

CK

Die User Zeitung

mit Sonderteil für Commodore,
Atari, Sinclair, TI 99/4A

Nr. 12/1 3. Jahrgang

Atari

- Reviews: The Music Studio
Graphic Arts
Department
- Kurse: Assemblerocke
6502-Maschinen-
sprache-Kurs
- Listings: Filecopy
Wombat

Sinclair Spectrum

- Magazin zum:
Spectrum
- Maschinensprache-Kurs
and Face

Sinclair QL

- Tips + Tricks
- Benutzeroberfläche

TI 99/4A

- Reviews
- Hexmonitor in C
- Combas

Commodore

- Vietnam
- Grafik-Kurs Teil 3

QL-Seiten

Computer Kontakt



ATARI ST

ASSEMBLER-PRAXIS AUF ATARI ST

Roland Löhrl

„ein Altmeister der Assembleranwendung, Herausgeber des Mikrocomputer Magazins MICRO MAG, veröffentlicht bei te-wi seine neueste Darstellung der Assemblerprogrammierung auf ATARI STs.“

Erklärt Grundlagen:

Begriffe und Werkzeuge der Assemblerprogrammierung, erforderliche Systemkenntnisse, systembezogene Erläuterung des 68000er Befehlsformats.

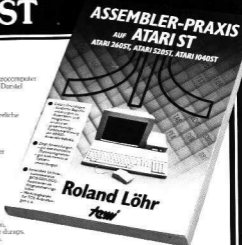
Zeigt Anwendungen:

Hat Ihnen mit Assembler: Aufruf von Assemblern; Steuern ihrer Optionen über Direktiva; Stellungnahme zu realen ATARI ST-Assemblern, Arbeiten in der ATARI ST-Programmierumgebung; Testprogramme zur Programmentwicklung; ein Editor, ein Parser; das Betriebssystem, BIOS Funktionen, BIOS Toolbox, GEMDOS Toolkit, das erweiterte XBIOS. Ansonsten, das Befehlsformat in Musterprogrammen für: E/A Routinen, Formulareisen, dez-bin Rechnerart, Suchverankerung, Adressverankerung, Entscheidungen, Schleifenkonstrukte, Unterprogramme, nummerierte Traps, Bedienen von Interfacebausteinen, Testfunktion, Testwerkzeuge, Datenerkennung, memory dumps, Floppy tests, Funktionen, schnelle RS232 Datenübertragung usw.

Entwickelt Hilfsprogramme:

BIOS Toolbox; GEMDOS-Toolkit; ein Editor, ein Parser; Arbeiten mit Toolkits. Die Programme des Buchs sind auf Diskette vom Autor erhältlich.

Ein Fachtext in klarer Sprache mit lesefreundlichem Druckbild, guter Bildelementation und umfangreichen Listings vor. Musterprogrammen (auf Diskette vom Autor erhältlich) ca. 300 Seiten, Softcover, DM 59,-



te-wi Verlag GmbH
Theo-Prosel-Weg 1
8000 München 40

Weitere te-wi-Bücher



NEC

DAS C-BUCH

Über 47 Kern der Industrie. Für strukturelle C-Programme. Über 800 Beispiele. Ansonsten auf zwei Disketten. ca. 500 Seiten, Softcover, DM 79,-

UNIX

Viele Titeln US Standardbook für UNIX-Programme. Eine vollständige Übersetzer und Einführung in die Version 3.00. 500 Seiten, Softcover, DM 79,-



LOGO - Jeder kann programmieren

(Daniel Witt)
Buch des Jahres in den USA. Best recommended von Pädagogen und deutschen Kulturministeren. Ein britischer Führer durch die ATARI-LOGO. Von Papert's Schüler D. Witt. 384 Seiten, A4, DM 59,-



M68000 FAMILIE, 2 Bde.
1411. Nachschl. ges. 968 Seiten
Genauige Motorola authentische Darstellung von CPU-68000-Architektur, Programmierung, Systemsoftware. Beinhaltet alle 68000-Bausteine sowie 68020, 68881, Bd 1. Grundlagen + Architektur. 668 Seiten, DM 79,-
Bd 2. Anwendung und Bausteine. 400 Seiten, DM 69,-



UMWELTDYNAMIK

30 Programme für systematische Umweltveränderungen auf allen BASIC-Rechnern. Das Buch enthält beides: Ein Programmiersystem zur Simulation dieser Produktformulierungen und 29 kommentierte Modellbeispiele wie Baumstricken, Heizungsbedarf, Nahrungsketten usw. Prospekt anfordern.
Von Hartmut Bessel, 480 Seiten, Softcover, DM 59,-



Mein ATARI Computer
Ihr persönliches Standardwerk deutscher ATARI-Akteure. Kompatibel ATARI 600, 800 System. Praktische Anleitung von Prof. M. S. Cook. 300 Seiten, zwei Disketten, DM 59,-

Spezialzeile lesen mit AEM-GRAPHIC
Prüfbar selbst in der Garage und Fernstudium. Einzigartiges Lesegerät mit ATARI-Graphiksystem. Von Prof. R. Köpcke. 201 Seiten, Softcover, DM 49,-



6502 - Programmieren in Assembler
Dieses Buch behandelt ausführlich die Assembler-sprache-Programmierung für den weiterverarbeiteten Mikroprozessor 6502. Lance Leverahl, 704 Seiten, Softcover, DM 79,-

Noch im Programm: Einführung in die Mikrocomputer-Technik, DM 66,-
Computer für Kinder, ATARI, DM 29,80



Liebe Leser,

eine etwas traurige Mitteilung müssen wir den Commodore-Lesern unter Ihnen machen: Wir werden ab dem nächsten Heft den Commodore-Teil einstellen. Der Grafik-Kurs wird allerdings zu Ende geführt. Sie werden jetzt sicher nach dem Grund fragen.

Wie auch bei den Computern selbst findet im Zeitschriftenbereich ebenfalls eine Konzentration statt. Dies hat bei uns dazu geführt, daß wir bei den Computertypen von Sinclair und Atari sowie dem TI 99/4A immer stärker wurden, dagegen bei Commodore gerade in der letzten Zeit Leser verloren haben. Dieser Prozeß wird sich weiter fortsetzen, weshalb wir uns in Zukunft noch stärker auf die restlichen drei konzentrieren wollen. Speziell den QL und den TI werden wir dabei noch mehr berücksichtigen.

Dagegen gibt es für die Atari-Leser eine interessante Neuigkeit: Am 22.12.86 erscheint die erste Ausgabe unseres neuen ATARI magazins. Lassen Sie sich dieses Heft nicht entgehen, auch die kleinen Ataris werden wir dort gebührend berücksichtigen. Neu ist auch unser Atari-Buchversand hier im Heft mit insgesamt 30 Titeln. Da ist für jeden etwas dabei.

Auch wer regelmäßig unsere Software-Cassetten bestellt, wird sich freuen. Ab sofort gibt's die neuen Programm-Cassetten für Atari, Spectrum und TI 99/4A mit den Listings aus Computer Kontakt.

Bis zum nächsten Mal
Ihr

Thomas Eberle, Chefredakteur

Brandheißer Knötterpreise

TI-99/4A

CGP 86-Programmierer	109,-
Iskra 2000 + 13 Disketten	188,-
Edmore 32-A-Erweiterung	289,-
Ed. 32-A-Erweiterung + Datensatz	279,-
Extended Basic II Plus	109,-
104 Binary Storage T5	85,-
Terminal Simulator II	206,-
Tri-Logic I (32-A-Erweiterung)	175,-
Speichererweiterung	88,-
Funktelegrafie (Digital) T1	179,-
Größtmöglicher Superrechner + Dig-Dig	179,-
+ Disketten + Speicher	36,-
Atari - Charakter - Text - Overlay	36,-
Defender, Dig-Dig, Car Wars, Moon-Moon	36,-
Wunsch Mail - Invaders	36,-
Back Rooms, Video Chess, Adventure	65,-
+ Reservierungswahl im Hand- und Software + Buchmarkt	

Schneider

CGP Plus mit Lasermonitor	350,-
CGP 8128 mit Farnsworth	1649,-
Schneider PC 2M-Mem. + 1 Laufwerk	1849,-
Schneider PC mit 2 Laufwerken	2279,-
Rechen-Diskette 1,2 MB, angeschlossen	799,-
FX 800, angeschlossen	1179,-
EX 800, angeschlossen	1479,-

Atari

16 11 808 - Pizzly 1000	399,-
16 11 1040 - 486 - 133 86 + 1600	719,-
Größtmöglicher	1989,-

Commodore

Commodore 4000	2099,-
Rechen-Diskette FX 800, angeschlossen	1179,-
Commodore EX 800, angeschlossen	1479,-
Commodore C 128 II	1299,-
Commodore Farnsworth 1981	799,-
Commodore Farnsworth 1981	599,-
Edmore-Diskette 1,2 MB + Datenspeicher	799,-
RA2 für 5 1/4" Disk	399,-
etc. + FX 800	1099,-
etc. + EX 800	1399,-
Edmore-Diskette 1,2 MB, angeschlossen	749,-
Commodore Plus II + 1600	699,-
Größtmöglicher Superrechner C 84	1399,-
Atari-Drucker Detron 521 C + Kabel + Fernprogramm	299,-

Verbindungsprogramme (Standard) bis DM 100,- (Mehrwert) Verkaufspreise (DM 3,-) (G 1, Neuzugabe (DM 11,20) (23), Ausland (DM 18,-) (26,-), Lieferung nur gegen Vorauszahlung oder per PKW. Auftrags nur für deutsche. Gesamtprogrammatische Computertipps angeboten gegen Zahlung eines Franchisegebührens.

CSV RIEGERT

Schloßhofstr. 2, 3324 Rucholdinghausen, Tel.: (0 51 41) 5 20 69

Jupitersoft GdBR

Neueste Software für
C 64, C 128, SVI 318/328,
VZ 200, Laser 210/310
ab **7,90 DM**

Adventures, Action, Anwender,
Grafik und Datenprogramme.

**Viel Software stark
im Preis reduziert.**

Kostenloser Katalog bei

Robert Goth Jupitersoft GdBR

Höfningerstr. 34b
86336 Eittingen
(0 91 41) 61 74



ATARI ST Tagestiefelpreis
Spectrum ab 248,-
Sinclair QL ab 448,-
Tastatur f. QL ab 198,-
Opus Discovery ab 389,-
Seiksha SP 1000 ab 699,-
Computer, Drucker + Zubehör
in unserer Gratisinfo
Unsere Kunden
ein frohes Weihnachtsfest
Ursula & Joachim Rudolph

INHALTSVERZEICHNIS

RUBRIKEN	
Vorwort	3
Topprogramm	5
Clubnachrichten	5
Atari-Buchversand	60
CK-Programmservice	62
Buchversand	99
Kleinanzeigen	116
Fundgrube, Inserentenverzeichnis, Impressum	121
USER-CLUB SINCLAIR	
ZX-Spectrum Maschinensprachkurs	10
Assembliertips für den Spectrum	12
ISO FACE 1,0	15
Textverarbeitungsprogramm	16
Graf Eddi - ein Grafkeditor	17
3 kurze Programme für Spectrum	24
Quercopy für Spectrum 16/48 K	26
Zitierschrift	28
Window-Routine	30
Respos für ZX-81	32
Indexomat für den QL	36
Microsoft-Basic auf dem QL	38
Grafische Benutzeroberfläche für den QL	39
Tips und Tricks für den QL	40
COMMODORE-NEWS	
Tips und Tricks zu C64 und Floppy 1514	46
Grafikkurs C64, Teil 3	47
Vietnam für den C64	51
Input de Luxe	55
Amiga-Spieleübersicht	57
Lattice auf Amiga-Platzen	58
Kuma-Seka, ein Makro-Assembler	59
ATARI	
Graphic Arts Department	65
Neuvs für den Gabentisch	66
Kurs: 6502 - Maschinensprache	68
Peters Assembler-Ecke	72
Spiel zum Abtippen: Wombel	78
Disk-Editor	81
Kalkulationsprogramm: Calc 800	82
Hardcopy auf Saikosa GP 500 AT	88
Speed-Tape für Atari XL/XE	89
Filecopy	90
Zeichensuchfinder	91
ATARI ST	
Profi Painter	95
Software-Test: ST-Heimfinanz	97
TI 99/4A	
Mechatronik-Eprommer 99	101
Neue Software: TI-Artist	102
Neuvs von Turbo-Assembler	103
Lange Programme speichern	104
C auf dem TI 99/4A	106
Hexmonitor	107
Spieltesting: SR002	109
U-Boot-Schlacht	112

Bei uns können Sie mitmachen

Computer-Kontakt ist die Homecomputerzeitung zum Mitmachen. Sie können bei uns Programme einsenden, Bücher besprechen, Spiele beschreiben, Tips und Tricks schicken, Fragen stellen und Ihre Meinung sagen. Wir haben für alles ein offenes Ohr. Damit wir aber Ihre Einsendung schnell bearbeiten können und alles mit rechten Dingen zugeht, müssen Sie folgende Punkte beachten:

1. Ihr Brief sollte ein Anschreiben mit Name, Anschrift, Telefon und Einsenddatum enthalten.
2. Geben Sie genau an, welches Gerät Sie haben. Läuft das Programm nur mit Speichererweiterungen oder Zusatzgeräten, müssen diese unbedingt angegeben werden.
3. Zu jedem Programm sollte eine Programmbeschreibung beiliegen. Diese kann mit der Schreibmaschine oder mit einem Drucker geschrieben sein. Der Zeilenabstand muß 2 Zeilen betragen, damit noch Korrekturen oder Anmerkungen eingefügt werden können.
4. Zu jedem Programm gehört grundsätzlich ein Listing und eine Cassette oder Diskette. Wenn Sie aber keinen Drucker haben, reicht auch der Datenträger. Speichern Sie zur Sicherheit das Programm zweimal ab. Cassetten und Disketten können wir nur zurücksenden, wenn Rückporto beiliegt.
5. Berichte, Spielebeschreibungen und Buchbesprechungen müssen ebenfalls zweizeilig geschrieben werden.
6. Wenn wir ein Programm von Ihnen abdrucken, vergüten wir ein Honorar für den einmaligen Abdruck und die Nutzung des Programms in unserem Cassettservice. Die Höhe des Honorars richtet sich nach der Länge und Qualität des Programms. Wir vergüten im allgemeinen bis zu 300 DM, für sehr gute Programme kann es auch mehr sein.
7. Mit der Einsendung gibt der Verfasser die Zustimmung zum Abdruck und erklärt, daß er Urheber der Texte und Programme ist und das ungeschränkte Nutzungsrecht daran besitzt. Sollte der Einsender Programme einschieken, an denen er kein Urheberrecht und kein Nutzungsrecht besitzt, hat er bei Abdruck durch uns etwaige Schadensersatzansprüche von seiten Dritter selbst zu tragen.

**Die nächste Ausgabe
»Computer Kontakt«
erscheint am 27.1.1987**

Topprogramm "Graf Eddi" für den Spectrum

Mit "Graf Eddi" lassen sich vorhandene Grafiken spiegeln, drehen, invertieren, verschieben, kopieren und vieles mehr. Auch eine 24-Zeilen-Hardcopy ist möglich.

Der Autor des Programms, Stefan Arbes, ist 17 Jahre alt und besucht das Friedrich-Dessauer-Gymnasium in Aschaffenburg. Neben der Programmierung von Rechnern beschäftigt er sich mit der hardwaremäßigen Anwendung von Computern bei der Steuerung im Bereich der Musik- und Bühnenelektronik.



**Computer-Kontakt
das Heft mit den
preisgünstigen
Kleinanzeigen**

Der Jejosoft- Preishammer hat zugeschlagen!

- QL (engl.) mit 640 K Speicher **DM 879.00**
- CL (engl.) mit 128 K Speicher **DM 444.00**
- Vorführung und Selbstabholung der Ware nur nach tel. Vereinbarung
- evtl. Leasing (Mietkauf)
- Abdeckhaube für QL **DM 34.00**
- Abdeckhaube für Spectrum **DM 19.00**

MICRODRIVE
**CARTRIDGE
STORAGE BOX**

4-Hubs 20 cartridges

18.50
4 Cartridges
DM **30.-**

Cartridge-Box mit 20 Cartridges **DM 158.00**
Multiface One, neue Version **DM 106.00**

Brandneu: Spectrum-Software
Der Stern des Druden (deutsches Adventure) **DM 49.00**

JEJOSOFT

Kruppstraße 9, 4040 Neuss 21
jeweils am Donnerstag 02 02 1 07 / 81 84
Di Mi Do 12 22 Uhr und 16 0 0 10 Uhr
Kontaktperson: Beate Jäger (24 Uhr)
Kontakttelefon: 0202 1 07 81 84

1000 DM Honorar

In der CK gibt es das Toplisting. Das läuft so, daß wir hier in der Redaktion von allen Programmeinsendungen das beste Programm herausuchen und in der CK als Toplisting abdrucken. Der Autor dieses Programms erhält dann als Honorar 1000.- DM.

Mitmachen können alle Programmierer mit den Geräten Atari, Sinclair und TI 99/4A. Ein Listing muß nicht unbedingt beiliegen, falls der Autor des Programms noch keinen Drucker hat. Werden Programme abgedruckt, die nicht zum Toplisting gewählt wurden, erhält der Autor dafür ganz normal das übliche Honorar. Beim Toplisting ist das Honorar für den Abdruck in den 1000.- DM schon enthalten. Mit der Einsendung seines Programms erklärt sich jeder Autor mit den einzelnen Punkten im Text "Bei uns können Sie mitmachen" einverstanden (siehe Seite 4 gegenüber).

Deshalb Leute aufgepaßt: Bei uns kann man Geld verdienen. Die Chancen für einen Abdruck, oder gar Gewinner des Toplistings zu werden, stehen bei uns immer gut.

Abo-Bestellschein

Ich möchte Computer-Kontakt in Zukunft regelmäßig zugeschickt bekommen. Meine Abo-Bestellung gilt ab der nächsten Ausgabe. Die Abodauer beträgt 6 Ausgaben, also ein Jahr und kann bis spätestens 4 Wochen vor Aboende wieder gekündigt werden. Der Abonnementpreis beträgt 33.- DM einschließlich Mehrwertsteuer und Versandkosten. Für Bestellungen aus dem Ausland wird es aber nur ein wenig teurer: Hier kostet das Abo 37.50.- DM.

Name/Vorname

Straße

PLZ

Ort

Ich bezahle wie folgt:

Ich bestelle ab Ausgabe:

Scheck liegt bei

Vorauskasse auf Postscheckkonto Karlsruhe Nr. 43423-756

Mir ist bekannt, daß ich diese Bestellung innerhalb 8 Tagen widerrufen kann und bestätige dies mit meiner Unterschrift. (Dieses Widerrufsrecht ist per Gesetz vorgeschrieben.)

Datum/Unterschrift

Diesen Bestellschein ausschneiden oder fotokopieren und an Computer-Kontakt, Postfach 1590, 7588 Bretten schicken.



USER-CLUBS

Atari-Clubs

Unser Atari-Club sucht noch Mitglieder. Wir sind derzeit 19 Uscr, die entweder einen 600/800XL bzw. einen 130XE besitzen. Bei genügend großer Mitgliederzahl ist eine monatlich erscheinende Club-Zeitung geplant. Ansonsten bieten wir eine Soft- und Hardware-Vermittlung an, auch eine eigene Software-Bibliothek ist vorhanden. Wir klären auch Fragen von Mitgliedern. Info gegen 80 Pf. Rückporto. Anfragen aus dem ganzen Bundesgebiet sind willkommen!

Michael Schele
Adalbr. 42
2300 Kuhl 1

Unser Atari-Club besteht seit 3 CK-Ausgaben. Wir beschäftigen uns mit den Atari-Computern 400/800 XL und XE. Programmieren wird in Maschinensprache und Basic. Anfängern und Fortgeschrittenen stehen wir mit Rat und Tat zur Seite. Jedes Clubmitglied erhält alle 3 Monate die Clubdiskette mit Utilities, Anwendungen, Spielen, dem Maschinensprachekurs und dem Clubmagazin. Weitere Infos gibt es bei:

Danuta
D. Brenne
Am Schützenbrunn 15
2120 Lüneburg

Wir, einige Atari-User aus Wuppertal, haben einen Com-

puter-Club gegründet und suchen noch weitere Mitglieder (600 XL - 130 XE). Ein Clubmagazin auf Diskette ist in Vorbereitung; es soll zum Selbstkostenpreis (ca. 5,- DM + Porto und Verpackung) abgegeben werden. Einca Clubbeitrag wollen wir möglichst nicht erheben, aber eventuelle Unkosten, die bei einem Treffen aufkommen (z.B. Getränke). Anfragen von Raubkopierern sind zwecklos. Info gegen Rückporto.

ACCW
Günther Wiedenstein
Gemarkter Ufer 15
5600 Wuppertal 2

Hallo Atari-User!

Wir wollen im Raum GF, WOB, BS, SZ und WF einen Computer-Club gründen. Auch eine Club-Bibliothek sowie Club-Treffs mit Unterhaltung und Erfahrungsaustausch sind geplant.

Oliver Hamann
Gladbeckweg 2
3170 Gifhorn

Mein Freund und ich sind beide 14 Jahre alt und suchen Kontakt zu anderen Atari-800XL-Besitzern mit Datensette oder auch Floppy. Sie sollten in unserem Alter sein und auch im Raum Duisburg wohnen.

Daniel Domnich
Sandstraße 3
4100 Duisburg 11

Kontakt gesucht!

Ich suche Kontakt zu Spectrum-Usern in der Gegend des Dümmer Sees (auch Urlaubler).

Michael Beise
Duisburg 454
4995 Sternewald 2

Ich möchte einen Sinclair-Club gründen, der sich mit dem 128er beschäftigen soll. Ein Schwerpunkt der Clubarbeit wird die Software sein. Wer Interesse hat, kann sich melden.

Schäfers Bauwerk
Schülerstraße 9
5484 Bad Honning
Tel. 026 33/9282

Atari-Club

Wir sind vermutlich ein Club mit den gleichen Interessen wie alle anderen und suchen Gleichgesinnte im Raum Lippstadt, Rietberg, Mastholte und Umgebung. Wir besitzen Atari 400 bis 130XE mit 16 K Bibom, Diskette, Cassette, einen Epson FX 80 und eine umfangreiche Bibliothek mit jeder Art von Programmen. Wir suchen keine Leute, die postlagernd arbeiten und nur Programme abstauben wollen. Wir erheben keinen Club-Beitrag und keine Portogebühren. Wir wollen Programme tauschen, Probleme lösen und Tips und Tricks weitergeben. Bei uns gibt es keine Altersgrenze. Für uns sind die Freaks richtig, die mal schnell mit dem Fahrrad oder dem Auto zusammenkommen. (Abholung ist auch möglich.)

Martin Schiller
Schülerstr. 58
4835 Rietberg 3/Mastholte

Führt auch Ihr Euch in Bezug auf Computerliteratur für Euren Spectrum immer mehr vernachlässigt??

S O N D E R H E F T
Sollt Ihr daher auch auf der Suche nach einer optimal abgestimmten Zusammenstellung von Tips & Tricks, Programmierkursen, Basisteilungen und Erfahrungsbereichen bezüglich dieses Homecomputers???

JA?

Dann haben wir vom Spectrum-Profii-Club hier genau das Richtige für Euch, nämlich das aktuellste... "Spectrum-Profii-Sonderheft". Es handelt sich hierbei um ein selten- und inhaltsreiches Heft, dessen Inhalt optimal abgestimmt wurde. Darin werden die besten Artikel aus unseren sämtlichen, bisher erschienenen Clubmagazinen genommen und hier zusammengefasst. Wir haben uns bemüht, möglichst viele Anwendungsbereiche zu treffen und zu behandeln, um dadurch ein umfangreiches Informationsangebot zu schaffen. Von Interface-Bauplänen über smarte Anwendungsbeispiele bis zum Poko, der den Spectrum bei Frustrationen in die ewigen technischen Jaggründe versetzt, ist wirklich fast alles vertreten. Dieses "Spectrum-Profii-Sonderheft" kann gegen Einsendung unserer Unkosten mit 10 DM bei uns angefordert werden (per Scheck oder per verbuchlicher NN-Bestellung, dann jedoch 15 DM). Spectrum-Profii-Club, Lärcherstr. 2, 8099 Maltenhof.

Wer sich für den Club selbst und dessen Leistungen interessiert, kann unter Beiliegung weiterer 5 DM Unkostenersatzung sich umfangreiches Informationsmaterial und ein Probeexemplar des Clubmagazins zusammen lassen. Nun aber als kleine Geschmackprobe einige Ausschnitte aus dem abgedruckten Inhalt des Sonderhefts: ● Baupläne einer Centronice- und Kompatron-Schnittstelle mit passender Software ● Auflistung von ROM-Routinen, Systemkodes und Spielekodes ● des walterten jede Menge Basic- und MC-Lesings ● Tips und Tricks für die Lebenszeit ● Computerkurse ● Erfahrungsbereiche ● Tests und so weiter...

★ Business-Software ★ ★ Commodore - Sinclair ★ ★ QL-Software & Zubehör ★

PC-FIBU	ab 199,-
PC-SM Business Paket	598,-
C64 FIBU	ab 98,-
C64 Kunden-Lief. Buchn.	99,-
Vzware-Vzaster-Multiplex	
SP-Teoword II	69,50
SP-Masterfile/MF-Print	69,-
SP-Statistik II	49,-
SP-Parasol-Compiler	je 98,-
Kompasur-E-Interface	149,-

Katalog DM 3,-
(Bitte Rechner angeben)

Fa. Lucker/CK
R.-Wagner-Straße 71
6239 Kriftel

Gewußt wo!?

Software + Hardware
- nur Markenprodukte -

- Atari
- Commodore
- Sinclair/Spectrum
- Schneider

Testen Sie uns!
Sie werden erstaunt sein!
Ihre Preisliste liegt bereit!

Annette Lange
Pf. 10 37 71 ● ☎ 04241/4882
D-2800 Bremen

DATENKASSETTEN

Deutsches Markenband alle Größen von C5-C93 lieferbar, z.B. C10 ab DM 0,60-DM 0,94 Copy-Service.

Laufend Sonderangebote. Interessant auch für Wiederverkäufer. Preisliste sofort erhältlich.

3,5" Atari 520/290
Preis auf Anfrage
5 1/4 HCPC Preis auf Anfrage
10 Disketten 5,25" HCPC
in Plastikbox (transparente Farbige) Preis auf Anfrage
Preis gelten ab 10 Stück.
5,25" Diskettenkopien auf Anfrage.

Holschuh Tapes
Kolbenstr. 67, 6140 Bensheim
Tel. 062 51 / 6265



LOGICAL-GEWINNER

Bei unserem etwas mitglückten Logical aus Heft 8-9/86 haben wir ja alle Einwendungen für die Auslosung genommen, da es keine einwandfrei richtige Lösung gab. Hier die Gewinner, die je ein Jahresabonnement Computer Kontakt erhalten:

- | | |
|--|--|
| ▶ Martin Grimm
Forstwinkl 3
8240 Schönbühl | ▶ Ludwig Philipp
Brennerstr. 18
8012 Octobrunn |
| ▶ Albert Krause
Auf dem Kuhl 62
5800 Hagen 1 | ▶ Peter Stock
Pfarrweilerstr. 2
8590 Marktredwitz |
| ▶ Wolfgang Labus
Previsivstr. 29
7000 Stuttgart 40 | ▶ Ute Eilermann
Weidenweg 1
2126 Adendorf |
| ▶ Robert Schuh
Im Lochgarten 11
5000 Köln 90 | ▶ Harald von Gorkom
Horner Str. 12
8221 Teitenhausen |
| ▶ Karin Wisniet
Schlopperweg 8
8458 Sulzbach-Rosenberg | ▶ Markus Wolf
Buchenweg 5
3305 Sickinge |

Wer sucht noch alte CK-Hefte?

Alle neuen Leser haben bei uns die Möglichkeit, die zurückliegenden Hefte nachzubestellen. Die Ausgaben von 1984 sind nicht mehr lieferbar. Bestellt wird mit untenstehendem Bestellschein. Die Lieferung erfolgt aber nur gegen Vorauskasse in Form von Briefmarken oder gegen Scheck.

Bestellschein für CK-Hefte

Ich möchte folgende CK-Hefte bestellen:

- | | |
|---|-------|
| Ex. Heft Januar '85 (4,50 DM) | |
| Ex. Heft März (4,50 DM) | |
| Ex. Heft April (4,50 DM) | |
| Ex. Heft Mai (4,50 DM) | |
| Ex. Heft Juni (4,50 DM) | |
| Ex. Heft Juli (4,50 DM) | |
| Ex. Heft August-September (4,50 DM) | |
| Ex. Heft Oktober (4,50 DM) | |
| Ex. Heft November (4,50 DM) | |
| Ex. Heft Dezember-Januar '85/'86 (5,50 DM) | |
| Ex. Heft Februar-März (5,50 DM) | |
| Ex. Heft April-Mai (5,50 DM) | |
| Ex. Heft Juni-Juli (5,50 DM) | |
| Ex. Heft August-September (5,50 DM) | |
| Ex. Heft Oktober-November (5,50 DM) | |
| Versandkosten (1-2 Hefte 1,40 DM,
2-4 Hefte 2,00 DM, 5-15 Hefte 3,00 DM) | |
| Summe | |

Meine Anschrift:

Der Bestellschein einlesen an den Computer Kontakt, Postfach 1640, 7518 Bretten.

Verlangen Sie mehr!

Datenverarbeitung mit dem Sinclair QL



Rainer Fischer
Datenverarbeitung mit dem Sinclair QL

1986, 196 S., kart., DM 42,-
ISBN 3-7785-1325-7

Das Buch zeigt zunächst die entsprechenden Möglichkeiten von Super-BASIC und führt den Leser dann in die Handhabung der Datenverarbeitungsprache von QL-ARCHIVE ein. Der Leser erhält dabei wichtige Hinweise für die Erstellung von strukturierten Programmen.

HALO LEMKE



Hajo Lemcke
Assembler mit dem C-64

1985, 227 S., 22 Abb., 6 Tab., kart., DM 38,-
ISBN 3-7785-1091-6

In der Maschinensprache nutzen Sie die Fähigkeiten Ihres Computers erst richtig aus. Diese zweibändige Einführung verhilft Ihnen dazu. Jeder Befehl bzw. jede Gruppe wird mit einem lauffähigen Beispiel vorgestellt, das sich später in große Programme einbauen läßt.

Cracker, Hacker, Datensammler



Thomas Tai
Cracker, Hacker, Datensammler

1986, 86 S., kart., DM 24,-
ISBN 3-7785-1236-6

Die Praxis der meist jugendlichen Kopierer steht im Mittelpunkt dieser mit vielen authentischen Fällen versehenen Schrift, die in einer Art Bestandsaufnahme des Software(schwarz)marktes beide Seiten zu Wort kommen läßt.

BESTELLCOUPON

einlesen an: Dr. Alfred Hüthig Verlag,
Im Weiher 10, 6900 Heidelberg

Name, Vorname

Strasse, Nr.

PLZ, Ort

Datum, Unterschrift



Hüthig

USER CLUB

sinclair®

Sinclair ist ein eingetragenes Warenzeichen der Sinclair Ltd.

Hallo Freunde,

noch immer ist nicht geklärt, ob, und wenn ja, wie es mit Sinclair-Produkten weitergeht. Im Sinclair-Teil dieser CK werdet ihr z.B. eine Kurznachricht über Thor, den neuen QL-kompatiblen Computer finden. Aus England kommen Meldungen und Gerüchte, wonach es bereits zwei neue Spectrum geben soll. Der eine, aus dem Hause Amstrad, sieht aus wie der 464 von Schneider, also mit vernünftiger Tastatur und eingebautem Recorder. Von dem anderen Gerät hört man die wundersamsten Dinge; es soll angeblich Fähigkeiten des Amiga erreichen. Wie gesagt, die Gerüchteküche brodelt, Fakten liegen nicht vor.

Im Vorwort der letzten Ausgabe habe ich an dieser Stelle das Programm "G-Force" angekündigt. Wie ihr sicher bemerkt habt, haben wir es dann kurzfristig streichen müssen.

Da "G-Force" Überlänge hat, wollen wir es auch jetzt nicht abdrucken. Ihr könnt das Programm aber preiswert direkt auf Cassette beim Verlag bestellen.

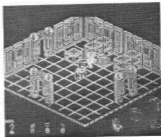
Jetzt aber zu den harten Tatsachen. Wir haben – wie üblich für den ZX 81, den Spectrum und den QL wieder eine Menge Informationen aufbereitet. Besonders erfolgreich ist, daß das Topprogramm des Monats wieder einmal für den Spectrum geschrieben wurde. Der Grafiker in diesem Heft wird sicher vielen Anwendern Freude bereiten. Ähnlich begeistert werden sicher die immer noch zahlreichen Besitzer eines ZX 81 über "Respos" sein. Auf jeden Fall ist der Sinclair-Teil wieder prall gefüllt; die anstehenden Feiertage dürften damit gerettet sein.

Rolf Knaore

Batman

Robin, der Freund und Helfer Batmans, ist entführt worden, und natürlich will ihn der Held sofort befreien. Dazu muß er aber erst einmal die sieben versteckten Teile seines Batmobils finden. Laut Anleitung sind rund 150 Räume zu durchsuchen, in denen es vor Gefahren nur so wimmelt. Bei Programmstart ist der Held schlecht ausgerüstet. Für eine erfolgreiche Suche müssen nun

aucher den Fahrzeugteilen noch andere Ausrüstungsgegenstände aufgetrieben werden. Die Batboots z.B. ermöglichen Batman das Überspringen diverser Gefahren. Weitere Extragaben machen Batman schneller oder vorübergehend unverwundbar; sogar ein Extraleben kann er erhalten. Mit der kompletten Ausrüstung und den Zagaben steigen Batmans bzw. des Spielers Chancen ungemein. Ist



Batman
Grafisch aufwendig gestaltet und in der Aufgabe anspruchsvoll

auch das Batmobil repariert, kann die Suche nach Robin endlich beginnen.

Bekanntlich sind neben Kinohits auch Comic-Helden beliebtes Material für Programmierer bzw. für Softwarefirmen, die sich allein durch den bekannten Namen schon ein Geschäft versprechen. Jüngstes Opfer dieses Trends ist Batman, der Held von Gotham City, der schon seit Jahrzehnten mit seinem immerjüngeren Helfer Robin in dieser Stadt für Ordnung sorgt. Bei der Bearbeitung durch Ocean entstand ein Action-Adventure, das sich hauptsächlich auf das schon übliche Sammeln und Suchen diverser Gegenstände konzentriert.

Das Programm ist mit Spielern wie "Knight Lore" oder "Gunfight" zu vergleichen; auch hier ist eine 3-D-ähnliche Animation vorhanden. Da es mittlerweile Umsetzungen solcher Programme gibt, ist es schade, daß den Programmierern zu diesem Thema nicht mehr eingefallen ist. Es muß aber auch gesagt werden, daß "Batman" grafisch sehr liebevoll gestaltet wurde und die gestellte Aufgabe so knifflig ist, daß man längere Zeit benötigt, um damit klarzukommen. Alles in allem ein nettes Programm, dem eine neue Spielidee aber gutgetan hätte.

System: ZX Spectrum 48 K
Hersteller: Ocean
Rolf Knaore

Paperboy

Der deutsche Vertrieb dieses Programms (Peter West Records) hat uns zum Test eine Vorabversion von "Paperboy" zur Verfügung gestellt. Mit diesen Versionen ist das immer so

eine Sache, da man nie genau weiß, was noch fehlt. Ich glaube aber, daß "Paperboy" bis auf den noch fehlenden Sound schon fast fertig ist. Wenn ihr dieses Heft in Händen haltet,

soll das Programm aber bereits in der endgültigen Version vorliegen.



Worum geht es nun bei "Paperboy"? Wer hin und wieder Spielhallen besucht, weiß sicher schon Bescheid, denn dort gibt es das Spiel schon länger. Es hat sich sogar zu einem echten Renner entwickelt. Hier kann ich gleich einmal erwähnen, daß die Umsetzung auf den Spectrum den Programmierern hervorragend gelungen ist, die Kopie steht dem Original kaum nach.

Die Spielhandlung ist recht simpel. Nach amerikanischem Vorbild muß ein kleiner Junge auf seinem Fahrrad Zeitungen ausliefern, d. h., die Zeitungen im Vorbeifahren in die entspre-

chenden Kästen (oder daneben) werfen. Dafür gibt es Punkte. In seiner Tasche hat der Paperboy jeweils 10 Exemplare. Sind diese verteilt, muß er vom Straßenrand ein neues Paket aufnehmen. Das klingt hier ziemlich einfach, erweist sich in der Praxis aber als wesentlich schwieriger. Der Bürgersteig, auf dem der kleine Kerl entlangsaust, ist nämlich nicht nur für ihn da. Mal kommen ihm Fußgänger oder Tiere entgegen, mal steht eine Mülltonne im Weg. Besonders hässig wird es, wenn er die sehr verkehrsreiche Straße überquert. Es gibt jedenfalls zahlreiche Überraschungen, die ich hier nicht alle verraten will. Das Spiel ist nicht in Level, sondern in Wochentage unterteilt. Nach jedem Tag bekommt der Spieler einen Service Report auf den Bildschirm, bevor es mit dem Streif weitergeht.

"Paperboy" bietet eine witzige Spielidee und -umsetzung und eine tolle Grafik. Mehr kann man kaum verlangen. Dieses Programm sollte bei keinem spielbegeisterten Spectrum-Freak in der Sammlung fehlen.

System: Spectrum 48 K
Hersteller: Elite
Ref: Knaorr

Hocus Focus

Jeremy, hauptberuflich Reporter, nebenberuflich Punk, soll für seinen Arbeitgeber, den Daily Shocker, eine Story recherchieren. Irgendein Professor hat eine tolle Erfindung gemacht, bei der es um Mutationen geht. Soweit die Rahmenhandlung zum Programm "Hocus Focus". Klingt doch eigentlich ganz interessant. Die Um-

setzung auf den Spectrum läßt davon aber nicht mehr viel übrig. Auf dem Bildschirm stellt sich "Hocus Focus" als übliches Lauf- und Suchspiel dar. Grafik und Sound liegen eher unter dem Durchschnitt. Man kann sich mit diesem Programm eine Weile beschäftigen, kann es, aber genauso gut bleibenlassen.

System: Spectrum 48 K
Hersteller: Quicksilver
Ref: Knaorr



Hocus Focus
Lauf- und
Sucharbeit
eines
Reporters

★ ZX-Spectrum ★ ZX-Spectrum ★ ZX-Spectrum ★

DISCOVERY 180	DM 388.-	GAMES PLAYER	
SANYO-MONITOR	DM 240.-	Joystick-Interface mit Regelung der	
PANASONIC-Drucker	DM 648.-	Spiegelschwindigkeit	DM 39.-
KX-P 1060		SPRACHSINTESEIZER	
KEMPTON-E-Drucker-		(JK-Tronics)	DM 79.-
Interface	DM 140.-	3-KANAL-SCHEIN-	
Tastatur (JK-Tronics)	DM 118.-	SYNTHESIZER	DM 79.-
dk/tronica-Drucker-		48-K-Spectrum-Kundendienst-	
Interface	DM 90.-	rückläufer mit zum Teil kleinen	
Trojan-Lightpen	DM 50.-	Fehlern, jedoch optisch erwandelt,	
48k-Erweiterung	DM 49.-	originalverpackt mit engl. Handbuch,	
Joystick-Interface	DM 20.-	Netzteil und kompl. Kabelsatz,	
		ohne Garantie	DM 148.-

Liste C02 mit weiteren Angeboten anfordern oder direkt bestellen bei

U. Kunz Junge Helden 3
D-7500 Karlsruhe 41

Telefonische Bestellannahme: 0721 / 48 26 76 (bis 20 Uhr)

**New's
New's
New's** Ständig die neuesten
Programme für alle
gängigen Rechner
(Commodore C16,
C64/128, Amiga,
CPC, Sinclair,
Atari, ST)

New's Softwarevertrieb
Inhaber Karl-Heinz Klug
Händler-TEL. 0211/489530

MN Michael Naujoks

Light Pen	DM 68.-	1942	DM 25.90
3-Kanal-Soundsynthesizer	DM 98.-	Dan Dare	DM 32.90
(JK Tr.) mit deutscher Anleitung und Software-		Dragon Lair	DM 32.90
Erweiterung	DM 98.-	Great Rider	DM 30.90
Spektrumsynthetizer	DM 140.-	Ghost Escape	DM 25.90
(JK Tronics)	DM 140.-	Heart Land	DM 34.90
Spectrum	DM 140.-	Knight Rider	DM 29.-
Multiface One	DM 109.-	Moonlight Madness	DM 25.90
JoyStick-Interface	DM 29.-	Nightmare Rallye	DM 27.90
JoyStick-Turbo-Pro	DM 35.05	Now Games 3	DM 32.90
(siehe Microcomputer)	DM 49.-	Revolution	DM 29.-
The Art Studio	DM 29.-	Street Hawk	DM 25.90
Tasword II	DM 38.-	The Trap Door	DM 25.90
Masterfile			

Hardware

Sinclair
Atari

Software

Spectrum 128 K		Atari ST	
Spectrum 128K	DM 429.-	Electronic Pool	DM 99.90
(6, 16, 32-Kurz)		Hacker II	DM 79.90
Austerlitz	DM 34.90	Karate	DM 69.90
Fairlight	DM 34.90	Leaderboard Tournament	DM 37.90
Hacker	DM 34.90	Mercenary	DM 69.90
Heroes of Jeand	DM 34.90	Protector	DM 59.90
The Planets	DM 34.90	PSION Schach	DM 79.90
QL	DM 449.-	Shuttle II	DM 59.90
(J. & B. Kaufmang)		Fight Wizard	DM 64.90
QL-Art	DM 99.-	Space Station	DM 59.90
Cartridge Doctor	DM 59.90	Star Gladiator	DM 79.90
Mort Villa Manor	DM 59.90	Temple of Asphal	DM 55.90
Orbis	DM 47.90	Wintergames	DM 79.90
Wanderer	DM 59.90		

Kostenlosen Katalog S/10 anfordern!

Als Preise zurück: Postpachtsache: DM 3,- bei Vorauskasse, DM 6,- bei Nachnahme.
Fordern Sie unseren kostenlosen Katalog an, Unbedingt Computer-Typ angeben.

Entwicklung & Vertrieb von
Computer Soft- und Hardware
Rottmannstr. 40, 6900 Weiskirchen

Hotline:
(06221) 46885

Maschinensprachekurs auf dem ZX-Spectrum

Teil 4: Programmaufbau, Ein- und Ausgabebefehle

Nachdem in den letzten Teilen dieser Serie schon verschiedene Befehlsgruppen erklärt wurden, wird diesmal schwerpunktmäßig auf die eigentliche Entwicklung eines MC-Programms eingegangen. So ganz nebenbei werden dabei noch die Ein- und Ausgabebefehle der Maschinensprache erläutert.

Wenn man sich an seinen Spectrum setzt, um ein MC-Programm zu schreiben, sollte man sich erst einmal auf einem Blatt Papier notieren, wie man den zur Verfügung stehenden Speicherplatz nutzen möchte. Dazu nimmt man folgende Werte zu Hilfe. Die Größe des Basic-Programmteils inkl. Variablen und möglicher Microdrive-Maps und Kanaldaten ist nötig, um den Beginn des freien Speicherbereichs festzustellen. Dies kann man, nachdem alle Variablen und Kanäle belegt worden sind, durch Abfrage der Systemvariablen `STKEND` (Peek 23653 + Peek 23654 *256) relativ genau feststellen. Man sollte dann noch ca. 1 KByte dazuzählen, da noch der Bereich für den Stack und `GOSUB`-Stapel folgt (siehe auch Speicheraufteilung im Spectrum-Handbuch, Kapitel 24). An dieser Adresse kann man nun mit dem Basic-Befehl `CLEAR >Adresse< RAMTOP` setzen, damit der Basic-Teil seine genauen Grenzen kennt. Der freie Speicherbereich für das MC-Programm beginnt nun genau 1 Byte über `RAMTOP`.

Um das Programm schreiben zu können, wird natürlich auch ein Assembler benötigt. Je nach Art handelt es sich dabei um einen ortsfesten (steht immer an der gleichen Stelle im Speicher) oder frei verschiebbaren Assembler. Aus der Anleitung sind nun seine Länge und (bei ortsfesten) die Startadresse zu entnehmen, um den Platz bei der Speichervergabe zu berücksichtigen.

Des Weiteren ist noch beim Sourcetest des Assemblers zu beachten, daß er sehr viel Platz verbraucht (auf 1 KByte Code kommen ca. 8 KByte Source, bei viel Kommentar sogar einiges mehr). Dies ist aber nur dann von Bedeutung, wenn man große Programme am Stück schreibt und diese mit geladenem Assembler testen will. So dürfte es normalerweise keine Speicherprobleme geben, da die meisten Assembler die verschiedenen Blöcke des freien Speichers immer auf deren Sicherheit vor gegenseitigem Überschreiben prüfen.

Hat man nun einen Platz für sein Programm gefunden, so kann das eigentliche Programmieren beginnen. Ein Programm besteht zumeist aus lauter kleinen, vorher im einzelnen getesteten Routinen, die dann von der Hauptroutine des Programms je nach Bedarf aufgerufen werden. Jede dieser Routinen besitzt verschiedene Merkmale: Welche Übergabeparameter stehen in welchen Registern oder werden aus welchen Speicherbytes geholt? Welche Register und/oder Speicherbytes werden von der Routine verändert und/oder enthalten ein mögliches Ergebnis? Daneben können noch Länge und Ausführungszeit der Routine für das Gesamtprogramm von Bedeutung sein. Mit dieser Methode kann man nun eigenständige Unterprogramme schreiben, die auch in anderen Programmen ohne Probleme verwendet werden können.

Daneben benötigt ein MC-Programm noch verschiedene Teile des Speichers für Daten. In diesen Bytes sind dann z.B. Daten über den Aufbau einer im Programm vorkommenden Figur und Programmvariablen (z.B. aktuelle Position der Figur) abgelegt. Dabei sollte man aber nicht vergessen, daß im

ROM des Spectrum schon umfangreiche Daten zur Verfügung stehen (z.B. der Zeichensatz ab Adresse 3D00h bis 3FFFh). Man kann also von seinem eigenen MC-Programm auf Routinen im Spectrum-ROM zugreifen, sollte sich aber vorher ein dokumentiertes ROM-Listing über den Buchhandel (z.B. "Das Sinclair Spectrum ROM", Hueber Verlag) besorgen, damit man auch weiß, welche Routinen vorhanden sind und wie diese arbeiten.

Das abgedruckte Listing ist ein Beispiel für den Aufbau eines einfachen MC-Programms. Begonnen wird mit der Hauptaufrufzeile für die jeweiligen Unterprogramme. Diese sind mit ihren Beschreibungen hinter der Hauptroutine abgelegt. Am Schluß folgen noch die definierten Bytes mit den Programmdaten. Die Hauptspringleiste des Programms, beginnend bei Label `START`, springt nach Beendigung aller UPs wieder an den Start der Routine. Damit man diese kleine Endlosschleife aber auch wieder unterbrechen kann, wird zu Beginn erst einmal eine `BREAK`-Routine aufgerufen, die über eine ROM-Einsprüngeadresse die Shift- und Space-Taste abfragt und das Ergebnis auswertet. Dazu wird noch ein Speicherbyte des Programms (Label `RETURN`) auf einen bestimmten Wert hin überprüft. Ist dieser vorhanden oder `BREAK` gedrückt, wird das MC-Programm nach Aktualisieren des Stacks verlassen. Das Label `RETURN` kann auch von einem anderen Programmteil aus gesetzt werden, um ein Beenden des Programms zu signalisieren. (Dies geschieht in diesem Beispiel bei der `STEUER`-Routine.) Eine solche `BREAK`-Abfrage sollte man an mehreren Stellen eines umfangreichen Programms einbauen, um vorkommende Endlosschleifen elegant zu unterbrechen.

Die zweite Routine, welche von der Endlosschleife aufgerufen wird, ist das `PRINT`-Unterprogramm. Es holt aus der durch das Label `AENDER` gekennzeichneten Speicherstelle den Wert und vergleicht ihn auf einen bestimmten Zustand (44h), der dem UP signalisiert, daß sich die Position der Figur geändert hat und neu gekennzeichnet werden muß (Zeilen 300-320). Dies ist ein Beispiel für die Übergabe von Ergebnissen an anderen UPs, hier von der `STEUER`- an die `PRINT`-Routine. Nachdem auf diese Weise festgestellt wurde, daß sich die Position der Figur verändert hat, wird diese als Label `PRINT1` gezeichnet. Dazu wird aus der in `BC` stehenden Position im Unterprogramm `UMWAND` die Bildschirmadresse mit Hilfe einer weiteren `ROM`-Routine errechnet. Diese Routine gibt in `HL` die dazugehörige Bildschirmadresse und in `A` die Position des Bits in dieser Adresse zurück. Nun weiß unser Programm, an welcher Stelle im Byte die Figur beginnt, also wie oft diese verschoben werden muß (Zeilen 480-510). Danach werden die ermittelten zwei Bytes an die vorbestimmten Stellen im Bildschirmspeicher geschrieben. Dies geschieht durch eine `XOR`-Verknüpfung (Zeilen 520-570), wodurch die Figur bei wiederholtem Schreiben an der gleichen Position gelöscht (Zeilen 330-360) und an die neue geschriebene wird. Der in Zeile 370 verwendete Befehl ist ein üblicher Trick, um das `A`-Register auf 0 zu setzen.

Der letzte Programmaufbau der Hauptspringleiste vor einem erneuten Durchlauf der Schleife wird zur Abfrage der `Cursor`-Lasten 5 bis 8 und 0 benötigt. Das Unterprogramm `STEUER` verwendet dazu den für uns neuen Befehl zur Abfrage von Ports. Dies sind die Verbindungen der Zentraleinheit

mit der Peripherie (Tastatur, Lautsprecher, Drucker, Microdrive). Für die Eingabe (von der Peripherie an den Rechner) wird der Befehl IN Zielregister (Adresse) verwendet. (Adresse bezieht sich hier nicht auf ein Speicherbyte, sondern auf die Adresse, unter der das abgefragte Gerät an den Rechner angeschlossen ist.) Beim Spectrum ist die eigene Peripherie sogar nur vom Zustand einzelner Bits der Adreßzahl abhängig (siehe auch die Tabelle der wichtigsten IN- und OUT-Adressen). Bei der Abfrage gibt es noch eine Unterscheidung zwischen den Befehlen IN A, (C) und IN A, (wert): Beim 2. Befehl gibt der Wert zwischen den Klammern die Adresse des Gerätes an, also eine 8-Bit-Adresse. Beim 1. Befehl ist aber neben dem C-Register auch noch das B-Register für die Adresse zuständig, so daß sich aus dem Wert des Registerpaars BC eine 16-Bit-Adresse des Gerätes bildet. Dies ist z.B. bei der Abfrage der Tastatur-Ports ausschlaggebend. Der Inhalt des abgefragten Ports, also der Wert, den das Gerät bei einer Anforderung

auf dem Datenbus oder einzelnen Bits davon liefert, wird in das vor der Adresse stehende Register geschrieben.

Bei der Ausgabe an ein Peripheriegerät (z.B. den Lautsprecher) wird der Befehl OUT (Adresse), Wert verwendet. Auch hier kann die Adresse durch einen 8-Bit-Wert oder den 16-Bit-Inhalt des BC-Registerpaars gekennzeichnet werden. Als zu übertragender Wert an das Peripheriegerät kann entweder eine feste Zahl oder ein Register angegeben werden. Dieser Wert wird dann gleichzeitig mit der Adresse (diese liegt auf der Adreßleitung) an den Datenbus ausgegeben und ausgewertet. Auch hier ist zu beachten, daß schon der Zustand eines einzelnen Datenbits das Gerät ansprechen kann (z.B. Bit 4 der Portadresse 254 des Spectrum/Lautsprecher, s. Tabelle). Im Programmteil STEUER werden nun die Tastatur-Ports der Cursor-Tasten abgefragt (Zeilen 780-820); anhand ihrer Bitwerte wird die aktuelle Position der Figur geändert. War zusätzlich noch die Taste 0 gedrückt,

so wird dies in der durch RETURN gekennzeichneten Speicherstelle signalisiert, wie auch im Label AENDER eine Änderung der Position dem PRINT-Teil mitgeteilt wird.

Liste einiger IN- und OUT-Adressen

Adr. 254

0. Adreßbit auf null

Bit 0,1,2 BORDER-Farbe wählen,
Bit 3 MIC-Eingang abfragen
Bit 4 Lautsprecher ein/an
Bit 6 EAR-Ausgang schaltbar

Adr. 253

1. Adreßbit auf null, Ausgabe an den Sinclair-Drucker, Adreßbit 3 und 4 für Microdrive

Adr. 223

5. Adreßbit auf null, Abfrage des Kempson-Joysticks, Bits 0-4 müssen abgefragt werden

Im Sinclair-Handbuch, Kapitel 23, sind die Tastatur-Portadressen angegeben.

Am Ende des Programms stehen dann die ganzen hercit-

gestellten Speicherstellen, die durch ein Label und die Assemblerbefehle DEFDB oder DEFWB festgelegt werden. Neben den bisher schon besprochenen Labels sind noch Daten über den Aufbau der Figur (hier stellen sie den Schriftzug "CK" dar) abgelegt, wobei als letztes Byte der Wert FFh steht, damit die PRINT-Routine das Ende der Liste erkennt (Zeilen 610-640). An dieser Stelle sind die Zeilen 400 und 410 zu beachten. In Zeile 400 wird mit dem Label FIGUR die tatsächliche Adresse dieses Labels geladen, während in 410 mit (AKTPOS) der Wert des Bytes bei der Adresse des Labels AKTPOS geladen wird (in diesem Fall sogar die beiden Bytes bei der Adresse).

Das abgedruckte Programm und die Tabelle mit den IN- und OUT-Adressen dürfen allen Lesern wieder ein großes Experimentierfeld eröffnen, so daß diese Lektion des Maschinen-sprachkurses hier beendet werden kann. Im nächsten Teil wird die Serie dann mit der Erklärung der Sonder- und Block-befehle abgeschlossen.

Thomas Beroldo

Figur-Steuerung

```
50 ; Hauptsprungleiste
60 ;
65 CALL PRINT1 ; zeichne Figur
70 XOR A ; lösche A
80 LD (RETURN),A ; lösche Endeerkennungsbyte
90 START CALL BREAK ; Programm beenden ?
100 CALL PRINT ; Figur zeichnen
110 CALL STEUER ; Bewegen der Figur
120 JR START ; Schleife wiederholen
130 ;
140 ;
150 ; BREAK-Routine
160 ; Abfrage BREAK-Taste und Label RETURN
170 ;
180 BREAK CALL RUFSA ; Aufruf ROM-Routine
190 JR NC,BREAK ; Shift + Space ist gedrückt
200 LD A,(RETURN) ; Lade Ende-kennungs Byte
210 CP #33 ; Vergleich ob auf Ende gesetzt
220 RET NZ ; Kein Ende, Routine verlassen
230 BREAK1 POP HL ; vor beenden Stack normalisieren
240 RET ; Programm beenden
250 ;
260 ;
270 ; PRINT-Routine
280 ; schreibt Figur an aktuelle Position
290 ;
```

```
300 PRINT LD A,(AENDER) ; Lade Aenderungskennung
310 CP #44 ; Vergleich ob geändert gesetzt
320 RET NZ ; nicht gesetzt, nicht zeichnen
330 CALL PRINT1 ; erst lösche der Figur
340 LD HL,(NEWPOS) ; Laden neue Position an ...
350 LD (AKTPOS),HL ; ... aktuelle Position
360 CALL PRINT1 ; zeichnen an der neuen Pos.
370 XOR A ; Lösche A
380 LD (AENDER),A ; Setze Aenderungskennung zurück
390 RET ; beenden dieser Routine
400 PRINT1 LD DE,FIGUR ; Laden Figurbytesbeginn
410 LD BC,(AKTPOS) ; Lade aktuelle Position
420 PRINT2 PUSH BC ;rette BC auf Stack
430 PUSH DE ; rette DE auf Stack
440 CALL UNWIND ; ermitteln der Bildschirmadresse
450 POP DE ; hole DE vom Stack
460 LD A,(DE) ; Lade Figurbyte
470 LD C,#00 ; Lösche C
480 INC B ; erhöhe zum richtigen Zahlen
490 PRINT3 SRL A ; rechtsschieben der Figur
500 RR C ; Übertrag in C schieben
510 DJNZ PRINT3 ; weiter schieben ?
520 XOR (HL) ; Lade Bildschirmbyte zur Figur
530 LD (HL),A ; Lade Figurbyte in Bildschirm
540 INC HL ; nächstes Bildschirmbyte
550 LD A,C ; Lade A mit Übertrag
560 XOR (HL) ; Lade Bildsch. zum Übertrag
570 LD (HL),A ; Lade Übertrag in Bildsch.
```

580	INC DE	; Zeiger auf nächstes Figurbyte	920	DEC B	
590	POP BC	; hole Position vom Stack	930	STEBL BIT 4,D	; Abfrage Taste "5"
600	DEC B	; verändere y-Position um 1	940	JR NZ,STEBL	
610	LD A,(DE)	; Lade Figurbyte	950	DEC C	
620	INC A	; ist Figurbyte Null ?	960	STEBL BIT 0,A	; Abfrage Taste "0"
630	JR NZ,PRINT2	; nein, also weiter	970	JR NZ,STEBL	
640	RET		980	LD A,#33	
650			990	LD I(RETURNS),A	; setze Endekennzeichen
660	IP	ermittelt Bildschirmadresse aus Position in BC	1000	STEBL LD HL,(AKTPOS)	; Lade HL mit Akt.Pos.
670	HL	enthält die Adresse	1010	RAC A	; lösche Übertragungslag
680	B	enthält die Position des Bits (Mez. Schieben)	1020	SBC HL,BC	; HL=Akt.Pos. minus neue Pos.
690			1030	RET I	; wenn gleich Routine beenden
700	UPWARD CALL #229A	; Aufruf ROM-Routine	1040	LD I(NEPOS),BC	; Lade NEPOS mit neuer Position
710	LD B,A	; Lade B mit Wert von A	1050	LD A,#44	
720	RET		1060	LD I(ANDEB),A	; setze Änderkennzeichen
730			1070	RET	
740			1080		
750	STEBL-Routine		1090		
760	Abfrage der Coursortasten und setzen der Position		1100	Bereich der Programmvariablen	
770			1110		
780	STEBL LD BC,#7FE	; Lade Portadresse der Tasten 1-5	1120	RETURN DEFB #00	; Endekennzeichen
790	IN A,(C)	; Lade den Portwert in A	1130	ANDEB DEFB #00	; Änderkennzeichen
800	LD B,A	; Lade B mit Wert von A	1140	NEPOS DEFB #0000	; neue PRINT-Position
810	LD BC,#EFFF	; Lade Portadresse der Tasten 6-0	1150	AKTPOS DEFB #0050	; aktuelle PRINT-Position
820	IN A,(C)	; Lade den Portwert in A	1160	FIGUR DEFB 111101001	; Beginn Figurbytes
830	LD BC,(AKTPOS)	; lade BC mit akt. Pos.	1170	DEFB 110001001	
840	BIT 2,A	; teste 2. A-Bit ("0")	1180	DEFB 110001001	
850	JR NZ,STEBL1	; nicht gedrückt, also kein ...	1190	DEFB 110001010	
860	INC C	; ... erhöhen I-Pos. um 1	1200	DEFB 110001111	
870	STEBL BIT 3,A	; Abfrage Taste "7"	1210	DEFB 110001001	
880	JR NZ,STEBL2		1220	DEFB 110001001	
890	INC B		1230	DEFB 111101001	
900	STEBL BIT 4,A	; Abfrage Taste "6"	1240	DEFB #FF	; Ende Figurbytes
910	JR NZ,STEBL3				

Assemblertips für den Spectrum

Teil 11: Treibersoftware für Epson-kompatible Drucker

Der Spectrum mit Interface 1, Discovery oder Wafadrive erlaubt die Verwendung "richtiger" Drucker mit serieller bzw. Centronics-Schnittstelle. Leider wird diese Möglichkeit von der Software in den Interface-ROMs nicht optimal unterstützt. Wir lernen in dieser Folge ein Utility kennen, das folgende Eigenschaften aufweist:

- verhält sich weitgehend wie ein Textkanal
- frei wählbarer rechter Rand (starre Einstellung auf 80 Zeichen je Zeile fällt weg)
- TAB-Kommando wird ausgeführt
- Blockgrafik wird gedruckt
- Basic-Schlüsselwörter können hervorgehoben und Grafik-Buchstaben unterstrichen werden

In der hier wiedergegebenen Form ist die Routine für Epson-kompatible Drucker, speziell für einen DMP-1101 ausgelegt. Ich hoffe, daß das Utility in dieser Version für möglichst viele Drucker geeignet ist. Gegebenenfalls sind kleine Änderungen erforderlich, beispielsweise zur Ausgabe der Zeichen von Code 127 bis 143 (Copyright-Symbol und Blockgrafik). Da nicht alle Blockgrafikzeichen des Spectrum im Epson-Zeichensatz verfügbar sind, werden manche Character durch Übereinanderdrucken von zwei oder drei Zeichen erzeugt. Die Codierung dieser Zeichenfolge ist in der Liste ab Label GRLST zu finden. Man kann hier ohne weiteres Änderungen vornehmen.

Die Zeilenbreite ist von 1 bis 254 frei wählbar. Gibt man 0 an, so führt die Routine keine Zeilenlängenkontrolle aus, und die Position des rechten Randes läßt sich durch Übermitteln von Steuer-codes an den Drucker festlegen. Die Verwendung von TAB ist in diesem Fall allerdings problematisch.

Das Utility wird über Strom #6 angesprochen (LIST #6 und PRINT #6; ...). Wer die Ausgabe lieber direkt auf Strom #3, den LPRINT-Strom, legen möchte, ändert im Programmteil INI die Adresse 23586 in 23580 (tritt an zwei Stellen auf), in den Programmteilen PRINT und PRNT1 LD A,6 in LD A,3 und streicht das OPEN #3; "L"-Kommando im Basic-Programm. Der FOR-

MAT-Befehl ist selbstverständlich nur bei Benutzung eines Interfaces mit RS-232-Schnittstelle erforderlich und muß an die eingestellte Baudrate angepaßt werden.

Das Maschinenprogramm ähnelt in Ablauf und Aufgabenstellung dem 32*42-Print-Programm aus den Assembler-tips in CK 8-9/86: Einbinden einer Zeichenausgaberroutine in das Strom- und Kanalsystem des Spectrum. Sieht man sich das Assemblerlisting näher an, entdeckt man - einmal abgesehen von der eigentlichen Character-Ausgabe, die bei Grafikbildschirm und Drucker natürlich grundverschieden ist - interessante Unterschiede. Was hat es mit den EXX-Statements auf sich, und weshalb werden

hier die benutzten Register gleichzeitig auf den Stack gerettet, während das bei der Bildschirmausgabe offensichtlich nicht notwendig war? Zur Beantwortung dieser Fragen führen wir uns noch einmal vor Augen, wie die Zeichenausgabe im Spectrum-Betriebssystem vonstatten geht. Der auszugebende Character (das kann ein ASCII-Zeichen, ein Steuercode, ein Basio-Token oder ein Grafikzeichen sein) wird in den Akku gebracht und durch RST \$10 über den gerade geöffneten Strom an die Ausgaberroutine eines Kanals weitergeleitet. Der RST-\$10-Mechanismus schaltet durch EXX auf das zweite Registerset der Z80-CPU um, ehe die Ausgaberroutine auferufen wird, und anschließend auf das erste Registerset zurück. Diese Methode stellt sicher, daß der Inhalt der

Register B, C, D, E, H, L im ersten Registeratz durch die Zeichenausgabe nicht verändert wird und somit nach Ausführung von RST \$10 weiterhin zur Verfügung steht.

So weit, so gut. Nun ruft unsere Routine jedoch ihrerseits durch RST \$10 eine Ausgaberroutine auf, nämlich die von Strom #4. Ohne weitere Maßnahmen hätte das zur Folge, daß das neuerliche EXX auf den ersten Registeratz zurückschaltet und die Strom-#4-Zeichenausgabe dessen Inhalt zerstört. Daher wechselt unsere Routine von sich aus auf das erste Registeratz und rettet seinen Inhalt auf den Stack, bevor weitere RST-\$10-Instruktionen folgen. Vor der Rückkehr ins Spectrum-Betriebssystem muß die alte Registersituation wiederhergestellt wer-

den. Außerdem ist es zwingend notwendig, wieder Strom #6 zu öffnen, d.h. zum aktuellen I/O-Strom zu machen, damit auch die noch folgenden Character über unsere Routine und nicht etwa über Strom #4 ausgegeben werden.

Diesmal wird zu Abwechslung ein etwas anderes Basic-Programm zur Erzeugung des Maschinencodes verwendet als in den bisherigen Folgen. Jede einzelne DATA-Zeile enthält als letzte Byte-Information eine Prüfsumme über ihren eigenen Inhalt. Tipplehler lassen sich so genauer lokalisieren und schneller korrigieren.

Bei dieser Gelegenheit möchte ich noch ein paar Worte zu den in dieser Serie immer wieder auftauchenden Basic-Programmen zum Generieren der Maschinprogramme sa-

gen. Im Laufe der Zeit erhielten wir Briefe und Anrufe von CK-Lesern, die Probleme mit diesen Programmen hatten. Ihre Schwierigkeiten ließen sich immer auf Tipplehler zurückführen. Man muß sich darüber im klaren sein, daß ein Fingebefehl nicht unbedingt zu der Meldung "Fehler in AS!" führt. Je nach Art der Fälschbeingabe kann das Basic-Programm z.B. auch mit der Fehlermeldung "Number too big" oder "Integer out of range" abrechnen. Wichtig ist auch die Kleinschreibung in den LET-A\$- bzw. DATA-Zeilen. Wer unbedingt Großbuchstaben eintippen möchte, muß in der DEF-FN-A-Anweisung die 39 durch eine 7 ersetzen (Großbuchstaben beginnen mit Code 65 = 48 + 10 + 7, Kleinschubstaben mit 97 = 48 + 10 + 39).

Michael Schramm

Epson-Druckertreiber

1 REM Treibersoftware fuer
Epson-Drucker am
Spectrum

Michael Schramm
Computer Kontakt 1986

```
10 RESTORE : CLEAR 64939
20 DEF FN A(X$)=CODE X$-48-39*
(X$)*9*
25 DEF FN B(X$)=16*FN A(X$(1))
+FN A(X$(2))
30 PRINT "Treibersoftware-Masc
hincode wird erzeugt - dauer
t nur einen Augenblick."
40 LET A=64940
50 FOR Z=1000 TO 1030 STEP 10
60 PRINT "Zeile ";Z;": ";
70 READ A$: LET S=0
80 IF LEN A$(4 OR LEN A$/2)>IN
T (LEN A$/2) THEN GO TO 200
90 FOR I=1 TO LEN A$-2 STEP 2
100 LET X=FN B(A$(I TO I+1))
110 LET S=S+X
120 POKE A,X: LET A=A+1
130 NEXT I
140 IF FN B(A$(I TO I))>S-256*I
NT (S/256) THEN PRINT "fehlerhaf
t!": STOP
150 PRINT "ok"
160 NEXT Z
```

```
170 PRINT "Maschinencode ist in
Ordnung."
180 INPUT "Zeilenbreite: ";Z
190 INPUT "Basic-Schlüsselwoer
ter fett- drucken (J/N)? ";A$
200 LET F=(A$="J" OR A$="j")
210 INPUT "Graphik-Buchstaben u
nterstreichen (J/N)? ";A$
220 LET U=(A$="J" OR A$="j")
230 CLOSE #3: CLOSE #4: CLOSE #
6: OPEN #3;"t": OPEN #4;"b"
240 FORMAT "b";4800
250 PRINT USR 64940,Z+512*(F+2*
U)
260 PRINT
270 PRINT "Ausgabe ueber Strom
#6"
280 STOP
1000 DATA "cd811ccda22dc52a225c7
cb52802cf172a535ce52b010b00e5cd5
516e1103fe7323723336c4233615233
64223010b00712370232323712370e1e
45b4f5ca7ed5222225cd12100fe73237
02372ed7b3d5cc3761b200000f5cd541
fd27b1b3e04cd0116d9e360"
1010 DATA "d65354cd62fe3e06cd011
6c1d1e1d9c92b7ec610e6f01805cb86c
bfe2b2b4ecb014e232006b930039110f
a574604b8dc59fe7a962806473e20d71
0fb72c92b7ea7c8353e00d7c92b36003
e0dd73e0ad7c92102fe7ea7cbbef81f7
a38bbfe06d828afe00d897"
1020 DATA "28d7fe0dd828dbfe10d8c
bfe0e16d8cbbecbc6fe10d8cb86fe20d
8d6a53823cb4ef528063e1bd73e45d73
```

```
e06cd0116f1f5cd100fc1c03e04cd011
63e1bd73e46d7c92b342b7e23a720069
635dc59fe347a23d67f300f7afdc018
6fe202004fdeb01c6d7c932*
1030 DATA *fe110192115ff470847e2
3a720fb10f910067ea7c83e08d77e23d
710f5cb56f520093e1bd73e2dd73e01d
77ad64fd7f1c03e1bd73e2dd7afd7c90
```

```
0fe002000f100f000f0f100f300f1f30
0f0f300f0f1f300f200f400ab00abf10
0a300a3f100abf300a700c7*
9000 SAVE "Druck.BAS" LINE 1
9010 STOP
9500 SAVE "Druck.CODE" CODE 6494
0,404
9510 STOP
```

Assembler-Listing

ORG 6490		LD SP, (ERRSP)	LD A, 00	list.
		JP STHRT	RET *10	
;	#### Treibersoftware für Epson-Drucker			
ERRSP = 23613		RAND NOP		
CHANS = 23611		XPOS NOP		ENTER DEC HL
PROB = 23630		PFLAG NOP		ENTRI LD (HL),0
				LD A, CR
				RET *10
PROB = 01655				RET M
STRT = 01754				ORA
REF_1 = 01070				LD A, 0
OPEN = 01601				JR C, TABUL
TKPR = 00C10				CP 6
STACKIN = 01001				RET C
UNSTACK = 02042				JR Z, PRCON
STHRT = 01076				CP 0
				RET C
				JR Z, PRWS
				CP 10
				RET C
				JR Z, ENTER
				CP 16
				RET C
				SET 7, (HL)
				CP 22
				RET C
				RES 7, (HL)
				SET 0, (HL)
				CP 24
				RET C
				RES 0, (HL)
				CP *
				RET 0
				ROR 165
				JR C, ORAPH
				BIT 1, (HL)
				PUSH AF
				JR Z, NBOLD
				LD A, ESC
				RET *10
				LD A, "E"
				RET *10
				NBOLD LD A, 0
				CALL OPEN
				POP AF
				PUSH AF
				CALL TKPR
				POP AF
				RET Z
				LD A, 4
				CALL OPEN
				LD A, ESC
				RET *10
				LD A, "F"
				RET *10
				RET
				GRAPH DEC HL
				INC HL
				DEC HL
				LD A, (HL)
				INC HL
				AND A

<p> JR Z,ORPHI (beim Zellenvorsch. SUB (HL) (Sonst rechten Rand DEC (HL) überblicksichtigen. CALL C,ENTRI INC (HL) ORPHI LD A,3 (Character -> Akku. INC HL (HL auf PFLAG. SUB 127 JR NC,RLKDR (Invertierte ASCII-Char. LD A,3 (122 bis 126) werden RES #,(17+1) (hier ausgegeben. CP * * (BIT # in FLAG# ist JR NZ,NBLNK (bei Vorliegen eines SET #,(17+1) (Blanks zu setzen, NBLNK RST #18 (sonst rückzusetzen. RET RLKDR CP 17 (Hier die Zeichen JR NC,NBLDR (127 bis 143 behand. LD HL,ORLST LD B,A (HL wird auf die zum INC B (Graphikzeichen ge- ORCH LD A,(HL) (hörnde Char.folge INC HL (in der Liste ORLST AND A (geprüft. JR NZ,SRCH (Bei ersten Char. DJNZ SRCH (der Folge sofort </p>	<p> JR NZDB ORLST LD A,(HL) AND A RET Z LD A,B5 RST #18 NZDB LD A,(HL) INC HL RST #18 JR ORLST NBLDR BIT 2,(HL) PUSH AF JR Z,NOUND LD A,ESC RST #18 LD A,-* RST #18 LD A,1 RST #18 NOUND LD A,0 SUB 79 RST #18 POP AF RET Z LD A,ESC </p>	<p> Drücken, die weiter- (Character in den (Akku. (bei Null fertig. (die Character der (Reihe nach überein- (anderdrücken. (die Ausgabe der (Graphikbuchstaben. (falls Bit 2 in (PFLAG gesetzt ist, (Steuercode für (Unterstreich- (ausgeben. (den Graphikbuchsta- (ben in einen norma- (len Buchstaben (umzuwandeln und (drücken. </p>	<p> RST #18 LD A,-* RST #18 LD A,0 RST #18 ORLST DEFB 0 DEFB #FC,0 DEFB #22,0 DEFB #F1,0 DEFB #F2,0 DEFB #F3,0 DEFB #F4,0 DEFB #F5,0 DEFB #F6,0 DEFB #F7,0 DEFB #F8,0 DEFB #F9,0 DEFB #FA,0 DEFB #FB,0 DEFB #FC,0 DEFB #FD,0 DEFB #FE,0 DEFB #FF,0 END </p>	<p> (Erforderlichenfalls (die automatische (Unterstreich- (wieder ausschalten. (Liste der Charact., (die statt der Zeich- (en mit Codes von (127 bis 143 ausge- (geben werden. (Jede Charact.folge (zu einem Zeichen (wird durch ein (Nullbyte abge- (schlossen. (Jede Charact.folge (schließt ein (Nullbyte ab. </p>
--	--	--	--	--

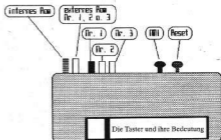
ISO-FACE 1,0 – die neue EPROM-Steckkarte für den Spectrum

In der Januar-Ausgabe der CK hatten wir das ISO-ROM von Individual Software vorgestellt. Aus dem gleichen Haas gibt es jetzt das ISO-FACE 1.0. Dieses Gerät bietet dem Anwender die Möglichkeit, bis zu drei zusätzliche ROMs einzusetzen, ohne einen Eingriff im Spectrum vorzunehmen; es handelt sich also um eine EPROM-Steckkarte. Optisch präsentiert sich der Zusatz eher unauffällig. Eine Black Box in der Art eines Joystick- oder Druckerinterfaces wird als erste Erweiterung an den Spectrum gesteckt. Die Anschlüsse sind natürlich durchgeführt, so daß weitere Peripheriegeräte angeschlossen werden können. Oben befinden sich einige Schalter:

- der Umschalter für internes/externes ROM
- ein Schiebeshalter für die Wahl eines der drei möglichen EPROMs

- ein NMI (nichtmaskierbarer Interrupt für Warmstart)-Taster
- ein Reset-Taster (für Kaltstart)

Der Spectrum-Benutzer hat jetzt erstmals die Gelegenheit, sein Original-ROM zu benutzen, einen Eingriff in den Computer zu vermeiden (ein evtl. Garantieverlust wird verhindert) und trotzdem mit anderen ROMs zu arbeiten. Dabei spielt ca keine Rolle, ob eigene EPROMs erstellt werden oder z.B. die ISO-ROMs zum Einsatz kommen. Verwendet werden können die EPROM-Typen 2716 bis 27128. Durch die Installation prellfreier Taster können die Möglichkeiten der ISO-ROMs (Monitor- oder Back-Up-Routine) direkt genutzt werden; zusätzliche Lötarbeiten am Spectrum entfallen. Unter dem Original-ROM steht dann auch endlich ein RE-



SET-Taster wie beim Spectrum+ zur Verfügung. Mit einem voll bestückten ISO-Face hat man die Wahl zwischen vier ROMs, was für die meisten Anwender ausreichen dürfte. Das Umschalten zwischen internem und externem ROM erfolgt abstrzfrei, lediglich die Wahl eines EPROMs muß in der Stellung "internes EPROM" erfolgen.

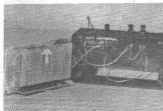
Zum Test waren dem ISO-Face das Monitor- und das Back-Up-ROM mitgegeben. Die Umschaltung und die Arbeit mit dem Original-ROM und externen EPROMs funktioniert einwandfrei. Bei dieser Gelegenheit weist der Hersteller darauf hin, daß die ISO-ROMs verbessert wurden. Das Back-Up-ROM ist jetzt speziell auch auf Beta-Disk-Benutzer zugeschnitten. Außerdem wurde eine Routine implementiert,

die den gesamten RAM-Speicherinhalt komprimiert. Das Monitor-ROM wurde ebenfalls verbessert. Das Adressenformat für den Monitor ist jetzt HI-LO (wie bei der Eingabe von Hexadezimalzahlen in Assembler). Außerdem kann beim Ausdrucken eine beliebige Zeichenbreite gewählt werden.

Wer mit seinem eigenen ROM nicht mehr zufrieden ist und einen Eingriff in seinen Spectrum bisher gescheut hat, ist mit dem ISO-Face (ebenso mit den ISO-ROMs) gut bedient. Der Preis von ca. 80,- DM erscheint angemessen.

Weitere Informationen: Individual Software V. Marohn Schürferstr. 41 4600 Dortmund 30

Rolf Knorr



Einfaches, aber solides Innenleben

Eines der besten Textverarbeitungsprogramme für den Spectrum

Es gibt mittlerweile eine ganze Reihe von Textverarbeitungsprogrammen für den ZX Spectrum. Das bekannteste und beliebteste ist wohl "Tasword Two", das wohl fast jeder Spectrum-Benutzer in seiner Bibliothek hat. Die englische Firma SAGA Systems hat sich dadurch aber nicht abschrecken lassen und vor kurzer Zeit ein neues Programm auf den Markt gebracht, das die Bezeichnung "The Last Word" trägt. Um es vorwegzunehmen: Wäre dieses Programm zeitgleich mit "Tasword" auf den Markt gekommen, hätte ich ihm einen großen Erfolg prophezeit. Heute



bin ich mir nicht mehr sicher, ob es viele User gibt, die von einem bewährten Produkt so einfach umsteigen wollen. Trotzdem, "The Last Word" ist hervorragend.

Das Programm besteht in der Hauptsache aus einem rund 14,5 KByte langen Maschinencodefile, der von Cassette geladen wird. Als Textspeicher verbleiben 22.000 Byte. Neben dem Hauptprogramm befinden sich auf der Cassette noch einige Files, die der Anpassung an verschiedene Massenspeicher und Drucker-Interfaces dienen. Berücksichtigt wurden folgende Systeme:

- Cassettenrecorder/Microdrive (bereits in "TLW" enthalten)
- Kempston KDOS

- Beta Disc
- Opas Discovery
- Gordon (mir leider unbekannt)
- Wafadrive
- Kempston-S-Interface
- Sinclair-Interface I RS 232-Port

Ich habe das Programm mit einem Epson-kompatiblen Drucker und dem Kempston-E-Interface getestet. Für diese Zusammenstellung werden die o.g. Routinen nicht benötigt; man kann also nach Laden des Hauptprogramms direkt arbeiten.

Nun zu den Möglichkeiten, die "The Last Word" dem Anwender bietet. Vorab sei noch gesagt, daß ich unmöglich alle Punkte ausführlich beschreiben kann, da sonst der Rahmen dieses Tests gesprengt würde. Ich hoffe aber, daß Interessenten genügend Informationen erhalten.

Beginnen möchte ich mit dem aktuellen Arbeits-Screen, der immer auf dem Bildschirm bleibt. Die unteren 20 Zeilen sind für den Text reserviert. Am oberen Rand befinden sich drei Statuszeilen, die permanent folgende Daten anzeigen:

Links: Zeilennummer / Spaltennummer / freier Speicherplatz

Mitte: Gewählte Zeichen pro Zeile / Groß- und Kleinschreibung / linker und rechter Rand / Tabulatorstop / Modus. In diesem Bereich werden auch besondere Bearbeitungshinweise gegeben.

Rechts: Anzeige für Word Wrap an bzw. aus / Blocksatz an bzw. aus / Überschreiben / Einfügen

Die Programmierer haben versucht, ihrem Produkt größtmöglichen Komfort und leichte Bedienbarkeit mitzugeben. Aus diesem Grund wurde auch darauf verzichtet, mit Menüs zu arbeiten; alle Funktionen werden über Tastenkombinationen aufgerufen. Notwendige Einblendungen und Hinweise erfolgen im oberen Statusfeld.

Der Anwender kann also ständig seinen Text im Auge behalten. Eine Ausnahme bildet hier eine Helppage, die auf Wunsch über den Text gescrellt werden kann.

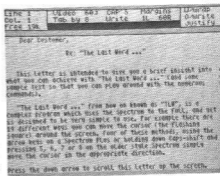
Wer sich ein wenig mit Textverarbeitungsprogrammen auskennt, wird die schriftliche Anleitung (leider zur Zeit nur in Englisch erhältlich) kaum benötigen, aber auch ein totaler Neuling kann sehr schnell die ersten Texte erstellen. Die Zeichenanstellung auf dem Bildschirm kann der vorhandenen Hardware angepaßt werden. Auf Tastendruck stehen dem Anwender 40, 60 oder 80 Zeichen zur Verfügung. 80 Zeichen waren aber selbst auf einem Monitor nicht gerade angenehm für die Augen. Optimal ist die 60-Zeichen-Variante, die sich auch als Briefformat eignet. Auch die BORDER-, PAPER- und INK-Farben können vom Programm aus geändert werden.

Nun zu den Bearbeitungsbeehlen, die direkt über die Tastatur aufgerufen werden (in der Regel mit der SymbolShift-Taste).

- Zeichen nach rechts/links
- Zeile nach oben/unten (über Cursor-Tasten)
- Tabulatorschritte

- Wort oder Zeile nach links/rechts
- Sprung an Textanfang oder -ende
- Sprung zu einer eingezehenden Zeilennummer
- Textscroll nach oben/unten
- Löschen eines Zeichens, Wortes, Textes oder einer Zeile
- Umschaltung Text einfügen/überschreiben
- Text nach links oder rechts schieben
- Text zentrieren und einiges mehr

Damit sind alle Optionen, die man von anderen Programmen her kennt, vorhanden. Auch weitere Standardfunktionen, wie z.B. das Einstellen der Ränder oder Setzen von Tabulatorstops, sind natürlich möglich. Das Word Wrapping, also der Umbruch eines Wortes, das den Rand überschreiben würde, funktioniert tadellos, ebenso der dabei ausgeführte Zeilenausgleich (=Blocksatz: die in einer Zeile vorhandenen Wörter werden auf die ganze Zeilenbreite verteilt). Beide Funktionen sind abschaltbar. Steuerodes, die dem Drucker z.B. eine andere Schriftart mitteilen, werden direkt in den Text geschrieben. Die dabei auftauchenden Grafiksymbole können ebenso wie die Zeilen-



Viel Information auf einem dennoch aufgeräumten Bildschirm

end- und Word-Wrap-Marker ausgeblendet werden, so daß auf dem Bildschirm nur noch der reine Text verbleibt.

Jetzt zu einigen Besonderheiten, die andere Programme dieser Art nicht bieten. Mit dem Befehl WORD COUNT kann z. B. abgefragt werden, wie viele Wörter und Zeichen man bereits geschrieben hat und in welcher Position der Cursor momentan steht. Interessanter ist aber CALCULATOR. Damit steht ein kompletter Taschenrechner zur Verfügung, der allerdings nur im oberen Statusfeld angezeigt wird. Neben den Grundrechnarten kann eine Vielzahl weiterer Operationen durchgeführt werden. Dazu dienen die Befehle SIN, COS, TAN, ASN, ACS, ATN, EXP,

SQR, INT, LN, VAL, SGN, ABS, PI, >, <, <>, >=, OR, AND und NOT. Das Ergebnis einer Rechnung kann nun entweder gelöscht, in einen der neun Speicher übernommen oder direkt in den Text eingesetzt werden. Die Zahl erscheint immer an der aktuellen Cursorposition. Wer Gefahr läuft, zu lange zu arbeiten, auch die interne Uhr abrufen oder die Zeit für einen akustischen Alarm einstellen.

Die Printer-Option ist ebenfalls vorbildlich. Was gedruckt werden soll, wird über die Zeilennummern eingegeben. Wird eine Kopie benötigt, kann dies auch vermerkt werden (bis zu 254 Kopien sind möglich); auch der Zeilenabstand kann einge-

stellt werden. Ähnlich arbeitet auch die SAVE-LOAD-Option. Neben der Wahl zwischen Microdrive und Cassette (oder dem eigenen Speichermedium) kann man z. B. angeben, welche Zeilennummern abgespeichert werden sollen.

"The Last Word" gehört mit zu den besten Textverarbeitungsprogrammen für den Spectrum. Die einfache Handhabung und die geringe Anfälligkeit gegen Bedienungsfehler, die große Leistungsfähigkeit und der hohe Komfort haben mich voll überzeugt. Warum ist dieses Programm nicht früher auf den Markt gekommen?

System: Spectrum 48 K
Hersteller: SAGA Systems
Ref: Kassic

Graf Eddi – ein Grafikeditor besonderer Art

Um eines von vornherein klarzustellen, "Graf Eddi" ist kein Programm zum Malen von Bildern. Man kann damit aber vorhandene Grafiken bearbeiten, so z. B. spiegeln, drehen, Teile löschen, invertieren, herausstreifen, verschieben, kopieren, Texte einblenden, Farben ändern, verschiedene Grafiken übereinanderlegen und das Bild den Farben gemäß schattiert auf einem Sinclair-kompatiblen Drucker als 24-Zeilen-Copy oder auf einem Normalpapierdrucker ausgeben.

"Graf Eddi" verwaltet Bildschirm Speicher, Workingpage, Sicherheitskopie und zwei Alternativseiten. Ferner stehen 2 Zeichenseiten bereit. Farb-Informationen werden getrennt vom übrigen Bild verwaltet, so daß die Möglichkeit besteht, die Farben des Bildes durch ein Schachbrettmuster zu ersetzen, um dann später die Originalfarben wieder zurückzuholen. Alle wesentlichen Teile des Programms sind in Maschinencode geschrieben; Geschwindigkeitsprobleme entstehen somit nicht.

"Graf Eddi" besteht aus 2 Listings. Listing 1 ist ein Hilfspro-

gramm, um den ca. 1500 Byte langen Maschinencode des Programms zu laden. Speichern Sie es nach dem Eintippen mit GOTO 9000 ab. Danach löschen Sie den Speicher und geben Listing 2, den Basic-Teil ein. Speichern Sie diesen mit GOTO 9880 ab. Nun laden Sie wiederum den MC-Lader und starten ihn mit RUN. Fehler in den DATA-Zeilen werden erkannt. Treten keine auf, fordert Sie das Ladeprogramm nach einiger Zeit dazu auf, Programm 2 zu laden. Kommen Sie dieser Aufforderung nach und speichern Sie das gesamte lauffähige Programm mit GOTO 9820 ab.

Die einzelnen Funktionen von "Graf Eddi" werden angewählt, indem man den Cursor auf den Schriftzug plaziert und Enter oder 0 drückt. Der kreuzförmige Cursor, der bei Pixel-Operationen eingeblendet wird, kann ebenfalls mit den Cursor-Tasten und 0 pixelweise bewegt werden (in Verbindung mit Symbol Shift um 8 Pixel). Mit der Taste 1 kommt man von diesem Cursor wieder ins Menü. Zu bearbeitende Bildausschnitte werden – so nicht anders erwähnt – durch die linke

untere und die rechte obere Ecke mit Hilfe des kreuzförmigen Cursors und der Taste 0 markiert. Grundsätzlich können sich Ursprungs- und Zielbildschirmteile ohne Beeinträchtigung überlagern.

Die Funktionen haben folgende Bedeutungen:

CHAR 1: Alternativzeichensatz 1

CHAR 2: Alternativzeichensatz 2

NORMCHAR: Sinclair-Zeichensatz

ROTATE: Der quadratische Bildschirmteil, markiert durch die linke untere und linke obere Ecke, wird gedreht.

MIRROR H: Der markierte Bildschirmteil wird an die Stelle, deren rechte untere Ecke festgelegt wurde, horizontal gespiegelt.

MIRROR V: Wie MIRROR H, es wird aber vertikal gespiegelt.

COPY: Der markierte Bildschirmteil wird an die Stelle kopiert, deren linke untere Ecke festgelegt wurde.

MOVE: Wie COPY, der Ursprungsbildteil wird aber gelöscht.

Reparatur-Schnelldienst/Ersatzteile für Spectrum! Rufen Sie uns an! Spectrum-Eprommer



- * 2716/2732/2732A/2764/27128
- * Intelligente Programmierung
- * Menü-gesteuert
- * zus. Centronics-Schnittstelle
- * Keine zus. Hardware erf.
- * Preis 248,- DM inkl. Software für EPROMMER und Drucker
- * Spez. Sockel zum Austausch Spectrum-ROM – 27128 24,80 DM
- * EPROM-Lötlösgerät 117,00 DM
- * Druckerkabel Centronics 37,00 DM

Alle Preise zuzügl. Versandkosten, Info anfordern.

ZX-Spectrum
G.M. Meyer
Rahserstraße 52
4083 Viersen 1
☎ 0 21 62 / 2 29 64

Jupitersoft GdbR

ZX Spectrum-, ZX81-,
C16-Software
ab 7,90 DM

Adventures, Action,
Anwender, Grafik
und Dateiprogramme –
C16 auch auf Disk.

**Viel Software stark
im Preis reduziert.**

Kostenloser Katalog bei

**Bernd Denk
Jupitersoft GdbR**

Fühlingsstr. 12
8831 Weiboldshausen
☎ (09141) 22 49 ab 18 Uhr



ATARI ST Tagestiefstpreis
Spectrum ab 248,-
Sinclair QL ab 448,-
Tastatur i. QL ab 198,-
Open Discovery ab 389,-
Seikusha SP 1000 ab 699,-
Computer, Drucker + Zubehör in unserer Gratisinfo
Unseren Kunden
ein frohes Weihnachtsfest
Ursula & Joachim Rudolfph

DELETE: Der markierte Bildschirmteil wird gelöscht.

FETCH: Der markierte Bildschirmteil wird in die Alternativseite 1 eingeblendet (ohne Farben).

INVERSE: Der markierte Bildschirmteil wird invertiert.

ZOOM: Ein durch die linke obere Ecke markierter, 32x22 Pixel großer Bildschirmteil wird 64mal vergrößert und kann bearbeitet werden. Der Cursor hat dabei eine OVER-Funktion. Zurückgebendert wird auf Originalgröße in das Gesamtbild, indem man den Cursor auf den Schriftzug STORE bringt. Ohne Veränderung erreicht man das Menü durch EXIT.

ATTR: Ruft folgende Untermenüpunkte auf:

1. **ATTRIBUTE:** Blendet den externen Attributspeicher in den Farb Speicher der Workingpage.
2. **CHISS:** Blendet ein

Schachbrettmuster ein.

3. **SET ATTR:** Nach Eingabe der Farbwerte können mit dem Cursor Attribute verändert werden.

4. **EXIT:** Bringt das Hauptmenü zurück.

RECALL: Holt die Sicherheitskopie in die Workingpage.

BACKUP: Kopiert die Workingpage in die Sicherheitskopie.

LD 1--5: Kopiert die Workingpage in die Alternativseite 1.

PRINT: Fügt Texte in das Bild der Workingpage ein. Dabei kann es in den untersten 2 Zeilen zu Farbverfälschungen kommen, die aber durch Zurückblenden der Farben mittels ATTR/ATTRIBUTE wieder rückgängig gemacht werden können.

Mit CAPS SHIFT und Taste 9 schaltet man auf den Grafikmodus um, in dem deutsche Sonderzeichen (Tasten A-G) verfügbar sind.

EXCHAR: Vertauscht Alternativzeichensätze miteinander.

OR S--1: Blendet Alternativseite 1 in die Workingpage ein.

OR S--2: Blendet Alternativseite 2 in die Workingpage ein.

SCHATT: Das Bild in der Workingpage wird den Farben des ext. Attributspeichers entsprechend für die Druckerausgabe schattiert.

EX S--1: Vertauscht Workingpage und Alternativseite 1.

EX S--2: Vertauscht Workingpage und Alternativseite 2.
SINCLAIR: Gibt eine 24-Zeilen-Copy (!) auf einem Sinclair-kompatiblen Drucker aus.

EIGEN: Ruft eine einsetzbare Hardcopy-Routine auf. Der Einsprungbefehl steht dann in Zeile 3500. Die Routine kann im Speicher bei 6500 liegen und darf 300 Byte lang sein.

SAVE: Speichert nach Eingabe des Namens (nur Enter führt

zum Menü) die Workingpage ab.

MSAVE: Speichert entsprechend SAVE auf Microdrive ab.

CAT: Microdrive-Katalog

LOAD: Lädt nach Eingabe des Namens ein Bild; die Länge muß 6912 Byte (Screen\$) betragen.

MLOAD: Lädt ein Bild entsprechend LOAD von Microdrive.

ERASE: Löscht nach Eingabe des Namens einen Code dieses Namens auf Microdrive.

LOAD CH: Lädt nach Eingabe des Namens einen Zeichensatz; die Länge muß 768 Byte betragen.

MLOAD CH: Lädt einen Zeichensatz entsprechend LOAD CH von Microdrive. Bei den Microdrive-Befehlen ist immer Drive 1 angesprochen.

