Zum Abheben  
Flugsimulatoren für den  
Spectrum 134

Ines  
Mehr als nur Text-  
verarbeitung 155

Tasword 2  
Textverarbeitung mit dem  
Spectrum 157

Mono  
Eine vielseitige  
Maschinencode-  
Programmierhilfe 158

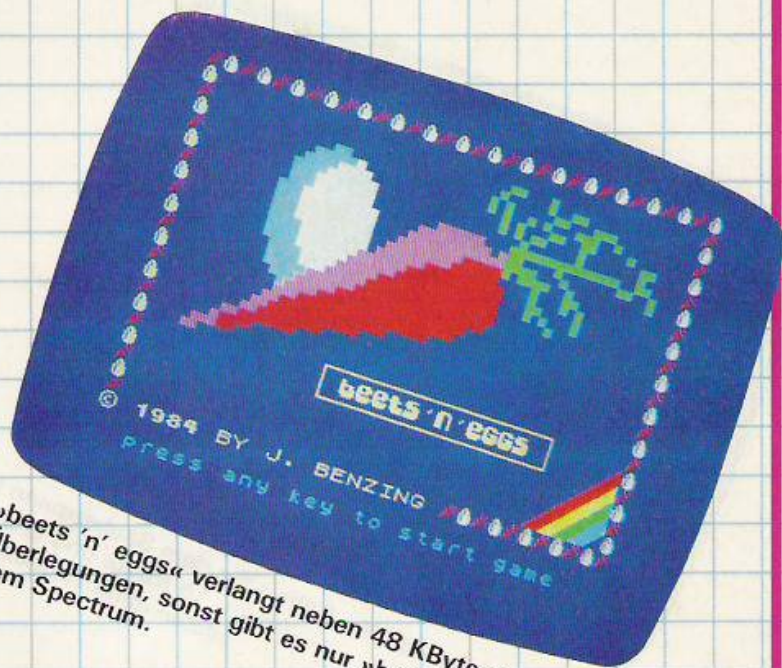
Multi  
Adreßdatei für Tasword 159

Meine liebsten Spielereien  
Eine nicht repräsentative  
Spiele-Hitliste 160

**ZX81**

Kopierschutz  
Harte Zeiten für Piraten 122

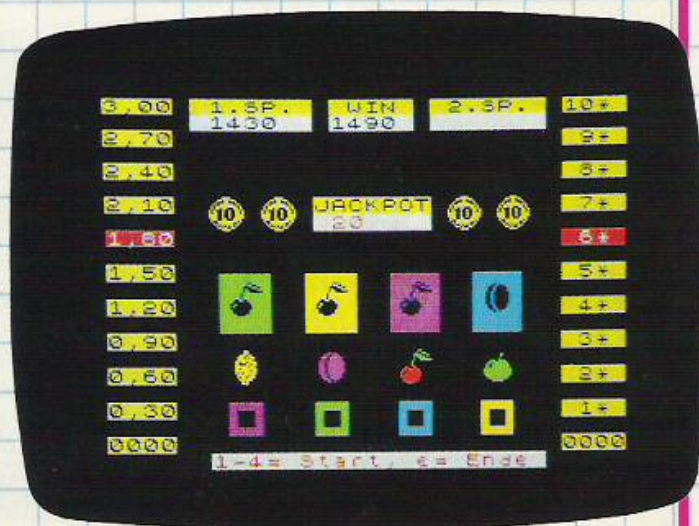
Autostart  
Programmstart ohne  
»Anschieben« 125

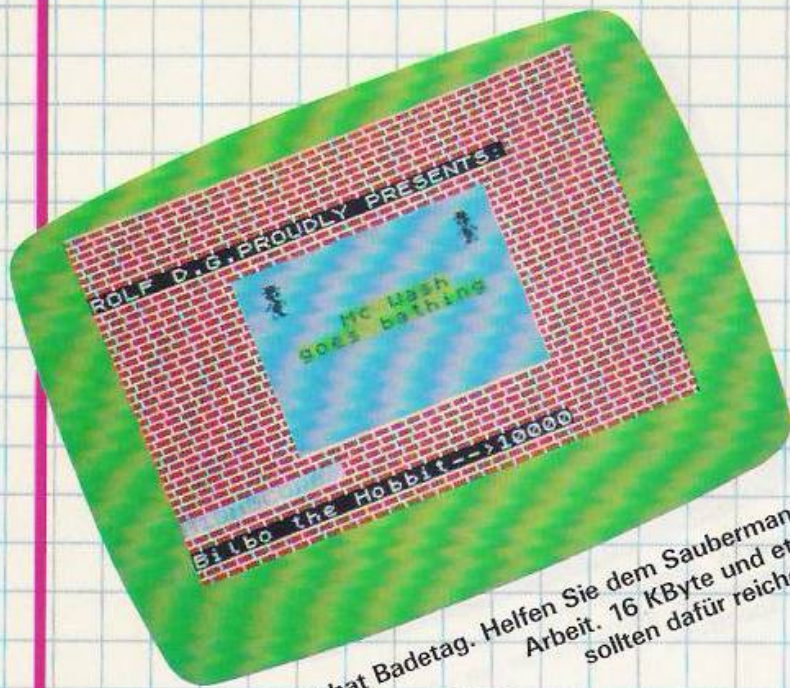


»beets 'n' eggs« verlangt neben 48 KByte strategische Überlegungen, sonst gibt es nur »broken eggs« auf dem Spectrum.

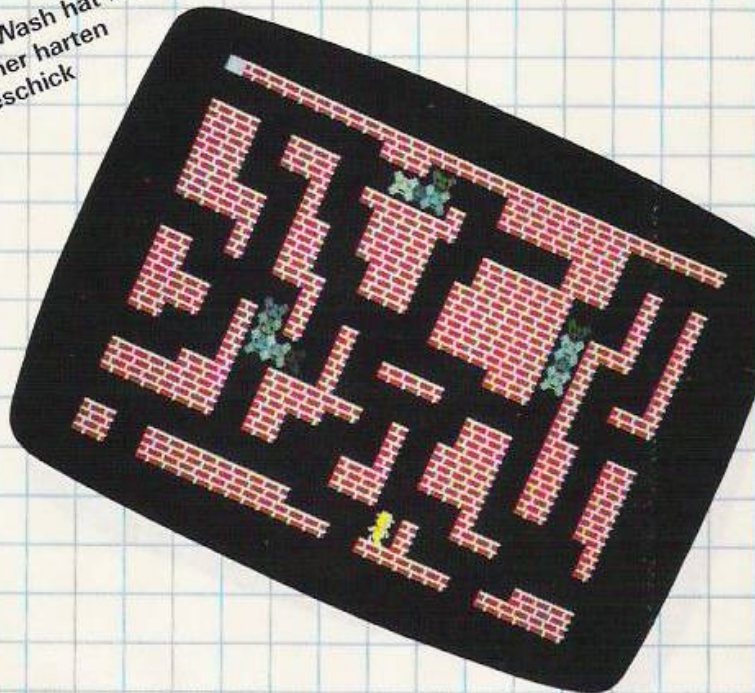


Jackpot ohne Geldverlust. Bingo für 48 KByte und Spielernaturen. Perfekte Automaten-simulation auf dem Spectrum.

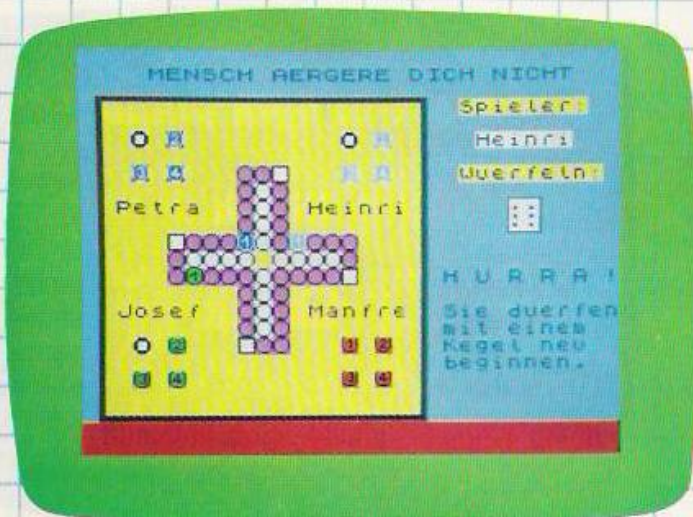




Mc Wash hat Badetag. Helfen Sie dem Saubermann bei seiner harten Arbeit. 16 KByte und etwas soltten dafür reichen.



Immer wieder beliebt: Mensch ärgere Dich nicht. Der Spectrum erspart den Würfeln und das Spielbrett. Mogeln entfällt ebenfalls.



## Spectrum-Listings

### Spiel listings

Jackpot ohne Geldverlust  
Bingo für 48 KByte und Spielernaturen — eine perfekte Spielautomaten-Simulation 37

Babylon  
Bringen Sie Ordnung in das farbige Chaos. Ein Strategiespiel für einsame Stunden mit dem 16-KByte-Spectrum 44

Kniffel  
Eine knifflige Sache für den 16-KByte-Spectrum und die maximal vier Mitspieler 47

Beets'n eggs  
Dieses Spiel verlangt neben 48 KByte strategische Überlegungen, sonst gibt es nur »broken eggs« auf dem Spectrum 52

Mc Wash  
Helfen Sie dem Saubermann bei seiner harten Arbeit am Badetag. 16 KByte und etwas Geschick soltten reichen 57

### Anwendungslistings

Mensch ärgere Dich nicht  
Der Spectrum erspart den Würfeln und das Spielbrett. Mogeln entfällt ebenfalls 99

Roulette  
Der Spectrum als Spielbank für bis zu vier Spielernaturen mit chronischem Geldmangel 107

Schach-Editor  
Zum Nachspielen, Protokollieren und Analysieren von Schachpartien mit Hilfe des 48-KByte-Spectrum 114

Hardcopy  
Der COPY-Befehl für Matrixdrucker und Interface 1 für Kopien in einfacher oder vierfacher Größe 62

## ZX81-Listing

### Spielisting

Zappel  
Punktejagd mit dem  
16-KByte-ZX81 67

### Anwendungslistings

Statistik  
Auswertung von Meßreihen  
mit dem ZX81 und 16-KByte-  
RAM 71

Kalender  
Rund um die wichtigen Ter-  
mine der Jahre 1582 bis 2399 84

### Utilities

Listschutz  
Tarnen und Täuschen mit  
dem ZX81 80

Scroll  
24 Zeilen auf dem Drucker 83

Overload  
LOAD und SAVE für  
ZX81-Programme über  
RAMTOP 92

Toolkit  
Software für RENUM, DELE-  
TE, FIND, HEXTODEC und  
DEZTOHEX 94

Clubadressen 43

Editorial 3

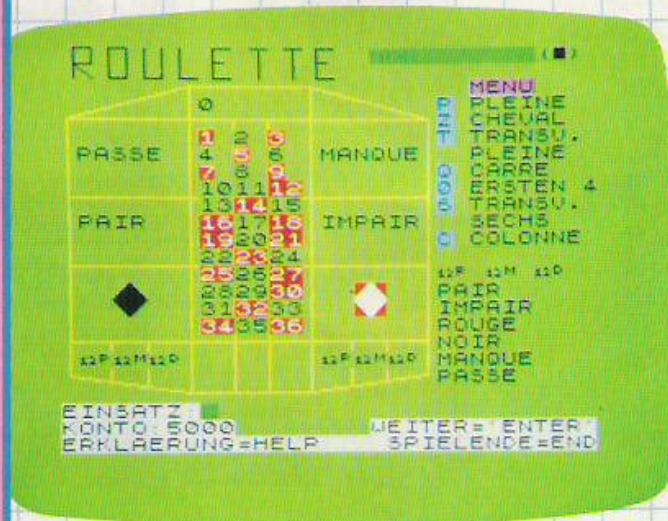
Impressum 162

Autorenverzeichnis 162

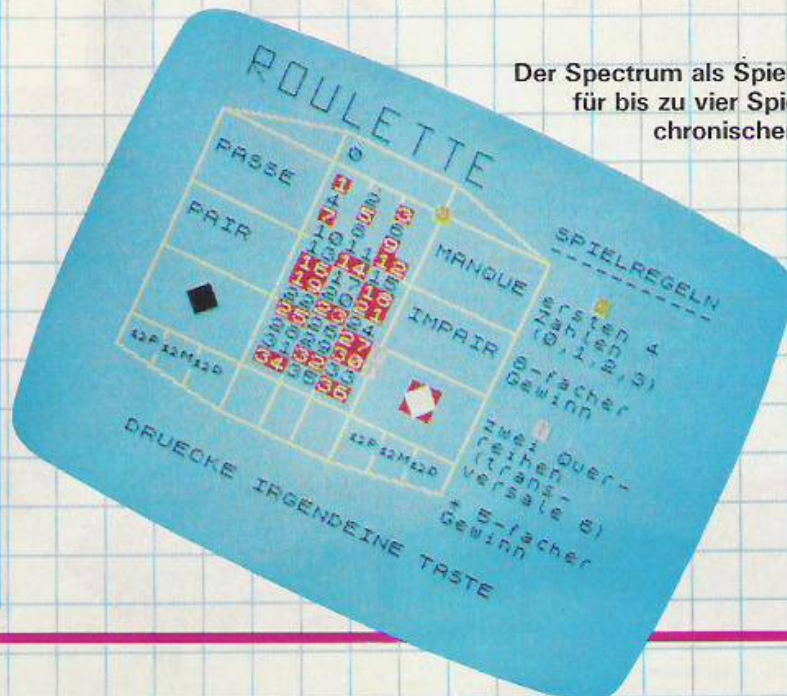


Schachfreunde  
aufgepaßt!  
Dieser Schach-  
Editor für den

48-KByte-Spectrum erleichtert  
das Nachspielen, Analysieren und  
Protokollieren von Schachpartien.

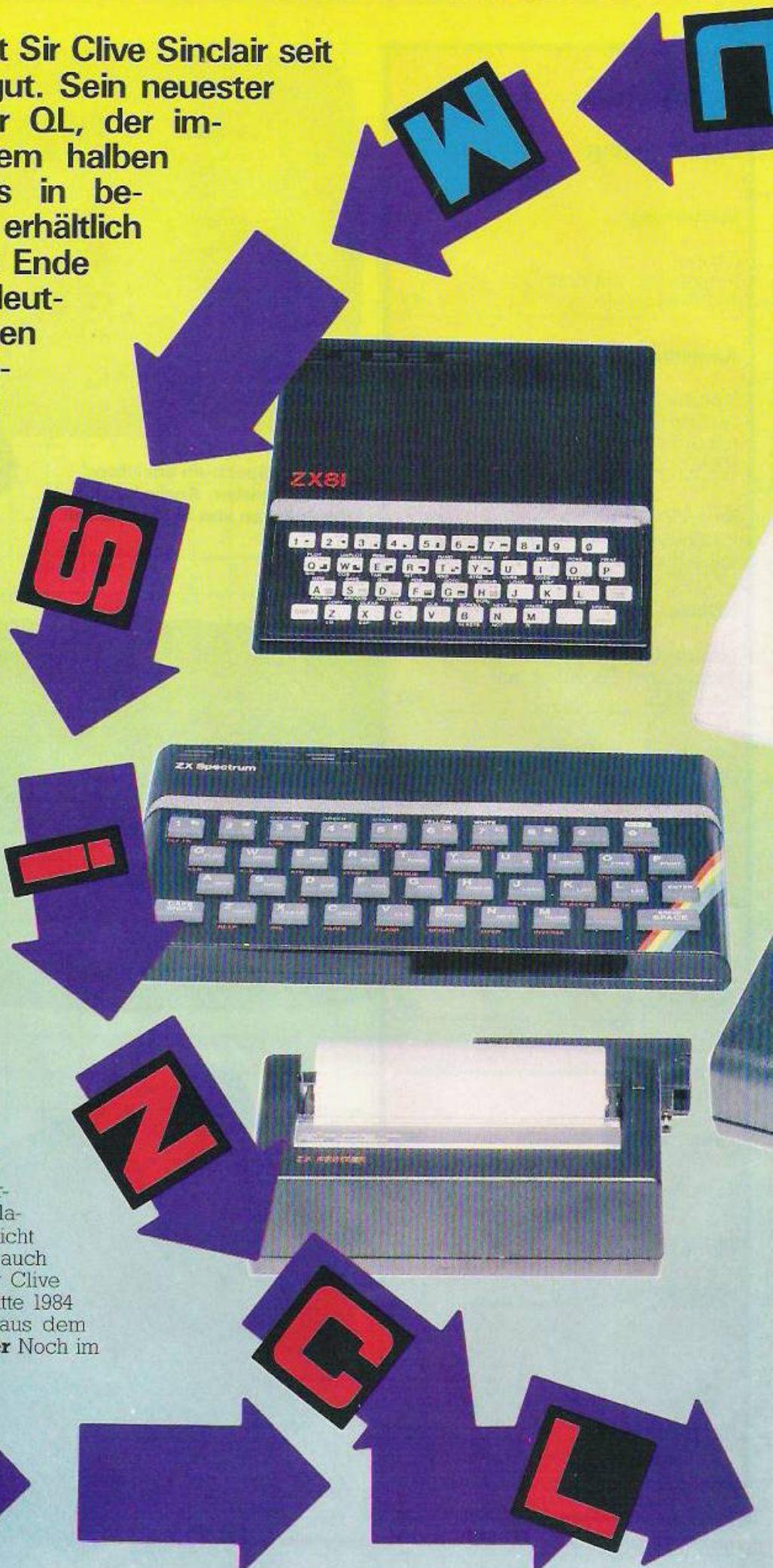


Der Spectrum als Spielbank. Roulette  
für bis zu vier Spielernaturen mit  
chronischem Geldmangel.



Für Überraschungen ist Sir Clive Sinclair seit Anfang 1980 immer gut. Sein neuester Streich ist der Sinclair QL, der immerhin nach gut einem halben Jahr Werbung bereits in begrenzten Stückzahlen erhältlich ist (in England) und ab Ende Oktober auch in der deutschen Version vorliegen soll. Noch überraschender ist die geänderte Absatzstrategie von Sinclair.

Hatte sich Sinclair bis Anfang 1984 nur um den britischen Markt gekümmert, begann plötzlich Interesse für den »Rest der Welt« aufzukommen. Stein des Anstoßes könnte unter anderem der Timex-Fall in den USA gewesen sein. »Timex übertrumpft Sinclair« lauteten voreilige Pressemeldungen noch im November 1983 und bezogen sich damit auf die Vorstellung neuer Geräte, die vom ZX81 und Spectrum abgeleitet wurden und mit denen Timex seinen eigenen Weg gehen wollte. Gemeint sind hier der TS 1500, ein ZX81 im Spectrum-Gewande, und der TS 2068, ein 48-KByte-Spectrum, mit ROM-Modul-Steckplatz. Gleichzeitig mit der Vorstellung dieser Geräte wurde der Preis der Vorgänger drastisch gesenkt und damit dem allgemeinen Preisverfall auf dem US-Markt angepaßt. Dies war dann auch schon »Time ex«. Zur Zeit ist die Timex-Produktionsstätte in Schottland, die schon immer für Sinclair mitproduzierte, mit Sinclair-Produktionen ausgelastet, so daß der Timex-Untergang nicht den europäischen Markt betrifft. Und auch bei Timex in Spanien wird eifrig für Clive produziert. **Blick über den Kanal** Mitte 1984 überschlugen sich die Meldungen aus dem Hause Sinclair: **Sinclair kommt näher** Noch im





Juni will Sinclair in Frankfurt eine Niederlassung gründen. Sie soll unter anderem die Aktivitäten des deutschen Generalimporteurs in Otto-  
brunn, Jürgen Schumpich, unterstützen. Weitere kontinentale Zweig-  
stellen sollen folgen, zum Beispiel im Herbst in Paris. Der offi-  
zielle Grund: Sinclair sieht in Europa große Absatzmärkte  
für seine Produkte und rechnet 1984 mit dem Verkauf von  
einer halben Million Computer. Eine andere Interpre-  
tation lautet freilich so: Nach dem Ausstieg von Ti-  
mex aus dem amerikanischen Markt bleibt Sin-  
clair nur noch die Flucht in den europäisch-  
kontinentalen Markt, da der Aufbau eines ei-  
genen Vertriebsnetzes in den USA derzeit  
außerhalb der Reichweite von Sinclair liegt.

Durch den ab Herbst erhältlichen QL  
soll der Absatz 1985 mehr als verdoppelt  
werden. Als Anwender kann man nun  
hoffen, daß Sinclair nun endlich seine  
Lieferversprechungen wenigstens  
zur Hälfte einhalten wird. Man  
müßte sich sonst fragen, worauf  
sich die prognostizierten Absatz-  
zahlen stützen.

Die neuen Büros werden sich  
laut Sinclair auch um die Ent-  
wicklung von Software in der  
jeweiligen Landessprache  
kümmern. Gleichzeitig mit  
dieser Ausweitung von Sin-  
clair übernimmt Schumpich  
den Vertrieb in Österreich  
und tritt damit die Nachfolge  
der Elektronova in Wien an.  
Inzwischen hat sich Sinclairs  
»Botschafter« in Bad Hom-  
burg etabliert und die Arbeit  
aufgenommen. Sinclairs »Hon-  
orarkonsul« Schumpich ist  
damit faktisch eine Konkur-  
renz entstanden. Dafür trat  
er in neue Märkte (Öster-  
reich) ein und hat seine An-  
gebotspalette um einige  
»Nicht-Sinclair-Produkte«  
erweitert. Die erste Nachricht  
auch Bad Homburg kam am  
1. August: **Sinclair Research**

#### **weitet Fertigungskapazität aus**

Bad Homburg/Cambridge, 1. Au-  
gust 1984 + + + Sinclair Research,  
Cambridge, plant eine drastische  
Erhöhung der Fertigungskapazität  
ihrer Produkte. Vom Home Computer  
ZX Spectrum sollen bis Ende 1984  
mehr als 200000 Einheiten im Monat  
hergestellt werden. Die monatliche Fer-  
tigung des neuen Sinclair Personal Com-  
puters QL wird auf 50000 Stück und die  
Fertigung des Taschenfernsehgerätes auf  
monatlich 20000 Einheiten erhöht. Die Produk-  
tionsmengen wurden drastisch heraufgesetzt,



um die erwartete hohe Nachfrage, insbesondere nach dem ZX Spectrum Computer in der Weihnachtssaison 1984, erfüllen zu können. In der Bundesrepublik Deutschland ist das Unternehmen mit der neu gegründeten Tochtergesellschaft, Sinclair Research Ltd, Branch Office Bad Homburg, vertreten.

Am 13. August erreichte uns folgende Meldung:

## Sinclair: Verstärkte Werbe- und Verkaufskampagne

Bad Homburg, 13. August 1984 — Sinclair Research Ltd., Branch Office Bad Homburg, realisierte in diesem Herbst in der Bundesrepublik Deutschland eine umfangreiche Werbe- und Verkaufsförderungskampagne.

Der mehrere Millionen-DM-Etat, der erstmals auch Rundfunkspots einschließt, soll in erster Linie die Basis für den Absatz des Sinclair Home Computers, ZX Spectrum, verbreitern. Darüber hinaus soll die Kampagne auch den neuen 32-Bit-Personal Computer, Sinclair QL, bekannt machen.

Im Rahmen der breit angelegten Verkaufs- und Werbestrategie erhält der Käufer seinen Spectrum 16 K mit sechs Spielkassetten inklusive. Beim Erwerb eines Spectrum 48 K gibt es acht Spiele inklusive.

Dieses Geschenkpaket, das als Eröffnungsangebot der deutschen Sinclair-Tochter an den Handel und Endverbraucher gedacht ist, wird ab September erhältlich sein.

Das Unternehmen Sinclair Research, das bisher in der Bundesrepublik Deutschland ausschließlich mit einem Generalimporteur in Ottonbrunn vertreten war, wird seine Verkaufs- und Werbeaktivitäten überregional verstärken. Über die bereits traditionellen Absatzkanäle hinaus, die Kauf- und Warenhäuser sowie Verbrauchermärkte, will Sinclair den Verkauf seiner Computer

über den Schreibpapier- und Bürowaren-Einzelhandel, Foto-, Kino-, Optik-Geschäfte sowie den Video-, Radio- und Fernsehhandel forcieren. Neue Distributionswege beschreitet Sinclair mit dem Sortimentsbuchhandel sowie dem Spielwaren- und Bastlerbedarfhandel.

## Einen Tag später hieß es dann: Personal Computer Sinclair QL Ende Oktober erhältlich

Bad Homburg, 14. August 1984 — Der 32-Bit-Rechner Sinclair QL ist Ende Oktober mit deutscher Tasta-

laufwerke mit einer Speicherkapazität von je 100 K für die Massenspeicherung, die sich mit sechs weiteren Microdrives auf 800 K erhöhen läßt. Hinzu kommt Netzwerkfähigkeit: 64 QLs können miteinander oder mit Sinclair Spectrum Home-Computern vernetzt werden.

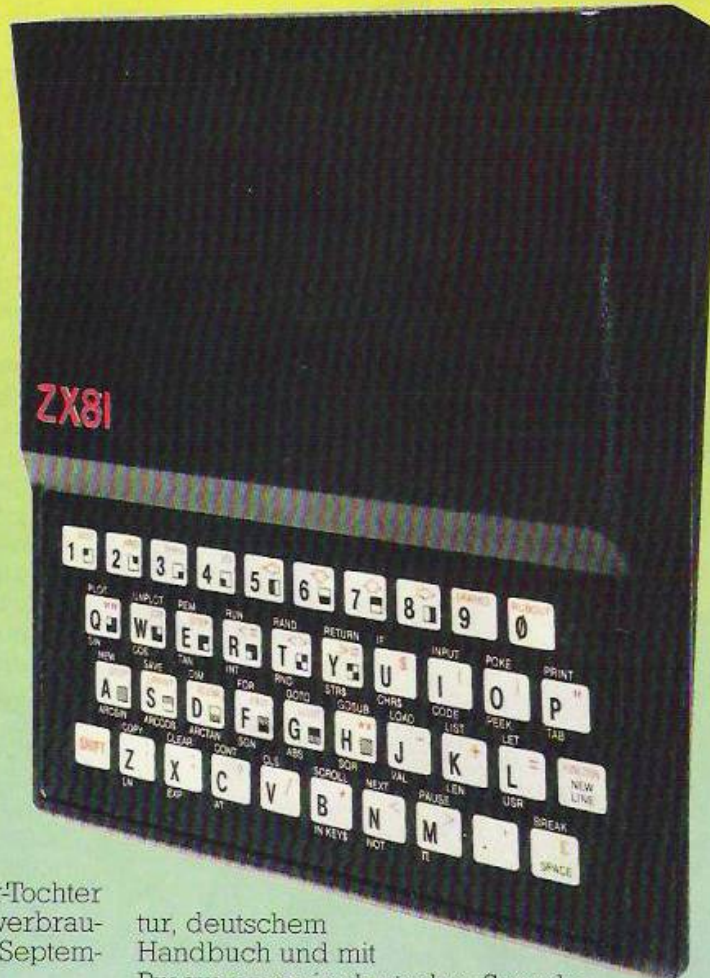
Gedacht ist der neue QL für Selbständige, Freiberufler, kleinere und mittelständische Unternehmen, Schulen und Universitäten, aber auch für anspruchsvollere private Anwender zu Hause. In der Bundesrepublik Deutschland wird der Personal Computer QL 1998,- Mark kosten.

**Im Angebot:** Und damit auch der Letzte merkt, daß sich auf dem Sinclair-Markt was tut, wurden die offiziellen Spectrum-Preise drastisch herabgesetzt und die Kunden mit Software-Dreingaben überrascht. Kostete in offiziellen Preislisten der 48-KByte-Spectrum noch 698 Mark, so ist er jetzt für 598 Mark zusammen mit acht der besten Spectrum-Programme im Angebot. Die 16-KByte-Version mit sechs Programmen kostet offiziell nur noch 498 Mark. Es wurden aber auch schon Preise von 477 Mark für die nackte 48-KByte-Ausführung gesichtet.

**Die Marktlücke:** Zwischen dem preiswerten Spectrum und dem QL klafft sowohl eine Preislücke als auch eine Leistungslücke. Es ist sicherlich wünschenswert, wenn Sinclair diese Lücke selbst schließt. Hier würde sich ein »getunter«

Spectrum mit einer guten Tastatur (aber software-kompatibel) und der Möglichkeit, die alte Peripherie weiterhin zu nutzen, anbieten. Ein »Spectrum 2« so um die 1000 Mark.

**Der QL:** Der Personal Computer QL ist für Oktober/November für den deutschen Markt angekündigt und soll neben dem »Preisvorteil« von knapp 2000 Mark (Preis in England umgerechnet 1600 Mark) über eine



tur, deutschem Handbuch und mit Programmen in deutscher Sprache erhältlich.

Mit diesem ungewöhnlich vielseitigen Gerät, das über vier Grundprogramme für Test- und Datenverarbeitung, Kalkulation und Grafik verfügt, will die kürzlich gegründete Tochtergesellschaft Sinclair Research Ltd., Branch Office Bad Homburg, den Markt für professionelle Computer nachhaltig beleben. In den QL eingebaut sind zwei Mikro-

Tastatur mit deutschen Umlauten, über ein deutschsprachiges Handbuch und eingedeutschte Programme verfügen. Ein Test des QL wird in einer der nächsten Ausgaben von Happy-Computer folgen.

**Drucksachen:** Beendet wurde die Produktion des Metallpapier-Druckers; dies wohl wegen der hohen Reparaturanfälligkeit und auch wegen der besseren und nicht viel teureren Konkurrenten. Sowohl der Alphacom 32, ein echter Thermo-Drucker aus dem Timex-Umfeld, als auch der Seikosa GP 50 S, ein richtiger Nadeldrucker, erfreuen sich wachsender Beliebtheit. Bei der kleinsten »Kreissäge« von Seikosa kann ich das sehr gut verstehen. Sie verwendet Normalpapier auf Rollen und erzeugt ein recht ordentliches Druckbild. Sie können sich von seiner Leistungsfähigkeit im Listingteil dieses Heftes überzeugen. Fast alle Listings und Hardcopies wurden mit dem GP 50 S erstellt. Nachteile: langsam und laut. Die ersten Geräte wurden mit einem Farbband mit auswechselbarem »Inker«, also einer Farbpatrone, ausgeliefert. Damit wurde gezielt so getan, als brauche man später nur diesen Pfennigartikel auszutauschen, statt des 22 Mark teuren Farbbandes. Doch Enttäuschung war vorprogrammiert. Die neuen Farbbänder haben fest eingebaute Farbpatronen und müssen, wenn sie »schwach« werden, komplett ausgetauscht werden. Auch für die alten Bänder gibt es, laut Aussage des deutschen Generalimporteurs Microscan, keine Inker nachzukaufen. Mit einer Farbpatrone habe ich so etwa 100 Meter Listings lesbar ausgedruckt. Und dann – Farbband war nicht zu bekommen – wurde gebastelt. Man nehme einen 3-mm-Bohrer und bohre in die Seite (egal welche) ein Loch in das Plastik. Als Zutat brauchen wir gute Stempelfarbe in kleinen Flaschen, wie sie für Datumsstempel angeboten werden. Von diesem billigen »Saft« geben wir nun etwa

drei Tropfen in das Loch. Das Filzstück im Inneren der Patrone saugt die Farbe auf, und mit Tesafilm wird das Loch wieder verschlossen. Ergebnis: Rund 20 Mark gespart und wieder ein gutes Druckbild.

**Ausgedient?** Auf dem Sektor ZX81 tut sich seitens Sinclair wohl nichts mehr. Auch die Software-Produzenten haben den einst so erfolgreichen Kleincomputer vergessen. Interessant ist jedoch nach wie vor der ZX81-Bausatz, der für alle ICs Steckfassungen hat und den man gleich seinen speziellen Erfordernissen beim Zusammenbau anpassen kann. Für Steuerungs-, Meß- und Regelungsaufgaben gibt es sicherlich keinen besseren Computer auf dem deutschen Markt für unter 100 Mark.

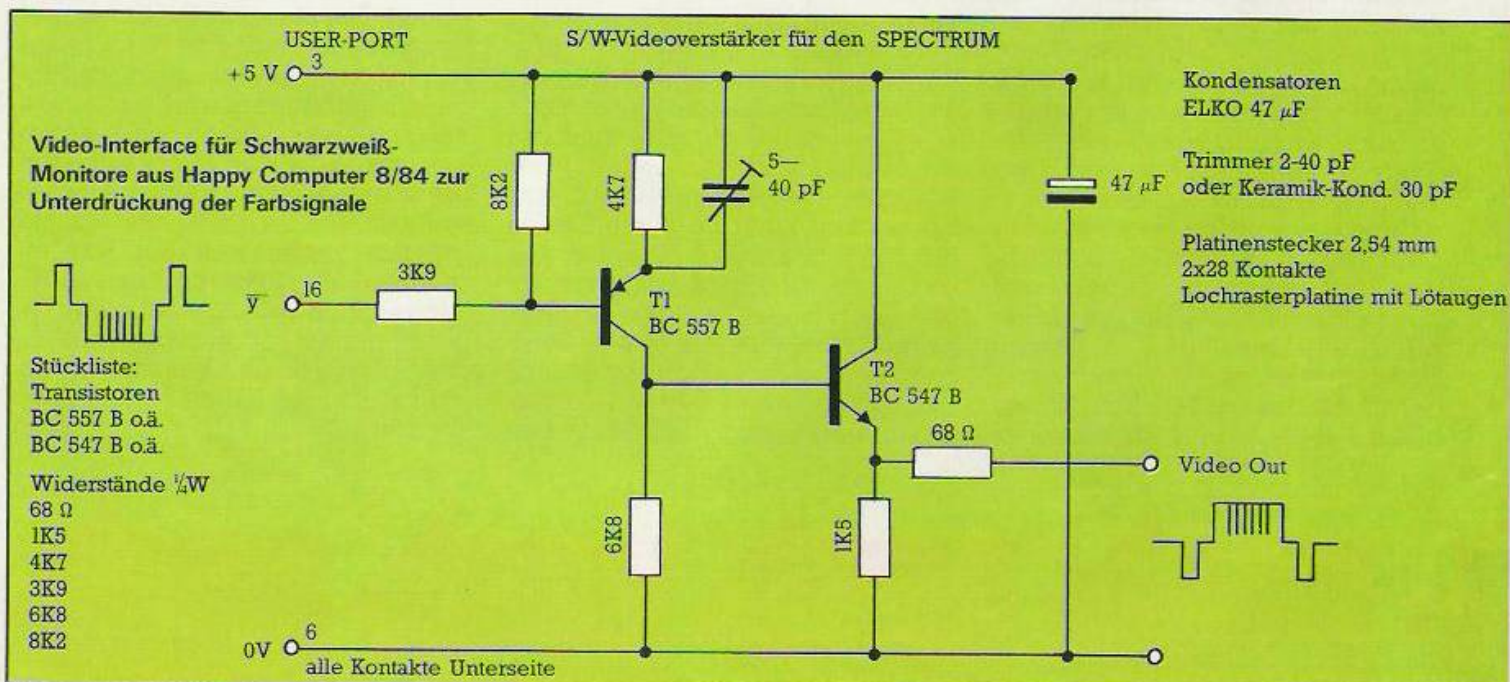
**Spielereien:** Das Interface II hat sich, wie zu erwarten war, nicht durchsetzen können. Daran hat zwar auch der Preis des Gerätes und der Spielmodule schuld, aber auch die etwas merkwürdige Belegung der Joysticks. Die ersten Spiele benutzen zur Steuerung die Cursor-Tasten (5,6,7 und 8). Danach richteten sich die Joystick-Interface-Hersteller und nach denen wieder die Software-Entwickler. Als dann Steve Sinclair mit seiner »Norm« kam, gingen nicht etwa die Software-Produzenten daran, ihre Programme um-

zustricken, sondern die Zubehörinterindustrie ersann die frei programmierbaren Interfaces. Da gibt es welche, die mechanisch dem Spiel angepaßt werden, andere werden durch mitgelieferte Software entsprechend gesteuert und wieder andere verfügen über genug eigene Intelligenz, um sich auf Bewegung in die richtige Richtung des Kundenwunsches zu merken. Letztgenannte sind zwar am komfortabelsten, aber auch am teuersten. Auch die Software-Häuser waren durch die verschiedenen »Normen« verunsichert. Die cleveren Programmierfirmen bauen nun ihre Programme so auf, daß der Benutzer am Programmstart gefragt wird, wie er seine Eingabe machen will beziehungsweise welche Tasten er benutzen oder mit seinem Joystick simulieren will. Wer sich neu ein Joystick-Interface kaufen will, dem rate ich zu einem möglichst universellen Interface ohne Software auf extra Tape. dk-tronics hat da recht ordentliche Angebote (in Deutschland im Mikrocomputer-Laden Berlin zu bekommen). (mk)



# Der schöne Koreaner: Ein

Nicht nur Drucker werden billiger, auch Monitore werden seit einiger Zeit zu interessanten Preisen angeboten. Für viele Anwendungen ist ein guter Schwarzweiß-Monitor nötig. Einer davon soll hier vorgestellt werden, der CDM 900/1200 von Monacor.



# Datenmonitor im IBM-Look

Der Monitor wird in zwei Bildschirmgrößen angeboten, jeweils mit grünem oder orangefarbenem Phosphor. Das formschöne, matte Kunststoffgehäuse in beige hat eine anthrazitfarbene Bildschirmmaske. Die Bildfläche ist leicht geneigt und hat eine reflexionsarme Oberfläche. An der Frontseite befindet sich der Netzschalter, eine LED-Betriebsanzeige und — hinter einer Klappe — die Einstellregler für Helligkeit und Kontrast.

Zwei Videobuchsen befinden sich an der Rückseite des Gerätes. Daneben der Schalter für den Abschlußwiderstand (75 Ohm) und die Regler für den horizontalen und vertikalen Bildfang für Bildgröße und Position.

Nach Lösen von sechs Schrauben läßt sich das Gehäuse entfernen. Auf einem Metallchassis sind Hauptplatine, Netzteil und Netztrafo montiert. Alle Verbindungen sind steckbar. Die Platinen haben einen Bauteileaufdruck und sind sauber aufgebaut.

Zum Lieferumfang gehört eine 16seitige ausführliche Anleitung. Der erste Teil — Anschluß, Inbetriebnahme, Fehlersuche und technische Daten — ist in deutsch, die Serviceanleitung in englisch geschrieben. Der technische Teil ist besonders ausführlich mit Platinen-

foto, Bestückungsplan, Blockdiagramm und Schaltplan mit Oszillogrammen. Alle Bauteile sind in einer Liste spezifiziert.

Dieser Monitor hinterläßt nicht nur auf den ersten Blick einen guten Eindruck. Das Gehäuse sieht ansprechend aus und der Bildschirm liegt in einem günstigen Blickwinkel. Die entspiegelte Oberfläche zeichnet zwar etwas weicher, erlaubt aber eine normale Arbeitsplatzbeleuchtung. Beim Betrieb an einem Spectrum brachte sowohl der Videoausgang in Farbe (1) als auch der in schwarzweiß (2) ein ruhiges, flimmerfreies Bild. Die Größe des Schriftfeldes (32 x 22 Zeichen des Spectrum) mißt beim CDM 1200 zirka 19 x 13 cm, beim CDM 900 zirka 14 x 7 cm. Die beiden Videobuchsen (Cinch) erlauben den Anschluß wei-

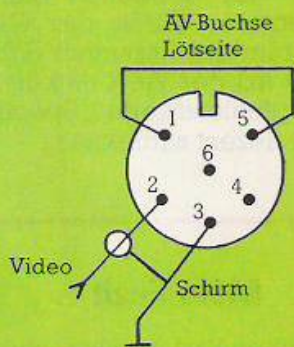
terer Monitore. Man kann zum Beispiel einen Farbfernseher mit AV-Buchse zusätzlich anschließen.

Mit zwei Bildschirmgrößen und zwei Farben stehen vier Geräte zur Auswahl. Der »Kleine« löst zwar auch 64 Zeichen pro Zeile noch gut auf, aber für längere Bildschirmarbeit ist der CDM 1200 wohl doch geeigneter. Ob Grün oder Orange die beste Farbe für einen Datensichtschirm ist, vermag ich nicht zu beurteilen. Im direkten Vergleich empfand ich orange sehr angenehm, aber grün — bei heller Schrift auf dunklem Grund — etwas leuchtender.

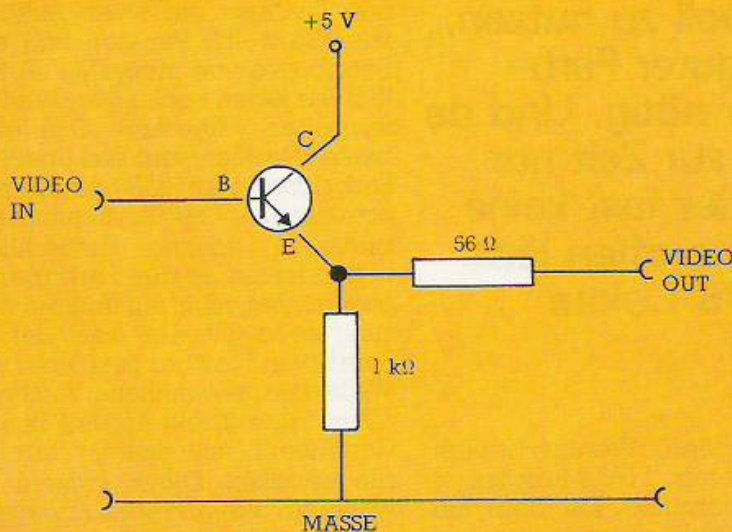
Doch gleich welche Schirmfarbe oder Schirmgröße, dieses Produkt aus Fernost ist bei einem Preis um 350 Mark gewiß keine schlechte Wahl. (Jürgen Howaldt)

## Technische Daten

Auflösung	: horizontal 1000/vertikal 350 Zeichen
Videobandbreite	: typ. 22 MHz
Videoeingangssignal	: 1 V + 0,5 V <sub>SS</sub> , 75 Ohm, Neg. Mod.
Horiz. Synchronisation	: 15,75 kHz ± 0,5 kHz
Vert. Synchronisation	: 47 bis 61 Hz
Ablenkung	: 90°
Bildfläche	: geätzt (non-glare)
Stromversorgung	: 220 V/50 Hz
Stromverbrauch	: 30 W (27 W bei 9 Zoll)
Abmessungen	: B 350 x H 278 x T 330 mm (12 Zoll) B 264 x H 220 x T 253 mm (9 Zoll)
Gewicht	: zirka 7 kg (12 Zoll) zirka 4,8 kg (9 Zoll)



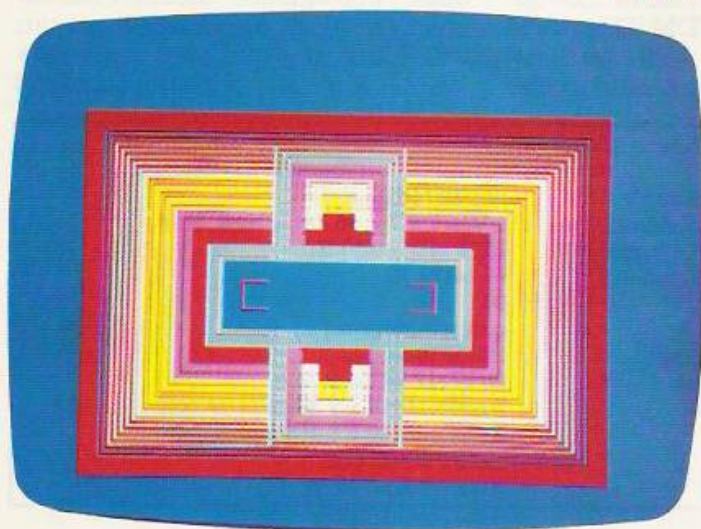
- 1, 5 = Schaltbrücke
- 4, 6 = NF
- 2 = Video
- 3 = Masse



Video-Interface aus Happy Computer 6/84 für Monitor-Anschluß am Spectrum

## RGB-Monitor: viel fürs Auge, nichts fürs Ohr

Der Spectrum kann grafisch mehr, als die Darstellung am Farbfernseher vermuten läßt. Um die Möglich-



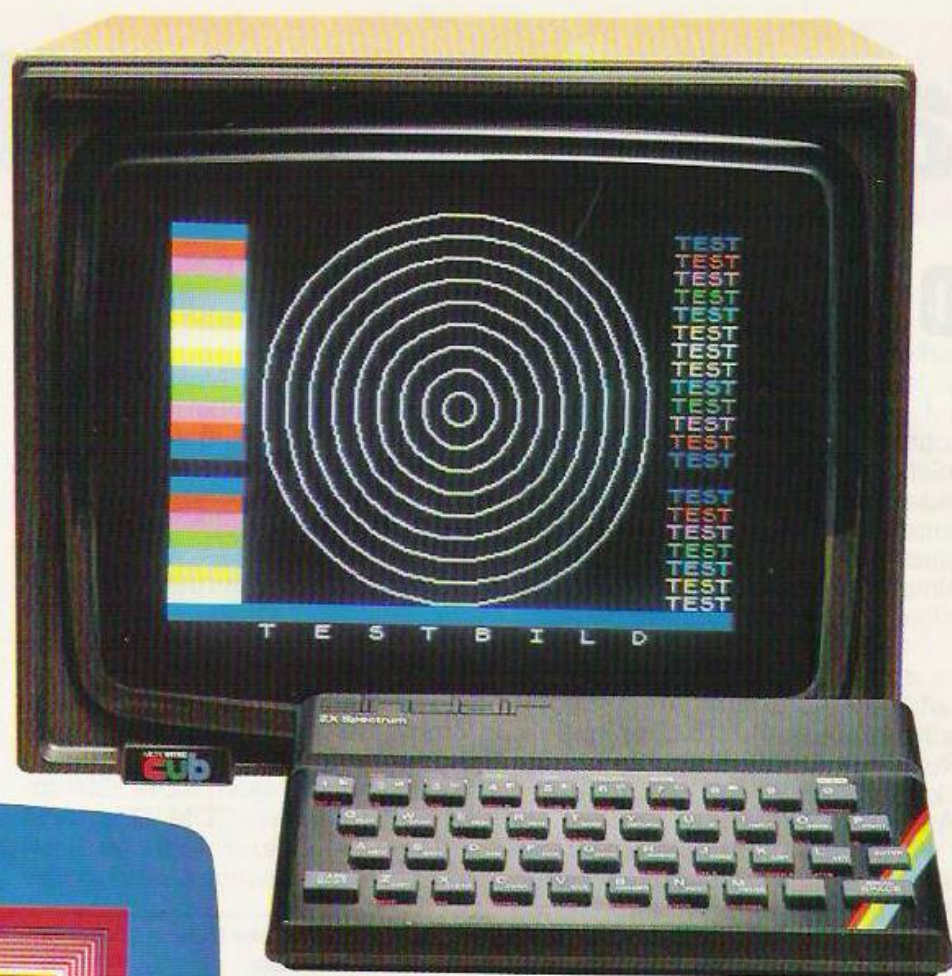
keiten voll zu nutzen, ist ein guter Farbmonitor nötig. Und da gibt es zur Zeit nur einen, der fast keine Wünsche offen läßt: den CUB Colour Monitor.

Ein (großes) Wermutstropfen gleich vorab: 1190 Mark kostet das Prunkstück, von dem ich mich nicht mehr trennen möchte. Dafür bietet der einzige RGB-Monitor mit Spectrum-Interface aber auch einiges. Die ungewöhnliche Brillanz seines 21 x 28 cm gro-

ßen Displays mit einer Auflösung von 585 x 452 Punkten und einer Bandbreite von immerhin 18 MHz, läßt Sie jeden noch so teuren Farbfernseher vergessen. Das eingebaute Interface setzt das Spectrum-Farbsignal in ein RGB-Signal um und erreicht damit ein fast flimmerfreies brillantes Farbbild. Farbschlieren und Schatten an den Farbübergängen tauchen nicht auf und die Auflösung ist so gut, daß auch die 64er Schrift von Tasword gut lesbar wird. Weitere ungewöhnliche Vorzüge:

Anschluß direkt an den Bus des Computers mit einem Durchführungstecker. Damit können also weiterhin alle anderen Erweiterungen gesteckt werden.

Zweiter Signaleingang für Geräte, die ein TTL-Signal für den Monitor zur Verfügung stellen. Die Eingänge sind mit einem Kippschalter um-



▲ Das Testprogramm zeigt die Farbskala und die Auflösung

◀ Dieses Bild läßt die »Farbfreude« erahnen

schaltbar. Damit ist der Monitor zum Beispiel auch für den Sinclair QL geeignet.

Das Gerät hat ein ansprechendes, stabiles Metallgehäuse mit den Maßen 298 (hoch) x 352 (breit) x 387 (tief) und einen Stromverbrauch von nur 65 Watt.

Bedauerlich ist, daß dieser Monitor keinen eingebauten Tonsignalverstärker hat. Ebenso fehlt ihm noch die bereits beantragte FTZ-Nummer der deutschen Bundespost. Letzteres dürfte der Grund sein, warum der Importeur Schumpich sich mit der Werbung für dieses empfehlenswerte Spectrum-Zubehör dezent zurückhält.

### Mein Fazit

Wer sich an Farbgrafiken, sei es in Spielen, sei es in ernsthaften Programmen, begeistern kann (so wie ich), der wird auf lange Sicht nicht um dieses Gerät herumkommen, auch wenn seine bessere Hälfte noch so sehr über den Preis entsetzt ist.

(mk)

# Mit Nadel und Faden



**Hat man eine interessante Schaltung zum Nachbauen gefunden, so taucht die Frage nach dem »Wie« auf. Von der »fliegenden« Verdrahtung, bei der ein Bauteil an das andere gelötet wird, bis zur geätzten Platine ist alles möglich. Während die erste Lösung weder betriebs- noch transportsicher ist, ist die letzte mit viel Arbeit verbunden. Aber es gibt noch eine andere Lösung, die sogar von Profis bei Einzelstücken genutzt wird. Gemeint ist der Aufbau mit einer Lochrasterplatine.**

**F**ür einige Anwendungen ist eine Platine mit Leiterbahnen günstig, aber meist muß man doch zu viele Bahnen unterbrechen und Querverbindungen legen. Auch die Positionierung der Bauteile ist nicht so einfach. Gut arbeiten kann man mit einer Lochrasterplatine mit Lötäugen. Die Bauteile können einfach platziert werden und die Verbindungen stellt man bei einfachen Schaltungen mit blankem Schweißdraht her. Sind mehrere ICs zu verdrahten, so ist es vorteilhaft, nur für die Versorgungsspannung und Masse eine Lötchiene aus blankem Schweißdraht zu legen und die anderen Verbindungen mit isoliertem Kupferdraht

herzustellen. Diesen Draht gibt es zum Beispiel von der Firma Vero in verschiedenen Farben und einer Dicke von 0,15 mm. Dazu passend gibt es einen Fädelstift. Er hat einen Spulenhalter und eine kleine Taste, um den Draht stramm zu halten. Für Ordnung im Drahtgewirr sorgen dann noch Verdrahtungskämme. Sie werden auf die Platine gesteckt und erlauben eine gebündelte Drahtführung mit sauberen Verzweigungen.

Der Lackdraht hat eine Isolierung aus Polyurethan, die beim Löten schmilzt (Dämpfe nicht einatmen). Mit einem LötKolben von 16 bis 30 Watt und einer nicht zu feinen Lötspitze erreicht man die benötigte

Temperatur von zirka 350°. Bei empfindlichen Bauteilen sollte der Draht vorher verzinnt werden, sonst kann er direkt mit einer spitzen Pinzette um ein Drahtende oder Kontaktpin gewickelt werden.

Für integrierte Schaltkreise braucht man bei dieser Technik einen Sockel. Damit schützt man das IC vor thermischer Überlastung und verhindert, daß sich die Schaltung bei einem Wechsel auflöst.

Bei einer »Investition« von etwa 20 Mark lassen sich dann auch komplexe Schaltungen noch sauber und übersichtlich aufbauen.

(Jürgen Howaldt)

# Centronics-Schnittstelle samt Software

Bei der selbstgebauten Centronics-Schnittstelle für den ZX-Spectrum dient die Z80A-PIO als Port-Baustein. Die Copy-Software erlaubt die Ansteuerung des Matrix-Druckers CP 80.

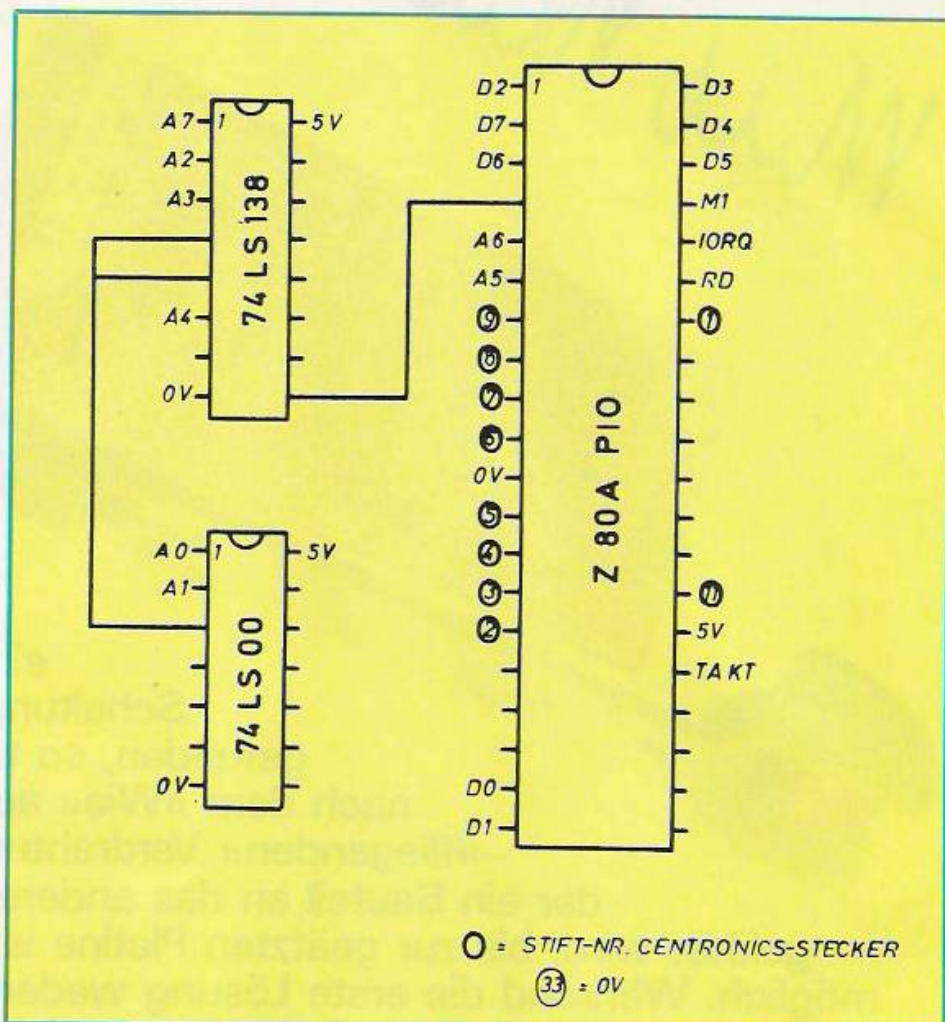
Da die Z80-PIO schon in fast jeder Zeitschrift eingehend erklärt wurde, wird hier darauf verzichtet. Die Schaltung wurde auf einer Lötspunkt-Platine mit Fädeldraht aufgebaut. Die Verbindung zum Computer erfolgt über Flachkabel und Spectrum-Stecker, zum Drucker über Flachkabel und Centronics-Stecker. Anhand des Schaltbildes dürfte es auch weniger geübten Lesern nicht schwer fallen, die Schaltung funktionsfähig nachzubauen.

Damit LPRINT und LLIST benutzt werden können, ist lediglich der Zeiger bezüglich der Druckroutine abzuändern. Für Copy existiert ein solcher Zeiger leider nicht, die Funktion kann somit nur über RANDOMIZE USR ... ausgeführt werden. Zur Arbeitsweise des Programms wird auf das kommentierte Assembler-Listing verwiesen.

Für die 48 KByte-Version kann nach CLEAR 63999 direkt der Objektcode mittels geeignetem Monitor eingegeben werden. Bei der 16 KByte-Version müssen die Sprung- und Call-Befehle entsprechend geändert werden. Der Speicherbedarf beträgt 268 Bytes.

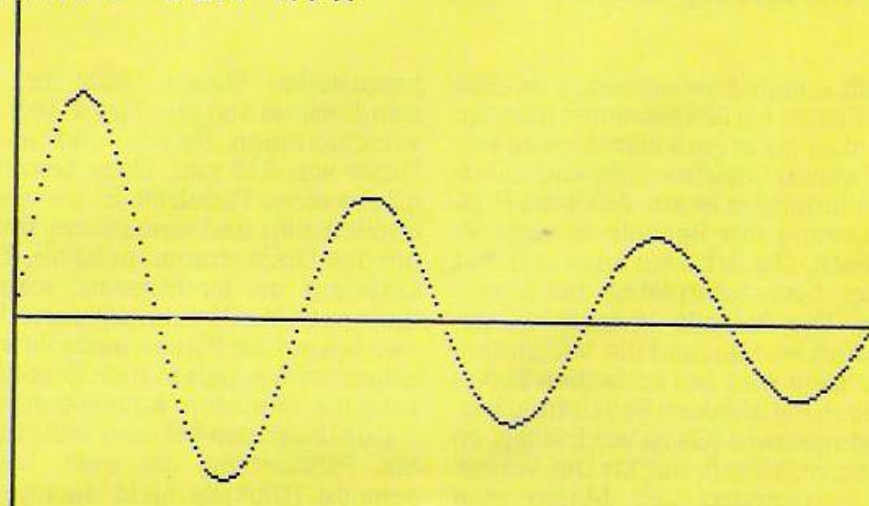
Das Programm wurde so geschrieben, daß beim ersten Aufruf (RANDOMIZE USR 64000) die Initialisierung durchgeführt wird. Bei jedem weiteren Aufruf (RANDOMIZE USR 64000) erfolgt eine Hardcopy des Bildschirms.

(Norbert Hugel)



Alle ⓪ sind Verbindungsstellen zum Drucker, alle anderen bezeichneten Pins führen zum entsprechenden Anschluß am User-Port des Spectrum. Alle 0-Volt-Anschlüsse sind an einer Stelle zu verbinden.

$$f(x) = \text{SIN } x / x$$



Mit dem hier vorgestellten Programm erstellte Hardcopy



ADR.	OBJ.CODE	LABEL	OP	OPERAND		
64000	FA00	C3B1FA	2	JP	INIT	Initialisierung
	FA03	F5	3	PUSH	AF	CPU-Register retten
	FA04	C5	4	PUSH	BC	
	FA05	D5	5	PUSH	DE	
	FA06	E5	6	PUSH	HL	
	FA07	3E1B	7	LD	A,1B	Zeilenvorschub 8/72"
	FA09	CDF0FA	9	CALL	PRINT	
	FA0C	3E41	12	LD	A,41	
	FA0E	CDF0FA	13	CALL	PRINT	
	FA11	3E08	14	LD	A,08	
	FA13	CDF0FA	15	CALL	PRINT	
	FA16	21FF3F	16	LD	HL,3FFF	Display-File 1.Drittel
	FA19	CD39FA	17	CALL	COPY	
	FA1C	21FF47	18	LD	HL,47FF	Display-File 2.Drittel
	FA1F	CD39FA	19	CALL	COPY	
	FA22	21FF4F	20	LD	HL,4FFF	Display-File 3.Drittel
	FA25	0606	21	LD	B,06	Nur 6 Reihen
	FA27	CD3BFA	22	CALL	NEXTR	
	FA2A	3E1B	23	LD	A,1B	Zeilenvorschub 1/6"
	FA2C	CDF0FA	24	CALL	PRINT	
	FA2F	3E32	25	LD	A,32	
	FA31	CDF0FA	26	CALL	PRINT	
	FA34	E1	27	POP	HL	CPU-Register zurueck
	FA35	D1	28	POP	DE	
	FA36	C1	29	POP	BC	
	FA37	F1	30	POP	AF	
	FA38	C9	31	RET		Ende
	FA39	0608		COPY :	LD B,08	8 Display-Zeilen
	FA3B	C5		NEXTR :	PUSH BC	
	FA3C	E5			PUSH HL	
	FA3D	CD5AFA			CALL PIXELR	
	FA40	E5			PUSH HL	
	FA41	CD82FA			CALL INITG	Listing. Ansteuerung des Matrix-
	FA44	CD97FA			CALL OUT	druckers CP 80
	FA47	E1			POP HL	
	FA48	CD5AFA			CALL PIXELR	
	FA4B	CD82FA			CALL INITG	
	FA4E	CD97FA			CALL OUT	
	FA51	E1			POP HL	
	FA52	012000	40	LD	BC,0020	
	FA55	09		ADD	HL,BC	Anfangsadr.naechste Zeile
	FA56	C1		POP	BC	
	FA57	10E2		DJNZ	-30>NEXTR	8 x durchlaufen ?
	FA59	C9	44	RET	-	
	FA5A	0604		PIXELR:	LD B,04	4 Pixelreihen
	FA5C	C5		NEXTPR:	PUSH BC	
	FA5D	11005B			LD DE,5B00	Printer-Buffer
	FA60	0E20			LD C,20	32 Byte/Pixelreihe
	FA62	0608		BYTE :	LD B,08	8 Pixel/Byte
	FA64	23	50		INC HL	
	FA65	CB06		NXTPIX:	RLC (HL)	High-Bit in Carry
	FA67	EB			EX DE,HL	
	FA68	CB16			RL (HL)	Carry in Buffer
	FA6A	EB			EX DE,HL	
	FA6B	CB0E			RRC (HL)	wieder zurueck
	FA6D	CB06			RLC (HL)	nochmal
	FA6F	EB			EX DE,HL	
	FA70	CB16	58		RL (HL)	

59	FA72 EB		EX DE,HL	
	FA73 13		INC DE	Bufferzeiger +1
	FA74 10EF		DJNZ -17>NXTPIX	8 Pixels ?
	FA76 0D		DEC C	
	FA77 20E9		JR NZ,-23>BYTE	32 Bytes ?
	FA79 01E000		LD BC,00E0	
	FA7C 09		ADD HL,BC	Anfangsadr.n.Pixelreihe
	FA7D C1		POP BC	
	FA7E 05		DEC B	
69	FA7F 20DB		JR NZ,-37>NEXTPR	4 Pixelreihen ?
	FA81 C9		RET	
	FA82 3E1B	INITG :	LD A,1B	Graphik-Mode normale Dichte
	FA84 CDF0FA		CALL PRINT	
	FA87 3E4B		LD A,4B	
	FA89 CDF0FA		CALL PRINT	
	FA8C 3E00		LD A,00	512 Bytes
	FA8E CDF0FA		CALL PRINT	
	FA91 3E02		LD A,02	
	FA93 CDF0FA		CALL PRINT	
78	FA96 C9		RET	
	FA97 21005B	OUT :	LD HL,5B00	Printer-Buffer
	FA9A 010001		LD BC,0100	256 Bytes
	FA9D 7E	AUSG :	LD A,(HL)	Muster in Akku
	FA9E CDF0FA		CALL PRINT	ausgeben
	FAA1 7E		LD A,(HL)	nochmal
	FAA2 CDF0FA		CALL PRINT	
	FAA5 23		INC HL	Bufferzeiger +1
	FAA6 0B		DEC BC	
	FAA7 78		LD A,B	
	FAA8 B1		OR C	Buffer leer ?
	FAA9 20F2		JR NZ,-14>AUSG	
90	FAAB 3E0D		LD A,0D	CRLF
	FAAD CDDEFA		CALL LPRINT	
	FAB0 C9		RET	
	FAB1 3E0F	INIT :	LD A,0F	Port A = Ausgang
	FAB3 D35F		OUT (5F),A	
	FAB5 3ECF		LD A,CF	Port B = Bitmode
	FAB7 D37F		OUT (7F),A	
	FAB9 3E01		LD A,01	Port B Bit 0 = Eingang
	FABB D37F		OUT (7F),A	
	FABD 3E67		LD A,67	Port B kein Interrupt
100	FABF D37F		OUT (7F),A	
	FAC1 3E80		LD A,80	Strobe = 1
	FAC3 D33F		OUT (3F),A	
	FAC5 2A4F5C		LD HL,(5C4F)	Pointer umstellen
	FAC8 010F00		LD BC,000F	
	FACB 09		ADD HL,BC	
	FACC 01DEFA		LD BC,LPRINT	
	FACF 71		LD (HL),C	
	FAD0 23		INC HL	
	FAD1 70		LD (HL),B	
110	FAD2 2100FA		LD HL,FA00	JP INIT loeschen
	FAD5 3600		LD (HL),00	
	FAD7 23		INC HL	
	FAD8 3600		LD (HL),00	
	FADA 23		INC HL	
	FADB 3600		LD (HL),00	
	FADD C9		RET	
	FADE FE0D	LPRINT:	CP OD	CR ?
	FAE0 2809		JR Z,+9>CRLF	

Listing. Ansteuerung des Matrix-druckers CP 80 (Fortsetzung)

FAE2 FE80	CP	80	Token ?
FAE4 380A	JR	C, +10>PRINT	
FAE6 D6A5	SUB	A5	
FAE8 C3100C	JP	0C10	im ROM weiter
FAEB CDF0FA	CRLF :	CALL PRINT	CR
FAEE 3E0A	LD	A, 0A	LF
FAF0 F5	PRINT :	PUSH AF	Zeichen retten
FAF1 F3	DI		Interrupt sperren
FAF2 3E7F	BUSY :	LD A, 7F	Break-Abfrage
FAF4 DBFE	IN	A, (FE)	
FAF6 1F	RRA		
FAF7 3011	JR	NC, +17>BREAK	Listing. Ansteuerung des Matrixdruckers CP 80 (Schluß)
FAF9 DB3F	IN	A, (3F)	Busy ?
FAFB 1F	RRA		
FAFC 38F4	JR	C, -12>BUSY	
FAFE F1	POP	AF	Zeichen zurueck
FAFF D31F	OUT	(1F), A	Ausgeben an Drucker
FB01 AF	XOR	A	Akku Nullsetzen
FB02 D33F	OUT	(3F), A	Strobe = 0
FB04 3E80	LD	A, 80	
FB06 D33F	OUT	(3F), A	Strobe = 1
FB08 FB	EI		Interrupt freigeben
FB09 C9	RET		
FB0A FB	BREAK :	EI	Interrupt freigeben
FB0B CF	RST	0B	Fehlerroutine
FB0C 0C	INC	C	Break.....
FB0D 00	NOP		Programm-Ende

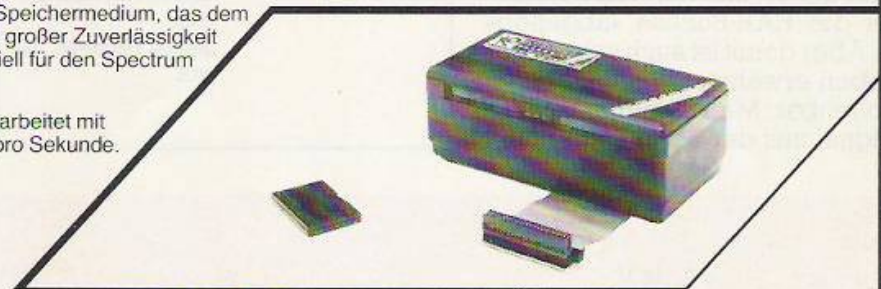
## Weltneuheit Wafadrive für Spektrum

Der Turbo-Dual-Drive ist ein hochentwickeltes Speichermedium, das dem Anwender einen schnellen Datenzugriff und bei großer Zuverlässigkeit ein günstiges Pheripheriegerät bietet, das speziell für den Spectrum konzipiert wurde.

Das System basiert auf Endlosband-Basis und arbeitet mit einer Datenübertragungsrate von 19200 Baud pro Sekunde.

### Spezifikationen

- Speicherkapazität über 240 KB (formatiert)
- Centronics-Schnittstelle
- RS-232 Schnittstelle
- Spectrum Erweiterungs Bus
- Betriebssystem mit folgendem Befehlssatz:  
NEW, FORMAT, CATALOG, SAVE, LOAD, ERASE, OPEN, PRINT, INPUT, CLOSE, CLEAR, CLS, MERGE, MOVE und spezielle Befehle zur Programmierung der RS-232 (z. B. Baudrate von 110 - 19200 B).
- sämtliche Copy- und Backup Kommandos möglich
- Maschinsprache LOAD & SAVE
- völlig unabhängiger I/O
- zusätzlicher 4 KB RAM-Speicher
- benutzt lediglich eine Adresse auf dem Bus



für  
Spectrum  
DM  
**499,-**

Preis versteht sich incl. 14% Mehrwertsteuer.  
Weitere Informationen direkt von uns gegen DM 1.30 Rückporto.

Händleranfragen auch aus dem Ausland erwünscht  
sowie Handelsvertretung gesucht.

Generalimporteur  
Vertrieb  
Service

**ncs**

**Nettetal Computer Systeme GmbH**

Klemensstraße 7 · D 4054 Nettetal 2  
Kaldenkirchen · Telefon 02157/1067  
Telex 17 215 732 · Teletex 215 732 = NCS

Für die Schweiz:  
**CN Valor AG** · Gurzelngasse 19  
CH-4500 Solothurn · Tel. (065) 23 41 44

# — MINI-PORT —

## Schalten und Walten über die »MIC«- und »EAR«-Buchsen

**E**in Schaltinterface mit so wenig Bauteilen? Nun ja, ein paar Einschränkungen muß man da schon in Kauf nehmen. Aber immerhin kann mit Basic ein Relais geschaltet und mit einem Taster eine Eingabe gemacht werden. Und ein paar Vorzüge hat die Schaltung auch. Sie läßt sich leicht nachbauen, ist preiswert und wird nur an die EAR- und MIC-Buchse angeschlossen. Damit läuft man nicht Gefahr, den Spectrum zu beschädigen und kann zur Speisung sogar eine Batterie verwenden.

### Schalten mit BEEP und Piep

Schnittstellen müssen adressiert werden, man muß dem Computer sagen, wohin er die Informationen schicken soll. Ein Interface muß dann auf »seine« Adresse ansprechen und die Informationen aufnehmen. Dieses Decodieren der Adresse kann man sich sparen, wenn man die Tonbuchsen verwendet. Der Spectrum weiß, daß der BEEP auf den Lautsprecher und die MIC-Buchse gehört und auf welchem Datenbit die EAR-Buchse »abgehört« wird. Aber damit ist auch schon eine der oben erwähnten Einschränkungen offenbar: Man bekommt nur ein Tonsignal aus der MIC-Buchse und

muß in die EAR-Buchse ein Tonsignal einspeisen. Das ist aber technisch kein Problem. Was als Handicap bleibt, ist die Tatsache, daß nur eine Ein/Aus-Funktion für jede Richtung zu erreichen ist, denn Tonfrequenzfilter wären zu aufwendig; da kann man lieber ein »richtiges« Schaltinterface bauen.

### Die Schaltstufe

Um einen breiten Betriebsspannungsbereich bei niedrigem Stromverbrauch zu erreichen, wird ein 4fach-CMOS-Gatter verwendet. Mit N1 wird das BEEP-Signal aus der MIC-Buchse verstärkt. Die Impulse

entladen den Kondensator C1 über den Widerstand R3 und die Diode D1. Gatter N2 arbeitet als Impulsformer und liefert das Schaltsignal für den Transistor T1. Die Diode D2 schützt den Transistor vor Spannungsspitzen, die beim Abschalten des Relais entstehen.

Die Dimensionierung der Bauteile ist nicht so kritisch. Gleiche Werte müssen aber R1 und R2 haben und R5 sollte nicht kleiner sein. Wenn das Relais mehr als 100 mA benötigt, muß für T1 ein anderer Transistor verwendet werden.

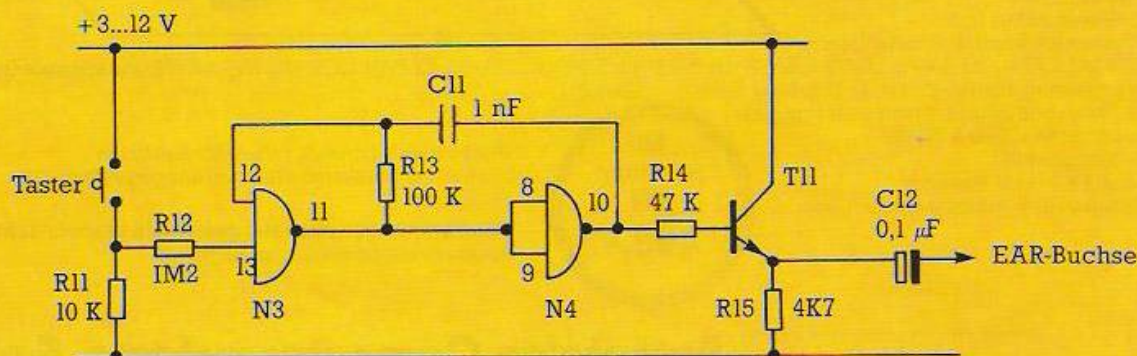
Die Betriebsspannung kann im Bereich von 3 bis zirka 12 V liegen, der Ruhestrom beträgt bei 5 V weniger als 0,2 mA. Das IC sollte einen Sockel erhalten und erst nach den Lötarbeiten eingesetzt werden. Für die Leitungen zu den Tonbuchsen ist abgeschirmtes Kabel zu verwenden.

### Der »Pieper« für die Eingabe

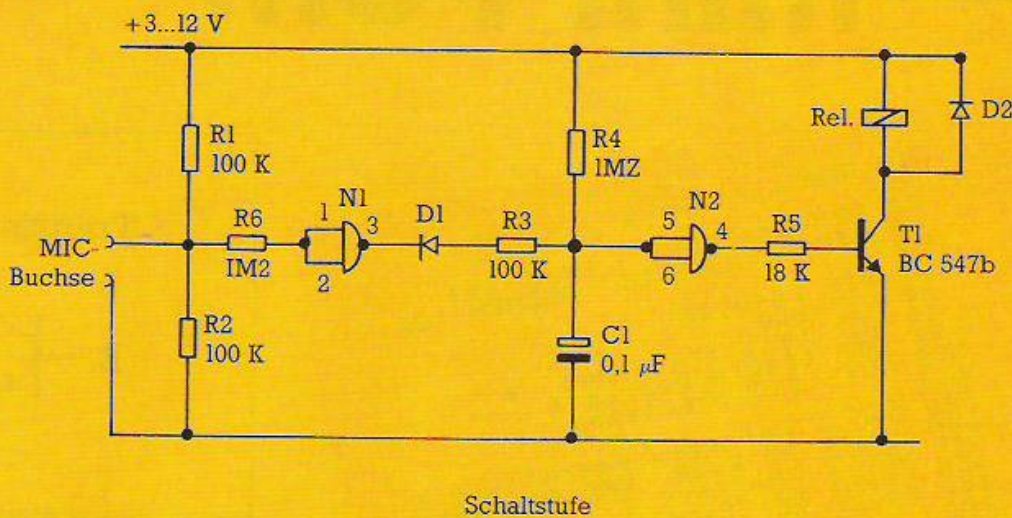
Über die EAR-Buchse ist der Spectrum ja nur für Tonfrequenzen zugänglich. Ein Rechteckgenerator ist mit den Gattern N3 und N4 aufgebaut, er schwingt, solange der Taster gedrückt ist. Der Transistor arbeitet als Impedanzwandler.

#### Stückliste

4 K7 = R15	1N4148 = D1, D2
10 K = R11	BC547b = T1, T11
18 K = R5	0,1 µF
	Tantal = C1, C12
47 K = R14	1 µF = C11
100 K = R1, R2, R3, R13	CD4010 = N1...N4
1 MZ = R4, R6, R12	
	Je Sockel, Taster, Klinkenstecker 3,5 mm
	Relais maximal 100 mA



Pieper (Tongenerator)



Auch hier müssen die Bauteilwerte nicht genau eingehalten werden, denn die Frequenz ist für die Funktion nicht so entscheidend.

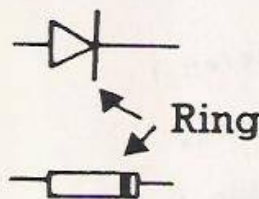
## Eingabe und Ausgabe

Die Programmierung aus dem Basic erfolgt für die Schaltstufe mit BEEP.

BEEP Zeitdauer, Tonhöhe (Handbuch Seite 137)

Diese Schaltung arbeitet ab einer Tonlänge von 0,02 Sekunden, und

## Diode



**1 N4148  
oder ähnlich**

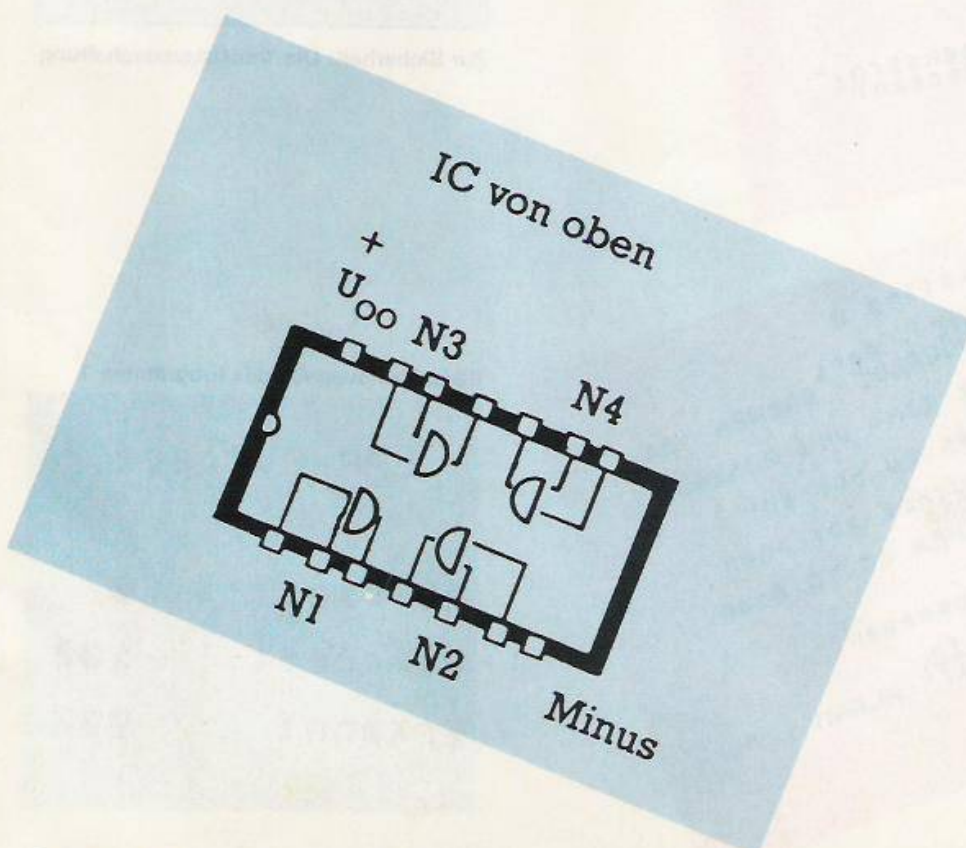
die niedrigste Frequenz liegt bei zirka -40. Das Relais schaltet für die Dauer des BEEP.

Die Abfrage der EAR-Buchse erfolgt bei jeder Tastaturabfrage. Mit »IN 254« erhält man ohne Ton den Wert 255, mit Ton den Wert 191. Die EAR-Buchse liegt ja auf dem Datenbit D6 mit dem Bitwert 64 (255-64=191). Weitere IN-Adressen im Handbuch auf Seite 160. Ein Tonsignal ist eine schnelle Folge von Impulsen. Um sicher zu sein, daß der Spectrum nicht gerade zwischen zwei Impulsen die EAR-Buchse prüft, sollte die IN-Abfrage in einer Schleife laufen.

## Software zum Testen

In Listing 1 kann die Abfrageschleife durch eine Eingabe dimensioniert werden. Wenn ein Tonsignal anliegt, wird die EAR-Buchse 100mal mit dieser Schleife abgefragt und das Ergebnis angezeigt. Damit ist zu prüfen, welche durchschnittliche Erkennung mit einem bestimmten Schleifenwert zu erreichen ist. Bei einem Wert um sieben müßte man auf 100 Prozent kommen. Soll das Programm nur die EAR-Buchse überwachen, so kann die IN-Abfrage in einer Endlosschleife laufen. Dafür gibt Listing 2 ein Beispiel. Wird ein Ton erkannt, so wird auf die Ausgabe verzweigt. Es erscheint eine Meldung auf dem Bildschirm, und das Ausgaberelais »blinkt« im Takt von BEEP und PAUSE.

(J. Howaldt)



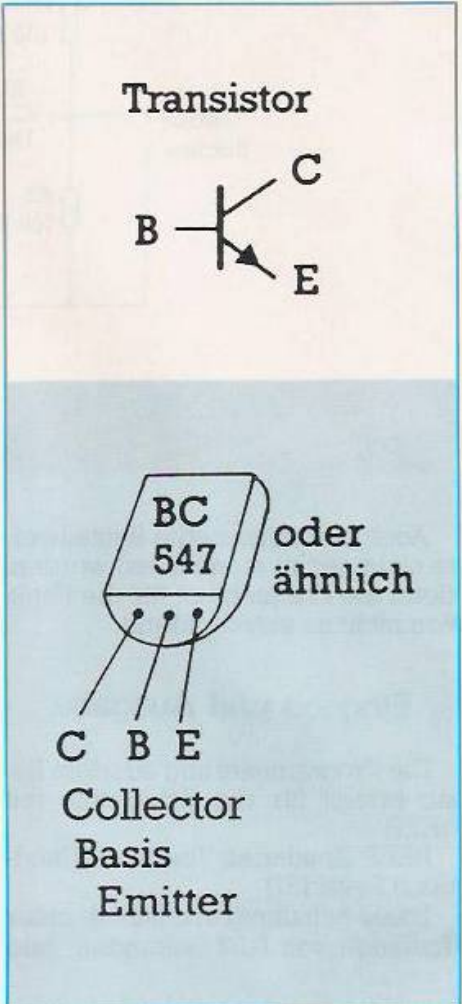
# MINI-PORT

```

100 REM Listing 1
110 REM *** MINI - Port ***
120 REM J.HOWALDT BREMEN '84
130 DIM t$(3,12)
140 LET e=0
150 LET t$(1)="Abfrage * "
160 LET t$(2)="geprueft : "
170 LET t$(3)="erkannt : "
180 REM & Abfragewiederholung &
190 PRINT AT 2,10;"EAR-Abfrage"
200 INPUT "Abfrage * ";X
210 PRINT #0;"Pieper druecken!"
220 PRINT AT 8,8;t$(1);X
230 REM # start wenn Piep #
240 IF a=IN 254
250 LET a=>191 254
260 LET a=255 THEN GO TO 0180
270 REM 100mal Pruefschleife
280 FOR Z=1 TO 100
290 LET piep=0
300 REM I und x-mal pruefen I
310 FOR P=1 TO X
320 LET a=IN 254
330 IF a=191 THEN LET piep=1
340 NEXT P
350 REM % Ergebnis ausgeben %
360 PRINT AT piep
370 PRINT AT 10,8;t$(2);Z
380 NEXT Z
390 REM Ende Pruefschleife
400 REM Wiederholung mit
410 INPUT "neuem Wert"
420 PRINT "..."
430 REM Abfrageschleife aendern;"
440 PRINT "FLASH 1;"
450 PRINT AT 10,12;
460 PRINT "ALARM I "
470 CLOSE 0 "(Taste druecken)"
480 RUN
    
```

```

100 REM Listing 2
110 REM MINI - Port
120 REM J.HOWALDT BREMEN '84
130 REM Test EIN- und AUSGABE
140 PRINT #0;"Kein Piep zu hoeren"
150 REM EAR-Buchse abfragen
160 LET a=IN 254
170 GO TO 90 THEN GO TO 0140
180 REM Alarm ausgeben
190 PRINT AT 10,12;
200 PRINT "FLASH 1;"
210 PRINT AT 10,10;
220 PRINT "ALARM I "
230 GO TO 100
    
```



Zur Sicherheit: Die Transistorbeschaltung

Bildschirmausgabe des Programms 1

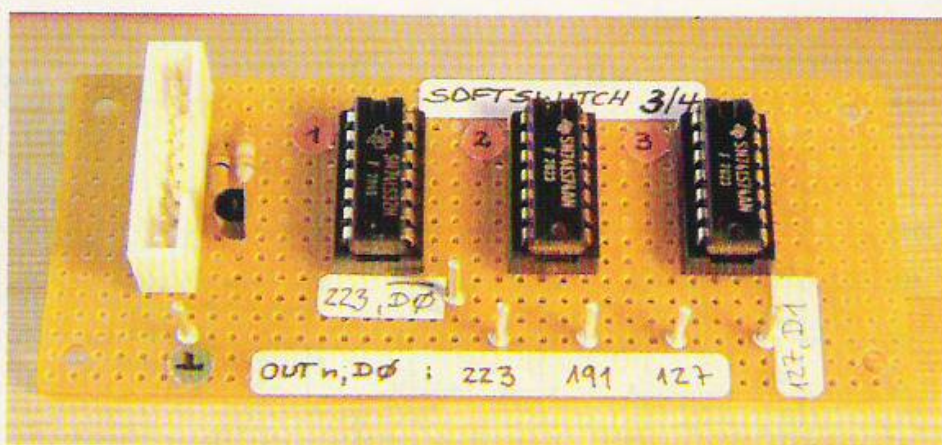
```

EAR-Abfrage

Abfrage * 5
geprueft : 100
erkannt : 99%
    
```

# Soft-Schalter

Eigentlich sollte der Softswitch 3/4 nur eine einzelne Schaltstufe werden. Gedacht für die RAM-Umschaltung bei einem 80-KByte-Spectrum (Happy-Computer 8/84, Seite 26). Aber dann



Muster-Aufbau der Schaltung

fiel ohne Mehraufwand noch eine zweite Stufe ab. Und wenn man die Möglichkeiten des Adreßdecoders ausnutzen will, dann läßt sich noch eine weitere Doppelschaltstufe ansteuern.

**S**oftswitch 3/4 bedeutet, daß über drei Adressen vier Schalter gesetzt werden können. Mit Schalter sind hier Flip-Flops gemeint, die den Wert eines Datenbits speichern.

Mit dem OUT-Befehl lassen sich die Schalter ansprechen, einzeln oder in beliebiger Kombination in einem Befehl, eine Eigenart der unvollständigen Decodierung.

## Die Schaltung

Die Decodierung mit dem ODER-Gatter IC1 liefert drei Freigabesignale. Die Flip-Flops in IC2 und IC3 übernehmen und speichern den Wert des anliegenden Datenbits bei Eintreffen eines dieser Freigabesignale. Die Ausgänge der Schalter sind in dem Schaltplan auch mit der entsprechenden OUT-Adresse bezeichnet. In Klammern gesetzt ist die Bezeichnung des Datenbits, das die EIN/AUS-Funktion bestimmt.

Der Transistor T1 arbeitet als Trei-

ber, um die Datenleitung D0 des Spectrum nicht mit drei Eingängen zu belasten.

## Die Anwendungsmöglichkeiten

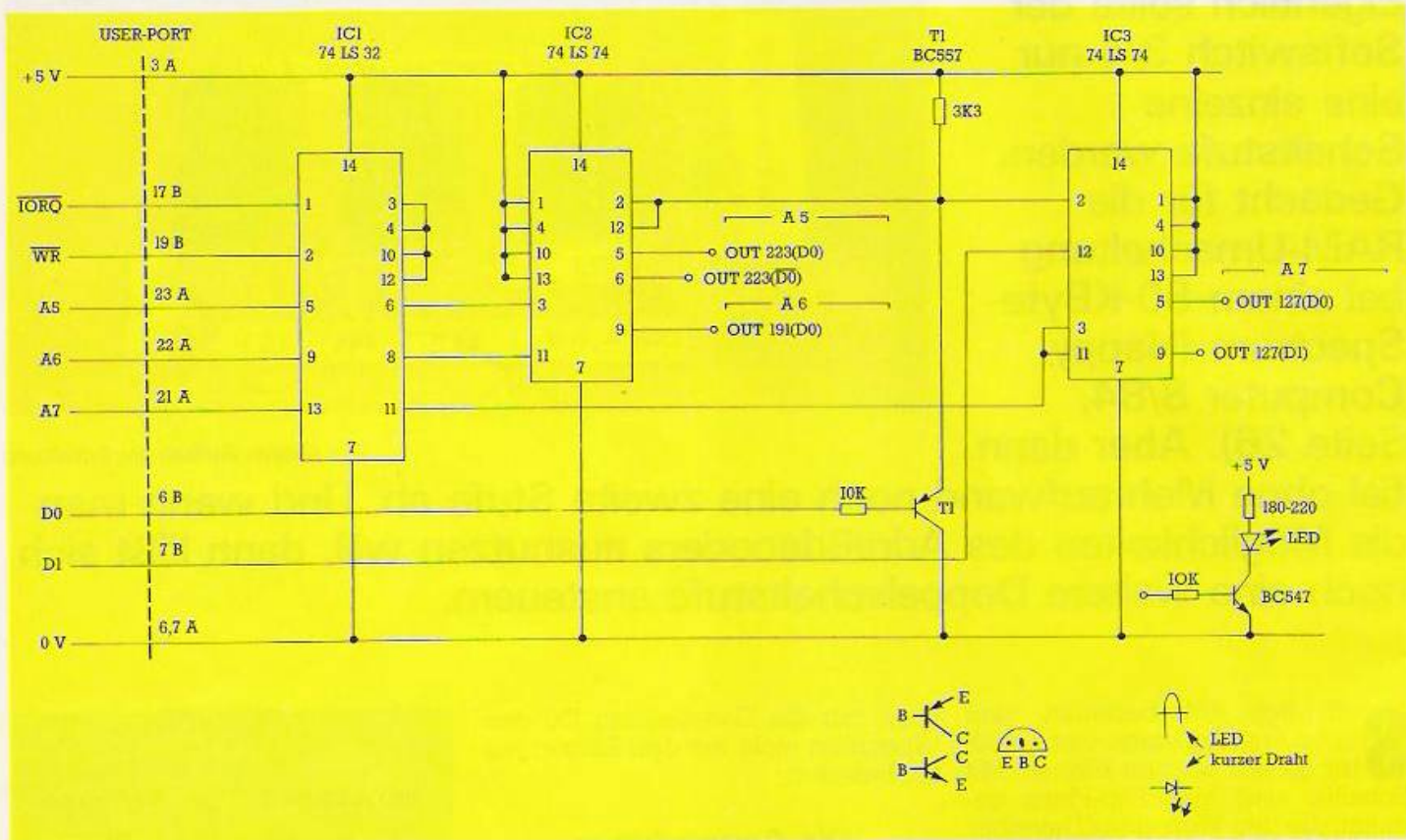
Für die schon zitierte RAM-Umschaltung muß die in Happy-Computer angegebene Beschaltung etwas geändert werden: Der Schalter wird ausgebaut. Von »gelb« nach »rot« wird ein Widerstand gelötet, Wert 3K9 bis 10K. Punkt »gelb« wird mit dem Ausgang OUT 223(D0) verbunden. Der Spectrum läuft jetzt auch ohne den Softswitch, weil durch den Widerstand ein Speicherbereich vorgewählt ist.

Mit einer Leuchtdiode läßt sich der gewählte Speicherbereich anzeigen. Größere Verbraucher schaltet man besser über ein kleines Relais, um dem Spannungsregler des Spectrum nicht so zu belasten. Sowohl Leuchtdiode als auch Relais müssen natürlich über einen

Adressbit			Adresse
7	6	5	OUT
		*	223
	*		191
*			127
	*	*	159
*	*		63
*		*	95
*	*	*	31

Adreß-Tabelle

# Soft-Schalter



Die Schaltung des »Softswitch«

```

1 REM Test RAM-Schalter
2:
10 PRINT AT 11,15;"0"
20 OUT 223,1: PAUSE 10
30 PRINT AT 11,15;"1"
40 OUT 223,0: PAUSE 10
50 GO TO 10

```

```
POKE USR "0",BIN 01000010
```

Listing »RAM-Schalter«

```

1 REM Blinker
2:
10 OUT 127,BIN 00: PAUSE 30
20 OUT 127,BIN 01: PAUSE 30
30 OUT 127,BIN 10: PAUSE 30
40 OUT 127,BIN 11: PAUSE 30
50 GO TO 10

```

Listing »Blinker«

zusätzlichen Transistor betrieben werden.

## Zur Software

Durch die getrennte Adressierung der Schalter ist die Programmierung sehr übersichtlich. Mit dem Befehl

»OUT Adresse,Wert«

werden die Flip-Flops gesetzt. Die Tabelle zeigt für die Schalter oder Schalterkombination die Adressen.

Die Schalter A5 und A6 werden mit dem Wert »0« und »1« aus- und eingeschaltet. Für A7 entsprechend mit »0« bis »1« und »0« bis »2«. In der binären Darstellung beim Listing »Blinker« wird die Zuordnung von

Datenbit und Schalterstellung deutlich. In Zeile 20 und 40 ist der Schalter mit dem Datenbit D0 auf »EIN«, in Zeile 30 und 40 der mit Datenbit D1; aber es ist natürlich einfacher, den Wert mit 0 bis 3 anzugeben.

Zum Testen der RAM-Umschaltung ist das zweite kleine Programm gedacht. Nachdem beide Bereiche initialisiert sind, kann das Listing eingegeben werden (das »0« im Grafikmodus). Dann wird in einem Bereich das UDG »O« zum »0« verwandelt. Was nun in der Schleife passiert, ist wohl klar: ein RAM-Bereich hat ein UDG »O«, der andere ein UDG »0«, die wechselweise auf dem Bildschirm erscheinen.

(Jürgen Howaldt)



# Ein Joystick-Interface

Haben Sie auch schon so viele »Leben« verloren, weil Sie sich bei einem rasanten Spiel in den Cursor-Tasten geirrt haben? Mit einem Joystick lassen sich die Bewegungsrichtungen viel leichter umsetzen — auch diagonale, mit denen man sich bei den Tasten besonders schwer tut.

Interface ist nicht gleich Interface. Die drei wichtigsten Verfahren der Joystick-Abfrage sind:

1. Sinclair-Norm: Geeignet für zwei Joysticks. Entspricht den Tasten-Halbreihen 1 bis 5 und 6 bis 0.

2. Sinclair-Norm: Geeignet für zwei Joysticks. Entspricht den Tasten-Halbreihen 1 bis 5 und 6 bis 0.

3. Kempston-Norm: Hier wird nicht mit der Tastaturabfrage gearbeitet, sondern mit den User-Adressen A5 bis A7. Das ergibt dann die Adressierung mit IN 31.

Wenn von Norm gesprochen wird, so ist das eigentlich nicht richtig. Eine Norm für die Joystick-Abfrage beim Spectrum gibt es nicht. Deshalb müssen die Software-Hersteller in einem Menü abfragen, welches Interface der User sein eigen nennt, und die Hardware-Hersteller lassen sich auch immer raffiniertere Geräte einfallen. Das geht hin bis zur freien Programmierbarkeit. Aber ist das wirklich bedienerfreundlich?

Doch zurück zum Bauvorschlag. Die folgende Schaltung ist »Kempston«-kompatibel, arbeitet also mit der Adresse IN 31. Näheres dazu im Softwareteil.

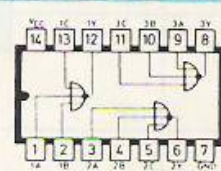
## Die Schaltung für Joystick und Trackball

Um den Aufwand gering zu halten, wird eine unvollständige Decodierung verwendet. Das heißt, mit IORQ und RD wird nur das Adreßbit A5 decodiert. Im praktischen Betrieb hat das keine Nachteile und ist allgemein üblich. Die Umsetzung der Joystick-Schaltbefehle auf den Datenbus wird mit dem invertierenden Bustreiber 74LS240 vorgenommen. Damit wird in Ruhestellung der Datenwert »0« erreicht und das Interface ist auch für den Trackball ge-

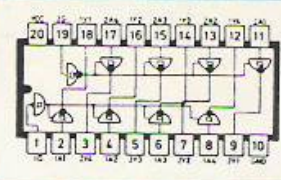
eignet. Die Anschlußbelegung für den Joystick entspricht der von Atari und Commodore. Zusätzlich auf der Buchse liegen noch Datenbit D5 und D6. Wenn der Eingang für andere Schaltaufgaben verwendet werden soll, stehen hier sieben Datenbits zur Verfügung. Falls Sie auch das achte Bit brauchen — es liegt an Pin 11 des LS240.

## Software zur Joystick-Abfrage

Für Hardware-Eingaben hat der Spectrum den Befehl »IN ADRESSE«. Damit läßt sich nicht nur die Tastatur abfragen, sondern auch ein Joystick-Interface. Mit IN 31 erhält man, bei angeschlossenem Interface, den Wert »0«. Die Werte für die verschiedenen Richtungen sind in



IC 74 LS 27



IC 74 LS 240

Die verwendeten IC's

der Tabelle aufgeführt. Bei gedrückter Feuertaste erhöhen sich die Werte um 16. Das Testprogramm zur Joystick-Abfrage gibt ein Beispiel, wie man die Werte verarbeiten kann. Mit der Zuweisung in Zeile 210 erhält man einen Augenblickswert, mit dem ab Zeile 290 gerechnet wird. Zuerst wird die Feuertaste geprüft. Sie liegt auf dem Datenbit D4 mit dem Bitwert 16. Ist der Wert größer oder gleich 16, so wird statt des Leerstrings das Wort »Taste« ausgegeben und die Variable »a« um 16 verringert. Das wiederholt sich dann entsprechend mit Bit D3 bis D0 mit den Bitwerten 8,4,2,1.

(Jürgen Howaldt)

```
JOY 31 - Test
IN 31 - Wert : 5
```

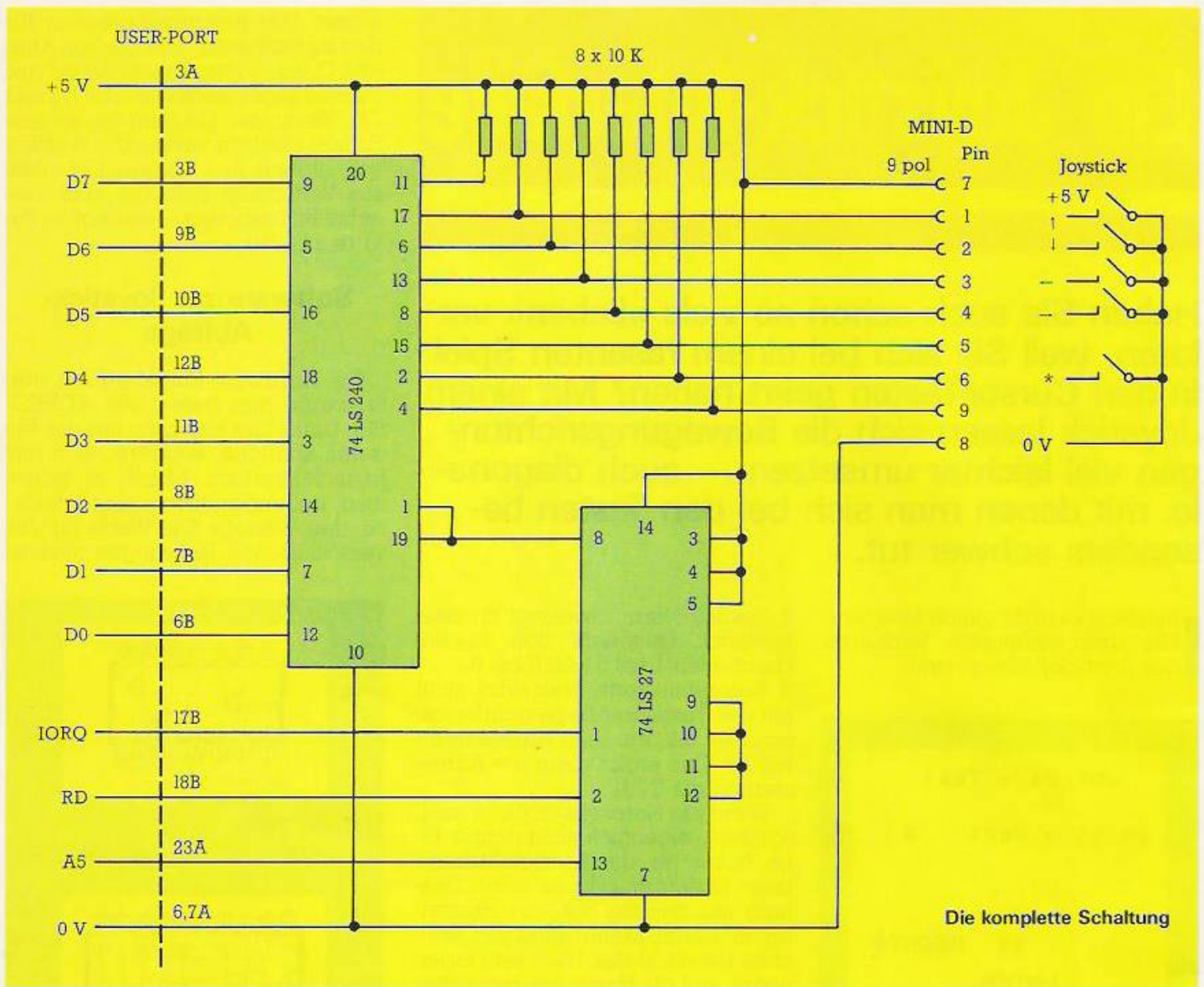
```
** RECHTS
UNTEN
```

```
Joystick-Werte
```

```
IN 31
  oben
  |
10  8  9
 |   |   |
L   2  0  1   R
i   |   |   |
n   |   |   |
k   |   |   |
s   |   |   |
  6  4  5
  |
  unten
```

Hardcopy der »Abfrage«

1. Cursor-Tasten-Simulation: Man fragt die Tasten ab, die schon eine Kennzeichnung für Richtungen haben. Der Nachteil für die Hardware: Es müssen zwei Halbreihen decodiert werden (hätte Sinclair doch die Cursor-Tasten bei der »6« beginnen lassen!).



Die komplette Schaltung

```

10 REM Test Joystick-Abfrage
20:
30 REM          JOY 31
40:
50 REM J.HOWALDT  BREMEN '84
60:
70 DIM b$(5,5)
80 LET b$(1)="RECHTS"
90 LET b$(2)="LINKS"
100 LET b$(3)="UNTEN"
110 LET b$(4)="OBEN"
120 LET b$(5)="TASTE"
130 LET c#=""
140:
150 REM Bild
160 PRINT AT 7,9;
    "JOY 31 - Test"
170 PRINT AT 10,7;
    "IN 31 - Wert: "
180 PRINT AT 16,16; "*"
190:
200 REM Wert Joystick pruefen
210 LET a=IN 31
220 PRINT AT 10,22; " "
230 PRINT AT 10,22; a
240 IF a>=32 THEN PRINT #0;

```

```

    AT 0,0;"Bitwert >= 32"
    GO TO 0200
0200: INPUT ""
0300: REM Richtung berechnen
0400 LET d$=c$
0500 IF a>=16 THEN LET a=a-16
    LET d$=b$(5)
0600 PRINT AT 20,13;d$
0700 LET d$=c$
0800 IF a>=8 THEN LET a=a-8
    LET d$=b$(4)
0900 PRINT AT 14,14;d$
1000 LET d$=c$
1100 IF a>=4 THEN LET a=a-4
    LET d$=b$(3)
1200 PRINT AT 18,13;d$
1300 LET d$=c$
1400 IF a>=2 THEN LET a=a-2
    LET d$=b$(2)
1500 PRINT AT 16,8;d$
1600 LET d$=c$
1700 IF a>0 THEN LET d$=b$(1)
1800 PRINT AT 16,19;d$
1900 GO TO 0200: REM Abfrage
2000:
9999 SAVE "JOY 31" LINE 0010

```

# bremse

Das Schaltbild zeigt, welche Kontakte benötigt werden. Vorweg noch einige Erläuterungen zur Funktion.

**RESET-Steuerung:** Über den Eingang RESET wird die CPU auf den Anfangszustand und der Programmzähler auf Null gesetzt. Um diesen definierten Zustand auch beim Einschalten der Betriebsspannung zu erreichen, wird ein Kondensator am RESET-Eingang über einen Widerstand aufgeladen und erzeugt damit den Power-On-Reset. Diesen Kondensator sollte man nicht ohne Vorwiderstand entladen (der Wert ist nicht so kritisch).

**Interruptsteuerung:** Die CPU hat zwei Steuereingänge (INT, NMI), die eine Unterbrechung der Programmbearbeitung bewirken. Über diese Leitungen können externe Geräte die Unterstützung der CPU anfordern. Mit einem einfachen Taster läßt sich natürlich keine ordentliche Interruptsteuerung realisieren, weil man doch etwas mehr

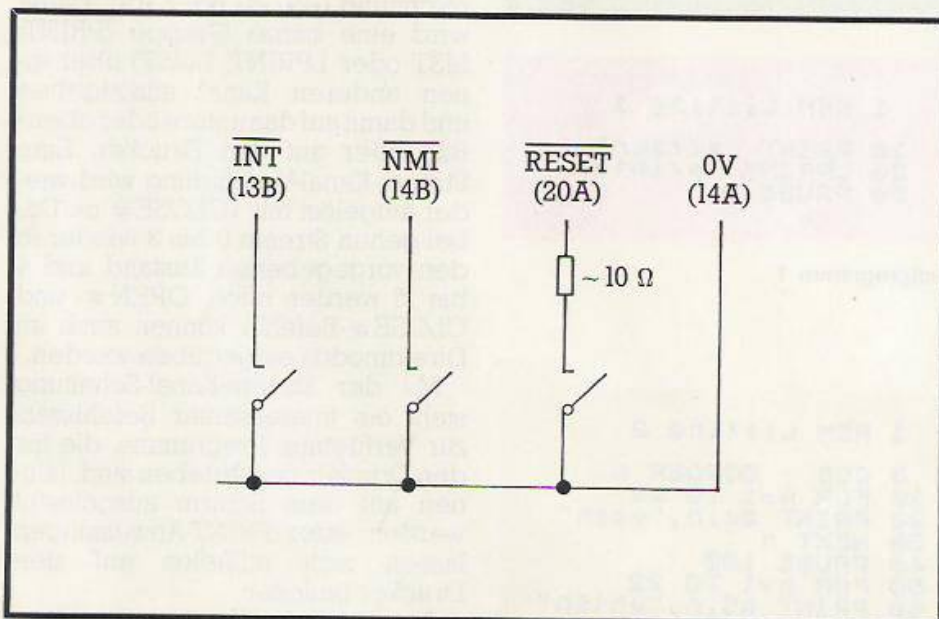
Hardware braucht, und die Software durch das Betriebssystem vorgegeben ist beziehungsweise durch das laufende Programm. Aber der geringe Aufwand rechtfertigt vielleicht einen Versuch.

Die **maskierbare Unterbrechung** (INT) läßt sich durch Software unterdrücken beziehungsweise beeinflussen; damit ist die Wirkung dieses Tasters eingeschränkt. Eine Methode der Maskierung führt auf die Speicheradresse 0038h. Da von dort die Tastaturabfrage aufgerufen wird, kann man sich eventuell mal aus einer Endlosschleife retten, ohne das Programm zu löschen. Ein Basic-Programm stoppt für die Zeit des Tastendrucks.

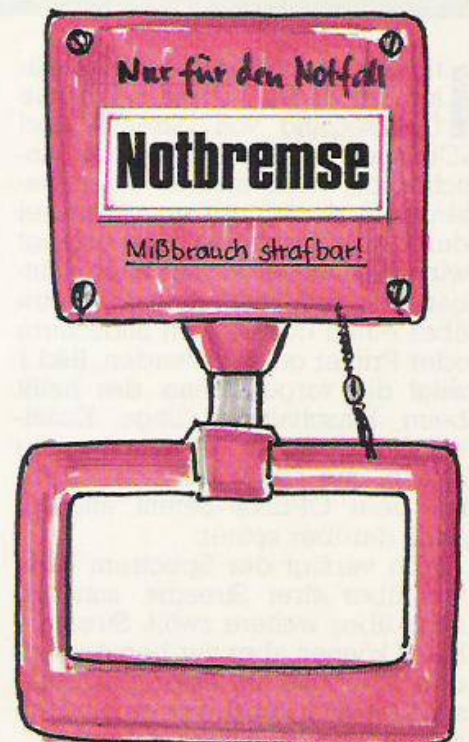
Die **nichtmaskierbare Unterbrechung** (NMI) bewirkt einen Aufruf der Speicherstelle 0066h und damit einen Kaltstart des Systems. Man braucht also nur den RESET- oder den NMI-Taster.

(Jürgen Howaldt)

Will man einen Kaltstart nicht über **USR 0** machen, oder hat sich ein Maschinenprogramm in einer Schleife gefangen, so greift man zum **9-V-Stecker**, um durch eine Unterbrechung der Versorgungsspannung einen **Power-On-Reset** zu erzwingen. Weder eine elegante noch eine material-schonende Prozedur. Wenn ein **Portstecker** vorhanden ist, so läßt sich mit zwei Tastern schnell **Abhilfe** schaffen.



