



DM 5.50 ÖS 48 SF 5.50

Magazin

Für alle Schneider Computer

1
Januar '86

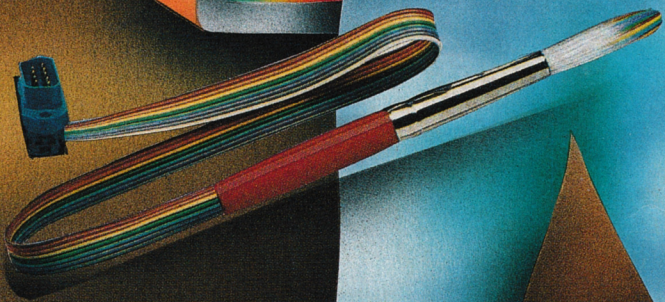
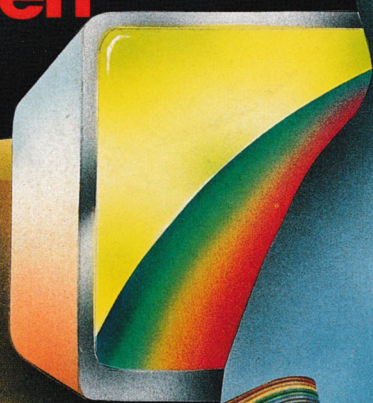
Topprogramm:
Showdown

Pascalkurs

Assemblerecke
Interrupts beim
CPC

Discdoctor
Einblick in jeden
Sektor

Floppykurs



neu
Noch mehr Seiten

pride utilities LTD



(Generalvertretung-Deutschland)

Peter Herzog · D-8240 Berchtesgaden · Postfach 2361
Doktorberg 20 · Telefon 086 52/6 11 99-26 91



* Anwender-Software für den aktiven Schneider-User *

RSX-TRANSMAT

RSX-Transmat ist ein bewährtes Programm, das Ihnen den TRANSFER von Kassetten Software auf 3" Diskette ermöglicht. Außerdem hebt TRANSMAT automatisch den BASIC-Listenschutz beim transferieren auf. RSX-TRANSMAT ist ein Anwenderprogramm auf das kein 3" Floppy Disc BENUTZER verzichten sollte. Funktionsfähig auf dem Schneider CPC 464/664/6128.

DEUTSCHE ANLEITUNG!!!

RSX-Transmat auf **Kassette** nur **DM 44.90**
RSX-Transmat auf **3" Disc** nur **DM 49.90**
RSX-Transmat auf **ROM** nur **DM 79.90**

TRANSMAT-Sonderinfo zum transferieren von über 60 Problemprogrammen. nur **DM 6.90**

Mehr über die hochwertige PRIDE UTILITIES-SOFTWARE erfahren Sie in unserem **GRATIS KATALOG**

Händleranfragen erwünscht

ODDJOB (464/664/6128)

ODDJOB, ein Diskettendienstprogramm das die vielen Möglichkeiten der 3" Diskettenstation kinderleicht handhaben läßt.

ODDJOB KANN

1. In der Directory alles verändern und editieren.
2. Gelöschte Files wieder »zurückholen«.
3. Kopieren von Sektoren. ID's werden angezeigt.
4. Sektoren vollständig auslesen und verändern in HEX und ASCII inkl. EDITOR.
5. Die Position eines Tracks/Sektors feststellen.
6. Diskettensoftware auf Kassette abspeichern.
7. Formatieren in DOPPELTER Geschwindigkeit.
8. Den BASIC-Listenschutz aufheben.
9. Die Laufwerkgeschwindigkeit um 20 % erhöhen.
10. DEUTSCHE ANLEITUNG u.v.a.m.

Ein **MUSS** für **JEDEN 3" Floppy-User!!!**

ODDJOB auf 3" Diskette nur

DM 59.90

Wie immer TOPSOFTWARE zum »benutzerfreundlichen« Preis

CPC-Kassetten: (464/664/6128)

Wintergames nur **DM 39.90**
VIEW TO A KILL nur **DM 29.90**
CODENAME MAT II nur **DM 29.90**
FRIDAY THE 13th nur **DM 34.90**
(NEU – Nichts für schwache Nerven)
SABRE WULF nur **DM 29.90**
ALIEN 8 nur **DM 34.90**
KNIGHT LORE nur **DM 34.90**
NIGHTSHADE nur **DM 34.90**
CRITICAL MASS **DM 39.90**
Hacker **DM 39.90**
Never Ending Story **DM 39.90**
Highway Encounter **DM 39.90**
Fighter Pilot (Deutsch) **DM 36.90**
Spy vs Spy **DM 39.90**
The Way of Exploding First
DM 34.90
Match Day nur **DM 34.90**
Starion **DM 34.90**
Sorcery 464 nur **DM 24.90**
They sold a Million **DM 39.90**
D. Th. Decatlon **DM 29.90**
Daley Thompson Supertest
DM 34.90
Slapshot **DM 34.90**
HYPERSPORTS **DM 34.90**
Yie are Kung Fu **DM 34.90**
ELITE **DM 39.90**
CYBERON – ULTIMATE
DM 39.90
GUNFRIGHT – ULTIMATE
DM 39.90

Ständig Neuheiten

CPC-Computer Dictionary 464/664/6128

(DAS Schneider SPRACHPROGRAMM)

- * EINZIGARTIG am Softwaremarkt
- * Über 20.000 fest gespeicherte Vokabeln
- * Über 10.000 Stichwörter
- * Minimale Zugriffszeiten
- * Selbst individuell ERWEITERBAR
- * Lernerfolg durch Vokabeltrainer
- * Nutzt die CPC und Floppy-Speicherkapazität voll aus
- * Eine GROSSE Hilfe in SCHULE und Beruf
- * Eine echte Bereicherung Ihrer Schneider-Softwaresammlung
- * Deutsche Anleitung und Menüführung
- * 3" und 5.25" Diskette

ENGLISCH/DEUTSCH nur **DM 69.90**
DEUTSCH/ENGLISCH nur **DM 69.90**

SET DEUTSCH/ENGLISCH
und ENGLISCH/DEUTSCH
nur **DM 119.90**

**SCHNELLVERSAND
GRATIS-KATALOG**

3" Disketten:

A View to a Kill **DM 49.90**
Airwolf **DM 49.90**
Fighter Pilot **DM 49.90**
Codename Mat II **DM 39.90**
Hyper Sports **DM 44.90**
Yie are Kung Fu **DM 44.90**
Spy vs Spy **DM 49.90**
Red Arrows **DM 49.90**
Slapshot **DM 49.90**

Anwendersoftware:

Masterfile Kassette **DM 99.90**
Masterfile 3" Disk. **DM 119.90**
Tasword Deutsch Kass. **DM 59.90**
Tasword-D 3" Disk. **DM 98.90**
Tasword-D 5.25" Disk. **DM 98.90**
Tasprint Kassette **DM 34.90**
Tascopy Kassette **DM 34.90**

Tasword-D+Tasprint
+ Tascopy 3" od. 5.25" **DM 149.90**

Tonkopfreinigungs und Entmagnetisierungskassette **DM 7.90**

Bei Bestellungen unter DM 70 werden DM 4.— Porto/Verpackung berechnet. Die Lieferung erfolgt per Nachnahme oder Vorkasse (Euro-Scheck). Telefonische Bestellung und Beratung ist möglich!!!



Liebe Leser,

unsere Erstaussage war ein voller Erfolg. Bis jetzt sind schon über 4000 Fragebogen bei uns eingegangen und täglich werden es mehr. Wenn möglich, werden wir die ersten Ergebnisse dieser Befragung schon im nächsten Heft veröffentlichen. Auch die glücklichen Gewinner der

Preise werden dann bekanntgegeben. Lassen Sie sich überraschen.

Die meisten Leser wollen das CPC-Magazin auf jeden Fall regelmäßig lesen. Damit das auch so bleibt, werden wir uns in Zukunft besonders anstrengen und aus dem CPC-Magazin eine lesernahe und leistungsfähige Zeitschrift machen. Wer also mit unserer Erstaussage nicht so ganz zufrieden war: Diesmal sind's gleich 16 Seiten mehr.

Eine leichte Veränderung hat es beim Titelbild gegeben. Hier meinte Data Media, wir würden der CPC-International zu ähnlich sehen. Da haben wir den Schriftzug eben geändert.

Zum ersten Mal bringen wir in dieser Ausgabe das Topprogramm des Monats. 1000 DM Honorar erhalten hier Andreas Zallmann und Christoph Schillo für ihr Superprogramm "Showdown in Tombstone City", bei dem Sie als Marshal von Dodge City zum Zweikampf mit skrupellosen Revolverhelden geordert werden.

Neu sind die 500 DM für den besten Trick oder Tip. Vielleicht haben gerade Sie diesen 500-DM-Tip in der Schublade, schauen Sie mal nach.

Und wer noch kein Abo hat, sollte bei unserem Logical mitmachen. Dort gibt es 30 Abos zu gewinnen. Ich bin sicher, Sie werden die Lösung ziemlich schnell gefunden haben.

Jetzt möchte ich Ihnen nur noch ein frohes Weihnachtsfest und ein glückliches neues Jahr wünschen, auch wenn meine Wünsche Sie vielleicht erst im neuen Jahr erreichen sollten.

Bis zum nächsten Mal
Ihr

Thomas Eberle, Chefredakteur

★★★★★★★★★★★★★★

Unser Weihnachts- angebot für Sie:

★★★★★★★★★★★★★★

Matrixdrucker

RITEMAN-INFORUNNER

- ☆ Traktor und Einzelblatt
 - ☆ 120 CPS – 9×9 Matrix
 - ☆ 40/66/80/132 Zeichen pro Zeile
 - ☆ Standard Centronics-Schnittstelle
 - ☆ Epson-Kompatibel
- alles für nur **DM 898.–** inkl. MWSt.

VALCOM I – Die serielle Schnittstelle für Ihren CPC

- ☆ für DFÜ, Drucker/Plotter, Rechner-Rechner-Kopplung
 - ☆ für Meß- und Labortechnik
 - ☆ einfache Bedienung durch ausführliches Handbuch
 - ☆ 5 verschiedene Schnittstellen für alle Ansprüche
 - ☆ Einbauschchnittstelle für den CPC 664
- Preis ab **DM 189.–** bis **DM 368.–**
 ☆ anschlussfertiges Komplettangebot (VALCOM I RS, datafons 21 d, MODEM I, Kabel)
 für nur **DM 538.–** inkl. MWSt.

Informationen zu diesen oder unseren anderen Produkten (Centronics DIN A3-Plotter, 5 1/4"-Floppy,...) erhalten Sie unter folgender Adresse:

**VALC Computer,
Peter Kohl, Waaggasse 4,
8230 Bad Reichenhall
Telefon 0 86 51 / 6 67 73**

DM 99.–

UNICON

Das Datenkommunikationspaket "UNiversal CONnection" ermöglicht Ihnen den Dialog bzw. den Programmaustausch mit Mailboxen, Datenbanken und anderen Benutzern.

Im Lieferumfang enthalten: UNICON-Software (Disketten- und Kassettenversion) mit ausführlicher Dokumentation sowie eine anschlussfertige Verbindung zum Akustikkoppler.

- Floppy und Drucker gleichzeitig anschließbar
- Dialog-Betrieb
- Parameter-Veränderungen
- Voll/Halb-Duplex
- Abspeichern auf Kassette/Disk
- Übertragen von Programmen (Hex und ASCII)
- Dienstprogramme
- Textverarbeitung
- BASIC-Erweiterung für eigene Anwendungen
- Deutscher Zeichensatz wählbar
- CPC 464, 664, 6128 (Version bei Bestellung angeben!)
- und, und, und

Kostenloses Info oder Bestellung bei:

**GERDES, Imperial Software
Systems, Heidegartenstr. 36
5300 Bonn 1, ☎ 02 28 / 25 24 74**

INHALTSVERZEICHNIS

RUBRIKEN

Vorwort	3
CPC-News	5
Bücher	6
Software-Service	29
Logical	47
Buchversand	58
Topprogramm des Monats	68
Leserforum	84
Clubs	84
Spielecke	85
Kleinanzeigen	87
Vorschau	90

SOFTWARE-REVIEWS/SPIELE

Death Pit	8
Macadam Bumper	8
Dragontorc	8
Hacker	9
Robin of Sherwood	9

SOFTWARE-REVIEWS/ANWENDERPROGRAMME

Diagramm	9
Teleterminal 300 S	10
ZEN-Assembler	10
CPC-Text/Adress	12
dBase II	14
Textmaster	21

BERICHTE

Der Joyce im Test	16
PIO-Interface und Epromer	18
Die Amstrad Consumer Show	19
Der sprechende Computer	20
RAM ist im kleinsten Schneider	22

SERIEN

Floppy-Kurs (Teil 2)	30
Z 80-Assemblerkurs (Teil 2)	33
CP/M – die neue Perspektive (Teil 2)	36
Pascal-Kurs	39
Z 80-Assemblertips (Teil 2)	44
Extended Basic	50

TIPS & TRICKS

Für Zwischendurch...	27
7 auf einen Streich (Teil 2)	48
Schriften mischen mit dem CPC 464	52
Der File-Examiner	53
Neue Programme mit Sesam	54
Machen Sie Ihre Disketten fit...	64

ANWENDERPROGRAMME

Dateiverwaltung für den Schneider	59
Lottoglück mit dem Schneider CPC	76
CPC-Orgel	80




Wir bringen einen Erfahrungsbericht über den neuen Joyce PCW 8256. Thomas Tai hat ihn getestet. Alles weitere lesen Sie auf Seite 16.



Vor einigen Wochen fand in London die Amstrad Consumer Show statt, auf der nur Amstrad/CPC Neuigkeiten gezeigt wurden. Unseren Bericht finden Sie auf Seite 19.

TOWN IN FEAR AND PANIC !

**MARSHAL MATT DILLON OF
TOMBSTONE CITY SHOT DOWN BY
CRUEL, BRUTAL BANDIT !**



BILLY THE KID

**MARSHAL MATT DILLON LEAVES
A WIFE, A COW AND 1000 \$.**

**BEST MARSHAL YET MATT DILLON
WHO EARNED 1000 \$.**

**NEW MARSHAL WANTED. GOOD WAGES
ARE YOU BRAVE ENOUGH TO ACCEPT
THIS DANGEROUS JOB (YEAH/NOP)?**

Spielen Sie Marshall und lassen Sie sich blaue Bohnen um die Ohren fliegen. Unser Topprogramm Showdown macht's möglich. Das Listing finden Sie ab Seite 68.

C 64-Drucker an Schneider Computer anschließen

Mit Hilfe des Interfaces Typ 29000 (ohne Pufferspeicher) oder Typ 29008 (mit 8 KByte Pufferspeicher) kann man Drucker, die für den Anschluß am C 64 gedacht sind, auch an Schneider-Computer anschließen. Der Adapter wird einfach zwischen Computer und Drucker geschaltet, so daß der Drucker weiterhin für den C 64 verwendbar ist. In der Version 29008 ist ein 8 KByte großer Pufferspeicher vorhanden, durch den gleichzeitig gedruckt und weitergearbeitet werden kann. Die Geräte sind für ca. 248.- DM (Typ 29000) bzw. ca. 298.- DM (Typ 29008) beim Hersteller verfügbar.

2 Computer an einen Drucker anschließen

Der neue Schnittstellen-ervielfacher Typ 2C/C/0 erlaubt, einen Drucker gleichzeitig an zwei Computer anzuschließen (Centronics-Schnittstellen). Im Gegensatz zu einfachen manuellen Umschaltern (die zudem meist wesentlich teurer sind), arbeitet dieses Gerät auch automatisch. Zunächst können beide Computer Daten senden. Der Computer, der dann zuerst anfängt auszudrucken, blockiert sofort den anderen. Erst wenn der gerade druckende Computer einige Sekunden keine Daten mehr sendet, kann der andere anfangen zu drucken. Auf diese Weise wird eine völlig automatische Umschaltung zwischen beiden Rechnern erreicht und die unerwünschte Mischung verschiedener Texte vermieden.

Der Adapter kostet ca. 248.- DM und ist sofort lieferbar.

Das Diskettenwerkzeug

Hier handelt es sich um ein Werkzeug zur Direktbearbeitung von Dateien und Datenträgern. Das Programm läuft unter CP/M und CP/M-Plus auf Rechner mit Z-80 CPU und benötigt eine TPA von lediglich knapp 40 Kilobyte, so daß es auch mit sehr kleinen Systemen einsetzbar ist.

Die Kommunikation Nutzer-Rechner erfolgt bei der Arbeit mit DWerk in deutschem Klartext über 6 verschiedene Arbeitsschirme aus mehreren Feldern. Das Redigieren des Datenträger-Inhaltes geschieht dabei vollschirmorientiert in Wordstar-Manier. Zu den Besonderheiten des Programms zählt ein simulierter Doppel-Cursor: Bei Editierbetrieb arbeitet man simultan in je einem Hexadezimal- und einem ASCII-Feld.

DWerk dient der Inspektion, der Reparatur und dem "Tuning" des Inhaltes von Disketten, RAM-Floppies und Bubble-Memories. Eine ausführliche Dokumentation erklärt die Organisation von CP/M-Datenträgern und führt eine Fallsammlung mit zahlreichen Anwendungsbeispielen von DWerk vor. Zu ihnen zählen Restaurieren gelöschter oder beschädigter Dateien, Einbau und Analyse von Kopierschutz-Mechanismen sowie die Eröffnung des Zuganges zu passwortgeschützten Dateien ohne Kenntnis des Passwortes.

Neben seiner Anwendbarkeit als Praxis-Werkzeug kann das Programm nach Meinung der Vertriebsfirma als Studien- und Lehrobjekt zu einem vertieften Verständnis von CP/M verhelpen. DWerk wird einschließlich eines leistungsfähigen Installationsprogramms in zahlreichen Diskettenformaten (3, 3 1/2, 5 1/4 und 8 Zoll) zu einem Einführungspreis von unter 100.- Mark geliefert.

Reihe
FREIZEIT & COMPUTER

**So haben Sie Ihren
Schneider-Computer
spielend im Griff.**

NEU

FREIZEIT & COMPUTER

Dieter Winkler

**DAS SCHNEIDER
CPC 6128
664
PRAXIS BUCH**

MIT KOMPLETTEM SOFTWAREFÜHRER

Alle Schneider-Computer im Vergleich • Die drei Betriebssysteme des CPC 6128 • Grundlagen des Commodore BASIC 1.1 • Die wichtigsten Schneider-Programmiersprachen • Wie man mit dem Computer arbeitet • Grafik, Sound und andere Anwendungen • Alle über Farbregener und Zahlen

SIGNUM
Computer-Bücher

Ca. 240 Seiten, DM 29,80
ISBN 3-924767-06-8

Ab sofort im
Buch- und Computer-
fachhandel erhältlich.

Mit diesem Buch erfahren Sie alles über den Einsatz von Computern in der Freizeit, beim Hobby und bei den verschiedenen Heim-Anwendungen.

Aus dem Inhalt:
Alle Schneider-Computer im Vergleich. Die drei Betriebssysteme des CPC 6128. Grundlagen des Locomotive BASICs 1:1. Die wichtigsten Schneider-Programmiersprachen. Was man mit den Computern anfangen kann. Grafik, Sound und nützliche Anwendungen. Alles über Peripheriegeräte, Zubehör und gesamte Software.

Fordern Sie unseren Computerbuch-Prospekt an.

Signum Medien Verlag GmbH
Eisenheimerstraße 59
8000 München 21
Tel. 0 89/5 70 58 92

SCHNEIDER Vertragshändler

RSE

Schneider	Kass.	Disk.
Vereinsverwaltung	57.-	69.-
Adressverwaltung	57.-	69.-
Faktura	57.-	69.-
Micro Expertensystem	19.80	29.80
Hardcopy	19.80	29.80
Kybernetia	27.90	39.80
TexPac	---	198.-
Roland Ahoy	39.-	49.-
Sorcery	35.-	
Tennis	39.50	
Pool Billard	39.50	
Manic Miner	19.-	
Ghostbusters	39.-	
Loco-Motion	11.95	
House of Usher	29.-	
Finders Keepers	11.95	
Nonterraqueous	11.95	
The wild Bunch	11.95	
Subsunk	11.95	
Short's Fuse	11.95	
Mr. Freeze	11.95	
Chiller	11.95	

COMMODORE Vertragswerkstatt

SOFTWARE
und
PERIPHERIE

Commodore	Hardware	a. Anfr.
CPC 464	grün	798.-
CPC 464	Farbe	1298.-
CPC 664	grün	1398.-
CPC 664	Farbe	1898.-
CPC 6128	grün	1598.-
CPC 6128	Farbe	2098.-
DDI-1		798.-
NLQ 401		798.-
FD-1		698.-
Joyce PCW 8256 inklusive Monitor, Floppy, Drucker, Textverarbeitungsprogramm		2490.-
Quickshot 1		24.90
Quickshot 2		29.80
Kunststoff-Abdeckhauben		
Rauchglas für:		
Commodore 64/20/16		17.90
Commodore 1541		24.90
Schneider CPC 464		24.90
Schneider CPC 664		24.90
Leerkassetten ab		-.99
Disketten 5 1/4		
DS/DD 10 Stück		27.90

Alle Preise inklusive MwSt. Versand per Vorkasse oder Nachnahme zzgl. 5.- DM Versandkosten. Info gegen 2.- DM in Briefmarken. Händleranfragen erwünscht.

R. Schuster Electronic
Obere Münsterstraße 33
4620 Castrop-Rauxel
☎ 0 23 05 / 37 70

Heft
12 '85



Kann nachbestellt werden. DM 7.- (inkl. Porto) in Briefmarken an

CPC-Magazin
Postfach 1640
7518 Bretten

DATA-SERVICE

Th. Piotrowski u. H. Langener
Grenzweg 47, ☎ 02594/83108
Mailbox: 02594/85991,
4408 Dülmen 1

Dataphon S21D	289.-
Dataphon S21D Komplettsset	389.-
S21D, Terminalpr., Datenkabel	
CPC 664 Grünmonitor	1398.-
CPC 664 Farbmonitor	1858.-
CPC 6128 Grünmonitor	1598.-
CPC 6128 Farbmonitor	2098.-
NLQ 401 Matrixdrucker	798.-
STAR SG 10 120 Z/s.	898.-
STAR SD 10 160 Z/s.	1298.-
Dun Darach (K)	32.-
Fantastic Voyage (K)	32.-
Frank Bruno's Boxing (K)	32.-
Gremlins (K)	32.-
Jewels of Babylon (K)	25.-
Return to Eden (K)	39.-
The Way of Exploding Fist (K)	39.-

Fordern Sie noch heute unsere umfangreiche Liste an!!!

Schneider

imfo IMBECK
hat's

imfo-Blätter, die es
in sich haben.
Für DM 3.- von imfo

Postfach 2015 51
2000 Hamburg 20

Was der CPC 464 alles kann

Von Martin Aschoff
Vogel-Verlag
156 Seiten, DM 28.-
ISBN 3-8023-0841-7

Einführungsbücher für die CPC-Computer gibt es eigentlich schon genug, wenn es auch verständlich ist, daß kein Verlag sich ein Geschäft entgehen lassen möchte. So hat nun auch der Vogel-Verlag ein Buch auf den Markt gebracht, das sich speziell an die CPC-Einsteiger wendet. Es soll das Schneider-Handbuch ergänzen und dem Neuling helfen, seinen Computer zu verstehen.

Gegliedert ist das Buch in drei Kapitel, die jeweils der Hardware, der Firmware und der Software gewidmet sind. Im 1. Kapitel wird kurz erläutert, wie der CPC intern arbeitet, was der Benutzer alles mitgekauft hat, wie er es benutzen kann und was es an Peripheriegeräten sonst noch alles gibt (diese Aufstellung ist aber unvollständig). Auch die verschiedenen Schnittstellen des Computers werden hier erläutert. Wer noch gar nichts über Computer weiß, wird sich über dieses Kapitel sicher freuen.



Das nächste Kapitel beginnt mit Erläuterungen über die Tastaturfunktionen, den Editor und den Basic-Interpreter. An dieser Stelle wird der Leser auch mit den ersten Basic-Befehlen konfrontiert. Weiter geht es mit den Grafik- und Soundmöglichkeiten und dem Betriebssystem, mit dem das Kapitel endet.

Die Software des CPC 464, das letzte Buchkapitel, ist auch gleichzeitig das umfangreichste. Allerdings wird hier nicht das Schneider (Locomotive)-Basic

erläutert, vielmehr hat der Buchautor hier verschiedene Programme zusammengetragen und als Listings wiedergegeben. Dabei sind einige durchaus nützliche Programme vertreten, andere wiederum sind die Arbeit des Eintippens kaum wert.

Wer sich dieses Buch kauft, um Basic zu lernen, ist schlecht beraten. Wer seinen Computer etwas besser kennenlernen will, kann das mit diesem Werk. Es muß aber ganz klar gesagt werden, daß erheblich bessere Bücher auf dem Markt sind, die für gleiches Geld mehr bieten.

Rolf Knorre

CPC 464 – Programmieren in Maschinsprache

Von Carsten Straush
Verlag Markt & Technik
270 Seiten, 46,- DM
ISBN 3-89090-166-2

Hier beschäftigt sich der Autor nicht nur mit der reinen Assemblerprogrammierung, er faßt das Problem mehr von unten an und beginnt mit der reinen Maschinspracheprogrammierung ohne Hilfsmittel. Er geht darauf ein, wie z.B. ein Basicprogramm im Speicher abgelegt ist oder wie Variablen abgespeichert sind. Dieses erweiterte Themengebiet macht sich auch im Umfang des Buches bemerkbar.

Der Autor beginnt sein Buch mit den verschiedenen Datenformaten und Zahlensystemen. Dann wird der Speicheraufbau des CPCs und die Restarts erklärt. Im Kapitel 2 wird ein Monitor programmiert, mit dem erste Schritte in MC unternommen werden können. In Kapitel 3 folgt dann eine Hardwareübersicht. Außerdem wird der Soundchip und der Schnittstellenbaustein 8255 erläutert. Kapitel 4 widmet sich ganz dem Z80A Prozessor. Hier wird in die Register eingeführt und ein Programm geschrieben, welches die Inhalte der Register vom Basic aus abprüfbar macht. In Kapitel 5, dem umfangreichsten, stellt der Autor dann den Befehlssatz des Z80 mit den Adressierungsarten und den Flags vor. Und schließlich wird im sechsten Kapitel ein Disassembler entwickelt und es werden

einige ROMroutinen aufgeführt, die dem Leser von Nutzen sein können.



Die im Buch entwickelten Programme muß man übrigens nicht eintippen, sie können auch beim Verlag bestellt werden.

Im Buch werden Beispielprogramme immer von Hand übersetzt und dann eingepoket. Das mag zwar gegenüber der Assemblerprogrammierung unverständlich erscheinen, trägt aber ganz wesentlich dazu bei, daß sich der Leser mehr in die Funktion des Z80 hineindenken kann. Kauft man sich dann noch einen Assembler, so kann man selbstverständlich sehr gut mit dem Buch arbeiten.

Andreas Zallmann

Schneider CPC 464 – Software und Trainingsbuch – Assemblerkurs

Von Tim Herbertson
Sybex Verlag
231 Seiten, DM 64,-
ISBN 3-88745-412-X

Der Assemblerkurs besteht aus einem Buch und einer Cassette. Der Autor beginnt hier allerdings nicht auf dem niedrigsten Level, der Maschinenprogrammierung, sondern auf einer höheren Stufe, der Assemblerprogrammierung. Zu diesem Zweck befindet sich auf der beiliegenden Cassette neben einigen anderen nicht so wichtigen Hilfsprogrammen (z.B. Zahlenumrechnung) ein recht ordentlicher Assembler. Zwar ist er nicht so vollkommen, wie ein

Profiassembler (z.B. DEVPAC von SCHNEIDER) und ist größtenteils in Basic geschrieben, aber für die ersten Schritte in Maschinensprache reicht er allemal aus.

Schon im ersten Kapitel werden Maschinenbefehle erklärt. Es folgen weitere Befehle, Register, Adressierungsarten, Grundrechenarten und logische Operationen. Die erste Multiplikation, die in Maschinensprache gar nicht so einfach ist, wird dann in Kapitel 6 programmiert. Im neunten Kapitel geht der Autor dann sogar auf die Interrupts ein und im zehnten Kapitel soll eine Grafikerweiterung in Maschinensprache mit Hilfe der RSX-Befehle programmiert werden.

Als krönender Abschluß des Buches wird eine Routine zum Zeichnen eines Kreises programmiert. Hierzu hat der Autor einen Algorithmus entwickelt, der ohne die Verwendung von Sinus und Cosinus auskommt. Die Grafikerweiterung pur und eine Demonstration derselben befindet sich ebenfalls auf der beiliegenden Cassette, so daß man die Erweiterung nicht eintippen muß.



Danach folgen noch einhundert Seiten Anhänge: Der Befehlssatz wird schematisch dargestellt und die Wirkung der Befehle auf die Flags erläutert. Im Anhang D werden schließlich Einsprungadressen ins ROM mit ihren Ein- und Aussprungsbedingungen angeführt. Erst im Anhang E folgen dann hexadezimale und duale Zahlen. Es wäre wohl besser gewesen, wenn man dieses Kapitel ziemlich an den Anfang gestellt hätte. Schließlich werden diese Darstellungsarten schon in allen Kapiteln fleißig angewandt. Anhang F beschreibt dann genau die Arbeit mit dem mitgeliefer-

ten Assembler und in Anhang G finden Sie die Lösungen zu den überall eingestreuten Aufgaben. Schließlich gibt der Autor noch Tips zur Konvertierung der Programme von Cassette auf Diskette und zur Speicherbelegung des CPC.

Insgesamt ist das Buch aber gut aufgebaut und sehr lehrreich. Hier wird anspruchsvoller Stoff gut verpackt an den Mann gebracht. Wer dieses Buch ernsthaft durcharbeitet und die Aufgaben löst, der hat ohne große Schwierigkeiten MC gelernt.

Andreas Zallmann

Start in die künstliche Intelligenz

Von Jeremy Vine
Vogel-Verlag
100 Seiten, DM 23.-
ISBN 3-8023-0863-8

Künstliche Intelligenz ist eigentlich ein Stichwort, das man bisher kaum mit Heimcomputern in Zusammenhang gebracht hat. Noch vor relativ kurzer Zeit waren nur Großcomputer in der Lage, entsprechende Datenmengen zu verarbeiten. Der enorme technische Fortschritt bei den Microcomputern hat jedoch dazu geführt, daß neben den natürlich immer noch beliebten Spielen auch andere, ernsthaftere Einsatzmöglichkeiten gesucht und gefunden wurden, wie eben auch die Materie der künstlichen Intelligenz.

Der Vogel-Verlag hat nun zu diesem Thema ein Buch auf den Markt gebracht, das sich speziell an die Benutzer des CPC 464 wendet. Es beginnt mit einer äußerst knappen Einführung des Themas. Der Autor startet danach sofort mit der Basic-Programmierung eines Dialogprogramms bzw. den Grundregeln dazu. Schritt für Schritt werden die wichtigsten Basicbefehle erläutert und anhand kurzer Beispiele veranschaulicht.

Erst ab Seite 57 geht es mit dem kompletten Dialogprogramm "Sigmund" richtig zur Sache. Warum das Programm allerdings englische Texte und Kommentare enthält bzw. warum diese nicht übersetzt wurden, ist nicht ersichtlich. Besonders neu ist die Programmidee auch nicht, es handelt sich

um eine Mini-Ausgabe von "ELIZA", was sicherlich einigen Lesern bereits ein Begriff ist.



Das Buch endet mit einem Basic-Schnellkurs, der aber nur wenige Seiten umfaßt. Wer sich wirklich für die Thematik der künstlichen Intelligenz interessiert, wird mit diesem Werk

kaum zufrieden sein. Es handelt sich meiner Meinung nach doch eher um eine Basic-Einführung, die sich aber nur auf einige wenige Befehle wie z.B. INPUT oder INKEY beschränkt.

Rolf Knorke

Der Computer-Shop in Haagen bietet für Ihren CPC:

Abdeckhauben aus Kunstleder für	
Grünmonitor	DM 25.-
Farbmonitor	DM 28.-
Keyboard 464/664/6128	DM 19.95
NLQ-Drucker	DM 19.95
Floppy	DM 17.95
Diskettenbox für 40 Disketten, ohne Schloß	DM 39.70
Kontrastscheibe	
Grün-/Farbmonitor	DM 43.-/46.-
Centronics-Drucker	DM 698.-
Software/Kassette	
Wintergames	DM 49.50
Bounty Bob	DM 43.90
Bruce Lee	DM 39.90
Spit-Fire 40	DM 39.-
Wintersport	DM 49.50
Nightshade	DM 43.90
Arcade-Hall of Fame	DM 43.90
RH-Büro 3", 5 1/4"	DM 69.-
RH-DAT 3", 5 1/4"	DM 79.-

Lieferbedingungen per Nachnahme oder Vorkasse (Scheck) zzgl. Versandkostenpauschale DM 6.-.

SFK-Elektroinstallations GmbH, Delsterner Str. 23, 5800 Haagen-Delstern
☎ 0 23 31 / 7 26 08

Gegendarstellung

In der Zeitschrift "CPC-Magazin" Ausgabe November/Dezember 1985 vom 28.10.1985 ist auf Seite 6 und 7 ein Beitrag unter der Überschrift "CPC-Perspektiven - Schneider auf der internationalen Funkausstellung 1985 in Berlin" enthalten, der unrichtige Behauptungen enthält, die ich wie folgt richtigstelle.