Stefan Arbes

MC-Generator

```

10 REM XXXXXXXXXXXXXXXXXXXXXXXX
20 REM XX GRAF EDDI
30 REM XX MC-Loader
40 REM XX (C) 1985 by STARCOMP
50 REM XX Tel.: 06021/97302
60 REM XXXXXXXXXXXXXXXXXXXXXXXX
70
80
100 CLEAR 32767
110 LET adr=62750
120 DEF FN f(a#)=16X(CODE a#(b)
-48-39X(a#(b)*9**)+CODE a#(b+1)
-48-39X(a#(b+1)*9**))
130 FOR a=1000 TO 1290 STEP 10
140 PRINT AT 10,0;"Bearbeite Ze
ile "ja
150 LET sum=0
160 READ a#,x
170 FOR b=1 TO 100 STEP 2
180 LET w=FN f(a#)
190 LET sum=sum+w
200 POKE adr,w
210 LET adr=adr+1
220 NEXT b
230 IF x(<)sum THEN PRINT "Fehl
er in Zeile "ja: STOP
240 NEXT a
245 GO SUB 9050

```

250 CLS : PRINT AT 4,0;"Alle Da
ten eingelezen."

260 PRINT ""*Bitte loeschen Sie
dieses Programm nun mit NEW,
und laden Sieden BASIC-Teil von
GRAFIK-EDDI ein."

270 STOP

1000 DATA "210000ed4bfcffaf328bf
53e01320af5cd12f77676cdbc5cd12f
776763ef6bc301c3ef6bd30283e09bc3
01b3e09bd30273e",5923

1005

1010 DATA "10bcc03e64bc20d601fff
fc97884febfb30cc4718c97884fec030c
310f57905fef730094f10b8798530b41
0f73209f579fe09",6043

1015

1020 DATA "3a09f530ec10a5000000c
5e53ebf903020c0b02247043efe0f10f
d473a0af5f60020147ea0f53a0bf5f60
02007f1a02f77e1",5702

1025

1030 DATA "c1c9f110f97e10f4219b9
bc53e7fdbfee00220060000ef710030
1ff01cde0f5c13ef7dbfee012002266
4c93eefdbfef5e6",6035

1035

1040 DATA "04200160f1f5e00020016
0f1f5e010200161f1e00120032610c93
ef7dbfee010200169c921009b1003210
0b6010010110040",4521

1045

1050 DATA "1ab61213230b78b120f6c
3baf60100032100ec1100ef1a3240f67
e123a40f67723130b78b120efc900210
0d111000001001b",4288

1055

1060 DATA "edb0c92100001100d1010
01bedb0c921009b1100d101001b10092
100b61100d101001bcd2ef6cd41f6c30
6f7f306c0c3af0e",4873

1065

1070 DATA "ed4bfeffed5bfcff2afaf
f151daf320af53e9132c6f63afeff4f3
afaff64c5444dcdf7f6c12003af10023
e01320bf5cd0cf5",7398

1075

1080 DATA "0d2d7bbd20e604257cba2
0d02100401100d101001bedb0c900c5e
53ebfcdac22f53ac6f68467f147047e0
710fde601b7e1c1",5818

1090 DATA "c93e01320af5af320bf5e
d4bfcffed5bfaaff141c61cd0cf50c79b
b20ff64c047ab820f2c90002100d110
04001001bedb0c9",6119

1095

1100 DATA "c5cd0cf50c0c1605cd0cf
50c1520f9c1c5041605cd0cf5041520f
9c1c516050d0cd0cf50d1520f9c1c50
5051605cd0cf505",5429

1105

1110 DATA "1520f9c1c93e01320bf5a
f320af5c3eaf6000000cd49f7ed4bfcf
fed5bfaaff141c3e9132c6f6af320af52
afeff3afcff4f3a",7263

1115

1120 DATA "feff6fcdcf7f6c52003af1
0023e01320bf5444dcdf5c10c2c79b
b20e604247ab020d0c3baf6ed4bfeffe
d5bfaaff2afcff14",7206

1125

1130 DATA "1caf320af53e9132c6f63
afeff4f3afcff64c5444dcdf7f6c1200
3af10023e01320bf5cd0cf50d2c7bbd2
0e604247cba20d0",6404

1135

1140 DATA "c3baf621004011009b010
01bedb0c9cd6b0ded4bfcffed5bfaaff1
41caf320af53e9132c6f63afcff4fcd
7f62003af10023e",6472

1145

1150 DATA "01320bf5cd0cf50c79bb2
0eb0478ba20e23ec93222f6cd0af6cdd
df73ec33222f6c92100b3cd3bf02100c
ecd3bf0c9010003",6375

1155

1160 DATA "3638230b78b120f0c9cd0
2f9cd06f7010000c5790f0f0fe0e0a05
f79e618ee4057d52158ff06001a77231
410fad1c1c5d579",5823

1165

1170 DATA "0f0f0f4fe0e0a06f79e60
3ee50677ef5e607cddaf006001150ffd
d2160ff1aa6dd77001001323dd2310f5f1e
6300f0f0fcddaf0",6244

1175

1180 DATA "06001150ffdd2160ff1ae
effa6ddb600dd77001323dd2310f0d12
160ff06007e12142310fac1043e20b02
0020c06003e10b9",4971

1185

1190 DATA "c250f0c3baf6fe0120032
11af9fe022003212af9fe0320032122f
9fe0420032112f9fe0520032132f9fe0
62003213af9fe07",5005

1195

1200 DATA "20032142f9fe00c0214af
9c9aaaa55555aaaa5555aaff55f4aaff5
5ffaa75555eaaa7755eaaa55aa55aa55a
a55aa005500aa00",6605

1205

1210 DATA "5500aaaaaaaa555555550
0000000000000000fffffffffffffffff2
100e91100f2010003edb03e70003e602
100e906100e2077",4904

1215

1220 DATA "0057230d79f6007a20f50
500570e2070f6007a20eac306f72100f
21100e9010003edb0c306f7cd02f9cd0
6f7ed4b76facd25",5611

1225

1230 DATA "f578a1feff200e30bfed4
376facdacc22cddb0b10e52100501100f
2010003edb0c900000000ed4b76fa3e9
132c6f6af320af5",6100

1235

1240 DATA "cd25f5ed4376facd6b0d3
e02cd0116ed4b76fa78a1feff0c878d61
63004feff02d255793274fac62030245
fcdc7f620043e20",6261

1245

1250 DATA "10023e0fc5d5d7d1c10c7
9bb20eb3a74fa4f0578ba20e2ed4374f
ac901ffffc906bf0e0079320af532c6f
621d1facdc7f620",6933

1255

1260 DATA "043e0110023e00772379c
6084f20ed78d60847d61030e5cd06f7e
d4b76fa78d6165779c6205f21d1fa7e3
20bf5cd0cf50c23",5620

1265

1270 DATA "79bb20f33a74fa4f0578b
a20eac3baf6000000000ed4bfcff3afbf
f3c90676f32c1faed5bfaaff593e9132c
6f6a4320af5cd0c7",7219

1275

1280 DATA "f62003af10023e01320bf
5c5d5c1cd0cf5c1041c2520e73ac1fa6
73afdf473afcff5f0c152d20d6c3baf
600af320bf53c32",6276


```

1920 RANDOMIZE USR (uu+412)
1930 GO TO au
2000 GO SUB 8000
2010 RANDOMIZE USR (uu+715)
2020 GO TO au
2100 GO SUB 8000
2110 RANDOMIZE USR (uu+451)
2120 RANDOMIZE USR (uu+412)
2130 GO TO au
2200 LET xx=USR (uu+1183): IF xx
=65535 THEN GO TO 19
2215 PRINT #0;AT 1,0;"STORE
EXIT"
2220 LET bu=0: PRINT AT oo,ppi 0
VER 1; FLASH 1;"■": GO SUB 7000
2225 IF oo=22 THEN GO TO 2250
2230 IF bu THEN PRINT AT oo,ppi
OVER 1;"■"
2240 GO TO 2220
2250 LET oo=21
2260 IF pp)=28 THEN GO TO au
2270 IF pp<=3 THEN RANDOMIZE US
R (uu+1282): GO TO au
2280 GO TO 2220
2300 CLS : PRINT AT 5,10;"MMMMM
MMMK" ;AT 6,10;"QATTRIBUTE";AT 7,
10;"QBBBBBBBB";AT 8,10;"QCHESS
E";AT 9,10;"QBBBBBBBB"
2310 PRINT AT 10,10;"QSET ATTR"
;AT 11,10;"QBBBBBBBB";AT 12,10;
"QEXIT E";AT 13,10;"QNNNNNNNN
L"
2320 PRINT AT o,p; OVER 1; INVER
SE 1;" ": FOR z=1 TO 20:
NEXT z
2330 IF INKEY#="6" OR CODE INKEY
#=10 THEN PRINT AT o,p; INVERSE
1; OVER 1;" ": LET o=o+2
X(o<12): GO TO 2320
2340 IF INKEY#="7" OR CODE INKEY
#=11 THEN PRINT INVERSE 1; OVE
R 1;AT o,p;" ": LET o=o-2
X(o<6): GO TO 2320
2345 IF INKEY#="8" OR CODE INKEY
#=13 THEN GO TO 2360+oX2
2350 GO TO 2330
2372 RANDOMIZE USR (uu+1124): GO
TO au
2376 RANDOMIZE USR (uu+1076): GO
TO 5000
2380 GO SUB 6000
2382 RANDOMIZE USR (uu+1138): RA
NDOMIZE USR (uu+1171): GO TO 230
0
2384 GO TO 19
2400 RANDOMIZE USR (uu+303)
2410 RANDOMIZE USR (uu+408)
2420 RANDOMIZE USR (uu+1171)
2430 GO TO au
2500 RANDOMIZE USR (uu+291)
2510 GO TO au
2600 RANDOMIZE USR (uu+703)
2610 GO TO au
2700 INK 8: PAPER 8: LET st=0: L
ET gr=0
2710 LET fl=1: GO SUB 7200
2720 GO SUB 7100
2730 IF st=1 THEN RANDOMIZE USR
(uu+412): GO TO 19
2740 GO TO 2720
2800 RANDOMIZE USR (uu+263)
2810 GO TO au
2900 RANDOMIZE USR (uu+236)
2910 GO TO au
3000 RANDOMIZE USR (uu+241)
3010 GO TO au
3100 RANDOMIZE USR (uu+809)
3110 GO TO au
3200 RANDOMIZE USR (uu+315)
3210 RANDOMIZE USR (uu+1171)
3220 GO TO au
3300 RANDOMIZE USR (uu+326)
3310 RANDOMIZE USR (uu+1171)
3320 GO TO au
3400 RANDOMIZE USR (uu+344)
3410 GO TO au
3500 COPY : REM Hier kann der Ma
schinencodeaufruf Ihrer Hardcopy
routine stehen
3310 GO TO au
3600 GO SUB 5200
3610 CLS : SAVE n%CODE 53504,691
2
3620 GO TO au
3700 GO SUB 5200
3710 CLS : SAVE X"m";lin%CODE 53
504,6912
3720 GO TO au
3800 CLS : CAT 1: GO TO au
3900 GO SUB 5200
3910 CLS : LOAD n%CODE 32768,691
2
3920 GO TO 2400
4000 GO SUB 5200
4010 LOAD X"m";lin%CODE 32768,69
12
4020 GO TO 2400
4100 GO SUB 5200
4110 ERASE "m";lin#
4120 GO TO au
4200 GO SUB 5100
4210 CLS : LOAD n%CODE 60416,760
4220 GO TO 4320
4300 GO SUB 5100

```

```

4310 CLS : LOAD %*m*;!;n#CODE 60
416,768
4320 RANDOMIZE USR (uu+488)
5000 IF INKEY$(<)"* THEN GO TO a
u
5010 IF INKEY$==" THEN GO TO 50
10
5020 GO TO 19
5100 CLS : PRINT "Name des Chara
ktersatzes:"
5110 INPUT AT 20,20;AT 0,10; LIN
E n#
5120 IF LEN n#=0 THEN GO TO 19
5130 RETURN
5200 CLS : PRINT "Name des Bilde
s:"
5210 INPUT AT 20,20;AT 0,10; LIN
E n#
5220 IF LEN n#=0 THEN GO TO 19
5230 RETURN
6000 INPUT "Paper: "ipa
6010 IF pa<0 OR pa>7 THEN GO TO
6000
6020 INPUT "Ink: "in
6030 IF in<0 OR in>7 THEN GO TO
6020
6040 INPUT "Flash: "ifl
6050 IF fl<0 OR fl>1 THEN GO TO
6040
6060 INPUT "Bright: "ibr
6070 IF br<0 OR br>1 THEN GO TO
6060
6080 POKE 23696,0; POKE 23695,12
8xfl+64xbr+8xpa+in
6090 RETURN
7000 PAUSE 0
7005 PRINT AT oo,pp; FLASH 0; OV
ER 1;"■"
7010 IF INKEY$="5" OR CODE INKEY
$=0 THEN LET pp=pp-1x(pp<0)
7020 IF INKEY$="8" OR CODE INKEY
$=9 THEN LET pp=pp+1x(pp<31)
7030 IF INKEY$="6" OR CODE INKEY
$=10 THEN LET oo=oo+1
7040 IF INKEY$="7" OR CODE INKEY
$=11 THEN LET oo=oo-1x(oo<0)
7050 IF INKEY$="0" OR CODE INKEY
$=13 THEN LET bu=1
7060 RETURN
7100 LET z$=INKEY$: LET zz=CODE
z$
7105 IF z$==" OR zz<8 OR (zz>11
AND zz<32 AND zz<>15) THEN GO T
O 7100
7106 BEEP .01,10; LET fl=0; GO
SUB 7200; LET fl=1
7110 IF zz=15 THEN LET gr=ABS (
gr-1); GO TO 7200
7120 IF zz=8 THEN LET pp=pp-1x(
pp>0); GO TO 7200
7130 IF zz=9 THEN LET pp=pp+1x(
pp<31); GO TO 7200
7140 IF zz=11 THEN LET oo=oo-1x
(oo>0); GO TO 7200
7150 IF zz=18 THEN LET oo=oo+1x
(oo<23); GO TO 7200
7160 IF z$=" STOP " THEN LET st
=1: RETURN
7180 IF zz>127 THEN GO SUB 7200
: GO TO 7100
7190 GO TO 7300
7200 IF oo>21 THEN GO TO 7230
7210 PRINT AT oo,pp; OVER 1; FLA
SH fl;bs(gr+1)
7220 RETURN
7230 PRINT #0; PAPER 0; INK 8;AT
oo-22,pp; OVER 1; FLASH fl;bs(
gr+1)
7240 RETURN
7300 IF gr=1 THEN GO TO 7400
7310 IF oo>21 THEN GO TO 7350
7320 PRINT AT oo,pp;CHR$ zz
7330 LET pp=pp+1x(pp<31); GO TO
7200
7350 PRINT #0;AT oo-22,pp; INK 8
; PAPER 8;CHR$ zz; GO TO 7330
7400 IF zz<97 OR zz>117 THEN GO
SUB 7200; GO TO 7100
7410 LET zz=zz+47; GO TO 7310
8000 LET xx=USR uu; BEEP .1,10;
IF xx=65535 THEN GO TO 19
8010 LET x1=INT (xx/256); LET y1
=xx-x1x256; POKE xy,y1; POKE xy+
1,x1; RANDOMIZE USR (uu+1444)
8020 LET xx=USR uu; IF xx=65535
THEN GO TO 19
8030 LET x2=INT (xx/256); LET y2
=xx-x2x256
8040 IF x2<x1 OR y2<y1 THEN G
O TO 8020
8050 RANDOMIZE USR (uu+1444); PO
KE xxyy,y2; POKE xxyy+1,x2
8060 RETURN
8100 BEEP .1,10; LET xx=USR uu;
BEEP .1,10; IF xx=65535 THEN GO
TO 19
8110 LET x1=INT (xx/256); LET y1
=xx-x1x256
8120 POKE 65534,y1; POKE 65535,x
1; RETURN
8200 LET xx=USR uu; BEEP .1,10;
IF xx=65535 THEN GO TO 19
8210 LET x1=INT (xx/256); LET y1
=xx-x1x256; POKE xy,y1; POKE xy+
1,x1; RANDOMIZE USR (uu+1444)
8220 LET xx=USR uu; IF xx=65535
THEN GO TO 19

```

```

8238 LET x2=INT (xx/256): LET y2
=xx-x2X256
8240 IF x1>=x2 THEN GO TO 8228
8250 RANDOMIZE USR (uu+1444): PO
KE xxyy,y2: POKE xxyy+1,x2
8260 RETURN
9000 RESTORE : DIM a$(33,8)
9010 FOR a=1 TO 33: READ a$(a):
NEXT a
9020 DATA "CHAR 1","CHAR 2","NOR
MCHAR","ROTATE","MIRROR H","MIRR
OR V","COPY","MOVE","DELETE","FE
TCH","INVERSE","ZOOM"
9030 DATA "ATTR","RECALL","BACKU
P","LD 1--S","PRINT","EX CHAR","
OR S--1","OR S--2","SCHATT","EX
S--1","EX S--2","SINCLAIR"
9040 DATA "EIGEN","SAVE","MSAVE"
,"CATALOG","LOAD","MLOAD","ERASE

```

```

,"LOAD CH","MLOAD CH"
9050 RETURN
9080
9810
9820 SAVE "EDDI" LINE 9999
9830 SAVE "EDDICODE"CODE 62750,1
500: SAVE "EDDI UDG"CODE USR "a"
,160
9840
9850 VERIFY "EDDI": VERIFY "EDDI
CODE"CODE : VERIFY "EDDI UDG"COD
E
9860 STOP
9870
9880 SAVE "EDDI": VERIFY "EDDI"
9890 STOP
9900
9999 CLEAR 32767: LOAD "EDDICODE
"CODE 62750: LOAD "EDDI UDG"CODE
USR "a": RUN

```

Poke Stripper

Nach "Strip Poker", "Peepshow" und "Samantha Fox Strip-Poker" liegt mit "Poke Stripper" nun ein weiteres Programm vor, in dem unbedeckte Damen eine Rolle spielen.

Das Programm bietet dem Spieler eine handfeste Poker-

Two Pairs - zwei Paare

One Pair - ein Paar

Beide Spieler (also ein Spieler und der Spectrum) erhalten zum Programmstart ein Kapital von 100 \$. Vor jeder Runde kommen davon 10 \$ in den

Hat man sie entblättert, endet das Spiel.

Da die Grafik ansprechend gestaltet und auch eine Spiel-motivation vorhanden ist, kann man das Spiel eigentlich als gut bezeichnen. Ich bin einmal gespannt, ob in Zukunft vielleicht

ein Programm auf den Markt kommt, in dem Männer ausgenutzt werden - natürlich nur im Rahmen der Gleichberechtigung.

System: Spectrum 48 K
Hersteller: Individual Software
Rolf Knaore



Handfester Poker
um Schlips und
Kragen

partie. Gespielt wird mit fünf Karten, die grafisch gut dargestellt sind. Folgende Wertungen werden gezählt:

Straight Flush - Straße aus einer Farbe
Four of a kind - 4 Karten eines Wertes

Full House - ein Paar und ein Drilling

Flush - alle Karten haben die gleiche Farbe

Straight - eine Sequenz von Werten

Triplet - ein Drilling

Pott. Danach können maximal 3 Karten neu ausgegeben werden, um das bestmögliche Blatt zu erreichen. Nun kann man den Einsatz stehenlassen oder erhöhen. Bis zu diesem Punkt handelt es sich also um eine ganz normale Pokerpattie.

Hat nun aber das auf dem Bildschirm dargestellte Mädchen nicht mehr genug Kapital, um mitzugehen, muß es ein Kleidungsstück ablegen. Das soll dem Programm die nötige Würze geben. Insgesamt bietet "Poke Stripper" drei Mädchen.

30 Games

Aus England erreicht uns ein Leckerbissen für Spectrum-Einsteiger. Es handelt sich um ein Software-Paket mit insgesamt 30 Spielen. Allerdings sind es keine neuen Titel, sondern leicht angestaubte Programme, die alle schon mal in den letzten drei Jahren veröffentlicht wurden. Ihre Qualität reicht von hervorragend bis eher mittelmäßig. Trotzdem kann man die Sammlung denjenigen, die bisher noch keine Programme haben, durchaus empfehlen. Da der Preis nicht zu hoch ist, wird sicher jeder zufriedengestellt. Folgende Titel sind enthalten:

"Chessplayer", "Micromouse", "Tubecube", "Invasion", "Escape", "Lazer Zone", "Strontium Dog", "Pyramania", "Blood 'n' Guts".

Wer seinen Spectrum schon länger besitzt, wird sicher das eine oder andere Programm noch kennen. Mit wenigen Ausnahmen handelt es sich um Action-Spiele aller Art. Ausgeliefert wird "30 Games" auf zwei Cassetten mit den entsprechenden Spielbeschreibungen.

System: Spectrum 48 K
Verschiedene Hersteller
Rolf Knaore

"Carpet Capers", "Antics", "Planetfall", "Gridrunner", "Bismark", "Meteo Storm", "Jet Set Gerlie", "3D Tunnel", "Stockmarket", "Fridge Frenzy", "The Valley", "Dragonsbane", "Demon Knight", "Neds Garden", "Space Intruders", "Mighty Magus", "Fall of Rome", "Draughts", "Xadom", "Detective", "20 Tons",



3 kurze Programme für den ZX Spectrum

"Elite-Code" ist die Auflösung des Spiels "Elite". Der von diesem Programm erstellte Code wird mit dem LOAD-Menüpunkt bei "Elite" geladen. Nach dem Starten des Spiels braucht man nur noch ein Objekt abzuschließen, um zur sagenumwobenen Elite des Weltraums zu gehören. Das Programm sichert den Code nach dem Start "normal" ab, wobei man darauf achten muß, daß der Header, der beim Laden sonst den Namen auf dem Bildschirm hervorruft, nicht auf Cassette abgespeichert wird, d.h., man darf den Recorder erst nach dem Header anschalten.

Das Programm "Streckung" verbreitert Text und Grafik auf dem Bildschirm. Nach jedem

Aufruf mit RANDOMIZE USR 30000 wird alles in Sekunden um ein Pixel breiter. Der Code wird automatisch abgespeichert und überprüft (VERIFY), worauf ein kleines Demo folgt. Die Routine ist übrigens auf jede beliebige Adresse verschiebbar.

Das Programm "Primzahlenberechnung" erstellt auf einem Spectrum 48K theoretisch bis zu 8000 Primzahlen. Die Zahlen werden relativ schnell ermittelt und im Array "A" gespeichert. Die Dauer der Berechnung nimmt mit dem Ansteigen der Primzahlen zu, da es nun immer mehr gibt, durch die eine Zahl geteilt werden muß, damit der Computer weitere Primzahlen feststellen kann.

Gunter Imeyer

Elite-Code

```

1 REM ELITE-CODE
2 REM von Gunther Imeyer
3
10 FOR n=304 TO 304+22: READ a
: POKE n,a: NEXT n
20 DATA 69,76,73,84,69,0,82,50
,48,48,0,64,255,255,25,14,0,7,19
9,255,97,11,158
30 FOR n=304+23 TO 304+39: POK
E n,0: NEXT n
40 FOR n=304+40 TO 304+102: RE
AD a: POKE n,a: NEXT n
50 DATA 35,3,1,0,0,0,70,4,89,1

```

```

27,0,0,255,255,255,255,255,0
,0,82,210,66,16,154,189,106,69,5
0,172,74,161,5,6,16,17,23,25,34,
15,0,0,15,41,0,39,60,13,31,13,0,
0,1,206,154,2,92,212,137,243,240
,149,0

```

```
60 SAVE "ELITE"CODE 304,102
```

Streckung

```

10 LET a=0: FOR n=304 TO 30020
: READ b: LET a=a+b: POKE n,b: N
EXT n: IF a<2671 THEN PRINT "F
ehler in Data !!": STOP
20 DATA 55,63,17,0,24,33,0,64,
126,31,35,27,56,11,43,182,119,35
,122,179,254,0,32,240,201,203,25
4,24,241
30 SAVE "Streckung"CODE 304,30
: VERIFY "CODE"
40 BORDER 0: PAPER 0: INK 7: B
RIGHT 1: CLS: PRINT "D E M O"
50 FOR n=-PI TO PI STEP .2: PL
OT 43,88: DRAW 170,0,n: NEXT n:
FOR n=-PI TO PI STEP .2: PLOT 12
0,6: DRAW 0,168,n: NEXT n: PAUSE
5: PAUSE 100
100 FOR n=0 TO 30: RANDOMIZE US
R 30000: NEXT n

```

Primzahlenberechnung

```

1 REM Primzahlenberechnung
2 REM von Gunther Imeyer
3
10 DIM a(8000): LET a(1)=2
20 LET b=1
30 FOR n=2 TO 1020
40 FOR c=1 TO b/2+1
50 IF n/a(c)<INT (n/a(c)) THE
N NEXT c: LET b=b+1: LET a(b)=n
: PRINT AT 0,0;n: NEXT n
70 NEXT n

```

Tantalus

Wieder einmal ist die Menschheit in Gefahr, wieder einmal droht ihr die totale Vernichtung – die Punkoiden greifen an.

Das neueste Programm von Quicksilver bietet eine im Science-fiction-Milieu angesiedelte Schlacht apokalyptischen Ausmaßes. Wer sich für Action-Schießspiele begeistern kann, wird hier gut bedient. In einem technologisch gestylten Labyrinth muß der Spieler einen



kleinen Roboter auf die Jagd nach den Punkoiden schicken, die in vielen Formen und ungeheurer Zahl auftreten. Da man nicht gleich bei jeder Kollision ein Leben verliert, macht die Ballerei eine Menge Spaß, zumal die Grafik wirklich hervorragend ist. Nur ist leider kein Sound vorhanden. Trotz der einfachen Geschichte wird "Tantalus" viele Freunde finden.

System: Spectrum 48/128 K
 Hersteller: Quicksilver
 Rolf Knorr

SPECTRUM * Hard- und Software zu günstigen Preisen * SPECTRUM

Das neue ISO-ROM für den Spectrum! Jetzt noch besser. Mit:

- **Neuer ohne Programmverlust!** • Deutsche Umlaute!
- **Reset Zeichenstift!** • Verbesserter CAT für MD!
- **Verkäufliche Syntax** für Microdrive oder Beta-Disk!
- **Erweiterter Editor** mit Cursor/Nein Programmieren!
- **Variable Zeichenbreite:** 32, 36, 42, 54 oder 64 Zeichen pro Zeile (nur bei Monitor-ROM!)

Dazu eine von 2 NMI-Routinen:

Erweiterer den HEX-Monitor: Unterbricht jedes Programm, ermöglicht Schreiben und Starten eigener MC-Routinen!
Oder die neue Backup-Routine: Überträgt sämtliche Programme auf Ihr Spielchemedrum! Dabei werden nur die **belegten Bytes abgelesen** (also nicht immer volle 48K). Die übertragenen Programme laufen natürlich auch ohne ISO-ROM.

Außerdem: **POKE-Option!** Sie können jederzeit Ihr Game unterbrechen, einen POKE eingeben und weiterpielen.
Unentbehrlich für alle Spieler!!!
 Backup-Routine gibt's für Microdrive, Optisk. Disk, Beta-Disk und Cassette (für alle anderen Speichermedien)! **Fördern Sie zum ISO-ROM unser Info!**

Komplett mit deutscher Anleitung und Einbauplan für Einbau direkt im Computer (auch für Parallelbetrieb mit Sinclair-ROM!)

Monitor-ROM: Jetzt nur 70,- DM **Backup-ROM: Jetzt nur 80,- DM**
 Auf Wunsch basen wir Ihr ROM für nur 25,- DM in Ihren Computer ein.

SYN - Das vollständige MC-Entwicklungspaket für den Spectrum!!!
 Lösung aller MC-Probleme. Sie beschreiben Macrossensibler mit extrem platzsparendem Textformat, **Disassembler, Debugger** mit Single-Step und Trace und **symbolischem Reassembler!** Unterstützt alle Speichermedien (Bitte bei Bestellung angeben)!

Amadeus - Das Spectrum-Musiksystem für alle 3-Kanal-Soundmodule. Synthesizer mit 7 Instrumenten, Notentast für bequeme Noteneingabe und Korrektur. Jetzt nur 45,- DM.

Zusammen mit 3-Kanal-Soundmodul nur 125,- DM.

Jetzt brandneu: **Das ISO-FACE!** Ermöglicht den Betrieb von bis zu 3 externen ROMs durch einfaches Anstecken an den Spectrum.

- 3 Steckplätze für 27128-er EPROMs!
- Umschaltbar Sinclair/Ext. ROM, abwärtsfrei, 3-fach Schalter für die ext. ROMs und präfixen Reset/NMI-Taster!
- **Durchgeführter Bus und stabiles Gehäuse!**
- **Das alles für sage und schreibe nur 100,- DM!!!**
- ISO-FACE komplett mit einem ISO-ROM nur 190,- DM!

BETA-PACK - Das Utility-Paket für Ihre Beta-Disk:

1. **BETA-TRANS** - Bringt Files von Cassette auf Diskette (auch headerlos!)
 2. **BETA-COPY** - Kopiert beliebige Files von Disk auf Disk!
 3. **BACKUP 48** - Schreibe Backup, lädt fast alle 48K auf einmal!
 4. **BETAADDRESS** - Findet Anfangsadressen von headerlosen CodeBlöcken!
 5. **BETA-TAPE** - Überträgt Diskettenprogramme zurück auf Cassette!
- Alle Programme arbeiten mit einem oder mehreren Laufwerken und allen DOS-Versionen (Bitte angeben)
 5 Topprogramme auf 1 Diskette: **Nur 70,- DM**

ISO-DAT1 - Endlich ein Dateiprogramm, das den Speicherplatz der Beta-Disk voll ausnutzt! Bis zu 5000 Einträge auf einer Disk, komfortable Screeneditor, deutsche Umlaute, Programm und Anleitung in Deutsch, sehr benutzerfreundlich!
 Jetzt nur 45,- DM!

BETA-BOOT - "boot"-File für jede Diskette! Zum Laden, Umbenennen, Löschen usw. nur noch File per Cursor anfahren. **Nur 25,- DM**

POKE-STRIPPER - Geben Sie Ihr POKefahrer auf und ziehen Sie sich warm an! Scharfen Sie an, spitzen Sie, trenn und Däre Sieger zu bleiben! Das Spectrum-Stripper mit den coolsten Sprüchen und den wackelsten Mädchen!
Nur 30,- DM

KEMPSTON PRO - Das Universal-Joystickinterface für den Spectrum!!!
 Kempston + Sinclair + Cursor-Interface in einem. **Jetzt nur 60,- DM**

Gratis-Info und Bestellungen (V-Scheck oder Nachnahme) bei:

Volker Marohn
Am Beilstück 30
4600 Dortmund 50

INDIVIDUAL
SOFTWARE

Telefonische
Bestellungen:
0231/716688

ZX Spectrum Hardware

Verlag Markt & Technik
 Von R. Egeler
 150 Seiten, DM 29,80
 ISBN 3-89090-092-5

Dieses Buch gibt allen Hardwarefans und solchen, die es werden wollen eine optimale Hilfestellung für ihre Bausteine in und am Spectrum. Daneben ermöglichen die vermittelten Kenntnisse ein leichteres Auffinden von Hardwarefehlern.

Das Buch gliedert sich in zwei Teile. Zunächst werden die verschiedenen internen Baugruppen des Spectrum erläutert. Begonnen bei der Z80-CPU, der Speicherorganisation, den vielfältigen Aufgaben des ULA usw. werden alle Baugruppen ausführlich anhand von Signalgebung und Schaltungsausdrücken erklärt. Die Beschreibung der Spectrum-Innenreie ist so gewählt, daß jeder Bereich erfaßt wurde. Daneben werden in einem gesonderten Kapitel Tipps für die Fehlersuche bei einem Defekt gegeben. Hinweise auf

die Anschaltung von Interface 1, welche man im Notfall aber auch einem der im Handel befindlichen Microdrive-Bücher entnehmen kann, fehlen leider. Im Anfang des Buches befinden sich noch zwei herausklappbare Schaltpläne der Spectrum-Versionen Issue 2 und 3, einschließlich der Unterschiede zu Issue 3b.

Der zweite und umfangreichere Teil des Buches befaßt sich mit eigenen Hardwareerweiterungen, die ohne großen Aufwand nachgebaut werden können. Alle in den Bausteinen vorkommenden Sonderbausteine werden - wie schon bei der Beschreibung der Sinclair-Hardware - anhand ihrer Anschlüsse und Funktionen genau erläutert. Der Reigen der Bausteine beginnt mit einer Einführung in die möglichen Adressierungsarten der I/O-Geräte und der Anleitung zum Aufbau einer PIO (parallel Input/Output), mit der dann die nachfolgenden Bausteine ohne Probleme oder Gefahren für

den Spectrum angesteuert werden. Neben einigen kleinen Bausteinen für Ein- und Ausgabe mit der PIO und einer Erklärung der PIO-Programmierung in Maschinensprache und Interrupts wurden von Autor R. Egeler noch drei große Bauanleitungen aufgezeichnet und dokumentiert.

Da ist zunächst die Anleitung zum Bau einer software-gesteuerten Hardwareuhr, die dank eines Akkus auch noch nach dem Ausschalten der Spectrum-Stromversorgung weiterläuft. So kann man z.B. die aktuelle Uhrzeit abfragen, um diese in einem Programm weiterzuverwenden.

Wer sich an die zweite Bauanleitung, einen Soundgenerator mit dem bewährten IC AY 3-8912, hält, kann eine Platine aufbauen, mit welcher der Spectrum in Sachen Musik nun auch seinen Konkurrenten Commodore und Schneider nicht mehr nachhinkt, da diese mit demselben IC bestückt sind. Neben einer kompletten Beschreibung des Platinaufbaus und der Funktionsweise

des Sound-ICs ist auch das Listing eines komfortablen Musik-Editors abgedruckt.

Da ja bekanntlich die Herstellung der Platine, das Beschaffen aller Bauteile oder Abtippen der Listings manchmal sehr umständlich ist, kann man im Anhang des Buches eine Bezugsquelle erfahren, über die Bausteine, Platinen oder Programme zu beziehen sind.

Die letzte Anleitung befaßt sich mit dem Aufbau eines Analog/Digital-Wandlers. Dieser wird im Buch als Speicheroszilloskop für den Spectrum verwendet, eignet sich aber auch für andere Zwecke der Meßwertfassung.

Nach dieser kleinen Inhaltsbeschreibung kann wohl jeder Interessierte selbst entscheiden, ob sich der Kauf des Buches lohnt. Zusammenfassend kann man wohl sagen, daß jeder, der mehr aus seinem Spectrum machen will, mit diesem Buch sehr gut beraten ist. Auch ist der Preis von unter 30,-DM für eine derartige Ansammlung von Know-how ein weiterer Pluspunkt.

Thomas Bortolotto

Quercopy für Spectrum 16/48 K

Jetzt können Sie nicht nur sehr lange Ausdrücke anfertigen, sondern auch sehr breite. "Quercopy" erstellt eine Bildschirmkopie im Querformat (24 Zeilen!).

Selbstverständlich ist die kurze MC-Routine nicht an einen

bestimmten Speicherplatz gebunden. Außerufen wird sie mit der Ladeadresse ("RANDOMIZE USR xxxx"), wonach die 32 Spalten hintereinander quer ausgedruckt werden.

Insbesondere Benutzer des Seikoha GP 50 \$ werden sich über den Querausdruck freuen, da der Nadelkopf nicht quadratisch ist und man jetzt zwischen

zwei "verzerrten" Bildern das bessere auswählen kann. Metallpapier-Freunde mit einer 48-K-Version werden es zu schätzen wissen, daß mittels Einstieg über "RANDOMIZE USR xxxx+5" erst das gesamte Querbild erzeugt und danach in einem Durchgang ohne Stoppen des Motors abgedruckt wird. Das gelingt aber nur, wenn das Programm nicht im Bereich 39000-47192 steht.

Falls dieser auf yyyyy+8129 verschoben werden soll, sind das 7. (LOW-) bzw. 8. (High-) Byte (hier 88 und 151) entsprechend mit der Adresse yyyyy-256 zu ändern.

Nach Eintippen und RUN speichern Sie das Programm bitte mit SAVE "quercopy" CODE 30000.152 ab.

Harald Hagemann

Quercopy

```
10 REM <quercopy> , h.h.
20 CLEAR 29999: RESTORE 100
30 LET fst=30000: LET sum=0: F
OR i=0 TO 151: READ j: LET sum=s
um+j: POKE fst+i,j: NEXT i: IF s
um<>15520 THEN PRINT "error!"
100 DATA 1,0,0,24,3,1,88,151,19
7,33,59,92,203,206,205,223,14,21
7,22,32,217,33,0,64,217,14,0,217
,221,33,27,91,217,6,3,217,14,0
```

```
110 DATA 6,8,203,6,221,203,0,30
,221,43,17,32,0,25,16,242,17,0,0
,221,25,13,32,232,17,248,255,221
,25,217,5,217,32,220,17,0,232,25
120 DATA 17,56,0,221,25,217,13
,217,32,202,241,245,254,0,32,41,2
29,205,205,14,225,35,217,21,217
,32,177,241,254,0,40,19,243,22,0
,33,0,152
130 DATA 6,10,205,244,14,33,128
,152,6,255,205,211,14,33,59,92,2
03,142,201,235,193,33,0,1,9,229
,213,235,33,0,91,1,0,1,237,174,24
,200
```

Low Price Software für den Spectrum

Ein Trend in der Softwarebranche ist nicht zu übersehen: Die Programme werden billiger. Die Firma Mastertronic war einst Auslöser dieser Welle. Mit Programm-Cassetten um 10,-DM hat sie sich inzwischen auf dem Markt etablieren können; Nachahmer gibt es auch schon. Dabei hat sich herausgestellt, daß preiswerte Software nicht unbedingt schlecht sein muß.

Das englische Softwarehaus Elite Systems Ltd., bekannt durch Titel wie "Frank Bruno's Boxing", "Commando" oder "Ghosts'n Goblins", hat vor einigen Wochen ein Label aus der Taufe gehoben, das dem neuen Trend folgen soll. Unter der Bezeichnung 2-99 Classics werden aber keine neuen Titel angelegt, sondern ausschließlich ältere, bereits mit Erfolg verkaufte Programme angeboten. Besonders interessant ist der Verkaufspreis, der sich schon aus der Label-Bezeichnung ableiten läßt: 2-99 steht für 2,99 englische Pfund, was etwa 11,-DM entspricht. Folgende Titel sind bisher für Spectrum-Besitzer erhältlich:

"Full Throttle", "3D Death Chase", "Skool Daze", "Valhalla", "Pool", "Scuba Dive", "Chess", "3D Star Strike", "Jasper", "3D Tank Duel", "Tornado Low Level".

Damit können besonders Neucinsteiger preiswert zu Programmen kommen, die zwar

schon älter sind, aber dennoch hervorragende Unterhaltung bieten. Die Reihe soll in nächster Zeit ständig erweitert werden. Auch andere Computertypen werden dann Berücksichtigung finden. Ein deutscher Lie-

ferant war zum Zeitpunkt der Drucklegung noch nicht bekannt, wird aber sicher nicht mehr lange auf sich warten lassen.

Rolf Knorre

Spectrum User Club Wuppertal



Informationen erhalten Sie (gegen
Einsendung von DM 0,50 Rückporto) von:
Rolf Knorre,
Postfach 20 01 02, 5600 Wuppertal 2



it's sinclair time



Die drei
Superspiele zum
Wahnsinnspreis
von je DM 24,-^(Cass)



2 Bücher unentbehrlich - ehrlich!
ZX 81 DM 49,-
Sinclair QL DM 69,-



Das schärfste für den
Spectrum - Peep Show.
Bitte Alter angeben.
DM 29,90

Actiongame mit
32 Screens.
32 Screens. 100%
Maschinensprache.
DM 19,-



Eine Reise
von New York
nach Ägypten.
Deutsches Grafik-
Adventure für
DM 16,-



Spezialitäten-Bestellschein

Best.-Nr.	Artikel	
N2	Paperboy	
N3	Light Force	
N3	Alles über den ZX81	
N4	QL unter Kontrolle	
N5	Peep-Show	
N6	Super-Story	
N7	G Force	
N10	1942	

Ich erwarte folgende
Bestellung:
 Nachnahme (max. 3,00 DM Versandkosten)
 Vorkasse
Bitte Beschriftung
des Versandbehälters mit
oder auf Nachnahmekonto
Einkaufspreise

Name/Vorname: _____ Nachname: _____

Adresse: _____

Computer Kontakt, Softwareversand, Postfach 1640, 7518 Bretten.

Zitterschrift

Bei "Zitterschrift" für den Spectrum 1648 K handelt es sich um ein kurzes Programm, mit dem man ein Wort (Satz) auf dem Bildschirm "zittern" lassen kann. Die Variablen haben folgende Bedeutung:

po1 Position des Wortes auf dem Bildschirm

po2 Position des Wortes auf dem Bildschirm
le Länge des Wortes
is das Wort selbst

Es ist zu beachten, daß man, um aus dem Programm herauszukommen, nicht Break drücken darf, sondern irgendeine andere Taste, weil das Programm den Zeichensatz in der Adresse 23606 verändert!

Torsten Rüsse

```

5 LET po1=10: LET po2=11: REM
  POSITION DES WORTES
10 LET x=0
20 LET p=0
30 LET a$=""
40 LET le=7: REM
  LAENGE DES WORTES
50 DIM i$(le)
60 REM UMWANDLER
70 LET i$="T.RINNE": REM
  DAS WORT WELCHES ZITTRT
80 FOR n=1 TO le
90 LET a$=a$+CHR$(CODE i$(n)
)-1)
100 NEXT n
105 REM ZITTERN
110 PRINT AT po1,po2;a$
120 POKE 23606,x+7
130 IF p=0 THEN LET x=x+1: IF
x=3 THEN LET x=x-1: LET p=1
140 IF p=1 THEN LET x=x-1: IF
x=-1 THEN LET x=x+2: LET p=0
150 IF INKEY$<"**" THEN POKE 23
606,0: STOP
160 GO TO 110
  
```

Core

"Core" ist ein neues Programm aus dem bei Programmieren so beliebten Science-fiction-Milieu. Die Story könnte jedenfalls Grundlage für einen interessanten Film sein. Commander Andrew Angello erhält den Auftrag, eine von der Erde installierte wissenschaftliche Station auf dem Planeten EROC 1 zu untersuchen, da man von ihr Notsignale aufgefangen hat. Als Angello per Teleporterstrahl dort eintrifft, lebt von den 720 Wissenschaftlern niemand mehr. Seine Aufgabe ist es nun, versteckte Teile eines Computers zusammenzutragen, um mit diesem

einen Gegenangriff gegen die Invasoren zu starten.

Der Programmtitel "Core" steht übrigens für "Cybernetic Organism Recovery Expedition". Laut Anleitung soll die Station aus über 1000 Räumen bestehen, die ich allerdings nicht nachgezählt habe. Auch hier sollte man wieder eine Karte zeichnen, um den Überblick nicht zu verlieren. Das Programm selbst ist grafisch gut gemacht, bietet jedoch kaum Überraschungen. Wie so oft muß man Teile untersuchen und aufnehmen und sich dabei natürlich vor zahlreichen Feinden in acht nehmen. Das Spiel ist dennoch guter Durchschnitt.

System: Spectrum 48 K
Roll Keycode

Schaltbild - CAD

Dieses Programm wendet sich an alle Hobby- und Amateurelektroniker, die ihre eigenen Schaltpläne erstellen. Will man selbst einen Schaltplan (z.B. eine Schnittstelle für den Computer) entwerfen, so geschieht dies zuerst in Gedanken und mit einigen Kritzeleien auf einem Blatt Papier. Das hier vorgestellte Programm erlaubt nun, die ersten Ideen für einen Schaltplan ordentlich aufzuzeichnen und auszudrucken.

Nach dem Start des Programms befindet man sich in einem Einstiegsmenü, welches z.B. die Änderung der Druckerart oder des Speichermediums ermöglicht. Bei Abspeicherung des Programms auf Microdrive werden automatisch alle erforderlichen Menü- und die Microdrive-Optionen (Cataloge, Erase, etc.) erweitert, so daß man beim Laden von Cartridges sofort eine erweiterte Version vor sich hat.

Nach einem Sprung in die Zeichenroutine kann das Schaltbild aufgebaut werden. Dies geschieht mit Hilfe der Cursor- und verschiedener Funktionstasten. Da das Programm über eine sehr umfangreiche Bauteilesammlung verfügt (vom einfachen Widerstand bis zu verschiedenen ICs und logischen Gattern), sind eigentlich alle normalen Schaltungen möglich. Daneben ist noch wahlweise eine automatische oder manuelle Beschriftung der Bauelemente sowie

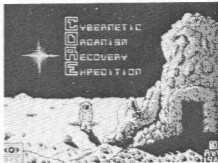
das Einfügen von beliebigen Texten in die Schaltung erlaubt. Bei der automatischen Beschriftung werden alle Bauteile gleicher Art gezählt und im Schaltplan mit entsprechender Bezeichnung versehen (z.B. R1, R2, ...). Alle Texte im Schaltplan werden mit 64 Zeichen je Zeile geschrieben.

Nach Verlassen des Zeichen-Editors läßt sich noch jedes erfaßte Bauteil mit einem Wert versehen, woraus sich eine Liste der Bauteile bildet. Diese kann dann zusammen mit dem Schaltplan auf einem Drucker (normale oder 4fache Größe) ausgegeben werden und ergibt so eine vollständige Dokumentation der Schaltung. Als Ergänzung ist noch ein ganzseitiger Text-Editor eingebaut, um weitere technische oder sonstige Daten zur Schaltung niederzuschreiben zu können.

Natürlich lassen sich alle erstellten Schaltpläne auf das aktuelle Speichermedium retten, um sie später einmal zu ändern oder zu erweitern.

Da das Programm zum größten Teil in Basic geschrieben wurde, ist schnelles Zeichnen zwar nicht so ohne weiteres möglich, aber die eigentliche Entwicklungsarbeit findet ja so wieso im Kopf statt.

Weitere Informationen:
A. Thiesen
Rathausstraße 70
5410 Höhr-Grenzhausen
Thomas Herold



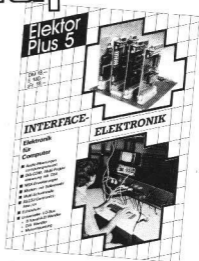
Brandneues Special

von

ELEKTOR

Interface- Elektronik

Auf dieses Heft haben die Computer-Freaks schon lange gewartet! Mit dieser Zeitschrift haben Sie die Möglichkeit sich Ihre entsprechende Schnittstelle selbst zu bauen.



Hier ein kleiner Themenauszug:

Grundlagen

Info-Artikel über Sinn, Zweck und Anwendungsmöglichkeiten von Interfaces, z.B. zur Steuerung von Robotern oder das Messen von Lautsprecherboxen

Marktberichte

Übersicht von industriell hergestellten Interfaces und Zubehör

Bauanleitungen

* verschiedene Computer und Peripheriegeräte sind anschließbar

- * RS232/Centronics-Schnittstelle
- * RS232-Analysator
- * C64/128-Schnittstelle
- * C64 Kassetteninterface
- * MSX-Erweiterungen
- * I/O-Erweiterung für MSX-Computer
- * Modem mit Selbstwahl-Interface
- * Universeller I/O-BUS
- * I/O-BUS-Karte Serielle Schnittstelle
- * I/O-BUS-Karte Kanalwähler
- * I/O-BUS-Karte Analogeingang
- * I/O-BUS-Karte Echtzeituhr

Schauen Sie mal rein, es lohnt sich!

Erscheinungstermin ist am
21. November 1986

Direkt beim Verlag oder im Fachhandel
erhältlich, DM 15,-

Elektor Verlag, Süsterfeldstr. 25, 5100 Aachen

Window

Mit der Window-Routine lassen sich beliebige rechteckige Bildschirmabschnitte manipulieren: SCROLL und ROLL in alle 4 Richtungen, CLS, Window abspeichern und wieder in den Bildschirmspeicher zurückladen (auch an andere Positionen!). Attribute werden - da das Programm die einzelnen Optionen pixelweise ausführt - nicht verändert. Die Handhabung der Window-Routine ist etwas kompliziert, denn es müssen 6 Parameter übergeben werden. Der Aufruf erfolgt mit `USR a: PRINT b, c, d, e, f, g`. Die Variablen haben folgende Bedeutung:

a: Gibt die Adresse des Maschinencodes an, der trotz seiner fast 500 Byte frei im Speicher verschiebbar ist.

b und c: Sie geben die obere linke Ecke des Windows an. Hierbei darf b Werte von 0-191 und c von 0-31 besitzen. Daraus ergibt sich, daß in der Breite immer nur ganze Print-Positionen abgespeichert werden können.

d: Bestimmt die Höhe des Windows und darf im Bereich von 1-192 liegen.

e: Gibt die Breite des Windows an und kann Werte von 1-32 annehmen.

f: Gibt die Adresse an, ab der das Window abgespeichert bzw. abgerufen werden soll. Der Wert ist außer bei den Optionen "Abspeichern" und "Zurückladen" überflüssig,

muß jedoch immer mitangegeben werden.

g: Sie gibt als letzte und wohl wichtigste Variable an, welche Funktion ausgeführt werden soll. Folgende Werte ergeben nachstehende Funktionen:

- 0 Scroll links
- 1 Roll links
- 2 Scroll rechts
- 3 Roll rechts
- 4 Scroll hoch
- 5 Roll hoch
- 12 Scroll runter
- 13 Roll runter
- 16 Window im Speicher ablegen
- 48 Window aus Speicher holen
- 128 Window CLS

Alle Scroll/Roll-Operationen werden pixelweise ausge-

führt. Bei falschen Werten (z. B. Window zu groß) wird die Fehlermeldung A ausgegeben. Abgespeicherte Windows kann man durch Ändern der x- und y-Position leicht duplizieren, womit sich schöne Effekte erzielen lassen. Beim Abspeichern des Windows sollte man darauf achten, daß man keine Daten zerstört. Es können allerdings nicht viele große Screens abgespeichert werden, da der ganze Spectrum-Screen (ohne Attribute) schon 6144 Byte lang ist. Der Speicherplatzbedarf eines Windows errechnet sich aus Breite*Höhe; so läßt sich leicht der Anfang des nächsten Windows bestimmen. Anstelle von festen Integerwerten kann man für b, c, d, e, f und g auch Variablen u.ä. einsetzen, was den Umgang mit der Routine wesentlich erleichtert.

Dirk Zwern

MC-Generator

```

1 INPUT "STARTADRESSE DES MCO
DES ? ";MCODE
2 LET A=MCODE
3 FOR B=1 TO 50
4 RESTORE (B+19)
5 LET C=0
6 FOR D=1 TO 10
7 READ E
8 LET C=C+E
9 POKE A,E
10 LET A=A+1
11 NEXT D
12 READ F
13 IF F<0 THEN FLASH 1: PRIN
T ""FEHLER IN DATAZEILE "JB+19"
""GELESENE PRUEFSUMME: "JF""ERRE
CHNETE PRUEFSUMME: "JC: FLASH 0:
BEEP .1,-35: STOP
14 PRINT "DATAZEILE "JB+19;" I
ST O.K."
15 NEXT B
16 CLS : PRINT BRIGHT 1;"
ALLE DATAS SIND O.K."
17 PRINT BRIGHT 1;" MIT GOTO
9990 KANN DAS PRO- GRAMM LND
DER ERZEUGTE CODE AB-"
18 PRINT BRIGHT 1;" GESP
EICHERT WERDEN."
19 PRINT BRIGHT 1;" BEI CO
NTINUE ERFOLGT EIN
KURZDEMO. *; STOP

```

```

20 DATA 243,223,254,58,192,231
,22,6,213,231,1673
21 DATA 285,138,28,285,153,38,
289,197,21,32,1218
22 DATA 243,193,121,254,7,56,2
8,254,12,48,1288
23 DATA 16,254,13,48,12,254,16
,48,8,254,907
24 DATA 48,40,4,254,128,32,26,
221,33,176,962
25 DATA 92,221,113,8,253,225,2
89,123,289,83,1528
26 DATA 95,193,121,193,65,79,1
22,167,48,3,1878
27 DATA 128,48,7,253,33,58,92,
251,287,9,1886
28 DATA 254,193,48,245,123,167
,48,241,129,56,1496
29 DATA 238,254,33,48,234,221,
283,8,182,32,1365
30 DATA 119,221,283,8,94,48,4,
128,138,71,1882
31 DATA 5,197,213,128,238,7,87
,128,238,248,1457
32 DATA 15,15,15,71,15,15,15,2
38,224,129,744
33 DATA 111,128,238,24,246,64,
138,183,289,213,1458
34 DATA 221,283,8,126,48,14,54
,8,35,29,722
35 DATA 32,258,289,193,4,21,32
,289,251,281,1482
36 DATA 22,8,221,283,8,86,32,5
8,221,283,1838

```

```

37 DATA 8,78,48,48,125,131,61,
111,283,62,859
38 DATA 245,29,48,12,43,283,62
,48,4,35,721
39 DATA 283,254,43,29,32,244,2
21,283,8,78,1299
40 DATA 48,7,241,48,5,283,254,
24,1,241,1864
41 DATA 289,193,4,21,32,151,25
1,281,24,185,1191
42 DATA 24,31,283,38,245,29,48
,12,35,283,868
43 DATA 38,48,4,43,283,198,35,
29,32,244,874
44 DATA 221,283,8,78,48,219,24
1,48,217,283,1462
45 DATA 198,24,213,221,283,8,2
14,197,213,128,1683
46 DATA 238,7,87,128,238,248,1
5,15,15,71,1838
47 DATA 15,15,15,238,224,129,1
11,128,238,24,1113
48 DATA 246,64,138,183,289,213
,221,283,8,118,1499
49 DATA 48,6,229,253,229,225,2
53,225,126,253,1839
50 DATA 119,8,35,253,35,29,32,
246,289,193,1151
51 DATA 221,283,8,118,48,3,229
,253,225,4,1288
52 DATA 21,32,198,251,281,289,
193,217,8,245,1567
53 DATA 197,213,229,217,8,21,4
8,88,197,213,1415
54 DATA 229,217,225,289,193,21
7,213,22,8,25,1558
55 DATA 43,289,126,245,43,29,3
2,258,217,213,1487
56 DATA 217,289,197,213,8,62,2
,8,128,238,1266
57 DATA 7,87,128,238,248,15,15
,15,71,15,823
58 DATA 15,15,238,224,129,111,
128,238,24,246,1344
59 DATA 64,138,183,289,193,197
,213,4,8,61,1182
60 DATA 245,8,241,48,15,229,25
3,225,221,283,1688
61 DATA 8,94,48,2,5,5,24,286,2
4,67,467
62 DATA 126,253,119,8,253,35,3
5,29,32,246,1128
63 DATA 289,193,221,283,8,94,4
8,2,5,5,972
64 DATA 4,21,32,174,213,128,23
8,7,87,128,1888
65 DATA 238,248,15,15,15,71,15
,15,15,238,869
66 DATA 224,129,111,128,238,24
,246,64,138,183,1381
67 DATA 289,241,221,283,8,78,4
8,3,119,24,1138
68 DATA 2,54,8,35,29,32,248,25
1,217,8,868
69 DATA 225,289,193,241,217,8,
251,281,8,8,1545
8999 REM WINDOW DEMO
9888 CLS : LIST 9888
9818 INPUT "X-POS. DES WINDOWS ?
" : IA
9828 INPUT "Y-POS. DES WINDOWS ?
" : IB
9838 INPUT "HOEHE DES WINDOWS ?
" : IC
9848 INPUT "BREITE DES WINDOWS ?
" : ID
9858 INPUT "ADRESSE FUER EVENTUE
LLEN ZUGRIFFEINES 2. WINDOWS ? "
: IE
9868 INPUT "FUNKTION ? " : IF
9878 PRINT #8:AT 1,0;"BITTE DRUE
CKEN SIE EINE TASTE !!": PAUSE 8
9888 FOR G=1 TO 100: LET Z=USR M
CODE: PRINT A,B,C,D,E,F: NEXT G
9898 PRINT #8:AT 1,0;"BITTE DRUE
CKEN SIE EINE TASTE !!": PAUSE 8
9998 LET A2=INT (MCODE/256): LET
A1=MCODE-256XA2
9991 POKE 23728,A1: POKE 23729,A
2
9992 CLEAR
9993 SAVE "WINDOW" LINE 1: SAVE
"WINDOWCODE"CODE (PEEK 23728+256
XPEEK 23729),588
9994 PRINT FLASH !:AT 18,8:"
VERIFY !
9995 VERIFY "WINDOW": VERIFY "WI
NDOWCODE"CODE (PEEK 23728+256XPE
EK 23729),588
9996 RUN
9999 REM

```

```

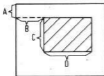
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
X 1986 BY G-S-S-T X
X 1986 BY DIRK ZWEERS X
X DALNERSTR. 15 X
X 5888 KOELN 41 X
X 8221/448769 X
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

```

Respos für ZX 81

Für die kleine Zahl unermüdlicher Fans des immer noch leistungsstarken ZX 81 ist es wichtig, weiterhin mit Neuerungen versehen zu werden.

"Respos" ist ein Maschinenprogramm, das viele Routinen zur Bildschirmbearbeitung enthält. Dabei geht es um die drei Grundfunktionen: Printen (FIL), Löschen (CLR) und Invertieren (INV). Ziel des Programms ist es, beliebige Rechtecke blitzschnell zu bearbeiten. Es bieten sich zwei Lösungen an: Die reihenweise Bearbeitung (RILF=reihenweise invertieren, löschen, füllen) und die Bearbeitung durch eine im Rechteck umlaufende Spirale (SILF=spiralg invertieren, löschen, füllen). Die letzte Methode hat den entscheidenden Vorteil, daß man die Spirale nach beliebig vielen Umläufen stoppen kann (Erzeugung von Rahmen). Auch ist im Hinblick auf optische Effekte ihre Geschwindigkeit steuerbar. Aus den beiden Grund-Routinen RILF und SILF setzen sich weitere zusammen. So gibt es abgekürzte Befehle für den ganzen Bildschirm und für ein symmetrisch zum Bildschirm liegendes Rechteck (Innenfeld ohne Rahmen). Schließlich erhält man noch 7 Effekte, die sich vor allem beim Invertieren verblüffend ausmachen.



Für die Bearbeitung eines Rechtecks ist es zunächst nötig, 4 Koordinaten für die Lage des Rechtecks einzugeben. "Respos" enthält zu diesem Zweck eine spezielle Routine, die es gestattet, alle erforderlichen Eingaben und den Startbefehl der gewünschten Routine in einer Basic-Zeile zu erledigen.

Rechteckkoordinaten

4 Zahlen sind in folgender Reihenfolge einzugeben:

A = Zahl der freien Zeilen über dem Rechteck

B = Zahl der freien Spalten links vom Rechteck

C = Zahl der Rechteckzeilen = Höhe des Rechtecks

D = Zahl der Rechteckspalten = Breite des Rechtecks

Befehle auf der Basis von RILF

Der Grundbefehl lautet: PRINT USR RILF, A, B, C, D, ART

Für A, B, C, D werden die erwähnten Rechteckkoordinaten eingegeben, für die Variable ART entweder CLR (löschen), INV (invertieren) oder FIL, X, wobei X der Code des zu printenden Zeichens ist. Für die Bearbeitung des gesamten Bildschirms gibt es RILF-TOTAL. Hier entfällt die Eingabe der Rechteckkoordinaten, so daß der reduzierte Befehl folgendermaßen lautet: PRINT USR RTOT, ART

Schließlich kann auch ein Innenfeld ohne Rand, also ein zum Bildschirm symmetrisch gelegenes Rechteck bearbeitet werden. Der Kurzbefehl für RILF-INNEN lautet: PRINT USR RIN, Z, ART. Hier gibt Z die Zahl der freien Zeilen um das Rechteck (Innenfeld) an, ART kann durch CLR, INV oder FIL, X ersetzt werden.

Befehle auf der Basis von SILF

Der Grundbefehl lautet hier: PRINT USR SILF, A, B, C, D, ART, BORDER, SLOW

Wie oben stellen A bis D die Rechteckkoordinaten dar, ART kann CLR, INV oder FIL, X sein. BORDER gibt die Zahl der auszuführenden Spiralaumläufe an: BORDER=0 bedeutet z.B., daß das Rechteck ganz von der Spirale ausgefüllt wird. SLOW steuert die Geschwindigkeit der umlaufenden Spirale: 0 gibt den schnellsten Umlauf an, mit zunehmendem SLOW-Wert wird das Tempo langsamer. Beispiel: PRINT USR SILF, 2, 5, 10, 12, INV, 2, 3 printet eine Rechteckspirale mit 2 Umläufen im Tempo 3 in Form einer Inversion der schon vorhandenen Zeichen.

PRINT USR STOT, ART, BORDER, SLOW ist die abge-

kürzte Version für SILF über den ganzen Bildschirm und PRINT USR SIN, Z, ART, BORDER, SLOW die Version für ein Innenfeld ohne Rand. Beispiel: Die Befehlsfolge PRINT USR STOT, FIL, 23, 3, 0; PRINT USR SIN, 2, INV, 1, 0 und PRINT USR RIN, 3, FIL, 8 erzeugt im höchsten Tempo (SLOW=0) einen Ma-chen Rahmen mit Sternchen um den ganzen Bildschirm, invertiert die innerste Zeile und füllt das Innenfeld grau aus.

Aus diesen 6 Grundbefehlen lassen sich beliebige eigene Effekte kombinieren, wie das Demo-Programm zeigt. Das Programm "Respos" enthält jedoch solche Effekt-Kombinationen im Maschinencode. Ihre Adressen befinden sich in einer Liste am Programmende. Damit wurde erreicht, daß der Aufruf allein durch Angabe der Effekt-Nummer möglich ist: PRINT USR EFF, NR, ART, SLOW

Wie man erkennt, ist NR die erste einzugebende Zahl, dann die ART der Bearbeitung und schließlich das Tempo. Die 7 Effekte (NR = 1 bis 7) sind am wirkungsvollsten bei der Inversion, machen sich aber auch beim Löschen des gesamten Bildschirms gut. Wer will, kann auch FIL, X versuchen; hier ist SLOW = 5 ein durchaus interessanter Wert. Folgende Möglichkeiten stehen bei den Effekt-Kombinationen zur Verfügung:

- EFF.1: Zeilenablauf von oben nach unten
- EFF.2: Spaltenablauf von links nach rechts
- EFF.3: Kontraktion horizontal
- EFF.4: Kontraktion vertikal
- EFF.5: Kontraktion horizontal und vertikal
- EFF.6: Koazentrisch wachsende Rechtecke
- EFF.7: Spirale von innen nach außen

Vorsichtsmaßnahmen

"Respos" enthält keine Absicherung gegen falsche Eingaben. Folglich müssen einige Dinge beachtet werden, will man nicht das gesamte Programm zum Absturz bringen. Die Summe der Rechteckkoordinaten A+C darf den Wert 22 nicht übersteigen, ebenso gilt

B+D < 33. Für Innenfeldbefehle RIN, Z und SIN, Z muß Z < 11 sein. Keine Eingabe darf fehlen oder zuviel vorhanden sein, denn die spezielle Routine erwartet die genau vorgeschriebene Zahl von Eingaben. Ferner darf der Code zu printender (FIL) Zeichen nur aus 0 bis 63 und 128 bis 191 gewählt werden.

Eingabe des Programms

Das Programm benötigt eine REM-Zeile mit 570 Byte (erzielbare Zeichen). Diese erzeugt man mittels REM-Füller (CK 2/86) oder indem man 1 REM mit 186 Zeichen eintrifft (5 Zeilen ganz, die 6. mit 25 Zeichen) und anschließend durch Editieren und Ummunern 2 REM und 3 REM erzeugt. Nun folgen die Befehle POKE 16510,0; POKE 16511,60; POKE 16512,2; POKE 16514,118 und POKE 16515,118. Durch New-Line ist jetzt nur noch die Zeile 0 REM zu erkennen. Wer einen REM-Füller benutzt, sollte auf jeden Fall auch POKE 16510,0 als Editorschutz eingeben.

Danach tippt man Listing 1 ab und startet mit RUN. Sollte ein falscher Wert enthalten sein, meldet der Computer über die eingebauten Checksummen, in welchem Statement der Fehler sitzt. Ein weiterer Hinweis ist nützlich, da das Listing immer wieder auf 0 REM zu rückspringen wird. Hier bringt z.B. die Eingabe von LIST 10, gefolgt von POKE 16420,100, Abhilfe. Damit stabilisiert sich das Bild.

Initialisierung

Nach Eingabe des Listings und Einpoken des Maschinenprogramms durch RUN wird das Listing ab 10 nicht mehr benötigt und sollte gelöscht werden. Dafür benötigt man folgende Initialisierung, damit die 7 Basis-Befehle von "Respos" wirksam werden können:

```
10 LET INV=145
20 LET CLR=150
30 LET FIL=154
40 LET RILF=16968
50 LET SILF=16976
60 LET RTOT=16987
70 LET STOT=16998
80 LET RIN=17016
90 LET SIN=17031
100 LET EFF=17049
```


Demo-Programm

Das Demo-Programm setzt "RESPOS" in 0 REM voraus, enthält die Initialisierung und zeigt die Wirkung der eingebauten Routinen sowie Möglichkeiten, durch geeignete Programmierung neue Effekte zu erzielen. Innerhalb des Programms wird an verschiedenen Stellen darauf hingewiesen, welcher "Respos"-Befehl ange-

wandt wird bzw. ab welcher Programm-Nummer die gezeigte Routine läuft, so daß man schnellen Zugang zur eigenen Anwendung findet.

Friedrich W. Buckel

Gegen Einsendung von 10,- DM an F. Buckel, Schulstr. 15, 7184 Kirchberg/Jagst erhalten Sie ein kommentiertes Assembler-Listing sowie "Respos" und das Demo-Programm auf Cassette.

```

360 GOSUB 700
370 LET T$="1698328440CD3F41F13
D328440CD3F41CD8840C110E7C9CD14
118DA21DA"
380 LET C$=3596
390 GOSUB 700
400 LET T$="41E536E2CDC241E1363
FC9218941E536C1CD141E1364CC9D0FE
73A2D0FF"
410 LET C$=4148
420 GOSUB 700
430 LET T$="FD5CC520FF1C4550FC
DA70FC905042184440C5E5CD0242E1712
3C110F5CD"
440 LET C$=3893
450 GOSUB 700
460 LET T$="024221C140713E9A992
087CD0242219B4071C9D0024221E1487
1CD824221"
470 LET C$=2764
480 GOSUB 700
490 LET T$="8A4071C9CD1542CDC44
0CFFFC01542CD03942CDE240CFFFC0254
2CD3741CD"
500 LET C$=4150
510 GOSUB 700
520 LET T$="C748CFFFC02542CD0394
2CD37417B328640CDE540CFFFC02422
1844871CD"
530 LET C$=3837
540 GOSUB 700
550 LET T$="2542CD5941CFFFC0224
221844071C02542CD03942CD06E41CFFFC
002420600"
560 LET C$=3312
570 GOSUB 700
580 LET T$="21E142097E21AE4277C
D2542CD04042CD0841CFFFE8F5A1C2E37
5DA000000"
590 LET C$=3630
600 GOSUB 700
610 PRINT "RESPOS IST FERTIG,ZE
ILEN 10 BIS 780 LOESCHEN"
620 SLOW
630 STOP
700 LET S=0
710 FOR R=1 TO 80 STEP 2
720 POKE A,16+CODE T$(I)+CODE T
$(I+1)-476
730 LET S=S+PEEK A
740 LET A=A+1
750 NEXT I
760 LET N=N+1
770 IF S<3 THEN RETURN
780 PRINT "FEHLER IN STATEMENT
";10+N*30

```

Listing 1

EINGABE-PROGRAMM ZU RESPOS

NACH ERZEUGUNG DER ZEILE 0 REM:

```

10 REM EINGABE ZU RESPOS
F. W. BUCKEL
7184 KIRCHBERG/JAGST
15 FAST
20 LET A=16514
30 LET N=0
40 LET T$="757600000001101011
B7AB320FBC07EC68077C9AF329B40360
0C92A0C40"
50 LET C$=2656
60 GOSUB 700
70 LET T$="E04B6440C5AF8928071
12100471910FDC1AFB522832310FDE05
886430C90"
80 LET C$=3352
90 GOSUB 700
100 LET T$="8840C9640C9CD9D40A
FB8C8BAC5434A3E21915F1000C54123C
DC0410FA"
110 LET C$=3716
120 GOSUB 700
130 LET T$="19C110F4C900CD9D403
AE140473A8840A790903D32844010421
5AFB8C805"
140 LET C$=3429
150 GOSUB 700
160 LET T$="23CDBD4010FAD14310A
FB8C8D511210019CDBD4010F7D14215A
FB8C8D52B"
170 LET C$=3743
180 GOSUB 700
190 LET T$="CDBD4010FAD1431DAFB
8C8D5112100A7ED52CDBD4010F5D13A8
44088C818"
200 LET C$=3930
210 GOSUB 700
220 LET T$="8D2A0C401E161620C9A
F3285403E20328740CDD440CAF32844
035183285"
230 LET C$=2734
240 GOSUB 700
250 LET T$="4018F12184407E23774
F3E1691912377C80A2377C9D574118D
ACD5741CD"
260 LET C$=3169
270 GOSUB 700
280 LET T$="E240C9080B703D32844
8C5CD741CD0C448C8840C110EEC9060
B783D3284"
290 LET C$=3307
300 GOSUB 700
310 LET T$="40C8CD57413E0132E14
0CDE248C110EEC906203E01328740C5C
53E209032"
320 LET C$=3193
330 GOSUB 700
340 LET T$="8540CD4C41F13D32854
8C4C41C1C084010E7C906163E01328
648C5C53E"
350 LET C$=3279

```

Listing 2

```

0 REM (ENTHAELT RESP0S)
10 REM DEMO ZU RESP0S (C) 5/86
BY F.W.BUCKEL SCHULSTR.15
7184 KIRCHBERG/JAGST
20 LET INU=145
30 LET CLR=160
40 LET FIL=184
50 LET RILF=16968
60 LET SILF=16976
70 LET RTOT=16987
80 LET STOT=16996
90 LET RIN=17016
100 LET SIN=17031
110 LET EFF=17049
120 LET PAUSE=200
130 LET TASTE=230
140 GOTO 500
200 FOR I=1 TO 80
210 NEXT I
220 RETURN
230 PRINT AT 19,23;T$
240 IF INKEY$="" THEN GOTO 240
250 PRINT AT 19,23;L$( TO 8)
260 RETURN

```

```

270 GOSUB TASTE
280 CLS
290 RETURN
300 DIM L$(32)
310 DIM Z$(8)
311 LET Z(1)=128
312 LET Z(2)=169
313 LET Z(3)=6
314 LET Z(4)=8
315 LET Z(5)=61
316 LET Z(6)=23
317 LET Z(7)=20
318 LET Z(8)=27
320 LET T$="ARTE-3"
330 RETURN
500 POKE 16418,0
510 PRINT USR STOT,FIL,23,5,0
520 PRINT USR RIN,5,FIL,8
530 PRINT AT 7,8;"DEMO-PROGRAMM
4,8;"VON F. U. DUCKEL"
540 PRINT USR SIN,3,INU,0,3
550 GOSUB PAUSE
560 PRINT USR RTOT,INU
570 GOSUB PAUSE
580 GOSUB PAUSE
590 PRINT USR RIN,4,CLR
600 PRINT AT 6,8;"AT 6,8;"AT 8,8;"
"AT 10,8;"AT 12,8;"
610 GOSUB PAUSE
620 PRINT AT 6,10;"IHENEISES"
AT 8,10;"IRALARTIGES";AT 10,9;"R
GANISATIONS-";AT 12,9;"YSTEM"
630 GOSUB PAUSE
640 PRINT AT 14,6;"7 NEUE BASIC
BEFEHLE";AT 16,6;"ZUR BILDGESTA
TUNG:"
650 GOSUB PAUSE
660 PRINT USR EFF,7,INU,1
670 PRINT USR STOT,CLR,0,1
680 PRINT AT 0,0;"SO PRINTET DE
R ZX-81,"EIN RECHTECK:"
690 FOR I=1 TO 12
700 PRINT AT 4+I,0;"
710 NEXT I
720 PRINT AT 20,0;"UND SO MIT
E S E S"
730 GOSUB 270
740 PRINT AT 21,0;"
"1,8,BEFEHL";"PRINT USR RILF,A,
B,C,D,FIL,49";USR RILF,5,8,12,16
,FIL,49
750 GOSUB PAUSE
760 PRINT AT 0,0;"1";TAB 0;"2";
TAB 0;"3";TAB 0;"4";TAB 0;"5"
770 GOSUB PAUSE
780 PRINT AT 0,0;"123456789";T
AB 7;"";TAB 7;"";TAB 7;"";TAB
7;"
790 GOSUB PAUSE
800 FOR I=1 TO 9
810 PRINT AT 14,8;CHR$(156+I)
820 NEXT I
830 PRINT AT 14,8;"[";TAB 8;"[";
TAB 5;"[";
840 GOSUB PAUSE
850 FOR I=1 TO 9
860 PRINT AT 5,8;"
"AT 4,26;"
870 PRINT AT 17,13;"COD L=49"
880 GOSUB TASTE
890 PRINT USR EFF,3,CLR,6
900 PRINT AT 23,0;L$;AT 23,0;"W
EITERE BEISPIELE:"
910 PRINT USR RILF,0,0,10,10,FI
L$,9
920 PRINT USR RILF,5,10,14,21,F
IL,130
930 GOSUB PAUSE
940 PRINT USR RILF,4,4,12,12,IN
U
950 GOSUB PAUSE
960 PRINT USR RILF,2,0,1,31,FIL
,61
970 PRINT USR RILF,0,20,21,1,FI
L,51
980 GOSUB PAUSE
990 GOSUB PAUSE
1000 PRINT USR EFF,2,CLR,10
1010 FOR I=1 TO 11
1020 PRINT USR RILF,I,I,11,21,FI
L,I+30
1030 NEXT I
1040 PRINT USR RILF,12,12,9,19,C
L
1050 GOSUB PAUSE
1060 PRINT USR RILF,12,12,9,19,I
NU
1070 GOSUB PAUSE
1080 PRINT AT 22,0;"
"PRINT USR,"SILF,A,B,C,D,ARY,BORD
ER,SLOW"
1090 PRINT USR EFF,1,CLR,7
1100 GOSUB PAUSE
1110 PRINT USR SILF,2,5,10,12,FI
L,180,1,0
1120 PRINT USR SILF,13,1,8,30,FI
L,61,2,0
1130 FOR I=1 TO 7
1140 PRINT USR SILF,13,1,8,30,IN
U,1,INT(4-I/2)
1150 NEXT I
1160 GOSUB TASTE
1170 PRINT AT 22,0;L$;L$;AT 22,0
;"3, BEFEHL";TAB 0;"PRINT USR AT
OT,INU"
1175 GOSUB TASTE
1180 PRINT USR RTOT,INU
1190 GOSUB TASTE
1200 PRINT AT 22,0;"
"
1205 GOSUB TASTE
1210 PRINT USR STOT,FIL,21,4,0
1220 GOSUB TASTE
1230 PRINT AT 23,0;L$;AT 22,0;"5
, BEFEHL";"PRINT USR RIN,4,FIL,
8"
1240 GOSUB TASTE
1250 PRINT USR RIN,4,FIL,8
1260 GOSUB TASTE
1270 PRINT AT 23,0;L$;AT 22,0;"5
, BEFEHL";"PRINT USR RIN,4,FIL,
8"
1280 GOSUB TASTE
1290 FOR I=0 TO 7
1300 PRINT USR SIN,I,FIL,Z(I+1),
1,0
1310 NEXT I
1320 GOSUB PAUSE
1330 PRINT AT 22,0;L$;L$;AT 23,0
;"PRINT USR RIN,8,FIL,X"
1340 GOSUB PAUSE
1350 FOR X=0 TO 63
1360 PRINT USR RIN,8,FIL,X
1370 NEXT X
1380 FOR X=191 TO 128 STEP -1
1390 PRINT USR RIN,8,FIL,X
1400 NEXT X
1410 GOSUB PAUSE
1420 PRINT AT 22,0;L$;L$;AT 22,0
;"7, BEFEHL";"PRINT USR EFF,10
,ART,SLOW"
1430 GOSUB PAUSE
1440 FOR I=1 TO 5
1450 PRINT USR EFF,5,INU,1
1460 NEXT I
1470 PRINT USR EFF,1,INU,7
1480 PRINT USR EFF,2,INU,7
1490 PRINT USR EFF,3,INU,4
1500 PRINT USR EFF,2,INU,4
1510 PRINT USR EFF,6,CLR,5
1520 PRINT AT 22,0;L$;L$
1530 PRINT USR STOT,FIL,23,4,0
1540 PRINT AT 7,6;"ANWENDUNGSBEI
SPIELE";AT 10,15;"ZU";AT 13,11;"
"
1550 GOSUB PAUSE
1560 FOR I=1 TO 11
1560 PRINT USR RILF,0,0,I,5+I,IN

```

```

U
1678 PRINT USR RILF,22-I,0,I,5+I
1680 PRINT USR RILF,0,27-I,I,5+I
1682 PRINT USR RILF,22-I,27-I,I,
1684 PRINT USR RILF,22-I,27-I,I,
1686 PRINT USR RILF,22-I,27-I,I,
1688 PRINT USR RILF,22-I,27-I,I,
1690 PRINT USR RILF,22-I,27-I,I,
1692 PRINT USR RILF,22-I,27-I,I,
1694 PRINT USR RILF,22-I,27-I,I,
1696 PRINT USR RILF,22-I,27-I,I,
1698 PRINT USR RILF,22-I,27-I,I,
1700 PRINT USR RILF,22-I,27-I,I,
1702 PRINT USR RILF,22-I,27-I,I,
1704 PRINT USR RILF,22-I,27-I,I,
1706 PRINT USR RILF,22-I,27-I,I,
1708 PRINT USR RILF,22-I,27-I,I,
1710 PRINT USR RILF,22-I,27-I,I,
1712 PRINT USR RILF,22-I,27-I,I,
1714 PRINT USR RILF,22-I,27-I,I,
1716 PRINT USR RILF,22-I,27-I,I,
1718 PRINT USR RILF,22-I,27-I,I,
1720 PRINT USR RILF,22-I,27-I,I,
1722 PRINT USR RILF,22-I,27-I,I,
1724 PRINT USR RILF,22-I,27-I,I,
1726 PRINT USR RILF,22-I,27-I,I,
1728 PRINT USR RILF,22-I,27-I,I,
1730 PRINT USR RILF,22-I,27-I,I,
1732 PRINT USR RILF,22-I,27-I,I,
1734 PRINT USR RILF,22-I,27-I,I,
1736 PRINT USR RILF,22-I,27-I,I,
1738 PRINT USR RILF,22-I,27-I,I,
1740 PRINT USR RILF,22-I,27-I,I,
1742 PRINT USR RILF,22-I,27-I,I,
1744 PRINT USR RILF,22-I,27-I,I,
1746 PRINT USR RILF,22-I,27-I,I,
1748 PRINT USR RILF,22-I,27-I,I,
1750 PRINT USR RILF,22-I,27-I,I,
1752 PRINT USR RILF,22-I,27-I,I,
1754 PRINT USR RILF,22-I,27-I,I,
1756 PRINT USR RILF,22-I,27-I,I,
1758 PRINT USR RILF,22-I,27-I,I,
1760 PRINT USR RILF,22-I,27-I,I,
1762 PRINT USR RILF,22-I,27-I,I,
1764 PRINT USR RILF,22-I,27-I,I,
1766 PRINT USR RILF,22-I,27-I,I,
1768 PRINT USR RILF,22-I,27-I,I,
1770 PRINT USR RILF,22-I,27-I,I,
1772 PRINT USR RILF,22-I,27-I,I,
1774 PRINT USR RILF,22-I,27-I,I,
1776 PRINT USR RILF,22-I,27-I,I,
1778 PRINT USR RILF,22-I,27-I,I,
1780 PRINT USR RILF,22-I,27-I,I,
1782 PRINT USR RILF,22-I,27-I,I,
1784 PRINT USR RILF,22-I,27-I,I,
1786 PRINT USR RILF,22-I,27-I,I,
1788 PRINT USR RILF,22-I,27-I,I,
1790 PRINT USR RILF,22-I,27-I,I,
1792 PRINT USR RILF,22-I,27-I,I,
1794 PRINT USR RILF,22-I,27-I,I,
1796 PRINT USR RILF,22-I,27-I,I,
1798 PRINT USR RILF,22-I,27-I,I,
1800 PRINT USR RILF,22-I,27-I,I,
1802 PRINT USR RILF,22-I,27-I,I,
1804 PRINT USR RILF,22-I,27-I,I,
1806 PRINT USR RILF,22-I,27-I,I,
1808 PRINT USR RILF,22-I,27-I,I,
1810 PRINT USR RILF,22-I,27-I,I,
1812 PRINT USR RILF,22-I,27-I,I,
1814 PRINT USR RILF,22-I,27-I,I,
1816 PRINT USR RILF,22-I,27-I,I,
1818 PRINT USR RILF,22-I,27-I,I,
1820 PRINT USR RILF,22-I,27-I,I,
1822 PRINT USR RILF,22-I,27-I,I,
1824 PRINT USR RILF,22-I,27-I,I,
1826 PRINT USR RILF,22-I,27-I,I,
1828 PRINT USR RILF,22-I,27-I,I,
1830 PRINT USR RILF,22-I,27-I,I,
1832 PRINT USR RILF,22-I,27-I,I,
1834 PRINT USR RILF,22-I,27-I,I,
1836 PRINT USR RILF,22-I,27-I,I,
1838 PRINT USR RILF,22-I,27-I,I,
1840 PRINT USR RILF,22-I,27-I,I,
1842 PRINT USR RILF,22-I,27-I,I,
1844 PRINT USR RILF,22-I,27-I,I,
1846 PRINT USR RILF,22-I,27-I,I,
1848 PRINT USR RILF,22-I,27-I,I,
1850 PRINT USR RILF,22-I,27-I,I,
1852 PRINT USR RILF,22-I,27-I,I,
1854 PRINT USR RILF,22-I,27-I,I,
1856 PRINT USR RILF,22-I,27-I,I,
1858 PRINT USR RILF,22-I,27-I,I,
1860 PRINT USR RILF,22-I,27-I,I,
1862 PRINT USR RILF,22-I,27-I,I,
1864 PRINT USR RILF,22-I,27-I,I,
1866 PRINT USR RILF,22-I,27-I,I,
1868 PRINT USR RILF,22-I,27-I,I,
1870 PRINT USR RILF,22-I,27-I,I,
1872 PRINT USR RILF,22-I,27-I,I,
1874 PRINT USR RILF,22-I,27-I,I,
1876 PRINT USR RILF,22-I,27-I,I,
1878 PRINT USR RILF,22-I,27-I,I,
1880 PRINT USR RILF,22-I,27-I,I,
1882 PRINT USR RILF,22-I,27-I,I,
1884 PRINT USR RILF,22-I,27-I,I,
1886 PRINT USR RILF,22-I,27-I,I,
1888 PRINT USR RILF,22-I,27-I,I,
1890 PRINT USR RILF,22-I,27-I,I,
1892 PRINT USR RILF,22-I,27-I,I,
1894 PRINT USR RILF,22-I,27-I,I,
1896 PRINT USR RILF,22-I,27-I,I,
1898 PRINT USR RILF,22-I,27-I,I,
1900 PRINT USR RILF,22-I,27-I,I,
1902 PRINT USR RILF,22-I,27-I,I,
1904 PRINT USR RILF,22-I,27-I,I,
1906 PRINT USR RILF,22-I,27-I,I,
1908 PRINT USR RILF,22-I,27-I,I,
1910 PRINT USR RILF,22-I,27-I,I,
1912 PRINT USR RILF,22-I,27-I,I,
1914 PRINT USR RILF,22-I,27-I,I,
1916 PRINT USR RILF,22-I,27-I,I,
1918 PRINT USR RILF,22-I,27-I,I,
1920 PRINT USR RILF,22-I,27-I,I,
1922 PRINT USR RILF,22-I,27-I,I,
1924 PRINT USR RILF,22-I,27-I,I,
1926 PRINT USR RILF,22-I,27-I,I,
1928 PRINT USR RILF,22-I,27-I,I,
1930 PRINT USR RILF,22-I,27-I,I,
1932 PRINT USR RILF,22-I,27-I,I,
1934 PRINT USR RILF,22-I,27-I,I,
1936 PRINT USR RILF,22-I,27-I,I,
1938 PRINT USR RILF,22-I,27-I,I,
1940 PRINT USR RILF,22-I,27-I,I,
1942 PRINT USR RILF,22-I,27-I,I,
1944 PRINT USR RILF,22-I,27-I,I,
1946 PRINT USR RILF,22-I,27-I,I,
1948 PRINT USR RILF,22-I,27-I,I,
1950 PRINT USR RILF,22-I,27-I,I,
1952 PRINT USR RILF,22-I,27-I,I,
1954 PRINT USR RILF,22-I,27-I,I,
1956 PRINT USR RILF,22-I,27-I,I,
1958 PRINT USR RILF,22-I,27-I,I,
1960 PRINT USR RILF,22-I,27-I,I,
1962 PRINT USR RILF,22-I,27-I,I,
1964 PRINT USR RILF,22-I,27-I,I,
1966 PRINT USR RILF,22-I,27-I,I,
1968 PRINT USR RILF,22-I,27-I,I,
1970 PRINT USR RILF,22-I,27-I,I,
1972 PRINT USR RILF,22-I,27-I,I,
1974 PRINT USR RILF,22-I,27-I,I,
1976 PRINT USR RILF,22-I,27-I,I,
1978 PRINT USR RILF,22-I,27-I,I,
1980 PRINT USR RILF,22-I,27-I,I,
1982 PRINT USR RILF,22-I,27-I,I,
1984 PRINT USR RILF,22-I,27-I,I,
1986 PRINT USR RILF,22-I,27-I,I,
1988 PRINT USR RILF,22-I,27-I,I,
1990 PRINT USR RILF,22-I,27-I,I,
1992 PRINT USR RILF,22-I,27-I,I,
1994 PRINT USR RILF,22-I,27-I,I,
1996 PRINT USR RILF,22-I,27-I,I,
1998 PRINT USR RILF,22-I,27-I,I,
1999 PRINT USR RILF,22-I,27-I,I,
2000 NEXT I
2002 GOSUB PAUSE
2004 PRINT AT 23,0;L6;AT 23,0;"I
NUMERISIONEFFEKTE AB 2020"
2006 PRINT USR STOT,FIL,0,0,1
2008 PRINT USR STOT,INU,0,1
2010 PRINT USR STOT,INU,0,1
2012 PRINT USR STOT,INU,0,1
2014 PRINT USR STOT,INU,0,1
2016 PRINT USR STOT,INU,0,1
2018 PRINT USR STOT,INU,0,1
2020 PRINT USR STOT,INU,0,1
2022 PRINT USR STOT,INU,0,1
2024 PRINT USR STOT,INU,0,1
2026 PRINT USR STOT,INU,0,1
2028 PRINT USR STOT,INU,0,1
2030 PRINT USR STOT,INU,0,1
2032 PRINT USR STOT,INU,0,1
2034 PRINT USR STOT,INU,0,1
2036 PRINT USR STOT,INU,0,1
2038 PRINT USR STOT,INU,0,1
2040 PRINT USR STOT,INU,0,1
2042 PRINT USR STOT,INU,0,1
2044 PRINT USR STOT,INU,0,1
2046 PRINT USR STOT,INU,0,1
2048 PRINT USR STOT,INU,0,1
2050 PRINT USR STOT,INU,0,1
2052 PRINT USR STOT,INU,0,1
2054 PRINT USR STOT,INU,0,1
2056 PRINT USR STOT,INU,0,1
2058 PRINT USR STOT,INU,0,1
2060 PRINT USR STOT,INU,0,1
2062 PRINT USR STOT,INU,0,1
2064 PRINT USR STOT,INU,0,1
2066 PRINT USR STOT,INU,0,1
2068 PRINT USR STOT,INU,0,1
2070 PRINT USR STOT,INU,0,1
2072 PRINT USR STOT,INU,0,1
2074 PRINT USR STOT,INU,0,1
2076 PRINT USR STOT,INU,0,1
2078 PRINT USR STOT,INU,0,1
2080 PRINT USR STOT,INU,0,1
2082 PRINT USR STOT,INU,0,1
2084 PRINT USR STOT,INU,0,1
2086 PRINT USR STOT,INU,0,1
2088 PRINT USR STOT,INU,0,1
2090 PRINT USR STOT,INU,0,1
2092 PRINT USR STOT,INU,0,1
2094 PRINT USR STOT,INU,0,1
2096 PRINT USR STOT,INU,0,1
2098 PRINT USR STOT,INU,0,1
2099 PRINT USR STOT,INU,0,1

```

```

2099 PRINT USR STOT,INU,0,1
2100 PRINT USR STOT,INU,0,1
2102 PRINT USR STOT,INU,0,1
2104 PRINT USR STOT,INU,0,1
2106 PRINT USR STOT,INU,0,1
2108 PRINT USR STOT,INU,0,1
2110 PRINT USR STOT,INU,0,1
2112 PRINT USR STOT,INU,0,1
2114 PRINT USR STOT,INU,0,1
2116 PRINT USR STOT,INU,0,1
2118 PRINT USR STOT,INU,0,1
2120 PRINT USR STOT,INU,0,1
2122 PRINT USR STOT,INU,0,1
2124 PRINT USR STOT,INU,0,1
2126 PRINT USR STOT,INU,0,1
2128 PRINT USR STOT,INU,0,1
2130 PRINT USR STOT,INU,0,1
2132 PRINT USR STOT,INU,0,1
2134 PRINT USR STOT,INU,0,1
2136 PRINT USR STOT,INU,0,1
2138 PRINT USR STOT,INU,0,1
2140 PRINT USR STOT,INU,0,1
2142 PRINT USR STOT,INU,0,1
2144 PRINT USR STOT,INU,0,1
2146 PRINT USR STOT,INU,0,1
2148 PRINT USR STOT,INU,0,1
2150 PRINT USR STOT,INU,0,1
2152 PRINT USR STOT,INU,0,1
2154 PRINT USR STOT,INU,0,1
2156 PRINT USR STOT,INU,0,1
2158 PRINT USR STOT,INU,0,1
2160 PRINT USR STOT,INU,0,1
2162 PRINT USR STOT,INU,0,1
2164 PRINT USR STOT,INU,0,1
2166 PRINT USR STOT,INU,0,1
2168 PRINT USR STOT,INU,0,1
2170 PRINT USR STOT,INU,0,1
2172 PRINT USR STOT,INU,0,1
2174 PRINT USR STOT,INU,0,1
2176 PRINT USR STOT,INU,0,1
2178 PRINT USR STOT,INU,0,1
2180 PRINT USR STOT,INU,0,1
2182 PRINT USR STOT,INU,0,1
2184 PRINT USR STOT,INU,0,1
2186 PRINT USR STOT,INU,0,1
2188 PRINT USR STOT,INU,0,1
2190 PRINT USR STOT,INU,0,1
2192 PRINT USR STOT,INU,0,1
2194 PRINT USR STOT,INU,0,1
2196 PRINT USR STOT,INU,0,1
2198 PRINT USR STOT,INU,0,1
2199 PRINT USR STOT,INU,0,1
2200 PRINT USR STOT,INU,0,1
2202 PRINT USR STOT,INU,0,1
2204 PRINT USR STOT,INU,0,1
2206 PRINT USR STOT,INU,0,1
2208 PRINT USR STOT,INU,0,1
2210 PRINT USR STOT,INU,0,1
2212 PRINT USR STOT,INU,0,1
2214 PRINT USR STOT,INU,0,1
2216 PRINT USR STOT,INU,0,1
2218 PRINT USR STOT,INU,0,1
2220 PRINT USR STOT,INU,0,1
2222 PRINT USR STOT,INU,0,1
2224 PRINT USR STOT,INU,0,1
2226 PRINT USR STOT,INU,0,1
2228 PRINT USR STOT,INU,0,1
2230 PRINT USR STOT,INU,0,1
2232 PRINT USR STOT,INU,0,1
2234 PRINT USR STOT,INU,0,1
2236 PRINT USR STOT,INU,0,1
2238 PRINT USR STOT,INU,0,1
2240 PRINT USR STOT,INU,0,1
2242 PRINT USR STOT,INU,0,1
2244 PRINT USR STOT,INU,0,1
2246 PRINT USR STOT,INU,0,1
2248 PRINT USR STOT,INU,0,1
2250 PRINT USR STOT,INU,0,1
2252 PRINT USR STOT,INU,0,1
2254 PRINT USR STOT,INU,0,1
2256 PRINT USR STOT,INU,0,1
2258 PRINT USR STOT,INU,0,1
2260 PRINT USR STOT,INU,0,1
2262 PRINT USR STOT,INU,0,1
2264 PRINT USR STOT,INU,0,1
2266 PRINT USR STOT,INU,0,1
2268 PRINT USR STOT,INU,0,1
2270 PRINT USR STOT,INU,0,1
2272 PRINT USR STOT,INU,0,1
2274 PRINT USR STOT,INU,0,1
2276 PRINT USR STOT,INU,0,1
2278 PRINT USR STOT,INU,0,1
2280 PRINT USR STOT,INU,0,1
2282 PRINT USR STOT,INU,0,1
2284 PRINT USR STOT,INU,0,1
2286 PRINT USR STOT,INU,0,1
2288 PRINT USR STOT,INU,0,1
2290 PRINT USR STOT,INU,0,1
2292 PRINT USR STOT,INU,0,1
2294 PRINT USR STOT,INU,0,1
2296 PRINT USR STOT,INU,0,1
2298 PRINT USR STOT,INU,0,1
2299 PRINT USR STOT,INU,0,1

```

Spindizzy

Zunächst gab es "Marble Madness", ein Programm, das besonders in Spielhallen ein Riesenhit wurde. Dann kamen die Adaptionen und Kopien für Heimcomputer. Auch "Spindizzy" fällt in diese Kategorie, ebenso wie z.B. "Gyroscope" und vielleicht "Quazatron".