Drei Tast-Schalter und ein Widerstand, mehr wird nicht benötigt. Die Anschlußbelegung ist dem Spectrum-Handbuch zu entnehmen. Die Taster sollten so angebracht werden, daß sie nicht versehentlich betätigt werden können.



# OPEN # und Close # —

## auch ohne Microdrive nutzbar

Zu den wenig dokumentierten Befehlen des Spectrum zählen »OPEN #« und »CLOSE #«. Das sonst so ausführliche Handbuch verweist auf die Verwendung mit dem Microdrive. Natürlich macht erst der Einsatz des Interface 1 mit seiner zusätzlichen Software diese Befehle voll einsatzfähig, aber was man schon in Verbindung mit einem ZX-Drucker damit anfangen kann, ist interessant. Und wer keinen Drucker hat, kommt auch nicht zu kurz, denn einige Möglichkeiten kann man auch auf dem Bildschirm nutzen.

Im Titel ist von Strömen und Kanälen zu lesen. Diese wörtliche Übersetzung von »Stream« und »Channel« ist vielleicht ganz anschaulich, wenn man an einen Datenstrom denkt, der zum Beispiel durch eine Tastatureingabe erzeugt wird. Weil dieser »Strom« auch sichtbar sein soll, muß dieser »Strom« über einen »Kanal« zum Bildschirm oder Printer gelenkt werden. Bild 1 zeigt die vorgegebene, das heißt beim Einschalten gültige Kanalschaltung. Diese Beziehung von Stream und Kanal läßt sich jedoch mit dem OPEN #-Befehl ändern, doch darüber später.

Nun verfügt der Spectrum nicht nur über drei Streams, sondern noch über weitere zwölf. Stream 4 bis 15 können aber nur benutzt werden, wenn sie auf einen Kanal geschaltet sind, sonst gibt es die Fehlermeldung »0 "Invalid stream"«.

```
1 REM Listing 1
10 PRINT "screen"
20 LPRINT "printer"
30 PAUSE 0
```

Testprogramm 1

```
1 REM Listing 2
5 CLS : BORDER 6
10 FOR n=1 TO 22
20 PRINT #4;n,"oben"
30 NEXT n
40 PAUSE 100
50 FOR n=1 TO 22
60 PRINT #5;n,"unten"
70 NEXT n
80 PAUSE 0
```

Testprogramm 2

Das Zeichen für Stream ist der Hash-Charakter »#«. Mit »PRINT #n« gibt man also eine PRINT-Anweisung über den Stream »n« aus.

Drei Kanäle stehen für die Ausgabe zur Verfügung; sie sind mit einem Buchstaben gekennzeichnet:

»k« unteres Bild (keyboard)

»s« oberes Bild (screen)

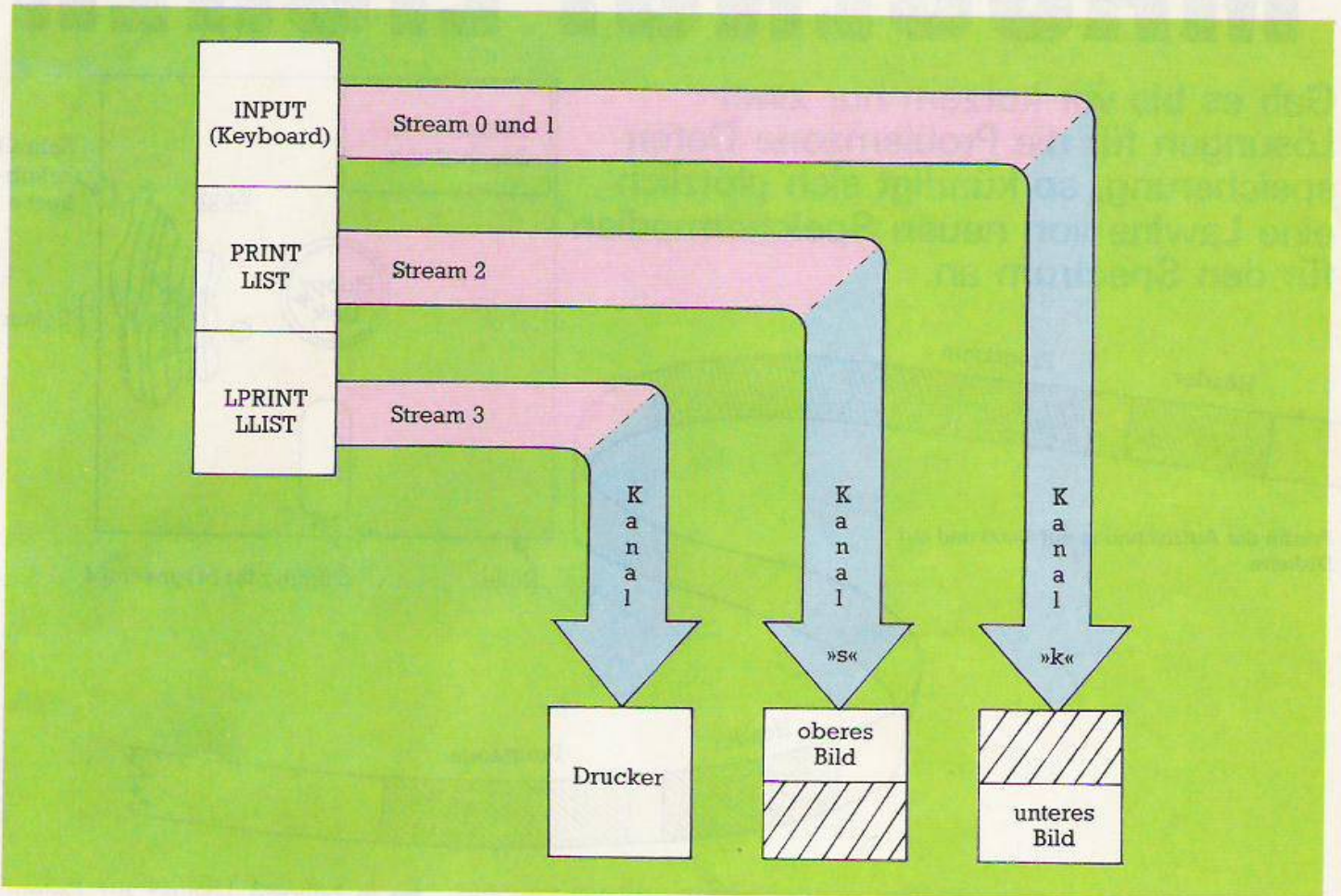
»p« Drucker (printer)

Übrigens, der Versuch, eine INPUT-Anweisung über Kanal »s« oder »p« auszugeben, wird mit der Meldung »J "Invalid I/O device"« scheitern.

Nun kann man nicht nur eine PRINT- oder LPRINT-Anweisung über einen anderen Stream ausgeben, sondern auch den Stream auf einen anderen Kanal schalten. Das geschieht in der Form: »OPEN # n,c\$«. Dabei ist »n« eine Zahl zwischen 0 und 15 und »c\$« die Kanalbezeichnung (»k«, »s« oder »p«). Damit wird eine ganze Gruppe (PRINT, LIST oder LPRINT, LLIST) über einen anderen Kanal ausgegeben und damit auf das untere oder obere Bild oder auf den Drucker. Eine Stream-Kanal-Verbindung wird wieder aufgelöst mit: »CLOSE # n«. Dabei gehen Stream 0 bis 3 wieder in den vorgegebenen Zustand und 4 bis 15 werden offen. OPEN #- und CLOSE #-Befehle können auch im Direktmodus eingegeben werden.

Mit der Stream-Kanal-Schaltung steht ein interessanter Befehlssatz zur Verfügung. Programme, die für den Drucker geschrieben sind, können auf dem Schirm ausgetestet werden, oder PRINT-Anweisungen lassen sich mühelos auf den Drucker umleiten.

Aus den vielen Kombinationsmöglichkeiten, eine Anweisung über einen anderen Stream auszugeben oder einen Stream auf einen anderen Kanal zu schalten, hier einige



▲ Bild 1. Normalfluß

▼ Bild 2. Flußmatrix

Anregungen für eigene Versuche. Vergleichen Sie dazu das Schema in Bild 2.

Aufgabe: Eine PRINT-Anweisung soll auf Zeile 24 erscheinen.

Lösung: PRINT # 0; 'Hallo' :Pause 0

Weil Stream 0 mit Kanal »k« verbunden ist, geht »Hallo« auf das untere Bild. Mit #3 wird die Begrüßung auf dem Drucker ausgegeben. Die PAUSE 0 verhindert nur ein Überschreiben durch die »OK«-Meldung.

Aufgabe: Die LPRINT-Anweisung aus Listing 1 soll auf Zeile 24 erscheinen.

Lösung: OPEN # 3; 'k'

Die Gruppe LPRINT,LLIST liegt auf Stream 3, der mit dem OPEN # -Befehl auf Kanal »k« geschaltet wird.

Aufgabe: Es wird ein Programm für die Ausgabe auf den Drucker geschrieben; es soll auf dem Bildschirm ausgetestet werden.

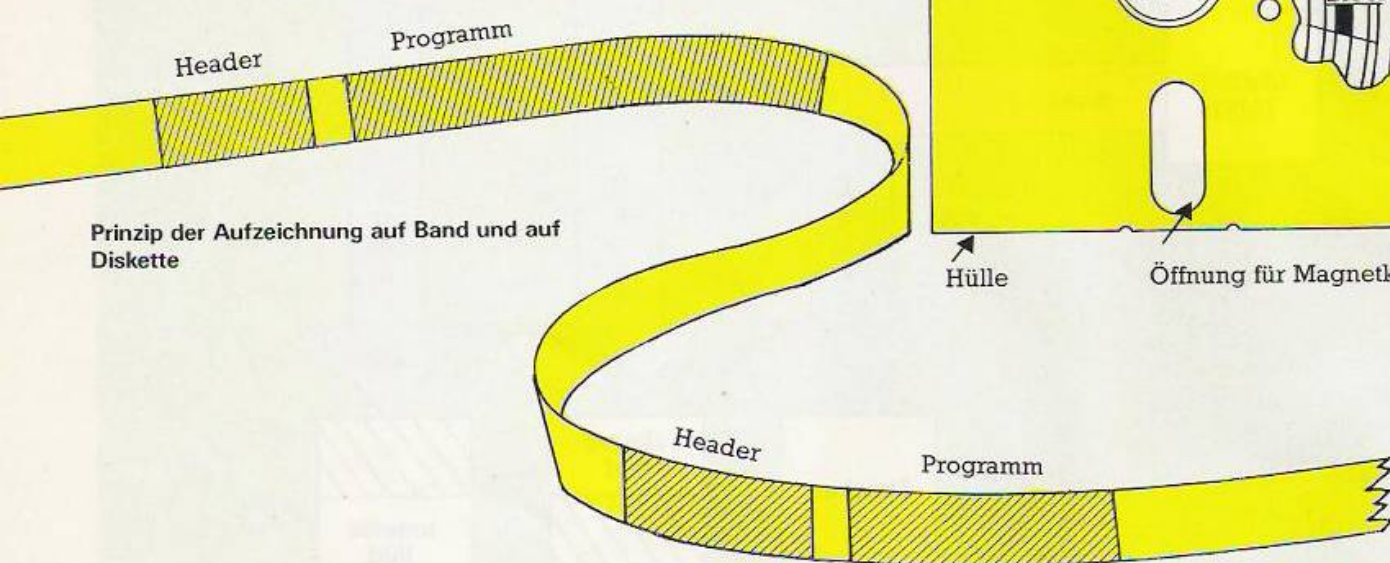
Lösung: Statt LPRINT wird PRINT verwendet; dann kann mit OPEN # 2; 'p' die Gruppe PRINT,LIST auf den Druckerkanal geschaltet werden. (Jürgen Howaldt)

Stream	Kanal			Befehls-Gruppe
	p	s	k	
# 0/1	X	X	*	INPUT, (Tastatur)
# 2		*		PRINT, LIST
# 3	*			LPRINT, LLIST
# 4				frei
⋮				
⋮				
# 15				
	Drucker	oberes Bild	unteres Bild	

\*) beim Einschalten vorgegebene Stream-Kanal-Schaltung

# MASSENSPEICHER

Gab es bis vor kurzem nur zwei Lösungen für die Problemzone Datenspeicherung, so kündigt sich plötzlich eine Lawine von neuen Speichermedien für den Spectrum an.



Die billigste Lösung zur Datenspeicherung war und ist der Kassettenrecorder für Audio-Cassetten, für den es preiswerte Datenkassetten mit kurzen Bändern gibt. Dem Preisvorteil steht die geringe Lade- und Lesegeschwindigkeit gegenüber. Und das wiederum liegt erstens an der Bandgeschwindigkeit, die von den Audiorecordern vorgegeben ist, und zweitens daran, daß die Daten nur in der Reihenfolge gelesen werden können, in der sie aufgezeichnet wurden. Wenn bestimmte Daten gesucht werden, muß erst das ganze Band danach abgesucht werden. Sie kennen das Problem sicherlich von Ihren Musikkassetten. Das gewünschte Musikstück ist mit Sicherheit nicht am Kassettenanfang. Entweder man läßt nun das Band komplett durchlaufen oder sucht mit Hilfe des Zählwerkes (wenn vorhanden) die richtige Stelle durch schnelles Vor- und Zurückspulen. Geeignete »Datenrecorder« gibt es ab etwa 100 Mark im Handel, sehr gute Ergebnisse erzielte ich mit dem abgebildeten Recorder von Schumpich und dem Sony TCM 3. Der Schumpich-Recorder

# IN MASSEN

der bietet als Zugabe eine Mithörmöglichkeit bei Aufnahme und Wiedergabe und arbeitet auch mit dem wählerischen ZX81 gut zusammen. Recorder ohne Kopfhörer- oder Lautsprecherbuchse sind nur bedingt geeignet, von Stereorecordern rate ich ab (es sei denn, sie lassen sich auf Mono umschalten).

## Es geht schneller

Schon auf der letzten Computer-Fair in London fiel mir ein sogenannter »Sprint«-Recorder auf. Dieses Gerät arbeitet gegenüber dem herkömmlichen Recorder mit der vierfachen Geschwindigkeit und ist voll kompatibel mit allen Software-Kassetten. Der von Challenge Research hergestellte und von Stephan Triebner importierte Sprint-



Ein Datenrecorder für Kompakt-Kassetten

## Keyboard, Interface, Joysticks, Trackball, Lightpen, Software

### Spectrum

**KEMPSTON Centronics Interface**  
TYP E mit Steuersoftware in ROM.  
LLIST, LPRINT und COPY (auch Supercopy mit 4facher Vergrößerung)  
DM 219,90

**KEMPSTON Centronics Interface**  
TYP S. Funktionen wie oben, jedoch Software auf Cassette.  
DM 159,90

**Spectrum KEYBOARD NACH INDUSTRIESTANDARD**  
in Metallgehäuse aus englischer Fertigung — 41 Tasten mit original Beschriftung — vergoldete Kontakte — zwei Shifttasten — große Leertaste  
— Microdrive Interface I ansteckbar — erstaunlicher Preis!  
RIKB2:DM 198,00

**Original Spectrum Kempston Joystick-Interface**  
DM 49,90

**Spectrum Interface für Trackball** (auch „Kempstonkompatibel“ für jeden Joystick) einfach anstecken  
DM 49,90



Für C-64, Spectrum usw.  
**Competition Pro-Joystick**  
nur DM 54,90

**Super Trackball** in schwerer Ausführung, sofort anschließbar an VC20, C64, Atari mit Interface auch an Spectrum.  
DM 79,90

**TELESOUND** brandneu: Spectrum-Ton über Fernsehlautsprecher. Ohne Löten einfach im Rechner anstecken; moduliert Tonsignal auf HF-Fernsehaustritt. Komplettpreis: DM 49,90

### Der Superrecorder SPRINT

Lädt und saved jedes Spectrumprogramm mit vierfacher Geschwindigkeit (z.B. 48K statt über 5 Minuten in nur 75 Sekunden) Bandgeschwindigkeit 18 cm sek. Voll kompatibel zu jedem normal aufgenommenen Programm. Einfach an Extension-Port anstecken. — keine externe Stromversorgung. — keine Überspielkabel. — Extension-Port am Recorder durchgeführt. — akzeptiert alle Tape Befehle (LOAD, SAVE, VERIFY). — digitale Aufzeichnungstechnik. — keine Aussteuer/Lautstärkeinstellung mehr. — große Datensicherheit! Der Preis- und Qualitätshit, wenn es darum geht, Daten schnell und kostengünstig zu speichern ohne das Aufzeichnungsformat zu ändern (kein Umstricken der Software nötig). Für 16-48K Rechner. Komplettlieferung DM 269,00



**LIGHTPEN** von D'kronics. Endlich können Sie bei Ihrem Spectrum direkt über den Bildschirm eingeben. Menüsteuerung mit Circle, Polygonzug Rechteck, Fill mit beliebiger Farbe, usw. Komplett: Lightpen & Interface & Software zu einem unglaublichen Preis: DM 89,90

### VISCOUNT DISCDRIVE und CONTROLLER (5 1/2")

Der absolute Star unter den Floppy-Systemen für den Spectrum, nicht nur was den Preis betrifft. — Controller einfach ansteckbar. — Alle Befehle im ROM (auch BACK für Kopien von einer Floppy auf eine andere!) — DOS belegt Speicherplatz oberhalb RAMTop, d.h. Umkopieren von Cassette auf Floppy außerst einfach. — formatiert ca 100K pro Floppy. — Directory mit 39 möglichen Fileinträgen. — große Datensicherheit. — Shughart-Laufwerk mit integriertem Netzteil!



Controller & Laufwerk & deutsches Handbuch  
**komplett DM 899,00**

### Software Spectrum:

3644 Rechen pro Zeile, ideal f. Textverarbeitung	SYS-64	DM 34,90
3 Pass-Assembler mit symbolischen Adressen	Assembler	DM 44,90
Diagnosemod, test jedes Maschinencodogramm	BUG	DM 34,90
Programmiersprache der Zukunft, ausführlich dokumentiert	FORTH	DM 79,00
DETELE, PENNUMBER, CLOCK, EXAMINE usw.		
jetzt für den Spectrum		
100 Maschinencodierungen für Ton, Bildschirm usw.	EXT-BASIC	DM 28,90
Datien sortieren, speichern, suchen	SUPERCODE	DM 39,90
Fast nicht über schlagbar, nachtr. 6-10 Zeile voraus	DATASAVE	DM 36,90
Das Originalspiel mit vier verschiedenen Bildschirmen	SUPERCHESS	DM 39,90
Wahrscheinlich das beste 3D-Spiel überhaupt	KONQ	DM 29,90
Das Originalspiel mit toller Grafik und Sound	3-D-COMBATZONE	DM 34,90
Bestehen Sie gegen die Trolch, Orcs, Sauris und Wargs in	PENGY	DM 29,90
Bingen Sie Proggz neu über die Straße und den Fuß	MORNA	DM 19,90
	JOGGER	DM 19,90
Oder fordern Sie unseren Info-Katalog (DM 3,-) an		
		1) Nur 48 K

### Software C-64:

Originalspiel: Sie flitzen auf dem Mond herum. Versuchen Sie, den Aliens zu entkommen, aber Vorsicht! Das Originalspiel mit vier verschiedenen Bildschirmen. Einmal der aufwendigsten Spiele für den C64/128. Das Originalspiel mit fantastischer Grafik. Sie sitzen am Cockpit und haben DAS 3-D-Spiel. Schon mit Joystick Musik komponiert? Ein Arcade-Spiel mit 500 verschiedenen Plänen. Das Originalspiel, das Sie bestimmt in Hektik bringt. Kleben Sie gern durch unterirdische Gänge? Aus Ihrem C64 und ein schwermögiger Synthesizer mit. Für alle Maschinencodierungen. Fantastisches Strategiespiel. Hier sind Sie. Voller 3-D-Grafik und SuperSound. Oder fordern Sie unseren Info-Katalog (DM 3,-) an

MOONBUGGY	DM 29,90
SKRAMBLE	DM 29,90
KONG	DM 34,90
MOTHERSHIP	DM 29,90
MEOCLYPE	DM 29,90
3-D-TIME TREK	DM 29,90
DANGING FEATS	DM 29,90
TRANSV. TOWER	DM 29,90
PANIC	DM 29,90
CHINA MINER	DM 29,90
ULTIMATH	DM 89,90
ASSEMBLER	DM 34,90
BIG BOSS	DM 29,90
VORTEX RAIDER	DM 29,90

## INFO-Katalog SI

Info-Katalog für Spectrum + C64, über 100 Seiten mit vielen Programmen direkt zum Abtippen, Tips + Info über Ihren Rechner, Peripherie und was man damit machen kann und natürlich jede Menge Programmbeschreibungen gegen DM 3,- in Briefmarken.

Alle Preise incl. MwSt. Bei Nachnahme zuzügl. DM 5,90. Bei Vorkasse mit Scheck zu zügl. DM 2,50. Ab DM 250, Warenwert porto- und verpackungsfreie Lieferung. Händleranfragen erwünscht.

STEPHAN TRIEBNER, Elektronische Datenverarbeitung, Postfach 12 72, 6103 Griesheim/Hessen, Tel.: 0 61 55/17 77

# MASSENSPEICHER

Recorder kostet 259 Mark und wird direkt auf den Bus des Spectrum gesteckt.

## Konsequente Weiterentwicklung

Die genannten Mängel der C-Kassette wurden auch bei der als Revolution gefeierten Microdrive-Entwicklung nur teilweise behoben. Im Cartridge des Microdrive befindet sich ein dünnes Endlos-Tonband, auf dem die Daten blockweise abgelegt und gelesen werden. Das Band jagt mit hoher Geschwindigkeit am Tonkopf vorbei und ist in etwa sieben Sekunden einmal »rum«. Ein gesuchtes Programm kann also, je nachdem, wie das Band gerade steht, in spätestens 14 Sekunden gefunden werden (diese »lange« Zeit wird benötigt, wenn das Band gerade am Anfang des gesuchten Programms steht und fast zweimal umlaufen muß, um Bandanfang mit dem »Directory« und Programmumfang zu finden). Ein »Pro-



Das umstrittene Microdrive-System



Das angekündigte Viscount-System

gramm« auf diesem Cartridge ist immer vorhanden: das Directory, in dem Filenamen, Filelänge und weitere Systemdaten abgelegt sind. Etwa zwei volle Spectrum-Speicher (RAM) passen auf ein Cartridge, welches allerdings zwischen 20 und 25 Mark pro Stück kostet. Das Laufwerk ist für etwa 250 bis 300 Mark zu haben, benötigt aber als Schnittstelle zum Spectrum das Interface I. Dieses Interface kostet ebenfalls 250 bis 300 Mark und bietet nebenher noch eine »V.24-Schnittstelle« für Drucker und andere Erweiterungen.

Auf dieses recht ordentliche Sinclair-Produkt will ich hier jedoch nicht weiter eingehen, dazu finden Sie zwei gesonderte Artikel in diesem Heft.

## Mehr Gerät für weniger Geld

Commander Turbo-Dual-Drive heißt eine Überraschung aus England, die von Nettetal Computer-systeme in Deutschland vertrieben wird. Dieses Speichermedium ist



# IN MASSEN

sowohl für den Spectrum als in Kürze auch für Commodore 64 erhältlich. Es funktioniert im Prinzip wie der Microdrive von Sinclair, ist jedoch als Doppellaufwerk konzipiert und hat neben der V.24- auch eine Centronics-Schnittstelle zum Anschluß eines Druckers eingebaut. Ein gesondertes Interface wird nicht benötigt, das Prunkstück für knapp 500 Mark wird direkt an den Spectrum angeschlossen. Kassetten dafür gibt es in fünf Längen: 17, 38, 70, 102 und 128 KByte Speicherkapazität stehen zur Verfügung. Diese Kassetten kosten je nach »Länge« um die 15 Mark. Einen ausführlichen Testbericht werden wir in einer der nächsten Ausgaben unserer »Happy-Computer« vorstellen. Neben dem erheblichen Preisvorteil ist noch positiv zu vermerken, daß das System keinen RAM-Speicherplatz im Spectrum belegt und somit alle vorhandene Software läuft.

## Jetzt geht es rund

Alle bisher genannten Speichergeräte verwenden als Datenträger ein mehr oder weniger normales Tonband. Seit wenigen Wochen jedoch, und leider auch nur in begrenzten Stückzahlen, sind richtige Floppy-Laufwerke erhältlich, die über ein spezielles Interface von Technology Reserach, mit dem Spectrum-Bus verbunden werden. Diese Einfach- oder Doppellaufwerke verwenden normale 5¼-Zoll-Disketten als Datenträger. Je nachdem, welches Shugart-kompatible Laufwerk sie verwenden, werden die Daten auf 40 bis 80 Spuren ein- oder zweiseitig in einfacher oder doppelter Dichte gespeichert. Knapp 400 KByte Speicherplatz sind das erreichbare Maximum (formatiert). Sie können sich sowohl das einzelne Interface für rund 500 Mark als auch ein komplettes System mit Doppellaufwerk für bis zu 1900 Mark bei Profisoft oder von Profisoft belieferten Händlern kaufen. Ausführliche Testberichte finden Sie übrigens in »Computer persönlich«, Ausgabe 18, ab Seite 166 und in »Happy-Computer«, Ausgabe 10, ab Seite 42.

## Geht es billiger oder preiswerter?

Wenn man der Ankündigung von Stephan Triebner glauben kann, dann bekommt man in Kürze bei ihm ein komplettes Floppy-Disk-System für unter 900 Mark. Es handelt sich hierbei um ein neues verbessertes Viscount-System mit Spectrum-Interface, Shugart-SA200-Laufwerk und eingebautem Netzteil. Gespeichert wird auf 5¼-Zoll-Diskette in 40 Spuren einseitig mit einfacher Schreibdichte. Dies ergibt etwa 100 KByte Speicherkapazität pro Diskette. Der benötigte RAM-Bereich des Controllers liegt an der oberen Grenze des 48-KByte-Speichers des Spectrum.

Ob die mit diesem System beschriebenen Disketten mit dem Profisoft-System zu lesen sind und umgekehrt, wird ein Test ergeben. Da aber beide Systeme die gleichen Laufwerke verwenden (auch Tech-

nology formatiert auf Befehl auf 40 Spuren), ist eine Kompatibilität zu erwarten.

## Kleiner bietet's keiner

Angekündigt wurden sie schon, zu sehen gab es bislang noch keines: 3-Zoll-Floppy-Laufwerke mit Spectrum-Interface von Schumpich.

Das komplette System soll unter 1000 Mark kosten und in seinen Abmessungen sich auf Spectrum-Format beschränken.

## Wo gibt's Software?

Die Speicherkapazität beträgt 145 KByte je Diskette, deren Preis bei 25 Mark liegen dürfte. Das System verfügt über eine Autostart-Möglichkeit. Weitere Informationen liegen nicht vor.

Wir werden Ihnen jedoch schnellstmöglich einen Testbericht liefern.



Das »hitverdächtige« Dual-Drive

# MASSENSPEICHER

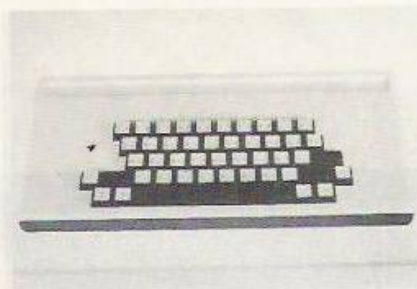


Der schnelle »Sprinter«

So schön es sein mag, daß der Verbraucher nun auch unter mehreren Disketten-Systemen wählen kann, so schwer wird die Entscheidung natürlich für die Software-Häuser, auf welchen Datenträgern sie in welchem Format ihre Software anbieten sollen. Im Moment gibt es nur für das Profisoft-System passende Software auf Diskette. Engagiert hat sich hier unter anderem die kleine Firma Strecker, die auch dieses System vertreibt. Wenn Sie sich (eventuell Weihnachten) ein richtiges Disketten-System kaufen (oder schenken lassen) wollen, folgen Sie meinem Rat: Marktbeobachtung und Preisvergleich lohnen sich.

(mk)

## SPECTRUM und ZX 81



**Spectrum Profitastatur** DM 198,—  
E-Look-Taste, 2 Caps-Shift  
2 Symbol-Shift zusätzliche Cursor-Tasten  
Spectrum-Platine kann eingebaut werden.  
Anschluß der Tastatur durch Stecker.

### Busverlängerung

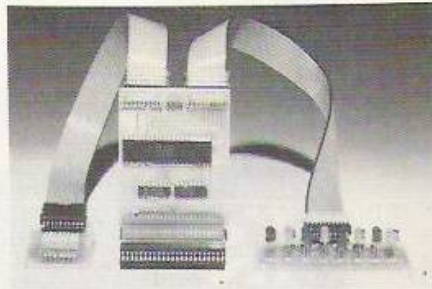
mit 20 cm flexiblem Kabel, Bausatz DM 34,—

### Profitastatur

ohne Gehäuse DM 159,—

### ZX81-Tastatur

mit Dauerfunktion auf allen Tasten DM 89,—



### Spectrum-PIO-Bausatz

zum Ansteuern von Lämpchen, Motoren, Eisenbahn u. vieles mehr  
Bausatz mit Steckadapter

DM 89,—

### dto. für ZX81

DM 79,—

### Eingabeeinheit

mit DIL-Schalter (im Bild links)

DM 24,90

### Ausgabeeinheit

mit LED (im Bild rechts)

DM 29,—

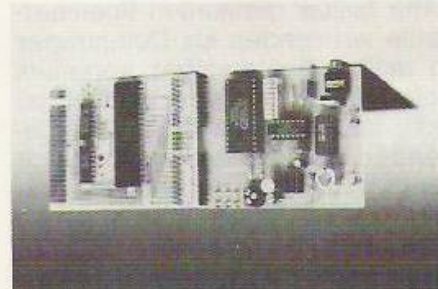
### Spectrumstecker 56polig,

2 Stück und 1 Zwischenstück

DM 34,—

### dto. für ZX81

DM 29,—



### Spectrum-Adapter

mit 2 Kontakt-Klammern DM 49,—

### Sondergenerator

für Spectrum mit ausführlicher Beschreibung und 20 K Musikeditor auf Kassette nur

DM 98,—

### Experimentierplatinen

für ZX81 oder Spectrum, 2 Stück DM 20,—

Ab Dezember lieferbar:

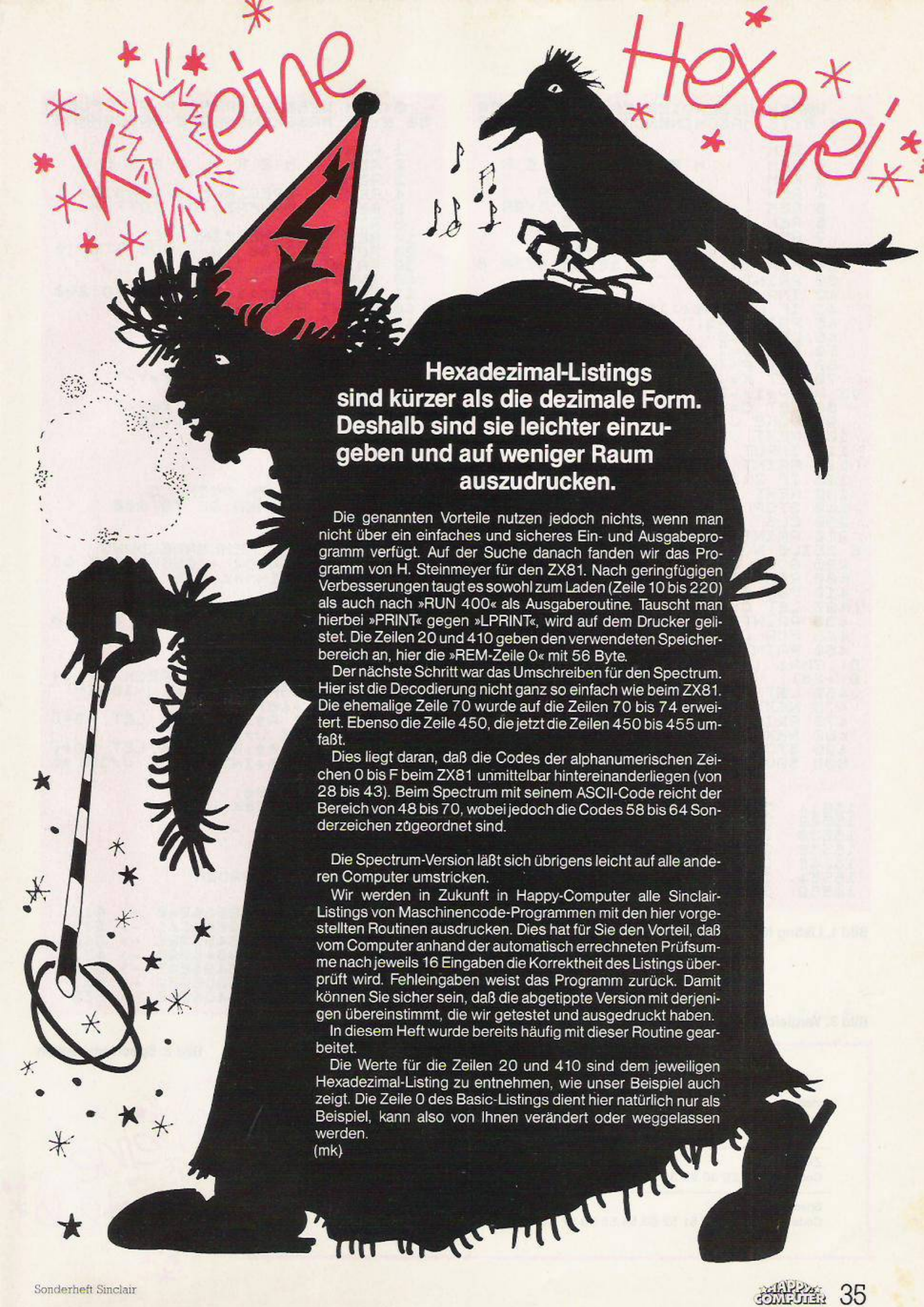
### Floppy-Interface

für Sinclair-Spectrum, Preis ca. DM 350,—  
für Laufwerke bis 1 MByte

Software Sonderangebote: **The Hobbit**, das englische Superspiel nur DM 45,—; **Penetrator** nur DM 25,—;  
Hobbit + Penetrator zusammen DM 65,—

Versand per Vorkasse + DM 3,— per NN + DM 4,70 · Prospekt kostenlos · Lieferung kurzfristig · Auf alle Geräte 6 Monate Garantie

**S. Egeler MC-Elektronik, Grünthal 21, 8201 Raubling** Telefon ab 18 Uhr (08035) 5666



## Hexadezimal-Listings sind kürzer als die dezimale Form. Deshalb sind sie leichter einzu- geben und auf weniger Raum auszudrucken.

Die genannten Vorteile nutzen jedoch nichts, wenn man nicht über ein einfaches und sicheres Ein- und Ausgabeprogramm verfügt. Auf der Suche danach fanden wir das Programm von H. Steinmeyer für den ZX81. Nach geringfügigen Verbesserungen taugt es sowohl zum Laden (Zeile 10 bis 220) als auch nach »RUN 400« als Ausgaberroutine. Tauscht man hierbei »PRINT« gegen »LPRINT«, wird auf dem Drucker gelistet. Die Zeilen 20 und 410 geben den verwendeten Speicherbereich an, hier die »REM-Zeile 0« mit 56 Byte.

Der nächste Schritt war das Umschreiben für den Spectrum. Hier ist die Decodierung nicht ganz so einfach wie beim ZX81. Die ehemalige Zeile 70 wurde auf die Zeilen 70 bis 74 erweitert. Ebenso die Zeile 450, die jetzt die Zeilen 450 bis 455 umfaßt.

Dies liegt daran, daß die Codes der alphanumerischen Zeichen 0 bis F beim ZX81 unmittelbar hintereinanderliegen (von 28 bis 43). Beim Spectrum mit seinem ASCII-Code reicht der Bereich von 48 bis 70, wobei jedoch die Codes 58 bis 64 Sonderzeichen zugeordnet sind.

Die Spectrum-Version läßt sich übrigens leicht auf alle anderen Computer umstricken.

Wir werden in Zukunft in Happy-Computer alle Sinclair-Listings von Maschinencode-Programmen mit den hier vorgestellten Routinen ausdrucken. Dies hat für Sie den Vorteil, daß vom Computer anhand der automatisch errechneten Prüfsumme nach jeweils 16 Eingaben die Korrektheit des Listings überprüft wird. Fehleingaben weist das Programm zurück. Damit können Sie sicher sein, daß die abgetippte Version mit derjenigen übereinstimmt, die wir getestet und ausgedruckt haben.

In diesem Heft wurde bereits häufig mit dieser Routine gearbeitet.

Die Werte für die Zeilen 20 und 410 sind dem jeweiligen Hexadezimal-Listing zu entnehmen, wie unser Beispiel auch zeigt. Die Zeile 0 des Basic-Listings dient hier natürlich nur als Beispiel, kann also von Ihnen verändert oder weggelassen werden.

(mk)

```

0>REM RESERVIERTER PLATZ FUER
56 BYTE MASCHINENCODE-PROGRAMME

1 REM
2 REM      H E X L O A D E R
3 REM
4 REM      (C) BY  QUENTER
5 REM                STEINMEYER
6 REM                6/1984
7 REM
10 REM SPEICHEREINGABE
20 FOR I=16514 TO 16562 STEP 8
30 PRINT I;" ";
40 INPUT A$
41 IF LEN A$<>16 THEN GOTO 200
45 PRINT A$;" => ";
46 LET C=0
50 FOR J=I TO I+7
60 LET P=(J-I)*2+1
70 LET X=(CODE A$(P)-28)*16+CO
DE A$(P+1)-28
80 LET C=C+X
90 POKE J,X
100 NEXT J
110 INPUT Z
115 PRINT C
120 IF Z<>C THEN GOTO 200
130 NEXT I
140 STOP
200 CLS
210 PRINT "ERROR - GEBEN SIE DI
E ZEILE NOCHEINMAL EIN"
220 GOTO 30
400 REM SPEICHERAUSGABE
410 FOR I=16514 TO 16562 STEP 8
420 LET C=0
430 PRINT I;" ";
440 FOR J=I TO I+7
450 PRINT CHR$ INT (PEEK J/16+2
8);CHR$ (PEEK J-16*INT (PEEK J/1
6)+28);
460 LET C=C+PEEK J
470 NEXT J
475 PRINT " => ";C
480 NEXT I
490 STOP
500 SAVE "HEXLOADER"

```

16514	372A362A373B2E20	=>	397
16522	37392A3700353120	=>	349
16530	393F002B3A2A3700	=>	318
16538	212200273E392A00	=>	267
16546	32263828202E332A	=>	368
16554	332834292A163537	=>	366
16562	342C372E32322A00	=>	331

Bild 1. Listing für ZX81

```

0>REM RESERVIERTER PLATZ FUER
56 BYTE MASCHINENCODE-PROGRAMME

1 REM
2 REM      H E X L O A D E R
3 REM
4 REM      SPECTRUM - VERSION
5 REM      MANFRED-D. KOTTING
6 REM
10 REM SPEICHEREINGABE
20 FOR I=23760 TO 23808 STEP 8
30 PRINT I;" ";
40 INPUT A$
41 IF LEN A$<>1 THEN GO TO 200
45 PRINT A$;" -> ";
46 LET C=0
50 FOR J=I TO I+7
60 LET P=(J-I)*2+1
70 LET Y=CODE A$(P)
71 IF Y>57 THEN LET Y=Y-7
72 LET X=CODE A$(P+1)
73 IF X>57 THEN LET X=X-7
74 LET X=X-48+(Y-48)*16
80 LET C=C+X
90 POKE J,X
100 NEXT J
110 INPUT Z
115 IF C=Z THEN PRINT C
120 IF Z<>C THEN GO TO 200
130 NEXT I
140 STOP
200 CLS : REM FEHLERMELDUNG
210 PRINT "ERROR - GEBEN SIE DI
E ZEILE NOCHEINMAL EIN"
220 GO TO 30
400 REM SPEICHERAUSGABE
410 FOR I=23760 TO 23808 STEP 8
420 LET C=0
430 PRINT I;" ";
440 FOR J=I TO I+7
450 LET A$=CHR$ INT (PEEK J/16+
48); LET B$=CHR$ (PEEK J-16*INT
(PEEK J/16)+48)
451 IF CODE A$>57 THEN LET A$=C
HR$ INT (PEEK J/16+55)
452 IF CODE B$>57 THEN LET B$=C
HR$ (PEEK J-16*INT (PEEK J/16)+5
5)
455 PRINT A$;B$;
460 LET C=C+PEEK J
470 NEXT J
475 PRINT " -> ";C
480 NEXT I
490 STOP
500 SAVE "HEXLOADER"

```

23760	5245534552564945	->	613
23768	5254455220504C41	->	570
23776	545A204655455220	->	544
23784	3536204259544520	->	479
23792	4D41534348494E45	->	584
23800	4E434F44452D5052	->	568
23808	4F4752414D4D4520	->	552

Bild 2. Spectrum-Version

Bild 3. Vergleichstabelle der Codes

Alphanu- meri- sches Zeichen:	0	1	2	3	4	5	6	7	8	9	A	B	C	D	F	E
ZX81- Code	28	29	30	31	32	33	34	35	36	37	38	39	40	41	43	42
Spectrum- Code	48	49	50	51	52	53	54	55	56	57	65	66	67	68	70	69



# Bingo-Jackpot

**Programme, die Geldspielgeräte simulieren, gibt es wie Sand am Meer. Noch eines? Ich glaube, daß sich »Bingo-Jackpot« wesentlich von anderen Programmen dieser Art unterscheidet.**

Normalerweise fehlt den Simulatoren der nötige Spielreiz, weil ein richtiges Risiko, Geld zu verlieren, nicht besteht. Meist wird auch der extrem kurze Spielablauf übernommen, und die Gewinnkombinationen sind von vornherein festgelegt. Fazit: Was beim »großen Bruder« durchaus Spannung erzeugt, nimmt den Simulatoren den Pfiff, und die Sache wird schon nach wenigen Durchläufen langweilig.

## Die Idee

Meine Idee war es, den Spieß umzudrehen und ein Programm zu schreiben, dessen Konzept für ein richtiges Gerät nicht anwendbar wäre, dafür einem Spiel im herkömmlichen Sinne aber sehr nahekommt. Daher gibt es bei »Bingo-Jackpot« eine variable Gewinnmöglichkeit, die vom Spieler Kalkulationsfähigkeit und Risikobereitschaft verlangt (ähnlich wie bei 17+4 oder Poker). Wie bei den Originalgeräten kann auf spannende Weise der Gewinn durch »Gambling« verändert werden. Um einen übergeordneten Spielsinn zu schaffen, hat das Programm eine Jackpot-Funktion, deren Auslösung von der kompletten Sammlung einer der beiden Gewinnmöglichkeiten (Symbol oder Farbe) abhängt (ähnlich wie bei Rommé oder Bingo). Die Jackpot-Auslösung erfolgt in Form von Serienspielen, in denen jeweils der bis dahin gesammelte Jackpot-Betrag gewonnen werden kann.

Es gibt die Möglichkeit, allein (Startkapital = 15 Mark) oder zu zweit (Startkapital = 10 Mark) zu spielen. Ein geschlossener Spielablauf ergibt sich dadurch, daß Minuszahlen im Spie-

lerkapital oder die Jackpot-Auslösung das Spiel regelrecht beenden. Die Spielzeit beträgt dabei meist zwischen 15 und 30 Minuten. Das Spielziel ist für den Einzelspieler, den Jackpot möglichst hochzutreiben und eine hohe Zahl von Serienspielen zu erzielen. Spielt man zu zweit, ist es wichtig, durch die Jackpot-Auslösung dem anderen zuvorzukommen.

## Bemerkungen zum Programm

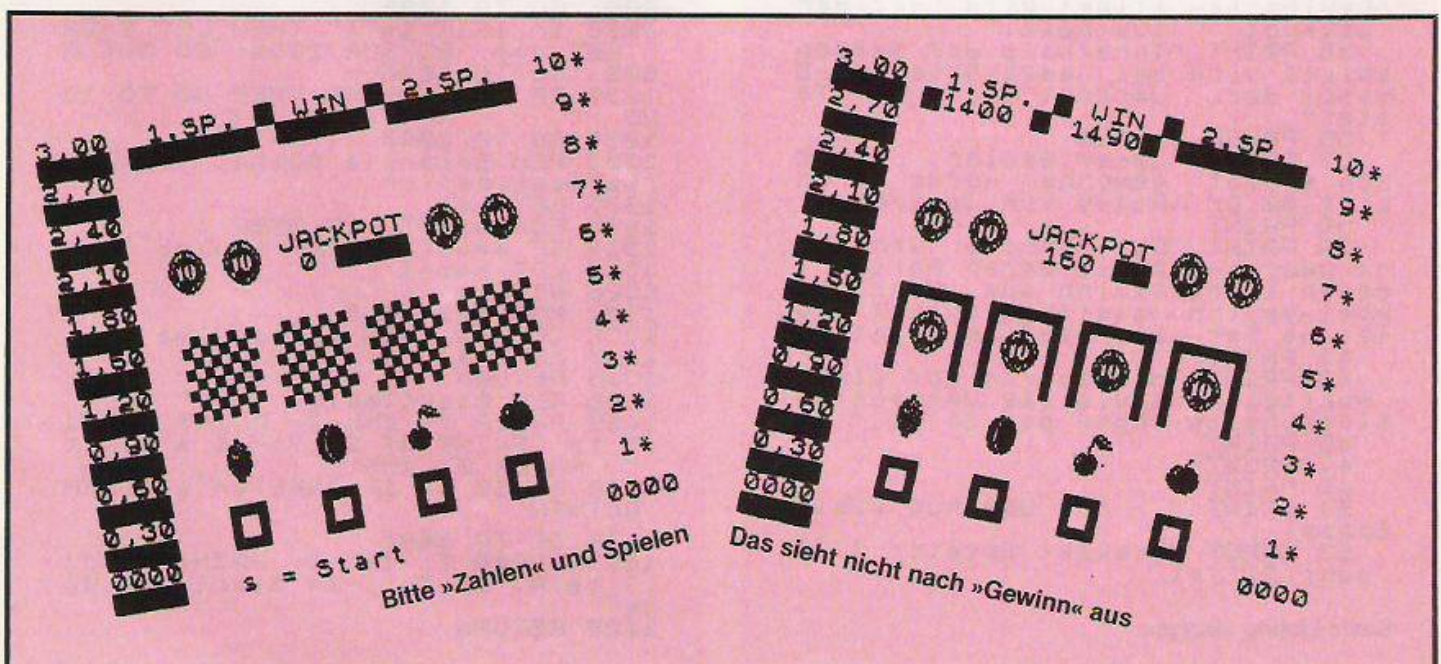
Die Zeilen 2950 bis 2970 sind absichtlich als Unterprogramme geschrieben worden (die Schleifenform war unübersichtlicher).

Der Einbau einer Stop-Möglichkeit war vorgesehen, hätte aber den Ablauf unnötig verzögert und zum Spielreiz nicht wesentlich beigetragen.

## Tips zum Abtippen

Welche Buchstaben im Grafik-Modus eingegeben werden müssen, ist im Unterprogramm 6500 ersichtlich. Die Zeilen 3 bis 51 (= Bedienungsanleitung) sowie alle REMs können weggelassen werden. Die Zeilen 1210-1225, 2950-2970, 4005-4020, 4055-4070, 6500-6570 können mit »EDIT« mit geänderten Zeilennummern schnell geschrieben werden.

(P. Bergen)



```

1 RANDOMIZE
3 REM Peter Bergen, Franken-
str. 29, 3200 Hildesheim
5 REM Bedienungsanleitung
7 GO SUB 5400: INK 0: PAPER 7
: INK 0: CLS: PRINT:
INGO - JACKPOT
9 PRINT
11 PRINT "Dieses Spiel simulie-
rt einen Geldspielautomaten.
Um einen Ge-winn zu bekommen, so-
llen 4 glei-che Symbole oder 4 g-
leiche Far-ben auf den Walzen e-
rscheinen."
13 PRINT "Jeder Walzenlauf kos-
tet 0,10DM. Der Spielstart (alle
e 4 Walzen) kostet daher 0,40DM.
Danach kannjede Walze einzeln n-
achgestartetwerden (-0,10DM)."
15 PRINT "Der Gewinn verringert
sich nach jedem Nachstart von
3,00DM um je0,30DM. Kann kein Ge-
winn erzieltwerden, ist der bis
dahin in-vestierte Einsatz ve-
rloren."
17 PRINT "Vor jeder Nachstartm-
oeglichkeit kann der Durchlauf b-
eendet wer- den. (Weiter mit Tas-
te)"
19 PAUSE 0: CLS
19 PRINT "Erscheint -1 als Sym-
bol, wird der Gewinn um eine S-
tufe zu- saetzlich verringert"
21 PRINT "Wurde gewonnen, ist
eine Erhoe- hung durch Risiko mo-
eglich."
23 PRINT
25 PRINT ; "----- Jackpot-Fun-
ktion -----"
27 PRINT
29 PRINT "Erscheint als Symbol
ein Gro- schen, wird das Jack-
pot-Konto um0,10DM vermehrt."
31 PRINT "Wurden alle 4 Farbe-
n oder alle 4 Symbole gewonnen,
wird der Jackpot ausgelost."
33 PRINT "Zur Ausloesung koenn-
en bis 10 Serienspiele gewonne-
n werden, deren Anzahl von dem
Gewinn be- stimmt wird, der den
Jackpot ausloest."
35 PRINT "Innerhalb der Serien-
spiele wird bei jedem Groschen-S-
ymbol der Jackpot gewonnen. (Tas-
te)"
36 PAUSE 0: CLS
37 PRINT "Jeder Gewinn, der sc-
hon einmal gewonnen wurde, brin-
gt 1,00 DM Abzug vom Jackpot."
38 PRINT
39 PRINT "Die Symbole >Grosche
n< oder >-1< sowie deren Farben l-
oesen keinen Gewinn aus. Einen Do-
ppel-gewinn (gleiches Symbol + g-
leiche Far- be) gibt es nicht."
41 PRINT
43 PRINT "Minuswerte auf einem
Punktekontosowie die Jackpot-Au-
sloesung be-enden das Spiel."
45 PRINT
47 PRINT
49 PRINT
51 PRINT " Und nun viel
Spaß!
53 INPUT "Wieviel Spieler ? (1
oder 2)";pl

```

Basic-Listing »Bingo«

```

55 LET hs=0
100 INK 0: BORDER 0: PAPER 7: C
LS
110 FLASH 1: PRINT AT 10,7;" Bi-
tte warten !": FLASH 0
1000 DIM a$(32,4): GO SUB 8005:
GO SUB 5500
1005 REM Definitionen und Spielf-
eldaufbau
1010 LET str=0: LET ki=0: LET pl-
e=1000: LET plz=1000: LET w=1: L-
ET jack=0: LET plp=21: LET jp=0:
LET geld=1000: DIM k(4): DIM i(
4): DIM l(4): DIM m(4)
1012 IF pl=1 THEN LET ple=1500:
LET geld=1500
1015 PAPER 0: CLS: GO SUB 7210
1020 GO SUB 7070
1025 IF pl=1 THEN LET plp=7
1030 REM Neudurchlauf
1035 LET gewinn=2950: INK 0: GO
SUB 7000: GO SUB 7100: GO SUB 71
70
1040 IF geld<0 THEN GO TO 5000
1045 GO SUB 2215
1050 PAPER 2: INK 7: GO SUB gewi-
nn
1055 GO SUB 3100
1060 REM 4*Walzenlauf
1065 GO SUB 1405: LET e=7: LET j
=1: GO SUB 2065
1070 LET e=12: LET j=2: GO SUB 2
065
1075 LET e=17: LET j=3: GO SUB 2
065
1080 LET e=22: LET j=4: GO SUB 2
065
1085 IF jack=1 THEN GO TO 4350
1090 GO SUB 2000
1095 GO SUB 1420
1200 REM Nachstart
1205 PAUSE 0
1210 IF INKEY$="1" THEN LET e=7:
LET j=1: GO SUB 2065: GO SUB 20
00: GO TO 1095
1215 IF INKEY$="2" THEN LET e=12
: LET j=2: GO SUB 2065: GO SUB 2
000: GO TO 1095
1220 IF INKEY$="3" THEN LET e=17
: LET j=3: GO SUB 2065: GO SUB 2
000: GO TO 1095
1225 IF INKEY$="4" THEN LET e=22
: LET j=4: GO SUB 2065: GO SUB 2
000: GO TO 1095
1230 IF INKEY$="e" THEN GO TO 10
25
1235 GO TO 1095
1300 REM Erlangte Symbol- und Fa-
rbenkombination
1305 LET b=1
1310 FOR a=1 TO 25 STEP 4
1315 IF d=a THEN LET k(j)=b
1320 LET b=b+1
1325 NEXT a
1330 FOR a=3 TO 6
1335 IF c=a THEN LET i(j)=a
1340 NEXT a
1345 RETURN
1400 REM Starttaste
1405 PAPER 7: INK 2: PRINT AT 21
,7;z$: PRINT AT 21,7;" s = Start
": PAPER 0: INK 7
1410 PAUSE 0: IF INKEY$="s" THEN
RETURN
1415 GO TO 1405
1420 PAPER 7: INK 2: PRINT AT 21
,7;z$: AT 21,7;" 1-4= Start, e= En-
de"
1425 RETURN

```

## Bingo-Jackpot

## Bingo-Jackpot

```

2000 REM Gewinnabastung
2005 FOR a=1 TO 4
2010 IF k(a)=5 OR k(a)=7 THEN LE
T Jp=Jp+10: BEEP .05,10: BEEP .0
5,30: LET ki=1: GO SUB 2215
2015 NEXT a
2020 FOR a=1 TO 4
2025 IF k(a)=6 THEN GO SUB 2900:
LET ki=1
2030 NEXT a
2035 IF ki=1 THEN GO TO 2050
2040 IF k(1)=k(2) AND k(1)=k(3)
AND k(1)=k(4) THEN GO TO 2125
2045 IF i(1)=i(2) AND i(1)=i(3)
AND i(1)=i(4) THEN GO TO 2100
2050 GO SUB 2900
2055 LET ki=0: RETURN
2060 REM Geld-10
2065 GO SUB 6000: GO SUB 1305: L
ET geld=geld-10: GO SUB 3000: RE
TURN
2070 STOP
2100 REM Gewinn-Markierung- und
Ermittlung
2105 FOR a=1 TO 4
2110 IF i(1)=(a+2) AND l(a)=2 TH
EN GO SUB 4100: GO TO 2145
2115 IF i(1)=(a+2) THEN LET l(a)
=1: GO SUB 4000+(a*5): GO TO 214
5
2120 NEXT a
2125 FOR a=1 TO 4
2130 IF k(1)=a AND m(a)=2 THEN G
O SUB 4100: GO TO 2140
2135 IF k(1)=a THEN GO SUB 4050+
(a*5): GO TO 2160
2140 NEXT a
2160 GO SUB 4200
2165 FOR a=1 TO 20
2170 BEEP .01,a
2175 INK 5
2180 FOR a=1 TO 21 STEP 2
2185 BEEP .01,a: PRINT AT a,1;"■"
2190 NEXT a
2195 PAPER 0: PRINT AT 4,7; z$
2200 GO SUB 3500
2210 GO TO 1035
2215 PAPER 7: INK 2: PRINT AT 7,
13;"Jp:"
2220 RETURN
2200 REM 1 Stufe weniger
2210 BEEP .2,-20
2220 LET gewinn=gewinn+2
2222 PAPER 6: INK 0: GO SUB gewi
nn-2
2224 PAPER 2: INK 7: GO SUB gewi
nn
2230 RETURN
2247 REM Bank
2248 GO TO 2950
2250 PRINT AT 0,1;"3,00"; AT 0,28
;"10*": LET f=300: LET s=10: RE
TURN
2252 PRINT AT 2,1;"2,70"; AT 2,28
;"9*": LET f=270: LET s=9: RET
URN
2254 PRINT AT 4,1;"2,40"; AT 4,28
;"8*": LET f=240: LET s=8: RET
URN
2256 PRINT AT 6,1;"2,10"; AT 6,28
;"7*": LET f=210: LET s=7: RET
URN
2258 PRINT AT 8,1;"1,80"; AT 8,28
;"6*": LET f=180: LET s=6: RET
URN
2260 PRINT AT 10,1;"1,50"; AT 10,
28;"5*": LET f=150: LET s=5: R
ETURN

```

```

2262 PRINT AT 12,1;"1,20"; AT 12,
28;"4*": LET f=120: LET s=4: R
ETURN
2264 PRINT AT 14,1;"0,90"; AT 14,
28;"3*": LET f=90: LET s=3: RE
TURN
2266 PRINT AT 16,1;"0,60"; AT 16,
28;"2*": LET f=60: LET s=2: RE
TURN
2268 PRINT AT 18,1;"0,30"; AT 18,
28;"1*": LET f=30: LET s=1: RE
TURN
2270 PRINT AT 20,1;"0000"; AT 20,
28;"0000": LET f=0: LET s=0: RET
URN
2272 GO SUB 5300: LET gewinn=295
0
2273 IF Jack=1 THEN GO TO 5000
2274 GO TO 1035
2280 RETURN
3000 REM Finanzregelung
3005 IF pl=1 THEN GO SUB 3300
3010 IF plp=7 THEN LET ple=geld:
GO SUB 3300: GO SUB 3210: RETUR
N
3015 IF plp=21 THEN LET plz=geld
: GO SUB 3300: GO SUB 3210: RETU
RN
3100 REM Spielerwechsel
3105 IF pl=1 THEN RETURN
3110 IF plp=7 THEN LET plp=21: L
ET geld=plz: GO SUB 3200: RETURN
3115 IF plp=21 THEN LET plp=7: L
ET geld=ple: GO SUB 3205: RETURN
3120 GO SUB 3210
3125 RETURN
3200 INK 0: PAPER 6: PRINT AT 0,
6;"1.SP.": PAPER 2: INK 7: AT 0,
20;"2.SP.": RETURN
3205 INK 7: PAPER 2: PRINT AT 0,
6;"1.SP.": PAPER 6: INK 0: AT 0,
20;"2.SP.": RETURN
3210 REM hs
3215 IF geld>hs THEN LET hs=geld
3220 PRINT AT 1,14;hs
3225 RETURN
3300 REM Geldausdruck
3305 PAPER 7: INK 0: PRINT AT 1,
plp;geld;"": RETURN
3500 REM Gambling
3505 PAPER 0: PRINT AT 21,7; z$;
PAPER 6: INK 2: AT 21,7;"R = Risi
ko, S = Stop"
3510 IF INKEY$="s" THEN LET geld
=geld+f: GO SUB 5100: GO SUB 300
0: RETURN
3515 IF INKEY$="r" THEN GO TO 35
25
3520 GO TO 3510
3525 LET r=INT (RND*10)+1
3530 IF r<7 THEN PAPER 2: INK
7: LET gewinn=gewinn+2: BEEP .1,
-20: GO SUB gewinn: PAPER 6: INK
0: GO SUB (gewinn-2): LET f=f-3
0: LET s=s-1: GO TO 3500
3535 PAPER 2: INK 7: LET gewinn=
gewinn-2: BEEP .1,20: GO SUB gew
inn: PAPER 6: INK 0: GO SUB (gew
inn+2): LET f=f+30: LET s=s+1: G
O TO 3505
3540 GO TO 3500
4000 REM JP-Verwaltung
4005 INK 3: PRINT AT 18,8;"■"; A
T 19,8;"■": LET l(1)=2: RETURN
4010 INK 4: PRINT AT 18,13;"■";
AT 19,13;"■": LET l(2)=2: RETUR
N

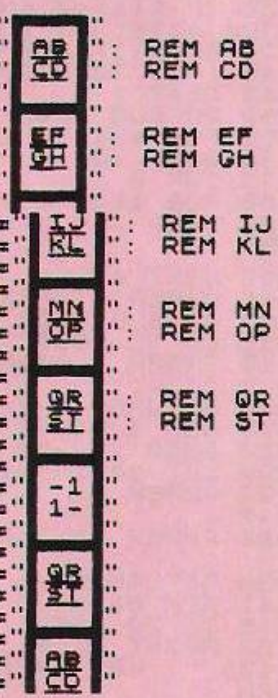
```

```

4015 INK 5: PRINT AT 18,18;"███";
AT 19,18;"███": LET l(3)=2: RETURN
N
4020 INK 6: PRINT AT 18,23;"███";
AT 19,23;"███": LET l(4)=2: RETURN
N
4055 INK 6: PRINT AT 15,8;"AB";A
T 16,8;"CD": LET m(1)=2: RETURN
4060 INK 3: PRINT AT 15,13;"EF";
AT 16,13;"GH": LET m(2)=2: RETURN
N
4065 INK 4: PRINT AT 15,18;"IJ";
INK 2: AT 15,18;"KL": LET m(3)=2
: RETURN
4070 INK 4: PRINT AT 15,23;"MN";
AT 16,23;"OP": LET m(4)=2: RETURN
N
4075 RETURN
4100 REM JP-Doppel
4105 FOR a=1 TO 20
4110 BEEP .01,30: PAPER 2: INK 7
: FLASH 1: PRINT AT 6,7;" 100 ";
AT 7,7;"Abzug";AT 6,21;"Abzug";A
T 7,21;" 100 ": BEEP .01,10
4115 NEXT a
4120 LET jp=jp-100
4125 FLASH 0: GO SUB 2215: GO SU
B 7240
4130 RETURN
4200 REM Jackpot voll?
4202 IF l(1)+l(2)+l(3)+l(4)=8 TH
EN GO TO 4300
4205 IF m(1)+m(2)+m(3)+m(4)=8 TH
EN GO TO 4300
4210 RETURN
4300 REM JP-Ausloesung
4305 LET jack=1: INK 5
4310 FOR a=1 TO 21 STEP 2
4315 BEEP .01,a: PRINT AT a,28;"
███"
4320 NEXT a
4325 FLASH 1: GO SUB 7240: FLASH
0: GO SUB 5400
4330 GO SUB 3500
4335 GO SUB 7100
4340 IF s<0 THEN GO SUB 5300: GO
TO 5000
4345 GO TO 1065
4350 GO SUB 4370
4355 GO SUB 2920: LET s=s-1
4358 GO TO 4335
4370 FOR b=1 TO 4
4375 IF k(b)=5 OR k(b)=7 THEN LE
T geld=geld+jp: GO SUB 5400: GO
SUB 3000
4380 IF k(b)=6 THEN GO SUB 2900:
LET s=s-1
4385 NEXT b
4390 RETURN
5000 REM ENDE
5005 PAPER 6: INK 0: PRINT AT 10
,7;"███";AT 11,7;"███";AT 12,7
;"███";AT 13,7;"███"
5010 PAPER 4: PRINT AT 10,12;"
███";AT 11,12;"███";AT 12,12;"
███";AT 13,12;"███"
5015 PAPER 5: PRINT AT 10,17;"
███";AT 11,17;"███";AT 12,17;"
███";AT 13,17;"███"
5020 PAPER 3: PRINT AT 10,22;"
███";AT 11,22;"███";AT 12,22;"
███";AT 13,22;"███"
5022 FOR a=10 TO -30 STEP -1
5023 BEEP .02,a: BEEP .01,a+2
5024 NEXT a
5025 PAPER 2: INK 7: PRINT AT 21
,7;" Noch einmal? (j/n) "
5030 PAUSE 0: IF INKEY$="j" THEN
GO TO 1010
    
```

```

5035 STOP
5100 REM Melodien
5105 BEEP .1,0: BEEP .1,0: BEEP
.1,0: BEEP .2,5: BEEP .2,9
5110 RETURN
5300 FOR a=1 TO 50: NEXT a
5302 BEEP .2,5: BEEP .2,5: BEEP
.1,7: BEEP .3,9: BEEP .1,5: BEEP
.2,7: BEEP .1,4: BEEP .3,5
5310 RETURN
5400 BEEP .2,5: BEEP .2,5: BEEP
.3,7: BEEP .3,5: BEEP .2,14: BEE
p .1,14: BEEP .35,14
5410 RETURN
6000 REM Walzensteuerung
6005 FOR a=1 TO 20: NEXT a
6010 LET v=INT (RND*27)-4
6015 FOR d=1 TO 27
6020 LET c=INT (RND*4)+3
6025 PAPER c: INK 0: PRINT AT 10
,e;a$(d);AT 11,e;a$(d+1);AT 12,e
;a$(d+2);AT 13,e;a$(d+3)
6030 IF d>v THEN GO SUB 6050: GO
TO 6060
6040 NEXT d
6045 GO TO 6015
6050 PAPER 0: PRINT AT 21,7;z$
6055 RETURN
6060 BEEP .002,10: IF d=1 OR d=5
OR d=9 OR d=13 OR d=17 OR d=21
OR d=25 THEN LET st=0: LET stf=0
RETURN
6065 GO TO 6040
6500 LET a$(1)="
6502 LET a$(2)="
6504 LET a$(3)="
6506 LET a$(4)="
6510 LET a$(5)="
6512 LET a$(6)="
6514 LET a$(7)="
6516 LET a$(8)="
6520 LET a$(9)="
6522 LET a$(10)="
6524 LET a$(11)="
6526 LET a$(12)="
6530 LET a$(13)="
6532 LET a$(14)="
6534 LET a$(15)="
6536 LET a$(16)="
6540 LET a$(17)="
6542 LET a$(18)="
6544 LET a$(19)="
6546 LET a$(20)="
6550 LET a$(21)="
6552 LET a$(22)="
6554 LET a$(23)="
6556 LET a$(24)="
6558 LET a$(25)="
6560 LET a$(26)="
6562 LET a$(27)="
6564 LET a$(28)="
6566 LET a$(29)="
6568 LET a$(30)="
6570 LET a$(30)="
6580 RETURN
6999 STOP
7000 REM Spielfeld
7010 INK 0: PAPER 6:
7020 FOR a=0 TO 20 STEP 2
7030 GO SUB 2950+a
7040 PRINT AT a+1,1;"███"
7050 NEXT a
7055 RETURN
7070 PAPER 6: INK 0: PRINT AT 0
,6;" 1.SP. █ WIN █ 2.SP. ";AT 1,6
;"███"
7080 PAPER 7: INK 2
7090 RETURN
    
```





## Bingo-Jackpot

## Bingo-Jackpot

```

7100 PAPER 6: INK 0
7105 FOR a=7 TO 22 STEP 5
7110 FOR b=10 TO 13
7120 PRINT AT b,a;"███"
7130 BEEP .02,b-10
7140 NEXT b
7150 NEXT a
7155 PAPER 7: INK 0
7160 RETURN
7170 GO SUB 7240
7180 PAPER 6: INK 0: PRINT AT 6,
13;"JACKPOT"; INK 7;AT 7,13;"███"
7190 LET z$=""
7200 RETURN
7210 PAPER 0: INK 3: PRINT AT 18
0,"███";AT 19,8;"███"; INK 4;AT 1
0,13;"███";AT 19,13;"███"; INK 5;A
T 18,18;"███";AT 19,18;"███"; INK
6;AT 18,23;"███";AT 19,23;"███"
7220 PAPER 0: INK 6: PRINT AT 15
0,"AB";AT 16,8;"CD"; INK 3;AT 1
5,13;"EF";AT 16,13;"GH"; INK 4;A
T 15,18;"IJ"; INK 2;AT 16,18;"KL
";AT 15,23; INK 4;"MN";AT 16,23;
"OP"
7230 RETURN
7240 PAPER 0: INK 6: PRINT AT 6,
7;"QR";AT 7,7;"ST";AT 6,21
7;"UV";AT 7,21;"WX"
7250 RETURN
8000 REM USG
8005 FOR a=USR "A" TO USR "U"-1
8010 READ b: POKE a,b
8015 NEXT a
8020 DATA 1,1,7,14,15,27,24,31
8025 DATA 128,128,96,176,224,120
,216,176
8030 DATA 27,31,21,15,11,7,2,3
8035 DATA 248,216,248,176,112,22
4,128,128
8040 DATA 0,3,14,20,29,59,59,59
8045 DATA 0,64,240,248,248,252,2
52,252
8050 DATA 59,59,59,29,29,14,3,0
8055 DATA 252,252,252,248,248,24
0,64,0
8060 DATA 0,0,1,1,1,2,2,2
8065 DATA 96,152,230,120,63,0,0,
0
8070 DATA 25,63,111,95,127,63,63
,15
8075 DATA 128,192,224,224,224,19
2,192,0
8080 DATA 0,0,0,0,0,14,27,55
8085 DATA 0,0,96,64,128,240,248,
252
8090 DATA 55,127,127,127,63,31,7
,0
8095 DATA 252,254,254,254,252,24
8,224,0
8100 DATA 3,13,17,39,95,91,162,1
78
8105 DATA 192,175,136,228,250,26
,77,77
8110 DATA 242,242,115,79,47,18,1
3,3
8115 DATA 79,79,30,250,228,168,4
8,192
8120 RETURN

```

Basic-Listing »Bingo« (Schluß) ▲

Hardcopy der Spielanleitung (rechts) ►

## ███ BINGO - JACKPOT ███

Dieses Spiel simuliert einen Geldspielautomaten. Um einen Gewinn zu bekommen, sollen 4 gleiche Symbole oder 4 gleiche Farben auf den Walzen erscheinen. Jeder Walzenlauf kostet 0,10DM. Der Spielstart (=alle 4 Walzen) kostet daher 0,40DM. Danach kann jede Walze einzeln nachgestartet werden (-0,10DM). Der Gewinn verringert sich nach jedem Nachstart von 3,00DM um je 0,30DM. Kann kein Gewinn erzielt werden, ist der bis dahin investierte Einsatz verloren. Vor jeder Nachstartmöglichkeit kann der Durchlauf beendet werden. (Weiter mit Taste)

Erscheint -1 als Symbol, wird der Gewinn um eine Stufe zusätzlich verringert. Wurde gewonnen, ist eine Erhöhung durch Risiko möglich.

## ----- Jackpot-Funktion -----

Erscheint als Symbol ein Groschen, wird das Jackpot-Konto um 0,10DM vermehrt. Wurden alle 4 Farben oder alle 4 Symbole gewonnen, wird der Jackpot ausgelöst. Zur Auslösung können bis 10 Serienspiele gewonnen werden, deren Anzahl von dem Gewinn bestimmt wird, der den Jackpot auslöst. Innerhalb der Serienspiele wird bei jedem Groschen-Symbol der Jackpot gewonnen. (Taste)

Jeder Gewinn, der schon einmal gewonnen wurde, bringt 1,00 DM Abzug vom Jackpot.

Die Symbole >Groschen< oder >-1< sowie deren Farben lösen keinen Gewinn aus. Einen Doppelgewinn (gleiches Symbol + gleiche Farbe) gibt es nicht.

Minuswerte auf einem Punktkonto sowie die Jackpot-Auslösung beenden das Spiel.

Und nun viel Spass!

## Bingo-Jackpot

## Bingo-Jackpot

### Programmablauf

Nach der Bedienungsanleitung (5-51) wird mit »pl« die Spielanzahl definiert. 1000 ruft die UDG ab (GS.8000) und definiert die Walzen (GS.6500). Die Änderungen für Einzelspieler werden in 1012 vorgenommen. 1015 und 1020 rufen Spielfeldteile ab, deren Ergänzung in 1035 erfolgt. 1040 prüft, ob noch Kapital vorhanden ist und verzweigt eventuell nach 5000 (Ende).

1045 ruft die Jackpot-Anzeige ab (GS.2215).

1050 markiert den momentanen Gewinnstand rot/weiß.

1055 initiiert den Spielerwechsel (GS.3100).

Nach Auslösung der Starttaste (GS.1405) werden durch die Zeilen 1065-1080 alle vier Walzen in Bewegung gebracht (GS.6000).

Nach jeder erfolgten Walzendrehung wird im GS.2065 festgestellt, welche Symbol-/Farbenkombination vorliegt (GS.1305), wonach 0,10 Mark vom Spielkapital abgezogen werden.

1090 führt zur Gewinnabstastung (GS.2000 siehe Anhang).

Durch Aufruf der Nachstarttaste wird die Nachstartmöglichkeit freigegeben (GS.1420).

Bei jedem Nachstart werden die gleichen Mechanismen wie beim Neubeginn in Gang gesetzt.

Das Ende der Nachstartmöglichkeit wird durch Gewinn = 2972 eingeleitet, wonach die Rückkehr zu 1035 erfolgt.

### Gewinnabstastung

2010 stellt innerhalb einer Schleife fest, ob das Symbol »Groschen« vorliegt und schließt mit »ki« = 1 jeden weiteren Gewinn aus.

2025 hat die gleiche Funktion für »-1«.

Beide Zahlen leiten die nötigen Konsequenzen ein (GS.2215 beziehungsweise GS.2900).

Durch Ki wird bis zum Gewinnstufenabzug (GS.2900) alles übersprungen. 2055 stellt ki wieder auf Null und führt zurück.

Durch die Schleifen werden alle vier Walzen abgetastet.

Wurde »Groschen« oder »-1« ausgeschlossen, stellt 2040 fest, ob ein noch nicht näher definierter Symbolgewinn vorliegt.

Im positiven Fall wird in Zeile 2125 festgestellt, ob das betreffende Symbol (K) schon einmal gewonnen wurde (M=2).

Durch GS.4100 würden dann die Konsequenzen gezogen.

Andernfalls wird durch Aufruf des betreffenden GS. (4055-4070) in Zeile 2135 das jeweilige Symbol mit weißem Untergrund auf der Anzeige dargestellt und mit »m=2« markiert.

2160 fragt durch GS.4200 nach, ob der Jackpot ausgelöst werden kann. Nachdem nun feststeht, daß ein regulärer Gewinn vorliegt, wird er durch die Zeilen 2165-2220 ausgelöst, wobei die endgültige Gewinnfestlegung durch das Gambling-GS. erfolgt (3500).

3500 führt mit neuem Kontostand zurück.

Die Farben werden durch den gleichen Ablauf ab Zeile 2045 überprüft.

### Jackpot-Auslösung

Haben die Zeilen 4200 oder 4205 den Jackpot als voll definiert (alle vier Symbole oder alle vier Farben), wird die Auslösung durch die Zeilen 4300 und 4390 vorgenommen.

Auch hier erfolgt die endgültige Festlegung der Serienspiele durch GS.3500.

Bevor die Serienspiele beginnen, wird in Zeile 4340 geprüft, ob noch Spiele vorhanden sind (S). Eventuell wird dann das Ende (GS.5000) eingeleitet. 4345 bewegt über GS.1065 die vier Walzen.

GS.4370 stellt fest, ob »Groschen« oder »-1« vorliegt und leitet die Konsequenzen ein (geld + jp bzw. Abzug einer Gewinnstufe).

Durch 4360 wird eine Schleife gebildet, die erst durch 4340 verlassen werden kann. (Spielende)

### Variablenliste

a und b	= universelle Variable für Schleifen
c	= Zufallsvariable für Farbe (1335, 6020, 6025)
d	= Variable für Walzendurchlauf (1315, 6015, 6025, 6030, 6040, 6060)
e	= Walzenposition (1065, 1070, 1075, 1080, 1210, 1215, 1220, 1225, 6025)
f	= Geldgewinn (2950-2970, 3510, 3530, 3535)
geld	= Kapital (wechselnd bei zwei Spielern; 1010, 1012,

gewinn	2065, 3010, 3015, 3110, 3115, 3215, 3305, 3510, 4375) = Variable für die Unterprogramme 2950-2970 (1035, 1050, 2205, 2920, 2922, 2924, 2972, 3530, 3535)
hs	= Highscore (angezeigt als »WIN«; 55, 3215, 3220)
i	= Farbenkontrollvariable (1010, 1335, 2045, 2110, 2115)
j	= Symbolposition (1065, 1070, 1075, 1080, 1210, 1215, 1220, 1225, 1315, 1335)
jp	= Jackpotwert (1010, 2010, 2215, 4120, 4375)
jack	= Jackpotauszahlungs-Kennzeichnung (1010, 1085, 4305)
k	= Symbolkontrollvariable (1010, 2010, 2025, 2040, 2130, 2135, 2150, 4375, 4380)
ki	= Gewinnstop bei »Groschen« oder »-1« (1010, 2010, 2025, 2035, 2055)
l	= Farbenkontrollvariable (1010, 2110, 2115, 4005, 4010, 4015, 4020, 4200)
m	= Systemkontrollvariable (1010, 2130, 4055, 4065, 4070, 4205)
pl	= Spieleranzahl (53, 1012, 1025, 3005, 3105)
plp	= Geldanzeigeposition (1010, 1025, 3010, 3015, 3110, 3115, 3305)
ple	= geld 1. Spieler (1010, 1012, 3010, 3115)
plz	= geld 2. Spieler (1010, 3015, 3110)
s	= Anzahl der Serienspiele (2950-2970, 3530, 3535, 4340, 4355, 4380)
v	= Zufallsvariable zum Stop der Walzendrehung (6010, 6030)

### Liste der Unterprogramme

1400	= Starttaste
1420	= Nachstarttaste
2000-2055	= Gewinnabstastung Aufruf der betreffenden Unterprogramme bei Gewinn oder Verlust.
2060-2070	= Geld -10
2100-2220	= Gewinnmarkierung (für Jackpot -100) und Gewinnermittlung Aufruf des Gambling (3500)
2900-2930	= 1 Gewinnstufe weniger
2948-2972	= Gewinnbank in Form kleiner Unterprogramme
3000-3015	= Finanzregelung für zwei Spieler
3100-3120	= Spielerwechsel
3200-3205	= Wechselanzeige
3210-3225	= Highscoreanzeige
3300	= Geldausdruck
3500-3540	= Gambling
4000-4075	= JP-Verwaltung (= Anzeigeaufruf der Farben und Symbole)
4100-4130	= -100 Abzug vom Jackpot
4200-4210	= Prüfung, ob Jackpot komplett
4300-4365	= JP-Auszahlung
4370-4390	= Prüfung auf »Groschen« oder »-1« während der Serienspiele
5000-5035	= Spielende
5100-5035	= Melodien
6000-6065	= Walzensteuerung
6500-6580	= Walzendefinition
7000-7065	= Gewinnleistenaufwurf
7070-7090	= Spieleranzeige
7100-7160	= Karierte Felder
7170-7200	= Jackpotanzeige
7210-7230	= Farben- und Symbolanzeige
7240-7250	= Groschenanzeige
8000-8120	= UDG

# Hier sind Sinclair-Clubs, die Ihnen Hilfe anbieten

## SBC Sinclair Benutzer Club

Erika Hölscher  
Ernst-August-Str. 5  
2730 Zeven  
Tel. 04281/6442

## Spectrum-User-Club Heidenheim

Brenzstraße 3  
7922 Herbrechtingen 1

## Sinclair Spectrum User Club

Thorsten Hock  
Elbestr. 101  
2850 Bremerhaven

## Spectrum-User-Club

Rolf Knorre  
Postfach 200 102  
5600 Wuppertal 2  
Tel. 0202/556689

## Spectrum-Club Hagen

Gerd Sonnenhol  
Lönsweg 19  
5800 Hagen 1

## ZX-Spectrum-Userclub

Grünstr. 5a  
4920 Lemgo 1

## Sinclair-Mikroclub

Am Porzhain 3  
5908 Neunkirchen/Zeppenfeld

## ZX-Userclub

Friedenstr. 44  
4000 Düsseldorf 1

## ZX-Club Deutschland und Umgebung

Aribert Deckers  
Postfach 967  
7000 Stuttgart 1  
Tel. 0711/225314

Da die Leistungsangebote der Clubs genauso verschieden sind wie deren Organisation, sollten Sie sich ruhig bei mehreren Clubs erkundigen.

Auch ist diese Clubliste sicherlich nicht vollständig. Deshalb: Clubleitungen, bitte melden.

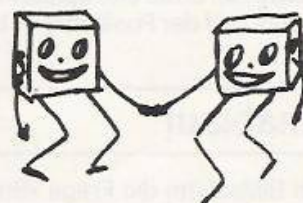
(mk)

## WITTICH

\*\*\*\*\* Der unglaubliche Versand \*\*\*\*\*  
Tel. 09443/453

Spectrum 48 K	469,-
Microdrive	189,-
Interface 1	189,-
Floppy-Controller	499,-
Diskettenlaufwerk	ab 499,-
Speichererweiterung 48 K	89,-
Profi-Tastatur	189,-
Fuller-Tastatur	239,-
Joystick	39,-
Joystick-Interface	39,-
Seikosha GP 500 A	599,-
Seikosha GP 550 A	799,-
Seikosha GP 50 S / 50 A	379,-
Druckerinterface	179,-
Commodore 64	649,-
Amstrad (Schneider)	949,-
Casio PB 700	389,-
Sinclair QL	1799,-

Computerversand Wittich  
Tulpenstr. 16, 8423 Abensberg, Tel. 09443/453



## ROLF STRECKER

Elektronik & Computer  
Vertrieb  
Luxemburgerstr. 76  
5000 Köln 1  
Tel.: (02 21) 41 77 89

## Multifile

Superadreiberverwaltung zum Erstellen von Serienbriefen  
mit Anrede + zahlreichen Sortierkriterien 49,-

## Mono

Leistungsstarker Macroassembler/Disassembler 59,-

## Disass

Disassembler mit Anzeige in HEX, DEZ., Memmonics 30,-

# Spectrum ZX81

Über 280 Artikel an  
Zubehör und  
Programmen!!!  
Katalog gegen Über-  
sendung von DM 3,80  
in Briefmarken erhältlich.

Händler-Anfragen erwünscht

## Stecker von Decker

Pf. 967, 7000 Stuttgart 1, 07 11/2253 14

	Stecker	Gehäuse
VC 20	Userport	9,85 4,15
VC 64	IEC-Port	9,85 4,15
CBM	Kassette	9,85 5,00
	Controlport	9,85 7,65
PET	80pin. Bus	25,-
VC 20	Audio/Video	3,50 (mit Geh.)
	Serial/E/A	3,50 (mit Geh.)
ATARI	13polig	15,- (mit Geh.)
	Gameport	9,85 7,65
Osborne	Modem-Port	9,85 7,65
TI 99/4	Joy-Port	9,85 7,65
	Kassette	9,85 weibl. 7,65
Telespiel-Stecker		9,85 7,65
		9,85 mann. 7,65

Busstecker	weiblich	männlich
APPLE II	12,-	NEU 18,-
VC 20	15,-	5,-
VC 64	18,-	5,-
ATARI 600XL/800XL	12,-	NEU 16,-
TI 99/4	16,-	5,-
Dragon-32	16,30	5,-
ZX 80/81 Lötöse	12,-	—
ZX 80/81 Print	8,50	—
ZX 80/81 Wire-Wrap	10,-	3,70
Spectrum Wire-Wrap	14,70	5,-

## Decker & Computer

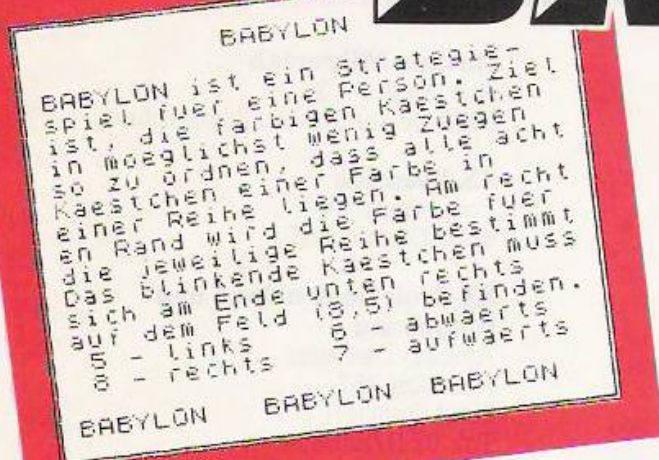
Pf. 967, 7000 Stuttgart 1, 07 11/2253 14

	für ZX80/81	für ZX-Spectrum
16 K-RAM-Platine	30,-*	—
(auf 48 K bzw. 56 K umbaubar)		
Autorepeat (fertig)	15,-*	—
Keyboard-Beeper	5,-*	—
Platine, leer	30,-*	—
Platine, fertig	30,-*	—
8 Bit Port-TTL Technik		
Platine, leer	30,-*	—
Platine, fertig	80,-*	—
Load Controller (leer)	5,-*	—
PIO Karte: leer	30,-*	30,-*
Bausatz	60,-*	60,-*
fertig	100,-*	100,-*
Video-Ausgang fertig	15,-*	15,-*
fertig, mit Kabel	30,-*	30,-*
Video Monitor	315,-	315,-
BASIC als EPROM	50,-	100,-
Busprobe: leer	40,-	40,-
fertig	150,-*	150,-*
Busplatine f. VG Leisten	30,-*	30,-*
SLOW-Modus f. ZX 80	38,-*	Restbestand!

Artikel mit einem \* Stern: bei Vorkasse erfolgt  
Lieferung im Inland spesenfrei

Autostart	Neu für ZX80/81	EPROM-Programmer
Leerplatine	35,-	Leerplatine 70,-
Bausatz o. EPROM	60,-	ZX-Hardwarebuch 40,-
		Fertigplatine 200,-

# BABYLON



Farbchaos am Bildschirm. Mit Überlegung und dem 16-KByte-Spectrum müssen Sie Ordnung schaffen. Mit seiner Hi-Score-Tabelle ist das Spiel »Babylon« ein toller »Abendfüller«.

Babylon ist ein Strategiespiel für eine Person. Das Ziel ist, farbige Kästchen am Bildschirm in eine bestimmte Ordnung zu bringen: Alle acht Felder einer Farbe müssen nebeneinander liegen; die Farbe für die jeweilige Reihe wird an der Seite bestimmt. Man ordnet die Kästchen mit Hilfe eines blinkenden Feldes, das in vier Richtungen bewegt werden kann. Versetzt man es zum Beispiel um eins nach rechts, dann befindet sich an der ursprünglichen Position des blinkenden Feldes die Farbe des Feldes rechts davon. Auf diese Weise kann man die Lage der farbigen Kästchen verändern. Am Ende des Spiels muß sich das blinkende Feld unten rechts (auf der Position 8,5) befinden.

## Programmablauf

Nach RUN sieht man auf dem Bildschirm die Frage »Brauchen Sie eine Anleitung? (J/N)«. Wenn man »J« eingibt, erscheint die Anleitung. Ein Druck auf irgendeine Taste genügt, um wieder zum Titelbild zurückzukehren. Drückt man nun auf »N«, dann wird der Bildschirm gelöscht und das Spielfeld aufgebaut. Nach kurzer Zeit wird man aufgefordert, eine Taste zu drücken, und das Spiel beginnt. Man steuert das blinkende Kästchen entweder mit den Tasten 5 — links, 6 — nach unten, 7 — nach oben und 8 — rechts oder mit einem Kempston-Joystick. Die Zahl der Versuche wird zu jeder Zeit im unteren Teil des Bildschirms angezeigt. Das Programm überprüft nach jedem Zug, ob man die Kästchen in die richtige Ordnung gebracht hat und beendet das Spiel, falls man es geschafft hat. Daraufhin erscheint entweder sofort die Hi-Score-Tabelle oder die Aufforderung, einen Namen einzugeben, je nachdem, wieviele Züge man benötigt hat. (Am Anfang des Spiels dürfen es höchstens 250 sein, um einen der fünf besten Plätze zu erreichen.) Sobald die Hi-Score-Tabelle auf dem Bildschirm zu sehen ist, erhält man durch den Druck einer Taste wieder das Titelbild.

Da das Programm für einen 16-KByte-Spectrum geschrieben ist, kann es Probleme mit dem Speicherplatz geben, wenn man versucht, die Anleitung zu lesen (es erscheint die Fehlermeldung »Out of memory«). In diesem Fall muß man alle REM-Zeilen und eventuell selbst eingefügte Zeilen entfernen.

Wenn nur ein Schwarzweiß-Fernseher zur Verfügung steht, gibt es die Möglichkeit, die Farben in Zeile 2050 zu ändern. In dieser Zeile steht jede Zahl für eine Farbe. Brauchbar ist die Kombination 0, 2, 3, 5 und 7.

### Erläuterung der einzelnen Programmteile

1- 40	In diesen Zeilen werden zwei Maschinenprogramme in den Speicher geladen. (32700 bis 32756)
50- 110	Dimensionierung der Felder und Definition von wichtigen Variablen
120- 180	Titelbild
200- 220	Spielfeld wird gezeichnet
2000-2190	Bestimmung der Farben
4020-4210	Verarbeitung der Eingaben während des Spiels
7000-7030	Ende des Spiels
8000-8260	Eingabe beziehungsweise Ausdruck der Hi-Scores auf dem Bildschirm
9000-9150	Anleitung

### Liste der Variablen

h\$ (6,15)	enthält die Namen und Punktzahlen der fünf besten Spieler
b (5,8)	enthält die Farbe des Feldes (x,y)
a (5)	enthält die Farben der seitlichen Felder
al, bl	enthalten die Position des blinkenden Feldes
v	gibt die Zahl der Versuche an
a, b, c, f, q, y, z und t\$	dienen als Steuervariablen

## Das Listing

Das Listing besteht aus zwei Routinen:

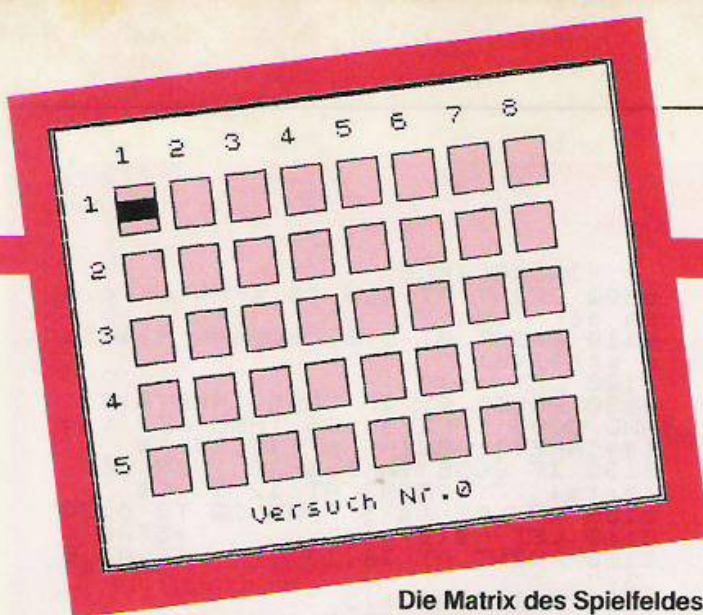
1. 9C40 — 9C4F Routine für Bordereffekte

Im Programm Babylon steht diese Routine von 32700 bis 32715. Sie dient dazu, das Programm durch Bordergrafiken interessanter zu gestalten. Man erhält mehr Effekte als die im Programm verwendeten, indem man den Wert des B-Registers verändert. Für den Benutzer von Babylon bedeutet das:

POKE 32701, x : PAUSE 10 : RANDOMIZE USR 32700  
(x ist eine Zahl zwischen 0 und 255).

2. 9C50 — 9C77 Routine zur Überprüfung der Lage der Felder

Diese Routine nimmt die Farbe eines seitlichen Feldes und vergleicht sie mit den Farben der übrigen Felder dieser Reihe. Wenn die Farben in allen Reihen richtig geordnet wurden, wird die Lage des blinkenden Feldes untersucht, und falls es sich



Die Matrix des Spielfeldes

unten ganz rechts befindet, lädt die Routine eine Zahl, die ungleich Null ist, in die Speicherstelle 32760. Diese wird dann durch das Basic-Programm abgefragt.

Im Programm Babylon steht die Routine von 32716 bis 32755.

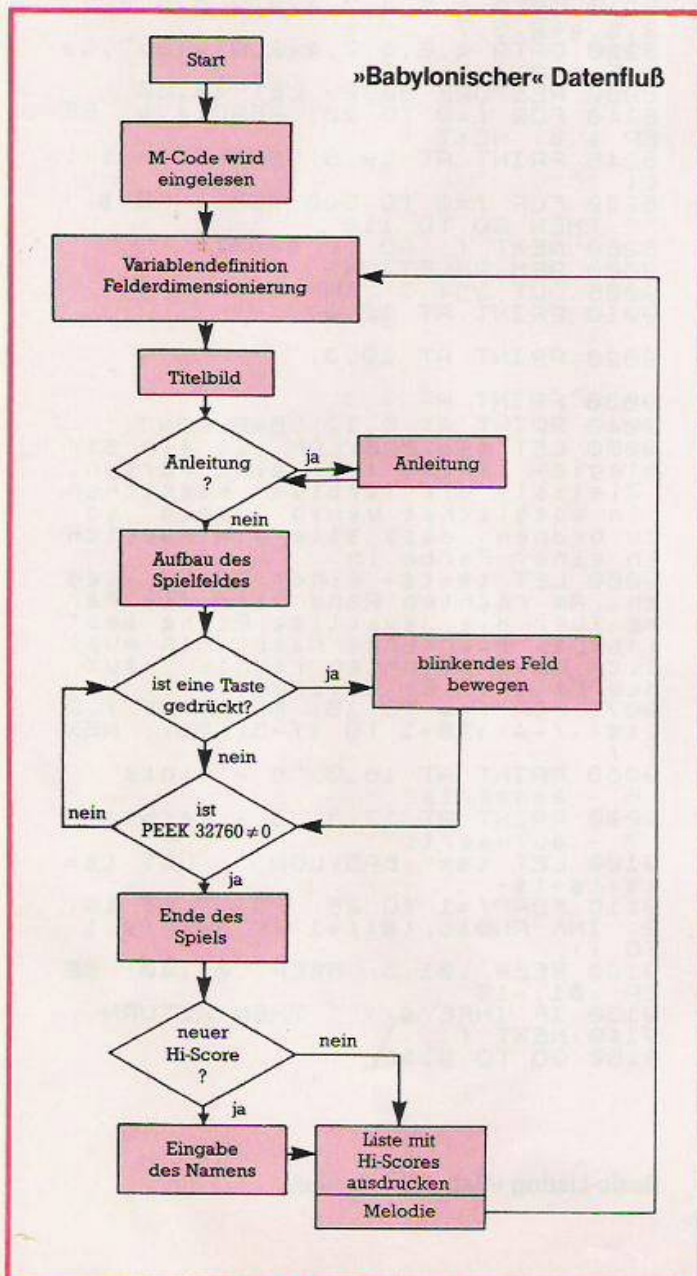
(Eva Kirchner)

```

1 REM BABYLON by EVA KIRCHNER
2 NEUENHAINERSTRASSE 8
3 REM 5000 FRANKFURT AM MAIN
4 CLEAR 32699
5
10 FOR f=32700 TO 32755: READ
a: POKE f,a: NEXT f
20 DATA 6,6,120,211,254,5,32,2
50,237,120,47,230,31,40,241,201
40 DATA 33,156,88,126,230,56,9
5,22,8,45,45,45,126,230,56,171,1
90,21,32,245,1,120,0,9,62,124,17
30,32,230,58,26,90,203,127,200,23
7,67,248,127,201
50 DIM h$(6,15): FOR f=1 TO 5:
LET h$(f)="FRED 250": NEXT
f
100 REM Menue
110 DIM b(5,8): DIM a(5): LET a
1=1: LET b1=a1: LET v=0
120 RANDOMIZE : POKE 32760,v
130 INVERSE v: FLASH v: BRIGHT
a1: BORDER a1: PAPER 7: INK a1:
CLS
140 PRINT AT 4,12;"BABYLON"
150 PLOT 8,8: DRAW 239,v: DRAW
v,159: DRAW -239,v: DRAW v,-159:
PLOT 6,6: DRAW 243,v: DRAW v,16
3: DRAW -243,v: DRAW v,-163: PRI
NT AT 10,3;"BRAUCHEN SIE EINE AN
LEITUNG";AT 12,13;"(J/N)"
150 POKE 32701,8: RANDOMIZE USR
32700: IF INKEY$="j" OR INKEY$=
"J" THEN GO SUB 9000: GO TO 100
170 IF INKEY$="n" OR INKEY$="N"
THEN GO TO 200
190 REM SPIELFELD
200 PRINT AT 19,9: FLASH a1;"BI
TTE WARTEN": PRINT AT 4,10;"
":AT 10,3;"
":AT 12,13;"
"
205 FOR f=v TO 96 STEP 24: FOR
a=v TO 168 STEP 24: PLOT 31+a,31
+f: DRAW 17,v: DRAW v,17: DRAW -
17,v: DRAW v,-17: NEXT a: NEXT f
210 FOR f=1 TO 5: PRINT AT f*3+
1,2:f: NEXT f
220 FOR f=1 TO 8: PRINT AT 2,f*
3+1:f: NEXT f
2000 REM FARBEN
2010 RANDOMIZE
2020 RESTORE 2050
2030 FOR a=0 TO 4
2040 READ z: LET y=z
2050 DATA 1,3,2,5,6
2060 FOR f=0 TO 7
2070 LET c=INT (RND*5)*3+4
2080 LET b=INT (RND*8)*3+4
2090 IF ATTR (c,b)=121 THEN PRIN
T AT c,b: PAPER z:" ":AT c+1,b:
PAPER y:" ": NEXT f
2100 IF f<8 THEN GO TO 2070
2110 NEXT a
2120 LET z=(PEEK 22660-65)/8
2130 PRINT AT 4,4: FLASH 1: PAPE
R z: INK 7:"-":AT 5,4:"-":
2140 FOR a=1 TO 5: FOR b=1 TO 8:
LET b(a,b)=ATTR ((a-1)*3+4,(b-1
)*3+4): NEXT b: NEXT a
2150 RESTORE 2050: FOR f=1 TO 5:
READ y: LET a(f)=y: NEXT f: FOR
f=1 TO 5: IF a(f)=z THEN LET a(
f)=a(5): LET a(5)=z: GO TO 2170
2160 NEXT f
2170 FOR f=4 TO 16 STEP 3: PRINT
AT f,28: PAPER a((f-1)/3):" ":A
T f+1,28:" ": NEXT f
2175 PRINT AT 19,6;"EINE TASTE D
RUECKEN"

```

Basic-Listing »Babylon«



```

2180 POKE 32701,42: RANDOMIZE US
R 32700
2190 IF INKEY$="" THEN GO TO 219
0
4000 REM SPIEL
4020 PRINT AT 19,6;" Versuch N
r.:v:" ": IF IN 31=1 OR INKE
Y$="8" THEN LET v=v+1: BEEP .01,
10: GO SUB 4100
4030 IF INKEY$="7" OR IN 31=8 TH
EN LET v=v+1: BEEP .01,20: GO SU
B 4130
4040 IF INKEY$="6" OR IN 31=4 TH
EN LET v=v+1: BEEP .01,30: GO SU
B 4150
4050 IF INKEY$="5" OR IN 31=2 TH
EN LET v=v+1: BEEP .01,40: GO SU
B 4190
4060 RANDOMIZE USR 32716: IF PEE
K 32750<>0 THEN GO TO 7000
4070 GO TO 4020
4100 IF b1=8 THEN RETURN
4110 POKE 23693,b(a1,b1+1): PRIN
T AT a1*3+1,b1*3+1;" ":AT a1*3+
2,b1*3+1;" ": POKE 23693,z*8+19
9: PRINT AT a1*3+1,b1*3+4;" "":A
T a1*3+2,b1*3+4;" "":
4120 LET b(a1,b1)=b(a1,b1+1): LE
T b(a1,b1+1)=254: POKE 23693,120
: LET b1=b1+1: RETURN
4130 IF a1=1 THEN RETURN
4140 POKE 23693,b(a1-1,b1): PRIN
T AT a1*3+1,b1*3+1;" ":AT a1*3+
2,b1*3+1;" ": POKE 23693,z*8+19
9: PRINT AT a1*3-2,b1*3+1;" "":A
T a1*3-1,b1*3+1;" "":
4150 LET b(a1,b1)=b(a1-1,b1): LE
T b(a1-1,b1)=254: POKE 23693,120
: LET a1=a1-1: RETURN
4160 IF a1=5 THEN RETURN
4170 POKE 23693,b(a1+1,b1): PRIN
T AT a1*3+1,b1*3+1;" ":AT a1*3+
2,b1*3+1;" ": POKE 23693,z*8+19
9: PRINT AT a1*3+4,b1*3+1;" "":A
T a1*3+5,b1*3+1;" "":
4180 LET b(a1,b1)=b(a1+1,b1): LE
T b(a1+1,b1)=254: POKE 23693,120
: LET a1=a1+1: RETURN
4190 IF b1=1 THEN RETURN
4200 POKE 23693,b(a1,b1-1): PRIN
T AT a1*3+1,b1*3+1;" ":AT a1*3+
2,b1*3+1;" ": POKE 23693,z*8+19
9: PRINT AT a1*3+1,b1*3-2;" "":A
T a1*3+2,b1*3-2;" "":
4210 LET b(a1,b1)=b(a1,b1-1): LE
T b(a1,b1-1)=254: POKE 23693,120
: LET b1=b1-1: RETURN
6999 REM Spiel ist zu Ende
7000 PAUSE 10: PRINT AT 19,6;"Ei
ne Taste druecken": RANDOMIZE US
R 32700
7005 BORDER 0: FOR f=0 TO 21: PR
INT AT f,0; PAPER 0;" ": PAUSE 2:
NEXT f
7010 PAUSE 5
7030 FOR f=21 TO 0 STEP -1: PRIN
T AT f,0; PAPER 1;" ": PAUSE 2: N
EXT f
8000 REM HIGH SCORE TABLE
8030 INK 6: PAPER 1: BRIGHT 1: C
LS
8040 PLOT 8,8: DRAW 239,0: DRAW
0,159: DRAW -239,0: DRAW 0,-159:
PLOT 6,6: DRAW 243,0: DRAW 0,16
3: DRAW -243,0: DRAW 0,-163
8050 IF v>VAL (h$(5) (11 TO )) TH
EN GO TO 8190
8060 PRINT AT 5,3;"Geben Sie ihr
en Namen ein."
8070 PRINT AT 12,13;"_____"
8080 PRINT AT 2,11; INK 5; FLASH

```

```

1;"INCREDIBLE"
8090 PRINT AT 4,3;"Sie haben ein
en der fuerf"
8110 PRINT AT 5,3;"besten Plaetz
e erreicht."
8120 LET t$=""
8130 PAUSE 0: IF CODE INKEY$>31
AND CODE INKEY$<128 THEN LET t$=
t$+INKEY$: PRINT AT 12,13;t$
8132 IF CODE INKEY$=13 THEN LET
t$=t$+" ": PRINT AT 12,13;t$
8135 IF LEN t$<5 THEN GO TO 8130
8140 LET h$(6)=t$+" "+STR$ v
8150 PRINT AT 12,13;" "":AT 6
,3;"
8160 PRINT AT 4,3;"
":AT 5,3;"
":AT 2,10;"
"
8170 FOR f=5 TO 1 STEP -1: IF UA
L (h$(f+1) (11 TO ))<=VAL (h$(f) (
11 TO )) THEN LET t$=h$(f): LET
h$(f)=h$(f+1): LET h$(f+1)=t$
8180 NEXT f
8190 FOR f=1 TO 5: PRINT AT f*3+
1,9:h$(f): NEXT f
8200 DATA q,2,q,4,2*q,5,2*q,9,q/
2,9,q/2,7,q/2,5,q*2,5,7
8210 DATA q,5,q,7,q/2,9,q/2,7,q/
2,5,q*2,5,7
8220 DATA q,5,q,7,q*2,9,q*2,7,q*
2,5,q*2,4,q*4,2
8230 RESTORE 8030: LET q=.30
8240 FOR f=0 TO 20: READ a,b: BE
EP a,b: NEXT f
8245 PRINT AT 19,8;"Noch ein Spi
el?"
8250 FOR f=0 TO 500: IF INKEY$<>
"" THEN GO TO 110
8260 NEXT f: GO TO 8230
9000 REM ANLEITUNG
9005 OUT 254,3: INK 3
9010 PRINT AT 12,3;"
"
9020 PRINT AT 10,3;"
"
9030 PRINT AT 4,3;"
"
9040 PRINT AT 2,12;"BABYLON"
9050 LET t$="BABYLON ist ein Str
ategie- spiel fuer eine Person.
Ziel ist, die farbigen Kaestchen
in moeglichst wenig Zuegen so
zu ordnen, dass alle achtKaestch
en einer Farbe in
"
9060 LET t$=t$+"einer Reihe lieg
en. Am rechten Rand wird die Far
be fuer die jeweilige Reihe best
immt.Das blinkende Kaestchen muss
sich am Ende unten rechts auf
dem Feld (8,5) befinden."
9070 FOR f=4 TO 15: PRINT AT f,2
;t$(f-4)*28+1 TO (f-3)*28): NEX
T f
9080 PRINT AT 16,3;"5 - links
5 - abwaerts"
9090 PRINT AT 17,3;"8 - rechts
7 - aufwaerts"
9100 LET t$="BABYLON ": LET t$=
t$+t$+t$+"
"
9110 FOR f=1 TO 28: PRINT AT 19,
2; INK RND*6;t$(f+1 TO 28);t$(1
TO f)
9120 BEEP .01,3: BEEP .01,40: BE
EP .01,-15
9130 IF INKEY$<>"" THEN RETURN
9140 NEXT f
9150 GO TO 9110

```

Basic-Listing »Babylon« (Schluß)

# Eine knifflige Sache

**Kniffeln ohne Schummeln, ohne Bleistift, ohne Zettel und Verrechnen: Ihr 48-KByte-Spectrum sorgt für einen reibungslosen Spielablauf.**

Ziel des Spieles ist es, wie sollte es anders sein, am Schluß die meisten Punkte zu haben. Dazu werfen Sie gleichzeitig fünf Würfel. Gefällt einem der Wurf nicht, so kann der Spieler bestimmte Würfel oder alle neu fallen lassen. Dies geschieht durch Drücken der Würfelnummer und »N«. Spätestens nach dem dritten Wurf ist jedoch die Wertung vorzunehmen. Soll kein Würfel neu fallen, dann statt »N« nur »O« tasten und die Wertung eingeben.

Die Eingabe der Wertung erfolgt über die Tasten 1 bis D, die Bewertungskriterien erläutert das Programm am Anfang. Jeder Spieler führt 13 Wertungen aus, dann erfolgt die »Abrechnung« und Namenszuteilung mit »Ehrung«.