1) Unrichtig ist die Behauptung, daß Herr Thomas Jacobi als Verfasser des Berichtes aufgrund besonderer Unfreundlichkeit des Standpersonals leider keine Informationen über die von der Firma Data Media angebotene Software und Hardware habe ergattern können. Richtig ist, daß Herr Jacobi am Ausstellungsstand der Firma Data Media GmbH auf der internationalen Funkausstellung 1985 in Berlin vorgesprochen hat und die kostenlose Übergabe eines Exemplars der seitens der Firma Data Media für den Computer "Schneider CPC 464" und "Schneider CPC 664" angebotene Speichererweiterung beehrte. Dieses Ansinn-

nen wurde von einem der Gesellschafter der Firma Data Media GmbH zurückgewiesen. Daneben beehrte Herr Jacobi keinerlei weitere Informationen.

2) Unrichtig ist die Behauptung, daß die Firma Data Media keine Speichererweiterung herausbringen werde. Richtig ist vielmehr, daß die Firma Data Media GmbH für den Computer "Schneider CPC 464" und "Schneider CPC 664" Speichererweiterungen mit einer Kapazität zwischen 64 und 512 Kilobyte zu Preisen zwischen 128,- und 498,- DM vertreibt.

Data Media GmbH,
Bereich Software, Dortmund

Anmerkung der Redaktion:

- Herr Jacobi hat um die leihweise Übergabe eines Exemplars zu Testzwecken gebeten.
- Bei der jetzt von der Firma Data Media angebotenen Speichererweiterung handelt es sich nicht um die gleiche Entwicklung, die damals auf der Funkausstellung angeboten wurde.

Macadam Bumper

Flipper sind die einzigen Spielhallenautomaten, die kaum eine zufriedenstellende Umsetzung auf den Computer erfahren haben. Das liegt zum Teil daran, daß die Eigenarten dieses Gerätes mit realistischen Geräuschen und Kugelbewegungen einige Schwierigkeiten bereiten. Der Hauptgrund liegt wohl aber auch bei der mangelnden Spielmotivation, die man nach zwanzig Spielen an demselben Gerät kaum noch verspürt.

Bei Macadam Bumper wurden die meisten dieser Hindernisse beseitigt. Natürlich kann bei keinem Computerspiel das richtige "Flipperfeeling" aufgenommen, da man verständlicherweise auf seitliche Druckknöpfe und Scoreboards verzichten mußte. Das wärs aber auch schon alles, was zur ungetrübten Flipperfreude fehlt. Die Kugel bewegt sich ruckfrei und auch der Sound holt alles aus dem Schneider Piepser heraus. Wer jetzt allerdings glaubt, den Expansionsport mit Markstücken vollstopfen zu müssen, der irrt: ein Druck auf die "C"-Taste genügt und die Investition ist getätigt.

Gespielt wird mit zwei Tasten, wobei auch die Stärke, mit der man die Kugel ins Spiel bringen will, bestimmt werden

kann. Nach einigen Spielen mit dem vorgegebenen Flipper verlangt es einen aber nach neuen Aufgaben und man kann nun der Phantasie freien Lauf lassen. Auf komfortable Art und Weise läßt sich ein vollkommen neues Flipperlayout schaffen. Nach dem Baukastenprinzip werden die einzelnen Komponenten ausgewählt und an die gewünschte Stelle gesetzt. Es gehört allerdings einige Übung dazu, einen funktionierenden Flipper aufzubauen, da die Kugel nirgendwo stecken- oder liegenbleiben darf. Zur Probe kann auch einfach das bestehende Layout verändert werden. Punktvorgabe und Spielgeschwindigkeit sind ebenfalls variabel.

Macadam Bumper ist für alle Flipperfreunde ein hervorragendes Programm. Bei der Umsetzung wurde alles aus dem Computer herausgeholt, was möglich war. Besonders begrüßenswert ist dabei die Möglichkeit, eigene Flipper zu entwerfen und abzuspeichern. Wer den guten alten Automaten vermißt, sollte sich das Programm trotzdem anschaffen, denn für das beste Layout wurde als erster Preis ein richtiger Flipper ausgesetzt.

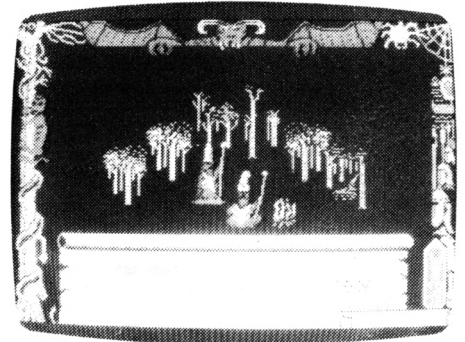
System: CPC 464/664
Hersteller: PSS
Preis: ca. 39.- DM

Thomas Tai

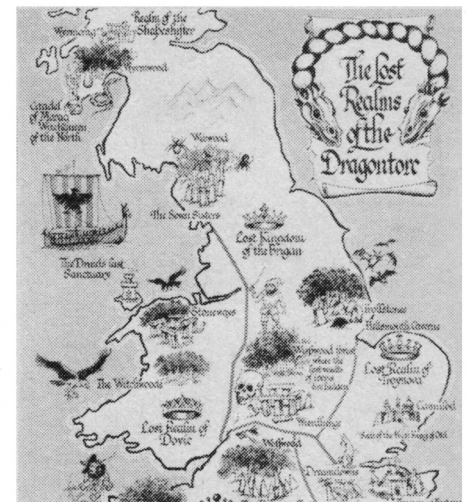
Dragontorc

Fünf Kronen sind es, die gefunden werden müssen, um Merlin vom bösen Morag zu befreien. Damit wäre wohl auch schon klar, daß es sich bei Dragontorc um ein Adventure mit einigen Besonderheiten handelt.

Da wäre erst einmal die Tatsache, daß man keinerlei Texte eingeben muß. Die Hauptfigur bewegt sich über die Tastatur. Wenn Kommunikation notwendig wird, kann ein kleines Menü eingeblendet werden, aus dem der Spieler dann die Aktionen abrufen kann. Ansonsten



läuft das Programm eher wie ein Trickfilm ab. Dazu trägt auch das neue Verfahren "Sensory Animation" bei, das meinen Geschmack aber nicht trifft. Durch das ständige (sanfte) Scrollen des Bildschirminhaltes hat man das Gefühl, starken Seegang auf einem Schiff zu erleben. Das ist aber subjektiv, muß also jeder selbst entscheiden.



Ausgerüstet mit ausführlichen (englischen) Instruktionen und einer kleinen Karte mit der Adventurelandschaft muß sich der Spieler auf die Reise durch eine überwiegend recht gute Grafik machen. Alles in allem ist Dragontorc ein Programm der Mittelklasse, was auch für den Schwierigkeitsgrad gilt.

System: CPC 464/664
Hersteller: Hewson Consultants

Rolf Knorre

VERGESSEN SIE ALLES, WAS SIE BISHER KENNEN ! JETZT GIBT ES RH-DAT

RH-DAT ist die Lösung Ihrer Probleme!

RH-DAT ist ein Dateiverwaltungsprogramm mit FREIER Maskendefinition (max. 21 Felder / max. 70 Zeichen pro Feld)
RH-DAT hält Ihre Daten im RELATIVEN Zugriff
RH-DAT kann nach jedem beliebigen Datenfeld suchen
RH-DAT findet einen Datensatz bei der Suche im Indexfeld selbst bei voller Datei (503/1003 Datensätze) in max. 4 Sekunden (Regelfall 1-2 Sekunden)
RH-DAT ist ein CP/M-Programm

Sie werden staunen, was RH-DAT noch alles kann !!!

RH-DAT ist erhältlich für den CPC 464, 664, 6128, 8256 auf 3"- und 5 1/4"-Disk

PREIS **DAS GIBT'S DOCH NICHT!** 503 Datensätze 79,- DM
1003 Datensätze 89,- DM

RH-BÜRO

Textverarbeitung, Adressenverwaltung und Mail-Merge
Testbericht in
CPC August 85, S. 17
NUR Disk 3"

5 1/4" (VORTEX)

Preis 99,- DM

TEL. BESTELLUNG
02 11/50 65-213

KOMPLETT-PAKETE		
1. RH - ADW	(Komfortable Adressverwaltung)	40,- DM
2. RH - TEXT	(Super-Textverarbeitungsprogramm)	40,- DM
3. RH - SYNTHESIZER	(Super-Synthesizerprogramm)	40,- DM
4. RH - VOKABULA	(Ausgerichtetes Vokabel-Lernprogramm)	30,- DM
5. RH - GRAPH	(Klasse Funktionsplotter)	30,- DM
		180,- DM

Jetzt als KOMPLETT-PAKET 50,- DM

Oder ein anderes Beispiel:

1. Drucker-Kabel	zum Anschluß eines Druckers mit CENTRONICS-Schnittstelle	58,- DM
2. Wie Komplet-Paket '5 für 1'		180,- DM
		238,- DM

Jetzt als KOMPLETT-PAKET 75,- DM



INTEGRAL HYDRAULIK

INTEGRAL HYDRAULIK & CO.

z. Hd. Herrn Hilchner, Am Hochhofen 108, 4000 Düsseldorf 11,
Tel. 02 11/50 65-213
Vertrieb von: H-Software

Death Pit

Die Aufgabeneinstellung bei Death Pit klingt einfach: Soviel Gold wie möglich sammeln und zum Ausgangspunkt bringen. Was aber wäre ein Actionspiel, wo so etwas gefahrlos möglich wäre.

In den etwa 80 Screens dieses Programms warten dann auch einige Überraschungen auf den Spieler, wenn er es wagt, die unterirdischen Höhlen zu betreten. Angelegt sind diese Höhlen und Gänge wie ein großes Labyrinth, aus dem man nur schwer wieder entkommen kann. Dazu kommt dann allerlei bisiges Getier bis hin zum ausgewachsenen Drachen, der einem das Leben besonders schwer machen kann. Obwohl das schon genug wäre, geht es noch weiter. Ein Zeitlimit ist zu beachten und der Vorrat an Energie und Sauerstoff muß von Zeit zu Zeit erneuert werden.

Für die Aufgabe stehen 5 Leben zur Verfügung, was in Anbetracht der vielen Hindernisse sicher nicht zuviel ist. Obwohl Death Pit nichts wirklich Neues bietet, ist das Programm dank der guten Grafik durchaus zu empfehlen.

System: CPC 464
Hersteller: Durell Software

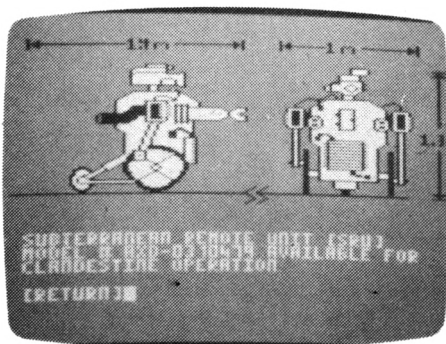
Rolf Knorre

Hacker

Welchen Computerfreak würde es nicht reizen, in fremden Datennetzen herumzustöbern. Leider sind aber die großen Computer in der Regel nicht an das Telefonnetz angeschlossen. Auch stellt uns die Post ein weiteres Hindernis in den Weg: die Telefonkosten. Selbst der Anruf einer seriösen, guten Mailbox kann Unsummen verschlingen, wenn der Husumer Hacker mit seinesgleichen in München Kontakt aufnehmen will.

Mit dem Spiel Hacker gibt es da allerdings keine Probleme mehr. Nach dem Einladen in den Computer werden wir ohne weitere Komplikationen zum richtigen Hacker. Die Anleitung ist denkbar kurz, denn sie enthält nur die Ladeanweisung. Alles weitere überläßt man dem Spieler, der nun alles ausprobieren und mit viel Phantasie herausfinden muß.

Bei Spielbeginn ist schon eine Verbindung hergestellt und wir müssen mit dem Computer weiter kommunizieren. Ganz nach unseren Vorstellungen wird natürlich ein Passwort verlangt und jeder wird bald auf das Wort "Hacker" kommen. Dann merken wir schon, daß wir auf der richtigen Spur sind, denn das System meldet, daß das Passwort geändert wurde. So versuchen wir es mit dem Namen unserer Freundin und erwartungsgemäß wird das nicht akzeptiert. Der angerufene Computer beginnt, die Verbindung abzubrechen.



Kurze Zeit vorher spielt das System verrückt und plötzlich befinden wir uns im Computer eines Elektrokonzerns. Wir erhalten die Kontrolle über einen kleinen Roboter, der sich in einem geheimen Tunnelsystem unter den Kontinenten bewegen kann. Mit Hilfe dieses hochmodernen Gerätes können wir nun den Weltfrieden retten, indem wir von Kontaktleuten in aller Welt Teile eines Geheimdokuments sammeln. Dabei erwarten den Spieler einige Überraschungen und Schwierigkeiten, bis er von der ganzen Welt als Retter umjubelt wird.

Das Spiel hat nicht nur eine sehr originelle Handlung, sondern erfreut den Spieler auch durch eine hervorragende grafische Umsetzung. Es hebt sich wohlthuend von anderen recht eintönigen Programmen auf dem Markt ab und kann jedem Spielefreak empfohlen werden.

System: CPC 464/664
 Hersteller: Activision
 Preis: ca. 44.- DM
 Thomas Tai

Robin of Sherwood

Haben Sie schon einmal etwas von dem "Rächer der Enterbten" gehört? Richtig, das ist Robin Hood. Jener tollkühne Angelsachse, der sich unentgeltlich in zahllosen Filmen und Büchern in den Kampf gegen die normannische Herrschaft stürzt. So einen Mann brauchte England, denn der gerechte König Richard widmete sich seinen Kreuzzügen und der herrschende Sheriff von Nottingham sah keine Veranlassung für eine Rebellion gegen die Normannen.

Der Spieler schlüpft nun in die Rolle von Robin Hood, der die angelsächsische Rebellion leiten und natürlich zum Sieg führen soll. So stolz man auf diese Wahl auch sein kann, zu Beginn des Spieles sieht es doch recht finster mit unserer Rebellion aus, denn der Held vegetiert im Kerker eines Sir Guy of Gisburne dahin. Jeder Adventureläie wird wochenlang an diesem Punkt festsitzen, denn wie entkommt man aus einem Verlies, das nur ein von einem unbestechlichen Wächter bewachtes Gitter als Ausgang hat. Als kleiner Tip sei hier erwähnt, daß man mit seinen Mitgefangenen im wahrsten Sinne des Wortes etwas "erreichen" kann. Außerdem sollte jeder das EXAMINE-Kommando möglichst oft benutzen, um wirklich die kleinste Einzelheit zu erfassen. Der total gefrustete Spieler kann auch in der Anleitung eine verschlüsselte Hilfe zu seiner Rettung finden.

Natürlich ist mit dem Entkommen aus dem Kerker auch eine Flucht durch das Schloß in den Wald verbunden und hier wird man dann auf den legendären Herne the Hunter treffen. Er ist der Herr des Waldes und gibt unserem Robin den Befehl, sechs Edelsteine zu sammeln und sie an ihren Ursprungsort zurückzubringen. Ist dies geschehen, hat er die magische Kraft, um die Rebellion zu leiten.

Man sieht, die Handlung ist außerordentlich spannend und wie von ADVENTURE INTERNATIONAL gewohnt,

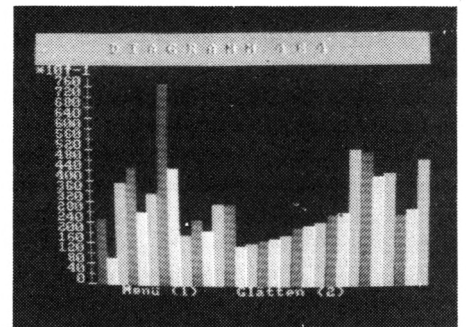
wurde das Ganze noch mit hervorragenden Grafiken unterlegt, welche die Motivation des Spielers um einiges erhöhen. Insgesamt kann man von einem rundherum gelungenen Adventure sprechen, das sowohl durch seine ausgefeilten Bilder, als auch durch eine spannende Handlung besticht. Es wird allerdings einige schlaflose Nächte kosten, bis der böse Sheriff von Nottingham besiegt ist.

System: CPC 464/664
 Hersteller: Adventure International
 Preis: ca. 49.- DM
 Thomas Tai

Diagramm 464

In unserer heutigen Zeit ist es fast unvorstellbar, ohne die grafische Darstellung von Zahlenwerten auszukommen. Egal welche Fernsehsendung aus dem Informationsbereich man anschaut, fast immer werden statistische Erhebungen, Vergleiche, Bilanzen etc. durch Grafiken dargestellt. Schnell hat man erkannt, daß die bildliche Umsetzung solcher Erhebungen ihre Bedeutung doch erst übersichtlich und interpretierbar macht. Eine Bundestagswahl ohne Torten- und Balkendiagramme ist heute nicht mehr denkbar. Aber auch an vielen anderen Stellen sind Diagramme von großer Bedeutung.

Die Firma Dynamics, die durch ihre preisgünstigen Softwarereihen für den



VC 20 und C 64 bekannt wurde, bietet jetzt auch allen Anwendern des Schneiders CPC 464 ein Programm mit dem Namen "Diagramm 464" an. Der Anwender muß hier nur seine Zahlenwerte oder Daten eingeben und schon können diese auf dem Bildschirm grafisch dargestellt werden.

Nach der Dateneingabe kann der Benutzer zwischen Balkendiagrammen, Strichdiagrammen und Funktionskurven wählen. Die Anzahl der Werte sowie der Anfangswert und die Position der X-Achse sind frei einstellbar. Auch die Bildung einer Mittelreihe (Nivellierung)

kann vorgenommen werden. Der Computer erkennt hier sog. "Ausreißer" und beseitigt diese. Hat man seine Diagramme erstellt, lassen sich die Werte auch abspeichern und jederzeit wieder einladen.

Leider gibt es keine Möglichkeit, diese Grafiken vernünftig auszudrucken, um diese dann einem breiteren Personenkreis zu zeigen. Auch die Darstellung am Monitor könnte noch ein wenig hochauflösender und mit einem größeren Farbspektrum versehen werden. Gleichzeitig wäre die Anpassung an die übrigen Schneider-Computer sinnvoll. Sobald uns detaillierte Informationen der neuen Version vorliegen, werden wir darüber berichten.

Diagramm 464 gehört zu der neuen, preisgünstigen Softwarereihe für Schneider-Computer, die aus diversen Archivprogrammen, dem Programm "Graphik Utilities 464" und einem Vokabeltrainer besteht. Der Preis für Diagramm 464 (Kassette) beträgt ca. 69.-DM. Die Dynamics-Programme sollen ab sofort bei vielen Computergeschäften und den Fachabteilungen einiger Warenhäuser erhältlich sein.

Markus Pisters

TeleTerminal 300 S

Spätestens seit dem großen Kinoerfolg "Wargames" träumen viele Computerbesitzer von der eigenen Hakerkarriere, zumindest aber von Spieleserien in Datenbanken und Mailboxen. CPC-Besitzer machen da keine Ausnahme. Das müssen sie auch nicht, da es die notwendigen Utensilien inzwischen auch für die Schneider-Computer gibt.



Getestet haben wir das Programm TeleTerminal 300 S, das schon bei anderen Computertypen erfolgreich eingesetzt

wurde. Benötigt wird ein Akustik-Koppler, der allerdings computerunabhängig ist. Im Test haben wir den AK 300 verwendet, der hervorragende Eigenschaften besitzt.

Nun aber zur Software, ohne die nichts läuft. Geliefert wird das Programm in einer Videokunststoffhülle, in der die Programmkassette, ein Verbindungskabel und eine deutsche Anleitung enthalten ist. Das Kabel dient der Verbindung zwischen Computer und Akustik-Koppler, der Anschluß erfolgt am Druckerport des CPC 464. Die Software – ein kurzes Maschinenprogramm – startet automatisch. Dem stolzen Besitzer steht nun im wahrsten Sinne des Wortes die Welt offen, da es überall Mailboxen gibt, in denen man herumstöbern kann. Das ist lediglich eine Frage der Telefonrechnung.

Folgende Optionen bietet TeleTerminal 300 S:

APPEND	Lädt ein Textfile von Band oder Diskette (auch die Vortex-Laufwerke werden unterstützt).
CAT	Katalog der auf Diskette/Kassette gespeicherten Files.
DOWNLOAD	Empfangene Daten werden gespeichert.
FARBE	4 mögliche Darstellungsarten.
INCREMENT	Kurzspeicher wird erhöht.
CHANGE	Der nächste Kurzspeicher wird eingeschaltet.
UPLOAD	Kurzspeicherinhalt wird gesendet.
LET	Der Kurzspeicherinhalt wird abgerufen und kann geändert werden.
MODUS	Umschalten von Halb- auf Vollduplexbetrieb.
PARITY	Parität umschalten.
READ	Der Textspeicher kann auf den Bildschirm gebracht und gelesen werden.
SAVE	Der Textspeicherinhalt kann auf Kassette/Diskette gespeichert werden.
WRITE	Erfassen von neuen Textfiles.
CLEAR	Löscht den gesamten Speicher.
COPY	Druckt den Textspeicherinhalt.

Das System wurde vom Programmierer auf FULL DUPLEX 8z1 eingestellt,

also auf Vollduplexbetrieb mit 8 Bit ohne Paritätsprüfung, 1 Stopbit. Mit dieser Einstellung kommt man in den meisten Fällen klar, die Änderung der Parameter ist aber leicht vorzunehmen.

Gut gelöst wurde auch die Option der Kurzspeicher. Hier können Userkennungen oder Paßwörter abgelegt werden, die man dann im Bedarfsfall auf Tastendruck senden kann. Im DOWNLOAD-Modus werden alle auf dem Bildschirm stehenden Daten im CPC gespeichert. Dadurch kann man auch längere Übertragungen empfangen, diese aber mit Rücksicht auf die Telefonrechnung erst nach beendeter Datenübertragung lesen oder ausdrucken.

TeleTerminal 300 S ist ein rundherum komfortables Programm, das kaum Wünsche offen läßt. Während des Tests sind keinerlei Fehler oder Mängel aufgetreten (bis auf eine fehlerhafte Übertragung aus Nordamerika, die aber auf die schlechte Verbindung zurückzuführen war.) Da man mit TeleTerminal auf eine zusätzliche serielle Schnittstelle (RS 232) verzichten kann und ein Komplettpaket erwirbt, ist auch der Preis von rund DM 100.- angemessen.

Rolf Knorre

Der ZEN-Assembler

Während man in den ersten Monaten nach dem Erscheinen des CPC Assembler/Disassembler noch mit der Lupe suchen mußte, werden diese jetzt von vielen Softwarehäusern angeboten. So auch das englische Softwarehaus KUMA, das den ZEN Assembler und Disassembler jetzt auf dem deutschen Markt anbietet. Auch hier handelt es sich um ein Programm, in dem Assembler und Disassembler fest miteinander verbunden sind. Auch andere Hersteller haben dieses Prinzip gewählt, z. B. ZAPP von HEWSON CONSULTANTS und DEVPAC von SCHNEIDER.

Beim DEVPAC bestehen Assembler und Disassembler allerdings aus separaten Programmen, welche man an eine beliebige (!) Stelle in den Speicher laden kann und sich so beide Programme gleichzeitig im Speicher befinden. Natürlich kann man dann von einem Programmteil durch Tastendruck in den anderen springen. Dieses Prinzip ist sehr benutzerfreundlich, da man sich seine Kombination selbst aussuchen kann. ZEN bietet diese Möglichkeit jedoch nicht.

Ansonsten hat ZEN eigentlich all die Features, Funktionen und Optionen, die man von anderen Assemblern ebenfalls kennt, ist aber im Gegensatz dazu extrem anwenderfreundlich.

Dem Programm liegt eine 100seitige Anleitung in Englisch bei, wobei die eigentliche Anleitung nur aus 10 Seiten besteht, der Rest ist ein Assemblerlisting von ZEN. Zudem ist die Druckqualität äußerst schlecht. Ohne gute Englischkenntnisse kann man mit diesem Programm praktisch nichts anfangen.

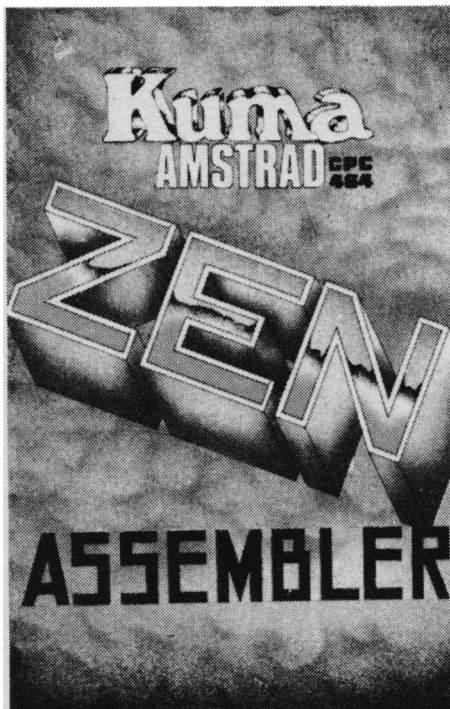
Auch das Laden ist so eine Sache. Das Programm wird nicht mit "RUN" geladen, sondern mit:

```
MEMORY 16383
LOAD "ZEN", 16384
CALL 16384
```

Ein kleines Vorprogramm hätte hier Abhilfe geschaffen.

Leider steht in der Anleitung nichts darüber, wie man das Programm auf Diskette konvertiert, deshalb hier für Diskettensitzer eine Anleitung:

```
TAPE.IN
MEMORY 16383
LOAD "ZEN", 16384
SAVE "ZEN", b, 16384, 6461
```



Ansonsten ist das Programm vollständig diskompatibel, d. h., alle Bereiche wirken auf das Diskettenlaufwerk in der gleichen Form, wie sie das mit dem Kassettenlaufwerk tun würden.

Nach dem Start meldet sich das Programm mit ZEN>. Insgesamt kann man dann 28 Funktionen anwählen. Leider existiert keine Helppage, so daß die Anleitung immer neben dem Computer liegen muß.

Der Disassembler ist ein Minimaldisassembler. Hier wurde nur das Notwendigste programmiert. Zudem ist er nicht

allzu schnell. Auch der Assembler ist ziemlich langsam. Die Eingabe und Editierung eines Textes ist hier so ziemlich das Umständlichste, was mir je unter die Augen gekommen ist. Auch das Listen und Löschen von Zeilen ist nicht so einfach und problemlos, wie es eigentlich sein sollte. Allerdings beherrschte der Assembler immerhin die Grundrechenarten und hat alle üblichen Pseudobefehle: ORG, END, DEFS, DEFB, EQU. Der Befehl DEFW fehlt allerdings.

ZEN ist ein Programm zum Arbeiten, allerdings sollte man eine sicher nicht zu kurze Einarbeitungszeit einplanen. Im

Prinzip ist der Assembler jedoch recht gut, nur beim Disassembler müßten noch zusätzliche Funktionen eingebaut werden. Der Assembler ist allerdings alles andere als komfortabel. Notwendig wäre noch ein vernünftiger Editor, ein Vorprogramm, eine Helppage und einige andere Punkte. Es schadet absolut nichts, wenn das Programm dadurch ein paar Kilobyte länger wird.

Hersteller: Avalon Software
 Vertrieb: Kuma
 Preis: ca. 40.- DM
 Andreas Zallmann

Die einzelnen Funktionen von ZEN

ASSEMBLE:	Das Textfile wird in ein MC-Programm übersetzt.
BYE:	Rückkehr zum Basic.
COPY:	Kopiert einen Speicherblock an eine andere Stelle.
DOWN:	Der Zeiger des Textfiles wird um eine angegebene Zahl nach unten verlagert.
ENTER:	Hier kann man das Textfile eingeben, welches assembliert werden soll.
FILL:	Füllen eines Speicherbereichs mit einer Zahl.
GOTO:	Aufruf eines Maschinenprogrammes.
HOWBIG:	Start und Endadresse des Textfiles wird ausgegeben.
IN:	Dieses Kommando zeigt eingelesene Daten (von einem angegebenen I/O Port) auf dem Bildschirm an.
KILL:	Das Textfile wird gelöscht.
LOCATE:	Im Textfile wird nach einem angegebenen String gesucht.
MODIFY:	Speicherbereiche können verändert werden.
NEW:	Damit können Zeilen des Textfiles editiert werden.
OUT:	Dieses Kommando sendet Daten an einen I/O Port.
PRINT:	Damit können Teile des Textfiles gelistet werden.
QUERY:	Speicherbereiche werden in HEX und in ASCII ausgegeben.
READ:	Liest ein Textfile von Kassette.
SORT:	Die Symboltabelle wird sortiert und ausgegeben.
TARGET:	Setzt den Zeiger des Textfiles auf den angegebenen Wert.
UP:	Der Zeiger des Textfiles wird um die angegebene Zahl nach oben verlagert.
WRITE:	Das Textfile oder ein Teil des Speichers wird auf Kassette geschrieben.
EXAMINE:	Die Werte der Register werden ausgegeben.
ZAP:	Teilbereiche des Textfiles können gelöscht werden.
CATALOG:	Das Band kann inspiziert werden.
DISASSEMBLE:	Ein Speicherbereich wird disassembliert.
UNSCRAMBLE:	Acht Bytes werden disassembliert. ZEN bemüht sich herauszufinden, ob es sich um Daten oder Programmbytes handelt.

Textverarbeitung mit CPC-Text/Adress

Ein brauchbares Textprogramm der unteren Preisklasse

Das Programm CPC-Text/Adress von Markt + Technik besteht aus zwei getrennten Programmen: Einem Textverarbeitungsprogramm und einer Adressverwaltung. Beide Programme sind gegenüber der von einem halben Jahr vertriebenen Version (V1.0) überarbeitet und erweitert worden (Version V1.1). CPC-Text/Adress ist sowohl als Disketten- als auch als Kassettenversion lieferbar (getestet wurde die Diskettenversion, Unterschiede zur Kassettenversion sind uns nicht bekannt).

Zur Programmdiskette wird ein 16 Seiten starkes Heftchen als Bedienungsanleitung mitgeliefert. Auf jeweils 7 Seiten wird die Bedienung für CPC-Text und CPC-Adress abgehandelt.

CPC-Text

Nach dem Laden erscheint das 10 Punkte große Hauptmenü. Es bildet das Rückgrat des sehr bedienungsfreundlichen Textverarbeitungsprogramms. Von hier aus gelangt man in weitere Untermenüs zum Erstellen aller wesentlichen Parameter für die Bearbeitung oder Ausgabe eines Textes. Es lassen sich von hier aus Texte laden (1), speichern (2), drucken (3) oder löschen (8). Sehr viel interessanter sind jedoch die Submenüs für die Bestimmung der Text- (5) und Druckerparameter (4).

Unter Textparameter versteht das Programm das Setzen der Tabulatoren und des rechten Randes (Zeilenbreite) sowie die Anzahl der Zeilen, die auf eine Druckseite passen sollen. Hier zeigen sich auch schon einige Grenzen des Programms. Die Zeilenbreite ist nur innerhalb des Bereichs 30 bis 79 Zeichen pro Zeile einstellbar. Das Bearbeiten breiterer Tabellen ist somit nicht möglich, doch meistens reicht eine Breite von 79 Zeichen aus.

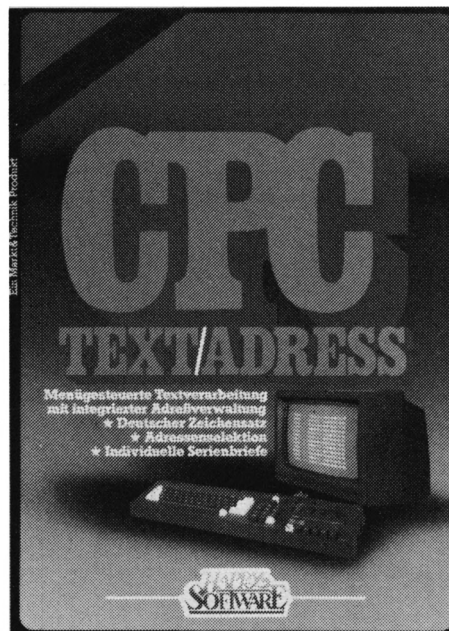
Was nützt ein Textverarbeitungsprogramm, wenn die Texte nicht optimal auf dem benutzten Drucker gedruckt werden können? Dieses Anpassen an den Drucker ist bei fast allen Textprogrammen schwierig. Problem Nummer 1 sind die deutschen Sonderzeichen, die bei vielen Druckern auf den exotischsten ASCII-Codes liegen. Im Submenü zu den Druckerparametern kann der Benutzer alle deutschen Sonderzeichen (leider fehlt das Paragraphenzeichen) auf beliebige ASCII-Codes legen und somit an den benutzten Drucker anpassen. Voreingestellt sind die Parameter für einen

Epson-kompatiblen Drucker, also auch für den NLQ 401 (sehr gut).