Das Programm zeigt einige Hundert aneinandergrenzende Ebenen, die eine fremde Welt darstellen sollen. Der Spieler muß mit einem Spezialgerät, das wie ein Kreisel aussieht, auf Erkundung gehen und wertvolle Edelsteine einsammeln. Leider kann das Gerät nur auf den

vorgegebenen quadratischen Flächen bewegt werden. Weicht es davon ab, ist es verloren. Gerade hier in der Steuerung liegt das größte Problem für den Spieler, denn durch die vielen Kurven und Hindernisse muß er sich voll konzentrieren und lange üben.

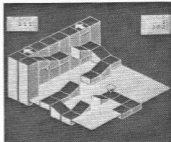
Erschwerend kommt hinzu, daß kein Joystick verwendet werden kann; die Steuerung muß über die Tastatur erfolgen. Unberührt läuft auch noch die Zeit ab, die man aber durch jeden gesammelten Edelstein wieder etwas verlängern kann. Zur Orientierung in diesem La-

abyrinth dient eine jederzeit abrufbereite Karte. Auch kann das Spiel eingefroren werden, wenn man eine Pause einlegen will.

"Spindizzy" mit seiner hervorragenden 3-D-Grafik und

den Bewegungen des Kreisels ist den Programmierern wirklich gut gelungen. Man muß es sich einfach ansehen.

System: Spectrum 48 K
 Hersteller: Electric Dreams
 Stephan König



Spindizzy
 Eine fremde
 Welt in
 3-D-Grafik

Indexomat für den QL

Eine der vielen praktischen Eigenschaften des QL ist, daß er automatisch auf Cartridge oder Floppy ein Programm mit Namen "boot" sucht, es lädt und startet. Es wäre also nützlich, wenn nach Einlegen von Disk oder Cartridge ein durchnumeriertes Verzeichnis der Programme erschiene und man zum Laden nur noch die gewünschte Programmnummer eingeben müßte.

Dies ist die Aufgabe von "Indexomat". Da der QL über mehrere Lademöglichkeiten verfügt, wird das Programm auf Extensions untersucht; das sind drei Buchstaben, die dem Programm mittels Strich (wie beim Unterstreichen üblich) angefügt werden. Die Extensions entsprechen voll der von Psion Software und dem Sinclair Club Leverkusen e. V. empfohlenen Normung.

Nachfolgend eine Liste der verwendeten Extensions:

_bas für Basic-Programme
_exe für EXEC_W-fähige Programme

_scr für abgespeicherte Bilder (Mode 8)

_cde für abgespeicherten Maschinencode (kann nicht vom "Indexomat" geladen werden) Hinzu kommen die Extensions der Psion-Software.

Wenn kein Extension gefunden wird, ruft man das Programm mit LRUN auf.

Bevor "Indexomat" verwendet werden kann, benötigt die Cartridge einen Header mit dem Namen inf100. Er wird mit folgenden Befehlen erzeugt:

```
OPEN_NEW #5, mdv1.inf
100: DIR #5, mdv1.CLOSE
#5
```

"Indexomat" ist auch für Disketten vorgesehen; man muß nur device\$="mdv1_" in device\$="fdkl_" ändern. Sollte auf der Cartridge oder Diskette nachträglich noch ein Programm eingefügt werden, so ist der Header nochmals zu erzeugen. Der Aufruf eines nicht vorhandenen Programms wird mit NO FILE quittiert.

Jürgen Maßberg

Indexomat

```
5 MODE 4
10 device$="mdv1_"
15 DEFINE PROCEDURE awindow:OPEN
#10,scr
20 WINDOW#10,465,205,30,10:CLS#1
0:PAPER#10,5:FOR scr=1 TO 100 ST
EP 2:BORDER#10,scr,2:NEXT scr:OP
EN#10,scr:WINDOW#10,465,205,30,1
0:BORDER#10,2,2:PAPER#10,4:INK#1
0,0:CLS#10
25 END DEFINE
30 DEFINE PROCEDURE r$X:CSIZE#0,
1,0:REPEAT in:AT#0,1,0:INPUT#0,"
File ? ",number$:IF number$<0"
THEN number=number$+2: EXIT in
35 IF "_scr" INSTR file$(number)
THEN MODE 8: LBYTES device$&fil
e$(number),131072:PAUSE:MODE 4:a
window:directory:r$X
40 IF "_bas"INSTR file$(number)
THEN LRUN device$&file$(number)
45 IF "_cde" INSTR file$(number)
THEN CLS#0:CSIZE#0,2,1:PRINT#0,
"Can't load it":PAUSE:CSIZE#0,0,
0:CLS#0:r$X
```

```
50 IF "_exe" INSTR file$(number)
THEN CLS: EXEC_W device$&file$(
number)
55 IF "_doc" INSTR file$(number)
THEN CLS#0:CSIZE#0,2,1:PRINT#0,
"Can't load PSION dates. Use GUI
LL":PAUSE:CSIZE#0,0,0:CLS#0:r$X
60 IF "_aba" INSTR file$(number)
THEN CLS#0:CSIZE#0,2,1:PRINT#0,
"Can't load PSION dates. Use ABA
CUS":PAUSE:CSIZE#0,0,0:CLS#0:r$X
65 IF "_grf" INSTR file$(number)
THEN CLS#0:CSIZE#0,2,1:PRINT#0,
"Can't load PSION dates. Use EAS
EL":PAUSE:CSIZE#0,0,0:CLS#0:r$X
70 IF "_dbf" INSTR file$(number)
THEN CLS#0:CSIZE#0,2,1:PRINT#0,
"Can't load PSION dates. Use ARC
HIVE":PAUSE:CSIZE#0,0,0:CLS#0:r$
X
75 IF "inf100"=file$(number) THE
N CLS#0:CSIZE#0,2,1:PRINT#0,"Hea
der info already read":PAUSE:CSI
ZE#0,0,0:CLS#0:r$X
80 IF LEN(file$(number))=0 THEN
CSIZE#0,2,1:PRINT#0,"No file":PA
USE:CSIZE#0,0,0:CLS#0:r$X
85 LRUN device$&file$(number):EN
D IF :END DEFINE
90 END IF
95 DEFINE PROCEDURE read_inf100
100 DIM file$(77,15)
105 OPEN_IN#5,device$&"inf100"
110 FOR info=0 TO EOF(#5)
115 FOR nui=1 TO 77
120 ios=nui
125 INPUT#5,file$(nui)
130 IF EOF(#5) THEN LET nui=100:
END IF
135 END FOR nui:END FOR info:CLO
SE#5
140 END DEFINE
145 DEFINE PROCEDURE directory
150 CURSOR#10,100,1:CSIZE#10,1,0
:UNDER#10,1:PRINT#10,"Mediename
="&file$(1)(1 TO 10):CSIZE#10,0,
0:CURSOR#10,0,25:UNDER#10,0
155 FOR filenum=1 TO 21
160 filenum2=filenum-2
165 AT#10,filenum2,1:PRINT#10,fil
enum2%."&file$(filenum)
170 END FOR filenum
175 FOR filenum=22 TO 40
180 filenum2=filenum-2
185 tab=filenum-21
190 AT#10,tab,19:PRINT#10,filenu
m2%."&file$(filenum)
195 END FOR filenum
200 FOR filenum=41 TO 59
```

```

205 filenum2=filenum-2
210 tab=filenum-40
215 AT#10,tab,39:PRINT#10,filenu
m2%."&file$(filenum)
220 END FOR filenum
225 FOR filenum=59 TO 77
230 filenum2=filenum-2
235 tab=filenum-58
240 AT#10,tab,58:PRINT#10,filenu
m2%."&file$(filenum)
245 END FOR filenum
250 END DEFINE
255 awindowread_inf10:director
Y:PSX

```

Computergrafische Experimente mit Pascal

Von Karl-Heinz Becker
und Michael Dörfler
Vieweg-Verlag
320 Seiten, 42.-DM
ISBN 3-528-04461-6

"Apfelmännchen", "Fraktale", "Mandelbrot", "Feigenbäume", "Julia-Mengen" - dies alles sind Begriffe, die durch Computerzeitschriften und Computerclubs geistern. Fast jeder Computerfan hat ein "Apfelmännchen"-Programm, doch ist den wenigsten klar, daß das Ganze nicht bloß aus schönen, bizarren Grafiken besteht, sondern gestandene Mathematik und Gegenstand aktueller Forschung in der Physik ist. Um so wichtiger ist ein Buch, das in die Materie einführt, die Sprache der Computerfans erläutert und zu eigenen Experimenten anregt.

Der Diplom-Informatiker Karl-Heinz Becker und der Diplom-Physiker Michael Dörfler, beide in der Lehrerfortbildung tätig, haben das vorliegende Buch über "Chaos und Ordnung in dynamischen Systemen" geschrieben. Dem ernsthaft interessierten Heimcomputerbesitzer ist das Buch sehr zu empfehlen. Es bringt all die Hintergrundinformationen, die man z.B. zum Verständnis des Programms "Mandelbrot" aus der CK benötigt.

Das Buch gliedert sich in drei Teile mit den Überschriften: Forscher entdecken das Chaos,

Bausteine für grafische Experimente, Pascal und die Apfelmännchen. Den vierten Teil bildet der Anhang. Hier werden Computersystem und Pascal-Dialekt, mit denen gearbeitet wurde, erklärt. Dieser Teil enthält nützliche Tipps zum Umschreiben der Programme in den eigenen Pascal-Dialekt.

Bei der Auswahl von Rechner (Apple) und Sprache (UCSD Pascal) haben die Autoren nicht sehr verbreitete Systeme ausgesucht. Aber zum Glück ist Pascal soweit standardisiert, daß die Anpassung an den eigenen Rechner keine Schwierigkeiten bereiten dürfte. Viele Programme sind auch in einer Art Metasprache formuliert, so daß der Algorithmus klar wird, das Ganze im Grunde jedoch von der Sprache unabhängig ist. Die Grafik-Routinen bedürfen aber in jedem Fall einer Anpassung, was nicht schwierig sein dürfte, da jeder Computerbesitzer ja wahrscheinlich die Grafik seines Rechners kennt.

Von den Programmen darf man allerdings keine hohen Geschwindigkeiten erwarten. Es liegt in der Natur dieser Probleme, daß sie sehr viel Rechenzeit benötigen, und der Spectrum ist schließlich kein Cray. Dennoch ist es erstaunlich, wie schnell dieser kleine Sinclair-Rechner mit dem Hisoft-Pascal rechnet. Ein Umsetzen der Programme in Basic sollten nur sehr gedul-

dige Menschen vornehmen, da sich die Laufzeiten dann immer im Bereich von Stunden bewegen (Feigenbaum auf Spectrum: ca. 55 Minuten in Basic, 1,5 Minuten in Pascal). Wer sich also mit dem Buch beschäftigen will, sollte die Anschaffung eines Pascal-Compilers nicht scheuen.

Das Angebot auf der dritten Umschlagseite sollte man ignorieren. Zehn Bilder aus dem Buch auf Diskette bzw. als Kopie eines Ausdrucks mit dem Laserdrucker für 200.-DM sind einfach zu teuer. Für das Geld könnte man großformatige Farbposter erwarten. Doch wer das Buch kauft, in dem ausreichend schöne Bilder vorhanden sind, muß das Angebot ja nicht

wahrnehmen. Ich empfehle, die Ausstellungskataloge der Forschungsgruppe "Komplex Dynamik" an der Universität Bremen zu bestellen. Anschrift und Details auf Seite 315 im Buch.

Ich kann das Buch jedem empfehlen, der mit seinem Computer mehr machen will als nur Ballerspiele oder "Pac Man". Man lernt etwas, das sicher auch in der Schule zu verwenden ist. Gerade die experimentelle Mathematik auf dem Heimcomputer kann vielleicht dazu beitragen, die (völlig unverständliche) Abneigung gegen Mathematik zu überwinden. Schwierig ist die in diesem Buch benutzte Mathematik nicht.

Rainer W. Gerling

Der Sinclair unter Kontrolle

Verlag Gigasoft
192 Seiten DIN A4
69.- DM

Dieses Buch ist eine Neuheit für den QL-User. Schlägt man es auf, so stoßt man auf ein sehr gut und übersichtlich gegliedertes Inhaltsverzeichnis. Schon Kapitel 1 zeigt, für wen das Buch gedacht ist. Es ist dies der Kenner der Maschinensprache. Auf eigenen Seiten wird kurz und knapp der Prozessor 68000/68008 erläutert. Diese Darstellung ist übersichtlich und informativ, doch eigentlich überflüssig, denn wer Maschinensprache kennt, bedarf einer solchen Einführung nicht.

Mit dem 2. Kapitel beginnt der eigentliche Wert des Buches. Hier wird eine Einführung in das QDOS-Betriebssystem des Sinclair-QL behandelt. Wichtige Punkte sind dabei das Multitasking und die Aufteilung des QL-Speichers.

Kapitel 3 erläutert die sogenannten "Traps" und deren Aufgaben. (Traps sind die Routinen des Betriebssystemes.)

In Kapitel 4 folgt dann endlich eine genaue Einzelbeschreibung der Arbeitsweise der Traps. Es werden die Aufrufparameter behandelt. Anschließend folgen Angaben, welche Register angesprochen werden.

Vektorisierte Hilfsprogramme beschreibet der Autor in Kapitel 5. Vektorisiert bedeutet, daß die Adressen der Hilfsprogramme in einer Tabelle stehen. Zur Anwendung muß der Benutzer dann nur noch den Standort in der Tabelle kennen. Dies folgt in Kapitel 6. Auch werden diese Hilfsprogramme hier noch genauer erläutert.

Kapitel 7 verläßt dann die Theorie und stellt 2 Basic-Programme und 2 Maschinenprogramme vor. "Trapper" in Basic geschrieben, ist eine kleine Einführung in die Maschinensprache. Sie ermöglicht die Eingabe von MC-Routinen und zeigt deren Auswirkung auf die Register.

Die restlichen Kapitel befassen sich mit dem maschinennahen Standort von Super-Basic und dem Microdrive.

Zusammenfassend läßt sich zu dem auf den ersten Blick etwas teuer erscheinenden Buch sagen, daß es viel Informationen bietet. Außerdem ist zu bedenken, daß dieses Buch nur in einer kleinen Auflage erscheinen kann und die Herstellungskosten erheblich sind.

M. L. Stinner

Das Buch kann über den CK-Verlag bezogen werden.

Microsoft-Basic auf dem QL

Für den CPM-Rechner und auch für IBM-PCs oder Kompatible gibt es sehr viele Software. Erstaunlich ist, wieviele Programme davon in Basic geschrieben sind. Insbesondere in den USA gibt es viele Basic-Spiele für den IBM-PC. Auch eines der besten Public-Domain-DFU-Programme, PC-TALK, ist in Basic geschrieben. Ich kenne auch einen guten 8086-Assembler (CHASM), der in Basic geschrieben ist. Eine Anpassung dieser Programme an den eigenen Rechner ist die Arbeit wert, da die Programmquellen meistens kostenlos sind. Auch CPM-Basic-Programme erleben gerade durch die Schneider-Computer und den Commodore 128 einen zweiten Frühling.

Der QL ist für eine solche Anpassung optimal geeignet. Mit der Möglichkeit, Prozeduren und Funktionen zu definieren, kann man die häufigsten Basic-Befehle aus dem Microsoft-Basic simulieren. Das Listing zeigt die Wichtigsten. Die Grafik- und Sound-Befehle müssen von Hand individuell angepaßt werden. Folgende Vorarbeit sollte jedoch schon gemacht sein:

1. Alle TABs müssen durch TOs ersetzt werden.
2. Alle Hochkommas, die Kommentare abgrenzen, müssen durch REMs ersetzt werden.
3. Falls es fehlt, ist nach allen THEN-Befehlen das GOTO einzufügen.
4. Nach allen NEXT-Befehlen müssen zusätzlich die Namen der Schleifenvariablen angefügt werden.
5. Vor allen ELSE-Befehlen ist ein Doppelpunkt einzufügen.

Drei Konstruktionen kann man nur von Hand ändern. 1. Die Konstruktion

```
10 WHILE bedingung
20 .....
30 wend
```

muß durch

```
10 REPEAT name
20 IF bedingung THEN EXIT
name
30 ....
40 END REPEAT name
```

ersetzt werden. 2. Wenn bei einer FOR-NEXT-Schleife nach dem FOR-Statement schon der erste Befehl der Schleife folgt, dann muß der Befehl in eine neue Zeile. Andernfalls gibt es Probleme mit der Kurzform der FOR-NEXT-Schleife (siehe QL-Handbuch). Ein Beispiel:

```
10 FOR i = 1 TO 10: a=b+45:
....
20 ....
30 NEXT i
muß durch
10 FOR i=1 TO 10
11 a=b+45: ....
20 ....
30 NEXT i
```

ersetzt werden. 3. Alle Abfragen mit Zeichen wie z.B. "IF a\$ >= "a" THEN" müssen durch den ASCII-Code ersetzt werden. Aus obigem Beispiel ergibt sich deshalb: IF CODE (a\$) >= 97 THEN. Das liegt an der Sortierreihenfolge des QL. Es werden nicht die ASCII-Codes verglichen, sondern die Zeichen in der Reihenfolge "AaBbCc" usw. (siehe Handbuch, Begriffe/Stringvergleich, Seite 48 bzw. 52).

Im Listing sind die wichtigsten Befehle umgesetzt, die man für einfache Befehle benötigt. Die Funktion INSTR (a\$, b\$) berücksichtigt keine Groß- und Kleinschreibung. Außerdem überdecken einige Funktionen/Prozeduren nicht die gesamte Breite der Möglichkeiten. Es sind manchmal nur die wichtigsten Parameter angegeben. Das ist aber dann diejenige Form, die am häufigsten benutzt wird.

Es ist sicherlich möglich, noch Verbesserungen vorzunehmen. Vielleicht fehlt auch der eine oder andere Befehl. Das vorgestellte Listing soll nur den Weg zeigen und ein Anfang sein.

Rainer W. Gerling

```
1 REMark Anpassung QL SuperBASIC an
2 REMark Microsoft Basic
3 REMark (c) 1986 by Rainer W. Gerling
4 REMark Version 1.1
5 REMark *****
6 DEFINE FuNction LEFT$(a$,b)
7 IF b=0 THEN RETURN ""
8 IF b>LEN(a$) THEN RETURN a$
9 RETURN a$(1 TO b)
10 END DEFINE
11 REMark *****
12 DEFINE FuNction RIGHT$(a$,b)
13 IF b=0 THEN RETURN ""
14 IF b>LEN(a$) THEN RETURN a$
15 RETURN a$(LEN(a$)-b+1 TO)
16 END DEFINE
17 REMark *****
18 DEFINE FuNction INSTR(a$,b$)
19 RETURN b$ INSTR a$
20 END DEFINE
21 REMark *****
22 DEFINE FuNction MID$(a$,A,b)
23 IF b=0 THEN RETURN ""
24 IF A>LEN(a$) THEN RETURN ""
25 RETURN a$(A TO A+b-1)
26 END DEFINE
27 REMark *****
28 DEFINE PROCEDURE COLOR (A,b)
29 INK A: PAPER b
30 END DEFINE
31 REMark *****
32 DEFINE PROCEDURE PLAY(a$)
33 REMark Ist nur eine primitive Lösung
34 BEEP 20000,10,100,1000,10,5
35 END DEFINE
36 REMark *****
37 DEFINE FuNction VAL(X$)
38 RETURN X$
39 END DEFINE
40 REMark *****
41 DEFINE FuNction STR$(X)
42 RETURN X
43 END DEFINE
44 REMark *****
45 DEFINE FuNction SGN(X)
46 IF X>0 THEN RETURN 1: ELSE IF X=0 THEN
  N RETURN 0: ELSE RETURN -1
47 END DEFINE
48 REMark *****
49 DEFINE FuNction asc(X)
50 RETURN CODE(X)
51 END DEFINE
52 REMark *****
53 DEFINE PROCEDURE files(a$)
54 DIR a$
55 END DEFINE
56 REMark *****
57 DEFINE FuNction space$(X)
58 RETURN FILL$(" ",X)
59 END DEFINE
60 REMark *****
61 DEFINE FuNction string$(X,a$)
62 RETURN FILL$(a$(1),X)
63 END DEFINE
64 REMark *****
65 REMark *****
```

Grafische Benutzeroberflächen für den QL

Das erste Desktop à la GEM für den QL brachte das britische Softwarehaus Eidersoft heraus. Mittlerweile gibt es zwei ähnliche Produkte aus Deutschland: "E.A.S.E." (Easily Applicable System Environment) von Gigasoft und "J.A.M." (Job Application Monitor) von zwei frankischen Autoren. Wir haben beide Programme getestet.

Das Programm "E.A.S.E." benötigt ca. 64 KByte, womit beim Grund-QL schon der meiste Speicherplatz belegt ist. Deshalb wird es beim Verlassen sofort gelöscht und nach einem erneuten Aufruf wieder geladen. In einem QL mit Speichererweiterung ist es ständig parat. "E.A.S.E." wird über Pull-Down-Menüs gesteuert; auf dem Bildschirm befindet sich ein Pfeil, der mit Maus, Joystick oder Cursor-Tasten in ein Menüfeld gebracht wird. Drückt man nun auf Maustaste, Feuerknopf oder Leertaste, wird der angesteuerte Menüpunkt ausgeführt. Damit der Bildschirm frei bleibt, gibt es am oberen Rand nur wenige Menüpunkte, von denen bei Anwahl ein Untermenü gleich einem Rollo herunterfällt. "E.A.S.E." bietet vier Begriffe: DESK, FILE, OPTIONS und EXIT. Der letzte Punkt dient zum Verlassen des Programms.

Hinter DESK verbergen sich vier Unterpunkte. Der erste, ABOUT E.A.S.E., ist der gleiche Betrug am Benutzer wie beim GEM auf dem Atari ST. Hinter CALCULATOR verbirgt sich ein Taschenrechner mit UPN-Eingabe wie bei HP-Taschenrechnern. Er verfügt über etliche wissenschaftliche Funktionen sowie eine Rechengenauigkeit, die Super-Basic entspricht. Was ich allerdings vermißt habe, sind Hexadezimalzahlen, die ein Computerbenutzer doch häufig braucht. Die Bedienung des Taschenrechners mit der Maus ist nur etwas für verspielte Zeitgenossen; eine Eingabe über Tastatur wäre angebracht. Dann gibt es

noch den Punkt GAME, ein einfaches Spiel, um die Zeit totzuschlagen. Gut ist, daß Taschenrechner und Spiel mehrfach (und gleichzeitig) angewählt werden können. Der vierte Punkt, PANEL, erlaubt das Einstellen diverser Systemparameter (z.B. Default Laufwerk).

Der Menüpunkt FILE macht so ein Desktop-Programm erst richtig interessant, da beim QL das Kopieren von Files ziemlich umständlich ist. Die Befehle sind lang, und auch das Löschen bedeutet viel Tipparbeit. Hier werden nun alle zu kopierenden/löschenden Files markiert (d.h. mit Pfeil anfahren und anklicken); anschließend wählt man ganz einfach COPY oder DELETE. Ein "Klick", und der Rest geht von alleine. Was leider fehlt, ist der Unterpunkt RENAME. Es gibt aber die Punkte FORMAT (den Namen der Cartridge muß man wirklich eintippen) und START. Damit kann man einen Job starten oder ein anderes File listen bzw. drucken. Auch ein Super-Basic-Programm kann gestartet werden. So wird die Fileverwaltung zu einem Kinderspiel; die lästige Tipperei entfällt. Cartridges oder Floppys, die man immer schon mal aufräumen wollte, werden jetzt von allem Datenschnott befreit.

Der letzte Punkt ist OPTIONS. Hier kann man einstellen, ob man seine Files als Icons oder als Text sehen möchte. Icons sind kleine Bilder, die den Filetyp angeben. QUILL-Files sind typischerweise kleine beschriebene Blätter, EASEL-Files kleine Diagramme usw. In der Option "Text" werden neben Filenamen auch Länge und Erstelldatum angegeben. Im normalen QDOS gibt es kein Datum, während z.B. Toolkit II alle Files damit versieht. Die Files können in diesem Modus auch wahlweise nach Namen, Länge und Datum sortiert werden.

Kommen wir zum "J.A.M." aus Franken. Hier sind alle Fen-

ster fest vorgegeben. Ein Verschieben ist auch nicht erforderlich, da alle Menüpunkte nur einmal angewählt werden können. Was ist also der Unterschied zu "E.A.S.E."? Zunächst gibt es kein Spiel. Aber wer ein Spiel will, der kauft kein Desktop. "J.A.M." ist nur 27 KByte lang, also wesentlich kürzer. Der Taschenrechner bietet weniger Funktionen, ist aber über Tasten zu bedienen. Es gibt einen Kalender, der den jeweils aktuellen Monat anzeigt. Man kann ihn vor- und zurückblättern, was sich bei der Terminplanung als praktisch erweist.

"J.A.M." verfügt am linken Rand über fünf Symbole für die unterstützten Laufwerke: zwei Floppys, zwei Microdrives und eine RAM-Floppy. Diese RAM-Floppy ist als schneller Zwischenspeicher eine große Hilfe beim Kopieren von Files. Hat man nur ein Laufwerk, so kopiert man von der ersten Diskette auf die RAM-Floppy und von da auf die zweite Diskette. Ein Zwischenspeichern auf Microdrive entfällt; der Kopiervorgang wird viel schneller.

Das File-Menü bietet auf den ersten Blick sehr viel mehr, als es später hält. Hat man sich näher damit beschäftigt, erkennt man, daß nur die Organisation anders ist. Alles, was mit "E.A.S.E." möglich ist, geht auch mit "J.A.M.". Ein wichtiger Pluspunkt ist aber die Option RENAME zum einfachen Ändern von Filenamen.

Während "E.A.S.E." ausschließlich Fileverwaltung und Desktop-Funktionen bietet, gibt es bei "J.A.M." eine gute Verwaltung von Jobs. Im JOB-Menü können sie alle gelistet, suspendiert oder gelöscht werden; auch Prioritäten lassen sich ändern ("J.A.M." nie auf Priorität Null setzen). Da QDOS keine Befehle zur Job-Verwaltung bietet, ist dies ein dickes Plus.

Unter der Rubrik "Options" bietet "J.A.M." einen Taschenrechner, einen Kalender, Me-

memory Info und einen Notizzettel. Eine kurze Notiz wird beim nächsten Start automatisch angezeigt. Der Taschenrechner bietet zwar nur die Grundrechenarten und Wurzelziehen, ist dafür aber über die Tastatur zu bedienen. Auch hier fehlen die so wichtigen Hexadezimalzahlen. Der Kalender zeigt beim Einsprung immer den aktuellen Monat; man kann nun beliebig vor- und zurückblättern. Es gibt auch die Möglichkeit, die Geschwindigkeit des Pfeils einzustellen. Benutzt man einen Joystick, so dauert es ein Weilehen, bis man die optimale Geschwindigkeit gefunden hat.

"J.A.M." ist kopiergeschützt, d.h., man kann zwei Kopien ziehen, dann weigert sich das Kopierprogramm, weitere zu produzieren. Alle anderen Methoden (QLone, COPY, "J.A.M." selbst) haben keine lauffähige Version erzeugt.

Beim Vergleich der beiden Desktop-Programme scheidet "J.A.M." besser ab: Es ist praxisorientierter und nicht so verspielt. Auch die Tatsache, daß es nur etwa die Hälfte an Speicherplatz benötigt, ist nicht unwichtig.

Vergleicht man die Preise ("J.A.M." ca. 69.- DM, "E.A.S.E." ca. 222.- DM), so ist zu berücksichtigen, daß "E.A.S.E." mit anschließertiger Maus und Giga-Basic (eine Basic-Erweiterung die alleine ca. 50.- DM kostet) geliefert wird. Die Gigasoft-Maus belegt allerdings neben dem Joystick-Port CTRL1 auch den ROM-Port, was für diverse Compiler- und Toolkit-Besitzer einige Probleme mit sich bringen dürfte.

Weitere Informationen: "E.A.S.E.": ABC-Electronic Andreas Budde Hängelstraße 30-32 4800 Bielefeld 1

"J.A.M.": T.S. Datensysteme Demistoße 45 8500 Nürnberg 80 Computerstudio Kreuzstraße 13 8000 München 2

Tricks und Tips für den QL

FOR und REMark

Das Super-Basic des QL kennt zwei Varianten der FOR-END-FOR-Schleife: die kurze und die lange Version. Beim Umschreiben von Basic-Programmen von anderen Rechnern kann es mit der kurzen Version Probleme geben. Diese Version liegt immer dann vor, wenn nach dem FOR in der Zeile noch ein weiteres Statement folgt und nicht etwa nur dann, wenn kein NEXT oder END FOR folgt. Eine Fehlerquelle ergibt sich, wenn auch ein REMark ein Statement ist. Um dieses zu demonstrieren, zeigt Listing 1 ein kleines Beispiel, das diesen Punkt verdeutlicht. Obwohl ein REMark den Programmablauf nicht beeinflussen sollte, tut es dies jedoch ganz entscheidend. Viele Basic-Programmierer schreiben einige Statements auf anderen Rechnern bereits hinter das FOR. Hier muß man beim Umschreiben auf den QL sehr aufpassen, denn solche Fehler sind manchmal schwer zu finden.

Löschen von Prozeduren und Funktionen

Hat man Prozeduren oder Funktionen im Programm, so kann es recht häufig vorkommen, daß man die eine oder andere auch mal löscht. Dazu verwendet der normale Program-

mierer den Befehl DLINE. Aber Vorsicht: Durch das Löschen der Programmzeilen wird der Prozedureintrag in den Systemvariablen nicht gelöscht. Das heißt, der Rechner erkennt den Namen der Prozedur/Funktion immer noch, findet sie aber natürlich nicht mehr.

Geben Sie dazu eine kleine Prozedur ein und löschen diese Zeilen mit DLINE. Verwenden Sie anschließend den Prozedurnamen als Variable. Jetzt werden verschiedene Fehlermeldungen erscheinen. In Einzelfällen kann der Rechner sogar abstürzen. Die einzige Hilfe: Das Programm save, dann einen Reset durchführen und das Programm neu laden.

Prozedur- und Funktionsnamen sollte man bei der ersten Eingabe immer mit Großbuchstaben schreiben. Sie werden dann bei weiteren Eingaben unabhängig von der Schreibweise groß geschrieben. Ist der Name einmal mit Kleinbuchstaben eingegeben, wird er immer mit Kleinbuchstaben geschrieben, gibt man den Namen gemischt ein, erfolgt die Schreibweise auch hier entsprechend. Eine Eingabekürzung (wie z.B. bei REMark) läßt sich so aber nicht generieren.

Im Block-Befehl gibt es einige kleine Fehler. Die Breite muß kleiner als 512 sein. Gibt

man 512 an, so passiert nichts, denn 512 wird als Null angesehen. Die maximale Breite eines Blocks ist 511 Pixel, 513 wäre Breite 1. Für die Höhe ist 256 der Maximalwert. Bei 257 gibt es eine klare Fehlermeldung.

Der Point-Befehl

Im deutschen QL (ROM-Version MGG) gibt es einen Fehler. Der Befehl POINT zeichnet nicht ein Pixel, sondern zwei. Das zweite ist oberhalb um ein Pixel nach links versetzt. Da alle anderen Zeichenroutinen (LINE, CIRCLE, BLOCK usw.), falls sie nur ein Pixel zeichnen müssen, auf die

POINT-Routine zurückgreifen, haben diese ebenfalls das Problem. LINE 50,50 TO 50,50 arbeitet fehlerhaft, LINE 50,50 TO 50,51 funktioniert jedoch richtig. Da das Pascal von Computer One ROM-Routinen anspricht, gibt es den Fehler auch hier.

SELECT in Prozeduren

Es wurde berichtet, daß in einigen englischen ROM-Versionen es nicht möglich ist, eine Übergabevariable in einer Prozedur als SELECT-Variable zu benutzen. Tests auf dem deutschen QL haben gezeigt, daß es hier geht. Besitzer englischer

Listing 1 demonstriert den Unterschied zwischen der kurzen und langen FOR-Schleife. Der dritte Teil zeigt, wie ein REMark die Schleife beeinflusst.

```

100 CLS
110 REMark Demo Lange Version
120 :
130 FOR i= 1 TO 10
140 PRINT i,i*1,i*3,i*4
150 END FOR i
160 PAUSE:CLS
170 :
180 REMark Demo kurze Version
190 :
200 FOR i=1 TO 10: PRINT i,i*1,i*3,i*4
210 PAUSE:CLS
220 :
230 REMark Demo lange Version mit REMark
240 :
250 FOR i= 1 TO 10: REMark
260 PRINT i,i*1,i*3,i*4
270 END FOR i

```

Listing 2 zeigt, wie man eine LLIST-Prozedur mit Druckersteuerung schreiben kann. Um Platz zu sparen, sollte man die REMarks am Ende einer jeden Zeile nicht mit abtippen. Zeile 250 kann ganz entfallen. Die Einstellung des rechten Randes läßt sich gegebenenfalls noch einbauen.

```

100 REMark LLIST für Sinclair QL
110 REMark Druckersteuerzeichen für TA Gabriele 9009
120 REMark R.W. Gerling
130 REMark Juni 1986
140 DEFine PROCEDURE LLIST
150 LOCAL esc$:esc$=CHR$(27) :REMark Codes für ESC und CR
160 OPEN#99,ser1 :REMark Kanal 99 auf Schnittstelle 1
170 TRA 1 :REMark Übersetzen auf deutschen ASCII
190 PRINT#99;esc$;CHR$(13);"P"; :REMark Reset des Druckers
200 PRINT#99;esc$;"a"; :REMark Auto Carriage Return ein
210 PRINT#99;esc$;CHR$(9); :REMark linken Rand setzen
220 rand=4:PRINT#99;CHR$(rand); :REMark auf 4te Schreibstelle
230 PRINT#99;esc$;"9";
240 PRINT#99;esc$;CHR$(22);CHR$(2); :REMark deutschen Zeichensatz für
250 :REMark Schreibmaschine
260 LIST#99,1000 TO
270 END DEFine LLIST

```




Uta Jäkel & A.K. Lintworth GbR



Hard- & Softwarevertrieb

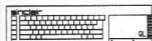
Marschhorst 2

2732 Klein - Neckelsen

Telefon: 04282 / 5615

ZX Spectrum 48 KB	249,-		Discovery 180 KB	399,-
ZX Spectrum plus 48 KB	329,-		Discovery 360 KB	599,-
ZX Spectrum 128 KB	449,-		Discovery 720 KB	799,-
Saga 1 Tastatur	129,-		Discovery 1440 KB	1199,-
Saga 3 Tastatur	249,-		Einbaulaufwerk 180 KB	199,-
Dk' tronics Tastatur	199,-		Einbaulaufwerk 720 KB	399,-
Kepton E Centronicsinterface	155,-		Beta Disk Controller 5.81	329,-
ZX Lprint III Druckerinterface	175,-		Beta Disk Komplettsystem 1 MB ab	749,-
Multiflex One (neue Ausführung)	155,-		SpeeDruS Schlagzeug-Synthesizer	149,-
Dk' Singleport Joystickinterface	29,-		Sprachsynthesizer	89,-
Dk' Doppelport Joystickinterface	45,-		Dk' 3-Kanal Soundsynthesizer	109,-
Kepton Pro Joystickinterface	59,-		Maus incl. Interface	ab 269,-
Art Studio Malprogramm	59,-		Tasword III (Cartridge oder Disk)	69,-

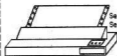
Achtung: Wir führen auch Software für den ZX Spectrum 128 KB !!!



QL 128 KB englische Ausführung	429,-
QL 128 KB deutsche Ausführung	459,-
Diskcontroller (Software in Epron) ab	299,-
Disk Komplettsystem (1 MByte) ab	699,-
Maus incl. Interface und Software	228,-
Super Diskcontroller + Centronicsinterface + TOOLKIT II + 512 KB auf einer Karte	799,-
512 KB Speicherverlängerung mit durchgeführten Bus (einfach ansteckbar)	389,-
QL 128 KB (englische oder deutsche Ausführung) + 512 KB Speicherverlängerung	799,-
Softwarepaket für Betriebe (Fibu, Adressverwaltung, Kalkulation, Serienbriefe, Lohn- und Gehaltsabrechnung, Lagerverwaltung, Faktura, läuft nur mit Disk)	890,-
Monitore (grün, bernstein) ab	248,-
QL Centronicsinterface	ab 139,-
CUB 653 Super Farbmonitor	1898,-
QSound / QPrint Interface	198,-
QL ART Super Grafikprogramme (Font Editor, Zoom, 2 Screens à 652 x 256 Pixel, sehr benutzerfreundlich, selbstvorführende Demo (lieferbar))	98,-

Die QL ART jedoch wesentlich umfangreicher (läuft nur mit Zusatzspeicher, Auflösung pro Grafikkarte 720 x 864 Pixel, Textured Fill, auf Epson komp. Druckern als DIN A4 oder DIN A2 - Poster ausdrückbar) 148,-

Cartridges	Stk.: 7,50	12 Stk.: 85,-	28 Stk. in Cartridge - Box	158,-
Cartridge Box	20,-	3,5" - 50er	Disketten Box mit Schloß	29,-
3,5" - Disk No Name 135 TPI	Stk.: 5,50	10 Stk.: 5,-	50 Stk.: 225,-	
3,5" - Disk Scotch SDD0 135 TPI	Stk.: 7,50	10 Stk.: 70,-	50 Stk.: 325,-	
3,5" - Disk Scotch DSD0 135 TPI	Stk.: 8,-	10 Stk.: 75,-	50 Stk.: 350,-	



Citizen 120 D (IBM + Epson kompatibel)	729,-
Seikosha MS15 Typendruckdrucker (RS232 + Centronics)	699,-
Seikosha SP 1080 (A / RS / I / UC)	749,-
Centronics GLP II (RS232 + Centronics, 180 Z/s)	598,-
Centronics GLP II + Aufsatztraktor	648,-
andere Drucker (Star, Oki, usw.)	auf Anfrage

Übrigens: Für alle Drucker, die wir verkaufen, können wir auch Farbbänder liefern!

Lieber Kunde, die Preise können sich zwischen Lagererstellung und Veröffentlichung dieser Anzeige schon geändert haben. Daher:

Ruf doch mal an!

Versand nur per Nachnahme oder Vorauskasse! Selbstabholung und Vorführung von Artikeln nur nach vorheriger telefonischer Terminabsprache!
Alle Preise zuz. Versandkosten zum Selbstkostenpreis!
Unsere Gesamtpreisliste erhalten Sie gegen 2,- DM in Briefmarken!

Übrigens: Wir exportieren auch ins Ausland!!!

Diese Anzeige wurde komplett und in Originalgröße mit unseren Graphikprogrammen **ART+** erstellt!

QLs sollten einmal berichten, ob es bei Ihnen geht.

Das Listen von Super-Basic-Programmen artet, falls man den Drucker auch an anderen Computern benutzt, häufig in eine ziemliche Spielerei mit den DIP-Schaltern aus. Eine einfache Lösung bietet dazu die Prozedur LLIST. Sie stellt den Drucker softwaremäßig auf die entsprechenden Parameter auf. Am Drucker sollte AUTO Carriage Return (CHR\$(13)) eingegeben werden, da der QL nur ein Linefeed (CHR\$(10)) sendet. Auch die Einstellung der Breite muß am Drucker erfolgen, da der Befehl WIDTH bei Listings nicht funktioniert. Alte Spectrum-Fans können am Drucker auch 32 Zeichen pro Zeile eingeben. Zusätzlich gibt es noch viele weitere Möglichkeiten, etwas einzustellen. Listing 2 zeigt eine mögliche Form. Damit man nicht ständig den LLIST-Teil mit auflistet, sollten die Programme immer

in Zeile 1000 beginnen. LLIST listet dann nur ab Zeile 1000.

Hilfsroutinen

Man kann natürlich unterhalb Zeile 1000 auch noch andere wichtige Prozeduren und Funktionen ablegen. Listing 3 zeigt drei Beispiele. Es sind dies der Befehl CAT für Katalog sowie die Prozeduren ZEIT und TAG. Der Befehl CAT verlangt eine Laufwerksnummer (1 für MDV1., 2 für MDV2., usw.) und gibt dann den Inhalt der Cartridge im Laufwerk aus. Mit ZEIT wird die Uhrzeit und mit TAG das Datum gestellt. ZEIT und TAG entsprechen in der Funktion den Kommandos TIME und DATE unter MS-DOS oder CP/M. Es gibt beliebige weitere Möglichkeiten, hier der Phantasie freien Lauf zu lassen und seine eigene Hilfsroutinen zu schreiben. Und alles geht in Basic vorstatten.

Rainer W. Gerling

Listing 3 zeigt drei Prozeduren: CAT, TAG und ZEIT. CAT erlaubt einen einfachen Katalog einer Cartridge und mit TAG und ZEIT läßt sich sehr einfach Datum und Uhrzeit setzen.

```

100 REMark Hilfsroutinen
110 REMark Demo und Anregung
120 REMark R. W. Gerling
130 REMark Juni 1986
140 REMark *****
150 Define PROCEDURE CAT(drive)
160 PRINT "Katalog von Laufwerk ";drive;" :
170 DTR "newscrivek" :
180 END Define
190 REMark *****
200 Define PROCEDURE ZEIT
210 LOCAL a$,b$,c$,d$,e
220 a$=DATE$:a$=INT( TO 4):b$=(6 TO 8) INSTR
   "JanFebMarAprMayJunJulAugSepOctNovDec"/3:
   c$=INT( TO 11):d$=INT( TO 17)
230 PRINT "Alte Zeit: ";a$
240 INPUT "Neue Zeit: ";b$:IF b$="" THEN PRINT a$:RETURN
250 d$=INT( TO 2):IF b$<0 OR b$>23 THEN GO TO 240
260 e$=INT( TO 5):IF b$<0 OR b$>5 THEN GO TO 240
270 SDATE a$,b$,c$,d$,e
280 PRINT
290 END Define
300 REMark *****
310 Define PROCEDURE TAG
320 LOCAL a$,b$,c$,d$,e,f,i
330 a$=DATE$:a$=INT( TO 4):b$=(6 TO 8) INSTR
   "JanFebMarAprMayJunJulAugSepOctNovDec"/3:
   c$=INT( TO 11)
340 PRINT "Altes Datum: ";a$;" ";b$;" ";c$
350 INPUT "Neues Datum: ";a$:" ";b$:" ";c$:
   " ";d$:" ";e$:RETURN
360 c$=INT( TO 2):IF a$<0 OR a$>31 THEN GO TO 350
370 i=INT( TO 8)
380 b$=INT(4-1 TO 5-1):IF b$<0 OR b$>12 THEN GO TO 350
390 i=i+(b$<0)
400 a$=INT(7-1 TO 8-1):IF a$<0 OR a$>30 THEN GO TO 350
410 e$=INT(a$+a$=DATE$:d$=INT( TO 14):f$=INT(15 TO 17):
   f$=INT( TO 3)
420 SDATE a$,b$,c$,d$,e,f
430 PRINT
440 END Define
450 REMark *****
460 END
470 PRINT "DOOS VERSION: ";VERSION
480 TAG:ZEIT

```

Spectrum Shadow ROM Disassembly

Von Gianhica Carri
Verlag Melbourne House
ISBN 0-86161-191-8

Vor einiger Zeit ist im englischen Verlag Melbourne House ein Disassembliertes des Interface 1 ROMs erschienen. Für jeden, der in Maschinensprache programmiert und der die M-Code Routinen im Interface 1 nutzen will, ist dieses Buch fast schon ein Muß.

Im ersten Teil wird ausführlich dargestellt wie man neue Befehle für das Sinclair Basic schreiben kann. Danach wird ausführlich eine 1303 Byte Basic-Erweiterung besprochen. Die neuen Befehle sind so gut, daß sie alleine den Kaufpreis schon wert sind. Im einzelnen handelt es sich dabei um einen Doppel-POKE-Befehl, einen String-POKE-Befehl sowie einen erweiterten BEEP-Befehl. Doch das ist nur der Anfang. Die Krüller sind ein pseudo Random-File-Befehl, der es erlaubt, gezielt einen Sektor vom Microdrive zu laden. Die Syntax ist: READ #s, n, wobei s der Strom und n die Nummer des Sektors im File ist. Das File muß vorher mit OPEN # geöffnet werden. Dazu gibt es noch einen Befehl, mit dem man ein bestehendes File auf der Cartridge erweitern kann. Er heißt RESTORE #s (s ist wieder die Strom-Nummer) und ermöglicht es, am Ende des Files weiterzuschreiben. Sehr hilfreich ist auch der erweiterte CAT-Befehl (er gibt Filetyp, Startadresse, Länge und Autostartzeile an), der Memory-Dump-Befehl und die Erweiterungen des RS 232-Kanals. Mit am besten finde ich den neuen EDIT-Befehl. Damit kann man ohne vorhergehendes LIST eine Zeile editieren. Man muß nur die Zeilennummer angeben.

Es versteht sich von selbst, daß die Befehlserweiterung für beide Versionen des Interface 1 ROMs angegeben ist. Und sollte es einmal eine dritte Version geben, so gibt es Hinweise, wie man die Maschinensprache den Routinen anspaßt. Alle Routinen sind auch komplett

im Quellcode gelistet und ausführlich kommentiert.

Der zweite Teil des Buches, das einmal nicht von einem Engländer geschrieben wurde, ist ein komplettes kommentiertes Listing des Shadow-ROM im Interface 1. Gelistet ist hier das ROM der Version 1. Die Änderungen in der Version 2 werden im Anhang erläutert. Die Qualität des Disassemblies steht der in "The Complete Spectrum ROM Disassembly", von Logan O'Hara im gleichen Verlag erschienen, nicht nach. Die Kommentare und Erklärungen sind ausführlich und gut.

Wer nicht weiß, welches ROM sich in seinem Interface 1 befindet, der schaue sich die Seriennummern an: bis 87315 ist die Version 1 eingebaut. Ein weiterer kleiner Test ist im Buch beschrieben. Das Buch kostet 8,95 Pfund, wenn man es in England beim Verlag bestellt. Dazu kommen noch 0,80 Pfund Versandkosten. Ich kann das Buch jedem, der in die Details des Interface 1 einsteigen will, wärmstens empfehlen.

An dieser Stelle noch ein Tip zum Bestellen in England. Viele Firmen nehmen keine Eurochecks, und wenn sie welche nehmen, dann kommen noch viele Gebühren dazu. Ich benutze zum Bezahlen in England immer eine Kreditkarte.

**Ihr direkter Draht
zur Redaktion:
Telefon 0 72 52 / 4 29 48
Für Briefe:
Computer Kontakt
Postfach 16 40
7518 Bretten**

Entweder braucht man eine Visa-Karte oder eine Eurocard. Vielleicht hat ja ein Bekannter oder Verwandter eine solche Karte. Das spart Gebühren und macht das Bestellen einfacher. Es geht damit sogar per Telefon, und die Rechnung gibt es in DM. Die Anschaffung einer Kreditkarte wegen des Computerhobbys lohnt sich allerdings nicht.

Rainer Gerling

Wir machen ein  auf!

CK

Fehlt Ihnen noch eine Ausgabe von
vom **85**er Jahrgang ? Oder mehrere ?
Oder sogar alle ?

Das muß nicht sein ! Wir schnüren jetzt
ein  . Und da geht echt die




ab.

Da ist das  drin und



Ja sogar



Und mit  setzen wir dem ganzen die
auf. Das sind **9** Hefte die am Kiosk

40 Mark und 50 Pfennige kosteten.

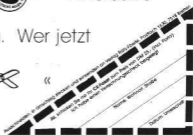
Und bei uns ? -   . Inclusive



und Verpackung. Wer jetzt

nicht den Coupon »  «

ist selber schuld.





COMMODORE-NEWS

Hallo Commodore-Freunde!

Wie im Vorwort des Heftes schon angekündigt, wird der Commodore-Teil in der CK nicht weiter erscheinen. Nur den letzten Teil des Grafikkurzes werden wir in Heft 2-3/87 noch bringen können.

Seit der ersten Ausgabe der CK im Mai 1984 war der Commodore-Teil dabei. Handelte es sich damals noch um unsere Hauptlesergruppe, so hat sich dies immer mehr verlagert. Die übergroße Konkurrenz ist zu stark; das hat sich gerade in den letzten beiden Ausgaben gezeigt, als die Commodore-Leser immer weniger wurden. Diese Situation zwingt uns deshalb, den Commodore-Teil aufzugeben. Wir finden das zwar schade, aber ein kleiner Verlag wie wir muß sich absolut an den Marktgegebenheiten orientieren.

Trotzdem, auch in diesem Heft haben wir noch einige interessante Beiträge auf Lager. Wie immer gibt es Programme, neue Software, Buchrezensionen sowie Tips und Tricks. Auch für dieses letzte Heft haben wir uns für unsere Leser noch einmal ins Zeug gelegt.

An dieser Stelle möchte ich mich bei all den Lesern bedanken, die trotz der großen Commodore-Zeitschriften-Konkurrenz uns treu geblieben sind. Da ist mancher dabei, der die CK seit ihren Anfängen kennt und nun – vielleicht etwas wehmütig – auf sie verzichten muß. All diesen Lesern wünschen wir alles Gute und natürlich weiterhin viel Spaß mit ihrem Computer, schließlich ist besagte Konkurrenz auch nicht gerade schlecht.

Thomas Eberitz

Boulder Dash III

Handlungsweise und Sinn des Spiels sind recht schnell erklärt, doch vielleicht gerade wegen seiner simplen Handlung fesselt einen "Steinschlag" (dritter Teil), bis man nicht mal mehr Kaffee kochen oder eine Zigarette rauchen will.

Getestet wurde die Cassettenversion für den Commodore 64. Man kann jedoch davon ausgehen, daß es Versionen für so ziemlich alle gängigen Computertypen gibt.

Es geht im Spiel darum, daß man mit einem kleinen Männchen namens Rockfort durch einen Screen flitzen und alle dort herumliegenden Diamanten einsammeln muß, um ins nächste, in der Regel wesentlich schwerere Bild zu gelangen. Wer jetzt weiterblättern und "Viel zu einfach" sagt, ist selbst schuld und hat sich schwer getäuscht. Bei seiner Suche nach den Diamanten muß Rockfort nämlich auf eine Menge achten. Wer beim Graben nicht aufpaßt, oder gar zu viele Steine lockert, wird recht schnell erschlagen bzw. blitzschnell eingesperrt. Doch damit nicht genug. Als weitere Schikane sausen noch einige Gestalten (Mäuler und Augen) durch die Höhle, von der man übrigens immer nur einen kleinen Ausschnitt sieht. Bei Berührung dieser Figuren verliert man eines von drei Leben. Biswellen liegen in einem Raum nicht ge-

nug Diamanten, um in die nächste Höhle zu gelangen. Dann muß man zu einem kleinen Trick greifen: Verrietet man ein Auge, so erhält man ungefähr 9 Diamanten. Sollten dies immer noch nicht ausreichen, so findet man in solchen Räumen noch lust blubbernde Lava, die man durch Einsperren mit Steinen in Diamanten verwandeln kann. Es besteht dann nur noch das Problem, an sie ranzukommen. Wenn das immer noch zu einfach erscheint, der kann sich nach überstandenen 4 Höhlen mit einem Puzzle auseinandersetzen, bei dem ein Leben zu gewinnen gibt. Dies geschieht übrigens alle 5 Punkte; hier erhält man ebenfalls einen Zusatz-Rockfort.

Die Anzeige aller wichtigen Informationen ist recht gut gestaltet. In der obersten Zeile ist während des ganzen Spiels, wie viele Diamanten man sammeln muß, wie viele Punkte es pro Diamant gibt, der Score usw. Zusätzlich werden einige spezielle Informationen angezeigt. So flammt z.B. der Bildschirm kurz auf, wenn der Ausgang offen ist. Auch mit Gags wurde wie schon bei den Vorgängern nicht gespart. Ist man unsere kleinen Held mit den Kniepfeilen zu langsam, so stampft ungeduldig mit dem Fuß.

Sollte man die Vorgänger schon besitzen, so lohnt es sich doch, den dritten Teil zu ka-

C 64 und PC 128

8/5 Modul	30,00 DM	Super-Hardcopy-Modul	45,00 DM
Hardware + Turbo Tape + Turbo Disk + Random-Ord		Druckt auf jedem Computer-Disk	
Supersoftwarepaket I	18,95 DM	8/5 Modul	39,00 DM
10 Programme auf Disk, oder Kassette, Anwendungs- und Notprogramme		Turbo Disk + Hardcopy + Random-Ord + 50 Dranger + Random-Ord	
10 Disketten 5 1/4		8/1 Modul	19,00 DM
Supersoft	14,50 DM	Turbo Tape	
8/4 Modul	33,00 DM	Turbo Disk + 50 1/4 und 5 1/4 Disketten	19,00 DM
Turbo Tape + Turbo Disk + DOS 5.1 + Random-Ord		Prof. Betriebssystem	49,00 DM
Modul 2000	45,00 DM	für schreibere Lesern, C64 und C63, Auktionen, Funktionsrechner, in-Banknote	
Turbo Tape + Turbo Disk + Monitor 5000 + 50 Dranger		Betriebssystem-	
oder siehe fast C64, PC 128		umschaltplatte 125	39,00 DM
Modul 3000	65,00 DM	Druckt auf 125 Dranger im 80-Minuten auf dem C28 weitere Betriebssysteme in Betrieb, Platte mit 100 Seiten, weitere Platten auf Anfrage	
Monitor + Turbo Disk + Turbo Tape + Hardcopy + Floppy + 50 Dranger + Backup + Random-Ord		Laserplatte	9,95 DM
Modus 8-128	49,00 DM		
Neuauflage des Labortagungs der 12/7-71 Version			
5,7" mit 16er Modult			
Alle Module im Gehäuse, mit Reset (außer bei 5,00 DM). Die Programme belegen keine Speicherplätze.			
24 Stunden Service, mit gegen 1,80 DM. Versand zu Hause und Commodore-Garantie, 1. 8. 1986			

**Klemmer & Schulte
Electronic**

Berrenratherstr. 496 - 5020 Hürth-Effern
Lieferadresse: Mo - Fr, 12.00-18.00 Uhr
Sa. 10.00-13.00 Uhr, 12.22-13.22 Uhr

fen. Das neue Spiel ist schneller und schwieriger als die beiden ersten Teile. Grafik und Sound sind von der Qualität her unver-

ändert gut; das Menü ist auch weiterhin optimal. Fazit: uneingeschränkt empfehlenswert.

Dirk Zwers

Parallax

Das Superraumschiff eines außerirdischen Volkes ist auf dem Weg zur Erde. Die Besucher kommen in friedlicher Absicht und nehmen sogar eine Delegation der Erde an Bord auf, bestehend aus den fünf erfahrensten Astronauten der Welt. Die Besatzung des Raumschiffs zeigt sich sehr gastfreundlich. Doch zufällig stolpert einer der Astronauten über gewisse Informationen von einem Computerhirn, die einen bevorstehenden Angriff auf die Erde aufdecken.

Das Superraumschiff mit der gastfreundlichen Besatzung verwandelt sich für die fünf Astronauten zum Deathstar, der die Welt zerstören will. Und wer kann das ändern? Sie, ja ganz genau Sie! Sie schlüpfen nämlich in die Rolle des todessmutigen Raumschiffpiloten, der die vier Astronauten-Kampfs der vier abgegrenzten Zonen des Raumschiffs bergen soll. Um das Tor zum jeweils nächsten Abschnitt zu öffnen, braucht man ein Paßwort, das man den Datenkarten der zu kidnappenden Wissenschaftler entnehmen muß. Am Ende tauscht man dann die vier gekidnappten Wissenschaftler aus den fünf Raumschiff-Zonen gegen die vier Astronauten aus und verschwindet dann in Windeseile in Richtung Heimatplaneten Erde.

"Parallax" ist eine ansehnliche Mischung aus Action-Ad-

venture und Ballerspiel, bekommt untermalt mit 3-D-Effekten und recht bombastischer Musik.

Auf der Suche nach den Hangars fliegen Sie mit Ihrem Raumkreuzer Ibis über das gigantische und gewaltig anmutende Schiff der Fremden. Die Hangars sind Hallen, in denen meist ein oder mehrere Computer stehen, die von Robotern oder Wissenschaftlern bewacht werden. Kidnappt man einen Wissenschaftler, so kann man dessen Datenkarte nehmen und diese in das Hauptkontrollsystem der Feinde (den Central Intelligence Unit C.I.U.-Computer oder auch Big One) eingeben, von wo aus man dann einen Buchstaben des Lösungswortes erhält.

Die Computer-Banken oder die Computer-Shops sind auch in vielen Hangars anzutreffen. Durch diese Systeme kann man Geld hebeln bzw. ausgeben, wenn man zur Datenkarte ein Paßwort hat. Hat man aber kein Paßwort und möchte Geld bei der Computer-Bank abbekommen, so behält dieser originale Computer einfach die Datenkarte.

Auf dem Weg von Hangar zu Hangar wird man natürlich von feindlichen Fliegern, Verteidigungsgeschützen, Boden- und Untergrundtruppen belästigt. Dieses Problem läßt sich aber durch Zielen und leichtes Antippen des Feuerknopfes bequem lösen. "Parallax" ist programmtechnisch ein Referenzwerk für alle Weltraumspiele. Die futuristische Musik wird von der dreidimensionalen Grafik sogar noch übertroffen. Es ist aber auch ein Spiel, dem man nicht gleich ansieht, daß sich eine höhere Handlung dahinter verbirgt. Den Schwierigkeitsgrad könnte man am treffendsten mit "anspruchsvoll" umschreiben.

System: C 64
Hersteller: Ocean
Stefan Selbach

Nuclear Embargo

"Nuclear Embargo" heißt ein neues Weltraumspiel von Eurogold. Spielidee und -handlung heben sich von den meist üblichen Schießspielen ab. "Rayol, der Perverse", seines Zeichens Tyrann auf Saturn, bricht den jahrhundertalten Vertrag eines Vorfahren, der Terra regelmäßige Uranlieferungen zuscherte. Die Regierung der Erde beschließt daraufhin, sich den notwendigen Rohstoff mit Gewalt zu beschaffen, und schickt ihren besten Piloten auf die gefährliche Reise. Hat dieser sich durch die Spielanleitung gelesen, die zwar in Deutsch, aber sehr klein gedruckt ist, bricht er auf und sieht den ersten der vier Bildschirme vor sich: Eine Sternenkarte mit einigen Punkten, die Saturnmonde symbolisieren. Per Knopfdruck düst man zum Ziel und erkennt unter sich die Mondoberfläche (zweiter Bildschirm). Jetzt bewegt man den Joystick, um mit Hilfe eines Geigerzählers das Uranlager zu finden, und zerstört mit der Bordkanone eine aufgestörte Verteidigungsbasis, was selbst Anfängern keine Probleme bereitet. Dann gerät man in den Transporterraum (dritter Bildschirm), der angeblich in hochauflösender Grafik dargestellt ist, und beamt einen Roboter auf die Mondoberfläche, eine je nach Magnetfeldstärke des Mondes mehr oder weniger schwierige Aufgabe. Ärgerlich ist nur, daß gerade dieser an sich schwierigste Teil des Spiels auf einem winzigen Bildschirmausschnitt stattfindet, während ansonsten immer noch der relativ unwichtige Transporterraum gezeigt wird.



Am Ziel angekommen sieht man den einzigen grafisch überzeugenden Bildschirm, das Uranfeld, wo der arme Roboter sich nun Uran stehend und bis auf ein Schutzfeld unbewaffnet mit Laserbatterien und Müllbe-seitigungsstationen auseinandersetzen muß. Nach erfolgreicher Rückbeamtung wird eindrucksvoll die Geschwindigkeit des C 64 demonstriert, denn der Uranzähler bewegt sich so schnell, daß es unmöglich ist, ihn abzulesen, und verschwindet dann wieder. Jetzt wiederholt sich der ganze Vorgang mit steigendem Schwierigkeitsgrad, d.h. stärkere Magnetfelder, mehr Laserbatterien usw. Obwohl die relativ friedliche Spielidee sowie die Verteilung der Handlung auf mehrere Schauplätze Beifall verdienen, ist das Spiel dennoch langweilig, weil die Spielaktionen entweder viel zu leicht oder viel zu schwierig sind und die echte "Action" eben fehlt.

System: C64
Hersteller: Eurogold
Audi Klein

Empire

Dieses Weltraumspiel unterscheidet sich von anderen Vertretern seiner Art durch eine einmalige Hintergrundhandlung. Der Spieler hat als Raumschiffpilot die Aufgabe, ein großes galaktisches Imperium aufzubauen. Er muß mit Sternensbasen aus Dutzenden von Sonnensystemen verhandeln, gegen Außerirdische kämpfen, Seuchen überleben und eine

Vielzahl von Spezialmissionen bewältigen. Beim Flug durch die Galaxien steht ihm ein Bord-Computer (Meldung in Deutsch) zur Verfügung, der viele wichtige Daten anzeigt, wie z.B. den Energievorrat des Raumschiffs, die Temperatur, die Radioaktivität, die beim Befördern bestimmter Güter zum Strahlentod führen kann, den Schild-Status sowie die



Richtung zur nächsten Sternbasis oder zu einem bestimmten Zeitpunkt (Marker), den man in einer Karte selbst setzen kann. Beim Flug durch die Sonnensysteme zeigt der Bord-Computer außerdem die Richtung zum nächsten System. Ein Treibstoff-Meter gehört ebenfalls zur Ausrüstung des Piloten.

Für die Lösung der Aufgabe, die Gründung eines Imperiums, benötigt man nun Material, das auch im All nicht überall anzutreffen ist. Um sich dieses Material zu beschaffen, fliegt man durch ein Warp-Loch und gelangt schließlich auf einen Planeten im betreffenden Sonnensystem. Von dort wird man gemeinerweise mit Raketen beschossen. Ist man allen Raketen ausgewichen oder hat sie gar zerstört, kann man mit einigem Geschick Fracht (Mineralien, radioaktive Erze und Arbeiter) aufsameln und zur zuständigen Sternbasis befördern. Bei ausreichendem Angebot an Material kann sie Gegenstände produzieren, die zur Gründung des Imperiums benötigt werden. Manche Sternbasen bieten Spezialmissionen an, die innerhalb eines gewissen Zeitlimits ausgeführt werden müssen. Auf der Reise durch das jeweilige Sonnensystem ist die System-Karte behilflich, auf dem "kleinen Trip" zwischen

den Systemen ist die Galaxis-Karte nützlich. Als besonderen Gag bietet die System-Karte eine Zoom-Option, mit der man den jeweiligen Abschnitt, in dem man sich befindet, vergrößern kann.

"Empire" ist ein Spiel mit einem recht anspruchsvollen Schwierigkeitsgrad. Schade ist nur, daß keine High-Score-Liste angelegt werden kann und der Schwierigkeitsgrad nicht frei wählbar ist. Ein Score wird nicht angezeigt; am Ende des Spiels erfährt der Spieler lediglich, wieviel Prozent von "Empire" er erreicht hat. Die Grafik ist gelungen, dafür stört der schlechte Sound bereits nach fünf Minuten ganz gewaltig. Zum Glück läßt sich das dauernde Gedudel mit einem Druck auf die Funktionstasten abstellen. Positiv ist außerdem zu vermerken, daß sich der jeweilige Spielstand abspeichern läßt und das Spiel nicht wie andere Weltraumspiele (z.B. "Space Invaders") in sinnlose Schachereien ausartet. Die Meldungen im Spiel sind – wie bereits erwähnt – in Deutsch geschrieben. "Empire" ist somit ein durchaus interessantes Spiel mit einem etwas vom Standard abweichenden Hintergrundgedanken.

System: C 64
Hersteller: Firebird-Software
Stefan Sebach

Tips und Tricks zur Hardware des C 64 und der Floppy 1541

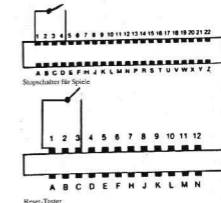
Diesmal stellen wir Ihnen drei Utilities vor, die Ihnen die Arbeit mit Ihrem Computer erleichtern sollen. Bevor Sie jedoch zu Ihrem Lötkolben greifen und sich an die Arbeit machen, müssen folgende Punkte klargestellt werden:

1. Durch das Öffnen verlieren Sie Ihren Garantieanspruch.
2. Wenn Sie im Löteten nicht geübt sind, proben Sie es erst einmal an einem Kabel!
3. Verwenden Sie kein Lötblei!
4. Überprüfen Sie jeden Arbeitsschritt genau!

So, und jetzt können wir zu den Bauanleitungen.

Einbau eines Spiele-Stop-Schalters

Mit diesem Schalter können Sie jedes Programm anhalten und zu einem anderen Zeitpunkt wieder fortsetzen. Benötigt werden zwei dünne Kabel und ein Schalter. Schalten Sie Ihren C 64 aus und entfernen Sie alle Kabel (Netzkabel, Floppykabel usw.). Drehen Sie Ihren Computer um und lösen Sie die Schrauben der Unterseite. Jetzt können Sie das obere Gehäuseteil vorsichtig hochziehen. Ziehen Sie den Stecker, der die Tastatur mit dem Computer verbindet (linke Platinen-



seite) und den Stecker für die LED (rechte Platinenseite) vorsichtig heraus, damit Sie das obere Gehäusestück vollständig abnehmen können.

In der rechten oberen Ecke der Platine sehen Sie nun den Modulport (Expansionsport). Hier löten Sie jeweils ein Kabel an Pin 1 (Masse) und Pin 4 (Interrupt) an (Bild 1). Vergewissern Sie sich, daß die Kabel sauber angelötet sind. Sie dürfen auf keinen Fall die Platine oder einen anderen Pin berühren. Danach müssen Sie nur noch die Kabel nach außen führen und durch einen Schalter verbinden.

Der Einbau eines Reset-Tasters

Ein Reset-Taster dient dazu, den Computer in den Ursprungszustand zu setzen, ohne jedoch den Speicher zu löschen (Kaltstart). Benötigt werden dazu zwei dünne Kabel und ein Taster. Entfernen Sie das Gehäuse Ihres Computers und ziehen Sie Tastatur- und LED-Stecker (wie in der ersten Bauanleitung beschrieben). Auf der oberen linken Seite der Platine sehen Sie den Userport. Löten Sie jeweils an Pin 1 (Masse) und Pin 3 (Reset) ein Kabel an (Bild 2). Vergewissern Sie sich auch hier wieder, daß die Kabel sauber angelötet sind, und führen Sie diese nach außen. Anschließend müssen Sie nur noch die Kabel mit einem Taster verbinden, und der Reset ist fertig.

Hardwaremäßiges Ausschalten der Lichtschränke in der Floppy

Wie Sie ja sicher wissen, kann man einseitige Disketten in den meisten Fällen auch beidseitig beschreiben, indem man ein zweites Loch an der linken Seite einwickelt. Dies wird überflüssig, wenn man die Lichtschränke der Floppy ausschaltet. Benötigt werden dazu wieder zwei Kabel und ein Schalter.

Schalten Sie Ihre Floppy aus und ziehen Sie den Netzstecker. Lösen Sie die Schrauben an der Unterseite der Floppy und öffnen Sie das Gehäuse. Auf der linken Platinenseite sehen Sie nun mehrere Stecker. Uns interessiert der Stecker mit der Bezeichnung P6. Um den Schreibschutz auszuschalten müssen Sie die beiden Kabel mit den Farben Orange und Lilas verbinden (Schalter). Vor-

Ihr direkter Draht zur Redaktion:
Telefon 0 72 52 / 4 29 48
Für Briefe:
Computer Kontakt
Postfach 16 40
7518 Bretten

sicht! Wenn der Schalter aktiviert ist, haben auch die Schreibschutzauflöcher keine Funktion mehr, und wichtige Daten können überschrieben oder gelöscht werden.

Gerold Herr

Grafikkurs für den C 64

3. Teil

Heute wenden wir zum ersten Mal die bisher erworbenen Kenntnisse an, um zwei nützliche Routinen zu entwerfen, die Sie in Ihren eigenen Programmen problemlos verwenden können.

Anwendung 1: Screen-Switch

Die erste Routine besteht aus einem Maschinenprogramm, mit dem wir zwei Bildschirmseiten verwalten können, die untereinander völlig unabhängig sind. Als praktische Anwendung kann man sich zum Beispiel vorstellen, daß die erste Bildschirmseite dazu verwendet wird, um ein Hauptmenü darzustellen, und die zweite Seite, um die jeweils verschiedenen Unterprogramme anzuzeigen. Ist ein Unterprogramm beendet, muß das Hauptmenü nicht erst wieder von Grund auf neu konstruiert werden, sondern es wird einfach wieder die erste Bildschirmseite aktiviert. Besonders sinnvoll läßt sich die im folgenden vorgestellte Routine verwenden, wenn Informationen dargestellt werden sollen, deren Bildschirmvorbereitung normalerweise zuviel Zeit in Anspruch nimmt, so daß es sich nicht lohnt, dies immer wieder neu durchzuführen. In einem solchen Fall sollten die benötigten Informationen einmal dargestellt und dann im Bedarfsfall immer wieder ein- bzw. ausgeblendet werden können.

Welche Anforderungen müssen nun an die Routine zur Verwaltung zweier Bildschirme gestellt werden? Zunächst einmal ist klar, daß man zwei Speicherbereiche von jeweils 1000 Byte benötigt (25 Zeilen zu je 40 Bildschirmzeilen), um zwei unabhängige Bildschirmseiten darstellen zu können. Zwischen diesen beiden Bereichen soll später in der Art eines Flip-Flop umgeschaltet werden. Weiterhin muß für jeden Bildschirm ein eigener Farbspeicher zur Verfügung stehen, damit jedes Zeichen auf dem Bildschirm in einer individuellen Farbe dargestellt werden kann. Wenn dann der jeweilige Bildschirm inaktiv geschaltet wird, muß der dem Bildschirm zugehörige Farbspeicher abgelegt werden, damit die Farbinformation nicht verlorengeht. Das wäre nämlich der Fall, wenn in den jeweils anderen Bildschirm geschaltet wird, da sich die beiden Bildschirme einen Farbspeicher teilen müssen. Und schließlich muß man sich, wenn von Bildschirm A auf Bildschirm B umgeschaltet wird, die aktuelle Cursorposition von Bildschirm A merken. Das ist sinnvoll, wenn man wieder zu Bildschirm A zurückkehren will, denn dann sollte sich der Cursor natürlich an der alten Stelle befinden.

Mit diesen Überlegungen können wir nun darangehen, das Problem zu programmieren. Dabei empfiehlt sich der Einsatz von Maschinensprache, damit die Routine später universell einsetzbar ist. Außerdem lassen sich bestimmte Operationen leider nur in Maschinensprache durchführen, und auch die Geschwindigkeit des fertigen Maschinenspracheprogramms ist nicht zu unterschätzen. Wer sich in der Prozessorsprache des C 64 nicht so gut auskennt, der braucht an dieser Stelle nicht zu verzagen, denn ein Basic-Lader für die Initialisierung des eigentlichen Programms ist in Bild 1 zu finden.

Zunächst ein paar Worte bezüglich der verwendeten Adressen und der gewählten Speicherkonfiguration. Das Maschinenprogramm beginnt bei \$9800 (*38912), der erste Bildschirm wird bei \$8000, der zweite bei \$8400 initialisiert. Der Farbspeicher für den ersten Bildschirm beginnt bei \$9000 (*36864) und belegt 4*256 Byte, weswegen die zweite Bildschirmseite bei \$9400 (*37888) beginnt. Wenn Sie nochmals in unseren letzten Kursteil hineinschauen, werden Sie sehen, daß die Adressen nicht zufällig so wie hier beschrieben gewählt wurden, sondern sich auf die Beispiele im zweiten Teil des Kurses beziehen. Nach dieser kurzen Einführung

definieren wir nun in Assemblersprache einige Konstanten, auf die man später mittels des vergebenen Namens zurückgreifen kann. Das Vereinfacht das Lesen des Listings wesentlich. Hier nun die Konstanten:

```
03 CIA_1 .DE $DD00 ; Basisadresse des CIA 1, Beschreibung
; siehe Grafikkurs Teil 2
04 VL_ZEICH .DE $3248+24 ; Register 24 des VIA, Video-RAM &
; Zeichengenerator
05 BILDZEIG .DE 648 ; Speicherzelle für Zeiger auf aktuellen
; Bildschirmfang, Hi-Byte
06 SPALTE .DE 211 ; Zeiger auf aktuelle Spalte
07 ZF_H F .DE 214 ; Zeiger auf aktuelle Zeile
08 SETCURS .DE $8732 ; Betriebssystemroutine, setzt den Cursor
; gemäß den Werten in ZEILE und
; SPALTE
```

Die Zahlen ganz am Anfang jeder Zeile sind Zeilennummern, damit auf bestimmte Zeilen innerhalb dieses Kursteils leichter Bezug genommen werden kann.

Zunächst beschäftigen wir uns mit der Initialisierung, d.h. mit den notwendigen Operationen, die durchgeführt werden müssen, damit später die Umschaltung mittels eines Aufrufs durchgeführt werden kann. Hierzu dienen die folgenden Maschinensprachebefehle, die dasselbe wie die vorgestellte Basic-Routine unter der Überschrift "das Video-RAM" im zweiten Teil dieses Kurses bewirken:

```
10 INIT LDA CIA_1
11 AND +%(11111)00
12 ORA +%(00000001) ; Block 1 aktiv schalten
13 STA CIA_1
14 LDA +%(00000100) ; Zeichengenerator auf Offset
; +496 = $9000
; jedoch keinen Offset für Video-RAM
; = $8000
15
16 STA VL_ZEICH
17 LDA #128
18 STA BILDZEIG ; Bildschirmzeiger korrigieren
```

Mit dieser Befehlssequenz ist die eigentliche Initialisierung schon abgeschlossen, und die aktuelle Bildschirmseite beginnt nun bei \$8000. Wie im vorhergehenden Kursteil könnten Sie nun durch POKE \$2768,42; POKE \$5296,1 in der linken oberen Ecke des Bildschirms einen weißen Stern erzeugen. Innerhalb der Initialisierungsroutine wird jetzt noch eine Cursorpositionstabelle eingerichtet, in der die aktuellen Cursorpositionen nach Zeile und Spalte festgehalten werden. Aus diesem Grund benötigt man für 2 Bildschirme 4 Byte, die jetzt belegt werden:

```
19 LDY #3
20 LDA #0
21 LOOP_001 STA CURSOR_Y ; Cursorpositionstabelle initialisieren
22 DEY
23 BPL LOOP_001
24 RTS
```

Wenn man nun von der gerade aktivierten Bildschirmseite zur deaktivierten schalten will, muß man sich überlegen, welches Register zur Bestimmung des aktuellen Zustands herangezogen werden soll. Um das festzustellen, sind in Bild 1 die Inhalte der für die

Register	symbolischer Name	Seite 1, \$8000	Seite 2, \$8400
56576	CIA_1	XXXX XX01	XXXX XX01
53272	VL_ZEICH	0000 010X	0001 010X
648	BILDZEIG	# 128	# 132

Bild 1: Relevante Register und ihre Inhalte

se Routine relevanten Speicherzellen und deren Inhalte in Abhängigkeit von der aktivierten Bildschirmseite in einer Tabelle aufgeführt. Hier steht X für ein Bit, dessen Wert unbekannt ist. Die ersten beiden Angaben sind binär zu lesen, die letzte dezimal. Offenbar kann man zur Abfrage des aktuellen Zustands das Register CIA_1 oder aber das Register BILDZEIG verwenden. Der Einfachheit halber wird hier das Register BILDZEIG gewählt:

```
26 FLIP LDA BILDZEIG ;feststellen, ob im 1. oder 2. Bildschirm
27 AND +%0000100
28 PHA ;und merken
```

Anhand von Bit 2 läßt sich eindeutig bestimmen, ob man sich zum Zeitpunkt des Aufrufs der Routine in dem ersten oder zweiten Bildschirm befand. Als nächste Aktion muß vor der Umschaltung auf die komplementäre Bildschirmseite die Cursorposition des noch aktiven Schirms gerettet werden:

```
29 LSR A ;Accu enthält #0 oder #2
30 TAY ;diesen Wert als Y-Index wählen
31 LDA *SPALTE ;die aktuelle
32 STA CURSOR_Y ;Cursorposition
33 LDA *ZEILE ;in der Tabelle
34 INY ;festhalten.
35 STA CURSOR_Y
```

Danach ist es notwendig, das Farb-RAM des aktivierten Bildschirms zu sichern. Das geschieht mit einer Verschiebungsroutine, die diesen RAM-Bereich kopiert. Da diese Routine zweimal mit unterschiedlichen Werten aufgerufen wird, muß man angeben, wo der zu kopierende Speicherbereich beginnt und wohin dieser Speicherbereich kopiert werden soll:

```
36 LDA *$D8 ;Beginn des Farb-RAM
37 STA VON+1
38 PLA ;gemerkten Wert abrufen
39 PHA ;und weiter merken
40 ORA *$90 ;es wird entweder nach $9000 oder
41 STA NACH+1 ;aber nach $9400 kopiert
42 JSR SCHIEBEN ;Verschiebung ausführen
```

Damit sind alle Operationen ausgeführt, die den zu deaktivierenden Bildschirm betreffen. Jetzt kann die Umschaltung und damit die Aktivierung des zweiten Schirms erfolgen:

```
43 LDA VL_ZEICH
44 AND +%00001111 ;Video-RAM vorbereiten
45 STA VL_ZEICH
46 PLA ;Bildschirmalt wiederholen
47 EOR +%0000100 ;und umschalten
48 PHA
49 ASL A ;auf Bitposition 4 schieben
50 ASLA
51 ORA VL_ZEICH ;neue Anfangsadresse
52 STA VL_ZEICH ;des Video-RAM setzen
```

Damit ist an dieser Stelle die komplementäre Bildschirmseite aktiviert. Nun muß der Bildschirmzeiger entsprechend der Tabelle in Bild 1 korrigiert werden:

```
53 PLA
54 PHA
55 ORA *$80 ;High-Byte des Bildschirm-Beginns
56 STA BILDZEIG ;korrigieren
```

Abschließend sind noch zwei Dinge zu erledigen: Zum einen muß das Farb-RAM des zweiten Schirms restauriert werden, zum anderen ist ebenfalls der Cursor für diese Bildschirmseite zu positionieren. Hier die entsprechenden Maschinensprachebefehle in Assembler-Notation:

```
57 ORA *$90 ;Farb-RAM aus RAM
58 STA VON+1 ;holen und nach
59 LDA *$D8 ;$D800
60 STA NACH+1 ;kopieren
61 JSR SCHIEBEN
62 PLA ;außerdem Cursorposition
63 LSR A ;aus der Tabelle
64 TAY ;holen
65 LDA CURSOR_Y ;und in die Register
66 STA *SPALTE ;für Zeile und
```

```
67 INY ;Spalte schreiben
68 LDA CURSOR_Y
69 STA *ZEILE
70 JMP SETCURSOR ;Cursor auf neue Position bringen
```

Mit dem letzten Befehl wird der Cursor gemäß den Werten für Zeile und Spalte aus der Cursor-tabelle neu gesetzt. Damit ist der Flip-Flop-Umschalter ausgeführt. Der Vorteil dieser Routine liegt nun darin, daß bei einem nochmaligen Aufruf derselben Routine wieder in den ursprünglichen Bildschirm zurückgesprungen wird. Es bleibt das Unterprogramm SCHIEBEN nachzutragen, welches zweimal aufgerufen wird:

```
72 SCHIEBEN LDA #0 ;Blockverschiebung
73 STA *$22
74 STA *$58
75 VON LDA *$XX ;XX wird durch das aufrufende Programm
angegeben
76 STA *$23
77 NACH LDA *$YY ;YY wird durch das aufrufende Programm
angegeben
78 STA *$59
79 PAGES LDX *$04 ;Anzahl der zu verschiebenden Seiten
80 LOOP_003 LDY *$00 ;Schleife
81 LOOP_002 LDA ($22),Y
82 STA ($58),Y
83 INY
84 BNE LOOP_002
85 INC *$23
86 INC *$59
87 DEX
88 BNE LOOP_003
89 RTS
```

Damit ist die Routine abgeschlossen. Der Vollständigkeit halber wird nun noch die Tabelle für die Cursorpositionen definiert:

```
90 CURSOR .DS4 ;4 Byte für Cursorpositionen reservieren
```

Das hier vorgestellte Assembler-Listing können Sie ohne Probleme mit einem eigenen Assembler neu eingeben und in Maschinensprache übersetzen, falls Sie einen solchen besitzen. Einen Basic-Lader für das eigentliche Maschinenprogramm finden Sie in Bild 2. Er ist selbsterklärend, weswegen hier auf das Programm nicht weiter eingegangen wird.

Ist das Maschinenprogramm durch das Basic-Programm erzeugt, wird die Routine INIT automatisch ausgeführt (SY: 38912). Sie kann im Bedarfsfall immer wieder aufgerufen werden. Die Umschaltung zwischen den beiden Bildschirmseiten erfolgt durch Eingabe von SYS 38943. Um einen ersten Eindruck über den Verwendungszweck für die Flip-Flop-Routine zu gewinnen können Sie das Listing aus Bild 3 abtippen. Sie können auch eigene Ideen im Umgang mit dieser Routine entwickeln.

Wir wenden uns nun einem weiteren, sehr interessanten Thema im Zusammenhang mit der Grafikprogrammierung zu: der Konstruktion von neuen Zeichen.

Anwendung 2: Zeichensatzprogrammierung

Haben Sie sich auch schon einmal darüber geärgert, daß Sie auf Ihrem Bildschirm nicht die deutsche Umlaute darstellen können. Für eine gute Textverarbeitung bildet aber gerade der deutsche Zeichensatz die Grundlage. Der C-64 besitzt 2 Zeichensätze zu je 256 Zeichen. Der erste umfaßt die Großbuchstaben und die Grafikzeichen, der zweite die Groß- und Kleinbuchstaben. Beide Zeichensätze sind vollständig im Anhang des C-64-Handbuchs aufgeführt (in der Tabelle der Bildschirmcodes). Die Umschaltung zwischen diesen beiden Zeichensätzen erfolgt normalerweise durch die gleichzeitige Betätigung der Commodore- und der Shift-Taste.

Bereits im ersten Teil dieses Kurses wurde beschrieben, wie sich ein dargestelltes Zeichen auf dem Bildschirm zusammensetzt. In Bild 4 wird am Beispiel des Zeichens A dieser Sachverhalt noch einmal gezeigt. Für jedes Zeichen steht grundsätzlich ein 8x8-Punkte-Feld zur Verfügung, innerhalb dessen man jeden einzelnen

Bild 2: MC-Generator

```

1 PR=0:FOR I=38912 TO 39066:READ A:POKE I,A:PR=PR+A+1:NEXT
2 IF PR<>19156 THEN PRINT"PRUEFSUMMENFEHLER!":STOP
3 SYS 38912:POKE 56,128:POKE 55,0:CLR
4 PRINT"(CLR)SCREEN- SWITCHING IST AKTIVIERT!"
5 PRINT"(DOWN)UMSCHALTUNG MIT SYS 38943"
6 PRINT"(DOWN)INITIALISIERUNG MIT SYS 38912":END
7 DATA 173,000,221,041,252,009,001,141
8 DATA 000,221,169,004,141,024,208,169
9 DATA 128,141,136,002,160,003,169,000
10 DATA 153,155,152,136,016,250,096,173
11 DATA 136,002,041,004,072,074,168,165
12 DATA 211,153,155,152,165,214,200,153
13 DATA 155,152,169,216,141,129,152,104
14 DATA 072,009,144,141,133,152,032,122
15 DATA 152,173,024,208,041,015,141,024
16 DATA 208,104,073,004,072,010,010,013
17 DATA 024,208,141,024,208,104,072,009
18 DATA 128,141,136,002,009,144,141,129
19 DATA 152,169,216,141,133,152,032,122
20 DATA 152,104,074,168,185,155,152,133
21 DATA 211,200,185,155,152,133,214,076
22 DATA 108,229,169,000,133,034,133,088
23 DATA 169,208,133,035,169,224,133,089
24 DATA 162,004,160,000,177,034,145,088
25 DATA 200,208,249,230,035,230,089,202
26 DATA 208,240,096

```

Bild 3: Flip-Flop-Routine

```

10 IF PEEK(38912)<>173 AND PEEK(56)<>128 THEN PRINT"BITTE ERST INITIALIS
IEREN!":STOP
11 POKE 53280,14:POKE 53281,6:SYS 38912
12 PRINT"(CLR,WHITE)DER(SPACE,GREY 3)ERSTE(SPACE,GREY 2)BILDSCHIRM IST A
KTIV":GOSUB 24
13 SYS 38943:PRINT"(CLR,WHITE)JETZT(SPACE,LIG.BLUE)IST(SPACE,GREY 2)DER
ZWEITE BILDSCHIRM AKTIV":GOSUB 24
14 FOR J=1 TO 5:SYS 38943:GOSUB 24:NEXT J
15 PRINT"(2DOWN,WHITE)GUTEN":PRINT"(2DOWN,LIG.BLUE)DAS IST BILDSCHIRM(SP
ACE,WHITE)1":GOSUB 24
16 SYS 38943:PRINT"(2DOWN,LIG.BLUE,6SPACE)TAG!":PRINT"(2DOWN,WHITE)DAS I
ST BILDSCHIRM(2SPACE,LIG.BLUE)2":GOSUB 24
17 FOR J=1 TO 5:SYS 38943:GOSUB 24:NEXT J
18 FOR I=1 TO 100:SYS 38943:NEXT:GOSUB 24
19 FOR I=32768 TO 32768+1000:POKE I,45:POKE I+22528,12:NEXT
20 SYS 38943
21 FOR I=33792 TO 33792+1000:POKE I,43:POKE I+21504,15:NEXT
22 FOR J=1 TO 5:SYS 38943:GOSUB 24:NEXT J
23 FOR I=1 TO 100:SYS 38943:NEXT:GOSUB 24:PRINT"(CLR)":END
24 FOR I=1 TO 1000:NEXT:RETURN

```

Bit	7	6	5	4	3	2	1	0	Inhalt
Byte 0				●	●				24
Byte 1			●	●	●	●			60
Byte 2	●	●					●	●	102
Byte 3	●	●	●	●	●	●	●		126
Byte 4	●	●					●	●	102
Byte 5	●	●					●	●	102
Byte 6	●	●					●	●	102
Byte 7									0

Bild 4: Bildschirmraster

Punkt setzen kann. Das gesamte Feld ist in 8 Byte zu je 8 Bit organisiert. Für ein gesetztes Bit muß dabei an der entsprechenden Stelle der Bitwert 1 geschrieben werden. Für das Zeichen A in unserem Beispiel sind die Inhalte der Bytes 0 bis 7 aufgeführt. Auf die Berechnungsweise wird an dieser Stelle nicht mehr gesondert eingegangen; im Anfangskapitel des Grafikforums können sie gegebenenfalls Ihre Kenntnisse über die Art und Weise der Berechnung wieder auffrischen.

Um nun einzelne Zeichendaten zu ändern, muß die Basisadresse des Zeichengenerators bekannt sein; sie liegt bei \$D000 (+\$5248). Und das bringt ein Problem mit sich: Weiter oben wurde die Basisadresse vom CIA 1 mit \$D000 angegeben. Damit liegen der Input/Output-Baustein und der Zeichengenerator an derselben Adresse. Aus diesem Grund sind die Daten des Zeichengenerators normalerweise nicht lesbar. Trotzdem kann man aber mit Hilfe eines Interrupt-Bits in einem Kontrollregister vom CIA 2 den Zeichengenerator einblenden. Das werden wir später ausnutzen, um den Zeichengenerator im Rechner in einen neuen Speicherbereich zu verschieben. Da der originale Zeichengenerator selbstverständlich als ROM ausgeführt ist, muß man notwendigerweise diesen in einen Bereich des RAM kopieren, damit er dort verändert werden kann. Wie viele Daten sind aber zu kopieren?

Ein Zeichen besteht aus einem Block von 8 Byte. Es existieren zwei komplette Zeichensätze zu je 256 Byte, wie oben angeführt, so daß der Zeichengenerator 4 KByte belegt. Wenn diese 4 KByte kopiert sind, dann muß man selbstverständlich die Register des VIC II benutzen, um diesen Zeichensatz zu aktivieren. Natürlich kann der kopierte Zeichensatz durch entsprechende POKE-Befehle manipuliert werden. Mit Hilfe dieser Vorgehensweise werden so schließlich auch die deutschen Umlaute entstehen. Vorher ist aber noch ein wenig Berechnungsarbeit erforderlich.

Zu Beginn ist zu überlegen, wo der neue Zeichensatz abgelegt werden soll. Da Zeichengenerator und Video-RAM im selben 16-K-Block untergebracht werden müssen, sind auch dafür Berechnungen notwendig. In unserem Beispiel legen wir das Video-RAM auf Adresse \$4C00 und den Beginn des Zeichengenerators auf \$5000. Damit dieser Bereich später nicht von einem Basic-Programm überschrieben werden kann, wird der obere RAM-Bereich gesperrt.

In Bild 5 sehen Sie die Zeichen, die in den neuen Zeichensatz implementiert werden sollen. Zusätzlich zu den deutschen Umlauten wurde als Sonderzeichen noch das wohlbekannteste Copyright-Zeichen aufgenommen. Die dezimalen Werte der einzelnen Bytes eines Blocks sind ebenfalls aufgeführt. Jetzt muß man sich nur noch überlegen, welche Zeichen auf der Tastatur den neuen Zeichen weichen sollen. Da die in Bild 5 dargestellten Zeichen ihre Hauptanwendung in der Textverarbeitung finden, werden wir hier nur die Zeichen des Klein-/Großbuchstabensatzes verändern.

Wenn durch B die Basisadresse des kopierten Zeichengenerators bezeichnet wird, durch N die Nummer des Zeichensatzes (0 oder 1) und durch Z die laufende Nummer des Zeichens, das verändert werden soll (im Bereich 0 bis 255), dann bestimmt man die Basisadresse BA dieses Zeichens mit Hilfe der Formel

$$BA = B + N * 2048 + Z * 8$$

Ist die Basisadresse BA ermittelt, so muß man einschließlich dieser Speicherzelle die 8 Byte des Blocks ändern, der dieses Zeichen bildet. Damit ist unser "Handwerkzeug" schon komplett, und wir können damit beginnen, die eigentliche Routine zu entwerfen. Dazu muß zunächst der Zeichengenerator kopiert werden:

- ```
10 POKE 56,76: POKE 55,0: CLR :REM oberen Basic-Bereich
 sperren
11 POKE 56334,127: POKE 1,51: REM Zeichengenerator kann
 gelesen werden
12 FOR I=53248 TO 57343: REM 4 KByte des Zeichengenerators
 kopieren
13 POKE I-32768,PEEK (I)
```

14 NEXT I

15 POKE 1,55: POKE 56334,129: REM Zeichengenerator wieder
 ausblenden

Durch diese Zeilen wurde der CIA 1 zeitweilig ausgeblendet, damit ein Zugriff auf den Zeichengenerator erfolgen konnte. Nachdem sämtliche Zeichendaten kopiert sind, muß der ursprüngliche Zustand wieder hergestellt werden, da der CIA 1 ansonsten keine seiner für den normalen Betrieb notwendigen Aufgaben erfüllen kann. In den nächsten beiden Zeilen wird dann der 16-K-Byte-Block angewählt, in dem das Video-RAM und der Zeichengenerator liegen. Außerdem werden die notwendigen Informationen in das Register #24 des VIA II geschrieben, um den Beginn des Bildschirms auf \$4C00 und die Basisadresse des zu manipulierenden Zeichensatzes auf \$5000 zu legen. Zusätzlich muß in das Register BILDZEIG der Wert \$4C (+76) geschrieben werden:

- ```
16 POKE 56576,(PEEK(56576) AND 252) OR 2:REM 16-
    K-Byte-Block wählen
17 POKE 53272,48 + 4: POKE 648,76:REM Zeichengenerator
    und Video-RAM
```

Jetzt können die neuen Zeichendaten an die entsprechenden Stellen geschrieben werden, um auf diese Art und Weise einen neuen Zeichensatz zu erhalten. Die Tabellenübersicht in Bild 6 verdeutlicht den Zusammenhang zwischen dem neuen und dem alten Zeichensatz.

Durch unsere Manipulationen werden natürlich Zeichen des ursprünglichen Schriftsatzes gelöscht. Die in der Tabelle dargestellte Belegung wurde so gewählt, daß die deutsche Normbelegung für diese Tasten aktiviert wird. In unserem Programm können die Zeichendaten für den Zeichensatz am einfachsten über DATA-Zeilen eingelesen werden. Dabei wird vorausgesetzt, daß die Basisadresse BA des jeweiligen Zeichens bekannt ist. Aus diesem Grund beginnt auch jeder Datensatz für ein Zeichen mit der Basisadresse BA:

- ```
18 FOR J=0 TO 7:REM 8 Zeichen zu bearbeiten
19 READ BA: FOR I=BA TO BA + 7: REM 8 Zeichendaten zu
 setzen
20 READ Z: POKE I,Z
21 NEXT I: NEXT J: END
```

Abschließend nun noch die Daten, die sich auf die Angaben in Bild 5 stützen:

- ```
22 DATA 22528,0,102,0,102,102,102,62,0
23 DATA 22744,102,0,60,102,102,102,60,0
24 DATA 22752,124,102,124,102,124,96,96,0
25 DATA 22760,195,60,102,126,102,102,102,0
26 DATA 22992,0,102,0,60,102,102,60,0
27 DATA 23000,102,0,60,6,62,102,62,0
28 DATA 23368,60,66,153,161,161,153,66,60
29 DATA 23504,102,0,102,102,102,102,60,0
```

Die Zeilen 22 bis 29 bestehen jeweils aus der Basisadresse BA des Zeichens und den 8 Zeichendaten, die das neue Zeichen formen. Wenn Sie das Programm ordnungsgemäß eingetippt haben, müssen Sie nach dem Start des Programms mittels der Commodore- und der Shift-Taste den zweiten Schriftsatz aktivieren. Danach erscheinen die neuen Zeichen bei Betätigung der entsprechenden Taste.