(Josef Kastl)

## Erläuterung der Variablen

Löschen des Bildschirms: Zeile 40

z	Zeilenzahl, die gelöscht werden soll
a	Koordinaten der ersten zu löschenden Zeile
c	x-Position des Würfels beim Zeichnen/Zählvariable
k	Würfelnummer
r	Zwischenwurf (1 kompletter Wurf besteht maximal aus drei Würfeln, das heißt es kann dreimal gewürfelt werden, bevor gewertet werden muß)
n	Initiale des Würfels, der nochmal fallen soll
u	Anstreichen des Würfels, der nochmal fallen soll
spi	Anzahl der Spieler
aspi	Aktueller Spieler
wurf	Anzahl der Würfe: erhöht sich bei jedem Spieler bis maximal 13 X aspi, da jeder Spieler die 13 Wertungen ausführen muß
p	Punkte aus der Wertung
o	Zeilenkoordinate der gewählten Wertung
xx, ttt	Parity-Variable als Sprungkriterien
h	Zählvariable
sss	Spielzähler
w	Spaltenkoordinate (abhängig von aspi)
a\$( )	Spielernamen
b\$( )	Spieler, der im Moment spielt
w(5)	Feld für die fünf Würfel zum Festhalten der Würfelaugen
g(4)	Gesamtsumme, pro Spieler eine
f( )	In diesem Feld wird für jede ausgeführte Wertung für jeden Spieler eine 1 gesetzt. Damit wird am Anfang jeder Wertung geprüft, ob diese schon mal ausgeführt wurde!
k( ), l( )	Felder, die bei der Wertung »Voll-Haus« verwendet werden
b(4)	Bonusfeld; für jeden Spieler wird das Array=35 gesetzt, wenn dieser den Bonus erhält
z(4)	Zwischensumme, pro Spieler eine
o(4)	Austauschfeld: je nachdem, ob Einzelspielwertung oder Rangliste, werden entsprechende Felder hierin gewechselt
t(4)	Punktezahle je Spieler in der Totalwertung
s\$	Name bei Siegerehrung

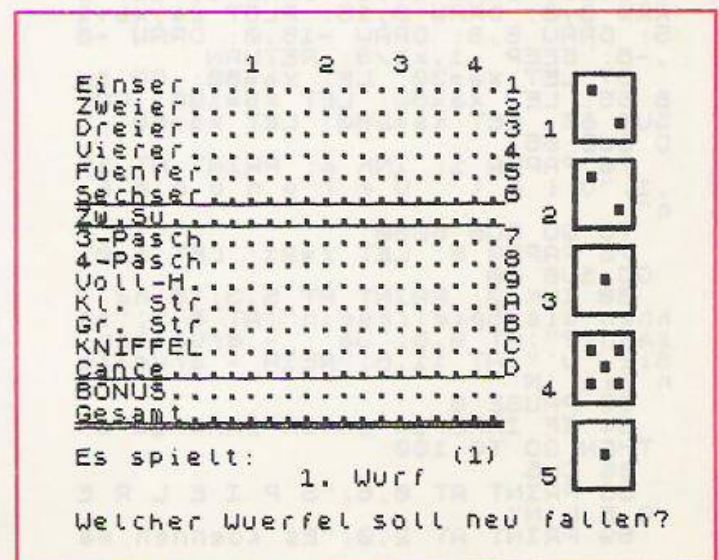
## Variablen-tabelle



Das Programm meldet sich startbereit



Fertig zum ersten Wurf



Kein toller Wurf





```

zahl ist"; AT 19,5;"auf 4 begren
zt."
132 INPUT "Wieviel Spieler? ";s
pi
135 IF spi>4 THEN FLASH 1: PRIN
T AT 18,5;"Es koennen leider nur
4"; AT 19,5;"Personen mitspielen
": PAUSE 150: FLASH 0: LET spi=
4
140 FOR x=14 TO 19: PRINT AT x,
5;
NEXT x
150 PRINT AT 13,3;"Mitspieler:"
160 DIM a$(4,9)
162 FOR x=1 TO spi
165 INPUT "Name: ";a$(x)
167 PRINT AT 11+2*x,15;a$(x)
169 NEXT x
171 BEEP .3,3: BEEP .3,7: BEEP
.3,10: BEEP .3,15
193 REM
200 REM      X X X X PUNKTETABELL
E X X X X
210 PAPER 4: LET z=21: LET a=0:
GO SUB 40
230 FOR x=1 TO 16
240 PRINT AT x,5;".....
"
250 NEXT x
270 PRINT AT 0,9;"1"; AT 0,13;"2
"; AT 0,17;"3"; AT 0,21;"4"
280 PRINT AT 1,0;"Einser": PRIN
T "Zweier": PRINT "Dreier": PRIN
T "Vierer": PRINT "Fuenfer": PRI
NT "sechser": PRINT "Zw.Su."
290 PRINT "3-Pasch": PRINT "4-P
asch": PRINT "Voll-H.": PRINT "K
l. Str.": PRINT "Gr. Str.": PRIN
T "KNIFFEL": PRINT "Cance"
300 PRINT "BONUS": PRINT "Gesam
t"
310 PAPER 3: PRINT AT 1,23;"1";
AT 2,23;"2"; AT 3,23;"3"; AT 4,23;
"4"; AT 5,23;"5"; AT 6,23;"6"; AT 8
,23;"7"; AT 9,23;"8"; AT 9,23; AT 1
0,23;"9"; AT 11,23;"A"; AT 12,23;"
B"; AT 13,23;"C"; AT 14,23;"D"
320 FOR x=1 TO 16: PAPER 6: PRI
NT AT x,8;"..."; AT x,12;"..."; AT
x,16;"..."; AT x,20;"...": NEXT
x
330 PLOT 0,120: DRAW 184,0: PLO
T 0,112: DRAW 184,0: PLOT 0,56:
DRAW 184,0
340 PLOT 0,40: DRAW 184,0: PLOT
0,38: DRAW 184,0: IF xx=1 THEN
RETURN
350 PAPER 4: PRINT AT 18,0;"Es
spielt:"
499 REM
500 REM      X X X X X SPIELABLAU
F X X X X X
501 REM
510 LET aspi=aspi+1
515 IF aspi>spi THEN LET aspi=1
520 LET wurf=wurf+1
530 IF wurf>13*spi THEN GO TO 4
000
540 PAPER 7: PRINT AT 18,11;a$(
aspi): PRINT AT 18,20;"(";aspi;"
)"
550 PAPER 4: PRINT AT 21,0;"Bit
te wuerfeln (beliebige Taste)"
555 BEEP .5,10: PAUSE 0
560 REM      X X X Die 3 Wuerfe x
x x
562 DIM w(5)
565 FOR x=1 TO 5: LET w(x)=0: N
EXT x
570 FOR r=1 TO 3
575 INK 7: PAPER 2: PRINT AT 19
,11;" ";r;" Wurf ": PAPER 4: IN
K 0
580 FOR k=1 TO 5
585 IF w(k)<>0 THEN GO TO 600
590 LET w(k)=INT (RND*6)+1
600 GO SUB 1000
610 NEXT k
620 IF r=3 THEN GO TO 690
640 FOR x=1 TO 6
645 LET n=0
650 PAPER 4: PRINT AT 21,0;"Wel
cher Wuerfel soll neu fallen?"
652 PAUSE 0
655 IF INKEY$="1" THEN LET n=1
656 IF INKEY$="2" THEN LET n=2
657 IF INKEY$="3" THEN LET n=3
658 IF INKEY$="4" THEN LET n=4
659 IF INKEY$="5" THEN LET n=5
660 IF INKEY$="0" THEN GO TO 69
0
661 IF INKEY$="n" OR INKEY$="N"
THEN GO TO 675
663 IF n<>1 AND n<>2 AND n<>3 A
ND n<>4 AND n<>5 THEN PRINT AT 2
1,0;"Sie haben falsche Taste gew
aehlt": BEEP 1,3: GO TO 650
665 LET w(n)=0
667 LET u=(n-2/3)*32
668 INK 1: PLOT 248,177-u: DRAW
0,-22: PLOT 249,177-u: DRAW 0,-
22: PLOT 250,177-u: DRAW 0,-22:
INK 0
670 NEXT x
673 IF r=3 THEN GO TO 695
675 PAPER 4: FOR x=0 TO 20: PRI
NT AT x,25;"      ": NEXT x
676 IF xx=1 THEN RETURN
680 NEXT r
690 LET o=0: LET p=0
695 PAPER 4: PRINT AT 21,0;"Was
wollen Sie werten?
697 FOR x=1 TO 3: BEEP .1,8: BE
EP .1,14: NEXT x: PAUSE 0
700 IF INKEY$="1" THEN LET o=1:
GO SUB 3100
705 IF INKEY$="2" THEN LET o=2:
GO SUB 3100
710 IF INKEY$="3" THEN LET o=3:
GO SUB 3100
715 IF INKEY$="4" THEN LET o=4:
GO SUB 3100
720 IF INKEY$="5" THEN LET o=5:
GO SUB 3100
725 IF INKEY$="6" THEN LET o=6:
GO SUB 3100
730 IF INKEY$="7" THEN LET o=7:
GO SUB 3200
735 IF INKEY$="8" THEN LET o=8:
GO SUB 3300
740 IF INKEY$="9" THEN LET o=9:
GO SUB 3400
745 IF INKEY$="A" OR INKEY$="a"
THEN LET o=10: GO SUB 3500
750 IF INKEY$="B" OR INKEY$="b"
THEN LET o=11: GO SUB 3600
755 IF INKEY$="C" OR INKEY$="c"
THEN LET o=12: GO SUB 3700
760 IF INKEY$="D" OR INKEY$="d"
THEN LET o=13: GO SUB 3800
765 IF o=0 OR o>13 THEN PRINT A
T 21,0;"Sie haben falsche Taste
gewaehlt": BEEP 1,3: GO TO 695
770 REM      X X X Punkte schreib
en x x x
771 LET f(o+aspi*15)=1
772 LET w=6+4*aspi
774 IF o>6 THEN LET o=o+1
776 IF p>9 THEN LET w=w-1
778 PAPER 6: PRINT AT o,w;p
780 IF o>6 THEN GO TO 850
800 REM      X X Zwischensumme er
hoehen x x
810 LET z(aspi)=z(aspi)+p
815 LET w=6+4*aspi
820 IF z(aspi)>9 THEN LET w=w-1
825 IF z(aspi)>99 THEN LET w=w-
1
830 PAPER 6: PRINT AT 7,w;z(asp
i)

```

Basic-Listing »Kniffel« (Fortsetzung)

```

832 PLOT 0,112: DRAW 184,0
835 IF z(asp1)<63 THEN GO TO 85
0
837 IF b(asp1)=35 THEN GO TO 85
0
840 PRINT AT 21,0;"Sie erhalten
einen ": FLASH 1: PRINT AT 21,1
9;"B O N U S !"
842 BEEP .2,10: BEEP .2,14: BEE
P .2,17: BEEP .2,15: BEEP .2,19:
BEEP .2,15: BEEP .2,19: BEEP .2,17: BEE
P .2,14: BEEP .2,10: FLASH 0
844 PRINT AT 15,6+4*asp1-1;"35"
: LET b(asp1)=35
846 LET p=p+35
850 REM X X Gesamtsumme erho
ehen X X
855 LET g(asp1)=g(asp1)+p
857 LET w=6+4*asp1
858 IF g(asp1)>9 THEN LET w=w-1
859 IF g(asp1)>99 THEN LET w=w-
1
860 PAPER 6: PRINT AT 16,w;g(as
Pi)
865 LET xx=1: GO SUB 330: LET x
x=0
890 LET xx=1: GO SUB 675: LET x
x=0
895 PRINT AT 19,11;" "
990 GO TO 500
999 REM
1000 REM X X X X X Zeichnen d
er WUERFEL X X X X X
1001 REM
1040 PAPER 7
1045 LET c=(k-1)*4+1
1048 PRINT AT c+2,25;k
1050 IF w(k)=1 THEN PRINT AT c,2
7;" " ;AT c+1,27;" A ";AT c+2,2
7;" "
1060 IF w(k)=2 THEN PRINT AT c,2
7;" D " ;AT c+1,27;" " ;AT c+2,2
7;" "
1070 IF w(k)=3 THEN PRINT AT c,2
7;" R " ;AT c+1,27;" B ";AT c+2,2
7;" "
1080 IF w(k)=4 THEN PRINT AT c,2
7;" D " ;AT c+1,27;" " ;AT c+2,2
7;" "
1090 IF w(k)=5 THEN PRINT AT c,2
7;" D " ;AT c+1,27;" B ";AT c+2,2
7;" "
1100 IF w(k)=6 THEN PRINT AT c,2
7;" D " ;AT c+1,27;" B R";AT c+2,2
7;" "
1120 PLOT 215,151-c*8: DRAW 0,24
: DRAW 24,0: DRAW 0,-24: DRAW -2
4,0
1122 PLOT 214,150-c*8: DRAW 0,26
: DRAW 26,0: DRAW 0,-26: DRAW -2
6,0
1125 BEEP .01,w(k)*3+20
1200 RETURN
2020 RETURN
2990 REM X X X X X WERTUNGEN
X X X X X
3000 REM X X Ueberpruefen, ob
gewaehlte Wertung noch frei ist
X X X
3010 IF f(0+asp1*15)=1 THEN GO T
O 3900
3020 RETURN
3100 REM X X X Einser bis Sec
hser X X X
3105 GO SUB 3000
3110 FOR x=1 TO 5
3120 IF w(x)=0 THEN LET p=p+0
3130 NEXT x
3140 RETURN
3200 REM X X X 3er-Pasch X X
X
3205 GO SUB 3000
3210 FOR x=1 TO 3
3215 LET h=0
3220 FOR y=x TO 4
3225 IF w(x)=w(y+1) THEN LET h=h
+1
3230 IF h=2 THEN GO TO 3250
3235 NEXT y: NEXT x
3240 LET p=0: RETURN
3250 LET p=w(1)+w(2)+w(3)+w(4)+w
(5)
3255 RETURN
3300 REM X X X 4er-Pasch X X
X
3305 GO SUB 3000
3310 FOR x=1 TO 2
3315 LET h=0
3320 FOR y=x TO 4
3325 IF w(x)=w(y+1) THEN LET h=h
+1
3330 IF h=3 THEN GO TO 3350
3335 NEXT y: NEXT x
3340 LET p=0: RETURN
3350 LET p=w(1)+w(2)+w(3)+w(4)+w
(5)
3355 RETURN
3400 REM X X X Voll-Haus X X
X
3405 GO SUB 3000
3410 FOR x=1 TO 3
3415 LET h=0
3420 FOR y=x TO 4
3425 IF w(x)=w(y+1) THEN LET h=h
+1: LET k(h)=x: LET l(h)=y+1
3430 IF h=2 THEN GO TO 3450
3435 NEXT y: NEXT x
3440 LET p=0: RETURN
3450 LET c=0: FOR x=1 TO 5
3455 IF k(1)=x OR l(1)=x OR l(2)
=x THEN GO TO 3470
3460 LET c=c+1: IF c=1 THEN LET
t=x: GO TO 3470
3465 IF w(t)<>w(x) THEN LET p=0:
RETURN
3470 NEXT x
3475 IF c<2 THEN LET p=0: RETURN
3480 LET p=25: RETURN
3500 REM X X X Kleine Strasse
X X X
3505 GO SUB 3000
3510 FOR x=1 TO 5
3515 IF w(x)=3 THEN GO TO 3525
3517 NEXT x
3520 GO TO 3590
3525 FOR x=1 TO 5
3530 IF w(x)=4 THEN GO TO 3540
3532 NEXT x
3535 GO TO 3590
3540 FOR x=1 TO 5
3545 IF w(x)=2 THEN GO TO 3555
3547 NEXT x
3550 GO TO 3565
3555 FOR x=1 TO 5
3560 IF w(x)=1 OR w(x)=5 THEN GO
TO 3595
3562 NEXT x
3565 FOR x=1 TO 5
3570 IF w(x)=5 THEN GO TO 3580
3572 NEXT x
3575 GO TO 3590
3580 FOR x=1 TO 5
3585 IF w(x)=6 THEN GO TO 3595
3587 NEXT x
3590 LET p=0: RETURN
3595 LET p=30: RETURN
3600 REM X X X Grosse Strasse
X X X
3605 GO SUB 3000
3610 FOR x=2 TO 5
3615 FOR y=1 TO 5
3620 IF w(y)=x THEN GO TO 3635
3625 NEXT y
3630 GO TO 3655
3635 NEXT x
3640 FOR y=1 TO 5
3645 IF w(y)=1 OR w(y)=6 THEN GO
TO 3660

```

```

3650 NEXT y
3655 LET p=0: RETURN
3660 LET p=40: RETURN
3700 REM      X X X Kniffel X X X
3705 GO SUB 3000
3710 LET x=w(1): IF w(2)=x AND w
(3)=x AND w(4)=x AND w(5)=x THEN
GO TO 3720
3715 LET p=0: RETURN
3720 LET p=50: RETURN
3800 REM      X X X (letzte) Chanc
e X X X
3805 GO SUB 3000
3810 LET p=w(1)+w(2)+w(3)+w(4)+
(5)
3820 RETURN
3900 PRINT AT 21,0;"Schon belegt
!!!
GO TO 695
4005 LET sss=sss+1: REM **Spielz
aehler**
4010 CLS : PAPER 4
4020 PRINT AT 1,2;".....
G E R E H R U N G " ; AT 3,2;"
4030 REM      x x Vergangenes Spie
l x x
4040 FOR x=1 TO spi: LET o(x)=g(x)
NEXT x: GO SUB 4500
4045 LET z=7: LET a=5: PAPER 6:
GO SUB 42
4050 PRINT AT 5,1;"Vergangenes S
piel:"
4060 LET aa=6: GO TO 4100
4070 REM      x x Alle Spiele (Tot
al) x x
4073 LET ttt=1
4075 FOR x=1 TO spi
4080 LET t(x)=t(x)+g(x)
4085 LET o(x)=t(x): NEXT x
4087 LET z=7: LET a=13: PAPER 5:
GO SUB 42
4090 PRINT AT 13,1;"RANGLISTE (n
ach ";sss;" Spielen):"
4095 LET aa=14
4100 GO SUB 4500
4105 IF ttt=1 THEN FOR x=1 TO 4:
LET g(x)=t(x): NEXT x
4110 FOR x=1 TO spi
4120 PRINT AT aa+x,2;x;" . Platz:"
4130 GO SUB 4300
4140 PRINT AT aa+x,12;ss$;" ";o(x)
;AT aa+x,25;"Pkte."
4150 NEXT x
4160 IF aa=6 THEN GO TO 4070
4180 LET d=.3: BEEP d,3: BEEP d,
5: BEEP d,7: BEEP d,5: BEEP d,8:
BEEP d,7: BEEP d,5: BEEP d,2: B
EEP d,3: BEEP d,12: BEEP d,10: B
EEP d,8: BEEP d,7: BEEP d,5: BEE
P d,7: BEEP d,3
4200 INK 0: PAPER 4: PRINT AT 21
,0;"Neues Spiel? (J/N)"
4210 PAUSE 0
4230 IF INKEY$="J" OR INKEY$="J"
THEN GO TO 13
4240 PRINT AT 21,0;" A U F W I
E D E R S E H E N"
4250 STOP
4300 REM      x Ermittlung des Spie
lernamens x
4310 FOR y=1 TO spi
4320 IF o(x)=g(y) THEN LET ss=a$(
y)
4330 NEXT y
4340 RETURN
4500 REM      x x Ordnen der Punkt
zahlen nach der Groesse x x
4510 LET x=0: LET x1=0
4520 LET x=x+1:
4530 IF o(x)=0 AND x1=0 THEN GO
TO 4570
4540 IF o(x)=0 THEN GO TO 4510
4550 LET y=x+1
4560 IF o(y)>o(x) THEN LET xx=o(x)
: LET o(x)=o(y): LET o(y)=xx:
LET x1=1
4562 GO TO 4520
4570 RETURN
5000 REM      X X X X X GRAFIKSOND
ERZEICHEN X X X X X
5005 RESTORE 5050
5010 FOR i=1 TO 7
5030 READ y$
5040 FOR x=0 TO 7: READ b: POKE
USR y$+x,b: NEXT x: NEXT i
5050 DATA "a",0,0,60,60,60,60,0,
0
5052 DATA "b",0,0,15,15,15,15,0,
0
5054 DATA "c",0,0,240,240,240,24
0,0,0
5056 DATA "d",0,0,0,0,15,15,15,1
5
5058 DATA "e",0,0,0,0,240,240,24
0,240
5060 DATA "f",15,15,15,15,0,0,0,
0
5062 DATA "g",240,240,240,240,0,
0,0,0
5070 RETURN
9000 REM SAVE - ROUTINE BAND
9010 SAVE "KNIFFEL" LINE 0
9020 VERIFY "KNIFFEL"
9030 STOP

```

Basic-Listing »Kniffel« (Schluß)

## Achtung! ZX-Spectrum User

Darauf haben Sie gewartet:  
Das **einzige** Joystick-Interface  
mit **durchgehendem Bus**.

**nur 85,- DM**

• bleibt immer am Spectrum • Bus durchgeführt - Anschluß bleibt erhalten • Peripherie wird am Interface angeschlossen, dadurch Schonung der empfindlichen Spectrum-Busleiste.

Arcade Joystick schwarz ..... **59,- DM**  
Competition Pro Joystick ..... **65,- DM**  
Arcade Professional Joystick ..... **139,- DM**  
Arcade Turbo Joystick ..... Lieferbar Ende Oktober

Alle Preise inkl. MwSt., Versand gegen Nachnahme oder Vorkasse (5,- DM Porto).  
Fordern Sie Infos an gegen Freiumschlag.

**Außenvertrieb in der BRD für EMAX- und SUZO-Produkte.**  
Fordern Sie unsere Händlerunterlagen an!

**EBR** Eckard Begerow  
Electronic- u. Computer-Zubehör  
VERTRIEB Postfach 30 · 8428 Rohr · Tel. 0 87 83/5 52

## M&T-Buchverlag



M. Gavin  
Astronomie-Programme für den  
ZX-Spectrum

September 1984, 268 Seiten  
Eine phantastische Reise in die Welt  
des Kosmos mit Ihrem ZX-Spectrum:  
Der Julianische Kalender · Die Mond-  
phasen · Eigene Satelliten starten · Ke-  
pler's Umlaufbahnen · Die Umlaufbahn  
Plutos · Interessant nicht nur für Hob-  
by-Astronome. **DM 32,80\***  
Best.-Nr. MT 732  
(\* DM 29,80 inkl. MwSt. zuzügl.  
DM 3,- Porto und Versand)

In guten Buchhandlungen,  
Computershops und Fachabteilungen  
der Kaufhäuser.

**Markt & Technik**  
Verlag Aktiengesellschaft  
Hans-Pinsel-Str. 2, 8013 Haar bei München

# Beets

**Helfen Sie Tante Emma in ihrem Laden beim Sortieren von Eiern und Gemüse. Benutzen Sie dazu Geschick, Glück und den 48-KByte-Spectrum.**

Nachdem Sie das Programm geladen haben, startet es von selbst. Zuerst erscheint das Titelbild. Gleichzeitig werden Sie aufgefordert, eine beliebige Taste zu drücken, um das Spiel zu beginnen. Nachdem das geschehen ist, zeichnet der Computer eine Art Regal, bestehend aus neun Reihen (durchnummeriert von 1 bis 9) und 14 Spalten.

Anschließend werden mittels RHD die einzelnen »Regalfächer« so verteilt, daß etwa die Hälfte der Fächer geschlossen und der Rest offen ist. Die oberste Reihe muß vor dem Spielbeginn immer geschlossen sein, da sich sonst sofort die Spielfiguren bewegen würden, ohne daß man einen Wert eingibt. Anschließend werden in der obersten Reihe die eigentlichen Helden des Spiels ausgedruckt — sieben Rüben und sieben Eier (in abwechselnder Reihenfolge).

Jetzt beginnt der Spaß. Sie werden zu einer Eingabe aufgefordert, eine ganze Zahl zwischen 1 und 9, entsprechend den Regalreihen 1 bis 9.

Sobald Sie eine Zahl eingegeben haben, verändert sich die betreffende Reihe wie folgt: Alle geschlossenen Fächer öffnen sich, alle geöffneten Fächer schließen sich, was soviel bedeutet, als daß der Inhalt eines offenen Fachs so weit es geht nach unten fällt.

Das Ziel des Spiels ist jetzt, die Rüben alle nach unten zu bringen, dabei aber die Eier (von freilebenden Hühnern) am Herunterfallen zu hindern, und das ganz allein durch »Invertieren« der einzelnen Reihen.

Eier, die unten ankommen, werden sofort zu Rührei und bringen keine Punkte mehr, wohl aber die, die heil geblieben sind. Je höher sie am Ende der Runde liegen, desto mehr Punkte.

Solange mindestens ein Ei heil bleibt, bekommen Sie ein Freispiel. Gehen alle »eggs« zu Bruch, ist das Spiel zu Ende. In diesem Fall erscheint Ihr Score mit dem Zusatz »GAME OVER«. Nach einem gewonnenen Spiel geht es nach Anzeige des Punktestands von vorne los, nach einem verlorenen Spiel erscheint nach einiger Zeit das Titelbild wieder und Sie können ein neues Spielchen wagen.

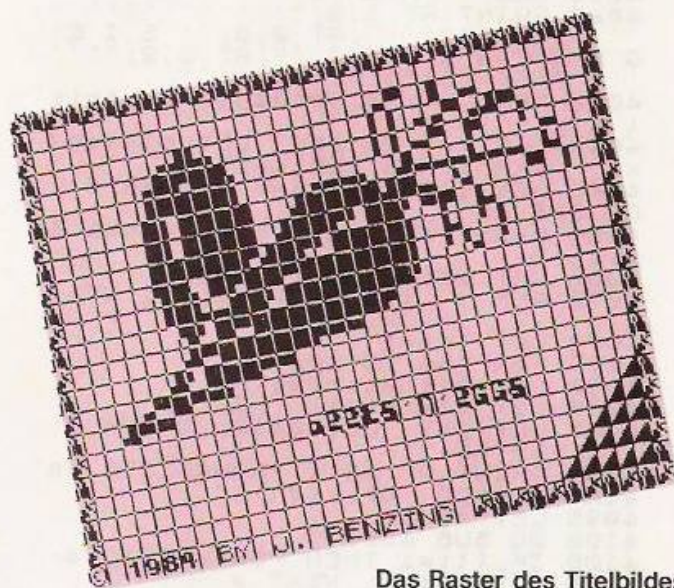
## Noch einige Hinweise zum Spiel

— Es gibt kein Unentschieden. Liegen die Rüben und Eier beim letzten Zug auf gleicher Höhe unter den selben Bedingungen, so verlieren Sie (einfach deshalb, weil die Eier vor den Rüben fallen). Also gar nicht erst soweit kommen lassen. Besser erst abhandeln, was auf den untersten Reihen liegt.

— Prüfen Sie vor jedem Zug die darunterliegenden Reihen, denn Sie können diese natürlich vor dem Zug zu Ihren Gunsten verändern oder zumindest das Beste daraus machen. Lassen Sie sich Zeit, es gibt keine Zeitbegrenzung.

— Jedes Ei, das heil geblieben ist, bringt (Reihe x 100) Punkte. Beispiel: Es bleiben zwei Eier in Reihe drei liegen, eines in Reihe sieben —  $300 \times 2 + 700 = 1300$  Punkte.

Nachdem beide Teile des Programms (program, bytes) geladen wurden, wird als erstes RAMTOP heruntergesetzt. Der Platz darüber wird für ein kleines Maschinenprogramm und für Daten, die das Titelbild betreffen, benötigt. Der nächste Schritt ist das POKEn der UDGs, die vollständig belegt sind.



Das Raster des Titelbildes

In Zeile 5200 wird ein Maschinencode-Programm in den Speicher geschrieben, so daß es später nur noch aufgerufen zu werden braucht.

Es geht weiter mit dem Titelbild, das ab Zeile 4000 entsteht. Hier wird oben genannter Maschinencode aufgerufen, der die Farbgebung des Titelbildes besorgt. Ich habe diesen Weg gewählt, weil es weit weniger mühsam ist, die Attribute als Zahlen einzutippen, als das ganze Titelbild im Direktmodus einzugeben. Das hier verwendete Verfahren geht außerdem (fast) genau so schnell.

## Programmbeschreibung

In Zeile 4500 findet die Abfrage nach dem Spielbeginn statt. Nach gedrückter Taste springt das Programm in Zeile 60 (Variablenliste).

Ab Zeile 100 wird das Spielfeld gezeichnet. Zeile 175 bis 190 verleihen dem Ganzen ein etwas plastisches Aussehen.

In Zeile 195 stehen die selbstdefinierten Ziffern als String. Warum, sieht man später. Zeile 200 numeriert die Reihen mit den oben genannten Zahlen durch.

Mit Zufallszahlen, in 16er Abständen, die als PLOT-Position gedeutet werden, folgt jetzt die Verteilung der einzelnen Fächer. Die Anfangskordinaten für Rüben und Eier stehen in 380 und 390.

# 'n' eggs



Das fertige Titelbild

Zeile 450 bringt zum ersten Mal die Spielfiguren zum Vorschein. Zeile 500: (siehe Variablenliste)

In Zeile 550 wird eine Zahl zwischen 1 und 9 verlangt. Wird eine kleinere oder größere Zahl oder aus Versehen ein Buchstabe eingetippt, so macht einen die Zeile 560 darauf aufmerksam. Im INPUT-Modus wird aus einem bestimmten Grund ein String anstelle einer Zahl verlangt. Tippt man versehentlich einen Buchstaben ein, so wird dieser nicht als Variable gedeutet. Somit entfällt also die Fehlermeldung »Variable not found« und das Programm geht zu 560.

Beachten Sie, daß Sie das Programm während des INPUTS nicht mit STOP unterbrechen können, da die Anführungszeichen, die gelöscht werden müßten, fehlen. Sie können natürlich die LINE-Funktion in Zeile 550 weglassen.

In Zeile 570 schließlich erfolgt die Umrechnung der Eingabe auf die PLOT-Position.

## Linienwechsel

Zeile 580 und 620 löschen beziehungsweise zeichnen einen Pfeil an der gerade verschobenen Reihe.

Der Tausch der beiden Variablen b und c ist nötig, um zuerst das alte Zeichen zu löschen und gleich darauf das neue zu zeichnen.

In Zeile 630 erfolgt das Vertauschen der Fächer mit Hilfe der OVER-Funktion, das heißt alle bereits vorhandenen Linien werden entfernt, auf einen leeren Platz wird eine Linie gesetzt.

Ab Zeile 1000 wird, getrennt für Eier und Rübchen, untersucht, ob der Weg nach unten frei ist.

Das geschieht mit der POINT-Funktion, bei gleichzeitigem Umrechnen der PRINT-(Eier, Rübchen) auf die PLOT-Position. Eine Untersuchung nur auf die Attribute ist nicht möglich, da diese die ganze Zeit über gleich sind.

Hat ein Ei das letzte Feld erreicht, so wird der Teilstring »s« des x-Wertes des gefallenen Ei's auf »1« gesetzt. Man braucht also nur zu testen, ob s\$ aus lauter »1«en besteht, wenn man wissen will, ob das Spiel zu Ende ist. Das ist einfacher und schneller als »IF s(1)=19 AND s(2)=19 AND s(3)=19...AND s(7)=19 THEN...« (19 ist die x-Koordinate des letzten Feldes). Natürlich kann man statt der »1« auch ein anderes Zeichen verwenden, aber »1« und »0« fallen nicht so sehr aus dem Rahmen.

Das gleiche gilt analog für g\$. Nach dieser Kontrolle geht's wieder zum INPUT nach 550.

Ist ein Spiel gewonnen, so werden in 2500 die Punkte berechnet. Diese Punktezahll wird zur Gesamtpunktzahl hinzugezählt. Die Zahl, die dabei herauskommt, wird in 2510 bis 2525 in den selbstdefinierten Zahlensatz umgewandelt. Zuerst wird die Gesamtpunktzahl »score« zu einem String gemacht, um jede Ziffer einzeln untersuchen zu können. Eine Schleife geht jetzt jede Position durch und wandelt die bestimmte Zahl in das entsprechende Zeichen um, das n\$ entnommen wird. In Zeile 2540 entsteht aus dem entstandenen String + »YOUR SCORE« ein neuer String.

Dasselbe gilt für ein verlorenes Spiel (natürlich ohne Punkte-zählung), und der String sieht etwas anders aus. Er steht in 3030.

Die Zeilen 3050 bis 3150 sorgen jetzt dafür, daß der Text (a\$) über den Bildschirm huscht. Das Verfahren dürfte klar sein: a\$ wird zuerst von der ersten bis zur neunten Position gedruckt, dann von der zweiten bis zur zehnten, von der dritten bis zur elften etc. Ist die Länge des Strings erschöpft, geht's von vorn los. So entsteht der Eindruck, als ob das ganze Ding von rechts nach links läuft.

In Zeile 3070 wird untersucht, ob ein Spiel gewonnen wurde. Wenn ja, dann geht es nach 4000 (Titelbild).

In 3300 findet das »GAME RESET« statt. Alle Spielfiguren werden gelöscht (3300), an die Stelle, wo die Laufschrift war, wird ein entsprechendes Leerfeld gesetzt (3320) und das fehlende Stück ersetzt. Der Pfeil am linken Rand verschwindet (3330). Anschließend prüft Zeile 3340, ob die oberste Linie noch steht. Ist dies nicht der Fall, so wird diese Linie eingezeichnet, um zu Beginn des Spiels ein frühzeitiges Herunterfallen der Rübchen beziehungsweise Eier zu vermeiden. (Der erste Zug, bei dem sich etwas tut, ist also das Bewegen der 9. Reihe.) Anschließend wird »k« neu dimensioniert, das heißt auf »0« gesetzt, und das Spiel geht wieder zurück zum erneuten Verteilen der Fächer (Zeile 300). Benutzen Sie zum Abspeichern die Routine in Zeile 9999, dann können Sie später wie gewohnt laden (LOAD "" oder LOAD "beets").

## Der Rest ist Hexadezimal

Wenn Sie das Programm abgetippt haben, so müssen Sie vor dem Abspeichern noch eine Reihe Zahlen mit dem kleinen Hilfsprogramm eingeben. Diese werden in einer separaten SAVE-Routine abgespeichert und auch wieder geladen (GO TO 9999).

(Jörg Benzing)

1&gt;REM

```

BEETS 'N' EGGS
FOR SPECTRUM 48k
© 1984 BY J. BENZING
HIRSCHBERGSTR. 33
7730 US-SCHWENNINGEN
TEL. 07720/31133

```

```

5 CLEAR 39999
6 LOAD "beets" CODE
10 BORDER 1: PAPER 1: INK 6: B
RIGHT 1: CLS
15 PRINT AT 11,8;"THIS GAME IS
...
20 GO SUB 5000
50 GO TO 4000
60 DIM s(7): DIM i(7): DIM g(7)
: DIM e(7): DIM k(7)
90 REM SPIELFELD
100 FOR x=19 TO 243 STEP 16
120 PLOT x,11: DRAW 0,160: BEEP
.003,30
130 NEXT x
150 FOR x=11 TO 171 STEP 16
160 PLOT OVER 1,19,x: DRAW OVER
1,224,0: BEEP .003,40
170 NEXT x
175 LET z=10
180 FOR x=21 TO 25: PLOT x,z: D
RAW 222,0: LET z=z-1: NEXT x
185 LET z=12
190 FOR x=244 TO 248: PLOT x,z:
DRAW 0,157: LET z=z-1: NEXT x
195 LET n$="LMNOPQRSTU"
200 FOR x=2 TO 18 STEP 2: PRINT
AT x,1;n$(11-x/2): BEEP .01,x:
NEXT x
250 LET score=0
290 REM REGALFÄCHER VERTEILEN
300 FOR x=0 TO 100
320 LET a=20+(INT (RND*14)*16)
330 LET b=27+INT (RND*8)*16
340 PLOT OVER 1;a,b: DRAW OVER
1;14,0: BEEP .003,25
350 NEXT x
360 REM KOORDINATEN FUER A,B
380 FOR x=1 TO 7: LET s(x)=1: L
ET g(x)=1: NEXT x
390 FOR x=1 TO 7: LET i(x)=(x*4
-1): LET e(x)=(x*4+1): NEXT x
395 REM KONTROLLSTRINGS FUER
SPIEL-ENDE
400 LET s$="0000000": LET g$="0
000000"
420 REM VARIABLE + SPIELBEGINN
450 FOR x=1 TO 7: PRINT AT s(x)
,i(x): INK 7;"A": PRINT AT g(x)
,e(x): INK 6;"B": NEXT x
500 LET a=19: LET b=27: LET c=1
: LET p=0
540 REM EINGABE EINER ZAHL UND
UMRECHNEN AUF PLOT-
POSITION
550 INPUT TAB 7; INK 0; PAPER 5
;" YOUR MOVE !"; PAPER 1;" "; LI
NE c$
560 IF CODE c$<49 OR CODE c$>57
OR LEN c$>1 THEN PRINT #0;TAB 7
; FLASH 1; INK 7; PAPER 2;" ONLY
1 - 9 ": FOR x=0 TO 10: BEEP .0

```

```

1,40-x: BEEP .01,30+x: NEXT x: P
AUSE 30: GO TO 550
570 LET c=VAL c$: LET c=11+(c*1
6)
575 REM FREI
580 INVERSE 1: PLOT 0,b-1: DRAW
2,0: DRAW 0,-2: DRAW 3,3: DRAW
-3,3: DRAW 0,-2: DRAW -2,0: DRAW
0,-2: INVERSE 0
510 LET b=c
620 PLOT 0,b-1: DRAW 2,0: DRAW
0,-2: DRAW 3,3: DRAW -3,3: DRAW
0,-2: DRAW -2,0: DRAW 0,-2
625 REM FACH
630 PLOT OVER 1;a,b: DRAW OVER
1;224,0
900 REM UNTERSUCHEN OB WEG FREI
NACH UNTEN. JA-DANN 1
FELD RUNTER -WIEDERHOLE
BIS WEG NICHT FREI.
1000 FOR x=1 TO 7
1020 IF POINT (i(x)*8,(22-s(x))*
8-13)=0 THEN PRINT AT s(x),i(x);
" ": LET s(x)=s(x)+2: PRINT AT s
(x),i(x): INK 7;"A": BEEP .005,2
2-s(x): GO TO 1020
1025 IF s(x)=19 THEN LET s$(x)="
1": PRINT AT s(x),i(x): INK 7;"D
": IF k(x)=0 THEN FOR z=0 TO 5:
BEEP .003,20-z: NEXT z: LET k(x)
=1
1030 NEXT x
1035 REM ALLE EIER UNTEN ?
1040 IF s$="1111111" THEN GO TO
3000
1045 REM SIEHE 900
1050 FOR x=1 TO 7
1060 IF POINT (e(x)*8,(22-g(x))*
8-13)=0 THEN PRINT AT g(x),e(x);
" ": LET g(x)=g(x)+2: PRINT AT g
(x),e(x): INK 6;"B": BEEP .005,1
0-g(x): GO TO 1060
1065 IF g(x)=19 THEN LET g$(x)="
1"
1070 NEXT x
1075 REM ALLE AUEBEN UNTEN ?
1080 IF g$="1111111" THEN GO TO
2500
2000 GO TO 550
2400 REM UMRECHNEN DER HOEHE
DER UEBRIGGEBLIEBENEN
EIER AUF PUNKTE
2500 FOR x=1 TO 7: LET p=p+(22-s
(x)-3)/2*100: NEXT x: LET score=
score+p
2505 REM SCORE MIT USERDEF. ZAHL
EN (n$)
2510 LET d$=STR$ score
2515 FOR x=1 TO LEN d$
2520 FOR z=0 TO 9
2523 IF d$(x)=STR$ z THEN LET d$
(x)=n$(z+1)
2525 NEXT z: NEXT x
2530 REM LAUFSCHRIFT FUER
PUNKTESTAND NACH EINEM
GEWONNENEN SPIEL
2540 LET b$="YOUR SCORE: "+d$+"
*"
2545 REM SIEHE 2505
2550 LET a$=b$+b$+b$
2600 GO TO 3050
3000 LET d$=STR$ score
3005 FOR x=1 TO LEN d$
3010 FOR z=0 TO 9: BEEP .01,z: B
EEP .01,z+3
3020 IF d$(x)=STR$ z THEN LET d$
(x)=n$(z+1)

```

```

3025 NEXT z: NEXT x
3027 REM LAUFSCHRIFT,TITEL,GAME
OVER + SCORE
3030 LET b$="EJUGH'I'JKKH * GAME
OVER * SCORE: "+d$+" *
3040 LET a$=b$+b$+b$
3050 LET z=0: LET count=0
3060 LET z=z+1: IF z>2/3*LEN a$+
1 THEN LET z=1: LET count=count+
1
3065 REM GAME OVER ODER WEITER ?
3070 IF count=2 AND g$="111111"
THEN GO TO 3300
3080 IF count=1 AND s$="111111"
THEN GO TO 4000
3090 REM PRINT LAUFSCHRIFT
3100 PRINT AT 10,12; PAPER z/15;
INK 9:a$(z TO z+8)
3110 PAUSE 3
3150 GO TO 3060
3200 REM GAME RESET
3300 FOR x=1 TO 7: PRINT AT s(x)
,i(x);" ": PRINT AT g(x),e(x);"
": NEXT x
3320 PRINT AT 10,12;" "
PLOT 84,91: DRAW 94,0: PRINT AT
10,12; OVER 1;"C C C C C"
3330 INVERSE 1: PLOT 0,b-1: DRAW
2,0: DRAW 0,-2: DRAW 3,3: DRAW
-3,3: DRAW 0,-2: DRAW -2,0: DRAW
0,-2: INVERSE 0
3335 REM NICHTIG OBERSTE LINIE
MUSS BEI SPIELBEGINN
DA SEIN
3340 IF POINT (20,155)=0 THEN PL
OT OVER 1;19,155: DRAW OVER 1;22
4,0
3345 DIM k(7)
3350 GO TO 300
4000 REM TITELBILD
4100 PRINT AT 0,0;"BABABABABABAB
ABABABABABABABABABABA"
4110 PRINT
4120 PRINT
4130 PRINT
4140 PRINT
4150 PRINT
4160 PRINT
4170 PRINT
4180 PRINT
4190 PRINT
4200 PRINT
4210 PRINT
4220 PRINT
4230 PRINT
4240 PRINT
4250 PRINT
4260 PRINT EJUGH'I
JKKH EA
4270 PRINT EER"

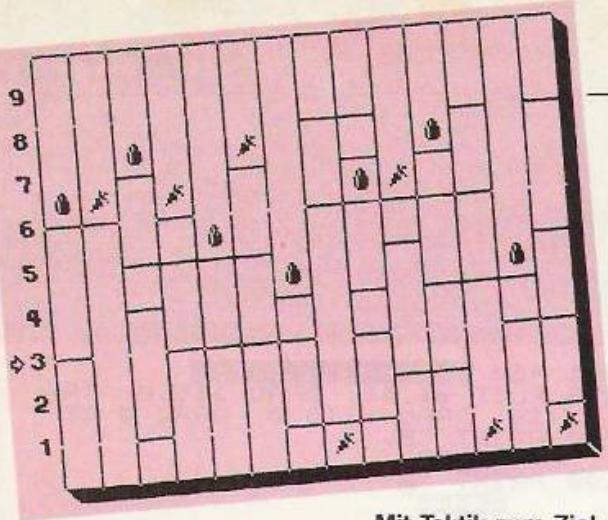
```

```

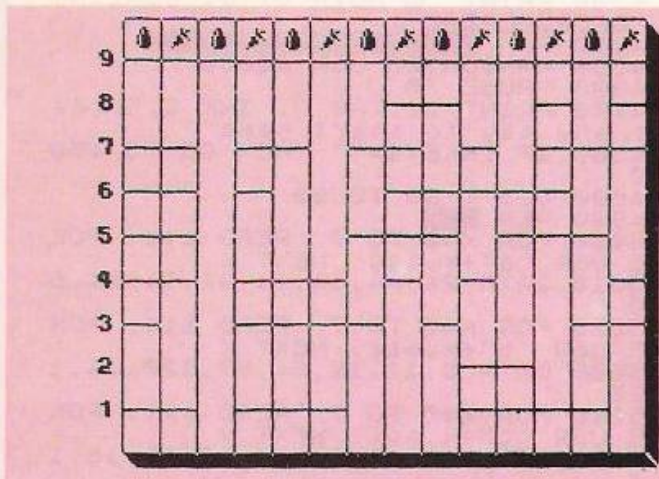
4271 REM UMRÄHMUNG TITEL
4275 PLOT 94,51: DRAW 115,0: DRA
W 0,-15: DRAW -115,0: DRAW 0,15
4280 PRINT "B
EEEE"
4290 PRINT "A
EEEE"
4295 PRINT "R
EEEEEA"
4300 PRINT "© MUTE BY J. BENZING
BABABABABAB"
4310 REM MC FARBE TITELBILD
4350 RANDOMIZE USR 40000
4390 PAUSE 70
4400 PRINT #0;TAB 2; INK 5;"pres
s any key to start game"
4500 IF INKEY$="" THEN GO TO 450
0
4600 CLS : GO TO 60
4900 REM UOG
5000 FOR x=0 TO 7: READ egg: POK
E USR "a"+x,egg: NEXT x
5010 DATA 24,44,44,94,94,94,94,6
0
5020 FOR x=0 TO 7: READ bee: POK
E USR "b"+x,bee: NEXT x
5030 DATA 8,11,12,54,57,120,96,1
28
5040 FOR x=0 TO 7: READ spi: POK
E USR "c"+x,spi: NEXT x
5050 DATA 16,16,16,16,16,16,16,1
0
5055 FOR x=0 TO 7: READ egg2: PO
KE USR "d"+x,egg2: NEXT x
5056 DATA 0,0,0,0,104,61,127,255
5057 FOR x=0 TO 7: READ spec: PO
KE USR "e"+x,spec: NEXT x
5058 DATA 1,3,7,15,31,63,127,255
5059 REM UOG TITELBUCHSTABEN
5060 FOR x=0 TO 47: READ let: PO
KE USR "f"+x,let: NEXT x
5070 DATA 192,192,252,204,206,25
4,254,124,112,112,252,112,112,12
0,126,62,124,224,252,126,14,254,
254,252,124,230,226,226,226,226,
226,226,124,206,206,252,192,254,
254,124,124,206,192,222,206,254,
254,124
5090 REM ZIFFERN 0-9
5100 FOR x=0 TO 79: READ num: PO
KE USR "i"+x,num: NEXT x
5120 DATA 0,60,102,102,102,102,6
0,0,0,12,60,12,12,12,12,0,0,60,1
02,6,60,96,126,0,0,60,102,12,6,1
02,60,0,0,60,106,103,126,12,12,0
,0,124,96,124,6,102,60,0,0,60,96
,124,102,102,60,0,0,126,102,6,12
,12,12,0,0,60,102,60,102,102,60,
0,0,60,102,102,62,6,60,0
5190 REM DATA MC FARBE TITELBILD
5200 FOR x=40000 TO 40011: READ
u: POKE x,u: NEXT x
5230 DATA 33,40,160,17,0,68,1,19
00,2,337,176,201
53000 RETURN
53990 REM SAVE
53999 SAVE "beets" LINE 5: SAVE "
beets"CODE 41000,704: PRINT AT 2
0,1;"VERIFY": VERIFY "beets": VE
RIFY "beets"CODE : BEEP .1,30

```

Basic-Listing »beets'n'eggs«



Mit Taktik zum Ziel



VARIABLENLISTE  
"beets 'n' eggs"

x,y	Allround Variable fuer Schleifen
s(i),j(i)	x und y Koordinaten fuer Eier
g(i),e(i)	x und y Koordinaten fuer Rueben
k(i)	Kontrolliert das Geräusch eines zerbrochenen Ei's
p	Punktezahl fuer ganz gebliebene Eier
score	Gesamtpunktezahl
a,b	PLOT Koordinaten fuer Faecher
c	Hoehle einer Reihe entsteht aus der eingegebenen Zahl
count	zaehlt die Durchgaenge der Lauf-schrift

Variablenliste

◀  
»Grundstellung«

Hexadezimal-Listing »beets 'n' eggs«

41000	4A4F4A4F4A4F4A4F	->	612	41352	4E4E4E4E4E4E4D4D	->	622
41008	4A4F4A4F4A4F4A4F	->	612	41360	5D4B534A4A4A4A4A	->	621
41016	4A4F4A4F4A4F4A4F	->	612	41368	4A4A4A4A4A4A4F4F	->	607
41024	4A4F4A4F4A4F4A4F	->	612	41376	4F4F4F4F4F4F4F4A	->	627
41032	4E4E4E4E4E4E4E4E	->	624	41384	4A4F4F4F4F4F4B4B	->	619
41040	4E4E4E4E4E4E4E4E	->	624	41392	534A4A4A4A4A4A4A	->	601
41048	4E4E4E4E4E4E4E4E	->	624	41400	4A4A4A4A4A4A4A4A	->	602
41056	4E4E4E4E4E4E4E4A	->	620	41408	4F4F4F4F4F4F4F4F	->	632
41064	4A4F4F4F4F4F4F4F	->	627	41416	4E4E4E4E4E4B4B53	->	623
41072	4F4F4F4F4F4F4F4F	->	632	41424	4A4A4A4A4A4A4A4A	->	602
41080	4F4F4C4C4C4C4C4C	->	614	41432	4A4A4A4A4F4F4F4F	->	612
41088	4C4A4A4A4A4A4A4F	->	699	41440	4F4F4F4F4F4F4F4A	->	627
41096	4E4E4E4E4E4E4E4D	->	623	41448	4A4A4A4B4B534A4A	->	603
41104	4D4D4F4F4F4F4F4F	->	628	41456	4A4A4F4F4F4F4F4F	->	622
41112	4F4F4C4C4C4C4C4C	->	614	41464	4F4F4F4F4F4F4F4F	->	632
41120	4C4C4C4C4C4F4F4A	->	615	41472	4F4F4F4F4F4F4F4F	->	632
41128	4A4F4D4D4D4D4D4D	->	615	41480	4E4E4E4B4B4A4F4F	->	616
41136	7D4F4F4F4F4F4F4F	->	678	41488	4F4F4F4E4E4E4E4E	->	627
41144	4F4F4F4C4C4C4C4C	->	617	41496	4E4E4E4E4E4E4E4E	->	624
41152	4C4C4C4C4C4C4C4F	->	611	41504	4E4E4E4F4F4F4F4A	->	624
41160	4E4E4E4E4E4E4D4D	->	622	41512	4A4E4E4E4E4E4E4E	->	620
41168	4F4F4F4F4F4F4F4B	->	628	41520	4E4E4E4E4E4E4E4E	->	624
41176	4B4B4B4F4C4C4C4C	->	608	41528	4E4E4E4E4E4E4E4E	->	624
41184	4C4C4C4C4C4C4C4A	->	606	41536	4E4E4E4E4E4E4A4F	->	621
41192	4A4F4F4F4F4F4D7D	->	671	41544	4E4E4E4E4E4E4E4E	->	624
41200	4F4F4F4F4F4F4B4B	->	624	41552	4E4E4E4E4E4E4E4E	->	624
41208	4B4B4B4B4C4C4C4C	->	604	41560	4E4E4E4E4E4E4E4E	->	624
41216	4C4C4C4C4C4C4C4F	->	611	41568	4E4E4E4E4E4A564A	->	624
41224	4E4E4E4E4E4E4D7D	->	670	41576	4A4F4F4F4F4F4F4F	->	627
41232	4F4F4F4F4F4F4B4B	->	620	41584	4F4F4F4F4F4F4F4F	->	632
41240	53534A4A4A4C4C4C	->	618	41592	4F4F4F4F4F4F4F4F	->	632
41248	4C4C4C4C4C4C4C4A	->	606	41600	4F4F4F4F4A56744F	->	671
41256	4A4F4F4F4F4F4D4D	->	623	41608	4E4E4E4E4E4E4E4E	->	624
41264	4F4F4F5F4B4B5353	->	648	41616	4E4E4E4E4E4E4E4E	->	624
41272	4A4A4A4A4A4A4F4C	->	601	41624	4E4E4E4E4E4E4E4E	->	624
41280	4C4C4C4C4C4C4C4F	->	611	41632	4E4E4E4A5674654A	->	635
41288	4E4E4E4E4E4E4D4D	->	622	41640	4A4F4F4F4F4F4F4F	->	627
41296	7D4F5F4B4B534A4A	->	630	41648	4F4F4F4F4F4F4F4F	->	632
41304	4A4A4A4A4A4A4C4C	->	696	41656	4F4F4F4F4F4F4F4F	->	632
41312	4C4C4C4C4C4C4C4A	->	606	41664	4F4F4A567465694F	->	719
41320	4A4F4F4F4F4F4D4D	->	623	41672	4E4E4E4E4E4E4E4E	->	624
41328	7D7B4B4B534A4A4A	->	703	41680	4E4E4E4E4E4E4E4E	->	624
41336	4A4A4A4A4A4A4F4F	->	607	41688	4E4E4E4E4E4A4F4A	->	617
41344	4F4F4F4F4F4F4F4F	->	632	41696	4F4A4F4A4F4A4F4A	->	612



# Sauberermann

Helfen Sie dem Spectrum-Sauberermann bei seinem Kampf gegen Zeit und Schwamm. 48 KByte RAM sind Voraussetzung, Geduld unverzichtbar.

Für Mc Wash ist großer Badetag. Aber er freut sich gar nicht darüber, denn für ihn bedeutet baden nur harte, gefährliche und spannende Arbeit! Der Grund: seine Badewanne steht in einem großen (sechs Bildschirme) Labyrinth. Darin fünfmal einen Eimer zu finden und zur Wanne zu bringen, ist schwer, zumal das Labyrinth von vielen wabbeligen Badeschwämmen bewacht wird, die sich gar nicht gerne ihre Eimer stehlen lassen! Sie wehren sich, indem sie hinter Mc Wash herlaufen, sobald er einen Eimer hält und versuchen, ihm diesen wegzunehmen und von neuem zu verstecken. Unter solchen Umständen muß unser Held innerhalb einer bestimmten Zeit sein Bad bereiten; wenn er es nicht schafft, oh grausame Strafe, erhält der Spieler keine Punkte!

## Programmtechnisches:

Tastenbelegung:

a-oben  
z-unten  
u-links  
i-rechts

(Es sind auch andere Tastenbelegungen möglich.)

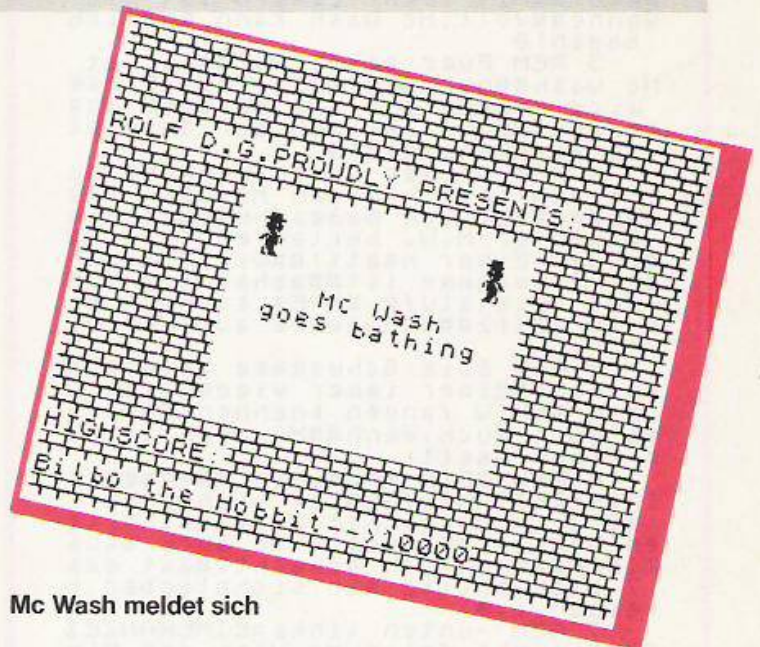
Im Maschinencodeteil sollten keine Veränderungen vorgenommen werden, die Grafik und das Labyrinth können jedoch selber gestaltet werden. Zum Aufbau des Labyrinths: Zum Beispiel die 2. Zeile, 1. Spalte des Labyrinths steht in Speicherzelle 30682 (=30670+12, das Labyrinth ist 12 Byte breit, was sich aus 3 (3 Screens breit) mal 32 (Screen=32 Spalten) mal 1 : 8 (eine Spalte=ein Bit!) ergibt.

»PEEK 30682« ergibt 128  
(=bin 10000000,1=mauer,0=weg)

## Eingabe der Programme

Zuerst das Maschinencode-Listing eingeben, dann das Basic-Programm; Basic-Programm SAVEn mit »SAVE 'MC WASH' line 99«, dann »SAVE 'mc wash' code 30000,3358«. Viel Spaß beim Spielen.

(Rolf Lakämper)



Mc Wash meldet sich

Speicherzelle	Funktion
31440	Flag (z.B. Farbe der Badewanne usw.)
31441	Farbe des Zeitzeichens
31442	Level
31443	Spalte Eimer
31444	Zeile Eimer
31445	Spalte Mc Wash
31446	Zeile Mc Wash
31447-31456	Arbeitsspeicher
31457	Bewegungsflag
31644-31673	Variablen (Spalte/Zeile) für Schwämme 1-15

(Basicteil)

hsc=highscore

sc=score

n\$=Name des Highscoreinhabers

lev=Level

Programmteile:

30000-30599	Scroll(re,li,u,o)
30600	Printroutine
30647	Zufallszahlgenerator
30670-31245	Labyrinth!!
31246	Unterprogramme
31501	Abfrage, ob Labyrinth an bestimmter Stelle Mauer oder Weg ist
31685	Bewegung Mc Wash
32017	Eimerabfrage usw.
32451	Bewegung Schwämme
32861	Initialisierung Schwämme
32920-33357	Grafik!

Variablen-tabelle: (für Maschinencodeteil)

## Basic-Listing »Mc Wash«

```

1 REM Instructions:#-----
---##Helfen Sie Mc Wash, seine B
ade##wanne zu fuehlen !##Hierzu
muessen Sie den Eimer##suchen, d
er in irgendeinem Teil##des sech
s Bildschirme grossen##Labyrinth
es versteckt ist!
2 REM @Unser Held muss diesen
Eimer##dann in die Badewanne ki
ppen!##Mit fuenf Eimern ist die
Wanne##voll, Mc Wash kann endlich
baden!@
3 REM Fuer seine Aufgabe hat
Mc Wash##nur begrenzt Zeit. Diese
wird zu##Anfang des Spieles ang
ezeigt.##Und ist von der Spielst
ufe ab-##haengig!@
4 REM An der Durchfuehrung se
iner##Arbeit hindern Mc Wash kle
ine##wabbelige Badeschwaemme, die
##hinter M.W. herlaufen, falls##
er den Eimer haelt!##Die Anzahl
der Schwaemme ist##abhaengig von
der Spielstufe.##(Es koennen bi
s zu 15!?)##Schwaemme auftreten!
)
5 REM @Die Schwaemme versteck
en den##Eimer immer wieder, falls
sie ##M.W fangen koennen.##Dies
es gilt auch, wenn##Mc Wash keine
n Eimer haelt!
6 REM @Die Anzeigen:##-oben l
inks=ZEITANZEIGE#Das blinkende T
wird in #gleichen Zeitabschnitt
en dunkler#Dieses wird auch akus
tisch#kenntlich gemacht.#Ist das
T blau, sollte man sich#lieber b
eeilen...@
7 REM -unten links=EIMERANZEI
GE##blinkt, falls Mc Wash den Eim
er#haelt.@
8 REM Viel Spass!#####@by R
olf Lakaemper#Uhuweg 6#4830 Guet
erstloh 11#TEL.05241/73430&
9 DATA 3,5,3,7,7,5,3,7,7,5,3,
8,8,5,3,8,8,5,3,7,7,5,3,3,4,5,7,
8,12,3,12,8
10 DATA 3,5,3,3,7,15,7,3,7,15,
3,8,12,15,3,3,12,15,3,5,5,4,3,3,
4,5,3,12,3,12,3,8
20 DATA 0,.1,5,.1,9,.1,12,.2,9
,.1,12,.5,9,.1,10,.2,6,.1,7,.2,1
2,.1,9,.3,5,.3
21 DATA 0,.1,5,.1,9,.1,12,.2,9
,.1,12,.5,9,.1,12,.1,14,.1,12,.1
,10,.1,9,.1,7,.1,5,.6
30 DATA .5,0,.25,7,.25,5,.5,7,
.5,0,.5,-1,.5,0,.5,-3,.5,0,.5,-4
,.25,5,.25,4,.25,5,.25,2,.25,0,.
25,-1,.5,5,.25,4,.25,2,1,4
40 LOAD "mc wash" CODE
100 CLEAR 29999
101 PAPER 0: INK 7: BORDER 4
102 LET n$="Bilbo the Hobbit":
LET hsc=10000
103 LET cl=120: LET wd=130: LET
sc=0
104 POKE 23675,152: POKE 23676,
128
110 GO TO 150
115 REM Unterprogramme
120 PRINT AT 0,0;
122 FOR i=1 TO 22
124 PRINT PAPER 2;"BBBBBBBBBBBBBB
BBBBBBBBBBBBBBBBBBBB"
125 BEEP .005,i+20
126 NEXT i
127 PRINT AT 0,0;
128 RETURN
130 FOR i=5 TO 15
132 PRINT AT i,8; PAPER 5;"
134 NEXT i
136 RETURN
138 PRINT AT 21,0;"PRESS ANY OL

```

```

RUBBER KEY": IF INKEY$="" THEN
GO TO 138
139 GO SUB cl: RETURN
140 FOR i=1 TO LEN a$
142 PRINT a$(i);
143 BEEP .003,10: NEXT i
144 PRINT : RETURN
150 REM Anfang
152 CLS : GO SUB cl
155 PRINT AT 4,0;"ROLF D.G.PROU
DLY PRESENTS:"
160 GO SUB wd
162 PAPER 5: INK 0
165 PRINT AT 10,13; PAPER 4;"Mc
Wash";AT 11,10;"goes bathing"
168 PRINT AT 7,9;CHR$ 146;CHR$
147;AT 8,9;CHR$ 149;CHR$ 148;AT
7,21;CHR$ 154;CHR$ 155;AT 8,21;C
HR$ 157;CHR$ 156
169 PAPER 0: INK 7
171 PRINT AT 18,0; FLASH 1;"HIG
HSCORE": PRINT : PRINT n$;"-->"
;hsc
172 RESTORE 10: FOR i=1 TO 64
174 READ a: BEEP .03,a
175 IF INKEY$<>"" THEN GO TO 18
2
177 NEXT i
180 GO TO 172
182 GO SUB 138
184 PRINT "Do you want to see i
nstructions?(j/*)"
185 IF INKEY$="" THEN GO TO 185
187 IF INKEY$<>"j" THEN GO TO 2
80
200 REM Instructions
201 GO SUB cl
202 LET t=23759
205 LET t=t+1
210 IF PEEK t=CODE "#" THEN PRI
NT : GO TO 205
220 IF PEEK t=13 THEN LET t=t+5
230 IF PEEK t=CODE "@" THEN GO
SUB 138: GO TO 205
235 IF PEEK t=CODE "&" THEN GO
TO 260
240 PRINT CHR$ PEEK t;
245 BEEP .0028,0+40*(t-INT (t/2
)*2)
250 GO TO 205
260 GO TO 150
280 REM Spielbeginn
285 GO SUB cl
290 LET a$="Select Skill Level
(1 TO 11)"
300 GO SUB 140
310 INPUT lev: IF lev<0 OR lev>
11 THEN GO TO 280
320 PRINT AT 3,0: LET a$="LEVEL
"+STR$(lev)
330 GO SUB 140
340 PRINT AT 5,0; LET a$=STR$(
20+lev-15)+" Badeschwaemme."
345 GO SUB 140
350 LET a$=STR$(7*5.12*(16-lev
))+" seconds time."
355 GO SUB 140
360 GO SUB 138
365 FOR i=9 TO 0 STEP -1: PRINT
AT 12,10;" ";i;" ": BEE
P .5,i-12: NEXT i
367 POKE 31530,16-lev
370 POKE 23672,0: POKE 23673,0
379 REM Hier Sprung in Maschine
ncode!
380 LET a=USR 32861
385 REM Auswertung
400 IF a=0 THEN GO TO 500
410 REM geschafft!
415 GO SUB wd: PRINT AT 10,12;"
WELL DONE !"
420 RESTORE 20:
430 FOR i=1 TO 27
435 READ a,b: BEEP b,a
440 NEXT i

```

```

450 GO SUB cl
452 GO TO 750
455 LET lev=lev+1: IF lev>11 TH
EN LET lev=11
460 PRINT AT 2,0;"Now try level
"+STR$ lev
470 GO TO 340
500 REM Leider verloren...
505 PRINT AT 10,12;"OUT OF TIME

610 FOR i=-20 TO 30: BEEP .03,i
: NEXT i
620 RESTORE 30
630 FOR i=1 TO 20
640 READ a,b: BEEP a,b
650 NEXT i
660 GO SUB 138
662 IF sc>hsc THEN GO TO 900
665 PRINT AT 10,10;
670 LET a$="ONCE MORE? (* /n)"
675 GO SUB 140
680 INPUT a$: IF a$<>"n" THEN G
O TO 103
700 LET a$="GOOD BYE"
710 GO SUB 140
720 STOP

```

```

750 REM Punkte
760 LET a=ATTR (0,0)
770 LET a=a-INT (a/8)*8
780 LET bonus=a*100*(lev+10)
790 LET a$="Bonus Points:"+STR$
bonus: GO SUB 140
800 LET a$="SCORE:"
810 GO SUB 140;
820 FOR i=1 TO bonus STEP 100
830 LET sc=sc+100
840 PRINT AT 1,6;sc
845 BEEP .005,40
850 NEXT i
852 PAUSE 4e4
855 GO SUB cl
860 GO TO 455
900 LET a$="You have beaten the
old record! Enter your name!":
GO SUB 140
910 INPUT n$: IF LEN n$>22 THEN
LET n$=n$( TO 22)
920 LET hsc=sc
925 LET a$=n$
930 GO SUB 140
940 GO TO 660

```

Basic-Listing »Mc Wash« (Schluß)

## Hexadezimal-Listing »Mc Wash«

```

300000 232323E50620C501 -> 570
300008 0008C521FE4711FF -> 835
300016 47EDB8C1C521FE4F -> 1248
300024 11FF4FEDB8C121FE -> 1252
300032 5711FF57EDB821FE -> 1154
300040 5A11FF5A010003ED -> 693
300048 B8C14806183E1890 -> 709
300056 E1E5C547110C0004 -> 755
300064 373FED521910F0C1 -> 924
300072 3E2091CB3FCB3FCB -> 974
300080 3F5F1600373FED52 -> 617
300088 180218A8E2091E5 -> 689
300096 073C5847373E0017 -> 366
300104 10FDA6432810C50E -> 769
300112 003E18904716011E -> 354
300120 17CD8877180EC50E -> 732
300128 003E18904716001E -> 353
300136 00CD8877C110A641 -> 900
300144 10C8E1C9E50620C5 -> 1106
300152 010008C521014011 -> 321
300160 0040EDB8C1C52101 -> 901
300168 48110048EDB8C121 -> 800
300176 0150110050EDB8021 -> 624
300184 0158110058010003 -> 198
300192 ED80C14806183E18 -> 794
300200 90E1E5C547110C00 -> 895
300208 04373FED521910FD -> 735
300216 C13E2091CB3FCB3F -> 964
300224 CB3F5F1600191802 -> 434
300232 18AD3E2091E6073C -> 733
300240 5847373E001F10FD -> 576
300248 A6432810C50E1F3E -> 593
300256 18904716011E17CD -> 520
300264 8877180EC50E1F3E -> 597
300272 18904716011E00CD -> 496
300280 8877C110A94110C8 -> 914
300288 E1C9E50618C50617 -> 911
300296 53CD9B0E4378E607 -> 886
300304 200511E0F8180311 -> 570
300312 E0FFE519EBE14806 -> 1271

```

```

300320 20E5D5C506067E12 -> 829
300328 241410FAC1D1E123 -> 904
300336 1310EE4110D22120 -> 629
300344 5811005801E002ED -> 657
300352 B0C14806203E1891 -> 710
300360 E1E5C547110C0004 -> 755
300368 ED521910FD3E20C1 -> 908
300376 90CB3FCB3FCB3F5F -> 1037
300384 1600191802189E78 -> 375
300392 3DE607584704373E -> 578
300400 001710FD43A62810 -> 981
300408 C53E20904F061716 -> 566
300416 011E17CD8877180E -> 952
300424 C53E20904F061716 -> 566
300432 001E00C06877C110 -> 699
300440 AC4110C9E1C91114 -> 917
300448 0119E50618C50617 -> 511
300456 587830C09E0E4378 -> 633
300464 E607200511200718 -> 354
300472 03112000E519EBE1 -> 766
300480 480620E5D5C50606 -> 763
300488 7E12241410FAC1D1 -> 868
300496 E1231310EE4110D0 -> 822
300504 21DF5A11FF5A01E0 -> 933
300512 02EDB8C14806203E -> 788
300520 1891E1E5C547110C -> 920
300528 000419373FED5210 -> 482
300536 FA3E20C190CB3FCB -> 1150
300544 3FCB3F5F16001918 -> 495
300552 02189A783DE60758 -> 686
300560 4704373E001710FD -> 484
300568 43A6280616011E17 -> 355
300576 180416001E00C53E -> 339
300584 20904F0600C08877 -> 721
300592 C11064411003E1C9 -> 1107
300600 7804D511200021E0 -> 643
300608 571910FD0947D173 -> 785
300616 2A7B5C5A1600EB29 -> 645
300624 2929EB19E578CD9E -> 1054
300632 0E060009D106081A -> 278
300640 77241310FAC9C9E5 -> 1071

```

```

30646  D5C5210000ED5B76 -> 889
30656  5C13064B1910FD22 -> 520
30664  765CC101E1C9FFFF -> 1548
30672  FFFFFFFF -> 2040
30680  FFFF801FFC000038 -> 977
30688  3FC01F9FFF838007 -> 966
30696  FC0000003FC00798 -> 666
30704  3E039C870CFFE701 -> 855
30712  01C980183E139CE0 -> 815
30720  0C00E73F01C98019 -> 661
30728  3E718C000C03E11E -> 585
30736  3F0993E108718006 -> 711
30744  7C03E11E3F0993E1 -> 826
30752  0871C09E7C91F99E -> 1147
30760  7F399267C1F3C09E -> 1219
30768  0091F98E7F2193E7 -> 1074
30776  01E38F9E0098098E -> 1024
30784  7F2192600FE38F80 -> 915
30792  679809C007F279260 -> 864
30800  0FE79F80641FC040 -> 920
30808  07201267C067801E -> 613
30816  641FC04800201267 -> 548
30824  C067801E673FF249 -> 934
30832  E0339007C7FE79F82 -> 1153
30840  6701F20807339007 -> 563
30848  C0079F8200010206 -> 691
30856  073383FFC0079812 -> 813
30864  0038027E4F3383FF -> 700
30872  CFFF8013F9F80E78 -> 1240
30880  4F3393FFC0FF81F3 -> 1361
30888  C079FE604F0193FF -> 1145
30896  C0FF9800C079FE61 -> 1263
30904  C101901FC40F8800 -> 716
30912  CE700001C1C00000 -> 704
30920  C4038E0400700000 -> 457
30928  080000800FC3CFE4 -> 781
30936  0073CFF8080FC384 -> 920
30944  0FF3CFE7F3F3CFFF -> 1644
30952  3F3FC0F9F3FF3CFE7 -> 1236
30960  F3F3CFFF3F3FC0F9F -> 1440
30968  3FF388000040003F -> 569
30976  0001C00000803800 -> 340
30984  004000200001C000 -> 289
30992  080399F9FE4FFF20 -> 1033
31000  3FF9C9FE487F9900 -> 1119
31008  024803267E01C300 -> 442
31016  4841990002400326 -> 397
31024  7E01C8004E419900 -> 683
31032  4241F326007FC9FE -> 994
31040  424F993C4049F325 -> 776
31048  007FC800424F9924 -> 661
31056  4049C327FE780800 -> 753
31064  404F983C4249C300 -> 689
31072  FE000E49C04F9828 -> 804
31080  4249CF000201C049 -> 614
31088  0FC0F992442498F3C -> 753
31096  03CFC048070CF9922 -> 875
31104  7209803CF04FC248 -> 896
31112  07C3810002098020 -> 602
31120  70780201E4038100 -> 995
31128  02499FE01E0300200 -> 490
31136  040399FE7E499FE6 -> 1002
31144  1E01927804C39800 -> 648
31152  00499FE79FE19001 -> 992
31160  3CC3980000400F00 -> 486
31168  8001900100CF9F3F -> 703
31176  FFC00F008000393C9 -> 941
31184  00CF8300000FCFFC -> 812
31192  C26780093FCF8300 -> 835
31200  000FC3E006678008 -> 679
31208  07CFF03FFFFFC000 -> 1219
31216  0007E3C0007FFF03F -> 991
31224  FFFFFFF042007E3C1 -> 1213
31232  E7FFFFFFF0FFFFFC -> 2013
31240  3FFFFFFF0000 -> 1338
31248  166FCH4530332461 -> 716
31256  13AAF0978E0C4839C -> 1205
31264  F7AD06C92759288E -> 988
31272  F792DE4114062E90 -> 1088
31280  6898B1FEA8564665 -> 1199
31288  8261A1506469E690 -> 1060
31296  5D70B410603F95DE -> 1036
31304  2FC1AB2EFDFAE17D9 -> 1095
31312  F1F80DEAE0FBC585 -> 1605

```

```

31320  5D4F386CAAFBDA2C -> 1022
31328  CB8E0EA41968A874 -> 1176
31336  1EBEE810232914ED -> 813
31344  AE2551AE1100C361 -> 789
31352  66F0A3FC4120618E -> 1090
31360  C1BA8F2CFBD1823E -> 1218
31368  3F76C5D6FC221D79 -> 1028
31376  907D075C3314E297 -> 1024
31384  69DDFC40C7A16EAD -> 1273
31392  F7C09A3584D87ADE -> 1338
31400  3CB3C78942646169 -> 943
31408  E273AE3735991D64 -> 905
31416  8EE28684F1F1E7FC -> 1599
31424  1F411AA9FC35806 -> 1024
31432  2222EEFF6E689400 -> 1052
31440  15C1057A0E00804 -> 559
31448  00FC16EA16F41676 -> 914
31456  16010000F5D5C507 -> 685
31464  0707E60721CE7747 -> 680
31472  CB50280104C89011 -> 692
31480  0400373FED521910 -> 482
31488  FDCB572804112001 -> 637
31496  19C1D1F1C9C5D5E5 -> 1508
31504  CD4F7B8383679E61F -> 899
31512  FE1F200079070707 -> 472
31520  E6073C0F0F0F4F18 -> 445
31528  010CC04F7B381C04 -> 508
31536  78FE18200D060079 -> 570
31544  0707073C3C3C0F0F -> 231
31552  0F4FCD4F7B38040D -> 574
31560  CD4F7BE1D1C1C979 -> 1356
31568  CDE47A1110C000C504 -> 765
31576  AFED521910FDC179 -> 1102
31584  E61FCB3FCB3FCB3F -> 1059
31592  5F16001979E60758 -> 588
31600  473E0004371F10FD -> 492
31608  43A62002AFC937C9 -> 899
31616  DD21D07A21D07A36 -> 1001
31624  072336C7233605D0 -> 610
31632  360550D0360616CD -> 647
31640  6D7E1829CF02A70C -> 688
31648  D206D000BAE0B2A0A -> 672
31656  D00AD0129A0C3515 -> 684
31664  D2095A1558152C11 -> 500
31672  AA114310D4003605 -> 762
31680  50D0360616C0507C -> 782
31688  01FE00E078E60FFE -> 1111
31696  0FCA0D70CB572025 -> 714
31704  DD361101D07E054F -> 724
31712  E61FFE1E200BDD0C6 -> 1012
31720  00F621C47579F61F -> 990
31728  4F0CDD4606C000078 -> 729
31736  DA0D7D187ACB5F20 -> 832
31744  22DD361102D04E05 -> 632
31752  DD46060D79E61FFE -> 946
31760  1F200800D0CB00F6 -> 754
31768  213075CDD0D7BDA0D -> 770
31776  7D1854CB472029D0 -> 801
31784  361103D07E06FE00 -> 681
31792  20110D7E05D06604F -> 790
31800  061600CB00F621EE -> 969
31808  7618054705DD4E05 -> 527
31816  CD0D7BDA0D7D1827 -> 760
31824  DD7E06D03611043C -> 709
31832  FE1720110600D07E -> 679
31840  05C6604FDDCB00F6 -> 1048
31848  215276180447DD4E -> 631
31856  05CD0D7BDA0D7DC5 -> 899
31864  DDCB0076280A11AE -> 783
31872  7C05E579CDE47AC9 -> 1443
31880  DD4606D07E05E61F -> 910
31888  4F16001E001808C5 -> 360
31896  D5CD8877D1C1C9CD -> 1481
31904  977C00C0977C004C0 -> 976
31912  977C00C0977C1DD -> 1182
31920  7105D07006DD4611 -> 765
31928  3EFA060810FDD0CB -> 1210
31936  005E2808C604D0CB -> 768
31944  009E1804D0CB00DE -> 832
31952  571E460D4606D07E -> 831
31960  05E61F4FCD0977C14 -> 845
31968  0CDD977C1404C097 -> 872
31976  7C1400C0977C1E28 -> 707
31984  0E03DDCB005E2804 -> 579
31992  1E500E02433E10D3 -> 482

```

```

320000 FE10FE433E000D3FE -> 1118
320008 10FE0D20EFD0C800 -> 978
320016 B6D07E05E6E0FEA0 -> 1402
320024 2020D07E05E6E07C6 -> 846
320032 405F06170E0E1622 -> 272
320040 CD977C140CCD977C -> 992
320048 140CCD977C140CCD -> 749
320056 977C47DD7E03E6E0 -> 1150
320064 B62025D0C8006E20 -> 819
320072 251E451626DD7E03 -> 546
320080 E61F4FDD4604CD97 -> 991
320088 7C140CCD977C1404 -> 660
320096 CD977C140CCD977C -> 993
32104 DDCB006E231A1EC5 -> 827
32112 162606160E00CD97 -> 458
32120 7C140CCD977C1404 -> 660
32128 CD977C140CCD977C -> 993
32136 3A795CDD4602B833 -> 804
32144 2ADD7E0130FE0C20 -> 929
32152 04010000C9DD7701 -> 547
32160 21000022785C1E1E -> 339
32168 0E64433E17D3FE10 -> 747
32176 FE433E00D3FE10FE -> 1118
32184 0D20EFD05E010E00 -> 614
32192 0600182ACD8877DD -> 751
32200 C6006E202E0D7E03 -> 741
32208 DD4605B882025D07E -> 896
32216 04D04606B3201CDD -> 786
32224 C600EE160506FF58 -> 817
32232 3E1303FE10FE3E00 -> 876
32240 D3FE4310FE4310EF -> 1124
32248 1520EAD0C8006E0A -> 1023
32256 C37E0D7E05E6E0FE -> 1381
32264 A0C2C37E0D7E06FE -> 1282
32272 16C2C37E0D7E05E6 -> 1119
32280 1FFE12D2C37E0E0E -> 1102
32288 DAC37E0D7E003D0D -> 1168
32296 7700E507FE022004 -> 648
32304 010100C921397E18 -> 443
32312 145A8446783B8C2C -> 643
32320 A03B8C46785A8446 -> 809
32328 783B3C2C0A1E0A56 -> 649
32336 234E23423E11D3FE -> 758
32344 10FE423E05D3FE10 -> 884
32352 FE0D20EF1D20E8CD -> 1036
32360 6D7E0C3C37E0DCB00 -> 1175
32368 AE16011E1706160E -> 892
32376 00CD977C00CD977C -> 972
32384 04CD977C00CD977C -> 977
32392 CD67773A765CE61F -> 1036
32400 FE1730F4473A775C -> 909
32408 4FE6E0FEA028E9FE -> 1474
32416 0028E5FEE028E157 -> 1099
32424 DD7E05E6E0BA28D8 -> 1248
32432 79E61FFE1F30010D -> 1129
32440 0D7B38CDD70040D -> 954
32448 7103C9DD3408D7E -> 846
32456 080E00FE02D0CB00 -> 702
32464 A62008D0CB006E6D -> 1081
32472 360800219B78D035 -> 647
32480 072004D03607043E -> 391
32488 14DD960247C623CD -> 901
32496 0F80D0CB006E6CA5E -> 962
32504 7F7EE6E047DD7E05 -> 1130
32512 E6E0B8202DD0CB00 -> 1139
32520 5E28277ED0BE0528 -> 771
32528 0F38033D18013C4F -> 899
32536 23462BDD0D783039 -> 894
32544 237E2BDD06E08280A -> 671
32552 38033D18013C474E -> 354
32560 1824CDB7773A775C -> 836
32568 E603233EFE012849 -> 703
32576 FE022822237E28FE -> 788
32584 0020087ED6604F06 -> 561
32592 1618033D474ECD0D -> 477
32600 7B303B4E23462B05 -> 653
32608 C3C47FC3967F237E -> 1451
32616 26FE16200806007E -> 491
32624 C6604F18E13C474E -> 831
32632 18DC7E4FE51FFE1E -> 994
32640 20010C0E23462B18 -> 829
32648 CD7E4FE61F20010D -> 717
32656 0D23462B180C57E -> 700
32664 E6E047DD7E05E6E0 -> 1331
32672 B82021D1C10C05D5 -> 1073

```

```

32680 7EE61F4F23462B16 -> 636
32688 001E00E5C0977C0C -> 751
32696 CD977C04CD977C0D -> 977
32704 CD977C0E1C1712370 -> 1158
32712 2B79E6E057D07E05 -> 1057
32720 E6E0BA202E5059C1 -> 1081
32728 0CC54E3E27DD4607 -> 683
32736 C80410FC42573E43 -> 752
32744 DD86075F79E61F4F -> 918
32752 E5CD977C0014CD97 -> 1097
32760 7C0414CD977C0014 -> 661
32768 CD977C0E1C00F80C1 -> 1246
32776 0523C2ED07E182C7E -> 791
32784 DDBE05C0237E2BDD -> 1033
32792 BE06C0E5C06D7EE1 -> 1282
32800 06FF583E16D3FE10 -> 914
32808 FE6E093473E02D3 -> 1064
32816 FE10FE4310EC3E00 -> 905
32824 D3FE094104210028 -> 808
32832 111E01AFED5210FB -> 809
32840 DDCB006E231A1EC5 -> 827
32848 06AFED523E002B8C -> 793
32856 20FCC3C87B219C7B -> 1114
32864 160FCDB7773A775C -> 813
32872 4FE6E028F5FEA000 -> 1232
32880 00FEE028E079E61F -> 1137
32888 FE1F28E6CDB7773A -> 1120
32896 775CE61FFE1730F4 -> 1041
32904 47CD0D7B38D47123 -> 828
32912 70231520C0C3807B -> 851
32920 0000000000000000 -> 0
32928 FF101010FF040404 -> 570
32936 07071F070F0F0703 -> 92
32944 C0C0F0C060E04080 -> 1328
32952 E00D0C080800E0E0 -> 1504
32960 0C1B37070706090D -> 142
32968 07071F070F0F0703 -> 92
32976 C0C0F0C060E04080 -> 1328
32984 C04020A0C0406030 -> 648
32992 070E170703070607 -> 74
33000 03030F0306070201 -> 40
33008 E0E0F0E0F0E0C0C0 -> 1808
33016 50D8ECE0E0D090B0 -> 1604
33024 070E0B0301010307 -> 44
33032 03030F0306070201 -> 40
33040 E0E0F0E0F0F0E0C0 -> 1808
33048 E070E8E0C0E060E0 -> 1528
33056 030204050302080C -> 37
33064 0303070004040001 -> 22
33072 E0E0F000101000C0 -> 920
33080 30F8F8F0F0703038 -> 1320
33088 000F080B07060E00 -> 64
33096 07071F0003080003 -> 64
33104 C0C0F0C060E04080 -> 816
33112 00F0C0D0E060E000 -> 1200
33120 011F1F0F0F0E0C1C -> 147
33128 03030F0306070201 -> 40
33136 E0E0F8E0B0F020C0 -> 1560
33144 90D8C8D0F0703038 -> 1224
33152 0C1D352507060E00 -> 158
33160 07071F070D0F0403 -> 87
33168 C0C0F0C060E04080 -> 1328
33176 30B8AC84E0607000 -> 1000
33184 091B13060F0E0C1C -> 135
33192 7FBF1F0F07030100 -> 375
33200 FF7F7F1C5D1CFFFF -> 1168
33208 FFFEFE787A38FFFF -> 1571
33216 FEFDF8F8F0F0E0E0 -> 1931
33224 000000070810101F -> 78
33232 000000E0100808F8 -> 504
33240 F81CF070E020E000 -> 1108
33248 1F1E0E0E05060700 -> 108
33256 FFF81E7E7E7E7E7F -> 1818
33264 000000000C1E1F0D -> 86
33272 000000003078F860 -> 592
33280 F0F8783000000000 -> 656
33288 0F1F1E0C00000000 -> 88
33296 0000183C3F1F190E -> 214
33304 0000183C3F896D0 -> 944
33312 F0D838FC3C180000 -> 848
33320 0F1B1C3F3C180000 -> 217
33328 0000183C3F1F190E -> 214
33336 0000183C3CF896D0 -> 944
33344 F0D838FC3C180000 -> 848
33352 0F1B1C3F3C180000 -> 217

```

## Hexadezimal-Listing »Mc Wash« (Schluß)

## Hardcopy mit Interface 1 und Matrix-Drucker

Das Programm erstellt wahlweise Hardcopies in Normalgröße (27 Sekunden) oder mit 4facher Vergrößerung (1 Minute 26 Sekunden) durch die selbstdefinierte Systemvariable ADDR 23681. Es ist 198 Byte lang, wird über RAMTOP gestartet und eignet sich für Spectrum (16 oder 48 KByte) mit Interface 1 und Epson-Drucker FX-80.

Zum Bau eines RS232-Verbindungskabels für Epson FX-80 mit RS232C-Interface kaufen Sie:

- a) ein fünfadriges, abgeschirmtes Kabel von zirka 1 Meter Länge
- b) einen 9-Pin-D-Stecker
- c) einen 25-Pin-D-Stecker

Löten Sie die Stecker so ans Kabel, daß die Pins wie folgt belegt sind; verwenden Sie den Schirm für Masse (Ground):

ZX Spectrum	—	Epson FX-80
Signal Pin	—	Pin Signal
TX Data	2 — 2	TXD
RX Data	3 — 3	RXD
DTR	4 — 20	DTR
CTS	5 — 11	2nd RTS
Ground	7 — 7	GND
+9V	9 — 6	DSR

Stellen Sie die DIP-Schalter des Epson-RS232C-Interface folgendermaßen ein:

Pin	Stellung
1-1	off
1-2	off
1-3	off
1-4	off
1-5	off
1-6	off
1-7	off
1-8	on

Das entspricht 19200 Baud Übertragungsgeschwindigkeit.

### So benutzt man das Programm

- Programm in Betrieb nehmen
- FORMAT "b";19200: OPEN #3;"b"
- CLEAR 64999 beziehungsweise 32199
- LOAD "hardcopy"CODE
- Programm aufrufen
- RANDOMIZE USR 65000 beziehungsweise 32200
- Wählen Normal/4fach
- POKE 23681,n
- Normal:  $0 \leq n \leq 127$
- 4fach:  $128 \leq n \leq 255$

### Programmeingabe

— Maschinencode eingeben

Das Programm verwendet nur relative Sprünge, jedoch ist der Hex-Code für »CALL OUTPUT« beim 16-KByte-Spectrum CD617E. Geben Sie CLEAR 64999 beziehungsweise 32199 ein und laden dann das Programm mit einem Assembler oder Hex-Lader (Sie haben sicher eins von beiden) über RAMTOP. Anschließend NEW.

— Basic-Teil für Steuercodes

Das Basic-Programm setzt für DATA1 bis DATA4 die wirklichen Adressen ein, POKET die Tabelle der Druckersteuercodes anschließend an das mc-Programm und SAVET das vollständige Hardcopy-Programm auf Kassette.

— Microdrive-Version

Ersetzen Sie in Zeile 9999 SAVE mit SAVE "\*"m";1;

Laden Sie das Programm nun mit LOAD "\*"m";1;"hardcopy"

CODE

— Sie wollen andere Steuercodes?

Die Data-Zeilen des Basic-Teils sind nach folgendem Schema aufgebaut:

Zeilennummer DATA ramtop+Zahl, n, c(1), c(2), ....., c(n)

c(1), c(2),: einzelne Druckersteuercodes

n: Anzahl der Steuercodes in dieser Data-Zeile

Sie können sich also Ihre individuelle Version herstellen, wenn Sie die Data-Zeilen entsprechend abändern.

— Sie wollen 24 Zeilen ausdrucken?

Fügen Sie zum Basic-Programm hinzu

9990 POKE ramtop+130,192

### Steuerzeichen

Zeile 100:

27,33,0 = ESC ! 27,64,0

Einschalten des Normalmodus

✓ 27,108,4 = ESC 1

Linken Rand so setzen, daß die 4f-Copy in die Mitte des Blattes gedruckt wird.

✓ 27,81,80 = ESC Q

Rechten Rand ganz außen setzen, damit genügend Platz für die Copy vorhanden ist.

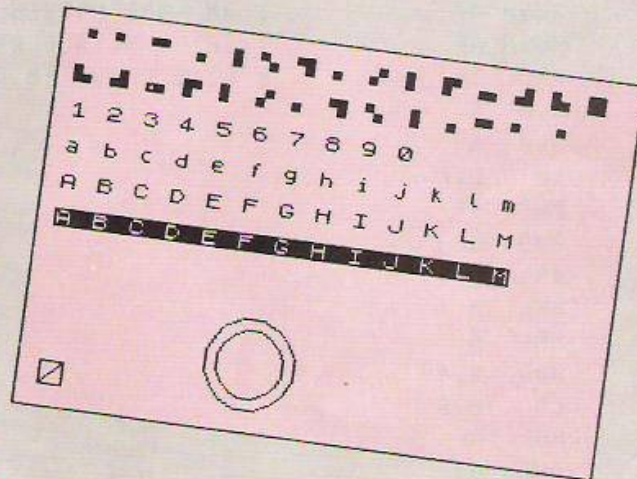
```

1 REM ZX Spectrum Hardcopy
2 REM
3 REM Copyright
4 REM 1984-07 Gunther Ritz
5 REM
10 LET ramtop=PEEK 23730+256*P
EEK 23731: LET add=ramtop+168
20 FOR f=1 TO 4
30 READ data,len
40 POKE data+1,INT (add/256):
POKE data,add-256*INT (add/256)
50 POKE add,len
60 FOR n=1 TO len
70 READ d: POKE add+n,d
80 NEXT n
90 LET add=add+len+1
99 NEXT f
100 DATA ramtop+7,12,27,33,0,27,
,108,4,27,81,80,27,65,8: REM dat
a1
110 DATA ramtop+161,5,27,108,0,
27,50: REM data2
120 DATA ramtop+140,5,27,42,5,0,
,1: REM data3
130 DATA ramtop+99,5,27,42,5,0,
2: REM data4
9999 SAVE "hardcopy"CODE ramtop+
1,add-(ramtop+1)

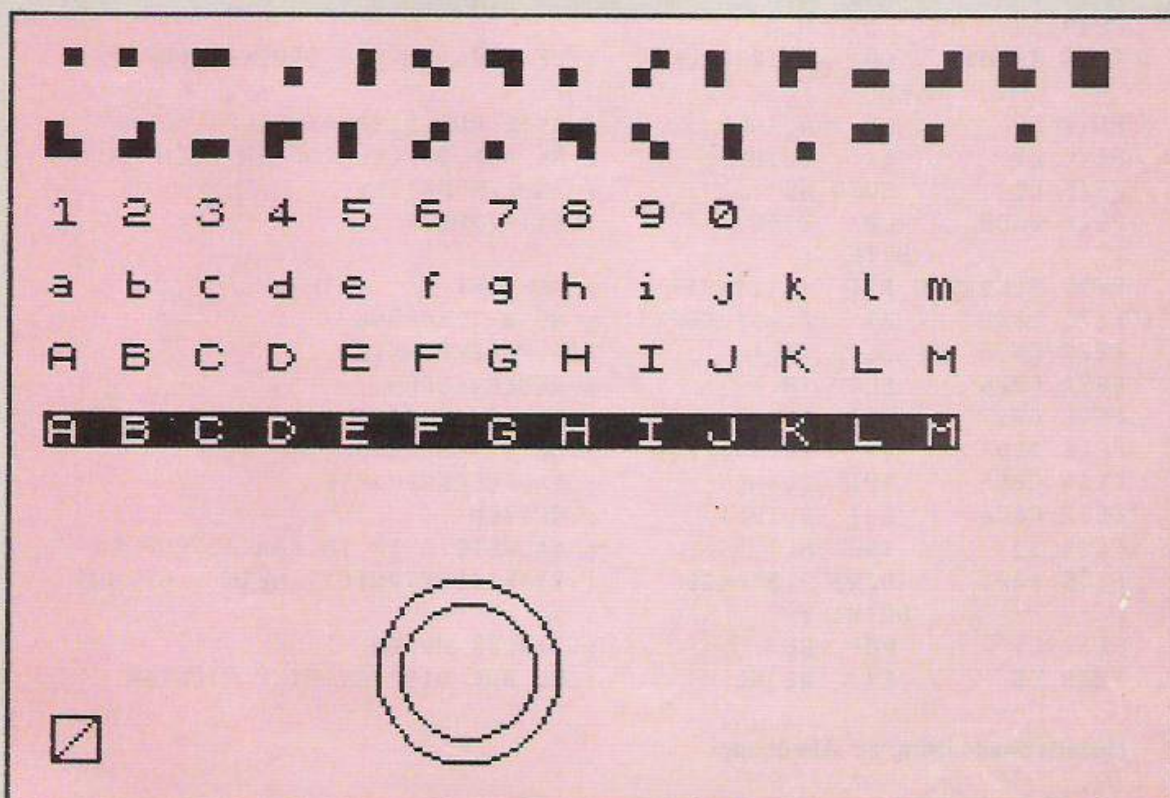
```

Basic-Listing  
zu »Hardcopy«

Musterausdruck  
in Normalformat ...



... und vierfach vergrößert.



```

Addr Hex      Op  Operand      ; Notiz

PTRSEL:
FDEB 3E03     LD  A,03      ; DRUCKERAUSGABE
FDEA CD0116   CALL 1601     ; WAHLEN

PTRINI:
FDED 210000   LD  HL,DATA1 ; STEUERCODES FÜR NORMALMODUS,
FDF0 46       LD  B,(HL)    ; LINKEN UND RECHTEN RAND SOWIE
FDF1 CD81FE   CALL OUTPUT  ; ZEILENVORSCHUB AUSGEBEN

START :
FDF4 AF       XOR  A        ; PIXELREIHENZÄHLER AUF 0
FDF5 F5       PUSH AF      ; UND RETTEN

LINE :
FDF6 0608     LD  B,08      ; PIXELREIHEN/DRUCKERZEILE
FDF8 FDCB477E BIT 7,(1Y+71) ; 4F-COPY ?
FDFC 2802     JR  Z,+2>CONV ; NEIN > SPRUNG
FDFE 0604     LD  B,04

CONV :
FE00 F5       PUSH AF      ; AM ENDE DIESES ABSCHNITTS
FE01 F5       PUSH AF      ; ZEIGT HL AUF DEN BEGINN DER
FE02 E638     AND  38      ; PIXELREIHE IM DISPLAY-FILE.
FE04 CB27     SLA  A
FE06 CB27     SLA  A
FE08 6F       LD  L,A
FE09 F1       POP  AF
FE0A E6C0     AND  C0
FE0C CB3F     SRL  A
FE0E CB3F     SRL  A
FE10 CB3F     SRL  A
FE12 C840     ADD  A,40
FE14 67       LD  H,A
FE15 F1       POP  AF
FE16 E607     AND  07
FE18 84       ADD  A,H
FE19 67       LD  H,A
FE1A 11005B   LD  DE,PTRBUF ; DE AUF DEN DES DRUCKERPuffers
COL :
FE1D 7E       LD  A,(HL)   ; BYTE AUS DISPLAY-FILE LADEN
FE1E EB       EX  DE,HL ; HL AUF DRUCKERPuffer RICHTEN
FE1F C5       PUSH BC   ; ZÄHLER RETTEN
FE20 0608     LD  B,08      ; BIT-ZÄHLER

BYTE :
FE22 FDCB477E BIT 7,(1Y+71) ; 4F-COPY ?
FE26 283D     JR  Z,+61>COPY1 ; NEIN > SPRUNG
FE28 CB26     SLA  (HL) ; 2 PAPERPUNKTE
FE2A CB26     SLA  (HL) ; RESERVIEREN
FE2C CB27     SLA  A ; PIXEL GESETZT ?
FE2E 3004     JR  NC,+4>FE34 ; NEIN > SPRUNG
FE30 CBCE     SET  1,(HL) ; 2 DRUCKERPUNKTE
FE32 C8C6     SET  0,(HL) ; SETZEN
FE34 23       INC  HL ; NÄCHSTE ADDR IM DRUCKERPuffer
FE35 10F1     DJNZ -15>FE2B ; BYTE ABGETASTET? NEIN > SPRUNG

BOTH1 :
FE37 C1       POP  BC   ; ZÄHLER HOLEN
FE38 EB       EX  DE,HL ; HL AUF DISPLAY-FILE RICHTEN

```

Hexadezimal-Listing zu »Hardcopy«



```

FE39 23      INC HL           ; NACHSTES BYTE DER PIXELREIHE
FE3A CB52    BIT 2,D          ; PIXELREIHE ZU ENDE ?
FE3C 28DF    JR Z,-33>COL     ; NEIN > SPRUNG
FE3E F1      POP AF          ; PIXELREIHENZÄHLER HOLEN,
FE3F 3C      INC A           ; ERHÖHEN
FE40 F5      PUSH AF         ; UND WIEDER RETTEN
FE41 10BD    DJNZ -67>CONV    ; ZEILE FERTIG ? NEIN > SPRUNG
PRINT :
FE43 FDCB477E BIT 7,(IY+71)  ; 4F-COPY ?
FE47 2829    JR Z,+41>COPY2   ; NEIN > SPRUNG
FE49 210000  LD HL,DATA4      ; DRUCKERSTEUERCODE FÜR
FE4C 46      LD B,(HL)        ; PLOTTERGRAFIK-MODUS MIT
FE4D CDB1FE  CALL OUTPUT      ; 512 DOTS BREITE AUSGEBEN
FE50 210058  LD HL,PTRBUF     ; HL AUF DRUCKERPUFFERBEGINN
FOUTPT:
FE53 7E      LD A,(HL)        ; 1 BYTE
FE54 07      RST 10           ; AUSDRUCKEN
FE55 7E      LD A,(HL)        ; NOCHMAL
FE56 07      RST 10           ; AUSDRUCKEN
FE57 23      INC HL           ; NACHSTES BYTE
FE58 10F9    DJNZ -7>FOUTPT   ; ZEILE BEDRUCKT? NEIN > SPRUNG
BOTH2 :
FE5A 3E0A    LD A,0A          ; LINE FEED
FE5C 07      RST 10           ; AUSGEBEN
FE5D F1      POP AF          ; PIXELREIHENZÄHLER HOLEN
FE5E FE80    CP B0            ; ALLES AUSGEDRUCKT ?
FE60 2825    JR Z,+37>END     ; JA > ABSCHLIESSEN
FE62 F5      PUSH AF         ; PIXELREIHENZÄHLER RETTEN
FE63 1891    JR -111>LINE     ; NACHSTE DRUCKERZEILE
COPY1 :
FE65 CB26    SLA (HL)         ; PAPERPUNKT RESERVIEREN
FE67 CB27    SLA A            ; PIXEL GESETZT ?
FE69 3002    JR NC,+2>FE6D    ; NEIN > SPRUNG
FE68 C8C8    SET 0,(HL)       ; DRUCKERPUNKT SETZEN
FE6D 23      INC HL           ; NACHSTE ADDR IM DRUCKERPUFFER
FE6E 10F5    DJNZ -11>COPY1   ; BYTE ABGETASTET? NEIN > SPRUNG
FE70 18C5    JR -59>BOTH1     ; JA > WEITER
COPY2 :
FE72 210000  LD HL,DATA3      ; DRUCKERSTEUERCODE FÜR
FE75 46      LD B,(HL)        ; PLOTTERGRAFIK-MODUS MIT
FE76 CDB1FE  CALL OUTPUT      ; 256 DOTS BREITE AUSGEBEN
FE79 21FF5A  LD HL,PTRBUF-1   ; 1 ZEILE GRAFIK
FE7C CDB1FE  CALL OUTPUT      ; AUSDRUCKEN
FE7E 18D9    JR -39>BOTH2     ; WEITER
OUTPUT:
FE81 23      INC HL           ; AUSGABE-ROUTINE:
FE82 7E      LD A,(HL)        ;
FE83 07      RST 10           ; B ENTHÄLT LANGE
FE84 10FB    DJNZ -5>OUTPUT   ; HL ENTHÄLT START-1
FE86 C9      RET
END :
FE87 210000  LD HL,DATA2      ; STEUERCODES FÜR
FE8A 46      LD B,(HL)        ; DRUCKERNORMIERUNG
FE8B CDB1FE  CALL OUTPUT      ; AUSGEBEN
FE8E C9      RET             ; RÜCKSPRUNG

```

Hexadezimal-Listing zu »Hardcopy«

65000	3E03CD0116218FFE	->	723
65008	46CD81FEAFF50608	->	1092
65016	FDCB477E28020604	->	705
65024	F5F5E638CB27CB27	->	1280
65032	6FF1E6C0C63FCB3F	->	1306
65040	CB3FC64067F1E607	->	1109
65048	846711005E7EEBC5	->	901
65056	0608FDCB477E283D	->	768
65064	CB26CB26CB273004	->	776
65072	CB0CE0BC62310F101	->	1295
65080	EB23CB5228DFF13C	->	1119
65088	F510BDFDCB477E28	->	1143
65096	2921A8FE46CD81FE	->	1154
65104	21005E7ED77ED723	->	841
65112	10F93E0AD7F1FE60	->	1223
65120	2825F518910626CB	->	935
65128	273002CBC62310F5	->	786
65136	18C521A2FE46CD81	->	1074
65144	FE21FF5ACD81FE18	->	1244
65152	D9237E0710FB0921	->	1094
65160	9CFE46CD81FEC90C	->	1281
65168	1B21001B6C041B51	->	307
65176	501B4108051B6C00	->	320
65184	1B32051B2A050001	->	157
65192	051B2A0500020000	->	81

## Maschinencode-Listing zu »Hardcopy«

Falls Sie bei Ihrem Drucker die Ränder nicht setzen können, lassen Sie die Steuercodes dafür einfach weg.

- ✓ 27,65,8 = ESC A  
Zeilenabstand auf acht vertikale Dotabstände setzen. Dadurch schließen die Druckzeilen lückenlos aneinander.  
Zeile 110:
- ✓ 27,108,0 = ESC 1  
Linken Rand ganz außen setzen. Falls Sie bei Ihrem Drucker

die Ränder nicht setzen können, lassen Sie diesen Steuercode einfach weg.

27,50 = ESC 2

Zeilenabstand auf Normalabstand setzen.

Diese Steuercodes werden ausgegeben, nachdem die Copy erstellt ist, um den Drucker wieder in seinen Normalzustand zu versetzen.

Zeile 120 und Zeile 130:

27,42,5,n1,n2 = ESC \* 5

Einschalten des 8-Nadel-Plottergrafik-Modus. Die Zahlen n1 und n2 errechnen Sie folgendermaßen:

n1 = Byteanzahl — 256 \* INT (Byteanzahl / 256)

n2 = INT (Byteanzahl / 256)

Byteanzahl pro Zeile:

256 bei Normalgröße in Zeile 120

512 bei 4fach in Zeile 130

Im Plottergrafik-Modus sind beim FX-80 die Dotabstände vertikal und horizontal gleich, das heißt es entsteht eine 1:1-Copy des Bildschirms ohne Verzerrungen.

Wählen Sie bei Ihrem Drucker einen dem Plottergrafik-Modus des FX-80 entsprechenden Modus und setzen Sie die entsprechenden Steuerzeichen in die Zeilen 120 und 130.

Mit Hilfe dieser Aufstellung können Sie das Programm an Ihren Drucker anpassen. Voraussetzung dafür ist allerdings, daß Ihr Drucker die Drucknadeln in der gleichen Weise wie der FX-80 ansteuert: obere Nadel = Bit 7.

Falls Ihr Drucker den Steuercode LINE FEED (CHR\$ 10) nicht kennt, schalten Sie Ihren Drucker auf automatischen Zeilenvorschub und fügen Sie zum Basic-Teil folgende Zeile hinzu:

9995 POKE ramtop+116,13

(Gunther Ritz)

# Tips und Tricks für den Spectrum

In diesem Artikel will ich eine schon überfällige Zusammenstellung nützlicher Tips für den Spectrum veröffentlichen, da ich diese sehr häufig gebrauche.

Muß der Cursor beim Programmieren häufig und schnell bewegt werden, so empfiehlt es sich, dessen Geschwindigkeit mittels POKE 23562,2 zu erhöhen. Die normale Geschwindigkeit ist durch POKE 23562,5 leicht wieder herzustellen. Der Tastatürklick ist durch POKE 23609,6 zu verlängern. Der POKE 2365,8 schaltet auf Großbuchstaben um, der POKE 23658,0 auf Kleinbuchstaben. Durch RANDOMIZE USR 0 wird der RAM Speicher vollständig gelöscht, so daß sich ein

Herausziehen des Steckers erübrigt. Durch POKE 23692,255 kann der Bildschirm 255 mal abgescrollt werden, ohne daß dieser die SCROLL? Frage stellt. POKE 23617,2 schaltet in den Grafik-Modus u. POKet man in dieselbe Adresse eine 0, so erhält man den erweiterten Modus. In der Adresse 23560 steht der Code der zuletzt gedrückten Taste. Möchte man die Anfangskordinaten einer Draw Anweisung bestimmen, so braucht man nur in die Adresse 23677 die x und in die Adresse 23678 die y Koordinate zu POKEn. Möchte man sein Basic-Programm vor fremden Zugriff schützen, so bietet POKE 23659,0 eine Möglichkeit. Drückt der Programm benutzer nämlich dann die BREAD-Taste, so gibt es einen Systemabsturz. Nur sollte man tunlichst darauf achten, daß während dem Ablauf des Programmes keine Meldung auf der untersten Zeile (der 24ten) erscheint. Auch eine Input Anweisung aus dem ROM sollte man vermeiden, oder vorher den POKE 23659,2 durchführen. Mit dem Befehl PRINT PEEK 23730+256\* PEEK 23731 läßt sich der derzeitige RAM-TOP feststellen. Mittels PRINT USR »a« läßt sich der Beginn der UDG Area feststellen. Um festzustellen, wieviel Bytes ein Basic-Programm verbraucht, bedarf es dem Befehl »PRINT«. Das Programm ist »;256\* (PEEK 23628-PEEK 23636)+ PEEK 23627-PEEK 23635;« Bytes lang. Um ein ZX81-Scroll zu simulieren, muß man nur die beiden Befehle POKE 23692,255: PRINT AT 21,31 ' ' AT 21,0; eingeben. Gibt man sehr lange Strings innerhalb eines Programmes ein (länger als der Bildschirm), so ist es ratsam, den Warnton auf ein Minimum zu reduzieren. Dies geschieht durch POKE 2308,0. Durch diese Tricks kann das Programmieren bequemer gestaltet werden.  
(Knut Bachmann)

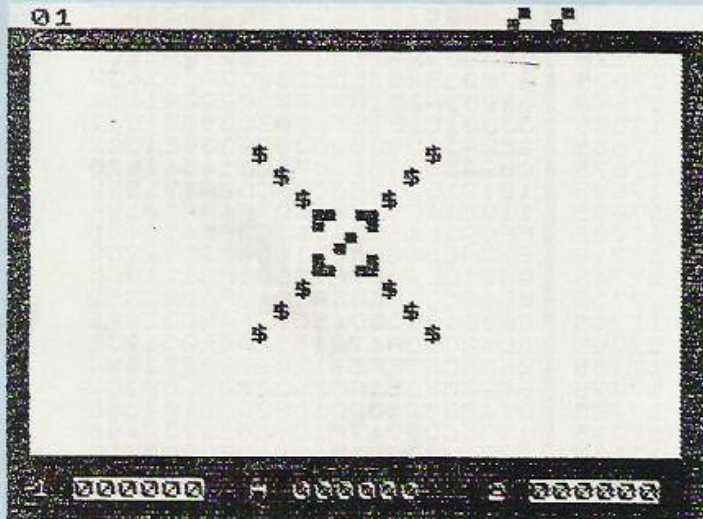


# Geisterjagd

## Ein »geistreiches« Spiel für den 16 KByte-ZX81

Ziel des Spiels: Punkte, Punkte, Punkte... Es gibt 12 Gegner. In der ersten Runde sind sie durch das String-Zeichen dargestellt. Später wechselt das von Runde zu Runde. Spieltasten sind »S, D, E und X«, was mir das Schnellste erschien. Mit »P« kann man das Spiel unterbrechen und mit NEWLINE beginnt es von vorn (Ausstieg aus der Schleife ist mit RUBOUT möglich, was aber keiner wissen muß!).

In der Bildmitte ist mein »Haus«. Hier beginne ich das Spiel. Auf Tastendruck ziehe ich aus, um mir die »Geister« einzufangen, die wirt über den »Acker« zappeln. Je nachdem, wie ich ihnen begegne, sehen sie mich und weichen aus. Es gilt, möglichst viele zu fangen, bevor das S blinkt. Das S ist der Schlüssel zu meinem »Haus«, ohne den ich nicht mehr hineinkomme. Wenn ich es verpasse, ihn zu holen, kommen alle Geister, die ich bisher (in diesem Durchgang) gefangen habe, wieder frei.



Spielbeginn

Habe ich den Schlüssel, passiert aber »Schreckliches«: Die übrigen Geister machen sofort kehrt und kreisen um mein Haus — mal links, mal rechts herum. Es wird mir recht schwer gemacht, hineinzukommen. Bin ich aber im Haus, so bleiben alle Geister, die ich in diesem Durchgang geschnappt habe, aus dem weiteren Spiel heraus. Sie werden über dem Spielfeld abgelegt. Während der ganzen Zeit verkürzt sich ein Zeitbalken, der unter dem Spielfeld liegt. Habe ich alle 12 Geister gefangen, bevor die Zeit abgelaufen ist, geht es in die 2. Runde! Schaffe ich es nicht, verliere ich meinen Spieler. Da ich deren drei besitze, die oben rechts am Rand auf ihren Einsatz warten, geht es mit der 2. Spielfigur weiter. Habe ich keine Spielfigur mehr, ist mein Spiel zu Ende und der 2. Spieler kommt zu seinem Recht. Wenn der Punktstand höher als der Höchststand ist, erfolgt ein Übertrag.

In der 2. bis 10. Runde erscheinen die »Geister« jeweils in einer anderen Verkleidung (Symbolwechsel!). In der 11. Runde beginnt das ganze von vorn.

Die Geschwindigkeit steigert sich von Runde zu Runde. Allerdings fällt sie nach je 10 Runden wieder ab, jedoch nicht auf den Anfangswert. Dieser wird erst nach der 100. Runde wieder erreicht, falls das je jemand schaffen sollte.



Spielende

\* ERSTELLUNG DES REM-BEFEHLS \*

GIB FOLGENDES PROGRAMM EIN UND STARTE MIT >RUN<

```
1 LET A$="0010350110420200640
5402900350542034035054000035011120
177032240205175020195012006"
2 FOR N=30000 TO 30025
3 POKE N,VAL A$( TO 3)
4 LET A$=A$(4 TO )
5 NEXT N
6 RAND USR 30000
```

GIB EIN: >POKE 16510,0<

JETZT KANN DAS LADE- UND LAUF-PROGRAMM VON 1 BIS 37 EINGEGEBEN WERDEN.