Problem Nummer 2 ist die Anpassung der Druckersonderfunktionen wie Fettdruck, Unterstreichen, etc. Auch hierfür wurde eine gute Lösung gefunden. Voreingestellt sind wieder 11 Epson-kompatible Funktionen. Aufgerufen werden sie später in Kombination der CONTROL-Taste mit einer anderen Taste der oberen Tastaturreihe (Taste 1 bis -). Diese 11 Steuerzeichen lassen sich problemlos mit neuen Funktionen belegen. Eine Anpassung an einen beliebigen Drucker mit einer Centronics-Schnittstelle steht nichts im Wege.

Aus dem Hauptmenü gelangt man natürlich auch in das eigentliche Textprogramm. Es zeigt sich ein gewohntes Bild: Die oberste Bildschirmzeile bildet die Statuszeile mit den Angaben, wo sich der Cursor momentan befindet (Seite, Zeile und Spalte). Darunter eine Art Zeilenlineal, das die Stellung der Tabulatoren kennzeichnet. Die restlichen 22 Bildschirmzeilen stehen für die Texteingabe zur Verfügung.

Zum Editieren des Textes sind diverse Funktionen vorhanden, die im wesentli-



chen über die 3-fach belegten Cursorarten ausgeführt werden. So ist es u. a. möglich, zum Zeilenanfang/Zeilenende oder zu einem Wort nach rechts oder links zu springen oder den Halbsatz links oder rechts vom Cursor zu löschen. Die Bewegungen des Cursors und das

Bildschirmscrolling sind für den Schneider recht schnell.

Störend wirkt sich die Funktion der DEL-Taste aus: Zwar wird das Zeichen links vom Cursor gelöscht, aber nur in der aktuellen Bildschirmzeile. Am Anfang der Zeile ist Ende. Ebenso fällt auf, daß die Zeichenausgabe auf dem Bildschirm oft hinter der Eingabe hinterherhinkt.

Erwähnt sei auch der Blocksatz, der automatische Trennvorschläge macht. Er arbeitet einwandfrei, aber sehr, sehr langsam.

CPC-Text ist für ca. 89,- DM ein brauchbares Textprogramm der unteren Preisklasse, welches durch seine Menüführung leicht und zuverlässig zu bedienen ist. Auch wer nur selten mit dem Programm arbeitet, findet immer wieder den problemlosen Einstieg. Im Gegensatz zur alten Version 1.0 ist es erheblich verbessert worden. Trotzdem führt z. B. eine fehlerhafte Eingabe beim Laden oder Sichern eines Textes zum Programmabsturz und Datenverlust. Auffällig ist auch das extrem langsame Ausdrucken eines Textes auf dem Drucker. Trotzdem ist CPC-Text weitaus besser als viele seiner Konkurrenten in dieser Preisklasse. Für den professionellen Einsatz ist es allerdings nicht oder nur sehr bedingt geeignet.

CPC-Adress

Als Beigabe befindet sich auf der Diskette noch das einfache Adressverwaltungsprogramm CPC-Adress. Innerhalb einer festen Maske lassen sich 100 Adressen verwalten. Die Maske ist etwas unschön und sehr störanfällig, zudem noch im 40-Zeichen-Modus. Ist ein Feldeintrag z. B. größer als 40 Zeichen, zerstört dieser Eintrag die Maske! (Warum nicht im 80-Zeichen-Modus, der CPC kann es doch?) Neben dem Eingeben, Löschen und Ändern von Einträgen lassen sich auch Adressaufkleber drucken. Die sehr vielseitigen Suchfunktionen erlauben auch das verkettete Suchen: So lassen sich problemlos alle Meyers selektieren, die im Postbereich 2000 wohnen.

CPC-Adress ist eine lobenswerte Ergänzung zum Textverarbeitungsprogramm CPC-Text und für den Hausbedarf ausreichend. Es allerdings als professionelle Adressverwaltung zu bezeichnen, wäre vermessen.

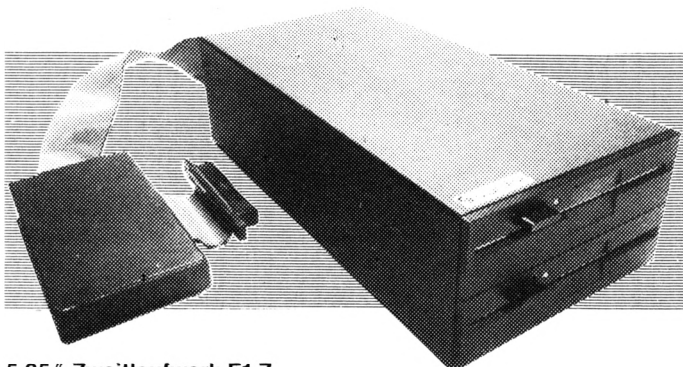
Manfred Walter Thoma

Speichern Sie wohl...

für den Schneider CPC 464 + 664*

5.25" 1,4 MB CP/M 2.2 VDOS 2.0

...neue Vortex-Diskstation F1
jetzt auch für CPC 664 und 6128.



5,25"-Zweitlaufwerk F1-Z

Das Laufwerk F1-Z kann als Zweitlaufwerk an die Schneider 3"-Diskettenstation DDI-1 angeschlossen werden und hat dieselbe Speicherkapazität wie das 3"-Laufwerk. Es ist identisch mit der Station F1-S jedoch ohne Controller und ohne CP/M.

Das mitgelieferte Programm SPARA erlaubt Ihnen das Lesen und Beschreiben von Disketten gängiger CP/M-Systeme, welche Ihre Disketten einseitig mit 40 Spuren verwalten. (Diese Einschränkung bedingt der Controller der Schneider DDI-1.)

Erwacht jedoch Ihr Interesse an 708 KB oder 1,4 MB, so können Sie Ihre F1-Z problemlos zur F1-S oder F1-D aufrüsten.

vortex Floppy-Disk-Station F1

jetzt mit VDOS 2.0: relative Dateien, Tracer, Disassembler, Assembler, BASIC-Befehlsweiterungen

Leistungen

- Ein (wahlweise zwei; von Anfang an, oder nachrüstbar) 5,25"-Slimline, 80 Track, DS/DD 6138 BASF-Laufwerk der modernsten Technologie mit 708 KB (1,4 MB), formatierter Speicherkapazität, 4 msec. Steprate, IBM 34-Formate.
- CP/M 2.2-Betriebssystem und Systemutilities
- Erweitertes BASIC-stand-alone-Diskettenbetriebssystem VDOS 2.0
- Ohne Soft- oder Hardwareänderungen kann ein Schneider 3"-Laufwerk über ein Adapterkabel angeschlossen werden. Softwarekonvertierung von 5,25" auf 3" und umgekehrt: kein Problem.

Preise

- F1/S Floppy-Disk-Station mit Controller und Laufwerk inkl. CP/M 2.2, VDOS und Handbuch
1198,— DM (unverbindliche Preisempfehlung)
- F1/D Floppy-Disk-Station mit Controller und zwei Laufwerken inkl. CP/M 2.2, VDOS und Handbuch
1698,— DM (unverbindliche Preisempfehlung)
- A1-S Aufrüstkit bestehend aus BASF-Laufwerk 6138 und Einbauleitg. **500,— DM (unverbindliche Preisempfehlung)**
- 5,25"-Zweitlaufwerk F1-Z + Programm SPARA
698,— DM (unverbindliche Preisempfehlung)
- Aufrüstkit A1-Z bestehend aus Controller, CP/M-Lizenz und Dienstprogrammen sowie Handbuch
548,— DM (unverbindliche Preisempfehlung)

64 KB bis 512 KB RAM-Erweiterung ... Druckerpuffer ... RAM-Floppy

vortex RAM-Erweiterung SP64 ...

— jetzt endlich läuft jedes Standard-CP/M-Programm

- (z. B. Wordstar, dBase, Multiplan)
- voll unter BASIC und CP/M einsetzbar
- das Betriebssystem der Karte (im ROM) ist nahtlos ins CPC-Betriebssystem eingebaut
- einfacher Einbau der Karte: kein Löten

SP 64, 64-KB-RAM	275.— DM (unverbindliche Preisempfehlung)
SP 128, 128-KB-RAM	348.— DM (unverbindliche Preisempfehlung)
SP 256, 256-KB-RAM	478.— DM (unverbindliche Preisempfehlung)
SP 320, 320-KB-RAM	528.— DM (unverbindliche Preisempfehlung)
SP 512, 512-KB-RAM	589.— DM (unverbindliche Preisempfehlung)

Sie erhalten unsere Produkte:

- in allen Karstadt-, Horten- und Quelle-Computercentern
- in den technischen Kaufhäusern Phora und Brinkmann

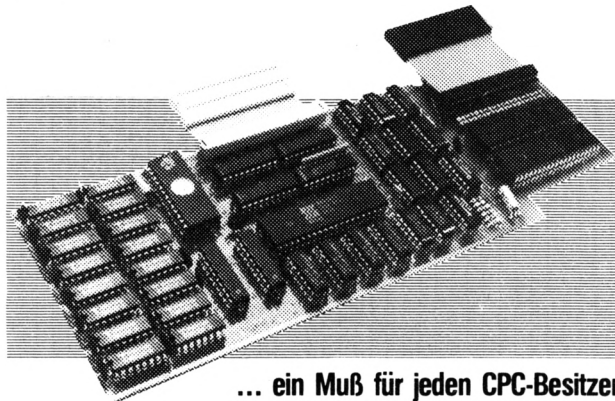
Bei Bezugsproblemen rufen Sie uns bitte an.

Fordern Sie unser kostenloses Informationsmaterial an.

* Speicherkarte für den 664 beinhaltet keine Basic-Erweiterung. Diese ist erst zu einem späteren Zeitpunkt als Nachrüstsatz (Eprom) zu beziehen.

Mit jedem unserer Produkte erhalten Sie den **vortex Service-Paß**. Mit diesem Paß garantieren wir Ihnen einen kostenfreien Anspruch auf alle Neuerungen und eventuelle Verbesserungen unserer Betriebssystemsoftware. Für soft- und hardwaretechnische Fragen im Zusammenhang mit unseren Produkten haben wir eine **User-Sprechstunde** eingerichtet. Montags und Donnerstags von 18 Uhr - 21 Uhr stehen wir Ihnen telefonisch zur Verfügung.

CP/M 2.2 ist ein eingetragenes Warenzeichen der Firma Digital Research · VDOS und vortex sind eingetragene Warenzeichen der Firma vortex GmbH · Wordstar ist ein eingetragenes Warenzeichen der Firma Micro Pro · dBase ist ein eingetragenes Warenzeichen der Firma Ashton Tate · Multiplan ist eingetragenes Warenzeichen der Firma Microsoft.



... ein Muß für jeden CPC-Besitzer

Die Programme **Wordstar**, **dBase** und **Multiplan** erhalten Sie zu sensationellen Preisen beim **M & T Software Verlag** oder bei **Vortex**. Alle Programme sind auf unsere RAM-Erweiterung abgestimmt und laufen mit ihr uneingeschränkt.

7106 Neuenstadt, Klingenberg 13 — Tel.: Abt. Marketing 07139/2160, Telex 728915 — Tel.: Abt. Software 0711/7775576

dBase II – Das Dateiverwaltungsprogramm schlechthin

Als ich mir vor ca. 3 Jahren meinen ersten Computer anschaffte (übrigens einen TI 99/4A), hatte ich von Bits und Bytes noch keine Ahnung. Er sollte mir als Videospieldienst dienen, aber auch dazu, ein bißchen in die Computerei hineinzuriechen.

Natürlich hörte ich auch von der Möglichkeit, mit dem Rechner Daten jeder Art zu verwalten. Und so kam ich auf den Gedanken, meine Schallplatten-sammlung, die aus ca. 500 Schallplatten besteht, zu katalogisieren, um diese Datei dann nach den verschiedensten Kriterien auswerten zu können. Etwa nach dem Motto: "Bei welchen Gruppen spielt der Musiker XY überall mit?" Also setzte ich mich mit kindlicher Naivität daran, ein solches Programm zu schreiben. Nach relativ kurzer Zeit schon war das Programm fertig, ich war aber meinem Ziel immer noch kein Stück näher gekommen. Wo lag der Haken? Ganz einfach: Erstens verfügte mein Rechner leider nur über eine Kapazität von reell nutzbaren 13 kByte für Basicprogramme. Aber selbst ein 64 kByte großer Hauptspeicher hätte keine Lösung meines Problems bedeutet. Und dies noch dazu mit einem Kassettenrekorder als Massenspeicher.

Kurze Zeit später stand die Anschaffung eines Diskettenlaufwerks ins Haus, wodurch ich wenigstens in der Lage war, durch Verwendung relativer Dateien die Menge der speicherbaren Daten im Direktzugriff drastisch zu vergrößern. Was jetzt folgte, kann ich nur als "großen Frust" bezeichnen, denn was mir vorschwebte, war so noch lange nicht durchführbar. Das zweite Manko war nämlich die auch hier sehr geringe Speicherkapazität. Total entnervt, legte ich also mein ehrgeiziges Projekt ad acta. Sauer war ich vor allen Dingen auf die Werbung der Computerhersteller, die dem unbedarften Anfänger suggeriert, schon mit dem Erwerb einer kleinen Grundausrüstung solche Dinge erreichen zu können. Dabei ist das einzige, was problemlos machbar wäre, eine kleine Adreßdatei oder ähnliches. Nur, dafür ist es bequemer und schneller, die Daten in einem Heftchen ständig parat zu haben, statt erst den Rechner einzuschalten, das Programm zu laden, die Datei zu laden, die Daten zu suchen,...

Dann kam der Schneider CPC 464 und kurze Zeit später das Vortex-Laufwerk mit seiner enormen Speicherdimension von 704 kByte pro Diskette. Rein theoretisch müßte es doch damit möglich sein? Aber woher bekommt man ein entsprechend leistungsfähiges Programm? Von der damals auf dem Markt befindlichen Software kam nur ein einziges in Frage, von dem wohl schon so ziemlich jeder den Namen gehört hat: dBase II! Das ist ein CP/M-Datenbanksystem, dessen Ruf ihm mindestens genauso weit vorausseilt, wie sein Preis es für die private

704 KByte pro Diskette

Anwendung praktisch unerreichbar machte. So war es jedenfalls damals. Inzwischen bietet der Verlag MARKT & TECHNIK diesen Star am Softwarehimmel in einer für den CPC speziell angepaßten Version für schlappe 200.- DM an. Da gab's für mich natürlich kein Zögern mehr. Diese Chance mußte ergriffen werden.

Was hat es nun mit diesem Mitglied des sagenumwobenen Triumvirates – Wordstar, Multiplan, dBase II – auf sich? Charakteristisch ist, daß es kein festgefügtes Schema gibt, sondern viel Wert auf äußerste Flexibilität gelegt wurde. Daraus ergibt sich, daß es sich bei dBase vielmehr um eine "Programmiersprache" handelt. Diese Sprache ist jedoch nicht so universell zu verwenden wie z. B. Basic. Ihr Ziel ist die Erstellung und Benutzung einer ganz auf die persönlichen Anforderungen zugeschnittenen Datei.

Eine Analogie zu Basic besteht allerdings, denn dBase ist ebenfalls eine Interpretersprache. Wer also ohne viel Lernaufwand schnell zum Erfolg kommen will, findet auch hierfür Möglichkeiten. Die dazu benötigten Befehle werden dann einfach im Direktmodus eingegeben, das heißt, nach Eingabe folgt sofort deren Ausführung.

Haben wir die Bedienungsanweisung auch nur recht oberflächlich gelesen, so ist uns bei der Lektüre doch sicher der Befehl CREATE aufgefallen. Selbst wer nur über Englischkenntnisse aus der Schulzeit verfügt, wird sofort ganz richtig schlußfolgern, daß es sich hierbei um den eigentlichen "Schöpfungsakt" handelt. Na, denn man zu!

Nach der erfragten Namensgebung für das File, das daraufhin auf Diskette eröffnet wird (natürlich hieß es bei mir "Platten"), erwartet der Rechner nun die Vorgabe der Struktur unserer Datei. Jeder Datensatz (quasi Karteikarte) kann dabei aus bis zu 32 Feldern bestehen und insgesamt bis zu 1000 Zeichen umfassen. Das dürfte selbst für ausgefallensten Anwendungen genügen!

Die Befehle von dBase II

ACCEPT	LIST
APPEND	LIST FOR
APPEND FROM	LOCATE
BROWSE	LOOP
CHANGE	MEMORY
CLEAR	MODIFY
COMMAND	NOTE
CONTINUE	OFF
COPY	ON
COUNT	PACK
CREATE	PICTURE
DATE	PRINT
DEBUG	QUIT
DEFAULT	READ
DELETE	RECALL
DELIMITED	RECORD
DESCENDING	REINDEX
DISPLAY	RELEASE
DO	REMARK
DO WHILE	RENAME
ECHO	REPLACE
EDIT	REPORT FORM
EJECT	RESET
ELSE	RESTORE FROM
ENDCASE	RETURN
ENDDO	SAVE
ENDIF	SAY
ENDTEXT	SELECT PRIM.
EOF	SELECT SEC.
ERASE	SET
EXACT	SKIP
FIELD	SORT ON
FILE	STEP
FIND	STORE
GET	STR
GO	STRUCTURE
GOTO	SUM
GO BOTTOM	TALK
GO TOP	TEXT
HELP	TOTAL
IF	TRIM
INDEX	UPDATE
INPUT	USE
INSERT	VAL
JOIN	WAIT
LINKAGE	WAIT TO

Wie sieht nun ein solches Datenfeld aus? Natürlich benötigt es erst einmal einen Namen. Dieser darf zehn Zeichen

ungeahnte Möglichkeiten

groß sein. Dann muß festgelegt werden, welchem Typ das Feld angehören soll. Die Wahl des Typs C entspricht in etwa dem, was man in Basic mit dem Begriff "String" umschreibt, also eine Aneinanderkettung beliebiger Buchstaben, Ziffern oder Symbole. Zahlen erfaßt man besser mit N, denn solche numerischen Felder lassen sich mit Rechenvorschriften verknüpfen und erlauben somit arithmetische Operationen. Seltener wird wohl der letzte Typ, L, eingesetzt werden. Er steht für logische Ausdrücke, also für die beiden Alternativen "wahr" oder "falsch".

Dann folgt die Wahl der Größe der einzelnen Felder, wobei auf die oben genannte Beschränkung der Gesamtkapazität zu achten ist. Abschließend kann die Zahl der Nachkommastellen vorgeählt werden, was natürlich nur bei numerischen Feldern sinnvoll ist.

Sind nun diese Angaben für alle von uns verwendeten Felder abgeschlossen, dürfen wir sofort mit der eigentlichen Datenerfassung beginnen. Haben wir uns für die Verwendung einer Indexedatei entschieden, so legt dBase die eingegebenen Daten automatisch sortiert ab. Eine Vorentscheidung für die Ordnung erspart spätere Wartezeiten bei entsprechenden Sortiervorgängen. Das bedeutet freilich nicht, daß wir uns für alle Ewigkeit auf diese Reihenfolge festlegen. Im Gegenteil, mit Hilfe anderer Indexedateien können wir unsere Daten beliebig umstrukturieren. Sogar spätere Änderungen und Ergänzungen im einmal gewählten Dateiaufbau sind kinderleicht machbar.

Da dBase II mit relativer Dateiverwaltung arbeitet, hängt die Gesamtkapazität praktisch nur von der verwendeten Diskettengröße ab, so daß wir zum Beispiel mit einem Vortex-Laufwerk bis zu 704 kByte Daten archivieren können. Um dieses gigantische Reservoir zu füllen, bedarf es schon einer unermüdlchen Arbeitswut! Auf eine normale Adreßdatei bezogen, bedeutet dies, daß uns im direkten Zugriff mindestens 3000 bis 5000 Adressen zur Verfügung stehen. Selbst bei meiner doch recht umfangreichen Anwendung bleibt noch Platz en masse übrig.

Anders sieht es leider bei 3"-Disketten aus. Ein Diskettenwechsel ist unter CP/M bekanntlich nicht so ohne weiteres möglich. Und dBase muß während der Arbeit ständig auf der Diskette vorliegen. Daraus ergibt sich, da dBase selbst schon 111 kByte Platz benötigt, ein Speicherplatz von nur noch 58 kByte. Für den einigermaßen sinnvollen Einsatz kommt man hier wohl kaum um die Anschaffung eines zweiten Laufwerkes herum.

Die Ausgabe und Auswertung der gespeicherten Daten ist genauso kinderleicht machbar, wie schon die Eingabe. Dazu dient der im Direktmodus zu gebende Befehl LIST. Er bewirkt die Auflistung aller Datensätze, wobei die auszugebenden Felder genau spezifiziert werden können. Auch eine Auswahl nach bestimmten Kriterien ist hier möglich: z. B. veranlaßt LIST FOR PLZ = "3000" .AND. "M" \$Name Name, Vorname, Straße, PLZ, Ort den Rechner, aus den vorhandenen Adressen diejenigen herauszusuchen, deren Postleitzahl 3000 lautet und deren Name mit dem Buchstaben M beginnt. Von diesen werden dann nur Name, Vorname, Straße, Postleitzahl und Ort gelistet, obwohl wir vielleicht noch weitere Felder, wie Bemerkungen, Kundennummern, Geburtstage oder ähnliches gespeichert haben. Ein denkbarer praktischer Einsatz hier-

für wäre beispielsweise der Druck von Adreßetiketten.

Um sämtliche Möglichkeiten von dBase II zu nutzen, ist es aber doch notwendig, eigene Commandfiles (Programme) zu erstellen. Dann nämlich sind alle denkbaren Auswertungen später durch einfachen Tastendruck möglich. Es gibt Befehle für Schleifenbildung, Abfrage von Bedingungen, Textausgabe, Formatierung von Zahlenwerten und viele mehr (siehe Kasten).

Bis zu 5000 Adressen

Es wäre müßig, sämtliche Anwendungen aufzuzählen, deshalb seien hier nur einige erwähnt: Lagerverwaltung mit automatischer Fakturierung, Rechnungsschreibung, Adreßkartei, Kundenstatistik, Umsatzanalyse, und, und, und... Mit ein bißchen Phantasie werden wohl jedem endlos viele Ideen kommen.

Nun noch ein paar Worte zur benötigten Gerätekonfiguration. Da dBase II ein CP/M Programm ist, brauchen wir zwingend ein Diskettenlaufwerk (natürlich auch aus anderen Gründen). Eine der vom Verlag MARKT & TECHNIK angebotenen Versionen ist für den neuen CPC 6128 konzipiert, die andere läuft nur auf einem CPC 464 mit der Speichererweiterung von Vortex. Mit dem kleinen Standard-CP/M ist leider absolut nichts anzufangen. Geliefert wird dBase II wahlweise im 3" oder 5.25" Format mit einer enorm umfangreichen Dokumentation. Wem dieses Material nicht ausreicht, der kann aus zahlreichen Literaturangeboten wählen.

Und noch ein Tip: Mit dem eingebauten Editor lassen sich dBase-Programme zwar schreiben, aber es mangelt ihm erheblich an Bedienungskomfort. Da die Programme aber auch über eine Textverarbeitung erstellt werden können, raten wir zu dieser Alternative.

Thomas Jacobi

G

Computerstore
GmbH

Hochstraße 11
8500 Nürnberg 80
☎ 09 11 / 28 90 28

SPIELE:

View to a kill	(K)	34.-
Bruce Lee	(K)	36.-
Dumbusters	(K)	36.-
One on One	(K)	36.-
Popeye	(K)	25.-
Archon	(K)	44.-
Beach Head	(K)	36.-
Centre Court	(K)	34.-
Cyrus-Schach	(D)	128.-
Pole Position	(K)	36.-
Exploding Fist	(K)	36.-
Superpipeline 2	(D)	44.-
Slapshot	(K)	25.-
Starion	(K)	29.-
Wizard's Lair	(K)	44.-
Dark Powers	(D)	59.-

Summer Games II (K) a. A.

ANWENDUNGEN:

Wordstar	(D)	199.-
dBase II	(D)	199.-
Multipan	(D)	199.-
Terminal Star	(D)	79.50
Supercalc II	(D)	198.-
Textomat plus	(D)	198.-
Basic-Compiler	(D)	69.-
Assemblerkurs	(K)	64.-
Tasprint	(K)	29.-

ZUBEHÖR:

MP 2	148.-
Joystick JY2	39.50
3"-Disketten	5 St. 59.-
	10 St. 110.-



Wir führen zu den original SCHNEIDER-Produkten Software, Bücher und Zubehör verschiedener Firmen wie DATA BECKER, VORTEX, CUMANA, ISS, RUSHWARE, MARKT & TECHNIK, SYBEX, VOGEL-Verlag usw.!

1 MB

Das DSB-51 System: 5 1/4 Zoll
Drive B mit 1 MB für CPC 464 u. 664

ab 659,-


2 MB

Das DSB-52 System: 2 x 5 1/4 Zoll
Drive mit je 1 MB für CPC 464 u. 664

ab 1398,-

3 Zoll 180 KB

399,-



5 1/4 Zoll 180 KB

599,-

Data-Service Bonn, 5300 Bonn, Kessenicherstr. 13, Tel.: 02 28 / 23 50 03



Das Komplettsystem Joyce PCW 8256

Der Joyce im Test

Amstrad/Schneider hat seinem Ruf als Hersteller von preiswerten und leistungsfähigen Computersystemen einmal mehr alle Ehre gemacht. Der JOYCE oder auch PCW 8256 ist ein leistungsfähiges Textverarbeitungssystem, das nicht allein wegen der Preislage einer guten Speicherschreibmaschine weit überlegen ist.

Die klugen Leute bei Amstrad wollten einen preiswerten, professionell einsetzbaren Computer, dessen Hauptaufgabe in der Textverarbeitung bestehen sollte. Um den Anforderungen gerecht zu werden, die man an ein solches Komplettsystem stellt, muß man sie erst einmal kennen. Was lag da näher, als einfach jemanden "vom Fach" zu befragen. Als Opfer wählte man sich die Sekretärin des Amstrad Chefs aus und die verriet den Entwicklern ihre Vorstellungen von einem guten Computer zum Büroeinsatz. Die Sekretärin hieß mit Vornamen JOYCE und so hatte der neue Computer auch gleich einen Namen.

Kommen wir zum System selbst: Für ca. 2500.- DM bekommt man einen großen Karton mit folgendem Inhalt: 1 Monitor mit eingebautem 3" Laufwerk, 1 Keyboard, 1 Drucker, 2 Systemdisketten und 2 umfangreiche Handbücher. Der Aufbau erweist sich als denkbar einfach: Tastatur und Drucker werden mit dem Monitor verbunden und wenn Strom fließt, kann es eigentlich schon losgehen.

Die gesamte Elektronik wurde im Monitorgehäuse integriert, womit ein relativ geringer Platzverbrauch gewährleistet ist. Diesem Umstand haben wir aber auch gleich einen sehr ärgerlichen Nachteil zu verdanken: die Tastatur erinnert verdächtig an den CPC 6128 und besitzt somit all seine Vor- und Nachteile. Der Zehnerblock wurde aus Platzgründen nicht von der übrigen Tastatur abgetrennt und bei den ersten Tippversuchen überkommen einen beklemmenden Engegefühl, weil die Tasten sehr dicht nebeneinander liegen. Nach einer kleinen Eingewöhnungszeit kann man das aber verschmerzen. Weniger schön ist der in die Zehnertastatur integrierte Cursorblock, wobei gleich zu erwähnen wäre, daß die Spezialtasten zur Textverarbeitung das gleiche Schicksal ereilt hat. Wer bei der Textverarbeitung mit vielen Zahlen hantiert und den Zehnerblock zur Eingabe nutzen will, muß diesen erst einmal durch ALT + FORM dafür definieren und verzichtet solange auf die Cursorstellung und die Funktionstasten zur Textverarbeitung. Die Mechanik der Tastatur ist gut. Sie vermittelt ein angenehmeres Schreibgefühl als beim CPC 464/664, ist aber gewöhnungsbedürftig.

Der Drucker vermittelt einen sehr schwächlichen Eindruck, doch davon darf man sich nicht abschrecken lassen, denn seine Bestimmung, nämlich das Bedrucken von Papier, erfüllt er sehr

gut. Maximal bringt er 90 Zeichen pro Sekunde zu Papier, doch bei Korrespondenzqualität sind es nur noch etwa 30 Zeichen pro Sekunde. Erfreulich ist der große Umfang des Zeichensatzes. So kann er griechische Buchstaben, acht internationale Zeichensätze sowie mathematische und andere Sonderzeichen zu Papier bringen. Als Matrixdrucker ist er natürlich auch grafikfähig und erlaubt Tabulatoren und Randeinstellung. Weiterhin verfügt er über einen halbautomatischen Einzelblatteinzug und einen aufsetzbaren Traktor für Endlospapier.

Da sämtliche Elektronik fehlt und man im Drucker nur die nackte Mechanik vorfindet, traut man ihm eigentlich nichts Außergewöhnliches zu, doch bei näherer Beschäftigung mit dem Gerät wird man seine Meinung schnell ändern. Die Schriftqualität ist sehr gut und alle Schriftarten sind gut lesbar. Besonders erwähnenswert erscheint die Möglichkeit, Proportionalchrift in Blocksatz auszudrucken. Der Drucker kann wirklich alles, was zu einer guten Textverarbeitung notwendig ist, und auch wenn der Geräuschpegel seikosaähnlich ist, muß man ihm das angesichts seiner anderen Merkmale verzeihen.

Die Monitoreinheit ist das wichtigste am ganzen System. Außer der ganzen Elektronik hat man dort auch das 3" Laufwerk eingebaut. Hält man einen gut berechneten Abstand vom Monitor, ist die Floppy gut erreichbar und bequem zu bedienen. Sie läuft fast vollkommen geräuschlos und bildet somit einen totalen Gegensatz zu dem schon erwähnten Drucker. Ohne Probleme können natürlich die für JOYCE vorgesehenen Disketten geladen werden. Darunter werden viele CP/M Plus Programme sein, wobei laut Bedienerhandbuch auch Disketten mit CP/M 2.2 gelesen werden können, soweit sie im Schneiderformat beschrieben wurden. Dies erwies sich bei unserem Test auch als richtig.

Bei der Installation von WORDSTAR für den CPC gab es allerdings einige Probleme, während TURBO PASCAL bald zur Mitarbeit überredet werden konnte. Bei beiden Programmen war es aber nicht möglich, Sicherheitskopien zu erstellen. Das DISCKIT antwortete dann einfach, daß es sich um ein altes Format handle. Ansonsten kann man mit dem Laufwerk voll zufrieden sein.

Wenn wir jetzt über den Monitor sprechen, müssen wir uns klar darüber sein, daß er einen Leckerbissen für alle "Textverarbeiter" darstellt. Mit seinen 90 Zeichen in jeder der 32(!) Zeilen kann JOYCE erheblich mehr, als die meisten anderen PCs. Die Schrift wird gut lesbar und völlig flimmerfrei dargestellt, so daß die bei anderen Computern mitgelieferten Augentropfen nicht nötig sind.

Die Software

Beginnen wir mit dem wohl wichtigsten Programm auf den Systemdisketten, der Textverarbeitung LOGOSCRIPT. Sie wurde speziell für JOYCE geschrieben und hier finden auch die Funktionstasten zur Textverarbeitung ihre Anwendung. Zuerst offenbart sich dem Anwender nach dem Booten eine Tabelle mit dem Disketteninhalt. Dabei stehen in jeder Spalte Texte gleichen Aufbaus. Für jede Spalte wird ein Textlayout erstellt, das automatisch eingeladen wird, wenn man beispielsweise wieder einen Brief schreiben will. Man hat dann gleich die Tabulatoren und Markierungen für Adresse und Absender richtig gesetzt und kann sofort loslegen. Für Leute, die ständig Texte gleichen Aufbaus schreiben, ist das eine nützliche Einrichtung. Die übrigen Anwender werden diese etwas umständliche Prozedur wohl erst verfluchen, sich aber dann nach einiger Zeit daran gewöhnen.

Im Zusammenhang mit der Speicherung von Texten müssen wir natürlich auch auf die eingebaute RAM-Disk von 112 Kbyte zu sprechen kommen. Dabei handelt es sich um einen Teil des insgesamt 256 Kbyte fassenden Speichers des JOYCE, der wie eine Floppy angesprochen werden kann. Legt man nun "M" als aktuelles Laufwerk fest, so spricht man mit Speicher- und Ladebefehlen immer die RAM-Disk an. Die Vorteile liegen auf der Hand: Daten können in unvorstellbaren Geschwindigkeiten geladen und wieder gespeichert werden. Dabei wird natürlich auch hier ein Inhaltsverzeichnis geführt. Als nachteilig könnte es sich erweisen, wenn der Strom einmal ausfällt. Dann sind alle Daten verloren. Die RAM-DISK stellt also einen sehr schnellen Zwischenspeicher dar, der bequem als Floppy angesprochen werden kann. Aus Gründen der Datensicherung wird aber immer zu einem Zwischenspeichern auf das richtige Laufwerk geraten.

LOGOSCRIPT

Bei der Textverarbeitung muß man den Programmierern ein großes Lob aussprechen. Die Bedienung von LOGOSCRIPT ist mit der gut durchdachten Menüsteuerung denkbar einfach. Mit den Funktionstasten kommt man in Untermenüs, die als kleine "Windows" in den aktuellen Bildschirm eingeblendet werden. Die Cursorstasten und zwei Spezialtasten reichen aus, um durch alle LOGOSCRIPT-Menüs zu kommen. Einfacher geht es schon bald nicht mehr. Für Profis gibt es noch den Expertenmodus. Hier kann man die einzelnen Funktionen

durch Eingabe von Abkürzungen ansprechen.