Wenn Sie die Routine verfeinern wollen, dann können Sie die Zeichendaten so ergänzen, daß die Tastenbelegung der DIN-Tastatur entspricht. Dazu müssen nur noch die beiden Zeichen "<" und ">" durch die Zeichen "!" und ":", ersetzt werden. Außerdem könnte man die Tabelle noch so ergänzen, daß die neuen Schriftzeichen invers darstellbar sind. Wenn Sie einmal mit dem Cursor auf ein neu definiertes Zeichen fahren, wird dieses Zeichen abwechselnd normal und invers dargestellt. Da aber zunächst die Definitionen für den inversen Schriftsatz noch nicht vorhanden sind,

wird als inverses Zeichen das der ursprünglichen Tastenbelegung entsprechende inverse Zeichen angezeigt. Wenn Sie also unsere DATA-Tabelle um die inversen Zeichen ergänzen, können Sie den gesamten deutschen Schriftsatz auch invers darstellen!

So, damit ist der dritte Teil des Kurses beendet. Im nächsten und letzten Teil des Grafikurses werden wir uns dann mit den verschiedenen Grafikmodi beschäftigen, die der C-64 zur Verfügung stellt. Bis dahin wünsche ich Ihnen jedenfalls viel Spaß beim Ausprobieren der diesmal vorgestellten Anwendungen.

Frank Lorenzki

Bild 5: Raster der neuen Zeichen









7 8 5 4 3 2 1 0 	Inhalt 195 80 102 126 102 102 102 0	7 8 5 4 3 2 1 0 	Inhalt 102 102 60 102 102 102 80 0
7 8 5 4 3 2 1 0 	Inhalt 102 0 60 102 102 102 80 0	7 8 5 4 3 2 1 0 	Inhalt 102 102 0 102 102 102 60 0
7 8 5 4 3 2 1 0 	Inhalt 102 0 102 102 102 102 60 0	7 8 5 4 3 2 1 0 	Inhalt 102 0 102 102 102 102 60 0
7 8 5 4 3 2 1 0 	Inhalt 124 102 124 102 124 96 96 0	7 8 5 4 3 2 1 0 	Inhalt 60 66 153 161 161 153 66 60

Bild 6: Alter und neuer Zeichensatz

Zeichennummer	altes Zeichen	neues Zeichen	Basissadresse BA des Zeichens
0	@	ü	22528
58	:	ö	22892
59	:	ü	23000
122	┘	ü	23604
27	[Ö	22744
28]	Ä	22760
28	┘	ß	22752
105	■	©	23368

Vietnam für den C 64

Ziel des Spiels ist es, möglichst weit in generisches Gebiet einzudringen, ohne den pausenlos angreifenden Kontrahenten zum Opfer zu fallen. Zunächst werden Maschinenprogramme und Grafik eingelesen, was ca. 4 Minuten dauert. Sobald Musik ertönt, können Sie Ihre Spielfigur mit dem Joystick (Port 2) bewegen. Sie sind natürlich nicht wehrlos und können durch Druck auf den Feuerknopf die feindlichen UFOs beschießen. Bei einem Treffer kassieren Sie Punkte. Sollten Sie selbst getroffen wer-

den, so erscheint eine Auswertung, aus der ersichtlich wird, wie viele Meter Sie in Feindesland vorgedrungen sind. Nun können Sie durch Druck auf die Space-Taste ein neues Spiel beginnen.

Sollten Sie sich einmal in einer ausgewogenen Situation befinden, so gelangen Sie mit der RUN-STOP/RESTORE-Taste wieder zum Titelbild. Tippen Sie dann GOTO 1300 auch ohne Cursor blind ein und nach Return beginnt das Spiel von neuem.

Christian Rohrbach

MC-Generator

```

0 POKES6,82
1 PRINT"VIETNAM"
2 PRINT" (C) BY RADDISH-SOFT GERMAN Y"
3 PRINT"PLEASE WAIT 4 MINUTES"
9 FOR I=900T0912:READQ:POKEI,Q:NEXT
10 DATA65,1,41,254,133,1,76,240,85,76,104,89,235
100 FORT=51200T051967:POKET,Q:NEXT
110 FORT=0T032:READQ:POKET+51200,Q:POKES1264+T,Q:POKES1328+T,Q:POKES1392+T,Q
120 NEXT
125 FORT=0T032:READQ:POKES1456+T,Q:POKES1520+T,Q:POKES1584+T,Q:NEXT
130 FORT=0T032:READQ:POKES1648+T,Q:POKES1712+T,Q:POKES1776+T,Q:NEXT
135 FORI=0T09:FORT=0T017:READQ:POKES1233+T+I*64,Q:NEXT:NEXT
136 FORI=1T022:READQ:POKES1840+I,Q:NEXT
139 DATA0,2,0,0,10,128,0
140 DATA10,128,0,14,192,0,2,0,0,10,128,0,10,128,0,10,128,0,10,128,0,13,192,0,10
145 DATA128,0,2,0,0,10,128,0,5,128,0,1,128,0,2,0,0,10,128,3,280,128,0,123,128
147 DATA0,46,128,0,15,192,0,10,128,0,2,0,0,10,128,0,9,64,0,9,0,2,0,0,10,128,0
149 DATA10,171,0,11,180,0,10,224,0,15,192,0,2,0,10,128,0
150 DATA10,128,0,8,192,0,8,192,0,12,192,0,12,0,0,12,0,0,10,128,0,8,128,0,8,192,0
151 DATA12,192
152 DATA0,12,192,0,12,0,0,10,128,0,8,128,0,12,128,0,12,192,0,12,192,0,0,192,0
154 DATA10,128,0,12,128,0,12,128,0,12,128,0,12,192,0,12,192,0,2,0,0,192,0,0,192,0,10,128,0,40,128,0,32
156 DATA160,0,48,60,0,48,12,0,240,48,0,10,128,0,8,128,0,8,128,0,12,192,0,12,192,0,0,192,0,3,192,0,3,192,0,10,128,0,8,160
160 DATA0,40,32,0,240,48,0,192,48,0,48,0,0,10,128,0,8,128,0,8,128,0,12,192,0,12,162 DATA192,0,3,240,0,8,128,0,10,128,0,10,0,0,12,0,0,15,0,0,15,0
163 DATA240,0,6,89,0,21,165,64,85,85,80,238,238,224,85,85,80,21,85,64,1,244
170 POKES3270,216

```

```

175 POKES3282,5:POKES3283,11
180 FORX=#32T08&5:READQ:POKEX,Q:NEXT
190 DATA20,169,51,133,1,169,0,133,95,13
3,90,133,88,169,208,133,96,169,240,133
200 DATA89,169,224,133,91,32,191,163,169
,55,133,1,88,96
210 BYBBS2:POKE50,160:POKE56576,PEEK(56
576)AND252:POKE53272,8:POKE648,192
220 B=57344:PRINT"L":CHR$(14)
230 READZ:IFZE=-1THEN1000
240 FORT=OT07:READQ:POKEB+ZE#T,Q:NEXT
250 BOTOZ30
300 DATA27,0,0,192,0,0,48,0,0,28,12,0,0,0
,48,0,0,0,29,0,3,0,0,0,192,0,30,85,85
310 DATA149,85,85,85,85,86,85,31,85,89,85,8
5,149,85,85,85,33,0,0,46,236,235,60,0,0
320 DATA35,0,3,14,58,234,234,63,195,36,0
,192,176,188,176,192,48,0,37,0,0,0,0,0
330 DATA15,58,59,58,0,0,0,48,236,187,170
,170,39,0,0,0,0,0,192,192,40,234,234
340 DATA238,250,235,58,62,15,41,186,171,
234,170,174,170,235,235,42,176,176,176
350 DATA176,240,240,240,204
360 DATA43,0,15,50,207,226,200,226,63,44
,0,235,34,255,34,136,34,255,45,0,252,47
370 DATA243,59,179,44,240,46,234,170,17,
251,174,171,171,254,47,186,174,174,255
380 DATA170,170,170,255,48,170,170,170,2
55,186,186,191,234,49,186,186,234,255
390 DATA170,170,171,254,50,171,171,171,1
90,235,234,234,255,51,0,0,0,0,15,58,63
400 DATA52,0,0,0,0,0,255,174,255,53,0,0
,0,0,0,255,174,255,54,0,0,0,0,255,170
410 DATA255,55,0,0,0,0,0,240,240,176,56,0
,3,3,0,3,3,0,57,176,176,176,176
420 DATA176,176,192,0,58,0,0,3,13,53,59,
58,58,59,0,15,245,21,81,85,245,173
430 DATA60,0,255,85,20,85,21,71,254,61,0
,192,124,87,85,30,250,170,62,0,0,0,0,192
440 DATA192,192,192,63,58,58,58,58,25
0,234,224,64,170,234,170,170,170,174,174
450 DATA171,65,170,234,171,171,170,187,1
70,170,66,171,171,171,171,171,170,234
460 DATA234,67,0,0,0,0,192,192,176
470 DATA68,0,63,53,213,85,85,85,69,20
4,119,117,85,85,85,85,70,48,3,207,125
480 DATA85,85,101,85,71,0,0,207,117,85,8
5,85,85,72,85,85,85,95,87,223,51,0
490 DATA73,85,85,117,85,119,205,15,0,74,
117,85,245,117,85,127,192,0
500 DATA75,3,0,195,195,195,195,195,7
6,0,0,0,12,12,12,12,12,99,0,195,13,61,53
510 DATA53,53,53,78,85,87,87,87,87,151,8
7,87,79,195,12,12,12,12,12,12,80,12
520 DATA48,48,48,48,48,48,81,53,213,2
13,217,213,213,213,82,87,87,87,108
530 DATA92,92,92,92,12,12,12,48,48,48
,48,48,84,48,48,48,192,192,192,192
540 DATA85,213,213,213,213,229,213,213,2
13,86,92,92,240,112,115,112,192,0
550 DATA87,48,48,48,48,204,192,192,0,88,
192,192,204,192,192,204,195,0,89,213
560 DATA213,61,53,53,15,3,0
570 DATA90,0,0,0,0,0,0,3,91,0,3,15,63,
251,235,235,235,92,0,0,192,243,192,174
580 DATA170,170,93,0,0,0,0,192,240,192,1
74,94,0,0,0,0,0,60,239,171,95,0,0,0,0,0
590 DATA15,250,170,96,0,3,3,3,207,254,17
0,170,97,192,240,188,172,175,171,170,170
600 DATA98,0,0,0,3,15,254,170,170,99,0,0
,204,255,187,187,235,235,100,0,0,0,3,15
610 DATA15,14,14,101,15,14,62,250,234,17
0,170,170
620 DATA102,171,171,171,171,171,172,172,
176,103,170,170,170,170,255,0,0,0,104
630 DATA171,171,175,190,250,58,58,58,105
,235,171,171,171,171,171,172,172,106
640 DATA0,0,0,3,63,254,250,254,107,0,0,4
8,255,254,170,170,108,14,14,14,250
650 DATA250,170,170,170,109,170,170,171,
172,172,176,192,192,110,192,192,0,0,0,0
660 DATA0,0,111,58,58,58,58,59,59,60,60,
112,176,176,192,192,0,0,0,113,239,235
670 DATA234,234,234,234,234,234,114,170,
234,255,191,170,170,170,115,171,172
680 DATA172,252,255,171,170,170,116,0,0,
12,63,251,234,170,170,117,0,0,0,204
690 DATA255,187,170,118,0,0,0,0,0,192,
240,119,0,0,0,15,15,62,62,120,0,0,0
700 DATA48,255,239,170,62,121,0,3,3,255
,254,234,234,234,122,236,236,172,172
710 DATA172,172,172,123,48,48,48,0,0,0,0
,0,0,124,234,234,234,234,234,234,234
720 DATA234,125,170,170,170,170,170,170
,170,170,126,85,85,85,85,85,85,85,85
730 DATA127,190,175,171,170,170,170,170
,170,128,190,255,235,170,170,170,170,170
740 DATA129,250,234,170,170,170,170,170
,170,130,234,234,234,234,234,233,235
750 DATA131,172,172,176,176,192,192,0,0
,132,234,234,255,0,0,0,0,133,170,170
760 DATA255,0,0,0,0,134,236,236,240,0,0
,0,0,0
770 DATA135,15,13,3,0,0,0,0,136,0,192,
124,215,53,13,3,253,137,0,3,13,53,215,0
780 DATA92,95,85,138,60,215,127,192,0,0
,0,192,139,3,13,55,220,240,0,0,0
790 DATA140,85,253,0,0,0,0,0,141,85,87
,92,240,240,240,240,142,112,92,115
800 DATA53,13,3,0,0,143,0,0,0,192,112,11
2,192,0,144,240,240,240,240,240,240
810 DATA240,145,0,0,0,0,3,3,0,146,240,
240,240,252,252,252,255,0
820 DATA147,0,0,0,0,0,4,1,17,148,0,1,21,
4,65,17,21,65,149,0,0,16,20,64,4,21
830 DATA150,0,0,1,0,4,17,5,4,151,85,81,6
9,69,65,17,20,5,152,5,68,85,85,81,69,1
840 DATA21,153,5,69,21,84,5,85,21,65,154
,64,0,0,16,64,0,68,84,155,21,81,69,21
850 DATA16,20,0,0,156,84,65,17,69,64,68,
65,1,157,68,68,68,64,4,4,65,0
860 DATA158,64,69,85,65,68,16,0,0,159,20
,85,80,20,20,0,0,0
870 DATA160,0,63,51,51,51,51,63,161,0
,3,15,3,3,3,3,162,0,63,3,3,63,48,48
880 DATA63,163,0,63,3,3,63,3,3,63,164,0,
51,51,51,63,3,3,165,0,63,48,48,63,3,3
890 DATA63,166,0,63,48,48,63,51,51,63,16
7,0,63,3,3,12,12,48,48,168,0,63,51,51
900 DATA63,51,51,63,169,0,63,51,51,63,3,
3,63
999 DATA=-
1000 AD=22000:FORI=1006T01125:READA#,PB:
C=0:FORN=1TULEN(A#)STEP2
1001 A=ASC(MID$(A#,N,1))-48:B=ASC(MID$(A
#,N+1,1))-48:A#A#7*(A#9)B#B#7*(B#9)
1002 POKEDAD,16#A#7+C#C#PEEK(AD):AD=AD+1
1003 NEXTA:PRINTI;" ";IFC>PBTHENPRINT"E
RROR":STOP
1004 PRINT"OK":NEXT

```

1005 80TD1200
 1006 DATA A90008B85638D86638D87638D88638D
 89638D8A638D8B638D8C63A9008DAC63, 3762
 1007 DATA BDAD638DAE638DAF638DD71648D7264
 8D73648D74648D7564A9018DAB63A000, 3763
 1008 DATA A9089900D89900D9A9209900C9900
 C188D0EDA90F8D18D4A96085FA96D85, 4010
 1009 DATA FCA250A000AD04DC297FF008C901F0
 0EAY2091FB4C7756A92191FB4C7756A9, 3852
 1010 DATA 2391FBCBA92491FB4C77563C8D008E6
 FCCAD01A95F85FD96185FEA000208D, 4922
 1011 DATA 56B1FDF078C901F017C902F016C903
 F015C904F014C905F013C906F0124C0E, 3739
 1012 DATA 574C64574CBA574C105B4C66584C8C
 584C1259185FD690185FDA5FE90085, 3271
 1013 DATA FE40A96085FB96D85FC18A5FB6928
 85FBA5FC69085FCCAD060A96085FB, 4933
 1014 DATA A99585FC18A5FB692885FBA5FC6900
 85FCCAD0F06018A5FB692885FBA5FC69, 4816
 1015 DATA 0085FC604C8D03208D5681FDF007AA
 20C8564C2557208D5681FDA20E45620, 3661
 1016 DATA BD56B1F88DB90218A2FB68D9028D8F
 A5FC90085FC9A9FB856A6185BD8A203, 4382
 1017 DATA A005B1BC91FB88D9F20FD5618A58C
 6905858CA58D6490858DCA4E4C8756, 4414
 1018 DATA 208D56B1FDF007AA20C8564C785720
 BD56B1FDA20E456208D56B1F8D8B902, 4079
 1019 DATA 18A5FB6A890285FBA5FC690085FCA9
 A9858CA96085BDA206A00D1BC91F88, 4511
 1020 DATA D0F920F205618A58C690858CA58D69
 00858DCA0E44C8756208D56B1FDF007, 4344
 1021 DATA AA20C8564CD157208D56B1FDA20E4
 56208D56B1F88DB90218A2FB68D90285, 4050
 1022 DATA FB85FC69085FCA9F7858CA960858D
 A204A004B1BC91FB88D9F20FD5618A5, 4726
 1023 DATA 8C6904858CA58D6490858DCA0E44C
 8756208D56B1FDF007AA20C8564C728, 4012
 1024 DATA 208D56B1FDA20E456208D56B1F8D8
 90218A2FB68D90285FBA5FC690085F, 4345
 1025 DATA A907858CA961858D8A202A005B1BC91
 FB88D9F20FD5618A58C690858CA58D, 4301
 1026 DATA 690858DCA0E44C8756208D56B1FD
 F007AA20C8564C7D8208D56B1FDA20, 4065
 1027 DATA E456208D56B1F8D8B90218A5FB68D9
 0285FBA5FC69085FCA911858CA96185, 4312
 1028 DATA 8DA203A00C81BC91FB88D9F20FD56
 18A58C690C858CA58D6490858DCA0E4, 4485
 1029 DATA 4C8756208D56B1FDF007AA20C8564C
 D338208D56B1FDA20E456208D56B1FD, 4137
 1030 DATA 8D890218A5FB68D90285FBA5FC6900
 85FCA935858CA961858D8A202A0031BC, 4178
 1031 DATA 91FB88D9F20FD5618A58C6903858C
 A58D690858DCA0E44C8756208D56B1, 4366
 1032 DATA FDF007AA20C8564C299208D56B1FD
 AA20E456208D56B1F8D8B90218A5FB68, 4064
 1033 DATA B90285FBA5FC690085FCA935858CA9
 61858DA204A005B1BC91FB88D9F20FD, 4585
 1034 DATA 5618A58C690858CA58D6490858DCA
 0E44C875678A97F8D0DCA9018D1AD, 3945
 1035 DATA A9028502A97885FBA9985FCA04460
 BD12D0A9188D11D0A9988D1403A9598D, 3964
 1036 DATA 1503584C9594C858DAD19D08D19D0
 2901F0F3C6021004A9018502A602BD6, 3040
 1037 DATA 808D21D08DA4608D12D0E000F003AC
 0059A011B9706399FFCF8D0F7A9FF8D, 4494
 1038 DATA 1DD0A00998986399F7C388D0F74C33
 58A011B9826399FFCF8D0F7A9008D1D, 4449
 1039 DATA D0A09098B26399F7C388D0F74C335B
 AD00DC2910F019AD00DC2908F049AD00, 3918
 1040 DATA DC2904F079AD00DC2901F001604CC3

5AADC763D00160A9008DC763AD83A38D, 2607
 1041 DATA B963AD86438D8A63AD00DC2904F00D
 AD00DC2908F00CA9008DC863A0A9018D, 3419
 1042 DATA C86360A9028DC86360AD8363CF9A
 0160AD8363E918A4A4A4A4A8B993C1C924, 4009
 1043 DATA 900160E8363AD8702C9038009EEB7
 02A9028D866260A9008D8702A9028D86, 3460
 1044 DATA 0260AD8363C919800160AD8363E918
 4A4A4A8A8B993C1C924900160CEB3A3AD, 3639
 1045 DATA B702C903B009EEB702A9018D8A0260
 A9008D8702A9018D860260AD8363E918, 3334
 1046 DATA 4A4A4A8A8B942C1C924900160AD8B702
 C905B006EEB7024CE65A9A008D8702A9, 3541
 1047 DATA 008D8602EEB6363EEB863EEB863EEB8
 63AD8C63C97D900A9018DC963A9008D, 4034
 1048 DATA 8C63EADA9608C917F007EEA9602032
 606020326018A5FB69285FBA5FC6900, 4083
 1049 DATA 85FCA9108DA940AC2F58A90085FDB9
 C085FEA00085F8858CA5FC858D81BC91, 4639
 1050 DATA FDB8D0F968DE6E8A5FC9C2D0EF60
 A6028D86608D11D08A96068A8A8A8A8, 5127
 1051 DATA 404CE5EADDC763F0160AD8C63F02C
 C901F014AD8963CF9D900A9018DC763, 4104
 1052 DATA 60EEB963EEB96360AD8963C9148006
 A9018DC76360EEB963EEB96360AD8A63, 4046
 1053 DATA C91EB00A9018DC76360CEB3CEBA
 6360ADC963F019B008DC963AD04DC29, 3886
 1054 DATA 038DCA63AD85638D8B63AD86638D8C
 63ADCA63F02CC901F014AD8B63CF9D90, 4244
 1055 DATA 06A9018DC96360EEB863EEB86360AD
 8B63C9148006A9018DC96360EEB863CE, 3932
 1056 DATA 8B6360AD86C63CF97F9006A9018DC963
 60EEB863EEB86360201FBC20545C089, 3513
 1057 DATA 5CA0CE8502AD8502F00160A90A08D85
 02AD8602F00EC901F04ACB702899463, 3635
 1058 DATA 8DB3610AC8702899C638D83A38D63A0C
 B7028998638D836360CEB202AD8B202F0, 4114
 1059 DATA 0160A90A8D8202AD8302F00EC901F0
 14AC8402B9A2638D8463A0AC402B9A2, 3684
 1060 DATA 638D846360AC402B9A2638D8463A0
 CEAF02ADAF02F00160A90A8D8A02AD80, 3847
 1061 DATA 02C901F0A2C902F014ACB102B9A263
 8D856360ACB102B9A2638D856360ACB1, 4046
 1062 DATA 02B9A2638D85636020C2F5C20D25D60
 ADC463F037C901F007C902F0754C758D, 3777
 1063 DATA AD8663C9809006A9008DC46363020A1
 5DA05281A7C92900160AD8A02C90590, 3641
 1064 DATA 05A9FF8D8602EEB6402EEB6360AD0A
 DC29038DC463E9018D8302AD8302F024, 3802
 1065 DATA 902F010A9F8E8D8563AD04DC293F69
 348D866360A91A8D8563AD04DC293F69, 3557
 1066 DATA 348D866360A9348D8663AD04DC297F
 69508D856360AD8563C918D006A9008D, 3462
 1067 DATA C4636020A15DA0081A7C92900160
 AD8402C9039005A9FF8D8402EEB6402CE, 3740
 1068 DATA 856360AD8563CF9E9006A9008DC463
 6020A15DA003B1A7C92900160AD8B402, 3665
 1069 DATA C9039005A9FF8D8402EEB6402EEB6402EEB63
 60A9C0858A90858A7AD8563E918A4A4, 4027
 1070 DATA 42858A7AD8563E9334A4A4A8A8A8A8
 1118A5A7692885A7AD856900858CA4C, 3755
 1071 DATA 8C8D60ADC563F037C901F007C902F0
 764CB15EAD8863C9809006A9008DC563, 3943
 1072 DATA 602AD850A5281A7C92900160AD8B1
 02C9059005A9FF8D8102EEB863, 3800
 1073 DATA 60AD04DC29038DC5638D8602C901F0
 28C900F0EDC902F010A91A8D6763AD04, 3639
 1074 DATA DC293F69348D8B6360A9F8E8D8763AD
 04DC293F69348D8B6360A9348D8B63AD, 3651

```
1075 DATA 04DC297F69508D76340AD8763C91C
B00A9008DC5636020AD5EA000B1A7C9, 3572
1076 DATA 24900160ADB102C9F3900A59FF8DB1
02EEB102CEB76360AD8763C9F9006A9, 3860
1077 DATA 00BDC5636020AD5EA0003B1A7C92490
0160ADB102C9F3900A59FF8DB102EEB1, 3681
1078 DATA 02EEB76360A99C085A8A90085A7ADB7
63E91B44A4A4A5A7ADB8663E93344A4A4, 3801
1079 DATA AAE000F01118A5A76928B5A75A8A69
0085ABC4CC85E6020F9520155C20C0, 3603
1080 DATA 5C20675B20B28520506020FC5E2032
5F20735F4C31EAADB663C97D9005A900, 3033
1081 DATA BDC463AD8663C97D9005A90085C63
ADBA63C97D900A97DB8DB63A9018DC7, 3977
1082 DATA 63AD8663C97D9005A9018DC7A6360AD
1E0DBDA663290BDO0BADA6632901D022, 3505
1083 DATA 60ADA6632902D00BADA6632904D00A
60A9008DC4632014606A9008DC56320, 3085
1084 DATA 146060ADA663290BDO160A9008DBA
6360ADAB63F0016078208CF5A01DC29, 3318
1085 DATA 10D0F9A9018DBA63C4C03A920A000
9900C9900C199008B9900D98BDOF1A0, 3801
1086 DATA 1AB9CB639930C08BDOF7A018B9E663
990C08BDOF7A013B9FD6399D6C08BDO, 4876
1087 DATA F7A005B970646A9A99D0C08BDOF5AD
7264F00EA018B911649920C18BDOF74C, 4386
1088 DATA EBCFA013B92B649920C18BDOF7A01F
B93C649998C18BDOF7A0049B863A9A0, 4312
1089 DATA 99AC18BDOF5A014B95C6499EBC18B
D0F7A9008D18D0460EAF63A003B9AC63, 4606
1090 DATA C90AD0EA90099AC6318B9AB63A901
99AB63BDD0E80E75644004B97164C9, 3924
1091 DATA 0AD00EA90099716418B97064690199
706488D0E80CEB163F00160A99A8DB1, 3647
1092 DATA 63EEB06348D0B63C9400D005A9008DB0
63ACB063A95ABD05D48D06AD48D0C0D48D, 4052
1093 DATA 0DD4A920BDO448D0B0D4A9218D04D4
8D08D4B908528D0D178D4948528D0D8A9, 3576
1094 DATA 8B528D0D84B9C8528D07D46028BF09
0910101B1C185A5B8528D5E5F60616263, 2762
1095 DATA 181D648566676767676767676896A6B
6C6D6E1D1C1B1D1CA6F707172737475, 2693
1096 DATA 7618777BDD7A7B1D7C7D7E7D7F7B0
817DB2E31C1D84858585858585858586, 3660
1097 DATA 1C1D1C46484C4D4E4F505152535455
565758593A3B3C3D3E3F404142432033, 2093
1098 DATA 34363436333436343637382E2F3031
322E2F30313239382E2F3031322E2F30, 1560
1099 DATA 31323925262728292A8788B9BA208B
6C8DBEBF202090202020919220202093, 2514
1100 DATA 949520969798999A999B999C999D9E9F0007
1E0F072A17074003074C22076221078F, 2477
1101 DATA 0807B01A07E80C07F6A007000D2207
001B0307001A12070024007002A1F07, 1003
1102 DATA 02D1007003B1807003E04070042205
07004D130700541F0700840407008F05, 849
1103 DATA 0700B0170700D4040700E81F012D04
01571901B417019B0C01B41101C92021, 1594
1104 DATA D51A01DA0101F41201100220E01000B
0701009A030100AF150100E40C031608, 1502
1105 DATA 03221F033601034C0C034D13036216
03791E03D302D321103E82203EA0703, 1571
1106 DATA 0E1003F30E03F30B03002B0B03004C
1B03008A0103009A150300B010030033, 1665
1107 DATA 030300E0B0300F321040203040719
040E904160D04230404400B04CC1304, 868
1108 DATA 5B010471180486D04471B04D0104
00D11040039B040053010400600F04, 1109
1109 DATA 064160400670504006A1404006F00
04007D1A04008A1104008B04008B19, 1109
.1110 DATA 040A05030400D31A0400E1050400EF
```

```
1705032005060605170605200B052719, 1121
1111 DATA 053514054415054002054E1B055612
055E1F056F0305710B058620059B0905, 1179
1112 DATA AS1E05B01005C20205D31905E71505
ED0B05F7020500C040500121F050011, 1695
1113 DATA 15050270E0500360B05003A1F0500
44B0500671505007903050086020500, 734
1114 DATA 211705009A1F0500A7220500AE0105
00BB170500C2005009C1F0500C0E05, 1635
1115 DATA 00B6160500D040500E31F060D2006
20026481F0605665C0E0667200671, 1422
1116 DATA 04067B01067C08067D11068B15069A
9905B20206D51906F20C6060200C0600, 1497
1117 DATA 591506009A200600D11B0600D60600
001B97489778970897DB9708972D9700, 2474
1118 DATA F7E000A77897783C7B00000000000000
000000000272892824252625202122, 1152
1119 DATA 2322212A2A2A2A2A2A2A0000000000
00000000001010102E2E2E2E10002D2D, 776
1120 DATA 2D2D2D2D2D2D000100010001000200
05090405122008011420051320130905, 540
1121 DATA 20051217093030814201309052013
090E042001020512200909D005120809, 445
1122 DATA 0E200051405122017050914200705
0B0F0D0D050E20040113201701122013, 492
1123 DATA 0308F0E2007010E1A200715142004
01132017011220170F080C20E090308, 486
1124 DATA 141320011513130512040500200801
02050E20130905202020202010150E, 546
1125 DATA 0B1405201310010305201401131405
20041215050308050E000000000005, 327
1200 AD=21000;FORI=1206TD1214;READA#,PB;
C=0;FORN=1TOLEN(A#)STEP2
1201 A=ABC(MID$(A#,N,1))-48;B=ABC(MID$(A#,
N+1,1))-48;A#A#*7*(A#);B#B#*7*(B#)
1202 POKEAD,16#A#*7*(C#*PEEK(AD);AD=AD+1
1203 NEXTA:PRINTI;" ";:IFC>PBTHENPRINT"
RROR";:STOP
1204 PRINT"OK";NEXT
1205 GOTO1300
1206 DATA 0D111113151115130D111113151111
100D1111131517151311100D0E10111, 541
1207 DATA 11110E0E0E0D0E10110E0D0E0D0B0A
0A0D0D0E0E0E0D0E10110E0D1101311, 445
1208 DATA 11110A675789E0D7EDB90A676789E0
67676D0A676789E03BEB9676D0AA26D, 3522
1209 DATA 67676767A2A20A26D67A20A2A20A
9E7F7F0A0A2A2A20A26D67A20A676D, 3552
1210 DATA 89676A7A20A0B0A0B0A0B0A0B0A0B0
040B040B0A0B0A0B0A0B0A0B0A0B0, 639
1211 DATA 06D04080A08050B050B050B050B04
0B06D04080A0B050B050B050B050B04, 230
1212 DATA 0B06D04080A4B5A84B5A84B5A84B5A8
5A84B5A84B5A84B5A84B5A84B5A84B5A8, 3399
1213 DATA 850A850A85A84B5A84B5A84B5A84B5
9E3A84B50A85A84B5A84B5A84B5A84B5, 4541
1214 DATA 9E3A84B50A84B5A84B5A, 931
1300 POKE690,1:PRINTCHR$(142);CHR$(8)
1301 FORI=45#64+49152T046#64+49152;POKEI
,255:NEXT
1302 FORI=46#64+4912T047#64+49152;POKEI,
0:NEXT;POKE46#64+49152=19,3
1303 POKE46#64+49152+22,3
1304 POKE56576,PEEK(56576)+AND251;POKE532
72,8;POKE648,192;POKE53270,216;POKE53262
,5
1305 POKE53283,15;POKE53269,244;POKE5327
6,255;POKE53277,255;PRINT"CL";
1306 POKE53286,1;POKE53287,5;POKE53288,1
0;POKE53285,11;POKE53289,10
1310 BY9900
```

Input de Luxe

Die INPUT-Funktion des C64 weist einige Mängel auf. So kann z.B. mit dem Cursor aus der Eingabezeile herausgeführt werden, bei Betätigung der CLR-Taste wird der Bildschirm gelöscht, Komma und Doppelpunkt können nicht ein-

gegeben werden usw. Dies alles wird mit "Input de Luxe" umgangen. Die Länge des Eingabefeldes und dessen Position müssen bei Aufruf des Programms angegeben werden. Nach Drücken der RETURN-Taste befindet sich die Eingabe in der Variablen IN\$. Innerhalb des Eingabefeldes können die

Cursor-Tasten sowie HOME, CLR, DEL und INST benutzt werden. Im Großschrift-/Grafikmodus werden keine Grafikzeichen angenommen. Als Underline-Cursor wird ein Sprite verwendet. Das Programm ist voll relokatable und kann daher an eine beliebige Stelle im Speicher gelogt werden (49152)

Funktionen: HOME: Cursor springt zum ersten Zeichen des Eingabefeldes; CLR: Eingabefeld wird gelöscht; DEL: Zeichen über Cursor wird gelöscht; INST: Einfügen einer Leerstelle, Cursortaste links und rechts; RETURN: Ende der Eingabe.

Susan Markowitz

```

100 REM          COMMODORE 64
110 REM
120 REM  INPUT-DELUXE - RELOKATIBEL
130 REM
140 REM AUFRUF:
150 REM SYS(STARTADRESSE),X,Y,L
160 REM X=BILDSCHIRMKOORDINATE (X)
170 REM Y=BILDSCHIRMKOORDINATE (Y)
180 REM L=LAENGE DES EINGABEFELDES
190 REM
200 REM EINGABE BEFINDET SICH IN IN$
210 REM
220 REM
230 REM LAENGE DES PROGRAMMS: 481 BYTE
240 REM
250 INPUT"STARTADRESSE";SA
260 FORI=SA TO SA+481
270 READP;POKEI,P;S=S+P;NEXT
280 IF$<>61726THENPRINT"PRUEFSUMMENFEHLER";END
290 PRINT"OK. ";END
300 DATA 032,253,174,032,158,183,224,040,176,095,134,020,169,000,133,250
310 DATA 138,010,010,010,105,024,141,000,208,133,249,144,007,173,016,208
320 DATA 009,001,133,250,032,253,174,032,158,183,224,025,176,059,138,010
330 DATA 010,010,105,058,141,001,208,134,113,162,000,134,114,134,041,169
340 DATA 040,133,040,032,087,179,138,024,101,020,133,020,133,003,144,001
350 DATA 200,200,200,200,200,132,021,152,024,105,212,133,004,032,253,174
360 DATA 032,158,183,134,252,232,024,144,003,076,072,178,160,063,169,000
370 DATA 153,192,003,136,016,250,169,255,141,192,003,173,021,208,009,001
380 DATA 141,021,208,169,015,141,248,007,173,134,002,141,039,208,164,252
390 DATA 169,032,234,136,208,252,132,198,132,251,196,252,240,026,169,000
400 DATA 133,251,152,010,010,010,038,251,024,101,249,141,000,208,042,041
410 DATA 001,005,251,005,250,141,016,208,132,251,032,228,255,164,251,201
420 DATA 028,240,108,201,147,248,090,201,157,240,078,201,148,240,072,201
430 DATA 019,240,187,201,029,240,056,201,013,240,117,201,032,144,217,201
440 DATA 092,144,017,072,173,024,208,041,002,208,004,104,024,144,201,104
450 DATA 201,193,144,196,201,128,144,003,041,127,044,041,063,145,020,173
460 DATA 134,002,145,003,200,196,252,144,003,136,208,142,024,144,139,200
470 DATA 196,252,144,134,136,208,131,240,057,192,000,240,001,136,024,144
480 DATA 235,164,252,136,169,032,145,020,136,016,251,169,000,240,162,152
490 DATA 072,200,196,252,176,012,192,000,240,018,177,020,136,145,020,200
500 DATA 208,239,136,169,032,145,020,104,168,024,144,192,104,136,240,249
510 DATA 240,032,132,253,164,252,136,177,020,201,032,208,016,200,136,136
520 DATA 177,020,200,145,020,196,253,208,245,169,032,145,020,164,253,024
530 DATA 144,215,173,021,208,041,254,141,021,208,164,252,136,177,020,201
540 DATA 032,208,003,136,016,247,196,252,200,132,251,174,085,165,134,012
550 DATA 202,134,013,169,073,133,069,169,206,133,070,032,231,176,133,073
560 DATA 132,074,032,163,182,165,251,032,117,180,160,002,185,097,000,145
570 DATA 073,136,016,248,165,251,240,041,200,177,020,201,160,176,022,201
580 DATA 129,176,015,201,064,176,008,201,032,144,013,044,105,064,044,105
590 DATA 127,044,105,191,044,105,127,044,105,064,145,098,200,196,251,208
600 DATA 216,096

```

READY.

PEEKs und POKEs zum Commodore 64

Von H. J. Liesert
Verlag Data-Becker
200 Seiten, 29,- DM
ISBN 3-89011-032-0

Sie haben richtig gelesen, Gegenstand dieser Besprechung ist ein Buch, das schon zwei Jahre auf dem Markt ist. Die nun vorliegende dritte, überarbeitete Auflage soll uns noch einmal dazu veranlassen, kurz hineinzublicken.

Wie der Titel schon sagt, beschäftigt sich das Werk mit den PEEK- und POKE-Befehlen, die dem C-64-Besitzer aufgrund des schwachen Basic in Fleisch und Blut übergehen müssen, damit er seinen Computer voll ausnützen kann. In leicht verständlichem Stil führt der Autor auch den Laien an die zu beherrschende Materie heran. Zunächst werden grundsätzliche Fragen über die Funktions-

weise von Interpreter, Betriebssystem und weitere elementare Dinge erklärt. Dann lernt der Leser anhand vieler Beispiele mit lauffähigen Listings zum Speichern von Grafiken, der Programmierung von Sprites oder der Erstellung eines eigenen Zeichensatzes den Umgang mit Speicherstellen und ihre Anwendung kennen.

Wie nützlich und gut das Buch ist, hat es in der Vergangenheit durch hohe Verkaufszahlen bewiesen. Einige kleine Fehler, die sich eingeschlichen hatten, sollen in der vorliegenden Auflage vollständig beseitigt sein. Das Buch wurde um einen Speicherbelegungsplan im Anhang erweitert, der alle wichtigen Speicherstellen kurz erklärt. Dem eigenen Experimentieren mit PEEKs und POKEs steht also nichts mehr im Wege.

Thomas Tai



enthält, findet sich das aus Editor, Compiler und Laufzeitsystem bestehende Pascal-System.

Der Editor erinnert stark an den gewöhnlichen Full-Screen-Editor des C 64, hat jedoch einige Vorteile aufzuweisen. So kann zum Beispiel die maximale Länge einer Programmzeile zwischen 0 und 80 frei gewählt werden; mit den Funktionsstufen bewegt man dann das Bildschirmfenster wie über ein Blatt Papier. Außer der Möglichkeit, Tabulatoren festzulegen, bietet er auch noch sogenannte Primary und Line Commands an, die dazu dienen, den Editor zu verlassen, den Source-Text ganz oder teilweise auf Diskette zu schreiben bzw. zu lesen und Zeilen einzufügen, zu löschen, zu kopieren oder zu verschieben. Es gibt auch zahlreiche Blockbefehle, mit denen einzelne Textblöcke verarbeitet werden können, sowie die nützlichen Befehle Find and Change, die längeres Durchsuchen des Source-Textes ersparen.

Der Compiler übersetzt den Quelltext im Speicher, was eine enorme Zeitersparnis auf Kosten des Speicherplatzes bedeutet und hinterläßt das Object-Programm im Basic-Speicher,

wo es beliebig gestartet, gespeichert und auch ohne das System geladen werden kann. Er bietet außer der Übersetzung noch die Kontrolle auf Syntaxfehler und die Möglichkeit, die Position von Runtime-Errors im Source-Text festzustellen, übersetzt mit wenigen Ausnahmen den gebräuchlichen Standard (bedeutendste Ausnahme ist eine einfache Potenzierungsfunktion) und reagiert auf aktive Kommentare (bei Assemblieren würde man sie Pseudo-OpCodes nennen), die zum Beispiel das Mitübersetzen von Quellprogrammen auf der Diskette erlauben. Dies ist bei nur 8 KByte Sourcecode, die in den Speicher passen, wirklich kein Luxus. Da das gut verständliche Lehrbuch und der durchaus brauchbare Compiler/Editor zusammen zu einem annehmbaren Preis angeboten werden, ist das System für Anfänger ungeschminkt empfehlenswert.

Axel Kleink

V24-Adapter für C 64/C 128

C 64/C 128-Computer haben eine eingebaute serielle Schnittstelle, die im TTL-Pegel an den User-Port geführt ist. Mit Hilfe des Interface Typ 88002 werden die TTL-Pegel in normgerechte V24-Signale gewandelt (ca. $\pm 9V$). Die notwendige Elektronik einschließlich Spannungswandler ist sehr kompakt im Steckergehäuse untergebracht. Eine Stromversorgung ist nicht erforderlich. Preis ca. 98,- DM.

Weitere Informationen: Wisemann Mikrocomputertechnik
Winchenbachstraße 3-5
5600 Wuppertal 2

Pascal mit dem C 64

Von Florian Matthes
Markt & Technik Verlag
220 Seiten, 52,- DM
ISBN 3-89090-222-7

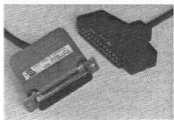
Ein günstiges Angebot von Markt & Technik ist "Pascal mit dem C 64". Hierbei handelt es sich um einen Pascal-Compiler/Editor, der nicht nur den Ansprüchen eines Anfängers genügen sollte, und um ein rund zweihundertseitiges Buch, das außer einer ausführlichen Einkleitung in diese Programmiersprache auch noch Tips und Tricks für Fortgeschrittene bietet.

Der Pascal-Kurs ist - wie bei dieser Sprache mit ihren strikten Formen nicht anders zu erwarten - anfangs recht trocken. Man wird in die grundsätzliche Systemstruktur mit Editor und Compiler eingeführt, lernt Syntaxdiagramme kennen und kämpft sich mit ihrer Hilfe von einfachen Operationen (Eingabe und Ausgabe, Auswerten von Ausdrücken) über Fallunterscheidungen bis zu den vielfältigen Schleifentypen durch. Besonders ausführlich werden da-

bei die für Pascal typischen komplexen Strukturen wie Prozeduren und Funktionen, hauptsächlich aber Rekursionen erklärt. Sachlich und nicht besonders originell, dafür aber gut verständlich, erläutert die Einführung selbst Sortieralgorithmen und Datenstrukturen wie die bekannte Baumstruktur.

Im darauffolgenden Teil (Tips & Tricks) erlernt man den Umgang mit Files zum Erstellen eines Directory-Programms und wird mit Systemadressen sowie der Einbindung von Maschinensprache in Pascal-Programme vertraut gemacht. Zusätzlich ist eine ausführliche Dokumentation zum beigelegten Compiler/Editor enthalten. Im Anhang findet man Listen von Syntaxdiagrammen, Fehlernummern, Laufzeitfehlern und Operatoren sowie einige Literaturhinweise. Eine Kurzzusammenfassung für Programmierer, die eine bestimmte Information suchen, fehlt leider.

Auf der Programmdiskette, die zusätzlich noch sechs beachtliche Demoprogramme



Einfaches, aber solides Innenleben

Amiga-Spieleübersicht

Wer die Leistungsdaten des Amiga betrachtet, denkt sofort, das sei der ideale Spielcomputer. Obwohl Commodore es gar nicht so gerne sieht, daß ihr Flaggschiff in diesen Ruf kommt, tun die Softwarehäuser doch alles, um den Amiga-Besitzer zum Spielen zu animieren. Dabei handelt es sich zum Teil um Adaptionen alter 64er Programme, aber auch um völlig neue Spiele mit neuen Ideen.

Die Programmierer der ersten Amiga-Spiele hatten sich nicht gerade mit Ruhm bekleckert, als sie C-64-Spiele umschrieben, ohne auf die besonderen Fähigkeiten dieses "Mediencomputers" einzugehen. Dies bessert sich aber zusehends. Immer mehr Spiele werden auf den Markt geworfen, und gerade zur Weihnachtszeit arbeiten die Kopiermaschinen auf Hochtouren. Deshalb ist es recht nützlich, sich einen kleinen Überblick über das aktuelle Angebot zu verschaffen. Die nun folgende Zusammenstellung von Spielen für den Amiga erhebt keinen Anspruch auf Vollständigkeit. Einige Programme wurden aufgrund ihrer schlechten Umsetzung bzw. Qualität bewußt weggelassen, andere waren zur Zeit der Entstehung dieses Artikels noch nicht auf dem Markt. Das Fehlen eines Programms in dieser Liste bedeutet also nicht unbedingt, daß es schlecht ist!

Bratacass

Das erste Spiel, das ausschließlich für die neue 68000er Generation geschrieben wurde, basiert auf einer völlig neuen Spielidee. Auf dem Bildschirm läuft ein richtiger Zeichentrickfilm ab, in den Sie eingreifen können. Es ist egal, was Sie tun und ob Sie etwas tun, "Bratacass" läuft unbeirrt weiter. Der Spieler schlüpft in die Rolle des Soldaten Kync, der nach einer Reihe von unglücklichen Ereignissen in das Kreuzfeuer zwischen Unterwelt und Regierung gerät. Alle Bemühungen, seine Unschuld zu beweisen, führen ihn auf den Planeten Bratacass.



Die Steuerung wird mit der Maus abgewickelt und ist recht gewöhnungsbedürftig. Im gesamten Spiel ist nichts genau festgelegt; es handelt sich um "Implied Action", d.h., das Programm interpretiert Ihre Mausbewegung in der aktuellen Situation z.B. als Bewegung, Handlung oder Kommunikationsversuch.

"Bratacass" besticht durch seine interessante und komplexe Handlung. Grafik und Sound bleiben allerdings weit hinter den Möglichkeiten des Amiga zurück. Es ist aber trotzdem ein empfehlenswertes Spiel für die Freunde von Action-Adventures.

System: Amiga 512 K
Preis: ca. 119.- DM

Little Computer People Project

Die Story von den kleinen Männchen im Computer ist ja nicht neu, und man hoffte deshalb auf eine spektakuläre Umsetzung. Doch sie blieb leider aus; man erhält biedere Hausmannskost. Das Männchen in Blockgrafik bewegt sich ruckartig, die Geräusche sind blechern, computerhaft, eben ganz so, wie sie beim Amiga nicht sein müßten. Der Reiz dieser hervorragenden Spielidee bleibt natürlich erhalten, doch in Anbetracht dessen, daß das Programm nur um ein Kartenspiel im Vergleich zur C-64-Version erweitert wurde, kommen doch arge Zweifel, ob man sich genug Mühe bei der Programmierung gegeben hat.

"Little Computer People" ist aber sicherlich ein Erlebnis, wenn man solch ein Programm noch nicht kennt. Der kleine Kerl auf dem Bildschirm geht einem ganz normalen Leben nach: Zähne putzen, fernsehen, Computer programmieren, essen, trinken, abwaschen, Hund füttern, tanzen, Klavier spielen. Auf Wunsch tritt er sogar mit uns in Verbindung. Wer das Spiel bereits kennt, sollte sein Geld besser anders anlegen.

System: Amiga 512 K
Preis: ca. 109.- DM

One On One

Eine der ersten ECA-Umsetzungen für den Amiga ist dieses Basketballspiel vom C 64. Leider ist die Grafik von sehr schlechter Qualität. Der Sound allerdings entschädigt für so manches. Zuschauergerummel, Ballgeräusche, Turnschuhquietschen, Schiedsrichterpfiffe und aufbrausender Jubel bei jedem Korb klingen da aus den Lautsprechern und versetzen den Spieler wirklich in eine Basketballhalle. Der hier verwendete digitalisierte Sound läßt erahnen, was in Zukunft noch auf dem Markt erscheinen wird. Das Ganze ist allerdings keine richtige Simulation des Basketballspiels, denn es spielen nur zwei Gegner auf einen Korb. Die Steuerung mit der Maus ist kaum möglich; Joysticks sollten auf jeden Fall genommen werden, wenn zwei Spieler gegeneinander antreten.

Jeder, der über die schlechte Grafik hinwegsehen kann, sollte sich "One On One" ruhigen Gewissens zulegen.

System: Amiga 256 K
Hersteller: ECA
Preis: ca. 90,- DM

Marble Madness

"Was lange währt, wird endlich gut", lautet ein altes Sprichwort, das in diesem Spiel wieder einmal seine eindrucksvolle Bestätigung fand. Monatelang war "Marble Madness" ein Zauberwort für Spiele-Freaks, das ihre Augen zum Leuchten brachte. Jetzt endlich ist es da, und wer es gesehen hat, weiß, daß alle Erwartungen weit übertroffen wurden.

Fantastische 3-D-Grafik in atemberaubend schnellem Scrolling, witzige Effekte und ein Supersound sind für das Spiel bezeichnend. Die Spielidee ist überraschend simpel: Eine Kugel muß durch 8 Level an Abgründen, Säurereampeln, Murrelffressern und Staubsaugern vorbei heil ins Ziel gebracht werden. Wer sich erst einmal vor den Computer gesetzt hat, wird von "Marble Madness" nicht so schnell wieder loskommen. Unzählige sehenswerte Effekte und eine hohe Motivation treiben den Spie-

ler-Freak zu Höchstleistungen an. Es handelt sich wohl um das beste Amiga-Spiel, das bisher das Licht der Welt erblickt hat.

System: Amiga 512 K
Preis: ca. 90,- DM

The Seven Cities of Gold

Noch eine der C-64-Umsetzungen auf den Amiga von ECA. Dieses Spiel ist allerdings nur aufgrund der relativ interessanten Handlung empfehlenswert. Sie bekommen ein Schiff und eine bestimmte Geldmenge, sollen neue Länder entdecken und natürlich Reichtümer zurückbringen. Das Zusammenstellen der Ausrüstung will gut überlegt sein, bevor es dann auf die große Reise durch die Weltmeere geht.

Weder Grafik noch Sound entsprechen den Möglichkeiten des Amiga. Selbst erfahrene Spiele-Freaks werden kaum Unterschiede zwischen der Version für den C64 und der für den Amiga feststellen können.

System: Amiga
Hersteller: ECA
Preis: ca. 109,- DM

Rogue

Ein typisches Labyrinthspiel, das auch gleichzeitig für den Atari ST herauskam. Die beiden Versionen unterscheiden sich trotz der unterschiedlichen Hardware kaum voneinander. Der Spieler steuert einen mühen Kämpfer durch 26 Levels, in denen es zunächst einmal nur um das Überleben gegen Monster und sonstige Kreaturen der Hölle geht. Wer allerdings tiefer in das Programm einsteigt, wird sich auch bald an die eigentliche Aufgabe heranwagen: Die Suche nach dem Amulett von Yendor.

"Rogue" ist sicherlich das beste Labyrinth-Spiel, das es derzeit für den Amiga gibt. Grafik und Sound sind hervorragend; die Programmierer gaben sich offenbar Mühe.

System: Amiga 512 K
Preis: ca. 89,- DM

Archon

Der Kampf des Guten gegen das Böse ist so alt wie die Menschheit. Zahlreiche Legenden hatten Einfluß auf Rollenspiele und Adventures für den

Computer. "Archon" ist nun das erste Strategiespiel, das sich mit diesem Komplex befaßt. Die Umsetzung vom C64 ist recht gut gelungen, vor allem der hervorragende Sound, der den Spieler erwartet. Durch geschicktes Einsetzen seiner Figuren werden wichtige Kräftequellen besetzt und Kämpfe zu eigenen Gunsten entschieden.

Insgesamt erscheint "Archon" wie die Variante eines abgemagerten Schachspiels, und tatsächlich hat man vor allem zu zweit viel Spaß. Es gehört zu den wenigen empfehlenswerten Spielen für den Amiga.

System: Amiga 512 K
Preis: ca. 109,- DM

Hacker II

Der Nachfolger des legendären "Hacker", das die 8-Bit-Spielwelt noch heute beschäftigt, ist wieder ein hochwertiges Produkt von Steve Cartwright.

Im Auftrag des CIA steuert der berühmte Hacker einen kleinen Roboter in einem Hochsicherheitsstrakt von Sibirien, in dem geheime Dokumente über die Vernichtung der USA lagern.

Abgesehen von der sehr politischen Handlung, mit der sich wieder ein hervorragendes Feindbild gegen die "bösen Russen" aufbauen läßt, ist "Hacker II" ein würdiger Nachfolger des ehemaligen Superhits. Man muß aber zur Lösung schon einige Stunden geistige Arbeit investieren. Ein Vergleich mit dem Schwierigkeitsgrad von "Hacker I" ist auf keinen Fall möglich. Für Profis mit gehobenen Ansprüchen ist "Hacker II" das ideale Spiel, mit dem der Amiga seinen Ruf

als Spielcomputer erstmals erfolgreich verteidigen kann.

System: Amiga 512 K

Nun noch zu zwei Spielen, die kürzlich angekündigt wurden.

Leader Board

Das alte bekannte Golfspiel vom C64 wurde hervorragend auf den Amiga übertragen. Sandbunker, abwechslungsreiche Grafik, kurz alles, was der ersten Version fehlte, hat nun dieses Programm.

Silent Service

Als U-Boot-Kommandant im Zweiten Weltkrieg geht es mitten hinein in die geheimnisvolle Welt der Unterwasser-Abenteuer. Solche kriegerischen "Spiclerien" sind allerdings nicht jedermanns Sache, und daher sollte man dieses Programm einmal kurz ansprechen, bevor man es erwirbt.

Der Trend auf dem Softwaremarkt für den Amiga geht eindeutig zum Spiel. Jedes annähernd erfolgversprechende Programm wird für ihn umgesetzt. Zu erwarten sind noch der "Game Maker" von Activision, "Deep Space", ein Elite-Clone von Psygnosis, und die Sportspiele "Arena" und "Winter Games".

Eines sollten Sie also bei allem ernsthaften Arbeiten mit dem Amiga nicht vergessen: Ihr Computer kann auch spielen, und wenn das Niveau weiterhin so rapide ansteigt, stehen wir bald vor einer Flut von hervorragenden Programmen, die man nicht nur spielt, sondern wirklich anwendet (z.B. zur Entspannung nach anstrengender Arbeit).

Thomas Tai

Lattice auf Amiga-Pfaden

Das renommierte amerikanische Softwarehaus Lattice, das Software-Entwicklern mit unzähligen Utilities und Compilern unter die Arme gegriffen hat, bringt nun auch eine Produktpalette für den Amiga heraus. An dieser Stelle soll zunächst nur ein kleiner Über-

blick über die Programme gegeben werden. Ein ausführlicher Test und eine Beurteilung folgen später.

Für den C-Programmierer dürfte natürlich der C-Compiler interessant sein, der allerdings schon etwas länger auf

dem Markt ist und nur der Vollständigkeit halber erwähnt sei. Schon hellhöriger wird man bei der "Mac Library", die 60 Funktionen zur Einbindung in C-Programme enthält. Mit ihnen kann man die Grafikfähigkeiten des Amiga besser nutzen und erspart sich das Zusammenbasteln eigener Funktionen aus mühsam herausgesuchten Systemadressen. So ganz nebenbei sind die Funktionen auch noch Macintosh-kompatibel. Wer also C-Programme vom Mac besitzt, kann diese ohne Probleme auf den Amiga übertragen.

Die "dBase III Library" soll eine Verbindung zum Standardprogramm der MS-DOS-Welt "dBase III" schaffen. Die Funktionen werden wieder von C genutzt und ermöglichen den Aufbau von Dateien, die "dBase III" lesen und auswerten kann, sowie das Erweitern von bereits existierenden "dBase III"-Dateien. Der Umstieg von MS-DOS auf den Amiga wird somit noch leichter, da die Datenbestände weiterhin genutzt werden können.

Kuma-Seka, ein Macro-Assembler für den Amiga

Assembler-Programme für den Amiga sind vorerst noch rar. Umso erstaunlicher ist es, daß sich im kleinen Assembler-Angebot ein Programm für den Commodore-Knüller befindet, das kaum Wünsche offenläßt. "Seka" heißt die angenehme Überraschung aus England, die von der Computer-Firma Kuma auf den Markt gebracht wurde und mit Sicherheit viele Freunde finden wird.

"Seka" ist recht einfach in der Handhabung. Der Assembler wird aus dem CLI heraus mit Namen aufgerufen; es genügt, "SEKA" einzugeben. Nach kurzer Ladezeit verlangt das Programm nach der Workbench-Angabe. Danach wird der User gefragt, wieviel Arbeitsspeicher dem Assembler zugeteilt werden soll.

Nun zu den Besonderheiten des Assemblers. "Seka" bietet

Die folgenden Utilities sind wieder für alle Programmierer interessant, ganz gleich, welche Sprache sie bevorzugen. Mit dem "Lattice Screen Editor" können Programme komfortabel und einfach wie mit einer Textverarbeitung editiert werden. Herausragendes Leistungsmerkmal ist hier die Multitask-Funktion, mit der mehrere Dateien gleichzeitig bearbeitet werden können. Zwischen ihnen ist auch ein Austausch von Zeichenfolgen möglich. Ebenfalls für die effektivere Arbeit mit Texten sind die "Lattice Text Utilities" gedacht. Mit dieser Sammlung von acht Programmen können Worte einer Datei gezählt, Zeichenfolgen gesucht und ersetzt, Dateien verglichen und Unterschiede aufgelistet sowie Batch-Dateien erzeugt werden.

Soviel für diesmal von den Lattice-Produkten. Dem interessierten User sei noch gesagt, daß die Firma Pfotenhauer Microcomputer-Anwendungen den Generalvertrieb der Produkte übernommen hat und deutsche Versionen in Arbeit sind.

Thomas Tai

einen Zeilen- sowie einen Screen-Editor, zwischen denen mit der ESC-Taste hin- und hergeschaltet werden kann. Der Zeilen-Editor verfügt über alle üblichen Funktionen, wie INSERT, EDIT, LOCATE usw.; beim Screen-Editor sind CUT, COPY und PASTE selbstverständlich. "Seka" ist in der Lage, absoluten oder voll-relokativen Code zu erzeugen, und verfügt über 30 Pseudo-Operatoren.

Der Befehl ORG veranlaßt den Assembler, absoluten Code zu erzeugen, der durch LOAD an die Adresse geladen wird. Sollte jedoch CODE eingegeben werden, so wird relokativer (voll im Speicher verschiebbarer) Code erzeugt. Durch EVEN wird die nachfolgende Anweisung auf eine gerade Adresse gesetzt, da der MC 68000 nur auf solche zu-

rückgreifen kann. Mit der Macro-Anweisung definiert man neue Befehle. "Seka" verfügt auch über Befehle, die den Drucker steuern. Bedingungen für die Assemblierung können durch IF, ELSE und ENDIF geschaffen werden. BLK hingegen definiert einen Block im Speicher, der mit einem Byte, Wort oder Langwort gefüllt wird.

Das wichtigste Kommando des Assemblers ist zweifellos A, das für assemble steht. Nachdem das Kommando eingegeben ist, verlangt "Seka" nach einigen Optionen. Werden diese mit RETURN beantwortet, so gibt er nur die fehlerhaften Zeilen aus. Die Option V steht für Video und bringt das Listing auf den Bildschirm; E und P bewirken seine Ausgabe auf einem Drucker. Sehr interessant ist auch die Option O. Alle Macros ohne .s oder .l werden im Source optimiert (z.B. wird an jeden Sprungbefehl, soweit dies möglich ist, ein .S angefügt).

Bemerkenswert ist auch der Befehl WO (Writocobjct). Mit ihm wird das assemblierte File, das relativ sein muß, als ein vom CLI aufrufbares Programm abgelegt. Dieses kann auch von der Workbench aus aufgerufen werden, sofern es mit einem Icon ausgestattet wird, oder als Basic-Routine, wenn es als .map-Datei im libs-Directory abgelegt wurde.

Ein auf der "Seka"-Disk mitgeliefertes Programm, "amiga-dos.l", wird zu einem Source-File geladen. In diesem Programm steht eine Reihe von DOS-Befehlen, wie z.B. OPEN, CLOSE, READ und

Anzeigenschluß

für
Nr. 2/3-87
Computer
Kontakt
ist der
2. Januar 1987.

WRITE, die mit JSR aufgerufen werden. So ist es möglich, mit einer MC-Routine eine andere zu laden, zu löschen oder zu erzeugen – und dies alles mit ein paar DOS-Aufrufen.

"Seka" verfügt auch über einen eingebauten Disassembler, für den Singel-Stepping und Modifizieren aller Register kein Problem darstellen. Alle Standardbefehle, z.B. für das Kopieren von Speicherblöcken, Springen zu einer Subroutine oder Starten ganzer Programme, sind vorhanden.

Das Handbuch des Assemblers umfaßt 35 Seiten. Die Erläuterungen sind in gut verständlichem Englisch gehalten, so daß man sich ohne weiteres zurechtfinden kann.

Boris Zentner

Einfach in der Handhabung durch den Full-Screen-Editor, auch ein Disassembler; ist dabei



A A R I

Koch
Peaks & Pokes zu Atari
500 XL/800 XL

257 Seiten
Dieses Buch enthält leicht verständlich den Umgang mit Peaks & Pokes. Es enthält viele interessante und wichtige Tricks, die sich bei verschiedenen Anwendungsumgebungen sowie sehr viele Erweiterungsprogramme. Zusätzlich wird der Aufbau des Atari 500XL/800XL erklärt.

Bestellnummer DB 1 DM 39,-

E. H. Carlson
Lerne Basic auf dem Atari

220 Seiten
Dieses Basic-Lernbuch ist besonders für Anfänger gedacht. Sie hier lernen, wie man Action, Wait- und Walk-Sprache programmiert. Hinweise, Erklärungen, Übungen und Wiederholungen werden in einer leicht verständlichen Art gegeben. Durch die vielen Zeichnungen und die leicht verständliche Sprache ist es besonders für Kinder bevorzugt geeignet.

Bestellnummer MT 7 DM 38,-

Julien Reschke
Atari Basic Handbuch

208 Seiten
Das vollständige Basic-Handbuch (ATARI) ist in Ihren Atari 500 und ganz so benutzbar. Das vollständige Basic-Handbuch wird beschrieben und anhand praktischer Beispiele erklärt.

Bestellnummer BY 13 DM 32,-

D. Scharfhan
Start mit Atari-Logo

220 Seiten
Hier handelt es sich um eine lernorientierte Einführung in die Computersprache Logo. Grafik, Text und Musik werden in zwölf Lektionen beigebracht. Auch welche Möglichkeiten Ihnen nutzt. Die Atari-Logo-Verfahren. Nach dem Logo geklärt wird, wie man das Logo neu einrichtet.

Bestellnummer V 2 DM 32,-

L. M. Schröber
Das Atari-Programmierhandbuch

390 Seiten
Hier werden moderne Konzepte von Programmieren im Hinblick auf die Atari-Systeme dargestellt. Sie lernen den Umgang mit Programmen in verschiedenen Umgebungen und deren Leituraufbau. Hier wird erklärt, wie Sie den 6552-Processor direkt programmieren. Wenn Sie dieses Buch durchgelesen haben, können Sie Ihren Atari in- und auswendig programmieren.

Bestellnummer MT 6 DM 52,-

Rabe/Schmidt
Spülen, lernen und arbeiten mit dem Atari

200 Seiten
Dabei werden Ihnen theoretische und praktische Kenntnisse vermittelt. Von Anfang an werden die Ihren Rechner (Atari 500) Schritt für Schritt besser kennen und benutzbar zu werden. Sie können Spiele zum Spaß.

Bestellnummer BY 14 DM 32,-

A. Hettinger/A. Heinz
Start mit Atari-Logo

194 Seiten
Nicht dem Durchschnittler dieses Buches werden Sie selbst in der Lage sein, Programme zu schreiben. Anleitungen für Grafik- und Soundmöglichkeiten über Plot und Text bis hin zu komplexen Spracherweiterungen reicht das breite Spektrum. Sie werden eigenhändig Basic-Komplexe wie komplexe dokumentierte, jede der Atari-Datei-Systeme die Einführung des Systems.

Bestellnummer V 3 DM 30,-

M. Vob
Das Schulbuch zu Atari
800 XL/800 XL

389 Seiten
Besonders für Schüler der Mittel- und Oberstufe geschrieben, enthält das Schulbuch viele interessante Problemlösungen und Lernprogramme, die besonders aufwändig und leicht beschreibbar sind. Sie ermöglichen ein intensives Lernen am Atari 800XL/800XL mit Ver Spaß.

Bestellnummer DB 9 DM 40,-

Tom Rowley
Sprühende Ideen mit Atari Grafik

240 Seiten
Dies ist ein Lehrbuch, das mit den Grafikerfahrungen des Atari in die Gestaltung von Objekten in Partikularien und die Entwicklung von Bildschirmtexten erklärt.

Bestellnummer TW 18 DM 48,-

A. J. Reschke
Was der Atari alles kann
Band 1

270 Seiten
Hier muß der Anwender schon die Grundlagen des Atari-Basic kennen und ein wenig Übung in Programmieren besitzen. Eine Vielzahl von gut durchdachten kleinen Programmen aus dem Bereich Hobby, Wissenschaft, Sport und Spiel werden vorgestellt.

Bestellnummer V 4 DM 35,-

Norbert Szczepanowski
Atari 130 XL, 600 XL, 800 XL

222 Seiten
Dieses Buch ist eine unverzichtbare Einführung in den Umgang mit dem Atari 130XL, 600XL und 800XL. Die gesamte Vorkenntnisse voraussetzt.

Bestellnummer DB 10 DM 35,-

Echter/Grohmann
Atari 600 XL/800 XL Intern

385 Seiten
Atari-Intern ist ein unverzichtbares Arbeitsmittel für jeden, der sich ernsthaft mit Technik und Betriebssystem der Atari-Computer 500/800XL auseinandersetzen will.

Bestellnummer DB 18 DM 48,-

A. J. Reschke
Was der Atari alles kann
Band 2

240 Seiten
Dieses Buch ist Band 1 enthält auch diese Buch eine ausgewählte Mischung aus praktischen Anwendungsprogrammen und Spiel. Mit 218. Computerwissen, Datenmanagement oder auch Trainingsprogramme in Verbindung mit dem entsprechenden Erläuterungen.

Bestellnummer V 5 DM 35,-

Stanley R. Trost
Atari-Programm-Sammlung

130 Seiten
Hier werden die Anwender ein Satz ausgearbeiteter Programme für die Atari-Computer geliefert. Eine Vielzahl von praktischen Beispielen mit Text, Ihre Computer nicht zu mühen.

Bestellnummer BY 11 DM 34,-

Voss
Das Basic-Trainingbuch zu Atari 600 XL/800 XL

380 Seiten
Das Basic-Trainingbuch zu Atari 600XL/800XL ist eine ausführliche, didaktisch gegliederte Einführung in die Atari-Basic. Sie sind den Beispielen über die Programmierung bis zum fertigen Algorithmus am Ende jedes der Programmierung.

Bestellnummer DB 11 DM 39,-

A. Hettinger/W. Krauß
Die Atari-Hitparade

190 Seiten
Die Atari-Hitparade ist eine Einführung in die verschiedenen Anwendungsumgebungen und enthält die Atari-Musik-Drucke. Die Atari-effektive und Musikstücke, aber auch komplette Spiele. Wir werden Sie (Atari-Hitparade).

Bestellnummer V 6 DM 32,-

Reschke/Wethoff
Das Atari Profibuch

300 Seiten
In dieser Werk finden Sie gebündelt alle wichtigen Informationen, um Ihre Atari-Systeme zu benutzen und Ihre Fähigkeiten zu erweitern. Ein Informationspaket, das keine Fragen offen läßt.

Bestellnummer BY 12 DM 42,-

Don Inman/Wulf Inman
Der Atari Assembler

270 Seiten
Mit diesem Buch können Sie die Programmierung in Assembler lernen, und Sie gleichzeitig mit der Anwendung des Atari-Assembler-Moduls auf Atari-Atari 400 oder 800XL mit Ver Spaß machen.

Bestellnummer ID 18 DM 39,-

BUCHVERSAND

Schneider
Strategiespiele, und wie man sie auf dem Atari 600 XL/900 XL programmiert
191 Seiten
Hier wird Ihnen eine Einführung in die hochkomplexe Welt der Strategiespiele geboten. Von einfachen Programmen mit System- und Basisdaten über komplexe Spielmechaniken, Suchverfahren bis hin zu komplizierten Programmen, die sich nicht von Hand entziffern lassen. Besondere Beispiele.
Bestellnummer DE 19 DM 25,-

Podde/McNir/Cook
Mein Atari-Computer
400 Seiten
Ein Handbuch, das für jeden Atari-Besitzer wichtige Informationen enthält und zur Lösung aller Atari-Probleme beiträgt. Es ist reich illustriert und enthält eine Vielzahl der für den einleitenden Interessierten so wichtigen Tabellen.
Bestellnummer TW 20 DM 59,-

Steiner/Stähler
GEM für den Atari 520 ST
344 Seiten
Dieses Werk ist eine Übersetzung in alle Sprachen, die GEM für den Benutzer anderswo spricht. Die zehnjährige Entwicklung findet eine neue Tje für die Bedienung und effektiv mit dem Atari ST arbeiten zu können.
Bestellnummer MT 21 DM 65,-

Jürgen Meier
WordStar für den Atari ST
436 Seiten
Dieses Buch ist so aufgebaut, daß der Leser mit WordStar schnell weiter vorankommt. Alle Funktionen erklärt. Auch auf die Bedeutung von Befehlen wird ausführlich eingegangen.
Bestellnummer MT 22 DM 49,-

Vinod Appin
Lernen mit Spaß - 40 Lernspiele für den Atari
189 Seiten
Interaktive Programme erwecken das Wissen in Logik, Geometrie, Mathematik, Englisch, Informatik, Deutsch sowie allgemeinen Themen und steigern die Lesefähigkeit. Neben dem Lernstoff sind die Leser, wenn Computer geschickt zu betreiben sowie die Programme in eigene Texte.
Bestellnummer MW 23 DM 29,90

Alfred Görjans
Utilities in Basic für Atari-Computer
120 Seiten
In diesem Buch finden Sie praktische Lösungen zu den Themen Programmierhilfen, Sound und Textverarbeitung, so z.B. die praktische Differenzierung. Ein vollständige Lösung von Basis-Zellen, automatische Programmierhilfen, Musterkollern oder auch die Weitergabe von Atari-Dokumenten und Modulen auf dem Diskette.
Bestellnummer V 24 DM 25,-

Das GROSSE
C. Lorenz
Das große Spielebuch für Atari, Band 1
103 Seiten
Aufwändige Computerspiele in Atari-Basic: Neben Spielen finden Sie hier eine Reihe hochinteressanter Anregungen für eigene Programme. S-P-Codes, Berechnung und Sortieren, Text und Text in Form, Fortprogrammierung usw.
Bestellnummer HO 25 DM 29,90

C. Lorenz
Das große Spielebuch für Atari, Band 2
208 Seiten
Dieses Buch enthält Programme für den Atari 600 XL/900 XL und ist eine Fortsetzung von Band 1. Es bringt eine Reihe neuer Spiele, Programme für Simulation und ein Kapitel über Grafik-Systeme.
Bestellnummer HO 26 DM 29,90

Visionen
Adventures, und wie man sie auf dem Atari 600 XL/900 XL programmiert
284 Seiten
Hier ist gezeigt, wie Adventures funktionieren. Wie man sie entwickelt und wie man eigene Adventures auf Atari-Computern für Atari XL programmiert. Hier kommt der künftige Adventure-Generator, der 999 Adventures programmieren zum Kinderspiel macht.
Bestellnummer GB 27 DM 39,-

Schwager
Atari Star-Taster
110 Seiten + Disk
Manchmal findet es sich ein umfängliches, kundenspezifisches Textmaterial für Ihren Atari (z.B. 48-KHHz). Das Buch gibt eine Einführung, die Diskette liefert ein weiteres Programm.
Bestellnummer SY 28 DM 64,-

Rugg/Feldman/Barry
30 Basic-Programme für den Atari
274 Seiten
Das Buch enthält sorgfältig getestete Spiele- und Grafikprogramme von Mathematik, Literatur und vielen anderen Anwendungsgebieten der täglichen Nutzung für Ihren Atari-Computer.
Bestellnummer O 29 DM 34,-

James/Geo/Earbank
Das Atari-Spielebuch für 600 XL/900 XL
194 Seiten
71 Spiele umfassen Spannung, Action und bewagte Grafik. Jedes Programm ist vollständig aufgeführt und vollständig erklärt. So wird auch der Neugierige mit der Funktion und dem Aufbau der Programmiersprache vertraut gemacht. In Form der Routine werden, erweiterbar und kann so gerne auf eigene Programme erweitert werden.
Bestellnummer V 30 DM 36,-

Buch-Bestellschein

Bitte liefern Sie mir folgende Bücher:

Anzahl	Bestell-Nr.	Einzel-Preis

Ich wünsche folgende Bezahlung:

Nachnahme (+ 5,70 DM Porto + Versandkosten)

Vorauskassa (keine Versandkosten)

Bei Vorauskassa bitte Betrag belegen auf Postcheckkonto-Nr. 43423 768 Stuttgart.

Name des Bestellers: _____

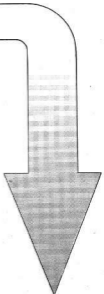
Anschrift: _____

PLZ/GA: _____

Titel: _____

Stempel/Unterschrift: _____

Coupon ausschneiden, auf Postkarte kleben und einsenden: Verlag Pitz-Eberle, Postfach 1640, 7518 Bretten.



CK-PROGRAMME

SINCLAIR

Paint (nicht veröffentlicht), Pyramide (7/84), Superlin (9/84), Drawer (9/84), Säulendiagramm (10/84), Große Buchstaben (10/84), Farben beim Spectrum (10/84), Prodomo (11/84), Isotext (12/84), Libelle (12/84), 3-D Schrift (12/84), Neues Zeichensatz (12/84), Krümmelmaier (1/85), Fast 1/5 (1/85)

Best-Nr. 8 1

Puzzle (4/85), Sprites mit Demo (3/85), Darts (5/85), Uhr (7/85), Roulette

Best-Nr. 5 10

Katalog (2/85), Software (2/85), Häufigkeit (2/85), Computer Figures (2/85), Ku Bernd (2/85), Sterngrafik (2/85), Maric Train (3/85), Senso (4/85)

Best-Nr. 5 11

Schaffner (5/85), Spectrum-Infosystem (5/85), Cowboy (6/85), Player's Dream (6/85), Tape (6/85), Soundexaminer (6/85), Hoverscraft (7/85)

Best-Nr. 5 20

ATARI

Lunar Lander (12/84), Car Race (7/84), Turbo Worm (1/85), Munstedagd (3/85), Bewegte Grafik (3/85), Digger (3/85), 15 und 3 (4/85), Bundesligastadion (3/85), 3-D Lobby (10/84), Zeichensatzeditor (2/85), Mira-Isotextstudio (9/85), Roby Doby (11/84), Musik-Editor (4/85), HELM

Nur bei Kassetteneinweisung. *Programme sind nur mit Erweiterung lauffähig.

Best-Nr. A 10 Diskette, Best-Nr. A 10a Kassette

Sound-Demo I (5/85), Sound-Demo II (nicht veröffentlicht), The Run and Jump Construction Set (6/85), Bank Panic (7/85) nur mit Endi Funktions-Plotter (5/85), Blockade (9/85), Jewel Eater (5/85), Zeichen-Assembler (7/85), Joystick-Controller (9/85), Horzontales-Scrolling (5/85)

*Converter (DOS II in DOS II) (9/85) *Nur auf Diskette

Best-Nr. A 11 Diskette, Best-Nr. A 11a Kassette

TI 99/4A

Bugler Time (12/84), Cowboy (7/84), Desert Right (9/84), Fossadenkettler (11/84), Hangman (nicht veröffentlicht), Mine-Pot (1/85), Nova-Madaga (1/85), Paschulte Jumper (5/84), Permanente Kleinbuchstaben (10/84), Pokalstengenerator (12/84), Screen Writter (11/84)

Bugler Time (12/84), Cowboy (7/84), Desert Right (9/84), Fossadenkettler (11/84), Hangman (nicht veröffentlicht), Mine-Pot (1/85), Nova-Madaga (1/85), Paschulte Jumper (5/84), Permanente Kleinbuchstaben (10/84), Pokalstengenerator (12/84), Screen Writter (11/84)

Best-Nr. B 1 Diskette, B 1a Kassette

Alpha Lock (2/85), Cube (3/85), Exotisch (4/85), Jünger (4/85), Macropedia (4/85), Mergle-Filer (3/85), Motor ON (2/85), Pooyan (2/85), Proglad (3/85), Reaktor (3/85), Vokabel (2/85), Achtung! Macropedia, Mergle-Filer und Proglad nur auf Diskette.

Best-Nr. B 10 Diskette, Best-Nr. B 10a Kassette

Balade pour Aokine (nicht veröffentlicht), Rugiminator (6/85), Super Disk-Katalog (5 und 7/85), Fuch des Pharo (6/85), Plot (6/85), Säulen-

diagramm (5/85), Calculator (7/85), Tester (6/85), Wurm-Quest (7/85). *Nur auf Diskette

Best-Nr. B 11 Diskette, Best-Nr. B 11a Kassette

*Golo/Jump (10/85), Caven (9/85), Crazy Man (11/85), Devil Quest (10/85), Elsieh-Star (11/85), High Res. Grafik mit Demo (11/85), Sound-
editor (10/85), Sprites-Utilities (11/85), Sektor D (9/85). *Nur auf Diskette

Best-Nr. B 12 Diskette, Best-Nr. B 12a Kassette

COMMODORE

Duell (7/84), Mäuserennen (9/84), Speicherplatzanzeige (10/84), Basic-Erweiterung (10/84), Through the wall (11/84), Maze-Ball (11/84),
Pulsfrequenzgenerator und Indikator (11/84), Grafik-Erweiterung (12/84), Berktele (12/84), Phoenix (1/85), Nüßbüchling (1/85)

Best-Nr. C 1 Diskette DM 20,-, Best-Nr. C 2 Kassette DM 16,-

Handballmanager (3/85), Delaher (2/85), Ghosts (4/85), Pugno (4/85), Dialekt (4/85), Mergle (2/85), Find (2/85), Screen-Designer (2/85),
Sci Des. Obj. COOD (2/85), Data-Gen (2/85), Rohrenfarben (3/85), Auto-Statist. I & II (4/85), Code 04 (5/85), File-Load (5/85), File-Gove (5/85)

Best-Nr. C 10 Diskette DM 20,-, Best-Nr. C 10a Kassette DM 16,-

Die Grafik-Erweiterung Teil I und II sowie die Basic-Erweiterung EX3 V 1.8 kann auf Diskette / Kassette inklusive Anleitung bezogen werden. (Siehe
auch CK-Hefte 10/84, 12/84 und 5/85)

Best-Nr. D685 Diskette DM 13,-, Best-Nr. K585 Kassette DM 10,-

FREIE AUSWAHL

3er Pack

5er Pack

3 Kassetten nach Ihrer Wahl nur **40,-** 5 Kassetten nach Ihrer Wahl nur **60,-**

3 Disketten nach Ihrer Wahl nur **50,-** 5 Disketten nach Ihrer Wahl nur **75,-**

M M S E R V I C E

Kassette
16,--

Paint (9/85), Window (9/85), 3-D Plotter (10/85), Superfit (10/85), Centripede (10/85), Ntho Joe (11/85)
Best.-Nr. S 30

Hörschutz (1/86), The Servant (1/86), Snake-Smasher (3/86), Aast (3/86), List/LIST (3/86), Programmzeilen löschen (3/86), Spectrum-Quickcopy (3/86), On Error Goto (3/86)
Best.-Nr. S 40

Ass. Tips Uhr (7/86), GUP-Druckeroutine (7/86), Datenverschüsselung (7/86), Pro Dos (7/86), Apletmännchen (7/86), Teile (7/86), Diatmaster (3/86), Spectrace (5/86)
Best.-Nr. S 50

neu
Assemblerlips Teil 9 (9/86), Autostartkiller (9/86), Variablenave (9/86), Headerchange (9/86), Assemblerlips Teil 10 (11/86), Jahrmarkt (11/86), Schreibschrift (11/86), Discovery Diskettenverz (11/86), Apletmännchengrafik (11/86), Maxi Floßkomma (11/86), HardcopyPrinter II (11/86), Grafikeditor (1/87), Quercopy (1/87), Ziffernschrift (1/87), Window (1/87)
Best.-Nr. S 60

Kassette
16,--

Diskette
20,--

Kassette
16,--

DL Designer 64K (10/85), Joypoint 64K (10/85), Musiccreator 64K (11/85), Chetfordaktor 64K (1/86), Unprotector V 1.0 16K (1/85), Key Maker 16K (1/85)
Best.-Nr. A 12 Diskette, Best.-Nr. A 12a Kassette

Chery Hazy (3/86), Mission X auf dem Atos (5/86), "Basic-Erweiterung (5/86), Mini-Billard (10/85), Zeichen-Zauberer (3/86), Sound-Demo (3/86), "Nur auf Diskette"
Best.-Nr. A 13 Diskette, Best.-Nr. A 13a Kassette

Revolver Kid (1/86), "Psy-Dos (7/86)", text im Grafikfenster (7/86), Rollerball (7/86), Kung Fu (9/86), "Disk-Menü (9/86), Tiron (9/86), "Nur auf Diskette"
Best.-Nr. A 14 Diskette, Best.-Nr. A 14a Kassette

neu
Der hungige Goff (11/86), Atos-Puzzler (11/86), Kartelverwahrung (11/86), Disc-Collector (11/86)", Mid-Disk-Programm (11/86)", MicroMon (nur für Kassettentherapie) (11/86), Wombel (1/87), Calc 800 (1/87), Diskeditor (1/87)", Speed Tape (1/87), Recopy (1/87)", Zeichensatzfinder (1/87)", Hardcopy-SP 50 AI (1/87)", "nur auf Diskette"
Best.-Nr. A 15 Diskette, Best.-Nr. A 15a Kassette

Diskette
20,--

Kassette
16,--

Bierböse (3/86), Büchchen (5/86), "Copy 5 (1/86), Hardcopy für Seikasha CP 100 (1/86), Lander (3/86), Moonriver (5/86), Polargrafik (3/86), T-Paint (3/86), 40-Zeichen-Mode (3/86), Turle Jumper (1/86)", "Nur auf Diskette"
Best.-Nr. B 13 Diskette, Best.-Nr. B 13a Kassette

Ti-Assembliediskette 1 Für die Ti-Assembliedisks haben wir auf vielfachen Wunsch eine Diskette mit den Quellcodes der überlieferten Assemblerprogramme Mazropede (4/85) und Super Disk Catalog (6 und 7/85) zusammengestellt. Zum Ändern und Lernen.
Best.-Nr. B 14 Diskette

Ti-Assembliediskette 2 Diese Diskette enthält die Quellcode-Dokumente für die Assemblerprogramme Adressdatei (4/86) und Disk-Uptime (8/86)
Best.-Nr. B 15 Diskette

"Wily der Wurm (7/86), Seikasha GP 50 Hardcopy (7/86), Tapan (7/86)", "Adressdatei (5/86), Soundeditor plus (9/86), Screen Editor (7/86), "Assembler-Sicht-Generator (7/86), The Plugue (9/86)", "Disk Utilities (9/86)", "Nur auf Diskette"
Best.-Nr. B 16 Diskette, Best.-Nr. B 16a Kassette

neu
Monopoly (9/86)", Maschinenkatalog (11/86), Buchstaben-Mix (11/86), Dat-Maker (11/86), Textverschüsselung (11/86), Combas (1/87)", Load/Save Long (1/87), U-Boot Kampf (1/87), SROCC (1/87), Hexmonitor in C (1/87)", "Nur auf Diskette"
Best.-Nr. B 17 Diskette, Best.-Nr. B 17a Kassette

Sonder-
modore
itten und
etten

Baloon (7/85), Rescue in the Stone-Age (5/85), Softlake (7/85), Spete Editor (9/85), Data Loader (6/85), Rohmannroutine (7/85)
Best.-Nr. C 11 Diskette DM 20,-, Best.-Nr. C 11a Kassette DM 16,-

Roddish One (10/85), 3-D-Hubschrauber Simulator (11/85), Head to Head Bathlon (1/86)
Best.-Nr. C 12 Diskette DM 13,-, Best.-Nr. C 12a Kassette DM 10,-

Formel One (3/86), Director in Farbe (3/86), Roddsh in the Jungle (5/86)
Best.-Nr. C 13 Diskette DM 13,-, Best.-Nr. C 13a Kassette DM 10,-

Silligen (7/86), Jumping Jive (9/86), Fasten Right (11/86), Hardcopy C64 (11/86), Time Loader (nur für Diskette) (11/86)
Best.-Nr. C 14 Diskette DM 13,-, Best.-Nr. C 14a Kassette DM 10,-

SOFT-
WARE-
BESTELL-
SCHEIN

Coupon ausschneiden und entsenden an:
Computer Kontakt, Softwareversand
Postfach 1440, 7518 Bredten

Anzahl	Bestell-Nr.	Bestell-Pack	Preis	Name des Bestellers
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____

Strasse _____ PLZ/Ort _____

Telefon _____ Datum/Unterschrift _____

Ich wünsche folgende Bezahlung:

Nachnahme (zuzüglich DM 4,70 Porto- und Versandkosten)

Vorauskassa (keine Versandkosten)

Bitte Scheck belegen oder auf Postcheckkonto Kartenzug 43420-736 überweisen.

ATARI®

INFOS - NEWS - BERICHTE - REVIEWS - TESTS - PROGRAMME - LISTINGS - ALLES FÜR ATARI-USER

Fröhliche Weihnachten

heißt es jetzt bald wieder. Auch dieses Jahr wird es sie geben: die Computer unter den Weihnachtsbäumen. Computerver- und -einkäufer stehen bereits jetzt voll im Streß. Wieder werden viele Computer mit falschen Vorstellungen gekauft, Laien aufgeschwatzt und an ahnungslose Grundschüler verschenkt.

Allen, die in dieser Saison ihren Computer bekommen haben und jetzt die CK in Händen halten, einen doppelten Glückwunsch: Zum Kauf dieses Heftes und, was sich aus ihrem Interesse für diese Seiten schließen läßt, vor allem zu ihrem

Atari, für den wir wieder all-er-hand zu bieten haben: Viele Hilfsprogramme für die 8-Bit-Ataris, Maschinensprachliches von Peter Finzel, eine Basic-Tabellenkalkulation und wie immer eine bunte Mischung aus Spielen, Infos und Tricks.

Für die großen Atris haben wir wieder einiges zusammengetragen. Zusätzlich gibt's für die großen und die kleinen Atris demnächst noch viel mehr im neuen ATARI magazin. Ein ausführlicher Hinweis findet sich in diesem Heft.

Bis nächstes Jahr verbleibe ich als Euer
Thomas Tausend

Atari-Leserfragen

Ich besitze einen Atari 800XL mit Cassettenrecorder. Nun habe ich vor kurzem eine Diskettenstation 1050 geschenkt bekommen. Da ich sehr viele Spielcassetten besitze und das ewige Warten leid bin, möchte ich die Spiele nun auf Diskette kopieren. Alle bisherigen Versuche sind jedoch fehlergeschlagen - wie muß ich vorgehen?

Basic-Programme können problemlos von Cassette geladen (CLOAD) und auf Diskette gespeichert werden (SAVE "D:prgramme.ext"), wenn man vorher ein DOS gebootet hat. Bei Maschinenprogrammen sieht die Sache jedoch anders aus: Meistens bestehen die Programme aus mehreren Teilen und sind mit einem Kopier-

schutz versehen. Selbst wenn diese Daten auf Diskette kopiert werden, kann man das Programm dennoch nicht starten, ohne es auf Diskettenbetrieb abzustimmen.

1. Ich beschäftige mich zur Zeit mit der Programmierung mathematischer Funktionen in Basic. Bosweilen kommen jedoch recht sonderbare Ergebnisse dabei heraus. Gerade bei sehr großen und sehr kleinen Zahlen sind die Ergebnisse oft falsch. Was kann man da machen?

2. In diesem Zusammenhang wurde mir erklärt, daß dieses Problem mit Maschinencode gelöst werden kann. Wie kann ich in die Maschinenprogrammierung einsteigen?

3. Bei laufenden Programmen schaltet der Atari dennoch in den Farbwechselmodus um. Wie kann ich dies verhindern?

1. Obwohl der Atari sehr viel genauer rechnet als z.B. ein Apple-, Commodore- oder Sinclair-Computer (einfacher Test: FOR N=1 TO 1000 STEP 0.1:PRINT N:NEXT N), treten auch bei den Atris Ungenauigkeiten auf. Diese resultieren aus Rundungsfehlern, da jeder Computer mit einer festen Anzahl an Stellen rechnet. Da das Auftreten solcher Fehler sehr schlecht vorhersehbar ist, kann man sie kaum verhindern.

2. Peter Finzels Maschinensprachkurs und sein kürzlich erschienenes Buch sollten Ihnen den Einstieg in die Maschinensprache ermöglichen. Wenn Sie jedoch glauben, in MS einfacher und genauer rechnen zu können, so werden Sie eine herbe Enttäuschung erleben, da Sie jede Rechenoperation aus elementaren Additions-, Subtraktions- und Schiebebefehlen zusammensetzen müssen. (Die Betriebssystemroutinen arbeiten auch von der MS-Ebene aus nicht genauer!)

3. Mit POKE 77,0 kann dies verhindert werden. Der POKE muß allerdings mindestens einmal in ca. 7 Min. durchgeführt werden. Am besten ist es, wenn Sie ihn in Joystick-Abfragen o. ä. einbauen.

Ich habe das Programm abgetippt. Bei RUN erscheint immer ERROR 8 AT LINE xxx. Zeile xxx ist jedoch einwandfrei. Wo liegt der Fehler?

Der Fehler liegt bei ERROR 8, nicht in der Zeile, die bei der Fehlermeldung gelistet wird.

ERROR 8 bedeutet, daß dem Computer beim Versuch, eine Zahl in eine Variable einzulesen (in der Regel mit READ), ein Text untergejubelt wurde. Da Zahlen in DATA-Zeilen schon als Text interpretiert werden, wenn ein Leerzeichen enthalten ist, kann dies ERROR 8 verursachen. Auch doppelte Kommas verursachen diesen Fehler. (Ein Leerstring ist ja schließlich auch keine Zahl!) Weniger häufig, aber durchaus vorkommend, sind Nullen, die als Buchstabe 0 eingegeben werden (oder 1 oder 1 statt 1). Es ist man jedoch keineswegs notwendig, gleich das ganze Listing nochmals Zahl für Zahl zu vergleichen. Ein kleiner Trick hilft weiter: Normalerweise befindet sich der READ-Befehl innerhalb einer FOR-NEXT-Schleife oder einer GO-TO-Schleife mit Vergleich und Zählvariable (wenn nicht, einfach einfügen: z.B. vor READ TEST = TEST + 1 schreiben). Bricht der Computer nach RUN mit ERROR 8 ab, so gibt TEST bzw. der aktuelle Wert der Zählvariablen (die Variable nach FOR) die Nummer des Datens aus, bei dem der Computer nicht mehr weiterkommt (einfach PRINT TEST oder eben die Schleifenvariable aus Ihrem Programm - statt TEST). Sollte die Schleife (wie bei TITAN) nicht bei 0 beginnen, so ist dieser Startwert (die Zahl nach dem "=" in der FOR-Zeile) von dem ermittelten Wert abzuziehen.

Jetzt brauchen Sie die DATAs nur noch nachzuzählen, wobei Sie beim ersten Datum beginnen und bis zu dem ermittelten Wert weiterzählen. Hier finden Sie (± 1) das defekte Datum. Dieser Vorgang ist so lange zu wieder-

holen, bis kein ERROR 8 mehr auftritt.

Gibt es einen Adapter, mit dem ich die Cassetten des Atari-Telespiels VCS 2600 verwenden kann?

Nein! Ist aber auch nicht besonders wünschenswert, wenn man einmal die Grafikqualität der beiden Geräte vergleicht!

Ich entdeckte vor kurzem eine Diskette "Ex. Basic". Kann ich damit Programme des TI 99/4A verwenden?

Nein! "Ex" ist eine gebräuchliche Abkürzung für "exten-

ded", also erweitert. Dieses Programm ist nichts anderes als ein erweitertes Atari-Basic!

Ich bin Besitzer der 80-Zeichen-Karte mit Oldranncchip. Mit PRINT USR (49152) wird diese Option aktiviert. Wie kann ich nun wieder in den 40-Zeichen-Modus schalten, ohne daß das Programm unterbrochen wird? Wie kann ich sonst hochauflösende Grafik und 80 Zeichen gleichzeitig betreiben?

Diese Frage gebe ich hiermit an unsere Leser weiter. Tips bitte direkt an Wolfgang Deutsch, Stephansplatz 3, 3000 Hannover 1.

Graphic Arts Departement

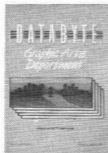
Im Rausch der Farben

Der erste Eindruck war etwas enttäuschend; schon wieder ein neues Malprogramm, und noch dazu im etwas blockartigen Grafikmodus 7. Nachdem die Demo-Bilder auch nicht gerade umwerfend waren, wollte ich es schon beiseite legen, doch ein Blick in die Anleitung des Programms förderte eine ganze Reihe von erstaunlichen Fähigkeiten zutage.