DANACH START MIT >RUN< UND EINGABE DES HEX-LISTING

> ZAPPEL < VON P. RICHTER

Programm zur Erstellung des REM-Befehls



\* \* H E X - L I S T I N G \* \*

>> Z A P P E L <<

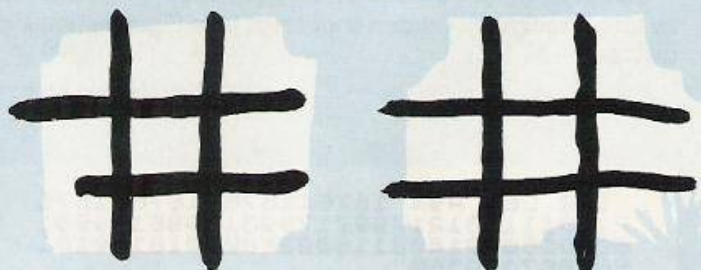
VON  
PETER RICHTER  
AM MAUBISHOF 22  
4044 KAARST 1  
02101/63257

16616	CD2A0ACD3D472A0C4011729	17335	360010FB06092310FD001373
16625	CF00193E97E5CD0A41E11189	17345	20F118B72B2C3A392A381510
16635	114200197719CD0A4137607	17355	000038352E2A31762B2C1191
16645	ED52771807060F72310690	17365	263700332E262D39002C1126
16655	FCC9117D44CD7A43CD1B1329	17375	3A39762E332600272E391270
16665	47119A44CD8430654CD1111	17385	392A762B55A6B88880A92042
16675	2F473E0C32E5403E9C2A855	17395	B4A8AD80A6B8A676292D2156
16685	0C4011B70219369D060A600	17405	34310029263800B800261253
16695	2310FD36AD2306052377811	17415	373A2A2D2A377628293A1354
16705	10FC2248462265482377896	17425	003C26373839003FA001197
16715	23232323369E3E1C2A0C596	17435	3126332C3826327626271346
16725	4011B802190606237710684	17445	00003700002600003800982
16735	FC2B2279414D06102310785	17455	0034760055293A002D261277
16745	FD0606237710FC2B227E1020	17465	38390002631312A001D1E1200
16755	41793278462100001803626	17475	7600582C2A2B26332C2A1370
16765	21000022E54622484622726	17485	337648292A2E330038351400
16775	624621124522AB41AF32943	17495	2E2A3100002E3839000311225
16785	CD400603CD9F4610FB2A1191	17505	2A2E292A377600733F001412
16795	0C402323361C233610CD730	17515	3A00002A00330029002A1134
16805	1F47CD7F462100007E32903	17525	762FB5A65AB8AA764DBF2348
16815	C54023FE0F2005060A2B861	17535	80A630B580B580A80B12435
16825	10FD22AB41ED5B0C402A1195	17545	760078B58A89A8780B72358
16835	31471313131AFE1C200B748	17555	AEA8AD89A8B7760786002252
16845	1B1AFE1C200C21C80A18876	17565	2B26332C00292E2A002C1299
16855	0D063C2B10FD18060696817	17575	2A2E38392A37000676001382
16865	232310FC22231473E262A884	17585	20E8002D343100292A331479
16875	0C4023232334BE200A2B768	17595	0038282D313A2A38382A1424
16885	34BE2002361C23361C2A787	17605	3100B876002E292633331568
16895	0C4006442310FD0E1206772	17615	002C2A2D003FA3700321357
16905	1E36002310FB2323230D794	17625	2E39392A76003026272A1497
16915	20F32A0C401116011911775	17635	370035263838002C3A391437
16925	1E0001220022D2400936746	17645	00263A2B1A760030292A1444
16935	072322E0402336842322974	17655	33331B1B1B76005C38351542
16945	D8401922DA402322E2401310	17665	2E2A31392638392A330E1502
16955	2322DE401922D64023361121	17675	002A38293D35760D14341516
16965	822322DC4023368109221094	17685	1517163D0E0C0FC0A471524
16975	D4402A0C4006122310FD1082	17695	110E44CD5743CD7346FE2194
16985	060D36000000002B10F8750	17705	FFCA3E46CD1147C7FE412435
16995	7832C8402A0C403AC5401251	17715	E13AC5400E0B12C680F52258
17005	4F118E42060CC51A233D1031	17725	0601CD2F47F10D20F3111986
17015	20FC2282403A7A4206051169	17735	EC43CD4C433ACD40FE002352
17025	3C10FD327A42C17113101318	17745	CA3B46CD5D45CD7446C32425
17035	E7180CD2081A061004801097	17755	BB42CD0A472185400E0C1935
17045	041C061A083E82327A42932	17765	0605C07EFE012025282B1894
17055	2A0C401173001922BE401003	17775	562B5ED51AFE802808CD2257
17065	3E3832C14011050021851063	17785	A245882802D1D53AC5402368
17075	40060C36001910FB2A0C942	17795	12F5CD27471C168012732458
17085	4011F8021906203683231084	17805	2372232D1C12310FD00D2128
17095	10FB2B22CF403E0432D11420	17815	200C2ABE4036800D11472207
17105	4021CC40360A2B36402A1122	17825	C90E0106031B1B131AFE1786
17115	0C40115A01193E067722930	17835	80280A10780D2807793C1903
17125	C24032C440AF32E4403E1657	17845	280AC947C906221B10FD2089
17135	050E01060C21844030FE1102	17855	18E306411310FD180CDD2299
17145	0020023E0477F579238E1340	17865	0A47CD3043CD574321852176
17155	2001353D2377232323F11187	17875	40060CC57EFE012006342010
17165	10E87721C640702370231506	17885	ESCDF845E123232323232421
17175	23702336163E0132CE401201	17895	C110EC3AC840FE00CA182539
17185	CD8B47FE76CAE740FE002108	17905	46CD1147C30A422A0C402250
17195	28F4C39D493AC740FE001864	17915	06052310FD0237EB820FB2243
17205	200411D343C9473AC8401515	17925	3AC54077CD27470605CD2279
17215	80FE0C200411E243C9111558	17935	DF4610FB21C84034C9112447
17225	C543C90604CD6643CD861798	17945	2344CDA843CD1F473ACB2441
17235	4310F8C90604CD6243CD1737	17955	404FCDBF460602CDDF462455
17245	864310F8C93E0018023E1446	17965	10FBBCD27470D20F0D112439
17255	80C5D52A0C40062123101386	17975	47C3AA41CD0A47114F442311
17265	FD0620237710FC1828C51624	17985	CD843CD1F47210000112167
17275	D52A0C4006C82310FD181523	17995	000006042B1B10FC7E471925
17285	05C5D52A0C401A233D201357	18005	1AC680903806201423132054
17295	FC131AFE7628057713231567	18015	18F221000011000006051727
17305	18F6131A13FE0028E9CD1756	18025	7EC680122B1B10F8CD112436
17315	2747D1C1C9C5D52A0C401941	18035	473AES46FE00C25141C32637
17325	06A92310FD0E0C0618231280	18045	7D4121CD40357EFEFFC82810

# Geisterjagd

18055 D505F02A0C40062123102107  
 18065 FD060A2B7EFE062802102206  
 18075 F83600C9C5E52A0C40062513  
 18085 1F2310FD060A052808281915  
 18095 7EFE0620F723233608212308  
 18105 CD4034E1C1C9E52ACF402972  
 18115 3AD14077FE0020053E042307  
 18125 2B18023E0032D14022CF2205  
 18135 40E1C9C53E261803C53E2593  
 18145 A60E012100000605F5782120  
 18155 FE02CC9F46F1348E2B202787  
 18165 0510F14E042BF53E06902394  
 18175 47F1D60A23237710FCC12746  
 18185 C9CD3D470605181E06142199  
 18195 CD2F47CD6243182206322387

18635 7620033E03C93E01C9112720  
 18645 2000060837ED5210FB7E2846  
 18655 FE8020033E04C93E03C92990  
 18665 E5F52AD2400E052606052913  
 18675 237C9220047D93280D102742  
 18685 F5061C2310FD00D20EBF13174  
 18695 1803F13E05E177C90E042978  
 18705 2AD440CD5349283606212902  
 18715 2B10FD0D20F32AD2400E3030  
 18725 04CD53492825062123102642  
 18735 FD0D20F32AD2400604233022  
 18745 CD5349281310F82AD4403132  
 18755 06042BCD5349280310F82861  
 18765 C93C3C3C3C097C92C07D3379  
 18775 93C90E042AD4400621282926  
 18785 10FDCD534928320D20F33178  
 18795 2AD2400E0406212310FD02857  
 18805 CD5349281F0D20F32AD23162  
 18815 400604CD5349281123102743  
 18825 F82AD4400604CD5349283187  
 18835 072B10F8C93C3C3C3C93176  
 18845 CD8B47FE77C80000FE353526  
 18855 CC5547CD274721C240E53435  
 18865 5E235623234E23AFBE283053  
 18875 051AB9CA3345E5EBCDD53680  
 18885 47EBE1FE0028101AFE003391  
 18895 201CAFBE2009CD7047FE3388  
 18905 002024182FAE240CD533049  
 18915 49201AE1CD2C4AC3C8453443  
 18925 2B4EB9202123AFBE200F3128  
 18935 23340605CDDA4610FBE13403  
 18945 CD2C4A180BE1CD2C4AC33431  
 18955 3445E1CD274A1812FE383356  
 18965 2604FEB820F2233601E13421  
 18975 CD2C4ACDF74718155E233388  
 18985 5618054E2346AF02237E3006  
 18995 C68077122B722B73C93A3417  
 19005 C5404F060C218240C55E3266  
 19015 235623E5237E7FE0020A43306  
 19025 237EFE0020481A89280663186  
 19035 2B360128183A3E00232E62987  
 19045 40E17EE5D5EBCDDA477E4142  
 19055 EBFEE00281121E64035AF3541  
 19065 BE281ED1E17EBCDC44773861  
 19075 18E3CD7947FE0028E8E13859  
 19085 E1C1C0CE4A23232323103629  
 19095 AD1842D1E1C118F4FE053897  
 19105 2807CDE948FE052005CD3549  
 19115 0F49EB1806D5EBCD95483727  
 19125 E1D112D5CDDA47EBE1C14322  
 19135 1AFE0628CDBFE8628C7FE3930  
 19145 0028C3180C5052846284E3417  
 19155 AF02C17323727912C9213547  
 19165 E54046352002360D217F3227  
 19175 401105001910FD463ACE3274  
 19185 40C0C94777CDB54721C73919  
 19195 407E238623BE280B77FE3588  
 19205 0C20063C3C23771801232974  
 19215 7EFE00284CD60F38272A3462  
 19225 BE403AC040CDDA477E7E4052  
 19235 00280B3AC040CDDA477E7E4052  
 19245 C040183EEBCD7947E9BFA4003  
 19255 00283522BE4018303AC53349  
 19265 40FE0020D22ABE407E7E3686  
 19275 0028073AC1404688201A3334  
 19285 3AC140C6807732C140183761  
 19295 0FCD3D47111FB43CD57433726  
 19305 CD6045C3EE422ABE40ED4092  
 19315 5BE2407C9238047D93303731  
 19325 052159491803210F49223092  
 19335 AB4A21CC4035200C360A3427  
 19345 2BCD5F4636CA1C4528B353687  
 19355 C39D49000000000000003165



18205 18100614180C060A18081740  
 18215 C50602CD2F47C1C9E5212784  
 18225 C80A2B7CFE0020FA10F52784  
 18235 E1C92A0C4006222310FD2508  
 18245 0E1606207EC6807723102326  
 18255 F9230D20F3C9D5E5CD233095  
 18265 47117644CD8643CD66432704  
 18275 CD8B47FE3520F0CD23472965  
 18285 E1D1C921E440BE2003342907  
 18295 3CC921E34006057A96282691  
 18305 20047B9628032B10F4C92546  
 18315 CDBB02444D51143E00232442  
 18325 04CDBD077EFE3820033E2648  
 18335 01C9FE3D20033E02C9FE2791  
 18345 2A20033E03C9FE29C03E2622  
 18355 04C93ACE40473CFE04202694  
 18365 023E0132CE40C94FCDB52801  
 18375 47798047DE05380478D62772  
 18385 044778C9E5CD8B47E1113052  
 18395 2000FE01200228C9FE022601  
 18405 20032319C9FE032004372434  
 18415 ED52C9FE04C023C93E013069  
 18425 328D483AC5404F2AD6402791  
 18435 23060D2B7EFE80280DB902653  
 18445 CC6F4810F406142B10FD02815  
 18455 18EB218D48342AD240062719  
 18465 11237EFE80280DB90C6F2963  
 18475 4810F406322B10FD18EB2819  
 18485 218D48342AD84028060E2553  
 18495 237EFE80280DB90C6F483048  
 18505 10F406132310FD18EB212771  
 18515 8D48342AD44006102B7E2674  
 18525 FE80280DB90C6F4810F43177  
 18535 06312310FD18EBC9C5E53165  
 18545 EB2182407E239320047E2862  
 18555 9228062323232318F1232572  
 18565 237EFE002006233600E12717  
 18575 01C9232318E0FE04283B3029  
 18585 FE032827FE022810060E2638  
 18595 2B10FD7EFE7620033E022889  
 18605 C93E04C91121000600C192551  
 18615 10FD7EFE8020033E01C93076  
 18625 3E02C9060F2310FD7EFE2980

18995 068077122B722B73C93A3417  
 19005 C5404F060C218240C55E3266  
 19015 235623E5237E7FE0020A43306  
 19025 237EFE0020481A89280663186  
 19035 2B360128183A3E00232E62987  
 19045 40E17EE5D5EBCDDA477E4142  
 19055 EBFEE00281121E64035AF3541  
 19065 BE281ED1E17EBCDC44773861  
 19075 18E3CD7947FE0028E8E13859  
 19085 E1C1C0CE4A23232323103629  
 19095 AD1842D1E1C118F4FE053897  
 19105 2807CDE948FE052005CD3549  
 19115 0F49EB1806D5EBCD95483727  
 19125 E1D112D5CDDA47EBE1C14322  
 19135 1AFE0628CDBFE8628C7FE3930  
 19145 0028C3180C5052846284E3417  
 19155 AF02C17323727912C9213547  
 19165 E54046352002360D217F3227  
 19175 401105001910FD463ACE3274  
 19185 40C0C94777CDB54721C73919  
 19195 407E238623BE280B77FE3588  
 19205 0C20063C3C23771801232974  
 19215 7EFE00284CD60F38272A3462  
 19225 BE403AC040CDDA477E7E4052  
 19235 00280B3AC040CDDA477E7E4052  
 19245 C040183EEBCD7947E9BFA4003  
 19255 00283522BE4018303AC53349  
 19265 40FE0020D22ABE407E7E3686  
 19275 0028073AC1404688201A3334  
 19285 3AC140C6807732C140183761  
 19295 0FCD3D47111FB43CD57433726  
 19305 CD6045C3EE422ABE40ED4092  
 19315 5BE2407C9238047D93303731  
 19325 052159491803210F49223092  
 19335 AB4A21CC4035200C360A3427  
 19345 2BCD5F4636CA1C4528B353687  
 19355 C39D490000000000000003165



# Geisterjagd

### Ablauf der Eingabe:

1. Programm zur Erstellung des REM-Befehls 2851 Byte
2. Run
3. Poke 16510.0
4. Eingabe des Lade- und Laufprogramms
5. Run
6. Eingabe des HEX-LISTING
7. Aufnahme mit Taste A
8. Das Programm läuft sofort!

Punkte gibt es pro Geist 50 beim Fangen, und 50 beim Nachhausebringen. Wenn alle 12 geschnappt sind, wird der Rest des Zeitbalkens ebenfalls in Punkte umgesetzt. Sollten Sie beim Probe-Spielen nicht über die 1. Runde hinauskommen, hier ein Tip: Von der Seite überfährt man die Kerle am besten, es sei denn, sie sehen sich gerade um! Ansonsten gilt: üben, üben, üben... Viel Spaß dabei!

Übrigens: Bei je 10 000 Punkten gibt es ein »Freispiel«. Bis zu fünf Spielfiguren werden angezeigt, jede Figur mehr nur registriert.

(Peter Richter)

```

1 FAST
2 SCROLL
3 PRINT "STARTADRESSE ?"
4 INPUT A
5 SCROLL
6 FOR N=A TO 19380 STEP 10
7 LET SUM=N-16515
8 SCROLL
9 PRINT N;" ";
10 INPUT A$
11 IF LEN A$ < 23 OR LEN A$ > 24 THEN GOTO 10
12 PRINT A$
13 LET B=0
14 FOR C=1 TO 20 STEP 2
15 LET D=(CODE A$(C)-28)*16+C
16 LET SUM=SUM+D
17 POKE (N+B),D
18 LET B=B+1
19 NEXT C
20 IF SUM <> VAL A$(21 TO ) THEN GOTO 23
21 NEXT N
22 GOTO 26
23 SCROLL
24 PRINT "          FALSCH EINGABE"
25 GOTO 7
26 CLS
27 SLOW
28 PRINT "SICHERE DAS PROGRAMM AUF BAND"
29 PRINT ">TASTE A<"
30 IF INKEY$="" THEN GOTO 29
31 IF INKEY$="A" THEN GOTO 32
32 GOTO 29
33 SAVE "ZAPPEL"
34 RAND USR 16515
35 CLS
36 PRINT "NOCH EIN SPIEL ? (J/N)"
37 IF INKEY$="J" THEN GOTO 33
38 GOTO 36
100 REM * NICHT IM SPIEL *
101 REM ** D-FILE ABHÄNGIGE **
    ** DATENBYTES = 0 **
    ** DRUCKPROGRAMM **
    
```

```

102 LET A$="1676116762167661676
71681116812179921799317995179961
80181801918021180221804018149181
501857319360"
103 FOR N=1 TO LEN A$ STEP 5
104 POKE VAL A$( TO 5),0
105 LET A$=A$(6 TO )
106 NEXT N
107 LPRINT " * * HEX - L I S
T I N G * *"
108 LPRINT "-----"
109 GOSUB 140
110 LPRINT "          >> ZAPPEL"
111 GOSUB 140
112 LPRINT "VON", "PETER RICHTER",
"AM MAUBISHOF 22", "4044 KARST 1",
"02101/63257",
113 LET Y=719
114 LET X=16515
115 FOR A=X TO X+Y STEP 10
116 LPRINT A;" ";
117 LET S=A-16515
118 FOR B=0 TO 9
119 LET N=2
120 LET H$="00"
121 LET P=PEEK (A+B)
122 IF A+B > 19358 THEN LET P=0
123 LET S=S+P
124 LET H=INT (P/16)
125 LET H$(N)=CHR$ (P-16*H+28)
126 LET P=H
127 LET N=1
128 IF P > 0 THEN GOTO 124
129 LPRINT H$;
130 NEXT B
131 LPRINT S
132 IF A+B > 19358 THEN STOP
133 NEXT A
134 GOSUB 138
135 LET Y=679
136 LET X=A
137 GOTO 115
138 LPRINT
139 LPRINT
140 LPRINT
141 LPRINT
142 RETURN
    
```

Lade- und Laufprogramm zu Zappel

# Statistische Auswertung von Meßreihen

Bei der täglichen beruflichen Arbeit mit statistischen Auswertungs- und Analyseprogrammen, die in APL geschrieben sind und über Netzwerkterminal auf einem Großrechnerverbund laufen, stellte sich die Frage, ob sich zumindest ein Teil dieser Anwendungen auch auf einem Computerwinzling wie dem ZX81 mit 16 KByte RAM in Basic implementieren und nachvollziehen lassen könnte.

Ich habe den Versuch unternommen, ausgehend von der beruflich genutzten Großrechner-Software, zunächst eine der in der technischen Statistik am häufigsten benutzten Anwendungen, das Auswerten von Meßwerten, auf den ZX81 umzusetzen. So entstand das Programm »TECSTAT I« (Technische Statistik). Die römische Eins soll darauf hinweisen, daß weitere ZX-Programme für statistische Anwendungen geplant sind und später realisiert werden sollen. Das Programm ermöglicht die statistische Auswertung von Meßreihen, mit im Vergleich zum Großrechner praktisch identischen Ergebnissen.

Natürlich mußten einige Einschränkungen in Kauf genommen werden. Während der Großrechner, bedingt durch die Vorteile der Sprache APL (sehr schnelle Vektoren- und Matrixoperationen) und durch die hohe CPU-Geschwindigkeit, praktisch sofort nach Abschluß der Meßwerte-Eingabe die Ergebnisse liefert, kann dies beim ZX-Programm, je nach Anzahl der eingegebenen Werte, unter Umständen ziemlich lange dauern. Außerdem mußte die Anzahl der möglichen Meßwerte auf maximal 1000 begrenzt werden, da hiermit die verfügbare Speicherkapazität von 16 KByte praktisch ausgelastet ist, während der große Bruder hier kaum Grenzen setzt. Allerdings ist der Grenzwert von 1000 für die Praxis durchaus akzeptabel.

Auf eine automatische Klasseneinteilung von ungruppiert eingegebenen Meßwerten, wie sie der große Bruder vornimmt, wurde verzichtet. Neben dem hierfür fehlenden Speicherplatz (die Anzahl möglicher Meßwerte hätte sonst erheblich verringert werden müssen) spielte für diese Entscheidung auch der Aspekt eine Rolle, daß bei dem Kapitel »Klasseneinteilung von Meßwerten« neben mathematischen Grundlagen auch eine Portion Erfahrung und Gefühl ins Spiel kommt. Deshalb soll dieses Thema einem später zu erstellenden und sich nur mit dieser Thematik befassenden Programm überlassen werden.

Auch auf Tests gegen Toleranzgrenzen wurde verzichtet. Ansonsten liefert das Programm jedoch für die eingegebenen Meßreihen eine Vielzahl von Informationen für den ernsthaften Statistiker.

Die unter Umständen sehr lange Laufzeit bei großer Anzahl von Meßwerten ist für den Hobbybereich praktisch bedeutungslos; hier steht ja normalerweise nicht der Zeit- und Entscheidungsdruck, wie er im Geschäftsleben oft an der Tagesordnung ist, hinter der Auswertung. Man findet in der Fachliteratur sicherlich auch Näherungsformeln, die in einem Basic-Programm eingebaut, schneller zu Ergebnissen führen; allerdings geht hierbei oft Information und Genauigkeit verloren. Im

übrigen wurde im ZX-Programm natürlich einiges, was bei der als Vorbild genommenen Anwendung subjektiv als nachteilig empfunden wurde, optimiert und umgestaltet. So wird zum Beispiel der nicht zu den Auswertungsergebnissen zählende Dialogverkehr mit dem Rechner bei Ausgabe der Auswertung auf dem Drucker unterdrückt.

Nicht enthalten im APL-Programm sind die Berechnung der Parameter Zentralwert (Median) und Variationskoeffizient. Die Berechnung dieser Werte wurde im ZX-Programm zusätzlich mitaufgenommen.

## Anwendungsbeschreibung

Das Programm ist selbststartend und meldet sich nach Beenden des Ladevorganges automatisch mit seinem Titel auf dem Display. Es wird sofort gefragt, ob eine Protokollierung auf einem angeschlossenen Printer gewünscht wird. Die Beantwortung dieser Frage mit »Nein« läßt die gesamte Auswertung nur auf dem Bildschirm erscheinen. Wird dagegen mit »Ja« geantwortet, erfolgt im weiteren Ablauf des Programmes zusätzlich ein Ausdruck der Auswertungsergebnisse.

Die Druckroutinen sind für den ZX-Drucker geschrieben worden, weil dieser die Möglichkeit bietet, durch die doppelte Auflösung mit Hilfe des »PLOT«-Befehles Balkendiagramme sehr gut dazustellen. Durch Umschreiben einiger Statements ist leicht eine Anpassung an andere Drucker zu erreichen.

Nur wenn ein Druckerprotokoll gewünscht wurde, wird vom Rechner eine Bezeichnung der Meßreihe erfragt. Dies soll die spätere Zuordnung bei mehreren Auswertungen erleichtern. Diese Eingabe kann maximal 32 Stellen lang sein; ist sie zu lang, wird sie durch die Software auf 32 Stellen gekürzt. Wird keine Bezeichnung gewünscht, kann auch nur »N/L« eingegeben werden. Die Ausgabe der Bezeichnung erfolgt auf dem Drucker und auf dem Bildschirm.

Nachdem die Angelegenheit mit dem Druckerprotokoll geregelt ist, will der Rechner erfahren, ob die Werte in gruppierter Form (Strichliste) vorliegen. Wenn ja, fragt der Rechner, um wieviele Gruppen es sich handelt. Anschließend werden für die Gruppen jeweils paarig die Meßwerte und die Häufigkeiten derselben in ihren Gruppen eingegeben. Bei großen Häufigkeiten werden die Abstände zwischen den einzelnen Eingaben etwas länger, weil der Rechner entsprechend oft den jeweiligen Wert erzeugen muß. Wurde die Frage nach der Gruppierung mit »Nein« beantwortet, muß dem Rechner bekanntgegeben werden, wieviele Meßwerte vorliegen. Danach sind

## Vergleich zwischen ZX-Programm und Großrechnerprogramm (APL) hinsichtlich der ausgegebenen Resultate

Meßreihe »Pgm-Laufzeitmessungen«, 16 ungruppierte Werte

Mittelwert	ZX: 43.0625 APL: 43.0625000000
Median	ZX: 43 APL: keine Angabe
Varianz	ZX: 8.0625 APL: 8.0625000000
Std.abweichung	ZX: 2.8394542 APL: 2.8394541729
Var.koeffizient	ZX: 6.5937978 APL: keine Angabe
Schiefe	ZX: -0.1120893 APL: -0.1120892981
Wölbung	ZX: -0.57192041 APL: -0.5719204058
Spannweite	ZX: 10 APL: 10.0000000000
Kleinster Wert	ZX: 38 APL: 38.0000000000
Größter Wert	ZX: 48 APL: 48.0000000000
VB f. Mittelw.(u)	ZX: 41.557625 APL: 41.5576254050
VB f. Mittelw.(o)	ZX: 44.567375 APL: 44.5673745950
VB f. Std.abw.(u)	ZX: 2.0974734 APL: 2.0974733514
VB f. Std.abw.(o)	ZX: 4.4013312 APL: 4.4013312375
VB f. Schiefe(u)	ZX: -1.2181324 APL: -1.2181323666
VB f. Schiefe(o)	ZX: 0.99395377 APL: 0.9939537704
VB f. Wölbung(u)	ZX: -2.7098372 APL: -2.7098372094
VB f. Wölbung(o)	ZX: 1.5659964 APL: 1.5659963978
Test auf NV	ZX Probengröße zu klein APL: Probengröße zu klein

Meßreihe »Wellendurchmesser«, 500 Werte in 15 Gruppen

Mittelwert	ZX: 13.4264 APL: 13.4264000000
Median	ZX: 13.42 APL: keine Angabe
Varianz	ZX: 0.013175391 APL: 0.0131753908
Std.abweichung	ZX: 0.11478411 APL: 0.1147841051
Var.koeffizient	ZX: 0.8549135 APL: keine Angabe
Schiefe	ZX: -0.043721135 APL: -0.0437206839
Wölbung	ZX: -0.082938245 APL: -0.0829382101
Spannweite	ZX: 0.7 APL: 0.7000000000
Kleinster Wert	ZX: 13.07 APL: 13.0700000000
Größter Wert	ZX: 13.77 APL: 13.7700000000
VB f. Mittelw. (u)	ZX: 13.416314 APL: 13.4163143152
VB f. Mittelw. (o)	ZX: 13.436486 APL: 13.4364856848
VB f. Std.abw. (u)	ZX: 0.10808313 APL: 0.1080831339
VB f. Std.abw. (o)	ZX: 0.12237772 APL: 0.1223777209
VB f. Schiefe (u)	ZX: -0.257788 APL: -0.2577875468
VB f. Schiefe (o)	ZX: 0.17034573 APL: 0.1703461791
VB f. Wölbung (u)	ZX: -0.51023018 APL: -0.5102301421
VB f. Wölbung (o)	ZX: 0.34435369 APL: 0.3443537219
Test auf NV	ZX: ...kann nicht verworfen werden APL: ...kann nicht verworfen werden
Häufigkeitstabelle	Werte ZX und APL identisch

die einzelnen Meßwerte einzugeben. In beiden Fällen erscheint die jeweils letzte Eingabe unten auf dem Schirm und wird mit jeder weiteren Eingabe um eine Zeile nach oben geschoben (Scrollen). Dadurch ist ein Überblick über die Eingabefolge gegeben. Bei eingeschaltetem Printerprotokoll erfolgt eine zeilenweise Auflistung der Eingaben auf dem Drucker.

Die Dateneingabe muß sehr sorgfältig erfolgen. Aus Rücksicht auf die Verarbeitungszeit wurde auf eine vollständige Syntaxkontrolle durch das Programm verzichtet, es ist nur die dem Rechner eigene Kontrolle wirksam. Sollte man sich verippt haben und bereits die »N/L«-Taste betätigt worden sein, bleibt nichts anderes übrig, als das Programm mit »BREAK« oder »STOP« abzuberechnen, mit »RUN« neu zu starten und mit der Eingabe neu zu beginnen.

Sofern die Meßwerte bereits in aufsteigend sortierter Form vorliegen, empfiehlt es sich, diese auch in dieser Reihenfolge einzugeben. Das Programm merkt, daß die Werte bereits sortiert sind und verzichtet auf einen eigenen Sortierlauf. Gerade bei großen Datenmengen kann hierdurch die Programmlaufzeit beträchtlich vermindert werden. Bei ungeordnet eingegebenen Werten führt das Programm eine Sortierung der Daten nach dem »Shell-Metzner-Verfahren« durch; damit kann sich die Laufzeit gegenüber der sortierten Eingabe, je nach Anzahl und Anordnung der Daten, gegenüber der sortierten Eingabe etwa vervierfachen.

Sollte die Anzahl der Eingaben die maximal zulässige Anzahl von 1000 überschreiten, gibt der Rechner eine Fehlermeldung aus; das Programm wird ohne Auswertung beendet.

Nach Abschluß der Dateneingabe, die vom Rechner gemeldet wird, ist die »N/L«-Taste zu betätigen. Der Bildschirm wird dunkel und der Rechner widmet sich nun mit ganzer Kraft der Auswertung. Damit hat er unter Umständen eine ganze Zeit zu tun (1000 sortiert eingegebene Werte benötigen etwa 7 Minuten, die unsortierte Eingabe der gleichen Werte führt zu einer Rechenzeit von etwa 26 Minuten).

Dann erscheinen die ersten Ergebnisse auf dem Bildschirm und gegebenenfalls auch auf dem Drucker. Diese Ergebnisse stellen die wichtigsten statistischen Kenngrößen dar. Es handelt sich hierbei um folgende Angaben:

- Anzahl Meßwerte
- Mittelwert
- Median (Zentralwert)
- Varianz
- Standardabweichung
- Variationskoeffizient
- Schiefe
- Wölbung
- Spannweite (Range)
- Kleinstwert
- Größtwert

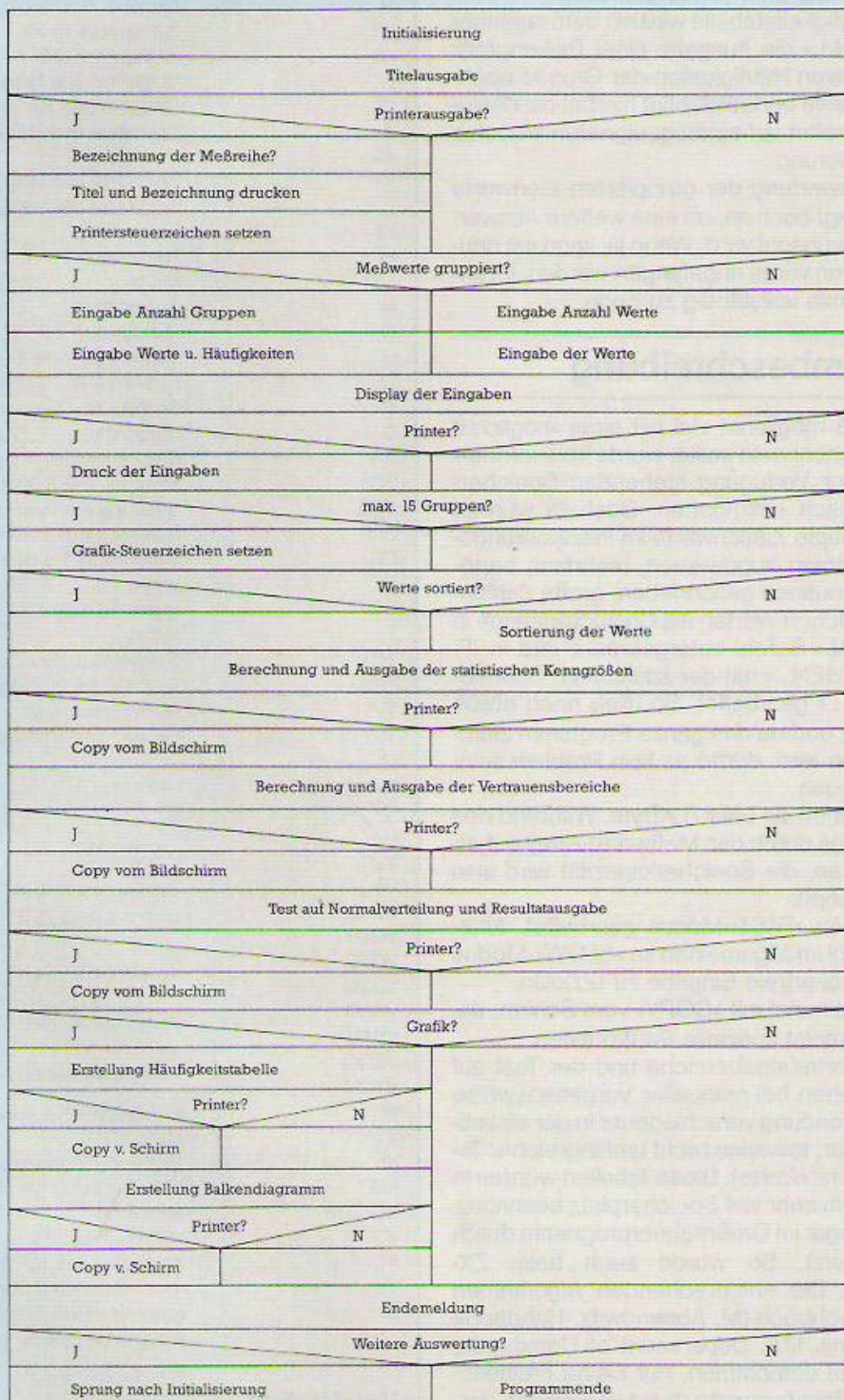


Die Berechnung der einzelnen Parameter erfolgte nach den allgemein gültigen und in der gängigen Fachliteratur enthaltenen Formeln. Es wird die maximal mögliche Stellenzahl ausgegeben, auf eine Einschränkung der Dezimalstellen durch Rundung wurde verzichtet.

Nach erneutem Betätigen der »N/L«-Taste ist der Rechner wiederum kurzfristig beschäftigt und meldet sich dann wieder mit einem neuen Block von Ergebnissen. Jetzt gibt er die Vertrauensbereiche für den Mittelwert, die Standardabweichung, die Schiefe und die Wölbung unter der Annahme, daß eine Normalverteilung vorliegt, mit 95prozentiger (zweiseitiger) Aussagewahrscheinlichkeit aus. Diese Angaben sind für den Statistiker besonders wertvoll, wenn es sich bei der Meßreihe

um Ergebnisse aus einer Stichprobe handelt und Rückschlüsse auf die Grundgesamtheit, aus der die Stichprobe entnommen wurde, gezogen werden sollen. Die Vertrauensbereiche zeigen an, in welchem Bereich sich die tatsächlichen Werte der Grundgesamtheit bewegen können.

Nach einem weiteren »N/L« wird das Programm veranlaßt, die vorliegende Meßreihe auf das Vorliegen einer Normalverteilung zu testen. Dies funktioniert jedoch nur, wenn mindestens 30 Werte vorliegen. Ist dies nicht der Fall, teilt der Rechner mit, daß die Probengröße für einen Test auf Normalverteilung nicht ausreicht. War die Meßreihe umfangreicher, wird das Testergebnis in der unter Statistikern üblichen Formulierung ausgegeben.



**Programmablaufplan**

Der weitere Programmablauf, der wiederum mit »N/L« angestoßen wird, hängt nun von der Art der Eingabedaten ab. Da, wie schon eingangs erwähnt, auf eine automatische Klasseneinteilung von ungruppierten Werten verzichtet wurde, ist für diese Daten die Auswertung an diesem Punkt zu Ende. Handelte es sich jedoch bei den eingegebenen Daten um gruppierte Werte (maximal 15 Gruppen, eine in der Praxis durchaus übliche Zahl), gibt der Rechner nach kurzer Denkpause eine Häufigkeitstabelle aus, die neben der absoluten Anzahl der Werte in den Gruppen auch die prozentuale Häufigkeit pro Gruppe und die aufakkumulierte Summenhäufigkeit der Gruppen enthält. Die Prozentangaben sind auf zwei Dezimalstellen gerundet. Sollte das manuelle Nachrechnen der Einzelhäufigkeiten zur Summenhäufigkeit aufaddiert keine genauen 100 Prozent ergeben, sondern vielleicht nur 99,99, liegt das an den im Rechner auftretenden Rundungsfehlern, die übrigens auch beim Großrechnerprogramm vorhanden sind.

Im Anschluß an die Häufigkeitstabelle wird mit dem nunmehr schon obligatorischen »N/L« die Ausgabe einer Balkengrafik angestoßen, die die relativen Häufigkeiten der Gruppe nochmals darstellt. Das Programm berücksichtigt hierbei die darzustellenden Werte und schaltet auf einen geeigneten Maßstab mit entsprechender Skalierung.

Damit ist auch die Auswertung der gruppierten Meßwerte beendet. Der Rechner fragt noch an, ob eine weitere Auswertung mit neuen Daten gewünscht wird. Wenn ja, kann mit neuen Eingabewerten ganz von vorne angefangen werden. Im anderen Fall ist das Programm vollständig zu Ende.

## Programmbeschreibung

Im Hinblick darauf, daß möglichst viel mit einer möglichst großen Datenmenge machbar sein sollte, wurde von vornherein versucht, mit dem zur Verfügung stehenden Speicherplatz möglichst ökonomisch umzugehen. Deshalb wurden zum Beispiel häufig benötigte Zahlenwerte im Initialisierungsteil des Programms Variablen zugewiesen, mehrfach benötigte Funktionen als Subroutinen geschrieben, große Zahlenwerte und Formeln mit solchen Werten als Characterstrings in Verbindung mit dem »VAL«-Befehl untergebracht und in IF-Bedingungen »IF A=0 THEN...« mit der sparsamen Schreibweise »IF NOT A THEN...« gearbeitet. So blieb noch etwas Freiraum für REM-Zeilen, und da das ganze Programm ziemlich geradlinig durchlaufen wird, dürfte es kein Problem sein, den Ablauf zu durchschauen.

Das Programm benötigt etwas über 8 KByte. Während des Laufes wächst es allerdings durch das Meßwerte-Array auf etwas mehr als 15 KByte an, die Speicherkapazität wird also praktisch ganz ausgeschöpft.

Es wird überwiegend im »FAST«-Modus gearbeitet. Allerdings wird ein Input-Befehl im allgemeinen im »SLOW«-Modus abgearbeitet, um eine flackerfreie Eingabe zu erzielen.

Das Drucken geschieht meist mit »COPY« vom Schirm, dadurch erübrigten sich zumeist spezielle Printroutinen.

Die Errechnung der Vertrauensbereiche und der Test auf Normalverteilung geschehen bei manueller Vorgehensweise üblicherweise unter Verwendung verschiedener in der statistischen Literatur enthaltener, teilweise recht umfangreicher Tabellen (t-Werte, u-Werte,  $\chi^2$ -Werte). Diese Tabellen würden in einem Computerprogramm sehr viel Speicherplatz beanspruchen und sind deshalb sogar im Großrechnerprogramm durch Rechenalgorithmen ersetzt. So wurde auch beim ZX-Programm vorgegangen. Die entsprechenden Algorithmen sind einem sehr guten Fachbuch (M. Abramowitz, Handbook of Mathematical Functions, U.S. Department of Commerce, Nat. Bureau of Standards) entnommen. Für kleine Freiheitsgrade (bis 3) werden die Tabellenwerte direkt verwendet, darüber kommen die teilweise recht umfangreichen Formeln ins

A0 ... A9	Ziffernwerte 0 ... 9
A\$	Hilfsstring
B0 ... B9	Zahlenwerte 10 ... 19
B\$	Leerzeilenstring
C0 ... C9	Zahlenwerte 20 ... 29
CHI	
CHIO	Chi <sup>2</sup> -Werte
CHIU	
D0 ... D2	Zahlenwerte 30 ... 32
E	Absolute Anz. Werte f. Häufigkeitstabelle
F	Gruppenwert für Häufigkeitstabelle
FG	Freiheitsgrad
F2	Hilfsvariable
G	Anzahl Gruppen
G1	Schiefe
G2	Wölbung
G\$	Hilfsstring
H	Eingabe- und Hilfsvariable
HT	Zahlenwert 1000
H1	Zahlenwert 100
I	Schleifen- und Hilfsvariable
IS	Eingabe- und Hilfsstring
J	Schleifen- und Hilfsvariable
J\$	Hilfsstring
K	Schleifen- und Hilfsvariable
L	Schleifen- und Hilfsvariable
L\$	Linienstring
M	Hilfsvariable
MD	Median
N	Anzahl Werte
O	Hilfsvariable
P	Printersteuerzeichen
Q	Stablänge im Balkendiagramm
R	Spannweite (Range)
S	Steuervariable f. Skalierung im Diagramm
SA	Standardabweichung
SG1	Streuung der Schiefe
SG2	Streuung der Wölbung
SQ	Summe der Quadrate der Werte
SX	Summe der Meßwerte
S2	Varianz
S\$	Sternchenzeile
T	t-Wert
TG1	Testwert für Schiefe
TG2	Testwert für Wölbung
TV	Bildschirmpointer
T1 ... T4	Errechnete Tabulatoren f. Ausgabe
U	
U2	Hilfswerte
U3	
U4	
V	Hilfsvariable
VA	Variationskoeffizient
VG10	
VG1U	Vertrauensbereich f. Schiefe
VG20	
VG2U	Vertrauensbereich f. Wölbung
VSAO	
VSAU	Vertrauensbereich f. Standardabweichung
VXQO	
VXQU	Vertrauensbereich f. Mittelwert
W	Prozentuale Häufigkeit
X	Meßwerte
XI	Meßwert-Inputvariable
XM	Hilfsvariable
XQ	Mittelwert
Y	Hilfsvariable
Y1	Steuerzeichen für Variationskoeff.errechnung
Y2	Steuerzeichen f. Sortier-Erfordernis
Z	Max. rel. Häufigkeit für Festlegung der Skalierung im Diagramm

### Variablenliste

Spiel. Es handelt sich hierbei um die Programmpassagen mit den etwas ominösen, durch viele Dezimalstellen erkennbaren Zahlen. Diese Zahlen wurden in voller vorliegender Länge übernommen, um die Rechen-Genauigkeit des ZX81 komplett auszuschöpfen. Der Test auf Normalverteilung wird nach der »Kumulantenmethode« (beschrieben in »Auswerten von Meßreihen«, DGQ 7, Deutsche Gesellschaft für Qualität e.V.) durchgeführt. Für alle anderen Berechnungen wurden die in der Statistik allgemein üblichen Formeln verwendet. Teilweise mußte darauf geachtet werden, daß unter keinen Umständen eine nicht erlaubte Division durch Null auftreten darf. So mußte vor Berechnung des Variationskoeffizienten geprüft werden, ob der Mittelwert ungleich Null ist. Hierfür wurde der Rechnung eine in Form eines Produktes aus den Operanden durchgeführte Prüfung vorgeschaltet, die vor Auftreten eines Überlaufers bereits erkennt, ob der Variationskoeffizient eventuell an die Grenze der Rechnerkapazität ( $10^{38}$ ) stößt. In diesem Fall wird der Koeffizient nicht mehr berechnet, als Ergebnis wird nur ein Hinweis ausgegeben.

Die sehr zeitraubende Sortieroutine wird benötigt, um den Zentralwert zu ermitteln. Sie wird aber nicht nur hierfür benötigt, sondern auch zur Aufbereitung der Daten für die Häufigkeitstabelle. Nebenbei fallen bei der Sortierung auch die Größt- und Kleinstwerte an.

Nach erfolgter Eingabe des Programmes und Beenden des Debuggings wird mit »GOTO 4270« auf Kassette gespeichert. Damit wird der automatische Start aktiviert.

## Betriebserfahrungen

Gewiß gibt es heute schon sehr preiswerte Taschenrechner, die das Ermitteln von Mittelwert, Standardabweichung, Varianz, Kleinst- und Größtwert auf Knopfdruck im Handumdrehen ermöglichen. Aber die anderen Werte wie Schiefe, Wölbung, Vertrauensbereiche sind nicht so einfach zu ermitteln. Insofern ist die leider etwas lange Programmlaufzeit ohne weiteres zu rechtfertigen, denn hier werden auch diese Werte geliefert.

Andererseits kam es darauf an, zu sehen, inwieweit es der kleine ZX81 mit Großrechnern aufnehmen kann. Hinsichtlich der Laufzeit schneidet er hier sehr schlecht ab. Aber, und das ist eigentlich etwas überraschend, die Rechengenauigkeit ist trotz der geringeren rechnerinternen Stellenzahl sehr gut. Eine größere Anzahl von Auswertungen, die auf dem Großrechner bei Schulungen, Vorfürungen und Tests erstellt wurden, sind auf dem ZX81 mit dem hier vorgestellten Programm wiederholt worden. Ein Vergleich der Ergebnisse zeigte praktisch identische Ergebnisse. Lediglich die letzten zwei bis drei Dezimalstellen zeigen kleinere Abweichungen, besonders bei der Errechnung von Schiefe, Wölbung und Vertrauensbereichen. Dies ist durch die teilweise sehr vielen Schleifendurchläufe mit sich langsam aufaddierenden Rundungsfehlern im Rahmen der ZX-Rechengenauigkeit begründet, hat aber auf die praktische Verwendbarkeit der Ergebnisse keinen Einfluß.

Abschließend soll nicht unerwähnt bleiben, daß das hier vorgestellte, zunächst mehr aus Neugier entstandene Programm mittlerweile ein festes Anwendungsgebiet gefunden hat, wenn es zum Beispiel darum geht, ob der Benzinverbrauch des Autos sich signifikant, also statistisch nicht mehr zufällig, verändert hat. Das gleiche gilt natürlich auch für Telefongebühren und ähnliche in größerer Anzahl vorliegende Daten. Das weitaus interessanteste Anwendungsgebiet sind natürlich Meßreihen, die aufgrund von Programm-Laufzeitmessungen oder bei der Optimierung von elektronischen Schaltungen entstehen. Hierbei sind eigentlich keine Grenzen in der Anwendbarkeit gesetzt. Deshalb soll das vorliegende Programm auch in der nächsten Zukunft durch weitere statistische Programme ergänzt werden. (Bernd W. Friedrich)

## Basic-Listing »Tecstat«

```

10 REM *****
20 REM *      TECSTAT I      *
30 REM *      STATISTISCHE  *
40 REM *      AUSWERTUNG   *
50 REM *      VON MESSREIHEN *
60 REM * (C)  B.W.FRIEDRICH *
70 REM * 6500 MAINZ 1  2/84 *
80 REM *****
90 FAST
100 REM INITIATIONIERUNG
110 LET A0=NOT PI
120 LET A1=SGN PI
130 LET A2=A1+A1
140 LET A3=A2+A1
150 LET A4=A2+A2
160 LET A5=A3+A2
170 LET A6=A3+A3
180 LET A7=A4+A3
190 LET A8=A4+A4
200 LET A9=A5+A4
210 LET B0=A5+A5
220 LET B1=B0+A1
230 LET B2=B0+A2
240 LET B3=B0+A3
250 LET B4=B0+A4
260 LET B5=B0+A5
270 LET B6=B0+A6
280 LET B7=B0+A7
290 LET B8=B0+A8
300 LET B9=B0+A9
310 LET C0=B0+B0
320 LET C1=C0+A1
330 LET C2=C0+A2
340 LET C3=C0+A3
350 LET C4=C0+A4
360 LET C5=C0+A5
370 LET C6=C0+A6
380 LET C7=C0+A7
390 LET C8=C0+A8
400 LET C9=C0+A9
410 LET D0=C0+B0
420 LET D1=D0+A1
430 LET D2=D0+A2
440 LET H1=VAL "100"
450 LET HT=VAL "1000"
460 LET G=A0
470 LET P=A0
480 LET Y1=A0
490 LET Y2=A0
500 DIM X(HT+A1)
510 LET X(HT+A1)=VAL "-1E38"
520 LET U=A0
530 LET XM=A0
540 LET A#="NUR MAX. 1000 WERTE
MOEGLICH"
550 DIM B$(D2)
560 DIM L$(D2)
570 DIM S$(D2)
580 FOR I=A1 TO D2
590 LET L$(I)="-"
600 LET S$(I)="#"
610 NEXT I
620 GOTO 1230
630 REM BEREICHUNG
640 REM SR ASK
650 LET TV=VAL "PEEK 16396+256*
PEEK 16397+760"
660 POKE TV,VAL "143"
670 SLOW
680 IF INKEY#(<)" THEN GOTO 680
690 IF INKEY#="" THEN GOTO 690
700 FAST
710 LET I#=INKEY#
720 RETURN
730 REM SR PRINT LINE
740 IF NOT P THEN RETURN
750 LPRINT
760 LPRINT L#
770 LPRINT
780 LPRINT
790 RETURN
800 REM SR N/L
810 PRINT AT C1,A0;"BITTE ""N/L

```



```

2060 IF P THEN LPRINT "GR. "; TAB
T1; I; " : X = "; XI; TAB C7; "H = "; TAB
T2; H
2070 FOR J=A1 TO H
2080 LET N=N+A1
2090 IF N<=HT THEN GOTO 2150
2100 SCROLL
2110 PRINT AT C0,A0;"1000 WERTE
UEBERSCHRITTEN", "AUSWERTUNG NICHT
MOEGLICH"
2120 SLOW
2130 PAUSE 200
2140 GOTO 4170
2150 LET X(N)=XI
2160 NEXT J
2170 NEXT I
2180 GOTO 1680
2190 REM STAT.KENNGROESSEN
2200 REM XQUER
2210 LET SX=A0
2220 FOR I=A1 TO N
2230 LET SX=SX+X(I)
2240 NEXT I
2250 LET XQ=SX/N
2260 REM VARIANZ
2270 LET SQ=A0
2280 FOR I=A1 TO N
2290 LET SQ=SQ+(X(I)-XQ)*(X(I)-X
Q)
2300 NEXT I
2310 LET S2=A1/(N-A1)*SQ
2320 REM STANDARDABWEICHUNG
2330 LET SA=SQR S2
2340 REM VARIATIONSKOEFFIZIENT
2350 IF VAL "ABS (XQ*SA)<1E-38"
THEN LET Y1=A1
2360 IF NOT Y1 THEN LET VA=(SA/X
Q)*H1
2370 REM WERTE SORTIEREN
2380 IF NOT Y2 THEN GOTO 2570
2390 LET M=N
2400 LET M=INT (M/A2)
2410 IF NOT M THEN GOTO 2570
2420 LET J=A1
2430 LET K=N-M
2440 LET I=J
2450 LET L=I+M
2460 IF X(I)<=X(L) THEN GOTO 253
0
2470 LET H=X(I)
2480 LET X(I)=X(L)
2490 LET X(L)=H
2500 LET I=I-M
2510 IF I<A1 THEN GOTO 2530
2520 GOTO 2450
2530 LET J=J+A1
2540 IF J>K THEN GOTO 2400
2550 GOTO 2440
2560 REM SPANNWEITE
2570 LET R=VAL "X(N)-X(1)+.5E-9"
2580 REM MEDIAN
2590 LET I=INT (N/A2)
2600 IF N/A2=I THEN GOTO 2530
2610 LET MD=X(I+A1)
2620 GOTO 2650
2630 LET MD=(X(I)+X(I+A1))/A2
2640 REM SCHIEFE U. WOELBUNG
2650 LET J=A2
2660 GOSUB 1040
2670 LET U2=U
2680 LET J=A3
2690 GOSUB 1040
2700 LET U3=U
2710 LET J=A4
2720 GOSUB 1040
2730 LET U4=U
2740 LET G1=N*U3/(((N-A1)*(N-A2)
)*SA*SA)
2750 LET G2=(((N*(N+A1)*U4)-A3*(
N-A1)*U2*U2)/((N-A1)*(N-A2)*(N-
A3)*S2*S2)
2760 REM AUSGABE STAT.KENNGROESS
EN
2770 PRINT TAB A3;"STATISTISCHE
KENNGROESSEN",TAB A3;L$( TO C6)
2780 PRINT "ANZAHL MESSWERTE.: "
;N
2790 PRINT ", "MITTELWERT.....:
";XQ
2800 PRINT "MEDIAN.....: "
;MD
2810 PRINT ", "VARIANZ.....:
";S2
2820 PRINT "STD.ABWEICHUNG....: "
;SA
2830 IF NOT Y1 THEN PRINT "VAR.K
OEFF.(0/0)...: ";VA
2840 IF Y1 THEN PRINT "VAR.KOEFF
.(0/0)...: ">=1E38"
2850 PRINT "SCHIEFE.....: "
;G1
2860 PRINT "WOELBUNG.....: "
;G2
2870 PRINT ", "SPANNWEITE.....:
";R
2880 PRINT "KLEINSTER WERT....: "
;X(A1)
2890 PRINT "GROESSTER WERT....: "
;X(N)
2900 IF P THEN COPY
2910 GOSUB 810
2920 REM BERECHN.VERTRAUENSBEREI
CHE
2930 REM T,CHI.F.PA=95 0/0 2SEIT
IG
2940 LET FG=N-A1
2950 IF FG<>A1 THEN GOTO 3000
2960 LET T=VAL "12.707"
2970 LET CHIO=VAL ".000982"
2980 LET CHIU=VAL "5.02"
2990 GOTO 3140
3000 IF FG<>A2 THEN GOTO 3050
3010 LET T=VAL "4.302"
3020 LET CHIO=VAL ".0506"
3030 LET CHIU=VAL "7.38"
3040 GOTO 3140
3050 IF FG<>A3 THEN GOTO 3100
3060 LET T=VAL "3.182"
3070 LET CHIO=VAL ".216"
3080 LET CHIU=VAL "9.35"
3090 GOTO 3140
3100 LET T=VAL "1.96+9.489538/(4
*N)+270.9799035/(98*N**2)+981.54
61612/(384*N**3)+150215.6332/(92
160*N**4)"
3110 LET F2=A2/(A0+FG)
3120 LET CHIO=VAL "FG*(1-F2-1.96
*SQR F2)*3"
3130 LET CHIU=VAL "FG*(1-F2+1.96
*SQR F2)*3"
3140 REM UB F. MITTELWERT
3150 LET I=T*SA/SQR N
3160 LET UXQ=XQ-I
3170 LET UXQ=XQ+I
3180 REM UB F. STD.ABWEICHUNG
3190 LET USAU=SA*SQR (FG/CHIU)
3200 LET USAO=SA*SQR (FG/CHIO)
3210 REM UB F. SCHIEFE
3220 LET SG1=SQR ((A5*N*(N-A1))/
((N-A2)*(N+A1)*(N+A3)))
3230 LET UG1U=VAL "G1-1.96*SG1"
3240 LET UG1O=VAL "G1+1.96*SG1"
3250 REM UB F. WOELBUNG
3260 LET SG2=SQR ((C4*N*(N-A1)**
A2)/((N-A3)*(N-A2)*(N+A3)*(N+A5)
))
3270 LET UG2U=VAL "G2-1.96*SG2"
3280 LET UG2O=VAL "G2+1.96*SG2"
3290 REM AUSGABE VERTR. BEREICHE
3300 PRINT " ZWEISEITIGE VERTRAU

```

Basic-Listing »Tecstat« (Fortsetzung)

## Basic-Listing »Tecstat« (Schluß)

```

ENSBEREICHE " ;L$ , " (MIT STAT.SIC
HERHEIT PA=95 0/0) "
3310 PRINT "FUER MITTELWERT:
" ,TAB A1;UX00;TAB B5; " . . . " ;UX00
3320 PRINT AT A9,A0; "FUER STD.AB
WEICHUNG: " ,TAB A1;USAU;TAB B5; " .
. . " ;USAU
3330 PRINT AT B2,A0; "FUER SCHIEF
E: " ,TAB A1;UG1U;TAB B5; " . . . " ;U
G10
3340 PRINT AT B5,A0; "FUER WOELBU
NG: " ,TAB A1;UG2U;TAB B5; " . . . " ;
UG20
3350 IF P THEN COPY
3360 GOSUB 810
3370 REM PRUEFUNG AUF NU
3380 LET A$=" NORMALVERTEILUNG "
3390 PRINT " PRUEFUNG AUF " ;A$;L
$
3400 IF N>=D0 THEN GOTO 3430
3410 PRINT " , , , " , "PROBENGROESSE ZU
KLEIN FUER TEST"; "AUF";A$; " . "
3420 GOTO 3490
3430 LET TG1=G1/S01
3440 LET TG2=G2/S02
3450 PRINT " , , , " , "MIT 95 0/0 AUSSA
GEWAHRSCHEIN- LICHEIT KANN DI
E ANNAHME EINER";A$;
3460 IF VAL "ABS TG1<1.96 AND AB
S TG2<1.96" THEN PRINT "NICHT " ;
3470 PRINT "VERWORFEN"
3480 PRINT "WERDEN. "
3490 IF P THEN COPY
3500 GOSUB 810
3510 IF NOT G OR G>B5 THEN GOTO
4170
3520 REM HAEUFIGKEITSTABELLE
3530 DIM E(G)
3540 DIM F(G)
3550 DIM W(G)
3560 LET Z=A0
3570 LET K=A0
3580 LET U=A0
3590 LET I=A1
3600 LET O=A1
3610 LET M=X(O)
3620 LET V=U+A1
3630 LET O=O+A1
3640 IF X(O)<>M THEN GOTO 3670
3650 LET V=U+A1
3660 GOTO 3630
3670 LET W(I)=VAL "INT ((U/N+.00
005)*1E4)/H1"
3680 IF W(I)>Z THEN LET Z=W(I)
3690 LET E(I)=V
3700 LET F(I)=X(O-A1)
3710 LET I=I+A1
3720 LET V=A0
3730 LET K=K+A1
3740 IF O<=N THEN GOTO 3610
3750 CLS
3760 PRINT TAB A6; "HAEUFIGKEITST
ABELLE";TAB A6;L$(A7 TO C5);TAB
C0; "REL. SUMMENG. WERT";TAB B4
; "ABS. (0/0) (0/0)";L$
3770 LET SH=A0
3780 FOR I=A1 TO K
3790 LET SH=SH+W(I)
3800 LET T1=A0
3810 IF I<B0 THEN LET T1=A1
3820 LET T2=B4
3830 IF E(I)<H1 THEN LET T2=B5
3840 IF E(I)<B0 THEN LET T2=B6
3850 LET T3=B8
3860 IF W(I)<H1 THEN LET T3=B9
3870 IF W(I)<B0 THEN LET T3=C0
3880 LET T4=C5
3890 IF I=G THEN LET SH=H1
3900 IF SH<H1 THEN LET T4=C7
3910 IF SH<B0 THEN LET T4=C8
3920 PRINT TAB T1;I;TAB A4;F(I);
TAB T2;E(I);TAB T3;W(I);TAB T4;S
H
3930 NEXT I
3940 PRINT L$
3950 IF P THEN COPY
3960 GOSUB 810
3970 REM GRAFIK REL.HAEUF.
3980 CLS
3990 LET S=VAL ".5"
4000 IF Z>C5 THEN LET S=A1
4010 IF Z>VAL "50" THEN LET S=VA
L "1.5"
4020 IF Z>VAL "75" THEN LET S=A2
4030 GOSUB 1130
4040 FOR I=A1 TO K
4050 LET T3=A3
4060 IF I>=B0 THEN LET T3=A2
4070 PRINT AT A5+I,T3;I;TAB A6;C
HR$ VAL "133"
4080 FOR J=A1 TO INT (W(I)/S)
4090 IF NOT W(I) THEN GOTO 4110
4100 PLOT B3+J,D2-I*2
4110 NEXT J
4120 NEXT I
4130 PRINT AT A5+I,A6;CHR$ VAL "
133"
4140 IF P THEN COPY
4150 GOSUB 810
4160 REM AUSWERT.ENDE
4170 CLS
4180 PRINT AT B0,B1; "E N D E"
4190 IF P THEN COPY
4200 PRINT AT C1,A0; "WEITERE AUS
WERTUNG ? (J/N)"
4210 INPUT I$
4220 IF I$<>"J" AND I$<>"N" THEN
GOTO 4210
4230 IF I$="J" THEN RUN
4240 PRINT AT C1,A0;B$
4250 STOP
4260 REM SAVE/AUTOSTART
4270 CLEAR
4280 SAVE "TECSTAT"
4290 RUN
* * *
8005 RESTORE 8030
8010 FOR i=1 TO 18
8015 READ P$
8020 FOR a=0 TO 7: READ b: POKE
USR P$+a,b: NEXT a: NEXT i
8030 DATA "a",60,66,129,129,129,
129,66,60
8032 DATA "b",60,126,195,195,195
,195,126,60
8034 DATA "c",66,137,153,169,137
,137,66,60
8036 DATA "d",66,153,165,137,145
,189,66,60
8038 DATA "e",66,157,165,141,165
,157,66,60
8040 DATA "f",66,137,153,169,189
,137,66,60
8042 DATA "g",255,129,129,129,12
9,129,129,255
8044 DATA "h",0,0,0,24,24,0,0,0
8046 DATA "i",0,0,0,0,0,0,0,1
8048 DATA "j",0,0,0,0,0,0,0,128
8050 DATA "k",128,0,0,0,0,0,0,0
8052 DATA "l",1,0,0,0,0,0,0,0
8054 DATA "m",0,0,0,24,24,0,0,1
8056 DATA "n",0,0,0,24,24,0,0,12
8
8058 DATA "o",128,0,0,24,24,0,0,
0
8060 DATA "p",1,0,0,24,24,0,0,0
8062 DATA "q",0,0,0,24,24,0,0,24
8064 DATA "r",24,0,0,24,24,0,0,0
8070 RETURN
9000 REM SAVE-ROUTINE BAND
9010 SAVE "MADN" LINE 0
9020 VERIFY "MADN"
9030 STOP

```

```
*****
*          *
*  TECSTAT I  *
*          *
*****
```

-----  
 STATISTISCHE AUSWERTUNG  
 VON MESSREIHEN  
 -----

WELLENDURCHMESSER

ZWEISEITIGE VERTRAUENSBEREICHE  
 -----  
 (MIT STAT.SICHERHEIT PA=95 0/0)

FUER MITTELWERT: 13.416314 ... 13.436486  
 FUER STD.ABWEICHUNG: 0.10808313 ... 0.12237772  
 FUER SCHIEFE: -0.257788 ... 0.17034573  
 FUER WOELBUNG: -0.51023018 ... 0.34435369

-----  
 AUFLISTUNG DER EINGABEWERTE  
 -----

```
GR.  1: X## 13.07  H##  1
GR.  2: X## 13.12  H##  4
GR.  3: X## 13.17  H##  4
GR.  4: X## 13.22  H##  8
GR.  5: X## 13.27  H## 13
GR.  6: X## 13.32  H## 24
GR.  7: X## 13.37  H## 38
GR.  8: X## 13.42  H## 56
GR.  9: X## 13.47  H## 72
GR. 10: X## 13.52  H## 88
GR. 11: X## 13.57  H## 104
GR. 12: X## 13.62  H## 118
GR. 13: X## 13.67  H## 122
GR. 14: X## 13.72  H## 120
GR. 15: X## 13.77  H##  1
```

-----  
 PRUEFUNG AUF NORMALVERTEILUNG  
 -----

MIT 95 0/0 AUSSAGEWAHRSCHEIN-  
 LICHKEIT KANN DIE ANNAHME EINER  
 NORMALVERTEILUNG NICHT VERWORFEN  
 WERDEN.

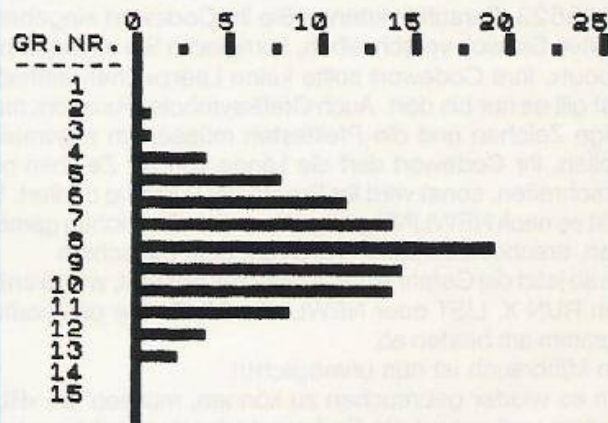
-----  
 HAEUFIGKEITSTABELLE  
 -----

GR. WERT	ABS.	REL. (0/0)	SUMMEN (0/0)
1 13.07	1	0.2	0.2
2 13.12	4	0.8	1.0
3 13.17	4	0.8	1.8
4 13.22	8	1.6	3.4
5 13.27	13	2.6	6.4
6 13.32	24	4.8	13.2
7 13.37	38	7.6	24.2
8 13.42	56	11.2	38.2
9 13.47	72	14.4	57.0
10 13.52	88	17.6	75.2
11 13.57	104	20.8	93.4
12 13.62	118	23.6	107.0
13 13.67	122	24.4	122.4
14 13.72	120	24.0	146.4
15 13.77	1	0.2	147.0

-----  
 STATISTISCHE KENNGROESSEN  
 -----

ANZAHL MESSWERTE.: 500  
 MITTELWERT.....: 13.4264  
 MEDIAN.....: 13.42  
 VARIANZ.....: .013175391  
 STD.ABWEICHUNG.....: 0.11478411  
 VAR.KOEFF. (0/0).....: 0.8549135  
 SCHIEFE.....: -.043721135  
 WOELBUNG.....: -.002938245  
 SPANNWEITE.....: 0.7  
 KLEINSTER WERT.....: 13.07  
 GROESSTER WERT.....: 13.77

-----  
 RELATIVE HAEUFIGKEIT (0/0)  
 -----



Der Programmablauf — mit sieben  
 Hardcopies dokumentiert

# List(iger) Schutz für Programme

**Benutzen ja, LISTen nein. So lautete die Aufgabenstellung für dieses ZX81-Programm, das ihre eigenen Programme vor Mißbrauch schützen soll.**

Zunächst geben Sie das Hilfsprogramm 3 ein und starten es mit RUN. Danach müssen Sie die Zahlen aus der rechten Spalte (ab Adresse 16513 = 249) des Password-Programms in den Computer eintippen.

Die Zeilen 10 bis 60 können Sie dann wieder löschen, Zeile 1 »STOP« ist hinzuzufügen.

Danach können Sie das Programm abSAVEN.

Beabsichtigen Sie, das Password-Programm in Ihre anderen Programme einzubauen? In diesem Falle brauchen Sie es nur einmal abzutippen und auf Kasette zu speichern. Diesmal jedoch zusammen mit dem Hilfsprogramm 2.

Richten Sie sich dazu nach den fünf aufgeführten Punkten:

Nummer 1 reserviert einen Speicherplatz, der auch nach dem Laden eines zweiten Programmes unverändert ist. Bei Nummer 2 beziehungsweise laden Sie das zu schützende Programm ein. Nummer 3 überträgt »PASSWORD« in den geschützten Bereich. Nummer 5 schließlich überträgt »PASSWORD« wieder, nachdem Sie Nummer 4 ausgeführt haben, in einen vorher geschaffenen Platz (wo es auch anfangs war). Die REM-Zeile 2 können Sie nach dieser Prozedur löschen. Es empfiehlt sich, POKE 16419,1 und LIST 1 (ohne Zeilennummern) hinzuzufügen. Um nun »PASSWORD« im Betrieb zu nehmen, starten Sie es bitte mit RUN,GOTO 0 oder RAND USR 16523. Daraufhin können Sie Ihr Codewort eingeben.

Sollten Sie sich verschreiben, korrigieren Sie am besten mit »Rubout«. Ihre Codewort sollte keine Leerzeichen enthalten, sonst gilt es nur bis dort. Auch Grafiksymbole, Funktion, mehrstellige Zeichen und die Pfeiltasten müssen im allgemeinen entfallen. Ihr Codewort darf die Länge von 32 Zeichen nicht überschreiten, sonst wird Ihr Programm vorzeitig codiert. Vorher ist es nach NEWLINE gültig. Wenn Sie alles richtig gemacht haben, erscheint daraufhin »0/0« auf dem Bildschirm.

Da ab jetzt die Gefahr eines Absturzes besteht, wie eventuell durch RUN X, LIST oder NEWLINE, SAVEN Sie das codierte Programm am besten ab.

Ein Mißbrauch ist nun unmöglich!!!

Um es wieder gebrauchen zu können, müssen Sie »RUN« eingeben und erneut ein Codewort (das richtige) benutzen.

Dann liegt das Programm in seiner ursprünglichen Form vor. Wenn Sie das falsche Codewort eingegeben haben, müssen Sie dieses und das richtige hinterher (natürlich getrennt) auf-führen. So ist es also möglich, ein Programm mehrmals zu co-dieren, was zusätzlich Sicherheit bedeutet.

Hier ist Ihrer Phantasie keine Grenze gesetzt!

Zum Beispiel:

```

1. Codierung
RUN
CODEWORT?      Eingabe   : VT0219xy
RUN
CODEWORT?      Eingabe   : 10HKODM
2. Decodierung
RUN
CODEWORT?      Eingabe   : VT0219xy
RUN
CODEWORT?      Eingabe   : 10 HKODM
    
```

(Jens Weber)

## Maschinencode-Listing

Adresse	Befehl		Eingabe
16509	RAND USR 0 XXXXX	(16523)	0,0,192,0, 249,212,28, 126,143,1,22, 0,0,118
16523,4,5 6,7,8	CALL 2602 CALL 16681	: CLS : DE= D__FILE -- 1	205,42,10 205,41,65
9	PUSH DE	:Für später	213
16530	EX DE,HL	:zu HL wird	235
1,2,3	LD BC,340	:340 addiert	1,84,1
4	ADD HL,BC	: +	9
5	EX DE,HL	: DE=Mitte D__FILE	235
6,7,8	LD HL,16688	: Zeiger auf T 1	33,48,85
9,16540,1	CALL 16674	: PRINT	205,34,65
2,3,4	CALL 699	:warten bis	205,187,2
5	LD A,L	:keine Taste	62,125
6	INC A	:mehr ge-	60
7,8	JR NZ,16542	:drückt ist: — 5	32,249
9,16550,1	CALL 699	:warten, bis	205,187,2
2	INC L	:eine Taste	44
3,4	JR Z,16549	:gedrückt: — 4	40,250
5	DEC L	:ist	45
6,7	LD BC,HL	:dann decodier-	77,68
8,9,16560	CALL 1981	:ung.	205,189,7
1	LD A, (HL)	:NEWLINE?	126
2,3	LD B,118	:dann zur	6,118
4	CP B	:Verknüpfung	184
5,6	JR Z,16624	: + : + 59	40,57
7	INC B	:Rubout?	4
8	CP B	:wenn nicht,	184
9,16570	JR NZ,16603	:dieses PRG. über- gehen : + 34	32,32
1,2,3	Call 16681	:HL = Anfang	205,41,65
4	EX DE,HL	:D__FILE	235
5	DOP DE	:DE = D Position	209
6	NOP	:der Eingabe	0
7	LD A,L	:noch keine	125
8	CP E	:Eingabe?	187



9,16580	JR Z,16596	:dann U-Prg: + 17	40,15
1,2	LD A,0	:Eingabe-Position	62,0
3,4,5	LD (16701),A	:um 1 verringern	50,61,65
6,7,8	LD HL,16700	:T 2 ein-	33,60,65
9,16590,1	CALL 16674	:schreiben,HL	205,34,65
2,3	DEC DE:DEC		
	DE	:als Zeiger	27,27
4,5	JR 16600	:Print,dann RET: + 6	24,4
6	LD A,H	:Unterprogr.:	124
7	CP D	:keine	186
8,9	JR NZ,16542	:Eingabe?dann	
		:—56	32,198
16600	PUSH DE	:zurück zur	213
1,2	JR 16542	:Tastatur 59	24,195
16603,4,5	LD (16671),A	:T 2 wird	50,61,65
6,7,8	LD HL,16700	:eingeschrieben,	33,60,65
9	POP DE	:DE auf Ziel,HL	209
16610	INC DE	:als Zeiger	19
1	PUSH DE	: +	213
2,3	LD A,118	:Zeile zu	62,118
4	EX DE,HL	:Ende?	235
5	CP (HL)	:dann zur	190
6,7	JR Z,16624	:Verknüpfung: + 8	40,6
8	EX DE,HL	: +	235
9,16620,1	CALL 16674	:T 2 wird gedruckt,	205,34,65
2,3	JR 16542	:dann zur Tasta- tur:—80	24,174
4	POP DE	:DE aus Stapel	209
5,6,7	Call 16681	:DE = D__FILE — 1	205,41,65
8,9	LD A,0	:keine Eingaben?	62,0
0,1,2	LD HL,DE:INC		
	HL	:dann zurück	98,107,35
3	CP (HL)	:zum	190
4,5	JR Z,16523	:Anfang:—111	40,143
6,7,8	CALL 3875	:FAST	205,35,15
9,16640,1	LD BC,16710	: 1 zu cod.Adr.	1,70,65
16642,3	LD HL,DE	:HL zeigt auf	98,107
4	INC HL	:Codewort	35
5	LD A,(BC)	:Verknüpfung	10
6	XOR (HL)	: +	174
7	LD (BC),A	: +	2
8	INC BC	:nächste Speicher- zelle	3
9,16650	LD A,C : 0	:letzte Zelle?	121,0
1	CP E	:dann	187
2,16653	JR Z,16662	:BASIC : + 10	40,8
4	INC HL	:Codewort zu	35
5,6	LD A,0	:Ende? Dann	62,0
7	CP (HL)	:von vorne, sonst	190
8,9	JR Z,16642	:normal : — 16	40,238
16660,1	JR 16644	:weiter : — 16	24,238
2,3	LD A,B : 0	: U Prg: Speicher	120,0
4	CP D	:zu Ende?	186
5,6	JR NZ,16654	: — 11	32,243
7,8,9	CALL 2602	:flow	205,42,10
16670,1,2	CALL 3883	:CLS	205,43,15
3	RET	:BASIC	201
4,5	LD B,0	:PRINT	6,0
6	LD C,(HL)	:ROUTINE	78
7	INC HL	: +	35
8,9	LDIR	: +	237,176
16680	RET	: +	201
1,2,3	LD HL,16396	: DE =	33,12,64
4	LD E (HL)	: Anfang	94
5	INC HL	: D__FILE	35
6	LD D,(HL)	: +	86
7	RET	: +	201
8,9,16690,1	TEXT 1 Länge	: + »Codewort«	10,48,52,41
2,3,4,5,6		: +	42,60,52,55, 57
7,8,9		: +	0,15,0
16700,1,2,3	TEXT 2 Länge	: + Freistelle	1,0,0,0

## Hilfsprogramm 1

### Bereichsende

Letzte Adresse: N 1 + 256 x N2 — 1

$N1 = (L.Adr. + 1) - INT((L.Adr. + 1)/256) \times 256$

$N2 = INT((L.Adr. + 1)/256)$

POKE 16649,62

POKE 16650,N 1

POKE 16651,185

POKE 16662,62

POKE 16663,N2

POKE 16664,184

### Bereichsanfang

Erste Adresse: X1 + 256 x X2

POKE 16640,X1

POKE 16641,X2

## Hilfsprogramm 2

16716,7,8	CALL 3875	:FAST	205,35,15
9,16720,1	LD HL,16508	:Für das Prg.	33,124,64
2,3,4	LD BC,202	:wird Platz	1,202,0
5,6,7	Call 2462	:geschaffen	205,158,9
8,9,16730	LD HL,32534	:dann wird	33,22,127
1,2,3	LD DE,16509	:es dorthin	17,125,64
4,5,6	LD BC,202	:übertragen	1,202,0
7,8	LDIR	: +	237,176
9,16740,1	CALL 3883	: +	205,43,15
2	RET	: +	201
3,4,5	LD HL,16509	:Das Prg. wird	33,125,64
6,7,8	LD DE,32534	:hinter	17,22,127
9,16750,1	LD BC,234	:RAMTOP	1,234,0
2,3	LDIR	:deponiert	237,176
4	RET	: +	201

39 Bytes

Eingabe: 2 REM genau 39 Leerzeichen  
10 FOR A = 16716 TO 16754  
20 INPUT B  
30 POKE A,B  
40 NEXT A

Geben Sie die Zahlen aus der rechten Spalte ein. Die Zeilen 10 bis 40 können Sie danach wieder löschen.

## Anwendung

1. POKE 16388,21 POKE 16389,127 NEW
2. Passwort und Hilfsprogramm eintippen beziehungsweise einladen.
3. RAN USR 16743
4. Anderes Programm ebenfalls einladen.
5. RAND USR 32741

## Hilfsprogramm 3

Eingabe des Hauptprogramms:

10 REM genau 190 Leerzei-  
chen.

Kontrolle: PRINT PEEK 16511 — 2  
POKE 16510,0 (Edit-Schutz)

10 FOR A = 16513 TO 16700

20 SCROLL

30 INPUT B

40 POKE A,B

50 PRINT A, PEEK A

60 NEXT A

Geben Sie die Zahlen aus der rechten Spalte ein.

# MC-Scroll

MC-Scroll ist eine Routine, die in der REM-Zeile des ZX81 Platz findet. Sie kann in jedes Basic-Programm vor- oder nachher geschrieben werden. Vorausgesetzt ist, daß die PRINT-Anweisung mit »AT« festgelegt ist.

Scroll-Programme gibt es zwar viele, aber nicht alle sind so kurz und simpel zu benutzen.

Das Programm kann: Scrollen in alle vier Richtungen auf einen Schlag; Bilder verschieben oder verschwinden lassen; Texte versetzen oder verdoppeln; Bilder und Grafik zeichnen. Ich habe zur Demonstration die Aufrufadressen auf die Cursor-tasten gelegt und die Startadresse auf 16514. Natürlich kann das Programm auch woanders in die REM-Zeile gePOKET werden.

Nun zur Eingabe: Starten Sie mit RUN und geben Sie die dezimalen Werte ein. POKEn Sie nun 16510,0 ein; dadurch wird die erste REM-Zeile auf Zeilennummer 0 gesetzt und vor EDIT geschützt.

Programmprüfung: Geben Sie folgendes Basic-Programm ein:

```
2 PRINT AT 10,10;»+«
3 IF INKEY$ = »6« THEN RUN USR 16514
4 IF INKEY$ = »7« THEN RUN USR 16541
5 IF INKEY$ = »8« THEN RUN USR 16557
6 IF INKEY$ = »5« THEN RUN USR 16584
7 GOTO 3
```

Das Test- und Eingabeprogramm kann schließlich gelöscht werden und das Scroll-Programm mit SAVE auf Tonband aufgenommen werden.

(Josef Herrmann)

```
1 REM 96 BELIEBIGE ZEICHEN
2 PRINT AT 14,4;"BY HERMANN
JOSEF OBERLAENDERW
EG 14 8415 NITTENA
U 1.6.1984

3 IF INKEY$="6" THEN RUN USR
16514
4 IF INKEY$="7" THEN RUN USR
16541
5 IF INKEY$="8" THEN RUN USR
16557
6 IF INKEY$="5" THEN RUN USR
16584
8 RUN
10 SAVE "MC-SCROLL"
20 PRINT AT 1,0;"**DEMOPROGRAM
ZUM MC-SCROLL** AB ZEILE >1
< STEHT INNEN IHR BASIC-PROGR
AM ZUR VERFUEGUNG "
30 PRINT AT 5,0;"DIE MC-ROUTIN
E FUNKTIONIERT, WENN PRINT >
AT < ANGEWANNT WIRD "
40 PRINT AT 10,0;"DRUECKEN SIE
JEZT TASTE 5/7/6/8"
50 IF INKEY$="" THEN GOTO 2
```

Basic-Listing zur Demonstration des Programms

16514	=	193	16516	=	69	16562	=	84	16563	=	93
16516	=	42	16517	=	12	16564	=	43	16565	=	62
16518	=	64	16519	=	1	16566	=	24	16567	=	1
16520	=	181	16521	=	2	16568	=	31	16569	=	0
16522	=	9	16523	=	34	16570	=	237	16571	=	184
16524	=	130	16525	=	64	16572	=	245	16573	=	151
16526	=	1	16527	=	33	16574	=	18	16575	=	43
16528	=	0	16529	=	9	16576	=	43	16577	=	27
16530	=	84	16531	=	93	16578	=	27	16579	=	241
16532	=	42	16533	=	130	16580	=	61	16581	=	32
16534	=	64	16535	=	1	16582	=	240	16583	=	201
16536	=	181	16537	=	2	16584	=	42	16585	=	12
16538	=	237	16539	=	184	16586	=	64	16587	=	35
16540	=	201	16541	=	42	16588	=	34	16589	=	93
16542	=	12	16543	=	64	16590	=	35	16591	=	62
16544	=	93	16545	=	94	16592	=	24	16593	=	1
16546	=	19	16547	=	1	16594	=	31	16595	=	0
16548	=	34	16549	=	0	16596	=	237	16597	=	176
16550	=	9	16551	=	1	16598	=	245	16599	=	151
16552	=	192	16553	=	2	16600	=	18	16601	=	19
16554	=	237	16555	=	176	16602	=	35	16603	=	19
16556	=	201	16557	=	42	16604	=	35	16605	=	241
16558	=	16	16559	=	64	16606	=	61	16607	=	32
16560	=	43	16561	=	43	16608	=	240	16609	=	201

Dezimal-Listing mit Adressen

# 24zeilige Display-Kopie auf dem Drucker

Laut Sinclair-Handbuch sind die beiden unteren Zeilen des Bildschirms für Input und Fehlermeldungen reserviert und lassen sich weder mit PRINT noch mit PLOT ansprechen. Daß dies nur bedingt zutrifft, dürfte den meisten ZX81-Freaks schon bekannt sein.

```

*****
*          ZEILE 1          *
*          ZEILE 2          *
*          ZEILE 3          *
*          ZEILE 4          *
*          ZEILE 5          *
*          ZEILE 6          *
*          ZEILE 7          *
*          ZEILE 8          *
*          ZEILE 9          *
*          ZEILE 10         *
*          ZEILE 11         *
*          ZEILE 12         *
*          ZEILE 13         *
*          ZEILE 14         *
*          ZEILE 15         *
*          ZEILE 16         *
*          ZEILE 17         *
*          ZEILE 18         *
*          ZEILE 19         *
*          ZEILE 20         *
*          ZEILE 21         *
*          ZEILE 22         *
*          ZEILE 23         *
*****

```

## Ausdruckversuch aller 24 Zeilen

Mit Hilfe der Systemvariablen D-FILE (PEEK 16936+256\*PEEK 16397) läßt sich die Lage des Bildspeichers feststellen und auch in die unteren beiden Zeilen ein oder mehrere Zeichen einPOKEIn. Außerdem besteht die Möglichkeit, durch temporäres Verändern der Systemvariablen DF-SZ (POKE 16418,0) die Ausführung von Befehlen wie PRINT AT 23,0; "TEXT" zuzulassen und damit ein 24zeiliges Display zu erreichen. Allerdings muß hierbei vor dem nächsten Inputbefehl die

Systemvariable auf ihren alten Wert zurückgesetzt werden (POKE 16418,2), da sonst das System zusammenbricht.

Bei der Dokumentation eines Programmes stellte sich das Problem, ein solchermaßen erstelltes 24zeiliges Display auf dem Drucker auszugeben. Zunächst widersetzte sich der Rechner allen diesbezüglichen Bemühungen. Es wurden selbst bei im Programm an entsprechender Stelle eingesetzten COPY-Befehl und bei veränderter Systemvariable DF\_SZ immer nur die üblichen 22 Zeilen auf dem Drucker ausgegeben.

Das Problem wurde schließlich mit der im Listing gezeigten kleinen Subroutine gelöst. Im Programm an der richtigen Stelle aufgerufen, stellt sie den gesamten Bildspeicher mit allen 24 Zeilen in einen alphanumerischen String (P\$), der sich dann einfach auf dem Drucker ausgeben läßt. Vor der Rückkehr aus der Subroutine wird der String zur Einsparung von Speicherplatz wieder auf eine Minimallänge dimensioniert.

Die kleine Subroutine läßt sich natürlich nicht nur temporär für Dokumentationszwecke, sondern auch dauerhaft in Programmen einbauen, wenn dies gewünscht ist. Die Routine benötigt selbst im FAST-Modus einige Sekunden, bis ausgedruckt wird. Sicherlich ließe sich auch eine kleine äquivalente USR-Routine schreiben, aber hier besteht immer das Problem, daß dann entweder Veränderungen des RAMTOP oder das EinPOKEIn in eine REM-Zeile erforderlich wird, so daß der Gewinn an Geschwindigkeit wohl meist durch den Aufwand zunichte gemacht wird. (Bernd W. Friedrich)

```

10 REM ERSTELLUNG EINES
20 REM 24-ZEILEN TESTDISPLAYS
30 LET S$="*****"
40 POKE 16418,0
50 PRINT S$;TAB 22;S$;AT 23,0;
S$;TAB 22;S$
60 FOR I=0 TO 23
70 PRINT AT I,0;"#";TAB 12;"ZE
ILE ";I+1;TAB 31;"#"
80 NEXT I
90 REM
100 REM AUSGABE AUF DRUCKER
110 GOSUB 150
120 POKE 16418,2
130 STOP
140 REM
150 REM SUBROUTINE DRUCK24
160 FAST
170 DIM P$(768)
180 LET PP=0
190 LET TV=PEEK 16396+256*PEEK
16397
200 FOR X=1 TO 791
210 IF X/33=INT (X/33) THEN GOT
O 240
220 LET PP=PP+1
230 LET P$(PP)=CHR$(PEEK (X+TV))
240 NEXT X
250 LPRINT P$
260 DIM P$(1)
270 RETURN

```

Die Druck-Subroutine umfaßt nur die Zeilen 150-270, der Rest ist ein Testrahmen für die Routine

## 3 Tips für den ZX81

Statt ein Programm mit LOAD'''' zu laden, kann man auch folgende ROM-Routine benutzen:  
FAST/RAND USR 836

Diese Routine stoppt das Programm nach erfolgreichem laden mit der Fehlermeldung C/0. Danach kann man das Programm problemlos listen und starten.

Der Befehl POKE 16510,0 ändert die erste Zeilennummer eines Programms in 0. Diese Zeile kann zwar noch gelistet werden, aber man kann Sie nicht mehr editieren.

Mit dem Befehl POKE 16418,0 erhält man statt 22 nun 24 Zeichen auf dem Bildschirm. Es ist nun sogar der Befehl PRINT AT 23,31 möglich.

# Rund um den

## Die Idee zu diesem Programm entsprang einer Diskussion, in welchem der kommenden Jahre wohl die Weihnachtsfeiertage und das Jahresende wieder so arbeitgeberfreundlich liegen würden wie 1983.

Das Ergebnis der Studien älterer und neuerer Lexika sowie anderer Literatur schlug sich in dem hier vorgestellten Programm »Rund um den Kalender« nieder. Die folgende Anwendungsbeschreibung soll aufzeigen, was das Programm leisten kann.

Nach dem Laden von Kassette startet das Programm automatisch und meldet sich auf dem Bildschirm. Hierzu ein kleiner Hinweis: Es sollte keine Panik aufkommen, wenn der Ladevorgang beendet ist, der Bildschirm aber zunächst noch dunkel bleibt. Das Programm durchläuft einen Initialisierungsteil im FAST-Modus; das dauert etwa zwei Sekunden.

### Datumseingabe

Der Rechner möchte nun das aktuelle Datum wissen. Die Eingabe wird gespeichert und später im weiteren Verlauf des Programms bei Bedarf wieder verwendet.

Das Datum ist prinzipiell in der Anordnung »TT.MM.JJJJ« einzugeben. Die Eingabe wird auf diese Syntax geprüft, Abweichungen führen zu einer Fehlermeldung und der Aufforderung zur Korrektur. Außerdem prüft das Programm, ob das eingegebene Datum überhaupt möglich ist; der 32.01.1984 wird genauso zurückgewiesen wie der 29.02.1983 (kein Schaltjahr!). Da das Programm für den gregorianischen Kalender konzipiert ist, werden auch nur Daten ab dem 15.10.1582, das ist der Zeitpunkt der gregorianischen Kalenderform, akzeptiert. Die Prüfung findet auch bei allen weiteren im Programm vorkommenden Datumseingaben statt. Soll das Programm aus irgendwelchen Gründen bereits hier abgebrochen werden, gibt man statt des Datums ein »X« ein. Dadurch wird das Programm ordnungsgemäß mit einer Endmeldung verlassen. Bei den später im Programm anstehenden Datumseingaben führt diese Eingabe normalerweise zurück ins Auswahlmü.

Das richtig eingegebene Datum quittiert der Rechner durch Ausgabe auf dem Display, ergänzt durch den bereits vom Programm selbsttätig ermittelten Wochentag. Nach Betätigen der N/L-Taste gelangt man nun ins Auswahlmü. Hier kann aus den angebotenen Möglichkeiten eine gewünschte Anwendung durch Betätigen der entsprechenden Taste ausgewählt werden. Die einzelnen Möglichkeiten sind im folgenden detailliert beschrieben.

### Wochentag

Wie auf dem neu erschienenen Schirmbild mitgeteilt, kann für ein vorgegebenes Datum der Wochentag ermittelt werden. Wird ein Datum eingegeben, erscheint nach kurzer Denkpa-

se das Datum und der zugehörige Wochentag auf dem Schirm. Bei Eingabe eines »X« wäre das Programm ins Menü zurückgekehrt. Bei bloßer Betätigung der N/L-Taste ohne weitere Eingabe nimmt der Rechner das aktuelle Datum an.

Nachdem sich der Rechner mit der Antwort auf die Eingabe gemeldet hat, kann gemäß dem unten auf dem Schirm sichtbaren Submenü ausgewählt werden, ob auf einem angeschlossenen Drucker eine Hardcopy des Schirmbildes erzeugt werden soll, ob ins Menü zurückgekehrt werden soll, oder ob die gleiche Abfrage für eine andere Datumsangabe durchgeführt werden soll.

Es besteht also die Möglichkeit, festzustellen, auf welchen Wochentag der nächsten Geburtstag fällt, oder ob der 1.1.2000 ein Samstag ist.

### Abstand

Hier läßt sich der Abstand zwischen zwei (eingegebenen) Daten in Tagen ermitteln. Der Computer antwortet mit der Meldung, wieviele Tage zwischen den beiden Daten liegen, wobei er es mit der Grammatik genau nimmt: bei nur einem Tag Abstand schreibt er statt »Tage« nur das Wort »Tag«.

Mit dieser Abfrage kann beispielsweise festgestellt werden, wie alt man ist, oder wieviele Tage es noch bis zur nächsten Weihnachtsbescherung sind.

### Tagesnummer

Dieser Programmteil ermittelt die Nummer eines eingegebenen Tages im Kalenderjahr. Der 15.11.1984 ist beispielsweise der 320. Tag im Jahr. Auch hier wird auf die Grammatik geachtet: bei gegenüber dem aktuellen Datum zurückliegenden Daten wird in der Ausgabe das Wort »ist« durch »war« ersetzt.

### Bestimmter Wochentag

Mit Eingabe »Samstag« und »24.12.1983« wird spezifiziert, daß man wissen möchte, in welchem der kommenden Jahre der 24.12. ein Samstag ist. Für die Lösung dieser Aufgabe benötigt der Rechner einige Zeit. Es vergehen etwa eineinhalb Minuten, dann erscheint auf dem Bildschirm eine Liste von Daten, die aufgrund der Abfrage ausgesucht wurden. Die Liste ist auf 20 Zeilen begrenzt. So hätte sich die eingangs geschilderte Diskussion leicht beenden lassen.

### Feiertage

Für ein vorgegebenes Jahr werden die Feiertage ermittelt. Bei den feststehenden Feiertagen stellt das kein Problem dar, da hier jeweils nur der Wochentag ermittelt werden muß. Problematischer ist es mit den beweglichen Feiertagen. Als Schlüssel wird das Osterdatum berechnet. Zwischen diesem und den meisten anderen beweglichen Feiertagen bestehen zahlenmäßige Zusammenhänge, die rechnerisch erfaßt werden können und schließlich zur Ermittlung der übrigen Feiertage dienen.

Da die Feiertage nicht bundeseinheitlich sind, sondern von Land zu Land variieren, wurde im Originalprogramm alles aufgenommen, was in Rheinland-Pfalz und in den angrenzenden Bundesländern als Feiertag zählt. So kam auch der Rosenmontag ins Spiel, der ja hier in Mainz zu den höchsten Feiertagen zählt. Nicht aufgenommen wurde Mariä Himmelfahrt (nur in Bayern ein gesetzlicher Feiertag). Bei Bedarf kann das Pro-

# Kalender

gramm mit geringfügigen Änderungen auf andere örtliche Verhältnisse umgeschrieben werden. Innerhalb der Logik wird auch die Tatsache berücksichtigt, daß es den 17.6. und den 1.5. zu Papst Gregors Zeiten noch nicht als Feiertage gab. Die Feiertage werden ebenfalls als Liste auf dem Bildschirm ausgegeben. Da die grundlegende Berechnung des Osterdatums auf einer Formel mit Hilfstabelle basiert, können Feiertage nur im Zeitraum 1583 bis 2399 berechnet werden.

## Kalender-Display

Diese Auswahl läßt sofort einen weiteren Schirm mit einem Submenü erscheinen. Man muß sich entscheiden, ob nur ein einzelner Monat oder ein ganzes Jahr angezeigt werden soll. Der Unterschied besteht darin, daß bei der Auswahl »Monat« nach dem Display des Kalenders mit der Taste »V« ins Menü zurückgekehrt wird, während bei der Auswahl »Jahr« durch den ganzen Jahreskalender geblättert wird, bevor der Rücksprung ins Menü erfolgt.

Da bei der Berechnung der Kalenderblätter keine Feiertage im Spiel sind, kann jeder Monat in den Jahren 1583 bis 9999 als Display erscheinen. Selbst für das Jahr 1582 ist über den Monatsabruf ein Display für die letzten Monate des Jahres möglich, der Oktober erscheint allerdings nur teilweise. Der Grund hierfür ist, daß bei der Kalenderreform einige Wochentage übersprungen wurden.

## Kalender-Druck

Ist ein Drucker vorhanden, kann mit dieser Auswahl ein vollständiger Kalender für ein beliebiges Jahr zwischen 1583 und 2399 ausgedruckt werden. Da am Ende des Kalenders eine Liste der Feiertage ausgegeben wird, ist hier die Auswahl etwas eingeschränkt. Allerdings dürfte der vorhandene Spielraum normalerweise allen vernünftigen Anforderungen genügen.

```

FEIERTAGE      1985
01.01.1985, DI  NEUJAHR
06.01.1985, SO  HL.3 KOENIGE
18.02.1985, MO  ROSENMONTAG
05.04.1985, FR  KARFREITAG
07.04.1985, SO  OSTERSONNTAG
08.04.1985, MO  OSTERMONTAG
01.05.1985, MI  MAIFEIERTAG
16.05.1985, DO  CHR. HIMMELFAHRT
26.05.1985, SO  PFINGSTSONNTAG
27.05.1985, MO  PFINGSTMONTAG
06.06.1985, DO  FRONLEICHNAM
17.06.1985, MO  T.D.DT.EINHEIT
01.11.1985, FR  ALLERHEILIGEN
20.11.1985, MI  BUSSTAG
24.12.1985, DI  HEILIGABEND
25.12.1985, MI  1.WEIHNACHTSTAG
26.12.1985, DO  2.WEIHNACHTSTAG
31.12.1985, DI  SILVESTER
  
```

Ausgabe der 85'er Feiertage

Am Anfang des Kalenders wird die Jahreszahl in Großschrift ausgedruckt. Durch Einfügen einiger temporärer PRINT-Befehle kann oberhalb der Jahreszahl noch ein sinniger Spruch oder eine Widmung ausgedruckt werden. Damit eignet sich der Printout recht gut als kleine Aufmerksamkeit für Bekannte und Verwandte. Der Kalender hat einen recht guten Anklang gefunden. Es hat sich gezeigt, daß hier das sonst so belächelte Metallpapier des ZX-Printers durchaus auch positive Seiten hat. Und Tante Ottilie hat sich besonders über den für sie bestimmten Kalender mit ihrem Lieblingsspruch gefreut.

## Programm beenden

Diese Auswahl ist der offizielle Weg, aus dem Programm auszusteigen.

## Kalender-Algorithmen

### Gregorianische Kalenderreform

Am 15.10.1582 wurde der gregorianische Kalender eingeführt und löste den bis dahin üblichen julianischen Kalender ab. Während er in den katholischen Ländern sofort Gültigkeit erlangte, wurde er in anderen Ländern teilweise erst Jahrhunderte später eingeführt.

### Schaltjahre

Ohne Rest durch vier teilbare Jahre sind Schaltjahre. Ausnahme von dieser Regelung: Von den Jahrhundertjahren sind nur diejenigen Schaltjahre, die auch ohne Rest durch vierhundert teilbar sind.

### Osterfest

Das Osterdatum läßt sich berechnen. Es stellt, wie bereits erwähnt, den Schlüssel zu den meisten anderen beweglichen Feiertagen dar. Berechnung: Man dividiert die Jahreszahl (j) durch vier, wobei etwaiger Rest unberücksichtigt bleibt und erhält so die Zahl  $q=j/4$ ; nun dividiert man  $j / 19$ , bleibt Rest a

$(M-11a) / 30$ , bleibt Rest b

$(j+q+b-D) / 7$ , bleibt Rest C

Dann ist der  $(28+b-c)$ te, für Zahlen über 31 der  $(28+b-c-31)$ te März beziehungsweise April der Ostersonntag. D und M sind der folgenden Tabelle zu entnehmen:

Zeitraum	D	M	Zeitraum	D	M
1582-1699	10	202 *)	2100-2199	14	204 *)
1700-1799	11	203	2200-2299	15	205 *)
1800-1899	12	203	2300-2399	16	206
1900-2099	13	204 *)			

\*) ergibt sich  $b=29$  oder  $b=28$ , so ist im ersten Fall mit  $b=28$ , im zweiten mit  $b=27$  weiterzurechnen.

Quelle: Kleine Enzyklopädie Natur, Verlag Enzyklopädie, Leipzig, 1957

Tabelle zur Errechnung des Osterfestes

## Rosenmontag

Der Rosenmontag liegt zwei Tage vor Aschermittwoch und damit 48 Tage vor dem Ostersonntag.

## Chr. Himmelfahrt

Dieser Feiertag, immer ein Donnerstag, liegt 39 Tage nach dem Ostersonntag.

## Pfingsten

Der Pfingstsonntag ist 49 Tage nach dem Ostersonntag.

## Fronleichnam

Fronleichnam fällt immer auf einen Donnerstag. Er liegt 60 Tage nach dem Ostersonntag.

## Bußtag

Dieser Feiertag hat keinen erkennbaren Zusammenhang mit Ostern. Er fällt immer auf einen Mittwoch, 11 Tage vor dem ersten Advent. Damit kann eine Verbindung zum Weihnachtsfest und damit zum Jahresende hergestellt werden.

Der Bußtag liegt x-Tage vor dem Jahresende.

Silvester	So	Mo	Di	Mi	Do	Fr	Sa
x	39	40	41	42	43	44	45

## Tag der deutschen Einheit

Gedenktag zur Erinnerung an den Volksaufstand in Ost-Berlin und der DDR am 17.6.1953. Für die Programmlogik ist wichtig, daß vor 1954 der 17.6. kein Feiertag war.

## Maifeiertag

Weltfeiertag der Arbeiterschaft am 1. Mai. Seit Beschluß der 2. Internationale im Jahre 1889, gesetzlicher Feiertag in vielen Ländern. Auch hier ist für die Programmlogik wichtig, daß vor 1889 der 1. Mai kein Feiertag war.

## Programmbeschreibung

Das Programm gliedert sich in drei Hauptabschnitte: Initialisierung, Subroutinen und Mainline.

Im Initialisierungsteil werden zunächst die häufigsten Zahlenwerte zu Variablen zugeordnet. Dies bringt eine nicht unbedeutliche Einsparung an Speicherplatz. Da das Programm recht umfangreich ist (etwa 14 KByte), mußte jede Möglichkeit der Speicherplatzreduzierung in Betracht gezogen werden, um es in den vorhandenen 16 KByte unterzubringen. Im übrigen hat sich diese Vorgehensweise bereits in früher erstellten Programmen bewährt und ist zum festen Bestandteil der meisten Eigenentwicklungen geworden.

### AUSWAHLEN

- 1 WOCHENTAG
- 2 ABSTAND
- 3 TAGESNUMMER
- 4 BESTIMMTER WOCHENTAG
- 5 FEIERTAGE
- 6 KALENDER-DISPLAY
- 7 KALENDER-DRUCK
- 8 PROGRAMM BEENDEN

Eingabe der gewünschten Berechnung

SEPTEMBER							1984
SO	MO	DI	MI	DO	FR	SA	
						1	
2	3	4	5	6	7	8	
9	10	11	12	13	14	15	
16	17	18	19	20	21	22	
23	24	25	26	27	28	29	
30							

Kalenderausdruck September 1984

Außerdem spezifizierte der Initialisierungsteil neben einigen Festtexten, die wiederholt benötigt werden, auch Tabellen mit Wochentags- und Monatsbezeichnungen sowie eine Tabelle mit der Anzahl Tage pro Monat. Besonders zu erwähnen ist die Tabelle 0\$. Sie enthält die Daten der Hilfstabelle zur Errechnung des Osterdatums und ist zwecks Platzersparnis im Charakterformat und in verkürzter Form vorhanden. Redundante Datenteile sind weggelassen, sie werden beim Lesen der Tabelle wieder hinzugefügt.

Die Subroutinen übernehmen den größten Teil der Berechnungen und die im Programm wiederholt anfallenden Arbeiten. Dies hat neben dem Fortfall von Redundanzen (Speicherplatzersparnis) den Vorteil, daß bestimmte Vorgänge getrennt ausgetestet werden konnten. Innerhalb der Routine fällt die Feiertagstabelle auf. Sie wurde hier, und nicht im Initialisierungsteil, untergebracht, weil sie bei wiederholtem Aufruf der Feiertagsroutinen jedesmal neu in Rohform zur Verfügung stehen soll.

Große Zahlenwerte sind im allgemeinen mit dem Befehl »VAL« verbunden. Damit wird ebenfalls Speicherplatz gespart.

Im »Mainline« wird im wesentlichen das Zusammenspiel der Subroutinen, je nach gestellter Aufgabe, gesteuert.

Die Aufbereitung des Kalenderdruckes erfolgt im FAST-Modus im Bildschirmspeicher. Nach Aufbereitung wird mit COPY gedruckt. Deshalb mußten keine speziellen Druckbefehle verwendet werden, und es konnte ein großer Teil der Displayroutinen eingesetzt werden. Die Ausgabe der großen Jahreszahl am Kalenderanfang erfolgt mit einer kleinen Routine, die auf den Zeichengenerator des ZX-ROM zugreift. Der Druck eines kompletten Kalenders dauert etwa acht Minuten.

Soll ein Kalenderspruch oder eine Widmung eingebaut werden, sind ein paar temporäre Print-Befehle zwischen die Zeilen 3710 und 3720 einzufügen.

Soll zur Ausgabe des Kalenders ein anderer als der ZX-Printer verwendet werden, sollte dieser grafikfähig sein. Eventuell ist das Programm den Gegebenheiten anzupassen (zum Beispiel Kalenderrahmen aus Sternchen zusammensetzen).

Bei Eingaben wird prinzipiell in den SLOW-Modus geschaltet, um eine flackerfreie Eingabe zu erzielen.

Besonders großer Wert wurde auf verständliche Dialoge und auf das Abhandeln falscher Eingaben gelegt. Es sollte nicht möglich sein, mit falschen Eingaben das Programm zum ungewollten Ausstieg zu bringen.

Nach erfolgtem Eintippen und Austesten des Programms wird mit dem Befehl »GOTO 5850« auf Kassette gespeichert.