Beim Schreiben von LOGOSCRIPT fällt auf, daß nicht wie sonst üblich die Spalte angezeigt wird, in der sich der Cursor befindet. In der Kopfzeile, die entsprechend numerierte Markierungen enthält, bewegt sich ein zweiter Cursor synchron zu dem des Anwenders. Wer auf anderen Programmen eingearbeitet ist, wird sich erst daran gewöhnen müssen, und eine genaue Zahlenangabe wäre dieser Lösung wohl doch vorzuziehen gewesen. Das wäre aber der einzige Schwachpunkt der Textverarbeitung.

Ansonsten bietet LOGOSCRIPT natürlich die üblichen Möglichkeiten der Definition und Verschiebung von Blöcken, Ausdruck von Überschriften und Fußzeilen, Text zentrieren, rechtsbündig schreiben, sowie dem automatischen Suchen und Ersetzen von Textstellen. Besonders erwähnenswert ist die Möglichkeit, Textstellen nach Belieben mit Hilfe besonderer Tasten ein- und auszublenden. So können bestimmte Sätze in einen Zwischenspeicher geschoben und bei Bedarf wieder eingeblendet werden. Da die ausgeblendeten Texte beim Speichern des Dokuments ebenfalls gesichert werden, kann man sich eine kleine "Floskelbibliothek" anlegen. Immer wiederkehrende Begrüßungssätze oder schwere Fachwörter brauchen nicht jedesmal eingegeben zu werden. Ein Tastendruck genügt.

Da es sich beim JOYCE um ein Komplettsystem handelt, gibt es natürlich auch keine Probleme mit dem Ansprechen des Druckers. Es müssen keine Steuerzeichenorgien veranstaltet werden, um beispielsweise eine Textstelle zu unterstreichen. Im Gegenteil, auch das Einfügen von solchen Steuerzeichen erfolgt über die Menüs: Cursor auf die hervorzuhebende Textstelle fahren, Menü anwählen, bei der gewünschten Option ENTER drücken, fertig! Die Steuerzeichen werden auch nicht durch sonderbare Zeichen dargestellt. Wenn man eine Zeile kursiv drucken will, erscheint vor der Zeile "(kurs)". Inversdruck und Unterstreichungen werden sogar auf dem Bildschirm ausgeführt.

Eine nette Möglichkeit für Leute, die von der Schreibmaschine auf JOYCE umgestiegen sind, ist der Sofortdruck. Das heißt, daß jede Zeile, die mit LOGOSCRIPT geschrieben wurde, nach dem Betätigen der RETURN Taste sofort ausgedruckt wird. So kann eine Schreibmaschine simuliert werden, wenn auch eine sehr teure.

LOGOSCRIPT ist eine hervorragende Textverarbeitung, die extra für die vorgegebene Hardware entwickelt wurde.

So ergeben sich viele Vorteile, wie beispielsweise die Funktionstasten zum Editieren und die problemlose Zusammenarbeit mit dem Drucker. Jeder Anfänger wird sich leicht einarbeiten können, da alles menügesteuert ist. Die Dokumentation, die einen kleinen Trainingskurs für Anfänger enthält, tut ein übriges dazu.

Auf den Systemdisketten befindet sich noch das schon erwähnte CP/M Plus. Auch unter CP/M kann die RAM-Disk angesprochen und wie ein richtiges Laufwerk benutzt werden. Über die

MALLARD BASIC

Neuerungen und Besonderheiten des CP/M Plus werden wir in einem gesonderten Test berichten. Ebenso verhält es sich mit dem MALLARD BASIC, für das im Gegensatz zu CP/M Plus eine umfangreiche Dokumentation beiliegt. Im Befehlssatz vermißt man Grafik- und Soundbefehle, womit die gewünschte Anwendung der Sprache im professionellen Bereich wohl auf der Hand liegt. So hat man auch eine Fülle von Befehlen zur Dateiverwaltung zur Verfügung. Dabei können sequentielle Dateien, Random Access Dateien und Jetsam (Direktzugriff mit Schlüssel-Dateien) erstellt werden.

Last but not least findet man auf den Systemdisketten das allseits bekannte wie beliebte DR. LOGO, die Programmiersprache mit der Schildkröte. In den Handbüchern wird nur der Befehlssatz dieser sehr einfachen Sprache angegeben. Es empfiehlt sich zur näheren Information, weitere Bücher zu Rate zu ziehen. Das sei auch jedem ans Herz gelegt, der sich näher mit CP/M Plus beschäftigen will. An diesen beiden Punkten liegen die einzigen Schwachstellen der ansonsten recht guten und umfangreichen Dokumentation.

Fazit: Mit dem JOYCE bekommt man ein preiswertes Textverarbeitungssystem, das hervorragend für kleine bis mittlere Betriebe und sonstige Büros sowie Computerheimarbeiter und Übersetzer geeignet ist. Die Handbücher erleichtern auch dem totalen Anfänger den Einstieg in die Welt der EDV und LOGOSCRIPT ist eines der leistungsfähigsten und bedienungsfreundlichsten Programme überhaupt. Durch CP/M Plus und das Mallard Basic lassen sich auch noch weitere Anwendungen wie Datenbanken und Kalkulationsprogramme finden. Der JOYCE ist also keine "Freakmaschine" mehr, sondern der erste "Schneider" in der großen PC-Familie.

Thomas Tai

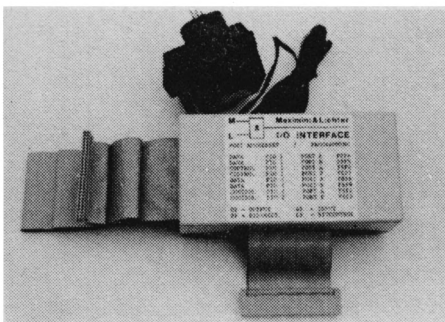
PIO-Interface und Epromer

Für den CPC-Elektronikfreak, der mehr aus seinem Computer machen will, gibt es jetzt diese beiden Erweiterungen. Damit kann der Computer dann zum Messen, Steuern, Regeln und Schalten verwendet werden.

Aktive Computerbenutzer, die auch das Innenleben ihres Spezies kennen, werden sich auf Dauer nicht mit den "normalen" Anwendungen und Spielen begnügen, sie wollen einfach mehr. Es ist ja auch so, daß in einem modernen Microcomputer Möglichkeiten stecken, die kaum ein Besitzer voll ausnutzt. Stichworte sind hier Messen, Regeln, Steuern und Schalten. Neben guten Kenntnissen der Hardware und der Programmierung sind hierzu aber Geräte notwendig, die es für die Schneider-Computer erst seit kurzer Zeit gibt. Zwei neue Produkte zu diesem Thema wollen wir heute vorstellen.

PIO-Interface

PIO ist die Abkürzung von Parallel Input/Output, also parallel Ein- und Ausgabe. Dieses PIO-Interface ist der Grundbaustein für weitergehende Anwendungen vielerlei Art. Ausgerüstet ist das Interface mit 2 Z80 A-PIOs die zum Z80 A-Prozessor in den CPC-Computern passen und eine Zusammenarbeit beider Geräte ermöglichen. Dem Benutzer stehen durch die beiden eingebauten PIOs insgesamt 32 Port-Leitungen zur Verfügung, die z. B. zum Steuern von Eisenbahnen oder Schrittmotoren oder zur Meßwertverarbeitung genutzt werden können (dem Anwender sind kaum Grenzen gesetzt). Verbunden wird das PIO-Interface über ein 50-poliges Kabel und einem entsprechenden Stecker direkt mit dem Erweiterungsport des CPC 464/664. Die Stromversorgung erfolgt über ein eigenes Netzteil, damit der Rechner nicht überlastet wird.



Das PIO-Interface

Die Port- und Interruptleitungen werden an der rechten Gehäusesseite mit einem ca. 35 cm langen Kabel herausgeführt. Durch diese sinnvolle Anordnung können weitere Schaltungen neben dem Computer installiert werden.

Passend zu dem PIO-Interface bietet der Hersteller folgende Ergänzungen an (bzw. werden in Kürze angeboten):



Der erweiterte CPC

- Simulationsgerät mit Schaltern und Leuchtdioden zur Simulation von Prozessoren
- Analog-Eingabe 0...10 V mit 8 Bit Auflösung
- dto. für Ausgabe
- Digital-Ausgabe mit Relais 24 V/2,5 A
- Digital-Eingabe mit Optokoppler
- Schrittmotoren-Modul

In dieser Aufstellung fehlt noch das Eprom-Programmiergerät (EPROMER), das bereits erhältlich ist. Mit ihm können nachstehende EPROMs bearbeitet werden:

- | | | |
|-------|---|--|
| 2716 | - | 2 K, |
| 2732 | - | 4 K, |
| 2732A | - | 4 K, |
| 2764 | - | 8 K, |
| 27128 | - | 16 K (von INTEL oder kompatibel), |
| 2516 | - | 2 K, |
| 2532 | - | 4 K, |
| 2564 | - | 8 K (von Texas Instruments oder kompatibel). |

Das Gerät kann nur in Verbindung mit dem PIO-Interface des Herstellers eingesetzt werden, über das es auch mit Strom versorgt wird. Die notwendige

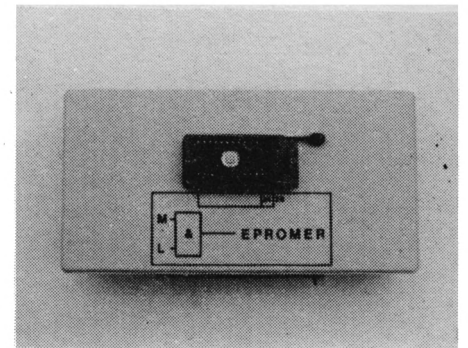
Treibersoftware liegt dem Gerät in einer Kassettenversion bei. Die Software auf Diskette gibt es gegen einen geringen Aufpreis. Das Programm ist voll menügesteuert, was die Anwendung erheblich erleichtert und Fehler vermeidet.

Möglich sind 4 Betriebsarten:

- Lesen EPROM in RAM
- Leertest EPROM
- Programmieren des EPROMs
- Vergleichen RAM-Inhalt/EPROM-Inhalt

Außerdem wird ein Maschinencode-Monitor mitgeliefert.

Der Umgang mit dem EPROM-Programmiergerät ist relativ einfach und erfordert keine großen Vorkenntnisse. Bei den Produkten (PIO-Interface und EPROMER) liegt eine ausführliche Bedienungsanleitung bei, die weitere Hilfestellung gibt.



Der Epromer

Für Interessenten hier noch die Preise:

PIO-Interface:	
Platinenversion	ca. DM 198.-
Gehäuseversion	ca. DM 218.-
EPROMER:	
Platinenversion	ca. DM 179.-
Gehäuseversion	ca. DM 205.-

Die Gehäuse bestehen aus hellgrauem Kunststoff und machen einen robusten Eindruck. Bastler werden vielleicht mehr Gefallen an der Platinenversion finden.

Die Amstrad Consumer Show

Schneider-Computer sind im Kommen, das belegen nicht nur die deutschen Verkaufszahlen, das zeigt sich auch an dem großen Interesse, das englische Käufer an den Tag legen, wenn es um Neuheiten für ihren Amstrad geht. Denn England war schon immer ein Indikator dafür, wie erfolgreich sich ein bestimmter Computer auch bei uns verkaufen kann.

Anfang Oktober fand in London eine reine Schneider/Amstrad Verbrauchermesse statt. Innerhalb von zwei Tagen brachte sie insgesamt 10 000 Besucher auf die Beine. Der englische Computermarkt hat ein neues Lieblingskind. Nach dem BBC-Computer, dem C 64 und dem Spectrum heißt der neue Rechner zweifellos Amstrad/Schneider. Das Interesse ist groß. Für knapp 200 Pfund geht der CPC 464 mit grünem Monitor weg wie warme Semmeln, der CPC 664 ist schon ab 299 Pfund überall zu haben. Dagegen können auch Commodore-Rechner, die als Gesamtpaket mit Spielen zu Schleuderpreisen angeboten werden, nicht ankommen.

Wie groß die Schar der Schneider-Besitzer in England inzwischen ist, bewies der große Andrang bei der Amstrad Computer-Show. Wer etwas sehen wollte, mußte sich erstmal in einer langen Schlange vor dem Eingang gedulden und dann zwei Pfund bezahlen, bevor er überhaupt Einlaß fand.

Innerhalb von nur drei Wochen hatte der Verlag Database Publications diese Zwei-Tage-Show aus dem Boden gestampft. Knapp 70 Aussteller boten an 115 Ständen ihre Neuheiten an. Dabei zeigte sich, wie stark sich der englische Markt trotzdem vom deutschen unterscheidet. Wird beispielsweise der Schneider bei uns hauptsächlich für den halbprofessionellen Betrieb gekauft, glauben die Engländer – und das wohl nicht zu unrecht – daß man mit dieser hervorragenden Kompaktanlage auch sehr gut spielen kann.

Spiele, Spiele und kein Ende

Wie gut sich ein Schneider auch für diesen Zweck eignet, bewies die Flut der angebotenen Programme. Wirklich Neues gab es dabei aber nicht. Vielfach wurden C 64 und Spectrum-Dauerbrenner einfach für den Schneider umgeschrieben und neu aufgelegt. Erwähnenswert und positiv ist lediglich, daß

die englischen Softwarefirmen sich von vornherein auf den Export einstellen und beispielsweise Spielbeschreibungen auch in deutsch liefern, ein echtes Novum. Neu ist auch, daß die meisten Programme gleichzeitig auf Kassette und auf 3 Zoll Diskette lieferbar sind. Und das, obwohl in England zumeist der CPC 464 verkauft wird.

Das gleiche Bild wie bei den Spielen zeigte sich mehr oder minder auch bei den Anwenderprogrammen. Firmen wie Tasman, die vor allem durch hervorragende Anwendersoftware für den Spectrum bekannt geworden sind, bieten ihre Palette nun auch für den Schneider an. Doch auch das ist hierzulande schon bekannt. Durch einen Exklusiv-Vertrag für Deutschland können deutsche Kunden bereits seit einigen Wochen bei der Firma Profisoft die deutsche Taswordversion erwerben.

Auch die Firma Schneider sicherte sich durch einen solchen Vertrag die Rechte für Deutschland für die Hisoft Angebotspalette, was allerdings nur für die Kassetten-Version gilt. Die Diskettenversionen der Programme Pascal, Devpac und C muß der interessierte Kunde weiterhin über England beziehen, es sei denn, andere deutsche Händler finden sich, die diese Programme in ihr Sortiment aufnehmen.

Zusätzlich zu diesen drei Spitzenprogrammen stellte Hisoft auf dieser Messe ein neues Grafikprogramm mit dem Namen "Font 64" für den Schneider vor. Mit diesem Utility ist es nicht nur mög-

lich, erstellte Zeichnungen zu spiegeln und rotieren zu lassen, sie können auch in beliebige Basic-, Maschinencode- und Pascal-Programme implementiert werden.

Eine weitere echte Neuheit konnte am dk'tronics-Stand bewundert werden. Neben einer Speichererweiterung für den CPC 464 auf 64 K oder gar 256 K gibt es jetzt auch einen Sprachsynthesizer für den Schneider. Eine Speichererweiterung für die Disketten-Schneider ist in Vorbereitung und soll Ende des Jahres ausgeliefert werden.

Auch diejenigen, die den Schneider "künstlerisch" nutzen wollen, werden sich freuen: die Firma Advanced Memory Systems Ltd. zeigte eine Maus, die man auch bei uns bald bei Schneider bekommen soll. Für knapp 80 Pfund sind in diesem Paket außerdem vier Programme enthalten, darunter ein tape-to-disc Kopierprogramm für die erstellten Grafiken.

Was Kopierprogramme angeht, so zeigte es sich, daß viele Anbieter mit einem allgemeinen Trend zu den Disketten-Schneidern rechnen. Sozusagen vorbeugend, damit nicht das gleiche passiert, was Spectrum-Besitzern mit ihrer Beta-Disc-Floppy passiert ist, bieten viele Tape-to-Disc-Programme an. Wie die Entwicklung aber wirklich verläuft, wird sich Anfang nächsten Jahres herausstellen. Denn da plant Database ihre zweite Consumer Show. Diese soll noch größer werden und drei Tage dauern.

Karina Krawczyk



Großer Andrang auf der Amstrad Consumer Show

Der sprechende Computer

Stereo Sound mit dem Speech Synthesizer

Die englische Firma dk'tronics ist schon seit Jahren bekannt für Peripheriegeräte, bisher allerdings nur für den Sinclair ZX Spectrum. Da auch in England die Amstrad/Schneider-Computer gut verkauft werden, liegt es auf der Hand, daß auch für diese Geräte Zubehör angeboten wird. Mit dem Speech-synthesizer ist das nun der Fall.

Aus der Gerätebezeichnung geht schon hervor, daß es sich um Spracherzeugung handelt, vorab aber kurz zu dem Gerät bzw. besser zu den Geräten selbst. Die Packung enthält

- 1 Sound-Interface
- 2 4-Zoll Lautsprecher
- 1 Softwarekassette
- 1 Anleitung

Das Sound-Interface wird direkt auf den Erweiterungsport (Floppy) gesteckt. Die Steckerleiste ist sinnvollerweise auf der Rückseite des Interfaces durchgeführt, so daß weitere Peripherie problemlos angeschlossen werden kann. Am Interface selbst befindet sich noch ein kleines Kabel, das mit dem Hifi-Stereo-Ausgang am Computer verbunden werden muß. Der letzte Schritt der Installation ist das Einstecken der beiden Lautsprecherkabel.

Die Sound-Einheit ist nun betriebsbereit. Alle vom Computer erzeugten Töne und Geräusche können jetzt über das Interface und die Lautsprecher wiedergegeben werden. Die dabei erzeugte Lautstärke ist beachtlich und übertrifft die des eingebauten Lautsprechers bei weitem. Gut und sehr wichtig ist es, daß die Lautstärke am Interface reguliert werden kann. Allerdings nur mit spitzen Fingern und etwas Mühe, da der Lautstärkeregler klein ausgefallen ist und sich schwer bewegen läßt. Auch der über die beiden Lautsprecher übertragene Klang ist nicht übel. Wer seine Nachbarn ärgern will, hat jedenfalls mit dem dk'tronics Gerät beste Chancen dazu. Neben der normalen Soundübertragung hat man nun auch die Möglichkeit, hervorragende Stereo-Effekte zu programmieren und zu erleben, ohne gleich die heimische Hifianlage zu benutzen.

Nun aber zu der Sprachausgabe, dem eigentlichen Zweck des Speech Synthesizers. Zum Einsatz muß erst die mitgelieferte Software geladen werden. Es handelt sich dabei um ein 2-teiliges Programm, einem Basic-Lader und einem 4 KByte umfassenden Maschinencode-

programm. Der M-Code kann im Speicherbereich von 16384 bis 39000 untergebracht werden. In diesem Bereich ist er frei verschiebbar. Nach beendetem Ladevorgang meldet sich der Computer mit einer dk'tronics Copyright Meldung, die auch gesprochen wird (also nicht erschrecken). Dem Anwender stehen nun 8 neue Basic-Befehle zur Verfügung, mit denen er auf 4 verschiedene Arten den Computer sprechen lassen kann.

1. Die Sprache wird in DATA-Zeilen abgelegt und über ein Basic-Programm eingelesen (geht auch ohne Software).

2. Die Sprache wird ebenfalls in DATA-Zeilen abgelegt, aber es erfolgt ein direkter Aufruf über einen Befehl.

3. Spracherzeugung über PRINT-Anweisungen.

4. Alles, was auf dem Bildschirm erscheint, wird ausgesprochen.

Die 1. Methode hat den Vorteil, daß die Treibersoftware nicht unbedingt geladen werden muß, da das Interface bereits die Phonetik im Speicher hat. Die auszusprechenden Laute werden Buchstabe für Buchstabe in Werte umgewandelt, die man aus der Anleitung entnehmen kann. Dort stehen auch diverse Wortbeispiele zum Ausprobieren. Die so ermittelten Werte können in DATA-Zeilen abgelegt und durch ein kleines Basic-Programm eingelesen werden. Unter Verwendung des neuen Befehls FEED kann dieses Programm entfallen, dafür muß vorher unbedingt die Software geladen werden. Der Weg zur Spracherzeugung ist ansonsten immer mit dieser Methode vergleichbar.

Noch einfacher und mir persönlich auch angenehmer ist allerdings die Spracherzeugung über PRINT-Anweisungen. Hier wird dem Computer über

den neuen Befehl SPON (Speech On) mitgeteilt, daß Sprache folgt. Was danach in einer PRINT-Anweisung steht, wird auch gesprochen.

Beispiel:
10 SPON
20 PRINT "Computer"

Man kann bei diesen Experimenten aber schnell den Unterschied zwischen Schreiben und Sprechen lernen, da die meisten Wörter doch am Anfang noch lange nicht so klingen, wie man es vielleicht erwartet hat. Das liegt zum großen Teil daran, daß der Speech Synthesizer in England geboren wurde und mit der deutschen Aussprache Probleme hat. Das Angenehme bei dieser Methode ist aber, daß man durch das Verändern einzelner Buchstaben und der ständigen akustischen Kontrolle doch nach und nach sein Ziel erreicht. So kann man sicherlich auch bei Methode 1 vorgehen, was aber den Nachteil hat, immer in der Wertetabelle nachschlagen zu müssen.

Die letzte Funktion des Interfaces ist wohl mehr als Gag gedacht. Man kann den Computer anweisen, alles, was auf dem Bildschirm erscheint, auszusprechen (auch Fehlermeldungen o. ä.). Es besteht z. B. die Möglichkeit, ein komplettes Programmlisting akustisch auszugeben, sprich vorlesen zu lassen. Mehr als einmal hat aber daran wohl niemand Freude.

Wer nun erwartet, daß die Stimme des CPC menschlich klingt, erwartet sicher zuviel. Es ist eine metallisch klingende Stimme, vergleichbar mit den in Science Fiction Filmen so gerne verwendeten Roboterstimmen. Mit etwas Übung bei der Programmierung kann man seinen Computer aber durchaus verstehen.

Durch die verschiedenen Möglichkeiten, Sprache zu erzeugen, wird wohl jedem Benutzer etwas geboten. Auch der Maschinencode-Programmierer kann der Anleitung einige nützliche Hinweise entnehmen. Ob der sprechende Computer allerdings notwendig ist, bzw. welche sinnvollen Einsatzmöglichkeiten die Spracherzeugung im privaten Bereich bietet, darüber kann man sich streiten. Zusammen mit der Ton-Verstärkung und dem Stereo-Effekt bietet der Speech Synthesizer auf jeden Fall einige Zeit Unterhaltung und Abwechslung vom "normalen" Computer.



Der Speech Synthesizer

Rolf Knorre

Textmaster

Das Textverarbeitungsprogramm aus dem Heim-Verlag

Der Darmstädter Heim-Verlag hat sich in der Vergangenheit speziell mit Büchern zu Triumph Adler Computern einen Namen gemacht. Mit dem Erscheinen der CPC-Computern wurden ebenfalls einige Bücher und Programme zu diesem sehr erfolgreichen Computer veröffentlicht. So auch ein Textverarbeitungsprogramm mit dem Namen TEXTMASTER für den CPC 464 und CPC 664.

Auf der mitgelieferten Programmdiskette befindet sich das Programm zweimal: je eine Version für den CPC 464 und den CPC 664. (Getestet wurde die 464-Version). Die beigelegte Bedienungsanleitung ist mit 24 Seiten ausreichend dokumentiert.

Nach dem Laden und Starten des Programms erscheint das Hauptmenü, in welchem alle Grundeinstellungen wie linker und rechter Rand, Tabulatoren, Schriftart des Druckers etc. ausgewählt werden können. Hier läßt sich auch der Zeichensatz (international oder deutsch) wählen. Leider ist beim deutschen Zeichensatz die Tastatur nicht komplett auf die DIN-Norm umgelegt worden, so daß sich z. B. der Doppelpunkt und das Semikolon auf den Tasten mit den eckigen Klammern befindet. Auch kann TEXTMASTER nur maximal 80 Zeichen pro Zeile verarbeiten. Das Erstellen von Tabellen mit mehr als 80 Zeichen Breite ist nicht möglich. Der linke und rechte Rand ist frei innerhalb dieses Bereichs einstellbar. Das Hauptmenü ist optisch und funktionell recht gut aufgebaut.

Maximal 80 Zeichen pro Zeile

Sind alle Anfangsparameter ausgewählt (sie können jederzeit geändert werden), gelangt der Benutzer mit der Eingabe eines "Z" in das eigentliche Textverarbeitungsprogramm. Für die Eingabe des Textes stehen 22 Bildschirmzeilen zur Verfügung. Die restlichen 3 Zeilen bilden die Status- und Tabulatorzeile. Die obere Tabulatorzeile (Zeilenlineal) zeigt die Position der gesetzten Ränder an. In der Statuszeile stehen neben den Angaben zur Cursorposition die Abkürzungen für alle möglichen Steuerbefehle, die in Kombination mit der Control-Taste abrufbar sind.

Wie von einem Textverarbeitungsprogramm gewohnt, kann man mit der

Texteingabe beginnen. Mit den Cursor-Tasten läßt sich der Cursor nur über die aktuelle Bildschirmseite bewegen. Gelangt der Cursor an den oberen oder unteren Bildschirmrand, so tut sich erst einmal nichts mehr. Jetzt muß erst eine neue Bildschirmseite angewählt werden. (SHIFT Cursor up/down). Gelangt man allerdings beim Schreiben in die unterste Bildschirmzeile, scrollt der Text erst einmal um 11 Zeilen nach oben, bevor es mit der Eingabe weitergehen kann. Diese Lösung ist sicherlich nicht optimal, aber das langsame Scrollen des CPCs ist schon ein Problem.

Zum Editieren eines Textes sind diverse Funktionen (Zeilen löschen und einfügen, Textteile kopieren etc.) vorhanden, die allesamt zwar zuverlässig aber doch recht langsam arbeiten. Oft holt sich das Programm weitere Informationen von der Programmdiskette. So auch für den Blocksatz, der zeitlich doch an den Nerven zerrt. (Leider ist ein rechtsbündiges oder zentriertes Formatieren nicht möglich.)

Eine gute Idee ist die Möglichkeit, einen Anschriftenrahmen in den Bild-



schirm einzublenden. Der Rahmen bestimmt einen 8 Zeilen großen Bereich, in dem man bei Fensterbriefumschlägen die Anschrift eintragen muß.

Nach genau 132 Textzeilen meldet sich das Programm mit einem Höllenlärm: Der Textspeicher ist voll und muß abgespeichert werden! Insgesamt lassen sich also rund 2 DIN A4 Seiten auf einmal erstellen und bearbeiten. Das ist nicht viel. Allerdings bietet das Programm die Möglichkeit, Texte verkettet auszudrucken.

Textmaster ist ein Textverarbeitungsprogramm der unteren Preis- und Leistungsklasse (ca. 89,- DM). Viele Ideen wie z. B. der Anschriftenrahmen sind gut und stellen für den Gelegenheitschreiber eine gute Hilfe dar. Auch durch den gelungenen optischen Aufbau und die Kombination aus Menü und SteuerCodes läßt sich problemlos damit arbeiten. Ein professionelles Arbeiten ist mit Textmaster allerdings nicht möglich (aber welches Textsystem kann diese Auszeichnung schon aufweisen).

Meiner Meinung nach sollte der Heim-Verlag das Programm noch verbessern (Zeilenbreite über 80 Zeichen, größerer Textspeicher, DIN-Tastenbelegung, besseres Scrollen), um konkurrenzfähig zu bleiben. Auch könnte man die Cursorbewegung sicherlich beschleunigen, wenn z. B. nach jeder vertikalen Cursorbewegung nicht mehr die gesamte (!) Statuszeile neu aufgebaut werden muß.

Manfred Walter Thoma



Eprommer Schneider CPC 464/664

Universeller EPROM-Programmer 4003

- Programmiert alle gängigen EPROM-Typen (z.B.: 2716,-32,-64,-128,2508,-16,-32,-64,...)
- Voll menügesteuerte Software auf Kassette
- Kein Schallen, Stecken oder Löten nötig
- Programmierspannung wird im Gerät erzeugt
- Verbindung zum CPC über Flachbandkabel und Interface-Karte
- Gleichzeitiger Anschluß der Floppy möglich
- Rote und grüne Leuchtdiode zur Betriebs-Art-Anzeige
- Kompl. mit 28 poligem Textool-Sockel
- Fertigergerät DM 289,50 ■ Bausatz mit Anleitung DM 239,-



EPROM Karte 64 KByte

Die ideale Ergänzung für jeden CPC

- Wahlweise bestückbar mit 2 - 64 KByte EPROM-Kapazität
- Arbeitet mit den EPROM-Typen 2716,-32,-64,-128
- Durchgeführter Erweiterungsbus (Floppy kompatibel)
- Autostart von BASIC- und/oder Assembler-Programmen
- Komplett mit umfangreicher und komfortabler Software
- Gleichmaßen für Profis und Einsteiger geeignet
- Fertigergerät DM 249,50 ■ Bausatz mit Anleitung DM 219,50

Drucker für alle CPC

EEDY 100-80 SPEEDY 100-80 SPEDDY 100-80 SP

- 100 Zeichen pro Sekunde schnell
- FX80 kompatibel
- Bis zu 142 Zeichen pro Zeile
- Optionaler Druckerpuffer
- Grafikfähig
- Kein doppelter Zeilenvorschub
- Direkt anschlussfähig
- Internationale Zeichensätze
- Gutes Preis-Leistungsverhältnis
- Kompl. mit deutschem und englischem Handbuch DM 739,-
- DruckerKabel CPC 464/664 DM 35,- ■ CPC 6128 DM 45,-
- Preise inkl. Mehrwertsteuer. Alle Artikel ab Lager lieferbar.

DOBBERTIN
INDUSTRIE-ELEKTRONIK
Braunsstraße 9, 6835 Brühl, Tel.: (06202) 714-17

RAM ist im kleinsten Schneider

Seit kurzem gibt es die VORTEX-Speicherkarten auf dem Markt. Damit hat sich der CPC 464 zum fast professionellen Personalcomputer gemausert. Bis 512 KByte gehen die Erweiterungen, Speicherplatz ist also kein Problem mehr.

Als die ersten Hinweise auf eine mögliche Erweiterung im Lande kursierten, habe ich gleich auf Verdacht bestellt. Das war im Juni. Ende September kam sie endlich an, die inzwischen geliebte SP128. Nun ja, gut Ding will Weile haben.

Was versteht man nun eigentlich unter einer Speichererweiterung? Wie Sie als Schneider-User wissen, stehen die 64 KByte des random-access-memory (RAM) nicht vollständig zur Verfügung, sondern das System knappst erst einmal 16 KB für den Speicher des Bildschirms ab und dann noch zusätzliche Kilos für alles, was ein System eben so braucht, seien es Sprungleisten, Merzzellen, Adressen und dergleichen Dinge mehr. Dadurch wird die Bewegungsfreiheit des Users z. B. bis auf ganze 42619 Byte eingeeengt, wenn eine Floppy angeschlossen ist. Und selbst das Betriebssystem CP/M schämt sich, daß es nur mit 44 KB aufwartet, weil es damit keine großen Programme wie WORDSTAR, dBASE usw. fahren kann.

Hier greifen nun die Speichererweiterungen hilfreich ein. Da die Z80-CPU nun einmal nicht mehr als 64 KB adressieren kann, benutzt man den gleichen Trick, den auch die Väter des CPC 464 benutzt haben, um die ROMs (read-only-memory) anzusprechen. Man legt gewissermaßen Speicherteile mit gleichen Adressen neben- bzw. übereinander an, von denen dann nur der Teil arbeiten muß, der gerade benötigt wird. Das ist ähnlich wie mit den Parallel-Hausnummern 13a, 13b, 13c. Solche Teile nennt

bank switching

man eine Bank. Die dazugehörige Umschalterei heißt "bank switching". Schauen Sie sich dazu die Figur 1 an. Im linken Teil sehen Sie die Bankverteilung des "nackten Schneiders", rechts das, was Vortex dazubringt. Figur 2 zeigt Ihnen, wodurch sich die einzelnen SP-Karten unterscheiden: Jedes durchkreuzte

Kästchen bedeutet einen Chip von der Größe, die darunter vermerkt ist. Wenn Sie sich nur eine kleine Karte gekauft haben, können Sie diese je nach Bedarf und Geldbeutel einfach durch Zustecken oder Austauschen von Chips bis auf maximal 15 Zusatzbänke zu je 32 KB aufblasen. Das ergibt insgesamt also 16 Bänke einschließlich Bank 0, dem Rest des Hauptspeichers. Es gibt allerdings auch eine Karte SP64/M (M wie Mager). Sie hat keinen Buspuffer und ist deshalb nicht ausbaufähig. Beide 64er-Karten erlauben keine Ramdisk.

Was Sie beim Öffnen Ihres frisch erworbenen Paketes in der Hand haben, ist in Figur 3 zu sehen. Nur die Felder "Z80 CPU" und "Gate array (ULA)" sind noch frei. Weiter finden Sie noch ein handlich schmales Teil im DIN A5 Format mit der Aufschrift "Benutzerhandbuch". Hier nun gleich meine erste und fast einzige Kritik: Es ist bestenfalls ein Benutzerhändchenheftchen mit 42 Seiten Umfang.