Das fängt bereits damit an, daß man alle 128 Farben in einem Bild verwenden kann. Welches andere Malprogramm kann das schon? "GAD" hat einen unglaublich umfangreichen Satz an Grafik-Funktionen, der alle bisherigen Programme wie "Atari-Artist" oder "MicroPainter" in den Schatten stellt. Ferner gibt es drei Editoren, mit denen Zeichensätze, Füllmuster und Pinselformen verändert werden können. Das ist aber noch nicht alles. Das Programm kann Ihre Kunstwerke auch zu Papier bringen und enthält ein Basic-Programm, mit dem "GAD"-Bilder in eigenen Programmen verwendet werden können.

Nach dem Laden des Programms (dauert sehr lange) hat man die Auswahl zwischen "Arts Gallery" und "Arts Bench". Letzteres ist das Malprogramm, zur "Gallery" kommen wir später. Zuerst die schlechte Nachricht: "Arts Bench" arbeitet ausschließlich in GRAPHICS 7, d.h. mit einer

Auflösung von nur 160x96 Punkten in 4 Farben. Warum nicht die höchste Auflösung mit 160x192 Punkten gewählt wurde, bleibt unklar. Mit diesem Modus arbeiten z.B. "Atari-Artist" und eine ganze Reihe anderer Malprogramme. Durch die geringe vertikale Auflösung wirken die Bilder immer etwas blockartig.



Hervorragend ist aber die Möglichkeit, mit allen 128 Farben des Atari zu arbeiten. Man verfügt über vier Farbtöpfe, die in jeder Zeile mit einem anderen Farbton gefüllt werden können. Sie wissen bestimmt (wenn Sie die Assemblerdecke der letzten Ausgabe gelesen haben), daß hierbei ein wenig mit DLLs gezaubert wird. In einer horizontalen Zeile können zwar nur vier verschiedene Farben verwendet werden, im ganzen Bild

Atari • Atari • Atari • Atari

Elektronische Schaltungen konstruieren mit dem ATARI
 Bildschirmorientiertes Arbeiten mit dem Joystick
 Widerstand, Kondensator, Diode, Transistor, Gator,
 Inverter, ICA usw.
 - Drucke-anscopy
 - Laden und Sägen auf Diskette
 - Bauteilebibliothek führen usw.

**ATARI
 XL/XE 64K
 40.- DM**

inkl. Versand
 Bestellung:
 Scheck/Check,
 Info kostenlos!
 Nachnahme
 = 6.- DM

Jürgen Dörr
 Einsteinst. 6
 6520 Worms 26
 ☎ 06241/34140

sind jedoch 128 möglich. Die Farbwechsel lassen sich bequem mit dem Joystick programmieren.

Die Menge der Grafik-Funktionen von "GAD" ist schon erstaunlich. Neben den regulären Funktionen wie Linien, Rechtecke, Kreise u.ä. bietet es noch ein zweifaches Zoom, Text in vielen verschiedenen Größen und Operationen zum Verdrehen, Spiegeln und Kopieren von Bildteilen. Das ganze Bild kann in alle Richtungen verschoben werden. Ellipsen sind möglich, und auch die Koordinaten des Cursors können abgefragt oder gezielt eingegeben werden. Damit man den Überblick nicht verliert, bietet "GAD" einen Help-Screen, in dem alle Befehle zusammengefaßt sind.

Eine wirklich nützliche Einrichtung sind die drei Editoren. Da wäre zunächst ein Font-Editor, mit dem Zeichensätze geladen, verändert und gespeichert werden können. Weiterhin gibt es einen Brush-Editor; er dient zum Verändern der Pinselform. Ein ganzer Satz von Pinseln kann auf Diskette gespeichert und bei Bedarf nachgeladen werden. Mit dem dritten Editor werden Füllmuster verändert und gefüllte Kreise und Rechtecke gezeichnet. Die Editoren sind reichlich mit Hilfsfunktionen ausgestattet.

Der Programmteil "Arts Gallery" dient zum Anzeigen und Ausdrucken von "GAD"-Bildern und kann auf eigene Disketten kopiert werden. Die Option zum Ausdrucken der Bilder funktioniert dann problemlos, wenn man einen der im Programm gespeicherten Drucker besitzt (Epson etc.). Andere Drucker können mit einem Menu angepaßt werden, allerdings sollte man sich dazu in seinem Drucker-Handbuch etwas auskennen. Grundvoraussetzung ist, daß Ihr Drucker einen Modus mit 960 Punkten pro Zeile besitzt. Damit dürften sich 1029-Besitzer schon auf dem Abstellgleis befinden. Der Ausdruck ist nur in einer einzigen Größe (ca. DIN A5) möglich, Farben werden als Graustufen dargestellt (wählbar).

Sie finden auch ein kleines Maschinenprogramm auf der Diskette, mit dem die Bilder in eigenen Programmen geladen werden können. Für Assembler-Programmierer ist sogar der Quellcode enthalten.

Die englischsprachige Anleitung von "Graphic Arts Departement" umfaßt 8 Seiten und enthält alle wesentlichen Informationen. Sieht man von der geringen Auflösung ab, dann überzeugt "GAD" durch die Unmenge an Funktionen.

Preis: ca. 70 DM, Disk System: alle Atari-Computer ab 48 K

Weitere Informationen:
 Compy Shop
 Gneisenaustr. 29
 4330 Mülheim

Peter Finzel

The Music-Studio

Mit diesem Programm bietet Ihnen Activision ein komplettes Tonstudio auf Diskette an; Ihr Atari wird zum elektronischen Notenblatt und Orchester. Sie müssen nur die Noten eingeben und die Instrumente auswählen, schon tönt die Melodie aus dem Lautsprecher. Wer mit Noten auf Kriegsfuß steht, kann seine Musik sogar mit der "Music-Paintbox" "malen". Einfacher geht's wirklich nicht mehr.

Das "Music-Studio" besteht aus drei Teilen: dem "Music-Editor", dem "Sound-Engineering-Room" und der "Music-Paintbox". Angenehmerweise befinden sich alle drei Teile permanent im Speicher, so daß nichts von Diskette nachgeladen werden muß.

Music-Editor

Die Zentrale von "Music-Studio" bildet der Editor. Der Bildschirm besteht aus zwei großen Notenzeilen (eine mit Violin-, eine mit Baßschlüssel) und einem Menü in unterem Teil. Die Auswahl geschieht mit dem Joystick (Symbol ansteuern und Knopf drücken). Eben-

so einfach werden die Noten eingegeben, indem man Notendlänge und Instrument anwählt und die Note ein- oder mehrmals auf der Notenzeile ablegt. Der Editor macht seinem Namen alle Ehre und erlaubt es, Noten zu löschen, einzusetzen und zu kopieren. Freilich lassen



sich auch Pausen, Vorzeichen und Punktierungen einsetzen; ebenso kann die Tonart frei gewählt werden.

Beim "Music-Studio" ist es möglich, bis zu 15 Instrumente in einem Musikstück einzusetzen. Merkwürdigerweise können nur drei Instrumente

gleichzeitig ertönen, obwohl der Atari doch über vier Tonkanäle verfügt. Sollte das vielleicht mit der Umsetzung vom C64 zusammenhängen? Lob verdient aber die Auswahl der Instrumente, die durch den Farbcode der Noten bestimmt wird. Auf diese Weise ist man nicht auf ein Instrument pro Stimme festgelegt, sondern kann praktisch mit jeder Note das Instrument wechseln. Auch bis zu 3 Textzeilen können eingegeben werden, die dann beim Abspielen mit den Noten über den Bildschirm laufen.

Tonteachnik

Im "Sound-Engineering-Room" können Sie nach Herzenslust mit den akustischen Möglichkeiten des Atari experimentieren. Acht Schieberegler verändern die ADSR-Charakteristik der 15 Instrumente, spezielle Verzerrungsmodi des POKEY-Chips erzeugen z.B. Schlagzeugeffekte. Die 15 Instrumente entsprechen dann direkt den Farbdarstellungen im "Music-Editor".

Paintbox

Wer mit Noten, Vorzeichen

und Pausen nichts anzufangen weiß, kann die Musik auch über die "Paintbox" erzeugen. Anstelle von Noten werden nun verschieden lange, farbige Balken auf die Zeile gemalt; je länger der Balken, desto länger der Ton. Durch Druck auf den Joystick kann diese Darstellung in normale Noten umgewandelt und mit dem "Music-Editor" weiter bearbeitet werden.

"The Music-Studio" wird auf einer Diskette mit den Versionen für Atari und C64 geliefert. Das englischsprachige Handbuch ist mit fast 60 Seiten recht ausführlich. Merkwürdig ist jedoch, daß eine Funktion nicht für den Atari übernommen wurde, die das Programm sicherlich noch aufwerten würde: gemeint ist der Ausdruck. Alle anderen Versionen von "Music-Studio" können die Noten zu Papier bringen.

System: Atari ab 48 K.
Preis: ca. 69,- DM, Disk

Weitere Informationen:
Compy-Shop
Gneisenauerstr. 29
4330 Mülheim
Peter Finzel

Neues für den Gabentisch

Die Tage werden kürzer, die Nächte länger, die Zeit für Computerspiele bricht an. Gerade rechtzeitig zum Weihnachtsfest kommen zwei neue Spiele vom AMC-Verlag sowie ein neu veröffentlichter Klassiker von Databyte.

Bilbo

Kennen Sie noch den Spielautomaten "Lady Bug"? Wenn Ihnen dieses an den Klassiker "Pacman" angelehnte Spiel Freude macht, dann ist "Bilbo" genau das Richtige für Sie. Der mit dem Joystick gesteuerte Zwerg rast durch ein Labyrinth, sammelt Punkte und wird dabei - wie nicht anders zu erwarten - von Geistern verfolgt. Das Spiel erhält seinen Reiz durch eine Anzahl von Drehtüren, die der Zwerg bei seiner Reise um-

legen kann, die Geister jedoch nicht. Sammelt man die Buchstaben EXTRA (in dieser Reihenfolge!), dann läßt ein Multiplikator den Score in die Höhe schnellen.

Unsere Beta-Version hatte zwar noch nicht alle Features, aber man kann schon jetzt sagen, daß das Spiel grafisch gut gestaltet und technisch perfekt programmiert ist. Nur die Spielidee ist schon etwas abgenutzt. Fällt den deutschen Programmierern außer "Pacman" und Konsorten nichts mehr Neues ein?

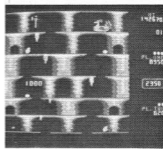
Wer Labyrinth-Spiele mag, wird mit "Bilbo" dennoch zufrieden sein. Der Preis für die Diskette ist mit ca. 20,- DM jedenfalls sehr günstig.

Tales of Dragons and Cavernen

Auch hinter diesem etwas langen Titel verbirgt sich ein neues Spiel vom AMC-Verlag. Es geht über vier Bildschirme, in denen man mit einem Seil von Stock zu Stock klettert, um einen wertvollen Vogel zu retten ("Donkey-Kong" läßt grü-

ben). Natürlich ist es fast unvermeidlich, daß einem dabei herabfallende Eier (!) und Steine behindern.

Lesder lag uns noch keine testfähige Version des Spiels vor. Die Bilder, die wir erhielten, machen allerdings keinen schlechten Eindruck. Auch dieses Spiel kostet rund 20,- DM auf Diskette.



Die Spiele Tales of Dragons and Cavernen aus dem AMC-Verlag



Aztec

Dieses Programm ist ein echter Klassiker unter den Computerspielen. "Aztec" war eines der ältesten Action-Adventures (allerdings nur für Apple-II-Computer). Nun gibt es eine Version für Atari-Computer, die nicht mit Tasten – wie beim Original – , sondern per Joystick gesteuert wird. Die großformatige Spielfigur, die stark an Indiana Jones erinnert, kann lau-

fen, rennen, kriechen, springen und noch einiges mehr. Gegenspieler sind einige recht gut animierte Monster wie Saurier, Löwen und Schlangen.

Ziel des Spiels ist es, in eine alte Pyramide einzudringen und den dort versteckten Goldschatz zutage zu fördern. Viele Hilfsmittel findet man in der Pyramide, wenn man fleißig die Knochenhaufen der zuvor gescheiterten Abenteurer durchsucht. Da das Spiel schon einige Jahre alt ist, sind Grafik und Bedienung teilweise nicht so ausgefeilt, wie man es heute gewohnt ist. Wer jedoch einen echten Klassiker in seine Software-Sammlung einreihen will, sollte bei "Aztec" zum Preis von ca. 60,- DM zugreifen.

Weitere Informationen:
AMC-Verlag,
Blücherstr. 17,
6200 Wiesbaden
Compy-Shop,
Gneissmastr. 29,
4330 Mülheim
Peter Flassd

Mercenary – Escape from Targ

Paul Woakes, der Autor von "Encounter", hat wieder einmal zugeschlagen! Für sein neues Werk "Mercenary" hat er sich reichlich Zeit gelassen. Doch das Warten auf dieses sehr komplexe Programm, eine Mischung aus Flugsimulator, Action und Adventure, hat sich gelohnt.

Sie schlüpfen in die Rolle eines Raumfahrers im 21. Jahrhundert, der aufgrund eines Maschinenschadens auf dem Planeten Targ notlanden muß. Während Sie unverletzt bleiben, wird das Raumschiff bis auf den Bordcomputer Benson, der Sie mit wichtigen Informationen versorgt, völlig zerstört. Die Ureinwohner des Planeten, die Palyars, kämpfen gegen die Mechanoids, eine übermächtige Roboter-Rasse, die den Planeten bevölkert. Sie mußten sich in eine Raumfestung, die in einer Umlaufbahn um den Planeten kreist, zurückziehen.

Die Aufgabe des Spielers besteht nun darin, genügend Geld für ein neues Raumschiff zu

verdienen, mit dem man von Targ entkommen kann. Zu Beginn steht Ihnen ein Gleiter zur Verfügung, mit dem Sie den Planeten erkunden können. Man erkennt eine Menge Gebäude, Brücken, Radarposten und sogar Reklameschilder.

Die Grafik der getesteten Atari-Version fasziniert durch die Vielzahl von Objekten, die alle in hochauflösender Vektor-Grafik dargestellt sind und sich flackerfrei mit beeindruckender Geschwindigkeit aufbauen. Es ist schon ein Erlebnis, wenn man z.B. über eine Brücke fliegt, hinabstürzt und schließlich unter ihr hindurchgleitet. "Elite" wirkt gegen diese schnellen 3-D-Grafik-Routinen wie ein Basic-Programm.

Aber damit nicht genug! Wenn Sie bestimmte Gebäude betreten, können Sie per Aufzug die Unterwelt Targs besuchen. In den unzähligen Zimmern und Korridoren finden Sie eine Reihe von nützlichen Gegenständen. Zehn davon darf man mitnehmen. Über den

H.G. Dreiser, Soft- und Hardware

Wir bekommen laufend die aktuellsten Produkte für den Atari 600/800/130 XL/28 sowie Atari ST. Nutzen Sie unseren Telefon- und Auftragservice zu den angegebenen Zeiten, damit auch Sie über die Neuheiten informiert sind.

Atari-Cassetten-Software		Atari-ST-Software	
Klick Start	9,90 DM	Fire Blaster	90,90 DM
Greatest Hits 1	31,90 DM	Warzone	90,90 DM
Smash Hits 5	39,90 DM	Football Factory	71,90 DM
HardBall	26,90 DM	Hecker 2	79,90 DM
Connel Game	34,90 DM	Strip Poker	62,90 DM
AirLine	11,90 DM	Electronic Pool	92,90 DM
Serfiqashe	11,90 DM	Kernie	72,90 DM
		Schach (DL)	79,90 DM
		(Schiffe 2)	99,90 DM
Atari-Disketten-Software			64,90 DM
Flight Sm.2	163,90 DM	Space Station	72,90 DM
Sun Star	42,90 DM	Starfighter	79,90 DM
Landerboard	60,90 DM	Water Games (DL)	77,90 DM
Hard Ball	67,90 DM	Mercenary (DL)	72,90 DM
Silver Service	80,90 DM	ST-Touki (DL)	95,90 DM
Smash Hits 5	52,90 DM	K-Spigh (DL)	114,90 DM
Mega Shads	77,90 DM		

Finden Sie unsere Größtkarte auf Dreiser Soft u. Hardware, im Ringweg 1, D-5200 Bonn 1, 0-220/28404. Mo. bis Fr. von 11:30-20:00 Uhr, Sa. von 14:00-18:00 Uhr oder Auftragsannahme rund um die Uhr.

Bordcomputer Benson erhalten Sie auch Job-Angebote wie das Transportieren eines bestimmten Gegenstands zur Raumfestung, oder Sie verbünden sich mit den Palyars, um ihnen im Kampf gegen die Mechanoids zu helfen. Bei erfolgreicher Mission erhöht sich Ihr Geldkonto.

Zum Glück läßt sich der Spielstand speichern, so daß man zu einem späteren Zeitpunkt weiterspielen kann. Dies ist auch bitter nötig, denn Raumschiffe sind bekanntlich sehr teuer. Noch einen kleinen

Gag bietet die Atari-Version. Schießt man das Commodore-Reklameschild ab, meldet Benson fröhlich: "Good Shoot".

Dieses Spitzenprogramm sollte in der Software-Sammlung von Atari-Fans, die in letzter Zeit nicht gerade mit guten Programmen überhäuft wurden, nicht fehlen.

System: Atari Computer, C 64, C 128
Preis: ca. 49,- DM Disk
ca. 39,- DM Cass.

Bernd Müller

AMC

VERLAG – WIESBADEN

Airten Stürmer, Blücherstr. 17, 6200 Wiesbaden

Software für den gehobenen Geschmack
ATARI 800 XL / 600 XL (64K) / 130 XE

Erhältlich:

PYRAMIDOS	AMC	DM 29,-
MIKE'S Slotmachine *	AMC	DM 19,-
THE SOUNDMACHINE	FPF	DM 29,80
DESIGNMASTER	FPF	DM 19,80

Neu im Programm:

BILBO*	AMC	DM 19,-
TALES OF DRAGONS AND CAVEMEN*	AMC	DM 19,-

(* 3er Pack DM 50,-)

Und alle 8 Wochen neu:

AMC-SOFT	AMC	DM 8,-
Das Disk-Magazin mit Game (Jahresabo. (6 Ausgaben) statt DM 48,- nur DM 40,-)		
Nachnahme (zuzügl. DM 5,- Versandkosten)		
Vorausgabe (keine Versandkosten)		

Händleranfragen erwünscht!
INFO KOSTENLOS ANFORDERN!

Atari-Maschinensprache für Anfänger

Das Assemblerbuch von Peter Finzel ist ein Kurs zur Einführung in die 6502-Maschinensprache, passend zum Atmas-II-Makrosassembler und speziell auf Atari-Computer zugeschnitten. Es werden keinerlei Vorkenntnisse in Assembler erwartet.

In drei großen Abschnitten wird Ihnen die Welt der Maschinenprogrammierung systematisch nähergebracht. Der erste Teil beschäftigt sich mit den Grundlagen wie Binär- und He-

xadezimalsystem und dem Aufbau des 6502-Prozessors. Dort erfahren Sie auch, was eine Direktive, ein Opcode und eine Adressierungssart ist.

Im zweiten Teil erwartet Sie der Befehlssatz des 6502-Prozessors. Viele Beispiele, die Sie gleich mit Atmas-II eintippen und ausprobieren können, zeigen die Befehle im praktischen Einsatz.

Leider nützt Ihnen die Kenntnis der 6502-Befehle we-

nig, wenn Sie mit den speziellen Eigenschaften des Ataris wie Custom-Chips, Interrupts und dem Betriebssystem nicht vertraut sind. Aus diesem Grund wurde der dritte Teil allein diesen Anwendungen gewidmet. Hier finden Sie ein Beispiel zur Ein-/Ausgabeprogrammierung, zur Player-Missile-Grafik und nicht zuletzt zwei Beispiele zur Interrupt-Technik.

Zu allen Listings gibt es eine ganze Menge an Hintergrundinformationen, die Sie bei eigenen Assembler-Projekten unterstützen. Nebenbei erhalten Sie noch eine Fülle praktischer Tips, z.B. wie man Assembler-Programme mit Basic koppelt.

wie man selbststartende Programme schreibt und vieles mehr. Ein alphabetisch geordnetes Assembler-Vokabular im Anhang des Buches wird Ihnen auch beim Lesen von Fachartikeln eine Hilfe sein, wenn Sie mit Begriffen wie LSB, Zero-Page oder DMA noch nicht vertraut sind.

Das Assemblerbuch umfaßt 169 Seiten im Format DIN A5 mit abwaschbarem Umschlag. Es kann zum Preis von 29,80 DM direkt über den Verlag Computer-Kontakt bezogen werden. Bestellschein siehe "take 9".

Oskar Schölsmann

6502-Maschinensprache

Ein Kurs für Atari-User - 3. Folge

Diesmal wird es ernst; wir beschäftigen uns mit dem ersten Teil des Befehlssatzes des 6502-Prozessors und bringen erste Programmbeispiele dazu. In der letzten Folge sind wir gerade bis zum LDA-Befehl vorgegangen und haben uns die wichtigsten Adressierungsarten näher angesehen.

Bevor wir uns mit der Besprechung des Befehlssatzes fortfahren, sollten wir uns noch mit einer wichtigen Einrichtung des 6502 vertraut machen, den Prozessor-Flags. Sie zeigen besondere Zustände an (wenn z.B. bei einem Rechenvorgang der Zahlenbereich überschritten wurde). Abhängig von diesem Flag kann der Programmablauf beeinflusst werden. Hier die Bedeutung der sieben Flags.

Z: Das Zero-Flag zeigt an, daß das Ergebnis der letzten Operation null war. Es muß sich dabei nicht um eine Rechenoperation handeln; auch wenn eine Null durch einen Ladevorgang in ein Register kommt, wird das Z-Flag gesetzt.

N: Das Negative-Flag zeigt an, wenn das Ergebnis der letzten Operation als negativ interpretiert werden kann. (Das Bit mit dem höchsten Wert ist gesetzt.)

C: Das Carry-Flag wird gesetzt, wenn ein Überlauf bei einem Rechenvorgang aufgetreten ist. Die größte Zahl, die der 6502 im Akku darstellen kann, ist 255. Sollte das Ergebnis einer Rechnung größer sein, so wird das C-Flag gesetzt. In vielen Fällen kann man sich dieses Flag auch als "neuntes Bit" des Akkus vorstellen.

V: Das Overflow-Flag zeigt an, ob bei Rechenoperationen mit vorzeichenbehafteten Zahlen ein Überlauf aufgetreten ist. Stellen Sie sich das bitte so vor: Das achte Bit gibt bei einer derartigen Zahl ihr Vorzeichen an. Findet bei einer Addition ein Überlauf vom siebten ins achte Bit statt, so würde das Vorzeichen verfälscht. Das Overflow-Flag wäre in diesem Fall gesetzt und zeigt an, daß das Vorzeichen nicht mehr stimmt.

B: Das Break-Flag wird gesetzt, wenn ein BRK-Befehl ausgeführt wird.

I: Das Interrupt-Disable-Flag kann eine bestimmte Gruppe von Interrupts (die IRQs) verhindern. Es empfiehlt sich allerdings beim Atari, dieses Flag nur zu verwenden, wenn man sich mit Interrupts auskennt. Setzt man dieses Flag, so nimmt der Compu-

ter z.B. keine Tastatureingaben mehr an!

D: Mit dem Dezimal-Flag läßt sich der Prozessor in einen dezimalen Rechenmodus umschalten. Er benutzt dann den sog. BCD-Code, mit dem in einer 8-Bit-Zahl zwei dezimale Ziffern untergebracht werden können. Die größte Zahl ist 99 (dargestellt als \$99); sollte das Ergebnis eines Rechenvorganges größer sein, so wird das C-Flag gesetzt.

Sie haben vielleicht bemerkt, daß die beiden letzten Flags eine gewisse Sonderstellung einnehmen, da sie keine Zustände anzeigen, sondern gewisse Eigenschaften des Prozessors umschalten.

Bei der nun folgenden Vorstellung der Befehle wird immer mitangegeben, welches Flag sie beeinflussen.

Laden und Speichern

Die zweifellos am häufigsten verwendeten Befehle dienen zum Laden und Speichern des Akkumulators. Den LDA-Befehl kennen Sie bereits aus der letzten Folge; er beeinflusst das N- und das Z-Flag. Die Umkehrung davon heißt STA (Store Accumulator) und dient zum Ablegen des Akkuinhalts im Speicher. Flags werden nicht

verändert. Die zulässigen Adressierungsarten können Sie (wie auch bei den folgenden Befehlen) aus Tabelle 1 entnehmen. Eine Kombination von Adressierungsart und Befehl ist erlaubt, wenn am Kreuzungspunkt ein Code in der Tabelle eingetragen ist. Die Codes brauchen Sie sich übrigens nicht zu merken, sie werden vom Assembler automatisch erzeugt.

Indexregister

Wie für den Akku gibt es auch Lade- und Speicherbefehle für die beiden Indexregister X und Y. Zum Laden des X-Registers dient der Befehl LDX, gespeichert wird es mit STX. Analog heißen die auf das Y-Register wirkenden Befehle LDY bzw. STY. Im Gegensatz zu den auf den Akku bezogenen Befehlen sind hier keine indirekten Adressierungsarten zulässig (s. Tabelle 1). Während LDX und LDY wiederum das Z- und das N-Flag beeinflussen, lassen die Speicherbefehle die Flags unverändert.

Mit diesen wenigen Befehlen (und einigen Assembler-Direktiven aus der zweiten Folge) kann man schon ein erstes einfaches Programm zusammenstellen. In Listing 1 verwenden wir die besprochenen Lade-

und Speicherbefehle, um drei Farbregister mit neuen Werten zu besetzen. Die Werte werden zunächst per Immediate-Adressierung in die Processor-Register geladen und anschließend mit absolut adressierten Store-Befehlen in die Schattenregister der Farben geschrieben. Die EQU-Direktiven am Anfang des Programms ordnen den Adressen der Schattenregister symbolische Namen zu, um das Ganze verständlicher zu machen. Durch die ORG-Anweisung wird das Programm an die Adresse \$A800 assembliert.

Das Programm ist für "Atmas-II" geschrieben, kann aber auch für andere Assembler modifiziert werden. Zum Starten des Programms gehen Sie in den Monitor (bei Atmas mit Control-P) und benutzen die GOTO-Funktion zur Adresse \$A800 (5-Zeichen nicht eingeben).

Den letzten Befehl des Programms haben wir bisher noch nicht besprochen. Wir werden das auf später verschieben, da seine Wirkungsweise recht kompliziert ist. Merken Sie sich nur, daß er BRK (für Break) heißt und die Rückkehr zum Monitor zusammen mit einer Anzeige der Processor-Register bewirkt.

Transferbefehle

Eine weitere Gruppe von Befehlen erlaubt den Austausch von Informationen zwischen den Registern. Die Liste der Transferbefehle lautet:

TAX: Der Akku wird ins X-Register kopiert.

TXA: Das X-Register wird in den Akku kopiert.

TAY: Der Akku wird ins Y-Register kopiert.

TYA: Das Y-Register wird in den Akku kopiert.

TSX: Dies ist der einzige Befehl, mit dem der Stapelzeiger gelesen werden kann. Sein Inhalt wird ins X-Register kopiert.

TXS: Mit diesem Befehl kann der Stapelzeiger mit einem neuen Wert (dem Inhalt des X-Registers) überschrieben werden.

Alle Transferbefehle außer TXS beeinflussen das N- und das Z-Flag; TXS verändert keine Flags. Hier wird eine Adressierungsart verwendet, die wir in der letzten Folge noch nicht besprochen haben. Sie nennt sich implied (implizit) und bedeutet nichts anderes, als daß der Befehl keinen Operanden benötigt und daher nur ein Byte lang ist.

Zählbefehle

Eine der hervorstechendsten Eigenschaften der beiden Index-Register ist, daß Sie problemlos auf- und abwärts zählen können. Dazu dienen die folgenden vier Befehle:

INX: Increment-X-Register. Zum Inhalt des X-Registers wird eine Eins addiert. Mehr oder minder entspricht das der in Basic oft benutzten Form $X = X + 1$.

DEX: Vom X-Register wird eine Eins subtrahiert.

INY: Zum Y-Register wird eine Eins addiert.

DEY: Vom Y-Register wird eine Eins subtrahiert.

Alle vier Befehle beeinflussen wieder das Z- und das N-Flag und benutzen die implizite Adressierung. Sie werden später noch sehen, daß man mit diesen Befehlen und Anweisungen zum Abfragen der Flags bequem Schließen programmieren kann. Während diese Befehle nur auf die Index-Register wirken, gibt es noch zwei weitere, die beliebige Speicherzellen herauf- und herabzählen können:

INC: Increment Memory. Addiert eine Eins zu einer beliebigen Speicherzelle.

DEC: Decrement Memory. Subtrahiert eine Eins von einer beliebigen Speicherzelle.

Die Adressierungen können Sie Tabelle 1 entnehmen. Auch diese beiden Befehle wirken auf das N- und das Z-Flag.

Addition und Subtraktion

Eine der wichtigsten Eigenschaften eines Mikroprozessors

ist sicherlich die Fähigkeit zu rechnen. Leider ist der 6502 (wie die meisten 8-Bit-Prozessoren) auf diesem Gebiet nicht eben opulent ausgestattet; er kennt nur Addition und Subtraktion. Alles andere (z.B. Division, Multiplikation) muß man selbst programmieren.

ADC: Add with Carry. Ein Byte aus dem Speicher wird zum Akku-Inhalt addiert. Da auch das Carry-Flag dazugezählt wird, kann man auf diese Weise mehrfach genaue Additionen ausführen. Das Ergebnis der Addition bleibt im Akku zurück; wurde der Zahlenbereich überschritten, so ist nach dem ADC-Befehl das Carry-Flag gesetzt. Vor

nutzt, nur diesmal in umgekehrter Bedeutung. Das C-Flag ist nämlich beim Auftreten eines Übertrags zum nächsten Byte gleich null. Aus diesem Grund muß das C-Flag auch vor einem Subtraktionsbefehl gesetzt werden.

ADC und SBC beeinflussen N-, Z-, C- und auch das V-Flag.

Verändern der Flags

Bei der Besprechung der letzten beiden Befehle haben wir gesehen, daß es möglich sein muß, die Flags gezielt zu setzen oder zurückzusetzen. Zu diesem Zweck gibt es eine Reihe von Befehlen:

SEC: Set Carry-Flag. Das C-Flag wird auf eins gesetzt.

Befehl	Implizit	Immediate	Zero-Page	Zero-Page,X	Zero-Page,Y	Relative	Relative,X	Relative,Y	Indexed,Direct,X	Indexed,Direct,Y
ADC		69	65	75		6D	7D	79	A1	71
CLC	1B									
CLD	5B									
CLV	5D									
DEC			C6	D6		CE	DE			
DEX	CA									
DEY	8B									
INC			E6	F6		EE	FE			
INX	EA									
INY	CB									
LDR		A9	AD	B5		AD	BD	89	A1	B1
LDX	A1	A2	A6	B6	B6	AE	BE	9E		
LDY	91	92	94	94		9C	9C	8C	E1	F1
SBC		E9	E5	F5		ED	FD	F9		
SEC	3F									
SED	F8									
SET	78									
STA			85	95		8D	9D	89	B1	91
STX			84	94	96	8C	9C	8C		
STY			84	94	96	8C	9C	8C		
TAX	AA									
TAY	A0									
TYR	90									
TEX	8A									
TEY	8A									
TXS	9A									

Tabelle 1: 6502-Befehle, Teil 1

jeder Addition sollte das Carry-Flag zurückgesetzt werden.

SBC: Subtract with Borrow. Der Befehl subtrahiert eine Speicherzelle vom Akkumulator und berücksichtigt einen Überlauf von vorhergehenden Rechenoperationen. Auch hier wird das Carry-Flag als Merker für einen Übertrag be-

CLD: Clear Decimal Mode. Der Befehl schaltet wieder in den binären Rechenmodus zurück (D-Flag gleich null).

CLC: Clear Carry. Das C-Flag wird auf null gesetzt.

SED: Set Decimal Mode. Das D-Flag wird gesetzt und bringt den Prozessor somit in die dezimale Rechenweise.

SEI: Set Interrupt Disable. Das I-Flag wird gesetzt und verriegelt eine Gruppe von Interrupts. Da Interrupts erst Thema einer späteren Folge sind, soll die nähere Besprechung dieses Befehls noch ein wenig verschoben werden.

CLI: Clear Interrupt Flag. Die IRQ-Interrupts werden wieder freigegeben.

CLV: Clear Overflow. Das Overflow-Flag wird gelöscht. Es gibt keinen ausdrücklichen Befehl zum Setzen dieses Flags.

Alle Befehle der zuletzt besprochenen Gruppe benutzen ausschließlich die implizite Adressierungsart und sind daher 1-Byte-Befehle.

Beispiel

Da alle Theorie bekanntlich grau ist, wollen wir uns nun ein Beispiel ansehen, in dem einige der zuvor besprochenen Befehle vorkommen. Listing 2 zeigt ein Assembler-Programm, das einige Zeichen am Bildschirm ausbildet. Da wir uns mit Unterprogrammen und Betriebssystemen noch nicht beschäftigt haben, werden wir den Bildschirm-Code der Zeichen einfach in den Video-Speicher schreiben. Dieser Speicherabschnitt umfaßt 960 Byte, die jeweils ein Zeichen auf dem Bildschirm festlegen (jedenfalls in GRAPHICS 0). Das erste Byte bestimmt das Zeichen in der linken oberen Ecke, die 39 folgenden Byte bilden die erste Zeile. Danach schließen sich wieder 40 Byte für die zweite Zeile an usw.

Da der Videospeicher beim Atari dank des Videoprozessors Antic nicht an einen bestimmten Speicherbereich gebunden ist, müssen wir uns seine Anfangsadresse besorgen. Wir finden sie in den Speicherzellen \$58 und \$59 der Zero-Page, die gewöhnlich mit dem Label SAVMISC bezeichnet werden. Deshalb befindet sich im Anfang des Listings eine Definition mit einer entsprechenden EQU-Direktive. Natürlich könnten wir auch die Zahlenwerte benutzen, aber denken Sie immer daran, daß ein Li-

sting so verständlich wie nur möglich sein sollte. Mit dem zweiten EQU wird eine Variable in der Zero-Page definiert, die wir benötigen, um die indirekte Adressierungsart einsetzen zu können.

Das Programm soll erlauben, daß die Zeichen an eine beliebige Stelle im Video-Speicher abgelegt werden. Daher wird zur Anfangsadresse in SAVMISC eine Konstante (die auch per EQU definiert wurde) addiert und das Ergebnis in ZEIGER abgelegt. Nehmen wir an, die Zeichen sollen ungefähr in der Mitte des Bildschirms auftauchen, so kann man z. B. 498 addieren. Mit mehreren indirekt-indizierten STA-Befehlen werden die Zeichen ab dieser Stelle ins Video-RAM geschrieben, die Weiterschaltung auf das nächste Zeichen erfolgt mit dem Y-Register, das zwischen zwei STA-Befehlen erhöht wird.

Interessant ist auch die Addition der Konstanten zu SAVMISC. Da es sich um 2-Byte-Werte handelt, ist eine mehrfach genaue Addition nötig. Das C-Flag wird auf Null gesetzt, dann werden zuerst die LSBs (die niederwertigen Bytes) addiert. Sollte dabei ein Überlauf auftreten, so bleibt dieser im Carry-Flag und wird bei der anschließenden Addition der MSBs (der höherwertigen Bytes) berücksichtigt. Deshalb darf vor dem zweiten ADC-Befehl kein CLC erfolgen! Die Zerlegung der Konstanten ABSTAND in LSB und MSB kann beim ATMAS-II recht einfach erfolgen. Man schreibt ABSTAND:L für das LSB bzw. ABSTAND:H für das MSB.

In der nächsten Folge werden wir die restlichen 6502-Befehle besprechen und uns dabei einige weitere (dann schon etwas kompliziertere) Befehle ansehen.

Peter Finzel

**Ihr direkter Draht
zur Redaktion:
Telefon 0 72 52 / 4 29 48
Für Briefe:
Computer Kontakt
Postfach 16 40
7518 Bretten**

Alle Atari-User!

Wenn Sie dieses Heft bereits von hinten betrachtet haben, dann sind Sie schon informiert: Neben der CK wird demnächst noch viel mehr für die Atari-Freunde aus unserem Verlag kommen: das ATARI magazin. Für User der kleinen 8-Bit-Ataris wie auch für die Besitzer der Atari-ST-Computer wird das neue Magazin eine geballte Ladung an Information bringen.

Geboten wird aber noch mehr: Die Besitzer von Atari-Computern aller Größen sind aufgerufen, durch Programm-Einsendungen am neuen ATARI magazin mitzuwirken. Auch Sie haben die Möglichkeit, zum ständigen freien Mitarbeiter zu werden und so Ihrem Hobby eine lukrative Seite abzugewinnen. Sind Sie interessiert? Dann schreiben Sie uns Name, Adresse und Telefonnummer. Wir setzen uns mit Ihnen in Verbindung.

ATARI magazin
Redaktion
Postfach 1640
7518 Bretten

Listing 1

```
*
* Drei Farbregister laden
*
COLOR1 EQU #2C5
COLOR2 EQU #2C6
COLOR4 EQU #2C8
*
ORG #A00
*
LDA #0A hellgrau
LDX #00 schwarz
LDY #0A hellgruen
STA COLOR4 Rahmen
STX COLOR1 Schrift
STY COLOR2 Hintergrund
BRK
```

Listing 2

```
*
* Einfache Textausgabe
*
SAVMISC EQU #58 Zeiger auf Video-Ram
ZEIGER EQU #D4 Variable in Zero-Page
ABSTAND EQU #9B Abstand zur Bildmitte
*
ORG #A00
*
CLC Carry auf 0
LDA SAVMISC erst die LSBs
ADC #ABSTAND:L addieren
STA ZEIGER und speichern
LDA SAVMISC+1 jetzt die MSBs
ADC #ABSTAND:H addieren
STA ZEIGER+1
LDY #0 Offset=0
LDA #*23 erstes Zeichen
STA (ZEIGER),Y
INY
LDY #*30 Offset=1
STA (ZEIGER),Y
INY
LDY #*35 Offset=2...
STA (ZEIGER),Y
BRK letztes Zeichen
```

16-K-Bibomon

Der 16-K-Bibomon ist ein Maschinensprach-Monitor, der fest in Ihrem Atari-Computer der Serien 800XL und 130XE eingebaut wird. Sie erhalten mit diesem Monitor eine Centronica-Parallel-Schnittstelle und eine Hardcopy-Routine in Ihrem Computer. Diese Hardcopy-Routine können Sie JEDERZEIT durch einen Testendruck auslösen. 4 verschiedene Drucker sind im Anwahlmenü der Hardcopy-Routine enthalten!

Ebenfalls fest im Computer eingebaut wird ein DOS. Sie brauchen also kein DOS mehr zu laden, um Programme zu speichern oder zu laden!

Auch den Monitor erreichen Sie jederzeit durch einen Testendruck. Sie können mit diesem Monitor Speicherbereiche -Diagnosieren, -Listen, -Vergleichen, -Verschieben, -Absuchen. Sie können Seitendoren einlesen, verändern, schreiben. Der Monitor verfügt über einen leistungsfähigen Zeilenassemblier. Und vieles mehr.

Mit dem Monitor bekommen Sie ein ausführliches, 70seitiges, deutsches Handbuch.

Der 16-K-Bibomon arbeitet mit der Speedy 1050 zusammen!

Das alles bekommen Sie exklusiv bei uns - im COMPY-SHOP!

16-K-Bibomon, für Atari 800XL	448,- DM
16-K-Bibomon, für Atari 130XE	DM 448,- DM
Einbauen des 16-K-Bibomon in Ihren Computer, zzgl.	80,- DM
Atari 800XL, 320 KRAM + 16-K-Bibomon	798,- DM
Atari 130XE, 192 KRAM + 16-K-Bibomon	849,- DM

— Testbericht in der COMPUTER KONTAKT 8/9-86 —

FORDERN SIE UNSEREN AUSFÜHRLICHEN PROSPEKT AN!
HÄNDLER-ANFRAGEN ERWÜNSCHT!

Speedy 1050

Speedy 1050 ist ein Hardware-Zusatz, der in Ihr Atari 1050-Laufwerk eingesteckt wird. Sie haben dann sofort mehrere Vorteile:

- Drei Speicherdichten:
 1. 88 KByte Single-Density
 2. 127 KByte Medium-Density
 3. 176 KByte Double-Density
- Drei Übertragungsgeschwindigkeiten:
 1. Atari 1050 Normal — ca. 19.200 Baud
 2. Speedy 1050 Normal — ca. 28.400 Baud
 3. Speedy 1050 Super-Speed — ca. 68.200 Baud
- Die Speedy 1050 ist voll programmierbar
- Formatiert jedes der drei Formate in 2 Sekunden
- Leichter Einbau, von jedem Laien ausführbar
- Benutzerfreundliche Software
- Arbeitet mit dem 16-K-Bibomon zusammen
- Kann kopierschutz Software kopieren
- Digitale Trackanzeige (Speedy T)
- Akustischer Fehlermelder (Speedy T)

WICHTIG! SPEEDY 1050 IST NICHT HAPPY-KOMPATIBEL!

Speedy 1050 N,	198,- DM
Speedy 1050 T, mit Trackanzeige und Summer	298,- DM
Nachrüstset Trackanz. u. Summer f. Speedy 1050 N	119,- DM
Amerikan. Handbuch Speedy 1050, mit ROM-Listing	50,- DM
Einbau der Speedy 1050 in Ihr Laufwerk, zzgl.	80,- DM
Atari 1050 mit Speedy 1050 T	698,- DM

FORDERN SIE UNSEREN AUSFÜHRLICHEN PROSPEKT AN!
HÄNDLER-ANFRAGEN ERWÜNSCHT!

Hardware - Software

Atari 800XL, mit 320 KRAM und 16-K-Bibomon ..	798,- DM
Atari 130XE, mit 192 KRAM und 16-K-Bibomon ..	849,- DM
Atari 1050, mit Speedy N	598,- DM
Atari 1050, mit Speedy T	698,- DM
Atari 1029, mit Grafikdruck-Programm	498,- DM
Grafikdruck-Programm, einzeln	39,- DM
Mannesmann MT 80, anschlussfertig für XL/XE	998,- DM
Okidata M1 182, anschlussfertig für XL/XE	1098,- DM
Okidata M1 192, anschlussfertig für XL/XE	1898,- DM
Printer-Interface für Atari XL/XE	148,- DM
Bbo Bumer,EPROMmer I. XL/XE, b. 32-K-EPROMs ..	248,- DM
SNAUT NEU! Fort 83 Compiler auf Disk oder Kas. ..	98,- DM
Design-Master, das Super-Programm für XL/XE	19,80 DM
Soundmaschine, ebenfalls Super!	29,80 DM

SOLANGE VORRAT REICHT:

RGB-Monitor für XL/XE, mit leichten Gehäuseschäden (Kratzer), anschlussfertig für XL/XE	698,- DM
ATARI POWER-Buch, viele Tips und Tricks	29,- DM
Die HEXENKÜCHE, Buch m. sehr vielen Tips u. Tricks ..	29,- DM
Die HEXENKÜCHE, Disk, z. Buch, m. allen Progr. ..	19,80 DM

WIR SIND FACHHÄNDLER FÜR DAS NEUE SEGA VIDEOSPIEL SYSTEM

FORDERN SIE UNSERE AUSFÜHRLICHE PREISLISTE AN!
KATALOG MIT CA. 170 SEITEN GEGEN 5,00 DM!

ALLE UNSERE PRODUKTE BEKOMMEN SIE IN ÖSTERREICH
BEI DER FIRMA AMV IN WIEN!



Pascal Software Atari XL/XE

kyan pascal für die XL/XE Serie DM 298,-

kyan pascal ist ein mit DOS 2.5 arbeitender Compiler für die Atari-XL/XE-Serie. Es umfasst den vollen Jensen-Wirth Standard und eignet sich für den Anfänger genauso wie für den fortgeschrittenen Programmierer. Es ist blitzschnell und hat folgende Eigenschaften:

- ★ 6502 Maschinencode Compiler erlaubt das Einbinden von Assemblersourcen
- ★ Bildschirmeditor
- ★ Stringbefehle
- ★ Atari-Grafik und Soundunterstützung
- ★ Source Code Linking, Chaining und Random-Files
- ★ 13stellige Floatingpoint-Genauigkeit
- ★ Mit Tutorial/Referenzunterlegen

kyan pascal wird auf einer ungeschützten Single Density Diskette geliefert und benötigt nur 48 K Speicherplatz und ein Laufwerk.

kyan pascal für Ihren Atari Computer bei Ihrem Fachhändler oder direkt von uns.

Händleranfragen: Witt Datentechnik, Am Denkmal 8
l.h. 4802 Halle/Westfalen, 052 01/40 06

Compy-Shop, Gneisenaustr. 29, 4330 Mülheim/Ruhr

Telefon 02 08-4971 69

PETER'S ASSEMBLIERECKE

Software-Blitter

Ein neues Wort ist in aller Munde, der Blitter. Der Amiga hat ihn, der ST bekommt ihn, aber was hat es eigentlich damit auf sich?

Es handelt sich um einen hochspezialisierten Chip, der nur zum Kopieren von Speicherbereichen konstruiert wurde. Besonders geeignet erweist sich ein Blitter zum Bewegen von Objekten auf einem hochauflösenden Bildschirm; daher auch der Name, der für "Block Image Transferer" steht. In der Assemblerecke wollen wir Ihnen ein Programm vorstellen, das die Arbeit eines Blitters simuliert und gleichzeitig über zwei neue Basic-Befehle verfügt.

Im Gegensatz zur Player-Missile Grafik oder Sprites, die über einen separaten Speicher für die Bilddaten verfügen, werden Bobs (Blitter Objects) direkt in den Bildspeicher kopiert. Dies hat den Vorteil, daß Objekte in fast beliebiger Größe und Anzahl bewegt werden können. Die einzige Einschränkung liegt in der vorhandenen Rechenzeit, denn je mehr bzw. je größere Bobs bewegt werden, desto größer ist auch die Datenmenge. Und diese Flut kann gewaltig sein. Während zur Bewegung eines Objektes mit PM-Gratik im günstigsten Fall die Veränderung eines einzigen Registers genügt, müssen bei Bobs häufig mehrere hundert oder tausend Bytes bewegt werden. Einem echten Blitter fällt das jedoch nicht schwer. Er kann per DMA sehr schnell auf den Speicher zugreifen, d.h. praktisch in jedem Arbeitstakt ein Byte lesen oder schreiben. Der 6502 benötigt dazu drei bis sieben Taktzyklen.

GET und PUT

Da unser Atari bekanntlich keinen Blitter hat, müssen wir uns mit einer Software-Lösung behelfen. Damit auch etwas

praktisch Verwertbares dabei herauskommt, werden wir dem Atari-Basic zwei neue Befehle (GET und PUT) hinzufügen. Sie kennen die beiden bereits? Irrtum, wer sich schon mit Basic-Dialekten für 16-Bit-Rechner befahrt hat, weiß, daß diese Befehle einen ganz neuen Sinn bekommen haben.

Mit GET kann man einem hochauflösenden Bildschirm einen rechteckigen Bereich entnehmen und seine Daten in einem String ablegen. PUT erledigt das Gegenteil; die Daten aus dem String werden in den HiRes-Screen zurückgeschrieben. Allerdings kann man den Bildausschnitt (nichts anderes als ein Bob) nun an jede beliebige Stelle des Bildschirms kopieren.

Diese beiden Befehle eröffnen dem Programmierer eine ganze Fülle neuer Möglichkeiten. Sie können nicht nur Objekte über den Bildschirm bewegen, sondern auch bei Windows beliebig sein und Symbole jeder Art anzeigen und wieder löschen.

Moderne Benutzeroberflächen wie das GEM des Atari ST (oder Intuition beim Amiga) arbeiten vorwiegend mit Symbolen (Icons) statt Texten. Wenn es Ihnen Spaß macht, können Sie mit GET und PUT ähnliche Programme schreiben. Sie benötigen dazu mehrere Symbole, die am Bildschirm gezeichnet und mit GET-Befehlen in Strings abgelegt werden. Durch PUT können diese Icons wieder sichtbar gemacht werden und so z.B. als Menü dienen. Auch Windows kann man mit unserem "Software-Blitter" verwalten. Bevor ein Window Teile des Bildes überschreibt, wird der Ausschnitt durch GET in einem String zwischengespeichert. Später läßt sich dieser per PUT an die alte Stelle zurückschreiben.

Zur Anwendung

Zunächst sollten Sie sich das Listing 1 näher ansehen. Es enthält im ersten Teil mehrere Demos mit Beispielen zu GET und PUT. Der zweite Teil (ab Zeile 3000) enthält das Maschinenprogramm des "Software-Blitter" als Data-Zeilen. Sollten Sie GET und PUT in eigenen Programmen einsetzen, brauchen Sie nur diesen Teil zu übernehmen, der das Maschinenprogramm in Page 6 ablegt. Die neuen Befehle können dann über USR angesprochen werden. Sinnvoll ist es auch, die Einsprungsadressen des in Page 6 liegenden Programms wie im Beispiel als Variablen GGET=1536 (für Grafik-GET) und GPUT=1539 zu definieren. Ein GET-Befehl wird dann so ausgeführt:

```
A=USR (GGET, X1, Y1, X2, Y2, ADR (A5))
```

Die Koordinaten X1, Y1 geben die linke obere Ecke des Bildausschnitts an, X2 und Y2 die rechte untere. Als letzter Parameter muß die Adresse eines Strings eingesetzt werden (natürlich kann man statt A5 beliebige Namen einsetzen). Achten Sie nur darauf, daß in der DIM-Anweisung des Strings ausreichend Speicherplatz reserviert wurde. Den Platzbedarf können Sie wie folgt berechnen:

$$\text{Bytes} = (X2 - X1 + 1) * (Y2 - Y1 + 1) + 2$$

Der PUT-Befehl wird ganz ähnlich aufgerufen (GPUT=1539):

```
A=USR (GPUT, X, Y, MOD, ADR (A5))
```

Bei PUT müssen Sie nur die linke obere Ecke angeben, da Höhe und Breite des Ausschnitts im String bereits enthalten sind. Der Parameter MOD bestimmt, wie PUT den Ausschnitt einsetzt. Dies geschieht durch folgende Codes:

O: OR-Modus. Der Ausschnitt wird dem bereits vorhandenen Bild durch eine Oder-Verknüpfung überlagert, z.B. ein Bob mit dem Hintergrund.

64: EXOR-Modus. Der Ausschnitt wird Exklusiv-Oder mit dem vorhandenen Bild verknüpft. Schreibt man den Aus-

schnitt ein zweites Mal an die gleiche Stelle, so verschwindet er wieder. Bobs können damit zerstörungsfrei über den Hintergrund bewegt werden.

128: PSET-Modus. Hier wird der Bildteil vom Ausschnitt einfach überschrieben. Diese Methode eignet sich zum Zurückkopieren von Hintergrundinformationen (s. Beispiel).

Noch ein Wort zu den Koordinaten. Der "Software-Blitter" ist auf GRAPHICS 8 zugeschnitten. Y kann daher Werte von 0 bis 191 annehmen, während für X anstatt 0 bis 319 nur der Bereich von 0 bis 39 zulässig ist. Diese Einschränkung ergibt sich aus dem Aufbau des Video-Speichers bei hochauflösender Grafik. Je acht Bildpunkte sind in einem Byte zusammengefaßt, und dies ist die kleinste Einheit, die unser "Blitter" verarbeiten kann. Die gewohnten Koordinaten von 0 bis 319 müssen daher für GET und PUT jeweils durch acht geteilt werden.

Man könnte den "Software-Blitter" natürlich auch so auslegen, daß er die Byte-Grenzen ignoriert und alle 320 möglichen X-Koordinaten zuläßt. Tatsächlich war die erste Version des Maschinenprogramms so ausgelegt, aber das Ergebnis war fatal. Das Programm erreichte nur noch ein Schnecken tempo, da nun zusätzlich die Bit-Positionen innerhalb der Bytes verschoben und eine Menge von Randbedingungen beachtet werden mußten. Es war für jegliche Bewegungseffekte viel zu langsam.

Hier hat nun ein Hardware-Blitter die Nase weit vorne, da er über sogenannte Barrel-Shifters verfügt. Stellen Sie sich das zu bitte folgendes vor: Wenn ein Byte um vier Bitstellen verschoben werden soll, muß der Prozessor vier Schiebepfeile (z.B. ROL) nacheinander ausführen. Ein Barrel-Shifter liefert das gleiche Ergebnis in einem einzigen Arbeitstakt, ein Zeitsparende, in der der Prozessor nicht einmal einen einzigen Befehl ausführen könnte.

Trotz dieser Einschränkung ist der "Soft-Blitter" sehr nützlich. Wenn Sie Listing 1 eingetippt und die Demos angeschaut haben, werden Ihnen be-

stimmt eine Menge neuer Einsatzgebiete einfallen.

Bewegung

Im Demo wird auch Bewegung mit dem "Soft-Blitter" erzeugt. Das Ergebnis ist zwar nicht ganz so rack- und flimmerfrei, wie man es sich wünschen würde, aber immerhin wird ein 40 mal 30 Punkte großes Objekt (ein Flugzeug) zerstückelfrei über den Hintergrund bewegt - und das in Basic!

Zu Beginn jedes Bewegungsschritts wird ein Ausschnitt des Hintergrunds in den String HG5 gerettet. Anschließend wird das Flugzeug aus GR5 mit einem PUT-Befehl (im Oder-Modus) dem Bild an dieser Stelle überlagert. Nun beginnt eine kleine Warteschleife, die dafür sorgt, daß das Flugzeug eine Weile sichtbar ist. Der in HG5 gespeicherte Hintergrund überschreibt dann mit einem erneuten PUT-Befehl (diesmal im PSET-Modus) das Bild mit dem zuvor geretteten Hintergrund. Damit ist der ursprüngliche Zustand des Bildes wiederhergestellt. Im nächsten Schleifenlauf kann das Flugzeug ein Stück entfernt neu gezeichnet werden. Da die Zeitspanne vom Löschen bis zum erneuten Erscheinen dank dem "Soft-Blitter" sehr kurz ist, entsteht der Eindruck einer Bewegung.

Wesentlich einfacher wäre die Bewegung mit dem EXOR-Modus zu erzielen, da hier der Buffer für den Hintergrund entfällt. Allerdings wird bei dieser Methode das Objekt ziemlich unansehnlich, wenn der Hintergrund dicht geschrieben ist. Probieren Sie es aber ruhig mal aus!

Auch das lästige Flackern des Objekts könnte man beseitigen. Man müßte nur einen zweiten GRAPHICS-8-Bildschirm anlegen und dann per Page-Flipping zwischen beiden hin- und herschalten. Während der "Blitter" in einem Bild arbeitet, wird das andere angezeigt, und sobald ein neues fertiggestellt ist, werden die Rollen getauscht. Diese Technik ist sehr wirkungsvoll und wäre vielleicht ein Thema für eine zukünftige Assemblerlecke.

Blitter intern

Wie immer werfen wir auch diesmal einen Blick hinter die Kulissen des Maschinoprogramms (Listing 2). Nach einigen Definitionen von Variablen, die zur Beschleunigung alle in die Zero-Page gelegt wurden, findet sich eine kleine Sprungtabelle zu den Routinen GET und PUT.

Die GET-Routine beginnt ihre Arbeit, indem sie alle Parameter vom Stack nimmt und in die zugehörigen Variablen einträgt. Danach wird die Größe des Ausschnitts berechnet und gleich geprüft, ob die Koordinaten richtig eingegeben wurden, d. h. der linke Rand nicht größer als der rechte ist. Die berechneten Größen werden in den String als erstes und zweites Byte eingetragen, damit die PUT-Routine später weiß, wie groß der Ausschnitt ist.

Im Unterprogramm BERZEIG (berechne Zeiger) wird die Anfangsadresse der Zeile im Video-Speicher berechnet, wo der Ausschnitt beginnt. Mit Hilfe dieses Zeigers kann die folgende Schleife (ab Label GETZEIL) Byte für Byte aus dem Video-Speicher entnehmen und per Unterprogramm PUTSTR in den gewünschten String eingetragen werden. Entscheidend ist, daß man die Bytes aus dem Video-Speicher nicht kontinuierlich entnehmen darf. Sehen Sie sich dazu Bild 1 an, das einen Teil des Video-Speichers darstellen soll. Beginnend am linken Rand des gewählten Ausschnitts müssen so viele Bytes entnommen werden, bis der rechte Rand angetroffen wird. Nun muß ein Sprung zum linken Rand der nächsten Zeile erfolgen. Die Größe der Lücke hängt somit von der Breite des Ausschnitts und der Anzahl der Bytes pro Zeile ab. Man nennt diese Lücke gewöhnlich das Modulo.

Diese Situation wäre eine echte Anwendung für einen Hardware-Blitter. Ein solcher Chip hat Register für die Breite des Ausschnitts und das Modulo, und zwar sowohl für die Quelle der Daten (in unserem Fall das Video-RAM) als auch für das Ziel (hier der Basic-String). Da die Daten im String kontinuierlich abgelegt wer-

den, wäre im vorliegenden Fall das Modulo eben null. Sie sehen, der Name "Soft-Blitter" ist nicht aus dem Ärmel geschüttelt; die vorliegende Routine imitiert tatsächlich die Funktion eines Hardware-Blitters, wenn auch in stark vereinfachter Form.

In ihrem Aufbau entspricht die PUT-Routine der gerade besprochenen GET-Funktion, nur ist die Richtung des Datenflusses umgekehrt. Die Quelle (String) besitzt nun einen Modulo von null, das Ziel (Video-RAM) einen Modulo von Zeilenlänge minus Breite des Ausschnitts. Die Modulo-Werte sind in beiden Routinen fest einprogrammiert und können daher nicht frei gewählt werden. Durch diese Maßnahme wird das Programm kürzer und schneller.

Computer-Kontakt das Heft mit den preisgünstigen Kleinanzeigen

Interessant ist vielleicht noch die Abfrage des Modus durch den BIT-Befehl. Dies ist eine kritische Stelle im Programm. Je länger die Abfrage dauert, desto langsamer wird das Programm, da sie für jedes bearbeitete Byte erfolgen muß. Ein Befehl mehr oder weniger kann sich dabei ganz hübsch auswirken. Die beste Lösung wäre sicherlich, für jeden Modus eine eigene "Blitter"-Routine zu schreiben. Nur leider wird dann das Programm für Page 6 viel zu lang.

Ein wenig Aufmerksamkeit

verdient auch die Berechnung der Zeilenadresse im Unterprogramm BERZEIG. Es handelt sich nicht um eine allgemeine Routine zur Multiplikation. Vielmehr werden hier trickreich Bit-Verschiebungen sowie Additionen eingesetzt, um eine Multiplikation mit 40 zu erreichen. Aufmerksame Leser der Assemblerlecke kennen dieses Prinzip bereits aus früheren Folgen. Warum nun gerade mal 40? Ganz einfach, das ist die Anzahl der Bytes pro Zeile in GRAPHICS 8.

Andere Grafikstufen

Obwohl der "Soft-Blitter" nur für den Einsatz mit GRAPHICS 8 konzipiert ist, kann man ihn auch mit einigen anderen Grafikstufen verwenden. Denkbar wären z. B. GRAPHICS 15 oder auch GRAPHICS 0. Im letzteren Fall könnte man recht effektiv mit Windows arbeiten, denn dem "Blitter" ist es egal, ob die bearbeiteten Daten aus einem Text- oder HiRes-Bildschirm stammen. Aufpassen muß man jedoch mit den Y-Koordinaten, die dann nur noch von 0 bis 23 zulässig sind (es erfolgt keine Prüfung!); außerdem finden die Modi OR bzw. EXOR des PUT-Befehls keine vernünftige Verwendung mehr. Ähnliche Einschränkungen ergeben sich mit den Farbgrafik-Stufen wie 7 oder 15. Da nun zwei nebeneinander liegende Bits die Farbinformation bestimmen, kommt es beim OR- bzw. EXOR-Modus zur Verfälscherung der Farben. GET und PUT im PSET-Modus bleiben jedoch durchaus brauchbar. Es lohnt sich bestimmt, damit zu experimentieren.

Peter Finat

Aufbau des Video-RAMs

000	001	002	003	004	005			039
040	041	042	043	044	045			079
080	081	082	083	084	085			119
120	121	122	123	124	125			159
160	161	162	163	164	165			199
200	201	202	203	204	205			239

Ablesen im String

4	2	042	043	082	083	122	123	162	163
↑	↑	Byte							

MC-Generator

```

100 REM *****
110 REM BLITTER-OBJECTS-(BOBS)-DEMO
120 REM
130 REM P. FINZEL 1986
140 REM *****
150 REM
200 DIM GR$(300),HG$(300)
210 BBET=1536: BPUT=1539
220 GRAPHICS 2+16
230 POSITION 2,4:PRINT #6;"SOFT-BLITTE
R DEMO"
240 POSITION 1,6:PRINT #6;"EINEN MOMEN
T BITTE"
250 GOSUB 30000
300 REM * Zeichne Flugzeug
305 GRAPHICS B:COLOR 1
310 PLOT 55,20:DRAWTO 63,0:POSITION 47
,12:GOSUB 700
320 DRAWTO 15,12:POSITION 0,20:GOSUB 7
00
330 PLOT 43,30:DRAWTO 38,20:PLOT 25,20
:POSITION 35,30:GOSUB 700
335 REM * Bild in String
340 X=USR(BGET,0,0,8,30,ADR(GR$))
390 REM * viele Flugzeuge...
400 GRAPHICS B+16:SETCOLOR 2,1,0
410 FOR X=0 TO 32 STEP 8
420 FOR Y=0 TO 140 STEP 35
430 A=USR(BPUT,X,Y,0,ADR(GR$))
440 NEXT Y:NEXT X
450 FOR Y=0 TO 140 STEP 35
460 FOR X=0 TO 32 STEP 8: SOUND 0,Y,10,
10
470 A=USR(BPUT,X,Y,64,ADR(GR$))
480 SOUND 0,0,0,0:FOR I=0 TO 7:GOSUB B
00:NEXT I
490 NEXT X:NEXT Y
500 REM * BEWEGUNG
510 M=69:GOSUB 900:BOUND 0,15,0,10
515 Y=100:FOR X=32 TO 0 STEP -1:Y=Y-2
520 A=USR(BGET,X,Y,X+7,Y+30,ADR(HG$))
530 A=USR(BPUT,X,Y,0,ADR(GR$))
540 FOR I=0 TO 4:GOSUB 800:NEXT I
560 A=USR(BPUT,X,Y,128,ADR(HG$))
570 NEXT X: SOUND 0,0,0,0
590 GOTO 400
700 REM * XI0-FILLROUTINE
710 POKE 765,1:XI0 18,#6,0,0,"B:"
790 RETURN
800 REM * 1/50 SEC. WARTEN
810 T=PEEK(20)
820 IF PEEK(20)=T THEN B20
830 RETURN
900 REM * BILD AUFBAUEN
910 GRAPHICS B:SETCOLOR 2,8,0
920 FOR I=0 TO 319 STEP 31:PLOT 159,M:
DRAWTO 1,M+90:NEXT I

```

```

930 I=0
940 G=I+I+M:IF G<=159 THEN PLOT 0,G:DR
AWTO 319,G:I=I+1:GOTO 940
990 RETURN
30000 REM * Soft-Blitter einrichten
30010 B=0:RESTORE 30100
30020 FOR A=1536 TO 1785:READ D:POKE A
,D:B=B+D:NEXT A
30030 IF S<>34981 THEN ? "DATEN-FEHLER
!":STOP
30090 RETURN
30100 DATA 76,6,6,76,98,6,216,104,104,
104,133,218,104,104,133,217
30110 DATA 104,104,133,220,104,104,133
,219,104,133,215,104,133,214
30120 DATA 56,165,220,229,218,176,1,96
,133,221,230,221,56,165,219
30130 DATA 229,217,176,1,96,24,105,1,1
33,219,32,239,6,165,221,32,239
30140 DATA 6,32,175,6,164,218,165,221,
133,216,177,212,32,239,6,200
30150 DATA 198,216,208,246,24,165,212,
105,40,133,212,144,2,230,213
30160 DATA 198,219,208,225,96,216,104,
104,104,133,218,104,104,133
30170 DATA 217,104,104,133,222,104,133
,215,104,133,214,32,228,6,133
30180 DATA 219,32,228,6,133,221,32,175
,6,165,221,133,216,164,218,32
30190 DATA 228,6,36,222,48,8,112,4,17,
212,80,2,81,212,145,212,200
30200 DATA 198,216,208,234,24,165,212,
105,40,133,212,144,2,230,213
30210 DATA 198,219,208,213,96,165,217,
133,212,169,0,133,213,6,212
30220 DATA 38,213,6,212,38,213,24,165,
212,101,217,133,212,144,2,230
30230 DATA 213,6,212,38,213,6,212,38,2
13,6,212,38,213,24,165,212,101
30240 DATA 88,133,212,165,213,101,89,1
33,213,96,162,0,161,214,230
30250 DATA 214,208,2,230,215,96,162,0,
129,214,230,214,208,2,230,215,96

```

Assemblerlisting

```

*****
* SOFTWARE-BLITTER V1.02
*
* PUT/BET fuer Graphik
* (Byte-Auflösung)
*
* Peter Finzel 1986
*****
* LAENGE EQU 40 Zeilenlaenge in BYTE
*
* SAVHSC EQU %58 Zeiger auf Video-RAM
*
* Variable in der Zero-Page
*

```

```

ZEIGER EQU #D4 Pointer in Screen
STRING EQU #D6 Pointer in String
BIAEHL EQU #D8 Zaehler
GRAND EQU #D9 Koordinaten: oberer Rand
LRAND EQU #DA linker Rand
URAND EQU #DB unterer Rand
RRAND EQU #DC rechter Rand
BLAENGE EQU #DD Laenge fuer Blitter
MODUS EQU #DE Modus fuer PUT 0: Oder
* 64:Exor 128: Pset
*
* DRB #0600
*
* Sprungleiste
*
* JMP GET
* JMP PUT
*
* GET-Routine fuer Bildausschnitte
* Aufruf:
* A=USR(1536,X1,Y1,X2,Y2,ADR(A#))
*
*
* CLD
* PLA Anz. der Parameter
* PLA Linker Rand MSB
* PLA l. Rand LSB
* STA LRAND
* PLA Oberer Rand MSB
* PLA Oberer Rand LSB
* STA GRAND
* PLA rechter Rand MSB
* PLA und LSB
* STA RRAND
* PLA unterer Rand MSB
* PLA und LSB
* STA URAND
* PLA Adresse des Strings
* STA STRING+1
* PLA
* STA STRING
*
* SEC Anz. der horizontalen
* LDA RRAND Bytes ermitteln
* SBC LRAND
* BCS LENOK ueberhaupt moeglich?
* RTS Linker > Rechter!
*
*
* LENOK STA BLAENGE Laenge merken
* INC BLAENGE Laenge= R-L+1
*
* SEC Anzahl der Zeilen
* LDA URAND ermitteln
* SBC GRAND
* BCS HOENOK moeglich?
* RTS Oben > Unten!
*
*
* HOENOK CLC Zeilenanzahl
* ADC #1 korrigieren
* STA URAND und in URAND eintragen
* JSR PUTSTR Hoehe & Breite des
* LDA BLAENGE Ausschnittes in
* JSR PUTSTR String eintragen
*
*
* MBB1 JSR BEREIG Zeiger in Video-Ram
*
*
* Soft-Blitter fuer GET
*
*
* GETZEIL LDY LRAND linker Rand
* LDA BLAENGE Byte-Zaehler
* STA BIAEHL einrichten
*
* GETBYTE LDA (ZEIGER),Y Wert aus Video/RAM
* JSR PUTSTR in String ablegen
* INY naechstes Byte
* DEC BIAEHL schon ganze Zeile
* BNE GETBYTE nein ->
* CLC Zeiger auf naechste
* LDA ZEIGER Zeile in Video-RAM
* ADC #ZLAENGE richten (plus 40)
* STA ZEIGER
* BCC GET2 jetzt MSB
* INC ZEIGER+1
*
*
* GET2 DEC URAND schon alle Zeilen?
* BNE GETZEIL nein -->
* RTS
*
*
* PUT-ROUTINE fuer Bildausschnitte
* Aufruf:
* A=USR(1539,X,Y,Mod,Adr(A#))
*
*
* PUT CLD
* PLA Anz. der Argumente
* PLA linker Rand MSB
* PLA und LSB
* STA LRAND
* PLA Oberer Rand MSB
* PLA und LSB
* STA GRAND
* PLA Modus-Byte
* PLA nur LSB!
* STA MODUS
* PLA Stringadresse
* STA STRING+1
* PLA
* STA STRING
*
*
*
* JSR GETSTR Anzahl d. Zeilen
* STA URAND des Ausschnittes
* JSR GETSTR Hor. Laenge des
* STA BLAENGE Ausschnittes
*
*
* ERH12 JSR BEREIG Zeiger in Video-RAM
*
*
* Soft-Blitter fuer PUT
*
*
* PUTZEIL LDA BLAENGE
* STA BIAEHL Zaehler Zeilenlaenge
* LDY LRAND linker Rand
*
*
* PUTBYTE JSR GETSTR Zeichen aus String
* BIT MODUS welcher Modus?
* BNI PSET -> 'PSET'
* BVS EXOR -> Ex-Oder
* ORA (ZEIGER),Y 'Oder'
* BVC PSET verzwaigt immer
* EOR (ZEIGER),Y Exor-Verkn.
* STA (ZEIGER),Y in Video-RAM
* INY naechst. Spalte
* DEC BIAEHL Zeile fertig?
* BNE PUTBYTE nein -->

```

```

CLC                               nächste Zeile
LDA ZEIGER                         Zeiger=Zeiger+40
ADC #ZLAENGE
STA ZEIGER
BCC PUT2
INC ZEIGER+1  MSB
*
PUT2  DEC URAND                     alle Zeilen fertig?
      BNE PUTZEIL                   nein -->
      RTS
*
* Zeiger auf erstes Byte berechnen
*
      ZEIGER = ORAND * 40
*
BERZEIG LDA ORAND                   LSB Zeiger=ORAND
        STA ZEIGER
        LDA #0                       MSB Zeiger=0
        STA ZEIGER+1
        ASL ZEIGER                     Zeiger=Zeiger*2
        ROL ZEIGER+1
        ASL ZEIGER                     Zeiger=Zeiger*2
        ROL ZEIGER+1                   ergibt ORAND*4
        CLC
        LDA ZEIGER                     Zeiger=Zeiger+ORAND
        ADC ORAND                       bisher berechnet
        STA ZEIGER                     Zeiger=ORAND*5
        BCC SCRADR1
        INC ZEIGER+1
SCRADR1 ASL ZEIGER                   Zeiger=Zeiger*2
*
* Routinen zur String-Verwaltung
*
GETSTR LDI #0                       Ein Byte aus
        LDA (STRING,I)                 dem String holen
        INC STRING                      und Stringzeiger
        BNE GS1                          weiterschalten
        INC STRING+1
GS1    RTS
*
PUTSTR LDX #0                       Ein Zeichen in
        STA (STRING,X)                 String eintragen
        INC STRING                      und Stringzeiger
        BNE PS1                          weiterzählen
        INC STRING+1
PS1    RTS

```

Wombel auf dem Eis

"Wombel", ein Spiel für Atari 800 XL, ist mehr als nur ein Reaktions- und Geschicklichkeitsspiel. Es verlangt durchdachtes Handeln, ohne daß Action und Spielwitz zu kurz kommen. Das Besondere an "Wombel" ist nämlich, daß Steine und blaue Eisblöcke von allein in ein anliegendes Feld hüpfen, was für den kleinen, weißen Pinguin Wombel zwar nicht sehr gefährlich ist, aber seine ganzen Pläne über den Haufen werfen kann. So bleibt "Wombel" immer spannend und spielerisch wert.

Wichtigstes Ziel ist es, drei blinkende Eisblöcke in eine Reihe zu schieben. Dazu muß man sich erst einmal eine geeignete Stelle des Spielfelds aussuchen, an die alte drei Blöcke geschoben werden können, ohne daß ein Hindernis (Stein, Schneewesen) dazwischen ist. Das ist gar nicht so einfach, meistens springt auch noch ein Stein oder Eisblock in den Weg, und man muß umdenken. Besonders schlimm ist es, wenn

ein Stein so hüpf, daß ein blinkender Eisblock nicht mehr verschoben werden kann. Was nun? Hierfür gibt es pro Runde eine blinkende Krafttablette, die für solche Fälle aufgehoben werden sollte. Man kann mit ihr aber natürlich auch einen Stein beseitigen, der im Weg liegt. Nimmt Wombel nämlich diese Tablette, so hat der Spieler kurze Zeit die Möglichkeit, Steine aufzulösen. Dazu fährt er auf einen Stein zu, bleibt direkt vor ihm stehen und drückt den Feuerknopf am Joystick. Berühren darf man aber während dieser Zeit die Steine nicht.

In großer Zahl gibt es auf Wombels Spielfeld blaue Eisblöcke, mit denen sich eine Art Wand bilden läßt, an die man die blinkenden Eisblöcke schiebt. Natürlich kann man die drei blinkenden Blöcke auch an der Spielfeldumrandung aufreihen, doch der Weg dorthin ist leider meistens versperrt. Außerdem läßt sich ein Eisblock, der am Spielfeldrand lehnt, in

die Gegenrichtung nicht mehr verschieben. Die Gefahr ist also groß, daß er eingeschlossen wird.

Zurück zu den zahlreichen blauen Eisblöcken. Wombel muß sie sehr oft aus dem Weg räumen oder gar auflösen, wenn einer nicht mehr verschiebbar ist. Außerdem kann man mit allen Eisblöcken Schneewesen überfahren, was aber nur von links oder rechts geht, damit es nicht zu einfach wird. Passen Sie aber immer auf, wo sich die zwei Eisstrahlwesen befinden. Sie sind immer übereinander, eines am oberen, eines am unteren Spielfeldrand. Sie tauchen an irgendeiner Position auf und bilden nach kurzer Zeit zwischen sich einen senkrechten Eisstrahl, den Wombel nicht berühren darf. Tauchen die Eisstrahlwesen also plötzlich direkt über und unter Wombel auf, sollte man sofort nach links oder rechts ausweichen, sonst verliert Wombel eines seiner drei Bildschirmeleben. Von Runde zu Runde wird die Reaktionszeit kürzer, genauso wie von Runde zu Runde Steine und Eisblöcke öfter springen und immer mehr

Schneewesen ihr Unwesen treiben. Fangen Sie ruhig einmal mit Runde 9 an (mit SELECT zu Beginn wählbar!), dann bekommen Sie gleich 9 Schneewesen auf einmal zu sehen. In Runde 10 sind es dann 10 Schneewesen gleichzeitig, das heißt, in jeder Zeile des Spielfelds eines! Sind Sie gut genug, können Sie sich am Ende des Spiels in die Highscore-Tabelle eintragen, die zu Beginn jedes Spiels wieder gezeigt wird.

Airg Löffler

ATARI 1029 ★ HARDCOPY ★

- Ausdruck in vier Größen: Von der Postkarte bis zur vollen DIN-A4-Seite
- Ausdruck invertiert
- Ausdruck gespeichert
- Ausdruck evtl. und gep. Beschriftung
- Ausschnittvergrößerung
- Ausdruck und Speichern des Ausschnitts

Demos in Originalgröße kostenlos.
Diskette XL/XE 64 KByte

40,- DM inkl. Versand-
Beschriftung

Schwarz/Schwarz oder Nachnahme (C-E-DM)

Jürgen Dörr
Einsteinstraße 6
6520 Worms 26
Tel. 0 62 41 / 3 41 40

```

0 REM
1 REM
2 REM
3 REM
4 REM
5 REM
6 REM
7 REM
8 GOSUB 12000
9 GOSUB 7000:1=0
10 REM
11 REM
12 REM
13 REM
14 REM
15 REM
16 P1=PEEK(53252):P2=PEEK(53253)
17 NP=PEEK(283):VP=PEEK(284):SOUND 1,1
18 .L
19 IF P1=1 OR P2=1 THEN POKE 1728,156:
CL=I7:FA=236:GOTO 200:PEEK(289)=100
20 IF P1=0 OR P2=0 THEN POKE 1728,16:C
L=I35:FA=120:GOTO 200:PEEK(289)=100
21 IF P1=2 OR P2=2 OR P1=4 OR P2=4 OR
PEEK(53263)=1 THEN 2000
22 IF RND(1)>.5 THEN GOSUB 250
23 IF ZL=1 THEN CL=ZL-9:GOTO 100
24 COLOR 228:PLOT X5,2:PLOT X5,13:R5=I
R1(RND(1)*10)+1:COLR 171:PLOT X5,2:P
LOT X5,13:ZL=UZ
25 POKE 87,L:POSITION 28,17: N6:50=PEE
K(L3):" ":POKE 87,1:IF PEEK(L3)=50 TH
EN 2000
26 IF RND(1) THEN ER=KR-1:GOSUB 800:IF K
R<L THEN POKE 718,124:POKE 1761,164
27 SOUND 2,L,1,L,1:POKE 53276,L:NR=L:POK
E 77,L:GOTO 70
28 REM
29 REM
30 IF ZL=1 THEN 50
31 XE=INT(5*NR+53):POKE 53282,XE:50UR
D 1,50,12,8
32 IF XE>PEEK(283) AND XE(PEEK(283))+8
THEN 2000
33 POKE 5375,L:GOTO 80
34 REM
35 REM
36 NR=INT(RND(1)*10)+1:YB=INT(RND(1)*
10)+1
37 LOCATE X8,Y8,CX:IF CX<135 AND CX
>2 THEN RETURN
38 XE=INT(RND(1)*3)-1:YY=INT(RND(1)*3
)-1
39 LOCATE X8+X8,Y8+YY,C1IF C<132 AND
C<135 THEN RETURN
40 SOUND 2,121,12,8,COLOR 32:PLOT X8,
Y8
41 X8=X8+X8:Y8=Y8+YY:COLOR 162:PLOT X
8,Y8:SOUND 2,255,12,8:FOR I=L TO 2
42 IF PEEK(53282)=4 OR PEEK(53253)=4
THEN 2000
43 NEXT I
44 COLOR CX:PLOT X8,Y8:SOUND 2,L,L,L,1:
RETURN
45 REM
46 REM
47 REM
48 X=(KP-48)/8:Y=VP/16-1:NM=X*8+48:Y
M=16+1
49 IF CL=17 THEN LOCATE X,Y,CX:IF CX=
64 THEN 970
50 LOCATE X,Y+1,CX:IF CX<132 THEN 710
51 COLOR CL:PLOT X,Y:POKE 287,YM:POKE 286,X
M:COLOR I2:PLOT X,Y:50=I2:SM=4:POKE 53
278,L
52 SC=5C+10:FOR I=X TO X+7 STEP -1:
50=50-SM:SOUND 2,50,10,10:POKE 287,I
53 IF PEEK(53263)=4 THEN IF NR=L THEN
NR=PEEK(28912+Y-3):GOSUB 900
54 NEXT I:SM=L:X=X-1:NM=X*8:TRAP 450
:LOCATE X-1,Y,CX:IF CX=32 THEN 430
55 COLOR CL:PLOT X,Y:POKE 286,L:POKE
287,L:FOR I=15 TO 3 STEP -0.4:SOUND 2,
280,12,I:NEXT I:SOUND 2,L,L,L
56 POKE 786,L:GOSUB 790:IF CL<17 THE
N 90
57 TRAP 90:LOCATE X,Y-1,CX:IF CX=37 I
NEN LOCATE X,Y-2,CX:IF CX=37 THEN 4700
58 LOCATE X+1,V,CX:IF CX=37 THEN GOSU
B 790
59 LOCATE X-1,V,CX:IF CX=37 THEN GOSU
B 780
60 GOTO 90
61 REM
62 REM
63 X=(KP-48)/8:Y=VP/16-1:NM=X*8+48:Y
M=16+1
64 IF CL=17 THEN LOCATE X,Y,CX:IF CX=
64 THEN 970
65 LOCATE X,Y+1,CX:IF CX<132 THEN 710
66 COLOR CL:PLOT X,Y:POKE 287,YM:POKE 286,X
M:COLOR I2:PLOT X,Y:50=I2:SM=4:POKE 53
278,L
67 SC=5C+10:FOR I=X TO X+14 STEP 2:
50=50-SM:SOUND 2,50,10,10:POKE 287,I
68 P3=PEEK(53263):IF P3=I OR P3=7 THE
N 2000
69 NEXT I:SM=L:Y=Y+1:YH=YM+16:LOCATE
X,Y+1,CX:IF CX=32 THEN 530
70 COLOR CL:PLOT X,Y:POKE 286,L:POKE
287,L:FOR I=15 TO 3 STEP -0.4:SOUND 2,
280,12,I:NEXT I:SOUND 2,L,L,L
71 POKE 786,L:GOSUB 790:IF CL<17 THE
M 90
72 TRAP 90:LOCATE X,Y+1,CX:IF CX=37 I
NEN LOCATE X,Y+2,CX:IF CX=37 THEN 4700
73 LOCATE X+1,V,CX:IF CX=37 THEN GOSU
B 790
74 LOCATE X-1,V,CX:IF CX=37 THEN GOSU
B 780
75 GOTO 90
76 REM
77 REM
78 X=(KP-48)/8:Y=VP/16-1:NM=X*8+48:
YH=YM+16
79 IF CL=17 THEN 2000
80 FOR I=15 TO 3 STEP -1:FOR II=107 T
O 190
81 IF PEEK(53263)=3 THEN 2000
82 SOUND 2,II,12,I:COLOR II:PLOT X,Y:
NEXT II:NEXT I:SOUND 2,L,L,L

```

```

485 IF CL=17 THEN LOCATE X,Y,CX:IF CX=
64 THEN 970
486 TRAP 700:LOCATE X-1,Y,CX:IF CX<132
THEN 710
487 POKE 786,FA:POKE 287,YM:POKE 286,X
M:COLOR I2:PLOT X,Y:50=I2:SM=3:POKE 53
278,L
488 SC=5C+10:FOR I=X TO X+7 STEP -1:
50=50-SM:SOUND 2,50,10,10:POKE 286,I
489 IF PEEK(53263)=4 THEN IF NR=L THEN
NR=PEEK(28912+Y-3):GOSUB 900
490 NEXT I:SM=L:X=X-1:NM=X*8:TRAP 450
:LOCATE X-1,Y,CX:IF CX=32 THEN 430
491 COLOR CL:PLOT X,Y:POKE 286,L:POKE
287,L:FOR I=15 TO 3 STEP -0.4:SOUND 2,
280,12,I:NEXT I:SOUND 2,L,L,L
492 POKE 786,L:GOSUB 790:IF CL<17 THE
N 90
493 TRAP 90:LOCATE X-1,Y,CX:IF CX=37 I
NEN LOCATE X-2,Y,CX:IF CX=37 THEN 4700
494 LOCATE X,Y+1,CX:IF CX=37 THEN GOSU
B 790
495 LOCATE X,Y-1,CX:IF CX=37 THEN GOSU
B 780
496 GOTO 90
497 REM
498 REM
499 X=(KP-48)/8:Y=VP/16-1:NM=X*8+48:Y
M=16+1
500 IF CL=17 THEN LOCATE X,Y,CX:IF CX=
64 THEN 970
501 LOCATE X,Y-1,CX:IF CX<132 THEN 710
502 POKE 786,FA:POKE 287,YM:POKE 286,X
M:COLOR I2:PLOT X,Y:50=I2:SM=3:POKE
53278,L
503 SC=5C+10:FOR I=X TO X+14 STEP -2:
150=50-SM:SOUND 2,50,10,10:POKE 287,I
504 P3=PEEK(53263):IF P3=I OR P3=7 THE
N 2000
505 NEXT I:SM=L:Y=Y-1:YH=YM-16:LOCATE
X,Y-1,CX:IF CX=32 THEN 530
506 COLOR CL:PLOT X,Y:POKE 286,L:POKE
287,L:FOR I=15 TO 3 STEP -0.4:SOUND 2,
280,12,I:NEXT I:SOUND 2,L,L,L
507 POKE 786,L:GOSUB 790:IF CL<17 THE
N 90
508 TRAP 90:LOCATE X,Y-1,CX:IF CX=37 I
NEN LOCATE X,Y-2,CX:IF CX=37 THEN 4700
509 LOCATE X+1,V,CX:IF CX=37 THEN GOSU
B 790
510 LOCATE X-1,V,CX:IF CX=37 THEN GOSU
B 780
511 GOTO 90
512 REM
513 REM
514 X=(KP-48)/8:Y=VP/16-1:NM=X*8+48:Y
M=16+1
515 IF CL=17 THEN LOCATE X,Y,CX:IF CX=
64 THEN 970
516 LOCATE X,Y+1,CX:IF CX<132 THEN 710
517 COLOR CL:PLOT X,Y:POKE 287,YM:POKE 286,X
M:COLOR I2:PLOT X,Y:50=I2:SM=4:POKE 53
278,L
518 SC=5C+10:FOR I=X TO X+14 STEP 2:
50=50-SM:SOUND 2,50,10,10:POKE 287,I
519 P3=PEEK(53263):IF P3=I OR P3=7 THE
N 2000
520 NEXT I:SM=L:Y=Y+1:YH=YM+16:LOCATE
X,Y+1,CX:IF CX=32 THEN 530
521 COLOR CL:PLOT X,Y:POKE 286,L:POKE
287,L:FOR I=15 TO 3 STEP -0.4:SOUND 2,
280,12,I:NEXT I:SOUND 2,L,L,L
522 POKE 786,L:GOSUB 790:IF CL<17 THE
M 90
523 TRAP 90:LOCATE X,Y+1,CX:IF CX=37 I
NEN LOCATE X,Y+2,CX:IF CX=37 THEN 4700
524 LOCATE X+1,V,CX:IF CX=37 THEN GOSU
B 790
525 LOCATE X-1,V,CX:IF CX=37 THEN GOSU
B 780
526 GOTO 90
527 REM
528 REM
529 X=(KP-48)/8:Y=VP/16-1:NM=X*8+48:
YH=YM+16
530 IF CL=17 THEN 2000
531 FOR I=15 TO 3 STEP -1:FOR II=107 T
O 190
532 IF PEEK(53263)=3 THEN 2000
533 SOUND 2,II,12,I:COLOR II:PLOT X,Y:
NEXT II:NEXT I:SOUND 2,L,L,L

```

```

740 COLOR 32:PLOT X,Y:GOTO 90
750 LOCATE X,Y:2,CH:IF CX=37 THEN POP
:GOTO 4700
755 LOCATE X,Y-1,CH:IF CX=37 THEN POP
:GOTO 4700
757 RETURN
760 LOCATE X,Y-2,CH:IF CX=37 THEN POP
:GOTO 4700
765 LOCATE X,Y+1,CH:IF CX=37 THEN POP
:GOTO 4700
767 RETURN
770 LOCATE X+2,Y,CH:IF CX=37 THEN POP
:GOTO 4700
775 LOCATE X-1,Y,CH:IF CX=37 THEN POP
:GOTO 4700
777 RETURN
780 LOCATE X-2,Y,CH:IF CX=37 THEN POP
:GOTO 4700
785 LOCATE X+1,Y,CH:IF CX=37 THEN POP
:GOTO 4700
790 POSITION 7,1:785:5:RETURN
795 REM WENN MAN DEN SCHWIMMSTUHN
800 IF 5 STRIG(1) THEN RETURN
810 SOUND L,L,L:L:POKE 1546,16:FOR I=2
80 TO L STEP -2:50:GOTO J,I,10,10
815 NEXT I:50:ND J,L,L,L:KR=KR-5:IF KR
(L THEN POKE 710,124:POKE 1761,184
820 GOTO PEEK(209)+829
825 X=CMP-483/8+1:Y=VP/16-1:GOTO 840
831 X=CMP-483/8-1:Y=VP/16-1:GOTO 840
832 X=CMP-483/8:Y=VP/16-2:GOTO 840
833 X=CMP-483/8:Y=VP/16+8:5:GOTO 840
840 LOCATE X,Y,6:IF G(2) THEN POKE 154
6,208:RETURN
850 FOR I=30 TO L STEP -1:50:GOTO I,I*90
,10,15-1/2:POKE 53760,I:COLOR INT(RND(
L)*4)+27:PLOT X,Y:NEXT I
860 SOUND J,L,L,L:5C=5C+50:GOSUB 790
870 COLOR 32:PLOT X,Y:POKE 1546,208:RE
TURN
875 REM WENN MAN DEN SCHWIMMSTUHN
900 POKE 1546,18:50:ND J,L,L,L:50:ND J
,L,L,L:5C=5C+30:GOSUB 790
910 FOR TT=15 TO L STEP -1:FOR II=70 T
O 100 STEP 15
920 POKE 38912+Y-3,RR:50:ND 2,II*2,10,
11:NEXT II:POKE 38912+Y-3,TT:NEXT TT
930 POKE 38912+Y-3,L:POKE 1546,208:RET
URN
945 REM WENN MAN DEN SCHWIMMSTUHN
970 POKE 1761,165:POKE 39190,165:FOR I
=L TO 14:50:ND 2,II*14,10,14-1/2
980 FOR II=L TO 5:NEXT II:POKE 710,II*1
0:NEXT I:5C=5C+100:GOSUB 790
990 COLOR 32:PLOT XT,YI:HT:L:YT:L:POKE
710,236:50:ND 2,L,L,L:NR=NR+G(20)
1000 REM WENN MAN DEN SCHWIMMSTUHN
1010 RESTORE 10100:AI=128:AZ=136
1020 FOR I=32590 TO 32752 STEP 2:POKE
I,1:POKE I+1,2:NEXT I
1030 FOR I=35630 TO 35800 STEP 2:POKE
I,1:POKE I+1,2:NEXT I
1040 FOR I=39360 TO 39457:READ C:POKE
I,C:NEXT I:RESTORE 10700
1060 FOR I=39560 TO 39557:READ C:POKE
I,C:NEXT I
1070 CH=144:CC=CHR(256):RESTORE 10800
1080 FOR I=1530 TO 1568:READ C:POKE I,
C:NEXT I:U=USR(1536):RESTORE 10400
1090 READ A:IF A=-1 THEN RETURN
1100 FOR I=L TO 7:READ B:8:POKE CC+I+A*80
,B:NEXT I:GOTO 1090
1110 REM WENN MAN DEN SCHWIMMSTUHN
1200 AI=128:AZ=136
1210 FOR I=1849 TO 1999 STEP 16
1220 RESTORE 10350+INT(RND(L)*95)*10
1230 FOR II=L TO 15:READ C:POKE 700,II
*10:POKE AI*256+II+1,C:POKE AZ*256+II+
1,C
1240 NEXT II:NEXT I
1250 RESTORE 1340+R(2)*10
1260 FOR I=L TO 9:READ C:POKE 38912+I,
C*8+48:IF C=L THEN POKE 38912+I,1
1270 POKE 38933+I,INT(RND(L)*16)*16+10
:POKE 38953+I,L:POKE 38923+I,L
1300 POKE 38943+I,C*I*20:NEXT I
1310 RETURN
1350 DATA 0,0,0,0,0,0,0,0,0,0,0,0
1356 DATA 0,0,0,0,0,0,0,0,0,0,0,17
1370 DATA 0,0,0,0,0,0,0,0,0,0,0,11,0
1380 DATA 0,0,13,4,0,16,0,0,0,0,0,0
1370 DATA 5,0,12,0,0,0,0,0,0,13,6
1400 DATA 0,7,0,17,10,16,0,6,0,11
1410 DATA 0,5,9,16,10,8,12,0,11,17
1420 DATA 0,4,12,7,0,8,7,7,12,4
1430 DATA 10,10,8,18,4,3,4,10,9,3
1440 DATA 9,10,8,14,3,15,5,17,2,13
1450 DATA 14,3,9,10,15,14,8,7,6,14
1460 DATA 14,8,14,0,0,6,15,2,3,17
1470 DATA 2,6,15,5,9,17,2,5,11,15
1480 DATA 13,9,12,5,2,15,10,6,17,15
1490 DATA 6,12,11,0,10,11,12,17,12,4
1500 DATA 11,7,9,17,10,18,15,6,17,11
1510 DATA 17,5,9,16,10,17,12,7,11,17
1520 DATA 11,4,12,7,13,0,14,7,12,4
1530 DATA 10,15,15,16,4,3,4,16,9,11
2000 REM WENN MAN DEN SCHWIMMSTUHN
2010 SOUND 1,1,1,L:POKE 538,1:50:ND L,
L,L,L:POKE 53251,L
2020 FOR I=100 TO 170 STEP 2:50:ND 2,I
,10,10:POKE 54279,132
2030 SOUND 2,II*5,10,10:POKE 54279,13
6:NEXT I:50:ND 2,L,L,L,L
2040 FOR I=L TO 4:POKE 53240+I,L:NEXT
I:FOR I=L TO 200:NEXT I
2050 SOUND 2,L,L,L:LB=LB-1:IF LB=L THE
N 9000
2060 GRAPHICS 18:GOSUB 1250:ZL=4:R5=10
:KR=L:CR=120:GOSUB 7820:GOTO 20
4000 REM WENN MAN DEN SCHWIMMSTUHN
4005 POKE 538,1:FOR I=L TO 4:POKE 5324
0+I,L:NEXT I:POKE 54286,255
4020 POKE 02,L:GRAPHICS L:POKE 559,L:P
OKE 752,1:POKE 789,202:POKE 780,184:PO
KE 711,226:POKE 710,228:POKE 750,CH
4030 BL=PEEK(508)+256:PEEK(561):POKE D
L*6,7:POKE DL*8,6:POKE BL*22,8
4040 ? " ***** "
4050 ? CHR$(140);" w o r l d "
4060 ? " ***** "
4070 ? CHR$(140);" (C)BY J/RG L/FFLER"
4080 ? " ***** "
4090 ? " "
4100 FOR I=L TO 7:?" :?+1;" "
4110 ? HIS(II*2+1,II*2+12);" .Level:..
*****":POSITION 25,II*2+5:RA(II+1
)
4120 IF RA(II+1)RE(II+1) THEN ? " " ;RE
(II)
4130 POSITION (30-LEN$(STR$(II+1))),C
2HI+5:?" P(II+1)" ""
4140 ? " "
*****:NEXT I
4150 ? " ***** "
4160 POKE 53277,3:POKE 704,12:POKE 705
,40:POKE 54279,136
4200 FOR I=L TO 3:50:ND I,L,L,L:NEXT I
:POKE 559,62
4210 RESTORE 4400+INT(RND(L)*6)*10
4220 READ A,B,C:50:ND 1,6,10,10:50:ND
2,8,10,8:50:ND 3,C,10,8
4230 FOR I=1 TO 30:IF STRIG(1) THEN ME
XT I:GOTO 4210
4240 FOR I=170 TO 160 STEP -0.2:POKE 5
3761,I:POKE 53765,I:POKE 53767,I:NEXT
I
4270 IF RET THEN GOSUB 1250:RETURN
4280 H=39:V=15:GOSUB 4390:H=32:V=17:G
OSUB 4390:H=32:V=19:GOSUB 4390:H=39:V=2
1:GOSUB 4390
4290 RW=1:POSITION 2,19:?" *X Press STA
RT to begin a new game!":POKE 752,L
4300 POSITION 2,17:?" *PRESS SELECT
to choose 1st LEVEL":CHR$(160):CHR$(177)
:
4310 IF PEEK(53279)=5 THEN GOSUB 4300