(Bernd W. Friedrich)

## Basic-Listing »Kalender«

```

10 REM *****
20 REM *RUND UM DEN KALENDER*
30 REM * (C) B.U.FRIEDRICH *
40 REM * 6500 MAINZ 1 1/84 *
50 REM *****
60 FAST
70 REM INITIALISIERUNG
80 REM -----
90 LET A0=NOT PI
100 LET A1=SGN PI
110 LET A2=A1+A1
120 LET A3=A2+A1
130 LET A4=A2+A2
140 LET A5=A3+A2
150 LET A6=A3+A3
160 LET A7=A4+A3
170 LET A8=A4+A4
180 LET A9=A5+A4
190 LET B0=A5+A5
200 LET B1=B0+A1
210 LET B2=B0+A2
220 LET B3=B0+A3
230 LET B4=B0+A4
240 LET B5=B0+A5
250 LET B6=B0+A6
260 LET B7=B0+A7
270 LET B8=B0+A8
280 LET B9=B0+A9
290 LET C0=B0+B0
300 LET C1=C0+A1
310 LET C2=C0+A2
320 LET C3=C0+A3
330 LET C4=C0+A4
340 LET C5=C0+A5
350 LET C6=C0+A6
360 LET C7=C0+A7
370 LET C8=C0+A8
380 LET C9=C0+A9
390 LET D0=C0+B0
400 LET D1=D0+A1
410 LET D2=D0+A2
420 LET H1=A6+C0
430 LET H4=H1*A4
440 DIM L$(D2)
450 FOR I=A6 TO C8
460 LET L$(I)="*"
470 NEXT I
480 LET A$=" BITTE "
490 LET B$="DATUM EINGEBEN:"
500 LET C$="TT.MM.JJJJ"
510 LET E$=" EINGABEFehler - KO
RAIGIEREN"
520 LET F$="WOCHENTAG"
530 LET G$="N/L"
540 LET J$="BERECHNUNG NICHT M
DEGLICH"
550 DIM O$(A7,A7)
560 LET O$(A1)="1515021"
570 LET O$(A2)="1717130"
580 LET O$(A3)="1818230"
590 LET O$(A4)="1920341"
600 LET O$(A5)="2121441"
610 LET O$(A6)="2222551"
620 LET O$(A7)="2323660"
630 DIM T$(A7,B0)
640 LET T$(A1)="SONNTAG"
650 LET T$(A2)="MONTAG"
660 LET T$(A3)="DIENSTAG"
670 LET T$(A4)="MITTWOCH"
680 LET T$(A5)="DONNERSTAG"
690 LET T$(A6)="FREITAG"
700 LET T$(A7)="SAMSTAG"
710 DIM M$(B2,A9)
720 LET M$(A1)="JANUAR"
730 LET M$(A2)="FEBRUAR"
740 LET M$(A3)="MÄRZ"
750 LET M$(A4)="APRIL"
760 LET M$(A5)="MAI"
770 LET M$(A6)="JUNI"
780 LET M$(A7)="JULI"
790 LET M$(A8)="AUGUST"
800 LET M$(A9)="SEPTEMBER"
810 LET M$(B0)="OKTOBER"
820 LET M$(B1)="NOVEMBER"
830 LET M$(B2)="DEZEMBER"
840 DIM T(B2)
850 FOR I=A1 TO B2
860 LET T(I)=D1
870 NEXT I
880 LET T(A2)=C8
890 LET T(A4)=D0
900 LET T(A6)=D0
910 LET T(A8)=D0
920 LET T(B1)=D0
930 LET H$=""
940 GOTO 3920
950 REM SUBROUTINEN
960 REM -----
970 REM SR DAT.EING.U.PRFG.
980 SLOW
990 LET D$=""
1000 INPUT D$
1010 FAST
1020 IF D$="X" THEN RETURN
1030 IF D$="" AND H$(">)" THEN LE
T D$=H$
1040 IF LEN D$(">")B0 THEN GOTO 120
0
1050 IF D$(A3(">")) OR D$(A6(">"))
." THEN GOTO 1200
1060 FOR I=A1 TO B0
1070 IF I=A3 OR I=A6 THEN GOTO 1
090
1080 IF CODE D$(I)<C8 OR CODE D$(
I)>D2+A5 THEN GOTO 1200
1090 NEXT I
1100 LET J=VAL D$(A7 TO )
1110 LET M=VAL D$(A4 TO A5)
1120 LET D=VAL D$( TO A2)
1130 IF J>1582 THEN GOTO 1160
1140 IF J<1582 THEN GOTO 1220
1150 IF M*A1+D<VAL "1015" THEN G
OTO 1220
1160 GOSUB 1310
1170 IF M<A1 OR M>B2 THEN GOTO 1
200
1180 IF D>A0 AND D<=T(M) THEN RE
TURN
1190 IF M=A2 AND D<=T(M)+5 THEN
RETURN
1200 PRINT AT B0,A0;L$;AT B0,A0;
E$
1210 GOTO 980
1220 GOSUB 1580
1230 PRINT AT B4,B0;D$;AT B0,A0;
J$;AT C1,A0;A$;G$;B$(A6 TO )
1240 SLOW
1250 INPUT D$
1260 FAST
1270 IF D$(">)" THEN GOTO 1240
1280 LET D$="X"
1290 RETURN
1300 REM SR SCHALTJAHRPRFG.
1310 LET S=A0
1320 IF J/A4(">")INT (J/A4) THEN RE
TURN
1330 IF J/H1(">")INT (J/H1) THEN GO
TO 1350
1340 IF J/H4(">")INT (J/H4) THEN RE
TURN
1350 LET S=A1
1360 RETURN
1370 REM SR TAG
1380 LET U=A0
1390 IF M>A2 THEN GOTO 1420
1400 LET X=D1*(M-A1)
1410 GOTO 1440
1420 IF S=A1 THEN LET U=A1
1430 LET X=VAL "INT ((306*M-324)
/10)"
1440 LET Z=J-A1
1450 LET A=VAL "U+Z*365+INT (Z/A

```

```

4) -INT (Z/H1)+INT (Z/H4)+X+D"
1450 LET X=A1+A-INT (A/A7)*A7
1470 RETURN
1480 REM SR ASK
1490 LET TV=VAL "760+PEEK 16396+
PEEK 16397*256"
1500 POKE TV,VAL "143"
1510 SLOW
1520 IF INKEY$("<") THEN GOTO 152
0
1530 IF INKEY$="" THEN GOTO 1530
1540 LET I$=INKEY$
1550 FAST
1560 RETURN
1570 REM SR CLRSCREEN
1580 FOR I=B4 TO C1
1590 PRINT AT I,A0;L$
1600 NEXT I
1610 RETURN
1620 REM SR WEITER
1630 PRINT AT C1,A0;" ";CHR$ VAL
"148";" ";CHR$ VAL "14
8";" ";CHR$ VAL "148"
;" ";CHR$ VAL "148"
;" ";CHR$ VAL "148"
1640 GOSUB 1490
1650 IF I$("<")="Z" THEN GOTO 1690
1660 PRINT AT C1,A0;L$
1670 COPY
1680 GOTO 1630
1690 IF I$="X" OR I$="U" THEN RE
TURN
1700 GOTO 1640
1710 REM SR OSTERN
1720 FOR I=A1 TO A7
1730 IF J<H1+VAL O$(I, TO A2) OR
J>H1-A1+H1+VAL O$(I,A3 TO A4) T
HEN GOTO 1770
1740 LET C=B0+VAL O$(I,A5)
1750 LET E=H1+H1+VAL O$(I,A6)
1760 GOTO 1790
1770 NEXT I
1780 RETURN
1790 LET O=INT (J/A4)
1800 LET F=J-(INT (J/B9)*B9)
1810 LET G=E-B1+F-(INT ((E-B1)*F)
/D0)*D0)
1820 IF NOT VAL O$(I,A7) THEN GO
TO 1840
1830 IF G=C8 OR G=C9 THEN LET G=
G-A1
1840 LET H=J+O+G-C-(INT ((J+O+G-
C)/A7)*A7)
1850 LET O=C8+G-H
1860 LET P=A3
1870 IF O<=D1 THEN RETURN
1880 LET O=O-D1
1890 LET P=A4
1900 RETURN
1910 REM SR FT-TAB.
1920 DIM K$(B8,C0)
1930 LET K$(A1)="0101 NEUJAHR"
1940 LET K$(A2)="0601 HL.3 KOENI
GE"
1950 LET K$(A3)=" 2ROSENMONTA
G"
1960 LET K$(A4)=" 6KARFREITAG
"
1970 LET K$(A5)=" 1OSTERSONNT
AG"
1980 LET K$(A6)=" 2OSTERMONTA
G"
1990 LET K$(A7)="0105 MAIFEIERTA
G"
2000 LET K$(A8)=" 5CHR.HIMMEL
FAHRT"
2010 LET K$(A9)=" 1PFINGSTSON
NTAG"
2020 LET K$(B0)=" 2PFINGSTMON
TAG"
2030 LET K$(B1)=" 5FRONLEICHN
AM"

```

```

2040 LET K$(B2)="1706 T.O.DT.EIN
HEIT"
2050 LET K$(B3)="0111 ALLERHEILI
GEN"
2060 LET K$(B4)=" 4BUSSTAG"
2070 LET K$(B5)="2412 HEILIGABEN
D"
2080 LET K$(B6)="2512 1.WEIHNACH
TSTAG"
2090 LET K$(B7)="2612 2.WEIHNACH
TSTAG"
2100 LET K$(B8)="3112 SILVESTER"
2110 RETURN
2120 REM FT.CALC.
2130 LET Y=A5
2140 LET X=X-5-VAL "120"
2150 IF X<=D1 THEN RETURN
2160 LET X=X-D1
2170 LET Y=A6
2180 RETURN
2190 REM SR TAB.UPDATE
2200 LET X$=STR$ X
2210 LET Y$=STR$ Y
2220 IF LEN X$<A2 THEN LET X$="0
"+X$
2230 IF LEN Y$<A2 THEN LET Y$="0
"+Y$
2240 LET K$(N, TO A4)=X$+Y$
2250 RETURN
2260 REM SR F.TAGE
2270 GOSUB 1920
2280 LET K$(A1,B7 TO C0)=STR$ J
2290 REM OSTERN
2300 GOSUB 1720
2310 LET N=A5
2320 LET X=0
2330 LET Y=P
2340 GOSUB 2200
2350 LET X=0+A1
2360 LET Y=P
2370 IF X<=D1 THEN GOTO 2400
2380 LET X=A1
2390 LET Y=A4
2400 LET N=A6
2410 GOSUB 2200
2420 REM KARFREITAG
2430 LET X=0-A2
2440 LET Y=P
2450 IF X>A0 THEN GOTO 2480
2460 LET X=D1+X
2470 LET Y=A3
2480 LET N=A4
2490 GOSUB 2200
2500 REM ABS.TAG F.OSTERN
2510 GOSUB 1310
2520 LET R=D1+C8+S+O
2530 IF P=A4 THEN LET R=R+D1
2540 REM ROSENMONTAG
2550 LET X=R-D0-B8
2560 LET Y=A2
2570 LET X=X-D1
2580 IF X<=C8+S THEN GOTO 2610
2590 LET Y=A3
2600 LET X=X-C8-S
2610 LET N=A3
2620 GOSUB 2200
2630 REM HIMMELFAHRT
2640 LET X=R+D0+A9
2650 GOSUB 2130
2660 LET N=A6
2670 GOSUB 2200
2680 REM PFINGSTEN
2690 LET X=R+D0+B9
2700 GOSUB 2130
2710 LET N=A9
2720 GOSUB 2200
2730 LET X=X+A1
2740 IF X<=D1 THEN GOTO 2770
2750 LET X=A1
2760 LET Y=A6
2770 LET N=B0

```



```

2780 GOSUB 2200
2790 REM FRONLEICHNAM
2800 LET X=R+D0+D0
2810 GOSUB 2130
2820 LET N=B1
2830 GOSUB 2200
2840 REM U.TAGE I.TAB.
2850 FOR K=A1 TO B8
2860 IF K$(K,A3)<>" " THEN GOTO
2910
2870 LET M=VAL K$(K,A3 TO A4)
2880 LET D=VAL K$(K, TO A2)
2890 GOSUB 1380
2900 LET K$(K,A5)=STR$ X
2910 NEXT K
2920 REM BUSSTAG
2930 LET X=C3-X
2940 LET Y=B1
2950 LET N=B4
2960 GOSUB 2200
2970 REM 1.MAI
2980 IF J>=1889 THEN GOTO 3010
2990 LET K$(A7)=L$( TO C0)
3000 REM 17.JUNI
3010 IF J>=1954 THEN RETURN
3020 LET K$(B2)=CHR$ A0
3030 RETURN
3040 REM SR KALP
3050 FOR Z=X TO A7
3060 LET N=N+A1
3070 IF J=1582 AND M=B0 AND N<B5
THEN GOTO 3120
3080 LET N$=STR$ N
3090 IF LEN N$<A2 THEN LET N$=CH
R$ A0+N$
3100 PRINT AT Y, (Z-A1)*A4+A3;N$
3110 IF N=U THEN RETURN
3120 NEXT Z
3130 RETURN
3140 REM SR KALENDER
3150 CLS
3160 FOR I=A2 TO C9
3170 PRINT AT A0,I;CHR$ A3;AT A2
,I;CHR$ VAL "131";AT B9,I;CHR$ U
AL "131"
3180 NEXT I
3190 FOR I=A1 TO B8
3200 PRINT AT I,A1;CHR$ A5;TAB D
0;CHR$ VAL "133"
3210 NEXT I
3220 PRINT AT A0,A1;CHR$ A7;TAB
D0;CHR$ VAL "132";AT A2,A1;CHR$
VAL "130";TAB D0;CHR$ VAL "129";
AT B9,A1;CHR$ VAL "130";TAB D0;C
HR$ VAL "129";AT A4,A3;"SO MO
DI MI DO FR SA";AT A1,A3;M$(
K);TAB C5;J
3230 FOR I=A2 TO C9
3240 PRINT AT A5,I;CHR$ A9
3250 NEXT I
3260 LET U=T(K)
3270 IF K=A2 THEN LET U=U+S
3280 LET N=A0
3290 LET Y=A7
3300 GOSUB 3050
3310 FOR Y=A9 TO B7 STEP A2
3320 LET X=A1
3330 GOSUB 3050
3340 IF N=U THEN RETURN
3350 NEXT Y
3360 RETURN
3370 REM SR JAHRESEING.U.PRFG.
3380 PRINT AT C0,A0;A$;"JAHR";B$
(A6 TO );AT C1,A1;"JJJJ"
3390 SLOW
3400 INPUT D$
3410 FAST
3420 IF D$="X" THEN RETURN
3430 IF D$="" THEN LET D$=H$(A7
TO )
3440 FOR I=A1 TO LEN D$
3450 IF CODE D$(I)<B8 OR CODE D$
(I)>D2+A5 THEN GOTO 3480
3460 NEXT I
3470 GOTO 3510
3480 GOSUB 1580
3490 PRINT AT B8,A0;E$
3500 GOTO 3380
3510 LET J=VAL D$
3520 IF J>=VAL "1583" AND J<=VAL
"2399" THEN RETURN
3530 GOSUB 1580
3540 PRINT AT B8,A0;J;"<-";J$;;A
T C1,A0;A$;G$;B$(A6 TO )
3550 LET I$=""
3560 SLOW
3570 INPUT I$
3580 FAST
3590 IF I$="" THEN RETURN
3600 GOTO 3560
3610 REM SR FT.AUSG.
3620 CLS
3630 PRINT TAB A8;"FEIERTAGE "
;K$(A1,B7 TO );
3640 PRINT K$(A1, TO A2);".";K$(
A1,A3 TO A4);".";K$(A1,B7 TO );"
;"T$(VAL K$(A1,A5), TO A2);TAB
B7;K$(A1,A6 TO B6)
3650 FOR I=A2 TO B8
3660 IF K$(I,A1)=CHR$ A0 THEN GO
TO 3680
3670 PRINT K$(I, TO A2);".";K$(I
,A3 TO A4);".";K$(A1,B7 TO );"
;"T$(VAL K$(I,A5), TO A2);TAB B7
;K$(I,A6 TO )
3680 NEXT I
3690 RETURN
3700 REM SR JUMBOZAHL
3710 CLS
3720 LET I$=STR$ J
3730 DIM L$(A8,D2+A1)
3740 FOR I=A1 TO A4
3750 FOR K=A1 TO A8
3760 LET C=PEEK (VAL "7680"+(COD
E I$(I)+A1)*A8-A9+K)
3770 FOR L=A8*I+A1 TO A8*I-A7 ST
EP -A1
3780 LET L$(K,L)=CHR$ ((C-A2*INT
(C/A2))*VAL "128")
3790 LET C=INT (C/A2)
3800 NEXT L
3810 NEXT K
3820 NEXT I
3830 FOR I=A1 TO A8
3840 PRINT AT I+B1,A0;L$(I)
3850 NEXT I
3860 COPY
3870 DIM L$(D2)
3880 RETURN
3890 REM MAINLINE
3900 REM -----
3910 REM HEADER
3920 CLS
3930 PRINT AT A3,A0;L$;TAB A4;"*
";TAB C7;"*";TAB A4;"* RUND UM D
EN KALENDER *";TAB A4;"*";TAB C7
;"*";L$
3940 DIM L$(D2)
3950 REM EINGABE AKT. DATUM
3960 PRINT AT C0,A0;A$;"HEUTIGES
";B$;AT C1,A1;C$
3970 GOSUB 980
3980 IF D$="X" THEN GOTO 5810
3990 GOSUB 1380
4000 GOSUB 1580
4010 PRINT AT B0,A8;"AKTUELLES "
;B$(A1 TO A5);".";AT B2,A3;D$;TA
B B9;T$(X);AT C1,A0;A$;G$;B$(A6
TO )
4020 SLOW
4030 INPUT I$
4040 FAST

```

Basic-Listing »Kalender« (Fortsetzung)

```

4050 IF I$<>"" THEN GOTO 4020
4060 LET H$=D$
4070 LET W$=T$(X)
4080 REM MENUE
4090 CLS
4100 PRINT "PROGRAMMSTART"; CHR$
VAL "146"; CHR$ VAL "146"; CHR$ VA
L "146"; "WOCHENTAG"; "
ABSTAND"; "TAGESNUMMER";
"BESTIMMTER WOCHENTAG";
"FEIERTAGE"; "KALENDER
-DISPLAY"; "KALENDER-DRUCK"
"PROGRAMM BEENDEN"
4110 GOSUB 1490
4120 IF I$="1" THEN GOTO 4220
4130 IF I$="2" THEN GOTO 4340
4140 IF I$="3" THEN GOTO 4560
4150 IF I$="4" THEN GOTO 4770
4160 IF I$="5" THEN GOTO 5150
4170 IF I$="6" THEN GOTO 5260
4180 IF I$="7" THEN GOTO 5590
4190 IF I$="X" THEN GOTO 5810
4200 GOTO 4110
4210 REM WOCHENTAG
4220 CLS
4230 PRINT TAB B0; "WOCHENTAG";
"BESTIMMUNG DES WOCHENTAGES FU
ER EIN VORGEgebenES DATUM:"
4240 PRINT AT C0,A0;A$;B$;AT C1,
A1;C$
4250 GOSUB 980
4260 IF D$="X" THEN GOTO 4090
4270 GOSUB 1380
4280 GOSUB 1580
4290 PRINT AT B0,A3;D$;TAB B9;T$
(X)
4300 GOSUB 1630
4310 IF I$="X" THEN GOTO 4090
4320 IF I$="U" THEN GOTO 4220
4330 REM ABSTAND
4340 CLS
4350 PRINT TAB B1; "ABSTAND";
"BESTIMMUNG DES ABSTANDES"; "ZWI
SCHEN ZWEI EINGEGEBENEN DATENIN
TAGEN:"
4360 PRINT AT C0,A0;A$;"1.";B$;A
T C1,A1;C$
4370 GOSUB 980
4380 IF D$="X" THEN GOTO 4090
4390 GOSUB 1380
4400 PRINT AT B0,A3;D$
4410 LET W=A
4420 GOSUB 1580
4430 PRINT AT C0,A0;A$;"2.";B$;A
T C1,A1;C$
4440 GOSUB 980
4450 IF D$="X" THEN GOTO 4090
4460 GOSUB 1380
4470 PRINT AT B0,B9;D$
4480 LET B=A
4490 GOSUB 1580
4500 PRINT AT B3,A3;"ABSTAND: ";
TAB B9;ABS (W-B);"TAG";
4510 IF ABS (W-B)<>A1 THEN PRINT
"E"
4520 GOSUB 1630
4530 IF I$="X" THEN GOTO 4090
4540 IF I$="U" THEN GOTO 4340
4550 REM TAGESNR.
4560 CLS
4570 PRINT TAB A9; "TAGESNUMMER";
"DER WIEVIELTE TAG DES JAHRE
S"; "IST EIN BESTIMMTES DATUM?"
4580 PRINT AT C0,A0;A$;B$;AT C1,
A1;C$
4590 GOSUB 980
4600 IF D$="X" THEN GOTO 4090
4610 GOSUB 1380
4620 GOSUB 1580
4630 PRINT AT B0,A6;"DER";TAB B5
;D$

```

```

4640 LET W=A
4650 LET J=VAL D$(A7 TO )-A1
4660 LET M=B2
4670 LET D=D1
4680 GOSUB 1160
4690 GOSUB 1380
4700 GOSUB 1580
4710 PRINT AT B3,A0;"IST DER ";A
B$ (W-A);". TAG DES JAHRES ";D$(
A7 TO )
4720 IF VAL (D$(A7 TO )+D$(A4 TO
A5)+D$(A1 TO A2))<VAL (H$(A7 TO
)+H$(A4 TO A5)+H$(A1 TO A2)) TH
EN PRINT AT B3,A0;"WAR"
4730 GOSUB 1630
4740 IF I$="X" THEN GOTO 4090
4750 IF I$="U" THEN GOTO 4560
4760 REM BEST. W.TAG
4770 CLS
4780 PRINT TAB A5; "BESTIMMTER W
OCHENTAG"; "WANN FAEHLT EIN DA
TUM IN DEN"; "FOLGENDEN JAHREN AU
F EINEN"; "BESTIMMTE Wochentag ?
"; " (Ausgabe von 20 zutreffende
N"; "DATEN)"
4790 PRINT AT C1,A0;A$;F$;B$(A6
TO );"
4800 SLOW
4810 INPUT I$
4820 FAST
4830 IF I$="X" THEN GOTO 4090
4840 IF I$="" THEN LET I$=W$
4850 FOR I=A1 TO A7
4860 IF I$=T$(I, TO LEN I$) THEN
GOTO 4900
4870 NEXT I
4880 PRINT AT B6,A0;L$;AT B8,A0;
E$
4890 GOTO 4800
4900 PRINT AT B0,A3;F$;";";TAB B
9;I$
4910 GOSUB 1580
4920 PRINT AT C0,A0;A$;"ANFANGS"
;B$;AT C1,A1;C$
4930 GOSUB 980
4940 IF D$="X" THEN GOTO 4090
4950 CLS
4960 PRINT I$;
4970 LET B=A0
4980 GOSUB 1380
4990 LET W=VAL D$(A7 TO )
5000 IF I$=T$(X, TO LEN I$) THEN
GOTO 5070
5010 LET W=W+A1
5020 IF W>VAL "9999" THEN GOTO 5
110
5030 LET D$(A7 TO )=STR$ W
5040 GOSUB 1100
5050 GOSUB 1380
5060 IF I$<>T$(X, TO LEN I$) THE
N GOTO 5010
5070 PRINT TAB B2;D$
5080 LET B=B+A1
5090 IF B<C0 THEN GOTO 5010
5100 PRINT TAB B5;"ENDE"
5110 GOSUB 1630
5120 IF I$="X" THEN GOTO 4090
5130 IF I$="U" THEN GOTO 4770
5140 REM F.TAGE
5150 GOSUB 1920
5160 CLS
5170 PRINT TAB B0; "FEIERTAGE";
"BESTIMMUNG DER FEIERTAGE FUER
"; "EIN VORGEgebenES JAHR IM ZEITRA
UM 1583 - 2399"
5180 GOSUB 3380
5190 IF D$="X" OR I$="" THEN GOT
O 4090
5200 GOSUB 2270
5210 GOSUB 3620
5220 GOSUB 1630

```

```

5230 IF I$="X" THEN GOTO 4090
5240 IF I$="U" THEN GOTO 5150
5250 REM KAL.DISPLAY
5260 CLS
5270 PRINT TAB A7;"KALENDER";CHR$
R$ VAL "150";"GRÜß" " AUSDRU
CK EINES JAHRESKALENDERS";TAB
6;"(ZEITRAUM 1583-2399)";AT B0,A
0;A$;"DRUCKER BEREIT MACHEN..."
5610 GOSUB 3380
5620 IF D$="X" OR I$="" THEN GOT
O 4090
5630 GOSUB 3710
5640 LET D=A1
5650 FOR L=A1 TO B2
5660 LET M=L
5670 LET K=L
5680 GOSUB 1380
5690 GOSUB 3150
5700 COPY
5710 NEXT L
5720 LET D$=" "+STR$ J
5730 GOSUB 1920
5740 GOSUB 2270
5750 GOSUB 3620
5760 COPY
5770 CLS
5780 COPY
5800 GOTO 4090
5800 REM PGM-ENDE
5810 CLS
5820 PRINT AT B0,B1;"PGM";CHR$
VAL "150";"ENDE"
5830 STOP
5840 REM SAVE
5850 CLEAR
5860 CLS
5870 SAVE "KALENDE"
5880 RUN

```

Basic-Listing »Kalender« (Schluß)

VARIABLENLISTE

A	HILFSVARIABLE
A0...A9	ZAHLENWERTE 0...9
A#	FESTTEXT
B	HILFSVARIABLE
B0...B9	ZAHLENWERTE 10...19
C	TABELLENWERT AUS OSTERDATUMSBERECHNUNG
C0...C9	ZAHLENWERTE 20...29
C#	FESTTEXT
D	TAG
D0...D2	ZAHLENWERTE 30...32
D#	EINGABEVARIABLE FUER DATUM
E	TABELLENWERT AUS OSTERDATUMSBERECHNUNG
E#	FESTTEXT
F	WERT AUS OSTERFORMEL
F#	FESTTEXT
G	WERT AUS OSTERFORMEL
G#	FESTTEXT
H	WERT AUS OSTERFORMEL
H1	ZAHLENWERT 100
H4	ZAHLENWERT 400
H#	AKTUELLES DATUM TT.MM.JJJJ
I	SCHLEIFENLAUFVARIABLE
I#	EINGABEVARIABLE, VORZUGSWEISE AUS "INKEYS"
J	JAHR
J#	FESTTEXT
K	SCHLEIFENLAUFVARIABLE
K#K)	FEIERTAGSTABELLE
	SP. 1- 4 DATUM TT.MM
	SP. 5 WOCHENTAG (1=SO,7=SA)
	SP. 6-20 BEZEICHNUNG
L	SCHLEIFENLAUFVARIABLE
L#<	HILFSSTRING, MEIST ALS LEERZEILE
M	MONAT
M#<	TABELLE DER MONATSNAMEN
O	ERRECHNETER OSTERTAG
O#<	KONSTANTENTABELLE F. OSTERDATUMSBERECHNUNG
	SP.1-2 VON JAHR (VERSCHL., NUR JAHRH.)
	SP.3-4 BIS JAHR (VERSCHL., NUR JAHRH.)
	SP.5 KONSTANTE (VERSCHL.,NUR EINER)
	SP.6 KONSTANTE (VERSCHL.,NUR EINER)
	SP.7 KONSTANTE
P	ERRECHNETER OSTERMONAT
Q	VARIABLE AUS OSTERFORMEL
R	OSTERSONNTAG ALS ABSOL. TAGESNR. IM JAHR
S	SCHALTJAHRINDIKATOR (1=JA/0=NEIN)
T<	TABELLE DER TAGESANZAHL IM MONAT
TV	BILDSCHIRMPÖINTER (LINKS UNTEN)
T#<	TABELLE DER WOCHENTAGSNAMEN
U	HILFSVARIABLE
V	HILFSVARIABLE
W	HILFSVARIABLE
W#	AKTUELLER WOCHENTAG
X	HILFSVARIABLE
Y	HILFSVARIABLE
Z	HILFSVARIABLE

ANM.: DIE "HILFSVARIABLEN" HABEN JE NACH PGM-TEIL VERSCHIEDENE BEDEUTUNGEN

# Overload — Lade- und Speicherrou- tine für Pro- gramme, die über RAMTOP liegen

Daten und Z80-Programme oberhalb von RAMTOP sind dem Zugriff des Betriebssystems entzogen, also gegen eine Verschiebung im Speicher und den NEW-Befehl gesichert. Außerdem ist die Lösung, ein Maschinencode-Programm über RAMTOP zu legen, nach unseren Erfahrungen praktischer als die Möglichkeit, Programme in überlangen REM-Zeilen unterzubringen, wo die Wahrscheinlichkeit gegeben ist, daß das System bei unvorsichtigem LISTen abstürzt.

Dieses Programm gleicht einen Fehler des ZX81-Betriebssystems aus. Oberhalb von RAMTOP gelagerte Programme lassen sich nämlich leider nicht durch den SAVE-Befehl des ROM abspeichern. Dieses kleine Maschinenprogramm, das auf einem ZX81 mit mindestens 16 KByte läuft, erlaubt die Abspeicherung. Um es aber dazu zu bringen, ist noch eine kleine Vorarbeit nötig:

POKE 16388,188 ; POKE 16389,127 ; NEW ; POKE 32700,0 ; POKE 32701,1 ; POKE 32702,255 ; POKE 32703,255 ; POKE 32704,201.

Nachdem dies geschehen ist, muß jetzt eine REM-Zeile mit 455 Byte Länge erzeugt werden. Nun wird der Hexcode unter Verwendung des HEXLOADER eingetippt. Dieser fragt jeweils 8 Byte und nach NEWLINE die dazugehörige Prüfsumme ab. Stimmen die Bytes und die Prüfsumme nicht überein, so muß die Zeile neu eingegeben werden.

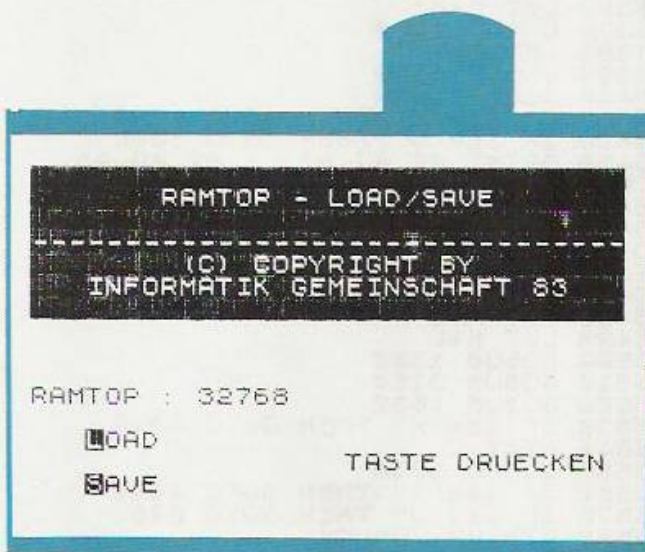
Ist nun das Programm vollständig in die REM-Zeile eingegeben, empfiehlt es sich, das Programm erst einmal abzuspeichern. Der Hexloader kann nun gelöscht und der Basicteil des »Ramtop-LOAD/SAVE«-Programmes eingetippt werden. Daß das jetzt vollständige Programm mehrmals mit RUN 310 abgesAVet wird, versteht sich von selbst.

Nach dem Autostart meldet sich das Programm mit dem im Bild dargestellten Display. Die Funktionsweise des Programmes wollen wir an einem Beispiel erläutern:

Die am Anfang durchgeführten POKE-Befehle haben bewirkt, daß RAMTOP verkleinert und ein Maschinencode-Programm oberhalb dieser Adresse abgelegt wurde. Bei PRINT USR 32700 erscheint auf dem Bildschirm die Zahl 65535.

Wenn Sie dieses Programm jetzt speichern wollen, so starten Sie das »RAMTOP-LOAD/SAVE« mit RUN. Wählen Sie mit »S« den SAVE-Modus. Bei der Abfrage wegen Änderung der Systemvariablen RAMTOP geben Sie »N« für Nein ein, und das Programm beginnt nach nochmaligem Tastendruck, vom aktuellen RAMTOP bis zur Adresse 32768, das Demoprogramm auf Kassette zu speichern. Nach Beendigung dieses Vorganges meldet sich der Rechner mit R/300, wenn alles richtig gelaufen ist.

Nun versuchen wir, das Programm wieder einzuladen. Nach dem Start mit RUN drücken Sie »L« und ändern RAMTOP wie-



Das Programm meldet sich

startbereit

```

16514  B7A6B269B4B596B1 => 1400
16522  B4A6A988B8A6BBAA => 1374
16530  000000000000000B => 59
16538  3433002C2A372631 => 331
16546  29003037263A382A => 338
16554  003A1B00373A2934 => 291
16562  0037342A322A3700 => 296
16570  0000000000000022 => 34
16578  20231F002C2A292A => 267
16586  3733001D7676FF7F => 753
16594  FF7F11CB12CD460F => 910
16602  D2A60310FE1B7A63 => 977
16610  20F32AD040CDEF40 => 1097
16618  CD124118F85E37C6 => 912
16626  13C89FE605C6044F => 894
16634  D3FF062310FEC046 => 1052
16642  0FD2A603061E10FE => 700
16650  0D20EDA710FD18DF => 965
16658  0023E62AD24037ED => 878
16666  52EB00E1213B40CB => 1109
16674  F6CD0702CF1AC900 => 894
16682  00000000000000DE7 => 436
16690  02AF322140CD3C41 => 654
16698  18FB0E0106003E7F => 485
16706  DBFE03FF1FD2A603 => 1349
16714  1717381E10F0F1FD => 882
16722  CB2146C21842CD3C => 855
16730  41FDCB21C62AD040 => 1066
    
```

Das Hexadezimal-Listing des Maschinencode-Programms

# RAMTOP

**RAMTOP-LOAD/SAVE**  
VON GERALD KRAUSE U. RUDO ROEMER  
6473 GEDERN 1

```

10 SLOW
20 LET A$="AKTUELLER WERT"
30 LET RT=PEEK 16388+256*PEEK
16388
40 RAND USR 16857
50 PRINT AT 10,0;"RAMTOP : ";R
T
60 PRINT AT 13,17;"TASTE DRUECKEN";AT 12,3;"LOAD";AT 14,3;"SAVE"
70 LET I$=INKEY$
80 IF I$<>"S" AND I$<>"L" THEN GOTO 55
90 PRINT AT 13,17;" ";AT 12,11;(A$ AND I$="L");AT 14,11;(A$ AND I$="S")
100 LET RUN=16596
110 IF I$="L" THEN LET RUN=16688
S
120 IF INKEY$<>" " THEN GOTO 120
130 PRINT AT 17,0;"SOLL RAMTOP GERENDERT WERDEN ? J/N"
140 IF INKEY$="" THEN GOTO 140
150 IF INKEY$="N" THEN GOTO 220
160 PRINT "AUF WELCHEN WERT ?"
170 INPUT ADR
180 RAND ADR
190 LET RT=ADR
200 LET I=USR 16935
210 PRINT "NEUER WERT VON RAMTOP : ";RT
220 IF I$="S" THEN LET RT=RT-1
230 LET A=INT (RT/256)
240 POKE 16592,RT-A*256
250 POKE 16593,A
260 PRINT "BEREIT FUER START DER OPERATION"
270 IF INKEY$<>" " THEN GOTO 270
280 IF INKEY$="" THEN GOTO 280
290 FAST
300 RAND USR RUN
310 SAVE "RAMTOP"
320 RUN

```

## Basic-Hilfsprogramm

```

16738 50CD3C4171CD1241 => 811
16746 18F8D51E94061A10 => 722
16754 DBFE17C67B7B38F5 => 1246
16762 10F5D12004FE5630 => 894
16770 B63FCB113066C97A => 1023
16778 A726C4C3164267A6 => 1037
16786 B26964B5309680B1 => 1307
16794 B4A8A996B8A66BAA => 1374
16802 90A89180A88465BE => 1304
16810 B7A8ACAD6980A7BE => 1372
16818 8080808080808080 => 1024
16826 808080AEB3ABB467 => 1271
16834 B2A689AEB080ACAA => 1349
16842 B2AAAE63B8A8ADA6 => 1392
16850 ABB930A49F0000CD => 1012
16858 2A0A11E0003E80D7 => 698
16866 1B7A6320F8010701 => 617
16874 CDF5082190411612 => 740
16882 7ED7231520FA0100 => 680
16890 03CD750805203E96 => 711
16898 D710FD010504CDF5 => 947
16906 0821A24106357ED7 => 668
16914 2310FB030000213B => 595
16922 40CBF6C00702CF0E => 948
16930 C900000000002A32 => 890
16938 40220440F901003E => 478
16946 C5017505C5ED7302 => 873
16954 40CFFF0000000000 => 526

```

derum nicht. Es wird nun von RAMTOP bis zur Adresse 32768 das Programm von der Kassette eingeladen. Tritt ein Ladefehler auf, so meldet der ZX dies mit F/300, andernfalls erscheint R/300.

Bei diesem Beispiel haben wir von einer Änderung der Systemvariablen RAMTOP abgesehen. Diese ist aber besonders wichtig, wenn wir ein Maschinenprogramm von 1000 Byte Länge laden wollen, uns aber nicht so viele Bytes oberhalb von RAMTOP zur Verfügung stehen. Der neue RAMTOP-Wert ergibt sich dann aus der Rechnung:  $32768 - 1000 = 31768$ . Da die Berechnung aber schwierig ist und wir auch nicht von jedem Maschinenprogramm die Länge kennen, empfiehlt es sich, bei jedem AbsAVen den aktuellen RAMTOP-Wert zu notieren, so daß man ihn beim Laden sofort zur Verfügung hat. Damit ergeben sich auch keine Schwierigkeiten, wie etwa eine Differenz zwischen gefordertem und vorhandenem Speicherplatz. Natürlich ist BREAK während eines LOAD- oder SAVE-Vorganges jederzeit möglich. Im einzelnen gliedert sich das Maschinenprogramm in folgende Abschnitte:

16592-16595	Daten: Anfangs- und Endadresse
16596-16680	SAVE-Routine
16688-16783	LOAD-Routine
16784-16854	Daten
16855-16917	PRINT-Routine
16918-16930	Fehlerabbruch-Routine
16931-16956	RAMTOP-Änderung

(R. Römer)

# Software-Toolkit

**Das Programm »Toolkit« für den ZX81 ist eine in Maschinensprache geschriebene Softwareerweiterung, die hauptsächlich einer komfortableren Programmerstellung und Editierung bei der 16-KByte-Version dient.**

Die Erweiterung belegt 2816 Byte, wobei 508 Byte das eigentliche Programm ausmachen und der Rest als Workspace, hauptsächlich für die RENUM-Routine, dient. Das Toolkit wird oberhalb von »RAMTOP« abgelegt und ist deshalb vor versehentlichem Löschen mit »NEW« gesichert. Die Erweiterung bietet folgende fünf zusätzliche Direkt-Mode-Befehle:

1. RENUM, aufgerufen durch:

RAND USR 29952 ohne Argumente

numeriert ein Basicprogramm neu, beginnend mit der Zeilennummer 10 in Zehnerschritten. Auf eine Wahlmöglichkeit der Anfangsnummer oder des Inkrementes wurde verzichtet, um das Programm nicht unnötig zu vergrößern. Dafür werden aber alle GOTOs und GOSUBs den neuen Zeilennummern angepaßt, das heißt ebenfalls umnummeriert, so daß das Programm mit den neuen Nummern sofort lauffähig ist. Bei sehr langen neu zu numerierenden Programmen sollte vor der Anwendung des Befehls der FAST-Modus eingeschaltet werden, da dadurch der Vorgang erheblich beschleunigt wird (zirka 2 Sekunden für ein Programm mit 250 Zeilen).

2. DELETE, aufgerufen durch:

FAST

PRINT USR 30259,zeile1,zeile2

löscht die Zeilen eines Programmes ab Zeile 1 bis Zeile 2-1. Dazu ein Beispiel: FAST N/L PRINT USR 30259,80,350 löscht die Zeilen 80 bis 340 beziehungsweise die letzte Zeile vor der Zeile 350. Vor diesem Befehl muß der FAST-Modus eingeschaltet werden, da sonst in den meisten Fällen der ZX81 nur noch durch Aus- und Einschalten wieder in den Normalzustand zu bringen ist.

3. FIND, aufgerufen durch:

PRINT USR 30374,"Text"

sucht in einem Basicprogramm die Zeichenfolge »Text« und listet alle Zeilen, die diese Folge enthalten. Sie darf aus beliebigen Zeichen bestehen, auch aus Grafikzeichen und den Befehlswörtern, die sich trotz L-Cursor mit SHIFT eingeben lassen (zum Beispiel THEN, AND, TO und so weiter).

Damit können zum Beispiel durch den Befehl:

PRINT USR 30374,"A="

alle Zeilen gelistet werden, in denen der Variablen A ein Wert zugewiesen wird. Da sich dieser Vorgang mit allen Variablen durchführen läßt, kann man damit feststellen, ob, wie oft und

welche Variablen bereits verwendet wurden. Weiterhin lassen sich zum Beispiel mit dem Befehl:

PRINT USR 30374,"THEN"

alle Zeilen listen, in denen der Befehl THEN vorkommt, oder mit:

PRINT USR 30374,"TO"

werden alle FOR-NEXT-Schleifen aufgelistet.

4. HEXDEZ, aufgerufen durch:

PRINT USR 30300,"Hexzahl"

ergibt den dezimalen Wert der »Hexzahl« im Bereich 0000...FFFF. Beispiel:

PRINT USR 30300,"3FFF"

ergibt 32767.

5. DEZHEX, aufgerufen durch:

PRINT USR 30338,Zahl (oder Variable)

ergibt den hexadezimalen Wert der Zahl im Bereich

0.....65535. Dazu ein Beispiel:

PRINT USR 30338,32768

ergibt 8000, wobei hier im Gegensatz zur HEXDEZ-Routine beide Werte, also der hexadezimale und der dezimale auf dem Schirm erscheinen.

Bei den Befehlen FIND und HEXDEZ dürfen als Argumente nach der USR-Adresse nur Zahlen und keine Variablen verwendet werden. Allerdings kann bei allen fünf Befehlen die Adresse zur besseren Verständlichkeit als Variable benutzt werden. Beispiel:

LET RENUM=29952

RAND USR RENUM

entspricht dem Befehl RAND USR 29952, ist aber bedeutend übersichtlicher.

Nun zur Eingabe des Toolkits in den Speicher des ZX81. Dazu dient ein Hilfsprogramm »Toolhilf«, welches in Listing 2 abgedruckt ist. Bevor das Hilfsprogramm eingetippt wird, muß noch RAMTOP durch: POKE 16389,117 und NEW auf die Adresse 29952 herabgesetzt werden. Jetzt kann mit der Eingabe des Hilfsprogrammes begonnen werden. Ist das Programm eingegeben, wird es mit »RUN« gestartet und die 509 Byte des Toolkits aus Listing 1 können in 16 Byte-Abschnitten, nach denen jeweils nach der entsprechenden Prüfsumme gefragt wird, eingegeben werden. Bei nicht korrekter Prüfsumme verlangt das Programm automatisch die letzten 16 Byte noch einmal. Ist die Eingabe der Hex-Daten abgeschlossen, zeigt das Programm selbständig die noch zu unternehmenden Schritte, die zur Aufzeichnung der Daten auf Band notwendig sind, auf dem Bildschirm an.

Das Programm »Tollkit« ist somit zur ständigen Benutzung auf Kassette aufgezeichnet.

Vor jedem Laden des Toolkits muß mit POKE 16389,117 und NEW die Speicherobergrenze auf 29952 herabgesetzt werden. Danach kann das Programm, das sich selbst initialisiert und durch ein abschließendes NEW den Speicher für Basic-Programme wieder freigibt, geladen werden. (J. Dietl)

Adresse	HEX-Daten	Pruefsumme
29952	21 7D 40 11 0B 77 01 00 00 7E FE 76 28 0E 05 ED	1356
29968	A0 ED A0 4E 23 46 23 09 01 03 18 ED 21 0A 00 22	1318
29984	FD 76 ED 43 01 77 21 7D 40 7E ED 4B 01 77 ED 43	1879
30000	FF 76 FE 76 C8 ED 4B FD 76 70 23 71 EB 21 0A 00	2166
30016	09 22 FD 76 EB 23 22 03 77 23 23 7E FE EC 28 0B	1577
30032	FE ED 28 07 FE 76 20 F2 23 18 0E 23 E5 22 16 40	1833
30048	DF CD D9 14 CD 8A 15 21 0B 77 11 00 00 78 BE 20	1551
30064	06 23 79 BE 2B 28 15 23 23 13 05 ED 4B FF 76 0B	1438
30080	ED 43 FF 76 78 B1 01 20 E4 E1 CF 2C 13 62 6B 29	2168
30096	29 19 29 AF 32 FC 76 CD FB 75 E1 E5 11 05 77 1A	1896
30112	FE 76 28 20 47 7E FE 7E 20 15 05 05 CD 9B 09 ED	2090
30128	5B 03 77 1A C6 01 12 13 1A 0E 00 12 23 C1 D1 70	1274
30144	23 13 18 DB 7E FE 7E 28 14 E5 ED 5B 03 77 1A D6	1782
30160	01 12 13 1A DE 00 12 CD 5C 05 E1 18 E7 23 EB E1	1581
30176	D5 22 16 40 DF CD D9 14 E1 01 05 00 ED 5B 1C 40	1649
30192	EB B7 ED 42 ED B0 EB 23 C3 29 75 01 05 77 11 E8	2131
30208	03 CD 19 76 11 64 00 CD 19 76 11 0A 00 CD 19 76	1191
30224	7D C6 1C 02 03 3E 76 02 C9 AF ED 52 3C 30 FB 19	1617
30240	C6 1B FE 1C 20 07 5F 3A FC 76 B7 7B C8 02 32 FC	1879
30256	76 03 C9 E7 CD 92 0D CD 8A 15 2A 16 40 7E CD B4	1920
30272	07 23 22 16 40 60 69 CD D8 09 E5 E7 CD 92 0D CD	1822
30288	8A 15 60 69 CD D8 09 D1 CD 5D 0A C9 E7 ED 5B 16	2089
30304	40 1A FE 0B C2 8A 75 21 00 00 06 04 13 1A FE 0B	1157
30320	28 0D D6 1C DA 8A 75 29 29 29 29 85 6F 10 ED 44	1497
30336	4D C9 E7 CD 92 0D CD 8A 15 78 CD 9C 76 78 CD A0	2321
30352	76 79 CD 9C 76 78 CD A0 76 AF D7 C9 0F 0F 0F 0F	1972
30368	E6 0F C6 1C D7 C9 2A 16 40 23 22 01 77 7E FE 0B	1595
30384	C2 8A 75 06 00 23 7E 04 FE 0B 20 F9 05 21 7D 40	1393
30400	E5 7E FE 76 20 05 E1 22 16 40 C9 23 23 23 23 ED	1687
30416	5B 01 77 13 23 E5 D5 C5 1A BE 20 04 13 23 10 F8	1474
30432	C1 D1 E1 28 09 7E FE 76 20 EA D1 23 18 D2 E1 C5	2340
30448	CD A5 0A 23 23 23 CD 7F 07 C1 18 C4 00 00 00 00	1237

Listing 1. HEX-Daten zu ZX81-Toolkit

Startadresse	: 29952	Hex	: 7500	
Endadresse	: 30460	Hex	: 76FC	
Anzahl Bytes	: 508	Hex	: 01FC	Programm ohne Daten
Anzahl gesamt	: 2571	Hex	: 0A0B	Speicherbedarf gesamt

Tabelle. Speicherbelegung

URENUM: 7500H	UC1: 76D8H	UDEL: 7633H	UDT0H: 7682H
UERR: 758AH	UF11: 76B5H	UF12: 76C0H	UF13: 76CBH
UF15: 76D4H	UF16: 76EEH	UFIND: 76A6H	UFND: 755BH
UHD1: 766CH	UHD2: 767FH	UHEXDEZ: 75FBH	UHT0D: 765CH
UJL33: 756DH	ULP1: 7509H	ULP2: 7529H	ULP3: 754AH
ULP4: 7577H	ULP5: 758CH	UNEW11: 75BFH	UNEW1: 759FH
UNEW2: 75C4H	UNEW3: 75DDH	UPH1: 76A0H	UPH: 769CH
UREN: 751CH	US1: 761AH	US2: 762DH	USUB: 7619H
VANZ1: 7701H	VANZNUM: 76FFH	VFLAG: 76FCH	VLEN: 7703H
VLINNUM: 76FDH	VNUM: 7705H	VNUMTBL: 770BH	

Startadressen der Routinen

## Listing 2. Hilfsprogramm zum Laden und Starten des Maschinencodeprogramms »Toolkit«

```

1 REM *****
2 REM *      TOOLHILF      *
3 REM * JOACHIM DIETL *
4 REM *   DAMMSTR. 3   *
5 REM * 7100 HEILBRONN *
6 REM *****
10 DIM A$(510)
20 LET J=1
30 CLS
40 LET CSUM=0
50 FOR I=1 TO 16
60 PRINT AT I,0;29951+J
70 PRINT AT 21,0;"BYTE-NR:";J
80 INPUT H$
90 IF LEN H$<>2 THEN GOTO 100
100 LET A=(CODE H$(1)-28)*16+CODE H$(2)-28
110 POKE 29951+J,A
120 LET A$(J)=CHR$ A
130 PRINT AT I,7;H$
140 LET CSUM=CSUM+A
150 LET J=J+1
160 IF J=509 THEN GOTO 240
170 NEXT I
180 GOSUB 360
190 IF I$="J" THEN GOTO 30
200 LET J=J-16
210 CLS
220 PRINT AT 0,0;"DIE LETZTEN 16 BYTES WIEDERHOLEN"
230 GOTO 40
240 GOSUB 360
250 IF I$="J" THEN GOTO 280
260 LET J=J-12
270 GOTO 210
280 CLS
290 PRINT "NUN FOLGENDE BEFEHLE IM"
300 PRINT "DIREKTMODE AUSFUEHREN."
310 PRINT AT 5,3;"1. FAST"
320 PRINT AT 7,3;"2. PRINT USR 30259,1,390"
330 PRINT AT 9,3;"3. BANDGERAET AUF          AUFNAHME SCHALTEN"
340 PRINT AT 12,3;"4. GOTO 390"
350 STOP
360 PRINT AT I,0;"CHECKSUM:";CSUM;" OK ?(J/N)"
370 INPUT I$
380 RETURN
390 SAVE "TOOLKIT"
400 FAST
410 FOR I=1 TO 509
420 POKE 29951+I,CODE A$(I)
430 NEXT I
440 NEW

```



29952	7500	URENUM:	LD HL,VBAS	30092	758C	ULP5:	INC DE
29955	7503		LD DE,VNUMTBL	30093	758D		LD H,D
29958	7506		LD BC,0	30094	758E		LD L,E
29961	7509	ULP1:	LD A,(HL)	30095	758F		ADD HL,HL
29962	750A		CP 76H	30096	7590		ADD HL,HL
29964	750C		JR Z,UREN	30097	7591		ADD HL,DE
29966	750E		PUSH BC	30098	7592		ADD HL,HL
29967	750F		LDI	30099	7593		XOR A
29969	7511		LDI	30100	7594		LD (VFLAG),A
29971	7513		LD C,(HL)	30103	7597		CALL UHEXDEZ
29972	7514		INC HL	30106	759A		POP HL
29973	7515		LD B,(HL)	30107	759B		PUSH HL
29974	7516		INC HL	30108	759C		LD DE,VNUM
29975	7517		ADD HL,BC	30111	759F	UNEW1:	LD A,(DE)
29976	7518		POP BC	30112	75A0		CP 76H
29977	7519		INC BC	30114	75A2		JR Z,UNEW2
29978	751A		JR ULP1	30116	75A4		LD B,A
29980	751C	UREN:	LD HL,10	30117	75A5		LD A,(HL)
29983	751F		LD (VLINUM),HL	30118	75A6		CP 7EH
29986	7522		LD (VANZ1),BC	30120	75A8		JR NZ,UNEW11
29990	7526		LD HL,VBAS	30122	75AA		PUSH DE
29993	7529	ULP2:	LD A,(HL)	30123	75AB		PUSH BC
29994	752A		LD BC,(VANZ1)	30124	75AC		CALL Z.99BH
29998	752E		LD (VANZNUM),BC	30127	75AF		LD DE,(VLEN)
30002	7532		CP 76H	30131	75B3		LD A,(DE)
30004	7534		RET Z	30132	75B4		ADD A,1
30005	7535		LD BC,(VLINUM)	30134	75B6		LD (DE),A
30009	7539		LD (HL),B	30135	75B7		INC DE
30010	753A		INC HL	30136	75B8		LD A,(DE)
30011	753B		LD (HL),C	30137	75B9		ADC A,0
30012	753C		EX DE,HL	30139	75BB		LD (DE),A
30013	753D		LD HL,10	30140	75BC		INC HL
30016	7540		ADD HL,BC	30141	75BD		POP BC
30017	7541		LD (VLINUM),HL	30142	75BE		POP DE
30020	7544		EX DE,HL	30143	75BF	UNEW11:	LD (HL),B
30021	7545		INC HL	30144	75C0		INC HL
30022	7546		LD (VLEN),HL	30145	75C1		INC DE
30025	7549		INC HL	30146	75C2		JR UNEW1
30026	754A	ULP3:	INC HL	30148	75C4	UNEW2:	LD A,(HL)
30027	754B		LD A,(HL)	30149	75C5		CP 7EH
30028	754C		CP 236	30151	75C7		JR Z,UNEW3
30030	754E		JR Z,UFND	30153	75C9		PUSH HL
30032	7550		CP 237	30154	75CA		LD DE,(VLEN)
30034	7552		JR Z,UFND	30158	75CE		LD A,(DE)
30036	7554		CP 76H	30159	75CF		SUB 1
30038	7556		JR NZ,ULP3	30161	75D1		LD (DE),A
30040	7558		INC HL	30162	75D2		INC DE
30041	7559		JR ULP2	30163	75D3		LD A,(DE)
30043	755B	*****		30164	75D4		SBC A,0
30043	755B	UFND:	INC HL	30166	75D6		LD (DE),A
30044	755C		PUSH HL	30167	75D7		CALL 55CH;LOESCHEN
30045	755D		LD (CHADD),HL	30170	75DA		POP HL
30048	7560		RST 18H	30171	75DB		JR UNEW2
30049	7561		CALL 1409H	30173	75DD	UNEW3:	INC HL
30052	7564		CALL 158AH	30174	75DE		EX DE,HL
30055	7567		LD HL,VNUMTBL	30175	75DF		POP HL
30058	756A		LD DE,0	30176	75E0		PUSH DE
30061	756D	UL33:	LD A,B	30177	75E1		LD (CHADD),HL
30062	756F		CP (HL)	30180	75E4		RST 18H
30063	756F		JR NZ,ULP4	30181	75E5		CALL 1409H
30065	7571		INC HL	30184	75E8		POP HL
30066	7572		LD A,C	30185	75E9		LD BC,5'
30067	7573		CP (HL)	30188	75EC		LD DE,(401CH)
30068	7574		DEC HL	30192	75F0		EX DE,HL
30069	7575		JR Z,ULP5	30193	75F1		OR A
30071	7577	ULP4:	INC HL	30194	75F2		SBC HL,BC
30072	7578		INC HL	30196	75F4		LDIR
30073	7579		INC DE	30198	75F6		EX DE,HL
30074	757A		PUSH BC	30199	75F7		INC HL
30075	757B		LD BC,(VANZNUM)	30200	75F8		JP ULP2
30079	757F		DEC BC	30203	75FB	*****	
30080	7580		LD (VANZNUM),BC	30203	75FB	UREXDEZ:	LD BC,VNUM
30084	7584		LD A,D	30206	75FE		LD DE,1000
30085	7585		OR C	30209	7601		CALL USUB
30086	7586		POP BC	30212	7604		LD DE,100
30087	7587		JR NZ,UL33	30215	7607		CALL USUB
30089	7589		POP HL	30218	760A		LD DE,10
30090	758A	UERR:	RST 8	30221	760D		CALL USUB
30091	758B		15				

Maschinencodeprogramm zu »Toolkit«

30224	7610		LD A,L	30353	7691		LD A,C
30225	7611		ADD A,10H	30354	7692		CALL UPH
30227	7613		LD (BC),A	30357	7695		LD A,C
30228	7614		INC BC	30358	7696		CALL UPH1
30229	7615		LD A,76H	30361	7699		XOR A
30231	7617		LD (BC),A	30362	769A		RST 10H
30232	7618		RET	30363	769B		RET
30233	7619	*****		30364	769C	UPH:	RRCA
30233	7619	USUB:	XOP A	30365	769D		RRCA
30234	761A	US1:	SBC HL,DE	30366	769E		RRCA
30236	761C		INC A	30367	769F		RRCA
30237	761D		JR NC,US1	30368	76A0	UPH1:	AND 0FH
30239	761F		ADD HL,DE	30370	76A2		ADD A,10H
30240	7620		ADD A,10H	30372	76A4		RST 10H
30242	7622		CP 10H	30373	76A5		RET
30244	7624		JR NZ,US2	30374	76A6	*****	
30246	7626		LD E,A	30374	76A6	UF1ND:	LD HL,(4016H)
30247	7627		LD A,(VFLAG)	30377	76A9		INC HL
30250	762A		OR A	30378	76AA		LD (VANZ1),HL
30251	762B		LD A,E	30381	76AD		LD A,(HL)
30252	762C		RET Z	30382	76AE		CP 11
30253	762D	US2:	LD (BC),A	30384	76B0		JP NZ,UERP
30254	762E		LD (VFLAG),A	30387	76B3		LD B,0
30257	7631		INC BC	30389	76B5	UF11:	INC HL
30258	7632		RET	30390	76B6		LD A,(HL)
30259	7633	*****		30391	76B7		INC B
30259	7633	UDEL:	RST 20H	30392	76B8		CP 11
30260	7634		CALL 0D92H	30394	76BA		JR NZ,UF11
30263	7637		CALL 158AH	30396	76BC		DEC B
30266	763A		LD HL,(4016H)	30397	76BD		LD HL,VBAS
30269	763D		LD A,(HL)	30400	76C0	UF12:	PUSH HL
30270	763E		CALL 7B4H; ZAHL UEBERLESEN	30401	76C1		LD A,(HL)
30273	7641		INC HL	30402	76C2		CP 76H
30274	7642		LD (4016H),HL	30404	76C4		JR NZ,UF13
30277	7645		LD H,B	30406	76C6		POP HL
30278	7646		LD L,C	30407	76C7		LD (4016H),HL
30279	7647		CALL 9DBH	30410	76CA		RET
30282	764A		PUSH HL	30411	76CB	UF13:	INC HL
30283	764B		RST 20H	30412	76CC		INC HL
30284	764C		CALL 0D92H	30413	76CD		INC HL
30287	764F		CALL 158AH	30414	76CE		INC HL
30290	7652		LD H,B	30415	76CF		LD DE,(VANZ1)
30291	7653		LD L,C	30419	76D3		INC DE
30292	7654		CALL 9DBH	30420	76D4	UF15:	INC HL
30295	7657		POP DE	30421	76D5		PUSH HL
30296	7658		CALL ASDH	30422	76D6		PUSH DE
30299	765B		RET	30423	76D7		PUSH BC
30300	765C	*****		30424	76D8	UC1:	LD A,(DE)
30300	765C	UHTOD:	RST 20H	30425	76D9		CP (HL)
30301	765D		LD DE,(4016H)	30426	76DA		JR NZ,UCRET
30305	7661		LD A,(DE)	30428	76DC		INC DE
30306	7662		CP 11	30429	76DD		INC HL
30308	7664		JP NZ,UERR; 7589H	30430	76DE		DJNZ UC1
30311	7667		LD HL,0	30432	76E0		POP BC
30314	766A		LD B,4	30433	76E1		POP DE
30316	766C	UND1:	INC DE	30434	76E2		POP HL
30317	766D		LD A,(DE)	30435	76E3		JR Z,UF16
30319	766E		CP 11	30437	76E5		LD A,(HL)
30320	7670		JR Z,UND2	30438	76E6		CP 76H
30322	7672		SUB 10H	30440	76E8		JR NZ,UF15
30324	7674		JP C,UERR; 7589H	30442	76EA		POP DE
30327	7677		ADD HL,HL	30443	76EB		INC HL
30328	7678		ADD HL,HL	30444	76EC		JR UF12
30329	7679		ADD HL,HL	30446	76EE	UF16:	POP HL
30330	767A		ADD HL,HL	30447	76EF		PUSH BC
30331	767B		ADD A,L	30448	76F0		CALL AASH
30332	767C		LD L,A	30451	76F3		INC HL
30333	767D		DJNZ UND1	30452	76F4		INC HL
30335	767F	UND2:	LD B,H	30453	76F5		INC HL
30336	7680		LD C,L	30454	76F6		CALL 77FH
30337	7681		RET	30457	76F9		POP BC
30338	7682	*****		30458	76FA		JR UF12
30339	7683	UDTOD:	RST 20H	30460	76FC	END	
30339	7683		CALL 0D92H	30460	76FC	VFLAG:	0
30342	7686		CALL 158AH	30461	76FD	VLINNUM:	0 0
30345	7689		LD A,B	30463	76FF	VANZNUM:	0 0
30346	768A		CALL UPH	30465	7701	VANZ1:	0 0
30349	768D		LD A,B	30467	7703	VLEN:	0 0
30350	768E		CALL UPH1	30469	7705	VNUM:	0 0 0 0 0 0
				30475	770B	VNUMTBL:	2048 Bytes

**Preisreduzierte  
Mängel Exemplare**  
(Einbandbeschädigungen)

- Alle Bücher zum halben Preis!
- Nur solange Vorrat reicht!
- Auslieferung erfolgt in der Reihenfolge des Auftragseingangs!

**T. Hartnell  
49 explosive Spiele für den Sinclair ZX81**



1982, 124 Seiten  
Dieses Buch enthält Programme für jedes Spiel, das Sie sich nur wünschen können, wie Galaktischer Angriff, Schmetterball, Dame, Raumschiff Enterprise, Todeslabyrinth, Viererreihe und ein 8-K-Abenteuerspiel Schatzsucher. Einige dieser Spiele laufen nur mit 1 K, wie z.B. Space Invaders.  
Einige Spiele entscheidet das Glück — durch den gefürchteten Sinclair-Zufallsgenerator — die anderen basieren auf Ihrer Fantasie und Geschicklichkeit und der Kapazität des Computers.

Bestellnummer AC 537 DM ~~29,80~~ 14,90

**E. Floegel  
Programmieren mit dem ZX81**



1982, 128 Seiten  
Die Programme in diesem Buch sind aufgeteilt in Spielprogramme, Schul- und andere Programme und Programme zur Datenverwaltung. Alle Programme sind abgeschlossen und lauffähig. Der Leser ist aber aufgefordert, diese Programme nicht als starr zu betrachten, sondern sie nach seinem Belieben zu erweitern oder abzuändern. Für alle diejenigen, die sich über Basic hinaus, mit der Programmierung des Prozessors Z80 beschäftigen wollen, ist ein Kapitel über die Verwendung von Maschinencode eingeschlossen.

Bestellnummer HO 342 DM ~~29,80~~ 14,90

**R. G. Hülsmann  
35 Programme für den ZX81**



1983, 186 Seiten  
Aus dem Inhalt: Allgemeine Tips und Hinweise · 10 Programme für den ZX81 mit 1 K-RAM (Meteor · Space Invader · Mondlandung · Irrgarten · Todeshöhle) · Unterprogramme in Maschinensprache (Hex-Loader I · Hex-Loader II · Screen-Loader · Down-Scroll · Left-Scroll · Right-Scroll) · Zwölf BASIC-Programme für den ZX81 mit 16 K-RAM (LP-Register · Kartei) · Sie haben mehr »RAM«? · (Haushaltsbuchführung · Star-Treck) · Drei Programme in Maschinensprache · (Ballspiele · Flipper · Game of Life)

Bestellnummer HO 407 DM ~~29,80~~ 14,90

**R. G. Hülsmann  
Viel mehr als 33 Programme für den Sinclair Spectrum**



1983, 138 Seiten  
Dieses Buch enthält zunächst einmal die Programme des Buches »35 Programme für den ZX81«. Sie wurden aber nicht einfach konvertiert, sondern teilweise wesentlich erweitert. Insbesondere wurden Fehler ausgemerzt und die Handhabung der Programme »Kartei und LP-Register« wesentlich verbessert.  
Aber es sind auch völlig neue Programme aufgenommen worden: »3-D-Graphik«, »Music-Computer« oder gar »Crazy-Kong«, eine Basic-Version des bekannten Spieles.

Bestellnummer HO 428 DM ~~29,80~~ 14,90

**W. Kremer  
ZX81 + Spectrum**



1983, 203 Seiten  
Sowohl ZX81 wie Spectrum verdienen es, daß die in ihnen steckenden Möglichkeiten erkannt und wahrgenommen werden. Vielfach in der Presse noch als »Spielecomputer« belächelt, läßt sich mit den Maschinen mehr als Pingpong und Sternenkampf anstellen. Die Skala reicht von Energiemanagement im Haus bis zur Textverarbeitung, von der Säulegrafik bis zur Verwaltung tausendfacher Daten.

Bestellnummer ID 567 DM ~~24,-~~ 16,-

**D. Hergert  
MEIN SINCLAIR ZX81**



1983, 176 Seiten  
Dieses Buch erklärt Ihnen, wie Sie Ihren ZX81 voll nutzen können. Schritt für Schritt lernen Sie, wie Ihr ZX81 an den Fernseher und den Kassettenscorder angeschlossen wird, so daß sie als System zusammenarbeiten, wie Sie die Tastatur nutzen, um Ihrem Computer Befehle zu geben, für Graphik-, Rechnungs-, Spiel- und andere Anwendungen Ihre eigenen Programme schreiben können, mit Ihrem ZX81 die Programmiersprache BASIC am besten einsetzen können und die bereits lauffähigen Programme in diesem Buch sofort zur Anwendung bringen.

Bestellnummer SY 405 DM ~~24,-~~ 12,50

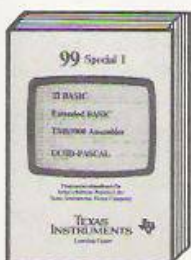
**T. Hartnell  
SINCLAIR ZX SPECTRUM**



Programme zum Lernen und Spielen  
1983, 232 Seiten, 105 Abbildungen  
Dieses Buch ist als Wegweiser zum Computergebrauch zu verstehen und dient als Werkzeug zur unmittelbaren Nutzung Ihres Spectrum. Es knüpft dort an, wo das Geräte-Handbuch aufhört. Schalten Sie beim Lesen das Gerät ein und üben Sie Ihre Fertigkeiten im Programmieren anhand der angeführten Programmbeispiele. Dieses Buch ist sowohl für den Leser, der gerade erst anfängt zu programmieren als auch für den Leser gedacht, der seine Programmierfähigkeiten erweitern möchte.

Bestellnummer SY 426 DM ~~24,-~~ 14,-

**99 Special I**



1983, 298 Seiten  
Das Buch führt vom spielerischen Beginn mehrschrittig aufbauend den TI 99/4A-Anwender zu komplexer Programmierung. Programmbeispiele sind nach steigendem Schwierigkeitsgrad in die Kategorien Spiele, Mathematik, Datenorganisation, Grafik usw. unterteilt; Beispiel-Programm Listings runden den Inhalt ab. Für Leute, die bereits wissen, was Basic ist und die ganze Palette von verfügbaren Programmiersprachen zum TI 99/4A erfahren wollen.

Bestellnummer TE 618 DM ~~49,80~~ 24,75

**G. Daubach  
Wörterbuch der Computerei**



1983, 144 Seiten  
Wer hat nicht bereits verzweifelt versucht, das »Computerchinesisch« zu verstehen! Hier hilft das Wörterbuch der Computerei mit seinen über tausend Begriffen. Außerdem sind die wichtigsten Begriffe erklärt. Ein handliches Nachschlagewerk für jeden, der sich mit Computerei beschäftigt.

Bestellnummer IW 584 DM ~~32,-~~ 16,-

A. Osborne

## Mikrocomputer-Grundwissen



Bestellnummer PW 156

**3. Auflage, 297 Seiten**  
Für alle, die in der Mikrocomputertechnik mitreden wollen oder müssen, ist dieses Buch der richtige Einstieg. Es werden keine Vorkenntnisse erwartet. Nach sechs Lernschritten beherrschen Sie die Materie und können mitreden, wenn es um das aktuelle Thema Mikrocomputer geht. Mit den Testfragen am Ende jedes Kapitels können Sie Ihren Lernerfolg kontrollieren und bestätigt sehen.

DM ~~38,-~~ **18,-**

A. Osborne

## Einführung in die Mikrocomputertechnik



Bestellnummer PW 138

**4. überarbeitete Auflage, 488 Seiten**  
Das schon legendäre Standardwerk spiegelt den allerletzten Stand der Mikrocomputertechnik wieder. Sie werden nicht nur die Komponenten eines Mikrocomputersystems richtig beurteilen, sondern auch zu Fragen der Programmierung, Vergleich alternativer Mikrocomputersysteme, Beurteilung zukünftiger Entwicklungen Stellung nehmen können.

Dieses Buch vermittelt alle Kenntnisse, die zum Verständnis heute existierender Mikroprozessoren und Mikrocomputer gehören.

DM ~~56,-~~ **33,-**

C. Lorenz/Ken Tracton

## 57 praktische Basic-Programme



Bestellnummer HO 559

**1979, 153 Seiten**  
Ein Buch mit technisch-wissenschaftlichen Programmen und einer großen Anzahl von Spielprogrammen in Basic. Ein Buch für jeden, der sich mit dem faszinierenden Hobby der Mikrocomputertechnik befassen will. Alle Listings sind in Basic und können auf den meisten Personal Computer-Systemen gefahren werden. Alle Programme wurden sorgfältig getestet. Zum Beweis ist für jedes Programm ein Protokoll des Probelaufs abgedruckt.

DM ~~38,-~~ **19,50**

D. Possin

## CBasic — CB80



Bestellnummer IW 572

**1983, 187 Seiten**  
CBasic ist eine Version, die sich von den meisten anderen BASIC-Dialekten abhebt, da sie keine Zeilenstruktur aufweist, sondern eine strukturierte Form anbietet, wie dies beispielsweise von Pascal her bereits bekannt ist. Dieses Buch ermöglicht die schnelle Einarbeitung in die Programmiersprache CBasic. Alle Eigenschaften werden mit Beispielen ausführlich erläutert. Syntaxdiagramme zeigen grafisch die Einsatzmöglichkeiten der Anweisungen.

DM ~~56,-~~ **28,-**

K. W. Hillerkus

## Basic aus der Praxis



Bestellnummer IW 573

**Typische Programmbeispiele für alle Berufe 1983, 163 Seiten**  
Dieses Buch enthält 30 lauffähige Programme aus den Arbeitsbereichen: Suchen — Schreiben — Rechnen — Sortieren. Sie sind an keinen Rechner gebunden, da sie unter CP/M und MBasic geschrieben sind. Sie entstanden aus der praktischen Arbeit heraus und haben sich deshalb bereits bewährt. Auch der »Newcomer« kann sie ohne Schwierigkeiten einsetzen.

DM ~~40,-~~ **20,-**

K. W. Hillerkus

## Basic aus der Praxis



Bestellnummer IW 583

**Programm-Beispiele für kaufmännisch orientierte Berufe 1983, 192 Seiten**  
Dieses Buch umfaßt 34 Programme und Routinen aus den Arbeitsbereichen: Suchen — Schreiben — Rechnen — Sortieren — Statistik. Die Programme sind unter CP/M in MBasic geschrieben und können somit leicht auf andere Rechner umgesetzt werden.

DM ~~40,-~~ **20,-**

H. Sterner/D. Herrmann

## Wirtschaft auf dem Apple II+, IIe



Bestellnummer IW 585

**Basic-Programme für den Anwender 1983, 200 Seiten**  
Diese Sammlung enthält 40 BASIC-Programme für den Apple II/IIe aus den Bereichen Finanzmathematik, Unternehmensforschung und Betriebswirtschaft. Wirtschaftliche Fragestellungen treffen jeden von uns, entweder als Steuerzahler, Sparer oder Kreditnehmer. Zahlreiche Beispiele erläutern die vollständigen Programme.

DM ~~38,-~~ **19,-**

L. Poole/M. Borchers

## 77 Basic-Programme



Bestellnummer PW 256

**1980, 195 Seiten**  
Dieses Buch ist für Leser gedacht, die eine Vielzahl von praktisch anwendbaren Basic-Programmen suchen. Es enthält 77 Kurzprogramme, die finanzielle, mathematische, statistische sowie allgemeine interessante Aufgaben wie Umwandlung von Einheiten etc. behandeln. Jedes Programm wurde sorgfältig getestet und anhand von Beispielen detailliert erläutert. Jedes Programm wird zunächst kurz beschrieben und später mit Hinweisen auf Änderungsmöglichkeiten der entsprechenden Programme aufgeführt.

DM ~~38,-~~ **19,50**

B. Pol

## Wie man in BASIC programmiert



Bestellnummer VO 376

**Einführung · Techniken · Fallstudien 1981, 367 Seiten**  
Im ersten Kapitel wird der Leser an die Grundlagen des Programmierens mit Basic herangeführt. Im zweiten werden die wichtigsten Programmierhilfsmittel besprochen. Das dritte Kapitel schließlich macht den Löwenanteil aus: Hier soll als Fallstudie ein (sehr) einfaches Lagerverwaltungsprogramm erstellt und daran die bis dahin erworbenen Erkenntnisse vertieft und erweitert werden.

DM ~~38,-~~ **15,-**

R. Baumann

## Spiel, Idee und Strategie programmiert in Pascal



Bestellnummer VO 574

**1983, 326 Seiten**  
Das Buch ermöglicht dem Leser zu spielen, sich zu unterhalten, seine geistigen Fähigkeiten auszubilden und gleichzeitig Kenntnisse im Programmieren mit Pascal zu gewinnen und zu vertiefen. Es verlangt und fördert vorausschauendes und schlußfolgerndes Denken, Analysieren und Kombinieren und führt vom Spielkonsum weg zum kreativen Umgang mit Spielen und etwas Mathematik.

DM ~~35,-~~ **17,50**

D. Price

## Strukturiertes Programmieren in Pascal



Bestellnummer ID 575

**1983, 235 Seiten**  
Dieses Buch gibt eine vollständige Einführung in die Programmiersprache Pascal. Erfahrung im Programmieren oder besondere mathematische Kenntnisse werden nicht vorausgesetzt. Gut verständliche Erklärungen und Musterprogramme erleichtern Ihnen die Anwendung neuer Programmier-Ideen. Gleichzeitig werden Sie mit dem Gedanken des »strukturierten Programmierens« vertraut gemacht, dessen Ziel die Erstellung von leicht lesbaren und leicht zu verwendenden Computerprogrammen ist.

DM ~~28,-~~ **14,90**

G. Schnell/K. Hoyer

## Mikrocomputer-Interfacebibel



Bestellnummer VV 639

**1984, 175 Seiten**  
Über die Anschaltung der Mikroprozessoren an Peripheriegeräte existiert leider sehr wenig allgemeine Information. Dieser Mangel wird durch die »Interfacebibel« in hervorragender Weise behoben. Die ausführlich kommentierten Interface-Programme sind in der leicht verständlichen, mikroprozessorunabhängigen Assemblersprache CALM geschrieben. Sowohl für den Studenten der Informatik und Datentechnik als auch für den einschlägig tätigen Ingenieur, Physiker und Techniker ein immer wieder hilfreiches Arbeitsbuch.

DM ~~38,-~~ **16,-**

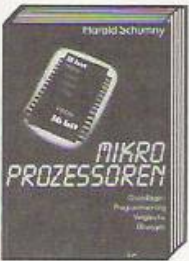
Alle Preise inkl. MwSt.  
Der Versand erfolgt mit offener Rechnung  
zuzügl. Porto und Versand

G. Tatzl  
**Praktische Problemanalyse**



1983, 319 Seiten  
Dieses Buch — aus den Erfahrungen langjähriger Praxis für die Praxis geschrieben — leistet einen Beitrag zur Behebung der vielzitierten Softwarekrise. Ohne den Leser in ein enges Denkschema zu pressen, wird bei Wahrung eines Mindestmaßes an Systematik vorwiegend die kreative und intuitive Seite angesprochen. Anhand einfacher bis mittlerer Beispiele wird in diesem Buch der Weg eines »springenden Funkens« verfolgt. Hieraus lassen sich optimale Problemgestaltungen ableiten.  
Bestellnummer VO 612 **DM ~~48,-~~ 22,50**

Norbert Hoffmann  
**Anwendung von Mikrocomputern**



Digitale Regelung mit Mikroprozessoren  
1983, 169 Seiten  
Das Buch ermöglicht, ein funktionsfähiges Mikroprozessor-Regelsystem zu entwerfen und zu realisieren. Vorausgesetzt wird die Kenntnis des Aufbaus und der Programmierung von Mikrocomputern.  
Inhalt: Grundlagen der digitalen Regelung — Praktische Probleme: Verbindung von Prozeß und Rechner/Verschiedene Detailprobleme/Typen digitaler Regler/Systemplanung — Vollständige Dokumentation eines Regelsystems mit dem AIM-65 als Regler — Entwurf eines Regelsystems mit dem AIM-65 als Regler — Entwurf eines Heizungssystems mit mehreren unabhängigen Heizkreisen.  
Bestellnummer VV 435 **DM ~~48,-~~ 24,-**

H. Schumny  
**Mikroprozessoren**



1983, 240 Seiten  
Das Arbeitsbuch besteht aus zwei Teilen, die auch selbständig nutzbar sind. Teil 1 behandelt »Grundlagen und Basisoperationen« und stellt eine Einführung für Anfänger dar, ist aber auch als Repetitorium für Fortgeschrittene geeignet. Teil 2 hat die »Programmierung im Maschinencode« zum Inhalt. Dies ist also der eigentliche Arbeitsteil. Durch besondere grafische Gliederung wird dieser Teil dem Einsteiger das Arbeiten erleichtern. Fortgeschrittenen wird dadurch mit schnellem Zugriff das Wiederholen oder Vertiefen ermöglicht.  
Bestellnummer VV 570 **DM ~~48,-~~ 24,-**

G. Oetzmann  
**Programmieren von Mikrocomputern 2**



Lehr- und Übungsbuch für die Rechnerreihen cbm 2001 und cbm 3001  
1981, 115 Seiten, mit zahlreichen Beispielen, 8 vollständigen Programmen und 32 Bildern  
Die Rechner der Serien cbm 2001 und cbm 3001 können in cbm-BASIC programmiert werden, einer BASIC-Variante mit verschiedenen herstellereigenen Besonderheiten. Dieser zweite Band der Reihe »Programmieren von Mikrocomputern« enthält eine umfassende Einführung in cbm-BASIC, ohne daß Programmierkenntnisse vorausgesetzt werden. Neben der Vermittlung der Sprache wird an verschiedenen Beispielen der Weg von der ersten Lösungsidee bis zum fertigen Programm demonstriert.  
Bestellnummer VV 313 **DM ~~29,80~~ 14,90**

Bitte freimachen

Postkarte  
Antwort



Markt & Technik  
Verlag Aktiengesellschaft  
Buchverlag  
Hans-Pinsel-Straße 2

8013 Haar bei München

**Lieferanschrift**

Liefern Sie bitte meine Bestellung an folgende Adresse

Name des Bestellers \_\_\_\_\_  
Anschritt \_\_\_\_\_  
PLZ \_\_\_\_\_ Ort \_\_\_\_\_  
Telefon \_\_\_\_\_

Bitte freimachen

Postkarte  
Antwort



Markt & Technik  
Verlag Aktiengesellschaft  
Buchverlag  
Hans-Pinsel-Straße 2

8013 Haar bei München

**Lieferanschrift**

Liefern Sie bitte meine Bestellung an folgende Adresse

Name des Bestellers \_\_\_\_\_  
Anschritt \_\_\_\_\_  
PLZ \_\_\_\_\_ Ort \_\_\_\_\_  
Telefon \_\_\_\_\_

Folgende im Heft abgedruckten Listings sind auch auf Kassette erhältlich:  
Bingo S. 37 Beets'n'Eggs S. 52  
Babylon S. 44 Kniffel S. 47  
Roulette S. 107 MacWash S. 57  
Best.-Nr. SI 003; DM 19,90\*  
\* inkl. MwSt. Unverbindliche Preisempfehlung

Alle Preise inkl. MwSt.  
Der Versand erfolgt mit offener Rechnung  
zuzügl. Porto und Versand

G. Oetzmann  
Programmieren von Mikrocomputern 5

Lehr- und Übungsbuch für die Rechnerreihen  
CBM 4001 und CBM 8001



1983, 119 Seiten  
Die Rechner der Serien CBM 4001 und CBM 8001 können in CBM-Basic programmiert werden, einer Basic-Variante mit verschiedenen herstellereigenen Besonderheiten. Dieser Band enthält eine umfassende Einführung in CBM-Basic, ohne daß Programmierkenntnisse vorausgesetzt werden. Neben der Vermittlung der Sprache wird an verschiedenen Beispielen der Weg von der ersten Lösungsidee bis zum fertigen Programm demonstriert.

Bestellnummer VV 346 ~~DM 29,80~~ 14,90

G. Oetzmann  
Programmieren von Mikrocomputern 7

Lehr- und Übungsbuch für Commodore-Volkscomputer



1983, 109 Seiten  
Commodore-Volkscomputer können in einer gegenüber der Standardversion deutlich erweiterten Variante der Programmiersprache Basic programmiert werden. Dieses Buch bietet eine umfassende Einführung in VC-Basic, ohne daß Programmierkenntnisse vorausgesetzt werden. Besonderer Wert wurde darauf gelegt, neben der Vermittlung der Sprache auch den Weg von der ersten Lösungsidee bis zum fertigen Programm an verschiedenen Beispielen zu demonstrieren.

Bestellnummer VV 576 ~~DM 29,80~~ 12,40

P. Kahlig  
Programmieren von Mikrocomputern 8

Assembler-Programmierung von Mikroprozessoren (8080, 8085, Z80) mit dem ZX81



1983, 185 Seiten  
Das Buch bietet Übungen zur Assembler-Programmierung der Mikroprozessoren 8080, 8085 und Z80, die mit dem Mikrocomputer Sinclair ZX 81 durchgeführt wird. Der Leser soll dadurch angeregt werden, sich vertiefte Hardware- und Software-Kenntnisse anzueignen. Aus dem Inhalt: Hilfsprogramm »KEY« — Eingabe von Konstanten und Registern — Löschen von Registern — Addition, Subtraktion, Multiplikation, Division und Kehrwert von 8-Bit-Daten — Gangang.

Bestellnummer VV 577 ~~DM 38,-~~ 19,-

W. Schneider  
Programmieren von Mikrocomputern 9

Einführung in die Anwendung des Betriebssystems CP/M



1983, 146 Seiten  
Das Buch gibt eine grundlegende Einführung in das für 8-Bit-Mikrocomputer bevorzugte CP/M-Betriebssystem. Es werden alle die Kommandos besprochen, die der Anwender im Normalfall benötigt. Aus dem Inhalt: Aufbau von Datenverarbeitungsanlagen — Programmiersprachen — Betriebssysteme — Grundlagen des CP/M-Betriebssystems — Starten des CP/M-Betriebssystems — CP/M-Dateinamen — Kopieren der CP/M-Betriebssystemdiskette — Das DIR-Kommando — Das STAT-Kommando — Das ED-Kommando — Das PIP-Kommando — Das TYPE-Kommando.

Bestellnummer VV 578 ~~DM 29,80~~ 14,90

★HAPPY★  
COMPUTER

BUCH-BESTELLKARTE

Liefern Sie mir zum Ladenpreis und gegen Rechnung:

Ich möchte vorab Ihren Buchkatalog

Anzahl	Bestell-Nr.	Titel	Einzel-Preis inkl. MwSt

Zuzüglich DM 3,- Versandkostenanteil. Bitte beachten: Es werden nur Festbestellungen berücksichtigt. Eine Rückgabemöglichkeit besteht nicht. Ausnahme nur bei Beschädigung.\*  
Genaue Lieferanschrift umseitig nicht vergessen!

Datum

Unterschrift

★HAPPY★  
COMPUTER

KASSETTEN-BESTELLKARTE

Liefern Sie mir zum Ladenpreis und gegen Rechnung:

Ich möchte vorab Ihren Softwarekatalog

Anzahl	Bestell-Nr.	Titel	Einzel-Preis inkl. MwSt

Zuzüglich DM 3,- Versandkostenanteil. Bitte beachten: Es werden nur Festbestellungen berücksichtigt. Eine Rückgabemöglichkeit besteht nicht. Ausnahme nur bei Beschädigung.\*  
Genaue Lieferanschrift umseitig nicht vergessen!

Datum

Unterschrift

# Mensch ärgere Dich nicht!

Jeder Spieler hat vier Kegel, die mit Ziffern (1,2,3,4) gekennzeichnet sind. Gewählt werden die entsprechenden Kegel mit den Tasten 1 bis 4. Kann nur mit einem Kegel gezogen werden, fährt der Spectrum damit automatisch. Stehen mehrere Kegel zur Auswahl, überläßt er Ihnen die Entscheidung. Wählen Sie einen nicht zum Ziehen berechtigten Kegel, meldet das der Computer. Wird ein Kegel »geschmissen«, befördert ihn der Spectrum in den Warteraum zurück.

Das Bewegungsfeld besteht aus 70 Positionen ( $z(x), y(x)$ ). Die Positionen mit  $x=1..40$  stellen einen Umlauf dar. Da aber der 4. Spieler bereits bei  $x=30$  einsetzt und ebenfalls einmal das Bewegungsfeld durchlaufen muß, wurde dieses bis auf  $x=70$  erweitert, wobei ab  $x=40$  eine Überlagerung auftritt. Die  $x$ -Werte können also von 1 bis 70 gehen. Je nach Spieler wird die entsprechende Differenz zum absoluten Start ( $x=1$ ) beim Bewegen des Kegels addiert ( $pd=10, 20$  beziehungsweise 30). Der Punkteendstand nach Bewegen eines Kegels liegt jedoch immer zwischen 0 und 40, so daß ein geringerer Aufwand betrieben werden mußte, um den Punktevergleich zur Ermittlung des Zielfeldes auszuführen. Die Umrechnung zwischen Punkteendstand und Positionsendstand erfolgt jeweils vor beziehungsweise nach dem Bewegen eines Kegels.

Die nötigen Pausen zum Lesen des Textes auf dem Bildschirm beziehungsweise beim Spiel zum Lesen des Kommentars wurden mit mehr oder weniger aufmunternden Melodien erzeugt.

(Josef Kastl)

Einer der Klassiker  
unter den  
Brettspielen,  
Mensch ärgere  
Dich nicht,  
kann mit diesem  
48-KByte-Spectrum  
am Bildschirm  
gespielt werden.



## Erläuterung zum Programm

Anmerkung:	Die im Flußdiagramm angeschriebenen Zahlen weisen auf die entsprechenden Programm-Zeilen hin! (UPR = Unterprogramm)
Zeile	
10-24	Variable definieren und =0 setzen
40	Vier Felder s, t, u, v definieren (entsprechen 1., 2., 3., 4. Kegel des jeweiligen Spielers)
50-54	Punktestände der Kegel =0 setzen
60	Spielerzähler =0 setzen
100-120	Titelbild
120-132	Auflisten der Spielregeln, während die Grafiken und das Bewegungsfeld generiert werden (GOSUB 7800, GOSUB 8000)
160	Löschen des Bildschirms
162-199	Eingabe der Spielerzahl und -namen
200-298	Grafik des Spielfeldes
290	Löschen des Kommentarfeldes
300-499	UPR (Unterprogramm): Würfeln und Würfelgrafik
500-550	UPR: 3mal würfeln
551-565	UPR: Spielerreihenfolge
1000-1100	Ablaufsteuerung
3001-3450	UPR: Bewegen eines Kegels
3595-3699	UPR: Einsetzen eines neuen Kegels
3740-3765	UPR: Wegfahren vom Start
3999-4050	UPR: Überspringen eines Kegels
4200-4320	UPR: Schmeißen eines Kegels
5000-5290	Nach »Hause«-Fahren
6000-6150	Siegerehrung
6170	Abschlußbild
7000-7250	UPR: Musik
7800-7870	UPR: Definition des Bewegungsfeldes
8000-8070	UPR: Definition der Grafiken
a =	Kreis
b =	Doppelkreis
c =	Kegel 1
d =	Kegel 2
e =	Kegel 3
f =	Kegel 4
g =	Viereck
h...r =	Würfelaugen

## Programmstruktur

## Erläuterung der Variablen

ka, kb, kc, kk =	Sprungvariable
k\$ =	Grafikzeichen des aktuellen Kegels
c\$, d\$, e\$ =	Namen der Sieger
f\$ =	Name des Verlierers
x\$ =	Ausweichstring
s(4), t(4), u(4), v(4)	
=	Felder der 4 Kegel jedes Spielers
zum Beispiel	
s(1) =	Kegel 1 von Spieler 1
s(2) =	Kegel 1 von Spieler 2
t(3) =	Kegel 2 von Spieler 3
t(4) =	Kegel 2 von Spieler 4
z(x), y(x) =	Zeile, Spalte, wo der sich bewegende Kegel hinspringt
a\$(aspi) =	aktueller Spielernamen
x,z,k,p,q,r,i =	Zählvariable bei FOR...NEXT-Schleifen
aspi =	aktueller Spieler
spi =	Anzahl der Spieler
w =	Würfelaugenzahl
anf =	Anfangspunktzahl des Spielers bei Einsetzen eines Kegels
pd =	Punktdifferenz zwischen Anfangspunktzahl und dem absoluten Zählbeginn des Bewegungsfeldes (z(1), y(1))
pa =	40 + pd: letzte Punktzahl vor dem nach »Hause«-Fahren
pb =	Maximale Punktzahl, die der Kegel haben darf, damit er »zuhaus« einfahren kann.
f =	Zählvariable der FOR...NEXT-Schleife während des Fahrens
fv =	f-1: kennzeichnet die vorhergegangene Position, wo entweder leeres Feld oder übersprungener Kegel geprinted wird.
m, n =	Koordinaten der Kegel in Wartestellung
maspi =	Ausweichvariable für aspi
xz, yz =	Position »zuhaus«

## Variablenliste





JOE's heitere Spieleammlung  
praesentiert:

M E N S C H

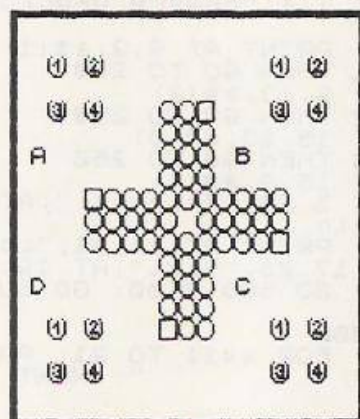
A E R G E R E

D I C H

N I C H T

V i e l U e r g n u e g e n

MENSCH AERGERE DICH NICHT



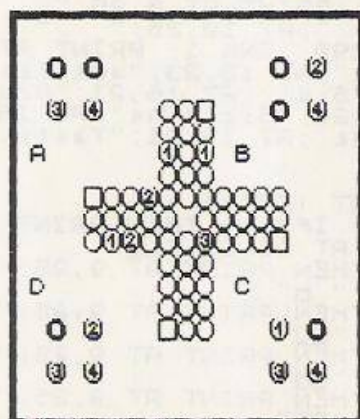
Spieler:

A

Wuerfeln:

Sie duerfen  
3mal wuer-  
feln, bis  
Sie eine  
6 haben.

MENSCH AERGERE DICH NICHT



Spieler:

C

Wuerfeln:



Sie bewegen  
den Kegel

②

Hardcops des Spielverlaufes

```

1>REM          / < < MENSCH AERGER
E DICH NICHT \ \ \
2>REM
3>REM          @ by Josef KASTL
4>REM          Sittling 27
5>REM          8425 Neustadt/Do
nau
6>REM
10>LET ka=0
11>LET kb=0
12>LET kc=0
13>LET kk=0
14>LET pn=1
15>LET pv=0
16>LET ss=0
20>LET k$=""
21>LET c$=""
22>LET d$=""
23>LET e$=""
24>LET f$=""
40>DIM s(4): DIM t(4): DIM u(4)
): DIM v(4)
50>FOR z=1 TO 4
52>LET s(z)=0: LET t(z)=0: LET
u(z)=0: LET v(z)=0
54>NEXT z
56>LET aspi=0
99>REM
100>REM          * * * * * TITEL * *
* * *
101>BORDER 2: GO TO 110
103>BEEP .01,3*k: BEEP .02,4*k:
BEEP .01,3*k
104>FOR x=0 TO k: PRINT "": NEX
T x: RETURN
110>PAPER 1: LET k=4: GO SUB 10
3: LET k=2: PAPER 6: GO SUB 103:
PAPER 5: GO SUB 103: PAPER 4: G
O SUB 103: PAPER 3: GO SUB 103:
LET k=4: PAPER 1: GO SUB 103
115>INK 6: PRINT AT 1,2;"JOE's
heitere Spieleammlung";AT 3,13;
"praesentiert:"
117>INK 1: PAPER 6: PRINT AT 6,
2;"M E N S C H": PAPER 5: PRINT
AT 9,7;"A E R G E R E": PAPER 4:
PRINT AT 12,16;"D I C H": PAPER
3: PRINT AT 15,19;"N I C H T "
119>PAPER 1: INK 4: PRINT AT 19
,1;"V i e l U e r g n u e g e
n"
120>GO SUB 7080: INK 0: GO SUB
160
121>PRINT AT 0,10;"SPIELREGELN"
;AT 2,0;"Das Spiel kann nur mit
einer '6' begonnen werden. Um die
se zu erhalten darf man 3-mal w
uerfeln,";AT 5,0;"Mit einer '6'
muss ein neuer Ke-gel eingesetzt
werden, sofern nicht alle im
Spiel sind. Mit dem eingesetzt
en Kegel muss so- fort vom Start
wegfahren wer- den. Bei Zuwid
erhandlung weist Sie der Spectr
um darauf hin."
123>GO SUB 8000
124>PRINT AT 15,0;"Bitte blaett
ern Sie um,";AT 17,0;"Druucken S
ie eine Taste!": PAUSE 0
125>CLS : GO SUB 160
126>PRINT AT 0,0;"Jeder Spieler
hat 4 Kegel in einer bestimm
ten Farbe,";AT 2,20;"C D E E";AT
Basic-Listing »MADN«

```

Wenn Sie heute bestellen und schon  
morgen spielen wollen, dann...

Nutzen Sie unseren heißen Draht!

**Joysoft**

Humboldtstraße 84, 4 Düsseldorf 1

02 11-680 14 03

```

4,0;"Gewählt werden die entspr
echen-den Kegel mit den Tasten 1
...4."
127 PRINT AT 7,0;"Kann nur mit
einem Kegel gefah- ren werden, f
aehrt der Spectrum mit diesem au
tomatisch. Stehen mehrere Kegel
fahrbereit zur Auswahl, uebe
rleasst er Ihnen die Wahl. Wae
hlen Sie einen nicht zum Fah
ren berechtigten Kegel, meldet
dies der Spectrum."
128 PRINT AT 16,0;"Wird ein Keg
el geschmissen, be- foerdert ihn
der Spectrum in denWarterraum zu
rueck."
130 GO SUB 7800
132 PRINT AT 20,0;"Druecken Sie
eine Taste, und das Spie
l kann losgehen." PAUSE 0
140 GO SUB 160: GO TO 162
160 CLS : LET k=21: PAPER 4: GO
SUB 104: RETURN
161 REM
162 REM * * * ANZAHL DER SPI
ELER * * *
163 INK 0: PRINT AT 4,2;"Wievie
le Personen wollen Mitspi
elen?";AT 7,2;"Die Anzahl der Spi
eler ist auf 4 begrenzt!";
AT 11,2;"Bitte geben Sie die Anz
ahl ein und druecken Sie 'E
NTER'."
164 INPUT "Wieviele Spieler? ";
spi
165 IF spi>4 THEN FLASH 1: PRIN
T AT 13,2;"Es koennen nur 4 Pers
onen";AT 19,2;"mitspielen!": PAU
SE 150: FLASH 0: LET spi=4
167 GO SUB 160
170 REM
171 REM * * * EINGABE DER NA
MEN * * *
172 INK 0: PRINT AT 1,2;"Wer sp
ielt mit?";AT 4,2;"Bitte geben S
ie die Vornamen ein.";AT 7,2;
"Pro Name werden nur 6 Buchsta
ben angenommen. Bitte kuerzen
Sie diese ggf.";AT 11,2;"Nach j
edem Namen 'ENTER' - Taste
druecken!!!"
174 DIM a$(spi,6)
175 FOR x=1 TO spi
176 INPUT "Spieler = ";a$(x)
177 PRINT AT 14+x,7;a$(x)
178 NEXT x
199 REM
200 REM * * * * * SPIELFELD
* * * * *
203 CLS : LET k=21: PAPER 6: GO
SUB 104
205 BORDER 4: PAPER 3: INK 0
210 FOR x=1 TO 40
212 PRINT AT z(x),y(x);"A": NEX
T x
216 PAPER 7: PRINT AT 7,11;"G";
AT 13,15;"G";AT 17,9;"G";AT 11,5
;"G"
218 FOR x=1 TO 4: PRINT AT 12,5
+x;"B";AT 7+x,10;"B";AT 12,15-x;
"B";AT 17-x,10;"B": NEXT x: PAPE
R 4
219 REM
220 REM * * * SPIELERDARSTEL

```

```

LUNG * * *
222 PAPER 1: INK 7: PRINT AT 5,
3;"C";AT 5,5;"D";AT 7,3;"E";AT 7
,5;"E"
224 PAPER 5: PRINT AT 5,15;"C";
AT 5,17;"D";AT 7,15;"E";AT 7,17;
"E"
226 PAPER 4: INK 0: PRINT AT 17
3;"C";AT 17,5;"D";AT 19,3;"E";A
T 19,5;"E"
228 PAPER 2: PRINT AT 17,15;"C"
;AT 17,17;"D";AT 19,15;"E";AT 19
,17;"E"
229 PAPER 5: FOR x=0 TO 2: PRIN
T AT x,0;"
": NEXT x: FOR x=3 TO
21: PRINT AT x,0;" ";AT x,20;"
": NEXT x
230 PLOT 10,2: DRAW 148,0: DRAW
0,148: DRAW -148,0: DRAW 0,-148
: PLOT 9,1: DRAW 150,0: DRAW 0,1
50: DRAW -150,0: DRAW 0,-150: PL
OT 8,0: DRAW 152,0: DRAW 0,152:
DRAW -152,0: DRAW 0,-152
239 REM
240 REM * * * * BESCHRIFTUNG
DES FELDES * * *
242 PRINT AT 1,4;"MENSCH AERGER
E DICH NICHT"
243 PAPER 5: PRINT AT 9,2;a$(1)
244 IF spi=1 THEN GO TO 250
245 PRINT AT 9,13;a$(2)
246 IF spi=2 THEN GO TO 250
247 PRINT AT 15,13;a$(3)
248 IF spi=3 THEN GO TO 250
249 PRINT AT 15,2;a$(4)
260 PRINT AT 3,22;"Spieler:";AT
7,22;"Wuerfel:"
270 PAPER 5: PRINT AT 14,21;"Lo
s geht's";AT 17,23;"Viel";AT 19,
25;" Spass!": GO SUB 7100: GO SU
B 290
288 GO TO 1000
290 PAPER 5: FOR x=11 TO 21: PR
INT AT x,21;" ": NEXT
x: RETURN
298 REM
300 REM * * * * * WUERFELN *
* * * *
301 PAPER 5: PRINT AT 9,25;"
";AT 9,25;" ";AT 10,25;" "
305 GO SUB 290: INK 1: PRINT AT
12,21;"Bitte";AT 13,23;"wuerfel
n";AT 14,28;"Sie!";AT 16,21;"Dru
ecken";AT 17,21;"Sie eine";AT 18
,21;"beliebige";AT 19,21;"Taste.
"
307 PAUSE 0
310 LET w=INT (RND*6)+1
312 PAPER 7: IF w=1 THEN PRINT
AT 9,25;"1";AT 10,25;"1k"
313 IF w=2 THEN PRINT AT 9,25;"
H";AT 10,25;"H"
314 IF w=3 THEN PRINT AT 9,25;"
ML";AT 10,25;"LQ"
315 IF w=4 THEN PRINT AT 9,25;"
HH";AT 10,25;"HH"
316 IF w=5 THEN PRINT AT 9,25;"
MN";AT 10,25;"PO"
317 IF w=6 THEN PRINT AT 9,25;"
OO";AT 10,25;"BB"
319 PLOT 200,88: DRAW 16,0: DRA
W 0,16: DRAW -16,0: DRAW 0,-16
321 RETURN

```



```

395 IF kk=1 THEN GO TO 4280
499 REM
500 REM      * * * * 3 x WUERFELN
* * *
505 GO SUB 290
507 PRINT AT 12,21;"Sie duerfen
";AT 13,21;"3mal wuerfeln";AT 14,21
";AT 15,21;"bis";AT 15,21;"Sie eine
";AT 16,21;"6 haben."
508 GO SUB 7120
509 LET z=3
510 GO SUB 300
513 LET z=z-1
515 IF w=6 THEN GO TO 3600
517 GO SUB 290
520 PRINT AT 12,21;"Sie haben";
AT 13,21;"noch ";z;AT 14,21;"Cha
nce(n)".
525 GO SUB 7140
530 IF z=0 THEN GO SUB 290; PRI
NT AT 12,21;"Sie haben";AT 13,21
";"Leider kein";AT 14,21;"Glueck
";AT 15,21;"habt.";AT 17,21;"
Der";AT 18,21;"Naechste";AT 19,2
1;" bitte."; GO SUB 7140; GO T
O 1000
535 GO TO 510
550 REM
551 REM      * * * SPIELERREIHENF
OLGE * * *
553 LET aspi=aspi+1
555 IF aspi=spi+1 THEN LET aspi
=1
560 IF s(aspi)>100 AND t(aspi)>
100 AND u(aspi)>100 AND v(aspi)>
100 THEN GO TO 553
565 RETURN
1000 REM
1001 REM      * * * * ABLAUFSTEUER
UNG * * *
1005 GO SUB 550
1010 IF aspi=1 THEN LET pd=0: LE
T anf=1
1012 IF aspi=2 THEN LET pd=10: L
ET anf=11
1014 IF aspi=3 THEN LET pd=20: L
ET anf=21
1016 IF aspi=4 THEN LET pd=30: L
ET anf=31
1020 PAPER 7: PRINT AT 5,23;a$(a
spi)
1025 LET pa=40+pd
1030 IF (s(aspi)=0 OR s(aspi)>(4
0+pd)) AND (t(aspi)=0 OR t(aspi)
>(40+pd)) AND (u(aspi)=0 OR u(as
pi)>(40+pd)) AND (v(aspi)=0 OR v
(aspi)>(40+pd)) THEN GO TO 500
1040 GO SUB 300
1042 IF w=6 AND (s(aspi)=0 OR t(
aspi)=0 OR u(aspi)=0 OR v(aspi)=
0) THEN GO TO 3600
1050 GO SUB 3000
1059 REM      * Noechmal wuerfeln *
1060 IF w=6 THEN GO SUB 290: PAP
ER 5: INK 0: PRINT AT 12,21;"Sie
duerfen";AT 13,21;"noch einmal
";AT 14,21;"wuerfeln.";AT 16,21;"
Sie hatten";AT 17,21;"eine '6'!";
GO SUB 7180; GO TO 1040
1075 GO SUB 290: PAPER 5: INK 0:
PRINT AT 14,21;"Der";AT 16,23;"
Naechste";AT 18,26;"bitte."; GO
SUB 7200
1100 GO TO 1005

```

```

3000 REM
3001 REM      * * * BEWEGEN EINES
KEGELS * * *
3005 REM      * * Punktevergleich
zur Bestimmung des zu bewegenden
Kegels * *
3010 LET pb=44-w
3015 IF (s(aspi)=0 OR s(aspi)>40
) AND (u(aspi)=0 OR u(aspi)>40)
AND (v(aspi)=0 OR v(aspi)>40) TH
EN LET k$="D": LET pv=t(aspi): G
O TO 3200
3020 IF (t(aspi)=0 OR t(aspi)>40
) AND (u(aspi)=0 OR u(aspi)>40)
AND (v(aspi)=0 OR v(aspi)>40) TH
EN LET k$="C": LET pv=s(aspi): G
O TO 3200
3025 IF (s(aspi)=0 OR s(aspi)>40
) AND (t(aspi)=0 OR t(aspi)>40)
AND (v(aspi)=0 OR v(aspi)>40) TH
EN LET k$="E": LET pv=u(aspi): G
O TO 3200
3030 IF (s(aspi)=0 OR s(aspi)>40
) AND (t(aspi)=0 OR t(aspi)>40)
AND (u(aspi)=0 OR u(aspi)>40) TH
EN LET k$="E": LET pv=v(aspi): G
O TO 3200
3040 GO SUB 290
3050 PRINT AT 12,21;"Mit welchem
";AT 13,21;"Kegel wol-";AT 14,21
";"len Sie";AT 15,21;"fahren?"
3055 PRINT AT 17,21;"D - Taste 1
";AT 18,21;"D - Taste 2";AT 19,2
1;"E - Taste 3";AT 20,21;"E - Ta
ste 4"
3057 PAUSE 0: PAUSE 0
3060 IF INKEY$="1" THEN LET k$="
D"
3062 IF INKEY$="2" THEN LET k$="
C"
3064 IF INKEY$="3" THEN LET k$="
E"
3066 IF INKEY$="4" THEN LET k$="
E"
3068 GO TO 3750
3072 LET pb=44-w
3080 IF k$="C" THEN LET pv=s(asp
i)
3082 IF k$="D" THEN LET pv=t(asp
i)
3084 IF k$="E" THEN LET pv=u(asp
i)
3086 IF k$="E" THEN LET pv=v(asp
i)
3090 IF pv=0 THEN GO TO 3105
3100 GO TO 3200
3105 IF (s(aspi)>pb OR s(aspi)=0
) AND (t(aspi)>pb OR t(aspi)=0)
AND (u(aspi)>pb OR u(aspi)=0) AN
D (v(aspi)>pb OR v(aspi)=0) THEN
GO TO 5010
3108 GO SUB 290
3110 PRINT AT 12,21;"Sie haben";
AT 13,21;"einen un-";AT 14,21;"b
eweglichen";AT 15,21;"Kegel";AT
16,21;"gewaehlt!";AT 18,21;"Bitt
e";AT 19,21;"tippen Sie";AT 20,2
1;"nochmal."
3115 GO SUB 7060
3120 GO TO 3040
3199 REM      * * * Bewegen eines
Kegels * * *
3200 IF pv>100 THEN LET pv=pv-10
0

```

Basic-Listing »MADN« (Fortsetzung)

Wenn Sie eine freundliche Bedienung  
und spielkundige Berater suchen, dann...

# Joysoft

Humboldtstraße 84  
4000 Düsseldorf 1, Tel. 0211-6801403

```

3203 LET pn=pv+w
3204 IF anf>pv THEN LET pv=pv+40
: LET pn=pn+40
3205 IF pn>pa THEN LET pn=pn+100
: IF pn=s(asp1) OR pn=t(asp1) OR
pn=u(asp1) OR pn=v(asp1) THEN G
O TO 5010
3208 IF pn>144+pd THEN GO TO 501
0
3210 GO SUB 290
3215 PAPER 5: INK 0: PRINT AT 12
,21;"Sie bewegen";AT 13,21;"den
Kegel"
3217 PRINT AT 15,24;k$
3218 FOR x=1 TO 10: BEEP .05,8:
NEXT x
3220 FOR f=(pv+1) TO pn
3230 BEEP .05,15
3240 IF asp1=1 THEN PAPER 1: INK
7
3250 IF asp1=2 THEN PAPER 5: INK
7
3260 IF asp1=3 THEN PAPER 2: INK
0
3270 IF asp1=4 THEN PAPER 4: INK
0
3280 IF ka=1 THEN RETURN
3285 IF f>pa THEN GO TO 5100
3300 PRINT AT z(f),y(f);k$
3320 LET fv=f-1
3325 IF fv=1 OR fv=11 OR fv=21 O
R fv=31 OR fv=41 OR fv=51 OR fv=
61 THEN PAPER 7: INK 0: PRINT AT
z(fv),y(fv);"G": GO TO 3350
3330 PAPER 3: INK 0: PRINT AT z(
fv),y(fv);"H"
3350 IF fv<>pv THEN GO SUB 4000
3380 NEXT f
3385 LET x$=k$
3387 IF pn>40 THEN LET pn=pn-40
3390 GO SUB 4200
3395 LET k$=x$
3400 IF k$="Q" THEN LET s(asp1)=
pn
3410 IF k$="D" THEN LET t(asp1)=
pn
3420 IF k$="E" THEN LET u(asp1)=
pn
3430 IF k$="E" THEN LET v(asp1)=
pn
3435 IF kc=1 THEN GO TO 5106
3440 LET pn=0
3450 RETURN
3590 REM
3595 REM * * * * EINSETZEN EI
NES NEUEN KEGELS * * * *
3600 IF s(asp1)=anf OR t(asp1)=a
nf OR u(asp1)=anf OR v(asp1)=anf
THEN GO TO 1050
3605 GO SUB 290
3610 PAPER 5: INK 0: PRINT AT 13
,21;"H U R R A !";AT 15,21;"Sie
duerfen";AT 16,21;"mit einem";AT
17,21;"Kegel neu";AT 18,21;"beg
innen."
3615 GO SUB 7020
3620 LET ka=1: GO SUB 3240: LET
ka=0
3630 IF s(asp1)=0 THEN PRINT AT
z(anf),y(anf);"C": GO SUB 3675:
GO SUB 3665: LET s(asp1)=anf: GO
TO 1060
3640 IF t(asp1)=0 THEN PRINT AT
z(anf),y(anf);"D": GO SUB 3680:

```

```

GO SUB 3665: LET t(asp1)=anf: GO
TO 1060
3650 IF u(asp1)=0 THEN PRINT AT
z(anf),y(anf);"E": GO SUB 3685:
GO SUB 3665: LET u(asp1)=anf: GO
TO 1060
3660 IF v(asp1)=0 THEN PRINT AT
z(anf),y(anf);"F": GO SUB 3690:
GO SUB 3665: LET v(asp1)=anf: GO
TO 1060
3665 LET pn=anf: GO SUB 4200: RE
TURN
3675 IF asp1=1 THEN LET m=5: LET
n=3: GO TO 3695
3676 IF asp1=2 THEN LET m=5: LET
n=15: GO TO 3695
3677 IF asp1=3 THEN LET m=17: LE
T n=15: GO TO 3695
3678 IF asp1=4 THEN LET m=17: LE
T n=3: GO TO 3695
3680 IF asp1=1 THEN LET m=5: LET
n=5: GO TO 3695
3681 IF asp1=2 THEN LET m=5: LET
n=17: GO TO 3695
3682 IF asp1=3 THEN LET m=17: LE
T n=17: GO TO 3695
3683 IF asp1=4 THEN LET m=17: LE
T n=5: GO TO 3695
3685 IF asp1=1 THEN LET m=7: LET
n=3: GO TO 3695
3686 IF asp1=2 THEN LET m=7: LET
n=15: GO TO 3695
3687 IF asp1=3 THEN LET m=19: LE
T n=15: GO TO 3695
3688 IF asp1=4 THEN LET m=19: LE
T n=3: GO TO 3695
3690 IF asp1=1 THEN LET m=7: LET
n=5: GO TO 3695
3691 IF asp1=2 THEN LET m=7: LET
n=17: GO TO 3695
3692 IF asp1=3 THEN LET m=19: LE
T n=17: GO TO 3695
3693 IF asp1=4 THEN LET m=19: LE
T n=5: GO TO 3695
3695 IF kk=1 THEN GO TO 4280
3696 PAPER 7: INK 0
3698 PRINT AT m,n;"B"
3699 RETURN
3740 REM * * Vom Start weg fah
ren * *
3750 IF k$="Q" AND (t(asp1)=anf
OR u(asp1)=anf OR v(asp1)=anf) T
HEN GO TO 3760
3752 IF k$="D" AND (s(asp1)=anf
OR u(asp1)=anf OR v(asp1)=anf) T
HEN GO TO 3760
3754 IF k$="E" AND (s(asp1)=anf
OR t(asp1)=anf OR v(asp1)=anf) T
HEN GO TO 3760
3756 IF k$="E" AND (s(asp1)=anf
OR t(asp1)=anf OR u(asp1)=anf) T
HEN GO TO 3760
3758 GO TO 3072
3760 GO SUB 290: PRINT AT 12,21;
"Sie haben";AT 13,21;"einen un-";
AT 14,21;"erlaubten";AT 15,21;"
Kegel ge-";AT 16,21;"wahlt!";AT
18,21;"sie muessen";AT 19,21;"v
om";AT 20,21;"Startpunkt";AT 21,
21;"wegfahren."
3763 GO SUB 7060
3765 GO TO 3040
3990 REM
3999 REM * * * * UEBERSPRINGE

```



```

N EINES KEGELS * * * *
4000 IF fv>40 THEN LET fv=fv-40
4002 IF fv=s(1) OR fv=t(1) OR fv
=u(1) OR fv=v(1) THEN PAPER 1: I
NK 7
4005 IF fv=s(2) OR fv=t(2) OR fv
=u(2) OR fv=v(2) THEN PAPER 5: I
NK 7
4010 IF fv=s(3) OR fv=t(3) OR fv
=u(3) OR fv=v(3) THEN PAPER 2: I
NK 0
4015 IF fv=s(4) OR fv=t(4) OR fv
=u(4) OR fv=v(4) THEN PAPER 4: I
NK 0
4020 IF fv=s(1) OR fv=s(2) OR fv
=s(3) OR fv=s(4) THEN PRINT AT z
(fv),y(fv);"D"
4025 IF fv=t(1) OR fv=t(2) OR fv
=t(3) OR fv=t(4) THEN PRINT AT z
(fv),y(fv);"D"
4030 IF fv=u(1) OR fv=u(2) OR fv
=u(3) OR fv=u(4) THEN PRINT AT z
(fv),y(fv);"E"
4035 IF fv=v(1) OR fv=v(2) OR fv
=v(3) OR fv=v(4) THEN PRINT AT z
(fv),y(fv);"E"
4040 IF pv>40 THEN LET fv=fv+40
4050 RETURN
4199 REM
4200 REM * * * * SCHMEISSEN E
INES KEGELS * * * *
4210 LET maspi=aspi
4220 FOR r=1 TO 4
4225 LET aspi=r
4227 LET kk=1
4228 LET x$=k$
4230 IF s(aspi)=pn THEN LET k$="
D": LET s(aspi)=0: GO TO 3675
4235 IF t(aspi)=pn THEN LET k$="
D": LET t(aspi)=0: GO TO 3680
4240 IF u(aspi)=pn THEN LET k$="
E": LET u(aspi)=0: GO TO 3685
4245 IF v(aspi)=pn THEN LET k$="
E": LET v(aspi)=0: GO TO 3690
4250 NEXT r
4260 GO TO 4310
4280 GO SUB 290: INK 0: PRINT AT
12,21;"OH WEH!";AT 13,23;a$(as
pi);AT 14,21;"Sie duerfen";AT 15
,21;"noch mal";AT 16,21;"von vor
ne";AT 17,21;"anfangen.";AT 19,2
1;"BERGERE";AT 20,26;"DICH";AT 2
1,27;"NICHT"
4285 GO SUB 7220
4290 LET ka=1: GO SUB 3240: LET
ka=0
4300 PRINT AT m,n;k$
4310 LET aspi=maspi
4312 LET kk=0
4315 LET k$=x$
4320 RETURN
5000 REM
5002 REM * * * * NACH 'HAUSE'
FAHREN * * * *
5005 REM * * Uorentscheidung
* *
5010 GO SUB 290
5020 PRINT AT 13,21;"Sie koennen
";AT 15,21;"nicht";AT 17,21;"fa
hren!"
5025 IF s(aspi)>pb AND t(aspi)>p
b AND u(aspi)>pb AND v(aspi)>pb
THEN GO TO 5040
5030 IF (s(aspi)>0 AND s(aspi)<4

```

```

0 AND s(aspi)<>pn) OR (t(aspi)>0
AND t(aspi)<40 AND t(aspi)<>pn)
OR (u(aspi)>0 AND u(aspi)<40 AN
D u(aspi)<>pn) OR (v(aspi)>0 AND
v(aspi)<40 AND v(aspi)<>pn) THE
N GO TO 5035
5032 GO TO 5040
5035 IF pn<145+pd THEN PRINT AT
19,21;"Waehlen Sie";AT 20,21;"ei
nen";AT 21,23;"anderen.": GO SUB
7060: GO TO 3040
5040 IF w=6 THEN PAUSE 50: GO TO
1060
5042 PRINT AT 19,21;"Bitte";AT 2
0,22;"diesmal";AT 21,23;"aussetz
en"
5050 GO SUB 7060
5060 GO TO 1070
5100 REM * * Nach 'Hause' fah
ren * *
5105 LET kc=1: GO TO 3400
5106 LET kc=0
5110 IF f=pa+1 THEN PAPER 3: INK
0: PRINT AT z(pa),y(pa);"B"
5115 IF f=pa+1 THEN LET fv=pa: G
O SUB 4000
5120 FOR f=1 TO (pn-140-pd)
5140 IF aspi=1 THEN LET yz=12: L
ET xz=5+f
5145 IF aspi=2 THEN LET yz=7+f:
LET xz=10
5150 IF aspi=3 THEN LET yz=12: L
ET xz=15-f
5155 IF aspi=4 THEN LET yz=17-f:
LET xz=10
5160 IF kb=1 THEN RETURN
5165 LET ka=1: GO SUB 3240: LET
ka=0
5170 BEEP .05,20
5200 PRINT AT yz,xz;k$
5210 IF f=(pn-140-pd) THEN GO TO
5240
5220 PAPER 7: INK 0: PRINT AT yz
,xz;"B"
5240 REM * * Ueberspringen ei
nes Kegels 'zu Hause' * *
5242 LET x$=k$
5245 LET fv=f-1
5250 IF fv=0 THEN GO TO 5280
5253 LET f=fv: LET kb=1: GO SUB
1140: LET kb=0: LET f=fv+1
5255 LET ka=1: GO SUB 3240: LET
ka=0
5257 LET fv=fv+140+pd
5260 IF s(aspi)=fv THEN LET k$="
D": GO TO 5270
5262 IF t(aspi)=fv THEN LET k$="
D": GO TO 5270
5264 IF u(aspi)=fv THEN LET k$="
E": GO TO 5270
5266 IF v(aspi)=fv THEN LET k$="
E": GO TO 5270
5268 GO TO 5280
5270 PRINT AT yz,xz;k$
5275 LET k$=x$
5280 NEXT f
5285 IF s(aspi)>100 AND t(aspi)>
100 AND u(aspi)>100 AND v(aspi)>
100 THEN GO TO 6000
5290 GO TO 3400
5999 REM
6000 REM * * * * * SIEGEREHRU
NG * * * *
6020 LET ss=ss+1

```

Basic-Listing »MADN« (Fortsetzung)

Wenn Sie Superpreise und eine Riesen-  
Programmauswahl haben wollen, dann...