Die ersten 12 Seiten beschreiben sehr genau den Einbau sowie die Erstinstallation für die VORTEX- und Schneider-Diskettenstationen. Wenn Sie sich wirklich danach richten, können Sie nichts falsch machen. Und Sie brauchen wirk-

lich keinen LötKolben. Zwei Schraubendreher und eventuell eine kleine Zange für das Kühlblech der ULA genügen zum Einbau. Soweit ist alles in Ordnung. Dann aber werden 39 Befehle auf 23 Seiten erklärt. Fünf Seiten davon beansprucht allein der Befehl `!Mon` (Monitor). Bleiben noch 18 Seiten für 38 Befehle. Zugegeben, der gut leserliche Kleindruck bringt manches und die Sternenkästchen um jeden Befehl erhöhen die Übersicht. Auch die knappen Befehlsbeschreibungen sind exakt und fatal errors habe ich nicht gefunden.

Alle Erläuterungen folgen dabei dem Schema:

Name, Format, Funktion, Bemerkung, Beispiel.

Und bei den letzten beiden Punkten hapert's. Das Beispiel beschränkt sich durchweg auf die Wiederholung des Formats mit vorgesetzter Zeilennummer. Die Bemerkungen müßten viel informativer sein. Frage: Müssen beim Aufruf eines MC-Codes durch `!CALL` mit Bankwechsel die Parameter vorher mit `!COMMON` übergeben werden? Das hätte nur Sinn, wenn sie in der neuen Bank an der alten Stelle stehen (Adressen werden übergeben). Wenn dort aber

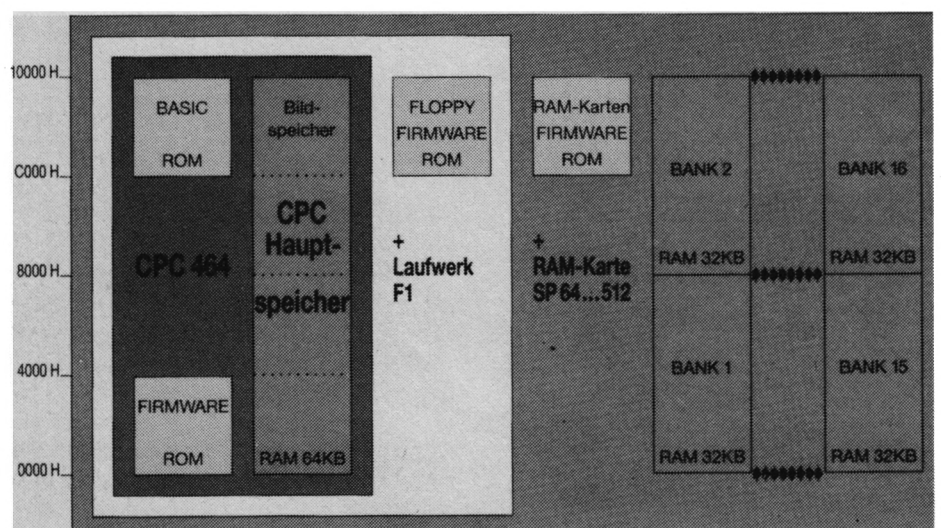


Fig. 1, Speicherorganisation des CPC

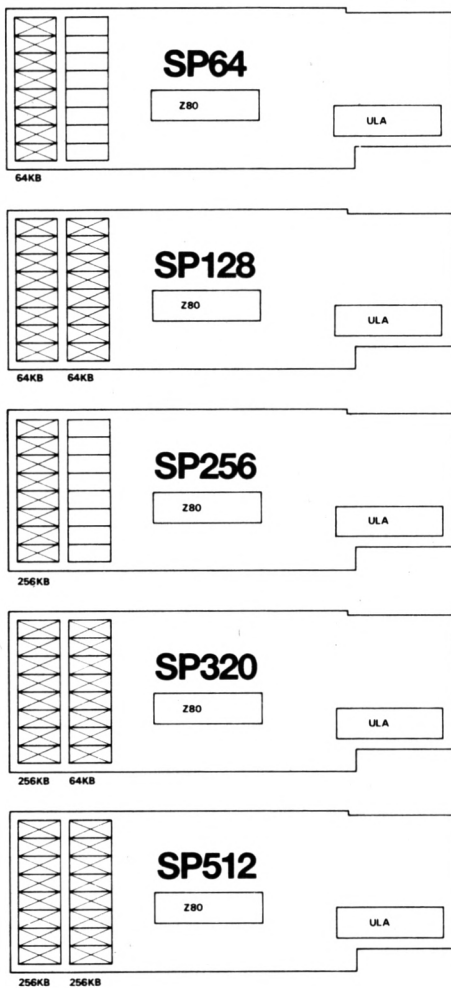


Fig. 2, Das 520er Konzept

schon was anderes steht? Wird das vorher gerettet? In der Anleitung steht dazu nichts.

Wie gesagt, das Einbauen ist problemlos. Nur Vorsicht, damit Sie dabei den beiden umzusteckenden ICs Z80 und ULA nicht die Beinchen brechen. Danach ist die RAM-Karte spurlos im Innern des CPC 464 verschwunden. Wenn Sie nun den gut verschraubten Rechner wieder umgedreht haben, ist bei den Geräten ohne Floppy die Arbeit getan, es sei denn, Sie hätten die CPU oder ULA verdreht. Dann müssen Sie die Arbeit noch einmal machen. Kaputtgehen könnte trotzdem nichts, sagt VORTEX, wir haben es aber nicht ausprobiert.

62 K CP/M

Für die Floppyisten geht die Arbeit danach erst richtig los. Mit Hilfe der zur Auslieferung gehörenden Kassette muß eine neue Systemdiskette erstellt werden. Die Stammprogramme FORMAT, COPY oder DISCOPY, CASCOPY oder CLOAD, MOVCPM und SYSGEN werden zur Arbeit gebeten und dann das Kernstück der Installation, das Programm PATCH.COM von der Kassette

geladen. Die Arbeitsvorgänge sind Schritt für Schritt im Heftchen erklärt. Endlich hat dann jeder sein Erfolgserlebnis: 62 K CP/M strahlt's uns entgegen. Genau genommen sehen Sie zunächst Bild 5 unmittelbar nach dem Einschalten. Trotzdem, mit 62 K CP/M sind die Wünsche eines jeden Freaks erfüllt, denn dicker geht's nimmer.

Das Programm PATCH.COM erzeugt auf der Diskette drei Zusatz-Files, nämlich RAMDISK.COM, SPOOL.COM und \$OSC.SYS. Mit diesen Dienstprogrammen können Sie unter CP/M die RAM-Diskette formatieren und den Drucker-spooler ein- und ausschalten. Sie haben außerdem die Qual der Wahl, ob das bei jedem CP/M-Neustart geschehen soll oder nicht. Die für sie beste Lösung werden Sie wohl trotz der Anleitungshilfe erst im Laufe der Zeit finden.

Gleiches gilt auch für den Systemvektor \$OSC, ohne den die Utilities INSTALL, FORMAT usw. nicht laufen, der aber bei DDT.COM und SAVE.COM Schwierigkeiten macht. Auf jeden Fall können Sie mit PATCH.COM alles wieder umfunktionieren, was tröstlich ist. Apropos PATCH: Der diensthabende Geist COPY muß umgepatcht werden. Er ist sonst nicht in der Lage, mit 62 CP/M fertigzuwerden. Von DISCCOPY steht nichts in der Anleitung.

Da sie mit Sicherheit zumindest am Anfang beide CP/M-Größen nebeneinander benutzen werden, empfehle ich Ihnen, die "großen" Disketten besonders zu kennzeichnen (z. B. mit einem roten Aufklebepunkt auf dem Herstelleretikett) und das neue, geflickte COPY unter dem Namen COPY62.COM darauf abzulegen. Um die Speicher- und Bankverteilung brauchen Sie sich unter CP/M nicht zu kümmern, das geht alles herrlich automatisch.

Bis jetzt sieht es so aus, als ob Sie zum Einbau einer der Speichererweiterungen unbedingt eine Diskettenstation haben müßten. Dem ist nicht so. Der CPC 464 meldet sich mit dem gleichen Bild 5, weist jedoch als himem 43899 Byte aus, mit Diskette dagegen nur 42615. An diese Zahlen sind wir bis auf ein paar Bytes gewöhnt. Sie stören nur bei BASIC-Programmen, welche die obere Speichergrenze exakt abfragen, um zum Beispiel Speicherplatz für Maschinenprogramme zu reservieren. Hier müssen Sie leicht korrigierend eingreifen und eventuell sogar die MC-Teile anpassen, wenn sie nicht relozierbar sind.

Anders sieht die Sache aus, wenn man die neuen Befehle mit IBOS aktiviert (siehe Bild 6). Statt des erwarteten Riesenspeichers meldet himem nur jämmerliche 32616 Byte! Keine Panik, das ist nur die Bank 0, der Rest vom Urspeicher. Bei allen anderen Bänken zeigt himem 32767 Byte an. Diese dürfen von uns nicht erhöht, nur erniedrigt werden. Die Bank 0 rutscht sogar automatisch auf nur noch 16360 Byte, wenn man den Befehl IVIDEO.ON eingibt. Dafür erlaubt diese Befehlsgruppe den Wechsel von kompletten Bildschirmdarstellungen in nur 0,3 Sekunden. Bei Vollausbau kann man so 16 Bilder mit dem Tempo von drei Bildern pro Sekunde über den Schirm huschen lassen.

Die niedrigen Bytezahlen pro Bank hängen mit dem bank switching zusammen. Sie können also ruhig eine Bank vollprogrammieren, schalten dann mit IBANK,n um und machen auf der nächsten weiter. Auf jeder Bank können die gleichen Zeilennummern verwendet werden. Sie geraten nicht durcheinander und können von Bank zu Bank mit IGOTO und IGOSUB angesprungen werden. Dabei darf die Zeilennummer sogar

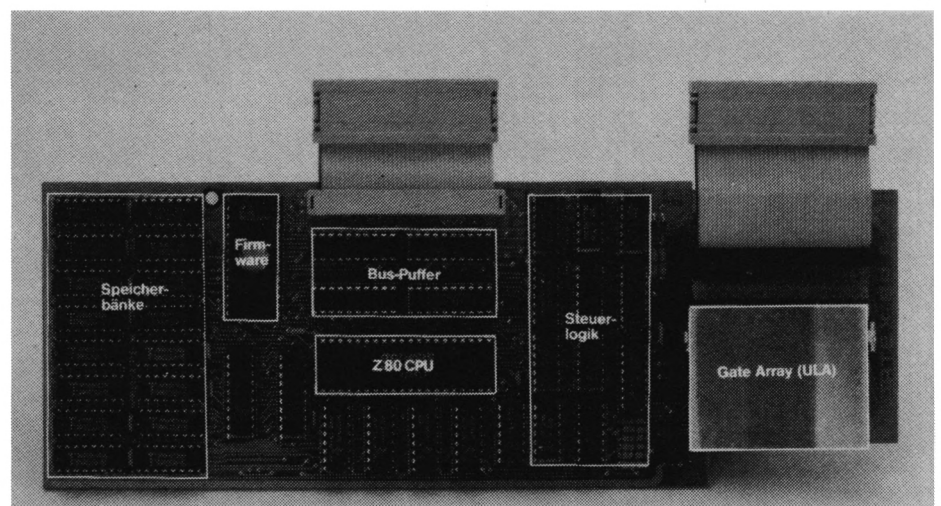


Fig. 3, Funktionsgruppe der Vortex RAM-Karte

	SP64/M	SP64	SP128	SP256	SP320	SP512
Ausbaubar	nein	ja	ja	ja	ja	nein
gepufferter Bus	nein	ja	ja	ja	ja	ja
freier Speicherplatz unter BASIC	64KB	64KB	96KB	160KB	192KB	288KB
freier Speicherplatz unter CP/M	60KB	60KB	60KB	60KB	60KB	60KB
RAM-Floppy	keine	keine	64KB	192KB	256KB	448KB
Betriebssoftware für BASIC	auf Cassette	in ROM	in ROM	in ROM	in ROM	in ROM
Betriebssoftware für CP/M	auf Cassette	auf Cassette	auf Cassette	auf Cassette	auf Cassette	auf Cassette
RAM-Speicher auf der Karte	64KB	64KB	128KB	256KB	320KB	512KB
ROM-Speicher auf der Karte	keiner	16KB	16KB	16KB	16KB	16KB
Spannungsversorgung	+5V	+5V	+5V	+5V	+5V	+5V
Temperaturbereich	-50°C bis +75°C	-50°C bis +75°C	-50°C bis +75°C	-50°C bis +75°C	-50°C bis +75°C	-50°C bis +75°C
Gewicht	170 Gramm	200 Gramm	230 Gramm	200 Gramm	230 Gramm	230 Gramm
Abmessungen (L x B)	18 x 10 cm	25 x 9 cm	25 x 9 cm	25 x 9 cm	25 x 9 cm	25 x 9 cm

Fig. 4, Technische Daten RAM-Erweiterungen

berechnet und in einer Variablen gespeichert sein.

Etwas anders sieht es mit den Variablen aus. Sie gelten im Prinzip nur in der Bank, in der sie definiert wurden. RENATE auf Bank 0 ist also eine andere RENATE als die auf Bank 1. Die Variablen müssen deshalb erst mit ICOMMON den anderen Bänken zugänglich gemacht werden. An dieser Stelle beginnt für den Nur-BASIC-Mann völliges Neuland. Die Variablenwerte müssen nämlich echt von einer Bank zur anderen geschaufelt werden. Und das kostet Geld. Bei Feldern geht es praktisch gar nicht (sagt VORTEX). Der Programmierer muß sich also vorher überlegen, wie er sein Programm so aufbauen kann, daß möglichst wenig Variablen zu transferieren sind und daß er mit wenigen Sprüngen auskommt. Er ist also gezwungen, sein Programm zu strukturieren.

Hier wäre noch eine dankbare Aufgabe für das Haus VORTEX: Schaffen Sie einen echten Common-Bereich, auf den alle Bänke zugreifen können, und in dem nur Variablen abgelegt werden, die schon am Programmanfang als common deklariert wurden.



Ähnlich sieht es bei der Ramdisk aus. Sie erlaubt zwar einen relativen Zugriff, kann aber nur Strings aufnehmen. Hier fehlen die schnellen Befehle CVx und MKx. Wenn man erst mit STR\$ und VAL arbeiten muß, so geht der Zeitvorteil einer Pseudofloppy wieder verloren.

Es ist unmöglich, alle RSX-Befehle hier zu erläutern. Deshalb haben wir sie nur dem Namen nach aufgezählt, zumal man ja bei einiger Phantasie schon aus dem Namen auf die Tätigkeit schließen kann. Alle Befehle, die es ohne RSX-Balken auch im Normal-Basic gibt, wurden in ihrer Funktion lediglich auf alle Bänke ausgeweitet.

Bank-BASIC

IBANK, IBASIC, IBOS, ICALL, ICOMMON, IDEV, IGSUB, IGOTO, IID, ILIST, ILOAD, IMON, INEW, IPEEK, IPOKE, IRAMCLOSE, IRAMFIELD, IRAMOPEN, IRAMREAD, IRAMWRITE, IRECORDS, IRUN, IRETURN, ISAVE, ISCREENS, ISCREEN.IN, ISCREEN.OUT, ISPOOL.ON, ISPOOL.OFF, IVIDEO.ON, IVIDEO.OFF.

BASIC-Grafikbefehle

IFAST, IFRAME, IGHAR, IGPAPER, IGPEN, IMASK, ISLOW, IUNMASK.

Die VIDEO-SCREEN-Befehle und die RAMdisk-Befehle können nicht gleichzeitig benutzt werden. Sie belegen den gleichen Speicher. Die Grafikbefehle wurden schon zusammen mit weiteren im CPC-Magazin 12/85, S. 7 als Bestandteil des VDOS-2.0 EPROMs beschrieben. Das beschriebene VDOS ist allerdings eine Vorablieferung gewesen, denn an der endgültigen Version wird noch entwanzt. Wie wir aber "aus gewöhnlich

gut unterrichteten Kreisen", nämlich dem VORTEX-Vertrieb, erfahren haben, soll der IC ab dem 15.11.85 verfügbar sein.

Noch ein nützlicher Hinweis am Rande. Sie werden sich auch schon geärgert haben, wenn noch 300 KB auf der VORTEX-Diskette frei und die Directory voll waren. Da man jetzt genug Speicher zur Verfügung hat, liegt es nahe, das Inhaltsverzeichnis auf 128 oder 256 Einträge zu erweitern. Mit der Diskette zum Dienstprogramm PARA.COM hat VORTEX die Programme DIR64.COM, DIR128.COM und DIR256.COM ausgeliefert. Diese laufen allerdings nicht unter 64 K-CP/M.

Der Monitor

Das Bonbon für den Maschinenprogrammierer kommt jetzt zum Schluß: der Monitor. Er ist ROM-resident und läßt sich von der Tastatur her ebenso aufrufen wie vom BASIC aus. Sie können damit die in BASIC-Programme eingebundenen MC-Teile während des Ab-

12 Kommandos

laufes kontrollieren und landen nach der Rückkehr im BASIC genau wieder hinter der Stelle, an der Sie es verlassen haben. Der Monitor hat zwölf Ein-Zeichen-Kommandos, die wir in Stichworten kurz vorstellen wollen:

- A Assembler. Mnemonics rein – Übersetzung im Speicher, fortlaufend an richtiger Stelle ab der Ladeadresse.
- B Breakpoints setzen; bis zu acht Stück gleichzeitig. Der Breakpoint übergibt die Steuerung dem Monitor. Weiter mit ENTER oder Abbruch mit ESC.
- D Dump eines Speicherbereiches. Hexadezimal und in der gleichen Zeile das ASCII-Zeichen. 256 Bytes auf einmal.
- F Füllen eines Bereiches mit beliebigem 8-Bit-Wert, also Codes zwischen 0 und 255.

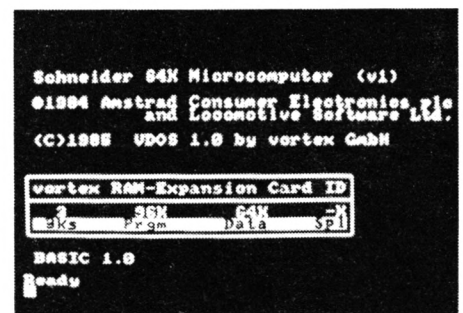


Bild 5. Der Bildschirm nach dem Einschalten

- I Input MC. Laden einer Binärdatei von Diskette oder Kassette in den originalen Speicherbereich. Dabei werden Ladeadresse, Länge und eventuell die Startadresse angezeigt.
- L Listet einen Speicherbereich in Form von Z80-Mnemonics, Länge kann angegeben werden, Zwischenstop mit ESC.
- M Memorycopy. Kopiert einen Bereich in einen anderen. (Keine Angabe, darüber ob intelligentes Kopieren möglich ist, das heißt, ob sich die Bereiche überlappen dürfen.)
- O Output MC. Das Gegenteil von I. Laden auf Diskette und Kassette möglich.
- P Printschalter. Nach einmaligem Drücken von "P" wird jede Schirmausgabe auf dem Drucker mitprotokolliert; auch Disassemblierungen und Dumps. Ein zweites "P" schaltet den Drucker wieder ab. Sehr nützlich!
- R Registeranzeige und -änderung. Der zweite Registersatz (Alternativregister) ist nicht veränderbar.
- S Speicherinhalt fortlaufend anzeigen und ändern. Dadurch wird die Eingabe kurzer MC-Teile ohne Lader möglich.
- T Tasten durch ein MC-Programm. Schrittweises Abarbeiten mit jeweiliger Registerangabe, nicht für ROMs.

Noch gibt es keine Basicprogramme auf dem Markt, welche die Vorteile der VORTEX-Speichererweiterungen nutzen. Im Augenblick profitieren davon allein die CP/M-Programme. TURBO-PASCAL, FORTRAN 80, WORDSTAR, MULTIPLAN, dBASE II usw. laufen hervorragend. Andere Quasi-Standardprogramme wie zum Beispiel TASWORD und TASPRINT müssen erst noch angepaßt werden, sei es vom Hersteller oder von begeisterten Chippis. Die wahren Vorteile werden wir erst erkennen, wenn einige Zeit vergangen ist und wir

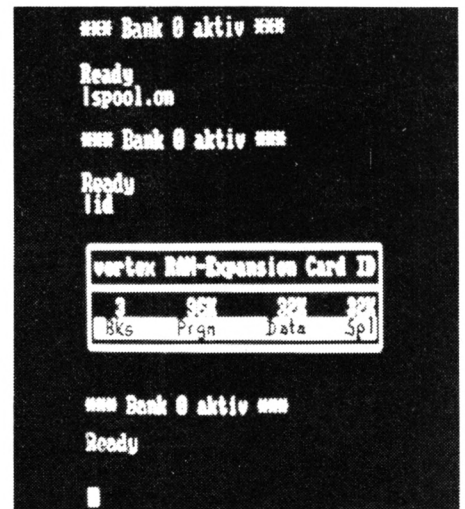


Bild 6. Bildschirm nach IBOS, ISPOOL.ON und IID

uns an die neue BASIC-Art gewöhnt haben und alle Tricks kennen! Es gibt viel zu tun Freunde, nichts wie RAM!

Prof. W. Tosberg

WANTED

Originelle Anwendungen für den Schneider CPC gesucht!

Wir suchen nach außergewöhnlichen Anwendungen, um einem größeren Publikum die Einsatzmöglichkeiten des CPC zugänglich zu machen. Die Palette kann von lustig, clever bis zu semiprofessionell reichen. Auch einfache, aber wirkungsvolle Lösungen sind willkommen. Nehmen Sie Kontakt mit uns auf, schriftlich oder telefonisch und berichten Sie uns über Art und Umfang Ihrer Lösung. Wir werden dann Ihr Projekt im CPC-Magazin vorstellen.

CPC-Magazin
Postfach 1640
D-7518 Bretten
Telefon 0 72 52 / 4 29 48



**Setzen Sie sich nicht
zur Ruhe -
werden Sie aktiv!**

Computerzeitschriften leben auch von der Aktivität ihrer Leser. Deshalb fordern wir alle auf, bei uns mitzuarbeiten.

Schicken Sie uns Ihre selbstgeschriebenen Programme. Wir suchen gute Spielprogramme, Tips & Tricks, Assemblerlistings und Hilfsroutinen.

Wenn wir nach einer Überprüfung der Meinung sind, daß Ihr Programm gut ist, werden wir es im »CPC - Magazin« oder in »Computer - Kontakt« veröffentlichen. Gegen ein angemessenes Honorar versteht sich.

CPC Magazin
Für alle Schneider Computer

Für zwischendurch...

Geschützte Programme kopieren

Wenn ein BASIC-Programm kopiert werden soll, das mit SAVE "Programmname", P abgespeichert worden ist, wird man auf erhebliche Probleme stoßen, wenn kein Kopierprogramm wie Speedmaster oder Syclone zur Verfügung steht. Dieses Problem tritt aber nicht nur beim Kopieren auf, sondern auch wenn man ein eigenes Programm auf die oben geschilderte Weise abgespeichert hat und dieses ändern will. Eine Lösungsmöglichkeit bieten die drei nachfolgenden POKE-Befehle, die in der angegebenen Reihenfolge eingegeben werden müssen.

POKE &AC02, &90

POKE &AC03, &C0

POKE &AC01, &C3

Danach kann das betreffende BASIC-Programm in der gewohnten Weise geladen und anschließend gelistet werden.

Speichervergrößerung

Wenn der Speicherplatz einmal sehr knapp wird, kann man mit dem Befehl SYMBOL AFTER 255 zusätzliche 120 Byte zur Programmierung freisetzen. Dabei handelt es sich dann um den Speicherbereich der frei definierbaren Zeichen. Es ist danach also nicht mehr möglich, frei definierbare Zeichen im Programm zu verwenden.

Das unbekannte RSX-Kommando

Probieren Sie bei Ihrem CPC doch einmal den RSX-Befehl IBASIC aus. Dieses Zusatzkommando bewirkt einen System-RESET. Der Unterschied dieses System-RESETs im Vergleich zu denen, die mit CALL&0 oder durch Drücken der Tastenkombination "SHIFT + CTRL + ESC" hervorgehoben werden, besteht darin, daß der bekannte Bildschirmvorspann fehlt, der immer nach dem Einschalten des Computers erscheint.

Auf dem Bildschirm steht nur "BASIC 1.0" und darunter "Ready". IBASIC läßt sich deshalb besonders gut in Programme einbauen, die bei Fehlbedienung gelöscht werden sollen.

Uwe Adam

Bildschirmmanipulation

Mit Hilfe des CRTC (Catode Ray Tube Controller oder zu deutsch Kathodenstrahlröhrensteuerung, auch unter dem Namen Video Chip bekannt) kann man die Bildschirmausgabe des CPC beeinflussen. Damit ist es dann möglich, die 18 Register des HD-6845 Video Chip vom BASIC aus zu manipulieren. Nehmen wir z. B. das Register Nr. 4. Durch Manipulationen am Inhalt dieses Registers ist es möglich, den Bildschirminhalt vertikal laufen zu lassen. Das hört sich zunächst sehr banal an, aber wenn man z. B. Laufschrift programmieren will, ist die Bildschirmmanipulation ideal.

Man muß hier also keine programmiertechnischen Klimmzüge machen. Außerdem bietet diese Methode einen weiteren nützlichen Nebeneffekt. Der Bildschirm wird

nicht nur innerhalb seiner Umrandung (Border) bewegt, sondern über seine ganze Höhe. Die Umrandung bleibt nur noch auf den Seiten stehen. Durch eine gute Farbgebung kann man so überraschende Effekte erzielen.

Wem die Darstellung der Laufschrift noch nicht genug ist, der kann durch Änderung der Werte hinter dem Komma in den Zeilen 80 und 90 des nachfolgenden Programms mit den diversen Möglichkeiten der CRTC-Manipulation experimentieren. Zeile 80 bezeichnet das CRTC-Register, Zeile 90 den Registerwert.

10 BORDER 10

20 INK 1, 3

30 INK 0, 6

40 CLS

50 PRINT

60 PRINT" Dies ist ein Test"

70 PRINT" Er demonstriert Laufschrift"

80 OUT &BC00, 4

90 OUT &BD00, 2

CPC — SOFTWARE — CPC

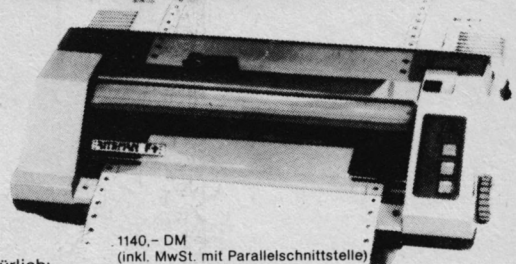
Suchen Sie professionelle Anwenderprogramme für Ihren CPC?

Info anfordern (1.- DM Briefmarken) bei:
UNICOM-Computertechnik, Postfach 21 04 05
 4100 Duisburg, ☎ 02 03 / 33 73 83,
 Händleranfragen erwünscht!

wiesemann 
 MIKROCOMPUTERTECHNIK
 ... Ihr Drucker-Spezialist!

Der Drucker, der gebaut ist wie das Papier, das er verarbeitet:
Flach.

Riteman F+



1140,- DM
 (inkl. MwSt. mit Parallelschnittstelle)
 Natürlich:
 Fx-kompatibel, Einzelblatt oder Endlospapier, Proportionalsschrift, echte Unterlängen, Grafik (auch 1:1, keine Verzerrungen), NLQ und vieles mehr. Riteman F+ kann auch steckerfertig (also inkl. aller Kabel, Stecker und ggf. Interface) z.B. für die folgenden Computer geliefert werden:

Apple, Atari, CBM, C64, HP-IB, IBM-PC, ITT, Kiss, Schneider, Tandy

Winchenbachstr. 3-5 **Telefon** **Telex**
 5600 Wuppertal 2 (02 02) 50 50 77 8 591 656 wwd

DIE NEUE ZEITSCHRIFT FÜR SCHNEIDER-FANS



TOP AUTOREN

schreiben

Richard Leinfellner

Programmierer des Spieles Hexenküche
von Palace Software

Thomas Binzinger

Programmierer von dem deutschen
Grafik-Text-Adventure
DARK POWERS

Frank Thielen
Thomas Barndt

Entwickler des Super BASIC-
Compilers mit Fließkommaarithmetik

TOP BEITRÄGE

3D-PLOT

TEAM-ANIMATOR

Kalorienwächter

Mineralien bestimmen

Optimizer

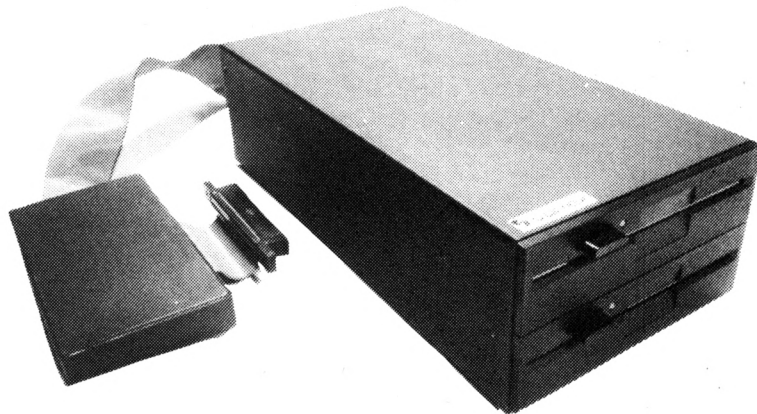
Nähere Info?
Probeheft?

Verlag **COMPUTER TEAM** Joachim Günster
Mühlenstraße 12, 5431 Boden
oder per **HOTLINE**
sofort und unverbindlich

0 26 02 / 6 00 80

Floppykurs

Teil 2



Die bekannteste Form der externen Datensicherung ist immer noch die sequentielle Datei. Obwohl die Diskettenstationen wesentlich schnellere und komfortablere Möglichkeiten bieten, unterstützt das Schneider BASIC ausschließlich die sequentielle Verwaltungsform, wobei die Vorzüge dieser Datei in der äußerst einfachen Handhabung und Programmierung bestehen.

Was ist eine Datei?

Allgemein könnte man sagen, daß eine Datei eine Liste von beliebigen numerischen Daten ist. Diese Aussage ist allerdings nur bedingt richtig. Nehmen wir als Beispiel eine Adressdatei eines Computer-Clubs: In dieser Datei befinden sich alle Anschriften der Mitglieder. Die Adresse eines Nicht-Mitglieds hat selbstverständlich innerhalb dieser Datei nichts zu suchen, obwohl es auch nur eine Anschrift ist. Ein Eintrag innerhalb einer Datei muß deshalb immer einen logischen Bezug zur gesamten Datei besitzen. Eine Datei ist also eine Sammlung von logisch gleichartigen Daten (Einträgen)!

Bleiben wir bei dem Beispiel der Mitgliederdatei. Nehmen wir an, daß der Club insgesamt 100 Mitglieder hat. Dann besteht die Mitgliederdatei aus insgesamt 100 Einträgen. Ein solcher Eintrag nennt sich DATENSATZ. Jeder Datensatz besteht wiederum aus einer bestimmten Anzahl von Feldern, die DATENFELDER genannt werden. Ein Datensatz der Mitgliederdatei könnte z. B. aus den Datenfeldern

Datenfeld 1: Name des Mitgliedes
 Datenfeld 2: Anschrift des Mitgliedes
 Datenfeld 3: Beitrag pro Monat

bestehen. Die Art eines Datenfeldes ist abhängig vom Inhalt des Feldes. Das Datenfeld 3 (Beitrag) ist ein numerisches Feld, da hier ein Wert, mit dem man rechnen kann, eingetragen wird. Die Anschrift des Mitgliedes (Datenfeld 2) ist ein alphanumerisches Feld: Es können beliebige Zeichen/Ziffern untergebracht werden. Zusätzlich gibt es noch die Alphafelder, die nur Buchstaben enthalten dürfen. Zu guter Letzt besteht ein Datenfeld außerdem aus einer bestimmten Anzahl von STELLEN (Zeichen).

Die sequentielle Datenverwaltung basiert selbstverständlich auch auf dem allgemeinen Dateiaufbau. Sequentiell (eng. sequal = Folge, Fortsetzung) bedeutet nichts anderes, als daß die Einträge (Datenfelder bzw. Datensätze) in einer bestimmten Reihenfolge hintereinander folgen. Die sequentielle Daten-

Sequentielle Datei

sicherung ist auf die frühe Verwendung von Magnetbandaufzeichnungsmaschinen zurückzuführen. Auf einem Magnetband können die Daten nicht anders abgelegt werden als hintereinander.

Sehen wir uns die Mitglieder-Datei einmal in einer sequentiellen Form an: Nach dem ersten Eintrag folgt der zweite, bis das Ende der Datei erreicht ist (Eintrag 100). Damit auf den letzten Eintrag zugegriffen werden kann, müssen alle Einträge vom Anfang der Datei an nacheinander überlesen werden. Hier wird auch schon der große Nachteil der sequentiellen Datenverwaltung deutlich: Ein Überlesen von vielen Daten ist sehr zeitaufwendig. Der notwendige schnelle Zugriff auf große Datenmengen ist damit nicht gewährleistet.