```

```

4330 IF PEEK(63279)=6 THEN POKE 559,1:
GOTO 4350
4340 GOTO 4310
4350 R1=RN:R2=RN:RA=RN:VL=0.76-RN*0.01
:VK=60-RN*3:VZ=6-RN*0.25:GOSUB 1250
4370 RETURN
4380 RN=RN+1:IF RN=10 THEN RN=1
4385 POSITION 36,17:?" CHR$(RN*176):"PO
R I=250 TO 50 STEP -15: SOUND 2,1,10,10
: NEXT I: SOUND 2,1,1,1: RETURN
4390 POSITION 1,1: FOR I=1 TO 37: CHR$(
CX): SOUND 2,1,10,10: SOUND 2,1,1,1: NEXT
I: RETURN
4400 DATA 96,121,162
4410 DATA 120,91,81
4420 DATA 121,96,81
4430 DATA 144,121,96
4440 DATA 100,91,72
4450 DATA 181,144,121
4460 DATA 121,181,193
4470 DATA 120,91,81
4480 DATA 144,193,243
4490 DATA 121,144,102
4500 DATA 120,162,182
4700 REM RELEVANCE
4701 IT=(50-PEEK(193))*100:POKE 53252,L
4705 POKE 530,1: SOUND 1,1,1,1: FOR I=15
TO 1 STEP -1: FOR IT=L TO 4: SETCOLOR I
I, I, I: SOUND 2,1MS*IT,10,1: NEXT I
4710 NEXT I: FOR I=L TO 3: POKE 53248+I,
L: NEXT I: SOUND 2,1,1,1: GRAPHICS 17:POK
E 700,102
4720 POKE 709,122:POKE 710,1:POKE 559,
62:DL=PEEK(560)+256*PEEK(561)
4730 POKE DL+8,7:POKE DL+9,7:POKE DL+1
1,2:POKE DL+23,2
4740 POSITION 4,1:?" w o w b e i"
4750 POKE 750,204:POSITION 7,6:?" H6:"C
C 1906 by J: CHR$(12):"PC L": CHR$(12)
:?"FLER":?"INT( )":*100
4760 POSITION 4,11:?" LEVEL":?" RN
4770 POSITION 4,13:?" SCORE":?" SC
4780 POSITION 4,15:?" BONUS":?" IT
4790 SOUND 1,1,1: POSITION 6,19:?" H6:
RELEVANCE
4810 VCL=ZL*4*Y5=10:RCL=L*5C=5C+1:RN=
RN*1:R1=R1+R2=R2+1:IF R1=R1 THEN R1=
1
4820 IF R2=20 THEN R2=10
4830 VL=VL-0.01:IF VL(0.3 THEN VL=0.3
4840 VK=VK-3:IF VK(20 THEN VK=20
4850 VZ=VZ-0.25:IF VZ(2 THEN VZ=2
4870 RET=1:GOSUB 4280:GOSUB 7820:GOTO
20
5000 REM RELEVANCE
5010 GRAPHICS 18:POKE 559,1:POKE 710,1
24:POKE 709,20:POKE 711,120:POKE 752,1
:POKE 622,2
5020 DL=PEEK(560)+256*PEEK(561)-1:POKE
560,PEEK(560)-1:POKE DL+6,7
5030 POKE DL+11:POKE DL+3,66:POKE DL+
4,PEEK(561):POKE DL+5,PEEK(561):POKE 756
,CHI:POKE 87,1
5035 FOR I=DL+6 TO DL+15:POKE I,PEEK I
+1:2: NEXT I:POKE 512,80:POKE 513,152:
POKE 559,62:POKE 1761,164
5040 POSITION 1,1:?" H6:?" SCORE":?" SC:POS
ITION 14,1:?" LEVEL":?" RN
5050 POSITION 1,1:?" H6:?" TIME":?" POSIT
ION 11,1:?" H6:?" LIVES":?" LD:
5060 COLOR 219:PLOT 1,2:COLOR 220:PLOT
1,2:DRANTO 10,2:COLOR 221:PLOT 19,2:R
ESTORE 5000+R1*100
5070 COLOR 172:PLOT 19,1:DRANTO 19,12:
COLOR 222:PLOT 19,13:COLOR 220:PLOT 18
,13:DRANTO 1,13:COLOR 223:PLOT 1,13
5071 COLOR 172:PLOT 1,1:DRANTO 1,3:CO
LOR 171:PLOT 85,2:PLOT 85,13:COLOR 175
5075 READ X, Y: IF X=55 THEN POKE 54206,
192:GOSUB 4500:RETURN
5080 IF X=L THEN COLOR Y:GOTO 5075
5085 IF X=L THEN PLOT 85,CX, Y:GOTO 50
75
5090 DRANTO X, Y:GOTO 5075
5095 REM RELEVANCE
5100 DATA -12,12,16,12,16,6,12,6,-12
5110 DATA 12,14,18,1,12,6,12,-2,5,2
5120 DATA 10,5,10,5,8,10,0,10,11,-5

```

```

5130 DATA 1,5,7,7,7,7,4,15,4,15,1,-10
5140 DATA 4,10,0,0,2,-1,1,0,0,0,4,7
5150 DATA 7,-11,5,12,5,-1,11,-12,7,12
5160 DATA 8,-17,5,17,11,-11,12,9,12,0
5170 DATA 37,-2,11,-11,0,-16,4,39
5195 REM RELEVANCE
5200 DATA -1,3,4,3,4,4,6,4,6,1.6
5210 DATA 1.0,3,8,-2,12,5,12,5,10,9
5220 DATA 10,9,11,-14,12,14,10,17,10
5230 DATA -0,0,10,0,10,4,0,12,7,0
5240 DATA 3,10,7,0,2,-7,10,9,3
5250 DATA 12,11,12,-0,12,8,11,-4,9,4
5260 DATA 7,-3,5,3,4,-7,1,9,2,-16,3
5270 DATA 16,5,-12,5,12,9,0,27,-15,6
5280 DATA -16,11,-4,10,99
5295 REM RELEVANCE
5300 DATA -1,12,10,3,10,11,10,12,-4
5310 DATA 12,7,12,-15,12,9,12,9,9,12
5320 DATA 9,12,0,0,0,-0,3,9,3,-2,4,2
5330 DATA 0,4,0,-15,4,17,4,17,0,2
5340 DATA -4,1,5,3,-3,5,1,7,5,7,-5,11
5350 DATA 0,11,0,0,0,11,5,11,-13,5,14
5360 DATA 6,14,11,10,11,0,37,-17,4,-7
5370 DATA 9,-2,0,99
5395 REM RELEVANCE
5400 DATA -2,0,2,3,10,3,-4,5,5,5,-2,9
5410 DATA 4,9,4,11,2,11,2,10,-0,10,10
5420 DATA 10,-0,0,10,0,10,0,0,0,7
5430 DATA -9,12,-11,12,-13,12,-15,10
5440 DATA 10,10,10,0,-15,4,17,4,17,6
5450 DATA 15,0,15,5,0,2,-1,0,7,8,7,5
5460 DATA 10,5,-12,7,10,7,-17,12,10
5470 DATA 12,-1,12,7,12,0,37,-16,5,-9
5480 DATA 7,-3,10,99
5495 REM RELEVANCE
5500 DATA -4,5,4,10,4,10,10,-14,1,14
5510 DATA 5,10,5,10,3,-1,4,5,4,5,7,1
5520 DATA 7,1,5,-2,0,2,11,1,12,3,12,3
5530 DATA 12,7,8,-12,17,7,17,9,15,9
5540 DATA 15,11,10,11,-0,9,0,12,-10,9
5550 DATA 10,12,0,2,-1,3,3,3,-7,5,11
5560 DATA 3,12,3,-15,3,15,4,-2,6,-4,6
5570 DATA -0,0,0,4,0,3,9,3,10,-7,9
5580 DATA -7,11,-12,-11,12,10,12,-0
5590 DATA 8,-11,6,15,0,-12,5,17,6,-13
5600 DATA 0,15,0,0,37,-16,4,-6,6,-9
5630 DATA 11,99
5655 REM RELEVANCE
5660 DATA -1,4,5,4,5,9,7,9,7,6,12,6
5670 DATA -10,3,15,3,15,4,-10,10,15
5680 DATA 10,-13,10,11,11,11,11,0
5690 DATA 9,0,9,11,3,11,3,0,15,-3,12
5700 DATA 8,12,0,2,-10,12,-9,12,-9
5750 DATA 10,-9,12,-2,12,-1,0,1,10
5660 DATA -3,0,3,10,-5,3,-11,4,11,5
5670 DATA 10,0,-12,9,13,5,10,5,0,37
5680 DATA -12,11,-10,4,-2,9,99
5695 REM RELEVANCE
5700 DATA -1,12,5,12,5,10,3,10,1,0,8
5710 DATA 0,8,6,1,6,1,3,14,7,14,7,10
5720 DATA 11,10,12,-7,12,9,12,0,12
5730 DATA 4,-12,12,10,-16,4
5740 DATA 18,4,18,6,0,2,-12,4,13,0,14
5750 DATA 8,-10,4,7,4,7,5,4,-10,7
5760 DATA 10,7,-5,7,7,-2,11,2,9,-7
5770 DATA 9,8,3,-0,11,7,11,7,12,-16
5780 DATA 10,16,11,0,37,-2,4,-6,9,-16
5790 DATA 10,99
5795 REM RELEVANCE
5800 DATA -3,4,0,4,18,12,7,12,1,0,7
5810 DATA 4,11,0,8,11,-2,12,2,4,-17,4
5820 DATA 10,4,-17,12,10,-1,0,0,2
5830 DATA -5,3,9,3,-14,3,10,3,-10,7
5840 DATA 17,0,10,9,-13,0,15,0,-3,5,4
5850 DATA 5,4,6,-10,10,10,-14,5,14
5860 DATA 6,10,0,-1,9,1,12,-0,0,5,11
5870 DATA 0,37,-2,4,-17,6,-17,10,99
5895 REM RELEVANCE
5900 DATA -4,0,4,3,1,1,12,4,12,4,10
5910 DATA -6,4,6,12,-9,4,4,4,-11,4,11
5920 DATA 11,12,12,14,14,14,-16,3
5930 DATA 10,3,10,12,16,12,16,7,0,2
5940 DATA -7,10,-10,11,-12,12,-5,0,-7
5950 DATA 7,7,1,5,3,-13,4,13,7,12,7
5960 DATA 12,9,13,9,13,10,-17,7,-15
5970 DATA 10,15,12,-2,4,2,6,4,0,37
5980 DATA -7,11,-10,7,-17,11,99
5995 REM RELEVANCE

```

```

6000 DATA -3,6,2,4,0,4,0,5,6,11,5,11
6010 DATA 3,3,-11,4,1,1,1,16,11,-14,3
6020 DATA 8,1,18,0,14,4,0,2,-16,7,17
6030 DATA 0,-12,9,12,3,-12,18
6040 DATA -3,4,3,12,-4,0,7,6,-3,12,5
6050 DATA 12,-1,3,1,0,-15,12,16,12,0
6060 DATA 37,-7,7,-7,11,-18,11,39,0
6500 COLOR 160:PLOT 12,10
6510 XT=INT(60*(L+3)*0.5)+1:YT=INT(60*(L+3)*0.5)
6520 LOCATE YT,XT,CH:IF CH(3) THEN 65
6530 COLOR 64:PLOT XT,YT:COLOR 32:PLOT
12,10:RETURN
7000 REM *****
7010 RETL:CH=144:GOSUB 4000:5C=L:KR=L
7120 L=L-1:K5=10:POKE 1761,164
7200 GOSUB 5000:POKE 532,L:POKE 503,6
POKE L,L:POKE 1,15:POKE 54273,136:HP=
44:VP=167:POKE 700,L
7630 SOUND L,L,1,0,POKE 203,XP:POKE 2
40,VP:POKE 206,L:POKE 207,L:POKE 39190
1,16:POKE 39193,L:POKE 53277,L
7640 POKE 53278,L:POKE 530,1:POKE 17,
POKE 20,L:POKE 39190,1
7650 RETURN
9000 REM *****
9005 POKE 54286,255:TRAP 40000
9010 ? CHR$(123):GRAPHICS L:POKE 755,L
:POKE 710,L:POKE 780,122:POKE 711,202
9015 DL=PEEK(5040)+256*PEEK(5041)
9020 POKE DL+0,6:POKE DL+9,6:POSITION
0,3:IF 86:"WOMEL GAME WOMEL":POSITI
ON 29,3:IF 86:"OVER":?
9030 POSITION 13,7:IF 86:"YOUR SCORE":?
5C:IF P(7) THEN 5300
9040 POSITION 9,10:?"PLEASE ENTER YOU
R NAME":POKE 736,CH
9050 POKE DL+19,6:POKE DL+20,6:POKE DL
+21,7:POKE DL+22,6
9055 COLOR 251:PLOT 23,13:COLOR 252:PL
OT 24,13:DRANTO 25,13:COLOR 253:PLOT 1
6,13
9060 POSITION 3,14:?"CHR$(140):POSIT
ION 16,14:IF 86:CHR$(148)
9065 COLOR 255:PLOT 23,14:COLOR 252:PL
OT 24,14:DRANTO 25,14:COLOR 254:PLOT 1
6,14
9070 POSITION 6,18:?"Press RETURN for a
new game!"
9080 X=4:OPEN #2,4,L,"X":5P5=".....
....."
9090 SOUND 2,L,L,L:GET WZ,6
9100 IF C=155 THEN FOR I=200 TO L STEP
-10:SOUND 2,I,10,16:NEXT I:GOTO 9140
9110 IF C=125 AND M4 THEN SOUND 2,60,
10,6:K=M-1:POSITION X,14:IF 86:CHR$(123
):15P5=C-I,X-3:="":GOTO 9090
9120 IF C(32 OR 6)90 OR K15 THEN SOUN
D 2,120,1,10:GOTO 9090
9130 POSITION X,14:IF 86:CHR$(G):SOUND
2,121,10,10:K=M+1
9137 5P5=C-4,X-13:CHR$(C):GOTO 9090
9140 CLOSE #2:FOR I=1 TO 8:IF 5C:P(1)
THEN 9160
9150 NEXT I:GOTO 9400
9160 X=37:Y=180:FOR II=0 TO X STEP -1:
P(II+1)=P(II):RAC(II+1)=RAC(II):RE(II+1)
=RE(II):MIS(M+2,Y+12)=MIS(M,Y)
9170 X=X-12:Y=Y-12:NEXT II:P(1)=5C:RAC
13=RA:RE(13)=R
9180 MIS(M+12,Y+12)=MIS(M,Y):MIS(M,Y)=
5P5
9200 GOTO 9400
9300 POKE DL+10,7:POKE DL+19,6:POSITIO
N 4,12:IF 86:"PUSH BUTTON!"
9310 POSITION 9,18:?"High-Scorer Nr.0
":P(7)
9320 IF 5TRIG(L) THEN 9390
9400 POP CPOP:GOTO 11
10000 REM *****
10010 DATA 91,56,91,121,144,121,182,0,
91,96,91,121,144,121,182,0
10020 DATA 91,81,76,81,76,91,81,91,81,
182,91,121,144,121,182,0
10030 DATA 91,81,76,81,76,91,81,91,81,
182,91,121,144,121,182,0
10040 POKE 700,100:RESTORE 10010:FOR I
-19300 TO 19347:READ C:POKE I,C:NEXT I
10050 REM *****
10100 RESTORE 10110:FOR I=1536 TO 1791
:READ C:POKE I,C:NEXT I:RETURN
10110 DATA 167,2,141,26,2,173,4,200,20
1,0,200,180,173,5,200,201,0,200,181,10
6,0,165,1,232,153
10120 DATA 0,224,0,200,9,162,0,174,0,1
73,128,2,131,281,7,280,11,230,283,16
7,130,142,7,212
10130 DATA 162,1,134,209,281,11,200,11
130,203,162,128,142,7,212,162,2,134,2
09,201,14,200,0,190,204
10140 DATA 170,204,162,3,124,209,201,1
3,200,0,220,204,230,204,162,4,124,209,
109,0,162,0,164,205,153
10150 DATA 0,132,153,0,140,153,0,133,1
53,0,141,232,200,224,19,200,230,70,123
6,76,165,6,162,0
10160 DATA 164,204,109,192,153,153,0,1
32,109,212,153,153,0,140,109,232,153,1
53,0,133,109,252,153,153,0
10170 DATA 141,232,200,224,19,200,226,
165,204,133,205,165,201,141,0,200,141,
1,200,167,0,165,0,164,70
10180 DATA 153,0,134,153,0,142,232,200
,224,17,200,244,162,0,164,207,109,16,1
54,153,0,134,153,0,142
10190 DATA 232,200,224,17,200,241,165,
207,133,200,165,206,141,2,200,230,190,
2,206,30,153,173,30,153,201
10200 DATA 164,200,24,169,169,141,30,1
53,174,31,153,109,132,153,141,0,210,23
2,224,40,200,2,162,0
10210 DATA 162,71,153,32,32,153,30
10220 REM *****
10230 DATA 12,30,22,30,20,12,20,24,
24,50,54,50,21,1,1,0,0,0,0
10240 DATA 40,120,80,120,120,40,40,50
,24,24,70,100,230,240,120,0,0,0,0
10250 DATA 0,0,32,70,0,120,120,194,182
,30,12,0,0,0,30,180,0,0,0,0
10260 DATA 0,0,4,0,0,1,1,67,102,100,40
16,16,0,30,54,0,0,0,0
10270 DATA 62,62,127,127,127,127,127,1
27,127,127,127,127,62,62,0,0,0,0
10280 DATA 0,0,0,24,60,63,123,119,2
21,255,255,226,126,60,0
10290 DATA 0,24,60,126,70,219,219,2
55,189,153,170,182,231,0
10300 DATA 0,0,24,60,126,255,153,219,2
55,255,219,70,182,60,231,0
10310 DATA 0,24,60,126,219,153,250,195
,182,60,24,60,182,195,129,0
10320 DATA 0,231,185,185,221,231,24,24
,60,182,219,189,219,182,126,0
10400 DATA 7,0,62,127,127,127,127,127,
62
10410 DATA 12,70,40,20,40,20,40,20,40
10420 DATA 15,0,102,0,60,102,102,60,0
10430 DATA 60,0,0,0,05,170,05,170,0,0
10440 DATA 63,20,40,21,42,21,42,0,0
10450 DATA 59,0,0,21,42,21,42,20,0,0
10460 DATA 61,0,0,84,160,84,160,20,40
10470 DATA 32,0,0,0,70,62,20,0,0
10480 DATA 62,20,40,84,160,84,160,0,0
10490 DATA 27,16,24,84,89,5,23,26,25
10500 DATA 28,24,37,00,6,80,74,20,00
10510 DATA 29,0,34,21,74,20,33,10,19
10520 DATA 30,4,84,34,0,64,10,30,36
10530 DATA 11,0,126,90,126,60,182,126,
0
10540 DATA 5,0,62,119,107,93,107,119,6
2
10590 DATA 2,0,56,124,95,111,30,60,0
10600 DATA 1
10700 DATA 24,60,44,60,60,12,12,24,24,
10,54,55,63,125,1,0,0,0,0,0
10710 DATA 24,60,52,60,60,40,40,24,24,
72,76,236,252,190,120,0,0,0,0
10720 DATA 0,0,64,192,0,0,2,6,35,100,2
00,135,120,0,60,204,0,0,0,0
10730 DATA 0,0,2,0,0,0,64,96,100,54,51
,17,1,0,24,51,0,0,0,0,0
10740 DATA 55,62,119,119,107,107,93,93
,107,107,119,119,62,62,0,0,0,0
10800 DATA 104,162,0,109,0,224,157,0,1
44,109,0,225,157,0,145,109,0,226

```



```

10010 DATA 157,0,146,189,0,227,157,0,1
47,232,224,255,200,227,96
10700 REM DISKEDITOR
10710 POKE 748,156:RESTORE 11000:FOR I
=52000 TO 52142:READ C:POKE I,C:NEXT I
10720 RESTORE 11100:FOR I=52200 TO 522
60:READ C:POKE I,C:NEXT I:RETURN
11000 DATA 72,120,72,152,72,120,174,10
,152,187,0,152,201,0,240,99,189,21,152
,141,21
11010 DATA 200,189,11,152,201,1,200,6,
254,0,152,76,126,152,222,0,152,254,41,
152,189
11020 DATA 41,152,201,0,200,67,169,0,1
57,41,152,189,11,152,201,1,240,15,222,
11,152
11030 DATA 222,11,152,189,11,152,254,1
1,152,76,175,152,254,11,152,254,11,152
,189,11,152
11040 DATA 222,11,152,160,185,172,150,
201,0,240,20,189,11,152,201,1,240,0,16
9,1,157
11050 DATA 11,152,76,202,157,169,0,157
,11,152,189,0,152,141,1,200,230,10,152
,173,10
11060 DATA 152,201,10,200,5,169,0,141,
10,152,104,160,104,170,104,80,64
11100 DATA 173,30,152,141,1,210,169,0,
141,10,152
11110 DATA 169,0,141,255,2,173,120,2,2

```

```

01,15,200,1,96,206,22,152,173,22,152,2
01,0,200,26,189,4
11120 DATA 141,22,152,162,0,189,192,15
3,72,189,76,154,157,192,153,104,157,76
,154,232,224,79,200,377,96
12000 REM DISKEDITOR
12010 DIM S$(20),M$(120),P(10),R0(10)
J,RE(10):FOR I=1 TO J:POKE 53240+I,L:IN
ENT I
12020 GRAPHICS 22:POKE 553,1:POKE 88,1
:POKE 89,130:7 R0:CHR$(125):POKE 53760
,24:POKE 53761,164:POKE 53763,207
12030 POKE 53765,164:POKE 53767,162:P0
KE 53769,200:POKE 53762,50:POKE 53764,
228:POKE 53766,101
12040 GRAPHICS 1:POKE 710,1:POKE 756,2
04:POKE 755,1:POKE 700,122
12050 DL=PEEK(560)+254:PEEK(561):POKE
DL+9,6:POKE DL+14,7:POKE DL+26,6
12060 POSITION 2,4:?" please wait for"
:POSITION 24,0:?" M 0 N 0 E I."
12070 POSITION 7,14:?"(C) 1986 by J"
:CHR$(12):"RG L":CHR$(12):"FILE"
12080 FOR I=L TO M:INIZ(141+I,141+I2)
="White MANUAL":NEXT I
12010 FOR I=1 TO 0:P(1)=5000-I*100:RA
I(I)=RE(I):Z=NEXT I
12120 GOSUB 1000:GOSUB 1000:GOSUB 100
10:GOSUB 1200:RETURN
12400 REM DISKEDITOR

```

Diskeditor

Mit diesem Diskeditor für alle Atari-Computer kann beliebig auf Sektoren der Diskette in Laufwerk I zugegriffen werden. Nach dem Start erscheint folgendes Menü:

```

xxxx R: Sektor xxxx lesen
C: angezeigten Sektor
verändern
W: angezeigten Sektor
schreiben
+: Sektor +1 lesen
-: Sektor -1 lesen
A: ASCII-Darstellung
H: HEX-Darstellung
?: Kommandos anzeigen

```

Zu den einzelnen Menü-Punkten:

xxxx R: Mit dieser Option wird ein beliebiger Sektor gele-

sen und angezeigt (z.B. best 361 R den ersten Sektor des Directory einer DOS-II-Diskette).

C: Hiermit kann man mit dem Cursor auf dem Bildschirm umherfahren und den angezeigten Sektor verändern. Je nach Modus werden nur die Zeichen 0-9 und A-F oder alle ASCII-Zeichen entgegengenommen. Mit RETURN gelangt man in den normalen Kommando-Modus zurück. Alle auf dem Bildschirm vollzogenen Änderungen wurden eingelesen, aber noch nicht auf die Diskette zurückgeschrieben.

W: Das Zurückschreiben eines veränderten Sektors ge-

schieht mit diesem Befehl. Vor-sicht, der Befehl wird ohne Rückfrage ausgeführt.

+; -: Mit diesen Komman-dos ist es möglich, vorwärts und rückwärts zu "blättern". Als Kurzform für + RETURN ist auch RETURN ohne Eingabe zulässig.

A: Umschaltung auf ASCII-Darstellung. Alle Bytes eines Sektors werden als ASCII-Zeichen dargestellt. Statt RE-TURN (155) wird ein SPACE ausgegeben. Soll ein RE-TURN-Zeichen eingefügt werden, so ist dies nur im Hex-Mode möglich. Nach dem Starten befindet man sich automatisch im ASCII-Mode.

H: Umschalten auf hexadezi-male Darstellung der Bytes.

?: Anzeige des Menüs; es kann auch "HELP" eingegeben werden.

Anwendungsbeispiel: Mit diesem Programm ist es möglich, gelöschte Files wieder lesbar zu machen. Das Directory einer DOS-II-Diskette befindet sich ab Sektor 361. Vor dem Filenamen werden 5 Byte mit der folgenden Bedeutung abgelegt: Das erste Byte ist das Statusbyte. Es gibt an, ob das File gelockt, gelöscht usw. ist. Byte 2 und 3 geben die erste Sektor-nummer des Files an, Byte 4 und 5 seine Länge. Bei einem gelöschten File hat das erste Byte den Wert 128. Setz man ihn auf 98, so ist das File wieder lesbar, sofern es nicht überschrieben wurde.

Herbert Barth

Disk-Editor

```

100 REM DISKEDITOR
110 REM
120 REM
130 DIM M$(37),A$(120),CMD$(7)
140 DIM HEX$(16)
150 HEX$="0123456789ABCDEF"
160 A$(120)="":SOUND:ASCII=SMUM
170 OPEN #1,4,0,"E:":OPEN #2,4,0,"E:"
180 FOR I=1 TO 37:READ A:W$(I)=CHR$(A)
: NEXT I
190 FOR I=1770 TO I+15:READ A:POKE I,A
: NEXT I
200 REM
210 REM COMMANDS
220 REM
230 GRAPHICS 0:POKE 82,2:POKE 83,39
240 POKE PEEK(560)+254:PEEK(561)+25,13
0

```

```

250 POKE 512,234:POKE 513,0
260 POKE 5428,192:PRINT "PRINT
270 PRINT "XXXX R: Sektor XXXX lesen"
280 PRINT " C: angezeigten Sektor
verändern"
290 PRINT " W: angezeigten Sektor
schreiben"
300 PRINT " +: Sektor+1 lesen"
310 PRINT " -: Sektor-1 lesen"
320 PRINT " A: ASCII Darstellung"
330 PRINT " H: HEX Darstellung"
340 PRINT " ? : Kommandos anzeigen"
350 REM
360 REM KOMMANDO-BEARBEITUNG
370 REM
380 POSITION 5,22:PRINT "C";
390 PRINT "KOMMANDO: ";
400 POKE 634,0:POKE 702,64:TRAP 300
410 CMD$="" : INPUT #1,CMD$
420 IF LEN(CMD$)=7 THEN 300
430 IF CMD$="?" OR CMD$="HELP" THEN 23
0

```

```

440 IF CMO$="M" AND ASCII=0 THEN 500
450 IF CMO$="H" AND ASCII=1 THEN 600
460 IF CMO$="+" OR CMO$="*" THEN 020
470 IF CMO$="-" THEN 800
480 IF CMO$="^" THEN 910
490 IF CMO$(LEN(CMO$))="R" THEN IF LEN
(STR$(VAL(CMO$)))=LEN(CMO$)-2 AND CMO$
(LEN(CMO$)-1,LEN(CMO$)-1)="" THEN 750
500 IF CMO$="C" AND ASCII=1 THEN 960
510 IF CMO$="C" AND ASCII=0 THEN 110
520 GOTO 300
530 REM
540 REM SEKTORVERSTÄRKUNG
550 REM
560 ASCII=1:POKE 82,4:POKE 83,35
570 PRINT "N":POSITION 4,1
580 PRINT "SEKTORNUMMER: ";SMUM
590 PRINT
600 FOR I=1 TO 120
610 A=ASC(AS(I)):IF A=155 THEN A=32
620 PRINT "V";CHR$(A);NEXT I:GOTO 300
630 REM
640 REM SEKTORVERSTÄRKUNG
650 REM
660 ASCII=0:IF R=0 THEN 300
670 PRINT "N":POSITION 8,1
680 PRINT "SEKTORNUMMER: ";SMUM
690 POKE 82,8:POKE 83,31:PRINT
700 FOR J=1 TO 120:ASC(AS(J)):J)=A:IN
TCI/163+1? NENS(A,0):A=CI/16-INT(CI/1
63)*16+1? NENS(A,0):":NEXT J
710 POKE 82,2:GOTO 300
720 REM
730 REM SEKTORVERSTÄRKUNG
740 REM
750 SMUM=VAL(CMO$)
760 CMO$=R:GOSUB 1350
770 IF ASCII=1 THEN 500
780 GOTO 640
790 REM
800 REM SEKTORVERSTÄRKUNG
810 REM
820 SMUM=SMUM+1:GOTO 760
830 REM
840 REM SEKTORVERSTÄRKUNG
850 REM
860 SMUM=SMUM-1:IF SMUM<1 THEN SMUM=1
870 GOTO 760
880 REM
890 REM SEKTORVERSTÄRKUNG
900 REM
910 IF R<0 THEN CMO$=R:GOSUB 1350
920 GOTO 300
930 REM
940 REM SEKTORVERSTÄRKUNG
950 REM
960 POSITION 3,22:PRINT "CHANGE hit
RETURN to quit !";
970 POSITION 4,3:PRINT "++";
980 GET M,I:PRINT CHR$(I);
990 IF I=155 THEN 1030
1000 IF PEEK(84)3 THEN ? "++++";
1010 IF PEEK(84)2 THEN ? "++++";
1020 GOTO 300
1030 POKE PEEK(94)+256*PEEK(95),PEEK(9
3)
1040 POSITION 4,3:FOR I=1 TO 120
1050 GET M,0:IF ASC(AS(I))<155 OR A<
32 THEN AS(I,3)=CHR$(A)
1060 NEXT I
1070 PRINT "++":GOTO 120:GOTO 300
1080 REM
1090 REM SEKTORVERSTÄRKUNG
1100 REM
1110 POSITION 1,22:PRINT "CHANGE hit
RETURN to quit !";
1120 POKE 82,8:POKE 83,31
1130 POSITION 8,3:PRINT "++";
1140 GET M,I:IF I=155 THEN 1260
1150 IF I<120 OR I>97 THEN 1140
1160 IF I<48 AND I>31 THEN 1140
1170 IF I<65 AND I>97 THEN 1140
1180 PRINT CHR$(I):IF PEEK(93)=26 AND
I<30 THEN PRINT "++";
1190 IF PEEK(93)=26 AND I=30 THEN PRIN
T "++";
1200 IF PEEK(93)=0 THEN 1140
1210 I=PEEK(84)
1220 IF I=12 AND PEEK(85)32 THEN POSI
TION 1,12:PRINT "++";
1230 IF I=2 THEN ? "++++";
1240 IF I=19 THEN ? "++++";
1250 GOTO 1140
1260 POKE PEEK(94)+256*PEEK(95),PEEK(9
3)
1270 POSITION 8,3:FOR J=1 TO 120
1280 GET M,0:A=A-48:IF A>9 THEN A=A-7
1290 GET M,I:I=40:IF I>9 THEN I=1-7
1300 AS(I,J)=CHR$(I+AM16):GET M,0:NEXT
I J
1310 PRINT "++":POKE 82,2:POKE 83,35:
GOTO 300
1320 REM
1330 REM SEKTORVERSTÄRKUNG
1340 REM
1350 R=USR(ADR(CMO$),CMO$,ADR(AS),SMUM)
1360 POKE 54206,192
1370 IF R=1 THEN RETURN
1380 IF R=129 AND SMUM=721 OR SMUM=104
1 THEN SMUM=SMUM-1
1390 POP :POSITION 2,22:PRINT "G";
1400 PRINT "CHANGE hit
RETURN to quit !";
1410 FOR I=1 TO 300:NEXT I:GOTO 230
1420 REM
1430 REM DATA FUER DISK HANDLER
1440 REM
1450 DATA 104,107,1,141,1,2,104,104
1460 DATA 141,2,3,104,141,5,3,104
1470 DATA 141,4,3,104,141,11,3,104
1480 DATA 141,10,3,32,83,220,132,212
1490 DATA 169,0,133,212,96
1500 REM
1510 REM DATA FUER DISK-INTERRUPT
1520 REM
1530 DATA 72,163,200,141,10,212,141,24
1540 DATA 200,165,0,141,23,200,104,64

```

Calc 800

"Calc 800" ist ein universelles, allgemein verwendbares mathematisches Programm. Es erlaubt, sowohl einfache wie auch relativ komplexe Probleme mathematischer, kaufmännischer und technischer Art mit dem Atari zu lösen. Es läßt sich auf Diskette abspeichern und bei Bedarf wieder laden. Mit der eingebauten Hardcopy-Funktion können beliebige Ausdrücke auf Atari- und Ep-

son-kompatiblen Printern angefertigt werden.

Jedes Arbeitsblatt (Spreadsheet) besteht aus einer variablen Matrix mit maximal 2016 Einträgen. Über die jeweiligen Spalten- und Zeilenummern werden die Matrixfelder angesprochen und können mit mathematischen Funktionen miteinander verknüpft werden. Der Vorteil liegt nun darin, daß

sämtliche Fixwerte und Berechnungsformeln nur einmal eingegeben werden müssen und damit beliebig viele Berechnungen möglich sind. Damit können verschiedene Modelle durchgespielt werden.

Das Programm beinhaltet zur leichten Handhabung eine HELP-Funktion, die jederzeit vom Eingabe-Modus aus anwählbar ist. Nach Benutzung von HELP wird wieder ins Kalkulationsprogramm zurückverweigelt und das Arbeitsblatt mit aktuellem Stand wieder aufge-

baut.

Folgende Kommandos sind möglich:

- Z - Fenster um eine Zeile abwärts
- A - Fenster zum Blattanfang
- F - Fenster zum Blattende
- SL - Fenster um eine Spalte nach links schieben
- SR - Fenster um eine Spalte nach rechts schieben
- SH - Fenster um beliebige Spalten schieben
- SV - Spaltenanzahl am Bildschirm ändern

N - Spaltenname eingeben
 S - Spaltensumme errechnen
 SO - Sortierung einer Spalte
 W - feldweise Werteingabe
 WV - Wert eines Feldes ändern
 O - Ausstieg aus dem Programm
 LF - alle Matrixfelder löschen (Formeln bleiben erhalten)
 L - Felder und Formeln löschen (Neuinitialisierung)
 LA - Arbeitsblatt laden
 SP - Arbeitsblatt speichern
 HR - Hardcopy anfertigen
 RR - Rückrechnen, d.h., alle Feldinhalte sämtlicher Spalten werden nach Formel oder Wertänderung aktualisiert.
 FE - Formeleingabe. Die berechneten Resultate werden in die jeweiligen Felder und Spalten eingetragen.

Eine Formel besteht aus einem oder zwei Operanden, und bei jedem gibt es 3 Möglichkeiten:

- a) eine Zahl; sie wird in jede Zeile der Spalte eingetragen.
 b) S und Spaltennummer. So

trägt z.B. S1 die Summe der ersten Spalte in jede Zeile der angegebenen Spalte ein.
 c) ZS und Spaltennummer. ZS2 bildet z.B. die Zeilensummen ab Spalte 2.

Mit zwei Operanden gilt das Format OPERAND1, OPERAND2, OPERATOR. Der erste Operand darf keine Zahl sein. Er muß K (Spaltennummer) oder S (Spaltennummer) lauten, also z.B. K3 und S2. Der zweite Operand kann eine Zahl oder K (Spaltennummer) oder S (Spaltennummer) sein, als letztes folgt der Operator:

+, -, *, / (Grundrechenarten), W (Wurzeln ziehen), ^ (Potenzieren), % (prozentuale Differenz zwischen Op. 1 und Op. 2)

Zu trennen sind die Operanden und der Operator durch Komma, Strichpunkt oder Doppelpunkt (wahlfrei).

Hier einige Beispiele:
 K1.K3,e multipliziert die Felder der Spalte 1 mit den Feldern der Spalte 3 und trägt die Ergebnisse in die Felder der angegebenen Spalte ein.

K4,S2,↓ addiert zu jedem Feld der Spalte 4 die Spaltensumme der Spalte 2 und trägt die Ergebnisse in die Felder der angegebenen Spalte ein.

K5,W berechnet die Wurzeln der Felder der Spalte 5 und trägt die Ergebnisse in die Felder der angegebenen Spalte ein.

K10,2,↓ dividiert jedes Feld der Spalte 10 durch den Faktor 2.5 und trägt die Ergebnisse in die Felder der angegebenen Spalte ein.

Achtung: Das Programm rundet alle Eingaben auf 2 Kommastellen ab.

Gerhard Leckat

KOMMANDO?	20 ZEILEN	5 SPALTEN	HILFE
=====S1=====S2=====S3=====			
BRUTTO	3% SKONTO	ZWI. SUMME	
1 10	-0.3	9.7	
2 20	-0.6	19.4	
3 30	-0.9	29.1	
4 40	-1.2	38.8	
5 50	-1.5	48.5	
6 60	-1.8	58.2	
7 70	-2.1	67.9	
8 80	-2.4	77.6	
9 90	-2.7	87.3	
10100	-3	97	
11110	-3.3	106.7	
12120	-3.6	116.4	
13130	-3.9	126.1	
14140	-4.2	135.8	
15150	-4.5	145.5	
16160	-4.8	155.2	
17170	-5.1	164.9	
2100	-6.3	2037	

EINGABE?HC

Bild 1: Hardcopy

Calc 800

```

10 DEN C=10000000 ; TO:BY GLEBERE
10000000:WLEN:1200
100 SETCOLOR 2,0,0
105 GRAPHICS 2
106 POSITION 1,0:7 06;" 3-DIG BOOK"
107 POSITION 1,1:7 06;"
108 POSITION 1,3:7 06;" 1 - ALTES BLAT
1"
109 POSITION 1,4:7 06;"
110 POSITION 1,5:7 06;" 2 - BLATT NEU"
110 POSITION 1,7:7 06;" ANLEGER"
150 ? 1? " AUSWAHL"
160 OPEN H1,4,0,"K:"
170 GET H3,Z
175 CLOSE H3
180 IF Z=49 THEN 4240
185 IF Z=50 THEN 220
199 GOTO 180
220 DIM H5(12),H5(1),F15(10)
230 DIM I5(2),FILES(13),HELFS(9),H5(10)
),FILHFS(12),ANS(8)
240 DIM SPRAWF(189):SPRAWN(13):=""
SPNA
H5(189):SPRAWF(1):SPRAWN(2):SPRAWN(3)
250 DIM FLANGS(252):FLANG3(13):=""
FLAN
G5(252):FLANGS(1):FLANGS(2):FLANGS(3)
304 SETCOLOR 2,0,0
305 GRAPHICS 2
307 POSITION 2,3:7 06;" ARBEITSBLATT"
310 POSITION 2,6:7 06;" DEFINIEREN "
315 POKE 756,204
320 ? 1? " BITTE IRGENDWIE TASTE DRÜ
CREN"
330 OPEN H2,4,0,"K:"
333 GET H3,T
335 CLOSE H3
340 GRAPHICS 0:7 CHR$(125)
355 TRAP 355
360 POSITION 27,2:7 "" :POSITION 0,
2:7 " " :POSITION 0,2:7 " " :IMPU

```

```

T H1:IF C=1+(C1+96) THEN 360
365 TRAP 365
370 POSITION 27,4:7 "" :POSITION 0,
4:7 " " :POSITION 0,4:7 " " :IMPU
T H1:IF C=1+(C1+96) THEN 370
375 TRAP 375
380 POSITION 27,6:7 "" :POSITION 0,6
:7 " " :POSITION 0,6:7 " " :IMPU
C1:IF (C1+(C1+25)*(C1)0) THEN 300
385 TRAP 385
390 POSITION 0,12:7 " " :IMPU
ANS:IF ANS="" THEN 1
90
392 IF LEN(ANS)>0 THEN 390
405 H=RL+1:G=M
410 DIM C=0:FS=0,H5(6),O=0,N=0
412 FOR Y=1 TO 10
413 FOR Z=1 TO M
414 O=V,Z)=0
415 NEXT Z
416 NEXT Y
420 ? CHR$(125)
450 IS="" :IT=0:IR=40
460 FOR Y=1 TO C1
470 C=V=(C=INT(O=C1)-INT(C7/C1)+1)
480 NEXT U
490 IF IS="" THEN 1740
500 J=0:K=0:5=0
510 IF IS="" THEN 1740
512 L=N1
520 5=0:IF L>17 THEN L=17
530 FOR I=1 TO L:POSITION 0,14:5=541:
? CHR$(252):3:NEXT I
570 POKE 732,0:POSITION 0,0:7 " " :IMPU
06 " " :ZEILEN " " :IMPU " " :SPALTE N
ILFE"
580 POSITION 0,1:FOR I=1 TO 39:7 "" :
NEXT I
590 FOR C=1 TO C1:POSITION (C=INT(O=C
13)-INT(C1/C1),1:7 "" :C-T:NEXT C
600 IF IS="" THEN 5000:GOTO 1150
610 GOSUB 1670:POSITION 0,22:7 " " :IMPU
:IMPU IS

```

```

624 IF IS="" THEN 610
626 IF IS="MC" THEN 12000
628 GOSUB 1832
630 IF ((IS="A")*(CR1(18))+(IS="E")*(CR1(18))) THEN 570
640 IF IS="E" THEN 1730:REM ANFANG
650 IF IS="A" THEN 500:REM ENDE
660 IF IS="SL" THEN 1720:REM SPALTE LI
NE5
670 IF IS="58" THEN 1710:REM SPALTE RE
CHTS
680 IF IS="58" THEN 1800:REM SHIFT
690 IF IS="Z" THEN 2630:REM ZEILE ADDO
ERTS
695 IF IS="W" THEN 5000:REM HELP
700 IF IS="50" THEN 2930:REM SORTIERE
SPALTE
710 IF IS="FE" THEN 1970:REM FORMEL EI
NEHEN
720 IF IS="RR" THEN 2120:REM RUECKRECH
NEH
730 IF IS="S" THEN 1260:REM SPALTENSUM
ME
740 IF IS="P" THEN 880:REM MERTEINGABE
750 IF IS="0" THEN 1800:REM SPALTENNAE
ME
760 IF IS="5W" THEN 1340:REM AENDERN 5
PALTENBREITE
770 IF IS="WW" THEN 1410:REM AENDERN M
ERT
780 IF IS="LF" THEN 2830:REM LOESCHE D
ATENFELDER
790 IF IS="L" THEN ? CHR$(125):RUN IRE
M LOESCHE DATEN U. FORMELN
810 IF IS="52" THEN 2740:REM SPEICHERN
820 IF IS="LA" THEN 4240:REM LADEN
830 IF IS="0" THEN ? CHR$(125):REM :EN
D
840 GOTO 570
880 GOSUB 1680
890 POSITION 0,0: ? "SPRANG(SPRANG-C,9NC)=""
SPRANG(SPRANG-C,9NC)=""
900 U=C+T:FLANGS(12NC-11,12NC)="ZAH
L"
910 FOR R=1 TO N1
920 R1=R-K:IF R1(1) THEN 940
930 POSITION C(V),R1+2: ? ""
935 TRAP 925
940 GOSUB 1670:POSITION 0,22: ? "SPRANG(S
PRANG):INPUT N0
950 IF LEN(N0)=0 THEN 940
954 IF LEN(N0) > 9 THEN 940
960 O(R,C)=VAL(N0):O(R,C)=INT(O(R,C)*1
00+0.5)/100:IF R1(1) THEN 1000
970 GOSUB 1640
980 IF (R1(16))*O(R,N1) THEN POSITION 0,
20: ? ""
990 GOSUB 2620
990 POSITION 0,21: ? ""
1000 NEXT R
1010 IF R1(16) THEN 1740
1012 IF O(R,C) < 0 THEN GOTO 1280
1015 GOTO 570
1050 GOSUB 1680:REM SPALTENNAE
ME
1060 U=C+T
1070 POSITION 0,0: ? "SPRANG(SPRANG-C,9NC)=""
SPRANG(SPRANG-C,9NC)=""
1080 GOSUB 1670:SPRANG(SPRANG-C,9NC)=""
SPRANG(SPRANG-C,9NC)=""
1083 POSITION 0,22: ? "EINGABE":INPUT
HILFS:SPRANG(SPRANG-C,9NC)=HILFS
1090 IF SPRANG(SPRANG-C,9NC)=""
THEN 1080
1110 IF (V(1)+(V(1))) THEN 570
1120 POSITION 0,2: ? "" :POSITION C(V)
+2: ? "" :POSITION C(V),2: ? SPR
ANG(SPRANG-C,9NC)
1130 GOTO 570
1150 POSITION 0,2: ? ""
1160 FOR C=1-T TO C1-T
1170 IF C=N THEN RETURN
1190 U=C+T
1200 POSITION C(V),2: ? SPRANG(SPRANG-C,9NC)
C)
1210 NEXT C
1220 RETURN
1280 GOSUB 1680:REM SPALTENSUM
ME
1270 U=C+T
1280 O(R,C)=0
1290 FOR R=1 TO N1:O(R,C)=O(R,C)+O(R,C
):O(R,C)=INT(O(R,C)*100+0.5)/100:NEXT
R
1300 IF (V(1)+(V(1))) THEN 1320
1310 POSITION C(V),20: ? ""
9310 POSITION C(V),20: ? O(R,C)
1320 IF IS="0" THEN 2210
1325 GOTO 570
1360 POSITION 0,0: ? "SPALTENNAE:SPRANG(S
PRANG-C,9NC)=""
1370 GOSUB 1680:IF C) THEN 1300
1372 C1=C
1375 GOTO 460
1410 XS="":REM AENDERN NERT
1420 GOSUB 1680
1430 PHILFS=FLANGS(12NC-11,12NC)
1431 POKE 756,204
1432 IF PHILFS=""ZAHL "" THEN PO
SITION 0,0: ? "SPRANG(SPRANG-C,9NC)=""
SPRANG(SPRANG-C,9NC)=""
1440 GOTO 1450
1440 GOTO 1470
1450 GOSUB 1670:POSITION 0,22: ? "SPRANG(S
PRANG):INPUT N5
1460 IF N5="" THEN 570
1470 POSITION 0,0: ? "SPRANG(SPRANG-C,9NC)=""
ZAHL VON 1 BIS "N1": ? ""
1475 TRAP 0000
1480 GOSUB 1670:POSITION 0,22: ? "SPRANG(S
PRANG):INPUT R
1490 IF (R(1)+(R(1))) THEN 1480
1500 R1=R-K1:U=C+T
1510 POSITION 0,0: ? "SPRANG(SPRANG-C,9NC)=""
SPRANG(SPRANG-C,9NC)=""
1515 TRAP 0000
1520 GOSUB 1670:POSITION 0,22: ? "SPRANG(S
PRANG):INPUT N1:O(R,C)=N1
1530 FLANGS(12NC-11,12NC)="ZAH
L"
1540 IF (V(1)+(V(1)))+(CR1(1)+(CR1(1))) TH
EN 1560
1550 GOSUB 1680
1560 IF O(R,C) < 0 THEN 1200
1570 GOTO 570
1580 GOSUB 1630:POSITION 0,0: ? "SPRANG(S
PRANG-C,9NC)="" ZAHL VON 1 BIS "N1": ? ""
1590 TRAP 0000:GOSUB 1670:POSITION 0,2
2: ? "SPRANG(SPRANG-C,9NC)=""
1620 IF (C(1)+(C(1))) THEN 1610
1630 POSITION 0,0: ? ""
1640 POSITION 0,12: ? "B I T T E M A R
T E N":RETURN
1650 POSITION C(V),R1+2: ? ""
1650 POSITION C(V),R1+2: ? INT(O(R,C)*100+0.
5)/100:RETURN
1670 POSITION 0,22: ? ""
1674 RETURN
1710 IF (C1-T)=M THEN 570
1712 T=T-1
1714 GOTO 1740
1720 IF (T)=0 THEN 570
1728 T=T+1:GOTO 1740
1730 K=N1-17:5=N1
1740 T=CHR$(125)
1750 FOR C=1-T TO C1-T:REM SPALTEN AUF
SCHNITT
1760 IF C=N THEN 1060
1770 U=C+T
1780 IF SPRANG(SPRANG-C,9NC)=""
THEN POSITION C(V),2: ? SPRANG(SPRANG-C,9NC)
C)
1800 FOR R=ABS(C)+1 TO ABS(C)+L
1810 R1=R-K
1820 POSITION C(V),R1+2:IF O(R,C) < 0 T
HEN PRINT INT(O(R,C)*100+0.5)/100
1830 NEXT R
1840 IF O(R,C) < 0 THEN POSITION C(V),2
0: ? O(R,C)
1850 NEXT C
1860 IF IS="A" THEN 520
1862 IF IS="LA" THEN 520
1865 S=5-L:GOTO 530

```

```

1700 POSITION 0,0:?"XXXXXXXXXXXXXXXXXXXX"
1705 REM TRAP 1905
1710 GOSUB 1670:POSITION 0,22:?"XXXXXXXX"
1715:INPUT ES
1720 IF (ES(1)+(ES-1)*M-C1) THEN 1710
1730 T-1-ES:GOTO 1740
1770 REM TRAP 8000:GOSUB 1600
1775 GOSUB 1600
1780 POSITION 0,0:?"XXXXXXXXXXXXXXXXXXXX"
1785 FLAG$(12MC-11,12MC)=""
1790 GOSUB 1670:POSITION 0,22:?"XXXXXXXX"
1795:INPUT PHILF$:IF PHILF$="" THEN 1790
1800 FLAG$(12MC-11,12MC)=PHILF$
1810 V=C+1
1820 FOR N=1 TO M1
1830 R1=R-K
1840 GOSUB 2260
1850 IF (V(1)+(V)*C1)+(R1(1)+(R1)*17) THEN
EN 2070
1860 GOSUB 1600
1870 NEXT R
1880 IF (R,C) THEN 1200
1895 GOTO 570
1920 FOR C=1 TO M
1925 PHILF$=FLAG$(12MC-11,12MC)
1930 IF PHILF$(1,3)="" THEN 2210
1940 IF FLAG$(12MC-11,12MC)="ZANL"
THEN 2210
1945 V=C+1
1950 FOR N=1 TO M1
1960 R1=R-K:R=RD(C):GOSUB 2260
1970 IF (V(1)+(V)*C1)+(R1(1)+(R1)*17) THEN
EN 2170
1980 IF (R,C) THEN GOSUB 1600
1990 NEXT R
2000 IF (R,C) THEN 1200
2010 NEXT C
2020 GOTO 570
2060 B$=FLAG$(12MC-11,12MC)
2065 LAENGE=0:FOR I=1 TO 12
2070 IF B$(I,I)="" THEN 2264
2075 LAENGE=LAENGE+1
2080 NEXT I
2085 TRAP 2200
2090 IF (VAL(PHILF$)) THEN (R,C)=VAL(PHILF$):RETURN
2095 TRAP 8000
2100 IF (B$(1,1)=""$)*LAENGE(4) THEN
(R,C)=R,C,VAL(B$(2,LAENGE)):RETURN
2105 IF B$(LAENGE,LAENGE)="" THEN C=1:
GOTO 2360
2110 IF B$(LAENGE,LAENGE)="" THEN C=2:
GOTO 2360
2120 IF B$(LAENGE,LAENGE)="" THEN C=3:
GOTO 2360
2130 IF B$(LAENGE,LAENGE)="" THEN C=4:
GOTO 2360
2140 IF B$(LAENGE,LAENGE)="" THEN C=5:
GOTO 2360
2150 IF B$(LAENGE,LAENGE)="" THEN C=6:
GOTO 2360
2160 IF B$(LAENGE,LAENGE)="" THEN C=7:
GOTO 2360
2170 IF B$(1,2)=""$ THEN 2570
2180 IF B$(1,1)=""$ THEN ZH=N:GOTO 238
0
2170 ZH=C
2200 B$=B$(2,LAENGE-2)
2210 F1=VAL(B$):F1$=STR$(F1)
2220 IF C=7 THEN 2520
2230 LAENGE=LEN(B$)
2240 B$=B$(LAENGE-(LAENGE-LEN(F1$)-2),
LAENGE)
2250 IF B$(1,1)=""$ THEN ZH=R:GOTO 244
0
2260 IF B$(1,1)=""$ THEN ZH=N:GOTO 244
0
2270 F2=VAL(B$)
2280 ON G GOTO 2460,2480,2500,2520,252
6
2285 LAENGE=LEN(B$)
2290 F2=VAL(B$(LAENGE-LAENGE-2),LAENGE
E3)
2450 ON G GOTO 2470,2490,2510,2530,252
6,2560
2460 (R,C)=CZM,F1:PF2:(R,C)=INT((R,C)
+1000+.5)/100:RETURN
2470 (R,C)=CZM,F1:WQ(CZM,F2):(R,C)=I
NT((R,C)+1000+.5)/100:RETURN
2480 (R,C)=CZM,F1:PF2:RETURN
2490 (R,C)=CZM,F1:WQ(CZM,F2):RETURN
2500 (R,C)=CZM,F1:PF2:(R,C)=INT((R,C)
+1000+.5)/100:RETURN
2510 (R,C)=CZM,F1:WQ(CZM,F2):(R,C)=I
NT((R,C)+1000+.5)/100:RETURN
2520 (R,C)=CZM,F1:PF2:RETURN
2525 (R,C)=INT((R,CZM,F1)+1000+.5)
/100:RETURN
2530 (R,C)=CZM,F1:PF2:(R,C)=INT((R,C)
+1000+.5)/100:RETURN
2535 (R,C)=CZM,F1:WQ(CZM,F2):(R,C)=I
NT((R,C)+1000+.5)/100:RETURN
2540 (R,C)=CZM,F1:WQ(CZM,F2):RETURN
2545 (R,C)=CZM,F1:WQ(CZM,F1):WQ100:100:
RETURN
2570 (R,C)=0
2580 LAENGE=0:FOR I=1 TO 12
2590 IF B$(I,I)="" THEN LAENGE=LAEN
GE+1
2600 NEXT I
2605 ZH=VAL(B$(2,LAENGE))
2610 FOR I=ZM TO C-1:(R,C)=R,C+C*(R
):NEXT I:RETURN
2620 IF R1+2*N THEN RETURN
2630 K=K+1:5=5+1
2640 POSITION 0,20:1 5
2650 PORE 04,23:1 1RETURN
2670 GOSUB 1670:J=J+1
2700 IF (S=N1)+(R=N1) THEN 570
2710 R=17+J:K=N+1:5=5+1
2720 POSITION 0,20:1 5:1 7:?"POSITION 10,
22:1 7:?"GOSUB 2650
2730 FOR C=1 TO C1-1
2740 IF FLAG$(12MC-11,12MC)=""
THEN 2700
2750 V=C+1
2760 POSITION C(V),19:IF (R,C)=0 THEN
2770
2785 ? (R,C)
2790 IF (R,C) THEN POSITION C(V),2
0:1 7 (R,C)
2800 NEXT C
2810 GOTO 610
2830 ? CHR$(125)
2840 GOSUB 1650
2850 FOR C=1 TO M
2860 FOR N=1 TO N1:(R,C)=R:NEXT R
2870 IF (R,C) THEN (R,C)=R,C+0
2880 NEXT C
2890 ? CHR$(125):GOTO 500
2920 GOSUB 1600
2940 ? CHR$(125)
2945 GOSUB 1650
2950 FL=0
2960 FOR N=1 TO M1-1
2970 IF (R,C) THEN (C=C+1,C) THEN 2990
2980 TEMP=R,C,C:(R,C)=R,C+(R,C):(R,C+1
,C)=TEMP:FL=1
2990 NEXT R
3000 IF FL THEN THEN 2950
3005 GOTO 1740
3040 ? CHR$(125)
3045 FILES=""FILES(1,2)=""$
3050 FILES(3)=0$
3055 OPEN H1.0.0,FILES
3060 ? H1:FLAMES
3070 ? H1:SPRANS
3075 ? H1:H1
3080 ? H1:N
3085 ? H1:M
3090 ? H1:C1
3095 FOR I=1 TO M
3100 FOR J=1 TO N
3105 ? H1:R(I,J)
3110 NEXT J
3120 NEXT I
3130 CLOSE H1:GOTO 1740
3140 GRAPHICS 0:CLR
3145 DIM B$(12),X$(1),F1$(10)

```

```

4244 DIM IS(2),FILES(11),HILF(9),MS(1
8),PHILF(12),ANS(8)
4245 DIM SPNAMS(109):SPNAMS(1)= "" :SPN
AMS(109)=SPNAMS(1):SPNAMS(2)=SPNAMS(1)
4246 DIM FLANGS(252):FLANGS(1)= "" :FL
ANGS(252)=FLANGS(1):FLANGS(2)=FLANGS(1)
4252 POSITION 3,10:"BLATTNAME":IMP
T ANS:IF ANS="" THEN 4252
4253 FILES="" :FILES(1,2)= "" :FILES(2)
=ANS
4254 TRAP $100
4255 OPEN #1,4,0,FILES
4256 INPUT #1:FLANGS
4270 INPUT #1:SPNAMS
4271 INPUT #1:#1
4272 INPUT #1:#1
4273 INPUT #1:#1
4274 INPUT #1:C1
4276 L=#1:C=C1:NR=40
4278 DIM C(0),FS(0),NS(6),ON(0),NO
4279 FOR Y=1 TO 6
4280 FOR Z=1 TO M
4281 OCT,Z)=0
4282 NEXT Z
4283 NEXT Y
4284 FOR I=1 TO M
4290 FOR J=1 TO M
4300 INPUT #1:HILF:O(J,I)=HILF
4315 NEXT J
4320 NEXT I
4330 M=#1+1:L=17:IS="" :LA":J=0
4331 IF #1(18) THEN L=#1
4332 FOR W=1 TO C1
4334 C(W)=COWINT(CR/C1)-INT(C7/C1)+1
4335 NEXT W
4340 CLOSE #1:GOTO 1740
5000 ? CHR$(125):POKE 756,204:POKE 82,
8:POKE 752,3
5005 SETCOLOR 2,0,0
5010 POSITION 0,0:"
[REDACTED]
5015 ? "
"
5020 ? "FOLGENDE KOMMANDOS SIND M.G.LIC
H:"
5040 ? :? " [ ] - Fenster um 1 Zeile abw
rts"
5050 ? :? " [ ] - Fenster zum Blattanf
g"
5065 ? :? " [ ] - Fenster zum Blattende"
5080 ? :? " [ ] - Fenster eine Spalte 1
inks"
5092 ? :? " [ ] - Fenster eine Spalte r
echts"
5095 ? :? " [ ] - Fenster um beliebige
Spalten schieben"
5098 ? :? " [ ] - Spaltenanzahl an Schi
ft ändern"
5070 ? :? " [ ] - Spaltenname eingeben"
5000 POSITION 0,22:" [REDACTED]-MEHR HE
LP [REDACTED]-ZUM PROGRAMM"
5005 OPEN #1,4,0,"K:"
5007 GET #1,X
5008 CLOSE #1
5090 IF X=155 THEN $100
5092 IF X=27 THEN GRAPHICS 0:GOTO 1740
5095 GOTO 5085
5100 ? CHR$(125):POKE 756,204:POKE 82,
8:POKE 752,3
5110 POSITION 0,0:" [REDACTED]
[REDACTED] SEITE 2 VON 6"
5115 ? "
"
5150 ? " [ ] - Errechne Spaltensumme"
5155 ? :? " [ ] - Sortiere eine Spalte"
5160 ? :? " [ ] - feldwweise Werteingabe"
5182 ? :? " [ ] - Wert eines Feldes "nd
ern"
5183 ? :? " [ ] - Leeren Felder (Forme
in bleiben)"
5185 ? :? " [ ] - Ausstieg aus CALC 800XL
"
5170 ? :? " [ ] - Arbeitsblatt laden"
5175 ? :? " [ ] - Arbeitsblatt speichern
"
5180 ? :? " [ ] - Hardcopy anfertigen"
5182 ? :? " [ ] - Neues Arbeitsblatt anl
egen"

```

```

5180 POSITION 0,22:" [REDACTED]-MEHR HE
LP [REDACTED]-ZUM PROGRAMM"
5190 OPEN #1,4,0,"K:"
5195 GET #1,X
5200 CLOSE #1
5210 IF X=155 THEN $100
5212 IF X=27 THEN GRAPHICS 0:GOTO 1740
5220 GOTO 5190
5300 ? CHR$(125):POKE 756,204:POKE 82,
8:POKE 752,3
5310 POSITION 0,0:" [REDACTED]
[REDACTED] SEITE 3 VON 6"
5315 ? "
"
5330 ? " [ ] - Nachrechnen, d.h. alle F
eldinhalte"
5332 ? :? " te aller Spalten werden
nach For"
5334 ? :? " weil-od. Wertänderung akt
ualisiert"
5350 ? " [ ] - Formel eingeben"
5355 ? :? " Das berechnete Resultat wir
d in die an"
5360 ? "gegebene Spalte eingetragen. B
ie Formel"
5363 ? "besteht aus einem oder zwei Op
eranden."
5365 ? "Bei einem Operanden gibt es 3
Möglich"
5370 ? "keiten"
5380 ? :? " a) eine Zahl (sie wird in
je00 Zeile"
5383 ? " der Spalte eingetragen)."
5390 ? :? " b) 3 und Spaltennummer,etw
a 51-tr"
5393 ? :? " die Summe der ersten Spalt
e in jede"
5395 ? :? " Zeile der angegebenen Spal
te ein."
5400 POSITION 0,22:" [REDACTED]-MEHR HE
LP [REDACTED]-ZUM PROGRAMM"
5410 OPEN #1,4,0,"K:"
5420 GET #1,X
5430 CLOSE #1
5440 IF X=155 THEN $100
5450 IF X=27 THEN GRAPHICS 0:GOTO 1740
5460 GOTO 5410
5500 ? CHR$(125):POKE 756,204:POKE 82,
8:POKE 752,3
5510 POSITION 0,0:" [REDACTED]
[REDACTED] SEITE 4 VON 6"
5515 ? "
"
5530 ? " [ ] 25 Spaltennummer, etwa 252
-bildet"
5535 ? " die Zeilensummen ab der Sp
alte 2"
5550 ? "Mit zwei Operanden gilt das Fo
rmel:"
5555 ? :? " OPERAND1:OPERAND2:OPERA
TOR"
5560 ? :? "Der erste Operand darf kein
e Zahl sein."
5565 ? "Er MUSS K(Spaltennummer),etwa
K3
"
5567 ? "oder S(Spaltennummer), etwa 52
lauten."
5570 ? "Der zweite Operand kann eine Z
ahl oder"
5575 ? "K(Spaltennummer) oder S(Spalte
nummer)"
5580 ? "sein. Als letztes folgt der Op
erator:"
5585 ? :? " [ ]:[ ]:[ ] (Grundrechnungsart
en)"
5590 ? :? " [ ] (Wurzelziehen): [ ] (Pot
enzieren)"
5595 ? :? " [ ] Proz. Differenz zwischen
Op.1 u.Op.2"
5600 POSITION 0,22:" [REDACTED]-MEHR HE
LP [REDACTED]-ZUM PROGRAMM"
5610 OPEN #1,4,0,"K:"
5615 GET #1,X
5620 CLOSE #1
5625 IF X=155 THEN $100
5630 IF X=27 THEN GRAPHICS 0:GOTO 1740
5700 ? CHR$(125):POKE 756,204:POKE 82,
8:POKE 752,3

```

IRATA

MERENDORF
PLATZ 8
1 BERLIN 10
030/3453061

VIDEOMEISTER



DER VIDEOMEISTER IST EINE ENTWICKLUNG UM BILDER VON CAMERA ODER VIDEO IN DEN ADAT ZU LEBEN MIT DEM PROGRAMM KANN MAN GRAUWERTE AENDERN UND AUCH FARBBILDER ERSTELLEN. DIE SOFTWARE HAT PROFISTANDARD

199.-

DRUCKINTERFACE

EINE VERBINDUNG VON COMPUTER ZUM DRUCKER OHNE VORHER SCHALTER ZU STELLEN ODER EIN PROGRAMM ZU LADEN. ES WIRD NUR IN DIE FLOPPY GESTECKT UND MIT DEM DRUCKER VERBUNDEN.

199.-

DAS HSB WIRD IN DIE 1050 EINGEBAUT. DAS LAUFWERK HAT NUN 178K UND WIRD ERHEBLICH SCHNELLER VON FAST JEDER DISK IST EINE COPY ZU ERSTELLEN. DER EINBAU IST EINFACH UND OHNE LÖTEN.

HIGH SPEED 1050

149.- SOUND
MEISTER

DER SOUNDMEISTER IST EIN INTERFACE MIT DEM MAN MUSIK ODER SPRACHE VOM RADIO ODER MICROFON IN DEN COMPUTER SPEICHERT. AUCH HIER GIBT ES ECHTE PROFISoftware DAZU. MIT DER 512K KARTE IST PLATZ FÜR 3 MINUTEN.

HEIKOHA HARDCOPY

GP 100	38.-
GP 100 AT	38.-
GP 500	38.-
GP 500	38.-
GP 550 AT	38.-

PRINT SHOP DRUCK

GP 100	28.-
GP 500	28.-

HRDCOPY

ATARI 1020	28.-
EPSON	28.-

PROGRAMME

VOKABEL	28.-
DATEI 7.8	38.-
CATLOG	28.-
PROTECT	28.-
UNDELETER	28.-
SUPERCOPY	38.-
KATALOG-INFO	88.-

SPEED MEISTER

DSR UND DUP LADEN IN 6 SEKUNDEN
SECTORCOPY 1 DISK 17 SEKUNDEN
SECTORCOPY DD 26 SEKUNDEN
62 SECTOREN BILD 4 SEKUNDEN
SPROMENNER ANSCHLIESSIG
CENTRONICS INTERFACE AUSGANG
24 BIT USER PORT
HIGH SPEED BOARD ERFORDERLICH

COMPUTERJOB

SEIT 1981
2700 LEUTE
EUROKAWART
INFO FREI

030 345 30 61

SUPERLABEL 2

BILDER UND SCHRIFTEN AUF ETIKETTEN BRINGEN WAR SCHON IMMER EIN TRAUM. ETIKETTEN IN DREI GROSSEN KANN MAN BENUTZEN. DER DRUCKER SOLLTE EPSON KOMP. SEIN

40.-

512K RAM 399.-

ENDLICH EINMAL RICHTIG SPEICHER HABEN. DAS WAR DER WUNSCH. JETZT KANN MAN DAS HABEN. DIE 512K KARTE IST LEICHT IN DEN XL ZU INSTALLIEREN.

129.-

EDITOR

BILDER UND SCHRIFTEN KANN MAN LEICHT MIT DEM PROGRAMM VERARBEITEN. GUT GEEIGNET FÜR KLEINE ZEITUNGEN ODER INSERATE. DAS GANZE PROGRAMM BESTEHT AUS 10 DISKETTEN

149.-

DANABONIC DRUCKER

KXD 1080	690.-
KXD 1091	781.-
KXD 1092	1050.-

FARBLENDER

SCHWARZ	24.-
ROT	24.-
BLAU	24.-
BRAUN	24.-

GRAFIKZEICHEN

DUCCHEG	10.-
EINSTEIGER	15.-
DISK BUCH	15.-
C/65 COMPILER	30.-
FILE FAX	30.-
TELFON TERROR	35.-
MSCINE LERNEN MIT EINEM SPIEL. ALLE ROUTINEN SICHTBAR.	

IRATA

MERENDORFPLATZ 8

BERLIN 10

```

5710 POSITION 0,0: ? "
5715 SEITE 5 VON 6"
5715 ? "
5720 ? "Zu trennen sind die Operanden
und der"
5740 ? "Operator durch Strichpunkt oder
r Doppel-"
5750 ? "punkt oder Komma." BEISPIELE"
5760 ? "
5771 ? : ? "K1;K3;N - multipliziert die
Felder
5772 ? " Spalte1 mit den Feldern der
Spalte2"
5774 ? " und tr'gt die Ergebnisse in
die an-"
5775 ? " gegebene Spalte ein."
5776 ? : ? "K4;S2;+ - addiert zu jeden
Feld der "
5777 ? " Spalte4 die Spaltensumme de
r Spalte2"
5778 ? " und tr'gt die Ergebnisse in
die an-"
5779 ? " gegebene Spalte ein."
5900 ? : ? "K3;N - berechnet die Wurzel
n der Fel-"
5902 ? " der der Spalte5 und tr'gt d
ie Ergeb-"
5904 ? " nisse in die angegebene Spa
lte ein."
5906 POSITION 0,22: ? " [RECHN]-MENUE NE
LP [ZUM-PROGRAMM]"
5910 OPEN M3,4,0,"K1"
5915 GET M3,K
5920 CLOSE M3
5925 IF K=155 THEN 5970
5930 IF K=27 THEN GRAPHICS 0:GOTO 1740
5940 GOTO 5910
5970 ? CHR$(125):POKE 756,204:POKE 82,
0:POKE 752,3
5975 POSITION 0,0: ? "
5980 SEITE 6 VON 6"
5980 ? "
5980 ? "K10)2.5;/ - dividiert jedes Fe
ld der"
5990 ? " Spalte6 durch den Faktor 2
.5 und "
5992 ? " tr'gt die Ergebnisse in die
angege-"
5995 ? " bene Spalte ein."

```

```

5998 ? : ? "K3;2;^ - quadriert jedes Fe
ld der"
5995 ? " Spalte5 und tr'gt die Ergeb
nisse "
5910 ? " in die angegebene Spalte ei
n."
5911 ? : ? "
5911 ? "
5911 ? " Zum Kennenlernen des Programm
s h"
5914 ? " nen sie das Arbeitsblatt DEMO
laden"
5915 ? "
5915 ? "
5918 ? " A C H T U N G ! ! ! "
5920 ? : ? " Das Programm rundet alle Z
angaben und"
5925 ? " Berechnungen auf 2 Kommastell
en ab."
5975 POSITION 0,22: ? " [RECHN]-BECHIMM
HELP [ZUM-PROGRAMM]"
5980 OPEN M3,4,0,"K1"
5982 GET M3,K
5984 CLOSE M3
5986 IF K=155 THEN 5900
5990 IF K=27 THEN GRAPHICS 0:GOTO 1740
5990 GOTO 5910
6000 END
8000 ? CHR$(125):POSITION 4,10: ? "FALS
CHE EINGABE ! ! ! ! ! : ? CHR$(253): ? CHR$(2
53):GOTO 1740
8100 POSITION 4,17: ? "BLATT RICHT VORN
UMBER ! ! ! : ? FOR Z=1 TO 150:WENT Z
8150 CLOSE M1: ? CHR$(125):GOTO 4252
12000 AD:PEEK(00)+PEEK(07)*256:Z=0
12050 OPEN M1,0,0,"P1"
12100 FOR I=0 TO 953
12110 D:PEEK(AD):AD=AD+1:Z=Z+1
12111 IF D<=8 AND D<=63 THEN D=D+32:GO
TO 12120
12112 IF D<=64 AND D<=95 THEN D=D-64:G
OTO 12120
12113 IF D<=120 AND D<=191 THEN D=D+32
:GOTO 12120
12114 IF D<=192 AND D<=223 THEN D=D-64
:GOTO 12120
12120 PUT M1,D
12130 IF Z>29 THEN Z=0: ? M1
12150 NEXT I
12200 CLOSE M1:GOTO 570

```

Hardcopy-Programm für Seikosha GP 500AT

Mit diesem Programm können Sie eine Hardcopy vom GRAPHICS-8-Bildschirm auf dem Seikosha GP 500AT ausgeben. Das Programm ist vollkommen in Maschinensprache geschrieben, und daher erfolgt der Ausdruck sehr schnell. Im Listing wird zuerst der Maschinencode in den Bereich von 50600-506FE eingelesen, dann erscheint ein kleines Demo auf dem Bildschirm. Das Hardcopy-Programm wird folgendermaßen aufgerufen:

```
OPEN #KANAL,8,0,"P:"
? #KANAL; CHR$(27);
```

```
CHR$(27); CHR$(57)
X = USR (1536, KANAL*16)
```

Falls während des Ausdrucks ein Fehler auftreten sollte, so steht seine Nummer in der Variablen X.

Noch ein wichtiger Hinweis: Nach meinen Informationen ist der Drucker 1029 von Atari kompatibel zum Seikosha GP 500AT. Somit könnte dieses Programm auch für den 1029 zu verwenden sein.

Matthias Wunder

```

100 REM *** Hardcopy fuer Seikosha
110 REM *** GP-500AT
120 REM ***
130 REM *** (c) Matthias Wunder
140 REM
150 9=0
160 FOR A=1536 TO 1790
170 READ I:9=9+1
180 POKE A,I
190 NEXT A
200 IF B<=30910 THEN ? CHR$(253):!*"DATE
NFEHLER! !":END
210 REM
220 REM *** Kleines Demo ***
230 REM
240 GRAPHICS 24:COLOR 1
250 FOR A=0 TO 319 STEP 10
260 PLOT 140,191:DRAMTO A,B
270 NEXT A
280 REM

```



```

290 REM *** Hardcopy ***
300 REM
310 KANAL=1
320 OPEN #KANAL,0,0,"P"
330 ? #KANAL,CHR$(27);CHR$(27);CHR$(57)
1)REM Select 9 Lines per Inch
340 I=USR(1536,KANAL,16)
350 IF I>=128 THEN ? *FEHLER *IX
360 END

30000 DATA 104,104,104,141,255,6,169
30001 DATA 0,133,203,160,0,162,3
30002 DATA 109,251,6,32,206,6,202
30003 DATA 16,247,162,7,169,0,157
30004 DATA 0,4,202,16,250,133,204
30005 DATA 169,7,166,203,224,109,200

```

```

30006 DATA 1,74,133,207,169,0,133
30007 DATA 205,133,206,165,204,32,169
30008 DATA 6,165,203,32,169,6,24
30009 DATA 165,00,101,205,133,205,165
30010 DATA 89,101,206,133,206,177,205
30011 DATA 162,7,42,72,144,20,130
30012 DATA 72,56,169,6,229,204,170
30013 DATA 169,0,56,42,202,16,252
30014 DATA 133,200,104,170,24,165,200
30015 DATA 125,0,4,157,0,4,104
30016 DATA 202,16,320,230,204,165,204
30017 DATA 197,207,144,160,162,7,109
30018 DATA 0,4,32,206,6,202,16
30019 DATA 247,200,192,40,144,140,165
30020 DATA 207,201,3,240,15,24,169

```

```

30021 DATA 7,101,203,133,203,169,155
30022 DATA 32,206,6,76,10,6,169
30023 DATA 1,133,212,169,0,133,213
30024 DATA 96,133,200,130,72,169,0
30025 DATA 133,209,166,200,240,11,24
30026 DATA 105,40,144,2,230,209,202
30027 DATA 76,179,6,24,101,205,133
30028 DATA 205,165,206,101,209,133,206
30029 DATA 104,170,96,133,200,130,72
30030 DATA 152,72,174,205,6,169,11
30031 DATA 157,66,3,169,0,157,72
30032 DATA 3,157,73,3,165,200,32
30033 DATA 06,228,132,200,104,160,104
30034 DATA 170,165,200,16,7,104,104
30035 DATA 165,206,76,162,6,96,64
30036 DATA 1,65,27

```

Speed-Tape für Atari XL/XE mit 64 KByte

Nachdem Sie "Speed-Tape" eingetippt haben, starten Sie es mit RUN. Sollten Datenfehler auftauchen, bessern Sie diese aus, und starten Sie das Programm erneut. Wenn das Programm fehlerfrei gelaufen ist, dann drücken Sie RESET und speichern es ab. Der Basic-Teil des Programms kann nun gelöst werden. "Speed"-Tape benötigt das unter dem OS-ROM liegende RAM. Es bleibt solange im Speicher, bis dieses RAM beschrieben wird (z.B. durch eine RAM-Disk oder wenn man den Computer aus-

schaltet). Weder NEW noch RESET gefährden "Speed-Tape".

"Speed-Tape" wird durch POKE 54017,253 oder RESET ausgeschaltet und durch POKE 54017,252 aktiviert. Nach diesem POKE-Befehl laufen sämtliche Cassettenoperationen mit höherer Geschwindigkeit ab. Alle Befehle bleiben gleich, und es kommt auch kein neuer dazu.

Programme, die mit "Speed-Tape" gespeichert wurden, lassen sich auch nur damit laden.

Will man ein Programm auf das "Speed-Tape"-Format umwandeln, muß man "Speed-Tape" laden und starten, schaltet es dann aus und lädt das betreffende Programm. Danach muß man "Speed-Tape" einschalten und das Programm speichern.

Da "Speed-Tape" mit sehr hoher Baudrate speichert, können leicht Übertragungsfehler auftreten. Deshalb sollte man immer Rückkopien in normaler Geschwindigkeit anfertigen! Es sollte auch nur eingeschaltet sein, wenn man es benötigt. Ansonsten ist es besser.

"Speed-Tape" auszuschalten, da sonst das Betriebssystem (das ins RAM kopiert wurde, damit das System nicht abstürzt) ungeschützt ist und bei unkontrollierten POKE-Befehlen leicht abstürzt. "Speed-Tape" erhöht lediglich die Baudrate von 600 Bit/sec auf 1150 Bit/sec, der Leader und die Pausen zwischen den Blöcken bleiben gleich lang. Daher kann erst bei größerem Datenmengen eine optimale Wirkung erzielt werden (ab ca. 16 KByte).

Andreas Fritinger

Speed-Tape 1,0

```

1000 REM *****
1002 REM *
1005 REM * SPEED-TAPE 1.0 *
1010 REM * (c) by FAM 1986 *
1020 REM * fuer alle XL,XE *
1030 REM * mit 64KByte *
1040 REM *
1050 REM *****
1060 SETCOLOR 2,0,2;SETCOLOR 4,0,4;SET
COLOR 1,0,10;POKE 752,1
1070 ? "1"? :? :? :? " SPEED-TAPE 1.0 "
1080 ? :? :? "POKE 54017,252 | SPEED-T
APE EIN"? :? :? "POKE 54017,253"
1090 ? "oder RESET | SPEED-TAPE AUS"
1100 GOSUB 10000;A=USR(20400);POKE 540
17,252

```

```

1110 ? :? :? :? "SPEED-TAPE ist aktivi
ert."
1120 POKE 60323,250;POKE 60328,2;A=007
04;POKE A,169;POKE A+1,250;POKE A+2,23
4
1130 POKE A+7,169;POKE A+0,2;POKE A+9,
234
1140 A=00B16;POKE A,32;POKE A,237
3000 POKE 752,0;END
10000 REM ---ROM INS RAM KOPIEREN---
10010 CHK=0;RESTORE 10100
10020 FOR I=20400 TO 20592;READ D
10030 IF D<255 THEN IF CHK<0=1000 THE
N ? "DATENFEHLER IN ZEILE"? :GOTO 10060
10040 IF D<255 THEN CHK=0;REST D
10050 CHK=CHK+0;POKE I,D;NEXT I;READ D
10052 IF D=1000;CHK THEN 10030
10054 RETURN

```

```

10060 ? PEEK(103);PEEK(104)+256;END
10100 DATA 104,120,169,0,141,14,212,16
9,252,141,1,211,169,192,2095
10110 DATA 133,204,169,207,133,206,169
0,133,203,133,205,160,3005
10120 DATA 255,169,253,141,1,211,177,2
03,72,169,252,141,1,211,3256
10130 DATA 104,145,203,136,192,255,200
2,235,230,204,165,204,197,3470
10140 DATA 206,200,225,169,216,133,204
,169,0,133,205,169,0,133,3170
10150 DATA 203,160,255,169,253,141,1,2
1,177,203,72,169,252,141,3407
10160 DATA 1,211,104,145,203,136,192,2
55,200,235,230,204,165,3209
10170 DATA 204,197,200,200,225,169,253
,141,1,211,169,64,141,14,3202
10180 DATA 212,00,96,0,1396

```


Zeichensatzfinder

Neue Zeichen auf dem Atari – um dieses Thema drehen sich in unserem Heft schon mehrere Artikel, so zum Beispiel die beiden bisher veröffentlichten Zeichensatzeditoren. Bei dem Hilfsprogramm unseres Stammlesers Norbert Hagemann geht es jedoch nicht darum, eigene Fonts Zeichen für Zeichen zu basteln, sondern fertige Zeichensätze auszuhauen, wie sie in vielen professionellen Programmen enthalten sind.

Als erstes sollten Sie sich eine DOS-II-Masterdisk erstellen, indem Sie eine Diskette formatieren und mit der DOS-II-Menü-Option "H" das DOS darauf schreiben. Tippen Sie dann das Programm fehlerfrei ein und speichern Sie es sicherheitshalber gleich einmal ab. Wenn Sie jetzt die neue Systemdisk einlegen und RUN eingeben, so sollte außer der Meldung "OBJ-File wird generiert..." eine immer länger werdende Reihe aus Sternchen erscheinen. Wird ein fehlerhaftes Datum entdeckt, so wird die entsprechende Zeile reklamiert. Wenn schließlich die O.K.-Meldung das Ende signalisiert, befindet sich das File "AUTORUN.SYS" auf der Diskette. Wenn Sie jetzt den Computer aus- und (möglichst mit gedrückter OPTION-Taste) wieder anschalten, so wird der Zeichensatzfinder automatisch geladen und gestartet. Das

Programm meldet sich mit einem schwarzen Bildschirm, auf dem unter der Copyright-Laufschrift ein graues Raster und ein vier Punkte umfassendes Menü erscheinen.

Versuchen wir als erstes Punkt 1: Speicher durchsuchen. Der Computer stellt jetzt den Speicherbereich ab Adresse \$E000 im Raster dar. Es handelt sich dabei um den Systemzeichensatz. Mit den Tasten "<" und ">" kann der dargestellte Bereich jetzt um 128 Byte verschoben werden. Die dargestellte Adresse wird dabei angezeigt. Wenn Sie zum Beispiel 40 mal (für 20 Pages) ">" drücken, so erscheint der internationale Zeichensatz, in dem auch die deutschen Umlaute enthalten sind. Als Adresse wird hier \$C000 ausgegeben. Wenn Sie jetzt A und E000 eingeben, so erscheint wieder der ROM-Font. Sie haben also manuell eine neue Adresse eingegeben. Mit RETURN übergibt Ihnen Sie die Adresssteuerung wieder den angegebenen Tasten. Außer diesen beiden Zeichensätzen werden Sie normalerweise wenig Glück bei der Suche haben, es sei denn, Sie hatten zuvor ein Programm im Speicher, dessen Zeichensatz noch vorhanden ist. Dies ist jedoch nur dann möglich, wenn Sie den "Zeichensatzfinder" nicht durch Booten starten, sondern umbenennen (z.B. in

FINDER.COM). Sie müssen dann, nachdem Ihr zu untersuchendes Programm unterbrochen wurde, ins DOS gehen und den "Zeichensatzfinder" manuell mit der "L"-Option laden.

Machen Sie sich am besten zuerst mit dem Erscheinungsbild eines Zeichensatzes vertraut. Im obersten, linken Kästchen steht das Zeichen, das als Leerzeichen verwendet wird. Ist hier ein anderer Buchstabe abgebildet, so sollten Sie mit "-" und "+" blättern. Sind die Buchstaben über die Kästchen hinweg verschoben, so helfen "+" und "-". Mit ESC kommen Sie wieder zum Hauptmenü zurück.

Versuchen Sie jetzt einmal, Funktion 2: Diskette absuchen. Hier fragt das Programm nach der Sektornummer, wobei 0 Default ist. Sollten Sie über eine Bootdiskette verfügen, auf der sich ein Programm mit eigenem Zeichensatz befindet, so können Sie die Diskette mit dieser Option nach Zeichensätzen durchsuchen, wobei die oben erwähnten Tasten nun dazu dienen, um 8, 4 oder 1 Sektor(en) vor- bzw. zurückzublätern. Nach 8 kann auch eine Sektornummer manuell eingestellt werden. Sollten bei manchen Sektoren Lesefehler irgendeiner Art auftreten, so deutet dies nicht immer auf ein defektes Programm hin – wahrscheinlich haben Sie den Kopierschutz erwischt.

Programme, die sich als File auf Diskette befinden, kann man mit Option 3 untersuchen. Sie werden unter Angabe des Namens in den Speicher geladen und können dort durchsucht werden – genau wie bei Punkt 1. Sollten Sie den Inhalt der Diskette nicht im Kopf haben, so genügt die Eingabe von RETURN, um ein Inhaltsverzeichnis zu bekommen, wobei dies mit einer beliebigen Taste fortgesetzt werden kann. Denken Sie daran, daß Sie bei Basis-Programmen nichts finden können. Die Zeichen sind hier normalerweise in DATA-Zeilen gespeichert, die erst nach RUN in den Speicher gepokt werden.

Ist es Ihnen gelungen, mit einer der drei obigen Optionen einen Zeichensatz auf den Bildschirm zu bekommen, so können Sie ihn nun mit der vierten und letzten Funktion abspeichern. Geben Sie einfach einen Namen ein. Wenn Sie sich erst den Inhalt der Disk ansehen möchten, so können Sie auch hier RETURN verwenden. Der Zeichensatz kann nun von 99% aller Zeichensatzeditoren eingelesen und weiterverarbeitet werden.

Mit dem Hilfsprogramm "Loaddemo" können Sie sich den "entfärbten" Zeichensatz ansehen. Ersetzen Sie "ZEICHENSATZ" durch den gewünschten Namen und starten Sie mit RUN.

Norbert Hagemann und
Thomas Tzouard

MC-Generator

```

10 REN
11 REN ----->
12 REN |
13 REN | Zeichensatzfinder |
14 REN |
15 REN | fuer alle ATARI-Computer |
16 REN |
17 REN |----->
18 REN |
19 REN | (c) June 1986 by |
20 REN | Norbert Hagemann |
21 REN | Behnstrasse |
22 REN | 1000 Berlin 65 |
23 REN | Tel. 830-493 88 92 |
24 REN |
25 REN |----->
26 REN | 26 REN | nur die RETURN-Taste ge- |
27 REN | Mit diesem Programm kann |
28 REN | man Zeichensätze suchen |
29 REN | die in anderen Programmen |
30 REN | verwendet werden. |
31 REN | Wurde ein Zeichensatz ge- |
32 REN | funden, so kann dieser auf |
33 REN | der Diskette abgespeichert |
34 REN | werden. |
35 REN | Das Programm hat folgende |
36 REN | Optionen: |
37 REN | |
38 REN | [1] Speicher absuchen |
39 REN | [2] Diskette absuchen |
40 REN | [3] Programm absuchen |
41 REN | [4] Datei abspeichern |
42 REN |
43 REN | Wird bei der Option 3+4 |
44 REN | anstelle des Filenamens |
45 REN |
46 REN | drueckt, wird der Disk- |
47 REN | Inhalt angezeigt. |
48 REN |
49 REN |----->
50 REN |
51 REN |
100 DIM D$(95):OPEN #1,0,0,"AUTORUN.SYS"
110 ? "OBJ-File wird generiert...."?
:TRAP 190:0=0
120 READ D$:READ P
130 ? "*"
140 FOR I=1 TO LEN(D$) STEP 2
150 M=ASC(D$(I))-40:I=ASC(D$(I+1))-40
160 D=(M-(M/9)*7)*64-(I/9)*7:0=0:P
UT #1,DIF 9:999 THEN 0=5-1000
170 NEXT I:IF 0=P THEN 120
180 ? : ? ? "Datenfehler in Zeile "PE

```


1 **Ohrschmaus:
Soundmachine**
Verknüpfung 10 Hilfkürven, Schiebereg. bis zu 5000 Notizen, auch vorprogrammierten Programmen nutzbar. Eingabe über Tastatur oder Joystick. Mit Demoverseil 2 Diskettenversion, ausführliches Handbuch.
ATARI 400 - 130 XE, ab 48K
Best.-Nr. AT 1 29,80 DM

2 **Lehrreich:
Das Assemblerbuch**
Klare Einblicke in Zahlensysteme, in Aufbau und Befehlsatz des 6502, in Programmierung der Custom-Chips. Pages-Memorie-Graph und Intermem-Technologie sowie in einigen anderen Kauderwelsch. Deutsch für Einsteiger geeignet. Lösung für ATMAS II Assembler. 196 Seiten DIN A5, druckt bei CK.
Best.-Nr. AT 10 29,80 DM

3 **Vielfalt:
Atari Power Superbuch**
Bauanleitungen, Lösungen, Tips & Tricks... 75 Seiten DIN-A4, nicht im Buchhandel erhältlich!
Best.-Nr. AT 3 29,- DM

4 **Nachschlag:
Die Hexenküche**
Aufschlüsselung für Ein/Ausstieger und Profis gleichermaßen: Tips & Tricks, Kniffe, Drehs etc. Maschinen-sprache-Programme als Listings. Tausend Fragen Atari ganz selbst an (und Sie auch!)
Best.-Nr. AT 4 29,80 DM

5 **Zuschlag:
Disk zu Hexenküche**
Dabei kann man viel Zeit sparen.
Best.-Nr. AT 5 19,80 DM

6 **Fix: ATMAS-II
Macro Assembler**
4 K Quelltext in 4 Sekunden assembliert! Erzeugung von Editiermodulen, Full-Screen Editor, sofort in hoch. Reichtext, integrierter Monitor. 50 seitiges Handbuch und Disk im Ringorder.
ATARI 400 - 130 XE
Best.-Nr. AT 6 Diskette 49,- DM

7 **Geistsparend:
ATMAS Toolbox**
Rechenroutinen I/O Makros, Cursorzugr., Fast-circle, Scrolling und noch einiges mehr. Auf Diskette mit Anleitung dabeist
ATARI 400 - 130 XE, ab 48K
Best.-Nr. AT 7 19,80 DM

8 **Praktisch: Monitor XL**
Verknüpfte Basis-Programme mit Mode-Routinen: eingeben, korrigieren, kopieren, Single-Step, Disk-Innen-spezifischer Directory-Anzeige, deutsche Fehlermeldungen auch für Basic und DOS. Der Basis-Speicherplatz bleibt unberührt, Auflöser und Disk.
ATARI 600 XL (64 K) / 800 XL / 130 XE
Best.-Nr. AT 8 19,80 DM

9 **Brandneu: Design Master**
Bedienung über Fenster-Technik, Auflösung 320 x 192, Fadenkreuz, Maßstabregler ein/ausblenken, 2 Screens gleichzeitig, über 122.000 Punkte im Druckauswurf, über 100 verschiedene Schriften, Hardcopy Drucker alle Matrix-Drucker (ab 8 Nadeln), Ausdruck in verschiedenen (3) Bytes möglich, ausführliche deutsche Anleitung.
ATARI 600 XL (64 K) / 800 XL / 130 XE
Best.-Nr. AT 9 Diskette 19,80 DM



CK Spezialitäten-Bestellschein

Name	Best.-Nr.	Stückzahl
AT 1	Soundmachine	
AT 10	Das Assemblerbuch	
AT 3	Atari Power Superbuch	
AT 4	Die Hexenküche	
AT 5	Disk zur Hexenküche	
AT 6	Macro Assembler	
AT 7	ATMAS Toolbox	
AT 8	Monitor XL	
AT 9	Design Master	

Ich wünsche folgende
 Bezahlung: Nach Prüfung 4,99 DM Versandkosten Vorkaufzahlung
 Bitte geben Sie mir die Vorzugsliste aller Bücher (ausgabe) oder auf Preisänderungen (Sonderheft) 42x10-156 zuversenden.