**Joysoft**

Ellenlange Preisliste anfordern!

Humboldtstraße 84  
4000 Düsseldorf 1, Tel. 02 11-680 14 03

```

6030 GO SUB 290
6040 INK 0: PRINT AT 15,21;"Sie
sind"
6045 IF ss=4 THEN PRINT AT 17,21
;"Leider";AT 18,22;"der";AT 19,2
3;"letzte"; LET r$=a$(aspi):
PAUSE 30: GO TO 6100
6050 PAPER 6: FLASH 1
6060 IF ss=1 THEN PRINT AT 17,21
;"1. Sieger"; LET c$=a$(aspi)
6070 IF ss=2 THEN PRINT AT 17,21
;"2. Sieger"; LET d$=a$(aspi)
6080 IF ss=3 THEN PRINT AT 17,21
;"3. Sieger"; LET e$=a$(aspi)
6085 GO SUB 7240: FLASH 0: PAPER
5
6088 GO TO 1075
6099 REM
6100 REM * * Siegerehrung * *
6110 PAPER 4: CLS : GO SUB 160
6120 PAPER 6: PRINT AT 1,2;"*****
*****";AT 2,2;
" S I E G E R E H R U N G ";AT
3,2;"*****
*****"
6130 PAPER 1: INK 7: PRINT AT 7,
6;"1. Sieger: ";c$;AT 9,6;"2. Si
eger: ";d$;AT 11,6;"3. Sieger: "
e$
6140 PAPER 7: INK 2: PRINT AT 16
,6;"Ungluecklicher";AT 17,6;"Ver
lierer: ";f$
6150 PAPER 4: INK 7: PRINT AT 21
,0;"Neues Spiel: j/n": PAUSE 0
6160 IF INKEY$="j" THEN GO TO 1
6162 IF INKEY$="n" THEN CLS : GO
TO 6165
6165 FOR x=0 TO 250 STEP 10: PLO
T x,0: DRAW 0,175: NEXT x
6166 FOR x=0 TO 175 STEP 5: PLOT
0,x: DRAW 250,0: NEXT x
6170 INK 0: PRINT AT 5,5;"D A N
K E";AT 8,9;"fuer die nette";AT
11,7;"U N T E R H A L T U N G";A
T 20,2;"Bis zum naechstenmal!":
6200 REM
7010 REM * * * * * MUSIK * * *
* *
7020 FOR x=1 TO 3: BEEP .3,0: NE
XT x
7025 FOR x=1 TO 3
7028 BEEP .3,3: BEEP .3,7: BEEP
.3,10: BEEP .5,15
7030 NEXT x
7032 RETURN
7050 FOR x=1 TO 5: BEEP .3,3: BE
EP .3,8: NEXT x: RETURN
7080 LET d=.3
7083 BEEP d,7: BEEP d,2: BEEP d,
7: BEEP d,8: BEEP d,10: BEEP d,1
0: BEEP d,10: BEEP d,10: BEEP d,
8: BEEP d,10: BEEP d,12: BEEP d,
14: BEEP d,12: BEEP d,10: BEEP d
,8
7085 RETURN
7100 LET d=.2
7105 BEEP d,3: BEEP d,7: BEEP d,
9: BEEP d,3: BEEP d,7: BEEP d,9:
BEEP d,8: BEEP d,8: BEEP d,9: B
EEP d,9: BEEP d,9
7110 RETURN
7120 LET d=.2
7123 FOR x=1 TO 2
7125 BEEP d,3: BEEP d,7: BEEP d,

```

```

12: BEEP d,15: BEEP d,19: BEEP d
,15: BEEP d,12: BEEP d,7
7127 NEXT x: RETURN
7140 LET d=.2
7145 BEEP d,3: BEEP d,3: BEEP d,
3: BEEP d,7: BEEP d,7: BEEP d,7
7150 RETURN
7160 LET d=.3
7163 FOR x=1 TO 3
7165 BEEP d,10: BEEP d,10: BEEP
d,10: BEEP d,7: BEEP d,3: BEEP d
,3: BEEP d,3: BEEP d,7
7167 RETURN
7170 RETURN
7180 LET d=.2
7185 BEEP d,10: BEEP d,14: BEEP
d,17: BEEP d,15: BEEP d,19: BEEP
d,15: BEEP d,19: BEEP d,15: BEE
p d,19: BEEP d,17: BEEP d,14: BE
EP d,10
7190 RETURN
7200 BEEP .6,10: BEEP .3,15: BEE
p .6,10
7205 RETURN
7220 DATA -5,-7,-9,-10
7225 RESTORE 7220: FOR x=0 TO 3:
READ a: BEEP .6,a: NEXT x: BEEP
1.8,-12
7227 RETURN
7240 LET d=.3
7245 BEEP d,3: BEEP d,5: BEEP d,
7: BEEP d,5: BEEP d,8: BEEP d,7:
BEEP d,5: BEEP d,2: BEEP d,3: B
EEP d,12: BEEP d,10: BEEP d,8: B
EEP d,7: BEEP d,5: BEEP d,7: BEE
P d,3
7800 REM
7802 REM * * * DEFINITION DES
BEWEGUNGSFELDES * * *
7810 DIM z(70): DIM y(70)
7815 RESTORE 7830
7820 FOR p=1 TO 70
7825 READ z(p): NEXT p
7830 DATA 11,11,11,11,11,10,9,8,
7,7,7,8,9,10,11,11,11,11,12,1
3,13,13,13,13,14,15,16,17,17,
16,15,14,13,13,13,13,13,12,11,11
,11,11,11,10,9,8,7,7,7,8,9,10,11
,11,11,11,11,12,13,13,13,13,13,1
4,15,16,17,17
7840 RESTORE 7855
7845 FOR q=1 TO 70
7850 READ y(q): NEXT q
7855 DATA 5,6,7,8,9,9,9,9,10,1
1,11,11,11,11,12,13,14,15,15,15,
14,13,12,11,11,11,11,11,10,9,9,9
,9,9,8,7,6,5,5,5,6,7,8,9,9,9,9,9
,10,11,11,11,11,11,12,13,14,15,1
5,15,14,13,12,11,11,11,11,11,10
7870 RETURN
7999 REM
8000 REM * * * * * FIGUREN *

```

Basic-Listing »MADN« (Schluß)



# ROULETTE

Roulette gilt als vielseitiges und interessantes Spiel. Da es 113 verschiedene Möglichkeiten gibt, einen Chip zu setzen, ist das Spiel geradezu prädestiniert für einen Computer.

Das Programm wurde so aufgebaut, daß zuerst eine ausführliche Erklärung der Spielregeln erfolgt. Es können maximal vier Spieler teilnehmen. Für jeden Spieler und die Bank (der Computer) wird ein Geldbetrag auf einem Konto verwaltet, über den der Spieler frei verfügen kann. Sodann kann ein Spieler nach dem anderen beliebig setzen. Sollte er einmal nicht mehr weiter wissen, kann er die HELP-Funktion aufrufen. Der Spieler hat ferner die Möglichkeit, innerhalb eines Setzblockes das Spiel durch die END-Funktion zu beenden; er wird dann verabschiedet und über seine Spieler-Qualität aufgeklärt. Für jeden Einsatz des Spielers wird naturgetreu ein Chip auf dem entsprechenden Platz des Roulette-Tisches gesetzt.

Haben alle Spieler ihre Einsätze gemacht, werden nochmals zum Vergleich alle gesetzten Chips auf dem Bildschirm ausgegeben und dann die zufällig gezogene Zahl angezeigt. Jetzt zeigt sich, wer das meiste Glück oder die beste »Geldnase« besitzt.

Bei der nachfolgenden »Auszahlung« werden die Gewinne ausgegeben. Sollte jemand kein Glück gehabt haben, kann er, falls noch Geld auf dem Konto ist, jederzeit neue Einsätze machen.

Jeder Spieler kann mit genügend Geld und Glück auch die Bank sprengen, wonach das Spiel zu Ende ist. Das Spiel ist ferner beendet, wenn kein Spieler mehr Geld auf seinem Konto hat oder alle Spieler vorzeitig ausgeschieden sind.

Das Programm wurde so geschrieben, daß falsche Eingaben vom System her abgefangen werden. So kann selbst Oma sich einmal »Spielbank-Luft« um die Nase wehen lassen.

(Andreas Eifert)

## ROULETTE

	0	
PASSE	1 2 3 4 5 6 7 8 9 10 11 12	MANQUE
PAIR	13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36	IMPAIR
◆		◆
12P 12M 12D		12P 12M 12D

**SPIELREGELN**

■ Volle Zahl (pleine)  
\*35-facher Gewinn

◆ Zwei verbundene Zahlen (cheval)  
17-facher Gewinn

## ROULETTE A

	0	
PASSE	1 2 3 4 5 6 7 8 9 10 11 12	MANQUE
PAIR	13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36	IMPAIR
◆		◆
12P 12M 12D		12P 12M 12D

(■) MENU  
P PLEINE  
N CHEVAL  
T TRANSV.  
Z PLEINE  
0 CARRE  
0 ERSTEN 4  
6 TRANSV.  
C SECHS  
C COLONNE

12P 12M 12D  
PAIR  
IMPAIR  
ROUGE  
NOIR  
MANQUE  
PASSE

```

1000>REM *
1001 REM * PROGRAM roulette
1002 REM * andreas eifert 1984
1003 REM *
1008 POKE 23558,0: POKE 23609,30
: POKE 23561,10
1009 LET b$="": LET null=0: LET
dur=0: BORDER 0: PAPER 0: INK 7
: CLS
1010 LET w$="

```

# TTE ROULE

```

1025 PRINT AT 8,0;w$;AT 15,4;"@
ANDREAS EIFERT (1984)"
1094 PRINT #0; PAPER 4; INK 0; F
LASH 1; AT 1,2;"ROULETTE - BITT
E WARTEN!": GO SUB 3050
1100 PRINT #0; AT 1,0; FLASH 1;"E
RKLAERUNGEN GEWUENSCHT ? (j/n)"
1110 IF INKEY$="n" THEN BEEP .1,
12: GO TO 1760
1120 IF INKEY$<>"j" THEN GO TO 1
110
1150 BEEP .1,12: BORDER 5: PAPER
5: INK 0: CLS : GO SUB 3160
1170 PRINT "Das Roulette gilt
als interes- santestes Gesells
chaftsspiel, daes von mehrerer P
ersonen gleich-zeitig gespielt w
erden kann und unerschöpfliche
Einsatz- und Gewinnmoeglichkei
ten bietet."
1180 PRINT "Es koennen von jed
em Spieler gleichzeitig versc
hiedene Ein- saetze auf Chancen
und Zahlen gemacht werden."
1210 PRINT "Als naechstes werd
en die ver- schiedensten Setzu
ngsmoeglich- keiten an einem Ro
uletetisch erklaert."
1220 GO SUB 3900
1260 PRINT BRIGHT 1; AT 5,25; FLA
SH 1; PAPER 6;"0"; AT 6,13;"0": P
RINT AT 6,22;"Volle Zahl"; AT 7,2
2;"(pleine)": PRINT AT 9,22;"*35
-facher"; AT 10,22;"Gewinn"
1270 PRINT AT 12,25; FLASH 1; PA
PER 7;"0"; AT 15,9;"00": PRINT AT
13,22;"Zwei ver-"; AT 14,22;"bun
dene"; AT 15,22;"Zahlen"; AT 16,22
;"(cheval)"; AT 18,22;"17-facher"
; AT 19,22;"Gewinn"
1280 GO SUB 3900
1310 PRINT BRIGHT 1; AT 5,25; FLA
SH 1; PAPER 6;"1"; AT 7,14;"1": P
RINT AT 6,22;"Querreihe"; AT 7,22
;"von drei"; AT 8,22;"Zahlen"; AT
9,22;"(trans-"; AT 10,22;"versale
"; AT 11,22;"pleine)"; AT 13,22;"*
11-facher"; AT 14,22;"Gewinn"
1320 PRINT AT 16,25; FLASH 1; PA
PER 7;"0"; AT 11,9;"00"; AT 12,9;"
00": PRINT AT 17,22;"Viereck"; AT
18,22;"(carre)"; AT 20,22;"8-fac
her"; AT 21,22;"Gewinn"
1330 GO SUB 3900
1350 PRINT BRIGHT 1; AT 5,25; FLA
SH 1; PAPER 6;"0"; AT 4,14;"0": P
RINT AT 6,22;"ersten 4"; AT 7,22;
"Zahlen"; AT 8,22;"(0,1,2,3)"; AT
10,22;"8-facher"; AT 11,22;"Gewin
n"
1360 PRINT AT 13,25; FLASH 1; PA
PER 7;"1"; AT 13,14;"1"; AT 14,14;
"1": PRINT AT 14,22;"zwei Quer-
"; AT 15,22;"reihen"; AT 16,22;"(tr

```

Basic-Listing »Roulette«

```

ans-"; AT 17,22;"versale 6)"; AT 1
9,22;"* 5-facher"; AT 20,22;"Gewi
nn"
1370 GO SUB 3900
1400 PRINT BRIGHT 1; AT 5,25; FLA
SH 1; PAPER 6;"0"; AT 18,8;"0"; AT
18,10;"0"; AT 18,13;"0": PRINT A
T 5,22;"1 Laengs-"; AT 7,22;"reih
e von"; AT 8,22;"12 Zahlen"; AT 9,
22;"(Colonne)"; GO SUB 3950
1410 GO SUB 3900
1430 PRINT BRIGHT 1; AT 5,25; FLA
SH 1; PAPER 6;"1"; AT 18,1;"1": P
RINT AT 6,22;"1. Dutzend"; AT 7,2
2;"1-12 (12p)"
1440 PRINT AT 9,25; FLASH 1; PAP
ER 7;"0"; AT 18,17;"0": PRINT AT
10,22;"2. Dutzend"; AT 11,22;"13-
24 (12a)"
1450 PRINT INK 7; BRIGHT 1; AT 13
,25; FLASH 1; PAPER 1;"0"; AT 18,
6;"0": PRINT AT 14,22;"3. Dutzen
d"; AT 15,22;"25-36 (12d)"
1460 GO SUB 3950: GO SUB 3900
1500 PRINT BRIGHT 1; AT 5,25; FLA
SH 1; PAPER 6;"2"; AT 11,3;"2": P
RINT AT 6,22;"gerade"; AT 7,22;"Z
ahlen"; AT 8,22;"2,4,6 etc."
1510 PRINT AT 11,25; FLASH 1; PA
PER 7;"1"; AT 11,18;"1": PRINT AT
12,22;"ungerade"; AT 13,22;"Zahl
en"; AT 14,22;"1,3,5 etc."
1520 GO SUB 3940: GO SUB 3900
1560 PRINT BRIGHT 1; AT 5,25; FLA
SH 1; PAPER 6;"2"; AT 16,18;"2":
PRINT AT 6,22;"rote"; AT 7,22;"Za
hlen"; AT 8,22;"7,9,12 etc"
1570 PRINT AT 11,25; FLASH 1; PA
PER 7;"1"; AT 16,3;"1": PRINT AT
12,22;"schwarze"; AT 13,22;"Zahle
n"; AT 14,22;"6,8,17 etc"
1580 GO SUB 3940: GO SUB 3900
1610 PRINT BRIGHT 1; AT 5,25; FLA
SH 1; PAPER 6;"1"; AT 7,18;"1": P
RINT AT 6,22;"erste"; AT 7,22;"Ha
elfte"; AT 8,22;"(1 - 18)"
1620 PRINT AT 11,25; FLASH 1; PA
PER 7;"2"; AT 7,3;"2": PRINT AT 1
2,22;"zweite"; AT 13,22;"Haelfte"
; AT 14,22;"(19 - 36)"
1630 GO SUB 3940: GO SUB 3280: G
O SUB 3160
1650 PRINT "Gewinnt die Null,
so wird ein Sonderfall eintre
ten, der im Programm genau be
sprochen wird."
1720 PRINT "Als Mindesteinsatz
wird DM 10.- verlangt und das Li
mit liegt bei DM 100,000.- pro Sp
iel."
1730 PRINT "Jeder Spieler hat e
in Konto mit einem Betrag von DM
5,000.- und kann ueber das Kont
o freiver- fuegen." "Der Compu
ter uebernimmt die Ge- schaefte
der Bank und verfuegt ueber ein
Konto mit DM 1000000."
1740 GO SUB 3280
1750 IF b$="HELP" THEN GO TO 232
5
1751 IF b$="1HELP" THEN GO TO 23
25
1765 FLASH 0: PAPER 5: BORDER 5:
INK 0: CLS : LET null=0: GO SUB
3160
1790 PRINT AT 5,0; PAPER 6;"SPIE
LEREINGABE"

```



```

1800 PRINT "Gebe die Anzahl der Spieler ein, die mitspielen wollen. (max. 4 Spieler)"
1810 IF INKEY$="" THEN GO TO 1810
1820 IF INKEY$<"1" OR INKEY$>"4" THEN GO TO 1810
1830 LET spi=VAL INKEY$: BEEP .1
1840 PRINT "Es wurden ";spi;" Spieler gewaehlt."
1845 DIM h$(spi,6): DIM n$(spi,10): DIM n(5): DIM a$(spi)
1846 PRINT : POKE 23658,8
1850 FOR i=1 TO spi
1860 INPUT "Gebe den Namen des ";i;" Spielers ein: ";LINE n$(i)
1870 PRINT "Spieler ";i;": ";n$(i): LET n(i)=5000: LET a(i)=0
1880 NEXT i: LET n(5)=1000000
1895 GO SUB 3280
1900 CLS : GO SUB 3160: GO SUB 3200: IF null=0 THEN DIM g$(spi,2,13)
1990 DIM c$(spi,15,36): DIM d$(spi,12,8): DIM f$(spi,3,13): DIM e$(spi,5): DIM c$(13,15): DIM k$(spi): DIM o$(spi): DIM p$(spi): DIM q$(spi): DIM r$(spi): DIM s$(spi)
1992 RESTORE 1995: FOR i=1 TO 13: READ c$(i): NEXT i
1995 DATA "ROUGE", "NOIR", "IMPAIR", "PAIR", "MANQUE", "PASSE", "12 P", "12 M", "12 D", "1. COLONNE", "2. COLONNE", "3. COLONNE", "ERSTEN 4 ZAHLEN"
2000 GO SUB 3300: FOR i=1 TO spi
2004 LET f(i,1,13)=8: LET o$(i)="0": LET p$(i)="0": LET q$(i)="0": LET r$(i)="0": LET s$(i)="0"
2005 FOR n=1 TO 6: LET f(i,1,n)=1: LET f(i,1,n+6)=2: NEXT n
2030 IF null=1 THEN FOR n=1 TO 6: LET f(i,3,n)=h(i,n): NEXT n
2070 NEXT i
2080 IF n(5)<=0 THEN GO TO 3450
2100 LET rnd=INT (RND*37)
2145 IF rnd=0 THEN LET null=1: FOR i=1 TO spi: LET e(i,1)=35: LET e(i,2)=1: NEXT i
2300 GO SUB 3280: FOR i=1 TO spi
2322 IF a(i)=1 THEN GO TO 2700
2325 BORDER 4: PAPER 4
2326 GO SUB 3630: GO SUB 3200
2340 PRINT AT 2,24: PAPER 3: BRIGHT 1: "MENU": AT 3,22: PAPER 5: "P": AT 4,22: "Z": AT 5,22: "T": AT 7,22: "0": AT 8,22: "3": AT 9,22: "6": AT 11,22: "C"
2345 PRINT AT 3,24: "PLEINE": AT 4,24: "CHEVAL": AT 5,24: "TRANSV.": AT 6,24: "PLEINE": AT 7,24: "CARRE": AT 8,24: "ERSTEN 4": AT 9,24: "TRANSV.": AT 10,24: "SECHS": AT 11,24: "COLONNE"
2350 PRINT AT 13,22: CHR$ 148: CHR$ 149: " ": CHR$ 148: CHR$ 150: " ": CHR$ 148: CHR$ 151: AT 14,22: "PAIR": AT 15,22: "IMPAIR": AT 16,22: "ROUGE": AT 17,22: "NOIR": AT 18,22: "MANQUE": AT 19,22: "PASSE"
2352 IF dur=1 OR b$="HELP" THEN GO TO 2500
2353 IF b$="1HELP" THEN GO TO 25
60

```

```

2355 PRINT #0: FLASH 1: PAPER 7: AT 0,0: "GEBE BEI DER EINGABE DEN CODE- BUCHSTABEN (HELLBLAU) ODER DAS GANZE WORT (AB 12P) EIN.": PAUSE 500
2357 PRINT #0: PAPER 6: AT 0,0: "SOLLTE DER SPIELER KEINEN SATZ MEHR MACHEN WOLLEN, EINFACH ENTER DRUECKEN!": PAUSE 500
2510 INPUT PAPER 7: AT 2,0: "ERKLARUNG=HELP SPIELEND=END": AT 1,0: ("KONTO: ";n(i)): AT 1,18: "WEITER='ENTER'": AT 0,0: "EINSATZ: ";LINE h$
2515 IF h$="" THEN LET b$=" ": GO TO 2700
2520 IF h$="END" THEN GO TO 4210
2530 IF h$="HELP" THEN LET b$=h$: GO TO 1140
2540 IF h$( TO 1)>CHR$ 57 OR h$( TO 1)<CHR$ 48 THEN GO SUB 3600: GO TO 2510
2545 IF VAL h$<10 OR VAL h$>1000 THEN PRINT #0: PAPER 7: AT 1,0: "Betrag DM10.- + 100000.- erlaubt": AT 0,0: "FALSCHER BETRAG!": PAUSE 300: GO TO 2510
2546 IF n(i)-VAL h$<0 THEN PRINT #0: PAPER 7: "DU GIBST MEHR GELD AUS, ALS DU AUF DEINEM KONTO HAST!": PAUSE 300: GO TO 2510
2547 IF VAL h$-INT (VAL h$)<>0 THEN PRINT #0: PAPER 7: "GEBE NUR GANZE EINSATZE EIN!": PAUSE 300: GO TO 2510
2550 LET n(i)=n(i)-VAL h$: LET n(5)=n(5)+VAL h$: LET k(i)=1
2560 INPUT PAPER 7: AT 2,0: "ERKLARUNG=HELP SPIELEND=END": AT 1,0: ("KONTO: ";n(i)): AT 0,0: ("EINSATZ: ";h$): "CHANCE: ";LINE a$
2565 IF a$="END" THEN GO TO 4210
2570 IF a$="HELP" THEN LET b$="1"+a$: GO TO 1140
2575 IF a$="12P" THEN GO TO 4250
2580 IF a$="12M" THEN GO TO 4260
2585 IF a$="12D" THEN GO TO 4270
2590 IF a$="PAIR" THEN GO TO 4280
2595 IF a$="IMPAIR" THEN GO TO 4290
2600 IF a$="ROUGE" THEN GO TO 4300
2605 IF a$="NOIR" THEN GO TO 4310
2610 IF a$="MANQUE" THEN GO TO 4320
2615 IF a$="PASSE" THEN GO TO 4330
2620 IF a$="C" THEN GO TO 4000
2625 IF a$="6" THEN GO TO 4040
2630 IF a$="0" THEN GO TO 4070
2635 IF a$="0" THEN GO TO 4090
2640 IF a$="T" THEN GO TO 4120
2645 IF a$="Z" THEN GO TO 4150
2650 IF a$="P" THEN GO TO 4180
2655 GO SUB 3600: GO TO 2560
2700 NEXT i
2701 IF rnd<>0 THEN LET null=0
2706 LET dur=1: FOR i=1 TO spi
2707 IF a(i)=0 THEN GO TO 2710
2709 NEXT i: GO SUB 3160: GO SUB 3200: GO TO 3400
2715 CLS : GO SUB 3160: GO SUB 3200: PRINT AT 0,22: "GESETZTE": AT 1,22: "CHIPS": AT 2,22: "-----"

```

```

2718 PRINT #0; FLASH 1; PAPER 5;
"WELCHE EINSÄTZE WURDEN GEMACHT
?"
2720 FOR i=1 TO spi
2725 IF a(i)=1 OR k(i)<>1 THEN G
O TO 2790
2730 PRINT AT i*3+1,22;n$(i)
2735 PRINT AT i*3+2,23; PAPER i+
3;CHR$(151+i)
2740 FOR n=1 TO 13
2745 IF g(i,1,n)=0 THEN GO TO 27
95
2750 PRINT AT g(i,2,n),g(i,1,n);
PAPER i+3;CHR$(151+i)
2755 NEXT n
2757 FOR y=1 TO 12
2759 FOR x=1 TO 3: IF d(i,y,x)=1
THEN PRINT AT y+4,x+5; PAPER i+
3;CHR$(151+i)
2760 NEXT x: NEXT y
2770 IF e(i,4)<>0 THEN PRINT AT
e(i,4),e(i,5); PAPER 3+i;CHR$(1
51+i)
2790 NEXT i
2800 PRINT #0; PAPER 6; AT 1,0; "
(Nichts geht mehr)
" AT 0,0; " R I E N N E U A
P L U S " : GO SUB 3000
2815 PRINT AT 16,22;"GEZOGENE";A
T 17,22;"ZAHL"; AT 19,22; FLASH
1; " ; rnd; "
2820 PRINT #0; AT 0,0; " : GO SUB
3280
2829 BORDER 5; PAPER 5; CLS
2830 FOR i=1 TO spi
2831 LET sch=0: LET j=5
2832 IF k(i)<>1 OR a(i)=1 THEN G
O TO 2975
2840 GO SUB 3630: PRINT AT 1,18;
"KONTO: ";n(i): PRINT PAPER 6; AT
3,0;"CHANCE: "; AT 3,16;"GEWINN: "
2843 LET gew=e(i,1)*e(i,2)*e(i,3
)
2844 LET go=2849: LET fak=e(i,3)
: LET d$="PLEINE": GO SUB 3800
2849 IF null=1 THEN GO TO 4500
2850 FOR n=1 TO 13
2855 LET gew=f(i,1,n)*f(i,2,n)*f
(i,3,n)
2865 LET go=2889: LET fak=f(i,3,
n): LET d$=c$(n): GO SUB 3800
2889 NEXT n: RESTORE 2992
2891 FOR h=0 TO 4
2893 READ w$: IF h=0 AND o$(i)="
0" OR h=1 AND p$(i)="0" OR h=2 A
ND q$(i)="0" OR h=3 AND r$(i)="0
" OR h=4 AND s$(i)="0" THEN GO T
O 2920
2902 FOR n=1 TO 36
2904 LET gew=c(i,h*3+1,n)*c(i,h*
3+2,n)*c(i,h*3+3,n)
2906 LET go=2912: LET fak=c(i,h*
3+3,n): LET d$=w$: GO SUB 3800
2920 NEXT n: NEXT h
2965 IF sch=0 THEN GO SUB 3630:
PRINT AT 1,18;"KONTO: ";n(i); AT
3,0; PAPER 6;"GEWINN: "; AT 3,16;"
GEWINN: "; PAPER 5; "" "In diesem
Durchgang hattest Du leider kei
n Glück."
2966 IF n(i)<=0 AND a(i)=0 THEN
PRINT "" "Dein Konto ist geplünd
ert und Du musst dem Spiel zusc
hauen." : LET a(i)=1
2967 GO SUB 3280: FOR n=1 TO 6:
LET h(i,n)=0: NEXT n

```

```

2975 NEXT i: GO TO 1980
2992 DATA "PLEINE","CHEVAL","TRA
NSU. PLEINE","CARRE","TRANSU. 6"
2993 LET j=5: GO SUB 3280: GO SU
B 3160: PRINT PAPER 6; AT 3,0;"CH
ANCE: "; AT 3,16;"Gewinn: "; PAPER
5; AT 25,1;"Konto: ";n(i): GO SUB
3630: RETURN
3010 FOR i=1 TO 40: BEEP .1,-6:
BEEP .005,-9: BEEP .07,3: NEXT i
3020 BEEP .01,24: BEEP .05,0: PA
USE 5: BEEP .001,48: BEEP .1,24:
BEEP .05,-4: BEEP .01,0: BEEP .
1,-12: PAUSE 10
3030 BEEP .005,15: BEEP .01,0: B
EEP .005,-36: PAUSE 5: BEEP .1,-
30: BEEP .01,-48: BEEP .1,4: BEE
P .1,0: RETURN
3080 RESTORE 3140: FOR n=1 TO 12
3100 READ p$: FOR f=0 TO 7
3110 READ a: POKE USR p$+f,a
3120 NEXT f: NEXT n
3140 DATA "a",1,3,7,15,31,63,127
,255
3141 DATA "b",128,192,224,240,24
8,252,254,255
3142 DATA "c",255,127,63,31,15,7
,3,1
3143 DATA "d",255,254,252,248,24
0,224,192,128
3144 DATA "e",0,0,38,97,34,119,0
,0
3145 DATA "f",0,56,36,56,32,0,0,
0
3146 DATA "g",0,34,54,42,34,0,0,
0
3147 DATA "h",0,56,36,36,56,0,0,
0
3148 DATA "i",0,126,126,126,126,
126,126,0
3149 DATA "j",129,66,36,24,24,36
,66,129
3150 DATA "k",16,56,124,254,124,
56,16,0
3151 DATA "l",24,24,24,255,255,2
4,24,24
3155 RETURN
3170 PLOT 8,160: DRAW 0,15: DRAW
8,0: DRAW 0,-8: DRAW -8,0: DRAW
8,-7: PLOT 24,160: DRAW 0,15: D
RAW 8,0: DRAW 0,-15: DRAW -8,0
3172 PLOT 40,175: DRAW 0,-15: DR
AW 8,0: DRAW 0,15: PLOT 56,175:
DRAW 0,-15: DRAW 8,0: PLOT 79,16
0: DRAW -8,0: DRAW 0,15: DRAW 8,
0: PLOT 72,168: DRAW 4,0
3175 PLOT 88,175: DRAW 8,0: PLOT
92,175: DRAW 0,-15: PLOT 104,17
5: DRAW 8,0: PLOT 108,175: DRAW
0,-15: PLOT 127,160: DRAW -8,0:
DRAW 0,15: DRAW 8,0: PLOT 120,16
8: DRAW 4,0: RETURN
3210 INK 6: PLOT 80,156: DRAW 56
,0: DRAW 56,-16: DRAW 0,-120: DR
AW -56,-8: DRAW -56,0: DRAW -56,
8: DRAW 0,120: DRAW 56,16
3220 DRAW 0,-144: PLOT 80,140: D
RAW 0,-128: PLOT 96,140: DRAW 0,
-128: PLOT 116,156: DRAW 0,-144:
PLOT 23,36: DRAW 0,-19: PLOT 40
,36: DRAW 0,-21: PLOT 135,36: DR
AW 0,-21: PLOT 152,36: DRAW 0,-1
9
3230 PLOT 4,140: DRAW 168,0: PLO
T 4,103: DRAW 168,0: PLOT 4,71:
DRAW 168,0: PLOT 4,36: DRAW 168,
0: INK 0

```

Basic-Listing »Roulette« (Fortsetzung)

```

3240 PRINT INK 7; PAPER 2; BRIGH
T 1; AT 5,8; "1"; AT 5,12; "3"; AT 6,
10; "5"; AT 7,8; "7"; AT 7,12; "9"; AT
8,12; "12"; AT 9,10; "14"; AT 10,8;
"18"; AT 10,12; "18"; AT 11,8; "19";
AT 11,12; "21"; AT 12,10; "23"; AT 1
3,8; "25"; AT 13,12; "27"; AT 14,12;
"30"; AT 15,10; "32"; AT 16,8; "34";
AT 16,12; "36"; AT 14,17; CHR$ 144;
CHR$ 145; AT 15,17; CHR$ 146; CHR$
147
3250 PRINT AT 3,8; "0"; AT 5,10; "2
"; AT 6,8; "4"; AT 6,12; "6"; AT 7,10
"8"; AT 8,8; "1011"; AT 9,8; "13"; A
T 9,12; "15"; AT 10,10; "17"; AT 11,
10; "20"; AT 12,8; "22"; AT 12,12; "2
4"; AT 13,10; "26"; AT 14,8; "2829";
AT 15,8; "31"; AT 15,12; "33"; AT 16
,10; "35"; AT 14,3; CHR$ 144; CHR$ 1
45; AT 15,3; CHR$ 146; CHR$ 147
3260 PRINT AT 6,1; "PASSE"; AT 6,1
5; "MANQUE"; AT 10,1; "PAIR"; AT 10,
15; "IMPAIR"; AT 10,1; CHR$ 148; CHR
$ 149; CHR$ 148; CHR$ 150; CHR$ 148
; CHR$ 151; AT 10,15; CHR$ 148; CHR$
149; CHR$ 148; CHR$ 150; CHR$ 148;
CHR$ 151; RETURN
3290 PRINT #0; AT 1,3; "DRUECKE IR
GENEINE TASTE"; PAUSE 0; BEEP .
1,12; CLS; RETURN
3320 PRINT AT 0,22; "KONTEN:"; AT
1,22; "-----"; PAPER 4; AT 3,22; "
BANK:"; AT 4,22; "DM "; n(5)
3340 LET we=6; FOR i=1 TO spi
3350 IF a(i)=1 THEN GO TO 3380
3360 PRINT PAPER 5; AT we,22; n$(i
); AT we+1,22; "DM "; n(i)
3370 LET we=we+3
3380 NEXT i; IF we<>8 THEN RETUR
N
3401 POKE 23658,0
3405 PRINT #0; FLASH 1; PAPER 5;
INK 0; AT 0,0; "DAS SPIEL IST VO
RBEI!"; AT 1,0; "NEUES
SPIEL (j/n)?
3410 IF INKEY$="j" THEN BEEP .1;
12; GO TO 1760
3420 IF INKEY$<>"n" THEN GO TO 3
410
3440 BEEP .1,12; NEW
3455 FLASH 1; BORDER 0; PAPER 0;
INK 7; CLS
3460 FOR i=1 TO 90 STEP 3.6; BEE
P .01,1/5; PLOT 127,87; DRAW 127
,i; PLOT 127,87; DRAW i+1,3,88;
PLOT 127,87; DRAW -127,i; PLOT 1
27,87; DRAW -i+1,3,88
3470 PLOT 127,87; DRAW 127,-i; P
LOT 127,87; DRAW i+1,3,-87; PLOT
127,87; DRAW -127,-i; PLOT 127,
87; DRAW -i+1,3,-87; NEXT i
3480 PRINT #0; AT 1,6; "BANK IST G
ESPRENGT"; PAUSE 200; GO TO 3400
3510 PRINT #0; AT 1,7; FLASH 1; "F
ALSCH EINGABE!"; PAUSE 200; RET
URN
3640 CLS; GO SUB 3160
3645 PRINT AT 0,10; FLASH 1; PAP
ER 4; n$(i); FLASH 0; PAPER 3+i; "
"; CHR$ (151+i); " "; RETURN
3650 PRINT AT x,y; PAPER i+3; CHR
$ (151+i); LET g(i,1,d)=y; LET g
(i,2,d)=x; LET f(i,3,d)=VAL h$
3690 GO TO 2510
3700 PRINT AT x,y; PAPER i+3; CHR
$ (151+i); LET x=x-4; LET y=y-6;
LET d(i,x,y)=1

```

Basic-Listing »Roulette« (Fortsetzung)

```

3730 IF a$="Z" OR a$="6" OR a$="
0" THEN RETURN
3740 GO TO 2510
3843 IF gew=0 THEN GO TO 90
3845 LET n(i)=n(i)+gew+fak; LET
n(5)=n(5)-gew-fak
3850 PRINT AT 1,25; n(i); PRINT A
T j,0; d$; AT j,21; gew; " "; LET
j=j+1; LET sch=1
3870 IF gew>fak*5 THEN PRINT FLA
SH 1; "*"
3880 RETURN
3910 GO SUB 3280; GO SUB 3160
3920 GO SUB 3200; PRINT AT 2,21;
"SPIELREGELN"; AT 3,21; "-----
---"; RETURN
3940 PRINT AT 17,22; "* 1-facher"
; AT 18,22; "Gewinn"; RETURN
3950 PRINT AT 17,22; "* 2-facher"
; AT 18,22; "Gewinn"; RETURN
4000 INPUT PAPER 7; AT 0,0; ("EINS
ATZ: "; h$; " CHANCE: colonne"; AT
1,0; "KONTO: "; n(i); AT 2,0; "Welche
Colonne? (1-2-3)"); LINE b$
4002 IF b$="" THEN GO TO 4004
4003 IF b$(1)>CHR$ 48 AND b$(1)<
CHR$ 52 THEN GO TO 4005
4004 GO SUB 3600; GO TO 4000
4005 LET b=VAL b$
4006 FOR n=1 TO 34 STEP 3
4007 IF rnd=n THEN LET f(i,2,10
)=1
4008 IF rnd=n+1 THEN LET f(i,2,
11)=1
4009 IF rnd=n+2 THEN LET f(i,2,
12)=1
4010 NEXT n
4015 LET d=b+9
4020 IF i=1 OR i=2 THEN LET x=18
; LET y=b+2+5+i; GO TO 3670
4021 IF i=3 OR i=4 THEN LET x=19
; LET y=b+2+3+i; GO TO 3670
4045 INPUT PAPER 7; AT 0,0; ("EINS
ATZ: "; h$; " CHANCE: transv. 6"; A
T 1,0; "KONTO: "; n(i); AT 2,0; "1. Z
ahl der transv. 6?"); LINE b$
4046 IF b$="" THEN GO TO 4050
4047 FOR n=1 TO 31 STEP 3
4048 IF b$=STR$ n THEN GO TO 40
51
4049 NEXT n
4050 GO SUB 3600; GO TO 4045
4051 LET b=VAL b$
4053 FOR n=0 TO 5
4054 IF rnd=b+n THEN LET s$(i)=
"1"; LET c(i,13,b)=5; LET c(i,14
,b)=1; LET c(i,15,b)=VAL h$
4055 NEXT n
4057 RESTORE 4620
4058 FOR n=1 TO 31 STEP 3
4059 READ x; LET x=x+1
4060 IF n=b AND i=1 OR i=3 THEN
LET y=7; GO TO 4063
4061 IF n=b AND i=2 OR i=4 THEN
LET y=14; GO TO 4063
4062 NEXT n
4063 LET j=x; LET n=y; GO SUB 37
00; LET x=j-1; LET y=n; GO SUB 3
700; GO TO 2510
4070 LET d=13; LET y=14; LET x=4
4074 IF rnd=0 OR rnd=1 OR rnd=2
OR rnd=3 THEN LET f(i,2,d)=1
4080 IF i=1 OR i=3 THEN LET y=7;
GO TO 4082
4082 GO TO 3670
4091 INPUT PAPER 7; AT 0,0; ("EINS
ATZ: "; h$; " CHANCE: carre"; AT 1,
0; "KONTO: "; n(i); AT 2,0; "kleinste
Zahl der carre?"); LINE b$

```

```

4092 IF b$="" THEN GO TO 4119
4093 IF b$(1)<CHR$ 48 OR b$(1)>C
HR$ 57 THEN GO TO 4119
4094 FOR n=3 TO 33 STEP 3
4095 IF b$=STR$ n THEN GO TO 41
19
4096 NEXT n: LET b=VAL b$
4097 IF b<1 OR b>32 THEN GO TO 4
119
4100 IF rnd=b OR rnd=b+1 OR rnd=
b+3 OR rnd=b+4 THEN LET r$(i)="1
": LET c(i,10,b)=8: LET c(i,11,b
)=1: LET c(i,12,b)=VAL h$
4103 RESTORE 4620: FOR j=1 TO 31
STEP 3: READ x
4104 IF b=j THEN LET y=10: GO T
O 4108
4105 IF b=j+1 THEN LET y=12: GO
TO 4108
4107 NEXT j
4108 LET j=x: LET n=y: GO SUB 37
00: LET x=j+1: LET y=n: GO SUB 3
700: LET x=j+1: LET y=n-1: GO SU
B 3700: LET x=j: LET y=n-1: GO S
UB 3700: GO TO 2510
4119 GO SUB 3600: GO TO 4091
4125 INPUT PAPER 7:AT 0,0:("EINS
ATZ:":h$:" CHANCE: transv.pl."):A
T 1,0:"KONTO:":n(i):AT 2,0:"1. Z
ahl der transv. pleine?"): LINE
b$: IF b$="" THEN GO TO 4130
4127 FOR n=1 TO 34 STEP 3
4128 IF b$=STR$ n THEN GO TO 41
31
4129 NEXT n
4130 GO SUB 3600: GO TO 4125
4131 LET b=VAL b$
4132 IF rnd=b OR rnd=b+1 OR rnd=
b+2 THEN LET q$(i)="1": LET c(i
,7,b)=11: LET c(i,8,b)=1: LET c(i
,9,b)=VAL h$
4135 RESTORE 4620: FOR n=1 TO 34
STEP 3: READ x
4137 IF n=b AND i=1 OR i=3 THEN
LET y=7: GO TO 4140
4138 IF n=b AND i=2 OR i=4 THEN
LET y=14: GO TO 4140
4139 NEXT n
4140 GO TO 3700
4155 INPUT PAPER 7:AT 0,0:("EINS
ATZ:":h$:" CHANCE: cheval"):AT 1
,0:"KONTO:":n(i):AT 2,0:"Gebe di
e 2 Zahlen ein:"): LINE b$:" "
LINE d$
4156 IF b$="" OR d$="" THEN GO T
O 4160
4157 IF b$(1)<CHR$ 48 OR b$(1)>C
HR$ 57 OR d$(1)<CHR$ 48 OR d$(1)
>CHR$ 57 THEN GO TO 4160
4158 LET b=VAL b$: LET d=VAL d$
4159 IF b+1=d OR b-1=d OR b+3=d
OR b-3=d THEN GO TO 4161
4160 GO SUB 3600: GO TO 4155
4161 IF d<b THEN LET x=b: LET b=
d: LET d=x
4162 IF rnd=b OR rnd=d THEN LET
p$(i)="1": LET c(i,4,b)=17: LET
c(i,5,b)=1: LET c(i,6,b)=VAL h$
4164 RESTORE 4620: FOR n=1 TO 34
STEP 3: READ x
4165 IF b=n AND d=b+1 THEN LET
y=10: GO TO 4175
4166 IF b=n+1 AND d=b+1 THEN LE
T y=12: GO TO 4175
4170 IF b=n AND d=b+3 THEN LET
y=9: GO TO 4176
4171 IF b=n+1 AND d=b+3 THEN LE
T y=11: GO TO 4176
4172 IF b=n+2 AND d=b+3 THEN LE
T y=13: GO TO 4176

```

```

4174 NEXT n
4175 LET j=x: LET h=y: GO SUB 37
00: LET x=j: LET y=h-1: GO SUB 3
700: GO TO 2510
4176 LET x=x+1: LET j=x: LET h=y
: GO SUB 3700: LET x=j-1: LET y=
h: GO SUB 3700: GO TO 2510
4185 INPUT PAPER 7:AT 0,0:("EINS
ATZ:":h$:" CHANCE: pleine"):AT 1
,0:"KONTO:":n(i):AT 2,0:"Gebe di
e Zahl ein:"): LINE b$
4186 IF b$="" THEN GO TO 4209
4187 IF b$(1)<CHR$ 48 OR b$(1)>C
HR$ 57 THEN GO TO 4209
4188 LET b=VAL b$
4189 IF b<0 OR b>36 THEN GO TO 4
209
4190 IF b=0 THEN LET e(i,3)=VAL
h$: LET e(i,4)=3: LET e(i,5)=12:
PRINT AT 3,12: PAPER 3+i:CHR$ (
151+i): GO TO 2510
4192 IF rnd=b THEN LET o$(i)="1"
: LET c(i,1,b)=35: LET c(i,2,b)=
1: LET c(i,3,b)=VAL h$
4195 RESTORE 4620: FOR n=1 TO 34
STEP 3: READ x
4197 IF n=b THEN LET y=9: GO TO
4201
4198 IF n+1=b THEN LET y=11: GO
TO 4201
4199 IF n+2=b THEN LET y=13: GO
TO 4201
4200 NEXT n
4201 GO TO 3700
4209 GO SUB 3600: GO TO 4185
4215 CLS : GO SUB 3150: PRINT AT
5,0: PAPER 5:"SPIELLENDE VON ":n
$(i): PRINT "Du beendest das S
piel mit einem Kontostand von ":
n(i):" DM."
4225 IF n(i)>5000 THEN PRINT "Du
u erspieltest einen Gewinn von",
n(i)-5000:" DM.": GO TO 4235
4226 IF n(i)=5000 THEN PRINT "Du
u hast gluecklicherweise kein G
eld verloren.": GO TO 4235
4230 PRINT "Du hast leider ":50
00-n(i):" DM","verspielt."
4235 LET n(i)=0: LET a(i)=1
4240 GO SUB 3280: GO TO 2700
4250 IF rnd<13 THEN LET f(i,2,7)
=1
4255 IF i=1 OR i=2 THEN LET y=i:
GO TO 4257
4256 LET y=i+12
4257 LET x=18: LET d=7: GO TO 36
70
4260 IF rnd>12 AND rnd<25 THEN L
ET f(i,2,8)=1
4265 IF i=1 OR i=2 THEN LET y=i+
2: GO TO 4267
4266 LET y=i+14
4267 LET x=18: LET d=8: GO TO 36
70
4270 IF rnd>24 THEN LET f(i,2,9)
=1
4275 IF i=1 OR i=2 THEN LET y=i+
4: GO TO 4277
4276 LET y=i+16
4277 LET x=18: LET d=9: GO TO 36
70
4280 IF rnd=0 THEN GO TO 4282
4281 IF (rnd/2-INT (rnd/2))=0 TH
EN LET f(i,2,4)=1
4282 LET d=4: IF null<>0 AND f(i
,3,4)<>0 THEN GO TO 4600
4284 LET x=11: LET y=i: GO TO 36
70
4290 IF rnd=0 THEN GO TO 4292

```

```

4291 IF (rnd/2-INT (rnd/2))<>0 T
HEN LET f(i,2,3)=1
4292 LET d=3: IF null<>0 AND f(i
,3,3)<>0 THEN GO TO 4600
4293 LET x=11: LET y=i+14: GO TO
3670
4300 IF rnd=0 THEN GO TO 4302
4301 IF rnd=1 OR rnd=3 OR rnd=5
OR rnd=7 OR rnd=9 OR rnd=12 OR r
nd=14 OR rnd=16 OR rnd=18 OR rnd
=19 OR rnd=21 OR rnd=23 OR rnd=2
5 OR rnd=27 OR rnd=30 OR rnd=32
OR rnd=34 OR rnd=36 THEN LET f(i
,2,1)=1
4302 LET d=1: IF null<>0 AND f(i
,3,1)<>0 THEN GO TO 4600
4303 LET x=16: LET y=i+14: GO TO
3670
4310 IF rnd=0 THEN GO TO 4312
4311 IF rnd=2 OR rnd=4 OR rnd=6
OR rnd=8 OR rnd=10 OR rnd=11 OR
rnd=13 OR rnd=15 OR rnd=17 OR rn
d=20 OR rnd=22 OR rnd=24 OR rnd=
26 OR rnd=28 OR rnd=29 OR rnd=31
OR rnd=33 OR rnd=35 THEN LET f(i
,2,2)=1
4312 LET d=2: IF null<>0 AND f(i
,3,2)<>0 THEN GO TO 4600
4313 LET x=16: LET y=i: GO TO 36
70
4320 IF rnd=0 THEN GO TO 4322
4321 IF rnd<19 THEN LET f(i,2,5)
=1
4322 LET d=5: IF null<>0 AND f(i
,3,5)<>0 THEN GO TO 4600
4323 LET x=7: LET y=i+14: GO TO
3670
4330 IF rnd=0 THEN GO TO 4332
4331 IF rnd>18 THEN LET f(i,2,6)
=1
4332 LET d=6: IF null<>0 AND f(i
,3,6)<>0 THEN GO TO 4600
4333 LET x=7: LET y=i: GO TO 367
0
4505 PRINT "Die Gewinnzahl ist
"0" und alle Einsätze werden ei
ngezogen, mit Ausnahme von Pair,
Impair, Noir, Rouge, Passe und Ma
nque."
4510 PRINT "In diesen sechs Fae
llen kann der Spieler den halben
Einsatz zu- rueckverlangen (1)
oder den Ein-satz fuer das naech
ste Spiel mit vollem Wert stehen
lassen (2).""Welche Regelung m
oechtest Du haben ? (1 oder 2
)"
4540 IF INKEY$="1" THEN BEEP .1,
12: FOR n=1 TO 6: LET f(i,2,n)=.
5: LET f(i,2,n+6)=0: LET f(i,2,1
3)=0: NEXT n: GO SUB 3630: PRINT
AT 1,18:"Konto: ";n(i):AT 3,0:
PAPER 6:"CHANCE: ";AT 3,16:"GEWIN
N: ": GO TO 2850
4545 IF INKEY$<>"2" THEN GO TO 4
530
4570 BEEP .1,12: FOR n=1 TO 6
4580 LET h(i,n)=f(i,3,n)
4590 NEXT n: GO TO 2975
4605 PRINT #0: PAPER 7: FLASH 1:
"ACHTUNG!"; FLASH 0:"Chance mit
";f(i,3,d):" DM be-"; "setzt (Nu
llregel)": PAUSE 200: PAUSE 200:
GO TO 2510
4620 DATA 5,6,7,8,9,10,11,12,13,
14,15,16
4900 SAVE "roulette" LINE 1000
4910 VERIFY "roulette"
4920 STOP

```

## Basic-Listing »Roulette« (Schluß)

## Spieldaufbau:

Aus Gründen der Speicherkapazität wurden auf REMarks verzichtet. Hier sind die Erklärungen zum Programm:

1000-1100	Titelbild und UDG
1100-1751	Erklärungen zum Spiel
1765-1895	Spielereingabe mit Kontenaufbau
1980-2070	Spielbeginn mit Feld-Dimensionierung und Einladen der Gewinn-Faktoren
2080	Abfrage: Bank gleich Null
2100	Ermittlung der Gewinnzahl
2300:2-2357	Beginn der Eingabe des einzelnen Spielers mit Menüaufbau
2510-2550	Eingabe des Einsatzes mit HELP- und END-Funktion und Überprüfung auf Richtigkeit
2560-2655	Eingabe der Chance mit HELP- und END-Funktion und Sprung zur entsprechenden Chance
2715-2820	Totale Chip-Ausgabe und Ausgabe der Gewinnzahl
2829-2975	Gewinnausgabe mit Abfrage der Spielerkonten
3000-3030	up Kugelrollen
3050-3155	up UDG
3170-3175	up Überschrift
3200-3260	up Roulette-Tisch
3300-3440	up Kontenausgabe und Abfrage nach weiterem Spiel
3455-3480	Bank gesprengt
3680-3730	up Ausgabe des gesetzten Chip und Einspeicherung der Koordinaten in Feld g() und d()
3843-3880	Gewinnausgabe (up)
3910-3950	up für Erklärungen 1100
4000-4209	Eingabe für Colonne, transv. 6, carre, transv. pleine, cheval, pleine
4215-4240	Spielende eines Spielers
4280-4333	Eingabe für noir, rouge, passe, manque, pair, impair
4505-4605	»Nullregel«

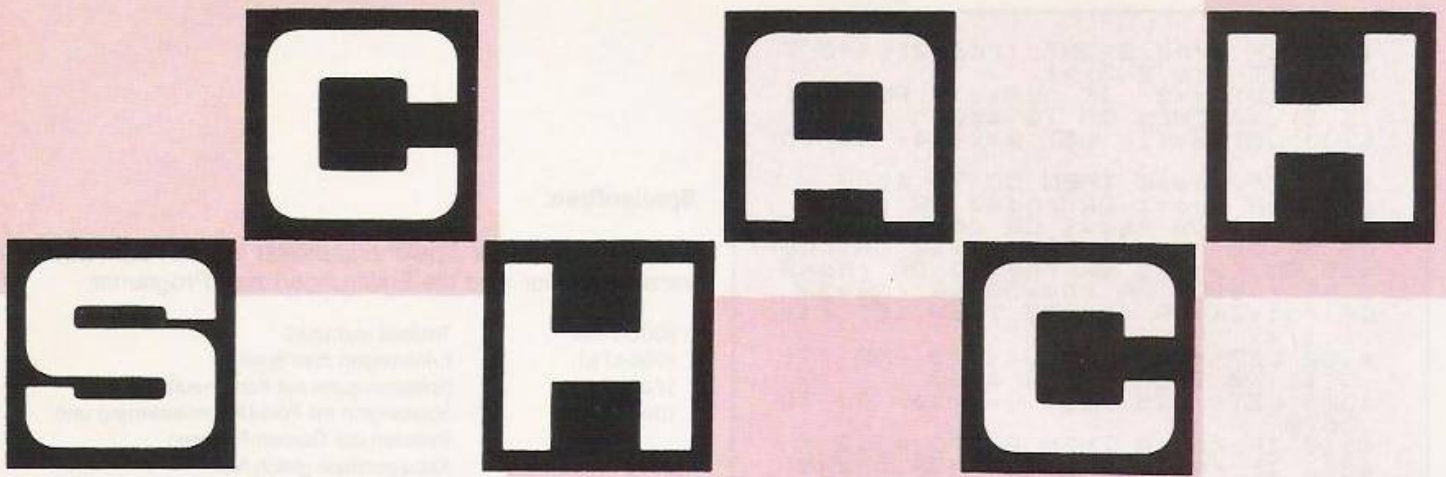
## Programmstruktur

## Variablen:

a\$	Eingabevariable
b\$	HELP-Variable
n\$(i)	Spielername
h\$	Einsatz-Variable
b	Zahleneingabe bei Chance
c()	Feld zur Gewinnberechnung (höhere Chancen)
d()	Feld zum Chipsetzen (höhere Chancen)
e()	Feld für Nullregel
f()	Feld zur Gewinnberechnung (niedrige Chancen)
g()	Feld zum Chipsetzen (niedrige Chancen)
i	Spielernummer (1 bis 4)
n(i)	Spielerkonto
n(5)	Bankkonto
rnd	Roulette-Zahl
spi	Spieleranzahl

o\$, p\$, q\$, r\$, s\$, a(i), dur, k(i), null, sch, sind verschiedene Schlüssel und Schalter, die in das Programm eingebaut wurden, um für ein Basic-Programm ansehnliches Tempo zu erreichen und das Spiel vom menschlich-logischen Gesichtspunkt her richtig aufzubauen.

## Variablenliste



## Editor

**Dieses Programm für den 48-KByte-Spectrum kann nicht Schach spielen. Aber es ist eine äußerst komfortable Hilfe für diejenigen Schachfreunde, die gerne die in Zeitschriften oder Zeitungen abgedruckten Partien nachspielen. Ich habe es auf einem normalen Schachbrett versucht, habe auch die angegebenen Varianten durchgespielt, mußte dann allerdings feststellen, daß ich die ursprüngliche Stellung vor der Variante vergessen hatte.**

Dieses Programm erlaubt es

- bis zu 64 Züge (128 Halbzüge) in ausführlicher Notation oder Kurznotation einzugeben, zu korrigieren und zu listen,
- eine Ausgangsstellung aufzubauen,
- eine Partie nachzuspielen. Hier wird Halbzug für Halbzug der gesamte Spielverlauf nachvollzogen (auf einem Bildschirmbrett). Der Verlauf kann unterbrochen werden, um eine Variante einzugeben. Anschließend kann der ursprüngliche Zustand auf dem Brett wieder hergestellt werden. Das Brett kann umgedreht oder auf dem Drucker ausgegeben werden.
- eine eingetippte Partie mitsamt Ausgangsstellung und Titel auf Band zu speichern oder vom Band zu laden.

### Programmbeschreibung

Nach dem Laden erscheint ein Bild auf dem Schirm. Ist dieses Titelbild fertig aufgebaut, werden die Graphics aus den DATA-Zeilen gelesen und die wichtigsten Variablen definiert.

Dann erscheint das Hauptmenü. Drücken Sie auf 1, dann kommen Sie in das Menü von »Zugeingabe«. Sie können jetzt wählen zwischen ausführlicher Notation (zum Beispiel b1 bis c3) oder Kurznotation (Sc3). Bei beiden Notationen ist die Eingabe der Rochaden identisch (0-0 = kurz, 0-0-0 = lang). Die ausführliche Notation bedarf keiner weiteren Erklärung, die häufiger verwendete Kurznotation jedoch hat es in sich.

Sie können alles genauso eingeben, wie es in Ihrer Zeitschrift oder Zeitung steht. Also Sf3 (Springer nach f3), b4 (Bauer nach b4), bc (Bauer auf b schlägt Figur auf c). Weiterhin Tad8 (Turm, der auf Spalte a steht nach d8), S8d7 (Springer, der auf Zeile 8 steht, nach d7). Für die Bezeichnung der ziehenden Figur müssen Großbuchstaben verwendet werden (kann beim Bauern entfallen, muß aber nicht). Also: Springer = S, König = K und so weiter. Bei einem schlagenden Doppelpauern muß genau festgelegt werden, um welchen es sich handelt. Es müßte dann also zum Beispiel heißen: b3c (der Bauer auf b3 schlägt Figur auf c4). Schlägt ein Bauer en passant, so wird das Programm allein damit fertig. Erreicht ein Bauer die Zeile 1 beziehungsweise 8, dann wird er automatisch in eine Dame umgewandelt, es sei denn, der Zug erhält als viertes Zeichen einen Buchstaben, der eine andere Figur bezeichnet: b8 S (Bauer nach b8 mit Umwandlung in einen Springer).

Jeder Halbzug wird in einem sechs Zeichen langen Teilstring gespeichert. Dadurch wird die Eingabe von Kommentarzeichen möglich: Bei der Kurznotation frühestens vom vierten Zeichen an (Sd4: +? heißt: Springer schlägt d4, schach, ein fragwürdiger Zug), bei der ausführlichen Notation zwischen den beiden Feldern und danach (b5xd4+ heißt: b5 schlägt d4, schach). Die INPUT-Routine, die die Züge entgegennimmt, erwartet stets zwei Halbzüge. Man gibt also zuerst den Zug von weiß ein, drückt ENTER, und dann springt der Cursor etwas nach rechts und der Zug von Schwarz kann eingegeben werden.

Möchte man die Eingabe beenden, muß man nur auf ENTER drücken (zweimal, wenn gerade der weiße Zug dran war), und man gelangt in ein Menü, das das Korrigieren eines Halbzuges, die Rückkehr zum Hauptmenü oder das Ändern der vielleicht falsch eingegebenen Notation ermöglicht. Wenn Sie f und ENTER drücken, dann können Sie korrigieren und anschließend die Eingabe fortsetzen. Bei n und ENTER wird die Notation geändert, dann können Sie fortfahren. Drücken Sie s und ENTER (für Schluß), dann wird ein Syntax-Test durchgeführt, bei dem die Züge auf ihre korrekte Schreibweise hin überprüft werden. Wird ein Fehler entdeckt, verlangt das Programm die sofortige Korrektur. Ist der Syntax-Test abgeschlossen, erscheint wieder das Hauptmenü. Durch Drücken auf 2 können die Züge — das Programm fragt: Drucker? (j/n) — aufgelistet werden. Mit 3 kommen Sie, wenn Sie während des

```

1 REM TU SCHACH LM
RE EDITOR NO
© ANDREAS VIEBKE
THURGAUER STR. 9
1000 BERLIN 51
TEL: (030) 4955689
-----
5 DEF FN I()=2*((F+E+1)/2-INT
((F+E+1)/2))
10 REM TABELLE
15 DIM B$(4,8,8)
20 DIM Z$(65,2,6)
30 DIM M$(8)
40 DIM N$(6): LET A$=""
50 DIM P$(1): LET H$=""
60 DIM F$(4): LET ZNR=1
65 DIM T$(64): LET S$=""
70 LET FELD=1: LET ORI=1
80 LET NOT=2: LET FLAG=1
90 RETURN
1000 REM MAIN
1020 CLS: INK 5: PLOT 16,16: DR
AW 223,0: DRAW 0,143: DRAW -223,
0: DRAW 0,-143: INK 0
1025 PRINT PAPER 5; AT 1,10: "
TU SCHACH LM"; AT 2,10: "
RE EDITOR NO";
1030 PRINT AT 5,6; "ZAHL MOEGLICH
GIBT ES:"
1040 PRINT AT 8,6; "1 ZUEGE EINGE
BEN..."; AT 9,6; "2 ZUEGE AUFLIST
EN..."; AT 10,6; "3 ZUG KORRIGIERE
N..."; AT 11,6; "4 STELLUNG AUFBAU
EN."
1050 PRINT AT 12,6; "5 EINGABE FO
RTSETZEN"; AT 13,6; "6 PARTIE NACH
SPIELEN"; AT 14,6; "7 AUF BAND SPE
ICHERN"; AT 15,6; "8 VOM BAND LADE
N..."; AT 16,10: "
1060 LET I$=INKEY$
1080 IF I$<"1" OR I$>"8" THEN GO
TO 1060
1090 IF I$="1" THEN GO TO 2000
1100 IF I$="4" THEN GO SUB 3500:
GO TO 1000
1110 IF I$="6" THEN GO TO 5000
1120 IF I$="3" THEN CLS: GO SUB
3060: GO TO 1000
1130 IF I$="2" THEN GO TO 1200
1140 IF I$="7" THEN GO TO 1160
1145 IF I$="5" THEN GO TO 2165
1150 IF I$="8" THEN CLS: INPUT
"DATEINAME?": LINE S$: LOAD S$
DATA B$(): LOAD S$ DATA Z$(): LE
T NOT=VAL Z$(65,2,1): LET ZNR=VA
L Z$(65,2,2 TO ): FOR F=1 TO 8:
LET T$(F*8-7 TO )=B$(4,F): NEXT
F: GO TO 1000
1160 CLS: INPUT "DATEINAME?":
LINE S$: "TITEL?": CHR$ 13: LINE
T$: FOR F=1 TO 8: LET B$(4,F)=T
$(8*F-7 TO ): NEXT F: IF LEN S$=
0 OR LEN S$>10 THEN GO TO 1160
1170 LET Z$(65,2,1)=STR$ NOT: LE
T Z$(65,2,2 TO )=STR$ ZNR: SAVE
S$ DATA B$(): SAVE S$ DATA Z$()
1180 VERIFY S$ DATA B$(): VERIFY
S$ DATA Z$(): GO TO 1000
1200 CLS: INPUT "DRUCKER? (J/N)
": LINE P$: PRINT T$: IF P$=
"J" OR P$="j" THEN LPRINT T$
1210 GO TO 2330
2000 REM NEUE ZUG EINGABE
2020 GO SUB 20
2030 CLS: PRINT "
ZUG EINGABE"
2040 PRINT "
BITTE ERST DEN ZUG VON WEISS,
UND DANN DEN GEGENZUG VON
SCHWARZ EINGEBEN.
ES HABEN BIS ZU 64 ZUEGE PLATZ."
2060 PRINT "
SIE HABEN ZWEI MOEGLICHKEITEN,
EINEN ZUG EINGEBEN.
1. FELD-FELD Z.B. g5-e7+
2. KURZNOTATION Z.B. b3!
ODER K97 ODER Thd1 ODER cd"

```

```

2080 PRINT "
KURZE ROCHADE 0-0
LANGE ROCHADE 0-0-0"
2100 PRINT "
INS MENU KOMMEN SIE MIT ENTER"
2110 PRINT "
--BITTE WAEHLLEN SIE 1 ODER 2--"
2120 LET I$=INKEY$
2140 IF I$<"1" OR I$>"2" THEN GO
TO 2120
2160 LET NOT=VAL I$
2165 CLS: PRINT "
ZUG EINGABE"
2170 IF ZNR>64 THEN GO TO 1000
2175 LET M$=""
2180 GO SUB 2290
2200 IF Z$(ZNR,1)=M$ OR Z$(ZNR,2
)=M$ THEN GO TO 3000
2220 LET ZNR=ZNR+1: IF ZNR<=64 T
HEN GO TO 2180
2290 INPUT (ZNR); ".": TAB 4: LINE
Z$(ZNR,1); TAB 11: LINE Z$(ZNR,2
): PRINT TAB 0; ZNR; ".": TAB 4; Z$(
ZNR,1); TAB 11; Z$(ZNR,2): RETURN
2300 REM TEST OB KORREKT
2320 CLS: PRINT "
SYNTAX-TEST"
2330 LET M$="": FOR F=1 TO 64
2340 IF PEEK 23689=2 THEN PAUSE
300: CLS
2350 LET N$=Z$(F,1): PRINT F; ".":
TAB 4; N$: IF P$="J" OR P$="j"
THEN LPRINT F; ".": TAB 4; N$:
2360 IF N$=M$ THEN GO TO 2480
2380 GO SUB 2500: LET Z$(F,1)=N$
2400 LET N$=Z$(F,2): PRINT TAB 1
1; N$: IF P$="J" OR P$="j" THEN L
PRINT TAB 11; N$
2420 IF N$=M$ THEN GO TO 2480
2440 GO SUB 2500: LET Z$(F,2)=N$
2460 NEXT F
2480 IF P$="J" OR P$="j" THEN LP
RINT
2490 BEEP .1,18: LET P$="": PAUS
E 300: GO TO 1000
2500 REM TEST
2510 IF N$(TO 5)="0-0" OR N$(
TO 5)="0-0-0" THEN RETURN
2515 REM NOTATION 3
2520 IF NOT=1 THEN GO TO 2950
2550 IF N$(1)="B" THEN LET N$=N$
(2 TO )
2560 IF N$(1)>="a" AND N$(1)<="h"
AND ((N$(2)>="a" AND N$(2)<="h"
) OR (N$(2)>="1" AND N$(2)<="8"
)) AND ((N$(3)>="a" OR (N$(3)<="a"
AND N$(3)<="h")) THEN RETURN
2580 IF N$(1)<>"T" AND N$(1)<>"S"
AND N$(1)<>"L" AND N$(1)<>"D"
AND N$(1)<>"K" THEN GO TO 2900
2590 IF (N$(1)="L" OR N$(1)="K")
AND N$(3)>"8" THEN GO TO 2900
2600 IF N$(3)>="a" AND N$(3)<="h"
AND N$(4)>="1" AND N$(4)<="8"
AND ((N$(2)>="a" AND N$(2)<="h")
OR (N$(2)>="1" AND N$(2)<="8"))
THEN RETURN
2620 IF N$(2)>="a" AND N$(2)<="h"
AND N$(3)>="1" AND N$(3)<="8"
THEN RETURN
2900 REM NOTATION 3 KORR.
2920 INPUT "FALSCH EINGABE: "; (
N$); ".": LINE N$: GO TO 2510
2950 REM NOTATION 3 TEST U KORR.
2960 IF N$(1)<"a" OR N$(1)>"h" O
R N$(2)<"1" OR N$(2)>"8" OR N$(4
)<"a" OR N$(4)>"g" OR N$(5)<"1"
OR N$(5)>"8" THEN INPUT "FALSCH
EINGABE: "; (N$); ".": LINE N$:
GO TO 2960
2980 RETURN
3000 REM MENU
3020 INPUT "FEHLER SCHLUSS NOT
ATION (; (NOT);) "; LINE I$
3050 IF I$="S" OR I$="s" THEN GO
TO 2300

```

Listing des Schach-Editors

```

3051 IF I$="N" OR I$="n" THEN LE
T NOT=(NOT=1)+1: GO TO 2180
3055 IF I$="F" OR I$="f" THEN GO
SUB 3060: GO TO 2180
3056 GO TO 3020
3060 INPUT "NUMBER? ";E;" FARBE?
(W/S) "; LINE I$
3070 LET E=INT ABS E: IF E<1 OR
E>70 THEN GO TO 3060
3080 INPUT (E);":":(Z$(E,(I$<>"
W" AND I$<>"w")+1));" ZUG? "; L
INE N$
3090 GO SUB 2500
3100 LET Z$(E,(I$<>"W" AND I$<>"
w")+1)=N$: PRINT ;TAB 18;N$
3110 RETURN
3500 REM STELLUNG AUFBAUEN
3510 CLS : LET ORI=1: LET FELD=1
: GO SUB 3800: GO SUB 3900
3515 PRINT AT 4,4;"AUFBAU" ; GU
SGANGSPOS." ; " WOESCHEN" ; " MURUEC
K" ; " SCHWARZ: S" ; " LEERFELD:
" ; " LEERTASTE" ; " Z.B. eIT" ; " ODER
a2Ls"
3520 INPUT "FELD,FIGUR,FARBE: ";
LINE F$
3530 IF F$="L " OR F$="l " T
HEN FOR F=1 TO 8: LET B$(1,F)="
": NEXT F: GO SUB 3900: G
O TO 3520
3540 IF F$="A " OR F$="a " T
HEN GO SUB 9800: GO SUB 3900: GO
TO 3520
3560 IF F$="Z " OR F$="z " T
HEN RETURN
3570 IF F$(1)>="a" AND F$(1)<="h
" AND F$(2)>="1" AND F$(2)<="8"
AND (F$(3)=" " OR F$(3)="B" OR F
$(3)="T" OR F$(3)="S" OR F$(3)="
L" OR F$(3)="D" OR F$(3)="K") TH
EN GO SUB 4000
3600 GO TO 3520
3800 REM DES BRETT ZEICHNEN
3805 LET A$=CHR$ 17+CHR$ 5: LET
H$=CHR$ 17+CHR$ 3: LET I$=" "
: LET I$=A$+I$+H$+I$: LET I$=I$+I$
+I$+I$+I$
3810 PRINT AT 3,15;: FOR F=2 TO
9: LET G=(F/2-INT (F/2))*2: FOR
H=1 TO 2: PRINT TAB 15;I$(G*5+(G
=0) TO 32+4*(G=1)): NEXT H: NEXT
F
3820 INK 3: PLOT 118,22: DRAW 0,
131: DRAW 131,0: DRAW 0,-131: DR
AW -131,0: INK 0
3830 IF ORI THEN FOR F=1 TO 8: P
RINT AT 2*F+2,13;9-F: NEXT F: PR
INT AT 20,15;"A B C D E F G H"
3840 IF NOT ORI THEN FOR F=8 TO
1 STEP -1: PRINT AT 2*F+2,13;F:
NEXT F: PRINT AT 20,15;"H G F E
D C B A"
3850 RETURN
3900 REM FIGUREN HINSTELLEN
3910 LET FLAG=1: FOR S=3 TO 17 S
TEP 2: FOR W=15 TO 29 STEP 2
3920 IF ORI THEN LET C$=B$(FELD,
9-(S-1)/2,(W-13)/2)
3925 IF NOT ORI THEN LET C$=B$(F
ELD,(S-1)/2,9-(W-13)/2)
3930 PAPER 8: PRINT AT S,W;
3935 IF C$=" " THEN PRINT INK 5;
":AT S+1,W;": GO TO 3990
3940 IF C$="B" OR C$="b" THEN PR
INT INK 7*(C$="B");"CD";AT S+1,W
;"HI"
3950 IF C$="T" OR C$="t" THEN PR
INT INK 7*(C$="T");"FG";AT S+1,W
;"HI"
3960 IF C$="S" OR C$="s" THEN PR
INT INK 7*(C$="S");"PQ";AT S+1,W
;"RS"
3970 IF C$="L" OR C$="l" THEN PR
INT INK 7*(C$="L");"JK";AT S+1,W
;"HI"
3980 IF C$="D" OR C$="d" THEN PR
INT INK 7*(C$="D");"TU";AT S+1,W
;"AB"

```

Listing des Schach-Editors (Fortsetzung)

```

3985 IF C$="K" OR C$="k" THEN PR
INT INK 7*(C$="K");"LM";AT S+1,W
;"NO"
3990 IF FLAG=1 THEN NEXT W: NEXT
S: BEEP .2,12
3995 PAPER 7: RETURN
4000 REM FIGUREN SETZEN, NEHMEN
4005 IF F$(3)<>" " AND (F$(4)="S
" OR F$(4)="s") AND F$(3)<>"a" TH
EN LET F$(3)=CHR$(CODE F$(3)+32)
4010 LET W=CODE F$-96
4020 LET S=VAL F$(2)
4025 IF W<1 OR W>8 OR S<1 OR S>8
THEN LET ERR=1: RETURN
4030 LET C$=F$(3)
4040 LET B$(FELD,S,W)=C$
4050 IF ORI THEN LET S=19-S*2: L
ET W=13+2*W
4060 IF NOT ORI THEN LET S=1+2*S
: LET W=31-2*W
4070 LET FLAG=2: GO SUB 3930
4090 RETURN
5000 REM PARTIE NACHSTELLEN
5010 CLS : FOR F=1 TO 8: LET B$(
2,F)=B$(1,F): NEXT F: PRINT T$
5020 LET FELD=2: LET ORI=1: LET
N$="": LET ERR=0
5030 GO SUB 3800: GO SUB 3900
5040 PRINT AT 4,4;"PARTIE" ; W"
" ; " S" ; " RICHTUNG" ; " ARIANTE
" ; " HALBZUG" ; " BRUCKEN" ; " UECHSE
L" ; " STELLEN" ; " MURUECK"
5045 FOR I=1 TO 65: FOR J=0 TO 1
5050 PAUSE 0: LET I$=INKEY$
5060 IF (I$="H" OR I$="h") AND E
RR<>2 THEN GO TO 5110
5070 IF I$="R" OR I$="r" THEN LE
T ORI=ORI=0: GO SUB 3830: GO SUB
3900
5080 IF I$="Z" OR I$="z" THEN GO
TO 1000
5090 IF I$="V" OR I$="v" THEN GO
TO 5300
5095 IF I$="D" OR I$="d" THEN LP
RINT "S$";: FOR F=8 TO 1 STEP -1
: LPRINT F;: FOR E=1 TO 8: LPRIN
T INVERSE FN I();B$(2,F,E);: NEX
T E: LPRINT : NEXT F: LPRINT " A
BCDEFGH"
5105 GO TO 5050
5110 GO SUB 6000: BEEP .02,36
5120 IF ERR=1 THEN GO SUB 9810:
PRINT AT J+6,0; FLASH 1;"?"
5130 IF ERR=2 THEN PRINT AT 9,5;
"END"; GO TO 5050
5140 NEXT J: NEXT I: LET ERR=2:
GO TO 5130
5200 GO TO 5050
5300 REM VARIANTE
5310 FOR F=1 TO 8: LET B$(3,F)=B
$(2,F): NEXT F: LET FELD=3: LET
J2=J: LET ER1=ERR
5320 PRINT AT 9,1;"W " ; " " S
": IF J>1 THEN LET J=0
5330 INPUT LINE N$
5332 IF N$(1)="W" OR N$(1)="w" T
HEN LET J=J=0: GO TO 5330
5333 IF N$="S " OR N$="s
" THEN GO SUB 3520: GO TO 5330
5335 IF N$(1)="Z" OR N$(1)="z" T
HEN LET J=J2: LET ERR=ER1: LET C
$="": PRINT AT 9,0;
C$;C$: LET FELD=2: GO SUB 3900:
GO TO 5050
5340 GO SUB 2500: LET M$=N$
5350 PRINT AT J+9,0;" ";AT J+9,7
;N$
5360 LET ERR=0: GO SUB 6044
5370 IF ERR=1 THEN GO SUB 9810:
PRINT AT J+9,0; FLASH 1;"?": GO
TO 5330
5380 LET J=J=0: GO TO 5330
6000 REM PER INTERPRETER
6030 LET M$=Z$(1,J+1): LET ERR=0
6040 LET N$="": IF M$=N$ THEN LE
T ERR=2: RETURN

```



```

6042 PRINT AT 6,0;" "" ";AT J+6
6043 I;" TAB 7;M$
6044 BEEP .01,24: PAUSE 0: LET I
$=INKEY$: IF I$<>"H" AND I$<>"h"
THEN LET J=J-1: RETURN
6044 LET J1=J: IF NOT J1 THEN LE
T J1=-1
6046 IF NOT=1 THEN GO TO 7000
6050 IF M$( TO 5)="0-0 " THEN G
O TO 6900
6050 IF M$( TO 5)="0-0-0" THEN G
O TO 6950
6070 IF M$(1)>"Z" THEN GO TO 800
0
6080 LET I$=M$(1): LET M$=M$(2 T
O )
6090 IF I$="T" THEN GO TO 8200
6100 IF I$="S" THEN GO TO 8400
6110 IF I$="L" THEN GO TO 8600
6120 IF I$="D" THEN GO TO 8800
6130 IF I$="K" THEN GO TO 9000
6900 REM MINZE ROCHSEN
6910 IF J THEN LET F$="e8": GO S
UB 4000: LET F$="c8k": GO SUB 40
00: LET F$="h8": GO SUB 4000: LE
T F$="f8t": GO SUB 4000: RETURN
6920 LET F$="e1": GO SUB 4000: L
ET F$="g1k": GO SUB 4000: LET F$
="h1": GO SUB 4000: LET F$="f1t"
: GO SUB 4000: RETURN
6950 REM MINZE ROCHSEN
6960 IF J THEN LET F$="e8": GO S
UB 4000: LET F$="c8k": GO SUB 40
00: LET F$="a8": GO SUB 4000: LE
T F$="d8t": GO SUB 4000: RETURN
6970 LET F$="e1": GO SUB 4000: L
ET F$="c1k": GO SUB 4000: LET F$
="a1": GO SUB 4000: LET F$="d1t"
: GO SUB 4000: RETURN
7000 REM NOTATIONEN
7010 LET I$=M$(4 TO 5)+B$(FELD,U
AL M$(2),CODE M$(1)-96): LET F$=
M$( TO 2): GO SUB 4000: LET F$=I
$: GO SUB 4000
7020 IF (F$(2)="1" OR F$(2)="8")
AND (F$(3)="b" OR F$(3)="B") TH
EN LET F$(3)=CHR$(CODE "D"+32*(
F$(3)="b"): GO SUB 4000
7030 RETURN
8000 REM BAUER
8010 LET FIG=CODE "B"+J*32: IF M
$(2)>"8" THEN GO TO 8150
8015 IF M$(2)<="8" AND M$(3)>="a
" AND M$(3)<="h" THEN LET F=VAL
M$(2): LET M$=M$(1)+M$(3): GO SU
B 8190: GO TO 8170
8020 GO SUB 8190: LET D=VAL M$(2
)
8025 IF B$(FELD,D+J1,C)<>" " THE
N GO SUB 8140: GO TO 8040
8030 LET D=D+J1: GO SUB 8140
8040 IF F$(2)<>CHR$(CODE "8"-7*
J) THEN RETURN
8045 IF M$(4)="T" OR M$(4)="S" O
R M$(4)="L" OR M$(4)="D" THEN LE
T F$(3)=CHR$(CODE M$(4)+J*32):
GO SUB 4000: RETURN
8050 LET M$(4)="D": GO TO 8045
8140 LET F$=M$(1)+STR$(D+J1): G
O SUB 4000: LET F$=M$( TO 2)+CHR
$ FIG: GO SUB 4000: RETURN
8150 GO SUB 8190
8155 FOR F=1 TO 8
8160 IF B$(FELD,F,C)<>CHR$ FIG T
HEN NEXT F: LET ERR=1: RETURN
8165 IF F-1<1 OR F+1>8 THEN LET
ERR=1: RETURN
8170 LET F$=M$(1)+STR$ F: GO SUB
4000
8175 IF B$(FELD,F-J1,D)=" " THEN
LET F$=M$(2)+STR$ F: GO SUB 400
0
8180 LET F$=M$(2)+STR$(F-J1)+CH
R$ FIG: GO SUB 4000: GO TO 8040
8190 LET C=CODE M$(1)-96: LET D=
CODE M$(2)-96: RETURN

```

```

8195 LET C=CODE M$(1)-96: LET D=
VAL M$(2): RETURN
8200 REM TURB
8205 LET FIG=CODE "T"+J*32
8210 IF M$(2)>"8" THEN GO TO 950
0
8220 GO SUB 8195
8230 LET E=CODE M$(1)-96: FOR F=
E TO 1 STEP -1
8240 LET C$=B$(FELD,D,F): IF F<>
E THEN IF C$<>CHR$ FIG THEN IF C
$<>" " THEN GO TO 8245
8242 IF C$=CHR$ FIG THEN GO TO 8
265
8244 NEXT F
8245 FOR F=E TO 8
8246 LET C$=B$(FELD,D,F): IF F<>
E THEN IF C$<>CHR$ FIG THEN IF C
$<>" " THEN GO TO 8252
8247 IF C$=CHR$ FIG THEN GO TO 8
265
8250 NEXT F
8252 LET E=VAL M$(2): FOR F=E TO
1 STEP -1
8253 LET C$=B$(FELD,F,C): IF F<>
E THEN IF C$<>CHR$ FIG THEN IF C
$<>" " THEN GO TO 8257
8255 IF C$=CHR$ FIG THEN GO TO 8
290
8256 NEXT F
8257 FOR F=E TO 8
8258 LET C$=B$(FELD,F,C): IF F<>
E THEN IF C$<>CHR$ FIG THEN IF C
$<>" " THEN GO TO 8262
8259 IF C$=CHR$ FIG THEN GO TO 8
290
8260 NEXT F
8262 LET ERR=1: RETURN
8265 LET F$=CHR$(F+96)+STR$ D:
GO SUB 4000
8270 LET F$=M$( TO 2)+CHR$ FIG:
GO SUB 4000: RETURN
8290 LET F$=M$(1)+STR$ F: GO SUB
4000: GO TO 8270
8400 REM SEMPER
8410 LET FIG=CODE "S"+32*J
8420 IF M$(2)>"8" THEN GO TO 950
0
8430 GO SUB 8195
8440 LET C$="": FOR E=-2 TO 2 ST
EP 4: FOR F=-1 TO 1 STEP 2
8450 LET G=D+E: LET H=C+F
8460 IF G>=1 AND G<=8 AND H>=1 A
ND H<=8 THEN GO SUB 8500: IF C$=
"ok" THEN GO TO 8550
8470 NEXT F: NEXT E: FOR E=-1 TO
1 STEP 2: FOR F=-2 TO 2 STEP 4
8480 LET G=D+E: LET H=C+F
8485 IF G>=1 AND G<=8 AND H>=1 A
ND H<=8 THEN GO SUB 8500: IF C$=
"ok" THEN GO TO 8550
8490 NEXT F: NEXT E: LET ERR=1:
RETURN
8500 IF B$(FELD,G,H)=CHR$ FIG TH
EN LET C$="ok"
8510 RETURN
8550 LET F$=CHR$(H+96)+STR$ G:
GO SUB 4000: LET F$=M$( TO 2)+CH
R$ FIG: GO SUB 4000: RETURN
8600 REM LEBENSER
8610 LET FIG=CODE "L"+J*32
8630 GO SUB 8195
8640 LET C$="": LET G=0
8645 FOR B=1 TO 8: FOR E=-1 TO 1
STEP 2: FOR F=-1 TO 1 STEP 2
8650 LET G=D+B*E: LET H=C+B*F
8660 IF G>=1 AND G<=8 AND H>=1 A
ND H<=8 THEN GO SUB 8500: IF C$=
"ok" THEN GO TO 8550
8670 NEXT F: NEXT E: NEXT B: LET
ERR=1: RETURN
8800 REM ORNE
8810 LET FIG=CODE "D"+J*32
8820 IF M$(2)>"8" THEN GO TO 950
0
8830 GO SUB 8630: IF ERR=0 THEN
RETURN

```

```

8640 LET ERR=0: GO TO 8230
9000 REM KORREKTUR
9010 LET FIG=CODE "K"+J*32
9020 GO TO 8830
9500 REM FIGUR REIHE SPALTE NACH
9510 IF M$(1) > "8" THEN GO TO 9600
9520 LET C=VAL M$(1): LET M$=M$(2 TO )
9530 FOR F=1 TO 8
9540 IF B$(FELD,C,F) <> CHR$ FIG THEN NEXT F: LET ERR=1: RETURN
9550 LET F$=CHR$(F+96)+STR$ C: GO SUB 4000: GO TO 9640
9600 LET C=CODE M$(1)-96: LET M$=M$(2 TO )
9610 FOR F=1 TO 8
9620 IF B$(FELD,F,C) <> CHR$ FIG THEN NEXT F: LET ERR=1: RETURN
9630 LET F$=CHR$(C+96)+STR$ F: GO SUB 4000
9640 LET F$=M$( TO 2)+CHR$ FIG: GO SUB 4000: RETURN
9800 LET B$(1,1)="TSLDKLST": LET B$(1,2)="88888888": LET B$(1,7)="88888888": LET B$(1,8)="tsldklst": FOR F=3 TO 6: LET B$(1,F)="": NEXT F: RETURN
9810 FOR F=1 TO 3: BEEP .1,24: NEXT F: RETURN
9900 REM START
9910 INK 5: PAPER 1: BRIGHT 1: OVER 1: INVERSE 0: FLASH 0: BORDER 1: CLEAR
9920 PRINT AT 2,15;" ";TAB 15;" ";TAB 13;" ";TAB 13;" ";TAB 15;" ";TAB 15;" ";TAB 10;" ";FOR F=1 TO 7: PRINT TAB 11;" ";NEXT F
9930 PRINT TAB 9;" ";TAB 8;" "
9940 LET F=0: LET G=11: FOR H=0 TO 30 STEP 3: PLOT 55+H,20+H/3:

```

```

DRAW 85,0: DRAW -15,10,1: DRAW 0,40
9950 DRAW 5,F: DRAW F,3: DRAW -27,F: DRAW F,G: DRAW G,F: DRAW F,G: DRAW -G,F: DRAW F,G
9955 DRAW -G,F: DRAW F,-G: DRAW -G,F: DRAW F,-G: DRAW G,F: DRAW F,-G
9960 DRAW -27,0: DRAW 0,-3: DRAW 5,0: DRAW 0,-40: DRAW -15,-10,1
: NEXT H
9970 OVER 0: INK 7: FOR F=0 TO 10: PLOT 50,170: DRAW -15,-25,F/15-PI: NEXT F: PLOT 30,130: PLOT 75,160: PLOT 180,145
9980 RESTORE : FOR F=USR "A" TO USR "U"+7: READ E: POKE F,E: NEXT F: GO SUB 0: GO SUB 9800: PAPER 7: INK 0: BORDER 7: GO TO 1000
9990 DATA 7,7,7,7,31,63,63,0,224,224,224,248,252,252,0
9991 DATA 0,0,1,3,7,3,1,0,0,0,128,192,224,192,128,0
9992 DATA 0,0,0,0,0,0,0,0,0,0,25,25,31,15,7
9993 DATA 0,0,0,152,152,248,240,224,3,3,3,7,7,15,31,0
9994 DATA 192,192,192,224,224,240,248,0,0,1,3,7,15,31,15,7
9995 DATA 0,128,192,128,48,248,240,224,1,1,7,7,1,1,31,15
9996 DATA 128,128,224,224,128,128,248,240,15,15,15,15,63,127
9997 DATA 0,240,240,240,240,240,252,254,0,0,1,1,1,5,14,31,28
9998 DATA 0,32,192,192,240,240,240,240,0,1,3,3,3,7,15,0
9999 DATA 240,240,224,224,224,224,248,0,1,35,33,63,63,7,15,7,128,195,132,252,252,224,240,224

```

Listing des Schach-Editors (Schluß)

Nachspiels Fehler entdeckt haben (Sinn-Fehler), ins Korrekturmenü.

## Einfacher als man denkt

Es ist nicht so kompliziert, wie es sich anhört. Wenn Sie erst einmal eine Partie eingegeben und nachgespielt haben, werden Sie merken, daß die Menü-Technik hervorragend geeignet ist, viele Funktionen in ihrer logischen Reihenfolge aufzurufen.

Manchmal hat man das Diagramm einer unvollendeten Schachpartie vor sich und darunter steht dann »wie gewann Schwarz?« (oder Weiß). Der Schach-Editor macht seinem Namen alle Ehre, denn durch Drücken der Taste 4 gelangt man zum »Aufbau«. Die aus dem Menü wählbaren Funktionen sind: Herstellen der Ausgangsposition (a), Abräumen des Brettes (l = löschen) und Zurückgehen ins Hauptmenü. Zuerst werden Brett (16 x 16 Zeichen) und Figuren gezeichnet. Dann erscheint eine INPUT-Zeile, und man kann entweder eine der drei oben genannten Funktionen ausführen, oder ein Feld auf dem Brett mit einer Figur besetzen. Dazu gibt man erst das Feld (zum Beispiel e4), dann die Figur als Großbuchstabe (D = Dame, B = Bauer und so weiter), und abschließend die Farbe (s für schwarz, bei weiß kann w entfallen) ein. Ein Feld kann gelöscht werden, indem nur seine Koordinaten eingegeben werden. Mit z kehren Sie ins Hauptmenü zurück.

Betrachten wir nun die sicher interessanteste Wahlmöglichkeit. Punkt 6 (Partie nachspielen). Nach Drücken auf 6 wird wieder das Brett gezeichnet und die Figuren werden aufgestellt. Links unten ist das Menü mit sieben Wahlmöglichkeiten

sichtbar, von denen zwei nur nach Drücken auf v (= Variante) zugänglich sind (w = Wechsel, s = Stellen). Doch zuerst sollen die anderen Funktionen erläutert werden. Das Brett kann mit r (= Richtung) umgedreht werden, so daß die schwarzen Steine unten sind. Wenn man auf h drückt, wird links oben (neben einem w für weiß und einem s für schwarz) der als nächstes auszuführende Halbzug und seine Zugnummer angezeigt. Nochmaliges Drücken von h bringt diesen Halbzug zur Ausführung. Mit d wird das Brett mit dem aktuellen Stand auf dem Drucker ausgegeben. Allerdings nicht als Hardcopy, sondern als verkleinertes Diagramm, das Großbuchstaben zur Kenntlichmachung der weißen und Kleinbuchstaben für die schwarzen Steine verwendet. Nun zur Taste v (= Variante). Wenn Sie diese Taste betätigt haben, erscheinen ein weiteres w und ein weiteres s. Sie können nun »von Hand« weitere Halbzüge eingeben, die zwar die Stellung auf dem Schirm verändern, nicht aber die Stellung, die vor dem Drücken auf v aktuell war.

Beachten Sie bitte, daß das Programm »weiß«, welche Seite am Zug war oder ist. Innerhalb einer Variante können Sie dies mit w (= Wechsel) ändern (das heißt Seiten vertauschen). Jeder in diesem Modus eingegebene Zug wird auf Durchführbarkeit überprüft und muß gegebenenfalls korrigiert werden (automatischer Sprung ins Korrektur-Menü). Sie können auch — innerhalb einer Variante — mit s (= Stellen) Figuren hinstellen oder entfernen. Dies funktioniert wie bei Punkt 4 des Hauptmenüs (Stellung aufbauen). Mit z kehren Sie in das jeweils nächsthöhere Menü zurück. Von »Stellen« kommen Sie damit zu »Variante« und von dort in den Nachspiel-Modus. Wenn Sie in letzteren Modus zurückgekehrt sind, werden die Figuren so



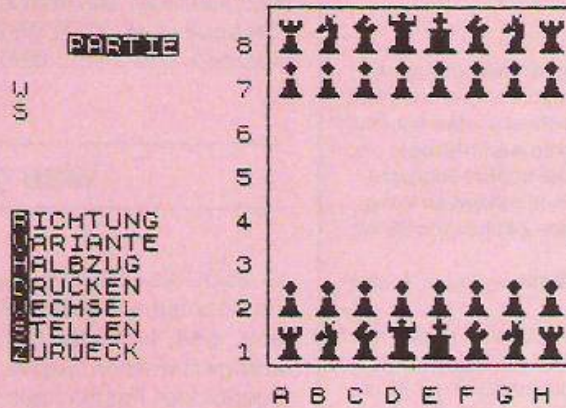
Mit dem Spectrum  
wird das Nachspie-  
len »tragbarer«



Das grafisch gut  
gestaltete Titelbild  
des Editors

Seppeur-Spasski  
(Ponziani-Eröffnung)

Grundstellung vor  
der Eröffnung



Seppeur-Spasski  
(Ponziani-Eröffnung)



Aktuelle  
Spielszene mit  
Protokoll

SCHACH  
EDITOR

**WAHLMÖGLICHKEITEN**

- 1 ZUEGE EINGEBEN....
- 2 ZUEGE AUFLISTEN...
- 3 ZUG KORRIGIEREN...
- 4 STELLUNG AUFBAUEN.
- 5 EINGABE FORTSETZEN
- 6 PARTIE NACHSPIELEN
- 7 AUF BAND SPEICHERN
- 8 VOM BAND LADEN....

Die Qual der Wahl:  
Das Menü mit  
8 Optionen

## Variablentabelle

BS(4,8,8)	Schachbretter und Titel der Partie. Schwarze Figuren liegen als Kleinbuchstaben (b,t,s,l,d,k) vor, weiße als Großbuchstaben (B,T,S,L,D,K).
Z\$(65,2,6)	Zug-Speicher.
MS(6),NS(6)	Zwischenspeicher für Züge.
A\$, H\$	Steuerzeichen für Brett-Farben.
C\$, I\$	Strings für vielerlei Aufgaben. Unter anderem INKEY\$.
P\$	Flag für Drucker an/aus.
T\$(64)	Eingabe-String für den Titel der Partie.
SS	Name der Datei.
NOT	Notation. 1 für ausführlich, 2 für Kurzform.
ORI	Orientierung. 1, wenn weiß unten, 0, wenn schwarz unten ist.
ZNR	Zugnummer.
FLAG, FELD, F\$(4)	FELD entscheidet darüber, mit welchem der drei Schachbretter gearbeitet wird. FLAG entscheidet, ob sämtliche Figuren gezeichnet werden oder nur eine. F\$ sieht zum Beispiel so aus: a1Ks oder a1k. Der String wird an die Zeichenroutine übergeben, die dann auf das Feld a1 einen schwarzen König stellt und das entsprechende Zeichen von B\$ mit k lädt.
B,C,D,E,F,G,H,S,W	Codes von Buchstaben, Brettkoordinaten, Laufvariable und so weiter.
FIG,J,J1	Für schwarz am Zug gilt J=1, J1=1. Für weiß am Zug gilt J=0, J1=-1. Gemäß der Eigenschaft, daß der Code eines Kleinbuchstabens um 32 größer ist als der des entsprechenden Großbuchstabens, wird die Variable FIG (Figur) mit dem Code der ziehenden Figur als Großbuchstabe plus 32 mal J geladen. Beispiel: Ein schwarzer Bauer ist am Zug — also ist J=1. FIG=CODE »B« + 32xJ (=CODE »b«!).
ERR	Error. Der Interpreter gibt Fehlermeldungen an den Nachspiel- oder Varianten-Modus. 1 für »Zug ist nicht ausführbar«, 2 für »keine weiteren Züge mehr da«, 0 für »alles in Ordnung«.
ERR1, J2	Zwischenspeicher für ERR und J1, wenn im Varianten-Modus gearbeitet wird.

aufgebaute Schachbretter aufnehmen kann. Tatsächlich werden davon aber nur drei benötigt. Eines für die Ausgangsstellung, eines für den Nachspielmodus und eines für den Variantenmodus. Das vierte hält den Titel der Partie — mit 8 x 8 = 64 Zeichen, also besagte zwei Zeilen. Eine ökonomische Lösung, wie ich finde.

So konnten sämtliche für die Partie wichtigen Daten in nur zwei String-Feldern untergebracht werden. Über andere Variablen gibt die Tabelle Auskunft. Nun zu wichtigen Programmteilen.

— Die Zeilen 2160 bis 2290 nehmen während der Zugeingabe die Züge an und stellen fest, ob die maximale Anzahl erreicht ist oder ob der Benutzer einen Leerstring eingegeben hat. Da es sich um dimensionierte Strings handelt, hat ein Leerstring natürlich immer eine festgelegte Anzahl an Spaces.

— Die Zeilen 2300 bis 2490 holen sich jeden einzelnen Zug nacheinander aus dem Zugspeicher und übergeben ihn an die Testroutine ab Zeile 2500. Zeile 2340 verhindert die Frage »scroll?« und macht dafür eine Pause.

## Was geschieht wo

— 2500 bis 2980: Syntax-Test. Jeder Zug wird untersucht im Hinblick auf seine Schreibweise. Fehleingaben wie T13a oder sh9, gA4, Lbf oder 33 werden herausgefiltert und müssen korrigiert werden. Bei Notation 1 kann jede beliebige Figur von irgendeiner Position auf dem Brett zu einer anderen ziehen.

— Die Zeilen 3000 bis 3110 stellen unter anderem eine Korrektur-Routine dar, für den Fall, daß nicht das Programm, sondern der Benutzer einen Fehler entdeckt hat. Für Fehler, die während des Syntax-Tests entdeckt werden, hält diese Routine eine eigene Korrekturmöglichkeit bereit.

— In den Zeilen 3500 bis 3600 werden alle Buchstaben entgegengenommen, die in dem Menü von »Stellung aufbauen« möglich sind. Siehe Variablentabelle unter F\$.

— Die Zeilen 3800 bis 3820 zeichnen das Brett (ohne Figuren). Die Variable I\$ enthält hier neun mal zwei Spaces, denen abwechselnd die Steuerzeichen 3 und 5 für die hellen und dunklen Quadrate auf dem Brett vorgeschaltet sind. Diese Steuerzeichen, die in Zeile 3805 vorerst in A\$ und H\$ abgelegt werden, können geändert werden, um dem Brett andere Farben zuzuordnen.

— Die Zeilen 3830 bis 3850 geben dem Brett die Koordinatenbezeichnung. Siehe Variablentabelle unter »ORI«.

— Von 3900 bis 3995 geht die Routine, die die Figuren — alle auf einmal oder einzeln — auf das Brett stellen kann. Die wichtigste von ihr verwendete Variable ist F\$.

— Die Zeilen 4000 bis 4090 sind dafür zuständig, daß einzelne Figuren stets richtig auf das Brett gestellt werden.

— 5000 bis 5200: Partie nachspielen. Unter anderem wird der Interpreter aufgerufen. Seine Fehlermeldungen werden verarbeitet. Hier befindet sich auch die Druckroutine, die ein einfaches Diagramm des Schachbretts auf dem ZX-Printer ausgibt.

— In den Zeilen 5300 bis 5380 werden die meisten Variablen aus dem Nachspielmodus zwischengespeichert. Unter anderem kann dann der Interpreter aufgerufen werden, ohne daß diese Variablen verloren gehen.

— 6000 bis 9020: Der Interpreter, das Herzstück. Hier werden die einzelnen Züge interpretiert. Bei der ersten Notation ist das ganz einfach: Feld, von dem gezogen wird, löschen. Figur, die vorher darauf stand, auf Zielposition stellen.

Bei der Kurznotation ist es unvergleichlich komplizierter. Ab Zeile 8000 wird jede Figur — entsprechend ihrer »Gangart« — bewegt.

hingestellt wie sie vorher waren, und Sie können weiter nachspielen. Ist der letzte Zug erreicht, erscheint das Wort »ENDE« auf dem Bildschirm.

Kommen wir jetzt zu dem Teil des Hauptmenüs, der das Speichern auf Band und das Laden vom Band ermöglicht. Wird die Taste 7 betätigt, dann wird nach dem Namen gefragt, den die Datei erhalten soll. Anschließend nach dem Titel der Schachpartie. Der kann bis zu 64 Zeichen — also zwei Zeilen — lang sein. Nach ENTER erscheint zweimal kurz hintereinander die bekannte Meldung »Start tape, then press any key«. Das Abspeichern dauert nur wenige Sekunden. Wenn Sie vom Band laden wollen, dann drücken Sie auf 8, geben den Dateinamen ein (kann auch ein Leerstring sein), und die entsprechende Datei wird in den Speicher geladen.

Es werden zwei String-Felder auf Band gebracht. (Das Programm erwartet übrigens, daß Sie zurückspulen und verifizieren.) Das erste Feld ist der Zugspeicher Zs(65,2,6). 64mal zwei Züge zu je sechs Zeichen haben dort Platz. Speicher Nummer 65,1 bleibt leer, damit das Ende auch bei vollem Speicher noch erkannt werden kann. Nummer 65,2 hält die Notation und die Anzahl der Züge, so daß mit Funktion 5 des Hauptmenüs eine angefangene und eventuell abgespeicherte Eingabe zu späterem Zeitpunkt fortgesetzt werden kann. Das zweite Feld ist BS(4,8,8). Es ist der Brett-Speicher, der vier

**Der Bauer:** Bei Vorwärtszügen wird überprüft, ob auf dem Feld vor der Zielposition ein Bauer steht. Wenn ja, wird dieser gelöscht; wenn nein, wird diejenige Figur gelöscht, die zwei Felder vor der Zielposition steht. Bei Seitwärtszügen dagegen wird überprüft, ob auf der Zielposition eine Figur steht. Wenn nein, wird die Figur auf dem Feld hinter der Zielposition gelöscht (en passant). Das Programm »weiß« nicht, wo en passant geschlagen werden darf, »sieht« aber, wenn ein Zug dieser Art auftaucht.

## Die Spielfiguren

**Der Turm:** Von der Zielposition ausgehend wird untersucht, ob von links, rechts, oben oder unten mit einem Turm auf dieses Feld gezogen werden kann, ohne eine andere Figur überspringen zu müssen (das ist für den Menschen selbstverständlich, nicht aber für einen Computer). **Der Läufer:** Wie beim Turm, jedoch Betrachtung der Diagonalen. **Der Springer:** Der Computer »schaut nach«, ob mit einem Springer auf die vorgegebene Zielposition gesprungen werden kann. Es wird nicht jede Position einzeln überprüft (es sind acht), sondern zwei Schleifen übernehmen diese Aufgabe. **Dame und König:** Die Bewegungen dieser Figuren setzen sich aus denen von

Turm und Läufer zusammen. Also werden auch die Routinen dieser Figuren aufgerufen. Das Programm »merkt« nicht, wenn ein König sich anmaßt, genauso weit ziehen zu wollen wie eine Dame. Aber das ist auch nicht Aufgabe dieses Programms. Der Interpret, von dem hier die Rede ist, reagiert auf folgende Fehler mit einem blinkenden Fragezeichen (der Zug wird ignoriert): Eine Figur versucht unerlaubterweise über eine andere hinwegzuspringen (ausgenommen Bauernzüge); oder eine Figur steht nicht dort, wo sie stehen sollte, um ihre Zielposition erreichen zu können; oder ist ganz und gar nicht vorhanden (abermals sind Bauern ausgenommen).

Wie gibt man das Programm ein?

— Für die Graphics benutzen Sie bitte die Graphics-Tabelle (1).  
— Zeichen, die im Listing invers gedruckt sind, wurden mit INV. VIDEO vor dem ersten und TRUE VIDEO nach dem letzten Zeichen erzeugt.

— Verwechseln Sie bitte nicht die Variable NOT (Notation) mit der öfter verwendeten logischen Funktion NOT.

— Nach dem Eintippen drücken Sie bitte CLEAR. Bringen Sie das Programm mit SAVE »SCHACH-ED.« LINE 9900 auf Ihre Kassette. Noch ein Tip zum Abspeichern von Schachpartien: Sollte beim Verifizieren ein Fehler auftauchen, dann können Sie mit GO TO 1180 den Verifizierungsvorgang wiederholen oder mit GO TO 1000 ins Hauptmenü zurückkehren.

(Andreas Viebke)

## MAIL-SHOP

### Computer-Peripherie

Inh. Georg Zeulner  
Alsterdorfer Str. 201  
2000 Hamburg 60



Aus unserem Lieferprogramm bieten wir an:

Preise  
inkl. 14% MwSt.