Die Anwendungen von sequentiellen Dateien beschränken sich damit auf kleinere Datenmengen, die zur weiteren Bearbeitung komplett in den Rechner geladen und anschließend komplett wieder zurück in die Datei (auf die Diskette/Kassette) geschrieben werden. Da sich aber nach dem Laden der Datei alle Daten (Datensätze/Datenfelder) im Rechner befinden, ist der Zugriff auf diese Daten extrem schnell.

So wird eine Datei beschrieben

Der erste und wichtigste Schritt zum Anlegen einer sequentiellen Datei ist das Eröffnen eines Datenweges vom Computer zum externen Speicherme-

dium (Diskettenstation oder Datenrekorder). Hier müssen wir zwei Datenrichtungen unterscheiden:

Computer – externer Speicher
 externer Speicher – Computer

Das Schneider-BASIC macht es uns hierbei sehr einfach. Die Befehle OPENIN (OPEN Input = Daten von der Diskette/Kassette holen) und OPENOUT (OPEN OUTput = Daten zur Diskette/Kassette senden) erledigen die gesamte Arbeit. Damit wird zwar ein Datenweg zwischen Computer und Diskette eröffnet, aber noch ist nicht bestimmt, in welche Datei die Daten später geschrieben bzw. aus welcher sie gelesen werden sollen. Den Befehlen muß deshalb ein gültiger Filename nachgestellt werden, der den Namen der zu bearbeitenden Datei wiedergibt:

OPENOUT "DATEN.DAT"

Dieser Befehl eröffnet eine neue Datei mit dem Namen DATEN.DAT zum Schreiben. In der Directory wird ein neuer Eintrag angelegt. Besteht schon ein File unter dem angegebenen Namen, wird das alte File in DATEN.BAK umbenannt.

OPENIN "DATEN.DAT"

Hierzu wird versucht, die Datei DATEN.DAT zum Lesen zu öffnen. Dazu muß natürlich diese Datei bereits auf der Diskette existieren. Ist dies nicht der Fall, erfolgt die Fehlermeldung "DATEN.DAT not found".

Nachdem wir soweit gekommen sind, legen wir jetzt die erste Datei auf der Diskette an.

1. Datenweg zum Beschreiben einer Datei eröffnen:

10 OPENOUT "TEST.DAT"

2. Die Datei wird mit 500 Datensätzen beschrieben, die aus jeweils einem numerischen Feld bestehen.

```

20 FOR wert=1 TO 500
30 PRINT#9, wert      :REM Diskette
40 PRINT wert        :REM Bildschirm
50 NEXT wert

```

Das Schreiben in die Datei findet grundsätzlich über den Datenweg Nummer 9 (#9) statt. Dieser Datenweg führt automatisch in die über OPENOUT eröffnete Datei! Die Schleife printet die Zahlen 1 bis 500 über den Datenweg #9 in die Datei TEST.DAT, die sich dann auf der Diskette befindet.

Noch kurz ein Wort zu PRINT: Wie auch bei der PRINT-Ausgabe auf dem Bildschirm, folgt nach dem auszugebenden Wert immer ein zusätzliches Carriage Return CHR\$(13), wodurch der Cursor in die nächste Zeile springt. Dieses Carriage Return ist Bestandteil des Datenfeldes! Hinter jedem Wert steht in der Datei also ein CHR\$(13).

Ist die Datei komplett auf der Diskette gesichert worden, muß der Datenweg zur Diskette wieder geschlossen werden. Das ist besonders wichtig, aber nicht nur, um diesen Datenweg für einen anderen Transfer wieder frei zu machen. Die Befehle zum Schließen eines Datenweges lauten entsprechend den Eröffnungsbefehlen: CLOSEIN (CLOSE INput = Schließen des Datenwegs, der mit OPENIN eröffnet wurde) und CLOSEOUT (CLOSE OUTput = Schließen des Datenwegs, der mit OPENOUT eröffnet wurde).

```

60 CLOSEOUT
70 END

```

Starten Sie jetzt das Programm und beobachten Sie genau das Verhalten der Diskettenstation. Nach dem Start des Programms läuft zunächst die Diskette an. Jetzt wird ein neuer Directory-Eintrag angelegt. Auf dem Bildschirm kann man deutlich verfolgen, welche Daten in die Datei geschrieben werden. Aber halt: Ungefähr beim Wert 100 ist festzustellen, daß die Diskettenstation nicht mehr läuft. Genau beim Wert 308 stoppt das Programm und die Diskette läuft wieder an. Nach einer kurzen Unterbrechung geht es dann weiter. Woher kommt nun dieses merkwürdige Verhalten?

Die Daten werden nicht direkt über den Datenweg in die Datei geschrieben, sondern zunächst im Computer zwischengespeichert. Erst wenn dieser Zwischenspeicher gefüllt ist, wird er komplett in die Datei geschrieben. Der Zwischenspeicher besitzt eine Größe von 2 kByte. Da gleichzeitig eine Schreib- und Lese-Datei eröffnet sein kann, besteht der gesamte Speicher aus 2 mal 2 kByte = 4 kByte. Vielleicht ist Ihnen auch jetzt die Wichtigkeit des CLOSEOUT-Befehls deutlich geworden. Mit ihm wird der restliche Inhalt des Speichers in die Datei geschrieben, erst

dann schließt er den Datenweg. Vergessen Sie den CLOSEOUT-Befehl, so wird die Datei mit Sicherheit nicht vollständig auf die Diskette geschrieben.

Datei lesen

Aus der Directory (CAT) ist zu entnehmen, daß sich ein 4 kByte großes File mit dem Namen TEST.DAT auf der Diskette befindet. Das ist unsere erste Datei! Lassen Sie uns diese Datei jetzt wieder in den Rechner einlesen:

```

10 OPENIN "TEST.DAT"
20 FOR zaehler = 1 TO 500
30 INPUT#9, wert
40 PRINT wert
50 NEXT zaehler
60 CLOSEIN
70 END

```

Das Programm dürfte inzwischen keine Schwierigkeiten mehr machen. Mit dem INPUT-Befehl lassen sich die Daten wieder aus der Datei lesen. Der INPUT-Befehl arbeitet hier einwandfrei, da ja nach einem Wert in der Datei noch zusätzlich ein Carriage Return (CHR\$(13)) steht. Dieses Carriage Return ist (neben dem Komma) für die INPUT-Anweisung immer die Kennzeichnung, daß hier das Datum (Einzahl von Daten) endet (genau wie bei der Eingabe mit Tastatur die ENTER-Taste). Das Programm gibt alle Daten der Datei wieder auf dem Bildschirm aus. Auch hierbei fällt sofort auf, daß die Daten zunächst in den Zwischenspeicher eingelesen und erst dann ausgegeben werden. Das geht bei der Diskette allerdings so schnell, daß es fast nicht auffällt. (Beachten Sie wieder den Wert 308.)

Meistens nützt es aber nicht viel, die Daten einfach nur auf dem Bildschirm auszugeben. Vielmehr sollen die Daten bearbeitet (geändert) und anschließend zurück auf die Diskette gesichert werden. Dazu muß der Computer sich die Daten zunächst einmal merken. Die einfachste Methode ist es, alle Daten in einem Array einzulesen.

```

10 DIM wert(500)
20 OPENIN "TEST.DAT"
30 FOR zaehler= 1 TO 500
40 INPUT#9, wert (zaehler)
50 NEXT zaehler

```

Im weiteren Verlauf des Programms können Sie die Daten beliebig ändern. Sind alle Daten bearbeitet, schreiben Sie die gesamte Datei (das Array wert(1) bis wert(500)) zurück auf die Diskette.

Diese Vorgehensweise (Datei in ein Array laden, Datei bearbeiten, Datei zurückschreiben) ist die bequemste und sinnvollste Methode, mit sequentiellen Dateien zu arbeiten. Der große Nachteil besteht in der Tatsache, daß die Datei nicht größer werden darf, als der Computer an Speicherplatz bietet.

End of File

Bisher war es noch sehr einfach, die Größe einer Datei festzustellen. Wir wußten, die Datei hat genau 500 Datensätze (hier Datensatz = Datenfeld), nicht mehr und nicht weniger. Was ist aber, wenn das nicht so ist?

Beim Sichern eines Programms jeder Art und auch beim Anlegen einer sequentiellen Datei wird als letztes (Steuer-) Zeichen das sogenannte "End of File" (&1A = 25 dezimal) in das File geschrieben (beim Programmfile folgen vorher 3 Nullbytes. Das bedeutet, daß das letzte Zeichen der Datei TEST.DAT ein CHR\$(25) ist. Immer dann, wenn der Computer bei einer Leseoperation auf ein "End of File" trifft, setzt er die Systemvariable EOF auf -1 (wahr), die sonst auf 0 liegt. Anhand dieser Systemvariablen läßt sich dann einfach und problemlos jedes Datenende feststellen.

```

10 OPENIN "TEST.DAT"
20 INPUT#9, wert
30 PRINT wert
40 IF EOF <> -1 THEN 20
50 CLOSEIN

```

Solange die Systemvariable EOF ungleich -1 ist, folgt ein weiteres Datum. Wird versucht, über das Dateiende hinaus ein weiteres Datum einzulesen, erfolgt die Fehlermeldung "EOF met". (Die Zeile 40 kann auch durch IF NOT EOF THEN... ersetzt werden.)

Der Zwischenspeicher

Beim Datentransport zwischen Computer und externer Speichereinheit werden, wie wir festgestellt haben, die Daten zunächst in einem Speicherbereich des Computers zwischengespeichert. Da gleichzeitig ein Ausgabe- und Eingabekanal geöffnet sein kann, muß der Zwischenspeicher eine Größe von 4 kByte besitzen. So wie im Programm ein Datenkanal (OPENIN, OPENOUT) eröffnet wird, werden 4 kByte RAM-Speicher für die Ein- und Ausgaberroutinen reserviert. D. h., es werden 4 kByte vom verfügbaren Speicher abgezweckt, und zwar immer die letzten 4 kByte des verfügbaren freien RAM-Speicherbereichs. Mit einem einfachen Test läßt sich das feststellen. Führen Sie zunächst einen Reset (CTRL/SHIFT/ESC) aus und starten dann das folgende kleine Programm:

```

10 PRINT HIMEM
20 OPENIN "TEST.DAT"
30 PRINT HIMEM
40 CLOSEIN

```

Zeile 10 gibt die Adresse des letzten freien Speicherplatzes vor dem Eröffnen eines Datenweges auf dem Bildschirm aus. Es sollte auch bei Ihnen die Adresse 42619 sein. Anschließend wird ein Datenweg eröffnet und wiederum die

letzte freie Adresse ausgegeben. Jetzt erhalten wir aber die Adresse 38523 und das sind genau 4 kByte weniger.

Aber gerade die letzten 4 kByte des Speichers sind sehr problematisch, da sich hier die Inhalte der Stringvariablen befinden. Der Computer muß sich diese Inhalte aller benutzten Variablen natürlich im RAM merken, die der CPC grundsätzlich vom RAM-Ende abwärts ablegt. Lassen Sie uns das noch einmal prüfen.

```
10 FOR i=0 TO 10
20 a$=a$+CHR$(65+i)
30 NEXT i
40 vp=@a$
50 adr=PEEK(vp+1)+256*PEEK(vp+2)
60 PRINT "Adresse :";adr
```

Mit Hilfe des Variablenpointers (siehe Folge 1) läßt sich die Adresse, wo der Inhalt liegt, sehr leicht ermitteln. Als Ergebnis erhalten wir die Adresse 42543, also ganz oben im freien RAM-Speicher! Erweitern Sie jetzt das obige Programm um folgende Zeilen und starten es erneut:

```
70 OPENIN "TEST.DAT"
80 adr=PEEK(vp+1)+256*PEEK(vp+2)
90 PRINT "Adresse :";adr
100 CLOSEIN
```

Nach dem Eröffnen eines Datenkanals liegt der Stringinhalt der Variablen a\$ ab der Adresse 38513!?

Immer dann, wenn ein Datenkanal eröffnet wird, müssen die Stringinhalte, die sich in den letzten 4 kByte des freien RAMs befinden, zunächst einmal "gerettet" werden. Das ist klar, denn sonst würden diese Inhalte durch die zwischengespeicherten Informationen zer-

stört werden. Dazu verschiebt der Computer alle Stringinhalte um (etwa) 4 kByte nach unten. Somit wird Platz für den Zwischenspeicher geschaffen. Nun stellen Sie sich aber vor, daß hier etliche Stringinhalte liegen. Jetzt wird das Verlegen des Inhalts schon eine sehr zeitaufwendige Sache. Auch können sich hier nicht mehr benutzte Stringinhalte befinden, die vollkommen überflüssig sind. (Z. B. werden einer Variablen nacheinander verschiedene Inhalte zugewiesen. Trotzdem liegen dann die alten Inhalte immer noch im Speicher, der Stringdescriptor zeigt allerdings nur auf den aktuellen Inhalt.)

Diese alten Inhalte brauchen natürlich nicht mit verlegt zu werden. Beim Verschieben findet deshalb gleichzeitig ein Aufräumen (Entmüllen, Garbage Collec-

Garbage Collection

tion) der Stringinhalte statt. Wenn Sie sich diesen gesamten Vorgang einmal bildlich vorstellen, können Sie den riesigen Aufwand dieser Aufräum- und Verschieberoutine ermesen. Dieser Vorgang kann je nach belegten Stringvariablen bis zu 10 Minuten (!) dauern. Während des Aufräumens können Sie mit dem Computer nichts anderes machen. Sogar ein Abbruch mit der ESCAPE-Taste ist nicht möglich! Das ist wirklich sehr lästig.

Dieses Problem läßt sich umgehen, indem im Programm vor der ersten String-

zuweisung (also in den ersten Programmzeilen) ein beliebiger Datenweg (Dummy) eröffnet wird. Jetzt geht das Verschieben noch blitzschnell, da ja nichts aufzuräumen ist. Gleichzeitig wird die obere Speichergrenze (HIMEM) um 4 kByte nach unten verschoben. Setzen wir nun das Speicherende auf diesen Wert (HIMEM), sind die oberen 4 kByte für die Belegung durch Stringinhalte gesperrt:

```
1 OPENOUT "DUMMY.DAT"
2 MEMORY HIMEM-1
3 CLOSEOUT
```

Gleichzeitig merkt sich der Rechner die Startadresse des geschützten Zwischenspeichers. Wird jetzt innerhalb des Programmes ein Datenkanal eröffnet, findet der Rechner immer einen freien 4 kByte großen Speicherraum vor, so daß ein Aufräumen nicht mehr notwendig ist.

Bei allen Programmen, die mit einem Datenkanal und vielen Stringvariablen arbeiten, sollte zu Beginn ein Zwischenspeicher angelegt und geschützt werden.

Im nächsten Heft

Das nächste Mal setzen wir die bisherigen Kenntnisse in die Praxis um und erstellen ein richtiges Dateiprogramm. Dabei werden die Begriffe Datensatz und Datenfeld mit Inhalt gefüllt und die Unterschiede zwischen PRINT, WRITE, INPUT und LINE INPUT untersucht.

Manfred Walter Thoma

PHASE 4

**Textverarbeitung
Tabellenkalkulation
Rechnen im Text
Adressenverwaltung**

**TEXTPROGRAMM – PHASE 4
– DIE NEUE GENERATION**

- ★ Deutsches Programm, deutsches Logo
- ★ In Maschinensprache geschrieben
- ★ Umfangreiches Handbuch

PHASE 4 MIT MAIL MERGE:

- ★ 500 Datensätze
- ★ 40 Sortierkriterien
- ★ Individuelle Massenbriefe
- ★ Dateiverwaltung (z.B. Bibliotheken, Karteien, Register usw.)
- ★ Menügesteuert
- ★ Etikettendruck
- ★ ... und vieles mehr!

PHASE 4:

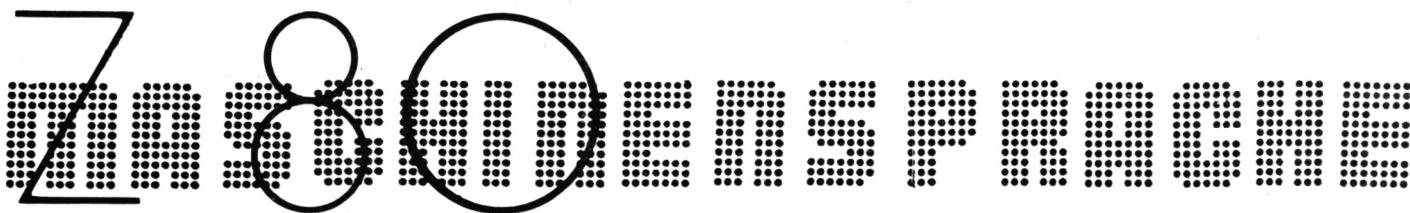
- ★ Zeichen löschen u. einfügen
- ★ Zeile(n) löschen u. einfügen
- ★ Absätze löschen u. einfügen
- ★ Begriffe suchen u. ersetzen
- ★ Wort oder Zeilenrest löschen
- ★ Zeilen aufbrechen u. anschließen
- ★ Bausteinverwaltung
- ★ Darstellungsbreite bis 240 Zeichen je Zeile
- ★ Fließtexteingabe
- ★ Randausgleich, Blocksatz
- ★ Linker Rand im Text frei definierbar
- ★ Gängige Druckertreiber sind installiert
- ★ Taschenrechnerfunktion u. Formelrechnung
- ★ Tabellenkalkulation im Text
- ★ Seitenumbruch, Zeilenanzahl frei wählbar
- ★ ... und vieles mehr!

Erhältlich auch in Computerfachgeschäften und den Fachabteilungen der Kaufhäuser.
PHASE 4: DM 225,72, PHASE 4 inkl. MAIL MERGE: DM 282,72 zzgl. Verpackung und Versandgebühr (9,50 DM)

SOFTWAREAGENTUR HEYNS
LIMBECKER PLATZ 7 · 4300 ESSEN 1 · TEL. 02 01 / 22 64 17

Für ganz Eilige:
02 01 / 22 64 17

SOFTWAREAGENTUR HEYNS
LIMBECKER PLATZ 7
4300 ESSEN 1



Teil 2

Grundsätzlich ist Maschinensprache dem Basic ähnlich. Auch hier gibt es Variablen und Befehle, nur sind diese weniger effektiv. Um z. B. den Basicbefehl PRINT in Maschinensprache zu schreiben, ist schon ein ziemlich umfangreiches Programm notwendig. Trotzdem kann man sagen: Wer Basic beherrscht, kann auch MC lernen.

In MC kann man Variablen keine eigenen Namen geben. Außerdem stehen nicht sehr viele zur Verfügung. Die wichtigsten Register sind A, B, C, D, E, H und L. Jedes Register ist 8 Bit breit und kann damit Werte von 0 – 255 annehmen (natürlich nur ganze Zahlen). Die Register BC, DE und HL lassen sich außerdem noch zu 16 Bitregistern zusammenfassen, so daß die Doppelregister dann Werte zwischen 0 und 65535 annehmen können. Das zuerst genannte Register ist Highbyte, das heißt, sein Inhalt wird mal 256 genommen, und das zuletzt genannte ist Lowbyte.

Die Register

Eines der wichtigsten Register überhaupt ist der Accumulator (Register A). Mit ihm kann man alle arithmetischen Operationen durchführen, während dies bei den anderen Registern nur bedingt möglich ist. Daneben gibt es noch die 16 Bit Indexregister IX und IY. Wichtig ist auch der Programmzähler PC, der die Adresse des augenblicklich abgearbeiteten Befehls angibt, oder die Register SP, R, I. Für das erste Programmieren sind diese aber nicht so wichtig, wir werden später noch detaillierter auf sie eingehen.

Enorm wichtig ist jedoch das Flagregister F. In MC kann man nicht einfach einen bedingten Sprung durch IF... THEN GOTO angeben, man kann nur sagen, wenn dieses Flag gesetzt ist, dann springe. Deshalb wollen wir zuerst die drei wichtigsten Flags behandeln:

CARRY (C)

Das Carryflag wird gesetzt, wenn bei einer Operation ein Über- oder ein Unterlauf eintritt. Wenn man z. B. 112 und 200 addiert, so lautet das Ergebnis 312 und paßt nicht mehr in 8 Bit.

ZERO (Z)

Das Zeroflag wird immer dann gesetzt, wenn bei einer Operation null herauskommt. Wenn man z. B. von 122 die Zahl 122 abzieht, so ist das Ergebnis 0 und das Zeroflag ist gesetzt.

SIGNUM (S)

Das Signumflag ist das Vorzeichenflag. Es gibt den Wert von Bit 7 nach einer Operation an. Anhand dieses Flags kann man sofort prüfen, ob eine Zahl negativ ist.

Die wichtigsten Befehle

LD – Der Ladebefehl

Mit LD kann man z. B. ein Register auf einen bestimmten Wert setzen. Um C auf den Wert 45 zu bringen, reicht der Befehl LD C, 45 aus. Aber auch ein Austausch von Zahlen zwischen den Registern ist möglich. LD A, B lädt den Wert des Registers B in das Register A. Register B selbst bleibt unverändert. Allerdings kann man nicht jedes Register aus jedem laden. Der Befehl LD A, HL existiert logischerweise nicht, da HL ein 16 Bit Register ist und A nur ein 8 Bit Register.

Nun zur indirekten Adressierung: Hier existiert der Befehl LD A, (30000). Was aber bedeutet das? A ist doch nur ein 8 Bit Register und 30000 eine 16 Bit Zahl. Doch beachten Sie den feinen Unterschied: Um die Zahl 30000 sind Klammern, und immer wenn das der Fall ist, so bedeutet das: indirekte Adressierung. Dieser Befehl hat nämlich die Funktion, daß A mit dem Wert geladen wird, der in der Speicherzelle 30000 steht.

Natürlich gibt es auch den Befehl LD (30000), A. So kann man auch Speicherzellen als Variablen verwenden. Selbstverständlich gibt es zu jeder Speicherstelle diese beiden Befehle. Noch ein Beispiel: Der Befehl LD A, (HL) lädt A aus der Speicherzelle, auf die HL zeigt. Also wenn HL 45124 wäre, würde A aus der Speicherzelle 45124 geladen.

INC – Der Erhöhebefehl

Mit INC wird der Operand um eins erhöht. INC A erhöht das Register A um eins. INC (HL) erhöht den Inhalt der Speicherstelle, auf die HL zeigt, um eins. Kommt bei einer solchen Operation 0 heraus, so ist Z gesetzt, wird die Zahl überschritten, ist C gesetzt. Natürlich kann man auch die Doppelregister erhöhen, z. B. INC DE. Allerdings werden die Flags nicht gesetzt.

DEC – Der Erniedrigebefehl

DEC ist das Gegenteil zu INC. Hier wird nicht addiert, sondern 1 subtrahiert. Ansonsten läuft der Befehl genauso ab, wie INC.

JP – Der Sprungbefehl

Mit JP springt man zu einer bestimmten Speicherzelle. JP 30000 würde z. B. den Rechner dazu veranlassen, zur Speicherstelle 30000 zu springen. Hier würde er nun versuchen, sein Programm fortzusetzen. Natürlich muß an dieser Adresse dann auch ein sinnvolles Programm stehen, sonst kommt es eventuell zu einem Absturz.

Selbstverständlich kann man JP auch von einer Bedingung abhängig machen. Soll nur gesprungen werden, wenn das Zeroflag gesetzt ist, lautet der Befehl JP Z, Adresse. Soll der Befehl nur ausgeführt werden, wenn das Zeroflag nicht gesetzt ist, heißt es JP NZ, Adresse. Dasselbe gilt auch für die anderen Flags. Ein JP Adresse hat immer zur Folge, daß der Rechner, ohne auf die Flags zu achten, zur angegebenen Adresse springt.

CALL – Der Unterprogrammaufruf

Mit CALL wird ähnlich wie bei GOSUB ein Unterprogramm aufgerufen. Ansonsten ist der Befehl identisch mit dem Befehl JP.

RET – Return

Mit RET kehrt man von einem Unterprogramm zurück zu der Stelle, von wo es aufgerufen wurde. Es wird dann der Befehl nach dem CALL abgearbeitet. Die CALLs können wie die GOSUBs geschachtelt werden, d. h., ein Unterprogramm ruft ein weiteres auf. Bei jedem CALL wird die Adresse auf einen Stapel gelegt. Bei einem RET wird diese

Adresse zurückgeholt und zum Befehl nach dieser Adresse gesprungen. Befindet man sich im Hauptprogramm, also in keinem Unterprogramm, und es tritt ein RET-Befehl auf, so springt der Rechner zum Basic zurück.

CP – Der Vergleichsbefehl

Mit CP wird eine Zahl oder ein Register oder der Inhalt einer Speicherstelle mit dem Accumulator (Register A) verglichen. Sind beide gleich, wird das Zeroflag gesetzt. Ist der Operand größer als A, wird das Carryflag gesetzt.

Einige Beispiele: CP 0 (Vergleich, ob Accu 0); CP B (Vergleich Accu/B-Register); CP (HL) (Vergleich Accu/Inhalt der Speicherzelle, auf die HL zeigt).

Beim CP-Befehl wird die spezielle Bedeutung des Accus deutlich. Vergleiche können nämlich nur mit dem Accu durchgeführt werden.

Wie sind Maschinenprogramme im Speicher abgelegt?

Ein Maschinenprogramm ist zunächst in normalen Speicherzellen abgelegt. Für ein Maschinenprogramm ab Adresse 40000 müssen wir den Speicherplatz erst einmal mit MEMORY 39999 schützen. Mit diesem Befehl weiß der CPC, daß er nur die Speicherstellen bis einschließlich 39999 benutzen kann.

Jeder Maschinenbefehl hat nun eine bestimmte Nummer. Z. B. hat der Befehl LD A, B die Nummer 120 und der Befehl RET die Nummer 201. Beide Befehle sind 1-Byte-Befehle. Ein Maschinenprogramm, welches das Register A aus dem Register B laden und danach zum Basic zurückkehren würde, hätte dann die folgende Form:

```
LD A, B
RET
```

Wollten wir dieses Programm ab Adresse 40000 ablegen, wären folgende Befehle notwendig:

```
MEMORY 39999
POKE 40000,120 (LD A, B)
POKE 40001,201 (RET)
```

Das erste MC-Programm

Und schon haben wir unser erstes Maschinenprogramm geschrieben. Tippen Sie die obengenannten Befehle ein (ohne die Kommentare in den Klammern) und das Maschinenprogramm befindet sich im Speicher. Es kann nun mit CALL 40000 aufgerufen werden. Versuchen Sie mal: CALL 40000. Daraufhin sehen Sie allerdings nichts. Das ist aber auch kein Wunder, da hier nun ein Register verändert wurde.

Es gibt nun auch noch 2 oder 3-Byte-Befehle. So ist z. B. der Befehl LD A, Zahl ein 2-Byte-Befehl. Der Opcode für LD A, Zahl lautet 62. Hinter dieser 62 muß als nächstes Byte die eigentliche Zahl folgen, mit der wir A laden wollen. Soll ab Adresse 40000 der Befehl LD A, 45 stehen, so wird das durch den Befehl POKE 40000,62 (Opcode für LD A, Zahl) und POKE 40001,45 (die eigentliche Zahl) bewerkstelligt. Dieser Befehl belegt nun die Speicherstellen 40000 und 40001; der nächste Befehl kann also erst ab Adresse 40002 folgen.

Jetzt noch ein Beispiel für einen 3-Byte-Befehl. Angenommen, wir wollen das Doppelregister HL mit dem Wert 14243 laden. Der Opcode für LD HL, Zahl lautet 33. Hinter diesem Opcode muß dann noch die 16-Bit-Zahl folgen. Und hier gibt es eine seltsame Regelung: Immer wenn 2-Byte-Zahlen direkt angegangen werden, wird zuerst das Lowbyte eingespeichert und dann das Highbyte. Bei 14243 ist das Highbyte 55, das Lowbyte ist 163. Der Befehl würde also ab Adresse 40000 folgendermaßen abgespeichert:

```
POKE 40000,33 (Opcode von LD HL, Zahl)
POKE 40001,163 (Lowbyte von 14243)
POKE 40002,55 (Highbyte von 14243)
```

Sie werden sich jetzt sicher fragen: Wie bekomme ich denn nun heraus, welcher Befehl überhaupt existiert und welchen Opcode oder welches Format er hat? Dazu werden wir in einem der nächsten Hefte eine ausführliche Tabelle bringen, in der alle diese Fragen behandelt werden.

Was ist ein Assembler?

Ein Assembler hat die Aufgabe, die Befehle in Zahlen zu übersetzen. Man kann also einen Befehl direkt eingeben, z. B. LD A, 4, und der Rechner übersetzt diesen. Wir wollen für unsere relativ kleinen Programme aber keinen Assembler benutzen, da wohl nicht alle Personen, die durch diesen Kurs MC lernen wollen, über ein Assemblerprogramm verfügen.

Nun aber endlich zu unserem ersten, größeren Maschinenprogramm. Wie Sie vielleicht wissen, befindet sich der Bildschirmspeicher des CPC im Bereich von 49152 bis 65535, ist also 16384 Bytes lang. Unser Ziel soll es sein, diesen Bildschirmspeicher vollständig mit der Zahl 255 zu füllen.

Wir wollen das Programm erst einmal in Basic schreiben, dabei aber nicht alle Möglichkeiten ausschöpfen, sondern uns auf das beschränken, was auch in MC zur Verfügung steht. Hier nun das Programm: Tippen Sie es ein und starten Sie es.

```
10 HL = 49152: REM Start des Bildschirmspeichers
```

```
20 BC = 16384: REM Zähler, 16384 Speicherzellen sollen mit 255 beschrieben werden
```

```
30 POKE HL,255: REM Speicherstelle, auf die HL zeigt, mit 255 füllen
```

```
40 HL = HL+1: REM Zeiger um eins incrementieren
```

```
50 BC = BC-1: REM Zähler um eins vermindern
```

```
60 IF BC <> 0 THEN 30: REM Solange der Zähler >0 ist, weiterfüllen.
```

Dieses Basicprogramm benötigt etwa 100 Sekunden, um den Bildschirm zu füllen.

Jetzt wollen wir dieses Programm in Maschinensprache übersetzen. Die beiden ersten Befehle sind klar, sie lauten in Maschinensprache:

```
LD HL, 49152
LD BC, 16384
```

Anschließend soll in die Speicherzelle, auf die HL zeigt, der Wert 255 geladen werden: LD (HL), 255. Nun soll der Zeiger um eins erhöht und der Zähler um eins vermindert werden:

```
INC HL
DEC BC
```

Jetzt wird es kompliziert: Ist BC <> 0, dann soll wieder zum Füllen gesprungen werden. Die Adresse wissen wir jetzt noch nicht, wir bezeichnen sie erst einmal mit dem Label SCHLEIFE. Hätten wir als Zähler ein 8 Bit Register, wäre dies alles kein Problem, denn wenn dieses Register Null erreicht, wäre das Zeroflag gesetzt und wir könnten als Befehl einfach JP NZ, Schleife nehmen (=springe, solange dieses Register nicht Null ist). Wir müssen aber ein 16 Bit Register benutzen, da 16384 in 8 Bit nicht hineinpaßt. Bei einem 16 Bit Register wird aber beim Decrementieren das Zeroflag nicht beeinflusst. Wir müssen also die Register B und C einzeln auf Null prüfen. Das machen wir so:

```
LD A, B; B in A laden, da nur A verglichen werden kann
```

```
CP 0; Vergleich, ob A gleich Null
```

Ist A jetzt gleich Null, wird das Zeroflag zurückgesetzt und wir können zur Schleife springen, da damit auch B ungleich Null ist und damit auch die Bedingung BC <> 0 gilt.

Das gleiche machen wir jetzt mit dem Register C:

```
LD A, C
CP 0
JP NZ, Schleife
```

Wenn der Rechner jetzt noch nicht zur Schleife gesprungen ist, dann heißt das, daß BC gleich Null ist. Damit ist der ge-

Das vollständige MC-Programm (Kasten 2)

So wird das Programm übersetzt: LD HL, 49152 hat den Opcode 33. Anschließend müssen Low und Highbyte von 49152 eingespeichert werden.

POKE 40000,33 (Opcode von LD HL,Adresse)
 POKE 40001,0 (Lowbyte von 49152)
 POKE 40002,192 (Highbyte von 49152)

Genauso verfahren wir mit dem zweiten Befehl:

POKE 40003,1 (Opcode von LD BC, Adresse)
 POKE 40004,0 (Lowbyte von 16384)
 POKE 40005,64 (Highbyte von 16384)

Nun kommt ab Adresse 40006 der Befehl LD (HL),255. Also ist die Adresse unseres Labels SCHLEIFE 40006. Der Rest des Programmes ist nun etwas kürzer dargestellt:

LD (HL),255	POKE 40006,54 (Opcode von LD (HL), Byte)
	POKE 40007, 255 (Wert, mit dem gefüllt wird)
INC HL	POKE 40008,35 (Opcode von INC HL)
DEC BC	POKE 40009,11 (Opcode von DEC HL)
LD A,B	POKE 40010,120 (Opcode von LD A,B)
CP 0	POKE 40011,254 (Opcode von CP, Byte)
	POKE 40012,0 (Wert)
JP NZ, 40006	POKE 40013,194 (Opcode von JP NZ, Adresse)
	POKE 40014,70 (Lowbyte von 40006)
	POKE 40015,156 (Highbyte von 40006)
LD A,C	POKE 40016,121 (Opcode von LD A,C)
CP 0	POKE 40017,254 (Opcode von CP, Byte)
	POKE 40018,0 (Wert)
JP NZ, 40006	POKE 40019,194 (Opcode von JP NZ, Adresse)
	POKE 40020,70 (Lowbyte von 40006)
	POKE 40021,156 (Highbyte von 40006)
RET	POKE 40022,201 (Opcode von RET)

Unser Programm wäre jetzt vollständig übersetzt. Fassen wir das Ganze noch in ein Basicprogramm:

```
10 MEMORY 39999
20 FOR I=40000 TO 40022: READ A: POKE I, A: NEXT
30 DATA 33, 0, 192, 1, 0, 64, 54, 255, 35, 11, 120, 254, 0, 194, 70, 156, 121, 254, 0, 194, 70, 156, 201
```

samte Bildschirmspeicher gefüllt, unsere Aufgabe ist also erledigt. Mit RET kehren wir zum Basic zurück.