Name des Bestellers _____ FOLIO _____
 Anschrift _____
 Coupon direkt/Friedrich, auf Postkarte kleben und einstecken an:
Computer Kontakt, Softwareversand, Postfach 1640, 7518 Bretten.

```

1578 DATA 449832835E44C833C80100C1E
A0B4A3F98B084B598B3E4E85C3E8A83D88E3E
88B3CA4D02F8F28F4E28, 394
1580 DATA 48428754C28F3482B4482BFF4C
A2B7D08E49F80FC18F82843484C8F4AB8A8A
8D324582888D334585A16C8A, 227
1590 DATA 88A22C28E3428754C2834485AF
A4885A28A4328754C28FF4CC918F8A4CC99F8
3EC93888C947888C94188, 324
1600 DATA 8AC93A98862840484C45A438E938
C98A98E2987A88884A3A88A86A226A3888F9
85A285A2284448A5A2A4A385, 291
1610 DATA AF84884C5A428F4A188C84C9649
28F6A8988A98885F85884C986A9B12CA984
2CA98A88888A18685AF85A2, 445
1620 DATA 981EEA3AC88A4A9812CA9842CA9
8885A44888A3885AF85A3A85A2D4A4CF4888
823CA328F6A8813A5A8A8A, 334
1630 DATA 85A9888A88881F8828754CA8A
492840484C8F4A85A2888A4A3F888A8A3C88A
9888888C9118882188A388, 324
1640 DATA A982A888888888989898983A987D
4283AC56E48FF851485148FC88A928186A88
A88A9882E6A1A8A928888888, 25
1650 DATA A888282148C810288C888888F8F8
12A93C2CA98F88882A9A888132A9F282348
888882888132A881C1A191715, 96
1660 DATA 131218888C888897858828838187
885818785828187858381878578786868588
4848383828181888888A68, 582
1670 DATA A22885A888A1A5A8A8A885A286A3
A98885A898885A8A28F8A18D9A4A599883888
88F78C83A881A285A6F8888C, 454
1680 DATA 7348A9888A62A8888F8A8C34819
D848C3A8838898888A85A6198138988138CA18
D8A818D8988891A88888F818, 283
1690 DATA A5A8672885A8882E6A18E8A28882
E6A3CA4888A18A5A86A6885A889882EA118A5
A2697885A29882E6A3C8A88, 393
1700 DATA 8C8A8728A22885A88A1A98985A2
8812A28A881F99A4A591A88818F8282A8CA88
F8A81E98891A888A9FF91A8, 345
1710 DATA 8888F8C88891A8282A88C4A2885
88A837A5A2854C8A5A48A4A4A4A286A4C48
2A8FC8A3882E98A6988988, 244
1720 DATA 1FC88A893A588286A4CA85A4C5D
8C8A4A89885A885A1F8A888886A4A28185A875
A8F5A8CA18F888F888A8A2, 175
1730 DATA 18A98C9D42832856E468A286284A
48284D48C8A8F78A218A9839D4283A9F29D44
83A94A94583989D4A83A988, 562
1740 DATA CC4CC74D948834C5E4A8988A3C
85A88A1A5A8A4A828D448218A98898A4283A9
889D4A83A9C9D4583A98899, 474
1750 DATA 4883A98A9485A4C5E4A28E4CA2
18A9839D4283A989D4A83A94A9D4583A98A98
4A83A98A9D4883285E4A8A88, 488
1760 DATA 2818A48288E4C686828148A27E
A84A9889D883A9D883A9883A9883A9883A9883
F8A8A9FF2CA988482828A88, 621

```

```

1770 DATA AA188D888889C8A19881FC888F7
F889D38439D881F888F7A988A1F88A4A88C
684828E4E8888A9D9F98934, 753
1780 DATA A9F998935A9FF299936A9F8989
3788889A817A9D899A3A989999A5A9FF99
9A3A9F999A378818E9A9F, 423
1790 DATA 8D993A88C2A3A9F889935888235
A9FF8D993A88823A9F889993788823788C84D
C3A4C88888738E928188269, 148
1800 DATA 8A8885A28A4A3A28A88881A291A8
C888F8E6A1E8A3C8A8F2A8A888C8F8C2199781F
89D4A299981F8E7429981F, 644
1810 DATA 8818E8A88888F44299921FCA8818
F688A4A4A22328E40A4A28814C8A98286A4C
A5A1285D4C28A94CA9C82823, 888
1820 DATA 48688888A898983A4A83988388
8818F3A9FF88823A8882388823888238889
3A88893888936889378828, 252
1830 DATA 28A881889384A99821F89574419
221F99734499A21F898A4A9621F99821F99A2
1F99C21F89A84A99E21F8818, 935
1840 DATA D8C88C11F8C8F1288E4E8A8C88
893A9888883888888888888888888888888888
3A88C1388C1388C137A889, 689
1850 DATA A9F99893A9FF9993599893A9
E9F99C4A8E8F478818E8A888A9F89983A9A9
8F99E3378818F3682858E8A9, 533
1860 DATA 8888128364828E4CA218A9839D
4283A989D4A83A94A93A58A98A88A8A88A988
9D4883285E4A831A988A88C, 589

```

```

1870 DATA 85A88A1A218A5A8D94A83A5A19D
4583A9149D4883A98894983A9859D4283285E
4C8888A88828C48E8A818887, 258
1880 DATA 8A4A28E4C8A4A8888888F837A2128D
C7448871FCA18F728814C8A9A5A8286A4C45
A1285D4CA98888F188781F, 532
1890 DATA 8DAF1F8881F8C8F1F8881F28A9
4A8A42832A88A8C613882888A88898A4499
891F881F87388A5D185A4A9, 924
1900 DATA 82A83C85888A1A96A881FA28785
A28A4A3A8881A8C88A8D91A28818F18A5A27D
6A4585A29882E6A3A98D28C, 215
1910 DATA 48C6A8C8A988F884F13CA188C283A
4828FF4CC91F888C2858A8C97A82848828FF
4C688882E182A8A5, 738

```

Beispiel

```

100 POKE 186,PEEK(186)-4:GRAPHICS BURE
STORE 148
110 DIM Z(29):FOR I=1 TO 29:READ D:Z(I)=CHR$(I):NEXT I
120 CLOSE #1:OPEN #1,4,8,"*:ZEICHENS.A
T2":Z=USR(ADR(Z)):CLOSE #1
130 POKE 156,PEEK(186)+81
140 DATA 188,162,16,169,7,157,66,3,165
,186
150 DATA 157,69,3,169,8,157,68,3,157,7
,2,3
160 DATA 169,4,157,73,3,76,86,228

```

*** BASIC-Zauberer ***

Die Super-Hilfsdiskette für Ihren ATARI!

Über 48 Files helfen Ihnen bei vielen Programmierproblemen - jeden Tag!

BASIC-Zauberer - das sind:

- * BASIC-Erweiterungen wie
 - AutoNumber
 - REM Killer
 - REM Marker & -Entmarker
 - COLOR-Maker
 - HEX, BIN->DEC -Wandlung
 - Block-Zeilen-Delete
 - Variablen-Lister...
- * Tolle Hilfsprogramme wie
 - Zeichensatzeditor
 - Farb-Zeicheneditor
 - TT-Titelmaschine
- * nützliche MS-Programme wie
 - Zeichensatz LOAD/SAVE/COPI
 - 128-Farben-Effekt
 - Bilder-Laderoutine

und vieles mehr! nur DM 29,-

Sie bestellen Sie per Nachnahme oder Scheck bei



Thomas Tausend, P.O. 17
 Tel.: (08376) / 1667
 8951 Sulzberg



Profi Painter – für den Atari ST

Auch das jüngste Kind der Data Becker-Erfolgsreihe wurde auf den Atari ST umgesetzt. Einschränkend muß aber gleich gesagt werden, daß der Autor die beiden Vorbilder (CPC und C64) nicht kannte und auch einen anderen Namen vorgesehen hätte. Weder Programmidee noch die grundsätzlichen Funktionen sind also abgekupfert. Vielmehr entstand erwartungsgemäß ein Programm, das von den Möglichkeiten und Fähigkeiten her jeden Konkurrenten auf einem 8-Bitler mit links aus dem Felde schlägt.

daß 1 MByte dazu ausreicht. Aber es ist ja noch nicht aller Tage Abend. Denn wie der Hersteller verstanden ließ, sind in Zukunft noch mehrere überarbeitete Versionen zu erwarten. Da aber noch weitere Schwachpunkte im Programm zu finden sind, die wir im folgenden noch näher betrachten werden, hätte man lieber noch ein paar Monate warten sollen, um dann sofort ein ausgereiftes Produkt auf den Markt zu bringen.

Nach der beschriebenen Voreinstellung offenbart sich ein

In der obersten Zeile finden wir schließlich wie üblich die Menüs von Profi Painter, mit denen wir uns gleich noch genauer befassen werden. In der Mitte thront das Zeichenfenster, das beliebig vergrößert und verkleinert werden kann, wie wir es schon aus der Diskettenverwaltung gewohnt sind.

Die freie Wahl des Bildformats hat auch einen ganz konkreten Hintergrund, denn mit dem Programm kann man sage und schreibe drei Bilder gleichzeitig bearbeiten. Mit den richtigen Einstellungen sind die Kunstwerke auch gleichzeitig sichtbar, und mit den Blockoperationen können Objekte zwischen den Files ausgetauscht werden. Dies erweist sich im praktischen Einsatz schnell als sehr nützlich, denn oft sind in einem Bild Objekte so gut gelungen, daß man sie auch in Zukunft weiter verwenden möchte.

Diese Blockmanipulationen beginnen zunächst mit der Definition des gewünschten Bildschirmbereichs, was mit dem bekannten Auswahlrahmen realisiert wird. Nun kann man den Block, der immer eine quadratische Form hat, löschen oder auch versetzen, indem man ihn einfach mit dem Mauszeiger an die gewünschte Stelle schiebt. Die ruckhaften und flimmernden Bewegungen des Bildschirmbereichs stören dabei allerdings sehr. Selbst klei-

ne Verschiebungen bewirken ein Flimmern, das ein genaues Arbeiten fast unmöglich macht und dem Anwender eigentlich auch nicht zugemutet werden kann. Der Grund für diese und andere Geschwindigkeitseinbußen sind sicherlich in der Tatsache zu suchen, daß Profi Painter in "C", also einer Hochsprache programmiert wurde, und dort wirken sich eben die Geschwindigkeitsunterschiede zum reinen Maschinencode verheerend aus.

Doch zurück zu den Blockoperationen. Der definierte Bereich kann natürlich auch kopiert und an beliebiger Stelle neu eingesetzt werden. Diese Stelle kann, wie bereits erwähnt, auch in einem anderen Fenster liegen, das gleichzeitig eröffnet wurde. Wenn diese Form der Objektübertragung zwischen zwei oder mehreren Files zu umständlich ist, oder kann den Block auch unter beliebigen Namen auf Diskette abspeichern und jederzeit in einem anderen Bild wieder einladen.

Neben diesen ganzen Möglichkeiten des Versetzens, Löschsens und Kopierens kann auch der Block an sich verändert werden. Invertieren, horizontales und vertikales Spiegeln sowie das Drehen um 90 Grad sind auf Maus-Knopfdruck realisierbar.

Die Zeichenfunktionen werden in der linken Leiste durch Symbole dargestellt. Selbstre-



Auswählen des Formates

Profi Painter richtet sich hauptsächlich an die Maler und Künstler unter den Computern. Technische Zeichnungen, Formulare usw. sollten von "Easy-Draw" oder "Gem-Draw" erstellt werden. Für Bilder aller Art, Plakate oder auch Hintergründe von Spielen ist der Einsatz des vorliegenden Programms jedoch ein Muß.

Nach dem Laden von Profi Painter kann der Anwender zwischen zwei Formaten wählen, in denen sein Bild entstehen soll. Man wundert sich allerdings, daß außer DIN A4 Hoch- und DIN A5 Querformat keine weiteren Formate möglich sind. Von einem Programm, das auf so einem schnellen und reichlich mit Speicherplatz ausgestatteten Computer läuft, kann man wenigstens noch DIN A4 quer und eigentlich auch DIN A3 quer verlangen. Man sollte meinen,

jungfräulich weißer Bildschirm, der von diversen Symbolen und Menünamen umrahmt wird. Das erste spontane Urteil: Optisch sehr ansprechend und sorgfältig aufgebaut. Auch beim praktischen Arbeiten entpuppt sich die Bildschirmaufteilung als sehr sinnvoll. Die rechte Leiste zeigt Symbole mit den Mal- und Zeichenfunktionen, die verschiedenen Strichstärken von GEM, den Maßstabs und das aktuelle Muster an, das in der linken Leiste geändert werden kann. Hier sind alle zur Verfügung stehenden Füllmuster dargestellt, wobei den erfahrenen Freaks sofort auffällt, daß nur die Standardmuster von GEM vorhanden sind. Große Enttäuschung macht sich gar breit, wenn deutlich wird, daß das Erstellen eigener Füllmuster nicht möglich ist. Dies ist sicherlich ein sehr gravierender Fehler, den der Konkurrent "Degas" nicht aufweist.



3 Bilder parallel bearbeiten

dead kann Profi Painter Kreise und Rechtecke mit oder ohne gerundete Ecken in beliebiger Form, an beliebiger Stelle und auf Wunsch mit den aktuellen Mustern gefüllt darstellen. Das Programm kann Linien ziehen und erlaubt auch das Freihandzeichnen. Dabei unterscheidet man aber noch zwischen Pinsel und Bleistift. Letzterer soll für feine Arbeiten eingesetzt werden, während der Pinsel in 24 verschiedenen Formen zur Verfügung steht und die größeren Vorarbeiten übernimmt.

Weniger selbstverständlich, aber ein Fast-Standard, ist die Sprühdose für die friedlichen Graffiti am Computer. Es stehen drei davon in verschiedenen Größen zur Auswahl. Deren Funktion läßt jedoch zu wünschen übrig. Bisher Verstand man unter sprayen, daß sich unter dem Mauszeiger Punkte bilden und sich die einzufräbende Fläche wie beim Einsatz einer richtigen Spraydose durch kontinuierlichen Tastendruck immer mehr verdichtet. Profi Painter füllt hier aus der Reihe. Der Tastendruck bewirkt nur den einmaligen Abdruck des Spraymusters. Weitergehende Effekte müssen durch Hin- und Herbewegen erreicht werden, was allerdings ein wenig verschmiert und unsauber aussieht. Daher ist der Sinn der Spraydose ein wenig fraglich, läßt sich derselbe Effekt doch mit dem Pinsel der entsprechenden Form erzielen.

Daneben gibt es noch das Füllen von Flächen und die etwas zweifelhafte Radiergummifunktion, mit der sich nur sehr große Teile der Zeichnung entfernen lassen. Feinarbeit wird mit der Lupe gemacht, und hier bietet Profi Painter eine nachahmenswerte Neuerung. Nach dem Anklicken des Lupensymbols erscheint in der linken oberen Ecke der Rahmen für den vergrößerten Bildteil und in der linken unteren Ecke das Fenster mit der Vergrößerung. Der Rahmen kann nun beliebig verschoben und gesetzt werden; im Lupenfenster scrollt die Pixel-darstellung mit. Sitz das Lupenfenster nicht ganz richtig, kann es wie ein ganz normales Window versetzt werden. Ist der richtige Ausschnitt erreicht, können die Pixel einzeln gesetzt

und gelöscht werden, wobei die Änderung auch gleich im Gesamtbild sichtbar wird.

Eine weitere Besonderheit, die vollkommen aus dem Rahmen fällt, ist die Textoption. Die Möglichkeiten, die sich hier eröffnen, sind faszinierend. Neben acht Größen stehen die Schrift-Typen fett, outlined, unterstrichen, kursiv und hell zur Verfügung, die man natürlich auch noch untereinander mischen kann. Außerdem ist das Laden von sechs weiteren GEM-Zeichensätzen möglich, zwischen denen man ganz einfach hin- und herschalten kann. Schier unglaublich ist, daß der Text in vier verschiedenen Richtungen geschrieben werden kann. Beschriftungen auf dem Kopf oder quer sind für das Programm kein Problem.

Erwähnenswert wären noch die vier Zeichenmodi, in denen man arbeiten kann. Gezeichnetes wird von DECKEND gadenlos überdeckt, TRANSPARENT läßt darunter liegendes durchschimmern, XOR führt eine XOR-Verknüpfung der beiden Flächen durch, und bei INVERS zeigt sich Weißes schwarz und Schwarzes weiß.

Nachdem ein Bild mit dieser Fülle von Funktionen erstellt wurde, soll es natürlich auch abgespeichert und nach Möglichkeit ausgedruckt werden. Beim Speichern stehen zwei Formate zur Verfügung, die nach Belieben eingestellt werden können. Auf diese Weise sind Profi Painter-Bilder ohne Probleme kompatibel zu "Degas" und "Doodle", so daß man beispielsweise ein Bild mit "Degas" weiterbearbeiten kann, indem man ein selbstdefiniertes Muster einfügt.

Der Ausdruck erfolgt entsprechend der Einstellung des Blattformats maßstabsgetreu. Dabei bedient sich Profi Painter einer Druckerdatei, die man jederzeit mit einem Texteditor ändern kann. Hier werden die Grafikcodes einzeln angegeben, so daß auch der No-Name-Drucker aus Hong Kong mit den entsprechenden Einstellungen Grafik ausspuckt.

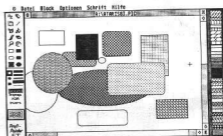
Profi Painter besitzt also einige beeindruckende Funktionen, wie die Lupe und die Textoption, die in dieser Form ein-

malig und vorbildlich sind. Dagegen ist die Geschwindigkeit an einigen Stellen nicht besonders hoch, auch eine freie Musterdefinition ist nicht möglich.

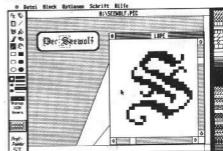
Man vermißt mehrere Blattformate, und auch das Handbuch verrät nur das, was der Anwender unbedingt wissen muß. Wer mit diesen Mankos leben kann, darf sich an einem optisch ansprechenden und in der besprochenen Weise sicherlich leistungsfähigen Programm erfreuen. Den anderen Anwendern sei empfohlen, auf neue Versio-

nen zu warten, denn neben vielen Verbesserungen soll auch noch ein Profi Painter in Farbe erscheinen. Trotzdem, Profi Painter kostet 99,- DM und bietet für einen knappen Hunderter relativ viel. Die leistungsfähigere Konkurrenz (z.B. "ST Art") ist schon im Anmarsch, wird aber sicherlich um einiges teurer sein.

System: Atari ST mit SM 124
 Hersteller: Data Becker
 Preis: 99,- DM
 Bezugsquelle: Data Becker
 Thomas Tai



Verschiedene Formen und Füllmuster



Lupe und Auswahlfahnen (gestrichelt)



Schrift-Beispiele

Dem Geld auf der Spur

Ein Programm für den Atari ST

Besonders Computerfreaks kennen dieses erschreckende "Wo ist nur mein Geld geblieben"-Gefühl. Am Anfang des Monats sah es noch recht gut aus, und nach zwei Wochen bahnt sich eine chronische Ebbe in der Kasse an. Ein Weg, dieser Entwicklung entgegenzuwirken, ist sicherlich, das schwer verdiente Geld an eine Vertrauensperson (Ehefrau, Kinder etc.) weiterzugeben, die sicherlich nie im Leben einen Pfennig zu viel ausgeben würden. Wer allerdings kein "Vertrauen" zu irgend jemandem hat (bei Geld hört ja bekanntlich die Freundschaft auf), wird den Computer zu Rate ziehen. "ST-Heimfinanz" ist nun ein komfortables Programm, mit dem die laufenden Ausgaben überwacht, in einer monatlichen Übersicht festgehalten und über einen längeren Zeitraum statistisch ausgewertet werden können.

Der Beginn der Arbeit mit dem Programm ist von diversen Voreinstellungen gekennzeichnet, die auf der Diskette abgespeichert und nie wieder erneuert werden müssen. Neben dem Benutzernamen, dem aktuellen Kassendatum sowie dem Monat und Jahr, in dem die Überwachung beginnen soll, muß man ein Passwort eingeben, das die Daten vor fremdem Zugriff schützt. Vergessen darf man den Code allerdings auch nicht, denn dann kommt man selbst nicht mehr an sie heran. Aber das ist nun mal der Tribut, den man der Datensicherheit zollen muß. Es ist erstaunlich, wieviel Mühe der Programmierer in den Schutz der Angaben gelegt hat. Die Daten sind auf Diskette so abgelegt, daß ein Manipulieren über Diskmonitor oder sonstige Hilfsmittel nahezu unmöglich ist.

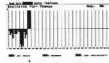


Bild 1: Monatsbilanz

Nach den Voreinstellungen müssen die einzelnen Ausgabenposten benannt werden, die regelmäßig oder häufig anfallen. Um die Übersicht zu behalten, sind nur 27 Posten möglich, so daß man gezwungen ist, die Ausgaben in größeren Gruppen



(z.B. Nahrungsmittel) zusammenzufassen. Haushalte, die ihr Geld häufig in sehr verschiedene Dinge stecken, werden dies vielleicht zunächst bemängeln, doch bei der Erstellung von Statistiken wird man schnell vom Gegenteil überzeugt, da hier eine sinnvolle Auswertung nur mit einer begrenzten Anzahl von Ausgabenposten möglich ist.

Nun kann man die harten Zahlen den jeweiligen Posten zuordnen. Leider liegt bei dieser Eingabe einer der gravierenden Schwachpunkte von "ST-Heimfinanz". Zunächst muß die jeweilige Ausgabe angeklickt werden, dann erscheint noch einmal ein Fenster, in dem die Eingabe von Geldbeträgen bestätigt wird und erst dann kann man wirklich loslegen. Besser wäre es gewesen, nur das entsprechende Feld anzuklicken und die Eintragung gleich auf dem Kontenblatt zu vollziehen. Bei einem Arbeitsschritt wie diesem, der sehr häufig genutzt wird, ist das Durcharbeiten durch Untermeritis, und sei es nur ein einziges, sehr lästig. Auch die Zahleneingabe ist nicht sonderlich komfortabel gestaltet. So müssen

Zahlen ohne Nachkommastellen trotzdem ein ".00" enthalten. Ansonsten wird die Eingabe nicht akzeptiert. Auch dies ist ein Fall, der häufig eintreten wird und lästige Tipparbeit beschert, die nicht sein müßte. Vorteilhaft bei der Eingabe ist aber der sog. "Limitbetrag", den man vor der Arbeit mit dem Programm eingibt. Wird dieser Wert mit einer Ausgabe überstiegen, warnt "ST-Heimfinanz" mit einem Alert-Window. Wie nützlich diese Warnung ist, ist eine andere Sache.

Die Arbeit mit "ST-Heimfinanz" gestaltet sich nun so, daß der Benutzer von Zeit zu Zeit die ausgegebenen Beträge eintippt, zum Monatsende einen Abschluß macht und einen neuen Monat anfangt, der wieder separat abgespeichert wird. Dabei ist es sehr nützlich, daß regelmäßig anfallende Ausgaben, wie Miete, Raten oder Beiträge, in eine automatische Monatsbuchung aufgenommen werden können. Zu Beginn jeden Monats klickt man "Fixkosten buchen" an, und die Beträge, die sowieso jeden Monat anfallen, werden automatisch abgebucht.

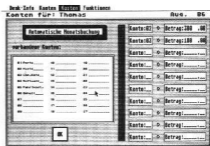
Nun reicht es natürlich nicht, die Ausgaben und Einnahmen fein säuberlich in den Computer einzugeben. Viel wichtiger ist die Auswertung, mit der man ja auch erst dem sprichwörtlichen "Loch im Geldbeutel" auf die Spur kommt. Die Auswer-

tung mit "ST-Heimfinanz" kann in tabellarischer und grafischer Form auf Bildschirm und Drucker ausgegeben werden. Dabei wird der Stand eines gewählten Monats mit dem Vormonat verglichen, ein Jahresdurchschnitt ermittelt, die Kassendifferenz zwischen Beginn und aktuellem Zeitpunkt und die Ausgaben und Einnahmen insgesamt errechnet. In einer Tabelle hat man zwar die genauen Werte für die Ausgaben, doch übersichtlicher und optisch ansprechender ist eine Grafik. Das Programm gibt sie auf dem Bildschirm aus, aber auch eine Hardcopy ist möglich. Die Ausgabenposten in Tabelle und Grafik können nach Namen, Beträgen und Nummern sortiert werden.

"ST-Heimfinanz" ist als komfortables und leicht zu bedienendes Programm für den Hausgebrauch einzustufen. Umfangreiche Kalkulationen lassen sich mit dem Programm natürlich nicht erstellen, aber das verlangt ja auch keiner. Beachtet werden müssen nur einige Bedienungsclippen, die einem mit der Zeit sehr schnell auf den Wecker fallen werden.

System: Atari ST
Bezugsquelle:
Ingenieur-Büro
Harald Zoschke

Thomas Tai



Fixkosten buchen

Das große LOGO-Buch zum Atari ST

Von Dr. Gerhard Sauer
Data Becker Verlag
297 Seiten, 49.- DM
ISBN 3-89011-122-X

LOGO ist in Amerika schon sehr lange als leicht zu erlernende Programmiersprache auch



für Kinder bekannt. In Deutschland fehlte bisher der richtige Durchbruch. Dennoch lieferte Atari zum ST zunächst LOGO kostenlos mit aus. Leider wurde diese Beilage inzwischen mangels Nachfrage eingestellt, und Interessierte müssen LOGO jetzt käuflich erwerben.

Ein Faktor, der LOGO zum Durchbruch verhelfen könnte, ist sicherlich das vorliegende Buch. Es ist ein hervorragend geschriebenes Lehrbuch, das die Seiten nicht mit aus dem Handbuch abgeschrieben Befehlsübersichten füllt, sondern auf 400 Seiten Erklärung pur bietet. Es richtet sich sowohl an den totalen Computerneuling als auch an erfahrene Freaks, die in LOGO einsteigen wollen.

Das Buch ist in 18 Lektionen aufgeteilt, von denen jede neue Sprachelemente erklärt und an Beispielprogrammen verdeutlicht. Der Autor beginnt mit dem Laden und Starten von LOGO und geht auf die Handhabung der Windows ein. Danach beginnt der Einstieg in die Sprache. Beginnend mit den einfachen Sprachelementen breitet er Stück für Stück alle Geheimnisse von LOGO vor

dem Leser aus. Er entwickelt dabei Programme, die den speziellen Eigenschaften von LOGO angepaßt sind und mit der steigenden Komplexität der erklärten Befehle ebenfalls leistungsfähiger und schwerer werden. Der Autor legt dabei sehr viel Gewicht auf die ausführliche Behandlung der Grafikfunktionen und der Realisierung von Künstlicher Intelligenz. So entwickelt er in einem Kapitel das bekannte selbstlernende Programm, das Tiere anhand von bestimmten Kriterien errät.

Zu Beginn jeder Lektion wird übersichtlich dargestellt, welche Sprachelemente im folgenden neu eingeführt werden, und am Ende findet der Leser einige Übungsaufgaben, anhand derer er sein Wissen überprüfen kann. Da aber leider keine Lösungen zu den Aufgaben vorhanden sind, muß man hier den einzigen Schwachpunkt des Buches aufdecken. Obwohl sich die Aufgaben teilweise sehr stark an den behandelten Stoff anlehnen, sind vor allem Laien oft unsicher und wären für die Lösungen sicherlich dankbar gewesen.

Das LOGO-Buch zum Atari ST ist das Standardwerk für alle, die den Computer der neuen Generation in dieser Sprache programmieren wollen. Es ist gut geschrieben und immer leicht verständlich. Wenn auch die Aufmachung und einige Stellen des Textes sehr an Schulbücher erinnern, sollte man sich doch nicht abschrecken lassen, denn es handelt sich nun mal um ein Fachbuch. Für eine eventuelle 2. Auflage sei aber empfohlen, den Text lockerer zu gestalten.

Thomas Tai

**Computer-Kontakt
das Heft mit den
preisgünstigen
Kleinanzeigen**

Das große Basic-Buch zum Atari ST

Von Frank Kampow und
Norbert Szczepanowski
Data Becker Verlag
268 Seiten, 39.- DM
ISBN 3-89011-121-1

Basic auf dem Atari ST ist ein heikles Thema, da der mitgelie-

ferne seine Nutzung und Bedeutung eingeführt. Dann geht es schon los mit der Programmierung.

Der Leser erstellt sein erstes Programm und im folgenden noch viele andere kleine Listings, die die Funktion der grundlegendsten Befehle verdeutlichen. Nachdem die Autoren diese Grundlagen gelegt haben, kommen komplizierte Strukturen an die Reihe, wobei die kleinen Beispielprogramme teilweise mit Flußdiagrammen erarbeitet werden. Zu jedem dieser Kapitel werden dem Leser Aufgaben gestellt, an denen er seinen Lernerfolg überprüfen und eventuelle Unklarheiten erkennen und nacharbeiten kann.

Die folgenden Kapitel behandeln komplexere Programmierertechniken. Von mehrdimensionalen Feldern und der Unterprogrammtechnik über einen sehr kurzen Sprung in die Grafik- und Soundprogrammierung bis hin zum Aufruf von GEM-Routinen wird der Bogen gespannt, wobei verständlicherweise nicht alles in großer Ausführlichkeit zur Sprache kommt. So sollte man die Kapitel über Grafik, Sound und GEM nur als kleinen Appetitstrecker betrachten. Ernsthaftige Anwendungen sind mit dem vermittelten Wissen kaum möglich.

Allen Unkenrufen zum Trotz kann man sagen, daß sich ein Anfänger die Logik des Programmierens am einfachsten und schnellsten mit Basic beibringt. Das war aufgrund der spärlichen Dokumentation des ST Basic nur schlecht möglich, und genau hier setzt das Basic-Buch zum ST ein. Es wendet sich zunächst an den völligen Neuling, der neben Anwenderprogrammen auch eigene Problemlösungen auf seinem Computer laufen lassen will, vom Programmieren aber kein Ahnung hat. Behutsam wird er in die Grundlagen eingeführt. Begriffe wie Dualsystem, Hexadecimalsystem, ASCII-Code oder Bit und Byte bekommen einen konkreten Hintergrund. Außerdem wird das Flußdiagramm

Der Anhang hat es dann noch einmal in sich. Eine Übersicht listet alle Befehle in alphabetischer Reihenfolge mit Format, Erklärung der Funktion und einem kleinen Beispiel auf. Außerdem werden die Fehlermeldungen in Deutsch erklärt.

Das Buch ist eine hervorragende Einführung in Basic, selbst für den totalen Neulingsteiger. Ein guter didaktischer Aufbau mit wenig Fachchinesisch sind weitere Merkmale des Werkes. Mit dem nützlichen Nachschlageteil bleibt es auch nach dem Durcharbeiten noch interessant und wird sich beim täglichen Programmierereinsatz bewähren.

Thomas Tai



Tom Rowley

Sprühende Ideen mit Atari Grafik



290 Seiten
Sprühende Ideen ist ein Lehrbuch, das mit den Möglichkeiten des ATARI als die Gestaltung von Objekten, in Farbgebung und die Entwicklung von Massenschnittstellen anführt. Für den Leser genügt Kenntnisse der Programmiersprache Basic – auch wenn das Buch gelegentlich die Vorteile der Maschinensprache zeigt.

Bestellnummer TW 104

DM 48,-

C. Lorenz

Das große Spielbuch für Atari



200 Seiten
Dieses Buch enthält eine Reihe aktueller Programme für den Atari 800 XL und 800 XL und ist eine Weiterführung von Band 1, dem großen Spielbuch für Atari. Es bringt eine Reihe neuer Spiele, Programme zur Soundbelegung und ein Kapitel über Grafik-Spiele mit dem Atari. Außerdem enthält es einige Tips zum Programmieren zum Zusammensetzen des Atari.

Bestellnummer H 820

DM 29,80

C. Lorenz

Das große Spielbuch für Atari



101 Seiten
Aufwändige Computerspiele in Atari-Basic. Neben Spielen finden Sie hier eine Reihe hochinteressanter Anregungen für eigene Programme. So Grafik, Bewegung und Scores, Grafik und Ton in FORTH. Fortprogrammierung usw.

Bestellnummer H 821

DM 29,80

Handbuch für Hacker



In diesem Buch finden Sie endlich alle Antworten auf die Fragen, die sich jeder stellt und bislang keine beantwortet konnte. Mit vielen Nummern und Datenbanken. Endlich wird hier erklärt, wie man mit Daten-P arbeitet. Viele Tips und Tricks zum Mailbedienen, Steigen Sie mit ein in die Unterwelt der DPU.

Bestellnummer 8

DM 38,-

Raeto West

C 64/SX 64 – Computer Handbuch.



688 Seiten
Insgesamt ein Jahr verstreut hat der Autor Raeto West mit der Anleitung und Dokumentation auf den C 64. Das Ergebnis seiner Arbeit: Das einzige einschlägigste 64er Buch, das neben Ihrem Computer liegen sollte. Das Buch ist so geprägt von laiden Fachwissen und umfangreicher Kompetenz. Es enthält über 300 Programmierungen aller 64er Funktionen – auch die schwierigsten, seltenen und meist genährten.

Bestellnummer TW 102

DM 68,-

Don Inman/Kurt Inman

Der Atari Assembler



278 Seiten, 82 Abb., ca. 190 Programme
Mit diesem Buch können Sie das Programmieren in Assembler lernen und sich gleichzeitig mit der Anwendung des Atari Assembler Moduls auf dem Atari 400- oder 800-Modell vertraut machen. Das Buch ist eine ausgewogene Einführung für Leser mit einigen Grundkenntnissen in Basic, setzt aber keinerlei Assembler-Kenntnisse voraus.

Bestellnummer ID 302

DM 38,-

Karl-Heinz Koch

ATARI Spiele programmieren



240 Seiten
Das Buch führt Schritt für Schritt in das Programmieren in BASIC ein. Dabei werden schon mit den ersten einfachen Definitionen grundlegende Grafikkenntnisse erlernt. So werden die Details und ihre Wirkung optisch erlernbar gemacht. Auf Verständlichkeit wird besonders Wert gelegt, was für Bücher dieser Materie leider keine Selbstverständlichkeit ist.

Bestellnummer BI 907

DM 22,-

Chaos Computer Club

Die Hackerbibel



290 Seiten
Überall in den Medien ist seit dem brennenden Zeit Raeto von den Hackern. Und wenn man "Hacker" sagt, meint man in diesem unsere Leute zu sein: Mitglieder des Chaos Computer Club Hamburg und seine Anhänger. Sie erlernt 130.000 DM Coup bei der Hamburger Sparkasse oder die diversen Affäre in Tageschau, bei Frank Erster und anderen; aber sie ist viel mehr als nur ein Handbuch an Buch von Ihnen. Nicht nur über das "Wie" des Hackens, sondern auch über das "Warum".

Bestellnummer H 906

DM 33,33

Julian Reschke/Andreas Wiethoff

Das ATARI Profibuch

286 Seiten. Das richtige Buch für Sie, wenn Sie einen ATARI 400, 800, 800 XL, 800 XL oder 130 XL besitzen. Das Buch enthält eine vollständige Liste aller Systemadressen und routinen mit Erläuterungen, detaillierten Informationen über Speicherbelegung und Ausnutzung des Betriebssystems, einen Tabellenteil mit Zeichencodes, Adressen, Könnigcodes, Tonhöhen, Farben, Fehlersignale sowie einen alphabetischen Nachschlageregister.

Bestellnummer SY 108

DM 42,-

Hal Hickman

Der ATARI als Musikbox

191 Seiten. Dieses Buch führt Sie in die Grundbegriffe der Musik ein. Erst danach werden die Möglichkeiten der Computermusik dargestellt. Das Buch ist für Anfänger geschrieben und zeigt, wie man mit einfacher Basic-Programmieren Melodien, Akkorde und Kanone erstellen kann. Sie müssen weder Noten lesen noch ein Instrument spielen können, um mit diesem Buch Ihren ATARI in ein mehrstimmiges Orchester zu verwandeln.

Bestellnummer MI 123

DM 29,80

BUCH-BESTELLKARTE

Bitte leihern Sie mir folgende Bücher:

Anzahl	Bestell-Nr.	Titel	Freigegeben oder Muttel

Name des Bestellers

Anschrift

Adress

Gebiet

Ich wünsche folgende Bezahlung:

- Nachnahme (+ 3,70 DM Porto) • Versandkosten
 Vorauskassa (keine Versandkosten)

Die Kontokausen bitte Scheck belegen oder auf Postcheckkonto Karlsruhe 43429-716 überweisen.

Stempel/Unterschrift

Copies abschneiden, auf Postkarte haben und einbinden in Computer Kontakt, Buchversand, Postfach 18 40, 75181 Bretten.



TI 99/4A

Liebe TI-Freunde!

In dieser Ausgabe gibt es viel Neues zu berichten. Zuerst die Meldung, auf die alle TI-Besitzer schon seit Monaten warten: Der neue TI, lange Zeit als Gerücht gehandelt, steht endlich vor der Tür. Er soll nach letzten Meldungen als Einschubkarte für die Peripheriebox aus-

führt sein. Auf dieser Karte befinden sich 640 KByte RAM, die bis auf 2 MByte ausgebaut werden können. Das Herz des neuen TI schlägt mit 12 MHz und wird vom Prozessor TMS 9995 gebildet. Dieser Prozessor ist der kleine Bruder des TMS 9900, der sich im TI 994A be-

findet. Der Datenbus des TMS 9995 ist intern 16 Bit breit, während nur 8 Datenleitungen nach außen führen. Im Prozessorchip ist zusätzlich ein 256-Byte-RAM integriert, auf das der Prozessor wesentlich schneller zugreifen kann als auf externe Speicherchips. Als Bildschirm kann ein 80-Zeichen-Farbmonitor benutzt werden. Eingebaute Unterstützungsrouitinen erlauben nun die Anwendung der Programme "TI-Writer" und "MS-Multiplan". Eine neue IBM-XT-Tastatur ist neben einem Modulport und einer RS-232-Schnittstelle ebenfalls vorgesehen. Die Kompatibilität ist assemblemäßig zu 95% gewährleistet, während das eingebaute Basic dem Microsoft Basic der IBM-Rechner entspricht und somit mit dem TI-Basic nicht mehr viel gemein hat. Die Maschine soll nach Mitteilungen von Myarc ca. 500 Dollar kosten und unter dem Namen Geneva auf den Markt kommen.

Zum nächsten Punkt: Der TI 99er Workshop Rheinland möchte zu Anfang des Jahres '87 einen TI-Treff in Köln veranstalten, der alle bisherigen Veranstaltungen dieser Art überbieten soll. Viele Clubs aus dem Ausland haben ihr Kommen bereits zugesagt; die Stadt Köln und einige ortsansässige Unternehmen unterstützen die Aktion finanziell. Nur die deutschen TI-Clubs haben bislang nicht viel Interesse gezeigt. Daher möchte ich an dieser Stelle alle TI-Benutzer und Inter-

essengeneinschaften, die Zeit und Lust haben, an einer solchen Veranstaltung teilzunehmen, auffordern, sich beim Workshop zu melden, damit die Teilnehmerzahl ungefähr abgeschätzt werden kann. Es wäre schade, wenn die Veranstaltung mangels Interesse nicht stattfinden könnte.

Doch nun zum heutigen Inhalt. Es gibt auch hier wieder eine kleine Sensation zu vermelden: Der TI würde um eine Programmiersprache erweitert. Ab sofort kann man mit ihm auch in C sprechen. Wer eine Speichererweiterung, ein Diskettensystem und ein Editor/Assembler-Modul besitzt, kann für ganze 10.-DM einen C-Compiler als Freeware erwerben und die neue Sprache ausprobieren. Als weitere Neuigkeit haben wir ein EPROM-Programmiergerät von der Firma Mechatronik getestet. Und der Aktivität unserer Leser verdanken wir einen interessanten Artikel über neue Software für das Grafikprogramm "TI-Artist". Zwei Spiele in Extended Basic, ein Utility zum Beschleunigen von Basic Programmen und ein Assemblerprogramm, das das Laden und Speichern von überlangen Basic-Programmen ermöglicht, runden die heutige Ausgabe ab. Ich wünsche euch wieder viel Spaß und hoffe, daß ihr auch weiterhin so aktiv bleibt wie in den vergangenen Wochen und uns eure persönlichen Erfahrungen und Ansichten mitteilt.

H.-P. Schwarz

TI-Software-Hitparade

So langsam scheint die Sache ins Laufen zu kommen, aber wie gesagt, ein paar mehr können es schon noch werden.

Der Spitzenreiter ist diesmal der alte, aber auf den Plätzen dahinter tut sich einiges. Das liegt auch daran, daß immer noch neue Software aus den USA nach Deutschland kommt.

- | | |
|----------------------|-----------|
| 1. TI-Artist II | (1) A,AS |
| 2. Micro Pinball | (-) S,AS |
| 3. Tennis | (3) S,AS |
| 4. War Games | (-) S,AS |
| 5. DM 1000 | (4) A,AS |
| 6. Bigfoot | (2) S,AS |
| 7. Artillerie | (9) S,BA |
| 8. Ret. t. P. Island | (10) S,MO |
| 9. Disk-Katalog (CK) | (6) A,AS |
| 10. TI-Runner | (8) S,AS |

Der neue Gewinner ist:

Ralf Bauer, Kelterstr. 16, 6962 Adelsheim-Sennfeld

Er erhält ein brandneues Softwarepaket aus den USA. Ihr seht, es lohnt sich, mitzumachen. Ab sofort nehmen übrigens alle Eisenlungen an der Verlosung teil.

Eure Tips müßt ihr wieder an folgende Anschrift schicken:

TI 99er Workshop Rheinland
Dept. Allgemein + Software
c/o Mike Heuser
Karl-Marx-Allee 18
5000 Köln 71

Heiße Bits für den TI mit dem Mechatronic-Eprommer 99

Von vielen TI-Freaks schrittweise erwartet, gibt es jetzt ein EPROM-Programmiergerät für den TI. Mit diesem Zusatz wird der gute alte TI zur Entwicklungsmaschine für TMS-9900-Maschinenprogramme. Zusammen mit den bereits erhältlichen EPROM- und GROM-Karten kann jeder TI-Besitzer nun seine eigenen Module herstellen.

Der Eprommer befindet sich in einem stabilen Metallgehäuse und wird in den Modulport eingeschoben, ist aber doppelt so lang wie ein normales Modul. Zwei Gummifüße an der Unterseite des Gehäuses stützen den gesamten Eprommer ab, damit die mechanischen Belastungen an Modulport und Eprommerplatine nicht zu Beschädigungen führen. Um das Netzteil der Konsole nicht zu überlasten, übernimmt ein separates Steckernetzteil die Spannungsvorsorgung. Auf der Gehäuseoberfläche befinden sich ein 28poliger IC-Nullkraft-Stecker, ein 36poliger Steckverbinder, der zur Aufnahme eines Moduls dient, und vier LEDs, die den Betriebszustand des Eprommers signalisieren. Alles in allem macht der gesamte Aufbau einen sauberen und vertrauenerweckenden Eindruck.

Mit dem Mechatronic-Eprommer kann man insgesamt 7 verschiedene EPROM-Typen programmieren. Die Skala reicht vom Typ 2716 mit 2 KByte bis zum Speicherriesen 27128 mit einer Kapazität von 16 KByte. Da der Eprommer vollständig softwaregesteuert ist, entfällt das lästige und fehlerträchtige Einstellen von Dip-Schaltern und Programmierspannungen. Die Betriebssoftware kann unter den Modulen Extended Basic, Editor/Assembler oder Minimemory gestartet werden. In jedem Falle sind außerdem eine 32-KByte-Speichererweiterung und ein Diskettenlaufwerk nötig.

Die Arbeit mit dem Eprommer ist sehr einfach. Zuerst steckt man ihn in den Modulport der Konsole. Danach wird das zum Betrieb der Software nötige Modul, beispielsweise das Extended-Basic-Modul, in die Steckerriste des Eprommers gesteckt. Erst jetzt darf man, laut Handbuch, die Konsole einschalten, da ein Modulwechsel bei eingeschaltetem System den Eprommer, die Konsole oder das Modul selbst beschädigen könnte. Nun wird das Steckernetzteil mit dem Eprommer verbunden. Dann kann es endlich losgehen. Die Betriebssoftware für den Eprommer wird automatisch bei der Auswahl des Extended-Basic-Moduls von Diskette geladen, und nach wenigen Sekunden erscheint das Hauptmenü auf dem Bildschirm, das zwei Auswahlmöglichkeiten erlaubt: EPROMMER COMMANDS und BUFFER COMMANDS. Unter dem Punkt EPROMMER COMMANDS kann man den EPROM-Typ auswählen und danach mit dem Inhalt des RAM-Puffers programmieren. Weitere Möglichkeiten unter EPROMMER COMMANDS bestehen darin, den Inhalt eines EPROMs in den RAM-Puffer zu lesen, seinen Inhalt mit dem RAM-Puffer zu vergleichen und zu überprüfen, ob ein EPROM völlig gelöscht ist. Als Pufferspeicher dient in allen Fällen der RAM-Bereich von +A000 bis +DFFF. Anwender-freundlich ist auch, daß man beliebig große Teile des EPROMs beschreiben kann und nicht gezwungen ist, für wenige Byte das ganze EPROM zu programmieren, was mit einer entsprechend längeren Wartezeit verbunden wäre. In diesem Zusammenhang soll auch erwähnt werden, daß EPROMs vom Typ 2764 und 27128 im sogenannten Fast-Programming-Mode gebrannt werden können, was eine erhebliche Zeitersparnis zur Fol-



USA-SPECIAL

S. ADAMS ADVENTURES
Adventure Modül DM 24,90
The Hulk, Spiderman, Gyanthor
Castle & Sauron's Starz
Jackie DeBarto DM 39,-
4er Pack DM 152,200,-

IN-KOPIE auch:
INFOCOM-Spieleplan:
Hörbücher DM 79,90
Parental DM 79,90
Zirk I DM 79,90
Zirk II DM 79,90
Zirk III DM 79,90
Diskette DM 99,-

und noch vieles mehr! Vorbestellung bewirkt sofortige Lieferung!

CSGD GRAFC

CSGD ermöglicht es, Zeichnungen, kleine Bilder, Symbole u. komplexe Transparenzen zu erstellen in elektronischer Form.
GSD + S DM 39,-
GSD + B DM 39,-
2-User-Edit DM 139,-
Kompatibel DM 249,-

MAUS-EDITOR

Basic, E-Basic und Pascal-sprachige Programm- & Text-Editoren erweiterter Disk-CMS, Backup, Merge-File, viele komplexe Funktionen.
Diskette mit 256 KByte, 15 x 74-Mikroprozessoren + 32 KByte (Mezz-Floppy)
Diskette nur DM 89,90

SUPER ACTION

3-D-Action-Programme DM 99,-
BTHAC Supergrafik-D DM 129,-
T-World 3-D-D DM 99,-
T-Form V. 4.3-D DM 99,-
Turbo Assembler-D DM 99,-

MADEIRA

Assembler-Kurs 104 DM 79,90
Assembler-Kurs 216-248 DM 149,-

WÄRDRE

WÄRDRE ist sehr powerful! Funktionen wie: Heap-Stack-Verwaltung, Disk-CMS, DM 139,-, MM-Speicher, 4000-Verfahren, CPS, Mega-RAM etc.

TI-ARTIST SUPPORT

by TI-ARTIST for more fun with your TI-ARTIST!

Bitte aus dem USA-Ausführer Testen diesen Heft!

Preisliste:
TI-ARTIST Version 2.2 DM 25,-
Art. Companion 1, 2 Disk. DM 120,-
Art. Companion 2, 2 Disk. DM 79,90
Art. Display Master DM 24,90
Art's 4-Box DM 109,-
Companion 1/2 komplett DM 249,-
Comp. 1+2 + Master + Extras DM 249,-
TI-ARTIST 10 Disk. Set DM 249,-

Da ist der Weihnachtsmann nicht!

TI-99/4A Total...

TORPEDO BASIC V. 3.0

Das stärkste Programm für Ihren TI.

Features:
30 Maschinenbefehle plus Utilities für Disk-Operationen, Vektor-, 40-Bit-Modul, Vektor-Filter, VDP-CPU-Speicherbussteuerung, PRINTV und POKED mit E-Basic, 32-Prüfung der betrieblichen Hardware & bei allen Modulen, Suchen & Ersetzen, ärmlichen Routinen, Basiskonsole (nachdem Bildschirm gesteuert), Cursor, 10 internationale Zeichensätze und noch vieles mehr!
Lieferumfang:
32 Maschinenbefehle mit 200wärtigen Funktionen und erweiterten Möglichkeiten, 32-Prüfung der betrieblichen Hardware für E-Basic u.a., Editor/Assembler, Inklusiv: IWD, Suchprogramm, Disk und Quick Reference Card.

Komplettpreis nur DM 129,-

SUPERINFO GRATIS

Das gibt's nicht für 99,- Tausch, sondern hier!

Rausch & Haub

Vertriebsgesellschaft d. d. r.

Postfach 32 00 12

5300 BONN 3

Preisangeb. & Best. Nr. Anwerd. DM 2,30

10.30. Voraus/Rechnung/Suchen

02 28 / 33 38 64

Montag-Freitag 11.00-18.00 Uhr

ge hat. Wenn der Programmiervorgang gestartet ist, erscheint in der untersten Bildschirmzeile die Adresse der in Arbeit befindlichen Speicherzelle, so daß man während der Programmierzeit wenigstens sieht, daß sich etwas tut.

Unter dem Hauptmenüpunkt BUFFER COMMANDS befinden sich Kommandos, mit denen der Pufferspeicher beeinflusst werden kann. Zunächst findet man die unterschiedlichsten Ladekommandos. Der Pufferspeicher kann aus anderen RAM-, GROM- oder VDP-Bereichen geladen werden; ebenso ist das Laden von Assembler-Files von Diskette sowohl im Tagged-Object-Format als auch im Program Image Format möglich. Eine wertvolle Ergänzung bietet ein Hex-Editor, mit dem sich einzelne Speicherzellen manuell ändern lassen. Um mühsam erstellte Speicherinhalte zu sichern oder den Inhalt fremder EPROMs zu archivieren, kann der gesamte Pufferin-

halt im Program-Image-Format auf Diskette gespeichert werden.

Computer-Kontakt das Heft mit den preisgünstigen Kleinanzeigen

Der Mechatronic-Eprommer ist für Programmierer, die eigene Module herstellen wollen, genauso zu empfehlen wie für Anwender, die EPROMs für völlig anderen Gebieten einsetzen wollen (z.B. Einplatinen-Mikrocomputer in Steuerungen). Durch die solide Ausführung und die flexible Software ist dieses Gerät auch für den professionellen Einsatz denkbar. Der Hobbyanwender muß auf jeden Fall neben den Anschaffungskosten von ca. 280,-DM auch noch den Preis für eine UV-Lampe zum Lösen der EPROMs einkalkulieren.

H. P. Schwaneck

Ein Künstler und seine Gesellen

Neue TI-Software

Es gibt seit einiger Zeit in Deutschland bei verschiedenen Händlern Bestrebungen, endlich wieder gute Software aus den USA zu importieren und zu vernünftigen Preisen anzubieten. Dies ist nicht zuletzt ein Verdienst des TI 99er Workshop Rheinland, der seine Partner immer wieder auf die Möglichkeiten des Softwareeinkaufs in den USA hingewiesen hat.

Seit einiger Zeit ist nun auch auf dem deutschen Markt das hervorragende Zeichen- und Grafikprogramm "TI-Artist" erhältlich. Nachdem in einem anderen Magazin schon ein detaillierter Testbericht veröffentlicht wurde, hier nur noch einmal eine kurze Zusammenfassung der wichtigsten Merkmale des "TI-Artist 2.0."

Es handelt sich um ein Programm, mit dem auf einfachste Weise hervorragende Grafiken auf dem Bildschirm erstellt werden können. Ebenso leicht lassen sich verschiedene Schriftarten und kleine Hilfsbilder, die man später zu Großbildern zusammenfügen kann, erstellen. Sogar der Aufbau elektronischer Schaltbilder ist möglich, da die gebräuchlichsten Zeichen (z.B. für Transistoren) schon im Programm vordefiniert sind. Diese können, bevor man sie positioniert, in alle Richtungen gedreht werden. Nachdem ein Zeichen die richtige Lage hat, wird es mit dem Joystick, einer Maus oder den Cursor-Tasten in das Bild eingebaut.

Im eigentlichen Hauptmenü findet man eine Vielzahl von Funktionen, die von der einfachen DRAW-Routine über CIRCLE- und FILL-Routinen bis hin zur Auswahl der Strichstärke gehen. Soweit die Kurzbeschreibung des "TI-Artist", der mittlerweile bei einigen deutschen Händlern für ca. 98,- DM zu erstehen ist.

Nun aber zu seinen Gesellen. In einer amerikanischen TI-Fachzeitschrift fand sich eines Tages eine Werbeseite, auf der

Zubehör für den "TI-Artist" angeboten wurde. Dieses Paket besteht aus insgesamt 10 (!) Disketten, voll mit Grafiken, Zusatzfiles, Hilfsroutinen und vor allem Schriftzeichen, die man ohne große Anstrengungen mit dem "TI-Artist" verarbeiten oder verbinden kann. Zuerst einmal gibt es da den "Display-Master". Mit diesem Programm kann man mit Hilfe des "TI-Writer" oder des E/A-Editors eigene Hilfsprogramme schreiben, um Bilder, die mit dem "TI-Artist" erstellt wurden, im Extended Basic auf den Bildschirm zu bringen. Endlich wird es auch möglich, einen farbigen Rahmen um den Bildschirm zu legen, indem man verschieden große Fenster definiert und mit Texten versehen kann. So können z.B. Balkendiagramme erstellt werden und mit dem "Display-Master" die dazugehörigen Kommentare über das Bild verteilt werden. Auch einige andere hilfreiche Befehle, wie z.B. Pause, Loop usw., erleichtern die Arbeit un-

ST kennt. Hier zeigt der TI wieder einmal, was wirklich in ihm steckt.

Auf zwei zusätzlichen Disketten liefert derselbe Hersteller, die Fa. Insecto Inc., weitere sog. "Artist-Extras". Zum einen befinden sich auf diesen Disketten einige sehr schöne Schrifttypen. Weiterhin erhält man einige Unterstützungsroutinen, die direkt in den "TI-Artist" eingebunden werden können, so z.B. die Input DSR für eine Maus, die es ja inzwischen auch hier zu kaufen gibt.

Den dritten Teil der Disketten bilden ca. 10 mitunter sehr gut gelungene Bilder, die entweder im "TI-Artist" oder auch im "Display-Master" verwendet werden können, sowie drei Konversions-Routinen, mit denen Grafiken anderer Programme "TI-Artist"-kompatibel werden.

Die weiteren 7 Disketten sind nun randvoll mit allen möglichen Schriftarten, Bildern und vor allem mit sogenannten In-

tended Basic mit einem sog. CSGD-File, das sich ebenfalls auf den "Artist-Extras"-Disketten befindet, erstellen lassen. Sie können mit einem Konversions-Programm auf den "TI-Artist" umgewandelt oder im Extended Basic benutzt werden. Einen besonderen Clou bieten alle Schriftarten, die sich sowohl in Normschwarz als auch invers darstellen lassen, so daß sich ihre Anzahl noch einmal verdoppelt. Eine kleine Auswahl dieser Instances sowie einiger weniger Schriftarten aus diesem Package haben wir diesem Artikel als Hardcopy beigefügt, damit man einmal einen Eindruck von der Leistungsfähigkeit dieses Pakets bekommt.

Insgesamt bekommt der Käufer für sein Geld eine fast riesig zu nennende Menge Software, die nahezu keinen Wunsch in punkto Grafik offenläßt. Auf den 10 Disketten befinden sich ungefähr 200 (!!) Routinen und Hilfsfiles, wobei aber die Masse nicht auf Kosten der Klasse geht. Ein weiterer Vorteil dieses Pakets ist ohne Zweifel, daß man nun nicht alle 10 Disketten auf einmal kaufen muß, sondern ganz gezielt auswählen kann, was für einen selbst nutzbringend oder erschwinglich ist. So ist es z.B. möglich, den "Display-Master" oder die "Artist-Extras" einzeln zu erwerben. Die 7 Companion-Disketten sind aufgeteilt in zwei Pakete zu 5 und 2 Disketten. Ebenso kann man alle Teile des Pakets miteinander kombinieren (z.B. 2 Companion-Disketten und den "Display-Master"), so daß den Möglichkeiten kaum Grenzen gesetzt sind.

Auf unser Betreiben hin hat sich die Fa. Rausch & Haub in Bonn dazu entschlossen, das gesamte Paket für die deutschen TI-User zu importieren. Die aktuellen Preise kann man der CK-Werbung oder der Mailbox des TI 99er Workshop Rheinland Tel. 02 28/65 82 32 entnehmen.



Neuestalteste Präsentation des TI-Artist

gemein. Zum Lieferumfang gehören neben einer sehr ausführlichen Befehlsliste und einer Betriebsanleitung 4 Demobil-der von einer Qualität, die man bisher nur von Rechnern der gehobeneren Preisklasse wie Commodore Amiga oder Atari

stances. Dies sind kleine Bildchen von ca. 5x5 cm, die an jede beliebige Bildschirmposition geschoben, vervielfältigt, kopiert und bearbeitet werden können. Ein weiterer Vorteil der Instances ist, daß sich Muster und Grafiken auch im Ex-

TI --ARTIST --COMPANION HIER EINE AUSWAHL DER SCHRIFTEN



Eine Vielfalt von Schriften

Noch ein Wort an alle diejenigen, die uns in letzter Zeit fragen, wie man denn wohl in den USA einkaufen könne. Hier einige Tips dazu. Leider gibt es keine Möglichkeit, kleine Geldbeträge an die Lieferanten zu überweisen, da in den allermeisten Fällen keine Bankverbindung bekannt ist. Auch Lieferung per Nachnahme (C.O.D.) kommt für die US-Händler nicht in Frage; das Mißtrauen ist drüben allgemein sehr groß. Die meisten Händler bieten drei Wege der Bestellung und Bezahlung an. Die erste Möglichkeit besteht darin, seiner Bestellung den Kaufpreis in bar

beizulegen. Man sollte aber dafür sorgen, daß es von außen nicht möglich ist, das Geld im Umschlag zu sehen oder zu ertasten. Dies kann man z.B. durch ein Stück schwarze Pappe verhindern, das man mit in den Umschlag legt. Wir haben bisher nur gute Erfahrungen mit dieser Art des Einkaufs in den USA gemacht. Für Leute, die eine Kreditkarte besitzen, gibt es nun auch noch die Möglichkeit, telefonisch bei den Händlern zu bestellen und bei der Bestellung die Nr. der Karte sowie das Gültigkeitsdatum anzugeben. Hier gewähren die meisten Händler auch noch einen klei-

nen Rabatt. Inwieweit der normalerweise TI-User nun über eine (in den USA sehr verbreitete) Kredit-Karte verfügt, entzieht sich unserer Kenntnis, der Vollständigkeit halber sei diese Möglichkeit hier aber auch erwähnt. Die dritte Möglichkeit ist die Zahlung per bankbestätigtem Scheck. Diese Zahlungsart sollte aber vor dem Einkauf mit der Bank und dem US-Händler abgeklärt werden. Hier nun noch die Adresse eines TI-Anbieters in den USA, der wohl mit das größte Angebot an Hard- und Software beinhaltet:

TEX-COMP

P. O. Box 33064
Granada Hills
CA 91344
USA

Gegen Zusendung von 2-
US\$ erhält man von dort einen
40seitigen Katalog mit allem,
was des TI-Users Herz begehrt.

Das war's vorerst wieder einmal vom TI 99er Workshop Rheinland. Wer Fragen bezüglich des "TI-Artist"-Companion-Packages hat, wende sich entweder an die bekannte Adresse oder direkt an Rausch & Haub GdBR, 5300 Bonn.

TI 99er Workshop Rheinland
Mike Heuser



Computer Kontakt
schließt sich diesem
Grüß an

Neues vom Turbo-Assembler

Der Turbo-Assembler hat sich zum Bestseller unter unseren angebotenen Programmen gemauert. Aufgrund der herausragenden Eigenschaften dieses Programms erscheint dies auch nicht weiter erstaunlich. Immerhin arbeitet der Turbo-Assembler mindestens doppelt so schnell wie der TI-Assembler; der Bedarf an Speicherplatz auf der Diskette von Quell- und Objectcode ist bis zu 3fach geringer, während die Zeit zum Laden der fertigen Programme 5mal kürzer ist. Dies sind schon sehr überzeugende Vorteile. Doch damit nicht genug. Der Turbo-Assembler ist der einzige Assembler für den TI, der auch ohne Diskettenlaufwerk auskommt und

notfalls auch mit Cassettentrecorder sehr gute Ergebnisse erzielt. Da aber gute Dinge auch noch verbessert werden können, wurden einige Schönheitsoperationen am Turbo-Assembler vorgenommen.

Aufgrund der konstruktiven Zusammenarbeit mit den bisherigen Benutzern wurde das Handbuch neu überarbeitet. Viele Tips und Anregungen flossen dabei in das neue Konzept mit ein, so daß der Umfang auf insgesamt 19 Seiten angewachsen ist. Das auf der Diskette enthaltene Katalogprogramm, das auch in der CK 6/85 als Listing des Monats erschien, wurde um einige wertvolle Op-

tionen erweitert. Der Speicherplatz für die Einträge wuchs auf fastliche 30 KByte an, so daß auch sehr umfangreiche Programmsammlungen erfaßt werden können. Eine von den Anwendern stürmisch geforderte Funktion wurde nun ebenfalls verwirklicht: Der gesamte Kataloginhalt kann jetzt auch als Datei auf einer Diskette gespeichert werden. Damit lassen sich Updates nun noch schneller durchführen. Wer bislang Schwierigkeiten mit der Druckersteuerung hatte, braucht ebenfalls nicht mehr zu verzweifeln. Eine zusätzliche Druckroutine bedient auch Drucker, die nicht dem Epson-Standard entsprechen. Neben diesen wesentlichen Erweite-

rungen kann man die permanente zur Verfügung stehenden Kleinbuchstaben schon fast als Spielerei ansehen. Dies alles ist - wie schon gesagt - im neuen Turbo-Assembler-Paket enthalten.

Wer schon einen Turbo-Assembler besitzt, der kann seine Originaldiskette einschicken. Gegen einen Unkostenbeitrag von 5,- DM erhält er die neueste Katalogversion und zusätzlich das erweiterte Handbuch. Wer keinen Turbo-Assembler hat, kann eine Freeware-Diskette mit dem neuen Katalog und einem Lader-Utility direkt bei H.-P. Schwaneck, Roggenkamp 3, 3300 Braunschweig anfordern.

Lange Programme auf dem TI 99 speichern

Das Extended Basic des TI 99 hat einen Nachteil: Bei angeschlossener Speichererweiterung können zwar Basic-Programme bis zu einer Länge von 24 KByte geladen werden, die aber dann nicht mehr auf Cassette, sondern nur noch auf Diskette gespeichert werden können. Das später folgende Maschinenprogramm für Extended Basic und 32-K-Erweiterung bietet Ihnen zwei Befehle zur Speicherung solcher Programme auf Cassette.

Will man den Grund für diese Eigenart des Basic verstehen, muß man einen tieferen Blick ins System werfen. In der Grundversion des TI 99 sind in der Konsole nur 16 KByte Speicher eingebaut, die vollständig unter Kontrolle des Video-Chips stehen. Erst die 32-K-Erweiterung schafft einen echten, direkt adressierbaren RAM-Bereich, in dem Basic-Programme gespeichert werden.

Die Cassetten-Routinen haben nun eine Eigenart: Daten von oder zur Cassette müssen über das Video-RAM geschickt werden. Soll ein Programm im kompakten "PROGRAM"-Format gespeichert werden, muß es vollständig als ein Block im Video-RAM liegen. Da dieses aber nur 16 KByte groß ist und die eigentlichen Video-Tabellen und Diskettenpuffer davon noch abgezogen werden müssen, dürfen Programme maximal rund 13 KByte haben, wenn sie auf Cassette gespeichert werden sollen.

Dieses Problem läßt sich nur in Maschinensprache lösen, indem das Basic-Programm in zwei nacheinander zu speichernde Blöcke aufgeteilt wird. Dazu haben Sie zwei Befehle: CALL LINK ("SVLONG") speichert ein überlanges Programm auf Cassette ab. CALL LINK ("LDLONG") lädt es wieder. Dabei erscheinen zwei-

mal die normalen Meldungen wie bei OLD und SAVE. Natürlich dürfen Sie nach dem Speichern oder Laden des ersten Blocks das Band nicht zurückschleusen, auch wenn der Computer Sie dazu auffordert. Bei kurzen Basic-Programmen, die sich normal speichern lassen, gibt das Programm die Fehlermeldung "Bad Value" aus, da es wegen eines möglicherweise erheblichen Zeitverlustes nicht sinnvoll ist, dieses Programm zu verwenden.

Am besten gehen Sie beim Speichern so vor:

- Das Maschinenprogramm eintippen oder laden.
- Das zu speichernde Basic-Programm einlesen.
- SAVE CS1 eingeben.
- Bei Fehlermeldung CALL FILES (1) eingeben und eine nicht vorhandene Zeile löschen; das bringt rund 1 KByte mehr Platz im Video-RAM.

- Erneut SAVE CS1 eingeben.
- Erst wenn jetzt wieder eine Fehlermeldung kommt, CALL LINK ("SVLONG") eingeben.

Das hört sich zwar recht kompliziert an, ist aber ganz logisch. Erheblich einfacher ist nun das Einlesen eines so gespeicherten Programms: CALL LINK ("LDLONG") eingeben und das Band starten. Sobald der erste Programmblock geladen ist, müssen Sie entsprechend den Meldungen auf dem Bildschirm den zweiten Block einlesen.

Noch ein Hinweis: Da das Maschinenprogramm im Low Memory liegt und hier für das Video-RAM einen 6-KByte-Buffer benötigt, dürfen solange keine anderen Maschinenprogramme mehr in diesen Bereich geladen werden, wie man die Cassetten-Routinen verwendet.

Martin Korfsta

MC-Generator

```

10 *****
20 :* POKELISTE FUER *
30 :* LOAD/SAVE LONG *
40 :* ERSTELLT AM 9.9.86 *
50 :* VON H-P SCHWANECK *
60 *****
100 DIM CH(40):: CALL CLEAR :: PRINT "ICH UEBERPRUEFE DIE DATA- ZEILEN. BITTE
WARTEN"
110 RESTORE 810 :: FOR I=1 TO 39 :: READ CH(I):: NEXT I
120 RESTORE 420 :: FOR Z=1 TO 39 :: SUM=0 :: FOR I=1 TO 16 :: READ A :: SUM=SUM+
A :: NEXT I :: IF SUM<CH(Z) THEN PRINT "EINGABEFehler IN ZEILE";Z*10+410 :: CALL
SOUND(100,110,0):: STOP
130 NEXT Z
140 PRINT "EINGABE KORREKT, JETZT WIRD GEPOKED"
150 RESTORE 420 :: ADR=9460 :: CALL INIT
160 READ A :: IF A<-1 THEN CALL LOAD(ADR,A):: ADR=ADR+1 :: GOTO 810
400 CALL LOAD(8194,36,244,63,240)
410 CALL LOAD(16368,76,68,76,79,78,71,37,222,83,86,76,79,78,71,37,112)
420 DATA 005,006,000,000,000,000,000,000,000,000,000,000,000,203,020,203,053
430 DATA 203,078,203,231,204,071,204,150,204,228,205,029,205,075,205,096
440 DATA 033,131,035,253,038,184,040,183,041,182,042,195,043,193,044,179
450 DATA 045,194,047,196,058,181,059,180,060,191,061,190,062,192,094,197
460 DATA 255,058,058,130,065,084,240,071,079,133,073,070,132,079,078,155
470 DATA 079,082,186,080,073,221,084,079,177,255,065,066,083,203,065,076
480 DATA 000,000,000,003,004,000,008,000,000,058,207,208,006,012,009,196
490 DATA 080,080,000,000,096,003,067,083,049,000,255,255,002,224,036,246
500 DATA 136,032,131,048,037,094,026,004,002,000,030,000,004,032,032,052
510 DATA 216,032,036,245,037,096,006,160,038,146,002,000,009,196,002,001
520 DATA 131,048,002,002,000,004,004,032,032,036,002,000,009,200,002,001

```


C auf dem TI 99/4A

```

*****
OHEAD LI R0, V0START      Program-Header beginnt Zeiger aufs Programm
      LI R0, L0START
      BLWP 010000

LOADEN MOV SLETLN, S10000 Aus dem Beginn der Zeilenliste S10000 berechnen
      DEC S10000

LISTEN LI R0, V0START+4   Zeiger auf Buffer nach Header
      LI R1, S10000        Startadresse des ersten Blocks
      LI R2, S12000        Länge des ersten Blocks laden
      BLWP 010000         Aus dem VOP-BAR übertragen
      MOV R0, S0FLECH     Mainline Programmblöcke wiederherstellen
      BL S10000          Casseffen-DBR initialisieren

L0M000 BL S10000         Casseffen-DBR aufrufen

L0T000 LI R0, V0START     Zeiger auf Programmbuffer in VOP-BAR
      MOV SLETLN, R0      Zeiger auf niedrigste Main-Adresse holen
      LI R0, S10000      R2-Programmende
      MOV R0, S0FLECH     Programmblöcke wiederherstellen
      BLWP 010000        Programm aus dem VOP-Buffer holen
      BL S0M000          Den VOP-BAR wiederherstellen

RETURN MOV R0, S0STARTUS  SPL-Subroutine-Linker
      BLWP 010000        R0-Adresse laden
      DATA %30          Sprung in den SPL-Interpreter
      BLWP 010000

*****

INT LI R0, PAD          PAD in den VOP-BAR kopieren
      LI R0, PAD+16
      BLWP 010000

P0T000 MOV S0M000, S1AC   Zeiger für die Casseffen-DBR laden
      MOV S0M000+2, S1AC+2
      LI R0, PAD+13
      MOV S0M000, R0
      MOV S1M000, S10000
      CLP S10000
      MOV S0M000, S10000
      BLWP 010000

D0M000 LI R0, S0000      Den VOP-BAR in Low Memory speichern
      LI R1, S0000VOP
      LI R2, S10000
      BLWP 010000
      RT

L0M000 LI R0, S0000      Den VOP-BAR aus dem Low Memory laden
      LI R1, S0000VOP
      LI R2, S10000
      BLWP 010000
      RT

*****

C000 MOV S0F000, S0STARTUS Casseffen-DBR aufrufen
      BLWP 010000
      DATA %30

O0000 LI R0, PAD+1      Zeiger auf Fehlerbyte in PAD
      BLWP 010000        Fehlerbyte in PAD
      MOV R1, R1+3       Inhibitorbyte aktivieren, Read, ablesen
      BE S00000          Wird ein Fehler angezeigt?
      RT                Nein! Rückprogramm

E0000 BL S0M000         J1 - VOP-BAR wiederherstellen
      LI R0, S0000      J2 - Error ausgeben
      BLWP 010000

*****

V0LINE DATA S0M000, S0F000 Vektoren nach S0F000
      S0F000            Die Base-Adresse sichern
      MOV S0M000, S0M000+1 ... und den Wert bewahren
      DEC S0M000+1

S0F000 MOV S0M000+4, S0M000 Aus DATA den Wert nach S0M000 schreiben
      MOV S0M000+4, S0M000
      LAMP S0F000         SPL-Workarea laden

D0M000 LI R0, S0F000+6   Den SPL-Workarea in S0M000 sichern
      LI R1, S0M000+4
      LI R2, 13

S0M000 MOV S0M000+4, R1+4
      DEC R2
      JNE S0M000+4

S0M000 AB S0M000, S0M000 SPL-Subroutine-Blitzpointer um 4 erhöhen
      MOV S0M000+4, R2   Den Wert nach R2 holen
      MOV R2, 0          ... und die Subroutine aufrufen
      MOV R0, R2+4       Zeiger auf Subroutine-Block in CPU-BAR-PAD
      DEC R0              Rückzugadressen laden
      DEC R0              Wert um vier Bytes dekrementieren

      MOV R0, S0M000     ... und in der Subroutine-Stack schreiben
      MOV S0M000, R0    Beriber über S0 schreiben
      BL S0M000         Sprung in den SPL-Interpreter

Q0M000 RECT S0M000, S0M000 Den SPL-Subroutine-Blitzpointer bereinigen
      MOV S0M000+4, S0M000 Die alten S0F000 wiederherstellen

L0M000 LI R0, S0F000+6   Den alten S0F000 wiederherstellen
      LI R1, S0M000+4
      LI R2, 13

L0M000 MOV S0M000+4, R0+4
      DEC R0
      JNE L0M000
      LAMP L0M000
      RTMP              Vor der Rückführung der Workarea wieder Laden
      RTMP              Rückprung zum aufrufenden Programm

*****

```

Der gute alte TI wird langsam zum Sprachenkünstler. Nach Assembler, Basic, Pascal und FortH ist nun sein kurzer Zeit auch die Programmiersprache C verfügbar. C ermöglicht das strukturierte, modulare Programmieren wie in Pascal; C-Programme laufen jedoch um einiges schneller als vergleichbare Pascal-Programme. Selbst gegenüber dem in der Anwendung eher etwas spröden FortH ist noch ein geringer Geschwindigkeitsvorteil vorhanden.

Der C-Compiler für den TI wurde von dem Amerikaner Clint Pulley entwickelt und liegt mittlerweile in einer zweiten verbesserten Version vor. Erstaunlicherweise genügte zur Entwicklung ein normaler TI 99 mit 32-K-Speichererweiterung, RS-232-Schnittstellenkarte und ein Diskettensystem mit 2 Laufwerken – ein weiterer Beweis für die Leistungsfähigkeit unserer texanischen Rechen- und Spielmaschine. Der C-Compiler entspricht den Spezifikationen des Standard C nach Ron Cain und Jeff Lomicka und läuft auf dem TI mit allen Erweiterungen, die für den Betrieb des Editor/Assembler-Moduls nötig sind. Dies wären 32-K-Byte-Speichererweiterung, Diskettensystem und selbstverständlich das Editor/Assembler-Modul selbst. Empfehlenswert, aber nicht unbedingt nötig, ist ein Drucker samt Schnittstellenkarte. Der C-Compiler versteht ein abgemergertes Standard C, das um viele nützliche Unterprogramm-Bibliotheken mit TI-99-spezifischen Befehlen ergänzt wird. Die Arbeitsweise entspricht dem One-Pass-System, d. h., der Programmcode wird in einem Durchgang auf Fehler geprüft und übersetzt, was den Nachteil hat, daß alle Referenzen zur Compilationszeit definiert sein müssen. Das bei der Übersetzung entstehende Produkt ist noch kein lauffähiger Objectcode, sondern Assembler-Quellcode im DIS/VAR-80-Format. Der lauffähige Objectcode entsteht dann im Assemblerdurchlauf.

Die Arbeit mit dem Compiler ist relativ einfach. Zunächst wird der C-Quelltext mit dem Editor des Assembler-Moduls oder dem TI-Writer eingegeben. Wichtig ist in diesem Zusammenhang nur, daß die erzeugte Textdatei ohne Steuerzeichen (Tabulator u.ä.) im DIS/VAR-80-Format auf der Diskette abgelegt wird. Nach Erstellung des Quelltextes wird der Compiler über die "Run Programm File"-Option des Assembler-Moduls geladen und gestartet. Nach Angabe des Dateinamens für die Quelldatei bietet sich für den Anwender eine interessante Option. Der Compiler fragt nun, ob zum erzeugten Assembler-Quellcode die entsprechende C-Programmzeile als Kommentar hinzugefügt werden soll. Wird diese Option gewählt, dann kann der Programmierer die Auswirkung jeder Programmzeile überprüfen und den entsprechenden Assemblercode nach Bedarf optimieren. Die Ausgabe des Assemblercodes kann sowohl auf einer Diskettendatei als auch auf dem Bildschirm erfolgen. Letzteres ist wesentlich schneller und empfiehlt sich vor allem bei den ersten Durchläufen, wenn noch Tipp-Fehler im Quellcode vorhanden sind.

Sobald ein Fehler während des Compilierens auftritt, stoppt der Compiler, und die entsprechende Zeilennummer wird auf dem Bildschirm angezeigt. Danach geht es mit Tastendruck weiter. Die Arbeit des Compilers kann jederzeit durch FCTN 4 gestoppt werden, so daß bei kleineren Fehlern am Anfang eines Programms nicht der gesamte Compilationsprozeß bis zum bitteren Ende abgewartet werden muß. Nach der erfolgreichen Compilation wird der Assembler aufgerufen und das Ausgabefile des Compilers zum lauffähigen Objectcode assembliert. Das fertige Programm kann mit der Editor/Assembler-Option "Load and Run" geladen und gestartet werden.

Falls Bibliotheken verwendet werden, müssen diese vor dem Programmstart ebenfalls geladen werden. Als Start-Label dient immer das Wort "Start".

Auf den beiden Disketten des Compilers befinden sich außer der eigentlichen Grundausstattung noch viele nützliche Bibliotheken und Utilities. Neben den TI-99-spezifischen Ein- und Ausgaberoutinen und Dateiverwaltungsroutinen wurden beispielsweise auch die Grafikmöglichkeiten des Extended Basic adaptiert. Durch diese Routinen kann man zwischen Grafikmode und 40-Zeichenmode umschalten, Character definieren, Bildschirmfarben setzen sowie Joysticks und Keyboard abfragen. Auch eine der stärksten TI-Waffen, die Sprites, wurden nicht vergessen. Man kann Sprites erzeugen, bewegen, deren Position abfragen und vieles mehr. Die Grafikmöglichkeiten des Extended Basic werden jedoch noch übertrifft, wenn man mit der Bibliothek der Bit-Map-Grafikbefehle arbeitet. Diese bietet zunächst die grundlegenden Befehle, um Pixel zu setzen sowie Linien und Rechtecke zu zeichnen. Eine zweite Befehlsgruppe ist zum Mischen von Text und Bit-Map-Grafik vorgesehen.

Den Nachteil, daß C99 nur die Integer-Rechnung beherrscht, gleicht die Bibliothek mit Floating-Point-Routinen wieder aus. Man sollte diese Berechnungen jedoch nur anwenden, wenn es unbedingt nö-

tig ist, da sie den Programmfluß stark verlangsamen. Die Besitzer eines Sprachsynthesizers können auf einen umfangreichen Wortschatz der Bibliothek "Speech" zurückgreifen. Wenn die mitgelieferten Bibliotheken nicht ausreichen, der kann fehlende Befehle selbst erzeugen, und zwar sowohl in Assembler als auch in C. Kleinere Assemblerprogramme lassen sich direkt im C-Quellcode einfügen.

Ein Handbuch im üblichen Sinne gibt es allerdings nicht. Die wichtigsten Anmerkungen, Bedienungs Hinweise und Beschreibungen sind auf mehreren Textfiles enthalten, die man zur besseren Übersicht ausdrucken sollte (ca. 40 Seiten). Wer allerdings noch nie in C programmiert hat, der muß sich auf jeden Fall ein entsprechendes Lehrbuch kaufen, um die ersten Schritte zu erlernen. Der Compiler ist als Freeware erhältlich, d.h., man muß zunächst nur die Auslagen für Disketten und Arbeitsaufwand bezahlen. Wer später der Meinung ist, der Compiler sei brauchbar, wird gebeten, dem Autor freiwillig 20 Dollar zu übermitteln. Dafür wird der Name des Käufers registriert, und er erhält immer die neuesten Updates. Wer also einen C-Compiler haben möchte, der kann diesen gegen einen frankierten Rückumschlag und 10,- DM Unkostenbeitrag bei mir beziehen.

H.-P. Schwack
Roggkamp 3
3300 Braunschweig

Hexmonitor in C am TI 99/4A

Seit im Frühjahr 1986 in Europa die Existenz eines C-Compilers aus Kanada bekannt wurde und dessen Autor Clint Pulley ihn freundlicherweise als Public-Domain-Software zur Verfügung stellte, hat sich langsam ein kleiner Anwenderkreis dieser Pascal-Tochter gebildet. Das vorliegende Programm soll auf die Existenz von C hinweisen, das am TI ohne weitere Hardware jedem Editor/Assembler-Besitzer zur Verfügung steht.

Enthält Version 1.0 des Compilers noch einige "Bugs", kann man dies von der Version 2.0, die zudem mit Floating-point-Routinen aufwartet, kaum mehr behaupten. Wieder einmal hat der alte TI den Anschluß an die derzeitige Computerwelt gefunden!

"Hexmonitor" gestattet das Screen-orientierte Editieren der Speicherinhalte in RAM, VDP und eventuellen GRAMS. Jeweils 64 Byte werden aufbe-

reitet und können auch mit einfachem Tastendruck zu Papier gebracht werden. Dank der prozeduralen Struktur von C ist jeder Teil des rekursiv aufgebauten Programms nahezu selbstdokumentierend. Es zeigt im Ablauf eine Titelseite und nach Betätigung einer beliebigen Taste eine Kommando-Menüleiste. Nun wird die Eingabe einer vierstelligen Hexadresse erwartet, R als Kennung für ROM/RAM, V für VDP und G für GROM/GRAM-Bereich. Innerhalb des Screens kann frei editiert werden; jedes Kommando wie Vor-, Rückwärtsblättern, neue Adresse oder Ausdruck, schreibt den Bildschirminhalt in den betreffenden RAM-Teil zurück.

Die Anpassung des Programms an andere Ausgabeme-

dien als PIO wird in der Prozedur "hardcopy" durch Austausch der Device in fopen ("DeviceName", "w") erzielt. Die C-Syntax entspricht dem Standard, der von Kernighan/R Ritchie vorgegeben ist. Es existieren zwar Einschränkungen gegenüber einem großen C-System, diese sind aber tolerabel. Grafik-, Sound- und Sprachmöglichkeiten des TI sind durch Basisanaloge Prozeduren voll erschlossen.

Die Arbeitsgeschwindigkeit der erzeugten Programme liegt trotz Programmierkomfort einer echten Hochsprache zwischen Forth und Maschinensprache. Wie in Forth, GPL oder Maschinensprache verzögern Floating-Point-Routinen aber den Programmablauf.

Karl Hagenbucher

Hexmonitor

```

/* HEXMONITOR Version 1.0
 *
 * Author: Hagenbucher, 1986/06/01
 *
 */

#include "hex_ptr.h" /* Bibliotheksumfassende .h
#include "hex_monitor.h"
#include "hex_util.h"
#define _NO_MATH_PRIMES
#include 
#define TRUE 1
#define FALSE 0

void main()
{ char *hex;
  int hex;
  while (getchar() != '\n')
    ;
  if (getchar() == 'R')
    hex = 0;
  else if (getchar() == 'V')
    hex = 0x00000000;
  else if (getchar() == 'G')
    hex = 0x00000000;
  else if (getchar() == 'A')
    hex = 0x00000000;
  else if (getchar() == 'C')
    hex = 0x00000000;
  else if (getchar() == 'M')
    hex = 0x00000000;
  else if (getchar() == 'S')
    hex = 0x00000000;
  else if (getchar() == 'D')
    hex = 0x00000000;
  else if (getchar() == 'E')
    hex = 0x00000000;
  else if (getchar() == 'F')
    hex = 0x00000000;
  else if (getchar() == 'G')
    hex = 0x00000000;
  else if (getchar() == 'H')
    hex = 0x00000000;
  else if (getchar() == 'I')
    hex = 0x00000000;
  else if (getchar() == 'J')
    hex = 0x00000000;
  else if (getchar() == 'K')
    hex = 0x00000000;
  else if (getchar() == 'L')
    hex = 0x00000000;
  else if (getchar() == 'M')
    hex = 0x00000000;
  else if (getchar() == 'N')
    hex = 0x00000000;
  else if (getchar() == 'O')
    hex = 0x00000000;
  else if (getchar() == 'P')
    hex = 0x00000000;
  else if (getchar() == 'Q')
    hex = 0x00000000;
  else if (getchar() == 'R')
    hex = 0x00000000;
  else if (getchar() == 'S')
    hex = 0x00000000;
  else if (getchar() == 'T')
    hex = 0x00000000;
  else if (getchar() == 'U')
    hex = 0x00000000;
  else if (getchar() == 'V')
    hex = 0x00000000;
  else if (getchar() == 'W')
    hex = 0x00000000;
  else if (getchar() == 'X')
    hex = 0x00000000;
  else if (getchar() == 'Y')
    hex = 0x00000000;
  else if (getchar() == 'Z')
    hex = 0x00000000;
  else if (getchar() == 'a')
    hex = 0x00000000;
  else if (getchar() == 'b')
    hex = 0x00000000;
  else if (getchar() == 'c')
    hex = 0x00000000;
  else if (getchar() == 'd')
    hex = 0x00000000;
  else if (getchar() == 'e')
    hex = 0x00000000;
  else if (getchar() == 'f')
    hex = 0x00000000;
  else if (getchar() == 'g')
    hex = 0x00000000;
  else if (getchar() == 'h')
    hex = 0x00000000;
  else if (getchar() == 'i')
    hex = 0x00000000;
  else if (getchar() == 'j')
    hex = 0x00000000;
  else if (getchar() == 'k')
    hex = 0x00000000;
  else if (getchar() == 'l')
    hex = 0x00000000;
  else if (getchar() == 'm')
    hex = 0x00000000;
  else if (getchar() == 'n')
    hex = 0x00000000;
  else if (getchar() == 'o')
    hex = 0x00000000;
  else if (getchar() == 'p')
    hex = 0x00000000;
  else if (getchar() == 'q')
    hex = 0x00000000;
  else if (getchar() == 'r')
    hex = 0x00000000;
  else if (getchar() == 's')
    hex = 0x00000000;
  else if (getchar() == 't')
    hex = 0x00000000;
  else if (getchar() == 'u')
    hex = 0x00000000;
  else if (getchar() == 'v')
    hex = 0x00000000;
  else if (getchar() == 'w')
    hex = 0x00000000;
  else if (getchar() == 'x')
    hex = 0x00000000;
  else if (getchar() == 'y')
    hex = 0x00000000;
  else if (getchar() == 'z')
    hex = 0x00000000;
  else if (getchar() == '0')
    hex = 0x00000000;
  else if (getchar() == '1')
    hex = 0x00000000;
  else if (getchar() == '2')
    hex = 0x00000000;
  else if (getchar() == '3')
    hex = 0x00000000;
  else if (getchar() == '4')
    hex = 0x00000000;
  else if (getchar() == '5')
    hex = 0x00000000;
  else if (getchar() == '6')
    hex = 0x00000000;
  else if (getchar() == '7')
    hex = 0x00000000;
  else if (getchar() == '8')
    hex = 0x00000000;
  else if (getchar() == '9')
    hex = 0x00000000;
  else if (getchar() == '+')
    hex = 0x00000000;
  else if (getchar() == '-')
    hex = 0x00000000;
  else if (getchar() == '=')
    hex = 0x00000000;
  else if (getchar() == '*')
    hex = 0x00000000;
  else if (getchar() == '/')
    hex = 0x00000000;
  else if (getchar() == '%')
    hex = 0x00000000;
  else if (getchar() == '^')
    hex = 0x00000000;
  else if (getchar() == '&')
    hex = 0x00000000;
  else if (getchar() == '|')
    hex = 0x00000000;
  else if (getchar() == '~')
    hex = 0x00000000;
  else if (getchar() == '!')
    hex = 0x00000000;
  else if (getchar() == '@')
    hex = 0x00000000;
  else if (getchar() == '#')
    hex = 0x00000000;
  else if (getchar() == '$')
    hex = 0x00000000;
  else if (getchar() == '%')
    hex = 0x00000000;
  else if (getchar() == '&')
    hex = 0x00000000;
  else if (getchar() == '*')
    hex = 0x00000000;
  else if (getchar() == '+')
    hex = 0x00000000;
  else if (getchar() == '-')
    hex = 0x00000000;
  else if (getchar() == '=')
    hex = 0x00000000;
  else if (getchar() == '<')
    hex = 0x00000000;
  else if (getchar() == '>')
    hex = 0x00000000;
  else if (getchar() == '?')
    hex = 0x00000000;
  else if (getchar() == '[')
    hex = 0x00000000;
  else if (getchar() == ']')
    hex = 0x00000000;
  else if (getchar() == '{')
    hex = 0x00000000;
  else if (getchar() == '}')
    hex = 0x00000000;
  else if (getchar() == '|')
    hex = 0x00000000;
  else if (getchar() == '<')
    hex = 0x00000000;
  else if (getchar() == '>')
    hex = 0x00000000;
  else if (getchar() == '?')
    hex = 0x00000000;
  else if (getchar() == '!')
    hex = 0x00000000;
  else if (getchar() == '@')
    hex = 0x00000000;
  else if (getchar() == '#')
    hex = 0x00000000;
  else if (getchar() == '$')
    hex = 0x00000000;
  else if (getchar() == '%')
    hex = 0x00000000;
  else if (getchar() == '&')
    hex = 0x00000000;
  else if (getchar() == '*')
    hex = 0x00000000;
  else if (getchar() == '+')
    hex = 0x00000000;
  else if (getchar() == '-')
    hex = 0x00000000;
  else if (getchar() == '=')
    hex = 0x00000000;
  else if (getchar() == '<')
    hex = 0x00000000;
  else if (getchar() == '>')
    hex = 0x00000000;
  else if (getchar() == '?')
    hex = 0x00000000;
  else if (getchar() == '[')
    hex = 0x00000000;
  else if (getchar() == ']')
    hex = 0x00000000;
  else if (getchar() == '{')
    hex = 0x00000000;
  else if (getchar() == '}')
    hex = 0x00000000;
  else if (getchar() == '|')
    hex = 0x00000000;
  else if (getchar() == '<')
    hex = 0x00000000;
  else if (getchar() == '>')
    hex = 0x00000000;
  else if (getchar() == '?')
    hex = 0x00000000;
  else if (getchar() == '!')
    hex = 0x00000000;
  else if (getchar() == '@')
    hex = 0x00000000;
  else if (getchar() == '#')
    hex = 0x00000000;
  else if (getchar() == '$')
    hex = 0x00000000;
  else if (getchar() == '%')
    hex = 0x00000000;
  else if (getchar() == '&')
    hex = 0x00000000;
  else if (getchar() == '*')
    hex = 0x00000000;
  else if (getchar() == '+')
    hex = 0x00000000;
  else if (getchar() == '-')
    hex = 0x00000000;
  else if (getchar() == '=')
    hex = 0x00000000;
  else if (getchar() == '<')
    hex = 0x00000000;
  else if (getchar() == '>')
    hex = 0x00000000;
  else if (getchar() == '?')
    hex = 0x00000000;
  else if (getchar() == '[')
    hex = 0x00000000;
  else if (getchar() == ']')
    hex = 0x00000000;
  else if (getchar() == '{')
    hex = 0x00000000;
  else if (getchar() == '}')
    hex = 0x00000000;
  else if (getchar() == '|')
    hex = 0x00000000;
  else if (getchar() == '<')
    hex = 0x00000000;
  else if (getchar() == '>')
    hex = 0x00000000;
  else if (getchar() == '?')
    hex = 0x00000000;
  else if (getchar() == '!')
    hex = 0x00000000;
  else if (getchar() == '@')
    hex = 0x00000000;
  else if (getchar() == '#')
    hex = 0x00000000;
  else if (getchar() == '$')
    hex = 0x00000000;
  else if (getchar() == '%')
    hex = 0x00000000;
  else if (getchar() == '&')
    hex = 0x00000000;
  else if (getchar() == '*')
    hex = 0x00000000;
  else if (getchar() == '+')
    hex = 0x00000000;
  else if (getchar() == '-')
    hex = 0x00000000;
  else if (getchar() == '=')
    hex = 0x00000000;
  else if (getchar() == '<')
    hex = 0x00000000;
  else if (getchar() == '>')
    hex = 0x00000000;
  else if (getchar() == '?')
    hex = 0x00000000;
  else if (getchar() == '[')
    hex = 0x00000000;
  else if (getchar() == ']')
    hex = 0x00000000;
  else if (getchar() == '{')
    hex = 0x00000000;
  else if (getchar() == '}')
    hex = 0x00000000;
  else if (getchar() == '|')
    hex = 0x00000000;
  else if (getchar() == '<')
    hex = 0x00000000;
  else if (getchar() == '>')
    hex = 0x00000000;
  else if (getchar() == '?')
    hex = 0x00000000;
  else if (getchar() == '!')
    hex = 0x00000000;
  else if (getchar() == '@')
    hex = 0x00000000;
  else if (getchar() == '#')
    hex = 0x00000000;
  else if (getchar() == '$')
    hex = 0x00000000;
  else if (getchar() == '%')
    hex = 0x00000000;
  else if (getchar() == '&')
    hex = 0x00000000;
  else if (getchar() == '*')
    hex = 0x00000000;
  else if (getchar() == '+')
    hex = 0x00000000;
  else if (getchar() == '-')
    hex = 0x00000000;
  else if (getchar() == '=')
    hex = 0x00000000;
  else if (getchar() == '<')
    hex = 0x00000000;
  else if (getchar() == '>')
    hex = 0x00000000;
  else if (getchar() == '?')
    hex = 0x00000000;
  else if (getchar() == '[')
    hex = 0x00000000;
  else if (getchar() == ']')
    hex = 0x00000000;
  else if (getchar() == '{')
    hex = 0x00000000;
  else if (getchar() == '}')
    hex = 0x00000000;
  else if (getchar() == '|')
    hex = 0x00000000;
  else if (getchar() == '<')
    hex = 0x00000000;
  else if (getchar() == '>')
    hex = 0x00000000;
  else if (getchar() == '?')
    hex = 0x00000000;
  else if (getchar() == '!')
    hex = 0x00000000;
  else if (getchar() == '@')
    hex = 0x00000000;
  else if (getchar() == '#')
    hex = 0x00000000;
  else if (getchar() == '$')
    hex = 0x00000000;
  else if (getchar() == '%')
    hex = 0x00000000;
  else if (getchar() == '&')
    hex = 0x00000000;
  else if (getchar() == '*')
    hex = 0x00000000;
  else if (getchar() == '+')
    hex = 0x00000000;
  else if (getchar() == '-')
    hex = 0x00000000;
  else if (getchar() == '=')
    hex = 0x00000000;
  else if (getchar() == '<')
    hex = 0x00000000;
  else if (getchar() == '>')
    hex = 0x00000000;
  else if (getchar() == '?')
    hex = 0x00000000;
  else if (getchar() == '[')
    hex = 0x00000000;
  else if (getchar() == ']')
    hex = 0x00000000;
  else if (getchar() == '{')
    hex = 0x00000000;
  else if (getchar() == '}')
    hex = 0x00000000;
  else if (getchar() == '|')
    hex = 0x00000000;
  else if (getchar() == '<')
    hex = 0x00000000;
  else if (getchar() == '>')
    hex = 0x00000000;
  else if (getchar() == '?')
    hex = 0x00000000;
  else if (getchar() == '!')
    hex = 0x00000000;
  else if (getchar() == '@')
    hex = 0x00000000;
  else if (getchar() == '#')
    hex = 0x00000000;
  else if (getchar() == '$')
    hex = 0x00000000;
  else if (getchar() == '%')
    hex = 0x00000000;
  else if (getchar() == '&')
    hex = 0x00000000;
  else if (getchar() == '*')
    hex = 0x00000000;
  else if (getchar() == '+')
    hex = 0x00000000;
  else if (getchar() == '-')
    hex = 0x00000000;
  else if (getchar() == '=')
    hex = 0x00000000;
  else if (getchar() == '<')
    hex = 0x00000000;
  else if (getchar() == '>')
    hex = 0x00000000;
  else if (getchar() == '?')
    hex = 0x00000000;
  else if (getchar() == '[')
    hex = 0x00000000;
  else if (getchar() == ']')
    hex = 0x00000000;
  else if (getchar() == '{')
    hex = 0x00000000;
  else if (getchar() == '}')
    hex = 0x00000000;
  else if (getchar() == '|')
    hex = 0x00000000;
  else if (getchar() == '<')
    hex = 0x00000000;
  else if (getchar() == '>')
    hex = 0x00000000;
  else if (getchar() == '?')
    hex = 0x00000000;
  else if (getchar() == '!')
    hex = 0x00000000;
  else if (getchar() == '@')
    hex = 0x00000000;
  else if (getchar() == '#')
    hex = 0x00000000;
  else if (getchar() == '$')
    hex = 0x00000000;
  else if (getchar() == '%')
    hex = 0x00000000;
  else if (getchar() == '&')
    hex = 0x00000000;
  else if (getchar() == '*')
    hex = 0x00000000;
  else if (getchar() == '+')
    hex = 0x00000000;
  else if (getchar() == '-')
    hex = 0x00000000;
  else if (getchar() == '=')
    hex = 0x00000000;
  else if (getchar() == '<')
    hex = 0x00000000;
  else if (getchar() == '>')
    hex = 0x00000000;
  else if (getchar() == '?')
    hex = 0x00000000;
  else if (getchar() == '[')
    hex = 0x00000000;
  else if (getchar() == ']')
    hex = 0x00000000;
  else if (getchar() == '{')
    hex = 0x00000000;
  else if (getchar() == '}')
    hex = 0x00000000;
  else if (getchar() == '|')
    hex = 0x00000000;
  else if (getchar() == '<')
    hex = 0x00000000;
  else if (getchar() == '>')
    hex = 0x00000000;
  else if (getchar() == '?')
    hex = 0x00000000;
  else if (getchar() == '!')
    hex = 0x00000000;
  else if (getchar() == '@')
    hex = 0x00000000;
  else if (getchar() == '#')
    hex = 0x00000000;
  else if (getchar() == '$')
    hex = 0x00000000;
  else if (getchar() == '%')
    hex = 0x00000000;
  else if (getchar() == '&')
    hex = 0x00000000;
  else if (getchar() == '*')
    hex = 0x00000000;
  else if (getchar() == '+')
    hex = 0x00000000;
  else if (getchar() == '-')
    hex = 0x00000000;
  else if (getchar() == '=')
    hex = 0x00000000;
  else if (getchar() == '<')
    hex = 0x00000000;
  else if (getchar() == '>')
    hex = 0x00000000;
  else if (getchar() == '?')
    hex = 0x00000000;
  else if (getchar() == '[')
    hex = 0x00000000;
  else if (getchar() == ']')
    hex = 0x00000000;
  else if (getchar() == '{')
    hex = 0x00000000;
  else if (getchar() == '}')
    hex = 0x00000000;
  else if (getchar() == '|')
    hex = 0x00000000;
  else if (getchar() == '<')
    hex = 0x00000000;
  else if (getchar() == '>')
    hex = 0x00000000;
  else if (getchar() == '?')
    hex = 0x00000000;
  else if (getchar() == '!')
    hex = 0x00000000;
  else if (getchar() == '@')
    hex = 0x00000000;
  else if (getchar() == '#')
    hex = 0x00000000;
  else if (getchar() == '$')
    hex = 0x00000000;
  else if (getchar() == '%')
    hex = 0x00000000;
  else if (getchar() == '&')
    hex = 0x00000000;
  else if (getchar() == '*')
    hex = 0x00000000;
  else if (getchar() == '+')
    hex = 0x00000000;
  else if (getchar() == '-')
    hex = 0x00000000;
  else if (getchar() == '=')
    hex = 0x00000000;
  else if (getchar() == '<')
    hex = 0x00000000;
  else if (getchar() == '>')
    hex = 0x00000000;
  else if (getchar() == '?')
    hex = 0x00000000;
  else if (getchar() == '[')
    hex = 0x00000000;
  else if (getchar() == ']')
    hex = 0x00000000;
  else if (getchar() == '{')
    hex = 0x00000000;
  else if (getchar() == '}')
    hex = 0x00000000;
  else if (getchar() == '|')
    hex = 0x00000000;
  else if (getchar() == '<')
    hex = 0x00000000;
  else if (getchar() == '>')
    hex = 0x00000000;
  else if (getchar() == '?')
    hex = 0x00000000;
  else if (getchar() == '!')
    hex = 0x00000000;
  else if (getchar() == '@')
    hex = 0x00000000;
  else if (getchar() == '#')
    hex = 0x00000000;
  else if (getchar() == '$')
    hex = 0x00000000;
  else if (getchar() == '%')
    hex = 0x00000000;
  else if (getchar() == '&')
    hex = 0x00000000;
  else if (getchar() == '*')
    hex = 0x00000000;
  else if (getchar() == '+')
    hex = 0x00000000;
  else if (getchar() == '-')
    hex = 0x00000000;
  else if (getchar() == '=')
    hex = 0x00000000;
  else if (getchar() == '<')
    hex = 0x00000000;
  else if (getchar() == '>')
    hex = 0x00000000;
  else if (getchar() == '?')
    hex = 0x00000000;
  else if (getchar() == '[')
    hex = 0x00000000;
  else if (getchar() == ']')
    hex = 0x00000000;
  else if (getchar() == '{')
    hex = 0x00000000;
  else if (getchar() == '}')
    hex = 0x00000000;
  else if (getchar() == '|')
    hex = 0x00000000;
  else if (getchar() == '<')
    hex = 0x00000000;
  else if (getchar() == '>')
    hex = 0x00000000;
  else if (getchar() == '?')
    hex = 0x00000000;
  else if (getchar() == '!')
    hex = 0x00000000;
  else if (getchar() == '@')
    hex = 0x00000000;
  else if (getchar() == '#')
    hex = 0x00000000;
  else if (getchar() == '$')
    hex = 0x00000000;
  else if (getchar() == '%')
    hex = 0x00000000;
  else if (getchar() == '&')
    hex = 0x00000000;
  else if (getchar() == '*')
    hex = 0x00000000;
  else if (getchar() == '+')
    hex = 0x00000000;
  else if (getchar() == '-')
    hex = 0x00000000;
  else if (getchar() == '=')
    hex = 0x00000000;
  else if (getchar() == '<')
    hex = 0x00000000;
  else if (getchar() == '>')
    hex = 0x00000000;
  else if (getchar() == '?')
    hex = 0x00000000;
  else if (getchar() == '[')
    hex = 0x00000000;
  else if (getchar() == ']')
    hex = 0x00000000;
  else if (getchar() == '{')
    hex = 0x00000000;
  else if (getchar() == '}')
    hex = 0x00000000;
  else if (getchar() == '|')
    hex = 0x00000000;
  else if (getchar() == '<')
    hex = 0x00000000;
  else if (getchar() == '>')
    hex = 0x00000000;
  else if (getchar() == '?')
    hex = 0x00000000;
  else if (getchar() == '!')
    hex = 0x00000000;
  else if (getchar() == '@')
    hex = 0x00000000;
  else if (getchar() == '#')
    hex = 0x00000000;
  else if (getchar() == '$')
    hex = 0x00000000;
  else if (getchar() == '%')
    hex = 0x00000000;
  else if (getchar() == '&')
    hex = 0x00000000;
  else if (getchar() == '*')
    hex = 0x00000000;
  else if (getchar() == '+')
    hex = 0x00000000;
  else if (getchar() == '-')
    hex = 0x00000000;
  else if (getchar() == '=')
    hex = 0x00000000;
  else if (getchar() == '<')
    hex = 0x00000000;
  else if (getchar() == '>')
    hex = 0x00000000;
  else if (getchar() == '?')
    hex = 0x00000000;
  else if (getchar() == '[')
    hex = 0x00000000;
  else if (getchar() == ']')
    hex = 0x00000000;
  else if (getchar() == '{')
    hex = 0x00000000;
  else if (getchar() == '}')
    hex = 0x00000000;
  else if (getchar() == '|')
    hex = 0x00000000;
  else if (getchar() == '<')
    hex = 0x00000000;
  else if (getchar() == '>')
    hex = 0x00000000;
  else if (getchar() == '?')
    hex = 0x00000000;
  else if (getchar() == '!')
    hex = 0x00000000;
  else if (getchar() == '@')
    hex = 0x00000000;
  else if (getchar() == '#')
    hex = 0x00000000;
  else if (getchar() == '$')
    hex = 0x00000000;
  else if (getchar() == '%')
    hex = 0x00000000;
  else if (getchar() == '&')
    hex = 0x00000000;
  else if (getchar() == '*')
    hex = 0x00000000;
  else if (getchar() == '+')
    hex = 0x00000000;
  else if (getchar() == '-')
    hex = 0x00000000;
  else if (getchar() == '=')
    hex = 0x00000000;
  else if (getchar() == '<')
    hex = 0x00000000;
  else if (getchar() == '>')
    hex = 0x00000000;
  else if (getchar() == '?')
    hex = 0x00000000;
  else if (getchar() == '[')
    hex = 0x00000000;
  else if (getchar() == ']')
    hex = 0x00000000;
  else if (getchar() == '{')
    hex = 0x00000000;
  else if (getchar() == '}')
    hex = 0x00000000;
  else if (getchar() == '|')
    hex = 0x00000000;
  else if (getchar() == '<')
    hex = 0x00000000;
  else if (getchar() == '>')
    hex = 0x00000000;
  else if (getchar() == '?')
    hex = 0x00000000;
  else if (getchar() == '!')
    hex = 0x00000000;
  else if (getchar() == '@')
    hex = 0x00000000;
  else if (getchar() == '#')
    hex = 0x00000000;
  else if (getchar() == '$')
    hex = 0x00000000;
  else if (getchar() == '%')
    hex = 0x00000000;
  else if (getchar() == '&')
    hex = 0x00000000;
  else if (getchar() == '*')
    hex = 0x00000000;
  else if (getchar() == '+')
    hex = 0x00000000;
  else if (getchar() == '-')
    hex = 0x00000000;
  else if (getchar() == '=')
    hex = 0x00000000;
  else if (getchar() == '<')
    hex = 0x0000000
```


SR002

In diesem Spiel für Konsole und Extended Basic geht es um die Segonen. Sie haben ein Raumgitter gebaut, mit dem sie der Galaxis die Energie entziehen, was deren Untergang bedeuten würde. Die Regierung der Vereinten Planeten rüstet

Sie, den Spieler, mit dem besten Kampfschiff aus und erteilt Ihnen den Auftrag, das Raumgitter zu vernichten. Das Schicksal der Galaxis liegt nun in Ihren Händen.