MATRIXDRUCKER für SPECTRUM mit ZX-INTERFACE-1  
SEIKOSHA GP-100AS mit 2k-Druckspeicher + RS-232C ... DM 598,-  
SEIKOSHA GP-550A mit RS-232C für EDV-Papier und Einzelblatt ... DM 998,-  
SEIKOSHA GP-700A color (V.24) für EDV-Papier und Einzelblatt ... DM 1598,-  
Datenleitung/Drucker-Kabel ... DM 98,-  
für SPECTRUM ohne ZX-Interface-1  
SEIKOSHA GP-50S ... DM 399,-  
SEIKOSHA GP-500A mit Interface und Kabel (COPY, LLIST + LPRINT) ... DM 848,-  
SEIKOSHA GP-550A mit Interface und Kabel (COPY, LLIST + LPRINT) ... DM 1098,-  
SEIKOSHA GP-700A mit Interface und Kabel (COPY, LLIST + LPRINT) ... DM 1558,-

Mit unserem Interface können Sie den Bildschirm in 2-facher Größe ausdrucken lassen!

MICROSCAN MS-15  
Schönheitschreib-Drucker  
Typenradrunder (Olympiatypen) ... Pr. a. A.  
Drucker für SINCLAIR ZX-81:  
GP-500S mit Adapt. für ZX-81 ... DM 429,-  
GP-500A mit Interface + Kabel ... DM 748,-  
GP-550A mit Interface + Kabel ... DM 1098,-  
GP-700A mit Interface + Kabel ... DM 1498,-  
Profi-Monitor für SPECTRUM + ZX-81:  
MICROSCAN MS-1264A bernst. 12" ... DM 379,-  
MICROSCAN MS-1265A grün 12" ... DM 399,-  
Dreh- und Kippgehäuse mit Standfuß

Schreiben Sie uns oder rufen Sie einfach an.  
**040/511 76 03**  
Die heiße Nr. für Ihren Computer!

## SPECTRUM Soft- und Hardware

**DAS**  
Textverarbeitungs-  
System für Ihren  
ZX Spectrum 48 K.

**INES**  
**DM 49.90**

(inkl. deutscher  
Anleitung)

exklusiv bei **ULTRASOFT**  
Kamperweg 167  
4000 Düsseldorf 12  
(Händleranfragen erwünscht)

**ZX Interface I**  
+ **Microdrive**  
zum  
**ULTRA-**  
Preis für **DM 488.00**

Außerdem die besten  
Programme für Ihren  
Spectrum. Fordern Sie  
unser Info an!!

**ULTRASOFT**  
Kamperweg 167  
4000 Düsseldorf 12

PROGRAMME FÜR ALLE  
sinclairs !

**SUPERSPIELE FÜR DEN ZX81**  
**ALLE ULTIMATE-SPIELE**  
**FÜR DEN SPECTRUM - ZUM**  
**PREIS VON NUR NOCH 22,- !**  
**UND VIELE ANDERE ARCADE-GAMES**

**UND FÜR DEN QL**  
**UNSER ERÖFFNUNGS-ANGEBOT**

**FORDERN SIE UNSEREN**  
**INFO-PROSPEKT AN!**  
**RECHNER ANGEBEN!**

**EMM SOFTWARE**  
BORIS BAGINSKI + PETER STIEDA  
SPONECKSTR. 8 / 8000 MÜNCHEN 60

LANGWEILT SIE IHR COMPUTER?  
OBERBRAUCHEN SIE WAS BESONDERES?  
↑ WIR SCHREIBEN DAS PRO-  
GRAMM, DAS SIE BRAUCHEN!  
BUSINESS, UNTERHALTUNG, AUF  
HOME-COMPUTERN ODER PC;  
FÜR DEN EIGENBEDARF ODER  
LIZENSIERT ZUM WEITERVER-  
KAUF → ALLES BEI **EMM**

## ABC-Elektronik

1) ZX-Spectrum, 48 K, Heimcomputer	Speichererweiterung, 16-48 K	89,-
	Speichererweiterung, 16-80 K	179,-
2) ZX-Microdrive-Massenspeicher	dk-tronics-Tastatur für ZX81 und Spectrum (Microdrive ge.)	169,-
3) ZX-Interface; 1 RS232/Netzwerk	Joystickinterface (Kempston)	
4) ZX-Interface; 2 Joystick/ RAM-Modul	+ Quick Shot 2	79,-
	RS232-Kabel für Interface 1	39,-
Cartridge 1 St. = 18,- 5 St. = 90,-	Floppy Disk-System für Spectrum,	
ZX-QL, der Supercomputer, 128 KRAM	397 K Test-Happy-Computer/CP	1199,-
2 x Microdrive; 2 x RS232; RGB	Floppy-Controller arbeitet mit	
1799,-	jedem 5 1/2"-Shugart-Laufw.	480,-
Centronics-Schnittstelle für Spectrum		
keine Software nötig! Kann COPY 4 x		
vergrößern! — Seikosha/Epson		179,-

Bestellungen (Scheck/Nachn.) an

**Andreas Budde;** Am Brodhagen 100, 48 Bielefeld 1;  
Telefon 05 21/89 03 81 (15.00-20.00), Telex 9 32 974 buude d



Wer Programme in Maschinensprache für den ZX81 schreibt und sie dann verkaufen möchte, hat sich sicher schon einmal gefragt, wie er das Programm sicher vor Raubkopien schützen kann. In diesem Artikel schlage ich Ihnen eine Methode vor, die den Softwarepiraten einige graue Haare wachsen lassen wird.

Sehen wir die Sache mal mit den Augen eines Hackers. Wir nehmen eine Kassette mit einem Programm in Maschinensprache, das wir kopieren möchten. Zuerst laden wir das Programm normal mit LOAD ''. Der erste Versuch ist die BREAK-Taste. Einige Programmierer verwenden in ihren Programmen auch Routinen des Betriebssystems. Manche davon kann man mit BREAK überlisten. Die Routine COPY gehört zum Beispiel dazu. Mißlingt dieser Versuch, drücken wir ein paar Tasten (möglichst solche, die nicht erlaubt sind) und suchen so einen Programmierfehler, der das Programm stoppt. Eine Methode, die erstaunlich oft zum Ziel führt! Haben wir immer noch keinen Erfolg, versuchen wir, das Programm direkt nach dem Laden zu stoppen. Wir schalten den Computer kurz aus und geben im FAST-Modus den Befehl RAND USR 836 ein. Dies ist ein Trick, der jedes Programm lädt, aber es nachher auch dann nicht ausführt, wenn das Programm mit Autostart geSAVEt wurde. Spätestens jetzt haben wir das Programm sichtbar im Speicher und können es beliebig oft kopieren.

Die Methode, die ich hier vorstellen möchte, verhindert den letzten, scheinbar immer funktionierenden Trick. Die beiden zuerst genannten Möglichkeiten, ein Programm zu knacken, müssen Sie selber durch geeignete Maßnahmen verhindern.

## Safety first

Gehen wir davon aus, daß der Hacker unser Programm im Speicher hat. Nun kann er damit nichts anfangen, weil unser Programm verschlüsselt ist! Um Daten zu verschlüsseln, gibt es verschiedene Methoden und Verfahren. Eines der sichersten Verfahren ist heute das Verschlüsseln mit dem DES-Algorithmus (Data Encryption Standard). So viel Aufwand zu treiben, lohnt sich aber für ein einfaches Programm meistens nicht. Eine wesentlich einfachere, aber noch hinreichend sichere Methode ist, die Daten mit einem, dem Knacker unbekanntem Byte einer XOR-Verknüpfung zu unterziehen. Die XOR-Funktion hat nämlich die Eigenschaft, daß ein Byte, zweimal mit einem beliebig anderen, aber beidesmal gleichen Byte verknüpft, wieder den alten Wert annimmt. Sehen wir uns zuerst die Wahrheitstabelle für die XOR-Funktion an:

	0	1
0	0	1
1	1	0

Beispiel:

**Originalbyte** 76H = 01110110B  
**XOR** F2H = 11110010B

**ergibt** 84 H = 10000100B

**nochmal** F2H = 11110010B

**ergibt unser** Originalbyte 76H = 01110110B

Das Prinzip ist nun klar: Wir verschlüsseln mit dieser XOR-Funktion unser Programm vor dem Aufzeichnen mit einem, nur uns bekannten Byte. Vor dem Start des Programms entschlüsseln wir das Programm dann mit der gleichen Funktion und dem gleichen Byte. Wenn der Hacker nach dem Entschlüsseln keinen Zugriff mehr zum Programm hat, haben wir was wir wollen. Das einzige Problem ist jetzt, woher nehmen wir ein Byte, das niemand kennt, das aber vor und nach dem SAVEn des Programms noch das gleiche ist, sich aber auch verändert. In den Systemvariablen des ZX81 finden wir ein geeignetes Byte. Es ist dies der Frame-Counter in der Adresse 4034H. Diese Systemvariable summiert die Anzahl Bilder, die erstellt werden. Im SLOW-Modus sind es 50 pro Sekunde. Solange der ZX81 im FAST-Modus kein Display erzeugt, bleibt der Wert unverändert. Außerdem wird die Variable beim SAVEn mit auf Band aufgenommen. Sie ist schlichtweg ideal für unseren Zweck.

Nach der Theorie nun die Programmstruktur im Einzelnen:

1. Aufruf der Entschlüsselungsroutine, die gleichzeitig auch Verschlüsselungsroutine ist.
2. SLOW-Modus einschalten
3. Hier ist das Programm, das Sie schützen möchten. Es sollte keine Möglichkeit geben, das Programm zu beenden, ohne daß der Speicher radikal gelöscht wird (mit NEW)! Falls Sie eine solche Möglichkeit einbauen, denken Sie immer daran, daß auch ein Hacker mal diese Taste drücken könnte!
4. FAST-MODUS einschalten. Von jetzt an verändert sich der Frame-Counter nicht mehr.

5. Aufruf der Verschlüsselungsroutine. SAVEn des Programms mit Autostart.

Das folgende Demonstrationsprogramm können sie direkt eingeben und sich davon überzeugen, daß das Hauptprogramm wirksam verschlüsselt wird. Da die Routinen FAST und SLOW benützt werden, müssen sie erst an Ihrem ZX81 testen, welche ROM-Version Sie haben. Je nach ROM-Version befinden sich diese Routinen an einer anderen Stelle im Betriebssystem. Geben Sie dazu RAND USR 3872 ein. Nun sollte Ihr ZX81 im FAST-Modus sein. Tippen Sie jetzt RAND USR 3880 ein. Nun sollten Sie wieder im SLOW-Modus sein. Ist dies der Fall, so haben Sie eine ältere Version des ROMs (was aber nichts ausmacht!). Ersetzen Sie aber im Listing die unterstrichenen Zahlen durch neue: Statt 230F nehmen Sie 200F und statt 2B0F nehmen Sie 280F.

## Programm-Eingabe

Zuerst müssen sie eine REM-Zeile mit genau 179 Leerzeichen erzeugen. Wenn Sie das im FAST-Modus machen, geht es wesentlich schneller als im SLOW-Modus! Zur Kontrolle geben Sie PRINT PEEK 16511 ein. Es muß als Resultat 181 herauskommen. Um den HEX-Code einzugeben, gibt es verschiedene Methoden. Eine davon habe ich als Beispiel gewählt:

```
1 REM ...197 Leerschläge...
2 FOR A = 16514 TO 16656
3 INPUT A$
4 POKE A,16*(CODE
  A$(1)-28)+CODE A$(2)-28
5 NEXT A
```

Wenn Sie nun alle Hex-Zahlen eingetippt und keinen Fehler gemacht haben, löschen sie die Linien 2 bis 5. Geben sie die folgenden Zeilen neu ein:

```
2 SAVE "TEST"
3 RAND USR 16514
4 RAND USR 16596
```

16514 ist die Startadresse des verschlüsselten Programms. Nach jedem Laden wird die Basic-Zeile 3 als erstes ausgeführt. 16596 ist die Startadresse des unverschlüsselten Programms. Hier wird das Programm nur beim allersten Mal gestartet.

Bevor Sie das Programm nun mit RUN 4 starten, empfehle ich Ihnen, eine Sicherheitskopie auf Band aufzunehmen. Vielleicht haben Sie sich doch irgendwo vertippt.

```

CD 8B 40      CALL CRYPT      ; Aufruf der
                                   ; Verschlüsselung

CD 2B 0F
C3 D4 40      CALL SLOW
                                   ; zum Hauptprogramm
                                   ;
2A 34 40      CRYPT: LD HL,(FRAMES)
7C            Ld A,H
AD            XOR L              ;Verknüpfung der
                                   ; beiden FRAME-Bytes
                                   ; zu einem Byte

01 63 00      LD BC, ENDPROG-BEGINPROG
                                   ; Länge des Programms

21 A1 40      Ld HL, BEGINPROG
                                   ; Start der
                                   ; Verschlüsselung

57            LD D,A
7A            CRYPT$: LD A,D
AE            XOR (HL)          ; verschlüsselt
77            LD (HL),A         ; und abgespeichert
23            INC HL
0B            DEC BC
78            LD A,B
B1            OR C
20 F7        JR NZ, CRYPT$1     ; noch ein Byte?
C9            RET               ; Nein -> zurück

BEGINPROG:   ; hier beginnt das zu
               ; schützende Programm

3A 33 30 34 35 2E 2A 37
27 26 37 2A 38 00 35 37
34 2C 37 26 32 32 00 00
00 00 00 00 00 00 00 00
00 3B 34 33 00 35 2A 39
2A 37 00 39 2E 2A 38 33
2A 38 1B
               ; Text. Gehört zum Demoprogramm:

               .DATA /UNKOPIERBARES/
               .DATA /PROGRAMM/
               .DATA /VON PETER TIESNES/

3E 00        STARTPROG: LD A,0
32 3C 40      LD (403C),A       ; Indikator für
                                   ; invers

CD 2A 0A      PROG$: CALL CLS
21 A1 40      LD HL,BEGINPROG
06 33        LD B,33H          ; Anzahl Text-Bytes
3A 3C 40      PROG$: LD A, (403C) ; invers oder normal
AE            XOR (HL)
D7            RST 10           ; Ausdruck
23            INC HL
10 F8        DJNZ PROG$2      ; noch ein Byte?
3A 3C 40      LD A, (403C)     ; Nein
EE80         XOR 80H          ; Indikator wechseln
32 3C 40      LD (403C),A
ED 4B 25 40  PORG$: LD BC,(LASTKEY)
03            INC B
78            LD A,B
B1            OR C
28 F7        JR Z,PROG$3      ; Taste gedrückt?
0B            DEC BC          ; Ja
CD BD 07      CALL DECODE
7E            LD A,(HL)        ; A := Code
EE 38        XOR 38H          ; ist es 'S'?
C2 D9 40      JP NZ,PROG$1     ; Nein -> PROG$1

ENPROG:      ; Ende des Hauptprogramms
               ;
CD 23 0F      CALL FAST
CD 8B 40      CALL CRYPT
21 36 41      LD HL,4136H      ; Adresse der Basic-
                                   ; Linie, die nach dem
                                   ; RET ins Betriebs-
                                   ; system ausgeführt
                                   ; werden soll.
22 29 40      LD (NXTLIN),HL  ; abspeichern in der
                                   ; entsprechenden
                                   ; Systemvariablen.
C9            RET             ; zurück ins
                                   ; Betriebssystem.

```



Dieses sehr einfache Demo-Programm schreibt lediglich bei jedem Tastendruck einen Text einmal normal und anschließend invers auf den Schirm. Wenn Sie aber die Taste »S« drücken, speichert sich das Programm einmal ab und kehrt wieder zu sich selbst zurück. Drücken sie also nun mal die Taste »S« und BREAKen Sie das Programm, sobald die breiten, schwarzen horizontalen Streifen auf dem Bildschirm erscheinen. Im Bildschirm haben Sie nun die verschlüsselte Version des Programms, und da, während Sie das Programm betrachten, der Frame-Counter weitergezählt hat, können Sie es niemals mehr entschlüsseln und starten. Hoffentlich haben Sie vorher eine unverschlüsselte Kopie auf Band aufgenommen. Sie können auch das Programm auf Band aufnehmen und es mit RAND USR 836 wieder laden. Obwohl Sie das Programm einwandfrei laden und dann kopieren können, ist es höchst unwahrscheinlich, daß Sie jemals eine funktionsfähige Version des Programms erhalten werden.

Wie gut ist nun dieser Schutz? Ich behaupte, für den geringen Aufwand haben wir einen Programmschutz erhalten, der fast optimal ist. Es gibt nur noch eine Methode, dieses Programm zu vervielfältigen: Man muß die Kassette analog von einem Kassettenrecorder zum anderen überspielen. Da der ZX81 aber ziemlich heikel ist, was die Qualität der Aufnahme angeht, weiß ich aus eigener Erfahrung, daß ein Erfolg mit dieser Methode sehr selten ist.

(Peter Tiesnes)



# Ein Autostart - auch für ZX 80

**L**eider haben alle drei Verfahren einen ganz wesentlichen Nachteil — beim Starten (und beim Absturz) muß man Hand anlegen:

\* bei Maschinenprogrammen im EPROM mit RAND USR in diese Programme einspringen

\* bei Basic-Programmen laden und mit RUN oder GOTO neu starten.

Bei Verwendung eines »Autostart« springt der Rechner beim Einschalten automatisch, ohne zusätzlichen Handgriff, in ein Programm in einem EPROM — daher der Name.

## Erster Schritt: eine ganz simple EPROM-Platine

Es gibt bereits einige EPROM-Karten. Wir haben einen Bausatz von Decker geprüft. Für die Steuerung suchte er einen neuen Weg. Dazu wird die Lücke zwischen Basic-ROM und RAM im Bereich von 8 KByte bis 16 KByte in vier Blöcke zu je 2 KByte (für EPROMs vom Typ 2716) oder zwei Blöcke zu je 4 KByte Länge (für 2732) zerlegt. Bei einem 2764 brauchen wir die vollen 8 KByte.

Um die drei verschiedenen EPROMs im gleichen Sockel einsetzen zu können, werden die Anschlüsse mit Drahtverbindung dem jeweiligen IC-Typ angepaßt. Die Platine ist für eine Bestückung mit 2716 geätzt; bei 2732 oder 2764 ist der Umbau eine Sache von zwei Minuten.

Damit ist die EPROM-Platine bereits fertig. Sie kann natürlich auch ohne den Autostart betrieben werden.

## Zweiter Schritt: der eigentliche Autostart

Der ZX gibt ein mehr oder weniger brauchbares Signal ab: RESET\*. Beim Einschalten der Stromversorgung oder beim Betätigen des Reset-Knopfes wird ein Kondensator über einen Widerstand geladen. Hat die Spannung am Kondensator einen gewissen Schwellwert erreicht, so gibt sie den Prozessor frei. Diese Spannung wird auch auf den

**Bei dieser Überschrift stellt man sich als erstes die Frage: Was ist das eigentlich, ein »Autostart«? Nehmen wir an, Sie bauen sich mit dem ZX80 oder 81 eine Steuerung, dann müssen Sie doch das Programm dazu von Kassette laden oder in einem (teuren) CMOS-RAM mit Akku-Pufferung speichern. Haben Sie das Programm in einem EPROM, müssen Sie zwar nicht laden, den Rechner aber immer noch von Hand starten.**

Bus geführt. Wir verwenden sie in einer kleinen Zusatzschaltung, die den ZX dazu bringt, daß er beim Starten nicht im Basic-ROM zu arbeiten beginnt, sondern auf unserer EPROM-Platine.

Dabei achten wir auf zwei Dinge:  
\* Der ZX verwendet zum Erzeugen des Fernsehbildes einen NMI-Generator. Den müssen wir beim Starten abschalten. Das geschieht mit wenigen Bytes gleich am Anfang in unserem EPROM.

\* Wenn wir ein Programm aus dem EPROM ausführen, mag das ja ganz nett sein, aber das 8-KByte-ROM brauchen wir auch — sonst können wir ja gleich das ROM durch ein EPROM ersetzen und uns die ganze Platine sparen. Darum ist die Schaltung so gebaut, daß wir auch auf das ROM zugreifen, also auch Basic-Programme benutzen können.

Der Reset-Knopf kann an der Platine oder intern im Rechner angeschlossen werden.

Damit ist die Autostart-Platine bereits fertig und wir können in unserem EPROM Programme unterbringen, die zum Beispiel als Basic-Programme geschrieben wurden. Beim Start werden sie von einem Verschiebe-Programm aus dem EPROM ins RAM befördert und dort mit Hilfe des Interpreters abgearbeitet.

Für viele Steuerungen ist es erforderlich, nach Systemabstürzen die Steuerung neu zu starten. Schließlich ist es nicht jedermanns Geschmack, nachts um halb drei in den

Keller zu gehen und die Heizungssteuerung neu anlaufen zu lassen. In diesem Fall verwendet man eine Überwachungsschaltung, die den Computer beaufsichtigt. Bei der Bundesbahn ist eine Sicherheitschaltung üblich, die man als »Totmann-Taste« bezeichnet: Wenn ein Lockführer nicht in regelmäßigen Abständen eine Taste drückt, bremsst der Zug automatisch.

Für eine Totmann-Taste benötigt man einen Zeitgeber, der vom Rechner über ein Ausgabeport, zum Beispiel spätestens alle fünf Minuten, zurückgesetzt wird. Geschieht das nicht, wird der Rechner durch einen Relaiskontakt nach Ablauf der Fünf-Minuten-Frist zurückgesetzt und der Autostart startet die Steuerung von neuem.

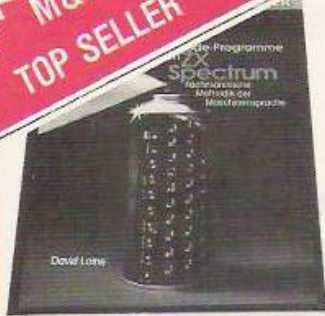
Der Autostart »läuft« mit dem internen 1-KByte-RAM und mit dem Original RAM-Pack von Sinclair sowie dem 16-KByte-RAM der Firma Dorsch. Letztere Platine kann auch auf 48 KByte beziehungsweise 56 KByte umgebaut werden. Beide Zusätze haben einen eigenen Refresh-Zähler. Ob alle RAM-Platinen mit einem Refresh-Zähler mit dem Autostart zusammenarbeiten, ist noch nicht bekannt.

Die Memotech-RAMs arbeiten übrigens nicht mit dem Autostart zusammen, ebenso wie sie bei einer Reihe von Portschaltungen »streiken«. Auch andere RAM-Platinen ohne eigenen Refresh-Zähler arbeiten nicht mit dem Autostart zusammen.

Fortsetzung auf Seite 162

# M&T-Buchverlag

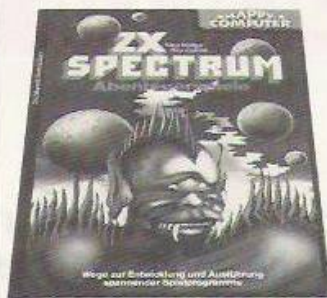
M&T  
TOP SELLER



D. Lane

### Maschinencode-Programme für den ZX Spectrum

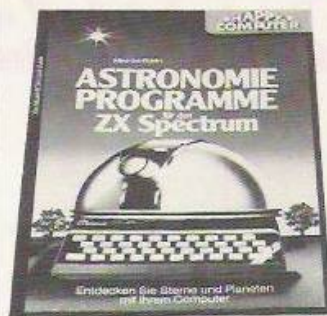
**Juni 1984, 203 Seiten**  
Nützliche Maschinencode-Programme mit Ihrem ZX Spectrum: Sortierung von Fließkommazahlen · Übernahme von Parametern direkt von einem Basic-Programm · Flußdiagramme · für Profis und solche, die es werden wollen.  
Best.-Nr. MT 702  
(Str. 29,50/6S 249,60) **DM 32,—**



T. Bridge/R. Carnell

### ZX-Spectrum Abenteuerspiele

**September 1984, 208 Seiten**  
Die Entstehungsgeschichte der Abenteuerspiele mit repräsentativen Beispielen für jede »Epoche«. Ein Programm speziell für Ihren ZX-Spectrum: »Das Auge des Sternenkriegers«, ein Grafik-Abenteuerspiel, das Sie in Atem hält!  
Best.-Nr. MT 712  
(Str. 29,80/6S 232,40) **DM 29,80**



M. Gavin

### Astronomie-Programme für den ZX-Spectrum

**September 1984, 268 Seiten**  
Eine phantastische Reise in die Welt des Kosmos mit Ihrem ZX-Spectrum: Der Julianische Kalender · Die Mondphasen · Eigene Satelliten starten · Kepler's Umlaufbahnen · Die Umlaufbahn Plutos · Interessant nicht nur für Hobby-Astronome.  
Best.-Nr. MT 732  
(Str. 27,50/6S 232,40) **DM 29,80**



### Im Land der Abenteuer

**1984, 146 Seiten**  
Ein Lösungsbuch für 14 spannende Rätsel- und Abenteuerspiele mit hochauflösender Grafik: Tod in der Karibik · Transsylvanien · Unternehmen Asteroid · Das geheimnisvolle Haus · Zauberer und Prinzessin · Das goldene Vlies · Zeitzone · Der dunkle Kristall · Die Klinge von Blackpool · Flucht aus Rungistan · Der Schlangenstein u.a.  
Best.-Nr. MT 699  
(Str. 27,50/6S 232,40) **DM 29,80**



J. Purdum

### Einführung in C

**1984, 304 Seiten**  
Die grundlegende Charakteristik von C-Operatoren, Variablen und Schleifen · Erstellung eigener Funktionen · Ein- und Ausgabeoperationen in C · Anlegen einer Adreßkartei · Einsatzmöglichkeiten in nahezu allen Bereichen · für Einsteiger und Fortgeschrittene.  
Best.-Nr. MT 561  
(Str. 63,50/6S 538,20) **DM 69,—**



K. Knecht

### Einführung in Forth

**1984, 218 Seiten**  
Ausführliche Informationen über die MMS Forth-Version der Computersprache Forth · syntaktische Grundlagen · zahlreiche Programmierbeispiele · der richtige Einstieg in das Programmieren mit Forth.  
Best.-Nr. MT 635  
(Str. 53,40/6S 452,40) **DM 58,—**



H. Stein

### Logo — Grafik, Sprache, Mathematik

**1984, 257 Seiten**  
Eine Einführung in Logo als Lehr- und Lernsprache · Grafikprozeduren · Zeichenkettenmanipulationen · Probleme der Rekursivität · Sprachbildung und Sprachforschung · Grundlagen der Arithmetik · mit umfassendem Glossar.  
Best.-Nr. MT 648  
(Str. 38,60/6S 327,60) **DM 42,—**



M.-P. Gottlob/G. Strecker

### Die Btx-Fibel

**1984, 119 Seiten**  
Eine Einführung in die Einsatzmöglichkeiten, die Funktionsweise und den Nutzen von Btx im privaten und professionellen Bereich · Aufbau, Funktion und Bedienung der Geräte ohne technischen Ballast erklärt · das neue Kommunikationssystem für jedermann.  
Best.-Nr. MT 519  
(Str. 27,50/6S 232,40) **29,80**



In guten Buchhandlungen, Computershops und Fachabteilungen der Kaufhäuser.

**Markt & Technik**  
Verlag Aktiengesellschaft,

Hans-Pinsel-Str. 2, 8013 Haar bei München, ☎ 089/4613-220  
Schweiz: M&T-Vertriebs AG, Alpenstr. 14, CH-6300 Zug, ☎ 042/223155  
Österreich: Rudolf-Lechner & Sohn, Heizwerkstraße 10, A-1232 Wien, ☎ 0222/677526



W. Pest

### Computerchinesisch für Einsteiger

**Juli 1984, 107 Seiten**  
Ein praxisnahes Lexikon, das Personal-Computer-Benutzern und solchen, die es werden wollen, das Lesen von Fachzeitschriften, Büchern, Bedienungsanleitungen und Datenblättern erleichtert · über 1000 häufig benötigte Fachbegriffe klar und verständlich erläutert · mit zahlreichen Abbildungen.  
Best.-Nr. MT 690  
(Str. 25,90/6S 218,40) **DM 28,—**



Personal Computer Lexikon

**1982, 136 Seiten**  
Über 1000 Suchbegriffe aus Hard- und Software · deutsch/englisch · ausführlicher Artikel zu jedem Suchbegriff · englisch/deutsch Register im Anhang · der ideale Einstieg ins Homecomputing · das unentbehrliche Nachschlagewerk für den Profi.  
Best.-Nr. MT 390  
(Str. 18,50/6S 154,40) **DM 19,80**

# KOMPATIBILITÄT

Viele Computerbesitzer haben dasselbe Problem: Wie schreibe ich ein Programm für meinen Computer um? Schwierigkeiten bereitet hierbei das Fehlen eines Basic-Standards.

Vom ZX81 zum Spectrum halten sich die Schwierigkeiten in Grenzen, vor allem, wenn der Programmierer beide Geräte kennt. Aber genau das ist oft nicht der Fall.

Schwieriger wird es noch, wenn ein Programm von einem völlig anderen Computer umgeschrieben werden soll.

Und noch ein Problem taucht auf: wenn »PEEK« und »POKE« benutzt werden, braucht man das Handbuch des anderen. Wobei das noch keine Garantie für die Lösung des Problems ist. In vielen Handbüchern steht nur wenig drin, die allerwenigsten sind so ausführlich wie Sinclairs.

Im folgenden werde ich die Befehle und Funktionen von ZX81 und Spectrum erläutern und gegebenenfalls Hinweise auf andere verbreitete Dialekte geben. Dort, wo ein Befehl nicht vorhanden ist, werde ich — soweit möglich — Wege zeigen, wie er simuliert werden kann.

Vorab noch einige Randbemerkungen: »RUBOUT« beim Kleinen

heißt »DELETE« beim Großen und mitunter »BACKSPACE« bei anderen; »NEWLINE« beim ZX81 heißt »ENTER« beim Spectrum und »RETURN« bei den meisten anderen.

#### »SLOW« und »FAST« (ZX81)

Um Zeit zu sparen, beispielsweise bei langen Rechenoperationen, kann der »FAST«-Modus eingeschaltet werden. Während dieser Zeit kümmert sich der Rechner nicht um das Fernsehbild, solange er nicht auf etwas warten muß (»INPUT« zum Beispiel). Dadurch hat er mehr Zeit zum Rechnen. Mit »SLOW« wird dieser Modus wieder ausgeschaltet.

#### »PRINT« (beide)

Ist wohl einer der am häufigsten gebrauchten Befehle überhaupt. Bei Sinclairs Kleincomputern sind die beiden Befehle identisch; die Farbinformationen müssen auf dem ZX81 weggelassen werden (»INVERSE«, »BRIGHT«, »PAPER«, »INK«, »OVER«, »FLASH«). Einige davon können jedoch mit einem kleinen Trick simuliert werden.

»PRINT AT zeile, spalte« kann voll-

ständig übernommen werden; das Raster ist dasselbe.

In anderen Dialekten ist »AT« in der Regel nicht möglich. Teilweise muß vorher dafür der Befehl »LOCATE' spalte, zeile« benutzt werden, andere wieder müssen den Cursor von der Home-Position (oben links) herunterzählen, das heißt: ein bestimmtes Steuerzeichen so oft drucken wie nötig.

#### »PLOT Spalte, Zeile« (ZX81)

Gibt es zwar auf beiden Rechnern, aber der ZX81 legt ein anderes Raster zugrunde, bedingt durch die geringere Auflösung.

Für die kleineren Computer ist die Spectrum-Auflösung mit Hilfe eines vor einiger Zeit in Happy-Computer veröffentlichten Programms möglich, oder auch mit dem »HRG«-Modul, einer Hardware-Erweiterung, die jedoch nach meinen Informationen nicht mehr im Handel ist.

Der Spectrum kann allerdings die ZX81-Grafik simulieren:

PRINT AT +; -INT (zeile/2), INT (spalte/2); <Grafikzeichen>

Die Berechnung der »PRINT«-Position beruht darauf, daß jeder »Punkt« des ZX81 halb so breit und halb so hoch ist wie eine Schreibstelle. Zudem werden bei »PLOT« die Zeilen von unten gezählt, bei »PRINT« von oben (übrigens bei beiden Computern).

Grafikzeichen steht für eines der folgenden vier Zeichen: » «, » «, » « « (Viertelquadrate auf den Tasten 1 — oben rechts, 2 — oben links, 4 — unten rechts und 7 mit Caps Shift — unten links).

Mit etwas Rechnen vor dem Druck-Befehl kann festgestellt werden, welches Zeichen gedruckt werden muß:

```
LET a$ = »-«: REM Grafikzeichen
IF zeile/2 <> INT (zeile /2) AND
spalte/2 <> INT (spalte/2) THEN
LET kont = 1 IF zeile/2 <> INT (zeile/2) AND
spalte/2 = INT (spalte/2)
THEN LET kont = 2 IF zeile/s = INT (zeile/2) AND
spalte/2 <> INT (spalte/2) THEN LET kont = 3 IF
zeile/3 = INT (zeile/s) AND
spalte/2 = INT (spalte/2) THEN LET
kont = 4 oder so:
LET kont = (INT (spalte/2) <> spalte/2 AND
INT (zeile/2) <> zeile/s) + 2*(IN T (zeile/2) <> zeile/2 AND
INT (spalte/2) = spalte/2) + 3*(INT (zeile/2) =
zeile/2 AND INT (spalte/2) <> spalte/2) + 4*(INT (spalte/2) =
spalte/2 AND INT (zeile/2) =
zeile/2)
```

Die erste Version erfordert zwar vier Zeilen, der Prozessor verarbeitet aber »IF...THEN« schneller als die in der zweiten Version benutzte Boole'sche Logik: Der in Klammern stehende Ausdruck ergibt den Wert 1, wenn der Ausdruck wahr ist, andernfalls 0. (Bei anderen Rechnern ergibt der wahre Ausdruck oft den Wert -1!)

Beide Versionen benutzen folgenden Hintergrund: Wenn die Spaltenzahl in dem »PLOT«-Befehl gerade ist, liegt der »Punkt« in der linken Hälfte der »PRINT«-Position. Bei gerader Zeilenzahl ist in der unteren Hälfte zu drucken. Aus beiden zusammen ergibt sich dann eben »unten links« oder »oben rechts«. In dem »PRINT«-Befehl muß es dann heißen: ...;a\$(kont)

Soll dieser Befehl in eine Schleife eingebaut werden, muß vorher »OVER 1« gesetzt werden, sonst löscht ein neues Zeichen ein altes.

#### »UNPLOT« Spalte,Zeile (ZX81)

Mit demselben »PRINT«-Befehl wie oben kann auch das realisiert werden, jedoch muß ein Leerzeichen gedruckt werden.

#### »INVERSE« (Spectrum)

kann auf dem ZX81 dadurch ersetzt werden, daß die inversen Zeichen benutzt werden.

#### »FLASH« (Spectrum)

läßt sich simulieren, indem abwechselnd das normale und das inverse Zeichen gedruckt werden, mit einer Pause (Warteschleife) von entsprechender Länge dazwischen.

#### »SCROLL« (ZX81)

ist ein auf dem Spectrum normalerweise überflüssiger Befehl. Wenn beim ZX81 in Zeile 21 gedruckt wurde und weiter geschrieben werden soll, unterbricht er das Programm mit der Fehlermeldung »Bildschirm voll«. Mit »SCROLL« wird der Bildschirminhalt um eine Zeile nach oben geschoben. Der Spectrum fragt hier automatisch »scroll?« und macht nach Tastendruck weiter. Im »ROM« gibt es die Scroll-Routine aber trotzdem. Sie kann mit »RANDOMIZE USR 3582« aufgerufen werden.

#### »COUNT«, »RAND« (ZX81)

sind dasselbe wie »CONTINUE« und »RANDOMIZE«, erscheinen in ZX81-Listings jedoch so, wie sie auf den Tasten stehen, also abgekürzt.

#### ! (Spectrum)

entspricht dem »\*\*« auf dem ZX81.

#### »DEF FN« (Spectrum)

kann sehr oft durch Benutzung von »VAL« beim ZX81, ersetzt werden.

Ein Beispiel:

```
DEF FN a(x) = x*x
```

```
LET x = 10
```

```
PRINT FN a(x)
```

kann auf dem ZX81 so formuliert werden:

```
LET A$ = "X*X"
```

```
LET X = 10
```

```
PRINT VAL A$
```

(Übrigens: das funktioniert auch auf dem Spectrum.)

#### »CLEAR« (Spectrum)

gibt es auch auf beiden Computern. Die Befehle haben jedoch nicht dieselbe Wirkung.

Mit beiden können die Variablen gelöscht werden. Auf dem Spectrum kann damit jedoch zusätzlich »RAMTOP« verschoben werden. Das ist beim ZX81 jedoch viel komplizierter und kann außerdem nicht im Programm erfolgen, denn die Be-

fehlsfolge muß mit »NEW« abgeschlossen werden.

Es gibt jedoch auch Möglichkeiten, »RAMTOP« beim ZX81 ohne »NEW« zu verschieben.

#### »SCREEN\$« (Spectrum)

hat zwei Bedeutungen. Die erste ist für den ZX81 unmöglich: Soll ein Bildschirm-Inhalt auf Band gespeichert werden, so kann das mit dem Befehl »SAVE« "Bild" »SCREEN\$« geschehen. Hier steht dieses Keyword für »CODE« 16384,6912. Bei Adressen 16384 beginnt der Spectrum-Bildschirmspeicher; er ist 6912 Byte lang.

Die andere Verwendungsmöglichkeit ist das Abtasten des Bildschirms auf ein bestimmtes Schriftzeichen hin. Das Format ist »SCREEN\$(zeile,spalte). Die selbstdefinierten Grafikzeichen können damit übrigens nicht abgefragt werden.

Diese Funktion ist beim ZX81 etwas schwieriger zu realisieren. Denn hier wurde — um den sehr knappen Platz in der Grundversion zu sparen — ein variabler Bildschirmspeicher angelegt (eigentlich genial, nicht?). Es geht aber auch:

```
LET a$ = CHR$(PEEK 16396 + 256*
PEEK 16397 + 1 + Zeile + 33 *
Spalte))
```

#### »BIN« (Spectrum)

ist recht praktisch. Duale (= binäre) Zahlen können damit direkt in den Spectrum eingegeben werden. Mit »PRINT BIN« 00001111 erhält man 15 als Ergebnis auf dem Bildschirm. Der ZX81-Benutzer muß hierfür ein kleines Programm benutzen (siehe Happy-Computer, Ausgabe 9/1984, Seite 69-71, Zahlenumwandlung).

#### »PAUSE« (Spectrum)

ist zwar auf beiden Computern verfügbar; allerdings sind auch hier wieder Unterschiede zu beachten:

Spectrum:

»PAUSE 0« wartet auf Tastendruck.  
»PAUSE Länge« ergibt eine Pause von Länge/50 Sekunden.

ZX81:

»PAUSE LÄNGE« mit »LÄNGE« größer als 32767 entspricht »PAUSE 0«.

Wenn Werte bis 32767 eingegeben werden, entspricht die Pause der Formel, die auch für den Spectrum gilt.

Die Pausen können jederzeit durch Drücken einer Taste verkürzt werden. Will man das vermeiden,

so muß die Pause durch eine Warteschleife (FOR i=1 TO 1000: NEXT i) ersetzt werden.

Bei beiden Computern wirkt sich die Pause auf die Systemvariable »FRAMES« aus. In dieser Variable werden die Fernsehbilder gezählt, 50 je Sekunde. Beim Spectrum ist diese Variable drei Byte lang. Während einer Pause wird nicht weitergezählt.

Beim ZX81 dagegen ist diese Variable zwei Byte lang und wird von dem Pause-Befehl benutzt. Bit 15 wird auf 0 gesetzt, Bit 0 bis 14 enthalten die Länge der Pause. Nun wird rückwärts gezählt, bis »FRAMES« 9 ist. Nun wird Bit 15 wieder 1 gesetzt. Wenn also diese Systemvariablen benutzt werden, muß auf eine Warteschleife zurückgegriffen werden.

#### »INPUT "text";var« (Spectrum)

kann der ZX81 in der Form nicht. Hier muß der im »INPUT« ausgegebene Text vorher mit »PRINT« ausgegeben werden.

#### »LINE« (Spectrum)

hat zwei mögliche Wirkungen, je nachdem, wo es verwandt wird.

Die erste Einsatzmöglichkeit ist das Abspeichern von Programmen mit automatischem Start in der durch »LINE« angegebenen Zeile:

```
»SAVE« "name"»LINE« zeile
```

(Entgegen anderslautenden Anleitungen muß dieser Befehl nicht im Programm stehen.)

Beim ZX81 kann ein Abspeichern mit automatischem Start nach dem Laden nur durch zwei Befehle im Programm erreicht werden:

```
9000 SAVE "NAME"
```

```
9010 GO TO ZEILE
```

Die zweite Verwendungsmöglichkeit ist die Unterdrückung der Anführungszeichen bei Zeichenketten-Eingaben. Damit wird gleichzeitig auch die Eingabe von »STOP« verhindert; denn die Anführungszeichen können nicht mehr mit »DELETE« entfernt werden:

```
»INPUT LINE« A$
```

Dies ist mit dem ZX81 nicht möglich.

#### »ATTR (Zeile,Spalte)« (Spectrum)

gibt den aktuellen Farbwert an dieser Position aus. Binär dargestellt ist es ganz leicht zu verstehen:

7. Bit: Flash an: 1, aus: 0 128

6. Bit: Bright an: 1, aus: 0 64

5.-3. Bit: Hintergrundfarbe 0-7

2.-0. Bit: Schreibfarbe 0-7

Der Wert wird dezimal ausgege-

ben. Dies entfällt natürlich auf dem ZX81.

#### »POINT (Zeile,Spalte)« (Spectrum)

gibt an, ob ein Pixel Vorder- oder Hintergrundfarbe hat (1 oder 0). Bislang habe ich keine entsprechende Möglichkeit für den ZX81 gefunden. Der unter »SCREEN\$« gezeigte Befehl gibt zwar auch für den Grafiken die richtigen Werte aus; bei geplotteten Punkten jedoch eine leere Zeichenkette.

Dies sind die wichtigsten Befehle und Funktionen, die sich auf den beiden Sinclair-Computern unterscheiden.

Nun zu einigen recht nützlichen Systemvariablen, die — natürlich — unterschiedliche Adressen haben und teilweise auch etwas anders arbeiten.

#### »FRAMES«

kann zur Echtzeitmessung herangezogen werden. Sie zählt die Anzahl der Fernsehbilder seit dem Einschalten (50 je Sekunde).

Beim Spectrum ist sie drei Byte lang und hat die Adressen 23672 bis 23674. 23672 wird alle  $\frac{1}{50}$  Sekunde um 1 erhöht, entstehende Überträge werden in die nächsthöheren Bytes übertragen.

Wenn die Zeitmessung beginnt, POKet man am besten die drei Bytes auf 0. Am Ende der Zeitmessung erhält man die Zeit in Sekunden mit  $>LET < t = (>PEEK < 23672 + 256 * >PEEK < 23673 + 65536 * >PEEK < 23674) / 50$ .

Wenn »BEEP« oder »PAUSE« benutzt werden, wird die Variable nicht weiter erhöht. Die Zeit muß also addiert werden.

Beim ZX81 umfaßt diese Variable nur zwei Byte und liegt in den Adressen 16436 bis 16437. Allerdings startet sie mit dem Wert 255 in beiden Adressen und wird jede  $\frac{1}{50}$  Sekunde heruntergezählt. Hier muß beim Start der Zeitmessung also 255 statt 0 gePOKET werden. Die Zeit in Sekunden erhält man dann durch  $>LET T = (65535 - (PEEK 16436 + 256 * PEEK 16367)) / 50$ .

Der Befehl »PAUSE« verändert diese Variable.

Auf beiden Computern wird nicht weitergezählt, während Programme geladen oder gespeichert werden.

**Verschiedene Systemvariablen, die benutzt werden können**

»LASTK« — enthält den Code der zuletzt gedrückten Taste, 16421 beim

ZX81, 23560 beim Spectrum, zwei Bytes, nur »PEEK«

»REDPEL« — Wartezeit vor Einsetzen der Tastenwiederholfunktion, nur Spectrum 23561, ein Byte, »POKE«

»REPPER« — Zeit zwischen zwei Wiederholungen einer Taste, nur Spectrum, ein Byte, »POKE«

Beide Werte auf 1 gePOKET, machen eine Tastatureingabe fast unmöglich.

»PIP« — Länge des Tastaturtones, nur Spectrum, ein Byte, »POKE«

»MODE« — gibt an, welcher Cursor K, G, L erscheint, 16390 beim ZX81, 23658 beim Spectrum, ein Byte. Kann beim Spectrum benutzt werden, um Großschrift einzuschalten: »POKE« 23658,8. Das ist übrigens nicht die Systemvariable »MODE«, die nicht gePOKET werden kann, sondern ein Flag (Kontrollbyte).

»COORDS« — enthält die Koordinaten des zuletzt geplotteten Punktes, 16438 (Spalte) und 16439 (Zeile) beim ZX81, 23677 (Spalte) und 23678 (Zeile) beim Spectrum.

»SCR CT« — zählt die Scrolls beim Spectrum. Der Wert ist stets um eins höher als die Anzahl der Scrolls, die ausgeführt werden, bevor mit der Meldung »scroll?« gestoppt wird. Wenn man immer wieder einen Wert größer als 1 »POKET«, wird diese Meldung unterdrückt.

»SEED« — erhält durch »RANDOMIZE« beziehungsweise »RAND« einen Wert und wird von »RND« benutzt. »RANDOMIZE« 10 entspricht »POKE SEED« 10. 16434 beim ZX81, 23670 beim Spectrum, zwei Byte.

»DF CC« — enthält die Adresse der letzten »PRINT«-Position im Bildschirmspeicher, 16398 beim ZX81, 23684 beim Spectrum, zwei Byte; die Schreibposition kann mit »POKE« verschoben werden.

»RAMTOP« — Ende des für »Basic« nutzbaren Speichers, 16388 beim ZX81, 23730 beim Spectrum, zwei Byte. Beim ZX81 wird die erste nicht mehr ansprechbare Adresse gespeichert, beim Spectrum die letzte ansprechbare Adresse.

»CHARS« — ist um 256 niedriger als die Startadresse des Zeichensatzes beim Spectrum, Adresse 23606, zwei Byte; kann gePOKET werden, um einen im »RAM« gespeicherten Zeichensatz zu benutzen, oder auch, um ein Programm unleserlich zu machen.

Diese Liste ist sicher noch nicht vollständig; denn findige Hobby-Programmierer werden noch andere Variablen zu benutzen wissen. Aber dies sind die wichtigsten, die öfter schon einmal auftauchen und durchaus Kopfschmerzen hinterlassen können, wenn man nicht weiß, was sie machen und wie sie umzusetzen sind.

Wenn statt »PRINT "A"« aus irgendeinem Grund »PRINT CHR\$ 38 (65)« benutzt wird, braucht man eine Tabelle des Zeichensatzes, um das Programm anpassen zu können. Der Zeichensatz des ZX81 ist völlig unkonventionell, der Spectrum benutzt weitgehend den »ASCII«-Code.

Der »ASCII«-Code speichert nur sieben Bits, kann also mit Codes über 127 nichts mehr anfangen. Das ist wesentlich bei der Datenfernübertragung über Interface 1 oder eine andere RS232-Schnittstelle (siehe auch Interface 1-Praxis). Zu guter Letzt noch einige Hinweise zum Umschreiben von Programmen anderer Computer, die mit Microsoft-»Basic«-Dialekten arbeiten.

Das Sinclair-»Basic« entspricht diesem weitgehend, es gibt jedoch einige Unterschiede, die beachtet werden sollten.

Der »LET«-Befehl ist bei den meisten anderen Computern optional, das heißt, es kann weggelassen werden. Ebenso kann oft hinter »THEN« das »GOTO« entfallen.

Einige Computer verlangen keine Zwischenräume hinter den Befehlen, bei anderen müssen diese sein.

## Keine »Kleinigkeiten«

Auf diese »Kleinigkeiten« kommt man meist schnell. Anders ist es mit folgenden Befehlen.

Die meisten Computer lassen es nicht zu, daß ein Feld zweimal dimensioniert wird, wie es bei Sinclair möglich ist (beispielsweise um das Feld zurückzusetzen). Einige müssen es mit »ERASE« löschen und dann neu dimensionieren, andere mit Schleifen zurücksetzen.

Die Zeichenketten- oder String-Verarbeitung ist bei Sinclair sehr komfortabel gelöst. Teilketten können mit A\$(erstes zeichen TO letztes zeichen) ausgedruckt werden. Da-

für gibt es im Microsoft-»Basic« meistens drei Befehle:

>MID<\$(name der zeichenkette, erstes zeichen, anzahl der zeichen) ist universell einsetzbar.

>LEFT<\$(name der zeichenkette, anzahl der zeichen) ergibt die ersten Zeichen.

>RIGHT<\$(name der zeichenkette, anzahl der zeichen) ergibt die letzten Zeichen einer Kette.

Ein Beispiel:

```
LET A$="KATZENFUTTER"
PRINT A$(TO 6) ergibt den Bildschirmausdruck »KATZEN«,
PRINT A$(7 TO) ergibt »FUTTER«.
```

(Der erste beziehungsweise zweite Parameter (=Wert) kann entfallen, wenn das erste respektive letzte Zeichen gemeint ist.)

In Microsoft-»Basic« sieht das so aus:

```
PRINT MID$(A$,1,6) für »KATZEN«
oder auch PRINT LEFT$(A$,6)
PRINT MID$(A$,7,6) oder RIGHT$(A$,6) für »FUTTER«
```

Dies sind die wichtigsten Unterschiede. Es würde etwas weit führen, hier alle Feinheiten aufzuzählen; besonders die diversen Steuerzeichen, die für die Cursor-Positionierung oder »CLS« etc. verwandt werden. Ein Teil davon ist aus der Zeichensatz-Tabelle zu entnehmen.

Als Abschluß noch ein Programm für Spectrum-Besitzer, die ZX81-Programme laden möchten.

## Tips für Umsteiger

Für die »Umsteiger« insbesondere, aber auch für andere ist in diesem Zusammenhang natürlich ein Programm sehr interessant, das ZX81-Programme für den Spectrum lesbar macht.

Ich habe das Programm »LOAD ZX81 Program into Spectrum« ausprobiert. Besonders hübsch ist die Sicherung gegen Mißbrauch: Jedes einzelne erhält einen Namen (bei Direktbezug) beziehungsweise eine Nummer, bestehend aus fünf Zeichen. Wenn man sich beim letzten Zeichen vertippt, hat man keine Korrekturmöglichkeit mehr. Das Programm verabschiedet sich.

Das Maschinenprogramm sitzt unsichtbar im Bildschirmspeicher, um möglichst wenig Speicherplatz zu beanspruchen.

Zur Funktion:

Nach dem Laden startet das Programm selbst und wartet auf das Kennwort. Da bei ZX81-Programmen die Lautstärke eine große Rolle spielt, wurde eine Routine eingebaut, mit der man die Lautstärke optimal einstellen kann. (Unter Umständen ist es nötig, vorher den Tonkopf des Recorders neu einzustellen.) Nachdem man die Lautstärke gemäß den Anweisungen eingestellt hat, kann man den Ladevorgang mit der SPACE-Taste starten.

Es erscheint der Hinweis: »Lade:«; wenn dahinter ein Name erscheint, wird das Programm geladen.

Das Programm erkennt Ladefehler und bricht dann ab. Der bis dahin geladene Teil steht aber zur Verfügung. Man kann diesen Teil nun direkt vom Menü aus speichern und überprüfen. Dies ist sinnvoll, wenn das Programm für den Spectrum zu lang ist und in zwei Teilen geladen werden muß oder wenn man die Absicht hat, mehrere Programme zu übersetzen.

Es besteht auch die Möglichkeit, direkt (oder nach SAVE) ein neues Programm zu laden. Wenn man sich ansehen möchte, was denn nun dabei herausgekommen ist, kann man auch »zu Basic zurück«. Damit wird allerdings das Übersetzungsprogramm gelöscht. Etwas geschockt sitzt man vor der »Einschaltmeldung« des Computers. Dennoch ist das ZX81-Programm noch im Speicher und kann nun nach Herzenslust angepaßt werden.

Einige Befehle werden schon beim Laden vom Ladeprogramm geändert: FAST und SLOW werden gar nicht erst übernommen. Andere Befehle werden gleich ersetzt. Dabei werden Maschinenprogramme in REM-Zeilen erkannt und nicht verändert.

Wenn ich auch mit recht vielen Ladefehlern zu kämpfen hatte (bei wirklich guten Aufnahmen), so ist doch dieses Verfahren sicherlich schneller und einfacher als das Ändern der Programme »zu Fuß« mit anschließendem Eintippen.

Die Erklärung des Programms ist zwar etwas sehr klein geschrieben, auch die Schriftqualität läßt noch Wünsche offen, doch inhaltlich reicht sie aus, zumal der Umgang mit diesem Programm sehr einfach ist.

(Erika Holscher)

## CODE Tabellen











































ZX81, ZX Spectrum, ASCII  
(deutsche Sonderzeichen in Klammern)





































Code	BI	SP	ASCII
0			
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11	"		
12	£		
13	§		
14	:		
15	?		
16	(		
17	)		
18	>		
19	<		
20	-		
21	+		
22	~		
23	*		
24	/		
25	:		
26	.		
27	'	Escape	
28	0		

Code	BI	SP	ASCII
29	1		
30	2		
31	3		
32	4		
33	5	!	!
34	6	"	"
35	7	#	#
36	8	\$	\$
37	9	%	%
38	A	&	&
39	B	'	'
40	C	(	(
41	D	)	)
42	E	*	*
43	F	+	+
44	G	,	,
45	H	-	-
46	I	.	.
47	J	/	/
48	K	0	0
49	L	1	1
50	M	2	2
51	N	3	3
52	O	4	4
53	P	5	5
54	Q	6	6
55	R	7	7
56	S	8	8
57	T	9	9

Code	BI	SP	ASCII
58	U	:	:
59	V	;	;
60	W	<	<
61	X	=	=
62	Y	>	>
63	Z	?	?
64		{}	{}@
65		A	A
66		B	B
67		C	C
68		D	D
69		E	E
70		F	F
71		G	G
72		H	H
73		I	I
74		J	J
75		K	K
76		L	L
77		M	M
78		N	N
79		O	O
80		P	P
81		Q	Q
82		R	R
83		S	S
84		T	T
85		U	U
86		V	V

Code	SI	SP	ASCII
87		W	W
88		X	X
89		Y	Y
90		Z	Z
91		[	[Å]
92		\	\O)
93		]	]O)
94		^	
95			
96		z	z
97		a	a
98		b	b
99		c	c
100		d	d
101		e	e
102		f	f
103		g	g
104		h	h
105		i	i
106		j	j
107		k	k
108		l	l
109		m	m
110		n	n
111		o	o
112		p	p
113		q	q
114		r	r
115		s	s
116		t	t
117		u	u
118		v	v
119		w	w
120		x	x

Code	SI	SP	ASCII
121		y	y
122		z	z
123		[	[Å]
124		\	\O)
125		]	]O)
126		^	^-O)
127			
128			Ende
129			
130			
131			
132			
133			
134			
135			
136			
137			
138			
139			
140			
141			
142			
143			
144			NDC Start
145			
146			
147			
148			
149			
150			
151			
152			
153			

Code	SI	SP	ASCII
154			
155			
156			
157			
158			
159			
160			
161			
162			
163			
164			NDC Ende
165			
166			
167			
168			
169			
170			
171			
172			
173			
174			
175			
176			
177			
178			
179			
180			
190			
191			
192			
193			
194			
195			
196			
197			
198			
199			
200			
201			
202			
203			
204			
205			
206			
207			
208			
209			
210			
211			

Code-Tabellen (Ende)



# BIN DEZ ODER HEX?

Wer sich mit der Programmierung von Computern beschäftigt, stößt irgendwann auf Hexadezimal- und Dualzahlen (Binärzahlen) — spätestens dann, wenn er Maschinencode lernen will. Damit taucht auch das Problem auf, eine Hex-Zahl in eine Dez- oder Bin-Zahl umzuwandeln oder auch umgekehrt.

Zahlen bestehen aus Ziffern, so wie Wörter aus Buchstaben. Diese Zahlensysteme sind Stellenwertsysteme, das heißt eine Ziffer hat je nach ihrer Stelle in der Zahl einen anderen Wert. Die »Grundwerte« jeder Stelle sind Potenzen der Basis des Zahlensystems. Die Potenzen beginnen von rechts mit Null und steigen nach links an (übrigens: eine Zahl hoch 0 ergibt immer 1). Jedes Zahlensystem braucht so viele Ziffern wie die Basis angibt.

Das hört sich komplizierter an als es ist. Nehmen wir uns doch erst mal unser Zehnersystem vor:

$10^3$	$10^2$	$10^1$	$10^0$
( 1000	100	10	1 )
6	4	3	1

In der Tabelle steht die Zahl 6431. Sie wird von unserem Gehirn »automatisch« zusammengesetzt aus  $6 \times 1000 + 4 \times 100 + 3 \times 10 + 1 \times 1$ .

Das Binärsystem funktioniert ganz genau so:

$2^3$	$2^2$	$2^1$	$2^0$
( 8	4	2	1 )
1	0	1	1

Das Dualsystem hat die Basis 2 (binär dargestellt: 10), und wir brauchen zwei Ziffern. In dieser Tabelle steht jetzt die Binärzahl 1011. Ihr dezimaler Wert ist  $1 \times 8 + 0 \times 4 + 1 \times 2 + 1 \times 1 = 11$

Das Hexadezimalsystem ist jetzt für uns kein großes Problem mehr. Es hat die Basis 16 (hexadezimal dargestellt: 10). Nur — wir kommen mit unseren zehn Ziffern nicht aus. Deshalb benutzen wir die ersten sechs Buchstaben des Alphabets mit den (dezimalen) Werten 10 bis 15.

$16^3$	$16^2$	$16^1$	$16^0$
( 4096	256	16	1 )
A	F	8	9

Die Hex-Zahl AF89 hat den dezimalen Wert

$$11 \times 4096 + 15 \times 256 + 8 \times 16 + 9 \times 1 = 49033.$$

Übrigens kann man in allen drei Systemen mit den »normalen« Rechenregeln arbeiten, denn die Systeme sind vom Aufbau her identisch. Das gilt für jedes Zahlensystem, egal zu welcher Basis.

Die jeweilige Rechnung, mit der ich oben die Dezimalzahl ermittelt habe, läßt sich auch in einem Programm einsetzen. Damit haben wir ein Umwandlungsproblem schon gelöst.

Nun muß es uns noch gelingen, Dezimalzahlen in Hex- und Dualzahlen zu verwandeln. Ein sehr einfacher — und universeller — Weg ist die wiederholte Division der Dezimalzahl durch die Basis des »Zielsystems«, in unserem Fall also durch 2 beziehungsweise 16. Die Reste ergeben die Zahl im Zielsystem, allerdings von hinten.

Sehen wir uns das einmal an unserem Beispiel von vorhin an:

13	:	2	=	6	Rest 1
6	:	2	=	3	Rest 0
3	:	2	=	1	Rest 1
1	:	2	=	0	Rest 1

Die dezimale 13 wird eine binäre 1101.

49033	:	16	=	3064	Rest 9
3064	:	16	=	191	Rest 8
191	:	16	=	11	Rest F (15)
11	:	16	=	0	Rest A (11)

Aus der dezimalen 49033 ist die hexadezimale AF89 geworden. Auch dieses Verfahren läßt sich leicht in ein Programm umsetzen.

Es bleibt noch die Umwandlung von Hexadezimalzahlen in Dualzahlen und umgekehrt. Dies geht besonders einfach, weil die Basis des Hex-Systems eine Potenz der Basis des Dualsystems ist. (Mit dem Oktalsystem Basis (8) besteht ein ähnliches »Verwandtschaftsverhältnis«.)

Eine Hexadezimalzahl kann ziffernweise »übersetzt« werden:

A	F	8	9
1011	1111	1000	1001

Umgekehrt geht es genauso einfach: Die Dualzahl wird von rechts ausgehend in Viererblöcke aufgeteilt, die dann einzeln übertragen werden.

Nebenbei: Diese Verwandtschaft ist der Grund dafür, daß das Hexadezimalsystem benutzt wird, eben weil es so einfach ist, die Zahl umzuwandeln. Denn Dualzahlen sind die einzigen, mit denen der Computer etwas anfangen kann. Aber sie sind sehr lang und unübersichtlich. Mit den Hex-Zahlen hat man ein handlicheres Format zum Aufschreiben der Werte, es schleichen sich nicht so leicht Fehler ein. Und trotzdem kann man — auch ohne Rechenkünstler zu sein — die Werte im Kopf umrechnen. Zu diesem Artikel wurde in der Happy-Computer 9/84 auf den Seiten 69 bis 71 ein Umwandlungsprogramm veröffentlicht. Damit kann der Benutzer jede Zahl bis 65535 in allen drei Systemen nebeneinander darstellen, in welcher Form er sie auch eingegeben hat.

(Erika Hölscher)

# Zum Abheben:

# Flugsimulatoren für den

Nur Fliegen ist schöner. Flugsimulationen sind die großen Renner auf dem Spielmarkt. Klar, daß sich mittlerweile für den Spectrum einiges an Software angehäuft hat. Wir haben uns die gesammelten Werke für verhinderte Piloten mal genauer angesehen.

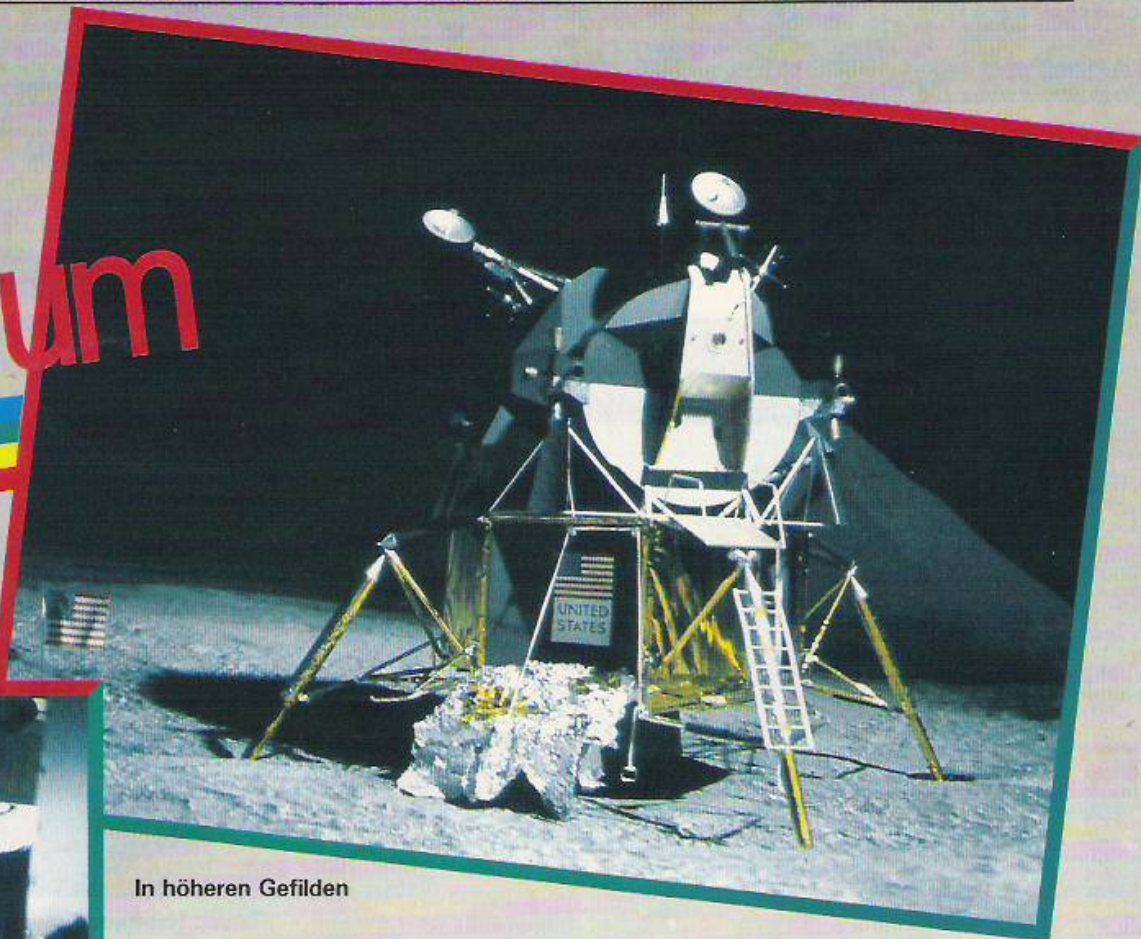


737-Cockpit im Flug



Oldie-Schau,  
Deutsches Museum,  
München

Spectrum



In höheren Gefilden



Für diese »Pionier-Leistung« hat es der »Psion Flugsimulator« zweifelsohne verdient, am Anfang unseres Testfeldes zu stehen. Nicht zuletzt auch deshalb, weil alle Flug-Programme der 2. Generation sich an diesem Vorreiter messen müssen.

Als der »Psion« auf den Markt kam, war er ein exotisches Juwel in der Spiele-Szenerie. Fernab von »Pac Man«- und »Space Invaders«-Aufgüssen, High Scores und sonstigen Klischees wurde dem Spieler urplötzlich ein Pilotenknüppel ins Händchen gedrückt. Und neben seiner Originalität besticht das Programm auch heute noch durch den hohen Spielwitz.

Drei Spielvariationen stehen dem furchtlosen Spectrum-Besitzer zur Auswahl: Start, Landung oder die Simulation beginnt mitten im Flug. Anfänger mit masochistischer Ader wagen sich gleich an die Landung heran: Ein Crash jagt den anderen. Wesentlich aufschlußreicher gestaltet sich da ein Start-Manöver. Nachdem sich der angehende Flug-Kapitän mit der umfangreichen (selbstverständlich englischen) Bedienungsanleitung vertraut gemacht hat, darf abgehoben werden. Zunächst wird die Maschine langsam

aber sicher immer schneller, die Landebahn scrollt mit schöner 3D-Grafik vorbei. Wenn man rechtzeitig aufs richtige Knöpfchen drückt, hebt das Flugzeug schließlich majestätisch ab — um kurz darauf wieder etwas weniger majestätisch abzustürzen. Denn, wie gesagt: Ich spreche von den Erfahrungen eines Anfängers.

Die Grafik des »Psion« ist eine angenehme Überraschung. Der »Pilot« hat die Wahl zwischen dem Blick aus dem Cockpit mit den bereits angesprochenen 3D-Effekten und dem Studium einer Landkarte. Diese gibt neben der augenblicklichen Position des Flugzeuges Aufschluß über anfliegbare Landebahnen und Besonderheiten der Umgebung, wie Gewässer und Gebirge. Die Sinclair-Tastatur ist mit den vielen Bedienungsmöglichkeiten nahezu ausgelastet, was für die Komplexität des Programms spricht.

Der »Psion Flugsimulator« hält, was er verspricht: Ein realistisches, spannendes Vergnügen, das einem so leicht nicht langweilig wird. Die Herausforderung ist groß und dank der detaillierten Grafik hat auch das Auge seine Freude. Ein moderner Klassiker der Spectrum-Software.

**R**unter kommen sie immer — fragt sich nur, in welchem Zustand. Diese alte Flieger-Regel gilt natürlich auch für die Flugsimulatoren. In unserer Übersicht, die übrigens keinen Anspruch auf Vollständigkeit erhebt, haben wir die Spectrum-Programme auf Herz und Nieren geprüft. Bügeln Sie also Ihren Fallschirm und stellen Sie das Rauchen ein: Denn jetzt gehen wir auf einen Höhenflug mit dem Computer.

### Der Pionier: Psion Flugsimulator

Er war der erste Simulator, der für den Spectrum auf den Markt kam.



Raumflug-Simulation



Tatort: Heathrow-Tower

## Der Killer: Night Gunner

»Night Gunner« ist ein etwas merkwürdiger Fall: Man könnte ihn nämlich von der Aufmachung her für einen echten Flug-Simulator mit viel Action halten — er ist aber leider keiner. Bei diesem Programm handelt es sich um ein ebenso simples wie fesselndes Schieß-Spiel, bei dem der Spieler in die Rolle eines wütenden Kampffliegers schlüpft, der seine Kollegen in den Lüften wie wild abdonnert. Etwas simulatorenhafte mutet er dank einiger Anzeigen an: Flughöhe, Schadensmeldungen etc. müssen beim Action-Spektakel zwar beachtet werden, sind aber eigentlich nur schmückendes Beiwerk.

»Night Gunner« ist ein tempo- und abwechslungsreiches Schießspiel mit guter Grafik. Die Flugsimulations-Elemente bleiben aber ganz klar untergeordnet, so daß dieser »Außenseiter« letztendlich in die umfangreiche Ecke der Ballereien zu schieben ist.

## Der Rasante: Fighter Pilot

Das Programm »Fighter Pilot« ist eine sehr gelungene Verknüpfung von Simulation und Actionspiel, ohne dabei zu einer wüsten Ballerei auszuarten wie »Night Gunner«. Das

Hauptaugenmerk liegt hier bei der Beherrschung der Fülle von Anzeigen und Steuerungsmöglichkeiten. Man soll ja nicht glauben, wieviele Knöpfchen so ein Flugzeug hat. Noch eine Parallele zum »Psion«: Auf Gummitasten-Druck hin erscheint eine Landkarte, auf der neben den üblichen Feinheiten wie Landebahnen und der Position des eigenen Fliegers der Standort des finsternen Feindes angegeben wird. Hier beginnt die eigentliche Raffinesse des Spiels. Kriegerisch und unmoralisch, aber äußerst spannend und unterhaltsam. Zunächst müssen Sie Ihr Flugzeug zum Gegner hin manövrieren, was dieser durch seinen unberechenbaren Kurs erschwert. Außerdem muß man sich von der Flughöhe her an den Kontrahenten herantasten, bis man dann dem gegnerischen Flieger gegenübersteht — oder auch nicht, da der Schlingel Ihnen vielleicht auf einmal im Rücken liegt. Geschossen wird auf Knöpfchendruck, wozu man nur einen gewissen Vorrat an Munition hat.

Die wichtigsten Steuerfunktionen von »Fighter Pilot« lassen sich auch per Joystick bewerkstelligen. Ein Menü erlaubt die Wahl zwischen Luft- und Landeanflug-Training oder den gnadenlosen »Combat-Mode«. Ganz Verwegene können auch eine

Landung im dichten Nebel probieren, die beim Anfänger aber sehr rasch mit einem gepflegten Absturz ein jähes Ende findet.

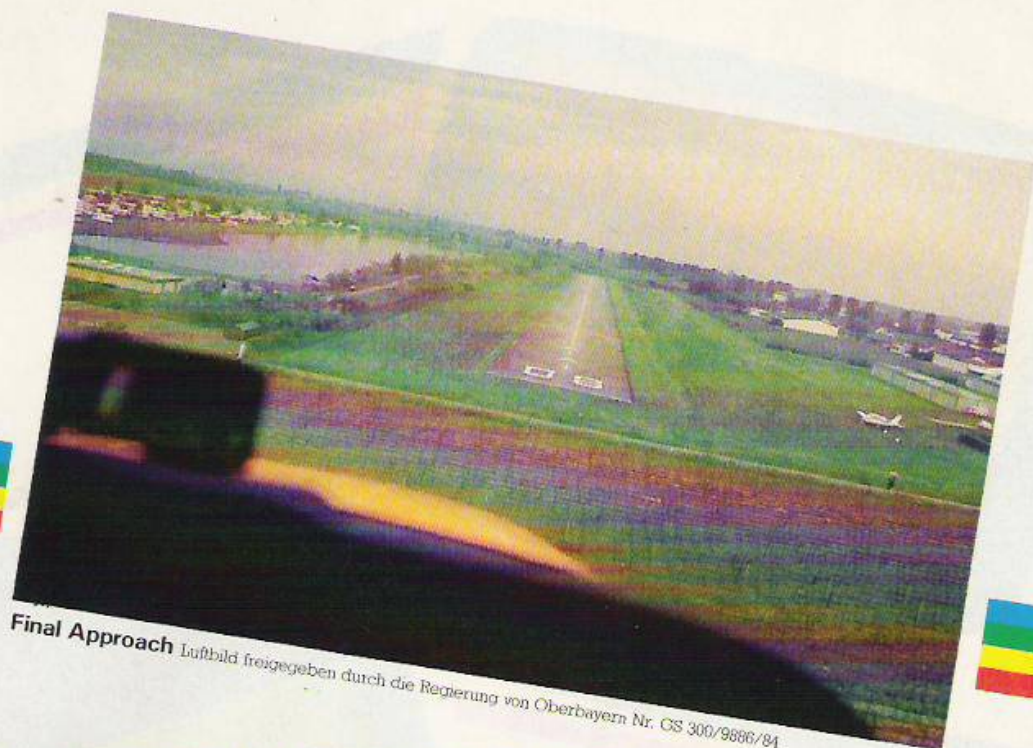
Kurz und schmerzlos gesagt: »Fighter Pilot« ist ein Knüller. Ein sehr schnelles Programm mit Spannung und Atmosphäre. Das Flugzeug reagiert sehr schnell auf die Eingaben des Spielers, so daß der realistischere »Psion« gegenüber »Fighter Pilot« regelrecht gemächlich wirkt. Luftkampf mit Niveau.

## Der Langweiler: Heathrow

Koordination ist alles bei »Heathrow«. Wie der Programmname schon vermuten läßt, muß der Spieler auf dem Londoner Flughafen für Ordnung sorgen. Die landewilligen Flugzeuge stehen schon Schlange.

Mit Flugsimulationen hat »Heathrow« nur im weitesten Sinn etwas zu tun. Der stets gleiche, auf Dauer äußerst eintönige Screen, liefert ein schmuckloses Radarbild. Das Spiel ist kompliziert, die englische Anleitung noch komplizierter und Spaß macht es auch keinen. Fazit: Lieber nicht.

Wer nach noch »Höherem« strebt, als über den Wolken zu schweben, schwenkt in die Erdumlaufbahn ein, um mit dem »Space Shuttle« einen



**Final Approach** Luftbild freigegeben durch die Regierung von Oberbayern Nr. GS 300/9886/84



**Jumbo in Heathrow**

defekten Satelliten zu bergen. Die Anleitung zu diesem Programm ist, wie es sich für eine Flugsimulation gehört, ebenso lang wie ausführlich geraten: neun engbeschriebene A4-Seiten. Nach Programmstart wird zwischen Keyboard- und Joystick-Steuerung gewählt und dann geht's richtig los.

Die Startrampe erscheint in manierlicher Grafik, der Countdown wird heruntergezählt und... das Shuttle startet. Ganz von allein. Ohne daß ich irgendein Knöpflein gedrückt hätte. Die Anleitung klärt auf: Der gesamte Start wird vom Autopiloten durchgeführt. Leicht frustriert ob dieser Bevormundung, beobach-

te ich, wie draußen die weißen Wölkchen vorbeiziehen und schließlich der Schwärze des Weltalls Platz machen. Nun darf ich erfreulicherweise auch mal was selbst machen, nämlich den zu bergenden Satelliten anfliegen. Einige Sekunden lang trudele ich umher, doch plötzlich — flacker, flacker! — greift der elende Autopilot wieder ein. Denn das ist so ein Pferdefuß im gesamten Programm: Macht man während eines Flugabschnitts etwas verkehrt, so stürzt man nicht ab oder explodiert (wie es sich für eine anständige Simulation hören würde). Nein, Big Autopilot is watching you. Der Computer führt das Manöver von sich aus

zu Ende — in allen Flugphasen. Da nützt es auch nichts, daß es für gemeisterte Abschnitte Punkte gibt.

Der Versuch, Flugsimulation und Punktejagd zu verknüpfen, hat mich nicht sonderlich befriedigt. Alte Flugfans werden sich etwas veräppelt vorkommen, wenn ihnen dieser lästige Autopilot dauernd in die Manöverversuche reinfuscht. Aber vielleicht lassen sich die Spieler, die mit dem üblichen Ablauf solcher Simulationen nichts anfangen konnten, von der Punkteklauberei begeistern.

## Der Düstere: Nightflite II

Einer der wenigen Simulatoren, die bereits auf der 16-KByte-Version laufen, ist dieser Linienflug für Nachtschwärmer. Es handelt sich um ein ganz »normales« Flugprogramm ohne irgendwelche kämpferischen oder astronomischen Einlagen. Ein Flugzeug soll — crashfrei — landen, starten und durch die Lüfte ziehen. Wie der Name bereits befürchten läßt, geschieht dies alles bei Nacht. Haken dieser Tageszeit: Die Grafik beim Blick aus dem Cockpitfenster ist ausgesprochen mickrig ausgefallen. Lediglich einige Flughafenlichter blinken einsam in der nächtlichen Finsternis.



Wo bin ich? — Karte des »Fighter Pilot«



Suche nach dem Feind: »Fighter Pilot«

Die Anordnung der Instrumente ist recht übersichtlich geraten und das Menü erlaubt eine Handvoll Variationen. Und bei einer Bruchlandung wird der Pilot ausführlich informiert, was zum unerfreulichen Bodenkontakt führte. »Nightflite II« leidet unter der mickrigen Grafik, bietet aber Besitzern eines 16-KByte-Spectrums durchaus ansprechende Flugunterhaltung. Im Vergleich zum »Psion« oder »Fighter Pilot« fehlt es aber an Atmosphäre und dem rechten Biß.

## Der Schlawfi: Airliner

Machen wir es kurz und schmerzlos: »Airliner« bietet selbst für 16 KByte recht wenig. Die Grafik scheint aus der Steinzeit der Computersoftware zu stammen und die Beeinflussung vonseiten des Spielers ist dürftig. Fazit: Es gibt nur bessere.

## Der Komplizierte: Air Traffic Control

Und zu guter Letzt noch ein »Seitenspringer«: Die Luftkontrolle ist

kein reinrassiger Flugsimulator, sondern vielmehr ein Strategiespiel für den geduldigen Denker. Ähnlich wie bei »Heathrow« schlüpft der Spieler in die verantwortungsvolle Rolle eines Fluglotsen, der den eiligen Luftverkehr koordiniert. Der Kassette liegen ein glücklicherweise ausführliches Handbuch und — zu meiner großen Freude — eine Schablone bei, die über die Spectrum-Tastatur gelegt wird. Dieses Schablönchen trägt sehr zur Übersicht über die zahlreichen Kommando-Belegungen bei. Ein vorbildliches Plus, das allerdings nicht über Zusatztastaturen paßt. Dieses umfangreiche Programm ist ein Tip für Liebhaber von komplexen, geruhsamen Spielen. Die grafische Gestaltung ist schlicht und übersichtlich, spielt aber nur eine Nebenrolle. Mehrere Schwierigkeitsstufen stehen zur Auswahl. Im Vergleich zu »Heathrow« ein deutlicher Fortschritt, für meinen Geschmack jedoch nicht allzu aufregend und auf Dauer einschläfernd.

Legen Sie nun die Sicherheitsgurte wieder an, denn wir beenden unseren himmlischen Trip in die Welt der Flugprogramme mit einer butterweichen Landung. Da jedes Spiel vom anderen abweicht und sowieso alles Geschmackssache ist, fällt es schwer, ein eindeutiges Fazit zu ziehen. Aber wenn Sie mich fragen würden, welche Kassetten ich mit auf eine einsame Insel nehmen würde, hieße die Wahl »Psion« und »Fighter Pilot«. Die meisten anderen Programme sind auch ihr Geld wert, wenn es auch die eine oder andere Bauchlandung gab. Und da wir gerade bei scherzigen Wortspielen sind: Das absolute »Absturz-Programm« war in jeder Hinsicht Airliner, der bei Falscheingaben hin und wieder mal abstürzt — im programmtechnischen Sinne.

(Heinrich Lenhardt)

# EINFACH SUPER!

Entdecken Sie jetzt mit  
»Happy-Computer«  
die Welt der  
Heimcomputer!

»Happy-Computer« bringt  
jeden Monat neue, tolle  
Tips und interessante Infor-  
mationen über alle wichtigen  
Heimcomputer-Systeme

»Happy-Computer«  
testet und bringt  
für Sie die Listings der  
schönsten Heimcomputer-  
Programme

»Happy-Computer«  
zeigt Software-  
Tests sowie pfiffige  
Programmier- und  
Anwendungsbei-  
spiele

»Happy-  
Computer«  
hat den großen  
privaten Kleinan-  
zeigen-Markt



Hardware-Tests sind ein wichtiger Bestandteil von Happy Computer. Auch Peripherengeräte, wie hier die »Daten-Recorder«, werden von der Redaktion getestet. Wenn Sie wissen wollen, welches Zubehör für welchen Heimcomputer geeignet ist, sollten Sie »Happy Computer« lesen.



Nicht nur neue Heimcomputer werden von »Happy-Computer« ausführlich getestet, sondern auch die jeweiligen Erweiterungen: Happy Computer berichtet über nachträglich aufgerüstete Heimcomputer.

»Happy-Computer« ist eine Publikation der

## Markt & Technik

Verlag Aktiengesellschaft

Hans-Pinsel-Str. 2, 8013 Haar, Tel. 0 89/46130

# Das Interface 1

Bereits bevor es im Handel erhältlich war, hat es viele Spekulationen über das Interface 1 gegeben. Als es dann erschien, gab es eine Reihe von Testberichten, aber nur wenig Information über Technik und Benutzung des Interfaces. Es ist auch viel Falsches geschrieben worden. In Happy-Computer (März 1984, Seite 18 ff) wird zwar der Microdrive ausführlich besprochen, über das Interface 1 jedoch wird wenig gesagt. Das möchte ich an dieser Stelle nachholen.

## Hardware und Maschinencode

Um zu verstehen, wie das neue ROM benutzt wird, müssen wir uns zuerst ein bißchen mit dem Z80-Mikroprozessor beschäftigen. Bild 2 zeigt die Anschlußbelegung des Z80. Uns interessiert nur die Speicher- und I/O-Kontrolle.

Was geschieht, wenn der Z80 einen Befehl lesen will? Mit dem Signal »MREQ« (Memory request) wird eine Speicheranforderung angezeigt. Gleichzeitig heißt dieses Signal, daß auf dem Adreßbus eine gültige Adresse liegt. Außerdem wird über M1 (Maschinenzyklus 1) angezeigt, daß ein Befehl gelesen werden soll (englisch: instruction fetch) und nicht etwa Daten.

## Das Interface-ROM

Das Interface 1 wertet diese Signale aus. Wenn über »MREQ« eine Speicheranforderung angezeigt wird und über »M1« ein »instruction fetch« signalisiert wird und eine bestimmte Adresse auf dem Adreßbus liegt, dann wird das Spectrum-ROM abgeschaltet und das Interface 1-ROM eingeschaltet. Wie das geschieht? In Abbildung 3 sieht man die Belegung des Userports des Spectrum; da gibt es den Anschluß »ROMCS« (ROM chip select). Über diese Steuerleitung kann das Spectrum-ROM ein- und ausgeschaltet werden. Es war also von vorneherein vorgesehen, das ROM des Spectrum zu erweitern.

Wenn das Interface 1 angeschlossen ist, wird also zwischen den beiden ROMs ständig hin- und hergeschaltet. Um im Detail zu verstehen, wie das Betriebssystem des Spectrum hin- und herschaltet, müssen wir uns mit Maschinensprache befassen.

Verbindung zum Microdrive	RS232-Schnittstelle	»Netzwerk«-Anschlüsse zu weiteren Spectrum-Einheiten	Bus-Anschluß: Hier wird der Spectrum angesteckt
---------------------------	---------------------	--	---

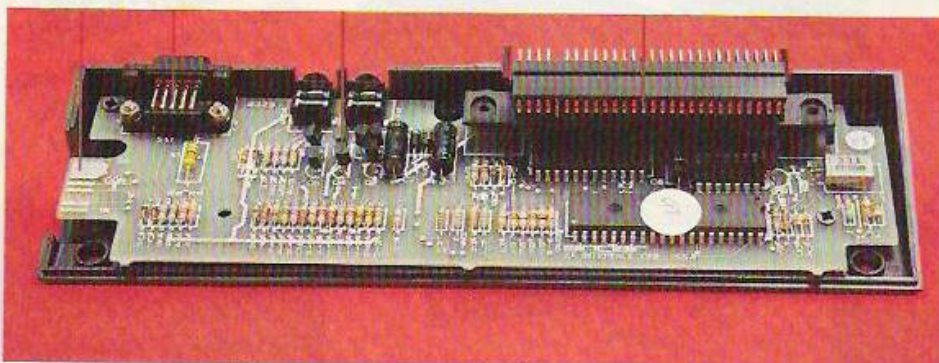


Bild 1. Innenleben des ZX-Interface 1

Zuerst möchte ich etwas zum Aufbau des Interface 1 sagen. Bild 1 zeigt das Innenleben des Interface. Man sieht den ULA-Chip und teilweise durch den Bus-Anschluß

verdeckt, links dahinter das neue ROM. Jawohl, das Interface 1 enthält ein 8-KByte-ROM, so daß der Spectrum mit Interface über stolze 24 KByte ROM verfügt.



Der Sprungbefehl »RST08« (das ist ein 1-Byte-Befehl, der an die Adresse 08H springt) wird zur Fehlerbehandlung benutzt. Das Byte, das auf den »RST08«-Befehl folgt, enthält den Fehlercode für die Fehlermeldung (FFH für o.k.; 00H für NEXT without FOR; 01H für Variable not found und so weiter).

Der »instruction fetch« auf die Adresse 08H bewirkt die Umschaltung auf das neue ROM. Das heißt, wann immer ein Fehler auftritt, wird auf das neue ROM umgeschaltet. Was das neue ROM dann macht, hängt von dem Wert in dem Datenbyte nach »RST08« ab.

## Der Datenfluß

Die Werte 00H bis 1AH (die alten Fehlermeldungen ohne o.k.) bewirken, daß im neuen ROM nochmals überprüft wird, ob der Befehl jetzt in Ordnung ist. Dadurch werden die Befehle MOVE, ERASE, CAT, FORMAT jetzt zum Leben erweckt beziehungsweise einige Befehle (LOAD, PRINT, INPUT, MERGE, SAVE, OPEN#, CLOSE#) erweitert und ergänzt.

Der Wert FFH bringt die neue Meldung »Program finished«. Darüber hinaus produziert das neue ROM auch noch neue Fehlermeldungen, wenn zum Beispiel die neuen Befehle falsch benutzt werden (Interface 1-Handbuch, Anhang 5).

Interessant sind die Werte 1BH bis 31H für das Datenbyte. Damit können gezielt neue Routinen im neuen ROM aufgerufen werden. So kann dann der Maschinencodeprogrammierer das neue ROM auch benutzen. Wer sich für Details interessiert, sei auf die Microdrive-Bücher verwiesen. Abbildung 4 zeigt eine Übersicht über diese Routinen.

Die Werte 33H bis FEH liefern die Fehlermeldung »Hook code error«, das heißt, falsches Datenbyte nach RST08. Der Wert 32H ist für Sinclair Research Ltd. reserviert.

## Die RS232-Schnittstelle

Über die Benutzung der Microdrives und des Netzwerkes gibt das Handbuch erschöpfende Auskunft. Deshalb braucht dazu nichts mehr gesagt werden. Ganz anders ist es dagegen mit der RS232-Schnittstelle. Um damit vernünftig zu arbeiten, muß man mehr wissen als im Handbuch steht.

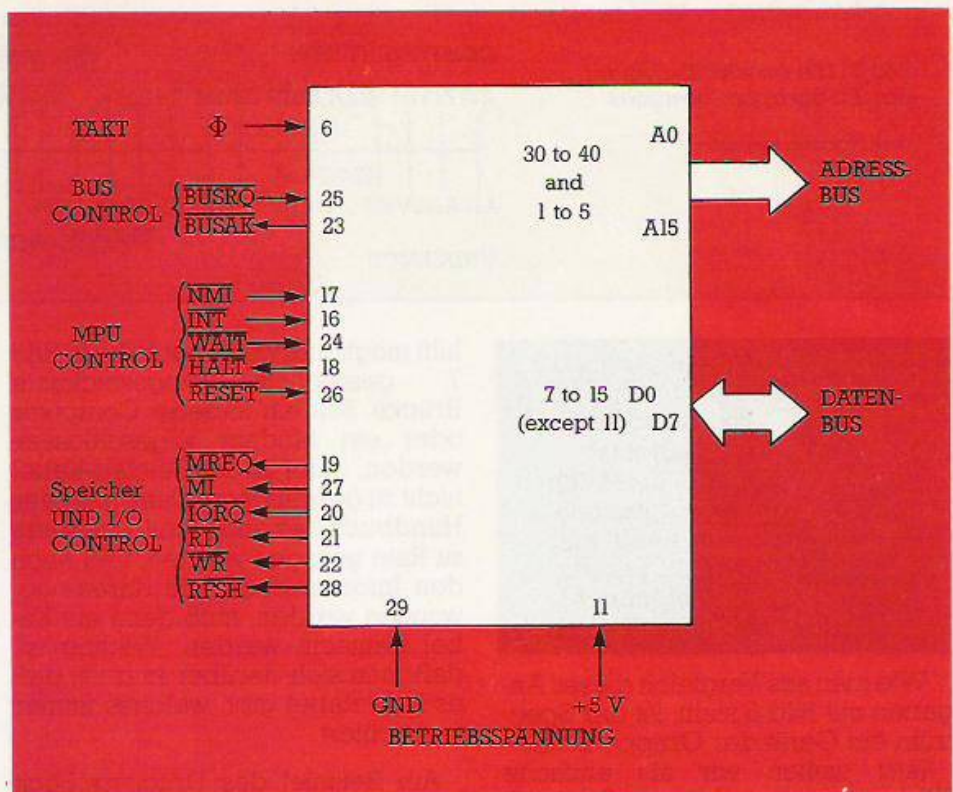


Bild 2. Die Anschlußbelegung des Z80-Mikroprozessors

Quelle: R. Zaks, Programmierung des Z80

Dazu erst einmal ein wenig Schnittstellen-Theorie: Die RS232-Schnittstelle ist eine serielle Schnittstelle, das heißt, die Bits eines Byte werden nacheinander und nicht gleichzeitig übertragen.

Dazu kommt, daß ein Byte vorneweg ein Startbit bekommt und hinten werden ein oder zwei Stoppbits angehängt.

Das ist alles noch einzusehen. Komplett wird die Verwirrung, wenn man sich die Belegung der Stecker ansieht. Um uns da Klarheit zu verschaffen, müssen wir vorne anfangen.

Bei der RS232-Schnittstelle werden alle Geräte in zwei Gruppen eingeteilt:

1. DCE = Data Communication Equipment, das sind Rechner, Modems und so weiter
2. DTE = Data Terminal Equipment, das sind Terminals, Drucker und so weiter

Aber es gibt viele Ausnahmen. Man muß fast immer das Handbuch zu Rate ziehen, um zu entscheiden, was man hat. Bild 5 zeigt die wichtigsten Anschlüsse einer RS232-Schnittstelle. Die Pin-Bezeichnung

gen beziehen sich auf den 25poligen Miniatur-D-Stecker, der Standard für diese Schnittstelle ist.

Dann muß man noch wissen: Low = logische Eins = Spannung zwischen -3 V und -15 V  
High = logische Null = Spannung zwischen +3 V und +15 V

Das klingt falsch, ist aber so richtig. Der Bereich zwischen -3 V und +3 V ist nicht definiert.

Das ist dann auch schon die Hardware der Standard-Schnittstelle. Daß die Spectrum-Schnittstelle mit ihrem 9poligen Stecker davon abweicht, ist offensichtlich.

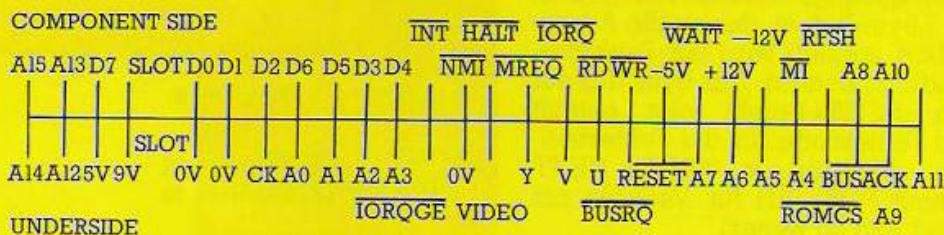
Jetzt kommt die entscheidende Frage: Wie wird etwas an den Spectrum über die RS232 angeschlossen? Dazu muß man wissen, was der Spectrum mit der Schnittstelle macht.

Die Anschlüsse des Spectrums sind:

PIN 2	Daten-Eingang (TX data)
PIN 3	Daten-Ausgang (RX data)
PIN 4	DTR-Eingang. Nur wenn das Signal »DTR HIGH« ist, kann der Spectrum senden. Wenn DTR »LOW« ist, dann sendet der Spectrum nicht.

**Bild 3. Die Anschlußbelegung des ZX-Spectrum-Userports**

Quelle: Spectrum-Handbuch



**PIN 5** CTS-Ausgang. Hier »sagt« der Spectrum, ob er empfangsbereit ist. »HIGH« heißt, ich kann empfangen, »LOW« heißt, ich bin nicht bereit.

**PIN 9** +9 V ist immer HIGH

Wie man aus Vergleich dieser Angaben mit Bild 5 sieht, ist der Spectrum ein Gerät der Gruppe DCE.

Jetzt wollen wir als einfache Übung ein Kabel herstellen, mit dem wir dann zwei Spectrum verbinden. Dazu brauchen wir ein 4adriges abgeschirmtes Kabel und zwei 9polige D-Stecker. Bild 6 zeigt, wie die Verbindungen gelötet werden müssen. Da man den Eingang des einen Spectrum mit dem Ausgang des anderen verbinden muß, kreuzen sich die Leitungen.

Haben wir das Kabel gelötet, können die beiden Computer verbunden werden; dann wird ein »b«-Kanal geöffnet, und die Programme können, wie im Handbuch beschrieben, ausgetauscht werden.

Kompliziert wird es, wenn wir den Spectrum mit einem anderen Gerät, zum Beispiel Drucker, Modem oder gar einem anderen Computer verbinden wollen. Dazu gibt es von Sinclair ein spezielles Kabel. Bild 7 zeigt die Verdrahtung. Das Sinclair-Kabel sollte sicherlich in den meisten Fällen richtig sein, wo ein Drucker oder Terminal angeschlossen werden soll. In einigen Fällen

hilft möglicherweise auch die in Bild 7 gestrichelt eingezeichnete Brücke. Soll ein anderer Computer oder ein Modem angeschlossen werden, dann ist das Sinclair-Kabel nicht zu gebrauchen. Dann muß das Handbuch des anderen Computers zu Rate gezogen werden, und nach den Informationen, die daraus gewonnen werden, muß dann ein Kabel gemacht werden. Wichtig ist, daß man sich darüber klar ist, daß es kein Kabel gibt, welches immer funktioniert.

Am Beispiel des Druckers Facit 4510 möchte ich versuchen klarzumachen, was zu tun ist. Bild 8 zeigt die Anschlußbelegung der RS232-Schnittstelle des Druckers, wie sie im Handbuch steht. Auch die Pfeile stehen so im Handbuch (leider sind nicht alle Handbücher so deutlich).

Benutzt man das Sinclair-Kabel und sind alle Parameter am Drucker (zum Beispiel Parität, Zahl der Stopbits, Baudrate) richtig gesetzt, so kann man drucken. Der Spectrum sendet immer, da er immer auf PIN 4=DTR das Signal auf HIGH hat (Bild 8: PIN 20 DTR immer HIGH). Der Drucker signalisiert jedoch auf PIN 19 über Ready/Busy, ob er bereit ist oder nicht. Der Spectrum merkt davon nichts, da PIN 19 nicht angeschlossen ist. Es können also Zeichen verloren gehen, wenn der Drucker Busy signalisiert, der Spectrum aber weiter sendet. Man sollte deshalb ein Kabel wie in Bild 9 verwenden.

Durch entsprechende Überlegungen muß man sich auch ein Kabel maßschneidern, um ein Verbindungskabel zu einem anderen Computer herzustellen. Das Prinzip, wie man so etwas macht, sollte jetzt klar sein.

Problematisch wird die Sache, wenn der Computer kein Hardware-Protokoll für das Handshaking benutzt. (Man spricht von einem Hardware-Protokoll, wenn die beiden Geräte über spezielle Leitungen durch Signalpegel anzeigen, ob sie bereit sind oder nicht.) Bei einem Software-Protokoll werden spezielle Zeichen über die Datenleitung geschickt. Die wichtigsten sind:

**XON/XOFF**

Es werden die ASCII-Zeichen DC1 und DC3 für ready/busy benutzt.

**ETX/ACK**

Es werden die Zeichen ETX und ACK benutzt.

(Siehe Bild 10 für die ASCII-Kontrollzeichen.)

Bei einem Software-Protokoll muß der Programmierer für das Senden der Kontroll-Codes an der richtigen Stelle Sorge tragen. Außerdem muß er natürlich, falls die Kontroll-Codes empfangen werden, für die richtige Reaktion des Programms sorgen. Die Details hierzu würden bei weitem den Umfang dieses Artikels sprengen.

(R.W. Gerling)

Datenbyte	Funktion
IBH	Tastatur Eingabe
ICH	Bildschirm-Ausgabe (Strom 2)
IDH	RE232 Input
IEH	RS232 Output
IFH	Drucker-Ausgabe (wie ICH, nur Strom 3)
2OH	Tastatur-Test (ob Taste gedrückt)
2lh	Startet und stoppt Microdrive-Motoren
22H-2CH	Microdrive-Routinen zum Anlegen, Lesen und Schreiben von Files
2DH-2EH	Öffnen und Schließen des Netzwerk-Kanals
2FH-30H	Empfangen und Senden eines Netzwerk-Pakets
3IH	Anlagen der neuen Systemvariablen

Quellen:  
Rodnay Zaks, Programmierung des Z80, Sybex-Verlag, Düsseldorf, 1982  
Spectrum-Handbuch, Interface 1-Handbuch, Sinclair Research Ltd.

Jan Logan, Spectrum Microdrive Book, Melbourne House, Tring, 1983, deutsche Ausgabe: Das Microdrive-Universum, Hueber Verlag, München, 1984

Jan Logan und Frank O'Hara, The Complete Spectrum ROM Disassembly, Melbourne House, Tring, 1983  
Das Modem-Sonderheft, MC, Franzis-Verlag, München  
Handbuch des Druckers Facit 4510

**Bild 4. Die neuen ROM-Routinen, die mit RST08 aufgerufen werden können.**

Quelle: Jan Logan, Spectrum Microdrive Book

PIN	Name	DCE	DTE
1	Chassis Ground Schutzwerde	—	—
2	TX data Sendedaten	Ein	Aus
3RX data	Empfangsdaten	Aus	Ein
4	Request to Send (RTS)	Ein	Aus
5	Clear to Send (CTS)	Aus	Ein
6	Data Set Ready (DSR)	Aus	Ein
7	Signal Ground Signalerde	—	—
20	Data Terminal Ready (DTR)	Ein	Aus

Bild 5. Die wichtigsten Anschlüsse einer RS232-Schnittstelle und die Kennzeichnung, ob Ein- oder Ausgang.  
Quelle: Modem-Sonderheft

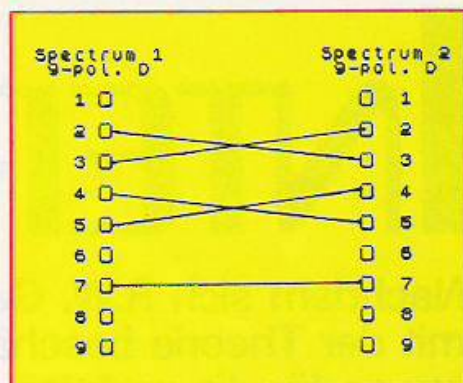


Bild 6. Ein Kabel zum Verbinden von zwei Spectrum-Computern über die RS232-Schnittstelle. Zur Verbindung von PIN 7 wird die Abschirmung verwendet

## 7-bit ASCII Code Tafel

### Kontroll Zeichen:

00	NUL	null
01	SOH	start of heading
02	STX	start of text
03	ETX	end of text
04	ENO	end of transmission
05	ENQ	enquiry
06	ACK	acknowledge
07	BEL	bell
08	BS	backspace
09	HT	horizontal tabulation
0A	LF	linefeed
0B	VT	vertical tabulation
0C	FF	form feed
0D	CR	carriage return
0E	SO	shift out
0F	SI	shift in
10	DL	data link escape
11	DC4	device control 4
12	DC3	device control 3
13	DC2	device control 2
14	DC1	device control 1
15	NAK	negative acknowledge
16	SYN	synchronous idle
17	ETB	end of transmission block
18	CAN	cancel
19	EM	end of medium
1A	ESC	escape
1B	FS	file separator
1C	GS	group separator
1D	RS	record separator
1E	US	unit separator

### Druckbare Zeichen (ZX-Spectrum):

	2	3	4	5	6	7
0	0	1	2	3	4	5
1	6	7	8	9	A	B
2	C	D	E	F	G	H
3	I	J	K	L	M	N
4	O	P	Q	R	S	T
5	U	V	W	X	Y	Z
6	[	\	]	^	_	`
7	a	b	c	d	e	f
8	g	h	i	j	k	l
9	m	n	o	p	q	r
A	s	t	u	v	w	x
B	y	z	{		}	~
C	`	!@	#	\$%	&'	(
D	)	*+,-	.	/:;	<=>	?@
E	[	^_`	abc	def	ghi	jkl
F	mno	pqr	stuv	wxyz	{ }	~
G	`	!@	#	\$%	&'	(
H	)	*+,-	.	/:;	<=>	?@
I	[	^_`	abc	def	ghi	jkl
J	mno	pqr	stuv	wxyz	{ }	~
K	`	!@	#	\$%	&'	(
L	)	*+,-	.	/:;	<=>	?@
M	[	^_`	abc	def	ghi	jkl
N	mno	pqr	stuv	wxyz	{ }	~
O	`	!@	#	\$%	&'	(
P	)	*+,-	.	/:;	<=>	?@
Q	[	^_`	abc	def	ghi	jkl
R	mno	pqr	stuv	wxyz	{ }	~
S	`	!@	#	\$%	&'	(
T	)	*+,-	.	/:;	<=>	?@
U	[	^_`	abc	def	ghi	jkl
V	mno	pqr	stuv	wxyz	{ }	~
W	`	!@	#	\$%	&'	(
X	)	*+,-	.	/:;	<=>	?@
Y	[	^_`	abc	def	ghi	jkl
Z	mno	pqr	stuv	wxyz	{ }	~

Bild 10. Die 32 ASCII-Kontroll-Zeichen und die ASCII-Codetafel für die druckbaren Zeichen des Spectrum.

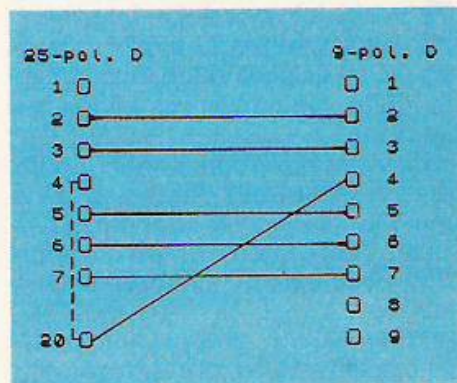


Bild 7. Das Sinclair RS232-Kabel

Quelle: Interface 1-Handbuch

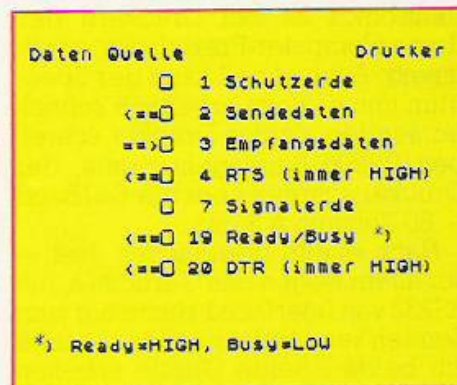


Bild 8. Anschlußbelegung der wichtigsten Anschlüsse der RS232-Schnittstelle des Druckers Facit 4510  
Quelle: Handbuch Facit 4510

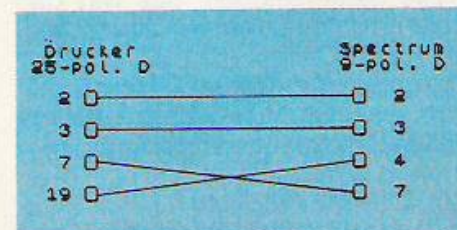


Bild 9. RS232-Kabel zum Anschluß des Druckers Facit 4510 an den ZX-Spectrum

# INTERFACE

1

**Nachdem sich R.W. Gerling hauptsächlich mit der Theorie beschäftigt hat, hier nun etwas für die praktische Arbeit mit der RS232-Schnittstelle, dem Microdrive und Tasword auf Cartridge.**

**A**ls ich die Frage beantworten mußte, über welche Schnittstelle ich meinen Drucker ansteuern wolle, entschied ich mich für eine RS232. Und zwar aus folgenden Gründen:

— Eine parallele Schnittstelle ist nur für die Steuerung eines Druckers geeignet. Ich wollte aber für einen später geplanten Akustikkoppler nicht noch eine Schnittstelle kaufen müssen.

— Das Kabel einer Centronics-Schnittstelle ist in der Länge sehr begrenzt und damit auch der Abstand Computer — Drucker. Ein RS232-Kabel kann aber sehr lang sein, 10 m und mehr.

— Die höhere Schnelligkeit einer Centronics ist bei Druckern der Home-Computer-Preisklasse zweitrangig. Auch seriell kann der Spectrum immer noch erheblich schneller senden als der Drucker schreiben (19200 Baud geht prima, der Drucker schreibt aber nur 640 Baud — 80 Zeichen à 8 Bit).

Nach einem gründlichen Test — vor allem wegen des Gerüchtes, die RS232 von Interface I könne nur zum Senden verwendet werden — kaufte ich beides. Beide Geräte arbeiten einwandfrei, nachdem die richtige Kabelverbindung hergestellt war. Mein Drucker ist ein Magnasonic PDH 8, tauglich zu Epson FX80, jedoch mit etwas anderen Steuerzeichen.

Die Steuerzeichen, das war das erste Problemchen. Sie haben häufig ASCII-Codes unter 32, was der Spectrum mit »Invalid Colour« quittiert. Deshalb habe ich für mich ein Mini-Drucker-Initialisierungsprogramm geschrieben:

```
»10 FORMAT "t";9600«
```

Baudrate entsprechend der Druckereinstellung festlegen. Interface I kann alle benötigten Daten senden und empfangen. Es ist egal,

**Der Spectrum mit angegeschlossenem Interface 1, Mikrodrive und Interface 2**

ob ich bei FORMAT den »b«- oder den »t«-Kanal anspreche.

```
20 OPEN #3;"t"
```

Strom 3 für den »t«-Kanal öffnen. Strom 3 deshalb, weil er LLIST und LPRINT verarbeitet. Dies sind Kürzel für LIST #3; und PRINT #3;.

```
30 OPEN #4;"b"
```

Strom 4 für binären Datentransfer öffnen. Über den »b«-Kanal können die Steuerzeichen gesendet werden, zum Beispiel PRINT #4; CHR\$ 27; CHR\$ 51; CHR\$ 50; (bei mir wird damit der Zeichenabstand festgelegt.) In den folgenden Zeilen sende ich die »Standard-Steuerzeichen«: Zeilenabstand, Schreibbreite, Zeilenanzahl je Seite, Anzahl der Leerzeilen (Perforation).

## Druckausgabe über die RS232-Schnittstelle

Das Programmchen muß allerdings zeilenweise gelöscht werden; NEW schließt die Ströme (obwohl das nicht schlimm ist, Hauptsache, der Drucker hat die Steuerzeichen empfangen).

Wenn man in einem Programm OPEN #-Befehle benutzt, sollte man am Programmfang auf jeden Fall CLOSE # einsetzen. Es gibt nämlich eine Fehlermeldung, wenn man versucht, einen bereits geöffneten Strom zu öffnen; ein Schließen eines nicht geöffneten Stroms führt der Computer jedoch aus.

Wichtig ist auch, daß der Drucker ONLINE ist. Der Computer wartet

nämlich auf das READY-Signal (erkennbar am schwarzen Bildschirmrand).

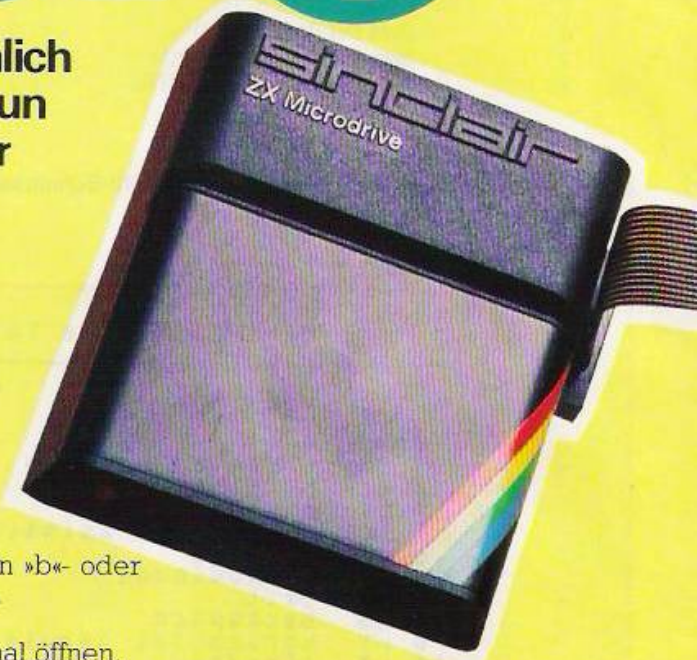
Wer einen Normalpapier-Drucker hat, möchte sicher gerne auch Tasword II auf diesem Drucker benutzen können. Die mitgelieferte Druckersoftware ist jedoch für Centronics geschrieben, also unbrauchbar. Meine Anpassung ist noch fehlerhaft, ich muß noch ein wenig tüfteln.