In dem folgenden Kästchen ist das vollständige Programm der besseren Übersicht halber noch einmal geschlossen aufgelistet. (Die Zahlen hinter den Befehlen brauchen wir später noch.)

	LD HL,49152	33,Low,High
	LD BC,16384	1,Low,High
SCHLEIFE	LD (HL),255	54,Byte
	INC HL	35
	DEC BC	11
	LD A,B	120
	CP 0	254,Byte
	JP NZ,SCHLEIFE	194,Low,High
	LD A,C	121
	CP 0	254,Byte
	JP NZ, SCHLEIFE	194,Low,High
	RET	201

Hätten wir jetzt einen Assembler, so könnten wir dieses Programm direkt eingeben, übersetzen lassen und starten. Aber da wir keinen Assembler benutzen wollen, müssen wir das Programm ab Speicherstelle 40000 ablegen und schützen deshalb den Speicher mit MEMORY 39999.

Hinter dem MC-Programm stehen schon die Opcodes der Befehle. Manche Befehle, wie INC HL, DEC BC, RET u. ä. sind 1-Byte-Befehle und können direkt übersetzt werden. Anderen müssen noch Adressen oder 1 Byte angehängt werden (siehe Kasten 2).

Nach dem Eintippen starten Sie das Programm mit RUN und schon befindet es sich im Speicher. Rufen Sie es nun mit CALL 40000 auf. Die Ausführung des Programmes dauert 0.5 Sekunden, d. h., das Maschinenprogramm ist 200 mal schneller als das Basicprogramm. Ein weiterer Vorteil ist der geringe Speicherbedarf. Dieses Programm benötigt nur 23 Byte Speicherplatz, während das Basicprogramm 122 Byte (ohne REMs) verbraucht. Hier ist der Unterschied beim Speicherplatz nicht so eklatant, aber bei anderen Anwendungen macht dies meist schon einiges aus.

Um dieses Maschinenprogramm abzuspeichern, müssen Sie SAVE "Name", b, 40000,23 eintippen. Um es wieder zu laden, geben Sie MEMORY 39999: LOAD "Name" ein.

Im nächsten Heft

Im nächsten Heft werden wir unser Programm dann noch verbessern, sowohl in Bezug auf Geschwindigkeit als auch in Bezug auf Speicherplatz. Auch wird dann wahrscheinlich schon ein Teil der angesprochenen Z80-Tabelle abgedruckt, da diese sehr umfangreich ist.

Andreas Zallmann

THE NEW YORK TIMES SUPER STORY

Sie sind ein kleiner Reporter der New York Times. Eines Tages erhalten Sie einen Anruf: Ein Unbekannter bietet Ihnen eine »Super Story« an. Doch noch während des Gesprächs wird der Mann erschossen. Folgen Sie der Spur des Mannes nach Ägypten und ergattern Sie die Sensationsfotos. Schaffen Sie das, haben Sie gute Chancen, zum Reporter des Jahres gewählt zu werden.



Dieses tolle deutsche Grafik-Adventure ist ein absolutes Muß für alle Schneider-CPC-Besitzer. Am besten bestellen Sie sofort die Kassette zum **Sensationspreis von nur DM 19,80** (zuzüglich DM 5,70 Porto und Versand bei Nachnahme oder ohne Versandkosten bei Vorkasse).

**Exklusiv bei Computer-Kontakt-Software,
Postfach 1640/0, 7518 Bretten**

Nur für CPC 464

CP/M – die neue Perspektive (Teil 2)

Heute werden wir bei unserem CP/M Kurs konkret: Es geht um die Bedienung und Benutzung des Betriebssystems, um die Tastatur und die CP/M-Kommandos.

Der erste und für den Benutzer wichtigste Schritt ist die selbstverständliche Bedienung und Benutzung des CP/M-Betriebssystems. Damit ist das Wissen über die Funktionen und Aufgaben der Tastatur und das problemlose Anwenden der vielen CP/M-Kommandos gemeint. Beginnen wollen wir heute mit dem Booten der Tastatur unter CP/M und den "eingebauten" Befehlen.

Noch ein wichtiger Hinweis!

Bevor wir beginnen, möchte ich noch ein paar Worte über den Aufbau des Kurses verlieren: Er soll nicht nur eine Einführung zur Benutzung des CP/M Betriebssystems und der Programmierung unter CP/M sein, sondern er soll vielmehr langfristig ein Hilfsmittel für Sie darstellen. Dazu entsteht innerhalb dieses Kurses eine Kartei, in der Sie auch später schnell einmal nachschlagen und die wichtigsten Informationen des beschriebenen Befehls entnehmen können. Die Karteikarten finden Sie in den "Kästen", die diesen Kurs begleiten. Fotokopieren Sie diese und schneiden Sie sich aus den Kopien die Karteikarten aus. Heften Sie diese entweder in einem Ordner ab oder kleben Sie die Kopien auf eine feste Karteikarte (DIN A7). In einem kleinen Karteikasten, der mit einem Indexregister versehen sein sollte, lassen sich die Karten dann gut unterbringen. Selbstverständlich können Sie die Kartei mit persönlichen Einträgen ergänzen. So erhalten Sie dann Ihre ganz persönliche CP/M-Kartei.

Die CP/M Bibliothek

Die Karteikarten sind aber Bestandteil des Kurses. Auch wenn Sie sich keine CP/M-Bibliothek anlegen wollen, sollten Sie diese Kästchen unbedingt beachten, da viele Informationen der Karten so aussagekräftig sind, daß auf eine weitere Beschreibung verzichtet wurde. Im Kurs finden Sie dann einen Hinweis auf die entsprechende Karteikarte.

Der erste Schritt: Das CP/M wird geladen!

Das Booten (Laden/Starten) des CP/M-Betriebssystems ist beim CPC-464/664 mit dem Befehl ICPM möglich. Selbstverständlich muß sich die Systemdiskette (Kopie) oder eine Diskette mit den Systemspuren im Laufwerk (A) befinden. Ist dies nicht der Fall, erscheint die Fehlermeldung: Failed to load boot sector – Retry, Ignore or Cancel?

Die eigentliche Fehlermeldung (Failed...) besagt, daß es nicht gelungen ist, den Boot-Sektor der Systemspur zu laden. Daraus ist zu schließen, daß sich entweder kein CP/M auf der Diskette befindet oder daß die Systemspuren defekt sind. Jetzt fragt Sie das System, ob Sie den Vorgang (R)etry: wiederholen, (I)gnore: ignorieren oder (C)ancel: abbrechen wollen. Mit der Betätigung der Tasten R, I oder C wird der Vorgang entsprechend wiederholt, ignoriert oder abgebrochen. Auf diese Meldung werden Sie bei der Arbeit mit CP/M immer dann treffen, wenn eine Aufgabe nicht ordnungsgemäß ausgeführt werden konnte.

Legen Sie jetzt eine Diskette mit den Systemspuren in das Laufwerk und wiederholen Sie mit (R) den Vorgang. Ohne weiteres sollte jetzt der Ladevorgang stattfinden (ca. 4 Sekunden) und die Meldung "CP/M 2.2 – Amstrad Consumer Electronics plc" auf pastellblauem Hintergrund erscheinen. Gleichzeitig sieht man das sogenannte "Prompt-Zeichen" (A<) auf dem Bildschirm. Jetzt besitzt das CP/M-Betriebssystem die Kontrolle über den Rechner und wartet auf eine Eingabe vom Benutzer. Bevor wir uns allerdings mit der Eingabe von Kommandos beschäftigen wollen, muß der Vorgang des Bootens etwas genauer beschrieben werden.

Das Booten

Das CP/M-Betriebssystem befindet sich auf den ersten beiden Spuren (Spur 0 und 1) einer Diskette. In diesen beiden Spuren liegen 4 verschiedene "Teilprogramme" (nennen wir es einmal so):

Spur 0/Sektor 0 :Boot-Sektor
Spur 0/Sektor 1 :Konfiguration-Sektor

Spur 0/Sektor 2–6 :unbenutzt!!
Spur 0/Sektor 7–8 :Anfang CCP
Spur 1/Sektor 0–8 :Ende CCP und BDOS

Der Boot-Sektor wird als erstes geladen. Dieses Programm lädt dann selbstständig die restlichen Programme der Systemspuren. Zunächst den "Konfiguration-Sektor" (lat. configuratio = Gestaltung). Er beinhaltet Informationen zum äußeren Erscheinungsbild des CP/M wie Bildschirmfarbe, Einschaltmeldung, Tastenbelegung usw. Den CCP (Console Command Prozessor) und das BDOS (Basic Operating System) haben wir bereits in der ersten Folge kennengelernt.

Selbstverständlich belegen diese Programme Speicherplatz im RAM des Computers, so daß der freie Arbeitsspeicher (auch Kernspeicher genannt) unter CP/M begrenzt wird. Hier die Adressen, wo die Programme im Speicher liegen. (Anm.: Alle hexadezimalen Angaben werden im folgenden mit einem nachgestellten "H" gekennzeichnet!)

C000H bis FFFFH: Bildschirmspeicher
AD00H bis BFFFH: BIOS (Setzt sich im Disketten-ROM fort.)

9F06H bis ACFFH: BDOS
9700H bis 9F05H: CCP

Weiterhin werden die ersten 256 Bytes des RAM-Speichers für wichtige Systeminformationen genutzt. Dieser Bereich nennt sich BASE- oder ZERO-Page:

0000H bis 00FFH: BASE-Page

Der Arbeitsspeicher

Als freier Arbeitsspeicher steht damit der Bereich 0100H bis 96FFH zur Verfügung. Das sind rund 37,5 kByte. Unter CP/M trägt der freie Arbeitsspeicher den Namen "TPS" (Transient Program Area). Das ist der Bereich, in den später Programme geladen werden.

Das Prompt-Zeichen

Nach dem erfolgreichen Laden (und Starten) des CP/Ms erscheint das sogenannte "Prompt-Zeichen" und dahinter

der Cursor: A>. Das Prompt-Zeichen besteht aus der Laufwerkangabe und aus dem Größer-Zeichen. Grundsätzlich unterstützt CP/M bis zu 16 Laufwerke (Diskettenstation/Festplatte). Jedes Laufwerk ist mit einem Buchstaben von A bis P bezeichnet. Der SCHNEIDER-CPC unterstützt unter CP/M allerdings nur zwei Laufwerke, also A und B. Das Prompt-Zeichen gibt immer das aktuell ausgewählte Laufwerk wieder. A> bedeutet, daß das Laufwerk A (Hauptlaufwerk!) ausgewählt ist. Alle Kommandos beziehen sich auf dieses Laufwerk. Das ">"-Zeichen trennt nur die Laufwerksangabe vom folgenden Kommando.

Das erste Kommando

Verfügen Sie über zwei Diskettenstationen, werden Sie sich sicher fragen, wie das zweite Laufwerk (B) ausgewählt werden kann? Geben Sie bitte A>B: {CR} ein (" {CR}" bedeutet, daß die Eingabe mit der ENTER-Taste abzuschließen ist). Haben Sie ein zweites Laufwerk angeschlossen und befindet sich eine Diskette im Laufwerk, so hat sich das Prompt-Zeichen in B> geändert. Als aktuelles Laufwerk ist jetzt Laufwerk B ausgewählt.

Haben Sie kein zweites Laufwerk angeschlossen oder ist keine Diskette im Laufwerk B, so erfolgt die Fehlermeldung: Drive B: disc missing - Retry, Ignore or Cancel? Legen Sie dann eine Diskette ins Laufwerk B und wiederholen (R) den Vorgang. Besitzen Sie kein zweites Laufwerk, hilft Ihnen nur noch ein Reset (SHIFT/ESC/CTRL). Vom Laufwerk B kommen Sie auf das Laufwerk A anlog mit dem Kommando B>A: {CR} zurück.

Damit haben wir das erste CP/M-Kommando zur Auswahl des aktuellen Laufwerks kennengelernt. Dazu finden Sie auch die erste Karteikarte, die noch weitere Hinweise zum Kommando gibt.

Was sind Kommandos?

Jede Zeichenfolge, die man dem Prompt-Zeichen eingibt, wird vom CP/M (CCP) zunächst einmal zwischengespeichert. Mit Betätigung der ENTER-Taste prüft das CP/M (CCP), ob es sich um eines der 7 "eingebauten" CP/M-Kommandos handelt (d:, DIR, ERA, REN, SAVE, TYPE, USER). Eingebaute Kommandos sind "Befehle", die das CP/M sofort und selbständig ausführen kann. Diese Kommandos sind Bestandteil des CP/M-Betriebssystems. Das Kommando "d:" haben wir bereits kennengelernt.

Ist das eingegebene Kommando nicht eingebaut, wird auf dem aktuellen Laufwerk nach einem CP/M-Programm mit dem Namen "....COM" gesucht und, falls es vorhanden ist, in den Arbeitsspeicher (TPA) des Computers geladen

und gestartet. Existiert auch kein Programm mit dem Namen "kommando.COM" auf der Diskette, erfolgt eine Fehlermeldung. Dabei wird das Kommando erneut mit einem nachgestellten Fragezeichen (?) ausgegeben.

```
A>hallo {CR}
HALLO?
A>
```

Als Beispiel wurde "hallo" als Kommando eingegeben. Da es sich um kein eingebautes Kommando handelt, sucht das CP/M auf der Diskette nach dem Programm "HALLO.COM". Wird es nicht gefunden, erfolgt die Fehlermeldung. Sie werden sicherlich bemerkt haben, daß das Kommando in Großbuchstaben ausgegeben wurde. Bevor das CP/M (CCP) das eingegebene Kommando überprüft, wird es erst einmal in Großbuchstaben umgewandelt. Von daher ist es egal, ob ein Kommando in Groß- und/oder Kleinbuchstaben eingegeben wird.

Mit den 7 eingebauten Kommandos ist das CP/M nicht gerade reich bestückt. Deshalb befinden sich auf der Systemdiskette mehrere sogenannte "transiente" Kommandos (transient = flüchtig). Dieser Begriff ist etwas verwirrend, da es sich schlicht und einfach um CP/M-Programme handelt. Im folgenden werden wir die transienten Kommandos deshalb als CP/M-Dienstprogramme bezeichnen. Diese werden wie normale Programme in den TPA geladen und gestartet. Mit Beendigung des Programms kehrt das System in die Kommandoebene (das Prompt-Zeichen erscheint) zurück.

Auf der Systemdiskette zum CPC befinden sich insgesamt 22 Dienstprogramme (transiente Kommandos). Sicherlich kennen Sie schon das Programm FORMAT zum Formatieren einer Diskette.

So wird eine Kommandozeile korrigiert!

Bei der Eingabe eines Kommandos kommt es natürlich öfters vor, daß man sich vertippt oder einfach ein nicht gewünschtes Kommando eingibt. CP/M bietet hier einige Möglichkeiten, eine Kommandozeile zu ändern, zu löschen oder zu ignorieren. Diese Editier-Funktionen werden immer durch die Kombination der CONTROL- und einer anderen Taste aufgerufen und direkt ausgeführt. So bewirkt z. B. die Kombination CONTROL X (abgekürzt "X"), daß die gesamte Kommandozeile gelöscht wird. Eine Zusammenfassung dieser Steuerzeichen finden Sie in den Karteikarten.

Besonders wichtig ist die Kombination CONTROL C (C). Mit "C" kann jedes laufende Programm oder Kommando

abgebrochen werden. Dabei wird zusätzlich der CCP und das BDOS von den Systemspuren nachgeladen. Erst dann gelangt das System wieder in die Kommandoebene (Prompt-Zeichen). Durch das Nachladen des CCPs und des BDOS wird sichergestellt, daß die Kontrolle einem vollständigen und intakten CP/M übergeben wird. Der Inhalt des TPA wird allerdings NICHT verändert. Dieser Vorgang nennt sich "Warmstart" (warm boot). Im Gegensatz dazu setzt der "Kaltstart" (cold boot, direktes Laden des CP/Ms mit ICPM) das System in den Urzustand zurück.

Noch ein Wort zu den Steuerzeichen: Die Steuerzeichen sind die nicht sichtbaren Zeichen mit den ASCII-Werten 0 bis 31. So entspricht

- ^@ = ASCII-Code 0
- ^A = ASCII-Code 1
- ^B = ASCII-Code 2
-
- ^Z = ASCII-Code 26
- ^[= ASCII-Code 27
- ^\ = ASCII-Code 28
-] = ASCII-Code 29
- ^^ = ASCII-Code 30
- ^_ = ASCII-Code 31

Wenn Sie im BASIC z. B. PRINT CHR\$(12) eingeben, so wird der Bildschirm gelöscht. CHR\$(12) entspricht aber auch "L! Geben Sie unter CP/M bitte einmal A>L {CR} ein, und siehe da, auch hier wird der Bildschirm gelöscht. (Allerdings mit einem zusätzlichen Fragezeichen, da das CP/M dieses Steuerzeichen nicht "versteh".) Mit diesem CPC-eigenen "Steuerzeichen" lassen sich interessante Dinge anstellen (Windows, Farben, Modis etc.), die auch für die spätere Programmierung unter CP/M sehr nützlich sein können. Einschränkend sei aber auch erwähnt, daß bei der Programmierung auf diese speziellen Steuerzeichen so weit wie möglich verzichtet werden sollte, um die Kompatibilität zu anderen CP/M-Computer zu gewährleisten.

Hier einige nützliche Steuerzeichen-Sequenzen:

A>^L{CR}	loescht den Bildschirm
A>^D@{CR}	MODE 0 (20 Zeichen)
A>^DA{CR}	MODE 1 (40 Zeichen)
A>^DB{CR}	MODE 2 (80 Zeichen)
A>^N@^ND{CR}	PAPER 0 bis PAPER 15
A>^O@^OD{CR}	PEN 0 bis PEN 15
A>^EAA	INK 0,1,1
A>^LABB	INK 1,2,2
A>^I@e	BORDER 0,0
A>^Z^A(^TY	WINDOW(#0),1,40,20,25

Beachten Sie aber immer, daß Sie bei den CPC-Steuerzeichen nicht in Konflikt mit den CP/M-Steuerzeichen kommen! Eine umfangreiche und detaillierte Darstellung aller Möglichkeiten, welche die CPC-Steuerzeichen unter CP/M bieten, finden Sie in meinem Buch "CPC-464/664 Praxis Band 3" vom Hüthig Verlag.

Tastatur unter CP/M I

Viele Steuerzeichen haben unter CP/M eine besondere Funktion. Sie lösen im allgemeinen eine komplexe Funktion aus und erzeugen keine Zeichen auf dem Bildschirm. Sie werden generell in Kombination mit der Control-Taste aufgerufen (Control-C = ^C).

^C: Warmstart. Lädt CP/M neu.

^E: Setzt Kommando in der nächsten Zeile fort.

^H: Löscht Zeichen links vom Cursor.

^I: Setzt Cursor auf den nächsten TAB-Stop.

^J: Line Feed. Entspricht ENTER.

Tastatur unter CP/M II

^M: Carriage Return. Entspricht ENTER.

^P: Leitet die Ausgabe auf den Drucker/Bildschirm um (Ein-/Ausschalter).

^R: Zeigt die Kommandozeile erneut an.

^S: Unterbricht die laufende Anzeige. Weiter mit einer beliebigen Taste.

^X: Löscht die gesamte Kommandozeile.

^U: Bricht Kommandozeile ab. Der Cursor bewegt sich in die nächste Zeile.

^V{CR}: Bricht Kommandozeile ab.

Fehlermeldungen allgemein I**kommando?**

Falsches oder unbekanntes Kommando. Das unkorrekte Kommando wird erneut mit einem Fragezeichen ausgegeben. Es handelt sich weder um ein eingebautes Kommando noch um ein Programm vom Typ ".COM".

DRIVE d: DISC MISSING

Laufwerk d: ist nicht oder nicht korrekt angeschlossen oder die Diskette fehlt. Vorgang wiederholen (R) oder abbrechen (C). Beim Abbruch (C) erscheint eine weitere Meldung (siehe nächste Karte).

Fehlermeldungen allgemein II**BDOS ERR ON d: SELECT**

Angesprochenes Laufwerk nicht aktiv. Mit der ENTER-Taste ist eine Rückkehr in die Kommandoebene des CP/Ms möglich.

BDOS ERR ON d: BAD SECTOR

Beim Zugriff auf das Laufwerk wurde ein nicht lesbarer Sektor angetroffen.

BDOS ERR ON d: R/O

Diskette ist schreibgeschützt. Kann auch dann auftreten, wenn die Diskette gewechselt wurde (^C).

Wildcard (Referenzzeichen)

* ?

*

Ersetzt den gesamten Anteil des Dateinamens oder den Datentypen. "*.BAS", selektiert alle Dateien vom Typ ".BAS".

?

Kennzeichnet (ersetzt) ein nicht relevantes Zeichen. "T????.BAS" selektiert alle Dateien vom Typ ".BAS", die einen 4-stelligen Namen haben, der mit "T" beginnt.

"?????????.???" = "*.*".

Aktuelles Laufwerk wählen

d:

A>d: {CR}

Wechselt das aktuelle Laufwerk auf das angegebene Laufwerk. Es sind nur die Angaben von A und B zulässig. Andernfalls erfolgt eine Fehlermeldung. Ist das gewählte Laufwerk nicht aktiv, erfolgt die Fehlermeldung DISC MISSING.

Typ-Kennzeichnungen**- TYP**

Die 3-stellige, durch einen Punkt vom Filenamen getrennte Typenkennzeichnung dient vor allem als weiteres Unterscheidungsmerkmal. Die Kennzeichnung kann aus beliebigen Buchstaben bestehen. Einige Kennzeichen werden vom CP/M unterstützt und sind zwingend:

.ASC: Assembler-Text	.LIST: Druck-File
.BAK: Back-up	.PRN: Druck-File
.BAS: BASIC-Programm	.SUB: SUBMIT-File
.COM: CP/M-Programm	.TXT: Text-File
.HEX: Intel-Hex-Format	.\$\$\$: Temporär-File

Die Kästchen auf dieser Seite können Sie kopieren, ausschneiden und dann auf Karteikarten kleben. So erhalten Sie im Laufe der Zeit eine nützliche CP/M Kartei.

Pascal-Kurs

Pascal hat in der letzten Zeit einen enormen Aufschwung erlebt. Das hängt sicher auch mit den neuen leistungsfähigen Computern zusammen. Auch die Schneider Computer stehen da nicht abseits. Praxisnah, leicht und locker bringen wir ab dieser Ausgabe einen Pascal Kurs mit Turbo Pascal.

Dieser Kurs will die Programmiersprache Pascal einer größeren Zahl von Computerfans näherbringen. Dies soll auf eine lockere Art und Weise mit vielen Beispielen und Tips geschehen. Das Ende des Kurses ist noch offen, wobei wir aber auch komplexere Programme erstellen wollen. Doch zunächst eine kleine Einführung.

Was ist Pascal?

Pascal ist eine "High Level" Programmiersprache, die von Professor Nikolaus Wirth an der technischen Universität Zürich entwickelt und nach Blaise Pascal, dem berühmten Philosophen und Mathematiker aus dem 17. Jahrhundert, benannt wurde. Sie eignet sich für allgemeine und technische Anwendungen. Ihr besonderer Vorteil liegt in der konsequenten Durchführung der strukturierten Programmierung, die zu übersichtlichen und leicht änderbaren Programmen führt. Pascal ist heute auf dem Wege, eine allgemein bekannte und vielbenutzte Programmiersprache zu werden. Schon im Informatikunterricht an den Schulen wird Pascal gelehrt. Weil das aber meist etwas trocken ist, wollen wir das hier im CPC-Magazin auf eine mehr praktische Weise tun.

Gehen wir zuerst einmal auf die Unterschiede zwischen Basic und Pascal ein. Basic ist die Hochsprache, die am weitesten verbreitet ist. Es gibt wohl keinen Heimcomputer, der nicht mit einem Basicinterpreter ausgerüstet ist. Darüber hinaus bietet Basic in der Interpreter-Form den Vorteil, interaktiv zu sein, d. h., eingegebene Befehle werden entweder sofort verarbeitet (Direktmodus) oder nach Programmänderungen ist das Programm sofort in der veränderten Form ablauffähig.

Bei Compilern ist das anders: Hier muß zunächst der sog. Sourcetext eingegeben werden, der anschließend compiliert wird. Erst dann ist das Programm ablauffähig. Bei jeder Programmänderung ist dann natürlich eine erneute Compilierung notwendig. Dies ist der Nachteil von Compilern. Zwar gibt es auch Basiccompiler, bei denen es sich aber natürlich genauso verhält.

Dieser Unterschied zwischen Interpreter und Compiler ist mit ein Grund für das Unbehagen, das Schüler bei Pascal haben, denn schließlich möchte man so-

fort sehen, was man gemacht hat. Statt dessen muß man erst den Compiler in Gang setzen. Dies führt zu Verzögerungen und zu Frustration, wenn der Compiler sich mit einer Fehlermeldung zurückmeldet. Dann muß der Fehler erst behoben werden, um dann den Compiler erneut zu starten. Unter Umständen kann dies zu einer sehr nervenden Aktion werden. Es soll aber nicht der Hauptvorteil der Compiler verschwiegen werden: Sie erzeugen Maschinencode, der natürlich mit einer Geschwindigkeit abläuft, die man von einem Interpreter nie erwarten kann.

Inzwischen gibt es eine Vielzahl von Pascalcompilern für jeden Computer. Auch hier gibt es Unterschiede, wobei

jeder Compiler seine spezifischen Vor- und Nachteile hat. Dieser Kursus hält sich aber eng an das Standardpascal, wie es von N. Wirth definiert wurde. Verschiedene Compiler bieten darüber hinaus noch Erweiterungen an. Bei Bedarf sollten Sie den entsprechenden Teil des Compilerhandbuches durchsehen. Die Programme dieses Kurses sind alle erprobt und laufen auf einem Schneider CPC 464/664 unter Turbo Pascal.

Der zweite Grund für das Unbehagen derjenigen, die Pascal lernen "müssen", ist folgender: Im Gegensatz zu Basic, wo man sich direkt an den Computer setzen und eine Idee ausprobieren kann, ist dies in Pascal nicht direkt möglich. Zum Programmieren in Pascal muß man sich

Erklärung der Fachbegriffe

Interpreter

Ein Interpreter ist ein Programm, das eine Folge von Befehlen in für den Prozessor des Computers verständliche Maschinensprache übersetzt. Die Abarbeitung ist dabei sequentiell, d. h. ein Befehl nach dem anderen. In der Regel sind Kontrollstrukturen vorhanden, die einen gesteuerten Programmablauf ermöglichen. Die Übersetzung erfolgt während der Abarbeitung des Programmes. Daraus entsteht auch der Hauptnachteil der Interpreter. Sie verschwenden einen großen Teil der Rechenzeit mit Übersetzen und der Suche von Befehlen. Der normale Interpreter muß einen großen Verwaltungsaufwand treiben. Man schätzt, daß ca. 70 - 80% der Rechenzeit für diese Verwaltung aufgewendet werden muß. Vorteil des Interpreters: Befehle können direkt ausgeführt werden.

Compiler

Ein Compiler ist genau wie der Interpreter ein Programm, das eine Folge von Befehlen in die für den Prozessor verständliche Maschinensprache übersetzt. Im Unterschied zum Interpreter erfolgt die Übersetzung vor der Ausführung des Programms. Dies hat zur Folge, daß als Programm reiner Maschinencode vorliegt, der während der Programmabarbeitung natürlich nicht mehr übersetzt werden muß und dementsprechend ist der Programmablauf sehr schnell. Nachteil: Bei Änderung der Befehlsfolge muß das Programm zunächst übersetzt werden, bevor man die Änderung testen kann.

Source Code

Als Source Code bezeichnet man einen Text, der die vom Compiler zu übersetzenden Befehle enthält. Der Source Text wird mit einem Editor erstellt und dann dem Compiler zur Verfügung gestellt.

Object Code

Als Object Code bezeichnet man den vom Compiler erzeugten Maschinencode. Er stellt das ausführbare Programm dar.

Editor

Der Editor ist ein Programm, mit dem man mehr oder weniger komfortabel Source Texte erstellen kann. Der Editor ist im weitesten Sinne eine Textverarbeitung, die aber auf spezielle Dinge für die Programmtexterstellung abgestimmt ist.

über die Struktur des Programmes erst im Klaren sein, bevor man es in den Computer eingeben kann. Die Arbeit des Pascalprogrammierers beginnt also am Schreibtisch. Dies löst natürlich Unbehagen aus, denn man möchte schließlich etwas sehen.

Wir wollen deshalb versuchen, die ersten Programme so kurz wie möglich zu halten, damit die Wartezeit nicht zu lange wird. Diese kleinen Programme werden dann, soweit sinnvoll, verbessert.

Schauen wir uns zunächst ein kleines Pascalprogramm an: Es berechnet den natürlichen Logarithmus von 2. Die den Programmzeilen vorangestellten Zahlen dienen der Orientierung und der Erklärung und müssen nicht eingegeben werden.

```
1 PROGRAM log (output);
2 BEGIN
3   write (ln(2))
4 END.
```

Dieser Text ist also unser Sourcetext. Er gliedert sich in verschiedene Teile: Zeile 1 zeigt dem Compiler, daß hier ein Pascalprogramm beginnt. Dies wird durch das Schlüsselwort "PROGRAM" erreicht. Daran schließt sich der Name des Programmes an, in unserem Falle "log". Das Segment "(output)" zeigt dem Compiler an, daß dieses Programm eine Ausgabe über das angeschlossene Terminal oder den Drucker macht.

Wir werden in diesem Kurs sämtliche Pascalschlüsselworte in Großbuchstaben schreiben, um sie besser erkennen zu können. PROGRAM ist also das erste Schlüsselwort, das wir kennenlernen.

Schlüsselwörter

Schlüsselwörter sind in ihrer Bedeutung festgelegt und dürfen vom Programmierer nicht verwendet werden. Als letztes Zeichen in der Zeile 1 steht ";". Das Semikolon trennt die einzelnen Zeilen des Sourcetextes voneinander und muß bis auf wenige Ausnahmen nach jeder Zeile stehen.

Die erste Ausnahme sehen wir bereits in Zeile 2: Nach dem Schlüsselwort BEGIN steht kein Semikolon. Das Schlüsselwort BEGIN zeigt dem Compiler den Beginn eines Programmumpfes an, dessen Ende durch das Schlüsselwort END markiert wird. Der Punkt hinter dem END zeigt dem Compiler, daß das Programm an dieser Stelle zu Ende ist. Nach dem letzten END eines Pascalprogramms muß also immer ein Punkt stehen.

Die Zeilen 1, 2 und 4 haben wir erklärt, bleibt nur noch die Zeile 3 übrig. In dieser Zeile wird vom Programm die Stan-

dardprocedur "write" aufgerufen. Was Procedures sind, werden wir im Verlaufe des Kurses noch genau erfahren. Die Funktion dieser Procedur ist mit dem "PRINT" in Basic zu vergleichen. Alles was hinter dem write in Klammern steht, wird ausgegeben. Dies können Variablen oder Texte sein.

Das write in Zeile 3 ist um 2 Zeichen gegenüber der oberen Zeile eingerückt. Dies hat den Sinn, die Programme leichter lesbar zu machen, weil auf diese Weise die Programmblöcke leichter voneinander zu unterscheiden sind. Man sollte sich an diese Schreibweise halten, um leicht lesbare Sourcetexte zu erhalten: Als Argument für die writeprocedur wird die Standardfunktion "ln" verwendet. Diese Standardfunktion berechnet den natürlichen Logarithmus von 2. Über die Standardfunktionen werden wir uns später aber noch ausführlich unterhalten.

Aufmerksame Betrachter werden festgestellt haben, daß in Zeile 3 auch kein Semikolon steht. Dies ist eine weitere Ausnahme dieser Regel. Vor dem letzten END des Programmes braucht (bei einigen Computern: darf) kein Semikolon stehen.

Nachdem dieses kleine Programm als Sourcetext mit Hilfe des Editors eingegeben wurde, startet man den Compiler und dann das entstandene Programm.