Das Raumschiff ist mit Gravobomben bewaffnet, die mit einem Zeitzähler ausgestattet sind. Jetzt müssen Sie die Bom-

ben so platzieren, daß sie über den Segonen Schiffen explodieren. Zur Zerstörung einer Gitterebene benötigen Sie eine Elementar bombe der Segonen. Dazu ist das gelbe Segonen schiff zu vernichten. An seiner Stelle erscheint dann eine Elementar bombe, die Sie auffangen müssen. Nun bringen Sie das Zielkreuz über einen der Energiekonverter und feuern

die Bombe ab. Sind die Konverter zerstört, gelangt man zur nächsten Ebene.

Gesteuert wird mit Joystick 1, wobei der Start mittels Feuer taste erfolgt. Im Verlauf des Spiels löst die Feuertaste dann die Gravobomben, Joystick unten die Elementarbomben aus.

Roland Toonen

```

100 CALL CLEAR : CALL SCREEN(2)
110 CALL CHAR(58,"7E7E007C7E067E7C",59,"7C7E067C7C6B6C66",60,"7C7E066666667E3C")
120 FOR I=1 TO 8
130 CALL COLOR(1,3,1)
140 NEXT I
150 DISPLAY AT(11,12):"ROLAND'S"
160 DISPLAY AT(14,13):":<<"
170 CALL MAGNIFY(3): CALL SCREEN(2)
180 FOR I=40 TO 139
190 READ A$
200 CALL CHAR(I,A$): NEXT I
210 DATA 7E7E007C7E067E7C,7E7E006060607E3C,7C7E066666667E3C,7C7E067C7C6B6C66,7E7E007C7E067E7E,08,0,0,1C3E3A3A3A3E3E1C,0E1E3E3E06060606
220 DATA 1C3E26060C1B3E3E,1C3E260E0E263E1C,0E1A3A3A3E060606,1E20203C063E3E1C,1C2203C2E2E3E1C,1C3E320206060606,1C3A3A1C3A3A3A1C
230 DATA 1C3E3A3A1E02221C,0,0,0,0,0,0,8888888888888888,8888888888888888,1111111111111111,1111111111111111,0F1F3F7FFFFFFF
240 DATA FFFFFFFF0E0C080,FOFBFCFEFFFFFFF,FFFFFFFF0F070301,2424242424242424,0,0,0,0F182B4A8888888888,8888888F90A0C080,F018D45211D1D111
250 DATA 51D111F109050301,0101010303030307,0F0F1F3F7FFFFFFF3C2,808080C0C0C0C0E0,FOFBFCFEFFFFFFF43,0000000001010103,070F1F3F7F796000
260 DATA 0000000080808080,EOFOFBFCFE9E0600,000000000000001,03070F1F3D300000,00000000000080,C0E0FOFBFCDC0000,0000000001010101
270 DATA 0202,00,8080,003D7C6160610054,54006160617C3D00,00BC3E860686002A,2A008606863EBC00,1C3477F0F7B7D7E7,F7B6D6E7F077341C
280 DATA 382CEE0FEFEDBE7,EF6D6E70FEE2C38,040A1B3B7A5B6A7B,7B7B5A6A3B1B0A06,20508BDCSEDA56DE,DEDESAS6DC88060
290 DATA 16367FEBABCBEBA,ABCBEF6C3C1C080B,686FE17D5D3D757,D715F3363C3B1010,1C3474F7B7B7B7B7,567637341C18080B,382CEEFEDEDEB
300 DATA 5A6EEC2C38181010,0000000103060D0F,070702030101,00000080C06080F0,E0E040C08080,7E7E007C7E067E7C,7C7E067C7C6B6C66,7C7E066666667E3C,0
310 DATA 3C7EFFFFFFF381,0,0,0,0040303C1F1E0F0F,0F0F1D1E38200000,0000000C18FBF0E0,DOFOFBFC3C0C0200,0000101C0E070707,080F1F1F3C304000
320 DATA 00020C3CF87BF0F0,EOFOFBFC3B0C0000,00302C2211080808,14102020434C80C0,030D32C204840808,10280804C4320E00
330 FOR I=1 TO 12
340 PRINT " @B H H @B" : PRINT " AC H H AC" : NEXT I
350 DISPLAY AT(12,13)SIZE(5):"xxyz"
360 DISPLAY AT(1,13)SIZE(5):"()*+," : DISPLAY AT(3,12)SIZE(7):"0000000" : DISPLAY AT(5,14)SIZE(3):"000"
370 FOR I=1 TO 40
380 A=INT((24-1+I)*RND)+1 : B=INT((32-1+I)*RND)+1 : CALL GCHAR(A,B,C): IF C>3 THEN 380 : CALL HCHAR(A,B,45)
390 NEXT I
400 CALL COLOR(5,15,5): CALL COLOR(6,15,7): CALL COLOR(2,11,1): CALL COLOR(3,14,1): CALL COLOR(4,14,1): GOSUB 1470
410 GOTO 1410
420 CALL MUSIC : CALL SPRITE(#1,80,16,160,128): CALL SPRITE(#8,96,1,90,128): CALL SPRITE(#2,100,8,1,128,20,-2)
430 CALL COLOR(5,15,5): CALL COLOR(6,15,7): DISPLAY AT(12,13)SIZE(5):" "
440 LJ=1 : PU=1000000 : RU=0 : SI=1005 : O=0 : E=0 : PU1=0 : WD=128 : R1=0 : R2=0 : R3=0 : R6=40 : R7=80 : RB=10

```

```

450 CALL PUNKTE(PU):: CALL SCHIFF(SI)
460 REM
470 R1=R1+1 :: R2=R2+1 :: R3=R3+1
480 IF R1=R6 THEN 490 :: IF R3=R8 THEN 520 :: IF R2>R7 THEN 500 :: GOTO 530
490 R1=R1+5 :: CALL SPRITE(#3,108,8,1,W0,20,2):: GOTO 480
500 IF LJ=1 THEN 510 :: CALL POSITION(#4,L1,L2):: IF L1>190 AND L1<=230 THEN 51
0 ELSE 530
510 LJ=0 :: CALL DELSPRITE(#4):: CALL POSITION(#1,L3,L4):: CALL SPRITE(#4,112,8,
1,L4,20,0):: GOTO 530
520 R3=R3+5 :: W2=INT((154-99+1)*RND)+99 :: CALL SPRITE(#5,104,11,1,W2,15,0):: G
OTO 480
530 D0=10 :: CALL KEY(1,K,S):: IF K=18 THEN 620
540 DE=10 :: CALL JOYST(1,X,Y):: CALL MOTION(#1,0,X#4):: CALL MOTION(#8,0,X#4)::
IF Y=-4 THEN 610
550 CALL COINC(#1,#2,14,T1):: CALL COINC(#1,#3,14,T2)
560 CALL COINC(#1,#4,14,T3):: CALL COINC(#1,#5,14,T4)
570 IF T1=-1 OR T2=-1 THEN 710 :: IF T3=-1 THEN 710 :: IF T4=-1 THEN 800
580 CALL POSITION(#1,P1,P2):: CALL LOCATE(#8,90,P2)
590 D0=0 :: CALL KEY(1,K,S):: IF K=18 THEN 620
600 DE=0 :: CALL JOYST(1,X,Y):: CALL MOTION(#1,0,X#4):: CALL MOTION(#8,0,X#4)::
IF Y=-4 THEN 610 ELSE 460
610 IF D=1 THEN 830 :: IF DE=10 THEN 550 :: IF DE=0 THEN 460
620 CALL POSITION(#1,A,B):: CALL SPRITE(#9,92,15,A,B,-100,0)
630 CALL SOUND(800,-7,4):: CALL COINC(#1,#2,14,T1):: CALL COINC(#1,#3,14,T2):: C
ALL COINC(#1,#4,14,T3):: IF T1=-1 THEN 710
640 IF T2=-1 OR T3=-1 THEN 710 :: CD=8 :: CALL MOTION(#9,15,0)
650 FOR I=128 TO 140 STEP 4
660 CD=CD+1 :: CALL PATTERN(#9,I):: CALL COLOR(#9,CD):: NEXT I
670 CALL JOYST(1,X,Y):: CALL MOTION(#1,0,X#4):: CALL MOTION(#8,0,X#4)
680 CALL COINC(#9,#2,10,T1):: CALL COINC(#9,#3,10,T2):: CALL COINC(#9,#4,10,T3)::
CALL COINC(#9,#5,10,T4)
690 IF T1=-1 THEN 1190 :: IF T2=-1 THEN 1250 :: IF T3=-1 THEN 1300 :: IF T4=-1 T
HEN 1360
700 IF D0=10 THEN 550 :: GOTO 600
710 SI=SI-1 :: CD=4 :: CALL DELSPRITE(#9,#5):: R3=0 :: D=0 :: CALL COLOR(#8,1)
720 FOR D=1 TO 3
730 FOR I=128 TO 140 STEP 4
740 CD=CD+1 :: CALL PATTERN(#1,I):: CALL COLOR(#1,CD):: CALL SOUND(70,-7,4)
750 NEXT I :: NEXT D
760 CALL DELSPRITE(#1)
770 CALL SCHIFF(SI):: IF SI<=1000.5 THEN 1410
780 FOR I=1 TO 300 :: NEXT I
790 CALL SPRITE(#1,80,16,160,128):: GOTO 550
800 IF WX=10 THEN 810 ELSE 710
810 WX=0 :: CALL DELSPRITE(#5)
820 D=1 :: CALL COLOR(#8,11):: GOTO 550
830 R3=0 :: CALL MOTION(#8,15,0):: CALL COINC(#8,#15,10,T1):: CALL COINC(#8,#16,
10,T2)
840 IF T1=-1 THEN 850 :: IF T2=-1 THEN 900 :: D=0 :: CALL COLOR(#8,1):: GOTO 550
850 FOR I=2 TO 14
860 CALL COLOR(#15,I):: CALL SOUND(40,-5,4):: NEXT I
870 CALL PATTERN(#15,68):: CALL COLOR(#15,2):: CALL COLOR(#8,1)
880 PU=PU+175 :: CALL PUNKTE(PU)
890 D=0 :: E=E+1 :: IF E>=2 THEN 950 ELSE 550
900 FOR I=1 TO 16
910 CALL COLOR(#16,I):: CALL SOUND(50,-5,4):: NEXT I
920 CALL PATTERN(#16,68):: CALL COLOR(#16,2):: CALL COLOR(#8,1)
930 PU=PU+175 :: CALL PUNKTE(PU)
940 D=0 :: E=E+1 :: IF E>=2 THEN 950 ELSE 550
950 LJ=1 :: R1=0 :: R2=0 :: R3=0 :: E=0 :: RU=RJ+1 :: IF RU>=5 THEN RU=1
960 R6=R6-5 :: IF R6<=10 THEN R6=10
970 R7=R7-5 :: IF R7<=15 THEN R7=15
980 R8=R8+2 :: IF R8>=60 THEN R8=60
990 CALL DELSPRITE(#2,#3,#4,#5)
1000 CALL POSITION(#1,P1,P2):: IF P2<=125 THEN 1010 :: IF P2>=129 THEN 1020 :: G
OTO 1030
1010 CALL MOTION(#1,0,4):: GOTO 1000
1020 CALL MOTION(#1,0,-4):: GOTO 1000

```

```

1030 CALL MOTION(#1,-7,0)
1040 FOR I=80 TO 91
1050 CALL PATTERN(#1,I)
1060 FOR O=1 TO 50 :: NEXT O :: NEXT I
1070 CALL DELSPRITE(#1)
1080 FOR I=3 TO 16
1090 CALL SOUND(-200,200,5,190,4,180,3):: CALL COLOR(5,15,I):: NEXT I
1100 IF RU=1 THEN 1110 :: IF RU=2 THEN 1120 :: IF RU=3 THEN 1130 :: IF RU=4 THEN
1140
1110 CALL COLOR(5,15,7):: CALL COLOR(6,15,13):: GOTO 1150
1120 CALL COLOR(5,15,13):: CALL COLOR(6,15,14):: GOTO 1150
1130 CALL COLOR(5,15,14):: CALL COLOR(6,15,5):: GOTO 1150
1140 CALL COLOR(5,15,5):: CALL COLOR(6,15,7):: GOTO 1150
1150 CALL PATTERN(#15,76):: CALL PATTERN(#16,76)
1160 PUI=PUI+75 :: PU=PU+PUI :: CALL PUNKTE(PU)
1170 SI=SI+0.5 :: CALL SCHIFF(SI)
1180 CALL SPRITE(#1,80,16,160,128):: CALL SPRITE(#2,100,8,1,128,20,-2):: GOTO 55
0
1190 CALL DELSPRITE(#2)
1200 PU=PU+75 :: CALL PUNKTE(PU)
1210 FOR I=10 TO 0 STEP -2
1220 CALL SOUND(50,-6,I):: NEXT I
1230 W=INT((240-15+1)*RND)+15
1240 CALL SPRITE(#2,100,8,221,W,20,-2):: GOTO 700
1250 CALL DELSPRITE(#3):: R1=0
1260 PU=PU+100 :: CALL PUNKTE(PU)
1270 FOR I=10 TO 0 STEP -2
1280 CALL SOUND(50,-6,I):: NEXT I
1290 WD=INT((240-20+1)*RND)+20 :: GOTO 700
1300 CALL DELSPRITE(#4):: R2=0
1310 LJ=1
1320 PU=PU+125 :: CALL PUNKTE(PU)
1330 FOR I=10 TO 0 STEP -2
1340 CALL SOUND(50,-6,I):: NEXT I
1350 GOTO 700
1360 WX=10
1370 PU=PU+200 :: CALL PUNKTE(PU)
1380 FOR I=10 TO 0 STEP -2
1390 CALL SOUND(70,-5,I):: NEXT I
1400 CALL PATTERN(#5,116):: GOTO 700
1410 DISPLAY AT(12,13)SIZE(5):"xyzz"
1420 RT=3
1430 RT=RT+1 :: IF RT>=14 THEN 1420 :: CALL COLOR(12,RT,I)
1440 CALL KEY(1,K,S):: IF K=18 THEN 1450 ELSE 1430
1450 CALL DELSPRITE(#2,#3,#4,#5):: CALL PATTERN(#15,76):: CALL PATTERN(#16,76)
1460 GOTO 420
1470 CALL SPRITE(#10,68,2,200,57,15,0):: CALL SPRITE(#15,76,2,88,57,15,0)
1480 CALL SPRITE(#11,68,2,150,193,15,0):: CALL SPRITE(#16,76,2,32,193,15,0)
1490 CALL SPRITE(#12,124,2,10,89,6,0)
1500 CALL SPRITE(#13,124,2,130,169,6,0)
1510 RETURN
1520 SUB PUNKTE(PU)
1530 B$=STR$(PU):: DISPLAY AT(3,12)SIZE(7):SEG$(B$,2,7)
1540 SUBEND
1550 SUB SCHIFF(SI)
1560 C$=STR$(SI):: DISPLAY AT(5,14)SIZE(3):SEG$(C$,2,3)
1570 SUBEND
1580 SUB MUSIC
1590 RESTORE 1640
1600 FOR I=1 TO 16
1610 READ D,T
1620 CALL SOUND(25*K,D,T,2,T+2,7,T*0.5,5)
1630 NEXT I
1640 DATA 2,600,5,800,5,800,5,800,2,600,5,800,5,800,5,800,2,600,5,800
1650 DATA 2,600,5,800,2,600,5,800,5,800,5,800,1,1500
1660 SUBEND

```

U-Boot-Schlacht

Dieses Spiel läuft auf dem TI mit Extended Basic. Zu Beginn erscheint das Titelbild. Um zu starten, muß die ENTER-Taste betätigt werden.

Zuerst zum Aufbau des Spielfeldes.

Sternchen (CHR\$(42)): Seesterne, durch die mit dem U-Boot hindurchgefahren werden kann. Abgeschossene Torpedos explodieren jedoch darin, so daß Torpedo und Seestern vernichtet werden. Grüne Mauern: Befestigungen, die nicht überfahren werden können. Beim Zusammenprall des U-Bootes mit der Mauer ist das Schiff verloren. Torpedos explodieren hier genauso wie bei den Seesternen. Gelbe Längsstriche: Schleusen, die durchfahren und abgeschlossen werden können.

Während des Spiels ist zeitweise ein Signal zu hören. Gleichzeitig erscheint ein T auf der linken bzw. rechten Seite des Bildschirms, das den entsprechenden Spieler darauf hinweist, daß sein U-Boot so bald wie möglich aufgetankt werden muß. Dazu muß es an den schwarzen Tank, der sich neben dem T befindet, herangeführt werden. Sollte dieser vom Gegner zerstört worden sein, so ist ein Auftanken nicht mehr möglich. Dies hat zur Folge, daß das U-Boot sich bald selbst zerstört. Dasselbe geschieht auch, wenn der Spieler nach dem Signal nicht rechtzeitig den Tank erreicht. Normalerweise ist dazu jedoch genug Zeit vorhanden.

Spiller 1 spielt auf der linken Seite. Seine U-Boote und Tor-

pedos sind schwarz, die von Spiller 2 auf der rechten Seite grau. Steuertasten für Spiller 1 sind die Pfeiltasten; Taste W dient zum Abfeuern des Torpedos. Es ist übrigens immer nur ein Torpedo pro Spiller möglich. Beim zweiten Betätigen der Schußtaste wird das alte Torpedo gelöscht und ein neues gestartet. Spiller 2 steuert ähnlich wie Spiller 1 mit den Tasten: I - nach oben, J - nach links, K - nach rechts, M - nach unten, U - Schußtaste. Jeder etwas erfahrene Programmierer wird die Tastatursteuerung leicht auf einen Joystick umschreiben können.

Die Torpedos werden unabhängig vom Spiel auf die Höhe des Gegners gelenkt und fliegen immer auf die gegenüberliegende Seite. Treffen zwei Torpedos aufeinander, sind beide vernichtet. Der Verlust eines U-Bootes erfolgt durch: Nichtauftanken, Zusammenstoß mit einem anderen U-Boot (damit sind auch die eigenen U-Boote

gemeint!), Zusammenstoß mit einem feindlichen Torpedo, Aufprall auf eine grüne Befestigungsmauer.

Das Spiel ist beendet, wenn ein Spieler kein U-Boot mehr besitzt.

Um den optischen Eindruck des Spiels zu verbessern, ist der gesamte Zeichensatz in Fettschrift umgewandelt worden. Vielleicht finden diese Zeilen (460-540) auch noch in anderen Spielen oder Textprogrammen ihren Platz. Zur Erhöhung der Laufgeschwindigkeit des Programms sind die meisten Programmzeilen bis zu 5 Bildschirmzeilen lang. Noch längere Zeilen hätten jedoch die Eingabe zu kompliziert gemacht. So könnten z.B. die Zeilen 120 und 130 in einer Programmzeile von ca. 8 Bildschirmzeilen zusammengefaßt werden, ohne daß sich der Computer mit LINE TOO LONG melden würde.

Frank Motzka

```

100 !U-BOOT SCHLACHT
110 ON BREAK NEXT :: GOTD 460 :: DIM G(2),H(2,5),P(2),R(2),S(2),W(2):: A,B,C,K,D
,X
120 CALL CHAR :: CALL CLEAR :: CALL COINC :: CALL COLOR :: CALL DELSPRITE :: CAL
L GCHAR :: CALL HCHAR :: CALL KEY :: CALL LOCATE
130 CALL VCHAR :: CALL MOTION :: CALL PEEK :: CALL POSITION :: CALL SCREEN :: CA
LL SOUND :: CALL SPRITE :: !0P-
140 FOR A=1 TO 2 :: C=3-A :: CALL KEY(A,K,X):: IF X=0 THEN 180
150 IF K=4 THEN 250 :: R(A)=R(A)-B*(K=0)+(K=5)*(R(A)>57):: S(A)=S(A)+B*(K=2)-
(K=3):: IF S(A)<1 OR S(A)>250 THEN S(A)=125+124*SGN(S(A))
160 CALL GCHAR(R(A)/B+.875,S(A)/B+.875,D):: IF D>103 AND D<121 THEN 340 :: IF W(
A)<0 THEN GOSUB 430
170 CALL LOCATE(A,R(A),S(A))
180 W(A)=W(A)-1 :: IF W(A)=0 THEN CALL HCHAR(20,A*27-24,B4):: CALL SOUND(-200,66
6,3):: CALL COINC(A,153,A*184-151,6,X):: IF X THEN GOSUB 440
190 IF W(A)=-25 THEN 340
200 IF P(A)=0 THEN 220 :: CALL POSITION(A+2,K,X):: CALL MOTION(A+2,2*SGN(R(C)-
K),P(A):: CALL PEEK(-31877,0):: IF 0 AND 32 THEN 270
210 CALL GCHAR(K/B+.875,X/B+.875,D):: IF D>32 THEN 240
220 CALL PEEK(-31877,0):: IF 0 AND 32 THEN 310
230 NEXT A :: GOTD 140
240 CALL DELSPRITE(A+2):: P(A)=0 :: CALL HCHAR(K/B+.875,X/B+.875,32):: GOTD 220
250 CALL SOUND(-1000,111,6,-3,6,333,6):: CALL GCHAR(R(A)/B+1.875,S(A)/B+.875,X)
:: IF X>32 THEN 200
260 P(A)=4*SGN(1.5-A):: CALL SPRITE(A+2,117+A,A*13-11,R(A)+B,S(A):: GOTD 200
270 CALL COINC(A+2,W,6,X):: IF X=0 THEN 300 :: CALL DELSPRITE(A+2,W):: P(A)=
0 :: R(C)=177 :: G(C)=G(C)+1 :: IF G(C)=6 THEN 370
280 CALL HCHAR(23,H(C,G(C)),128):: S(C)=H(C,G(C))*B-7 :: CALL SPRITE(W,123-C*3,
C*13-11,R(C),S(C):: CALL HCHAR(20,C*27-24,104):: GOTD 220
290 ALL HCHAR(23,H(C,G(C)),128):: S(C)=H(C,G(C))*B-7 :: CALL SPRITE(W,123-C*3,C
*13-11,R(C),S(C):: CALL HCHAR(20,C*27-24,104):: GOTD 220
300 CALL COINC(A+2,W+2,6,X):: IF X THEN CALL DELSPRITE(A+2,W+2):: P(1),P(2)=
0 :: GOTD 220
310 CALL COINC(A,W,6,X):: IF X=0 THEN 220 :: CALL DELSPRITE(A,W):: R(1),R(2)
=177
320 FOR X=1 TO 2 :: G(X)=G(X)+1 :: IF G(X)=6 THEN 360 :: CALL HCHAR(23,H(X,B(X))

```



```

/B+.875,128):: S(X)=H(X,G(X))*8-7
330 CALL SPRITE(WX,123-X*3,X*13-11,177,S(X)):: NEXT X :: GOTO 220
340 CALL DELSPRITE(WA):: CALL SOUND(-700,-5,4):: R(A)=177 :: W(A)=INT(80*RND)+10
  :: CALL HCHAR(20,AK27-24,104):: G(A)=G(A)+1 :: IF G(A)=6 THEN 370
350 CALL HCHAR(23,H(A,G(A)),128):: S(A)=H(A,G(A))*8-7 :: CALL SPRITE(WA,123-3*A,
AK13-11,R(A),S(A)):: GOTO 220
360 IF X=1 THEN G(2)=G(2)+1
370 FOR A=1 TO 600 :: NEXT A :: CALL DELSPRITE(ALL):: CALL SCREEN(5):: DISPLAY A
T(5,1)ERASE ALL:RPT$( "=",36)&" GAME OVER "&RPT$( "=",36)
380 IF G(1)=G(2) THEN DISPLAY AT(12,1):"SIE SIND BEIDE GENAUSO MIES": "ALS UGB00
T KOMMANDANT": "ES GIBT KEINEN SIEGER" :: GOTO 400
390 DISPLAY AT(12,1):USING "SIEGER : SPIELER #":-(G(1)<6)-2*(G(2)<6)
400 DISPLAY AT(23,1):"NOCH EIN SPIEL ? <J> <N>"
410 CALL KEY(C,K,A):: IF K=74 THEN RUN 550 ELSE IF K>78 THEN 410
420 STOP
430 CALL COINC(WA,153,AK168-127,6,X):: IF X=0 THEN RETURN ELSE CALL GCHAR(20,AK2
5-21,X):: IF X<>45-A THEN RETURN
440 CALL GCHAR(20,AK25-21,X):: IF X<>45-A THEN RETURN
450 CALL SOUND(-500,131,6,555,6,192,6):: W(A)=INT(90*RND)+1 :: CALL HCHAR(20,AK2
7-24,104):: RETURN
460 CALL CHAR(48,"3C7E6666667E3C001B383B1B1B1B3C0003C7E660E1C3E7E003C7E6660C667E3C
")
470 CALL CHAR(52,"606C6C7E7E0C0C007E7E707C0E7E3C003E7E707C66663C007E7E0E1C3B707"
)
480 CALL CHAR(56,"3C66663C66663C003C66663E0E3E3C",61,"00007E7E007E7E",63,"003C7E
660E1B001B")
490 CALL CHAR(65,"3C7E66667E6666007C66667C66667C003C7E66660667E3C",68,"7B7C666666
7C7B007E7E607B607E7E007E7E507B606060003C7E6066667E3C")
500 CALL CHAR(72,"66667E7E66666003C3C1B1B1B3C3C000E060606667E3C006666C7B707B6C66
")
510 CALL CHAR(76,"60606060607E7E0042667E666666600667E7E7E66666003C7E6666667E3C
")
520 CALL CHAR(80,"7C66667C660606000000007E7E0000007C66667C7B6C66003E7E703C0E7E7C
")
530 CALL CHAR(84,"7E7E1B1B1B1B1B0066666666667E3C0066666666663C1B00636363B6B7F3E
")
540 CALL CHAR(88,"63773E1C3E77630066667E3C1B1B1B007E7E0E1C3B7E7E")
550 CALL CHAR(104,RPT$( "F",24)&RPT$( "OF",16),108,"FFFFFFFF&RPT$( "FO",16)&RPT$( "F
",16),112,"00000000FFFFFFFF")
560 CALL CHAR(113,"F0F0F0F0&RPT$( "O",17)&"F0F0F0F0F0F0F0F0FFFFFFFF00000000F0F0F
0F",128,"3C4299A1A199423C")
570 RANDOMIZE :: CALL CLEAR :: CALL COLOR(1,5,2):: CALL SCREEN(2)
580 CALL CHAR(92,"0F1621293D5E40433C110A121108070000C0202020A0A040B031499A2C40B"
)
590 DISPLAY AT(3,1):CHR$(12B):" BY FRANK MOTZKAU \~:" WORM PRODUCTIONS J_"
600 DISPLAY AT(9,1):"h h jlrthhrthrhjhnm":"h h prjnsqjwjmjmjm jm":"hph oqjmwj
jnsnjnsj ja":"koq koo oo oo kq"
610 DISPLAY AT(15,1):"shthhrh hjm shnthhrh hjhhainrjnkqhhhj h h jkqkqhh jm "
620 DISPLAY AT(17,1):"shhjnseh hjnrphohjnseh h ja ooq oo o okooqo o oo o o kq":
: : "SPIELSTART = ENTER"
630 FOR A=2 TO 8 :: CALL COLOR(A,16,1):: NEXT A :: CALL COLOR(10,12,1,11,12,1,13
,16,1)
640 CALL KEY(O,K,A):: IF K<>13 THEN 640 :: CALL CLEAR :: CALL COLOR(1,5,5,2,2,5,
9,8,10,13,5,12,16,2,13,12,5)
650 CALL CHAR(128,RPT$( "1B",8),117,"001B0B0B187EFF7E0000BEC3BE0000000007DC37D00
0000001B101B187EFF7E",43,"01010707071FFF3FB0C0E0E0F0FFB")
660 R(1),R(2)=177 :: CALL VCHAR(8,17,104,16):: CALL HCHAR(1,1,96,224):: CALL HCH
AR(23,31,104,34)
670 CALL HCHAR(23,31,104,34):: CALL VCHAR(8,16,104,16)
680 FOR A=1 TO 50 :: CALL HCHAR(RND*13+10,INT(RND*30)+2,42):: NEXT A :: CALL HCH
AR(19,24,104,12)
690 CALL HCHAR(20,4,44):: CALL HCHAR(20,29,43):: CALL HCHAR(20,30,104,12):: CALL
HCHAR(21,24,104,18):: CALL HCHAR(22,24,104,11)
700 CALL HCHAR(23,8,120,4):: CALL HCHAR(23,27,117,4):: CALL HCHAR(20,7,128,3)::
CALL HCHAR(20,24,128,3):: CALL HCHAR(23,7,128,3)
710 S(1)=49 :: CALL HCHAR(19,1,104,9):: CALL HCHAR(23,24,128,3):: P(1),P(2)=0 ::
G(1),G(2)=1
720 FOR A=1 TO 5 :: H(1,A)=B-A :: H(2,A)=25+A :: NEXT A
730 S(2)=201 :: W(1)=INT(40*RND)+10 :: W(2)=INT(40*RND)+10 :: CALL SPRITE(#1,120
,2,177,49,#2,117,15,177,201):: GOTO 140

```

Combas

Dieses Programm läuft auf dem TI 99/4A mit Extended Basic und Diskettenlaufwerk. Es packt auf sinnvolle Weise möglichst viele Statements eines unter DSK1.WORK,

MERGE abgespeicherten Programms in eine Zeile und speichert das Ergebnis unter DSK1.CBAS, MERGE ab. Man kann beim Programmwurf ruhig ein Statement pro Zeile schreiben (der Übersicht wegen), weil "Combas" die Geschwindigkeitsoptimierung

übernimmt. Danach laufen TI-Basic-Programme sehr viel schneller.

Zu beachten ist allerdings, daß "Combas" Unterschiede bei der Zeichendefinition von TI-Basic zu Extended Basic nicht berücksichtigt! Falls das zu bearbeitende Programm die

Zeilennummer 200 nicht enthält, so ist eine REM-Zeile mit der Nummer 200 einzufügen, weil der Interpreter wahrscheinlich nach einer Kombination von RESTORE *XX : OPEN *XX in's Schleudern kommt.

Diethelm Schädlér

```

100 !          COMBAS
120 !  DIETHELM SCHÄDLER
130 !  KRANICHSTRASSE 11
140 !  6700 LUDWIGSHAFEN
160 !  BENÖTIGTE GERÄTE:
180 !  DISK-DRIVE
190 !  EXBASIC
210 !  ZU OPTIMIERENDES PG ->
220 !  WORK (MERGEFORMAT)
240 !  OPTIMIERTES PG ->
250 !  CBAS (MERGEFORMAT)
260 !
270 DATA 131,132,134,136,139,147,150,152,154,161,163,168
280 A$=CHR$(130):: B$=CHR$(0):: C$=CHR$(147):: A=255 :: B=163 :: C=3 :: D=0 :: E
=164 :: F=12 :: G=1 :: H=162 :: DIM ZNR$(5),CH$(12):: FOR I=G TO F :: READ CH ::
CH$(I)=CHR$(CH):: NEXT I
290 DEF FPP(X)=INT(X/2)*2-X :: DEF FZNR$(X$)=SEG$(X$,G,2):: DEF FVGL(I)=POS(X$,C
H$(I),C):: DEF FZV(X$)=POS(X$,CHR$(201),C):: E$=CHR$(A)&CHR$(A):: OPEN #B:"DSK1.
WORK",VARIABLE 163 :: ZNR$(0)=E$
300 IF EOF(B)THEN 340
310 LINPUT #B:X$ :: IF POS(X$,C$,C)THEN CALL ZIEL(ZNR$(3),FZNR$(X$),ANZ)
320 IF FL=D THEN CALL ZIEL(ZNR$(3),FZNR$(X$),ANZ):: FL=G
330 ZV=FZV(X$):: IF ZV THEN X$=SEG$(X$,ZV+G,A):: CALL ZIEL(ZNR$(3),FZNR$(X$),ANZ)
:: GOTD 330 ELSE 300
340 RESTORE #B :: OPEN #E:"DSK1.CBAS",VARIABLE 163
350 IF EOF(B)THEN CLOSE #B :: CLOSE #E :: END
360 LINPUT #B:X$ :: FOR I=D TO ANZ :: ZN$=FZNR$(X$):: PP=POS(ZNR$(I),ZN$,G):: IF
PP>D AND FPP(PP)<D OR FL=D THEN 400
370 NEXT I
380 L=LEN(X$):: LX=LX+L-2 :: IF LX>H THEN 400
390 PRINT #E:A$&SEG$(X$,C,L-C):: GOTD 420
400 IF X$=E$ THEN PRINT #E:B$ :: PRINT #E:X$ :: GOTD 350
410 PRINT #E:B$ :: L=LEN(X$)-G :: PRINT #E:SEG$(X$,G,LX):: FL=G
420 FOR I=G TO F :: IF FVGL(I)>D THEN FL=D :: GOTD 350
430 NEXT I :: GOTD 350
440 SUB ZIEL(ZNR$(3),ZN$,ZNR)
450 FOR I=0 TO ZNR :: PP=POS(ZNR$(I),ZN$,1):: IF PP<0 AND INT(PP/2)*2-PP=-1 THE
N SUBEXIT
460 NEXT I
470 IF LEN(ZNR$(ZNR))<254 THEN ZNR$(ZNR)=ZNR$(ZNR)&ZN$ ELSE ZNR=ZNR+1 :: ZNR$(Z
N$)=ZNR$(ZNR)&ZN$
480 SUBEND

```

Klarstellung!

Aus gegebenem Anlaß möchten wir hier eventuellen Mißverständnissen oder Verwechslungen vorbeugen. In der letzten Zeit sind in einigen Teilen Deutschlands die TI-Workshops wie Pilze aus dem Boden geschossen. Wir wurden vielfach angerufen und gefragt, ob wir nun Filialen eröffnet hätten. Dem ist aber nicht so. Es

gibt weiterhin nur einen TI 99er Workshop Rheinland!

Wir möchten hier feststellen, daß wir mit keinem dieser Workshops mehr oder weniger zu tun haben als mit jedem anderen Club. Sollte sich der eine oder andere Workshop mit dem Hinweis auf unsere Gruppe irgendwelche Vorteile verschaffen wollen, bitten wir um eine kurze Nachricht an:

TI 99er Workshop Rheinland
Dept. Allgemein & Software

c/o Mike Heuser
Karl-Mars-Allee 18
5000 Köln 71

Tel. 02 21/70 39 79
(19.00 - 22.00 Uhr)

Ihr könnt aber auch eine Nachricht in unserer Mailbox, die selbstverständlich auf einem TI läuft, unter der Nummer 02 28/65 82 32 hinterlassen.

Es sei hier nochmals betont, wie auch schon mit den Work-

shops persönlich besprochen, daß wir nichts gegen die Mitglieder oder auch die Clubs als solche haben. Wir versuchen nur, unseren guten Namen gegen evtl. Mißbrauch zu schützen.

Wir werden auch weiterhin mit allen Clubs, die dies wünschen, hoffentlich fruchtbar zusammenarbeiten. Also schreibt uns, wenn ihr als Club oder auch als Einzelperson an einer Zusammenarbeit interessiert seid.



TI Special



Tuning: 128K RAM plus Centronics- Interface

Stabiles Metallgehäuse, Versorgung über Steckernetzteil, durchgeführter Expansionsbus 4K ROM gesockelt, 32K RAM direkt eingebündelt, ermöglicht TI-Writer, Ext. Basic etc. RAMdisk, Centronics Druckeranschluß, leicht verständliches Handbuch.

Best.-Nr. TI 1 400,- DM



Komfort: TI-Maus

Maus mit Anschlußkabel, 5-V-Steckernetzteil, Software auf 5.25" Disk.

Best.-Nr. TI 2 296,- DM



Lehrreich: TI Assembler Kurs II

Eine Einführung in die Assemblerprogrammierung mit dem TI 99/4A.
300 Seiten inkl. Diskette.

Best.-Nr. TI 6 80,- DM



Lang erwartet: TI-Assembler-Kurs III

Die logische Fortsetzung zum TI-Assembler-Kurs II. Bietet noch mehr Einblick in das Innenleben Ihres TI.

Best.-Nr. TI 4 79.90 DM



Zeitsparend: Turbo-Assembler

Doppelt so schnell wie Editor/Assembler, weniger Platzbedarf auf Disk, schnelleres Laden von Disk.

Anforderungen:
Konsole + Ext. Basic, Cassettrecorder, 32K RAM Erweiterung

Best.-Nr. TI 5 99,- DM

Nützlich: TMS 9900 Assembler

DAS Nachschlagewerk für TI-Assembler-Programmierer. Der Prozessor, seine Befehle, Beispieldiagramme, Speicheraufbau, Tabellen, Index...

Anforderungen:
Editor/Assembler, Extended Basic

Best.-Nr. TI 3 49,- DM



CK SPEZIALITÄTEN-BESTELLSCHEIN



Preis	Best.-Nr.	Artikel	Preis inkl. MwSt.
	TI 1	128K Ram plus Cent. Interf.	
	TI 2	TI Maus	
	TI 3	TMS 9900 Assembler	
	TI 4	TI-Assembler Kurs III	
	TI 5	Turbo Assembler	
	TI 6	TI Assembler Kurs II	

Name des Bestellers

Anschuß

PLZ/City

Ordnung/unterschiedl.

Ich wünsche folgende Besorgung

Nachnahme (inkl. 5,70 DM Versandkosten)

Vorauskassa (keine Versandkosten)

Bei Vorauskassa bitte Scheck belegen oder auf Finanzbuchkonto Kontonr. 43422-756 überweisen

Coupon ausschneiden, auf Postkarte kleben und einreichen an:
Verlag Rötz-Eberle/CK-Software, Postfach 1640, 7518 Breden

ZX-SPECTRUM 48K zu verkaufen da: **Troick-Testatur** (Zwei-Stück) + über 150 Programme - von **Tasword II** bis **Deception** + eingebauter Joystickinterface + **9-SW-Fernseher** + **Seiko-GP-100 A** Mitteldrucker + **Druckerinterface ZXPrinter II** + Bücher + Zubehör für zusammen nur **950,- DM**. **Leonhard Fuchs**, Staufenackstraße 16, 7340 Gaisingen/1, ☎ 0 70 31 / 6 78 51

Verkauf von ZX81 / 16 K-Software, **Infor**, **N. Lange**, **Heckhauz 23, 7777 Seem-Bier**

●●● Halo Spectrum-User ●●● Zu verkaufen: 1 Microdrive (30 - DM) + Interface 1 (80 - DM) + 17 Cartridges je 5,- DM. Alles zusammen 250,- DM. Suche die Kontakt zu Opus-Usern und u.U. Programmkaufclubs. Jörgsch, Postfach 1006 10, 40 40 Neuss 1

Suche Programme für Spectrum + für **Wetterstation ELU** und **ELV WS 7000**. **Dieter Kruse**, Schlierstraße 16, 4650 Brämsche 1

Verkaufe ZX-Spectrum 48K, ZX-Printer (defekt), **Seiko-GP 500A**, diverse Literatur und Software zu günstigen Preisen. ☎ 02 06 1 / 34 28 nach 19 Uhr

●●● **Tausche Spectrum-Spiele** ●●● (Gass.) gegen Atari (Disk/Gass.) oder Spectrum-Programme (Gass.). Liete an **Andreas Bergmatsch**, Abnstraße 5, 5780 Arnsberg 1

●●● **Hallo Spectrum-Freaks** ●●● Suche Software aller Art. Suche auch nach guter Drucker! Schreibt an B. Jungshöfel, Herwegweg 10, 2993 Esterwegen. ☎ 0 59 55 / 5 89

ZX-Spectrum + Profitestatur + **Kempat**, **Joy-Off + viel Literatur + Software**. **VHB 490,- DM**. ☎ 0 88 46 / 6 58

● **Sinclair QL** ● Tausche oder verkaufe Originalprog.: **Machine Code Editor/Assembler** (McDraw H87 70 - DM, Flight Simulator (Microcode) 40,- DM, Gedächtn: Pison Chess, QL-Art s.a. U. Schmidt, Schloppstüchel 36, 5490 Lnz

Sinclair-Spectrum + Opus 180 K + Testatur + Disketten + Software + Monitor, komplett gegen Gebot. ☎ 0 72 46 / 88 75, 18 Uhr

Verkaufe Spectrum 48 K + Microdrive + Interface I + HiSoft Pascal und Assembler + div. Spiele und Bücher, alles 1a-Zustand und orig. verpackt. für 500,- DM. ☎ 0 99 01 / 15 84, nur Sa. oder So.

● **Verkaufe Beta-Disk-Controller** ● Ausführung 4, 12, 89/280,- DM. **Stegfried Hauger**, Hauptstr. 68, 7734 Brighatal

Verkaufe: Interface 1, 2 M'Drives, 22 Cartr., ZX1 Print 3, Monitorinterface. ☎ 0 61 62 / 33 84

Spectrum 48K + gute Testatur + Interface 1 + 2 Laufwerke, alles inkl. Netzeil im Testaturlager. Ferner Drucker GP 50 S, 200 Cassetten und 12 Cartridges mit vielen guten Programmen, viel Literatur. Komplett für 600,- DM VHB. ☎ 02 91 / 3 34 05

●●● **Spectrum 80 K, Kompaktbox** ●●● inkl.: Cass-Laufwerk, Opus-Controller, Floppy 6,25 Zoll 1 MB, Saga 1-Tast., Sound - Joyst., CK Interface, Resel, Centronics, Disketten, Software-Literatur + Handbücher kompl. 600,- DM. ☎ 02 08 / 83 52 78, bis 18 Uhr

●●●● Topf ●●●●

Biele Software, Bücher, 200 Magazins s.u.m. zu **Spottpreisen!** **Thomas Schröber**, Hohlstraße 11, 5791 Sielbach. Info 1,- DM. **Centr. Drucker-Kabel (Opus) 26,- DM**

Kempston Centron, E + L-Print II, je nur 88,- DM! ☎ 0 69 93 / 74 90, ab 18.30 Uhr!

Verkaufe Spectrum 48K, Fuller FDS-KB, Beta-Disk-Contr. 4.7 DOS, 2 Drucker-Int. (1 Kempst. 1 ZX-L-Print), Lichtgriffel, Tastaturleben, Ersatz-Ura-Spulte, Tasword-Masterfile-Tasprint-Tasmerge, RTTY-Soft- und Hardware-Oscar 10, auch auf Diskette, Presse VHS, auch Einzelverkauf. ☎ 0 81 95 / 5 47

●●● **Bücher für ZX-Spectrum** ●●● 50% unter Neupreis! **ZX-Spect. Tips + Tricks** von **Data Becker**, weitere Kritisik und Programme mit dem **ZX-Spect.**, **Machinecode** von **Birkhäuser**, **Basic-Handbuch** und **Pascal** auf dem **ZX-Spectrum**. ☎ 082 33 / 47 10, ab 18 Uhr

Atari

●●●● **ATARI 600 XL / 800 XL** ●●●● 256 KB Speicherverweigerung, XE kompatibel, inklusive Software: a) RAMDISK (alle Atari-Formate) b) Kopierprogramm (1 Diskwechsel) c) Adressverwaltung (2000 Adressen). Preis: 279,- DM. Info gegen Rückporto bei: **Völler Rasche**, Gellstraße 22, 4970 Bad Oeynhausen 1. ☎ 0 57 31 / 2 25 20

ATARI 260 ST: 1 Megabyte Aufrüstung, steckbare Platine ohne Lötarbeiten,EPROMsteckkarte und User-Port-Karte. Für 400/800: OS Rev. B und es laufen alle Programme. **Happy 1080** und Highchip. ☎ 02 02 / 40 26 14

Verkaufe Originaldiskette Atlas II, Makroassembler mit Anleitung 26,- DM, Originalkassette Forbidden Forest für nur 10,- DM. ☎ 0 22 38 / 5 72 91

Suche Atari-Software, nur D, Z, B, G,unshio, Q,ulwi und gute Anwendungssoftware, Tausch möglich, z. B. Silent Service, Print Shop, Wizard usw. Liete an **R. Lobert**, 4700 Hamm 3, W.-v.-Braun-Strasse 8

Seiko-GP-100 AT für Atari 600/800 XL, Disketten-schüß. 8 Monate alt, wenig benutzt = 150,- DM. ☎ 0 30 / 8 53 15 75

●●● **Achtung Software-Besitzer** ●●● Suche - dringend - Simulationen, drucke CNC-Steuerung (auf 800 XL), Zahle gut. Angebote bitte an: J. Proiwach, Gudenstraße 10, 8765 Eferbach, ☎ 0 93 72 / 64 78, ab 18 Uhr

Verkaufe Abfrage- und Übersetzungsprogramm für Atari 800 XL. Es sind über 4000 englische und deutsche Wörter gespeichert. **Cassette für 20,- DM** bei J. Sznel, Herderstr. 4, 6093 Burghausen

● **Atari KX**: Software auf Cassette ● ab 80 Pf., keine Raubkopien. Liete gegen 80 Pf. Rückporto bei **Michael Wagner**, Altburgberg Str. 64, 6300 Alsfeld

ATARI XL/XE: Tausche Software Hebe z. B. Int. Karte, Last V8. ☎ 0 92 72 / 59 43 ab 18 Uhr

Verkaufe: Rileyman F mit Interface für Atari (Centronics), 100% Epson-Komp., 9x9 Matrix, NP 1100 - DM. 1/2 Jahr alt, VHB 600,- DM. ☎ 0 26 25 / 46 12

Wir suchen einen Redaktionsvolontär!

Die CK-Computer Kontakt erscheint jetzt schon im 3. Jahrgang und hat bei den Lesern der Geräte Sinclair, Atari und TI 99/4-A einen festen Platz erreicht und befindet sich weiter im Aufwind. Nicht zuletzt verdanken wir diesen Erfolg unseren hervorragenden Mitarbeitern. Für den weiteren Ausbau der CK-Computer Kontakt, des Schneider-Magazins sowie unseres neuen Atari-Magazins suchen wir jetzt einen Redaktionsvolontär, der mit uns an den bestehenden Objekten arbeiten und auch neue Ideen angehen will.

Wir stellen uns einen Computerfreak mit Begabung zum Schreiben vor. Er sollte gute bis sehr gute Kenntnisse in Basic, Assembler und MS-DOS vorweisen können oder sich zumindest zutrauen, fehlende Bereiche schnell zu erlernen. Besonders wichtig ist uns aber, daß er weiß, was er will und wir uns auf ihn bei allen Arbeiten verlassen können. Seine Computererfahrungen sollte er bevorzugt auf Atari 800XL/130XE oder Atari ST gesammelt haben.

Das ideale Alter unseres neuen Redaktionsvolontärs wäre ca. 25 Jahre, er kann aber auch älter oder jünger sein. Was zählt, ist die Qualifikation. Beginnen könnte unser neuer Mann sofort bei uns, in unseren Büroräumen ist genug Platz. Die Stelle wird nach Tarif vergütet.

Wenn Sie diese Aufgabenstellung reizt und sich vorstellen können, ihr Hobby zum Beruf zu machen, so wenden Sie sich schriftlich an uns.

Verlag Rätz-Eberle, Postfach 1640, 7518 Bretten

Flensburger Computer Versand

M.-L. Sander - Am Soot 4 - 2390 Flensburg - 0461 / 32533

5 1/4"-Disketten	2D 10er	10.-	CDI Hitrans mit NT-Mögl.-DFU-Interface	222.-
3 1/2"-Disketten	1D 10er	39.-	f. Spectrum	100.-
3 1/2"-Disketten	2D 10er	45.-	Spectrum-Grafiktablett (das Größtelle)	400.-
3"-Maxell CF2	10er	95.-	Mirage Microdriver	100.-
			Philips MSX-2 VG8235	1450.-

Ab 300,- DM bequeme Teilzahlungsmöglichkeit.

●●●● **ATARI** ●●●● Ich liete für 10,- DM ein Malprogramm an. Die genauen Bilder, die man natürlich abspeichern kann, lassen sich für selbstprogrammierte Spiele verwenden. Wie? Das steht in der Beschreibung, die mitgeschickt wird. Schicke 10,- DM und Deine Adresse an: Walter Derschlag, Neubachstr. 42, 8520 Worms 26

● **Atari 130 XE** ● Suche Module, Software, möglichst auf Diskette, vor allem gute Spiele. Suche auch Farbmonitor. Informationen liete an T. Karacayir, Ostensalze 47, 4700 Hamm 1

Suche Kontakt zu Atari-Usern! Habe selbst einen 130XE + Floppy. Meldet Euch bei Jörg Soltmann, Königstr. 32, 6750 Kaiserslautern

Verkaufe Computeranleitungen, drucke Atari-Listings. Listen gegen 2,50 DM in Briefmarken von H. W. Herrmann, Mohrweg 21, 5628 Helligenhausen

Atari 800 XL. Suche Programme aller Art (Disk.) R. Möler, Kellernweg 16, 69-66 Gornheimtal

Atari XL/XE-Software! Verkaufte wegen Systemwechsel meine kompl. Programme zu 30-70% des NP. Beispielt: Print Shop, Typewriter, Ultima 3, Arcade Mach. Nur Originale, keine Raubkopien! Liete anfordern bei: M. J. Meyer, Kreuzmattweg 15, CH-4144 Arlesheim/Schweiz, Atari!

Verkaufe Atari 800 XL + 1018, Progr.-db. Bchner, Stck, VHB 600,- DM. ☎ 0 21 29 / 5 01 05

extrem stabile JOYSTICK'S (Acht-Wege-Fahrhebel)

- jahrelang bewährt im härtesten SPIELHALLEN-EINSATZ
- passend für:
C 64, C 128 / Atari 400, 800, 130 /
Schneider CPC / SVI 318, 328 u.v.a.
auch in Ausführung für TI-99/4A
- von alterfahrenem MÜNZ-AUTOMATEN-HERSTELLER handwerklich gefertigt.
- alle Schaltfunktionen selbstverständlich über Mikroschalter
- 1 Jahr Garantie - DM 98.-

Lieferung kurzfristig per NN plus DM 6,50 Porto und Versandkosten, oder gegen Vorkasse DM 98.- (porto- und verpackungsfrei) Postgiro-Konto Köln 3636 23 - 500, oder durch Verrechnungsscheck.

AUTOMATEN - SERVICE ALAIN KRAWIETZ
BAUERBANKSTR. 27 TEL. 02 21 / 36 63 49 5000 KÖLN 51



Hahe Lösungen zur The Dark Crystal, Atlantis, Cromwell-Hölle, The Mask Of The Sun (de+Engl.), Mül Grad Nord, Schloß des Grauens, Abenteuer im Weltraum, The Serpent's Star, The Dallas Quest, Nebelungen, Je 10.—DM-Scheine belegen! Zuschriften an: D. Kick, Weberweg 2, 8590 Marktredwitz

Verkaufe Atari-800XL-Software. Liste bei Raff Pietzka, Pfalzstraße 47 a, 4250 Bttrpp. 80 Pf. Rückporto!

800XL ● ● 924 Programme ● ● wegen Systemwechsel zu Schmeißerpreis zu verkaufen. Nur Disk. Größtliste anfordern! Karl-Heinz Harlmann, Casa Graziella, CH-6518 Gorduno, Schweiz

● ● ● Staubkiller ● ● ●

Aberkhauben für Atari 800/800 XL/1050, C64/1541, Kivari 14.95 DM, Kunstflügel 19.95 DM, VK oder NH, Bei M. Köhn, Ulmerstr. 16, 4100 Duisburg 17, ☎ 021 36/3 1977, nach 17 Uhr. Bei Vorkasse keine Gebühr!

Super Lotto-Programm geschrieben auf Atari 520 ST+. Mit VEW-Systemen, eigene Zielungen, Gewinn-Tabellen, kurz gesagt, ein Programm mit vielen Möglichkeiten, 50 - DM oder 40.-€fr. einlesen an D. Steiger, Wittingerstraße 164, CH-4058 Basel

● ● ● ATARI ST ● ● ●

Software für Eretschkane (Pomox), Bio-rhythmus, Reversi, Solitär u.s.m. Info anfordern bei: Soft, Postfach 1619, 4630 Gütersloh

Verkaufe Atari-4-Farb-Plotter 1020, Direktanschub, VHB 220.- DM. ☎ 040/4802859

● ATARI ● ● ● RABE I ● ● ● ATARI ● ●
Ein Programmier-, Testunterstützungs-, Fehlerbuch- und Programmanalyse-System ermöglicht Einblicke in geschützte Software. Per Tastendruck jederzeit aufrufbar. Preis 79.- DM, EPHOM + Disk. Info bei A. Kern, Pfalzstraße 5, 8591 Bachmehring, ☎ 060 71 / 12 25

● Lohn- und Einkommensteuer ●
Super Jahresausgleich mit Analyse, jährl. Aktualisierung ohne Neukauf, Cassette 60.- DM, Diskette 73.- DM für Atari. Info gegen Rückporto von H. Iichen, Niederfelder Straße 44, 8072 Mönching, ☎ 06459 / 16 69

Atari 400/800/XL/XE

Verkaufe Spiele + Anwendungen. Z. B. von ECA, Synapse, Lucasfilm auf Disk, Cass. und Modul. Preise von 5.- bis 30.- DM. Liste anfordern, ☎ 023 04 / 4 15 10. Verkaufte außerdem Atari 1020 Drucker mit Handcopy-Programm (Top Zustand), VHB 380.- DM

Suche zuverlässigen Tauschpartner für Atari XL (nur Disk). ☎ 07262/89052

Verkaufe Atari 800XL + Recorder 1010 + Floppy 1050 + 4 Module + 4 Disketten + 8 Cassetten für nur 700.- DM. ☎ 02 21 / 48 17 22

Verkaufe Atari 600 XL + Datas (+ Spiele, Progr., sehr billig. Suche C64 + 1541 günstig. ☎ 029 41 / 808 31. Übernahme art. Porto!

● ● ● ATARI 800 zu verkaufen ● ● ●
Mit Basic, engl. - dtsch. Handbüchern und weiteren Büchern über den Atari, VHB 230.- DM. ☎ 040/829602

Verkaufe Atari-Software auf C + D. Info gegen 80 Pf. Rückporto von Markus Krays, Seebadlerstr. 6, 5024 Pulheim

Suche Atari-Floppy 1050, bis 200.- DM. Außerdem Software und def. Hardware gesucht. ☎ 04621 / 526 67 nach 19 Uhr

ATARI 1050: Suche Master DOS 2.0 und 2.5 Disk für 15.- DM pro DOS + Porto. U. Kostom, Pappelweg 22, 5205 Sankt Augustin 1, ☎ 022 41 / 35 63 89

● ● ● Atari 800 XL ● ● ●

Verkaufe gute MS-Spiele für 5.- DM/ Cass. (auch Tausch). Goonies, Decathlon, Hero usw. Gunter Starke, Beethovenstraße 1, 8943 Biberhausen. Antwort garantiert!

ATARI 800XL + 64 K (1/2 J. alt) mit 3 Modulen (Missile Command, Tennis, Pole Position) und Recorder XC 11 zu verkaufen. Preis 250.- DM, ☎ 046 69 / 12 80, Raneo Nitzel, Kirt 2, 2265 Ladelund

● Dringend ● Suche für Atari: Kyan Pascal, Comal, C. Forth, u.v.m. ☎ 02 73 / 7 28 23, Michael Fiedler, Hanserplatz 5, 5758 Fröndenberg (nach 16 Uhr)

Atari: Verk. Orig.-Spiele auf Disk. Paint, Cavelord, Brittain, Je 20.- DM, Conner 26.- DM, Arcade Machine 35.- DM, Christian Bergmeister, Kriegerstraße 26, 8600 Augsburg 1, ☎ 08 21 / 40 46 90

Hallo XL/XE-Besitzer! Ich verkaufe Teile meiner Programmsammlung auf Cassette und Diskette. Info: Ludger Kerkeberg, Reischer 40, 4408 Dülmen, ☎ 025 94 / 24 00 ● Billig!

● Atari XL/XE ● Suche: Sommergame 8 + Wintergame. ☎ 093 72 / 59 43, Bernd verlangert!

● Für Atari 800: Hyperblast (3) 10.- DM ● Atari-Größtliste: 15.- DM ● Spielbuch: 15.- DM ● Alles VHS. Info: S. Rauch, Mörscher Straße 17, 7552 Durnheim

● ● ● Atari XL/XE ● ● ●

Public Domain Software (ungebittelt) 4 Disketten voll. Preis: nur 50.- DM, mit Anleitungen in Deutsch! Bitte DOS-Version angeben. Bestellung an: Lutz Marschin, Postfach 49, 3258 Aerzen. Das Programm wird in einem Ordner geliefert + 1 leere Diskette extra!

● ● Hallo Atari-XL/XE-Freaks! ● ●

Minimale Spielspendenkette ist auf Bestellung mit ca. 40 Spielen! 90-DM-Schein an: J. Schwarzer, Th.-Heuse-Straße 37, 8660 Mönchberg 100 Sp. = 50.- DM!

● ● ● Software für Atari XL ● ● ●

Neueste Software aus allen Bereichen (Anwenderprog., Geschäftsprogramme und jede Menge Spiele). Z. B. Disketten 15 Top-Tren 20.- DM. ☎ 02 41 57 71 16

● ● ● Suche Software ● ● ●

Ich suche Software für meinen Atari 800 XL. Bitte nur auf Cassette. Preis nach Absprache. Frank Bockle, Krokodieweg 50, 7000 Stuttgart 31

800 XL Atari: Suche für meinen Atari Basic-Module sowie Software. Wenn es geht, nur auf Cassette. Schreibt Angebote an Jörg Affeldt, Bahnhofstraße 4, 4100 Duisburg 18

••• ATARI •••

Verkaufe Drucker 1027 und Datensette, beide wenig gebraucht, Schickt Eure Angebote mit Preisvorstellungen an Peter Jüngermeier, Bismarckring 104, 6301 Heuchelheim, ☎ 06 41/55313

Suche DOS-2 und Astrologieprogramm für Atari 800 XL Ludwig Bürger, Karmannstraße 15, 2082 Uetersen ☎ 0 41 22/4 20 36, Anrufe nach 19 Uhr.

• **Top-Akzept für Atari XL/XE**
Musik-Maschine II: • Synthesizer; Direct-Play/2; Manuale/veracht; instr.; neue Superflythru! • Converter: Einbaue des Scripts in Basic-Programme • 30 Demos • I/II-Interface • Als Komp. per teptext oder einzeln erhältlich zum absoluten Sonderpreis • ☎ 0 61 44/17 98 oder 0 61 35/33 84, nur Sa., So!

Texas Instruments

1000 KB Programme für TI-99, komplett für 100,- DM, auch einzeln und viel Lit. M. Buchberger, Guardianstraße 22, 8000 München 70

Assembler-Programme: Disk-Editor, GP, Disassembler, Fast-Cooly für E/A, RAM, XB + 32 K + Disk. Info gegen Fremmschlag. M. Eichhorn, Ziegelsteck 1, 6240 Königstein 4

••• TI-99/4A •••
 Adressverwaltung, Textverarbeitung + div. Finanzrechnung. Je 9,90 DM. komplett 24,90 DM zuzüglich 5,10 DM Porto und Cassette. Info 1,10 DM. W. Schmitz, Bendstraße 35, 4780 Werl

Verkaufe gebraucht für den **TI-99**: Statist.-Modul + 994002 Printer interface + GP 50 A mit 5 Rollen Papier. Ab 16 Uhr ☎ 0 81 88/22 05 bei Vittzhum

Suche neue und gute Freeware für den TI mit 32 K! ☎ 02 01/53 06 67

Verkaufe TI-99/4A + Ex. Basic + deutsches Handbuch + Rec. Kabel + Joystick = 150,- DM, Diskettenlaufwerk M. PHP 1850 = 100,- DM, 32-K-Byte-Speichererweiterung - batteriegepuffert = 150,- DM, Module, Spiele auf Cassette und Listings, Info gratis von Markus Thiemer, Schwedenstr. 2, 6203 Hochheim

TI-99/4A + Zubehör, VHS 3700,- DM. ☎ 021 74/4 06 54

Verkaufe: TI-99/4A + Ex. Basic TI Plus + MBI-Cent, + 32 KRAM-Ext. + Sekunia GP 100 Merk T. u. div. Bücher, z. B. TI-SP. 1 + 2 + Spezialkäfte + TI-Magazine ab Nr. 1, mit Cass.-Software + weitere Cass.-Software + Joystick, komplett. Preis: VHS, Thomas Schmid, Kernstraße 21, 7430 Metzingen

TI-99/4A + P-Box + RS 232 + 32 KRAM + TI-Laufwerk zu verkaufen, inklusive Module: Mini-Memory + Microsoft Multiplan TI Writer + Datenvens. und Analyse + Text-, Datei- und Finanzbuchhalt. + TI-Ext. Basic + Disk Manager II mit Zeitrhyth. Monitor. ☎ 0 78 39/7 45

••• TI-99/4A •••
 + P-Box (32 Kint.) + Disklaufwerk und Contr., + Ex-Basic-E/A-Modul + Speech-Synth. Assembler-Kurs + Bücher und Hefte KCI + TI-Reviser + Disketten (auch Spiele) + Recorder mit Kabel usw. Preis VHS, Andreas Zeitl, Weilerstraße 12, 6140 Bensheim, ☎ 0 62 51/6 33 98, Konsole defekt!

• 32-K-Erweiterung für den TI 99/4A • nur 140,- DM! Batterielsg., durchgeh. 5H. Bus, inkl. Akkuzell Gehäuse dazu: 10,- DM, Assembler/Disassembler: 20,- DM! Manual 5,- DM! K. Mehr, Friesweg 38, 4133 Neuk.-Vuyit, ☎ 0 26 45/2 15 07 oder 54 14

TI 99/4A Konsole und V24-Karte für Exp. Box gegen Geldbot bei: Albertweg, Auesfeld, 15, 7400 Tübingen, ☎ 0 70 71/6 15 58

TI 99/4A + Ext. Bas. + div. Module + Exp. Box + Disk + 32 K + Lit. Preis: 1100,- DM. ☎ 05 11/77 35 99

Commodore

Universale Datenverwaltung in Assembler für C 126/80 Zohn, Disk mit Disk-Ass-120, für 30,- DM (a. NN), Peter Hahlscher, Am Wall 22, 4401 Speerbeck

C 64, Floppy 1541, Speed-DOS-E, komplett 790,- DM, ggf. auch einzeln zu verkaufen, ☎ 05 21/1 68 24/15 23 76

20 C 64-Programme auf Disk für nur 20,- DM inkl. Versand, Vorschick an Uwe Röhner, Karstr. 24, 7552 Durnmenheim. Info nur gegen Fremmschlag!

Verkaufe C 64 + Datensette + Software, VHS 250,- DM. Suche Microdrive-Cartridge! Meidet Sucht nachmittage bei Felix, ☎ 02 41/7 81 95, außer an Wochenenden!

C-64-Spiele- und Anwendungsprogramme (C/D) gesucht. Schrebt an Fritzj Drom, Wilhelm-Böhmert-Straße 4, 2800 Bremen 33

••• Super Neu! •••
 Textverarbeitungsprogramm verwaltet Adressen und Serienbriefe 100% MCI! Luft auf C64, 20, 16 usw. 35,- DM, Börsenprogramm für Aktien und Commodities inkl. RSI-Momentenkurve, versch. Gliederungsschritte, für C 64/35,- DM, Alle Preise + 3,- DM Porto, Bel O. Hauke, Lichtenbergerstraße 90, 4019 Mönheim

VC-20-Software zu K.O.-Preisen. Wo? Bei F. Segovia, Nordwalder Str. 83, 4407 Elmstedten. Info gegen 1,20 DM! Tausche auch! Suche Floppy für VC-20! Bitte bis 200,- DM! Muß funktionieren!

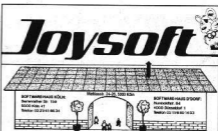
Suche jemanden in Berlin, der komplettes Hausverwaltungsprogramm für 50 Wohnmehnhäuser für mich schreibt. Wirtschaftliche Gegenüberstellung, Mietabhöhung usw., Näheres mündlich, möglichst Commodore 128, keine Stämper, Gage VHS. Gerhard Schibba, Achener Str. 41, 100 Berlin 31

Sonstiges

••• NEU •••
 Music-Power-Club, Postfach 1109, 82716 Rest/Wirtil, bietet: 1 Hinstänge + Clubinfo + Supercassette gegen 10,- DM-Scheitel! Nur solange Vorrat reicht!

Suche immer noch für den Intelsystem B 17 Bomber, Utopia und Treasure of Terminus. Ich kaufe diese Spiele. Ruf doch an ☎ 08 51/3 22 64 Günther Beyer, Dr.-Winkelhof-Str. 90, 6390 Passau

Ich programmiere alle EPROM-Typen für Spectrum, QL, Atari ST und alle anderen Computer. Anfragen unter ☎ 09 61/44 85 20 ab 18 Uhr, D. Appl



Spectrum 48/128 K Preis Spectrum 65/128 K Preis

2112 AD	35,00	Soy Hunter	35,00
30 Games	35,00	Starquake	35,00
A View to a Kill	35,00	Street Hawk	29,00
Ariflow	29,00	Superblast	29,00
Arade-Hall of Fame	32,00	Superschess 3.5	37,00
Arenas	69,90	Swords and Sorcery	39,00
Art Studio	49,00	Tarotus	32,00
Asterts	32,00	Tarzan	29,00
Astro Clone	35,00	Taxi Cab	35,00
Back to the Future	39,00	Thaatos	35,00
Berry Mc Guigans Boxing	39,00	The Hacker	29,00
Barnham	32,00	The Never-ending Story	35,00
Battle of the Planets	35,00	The Way of the Tiger	32,00
Biggles	25,00	Theatre Europe	35,00
Bobby Bearing	29,00	They said a Million I	35,00
Born-Jack	29,00	They said a Million II	35,00
Bruba Law	32,00	Three Weeks in Paradise	39,00
Carnelot Warriors	39,00	Timetrax	39,00
Colossus 4.0	32,00	Toadrunner	32,00
Combat Lynx	32,00	Tomahawk	35,00
Commando	29,90	Turbo Euprit	39,00
Classical Mass	32,00	Turbo	32,00
Dandy	29,90	Wanted Gunlight	39,00
Demolitor	39,00	War	29,00
Elite	65,00	Writer Games	39,00
Equinox	32,00	World Cup Carnival	39,00
Fairlight	35,00	World Series Baseball	32,00
Fighting Warrior	35,00	World Series Basketball	39,00
Football Manager	39,00	Year at Kung Fu	32,00
FP Compiler	39,00	Zoics	32,00
Frank Blinn's Boxing	35,00	Zorm	32,00
Franka goes to Hollywood	35,00	"V"	35,00
Galvan	29,00		
Ghosts in Goobins	23,00		
Gladitor	35,00		
Goonies	69,00		
Graphic Advent. Creator	35,00		
Great Escape	35,00		
HERY000R	35,00		
Heavy on the Magic	39,00		
Herb's Dunny Run	35,00		
Highway Encounter	35,00		
Hyper Sports	32,00		
Impossible Mission	35,00		
Infiltrator	32,00		
International Karate	32,00		
It's a Knock Out	29,90		
I, of the Mask	39,00		
Jack the Hippo	31,00		
Juggernaut	35,00		
Kingslayer	29,00		
Kung Fu Master	49,00		
Laser Basic	29,00		
Laser Complex	49,00		
Legende of the Amazon V.	32,00		
Lightforce	29,90		
Lord of the Rings	55,00		
Macadam Bumper	35,00		
Mermaid Madness	35,00		
Mickis	32,00		
Mindshadow	39,00		
Mission Creeps	32,00		
Monty on the Run	32,00		
Movie	32,00		
N.O.M.A.D.	29,00		
On the Run	32,00		
Paperboy	32,00		
Ping Pong	32,00		
Princess	35,00		
Quazatron	32,00		
Rabul Planet	35,00		
Recue on Fractus	35,00		
Revolution	35,00		
Rock n' Wrestle	29,00		
Sobolux	32,00		
Slit Combat	29,00		
Sesame the Fox Strip Paker	35,00		
Shizpzhenns	29,00		
Spiffie 40	39,00		

Spectrum QL Preis

Arcadia	39,00
Better Basic Expert Syst.	39,00
Blocklands	59,00
Cambridge Media Manager	65,00
Cosmos	49,90
Droidzone	59,00
Eye-Q V2.0	79,00
Monville Manor	69,00
Nucleon	59,00
Ornelo	59,00
Panthe	55,00
Professional Astrologer	159,00
QL Assembler	69,00
QL Bouncer	39,00
QL Camera	39,00
QL Fictionry	45,00
QL Jacket	59,00
QL Macro Assembler	119,00
QL Monitor Storm	39,00
QL Mirror	69,00
QL Paint	66,00
QL Cuboids	69,00
QL Hoverski	39,00
QL Touch and Go	39,00
Reversi V2.0	39,00
Skull	49,00
Super Astrologer Deluxe	69,00
Super Astronome V1.4	49,00
Super Backgammon V3.0	49,00
Super Media Manager	109,00
Super Monitor Disassembler	99,00
Super Sprite Generator	79,00
Supercharge Compiler	79,00
Superfont V2.0	79,00
TechQL	119,00
The Editor	69,00
The Last King of Zulu	69,00
The Lost Phoenix	49,00
Turbo	199,00
Turbo Toolkit	69,00
Wardner	59,00
West	49,00
Worbench Assembler	49,00
Wroom	49,00



Ergänzung zur RAM-Disk von Michael Gehret

In Heft 8-9/86 hatten wir auf Seite 95 die RAM-Disk von Michael Gehret vorgestellt. Dazu teilte uns der Autor einige Neuerungen mit. In der jetzt vorliegenden Version 2.0, die insgesamt ein 1-Megabyte-Laufwerk simuliert, werden aus den TODISK-Ordern nun auch Unterordner, also komplette Anwendungen kopiert. Schreib- und Leseoperationen werden auf dem Bildschirm angezeigt. Das Programm funktioniert auch mit Festplatten und kostet rund 100,- DM. Wer die alte Diskette einschickt, erhält gegen 10,- DM ein Update.

Weitere Informationen:
Michael Gehret Ueware
Sülicherstraße 15
8944 Gröbenbach

Pil-Software Darmstadt

Unser Club besteht schon seit über einem Jahr. Schwerpunkte unserer Arbeit sind: Hilfestellung für Anfänger, Umgang mit Dienst- und Anwenderprogrammen, Erlernen und Programmieren verschiedener

Computersprachen und anderes mehr. Hierzu bieten wir beispielsweise folgende Leistungen: regelmäßige Treffen, Betreuung durch unsere Systemsprecher, Software-Bibliothek, PDS, Club-Zeitung, Zeitschriften-Abo und Fachliteratur.

Der Club-Beitrag wird jährlich entrichtet und ist nach Altersgruppen gestaffelt. Mitmachen können Computer-Freaks mit allen Rechnern.

Wir suchen noch Mitglieder im gesamten Bundesgebiet und im Ausland, um weitere Bezirksgruppen gründen zu können. Infos sind gegen Rückporto beim Club erhältlich.

Peter Schach
PI-Software Darmstadt
Grillpazenzstraße 25
6100 Darmstadt 12
Tel. 061 51/37 325

Neue Pascal-Version

Von Kyan-Pascal (wir berichteten in Ausgabe 6/7) ist jetzt die überarbeitete Version 2.0 erhältlich. Auffällige Neuerung ist die sog. KIX-Benutzeroberfläche, die den Umgang mit Editor und Compiler vereinfacht, da nicht in das DOS zurückgesprungen werden muß.

Weiterhin gibt es nun zum Kyan-Pascal zwei Toolkit-Disketten mit einer Menge zusätzlicher

cher Pascal-Routinen. Das erste Toolkit enthält System-Utilities, das zweite nennt sich "Atari Advanced Graphics" und enthält Routinen zur 3-D-Transformation sowie zur PM-Grafik. Wir werden in einer der nächsten Ausgaben noch genauer darüber berichten.

Weitere Informationen:
Compy-Shop
Gneisenaustr. 29
4330 Mülheim

Peter Finzel

Ein neuer QL?

Aus England erreicht uns die Meldung, daß ein neuer Computer mit der Bezeichnung Thor auf den Markt kommen soll. Das alleine ist natürlich noch keine besondere Nachricht. Für Sinclair-, speziell für

QL-Benutzer ist aber interessant, daß er voll QL-kompatibel sein soll. Gerüchten zufolge soll in Thor die Original-QL-Platine stecken, die nur leicht modifiziert wurde. Leider war es noch nicht möglich, ein Testgerät zu erhalten. Wir bemühen uns aber intensiv darum und hoffen, es bis zur nächsten Ausgabe zu schaffen. Vorab schon mal ein Foto des neuen Computers und einige technische Daten.

- 640 KByte RAM
- eingebaute 3,5-Zoll-Floppy
- optional 20-MByte-Harddisk
- Steckplatz für User-EPROM
- Echtzeituhr
- IBM-ähnliche Tastatur
- Schnittstellen für Maus, Centronics- oder serielle Drucker usw.

Rolf Knorr



Dieses Ding ist voll QL-kompatibel und heißt Thor

SPECTRUM - QL - ATARI - SPECTRUM - QL - ATARI - SPECTRUM - H.G. Dreeser, Soft- und Hardware

Wir bekommen laufend die aktuellsten Produkte für den Spectrum QL, Atari 800/800/130 XL/05E sowie Atari ST. Nutzer Sie unseren Telefon- und Auftragservice zu den angegebenen Zeiten, damit auch Sie über die Neuheiten informiert sind.

Spectrum	QL	Multiville Master	82,80 DM
Lightforce	24,90 DM	Mutation	52,80 DM
Great Escape	24,90 DM	Orwell	46,90 DM
Druid	24,90 DM	Vision	46,90 DM
Galan	24,90 DM	Wanderer 3D	52,90 DM
Infiltrator	33,90 DM	ToolKit 800 3	147,90 DM
Dan Dare	31,90 DM	Macro Assembler	169,90 DM
Dragons Lair	31,90 DM		
Spectrum 128		Hardware	
Fairlight	39,90 DM	AMEX Mouse	(Spec) 209,00 DM
Overblatters	39,90 DM	CS 101	429,00 DM
Hacker	39,90 DM	Spectrum 128 (H)	309,00 DM
Nodes of Yesod	39,90 DM	Discovery one/140	309,00 DM
Pierate	39,90 DM	Discovery one/128	309,00 DM
Sweeps World	39,90 DM	Multitalk One	199,00 DM
Odinator	34,90 DM	Cartridges	4 St. ab 29,90 DM

Formen Sie www.Gesellschaft.de

Dreeser Soft- u. Hardware, im Rosenhof 6, D-5300 Bonn 1, 91 02 26/24 40 DM

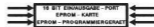
Mo. 10-17, von 11:00-20:00 Uhr, Sa. von 10:00-18:00 Uhr oder Auftragsformulare rund um die Uhr

SPECTRUM - QL - ATARI - SPECTRUM - QL - ATARI - SPECTRUM

SPECTRUM SERVICE

REPARATURZEIT CA. 5 WERKTAGE - 8 MONATE GARANTIE
KOSTENVORANSCHLAG AUF WUNSCH (KOSTENLOS)

MULTICARD 1.0

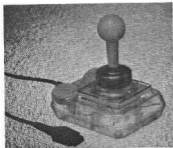


8 GARANTIE IN EINEM - UNZÄHLEIGER ANWENDUNGSMÖGLICHKEITEN
FORDERN SIE VON UNS WEITERE INFORMATIONEN AN
SUPERPREIS NR 169.00 DM inkl. MwSt



BELKENHEID COMPUTERTECHNIK
WELLINGER WEG 54 - 46113 BSLM

TEL. 05408 - 8647



Joystick in durchsichtigem Kunststoffgehäuse

Der Competition-Pro-Joystick ist robust, schnell und präzise. Spiele-Fans wissen das. Neu am Competition-Pro ist jetzt das transparente Kunststoffgehäuse. Da ist Joystick-Technik sichtbar gemacht, und man kann während des Spiels sehen, wie's funktioniert. Natürlich ist auch der neue Joystick mit Mikroswitchern und 8-Wege-Steuerung ausgestattet.

Weitere Informationen:
Dynamics GmbH
Friedensalle 35
2000 Hamburg 50

Der Competition-Pro gewährt Einblicke auf Solides

Soundbox für C 64/C 128

Für die User von C 64 und C 128 mit einem Monitor ohne Lautsprecher ist diese Erweiterung interessant. Die Soundbox stellt einen kompakten, preiswerten HI-FI-Verstärker mit eingebautem Lautsprecher dar, der an der Audio/Video-Buchse des Computers angeschlossen wird. Auch der Monitor arbeitet nach wie vor an diesem Anschluß (bzw. der Antennenbuchse), da alle Leitungen durchgeschleift wurden. Die Stromversorgung erfolgt über vier Mignon-Batterien; der Anschluß eines Netzgerätes ist nicht vorgesehen. Die Lautstärke wird links am Gerät geregelt; eine rote LED dient als Funk-

tionsanzeiger. Das Potentiometer arbeitet leider nicht ganz linear, so daß man bei höheren Lautstärken ein ganz schönes Fingerspitzengefühl braucht.

Es ist schon erstaunlich, was die circa 9x9x5 cm große Box an Sound zu bieten hat. Nur bei tieferen Bafitönen hat sie ihre Probleme, was bei solch geringen Abmessungen nicht verwunderlich ist.

Die Soundbox ist zwar keine Alternative zum Anschluß des Computers an die Stereo-Anlage, stellt aber sicher eine sinnvolle Ergänzung zum Monitor ohne Lautsprecher dar.

System: C 64/C 128
Hersteller: Hirschmann & Theis GmbH
Preis: ca. 78.- DM
Stefan Selbach

Fundgrube

Wer mit einem 3"- oder 3,5"-Laufwerk arbeitet und bisher nicht wußte, wohin mit den teuren Disketten, sollte sich einmal die neue Multiform-Diskettenbox ansehen. Im Gegensatz zu herkömmlichen Boxen ist sie zusammenklappbar und benötigt nur relativ wenig Platz; sie hat dann ungefähr das Format eines mittelstarken Buches. Allerdings können auch nur 12 Disketten untergebracht werden, was für viele Anwender aber ausreichen dürfte.

Weitere Informationen:
Jeposoft
Kruppstraße 9
4040 Neuss 21

Impressum

Herausgeber: Dipl.-Wirt.-Ing. (FH) Thomas Eberle
Worms/Rhein
Chefredakteur: Thomas Eberle
Technische Redaktion: Werner Räte
Redaktion: Helmut Fischer
Rainer Kaltenbrunn
Stadredakteur: Hans Müller
Mitarbeiter: Rolf Kauer
Dieter H. P. Schwesek
Thomas Bertsch
Michael Schürmann
Rainer W. Dörfling
Christian Eberle
Dipl.-Ing. Peter Probst
Thomas Linnert
Karlheinz Hirschner
Martin Kotalla

Verantwortlich: Gabriele Herbig
Titelbild: Rainer Grötsch
Anzeigen: Arno Weiß
Er gibt die Anzeigen-Preise für den Verlag
von J. L. 1986
Montage: Friederike Melchen
Satz: Druckermeister
7941 Veltheim/Elz
Druck: Verlag und Industrie-Druckerei
Graf & Co KG
7730 Völklingen-Schwemmen
Vertrieb: Verlagsanstalt
6200 Wiesbaden
Ansehen des Verlags: Verlag Räte-Eberle
Postfach 243
Mellaachstrasse 11
7114 Bietzen
Tel. 071 52-4248

Haftung und Programmierung werden garantiert von der Redaktion übernommen. Sie müssen Ihre von Redaktion Entnommenen, falls es sich um unentgeltliche oder werblich bezahlte oder gewerblich bezahlte Kopien handelt, dem aus, daß die Angaben werden. Mit der Einreichung von Manuskripten und Zeichnungen geht der Verleger die Zustimmung zum Abdruck in das vom Verlag Räte-Eberle bezahlte Honorar ein. Die Redaktion ist für die Druckfertigkeit der Manuskripte und Zeichnungen verantwortlich. Die Redaktion ist für die Druckfertigkeit der Manuskripte und Zeichnungen verantwortlich. Die Redaktion ist für die Druckfertigkeit der Manuskripte und Zeichnungen verantwortlich.

Computer Kontakt erscheint zweimonatlich am letzten Montag des Vormonats und kostet pro Heft 5,50 DM



lin geschmackvolles Design rundet den positiven Eindruck der Soundbox ab

Star-Painter von Sybex

Achtung: Der "Star-Painter" von Sybex kostet nicht wie irrtümlich im letzten Heft angegeben 79.- DM, sondern nur 64.- DM. Wir bitten dieses Versehen zu entschuldigen.

Inserentenverzeichnis

Albs	S. 108
AMC-Verlag	S. 67
Annette Länger	S. 6
Bellverland	S. 121
C & M Meyer	S. 17
Compy-Shop	S. 71
CSV Fliegen	S. 3
Dressler	S. 87/121
Elektor-Verlag	S. 29
Flanzburg	S. 117
Computerverband	S. 117
Hofschuh	S. 6
Hoffmy-Verlag	S. 44
Individual Software	S. 25
Irata	S. 87
Jakob & Kleinwirth	S. 41
Jeposoft	S. 5
Joystick	S. 119
Jupitersch	S. 3/17
Jürgen Dör	S. 65/76
Kleinher + Schulte	S. 4
Kunst	S. 9
Kornwatz	S. 118
Lücker	S. 6
Naxos	S. 9
new's	S. 9
Rausch + Haub	S. 101
Ritz-Eberle	S. 124
Schiffbauer	S. 120
Schiffbauer	S. 2
telex-Verlag	S. 123
T. S. Darfery-Systeme	S. 123
Ufferkamp	S. 120
Unicom	S. 3/17/120
Wagner	S. 120

ATARI magazin

Das unabhängige Magazin für alle Ataris



**Neu: Ab dem 22.12.
bei Ihrem
Zeitschriftenhändler!**

Tips für die 1050-Floppy
mit Happy-Enhancement