Eine COPY-Routine ist in diesem Heft veröffentlicht. H. Ritz teilt meine Meinung bezüglich Schnittstellen. Er arbeitet auch schon länger mit Interface I.

Ich habe diese Routine mit den passenden Steuercodes so geschrieben, daß ich sie mit einem kurzen Basic-Teil immer MERGEN kann. Allerdings habe ich sie auf Strom 4 (siehe oben) gelegt: POKE RAMTOP +3, 4, damit ich die Ströme nicht umschalten muß (Listing 1).

In Zeile 9810 muß für lfach/4fach-Copy ein Wert entsprechend der Beschreibung eingegeben werden. Zeile 9830 muß für den 16-KByte-Spectrum geändert werden.

Ab Zeile 9840 sende ich meine normalen Steuercodes: Zeilenabstand, Zeilen je Seite, davon Leerzeilen (Perforation), Zeichen je Zeile. Der Formularvorschub muß hier von Hand durchgeführt werden. Das funktioniert hervorragend.



# PRAXIS



Diese Routine habe ich natürlich auf Microdrive, ebenso wie »DRINT«. Es sind Arbeitsprogramme, und bei solchen fallen Wartezeiten durchaus ins Gewicht. Bei Spielen macht es nicht viel aus, wenn man einige Minuten Ladezeit hat, aber wenn es »weitergehen« soll, stört es doch.

Und damit sind wir beim 2. Teil: Microdrive.

Bei mir hat es bislang einwandfrei funktioniert. Ich mache zwar jedes Mal ein VERIFY (dauert ja nicht lange), aber bislang war es überflüssig. Auch Tasword II habe ich auf Cartridge. Es ist nicht weiter schwierig, es umzuschreiben. Nebenbei: Beim Formatieren meines neuen Cartridges hatte ich normalerweise 95 KBy-

te erreicht, oft 98 KByte, jeweils im ersten Anlauf. Übrigens kann ein Programm mit Laderoutine, Bild und so weiter durchaus länger sein als 48 KByte. Das heißt also, daß auch bei solchen Werten nicht unbedingt zwei volle Programme auf ein Cartridge passen müssen.

Eine andere Anwendung ist für Microdrive etwas interessanter als das Speichern von Spielen: Dateiverarbeitung. Es gibt durchaus einen Weg, nicht die ganze Datei la-

den und speichern zu müssen: Man nimmt einfach die Nummer des Datensatzes mit Hilfe der Funktion STR\$ in den Speichernamen des Satzes. Dadurch gewinnt man sehr viel Platz im Computer für ein wirklich »idiotensicheres« Programm, das alles kann. Aber auch die Programme kann man als Einzelteile auf

Cartridge SAVEn, jeweils eins lädt dann das nächste angewählte. So verfahren auch die »großen« Computerprogramme. Dies hat vor allem den Vorteil, daß maximal ein Datensatz verloren gehen kann. Auf diese Weise kann auch im Spectrum ein Programm laufen, das wirklich professionelle Möglichkeiten bietet.

Das Lösungsprinzip ist zwar einfach, die Ausführung braucht jedoch einige Zeit. Deswegen ist es noch nicht fertig. Das Microdrive wäre auch interessant für Programme, die eine Vielzahl von SCREENs verarbeiten müssen. Denn ein voller Bildschirmspeicher braucht 6912 Bytes. Auf ein Cartridge 95 KByte (= 97280 Bytes) passen dann ohne Programm 14 Bilder. Nicht schlecht, oder?

Trotz häufigem Gebrauch (= dauernd) ist bei mir noch kein Cartridge defekt. (Aber mein Computer wird auch nicht heiß; und die Originaltastatur tut es auch noch, auch die Schrift ist noch drauf. Und das trotz einem Jahr Dauerbetrieb, öfterem Herunterfallen und diversen Katzen-Spaziergängen. Ich weiß nicht, woran es liegt.)

Es gibt Computer-Benutzer, die mehr von Floppys halten. Nun, eine Floppy für den Spectrum kostet auch einen guten Tausender. Auf eine Diskette geht mehr drauf, das stimmt. Aber wenn man den Platz nicht für Spiele, sondern — so wie ich — für Arbeitsprogramme und feste Programmteile (Subroutinen) benutzt, braucht es seine Zeit, bis ein Cartridge voll ist. Wann fängt der Punkt an, wo Floppys — abgesehen von dem bei Sekunden nun wirklich geringen Zeitunterschied — kostengünstiger sind?

1 Laufwerk: 1200 Mark, 1 Drive: 500 Mark, 1 Diskette 200 K: 5 Mark, 1 Cartridge (95 k): 19,50 Mark. Bei 20 Programmen brauche ich maximal 960 KByte Platz. Das sind 11 Cartridges oder 5 Disketten.

Nun, bis es soweit ist, dauert es wirklich seine Zeit. Die Leute, die solche Mengen verarbeiten, benutzen normalerweise keinen Spectrum (obwohl ich zwei kenne, die ihre komplette Firmenverwaltung mit Spectrum + Kassettenrecorder machen). Es geht halt alles. Und selbst so ist der Computer immer noch schneller als der Mensch. Das soll übrigens nicht heißen, daß ich etwas gegen Floppys habe! Nur im

Verhältnis zum Preisunterschied steht der Leistungsunterschied nicht.

Noch ist keine Software auf Cartridge erhältlich. Sollten aber genügend Leute das Microdrive haben, kommt sicher auch das. Bis dahin gibt es ein Programm (TRANS EX-PRESS), das das Kopieren von Kassette zum Microdrive und umgekehrt erlauben soll. Ich habe es zunächst mit meinen eigenen Programmen versucht (SENSO). Das ging ganz gut (meistens). Die SAVE- und LOAD-Befehle wurden automatisch angepaßt.

Nicht funktioniert hat es mit Tasword, obwohl dieses Programm in keiner Weise LIST-geschützt ist. Auch bei kommerziellen Spielen hatte ich Probleme.

Von Microdrive zu Kassette (BACK UP-Kopie) machte ähnliche Schwierigkeiten. Hier werden SAVE und LOAD nicht geändert. Als Nachteil empfinde ich, daß dieses Programm

- aus mehreren Einzelprogrammen besteht
- nicht auf Cartridge erhältlich ist (Utility!)
- nicht mit sich selbst kopiert werden kann
- es nicht fehlerfrei läuft.

Allerdings hat es bisher noch keine Konkurrenz...

Nun, das Microdrive ist noch verhältnismäßig neu; wie bei allem Neuen folgt die Software erst nach einiger Zeit. Mein System arbeitet gut und mit ein bißchen Tüftelgeist werden sicher weitere Möglichkeiten gefunden. Hier sind auch und gerade für Hobby-Programmierer noch viele Möglichkeiten offen.

## Anpassung von Tasword an Microdrive und RS232-Drucker

Da Tasword ein echtes Arbeitsprogramm ist, das ich sehr oft brauche, wollte ich es natürlich gerne auf Cartridge speichern sowie an meinen neuen, für Textverarbeitung geeigneten Drucker anpassen.

Es ist recht einfach, Änderungen in das Programm einzubauen, denn es ist ausdrücklich vorgesehen, daß der Benutzer dieses Arbeitsmittel an seine Geräte anpaßt.

```
10 CLEAR #: FORMAT "b";9600: 0
PEN #3;"b"

11 CLS : LET a=USR 64330: GO T
0 20

15 POKE 23609,2: CLEAR 31999:
LET q$="": GO SUB 4000: LOAD *"m
";1;"TAS 2"CODE : CLS : LET a=US
R 59081: GO TO 10

90 LET i=NOT PI: LET b=CODE a#
: IF b<97 THEN LET b=b+32
```

Die Anpassung an den Drucker war nicht weiter schwierig, nachdem H. Ritz (COPY-Programm) mir den »Dreh« verraten hat. Meine Anpassung in Programmform zeigt das entsprechende Listing.

Die Erklärung dazu:

In Zeile 10 werden eventuell vorher schon geöffnete Ströme gelöscht, die Baudrate eingesetzt (19200 geht auch, es muß nur dieselbe sein wie beim Drucker) und der »b«-Kanal auf Strom 3 gelegt. Über den »b«-Kanal können auch die Steuerzeichen gesendet werden.

Zeile 10 ist zusätzlich eingefügt, die ursprüngliche Zeile 10 ist jetzt Zeile 11.

Die ursprüngliche Zeile 11 enthielt eine BEEP-Routine, die aus Platzgründen entfällt.

In Zeile 200 habe ich die Steuerzeichen eingefügt, die mein Drucker braucht, damit der Formularvorschub stimmt. Das geht vermutlich auch anders (im Code), aber so war es am einfachsten. Damit wird der Drucker jedes Mal auf das »Tasword-Format« eingestellt für den Fall, daß ich zwischendurch etwas anderes gedruckt habe.

In den Zeilen 275 und 285 wurde LPRINT durch PRINT #3; ersetzt und aus Platzgründen IF c<>0 durch IF c.

In Zeile 290 wird nach dem Druck ein Formularvorschub durchgeführt.

Etwas schwieriger war es schon mit meiner Anpassung an das Microdrive. Das verbraucht recht viel Platz, weil ich für das Speichern von Textfiles vorgesehen hatte, zwischen Cartridge und Kassette wählen zu können. Doch der Code beginnt schon bei Adresse 32000. Daher habe ich einige Streichungen und Kürzungen vorgenommen. Es sind also ein paar Zeilen mehr.

```

180 IF i>NOT PI THEN PRINT AT 90 LET i=NOT PI: LET b=CODE a$
i,31; FLASH 1;CHR$ b;: GO TO 500 : IF b<97 THEN LET b=b+32
190 GO TO 80 1005 PRINT AT 8,0;"Name?": LET i
200 PRINT #3;CHR$ 27;CHR$ 51;CH =10: LET j0=0: GO SUB 6000
R$ 50;CHR$ 27;CHR$ 67;CHR$ 52;CH 1010 IF LEN a$>10 THEN LET a$=a
R$ 27;CHR$ 78;CHR$ 8: CLS : GO S $( TO 10)
UB 4000: PRINT AT 4,8;"PRINT OPT 1020 IF LEN a$=0 THEN LET a$="?
IONS": PRINT " just press ENTER ??
for default values given in
brackets"
275 LET c=PEEK 62470: IF c THEN
PRINT #3;CHR$ c
280 RANDOMIZE USR 60038
285 LET c=PEEK 62471: IF c THEN
PRINT #3;CHR$ c
700 CLS : LET i=8: ERASE "m";1;
"TAS 1": SAVE *"m";1;"TAS 1" LIN
E 15
710 ERASE "m";1;"TAS 2": SAVE *
"m";1;"TAS 2"CODE 54784,10751
770 PRINT AT 19,0;
780 VERIFY *"m";1;"TAS 1": PRIN
T AT 21,0;"tasword: basic O.K.;"
;AT 19,0;
790 VERIFY *"m";1;"TAS 2"CODE 5
4784,10751: PRINT AT 21,20;" m/c
ode O.K.": GO TO 25
1041 PAUSE NOT PI: IF x$="m" OR
x$="M" THEN GO TO 25
1050 PRINT AT 12,0;"Verify ";a$;
?": LET i=14: GO SUB 920: IF i=
0 THEN GO TO 20
1100 CLS : GO SUB 900: VERIFY a$
CODE b,a
1110 PRINT AT 21,6;"ok": GO TO 2
5
2000 CLS : PRINT AT 8,0;"Name?":
LET j0=NOT PI: LET i=16: GO SUB
6000
2020 INPUT "Cass/Micro? ";x$: IF
x$="C" OR x$="c" THEN GO TO 20
30
2022 PRINT AT 18,9;"Cartridge fe
rtig?": PAUSE NOT PI: LET b=FN p
(62216): LOAD *"m";1;a$CODE (a+b
),((FN p(62221)+22)*64-a): GO TO
10

```

Zeile 15 enthält den geänderten Ladebefehl; den Namen mußte ich ebenfalls ändern, weil auf Cartridge nicht zwei Files denselben Namen haben dürfen. GOSUB 11 entfällt (siehe oben).

Die erste SAVE-Routine beginnt bei Zeile 700:

Der ERASE-Befehl wurde eingebaut, um einer Fehlermeldung vorzubeugen. Es kann nämlich sehr wohl ein nicht vorhandenes File gelöscht, nicht aber ein vorhandenes überschrieben werden.

Das GOSUB 800 ist natürlich jetzt überflüssig, ebenso wie die Frage nach dem VERIFY und die daraus resultierenden Befehle.

Zeile 900 wurde etwas gekürzt, ebenso wie die Zeilen 1005, 1010, 1020. Dafür wurde die Zeile 1021 und 1022 sowie 1041 eingefügt. NOT PI steht übrigens kurz für 0.

Die notwendigen Tasword-Änderungen sind diesen Programmzeilen zu entnehmen

Die Speicherung von Konstanten nimmt sehr viel Platz in Anspruch. Analoge Änderungen wurden ab Zeile 2000 eingebaut:

Die Zeilen 2000 und 2020 sind gelöscht worden, dafür ist Zeile 2010 jetzt Zeile 2000, und Zeile 2020 und Zeile 2022 wurden eingefügt.

Es sollte bei der Arbeit mit Microdrive beachtet werden, daß kein Textfile von Cartridge mit " " geladen werden kann. Dafür ist in dieser Version keine Sicherung eingebaut worden.

In Arbeit befindliche Texte oder solche, die ich sehr oft brauche, speichere ich auf Cartridge. Sicherheitskopien oder Textfiles zum Verschieben (Warensendung!) auf Kasette.

Ich habe versucht, die Originalversion nicht allzu sehr zu ändern. Es sind für mich noch weitere Feinheiten einzubauen, doch das kann nur jeder für sich tun, denn diese Feinheiten sind abhängig vom Verwendungszweck.

(Erika Hölscher)

An einem kleinen Beispielprogramm möchte ich hier zeigen, wie man eine ganze Reihe von »Standard-Fehlern« vermeiden kann.

Die erste und nicht unbedingt einfachste Teilaufgabe besteht darin, möglichst genau und am besten schriftlich festzulegen, was das Programm überhaupt tun soll. Anders ausgedrückt: das umzusetzende Problem muß so genau wie es geht dargestellt werden. Hierbei (und auch bei späteren Teilaufgaben) ist es von Vorteil, wenn man jemanden zum »Ausprobieren« hat. Man versucht, ihm die Absicht zu erklären. Wenn er es versteht, ist schon viel gewonnen.

Machen wir das noch mal für unser Beispiel:

Problemstellung:

Das Programm soll den Benutzer im Bruchrechnen testen.

- Erweitern
- Kürzen
- Addieren, Subtrahieren
- Multiplizieren, Dividieren.

Das sieht nur auf den ersten Blick harmlos aus. Beim genaueren Hinsehen erkennt man, daß hier eine Reihe von Teilproblemen zu lösen sein wird: KGV (Kleinstes Gemeinsames Vielfaches), GGT (Größter Gemeinsamer Teiler), Eingaberoutinen, damit der Benutzer das Berechnen von Teilergebnissen nicht dem Computer überlassen kann. Aber soweit sind wir noch nicht. Jetzt kommt erst einmal der zweite Schritt:

## Das Flußdiagramm

Dies ist die Hauptarbeit beim Programmieren. Hier wird nicht nur der Programmablauf festgelegt, sondern — notwendigerweise — werden auch schon die wichtigsten Variablen bestimmt, denn die Art der Wertablage ist ja mitbestimmend für den Programmablauf.

Damit wir den groben Ablauf des geplanten Programms festlegen können, brauchen wir ein Flußdiagramm (Programmablaufplan, PAP). Dazu müssen wir natürlich die in solchen Diagrammen benutzten Symbole kennen.

Dazu gibt es ein paar kleine Regeln, die man beachten sollte:

- Der Verbindungspfeil mündet immer von oben in ein Symbol.
- Die Verwendung von Basic-Befehlen im PAP ist zu vermeiden. Das erleichtert es anderen, dieses Programm für ihren Rechner zu verwenden, weil sie besser verstehen können, was geschehen soll.

# So programmiert

**Basic ist derzeit die verbreitetste Computersprache. Vor allem die leichte Erlernbarkeit verleitet viele Hobby-Programmierer dazu, »einfach drauflos zu tippen«. Daraus folgt jedoch, daß die entstandenen Programme unübersichtlich und schwer zu korrigieren oder zu erweitern sind.**

— Der Plan sollte großzügig angelegt werden, um Platz für Ergänzungen und Bemerkungen zu haben.

— Sich kreuzende Verbindungspfeile sind zu vermeiden.

Jetzt nehmen wir uns einen Schmierzettel zur Hand und zeichnen unser Programm im groben auf. Schmierzettel deshalb, weil es sicher nicht im ersten Anlauf hinkommen wird.

Das Ergebnis nach zehn Skizzen habe ich in Bild 2 dargestellt.

Sicher sind hier verschiedene Lösungen möglich. An diesem Plan können wir aber jetzt schon erkennen, welche Programmteile die meisten Probleme aufwerfen werden: ich habe sie deshalb schon als Unterprogramme gekennzeichnet.

Diesen Teilen wenden wir uns nun zu. Zuerst müssen wir uns überle-

gen, wie unser Programm die Aufgabe auswählen soll. Das ist gleichzeitig mit der Lösungseingabe das Schwierigste. Für so schwierige Einzelteile erstellen wir eigene Flußdiagramme. Diese machen wir so aufgeschlüsselt wie nötig. Den Aufgabentyp haben wir ja am Anfang des Programms einzugeben. Von diesem Wert hängt es ab, wie es weitergeht. Schreiben wir uns doch mal auf, wie diese Aufgaben aussehen müssen:

Kürzen und Erweitern:

— Zähler und Nenner müssen gezeigt werden.

— Beim Erweitern muß außerdem der neue Nenner angezeigt werden.

Rechnen mit Brüchen:

— alle obigen Teilaufgaben, aber zweimal.

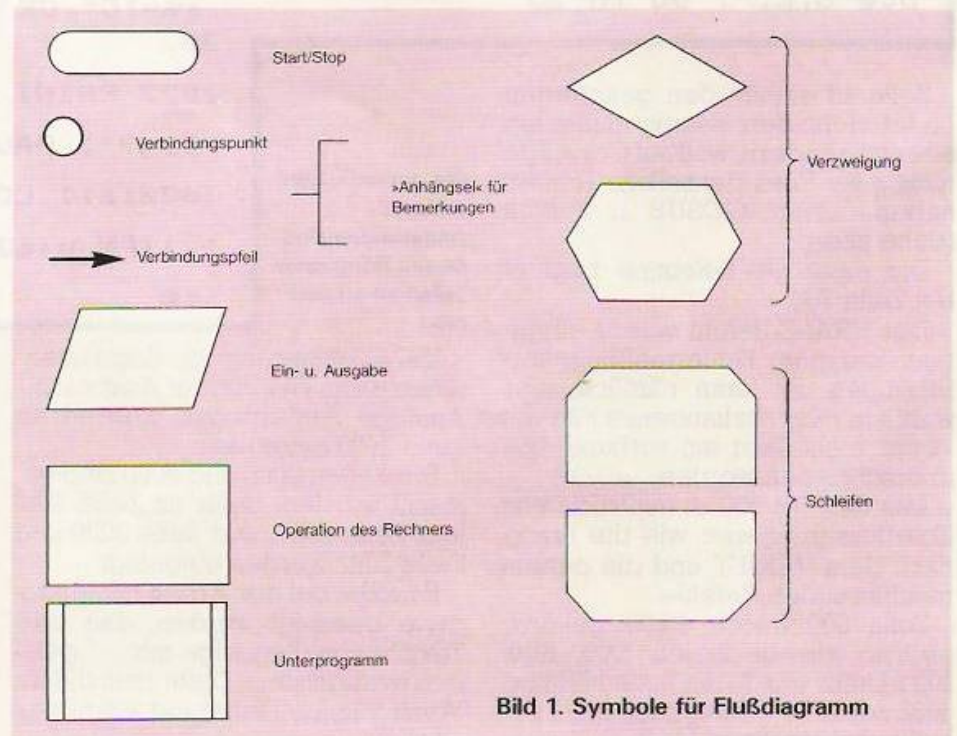


Bild 1. Symbole für Flußdiagramm



# man richtig

Dazu kommen noch solche Kleinigkeiten wie Aufgabenzähler erhöhen, Lösung speichern, Text zur Aufgabe ausgeben. Es ist zweckmäßig, möglichst viele Variablen mit demselben Namen ansprechen zu können. Die Einzelteile der Aufgaben legen wir daher in einem Feld ab: A\$(4,\*\*). Dann können sie mit einer Schleife ausgegeben werden. Dieselbe Speicherung bietet sich natürlich dann auch für die Lösung (L\$(2,\*\*)) und die Eingabe des Benutzers an (E\$(2,\*\*)).

Selbst mit dem Plan nach Bild 3 sind noch nicht alle Probleme klar. Aber wir sind trotzdem schon ein ganzes Stück weiter.

Auf die Entwicklung der »Kleinigkeiten« wie GGT, KGV verzichte ich hier. Die Verfahren sind so einfach, wenn man erstmal drauf gekommen ist, daß hierfür das Listing des Programms genügt.

Die Eingabe der Lösung ist vom Ablauf her recht einfach:

- >PRINT<-Position festlegen
- Tastatur abfragen
- Eingabe in Zeichenkette speichern
- und von vorne.

Es muß natürlich eine Abbruchbedingung festgelegt werden.

Bevor Sie jetzt weiterlesen, sollten Sie versuchen, die PAP's zu verstehen. Und vielleicht auch zu ändern beziehungsweise verbessern.

## Die Arbeit am Computer

Nachdem alles das feststeht und die Hauptprobleme gelöst sind, können wir jetzt unseren Computer einschalten. Hier möchte ich kurz auf die Bedeutung der REM-Zeilen kommen: Sie sollten außer dem Hinweis, was in diesem Programmteil geschieht, auch eine Liste der benutzten Variablen enthalten, getrennt nach solchen, die nur benutzt werden und solchen, die verändert werden. Eine Auflistung der Schlei-

fenvariablen kann auch nicht schaden. Diese REM-Statements programmieren wir zuerst. Bei den Unterprogrammen setzen wir auch gleich auf die höchste vorgesehene Zeilennummer das RETURN. Bei späteren Tests von Teilprogrammen erhalten wir dann keine lästigen Fehlermeldungen. Unser Programm sieht also jetzt so aus:

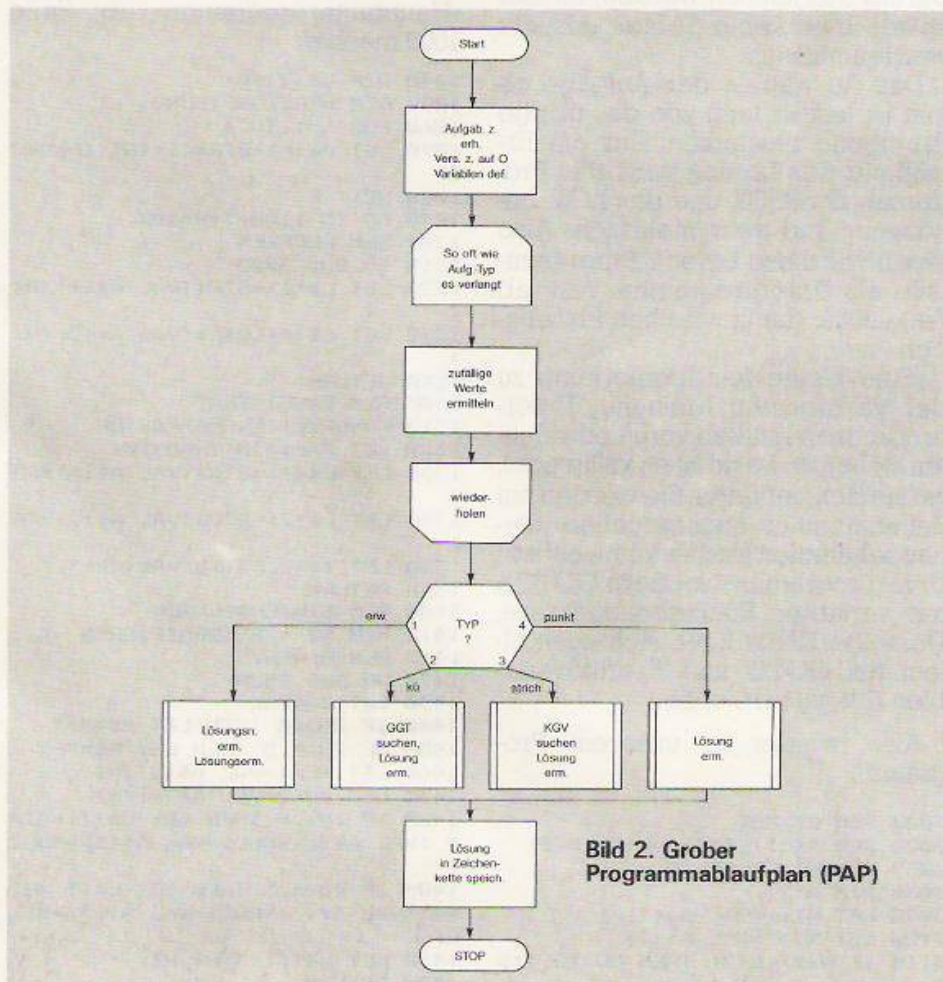


Bild 2. Grober Programmablaufplan (PAP)

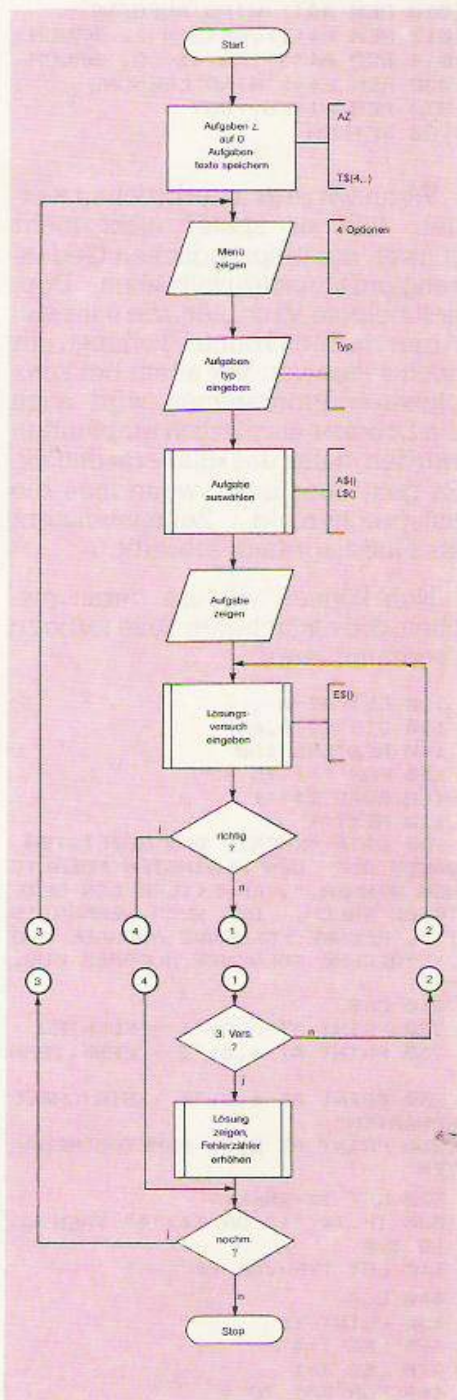


Bild 3. Feinplan Unterprogramm-Aufgabe auswählen

```

10 REM *** BRUCHRECHENTEST **
20 REM
100 REM VARIABLE INITIALISIEREN
110 REM AZ, T$( )
200 REM MENUE DARSTELLEN
300 REM AUFGABENTYP EINGEBEN
310 REM TYP
350 REM AUFGABE AUSWAHLEN
360 GO SUB 1000
370 LET AZ=AZ+1
400 REM AUFGABE ANZEIGEN
410 REM A$(4,*)=AUFGABE
420 REM T$( )=TEXT
430 REM R#=RECHENZEICHEN
600 REM LOESUNG EINGEBEN
610 REM E$( )=EINGABE
620 REM I,J=POSITION
800 REM VERGLEICH
810 REM E$( ), L$( )
900 REM INKEY#=SUBROUTINE
990 RETURN
1000 REM SUBROUTINE AUFGABE
1010 REM A$( ) WIRD AUFGABE
1012 REM A$(1),A$(2)= 1. BRUCH
1014 REM A$(3),A$(4)= 2. BRUCH
1020 REM L$( ) WIRD LOESUNG
1030 REM HI=KGV/GGT
1900 RETURN

```

Wenn wir jetzt unterbrochen werden, fällt es später nicht mehr schwer, die ursprünglichen Gedankengänge nachzuvollziehen. Doppelt belegte Variablen, die sehr viel Ärger machen können, kommen nur noch selten vor. Vor allem bei komplexeren Programmen wird auch ein Drucker als nützlich empfunden werden, damit das »Blättern« entfällt. Es geht aber auch, wenn man die entsprechenden Zeilennummern ins Flußdiagramm schreibt.

Nun können wir uns daran machen, die einfacheren Teile fertig zu programmieren:

```

120 LET AZ=0
130 DIM T$(4,64)
140 RESTORE 180
150 FOR I=1 TO 4
160 READ T$(I)
170 NEXT I
180 DATA "KUERZE DEN GEZEIGTEN
BRUCH AUF DEN KLEINSTEN MOEGLIC
HEN NENNER.", "ERWEITERE DEN GEZE
IGTEN BRUCH. DER NEUE NENNER IS
T .", "RECHNE FOLGENDE AUFGABE AUS
.", "RECHNE FOLGENDE AUFGABE AUS.
"
210 CLS
220 PRINT AT 2,0; "1 - KUERZEN
230 PRINT AT 4,0; "2 - ERWEITERN
"
240 PRINT AT 6,0; "3 - STRICHREC
HENARTEN"
250 PRINT AT 8,0; "PUNKTRECHENAR
TEN"
320 LET I#=INKEY#
330 IF I#<"1" OR I#>"4" THEN GO
TO 320
340 LET TYP=VAL I#
440 CLS
450 PRINT T#
460 LET I=4
470 LET J=1
480 FOR K=1 TO 4
490 PRINT AT I+(K>2),J+(K/2=INT
(K/2))*10;A$(I)
500 NEXT I

```

```

510 PRINT AT 4,0;R#
520 PRINT OVER 1;AT 4,1;" _____
"
530 PRINT OVER 1;AT 4,11;" _____
" AND R#<>" "
630 DIM E$(2,10)
640 FOR K=1 TO 2
650 LET I=7+K
660 LET J=0
670 LET HILF=1
680 LET J=J+1
690 PRINT AT I,J;"#"
700 GO SUB 900
710 IF I#<CHR# 13 THEN GO TO 75
0
720 LET E$(K,HILF)=I#
730 LET HILF=HILF+1
740 IF HILF<=10 THEN GO TO 680
750 NEXT K
820 FOR K=1 TO 2
830 IF VAL E$(K)=VAL L$(K) THEN
GO TO 870
840 NEXT K
850 PRINT AT 21,0;"DAS WAR RICH
TIG."
855 FOR L=1 TO 1000
856 NEXT L
860 GO TO 200
870 PRINT AT 21,0;"DAS WAR LEID
ER FALSCH."
875 FOR L=1 TO 1000
876 NEXT L
880 GO TO 200
910 LET I#=INKEY#
920 IF I#<"0" OR I#>"9" THEN IF
I#<>CHR# 13 THEN GO TO 910

```

Das waren die einfacheren Teile des Programms. Obwohl auch hier nicht alles auf Anhieb stimmte. Dennoch kann das Programm insoweit bereits ausgetestet werden, so daß später hier keine Fehler gesucht werden müssen.

Das Auswählen der Aufgabe allein ist fast so lang wie das übrige Programm zusammen. Für die Ermittlung der Lösung wird das Programm den GGT und das KGV benötigen. Und zwar mehrfach. Also legen wir diese beiden Programmteile als Unterprogramme fest, ab Zeile 3000, damit wir noch Platz haben.

(Eine kleine Randbemerkung zu der verbreiteten Meinung, Unterprogramme müßten vorne oder hinten stehen: in Basic ist es völlig egal, wo sie sich befinden. Sie werden mit Zeilennummer angesprochen. Genau genommen sind es keine echten Unterprogramme, sondern GOTO's mit variabler Rücksprungadresse. Derselbe Effekt ließe sich auch allein mit GOTO und Kontrollvariablen (Flags) erreichen.)

Also, wieder zu unserem Programm:

```

3000 REM UP GGT
3020 REM A$(1),A$(2)>N(1),N(2)
3030 REM HI=ERGEBNIS
3050 DIM N(2)
3090 LET N(1)=VAL A$(1)
3100 LET N(2)=VAL A$(2)
3110 IF N(1)<N(2) THEN GO TO 315
0

```

```

3120 LET GR=N(1)
3130 LET KL=N(2)
3140 GO TO 3170
3150 LET GR=N(2)
3160 LET KL=N(1)
3170 LET HI=KL
3180 IF GR/HI=INT (GR/HI) AND KL
/HI=INT (KL/HI) THEN GO TO 3210
3190 LET HI=HI-1
3200 GO TO 3180
3210 RETURN

```

In HI ist jetzt der größte gemeinsame Teiler enthalten. Nebenbei: wenn hier 1 herauskommt, sind die Zahlen teilerfremd. Das ist aber für die weitere Verarbeitung uninteressant.

```

3500 REM UP KGV
3510 REM A$(2),A$(4)>N(1),N(2)
3520 REM HI=ERGEBNIS
3530 DIM N(2)
3540 LET N(1)=VAL A$(2)
3550 LET N(2)=VAL A$(4)
3560 IF N(1)<N(2) THEN GO TO 360
0
3570 LET GR=N(1)
3580 LET KLN(2)
3590 GO TO 3630
3600 LET GR=N(2)
3610 LET KL=N(1)
3630 LET HI=GR
3640 IF HI/KL=INT (HI/KL) THEN G
O TO 3660
3650 LET HI=HI+GR
3660 RETURN

```

Damit hat das Programm die nötigen Unterprogramme, um die Lösungen ermitteln zu können.

Nun können wir uns an das »Hauptunterprogramm« ab Zeile 1000 machen.

```

1040 DIM L$(2,10)
1050 REM WERTZUWEISUNG
1060 FOR K=1 TO 4
1070 LET A$(K)=STR$(1+INT (RND*2
0)
1080 NEXT K
1090 GO TO 1100+TYP*100
1200 REM KUERZEN
1210 GO SUB 3000
1220 LET L$(1)=STR$(VAL A$(1)/HI
)
1230 LET L$(2)=LSTR$(VAL A$(2)/HI
)
1240 RETURN
1300 REM ERWEITERN
1310 FA=ERWEITERUNGSFAKTOR
1320 LET FA=1+INT (RND*20)
1330 LET L$(1)=STR$(VAL A$(1)*FA
)
1340 LET L$(2)=STR$(VAL A$(2)*FA
)
1350 LET T$(2,53 TO)=A$(2)+". "
1360 RETURN
1400 REM STRICHRECHNEN
1410 REM R# = RECHENZEICHEN
1415 REM HI=KGV
1420 GO SUB 3500
1430 LET ZU=RND
1440 IF ZU>.5 THEN LET R#="+ "
1450 IF ZU<.5 THEN LET R#="- "
1460 LET MUL1=VAL A$(2)/HI
1470 LET MUL2=VAL A$(4)/HI
1480 IF ZU>.5 THEN LET L$(1)=STR
$( VAL A$(1)*MUL1+VAL A$(3)*MUL2
)
1485 IF ZU<.5 THEN LET L$(1)=ST
R$( VAL A$(1)*MUL1-VAL A$(3)*MUL
2)
1490 LET L$(2)=STR# HI
1495 RETURN

```

```

1500 REM PUNKTRECHNEN
1510 REM R# = RECHENZEICHEN
1520 LET ZU=RND
1530 IF ZU>.5 THEN GO TO 1580
1540 LET R#="*"
1550 LET L$(1)=STR# (VAL A$(1)*V
AL A$(3))
1560 LET L$(2)=STR# (VAL A$(2)*A
$(4))
1570 RETURN
1580 LET R#=":"
1590 LET L$(1)=STR$(VAL A$(1)*VA
L A$(4))
1600 LET L$(2)=STR# (VAL A$(2)*V
AL A$(3))
1610 RETURN

```

Damit wäre das Größte geschafft. Das Programm ist sicher kein Superprogramm, aber das sollte auch gar nicht erreicht werden. Wichtig war mir, zu zeigen, wie man ein solches Problem angeht. Wenn wir jetzt noch Feinheiten einbauen möchten, ist das relativ einfach. Beispielsweise könnte man drei Versuche zulassen, bevor eine Aufgabe als falsch gewertet wird.

Wir haben damit nicht nur das Programm entwickelt, sondern auch schon die halbe Dokumentation fertig.

Übrigens ist das Programm nicht zum Abtippen gedacht, es ist nur ein Beispiel:

### Das endgültige Listing

```

10 REM *** BRUCHRECHENTEST **
20 REM
100 REM VARIABLE INITIALISIEREN
110 REM AZ, T$( )
120 LET AZ=0
130 DIM T$(4,64)
140 RESTORE 180
150 FOR I=1 TO 4
160 READ T$(I)
170 NEXT I
180 DATA "KUERZE DEN GEZEIGTEN
BRUCH AUF DEN KLEINSTEN MOEGLI
HEN NENNER.", "ERWEITERE DEN GEZE
IGTEN BRUCH. DER NEUE NENNER IS
T ", "RECHNE FOLGENDE AUFGABE AUS
.", "RECHNE FOLGENDE AUFGABE AUS.
"
200 REM MENUE DARSTELLEN
210 CLS
220 PRINT AT 2,0;"1 - KUERZEN
230 PRINT AT 4,0;"2 - ERWEITERN
"
240 PRINT AT 6,0;"3 - STRICHREC
HENARTEN"
250 PRINT AT 8,0;"PUNKTRECHENAR
TEN"
300 REM AUFGABENTYP EINGEBEN
310 REM TYP
320 LET I#=INKEY$
330 IF I#<"1" OR I#>"4" THEN GO
TO 320
340 LET TYP=VAL I#
350 REM AUFGABE AUSWAELHEN
360 GO SUB 1000
370 LET AZ=AZ+1
400 REM AUFGABE ANZEIGEN
410 REM A$(4,*)=AUFGABE
420 REM T$( )=TEXT
430 REM R#=RECHENZEICHEN
440 CLS
450 PRINT T#
460 LET I=4
470 LET J=1
480 FOR K=1 TO 4
490 PRINT AT I+(K/2),J+(K/2)=INT
(K/2)*10;A$(I)

```

```

500 NEXT I
510 PRINT AT 4,8;R#
520 PRINT OVER 1;AT 4,1;"-----
"
530 PRINT OVER 1;AT 4,11;"-----
" AND R#<>" "
600 REM LOESUNG EINGEBEN
610 REM E$( )=EINGABE
620 REM I,J=POSITION
630 DIM E$(2,10)
640 FOR K=1 TO 2
650 LET I=7+K
660 LET J=0
670 LET HILF=1
680 LET J=J+1
690 PRINT AT I,J;"#"
700 GO SUB 900
710 IF I#=CHR# 13 THEN GO TO 75
0
720 LET E$(K,HILF)=I#
730 LET HILF=HILF+1
740 IF HILF<=10 THEN GO TO 680
750 NEXT K
800 REM VERGLEICH
810 REM E$( ), L$( )
820 FOR K=1 TO 2
830 IF VAL E$(K)=VAL L$(K) THEN
GO TO 870
840 NEXT K
850 PRINT AT 21,0;"DAS WAR RICHTIG."
855 FOR L=1 TO 1000
856 NEXT L
860 GO TO 200
870 PRINT AT 21,0;"DAS WAR LEIDER FALSCH."
875 FOR L=1 TO 1000
876 NEXT L
880 GO TO 200
900 REM INKEY$-SUBROUTINE
910 LET I#=INKEY$
920 IF I#<"0" OR I#>"9" THEN IF
I#<>CHR# 13 THEN GO TO 910
990 RETURN
1000 REM SUBROUTINE AUFGABE
1010 REM A$( ) WIRD AUFGABE
1012 REM A$(1),A$(2)= 1. BRUCH
1014 REM A$(3),A$(4)= 2. BRUCH
1020 REM L$( ) WIRD LOESUNG
1030 REM HI=KGV/GGT
1040 DIM L$(2,10)
1050 REM WERTZUWEISUNG
1060 FOR K=1 TO 4
1070 LET A$(K)=STR$(1+INT (RND*2
0))
1080 NEXT K
1090 GO TO 1100+TYP*100
1200 REM KUERZEN
1210 GO SUB 3000
1220 LET L$(1)=STR$(VAL A$(1)/HI
)
1230 LET L$(2)=LSTR$(VAL A$(2)/HI
)
1240 RETURN
1300 REM ERWEITERN
1310 FA=ERWEITERUNGSFAKTOR
1320 LET FA=1+INT (RND*20)
1330 LET L$(1)=STR$(VAL A$(1)*FA
)
1340 LET L$(2)=STR$(VAL A$(2)*FA
)
1350 LET T$(2,53 TO)=A$(2)+". "
1360 RETURN
1400 REM STRICHRECHNEN
1410 REM R# = RECHENZEICHEN
1415 REM HI=KGV
1420 GO SUB 3500
1430 LET ZU=RND
1440 IF ZU>.5 THEN LET R#="+"
1450 IF ZU<=.5 THEN LET R#="-"
1460 LET MUL1=VAL A$(2)/HI
1470 LET MUL2=VAL A$(4)/HI
1480 IF ZU>.5 THEN LET L$(1)=STR
$( VAL A$(1)*MUL1+VAL A$(3)*MUL2
)
1485 IF ZU<=.5 THEN LET L$(1)=ST
R$( VAL A$(1)*MUL1-VAL A$(3)*MUL

```

```

2)
1490 LET L$(2)=STR# HI
1495 RETURN
1500 REM PUNKTRECHNEN
1510 REM R# = RECHENZEICHEN
1520 LET ZU=RND
1530 IF ZU>.5 THEN GO TO 1580
1540 LET R#="*"
1550 LET L$(1)=STR# (VAL A$(1)*V
AL A$(3))
1560 LET L$(2)=STR# (VAL A$(2)*A
$(4))
1570 RETURN
1580 LET R#=":"
1590 LET L$(1)=STR$(VAL A$(1)*VA
L A$(4))
1600 LET L$(2)=STR# (VAL A$(2)*V
AL A$(3))
1610 RETURN
3000 REM UP GGT
3020 REM A$(1),A$(2)>N(1),N(2)
3030 REM HI=ERGBNIS
3050 DIM N(2)
3090 LET N(1)=VAL A$(1)
3100 LET N(2)=VAL A$(2)
3110 IF N(1)<N(2) THEN GO TO 315
0
3120 LET GR=N(1)
3130 LET KL=N(2)
3140 GO TO 3170
3150 LET GR=N(2)
3160 LET KL=N(1)
3170 LET HI=KL
3180 IF GR/HI=INT (GR/HI) AND KL
/HI=INT (KL/HI) THEN GO TO 3210
3190 LET HI=HI-1
3200 GO TO 3180
3210 RETURN
3500 REM UP KGV
3510 REM A$(2),A$(4)>N(1),N(2)
3520 REM HI=ERGBNIS
3530 DIM N(2)
3540 LET N(1)=VAL A$(2)
3550 LET N(2)=VAL A$(4)
3560 IF N(1)<N(2) THEN GO TO 360
0
3570 LET GR=N(1)
3580 LET KLN(2)
3590 GO TO 3630
3600 LET GR=N(2)
3610 LET KL=N(1)
3630 LET HI=GR
3640 IF HI/KL=INT (HI/KL) THEN G
O TO 3660
3650 LET HI=HI+GR
3660 RETURN

```

Sollte der Speicherplatz nicht für die vielen REM-Zeilen genügen, so sollten diese aber im PAP notiert werden. Dadurch findet man nachher im Programm eher die Stelle, an der etwas zu ändern oder verbessern ist.

Beim Durcharbeiten dieses Artikels ist vielen sicher aufgefallen, daß die Vorarbeiten auf dem Papier mehr Zeit in Anspruch nehmen als das Programmieren selbst. Dafür entfällt aber hinterher ein Großteil der Fehlersuche. Ein Programm, das vollständig fehlerfrei auf Anhieb läuft, ist eine absolute Rarität, auch bei erfahreneren Computereckern. Normalerweise kann der Programmierer nicht ununterbrochen bis zum Ende an seinem Programm sitzen. Daher schleichen sich ohne diese Vorarbeiten sehr viele überflüssige Fehler ein.

(Erika Hölscher)

# BÜCHER FÜR SINCLAIR-FANS

Bücher für und über Sinclair-Computer sind reichlich

geschrieben worden. Nur wenige sind bemerkenswert.

Erstaunlich ist die Tatsache, daß fast alle Bücher, die mir gefallen, aus ein und demselben Verlag stammen. Wo hat der Birkhäuser-Verlag die »gute Nase« für gute Bücher her?

Aber jetzt der Reihe nach:

»Alles über Sinclair-Computer« ist der Titel eines neuen Buches, welches sowohl alles über Steve Sinclair, über die »Szene« um die Computer und fast alles über das verfügbare Zubehör aussagt. Dabei ist sowohl die Hardware als auch die Software gemeint.

Urteil: Unverzichtbar in der Sinclair-Buchecke.

ISBN 3-7643-1625-X, Preis 29,80 DM

Ebenfalls neu auf dem Büchermarkt ist »Einfache Zusatzgeräte für ZX Spectrum, für ZX81 und Jupiter Ace« von Owen Bishop. Dieses Bastelbuch beschreibt Schaltungen, die einfach und billig aufzubauen sind und den Nutzwert Ihres Computers beachtlich erweitern. Das angebotene Bastelprogramm umfaßt unter anderem einen Impuls-Detektor, einen Bild-Digitalisierer, Modellsteuerungen und Schaltungen für eine Wetterstation. Selbstverständlich fehlt die notwendige Software dazu nicht.

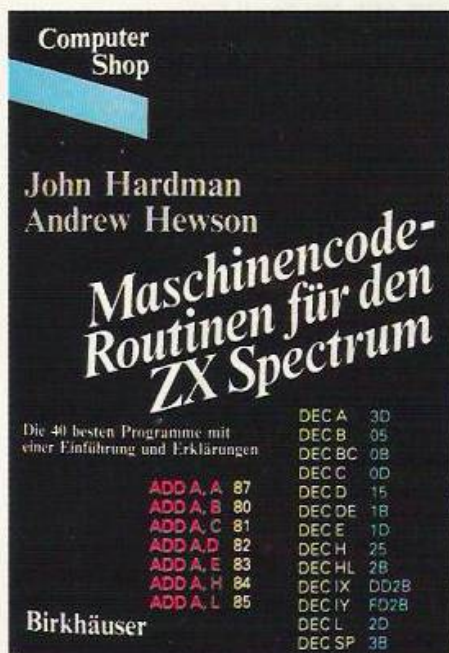
Urteil: Empfehlenswert für Bastler. ISBN 3-7643-1589-X, Preis 29,80 Mark.

26 Fertigprogramme und eine gute Einführung in die Programmierung finden Sie in dem Band »Sinclair ZX Spectrum, Programmieren leichtgemacht« von Ian Stewart und Robin Jones.

Urteil: Gute Lektüre für Spectrum-Anfänger.

ISBN 3-7643-1491-5, Preis 29,80 Mark.

Für Fortgeschrittene mit Maschinensprache-Ambitionen haben die beiden vorgenannten Autoren das Buch »ZX Spectrum Maschinencode« verfaßt. Natürlich sind neben dem Lehrstoff auch gute fertige Routinen abgedruckt, und ein komplet-



Mein »Lieblingsbuch«

tes Listing aller Z80-Opcodes vervollständigt das Werk.

Urteil: Unentbehrlich für Maschinencode-Einsteiger.

ISBN 3-7643-1535-0, Preis 27,80 Mark.

Freunde des ZX81 sollten sich das Buch »Maschinencode und besseres Basic« zulegen, welches auch für Spectrum-Benutzer einiges bietet. Urteil: Gutes Lehr- und Nachschlagewerk.

ISBN 3-7643-1492-3, Preis 32,— Mark.

Noch mehr aus dem Spectrum herauszuholen, wird Ihnen mit Hilfe des Bandes »Weitere Kniffe und Programme mit dem ZX Spectrum«, ebenfalls von Stewart/Jones gelingen.

Urteil: Sehr gut für alle Spectrum-Programmierer.

ISBN 3-7643-1532-6, Preis 32,— Mark.

Ein wahrer Leckerbissen für programmierfaule Spectrum-Enthusiasten ist »Maschinencode-Routinen für den ZX Spectrum« von John Hardmann und Andrew Hewson. 40

tolle fertige Maschinencode-Programme, die sich leicht an eigene Programmierwerke anpassen lassen, ergeben folgendes Urteil: Unverzichtbare Fundgrube.

ISBN 3-7643-1559-8, Preis 29,80 Mark.

Die Autoren Ian Stewart und Robin Jones bieten auch für den ZX81-Fan etwas: »Sinclair ZX81, Programme, Spiele, Grafik.« Diese leichtverständliche Einführung ist zwar speziell auf den ZX81 zugeschnitten, aber auch für Spectrum-Besitzer lesenswert. 50 fertige Programme runden das »Lehrbuch« ab. Urteil: Sehr gutes Einsteiger-Buch. ISBN 3-7643-1398-6, Preis 28,80 Mark.

Zwei neue Bücher des Autors Jochen Merz sollten ebenfalls nicht in Ihrer Buchsammlung fehlen: »Maschinencode-Handbuch für den ZX Spectrum« und »Microdrive-Handbuch für den ZX Spectrum«. Das Maschinencode-Handbuch ist zugleich Lehr- und Nachschlagewerk und beschäftigt sich unter anderem ausführlich mit dem ROM des Spectrum und seinen nutzbaren Funktionen. Das Microdrive-Buch ist für Interface 1-Benutzer ein sehr gutes deutschsprachiges und leicht verständlich geschriebenes Handbuch. Im zweiten Teil des Werkes wird quasi als Dreingabe ein Programm »erweitertes Basic« gelistet und beschrieben, welches das gute Spectrum-Basic durch seine neuen Befehle unschlagbar macht. Für Leute wie mich, die zu faul zum Abtippen sind, gibt es auch für 10 Mark mehr das Buch mit einer Programmkassette.

Das Maschinencode-Handbuch mit der ISBN 3-923985-02-9 kostet 25,— Mark, das Microdrive-Handbuch, ISBN 3-923985-04-5, D 29,— Mark. Beide Bücher sind bei Profisoft in Osnabrück erhältlich.

(mk)

Im Handbuch findet man ab Seite 159 Angaben zu IN und OUT. Trägt man die Werte in eine Tabelle ein, so wird sofort das System der Kodierung deutlich. Die Adressen A0 bis A4 sprechen den Baustein an (zum Beispiel A0 die Tastatur, A2 den Drucker) und mit A8 bis A15 wird für die Tastatur die Halbreihe selektiert. Für zusätzliche Hardware, wie zum Beispiel ein Joystick-Interface, können Bit A5 bis A7 verwendet werden. Beim Spectrum sind sowohl Adreß- wie auch Datenbus low-aktiv, das heißt ein gesetztes Bit hat niedrigen Pegel (0V), alle anderen sind »high«.

Einen Adreßwert errechnet man durch Subtraktion des Bitwertes vom 16-Bit-Höchstwert (65535). Für den 8-Bit-Datenbus gilt entsprechend der Höchstwert 255.

### Die IN-Funktion zur Abfrage

Wie die Tabelle zeigt, werden für die Tastatur nur Datenbit D0 bis D4 belegt. Das entspricht den fünf Tasten einer Halbreihe von außen nach innen. Das Ergebnis der IN-Funktion ist vom Bitwert der gedrückten Taste abhängig. Auch mehrere gleichzeitig gedrückte Tasten lassen sich auswerten, was ja zum Beispiel bei INKEYs nicht möglich ist.

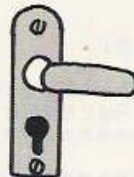
Bei allen Tastaturadressen wird auch die Ear-Buchse auf D6 abgefragt. Hat man ein entsprechendes Interface, so stehen mit den User-Adressen natürlich alle acht Datenbits zur Verfügung.

### Die OUT-Anweisung zur Ausgabe

Ohne Interface kann man eigentlich mit der OUT-Anweisung nicht viel anfangen. Zur Steuerung von Lautsprecher, Border und Drucker hat der Spectrum ja bessere Befehle. Aber versuchen kann man es ja mal.

### Zum Probieren

Die Kurzprogramme »Border« und »Lautsprecher« arbeiten mit der OUT-Anweisung. Ein Beispiel für die IN-Funktion ist »Ear-Buchse«. Mit Musik lassen sich Kurven auf den Schirm plotten. Natürlich ist das kein Pegelmesser, weil nur die Dauer von »laut« und »leise« für das Auf und Ab verantwortlich ist.



Mit den Basic-Befehlen IN und OUT kann man den Spectrum dazu bringen, sich mit der angeschlossenen Hardware zu beschäftigen. Und dazu gehört auch die Tastatur, die Mikrofon- und Kopfhörer-Buchse und der eingebaute Lautsprecher.

Das binäre Adreß- und Datenbitmuster zeigt Listing 1. Jeweils in einer Schleife werden die Dezimalwerte für Adresse und Daten in binäre Werte umgerechnet und wie in der Tabelle dargestellt. Die Tastatur- oder Joystickabfrage läßt sich so ganz anschaulich nachvollziehen. In einer Endlosschleife läuft die Datenabfrage, so daß die Adresse nur nach BREAK und RUN zu ändern ist. Natürlich kann man sie mit BIN auch binär eingeben und erspart sich damit eventuell das Umrechnen.

(Jürgen Howaldt)

## Liste der Variablen

a(16) = Adressfeld  
 d(8) = Datenfeld  
 adr = IN-Adresse  
 dw = Rest Datenwert  
 aw = Rest Adresswert  
 bw = Bitwert  
 i,n = Schleife

Adress- und Datenbitmuster  
 Adresse: 65278      Daten: 255  
 1111110 1111110      1111111  
 Neue Adresse nach BREAK + RUN

```

10 REM Listing 1
20 DIM a(16): DIM d(8)
30 PRINT AT 2,3:
"Adress- und Datenbitmuster"
40 GO SUB 0220: REM Adresse
50 REM =====
60 REM     Daten
70 REM =====
80 FOR i=0 TO 1 STEP 0
90 LET dw=IN adr
100 LET bw=128
110 PRINT AT 10,20;
120 PRINT "Daten: ";dw
130 PRINT AT 12,20;
140 FOR n=8 TO 1 STEP -1
150 LET d(n)=0
160 IF dw>=bw THEN LET d(n)=1
: LET dw=dw-bw
170 LET bw=bw/2
180 PRINT d(n);
190 NEXT n
200 NEXT i
210 REM =====
220 REM     Adresse
230 REM =====
240 LET bw=32768
250 INPUT "Adresse: ";adr
260 PRINT AT 10,0;"Adresse: ";
270 PRINT adr
280 LET aw=adr
290 FOR n=16 TO 1 STEP -1
300 LET a(n)=0
310 IF aw>=bw THEN LET a(n)=1
: LET aw=aw-bw
320 LET bw=bw/2
330 IF n=8 THEN PRINT " ";
340 PRINT a(n);
350 NEXT n
360 PRINT AT 20,1;
"Neue Adresse nach BREAK + RUN"
370 RETURN
    
```

Listing Adreß- und Datenbitmuster

```

10 REM EAR-Buchse
20 LET y=0
30 CLS
40 FOR x=0 TO 255 STEP .1
50 LET in=IN 254
60 IF in=255 AND y>0
THEN LET y=y-0.5
70 IF in=191 AND y<173
THEN LET y=y+1.5
80 PLOT X,Y
90 NEXT X: GO TO 30
    
```

Kurzlisting »EAR-Buchse«

```

10 REM BORDER-Farbe
20 FOR n=255 TO 248 STEP-1
30 OUT 254,n
40 PRINT AT 10,5;n
50 PAUSE 50
60 NEXT n: GO TO 20
    
```

Kurzlisting »BORDER-Farbe«

```

10 REM Lautsprecher
20 OUT 254,255
30 OUT 254,239
40 GO TO 20
    
```

Kurzlisting »Lautsprecher«

ADRESSEN																DATEN												
Max.Wert: 65535																Max.Wert: 255												
Bit-Wert	32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1	128	64	32	16	8	4	2	1				
A	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	D	7	6	5	4	3	2	1	0			
Tastatur																	Ea	V	C	X	Z	Cs						
IN 65278																	Ea	G	F	D	S	A						
IN 65022																	Ea	T	R	E	W	O						
IN 64510																	Ea	S	4	3	2	1						
IN 63486																	Ea	6	7	8	9	0						
IN 61438																	Ea	Y	U	I	O	P						
IN 57342																	Ea	H	J	K	L	En						
IN 49150																	Ea	B	N	M	Ss	Sp						
IN 32766																												
Lautsprecher, Mic, Border																												
OUT 254																												
Drucker																												
IN 251																												
OUT 251																												
Interface & Microdrive																												
IN 254																												
IN 247																												
IN 239																												
USER-Adressen																												
31																												
63																												
96																												
127																												
159																												
191																												
223																												

IN/OUT-Adressen &amp; Daten des Spectrum

# Mehr als Textverarbeitung:

nes 64 bietet — in Kurzform — folgende Möglichkeiten: bis zu 251 Zeichen per Zeile; 22.000 Zeichen (entspricht 11 Seiten oder 350 Zeilen) können auf einmal gespeichert werden; glattes Scrolling; zehn Datenbasis-Operationen; Mailing List-Verarbeitung; alle Druckereigenschaften, auch für ZX-Drucker; Setup-Möglichkeiten; Microdrive-kompatibilität. Es wurde von den Jugoslawen T. Jakopin geschrieben und besteht fast nur aus Maschinencode-Routinen. Geliefert wird es von Ultrasoft mit einer ausführlichen deutschen Bedienungsanleitung (40 Seiten DIN A5), die sehr verständlich geschrieben ist. Das Programm wird geladen mit LOAD " " und meldet sich mit »Create or edit« sowie einer Firmennummer, die nach Aussage des Programmschreibers nicht gelöscht werden kann, ohne das Programm zu zerstören. Hier wurde ein neuer Schutz eingeführt, um das Kopieren zu erschweren. Der Vertreiber führt eine Liste über alle ausgelieferten Programme und kann somit leicht den Namen des Käufers herausfinden.

Wird nach dem Laden die Frage »Create or edit« mit »e« beantwortet,

# INES

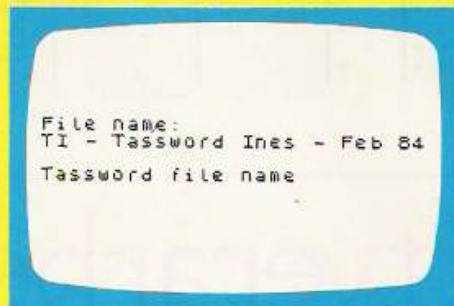
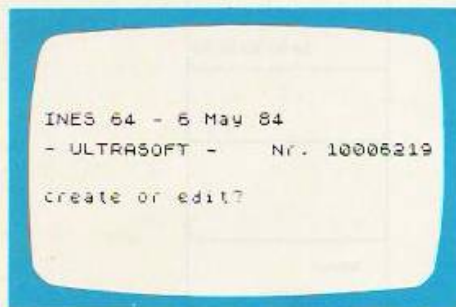
**Das Textverarbeitungsprogramm INES erlaubt vom Briefschreiben, Ändern und Finden von Datensätzen bis zum automatischen Sortieren fast alles, was man sich wünschen könnte.**

verlangt der Computer den File-Namen des zu ladenden Programms. Man kann hiermit bereits bestehende Ines-Programme laden.

Beantwortet man die Frage mit »c«, was soviel wie kreieren bedeutet, wird der Bildschirm schwarz und an der linken oberen Ecke erscheint ein weißer Cursor. Jetzt darauf loschreiben, dachte ich mir, und siehe da, es klappte wirklich. Der einprogrammierte Zeichensatz mit deutschen Umlauten erlaubt ein Schreiben von 64 Zeichen je Zeile. Etwas gewöhnungsbedürftig sind die verschiedenen Zeichen schon: So hatte ich Probleme, zwischen N und M zu unterscheiden. Angenehm fällt das glatte Scrolling auf. Die oberste Zeile rutscht automatisch nach oben, wobei unten eine Zeile zur weiteren Arbeit frei wird. Möchte man den geschriebenen Text verbessern oder ändern, muß man in den Editor-Modus übergehen, den man durch gleichzeitiges Drücken der Caps-Shift- und Edit-Taste erreicht.

Jetzt fängt die Sache an, Spaß zu machen. Durch Betätigen der Tasten b, l, n und e kann im Text beliebig geblättert werden. Der Cursor

ist frei über den gesamten Bildschirm zu bewegen. Der Anwender hat auch noch die Möglichkeit, durch Drücken der  $\uparrow$ -Taste verschiedene Parameter zu ändern: Bildschirm weiß/schwarze Schrift; linke und rechte Druckrandeinstellung; Zeilenabstand; Umschalten auf den Epson FX-Drucker und so weiter. Änderungen der Parameter erfolgen mit den Cursortasten 5 beziehungsweise 6.



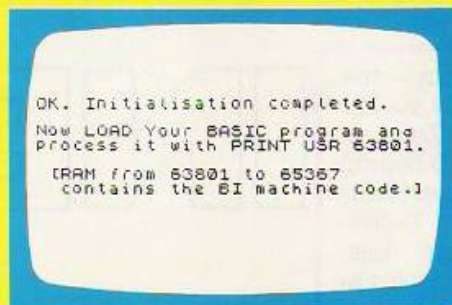
Ebenso einfach ist es, einzelne Zeichen, ganze Zeilen oder aber Textabschnitte innerhalb eines Datensatzes zu löschen beziehungsweise zu ergänzen. Wer viel Texte schreibt und diese später ändern will, findet bei Ines einige Hilfsroutinen. Eine davon erlaubt zum Beispiel das Umformatieren eines Textes, der auf 80 Zeichen pro Zeile geschrieben wurde. Soll der Text auf Manuskriptgröße gebracht werden (35 Zeichen/Zeile und 1 Zeile Korrigierabstand), ist dies leicht möglich. Ines verlangt zuerst, daß die Markierungspunkte gesetzt werden, da nur innerhalb dieser Markierungen gearbeitet wird.

Die Bedienungsanleitung hilft erfolgreich. Nach zirka einer Stunde kannte ich alle wichtigen Befehle und konnte die Anleitung beiseite legen. Auf eine Besonderheit will ich noch etwas näher eingehen, nämlich auf die Sort-Sequenzen.

Durch einen Escape-Sequenz-Begrenzer können Tabellen erstellt werden, für die man sonst sehr viel Zeit verwenden mußte. Das Programm hat eine ausgeklügelte Maschinencodieroutine, die das Sortieren von Datensätzen zur wahren Freude macht. Am besten ist es, die Beispiele in der Anleitung auszupro-

bieren, um die Leistungsfähigkeit dieser Routine zu erfahren.

Für die Drucker RX und FX sind die Steuerroutinen fest eingearbeitet. Es können alle Möglichkeiten genutzt werden, so zum Beispiel Breitschrift, komprimierte Schrift, Fettdruck, verschiedene Schriftarten, und dies alles innerhalb eines Datensatzes. Für den Sinclair ZX-Drucker ist ein eigener Befehl vorhanden, der den LPRINT-Befehl simuliert. Das Verwenden eines anderen Druckers ist ebenfalls möglich. In der Bedienungsanleitung sind die Adressen und Codes zur Steuerung des Druckers angegeben. Dem etwas fortgeschrittenem Spectrum-Besitzer und Maschinencodiker dürfte es keine Mühe bereiten, die Druckersteuerung auszuprogrammieren.



Neu geschriebene Texte lassen sich leicht abspeichern, ebenso ist es möglich, verschiedene Textabschnitte zusammen zu laden, wobei auch hier verschiedene Varianten anwählbar sind. Soweit zum Programm Ines 64.

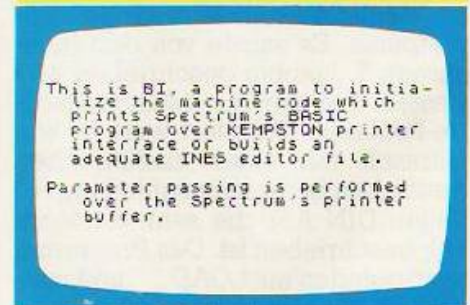
Auf der Kassette sind noch drei Zusatzprogramme. Das erste Programm wird geladen mit LOAD "BI". BI — Basic zu Ines — erlaubt, vorhandene Basic-Programme in eine Ines-Text-beziehungsweise Datendatei umzuschreiben. Dazu muß das Programm zuerst initialisiert werden. Da das Programm in ein Maschinencodieroutine umgesetzt wird, soll zuerst die Startadresse eingegeben werden, die auf jeden Fall unter 63500 liegen muß, da sonst das Dienstprogramm zerstört wird. Weiter wird die größte Zeilenlänge, die erste und die letzte Zei-

lennummer verlangt. Das Programm darf nicht selbststartend sein. Hat der Anwender alle Angaben richtig beantwortet, meldet sich der Computer mit »O.K. Initialisation completed. Now LOAD Your Basic program and process it with PRINT USR 63801«. Das Basic-Programm wird nun geladen und innerhalb von Sekunden steht eine fertige Basic-Datei zur Verfügung. Auch hier gilt: selbst ausprobieren und staunen.

Das zweite Hilfsprogramm nutzt denen, die bisher mit Tasword II gearbeitet haben. Bereits bestehende Tasword-Dateien werden wie beim »BI« in eine Ines-Datei verwandelt.

Selbst die längste Tasword-Datei bereitet dem Ines keine Schwierigkeiten, da Ines eine halbe Seite mehr Platz zur Verfügung hat. Geladen wird es mit LOAD "TI". Das Initialisieren ist nicht schwer und wird im Handbuch genau beschrieben.

Das dritte Hilfsprogramm nennt sich Gens to Ines. Wer viel in Maschinencodes arbeitet und ein Freund des Programms Gens von Hisoft ist, hat die Möglichkeit, Gens-Programme in eine Ines-Datei umzuschreiben. Die 64 Zeichen/Zeile kommen gut zur Geltung und bieten



eine bessere Übersicht. Auch hier übernimmt der Computer die Hauptarbeit.

Das Programm lohnt sich für den ernstesten Spectrum-Anwender auf jeden Fall. Es kostet 49,90 Mark. Hervorzuheben ist das schnelle Arbeiten bei allen Befehlen und das umfangreiche Handbuch. Ein Monitor ist sehr zu empfehlen, da 64 Zeichen eine gute Auflösung des Bildschirms verlangen. (Hans Merkl)



# Textverarbeitung mit Tasword II

**Mit Tasword II wird für den Spectrum ein wirklich gutes Textverarbeitungsprogramm geboten. Ich habe meine Artikel für dieses Heft damit erstellt.**

Das Programm selbst wurde zwar nicht auf Deutsch umgeschrieben, aber das fällt nicht weiter ins Gewicht. Es wird eine ausführliche deutsche Beschreibung mitgeliefert und eine Tastaturschablone, die allerdings etwas dünn ausgefallen ist und auch nicht sofort aufgelegt werden kann. Man muß erst noch die Löcher für die Tasten hineinschneiden ... Außerdem befindet sich auf der Kassette ein deutschsprachiges Textfile, das weitere Einsteigerhilfen bietet. Da dieser »Tutor« etliche Fehler enthält, hat der Neuling ein sehr schönes Übungsfeld. Vermutlich sind extra deshalb so viele Tippfehler drin ...

Einige Fehler haben sich sowohl in der Beschreibung als auch im Tutor und in der zweiten Help-Seite eingeschlichen: »Unjustify Line« liegt nicht auf Taste K, sondern auf Taste H. Auf der Tastaturschablone ist es richtig. Und an anderer Stelle muß es heißen: »Symbol Shift + U« und nicht »Symbol Shift + G« (Cursor zum Ende des Textes).

Das ausführliche Handbuch (keine Selbstverständlichkeit!) wird sehr schnell überflüssig. Die Bedienung des Programms ist nämlich sehr einfach. Es verfügt über alle Funktionen, die man bei einem solchen Programm sucht:

**\* Cursorbewegungen:**

- alle vier Richtungen
- Ende oder Anfang des Files
- Ende des letzten Wortes/Anfang des folgenden Wortes
- 22 Zeilen vor oder zurück (das entspricht einer Bildschirmseite)

**\* Einfügungen:**

- Buchstaben (erst ein Leerzeichen, das dann überschrieben wird)
- Zeilen
- Abschnitte (es werden jeweils bei Erreichen eines Zeilenendes Leerzeilen eingefügt, die dann überschrieben werden)

- Abschnitte kopieren oder verschieben
  - Zeilen löschen
  - Wörter suchen und ersetzen
  - \* Formatieren:
    - automatischer Randausgleich (wahlweise)
    - Hinüberziehen des letzten Wortes in die nächste Zeile (wahlweise)
    - Zeile wieder »normalisieren« (das heißt, die für den rechten Randausgleich nötigen Leerzeichen wieder löschen)
    - Zeile zentrieren, nach links oder rechts schieben
    - Absätze wieder neu einrichten (beispielsweise nach Einfügungen)
    - rechts und links beliebige Ränder setzen (der Randbereich wird gelb unterlegt)
  - \* Input/Output:
    - >SAVEN< des Textfiles
    - >SAVEN< einer modifizierten Tasword-Version (nach Anpassung an Microdrive oder Drucker)
    - >LOADEN< eines Textfiles
    - >MERGEN< eines Textfiles
    - Drucken über ZX-Printer
    - Drucken über Epson-Drucker
    - Definieren der Steuerzeichen für den Drucker
- (Nach der Rückkehr aus dem I/O-Menü müssen die Ränder neu gesetzt werden.)

Im Programm sind zwei >HELP<-Seiten enthalten. Hier wurden jedoch die auf der Tastatur-Schablone verwandten Kurzzeichen eingesetzt, so daß nur beide zusammen wirklich nützlich sind.

Übrigens stehen die Umlaute nicht auf dieser Schablone. Auf dem Bildschirm sind die großen und kleinen Umlaute sehr schlecht zu unterscheiden.

Das Programm kann auf den ZX-Printer drucken. Hierbei ist jedoch zu beachten, daß die kleine Schrift nicht besonders gut lesbar ist. Wenn auf Vergrößerung geschaltet ist, wird auch in der großen Normal-

schrift gedruckt, also nur 32 Zeichen je Zeile. Es können auch zwei »Steuerzeichen« in den Text eingefügt werden: >LARGE PRINTING ON< und >LARGE PRINTING OFF<. Der ZX-Drucker schreibt dann doppelt hoch.

Es wurde auch berücksichtigt, daß der Spectrum-Benutzer, der ein solches Programm benötigt, wahrscheinlich mit einem Normalpapier-Drucker arbeiten möchte. Die von mir getestete Version enthält die notwendige Software, um einen Epson-Drucker über eine Centronics-Schnittstelle anzusteuern. Die Steuerzeichen sind auf die Grafikzeichen gelegt. Dieser Drucker verwendet dann seinen eigenen Schriftsatz. Damit die Umlaute auch gedruckt werden, muß er schon auf den deutschen Zeichensatz eingestellt sein.

Das Programm läuft auch dann fehlerfrei, wenn Interface I angeschlossen ist. Es läßt sich ganz einfach so umschreiben, daß es auf Cartridges gespeichert werden kann und auch die Texte dort ablegt.

Obwohl das Programm teilweise in Basic arbeitet, ist es keineswegs zu langsam. So schnell, wie das Programm aufnehmen kann, tippt kaum jemand. Sollte der Benutzer — wie auch immer — ins 'Basic' gekommen sein, kann er das Programm getrost mit >RUN< neu starten. Der Text ist als Code (ab Adresse 32000) über >RAMTOP< abgelegt. Diese Texte kann man übrigens auch in eigene Programme einbauen (Anleitungen zum Beispiel) und mit folgendem Mini-Programm ausdrucken:

```
10 FOR I = 32000 TO 32000 + LAENGE-1
20 PRINT CHR$( PEEK I);
30 NEXT I
```

Der Ausdruck erfolgt allerdings im normalen Spectrum-Schriftsatz, das heißt, mit 32 Zeichen je Zeile und ohne Umlaute. Aber es geht.

Es ist für »Spectrum-Verhältnisse« ein gutes Programm, das sich durchaus mit »professionellen« Programmen der gleichen Art messen kann. Beim »Wettschreiben« mit einer Freundin, die auf einem Triumph Adler PC mit einem entsprechenden Programm arbeitete, gewann ich. Wir schreiben beide im Zehnfingersystem, und ich benutzte die Original-Spectrum-Tastatur. Wer zudem noch über einen Normalpapier-Drucker verfügt, kann seine Schreibmaschine verschrotten. Das Programm kostet bei Profisoft knapp 100 Mark.

(Erika Holscher)

# Mono, eine vielseitige

# Maschinencode-Hilfe

**Wer die Möglichkeit seines Spectrum voll ausnutzen will, wird über kurz oder lang eigene Programme in Maschinensprache schreiben. Besonders bei Spielen und bei schneller Grafik besticht die Geschwindigkeit von Maschinencode-Programmen.**

**D**och die Eingabe eines solchen Programms und insbesondere das Austesten erfordern sehr viel Mühe, wenn man allein mit dem Basic-Betriebssystem arbeiten muß. Ein Assembler, ein Disassembler und ein Debugger stellen hier eine unschätzbare Hilfe dar. Seit geraumer Zeit gibt es solche Programme auch für den Spectrum. Wir haben »Mono«, das alle drei genannten wichtigen Teilfunktionen enthält, für Sie getestet.

Mono läuft auf einem 48-K-Spectrum ohne besondere Erweiterungen und belegt dort den Speicherbereich ab der Adresse 40960. Im unteren Adreßbereich kann weiterhin mit Basic gearbeitet werden, ohne daß sich die beiden Systeme stören. Auch ein Hin- und Herschalten zwischen Basic und »Mono« ist jederzeit möglich, so daß auch kombinierte Basic/Maschinencode-Programme schnell und ohne Mühe erstellt werden können.

Nach dem Laden meldet sich das Programm mit einer Kopfzeile und wartet mit blinkendem Cursor auf Befehle. Schon nach den ersten Eingaben fällt dem Benutzer einer der größten Vorteile des Programms auf: Es besitzt nicht wie das Sinclair-Basic einen umständlichen Zeileneditor, sondern einen kompletten Bildschirmeditor à la Commodore. Der ganze Bildschirm wird also zu einem Arbeitsfeld, in dem Änderungen in Hex-Dumps oder Listings einfach dadurch vorgenommen werden, daß man mit dem Cursor an die entsprechende Stelle wandert und dort den gewünschten Text einschreibt. Selbstverständlich stehen auch Funktionen zum Einsetzen und Löschen von Zeichen zur Verfügung. Einfacher und schneller geht es wirklich nicht mehr.

Mit diesem Editor ist schnell das

erste Assembler-Programm eingegeben. Es können dabei alle Z80-Mnemonics benutzt werden, wobei allerdings die Syntax geringfügig von der Zilog-Norm abweicht. Dies kommt aber dem geübten Basic-Programmierer auch zugute, da hier im Gegensatz zu Standard-Assemblern auch Sprünge zu Zeilennummern eines Programms möglich sind.

Ein fertiges Programm kann nun mit dem »ASS«-Befehl als richtiger Maschinencode in den Speicher eingeschrieben werden. Die Übersetzungsgeschwindigkeit ist dabei durchaus akzeptabel und die eventuell auftretenden Fehlermeldungen sind recht präzise, so daß eine fehlerhafte Zeile ohne langes Rätselraten berichtigt werden kann. Diese Features heben aber »Mono« noch nicht von anderen Assemblern ab. Das eigentlich Besondere an diesem Programm ist die Möglichkeit, sogenannte Makros zu definieren. Als Makros bezeichnet man mehrere Z80-Befehle, die unter einem neuen Namen zusammengefaßt werden. Wenn man etwa den im Z80-Befehlssatz nicht vorkommenden Befehl »LD HL,BC« benutzen möchte, so definiert man sich einfach unter diesem Namen ein Makro aus den beiden Instruktionen »LD H,B« und »LD L,C«. In Zukunft kann man das neue Makro wie einen ganz normalen Z80-Befehl benutzen. Es kann sogar zur Definition von weiteren Makros mit eingesetzt werden. Da auch Parameter vom Namen eines Makros in den eigentlichen Code, für den das Makro steht, übernommen werden können, bieten sich dem Programmierer unglaublich vielfältige Möglichkeiten. Auf diese Weise können für spezielle Anwendungen ganz neue Befehlssätze erstellt und auch auf Band

abgespeichert werden. So werden Assembler-Programme kürzer und lesbarer, ohne dabei ihren Geschwindigkeitsvorteil gegenüber höheren Programmiersprachen zu verlieren.

Positiv fiel uns auf, daß auch der eingebaute Disassembler jedes neue Makro sofort in seinen Sprachschatz aufnahm und so die Programme im Speicher genauso aussahen wie sie im Programmtext geschrieben wurden. Ein weiterer Vorteil eines zusammenhängenden Systems aus Assembler und Disassembler ist die Möglichkeit, direkt aus dem Speicher in ein für den Assembler verständliches Listing zu disassemblieren. Mit dieser Funktion können auch bereits fertige Programmteile noch einmal bequem bearbeitet werden, zumal die absoluten und relativen Sprungadressen soweit möglich in Zeilenadressen umgewandelt werden.

Wie im Basic, so stehen auch bei »Mono« die üblichen Befehle zur Kassettenaufzeichnung zur Verfügung (LOAD, SAVE, VERIFY). Da die Speicherung im Sinclair-Standardformat erfolgt, dürfte es hier keine Probleme geben, ein mit »Mono« abgespeichertes Programm mit dem normalen Basic-Befehl LOAD " " CODE zu laden. Wie bei einem solchen Programm selbstverständlich, können alle wichtigen Ausgaben, wie Listings und Hex-Dumps, auch ausgedruckt werden.

Zusammenfassend kann man sagen, daß »Mono« ein gelungenes System zur schnellen und komfortablen Maschinencodeprogrammierung darstellt, das besonders durch die einfache Handhabung auch für Neulinge, schnell eine fast unverzichtbare Hilfe wird. Das Programm kostet bei der Firma Strecker 59 Mark. (mk)

# Eine Adreßverwaltung für Tasword

Mit Textverarbeitungsprogrammen erstellte Serienbriefe konnten bisher zwar vervielfältigt werden, hatten jedoch den Nachteil, daß die individuelle Adreßverwaltungsträger eingeben werden mußten. Die Ankündigung und Anrede manuell nachträglicherig: Zum Test bestellte ich mir eine Kassettenversion des Programms von Strecker.

Das Programm ist ausschließlich in Maschinensprache geschrieben, es nutzt 64 Zeichen pro Zeile, bietet deutsche Umlaute sowie diverse Optionen. Jegliche Eingaben, Korrekturen, Kommentare und Fehlermeldungen erfolgen direkt in die Bildschirmmaske. Ebenfalls werden die angewählten Menüs mit eingeblendet.

Gestartet wird mit dem Hauptmenü. Bei der Eingabe der ersten Adresse fiel auf, daß nicht mit einem vollen Bildschirmeditor gearbeitet wird, vielmehr erfolgen die Eingaben feldweise. Die Korrekturmöglichkeit besteht darin, daß das Programm für jede erfolgte Eingabe eine Bestätigung verlangt. Bei Nichtbestätigung löscht Multifile den letzten Eintrag, so daß dieser Vorgang wiederholt werden kann. Dieses Verfahren hört sich zwar umständlich an, hat aber den großen Vorteil, daß Bedienfehler so gut wie ausgeschlossen sind. Als Eingabefelder stehen zur Verfügung:

Name, Straße, Wohnort, Telefon und zwei frei zu belegende Kennzeichenfelder.

Der jeweilige Status des Programms, die Anzahl der belegten und freien Adressen, die Dateinummer und die Adressenpointer werden ständig angezeigt. Neben der Eingabe ist das Verändern (zum Beispiel neue Telefonnummer) und das komplette Löschen von Adressen möglich. Hat das Programm einen gesuchten Eintrag gefunden, so zeigt es diesen an und fordert wiederum eine Bestätigung. Das Speichern und Laden der Bestände zeigt einen weiteren Vorteil von Multifile. Sämtliche Angaben über den Belegungsgrad und die Dateinummer werden mit überspielt, ohne direkt

[ Home		MULTIFILE	[ 1984
Name :	.....	Selektiert:	
Straße :	.....	FD (j/n)	
Wohnort :	.....	Dateinummer: 000	
Telefon :	.....		
Kenn. 1 :	.		
Kenn. 2 :	.		
Floppy			
1 = Datei Laden		2 = Datei speichern	
3 = Brief Laden		4 = Brief speichern	
5 = Katalog		6 = B. Löschen	
7 = Hauptmenue			
File <000>	Belegt <000>	Frei <290>	

Die Bildschirmmaske von Multifile

das gesamte Programm abzuspeichern.

Ein eigenes (eingeblendetes) Menü befaßt sich ausschließlich mit dem Sortieren. Es kann nach Namen, Postleitzahlen und den beiden Kennzeichen sortiert werden. Das ganze File kann dann in Listenform ausgedruckt werden. Der Sortiervorgang dauert meiner Meinung nach etwas lange.

Der eigentliche Vorteil des Programms dürfte allerdings in den nachfolgenden Optionen liegen. Ein mit dem Textverarbeitungsprogramm Tasword II erstellter Brief kann in Multifile eingeladen werden. Multifile ist in der Lage, auf die vorhandenen Adressen zurückzugreifen und den Brief mit Anschrift des Empfängers sowie dessen persönlicher Anrede zu versehen. Der Autor scheint jedoch nichts von der Emanzipationsbewegung zu halten, denn die Anrede bezieht sich nur auf Herren. Es kann an einzelne

Adressen, bestimmte Postleitzahlbezirke und an, nach den frei definierten Kennzeichen, ausgewählte Adressen geschrieben werden. Den in Multifile enthaltenen Brief kann man sich auf dem Bildschirm ansehen. Dieses ist die einzige Option, bei der die Bildschirmmaske verlassen wird. Brief und Datei können separat geladen, gespeichert und auch gelöscht werden.

Alles in allem ist das Programm Multifile eine wertvolle Hilfe bei der Verwaltung von Adressen und bei der Erstellung von Serienbriefen. Die Idee des Autors, mit einer solchen Bildschirmmaske zu arbeiten, dürfte den Abstand zwischen dem Spectrum und manchem größeren Computer verkleinert haben; bei Verwendung mit Diskettenlaufwerken wird er sich sicherlich nochmals verringern. Das Programm kostet 49 Mark und läuft auf Kassette, Microdrive und Diskette.

(mk)

# Meine liebsten

## SPIELEREREIEN

Die gute Grafikauflösung macht's möglich:

Der ZX-Spectrum ist ein idealer »Spiel-

gefährte«, für den es eine riesige Auswahl

an Computerspielen gibt. Mit welchen

Programmen er am schönsten beept und

flackert, verraten wir Ihnen in diesem Artikel.

Die Spieleflut für den ZX-Spectrum kommt in erster Linie aus den Gefilden von Queen, Ale und Bowlerhüten – England. Zu überschauen ist das Angebot nicht mehr, doch hin und wieder gibt es außergewöhnliche Programme, die zu den Klassikern der Software-Szene avancieren.

Beginnen wir gleich mit dem »geistreichen« Spielvergnügen der Adventures. Ein echter Reißer dieser Kategorie war und ist »The Hobbit«, ein fesselndes Abenteuerspiel, das sich thematisch eng an Tolkiens »Herr der Ringe« anlehnt. Neben der schönen Grafik besticht das Programm durch die für Adventures bahnbrechende »Animation«. Sämtliche Figuren, die in »The Hobbit« vorkommen, führen ein gewisses Eigenleben. Um das sehr anspruchsvolle Programm zu lösen, ist man auf Mitarbeit und Korrespondenz angewiesen. Ebenfalls sehr eigenwillig und ungewöhnlich ist »Valhalla«. Wie der Titel bereits vermuten läßt, spielt dieses Adventure in göttlichen (inklusive Thor und Odin) Gefilden. Das Prinzip der »Animation« wurde bei »Valhalla« drastisch weiterentwickelt. Sämtliche Personen agieren höchst eigenständig

und mitunter auch ausgesprochen kämpferisch. Ein recht reizvolles Programm, das sich bereits kräftig von der traditionellen Form des Adventures fortentwickelt hat.

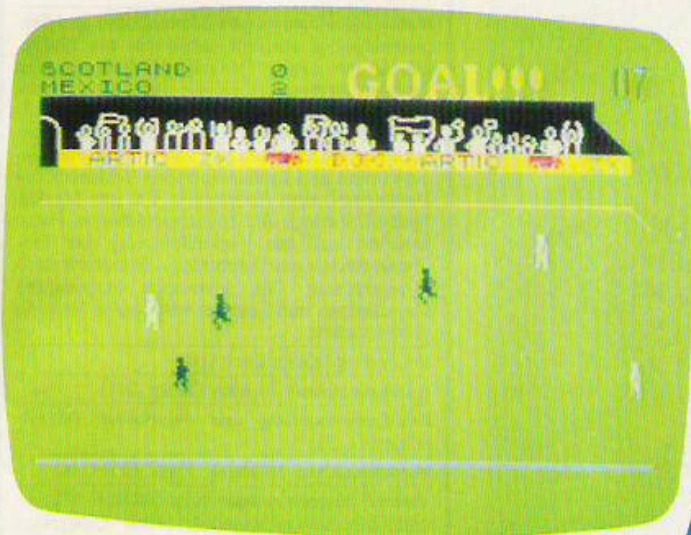
### Todesmutiger Tiefflieger

Weniger mit Worten als vielmehr mit der Tastatur (oder dem Joystick-Interface) wird bei den Geschicklichkeits-Spielen agiert. Gerade in dieser Kategorie hat sich eine ganze Latte von Kassetten für den Spectrum angesammelt, die in Sachen Spielwitz und Abwechslungsreichtum nichts zu wünschen übrig lassen. Zu meinen persönlichen Favoriten gehört »Fred«, ein putziges Männlein, das sich durch eine Pyramide schlägt, um diverse Schätze einzusammeln. Neben der witzigen Grafik und der hohen Spielmotivation begeistert das Programm durch seinen Abwechslungsreichtum. Bei jedem neuen Spielstart wird die Pyramide immer etwas anders aufgebaut. Gespenster, Mumien und Vampire inbegriffen.

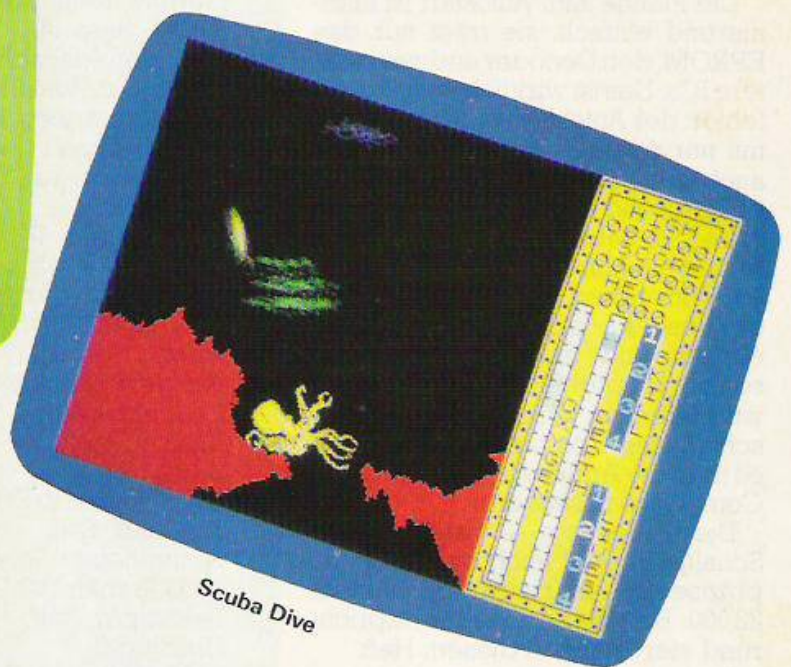
Wesentlich friedlicher, als der Titel es vermuten läßt, geht es bei »Tornado Low Level« zu. Obwohl geflogen wird, ist es weder ein Schieß-

spiel noch eine Flugsimulation. Vielmehr ist Reaktion gefragt, um mit dem teuren Flieger möglichst tief durch die Landschaft zu düsen, ohne an Strommasten oder anderen Widrigkeiten anzuschneppern. Von den Lüften in die Meere: »Scuba Dive« ist ein unwiderstehliches Geschicklichkeitsspiel, dessen originelle Grafik einiges zum Spielvergnügen beisteuert. Ein schatzklüsternder Taucher paddelt durch die Meerestiefen, um Perlen und andere schöne Dinge zu mopsen. Doch Vorsicht vor dicken Fischen und knurrigen Kraken, die unsere feuchte Mission nur allzusehnell gewaltsam beenden. Und vom Wasser geht's gleich in die Berge, wo ein Klassiker beheimatet ist, zu dem es eigentlich nicht mehr viel zu sagen gibt: »Manic Miner«. Wer ihn kennt, der liebt ihn, wer ihn nicht kennt, sollte auf dieses Spiel mal ein Auge werfen. Der Bergarbeiter Willy tappt durch über zwei Dutzend Screens, um kostbare Schlüssel einzusammeln. Ein fesselndes Vergnügen für lange Winterabende. Ebenso wie das Nachfolgespiel »Jet Set Willy« ein Bestseller, der sich ausgesprochen Käuferbeliebtheit erfreut.

# SPIELERREIEN



World Cup Football



Scuba Dive

## Ganz schön sportlich

Apropos Abwechslung: Wer vertrackte Actionspiele mag, wird bei »The Prize« nicht zu kurz kommen. Durch schier unendlich viele Räume und Kammern zischt unser Raumschiff, um Schlüssel zu finden, die gemeinerweise auch noch in der richtigen Reihenfolge aufgeklaut werden müssen. Gefahren in Form abstrakter Gebilde gibt es in rauhen Mengen. Ein Programm, das an den Klassiker »Shamus« für die Atari- und Commodore-Heimcomputer erinnert. Tempo ganz anderer Art bringen Sportspiele. »Daley Thompson's Decathlon« bietet Leichtathletik-Freuden unter der Schirmherrschaft des schnauzbärtigen Zehnkampf-Olympiasiegers. In Anlehnung an den Spielhallen-Renner »Hyper Olympics« entstand ein recht witziges Sportspiel, das den Spectrum-Besitzer mit Hammerwerfen, Weitsprung und anderen Sportarten konfrontiert. Und wer es Beckenbauer schon immer mal zeigen wollte, ist mit »World Cup Football« recht gut bedient. Eine

Fußball-Simulation, die nicht ohne kleine Schwächen ist, bei Joysticksteuerung aber durchaus Spaß macht.

Abschließend ein Strategiespiel, das viel mit diesem schönen Rasensport zu tun hat: »Football Manager« macht aus Ihnen den sorgengeplagten Trainer einer Fußballmannschaft. Spielertransfers, Verletzungspech und die lieben Finanzen müssen bewältigt werden. Sportlicher Erfolg heißt das große Ziel. Sie starten mit Ihrem Team in der 4. Division (vergleichbar mit unseren Amateur-Verbandsligen), um sich Saison für Saison in die Bundesliga vorzuschieben. Sechs Schwierigkeitsstufen sorgen für faire Bedingungen, egal ob Sie Anfänger oder Profi sind. Ein sehr originelles Programm, das auch langfristig zu fesseln vermag. Strategie, Risikobereitschaft und etwas Glück sind gefragt. Schöne Grüße an Dietrich Weise.

Soweit einige »Rosinen« aus dem Spieleangebot für den Spectrum. Alle hier vorgestellten Programme laufen übrigens nur auf der 48-KByte-Version. Erfreulich übrigens, daß fast alle Programme nicht mehr als 30 bis 35 Mark kosten. Ausnahmen: aufwendige Adventures wie »The Hobbit« oder »Valhalla«, für die ungefähr das Doppelte hingeblickert werden muß. Und noch ein wichtiger Tip: Hüten Sie sich vor Blindkäufen! Die Aufmachung der Kassettenhülle mag noch so verlockend sein, was letztendlich zählt, ist die Qualität des Spiels. Und die erkunden Sie am besten bei einer Vorführung im Fachhandel, dann steht dem ungetrübten Spielvergnügen nichts mehr im Wege.

(Heinrich Lenhardt)

Für die meisten Schaltungen, die einen Autostart erfordern, reicht das interne RAM mit Sicherheit aus. Falls 1 KByte zu wenig sind, kann man im Rechner auf 2 KByte erweitern. Bei besonders großem Speicherbedarf kann man im Adreßraum von 32 KByte bis 48 KByte in zwei EPROMs (auf einer zusätzlichen Platine) weitere 16 KByte an Daten unterbringen; RAM-Platinen können am Busstecker angesteckt werden.

## Der Aufbau: verblüffend einfach

Die Platine zum Autostart ist faszinierend einfach: sie trägt nur das EPROM, den Decoder und zwei weitere ICs. Das ist wirklich kein Druckfehler: der Autostart wird tatsächlich mit nur zwei ICs bewirkt. Das sind auch keine Spezial-ICs, sondern ein 74 LS 32 und ein 74 LS 74, ganz normale TTL-Bausteine. Weitere Bauteile sind ein Kondensator, eine Diode, ein Widerstand, ein Busstecker und der Reset-Knopf. Der Aufbau ist also einfach und unkritisch. Der Anschluß ist problemlos: Die Platine wird lediglich hinten an den Busanschluß des Computers gesteckt — es ist keine Änderung im Innern des Computers notwendig.

Die Anleitung — natürlich mit Schaltplan — ist leicht verständlich, präzise und sehr ausführlich: zirka 22000 Buchstaben, das entspricht rund vier Seiten in diesem Heft.

### Preise:

Leerplatine mit Anleitung 30 Mark  
Bausatz mit Anleitung/ohne EPROM ..... 45 Mark  
Die EPROMs können nach Kundenwunsch programmiert werden.

Hersteller: Decker & Computer, Stuttgart.

Diese Firma hat für ZX80/81 unter anderem bereits den kleinsten Auto-Repeat entwickelt (mit nur einem IC — aber einer Anleitung mit zirka 41 000 Buchstaben), außerdem für ZX80/81 und Spectrum die kleinste PIO-Karte (mit nur zwei ICs — dafür aber der längsten Anleitung mit zirka 75 000 Buchstaben).

Ein ZX-Hardwarebuch stammt ebenfalls von Decker & Computer.

Für den Floppy-Controller, der seit mehr als einem Jahr als Labormuster seine Dienste tut, soll jetzt eine Leerplatine angeboten werden: Preis zirka 35 Mark.

Wir sind gespannt darauf!

(mk)

## Autorenverzeichnis

Bachman Knut,  
Ölbergstr. 3,  
5000 Köln 41  
Benzig, Jörg,  
Hirschbergstr. 33,  
7730 VS-Schwenningen  
Bergen, Peter,  
Frankenstr. 29,  
3200 Hildesheim  
Dietl, Joachim,  
Dammstr. 3,  
7100 Heilbronn  
Eifert, Andreas,  
Blumenstr. 44,  
7016 Gerlingen  
Friedrich, Bernd W.,  
Ludwigsburger Str. 2,  
6500 Mainz  
Gerling, Reiner W.,  
Hollergasse 16,  
8551 Heroldsbach  
Herrmann, Josef,  
Oberländerweg 14,  
8415 Nittenau  
Howaldt, Jürgen,  
Kölner Str. 22,  
2800 Bremen 41  
Hölscher, Erika,  
Ernst-August-Str. 5,  
2730 Zeven  
Hugel, Norbert,  
Auf dem Leimen 4,  
6718 Grünstadt  
Kastl, Josef,  
Sittling 27,  
8425 Neustadt/Donau  
Kirchner, Eva,  
Neuenhainer Str. 8,  
6000 Frankfurt/Main  
Lakämper, Rolf,  
Uhuweg 6,  
4830 Gütersloh 11  
Lenhardt, Heinrich,  
Happy-Computer  
Merkel, Hans,  
Franz-Schubert-Str. 14,  
6742 Herxheim/Pfalz  
Richter, Peter,  
Am Maubishof 22,  
4044 Kaarst 1  
Ritz, Gunter,  
Stelzer Str. 43,  
A-4020 Linz  
Römer/Krause,  
Informatik Gemeinschaft 83,  
Schillerstr. 18,  
6473 Gedern 1  
Tiesens, Peter,  
Schloßbergweg 8,  
CH-5400 Baden  
Viebke, Andreas,  
Thurgauer Str. 9,  
1000 Berlin 51  
Weber, Jens,  
Flotenkamp 25,  
2358 Kaltenkirchen

## Impressum

**Herausgeber:** Carl-Franz von Quadt, Otmar Weber

**Chefredakteur:** Michael M. Pauly (py)

**Redakteur:** mk = Manfred Kötting (177)

**Redaktionsassistentz:** Dagmar Zednik-Djadja (237)

**Fotografie:** Janos Feitser, Titelfoto: Alex Kempkens

**Layout:** Leo Eder (Ltg.), Alexander Gerhardt

### Auslandsrepräsentation:

**Schweiz:** Markt&Technik Vertriebs AG, Alpenstrasse 14, CH-6300 Zug.

Tel. 042-223155/56, Telex: 862329 mut ch

**USA:** M&T Publishing, 2464 Embarcadero Way, Palo Alto, CA 94303; Tel. 415-424-0600; Telex 752381

**Manuskripteinsendungen:** Manuskripte und Programm Listings werden gerne von der Redaktion angenommen. Sie müssen frei sein von Rechten Dritter. Sollten sie auch an anderer Stelle zur Veröffentlichung oder gewerblichen Nutzung angeboten worden sein, muß dies angegeben werden. Mit der Einsendung von Manuskripten und Listings gibt der Verfasser die Zustimmung zum Abdruck in von der Markt&Technik Verlags AG herausgegebenen Publikationen und zur Vervielfältigung der Programm Listings auf Datenträger. Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte und Listings wird keine Haftung übernommen.

**Herstellung:** Klaus Buck (180)

**Anzeigenverkauf:** Brigitta Fiebig (211)

**Anzeigenverwaltung und Disposition:** Patricia Schiede (172)

**Vertriebsleitung, Werbung:** Hans Hörll (114)

**Vertrieb Handelsauflage:** Inland (Groß-, Einzel- und Bahnhofsbuchhandel) sowie Österreich und Schweiz: Pegasus Buch- und Zeitschriften-Vertriebs GmbH, Pfeninger Straße 100, 7000 Stuttgart 80 (Möhringen), Telefon (0711) 72004-0

**Bezugsmöglichkeiten:** Leser-Service: Telefon 089/46 13-238. Bestellungen nimmt der Verlag oder jede Buchhandlung entgegen.

**Bezugspreis:** Das Einzelheft kostet DM 14,-.

**Druck:** E. Schwend GmbH, Schmollerstr. 31, Schwäbisch Hall.

**Urheberrecht:** Alle im Sinclair-Sonderheft erschienenen Beiträge sind urheberrechtlich geschützt. Alle Rechte, auch Übersetzungen, vorbehalten. Reproduktionen gleich welcher Art, ob Fotokopie, Mikrofilm oder Erfassung in Datenverarbeitungsanlagen, nur mit schriftlicher Genehmigung des Verlages. Anfragen sind an Hans Hörll zu richten. Für Schaltungen und Programme, die als Beispiele veröffentlicht werden, können wir weder Gewähr noch irgendwelche Haftung übernehmen. Aus der Veröffentlichung kann nicht geschlossen werden, daß die beschriebenen Lösungen oder verwendeten Bezeichnungen frei von gewerblichen Schutzrechten sind. Anfragen für Sonderdrucke sind an Klaus Buck zu richten.

©1984 Markt & Technik Verlag Aktiengesellschaft, Redaktion »Happy-Computer«.

**Verantwortlich:** Für redaktionellen Teil: Michael M. Pauly.

Für Anzeigen: Hannelore Schmidt (152).

**Vorstand:** Carl-Franz von Quadt, Otmar Weber

**Anschrift für Verlag, Redaktion, Vertrieb, Anzeigenverwaltung und alle Verantwortlichen:**

Markt&Technik Verlag Aktiengesellschaft, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon 089/46 13-0, Telex 5-22052

### Telefon-Durchwahl im Verlag:

**Wählen Sie direkt: Per Durchwahl erreichen Sie alle Abteilungen direkt. Sie wählen 089-46 13 und dann die Nummer, die in Klammern hinter dem jeweiligen Namen angegeben ist.**

# DAS VOLLELEKTRONISCHE GESCHENKPAKET

INSTANT COMPUTING FÜR EINSTEIGER



**H**omecomputing – das endlose Vergnügen mit Spiel- und Nutzprogrammen für jedermann! Frohe Botschaft für alle, die noch keinen Computer haben und meinen, dieser Spaß sei für sie zu kompliziert. Ohne schwindelerregende Lernprozesse genießt man sofort Instant Homecomputing durch Einsteiger-Modelle, die darauf programmiert sind, dem Neuling alles spielend beizubringen! Günstige Nachricht für alle, die einen Fernseher und einen Kassettenrecorder besitzen: Sie haben bereits einen wesentlichen Teil der Computer-Anlage. Den Rest schreiben Sie am besten auf die Liste der Wünsche, die Sie sich selber erfüllen. Weil es einfach Sachen gibt, die man sich nur selber schenken kann.

## Instant Computing Paket A (ICA):

Inhalt: 1 Heimcomputer SINCLAIR SPECTRUM 48K ①, 1 Drucker SEIKOSHA GP-50S ②, 8 Programm-Kassetten ③. Der SINCLAIR SPECTRUM ① ist das Gerät, mit dem Homecomputing zur Perfektion gelangt ist, von Millionen Anwendern in aller Welt getestet und für ideal brauchbar befunden: ein Kleingerät der unendlichen Möglichkeiten, für Spielereien wie für Profi-Aufgaben geeignet dank eines Systems, das durch ein Angebot revolutionärer Erweiterungsgeräte mit dem Können und den Bedürfnissen des Benutzers wächst. – Zu einem Heimcomputer, der Freude macht, gehört ein stabiler, tüchtiger Drucker, der wenig Umstände macht und unermüdlich leistungsstark ist. Als dieses Gerät hat sich der SEIKOSHA GP-50S bereits voll bewährt: ein Normalpapier-Drucker mit eingebautem Interface für SINCLAIR ZX81 und SINCLAIR

SPECTRUM. – Und dazu die Software: Acht der erfolgreichsten Programm-Kassetten ③ in einem Sortiment, das die vielseitigen Spielmöglichkeiten beim Homecomputing voll erschließt: Simulationen wie FLIGHT SIMULATION und CHEQUERED FLAG (Autorennen), Brettspiele wie SCHACH, BACKGAMMON und REVERSI, Cartoons wie COOKIE und PSSST und Weltraumabenteurer wie JET PAC. Preis für ICA: DM 798,-

## Instant Computing Paket B (ICB):

Inhalt: 1 Heimcomputer SINCLAIR SPECTRUM 48K ①, 1 THURNALL DISKETTEN STATION ④, 8 Programm-Kassetten ③. Das normale Speichermaterial für Heimcomputer sind Kassetten. Das normale Speicher-material für Profi-Computer sind Disketten. Die THURNALL DISKETTEN STATION ④, eine brandneue Sensation, gehört zu den revolutionären Erweiterungsgeräten, mit denen der anspruchsvolle Anwender einen einfachen SPECTRUM zu einem vollwertigen Profi-Gerät machen kann: ein Disketten-Laufwerk mit einer 150K-Kapazität pro Diskette. Preis für ICB: DM 1398,-

## Instant Computing Paket C (ICC):

Inhalt: 1 Computer SINCLAIR QL, einschließlich der Software-Programme DATENBANK, TEXTVERARBEITUNG, FINANZPLANUNG und GRAFIK ⑤. Kaum je wurde ein Computer mit solcher Spannung erwartet, mit soviel Beifall begrüßt wie das Gerät, das sein Erfinder Clive

Sinclair stolz als den QL (Quantum Leap – Hand des Computer-Spielers. Unsere Marke Quanten-Sprung) der Computer-Technologie bezeichnet: ein Kleincomputer mit der Profipower: 128K, 32-Bit Processor. Zwei eingebaute Microdrives. Hochauflösendes Farbdisplay. Profi-Tastatur. Englische Originalfassung. Deutsches Handbuch wird nachgeliefert. Preis für ICC: DM 1698,-

## Instant Computing Paket D (ICD):

Inhalt: 1 Heimcomputer SINCLAIR ZX81 als Bausatz ⑥. Der SINCLAIR ZX81, seit Jahren weltweit beliebt als das klassische Einsteiger-Modell und inzwischen schon zum Taschengeld-Preis erhältlich, ist als ZX81-Bausatz ⑥ die ideale Anschaffung für Elektronik-Freunde und Do-It-Yourself-Freaks. Preis für ICD: DM 98,-

## Instant Computing Paket E (ICE):

Inhalt: 1 ISS DATENRECORDER ⑦. Der ISS DATENRECORDER ⑦, eine Neuheit, die erst in diesen Tagen auf den Markt kommt, bietet zusätzlich zur üblichen Ausstattung von Kassettenrecordern viele Funktionen und Annehmlichkeiten, die den Umgang mit dem Computer erleichtern. Preis für ICE: DM 129,-

## Instant Computing Paket F (ICF):

Inhalt: 1 Joystick QUICK SHOT inklusive Interface mit 2 Anschlüssen ⑧. Der Joystick ist der Steuerknüppel in der

Hand des Computer-Spielers. Unsere Marke ⑧ ist so stabil, daß sie die Vibrationen der heißesten Spiele aushält. Das Interface mit 2 Anschlüssen ermöglicht das Spielen mit Joysticks. Preis für ICF: DM 79,-

DER COMPUTER-AUSSTATTER  
COMPUTER ACCESSOIRES INT'L GmbH  
Jägerweg 10 · 8012 Ottobrunn

## DER INSTANT ACTION COUPON

Ich bestelle zur sofortigen Lieferung

per Vorausschick

per Nachnahme, zuzüglich Nachnahmegebühren

Stück	Artikel	Preis DM
	Instant Computing Paket A (ICA)*	798,-
	Instant Computing Paket B (ICB)*	1398,-
	Instant Computing Paket C (ICC)*	1698,-
	Instant Computing Paket D (ICD)*	98,-
	Instant Computing Paket E (ICE)*	129,-
	Instant Computing Paket F (ICF)*	79,-

\*Inhalt der einzelnen Pakete siehe Anzeige. Alle Preise inklusive gesetzliche Mehrwertsteuer. Produkt- und Preisänderungen vorbehalten.

Name: \_\_\_\_\_

Straße: \_\_\_\_\_

PLZ/Ort: \_\_\_\_\_

Datum: \_\_\_\_\_ Unterschrift: \_\_\_\_\_

COMPUTER ACCESSOIRES INT'L GmbH  
Jägerweg 10 · 8012 Ottobrunn

# SINCLAIR spricht DEUTSCH mit Ihnen

Ladies & Gentlemen, now we are in Germany. Genau gesagt in Bad Homburg. 15 Minuten nahe am Flughafen Frankfurt. Was Sie das angeht? Nun gut, Sie haben in dieser Zeitschrift sehr viele Informationen über die technologische Qualität – Hardware und die Software mit tausenderlei Möglichkeiten – gelesen. Sie sollten dabei aber noch eines bedenken: SINCLAIR in Deutschland ist nicht allein vorzügliche Technik, die schlichtweg im Preis-/Leistungsverhältnis ● unvergleichlich ist; SINCLAIR heißt auch ein Service, der seinesgleichen sucht. Beispielsweise lassen wir Sie im Kundendienstfall nicht warten. Unser Service-Partner M/A/I steht Ihnen mit seinem flächendeckenden Service-Netz zur Verfügung. Und wir geben Ihnen guten Gewissens auf jedes Gerät ein Jahr Garantie. Und was Sie besonders interessieren sollte: Mit uns von SINCLAIR können Sie jetzt jederzeit DEUTSCH reden.

Ihr

*Ulrich O. Wolf*  
Ulrich O. Wolf  
Sales-Manager

SINCLAIR Computer gibt es praktisch überall: z. B. im Fachhandel, in den Fachabteilungen der Warenhäuser, in gut geführten Radio-/Fernseh-/Phono-Fachgeschäften. Doch weitere SINCLAIR-Händler sind uns immer willkommen. Infos für Profis wie für Newcomer gibt's von SINCLAIR Research Ltd., Bad Homburg v.d.H.

Sinclair Research Ltd.  
Niederlassung Deutschland  
Hessenring 83 · 6380 Bad Homburg  
Tel. 06172/25071 · Telex 418103

**sinclair**