Die Verwendung von Texten zeigt das folgende Beispiel:

```
1 PROGRAM text (output);
2 BEGIN
3   write ('Hallo')
4 END.
```

Die Zeilen 1, 2 und 4 sind uns ja bereits vertraut. Allerdings heißt jetzt das Programm nicht mehr log sondern text. In Zeile 3 wird wieder die Procedur write aufgerufen, diesmal aber mit einem Text zwischen den Klammern. Der auszugebende Text muß dabei durch Hochkomma eingegrenzt sein. Diese beiden kleinen Programme zeigen die Grundstruktur eines Pascalprogrammes und führen auch zu einem Ergebnis. Allerdings wird wohl jeder einsehen, daß diese Programme noch nicht besonders leistungsfähig sind. Deshalb wollen wir sie jetzt ein wenig verbessern. Vergleichen Sie dazu das Listing von log1 im Kasten.

Was ist daran neu? Die erste Erweiterung ist in Zeile 1 zu sehen. Hinzugekommen ist "input". Dies zeigt dem Compiler, daß neben Ausgaben auch Eingaben vom Programm verarbeitet werden. In Zeile 2 lernen wir ein neues Schlüsselwort kennen: "VAR" sagt dem Compiler, das jetzt die Variablendeklaration erfolgt. Im Gegensatz zum Basic, wo die Variablen jederzeit beliebig be-

nannt und benutzt werden können, ist dies in Pascal nicht möglich. Alle im Programm vorkommenden Variablen müssen vorher im Programmkopf deklariert werden.

Dies mag auf den ersten Blick recht umständlich erscheinen, hat aber einen guten Grund: Im Gegensatz zu Basic, das nur die Variablentypen INTEGER, REAL, ARRAY und STRING kennt, ist es in Pascal möglich, nahezu beliebige Variablentypen zu definieren. Außerdem wird man durch die vorherige Deklara-

Variablendeklaration

tion dazu gezwungen, sich über die Struktur des Programmes Gedanken zu machen. In unserem Fall wird eine Variable mit dem Namen "x" deklariert, die vom Typ REAL ist. Eine Variablendefinition hat immer folgendes Aussehen: Name der Variablen, Typ der Variablen.

Die nächste Änderung ist in Zeile 4 zu sehen. Hier wird die Standardprocedur "read" aufgerufen. Sie dient dazu, Eingaben zu machen. In den Klammern nach dem Bezeichner "read" steht der Name der Variablen, an die der eingegebene Wert übergeben werden soll. Die Variable muß natürlich vorher deklariert werden. Die letzte Änderung ist auch in Zeile 4 zu sehen. Um von beliebigen Zahlen den Logarithmus berechnen zu können, müssen wir den gewünschten Wert natürlich an die Funktion übergeben. Dies geschieht durch die Angabe des Parameters (x) in der Logarithmusfunktion.

Wir haben unser Programm nun so weit erweitert, daß wir Eingaben machen können und diese Eingaben verarbeitet werden. Auch unser Programm "text" soll erweitert werden. Es heißt jetzt "text 1" (siehe Kasten). Anmerkung: In Zeile 2 sind ein ä und ein ü ausgedruckt. Diese beiden Buchstaben müssen durch die Zeichen "eckige Klammer auf und eckige Klammer zu" ersetzt werden.

Auch hier wollen wir die Änderungen beschreiben. In Zeile 1 sind sie ähnlich wie im Programm log1. Der Programmname ist geändert und "input" ist dazugekommen.

In Zeile 2 ist wieder eine Variablendeklaration zu sehen. Dieses Mal wird eine Stringvariable mit der Länge von 30 Zeichen deklariert. An dieser Stelle der Hinweis, daß die Stringvariablendeklaration bei einigen Compilern abweicht. Dies kann man aber im Compilerhandbuch nachlesen.

Die Zeilen 4 und 7 enthalten eine neue Standardprocedur "clrscr". Diese Procedure bewirkt das Löschen des Bildschirms.

log 1

```

1 PROGRAM log1 (input,output);
2 VAR x : real;
3 BEGIN
4   read (x);
5   write (ln(x))
6 END.

```

text 1

```

1 PROGRAM text1 (input,output);
2 VAR Name : stringä30ü;
3 BEGIN
4   clrscr;
5   writeln ('Bitte geben Sie Ihren Namen ein und druecken Sie <ENTER>');
6   readln (Name);
7   clrscr;
8   write ('Hallo ',Name,', hier ist dein Pascalcompiler!')
9 END.

```

log 2

```

1 PROGRAM log2 (input,output);
2 VAR x : real;
3 BEGIN
4   clrscr;
5   REPEAT
6     write ('Bitte Zahl eingeben (Ende durch 1): ');
7     readln (x);
8     IF x>0 THEN writeln ('Der natuerl. Logarithmus von ',x,' ist: ',ln(x))
9       ELSE writeln ('Zahl liegt ausserhalb des Definitionsbereiches !');
10  UNTIL x=1;
11  write ('Programm beendet!')
12 END.

```

text 2

```

1 PROGRAM text2 (input,output);
2 VAR Name : stringä30ü;
3   Alter : integer;
4 BEGIN
5   REPEAT
6     clrscr;
7     writeln ('Bitte geben Sie Ihren Namen ein und druecken Sie <ENTER>');
8     readln (Name);
9     IF Name <> '' THEN
10      BEGIN
11        writeln (Name,' bitte gib dein Alter ein. ');
12        readln (Alter);
13        clrscr;
14        write ('Hallo ',Name,', ich muss sagen, du siehst nicht wie ',Alter);
15        writeln (' Jahre aus. ');
16        writeln;
17        writeln ('Bitte Taste druecken!');
18        readln;
19      END;
20  UNTIL Name = '';
21  write ('Programm beendet')
22 END.

```

Zeile 5 gibt eine Meldung auf dem Bildschirm aus. Das writeln bewirkt im Gegensatz zu write die Ausgabe von <CR>, d. h., der Cursor wird in die nächste Zeile plaziert. In Zeile 6 wird die Eingabe der Variablen "Name" zugeordnet. Auch hier bewirkt readln, daß der Cursor in die nächste Zeile plaziert wird.

Zeile 8 zeigt die gemischte Verwendung von Texten und Variablen bei der Ausgabe. Die auszugebenden Teile werden durch Komma voneinander getrennt.

Unsere Programme sind nun schon leistungsfähiger geworden, denn wir können jetzt Werte eingeben und erhalten verschiedene Resultate. Ganz zufrieden können wir allerdings noch nicht sein. Es ist lästig, für jede Berechnung das Programm neu starten zu müssen. Außerdem ist das Programm log1 noch nicht abbruchsicher, denn die Eingabe von Werten kleiner oder gleich Null führt zu einem unzulässigen Argument für die Logarithmusfunktion. Wir wollen diese Änderungen jetzt durchführen. (Siehe Listing log2).

Die Zeilen 1 bis 3 haben sich bis auf den anderen Programmnamen nicht geändert. In Zeile 4 wird der Bildschirm gelöscht (für die optische Aufbereitung des Programms). In Zeile 5 steht ein weiteres Schlüsselwort für Pascal. "REPEAT" bewirkt, daß der folgende Block so lange ausgeführt wird, bis die Abbruchbedingung erfüllt ist. Diese Abbruchbedingung steht hinter dem Schlüsselwort "UNTIL", hier in Zeile 10. Alles was zwischen REPEAT und UNTIL steht – hier die Zeilen 6 bis 9 – wird so oft ausgeführt, bis die Bedingung, die hinter UNTIL steht – hier $x=1$ – erfüllt ist.

Diese Konstruktion ist also eine Art Schleife mit Abbruchbedingung, bei der die Anzahl der Schleifendurchläufe im voraus noch nicht bekannt ist. Zu beachten ist dabei, daß der Schleifenrumpf mindestens einmal ausgeführt wird, da die Überprüfung der Bedingung erst am Ende des Blockes erfolgt. In unserem Beispiel wurde die Bedingung $x=1$ gewählt, d. h., zur Beendigung des Programms muß man eine 1 eingeben. In Zeile 11 wird das Programmende dann auch angezeigt.

IF...THEN...ELSE

Jetzt noch zu den Zeilen 6–9: Die Zeile 6 gibt eine Meldung aus, die dem Benutzer des Programms eine Anweisung gibt. Zeile 7 liest den eingegebenen Wert und weist ihn der Variablen x zu. In den Zeilen 8 und 9 befindet sich eine Konstruktion, die wohl jeder Programmierer kennt: Die IF...THEN...ELSE Konstruktion. Hier wird realisiert, was beim

Die Pascalschlüsselworte

PROGRAM	Zeigt dem Compiler an, daß hier ein Pascalprogramm beginnt. Parameter (Files): Programmname, benutzte Geräte. Beispiel: PROGRAM log (input, output);
BEGIN	Zeigt dem Compiler den Beginn eines Blocks an. Parameter: keine
END	Zeigt dem Computer das Ende eines Blockes an. Parameter: keine
REPEAT	Anfang einer Schleifenkonstruktion, für die die Anzahl der Schleifendurchläufe nicht vorher feststeht. Parameter: keine
UNTIL	Ende der mit REPEAT angefangenen Schleife. Parameter: Abbruchbedingung Beispiel: UNTIL x=1
VAR	Zeigt dem Compiler den Beginn der Variablendeklaration an. Parameter: Variablenname: Variablentyp Beispiel: VAR Name : byte x : integer y : real test : boolean ...
IF THEN	Anweisung zur Verzweigung innerhalb eines Programmes. Parameter: Bedingung Beispiel: IF $x>1$ THEN write (x)
ELSE	Option bei der IF THEN Konstruktion. Ist die Bedingung erfüllt, wird der Teil nach THEN ausgeführt. Ist sie nicht erfüllt, der Teil nach ELSE. Parameter: keine Beispiel: IF $x>1$ THEN write (x) ELSE write (x+1) Besonderheit: In der Zeile vor ELSE darf kein Semikolon stehen.

Standardfunktionen und Standardprocedures

(Weitere folgen in den nächsten Ausgaben.)

write	Allgemeine Ausgabeprocedur. Es können sowohl Texte als auch Variablen ausgegeben werden (auch gemischt). Beispiele: write (x) write ('Hallo') write ('Hallo', name)
writeln	Wie write, nur daß der Cursor in die nächste Zeile plaziert wird.
read	Allgemeine Eingabeprocedur. Es werden Werte eingelesen und der bezeichneten Variablen zugewiesen. Es erfolgt eine Überprüfung, ob der Werttyp mit dem Variablentyp übereinstimmt. Beispiel: read (x) read (name) Die Variablen müssen natürlich vorher deklariert werden.
readln	Wie read, nur daß der Cursor in die nächste Zeile plaziert wird.
ln(Wert)	Standardfunktion zur Berechnung des natürlichen Logarithmus. Als Wert wird Integer oder Real zugelassen. Das Ergebnis ist Real.
clrscr	Standardprocedur zum Löschen des Bildschirms.

Programm log1 noch unzureichend war. Der eingegebene Wert wird zunächst überprüft, ob er in den gültigen Grenzen liegt. Wenn ja, dann wird der Logarithmus berechnet und ausgegeben, wenn nein, dann wird eine entsprechende Meldung ausgegeben.

Diese Konstruktion hat also folgende Wirkung: Wenn Bedingung erfüllt, dann Aktion 1, ansonsten Aktion 2. Die Abfolge "ELSE Aktion 2" in dieser Konstruktion ist nicht zwingend vorgeschrieben, es ist auch möglich, nur die Konstruktion IF...THEN... zu verwenden. Bei Nichterfüllung der Bedingung wird mit der Abarbeitung dann bei der nächsten Anweisung nach dem IF THEN Block fortgefahren.

Auch unser Textprogramm kann noch verbessert werden. Betrachten wir dazu das Listing text 2.

Die erste Neuerung ist in Zeile 3 anzutreffen. Hier wird eine weitere Variable deklariert, die den Namen "Alter" hat und vom Typ INTEGER ist. Hier soll das Alter abgespeichert werden. In Zeile 9 wird überprüft, ob bei der Eingabe nur ENTER gedrückt wurde. Nur wenn dies nicht der Fall ist, wird der folgende Block ausgeführt. Diese Überprüfung ist notwendig, weil als Abbruchbedingung für die REPEAT UNTIL Schleife eine Leer-eingabe für die Variable Name ausgewertet wird.

An diesem Listing fällt noch folgendes auf: In Zeile 10 ist ein weiteres BEGIN und in Zeile 19 ein weiteres END. Diese beiden Schlüsselworte kennzeichnen den zur IF THEN Konstruktion gehörenden Block. Benötigt man nach IF THEN nur eine Anweisung, so können BEGIN und END entfallen. Sind dagegen mehrere Anweisungen notwendig, so müssen diese durch BEGIN und END eingeschlossen werden.

Das writeln in Zeile 16 bewirkt die Ausgabe einer Leerzeile. Das readln in Zeile 18 wartet auf das Drücken der ENTER-Taste, damit der Bildschirm nicht sofort gelöscht wird, bevor man den ausgegebenen Text gelesen hat.

Bis zum nächsten Mal können Sie ja folgende Programmieraufgabe lösen: Es soll ein Programm erstellt werden, das einen Wert einliest und die Quadratwurzel bzw. das Quadrat des eingegebenen Werts ausgibt. Und zwar soll das Quadrat ausgegeben werden, wenn der Wert negativ ist und die Quadratwurzel, wenn der Wert positiv ist. Bei der Eingabe von 0 soll das Programm stoppen. Die dazu notwendigen Standardfunktionen sind "sqr" für das Quadrat und "sqrt" für die Quadratwurzel. Die Lösung dieser Aufgabe werden wir beim nächsten Mal vorstellen.

H. J. Janke

Lesermeinungen

Eure Professionalität wird euch den Erfolg garantieren.
W. Jäger, Ortenau

...entschloß ich mich zum Kauf, und ich muß sagen, ich habe es nicht bereut.
M. Mikat, Elchingen

Der positive Gesamteindruck eurer Zeitschrift hat mich bewogen, sie zu abonnieren.
W. Buchner, Nürnberg

Diese Zeitschrift finde ich die beste, die ich je gesehen habe.
F. Reiter, München

Ich habe ihr Magazin gekauft und bin nicht unzufrieden. Weiter so!
S. Hinker, Auenwald

Die Programme sind sehr unübersichtlich.
R. Barie, Mutterstadt

Programmlistings, deren Struktur nur schwer zu erkennen ist.
R. Pekx, Duisburg

Die Gliederung des CPC-Magazins könnte besser sein.
W. Gerth, Laufenberg

Unvoreingenommene Berichterstattung: nicht wie andere Computerzeitschriften.
T. Lukaschewski, Schwieberdingen

Die Qualität der Seiten: Ich bin von CPC-International besseres gewohnt.
T. Willinski, Herne

Schrift der vielen Listings läßt noch zu wünschen offen.
M. Abendroth, Remscheid

Tendenz dieser Zeitschrift geht zum semiprofessionellen Anwender.
M. Köhne, Kyllburg

Das Niveau der Serien ist ein wenig zu niedrig.
H. Adam, Neunkirchen

JEDEN MONAT



Für den besten
TIP oder TRICK
rund um die SCHNEIDER
COMPUTER

Magazin
Für alle Schneider Computer

Kennwort:
»Tips und Tricks«

CPC-Magazin
Postfach 1640
7518 Bretten

Z80 – Assemblertips – Teil 2

Wollten auch Sie schon immer mal wissen, was Interrupts sind und wie man sie programmiert? Lesen Sie dazu unsere Assemblertips, Teil 2.

Heute behandeln wir die Programmierung von Interrupts auf dem Schneider CPC. Sie ist bei Spielen aber auch bei Anwenderprogrammen äußerst hilfreich, bei manchen Problemstellungen sogar absolut notwendig. Außerdem soll der Aufbau des Bildschirmspeichers dargestellt werden.

Was sind Interrupts?

Wenn der Prozessor ein Maschinenprogramm abarbeitet und es erfolgt ein Interrupt, unterbricht der Rechner sofort das laufende Programm und springt zu einer Interruptbehandlungsroutine. Ein Interrupt ist also eine Unterbrechung, die man gezielt für eigene Aufgaben nutzen kann. Es gibt nun zwei Arten der Interruptprogrammierung: die Hardware-Interruptprogrammierung und die Software-Interruptprogrammierung.

Die Hardwareprogrammierung ist schwer zu handhaben und äußerst unkomfortabel zu programmieren. Wir werden uns deshalb auf die Softwareprogrammierung beschränken.

Im CPC ist die Programmierung der Interrupts folgendermaßen vorgesehen: Jede dreihundertstel Sekunde springt der CPC über einen Hardwareinterrupt zu einer bestimmten Adresse im ROM. Hier werden die Interruptwarteschlangen überprüft und eventuell wird zu einer Interruptbehandlungsroutine gesprungen, die dann den Interrupt bearbeitet.

In den Interruptwarteschlangen muß der Benutzer eintragen, ab welcher Adresse seine Behandlungsroutine steht, welche Priorität diese Routine hat und um welchen Interrupt es sich handelt. Für all diese Einträge gibt es spezielle ROM-Routinen, die den Umgang mit Interrupts einfach und bequem machen.

Die Interrupts

1. FRAME FLY BACK – Das Bild auf dem Monitor des CPC wird ja durch einen Elektronenstrahl erzeugt. Dieser schreibt das Bild von oben nach unten. Ist er unten angekommen, muß er wieder nach oben links gelenkt werden. Während dieser Zeit wird der Bildschirm nicht weiter beschrieben. Sobald sich der Strahl auf dem Rückweg befindet, wird diese Interruptwarteschleife abge-

arbeitet. Dieses passiert jede fünfzigstel Sekunde.

2. TICKER – Dieser Interrupt tritt ebenfalls jede fünfzigstel Sekunde auf.

3. FAST TICKER – Dieser Interrupt wird jede dreihundertstel Sekunde abgearbeitet.

Für alle Interruptarten gibt es ROM-routinen, die einen Interrupt in die Interruptwarteschleifen einreihen oder diesen daraus löschen. Kommen wir zu nächst zum Einreihen der Routine. Folgende Parameter müssen an die ROM-routine übergeben werden.

DE – Interruptbehandlungs-routinenadresse

DE zeigt auf die Adresse der Behandlungsroutine, zu der bei einem entsprechenden Interrupt gesprungen werden soll.

C – ROM Auswahl

Selbstverständlich können auch Routinen im ROM angesprungen werden. Es kann ein ROM von 0–251 ausgewählt werden, wenn der Interrupt vom Typ "Far" ist. Wird ein RAM ausgewählt, sollte C 255 sein.

HL – Aufbaublock

HL zeigt auf 9 freie Bytes, die das Betriebssystem braucht, um den Interrupt in die Interruptwarteschleife einzubinden.

B – Ereignisklasse

Jedes Bit im B-Register hat eine spezielle Funktion:

- Bit 0 = 1 Die Interruptbehandlungsroutine liegt im zentralen 32 K-RAM.
- Bit 0 = 0 Die Sprungadresse ist vom Typ "Far", das B-Register wird beachtet.
- Bits 1–4 Diese geben die Priorität an.
- Bit 5 Muß immer 0 sein.
- Bit 6 = 1 Es handelt sich um einen Expressinterrupt, d. h., er ist besonders eilig. Expressinterrupts haben eine höhere Priorität wie normale Interrupts mit der höchsten Priorität.
- Bit 6 = 0 Es handelt sich um keinen Expressinterrupt
- Bit 7 = 1 Asynchroner Interrupt: Diese Interrupts werden sofort in die Interruptausführungskette eingereiht. Handelt es

sich sogar noch um einen Expressinterrupt, wird er auf der Stelle ausgeführt.

Bit 7 = 0 Synchroner Event: Der Rechner sucht noch die ganze Interruptwarteschleife nach Interrupts durch und reiht sie entsprechend ihrer Priorität in die Interruptausführungskette ein.

Die Parameter sind für die zwei Interruptarten identisch. Die Einsprungsadressen ins ROM sind allerdings verschieden. Die Routinen bewirken, daß der entsprechende Interrupt, der durch die Register festgelegt wird, in die Interruptkette eingereiht und dann bald ausgeführt wird. Ein Interrupt sollte erst eingerichtet werden, wenn sich auch schon die Behandlungsroutine dieses Interrupts an der richtigen Stelle im Speicher befindet, da diese Routine per Interrupt bald angesprungen wird. Wenn sich diese dann nicht dort befindet, kommt es eventuell zu einem Absturz des Rechners.

KL NEW FRAME FLY BACK – &BCD7
KL NEW FAST TICKER – &BCE0

Selbstverständlich kann man einen Interrupt auch wieder aus der Interruptwarteschleife ausreihen. Dazu ist es notwendig, in HL die Adresse des Aufbaublockes zu übergeben und dann die Routine anzuspringen.

KL DEL FRAME FLY BACK – &BCDD
KL DEL FAST TICKER – &BCE6

Ein Anwendungsbeispiel

Verlassen wir die graue Theorie, schauen wir uns die Interruptprogrammierung an einem Beispiel an. Angenommen, wir wollen jede Sekunde die aktuelle Borderfarbe um eins erhöhen. Wenn sie 27 übersteigt, soll sie natürlich wieder den Wert 0 annehmen. Wir bedienen uns der ROMroutinen SCR SET BORDER (&BC38) und SCR GET BORDER (&BC3B). SCR GET BORDER liefert in B und C die aktuelle Borderfarbe, SCR SET BORDER stellt den Border auf die Farben aus den Registern B und C ein.

Die Routine soll bei Hex 8000 beginnen. Wir wählen als Interrupt ein FRAME FLY BACK aus. Da dieser fünfzigmal in der Sekunde ausgelöst wird, richten wir noch einen Zähler ein, der von 50 bis 0 herunterzählt. Kommt die-

ser bei 0 an, wird der Border einmal gewechselt und unser Zähler erneut auf 50 gesetzt. Listing 1 ist das komplette Assemblerlisting, welches mit dem sehr guten Schneiderassembler "DEVPAK" erstellt wurde. Für Personen ohne Assembler ist das Programm noch einmal in Form von DATAs abgedruckt. Diese müssen Sie eingeben und dann mit RUN starten.

Mit CALL &8000 startet das Programm und unsere Routine wird in die Interruptkette eingereiht. Jede fünfzigstel Sekunde erfolgt ein Aufruf, durch den Zähler wird aber die eigentliche Routine, die Änderung der Borderfarbe, nur jedes fünfzigste Mal ausgeführt (einmal pro Sekunde) wie das ja geplant war.

Mit CALL &800E wird schließlich die Routine wieder aus der Interruptkette ausgereiht, das Wechseln des Borders stoppt sofort.

Was ist ein Bildschirmspeicher?

Im Bildschirmspeicher wird das augenblicklich auf dem Bildschirm befindliche Bild gespeichert. Es gibt dabei verschiedene Arten. Der Commodore hat z. B. einen 1 K großen Textspeicher. Der Bildschirmspeicher ist hier in 40 mal 25 Printpositionen unterteilt ($40 \times 25 = 1000$) und jeder Position wird ein Byte, also ein Wert zwischen 0 und 255 zugeordnet. Dieser Wert stellt dann einen Buchstaben oder ein festes Grafikzeichen dar.

Der Bildschirmspeicher

Diese Methode hat den Vorteil, daß Zeichen schnell dargestellt werden können und für den Bildschirmspeicher verhältnismäßig wenig Platz verbraucht wird. Der große Nachteil besteht aber darin, daß man Text und Grafik nicht mi-

schen kann. Deshalb sind beim Commodore auch keine Plot- und Drawbefehle vorhanden. Zur Ehrenrettung des Commodore muß man aber erwähnen, daß auf einem anderen Bildschirmspeicherformat ebenfalls Grafik möglich ist.

Beim CPC ist der Bildschirmspeicher ganz anders organisiert. Es handelt sich um einen echten Bildschirmspeicher, d. h., jeder Punkt wird für sich gespeichert, so daß Text und Grafik beliebig mischbar sind. Wie diese Punkte abgespeichert werden, das werden wir uns jetzt näher ansehen.

Der Bildschirmspeicher des CPC 464 liegt im Bereich von 49152 bis 65535. Er ist also 16384 Bytes oder 16 K groß. Die Darstellung der Punkte ist natürlich vom jeweiligen Bildschirmmodus abhängig, aber grundsätzlich identisch. Tippen Sie folgendes Programm ein, daran wird der Aufbau des Bildschirmspeichers (BSSP) besonders deutlich:

```
MODE 2: FOR I=49152 to 65535: POKE I, 255: NEXT
```

Durch diesen Befehl werden alle Bits des Bildschirmspeichers gesetzt. Zuerst die Punkte der obersten Reihe der ersten Textzeile, dann die Punkte der obersten Reihe der zweiten Textzeile bis schließlich die Punkte der obersten Reihe der fünfundzwanzigsten und damit untersten Textzeile gesetzt sind. Anschließend wird die zweite Reihe der obersten Textzeile gesetzt, dann die nächste bis zur fünfundzwanzigsten Zeile. Das geht so weiter, bis dann alle 8 Reihen einer Textzeile gesetzt sind.

Eine Reihe besteht aus 80 Bytes. Bei 25 Zeilen ergibt sich ein Speicherbedarf der obersten Reihen der Textzeilen von 25 mal 80 Bytes = 2000 Bytes. Bei acht Reihen ergibt dies einen Speicherbedarf von exakt 16000 Bytes. Der Bildschirmspeicher ist aber 16384 Bytes lang, da-

mit stellt sich die Frage, wo die restlichen 384 Bytes geblieben sind. Die Antwort ist einfach: Nach jeweils 2000 Bytes (25 Reihen) sind 48 unbenutzte Bytes eingefügt. Das hat gute Gründe. So kann der Prozessor mit 2048 Bytes Unterschied viel besser rechnen als mit 2000 Bytes. Außerdem kann der Rechner die 48 Bytes beim Scrollen als Zwischenspeicher benutzen. Auch Sie können diese 384 Bytes als kurzfristigen Zwischenspeicher nutzen. Denken Sie aber daran, daß bei CLS oder MODE der gesamte Bildschirm gelöscht wird. Bei diesen Befehlen gehen Ihre Daten deshalb verloren.

Der Aufbau der 80 Bytes einer Reihe hängt vom augenblicklichen Modus ab. Schauen wir uns die einzelnen Modi einmal an:

MODE 2

Hier haben wir 640 Punkte pro Zeile. Da 80 Bytes oder 640 Bits dafür zur Verfügung stehen, ergibt sich daraus, daß pro Punkt ein Bit Speicherplatz verbraucht wird. Im Mode 2 gibt es ja nur zwei Farben, nämlich Hintergrund- und Vordergrundfarbe. Ist ein Bit 0, so hat dieser Punkt die Hintergrund- und Vordergrundfarbe oder ist aus. Ist ein Bit 1, leuchtet der Punkt in der Vordergrund- oder Zeichenfarbe. Die Reihenfolge ist auch recht einfach. Von links nach rechts kommt Bit 7, dann Bit 6 bis Bit 0.

MODE 1

Im Mode 1 ist die Sache schon komplizierter. Es müssen jetzt vier Zustände oder Farben dargestellt werden. Dazu braucht man 2 Bits pro Punkt. Der Bildschirmspeicher müßte also doppelt so groß werden. Da das den CPC Entwicklern doch zu groß geworden wäre, wurde die horizontale Auflösung halbiert. Damit stehen nur noch 320 Punkte zur Verfügung und 320 Punkte mal 2 Bits

Borderwechsel per Interrupt

```
10 REM =====
20 REM          BORDERWECHSEL PER INTERRUPT - by Andreas Zallmann
30 REM =====
40 MEMORY 32000
50 FOR i=1 TO 65:READ a:POKE i+32767,a:sum=sum+a:NEXT
60 IF sum<>9173 THEN PRINT "ERROR IN DATA...":END
70 CLS:PRINT "OK.":PRINT:PRINT "START MIT CALL &8000
          ENDE MIT CALL &800E"

80 LOCATE 1,20:END
1000 DATA 33,65,128,17,20,128, 6,129, 14,255,205,215,188,201, 33, 65,128,195,22
1,188,243,229,213,197,245,221,229,253,229, 33, 64,128, 53, 32, 19, 54, 50,205, 5
9,188
1010 DATA 120, 60,254, 28,250, 49,128, 62, 0, 71, 79,205, 56,188,253,225,221,22
5,241,193,209,225,251,201, 1
```

ergeben wieder unsere 80 Bytes. War im Mode 2 ein Byte für 8 Punkte zuständig, so ist jetzt ein Byte nur noch für 4 Punkte zuständig.

Bildschirmauflösung

Die Anordnung der Punkte ist jetzt folgendermaßen: Ein Byte wird in zwei Nibbel (4 Bits) unterteilt. Die jeweils ersten Bits der Nibbles ergeben die Farbe des ersten Punktes, die zweiten Bits die Farbe des zweiten Punktes, usw. Hätte ein Byte den Dualwert 00001111, hätten alle Punkte die Farbe 10. (Es wird von rechts nach links gelesen.) Ein Byte mit dem Dualwert 1010000 würde folgendes bedeuten: Punkt 1 und Punkt 3 haben die Farbe 01, Punkt 2 und 4 haben die Farbe 00 (Hintergrund). Probieren Sie einmal selbst aus, was sich durch Poken von Werten in die Speicherstelle 49152 ergibt.

MODE 0

Das Prinzip von Mode 0 ist dem von Mode 1 ähnlich. Da wieder doppelt so viele Bits (vier) für die jetzt 16 Farben notwendig sind, wurde die Auflösung erneut auf 160 halbiert. Die Bits eines Bytes sind nun folgendermaßen aufgeteilt:

```
7 6
3 2
5 4
1 0
```

In diesem Modus werden pro Byte zwei Punkte dargestellt. Um die Farbnummer des ersten Punktes zu erhalten, schreiben Sie die Werte der Bits 7, 3, 5 und 1 wie oben untereinander und Sie erhalten den Wert, wenn Sie die Bits von unten nach oben hintereinanderschreiben. Um die Farbe des zweiten Punktes zu ermitteln, müssen Sie nur mit den Bits 6, 2, 4 und 0 das gleiche tun.

Damit wäre der Aufbau des Bildschirmspeichers geklärt. Zu erwähnen

ist noch, daß die Farbnummer (z. B. 01) nicht die Farbe im Handbuch ist (01 = blau), sondern die Penfarbe. Übrigens, wenn von 160 Punkten Auflösung die Rede ist, werden Sie sich vielleicht wundern. Auch in Mode 0 kann man noch bei PLOT 639,0 einen Punkt sehen. Das kommt daher, weil die CPC Entwickler bemüht waren, in allen drei Modi die gleichen Koordinaten zu verwenden. So wird bei Mode 1 und Mode 0 halt gerundet. Probieren Sie mal diese Zeile aus:

```
MODE 0: PLOT 0, 0: PLOT 1, 0: PLOT
3, 0
```

Sie sehen nur einen Punkt. Alle anderen Punkte werden auf den vorherigen gesetzt.

Im nächsten Heft

Im nächsten Heft behandeln wir ROMroutinen zum Plotten und Zeichnen in Maschinensprache. Als Anwendungsbeispiel wird eine Fillroutine erstellt.

Andreas Zallmann

Listing 1

```
8000          010          ORG #8000          ;Routine beginnt bei Hex 8000
8000 214180   020          LD HL, frei        ;HL zeigt auf 9 freie Bytes für Block
8003 111480   030          LD DE, routin     ;DE zeigt auf die Routine
8006 0681     040          LD B, %10000001 ;Asynchroner Interrupt
8008 0EFF     050          LD C, 255         ;RAM ausgewählt
800A CDD7BC   060          CALL #BCD7       ;Interrupt auslösen
800D C9       070          RET              ;und zum BASIC
800E 214180   080          LD HL, frei        ;HL zeigt auf Interruptblock
8011 C3DDBC   090          JP #BCDD        ;Ausreihen und zurück
8014 F3       100          routin: DI       ;Interrupts sperren
8015 E5       110          PUSH HL         ;Alle Register retten
8016 D5       120          PUSH DE
8017 C5       130          PUSH BC
8018 F5       140          PUSH AF
8019 DDE5     150          PUSH IX
801B FDE5     160          PUSH IY
801D 214080   170          LD HL, zaehl     ;HL zeigt auf Zähler
8020 35       180          DEC (HL)        ;Erniedrigen
8021 2013     190          JR NZ, end      ;<>0 Routine nicht ausführen, weiterspr.
8023 3632     200          LD (HL), 50     ;Ansonsten: Zähler wieder auf 50 und
8025 CD3BBC   210          CALL getbor     ;Borderfarbe holen
8028 78       220          LD A, B         ;in Accu laden
8029 3C       230          INC A          ;Borderfarbe erhöhen
802A FE1C     240          CP 28          ;Borderfarbe größer 27 ?
802C FA3180   250          JP M, weit     ;Nein: Weiterspringen
802F 3E00     260          LD A, 0        ;Ja: Borderfarbe = 0
8031 47       270          weit: LD B, A        ;Register für Rom-
8032 4F       280          LD C, A        ;routine herichten
8033 CD38BC   290          CALL setbor    ;Neuen Border setzen
8036 FDE1     300          end: POP IY     ;Alle Register
8038 DDE1     310          POP IX        ;wieder zurückholen
803A F1       320          POP AF
803B C1       330          POP BC
803C D1       340          POP DE
803D E1       350          POP HL
803E FB       360          EI           ;Interrupts freigeben
803F C9       370          RET          ;und zurück
8040 01       380          zaehl: DEFB 1 ;1 Byte für den Zähler
8041          390          frei: DEFS 9   ;9 Bytes für Interruptblock
BC38          400          setbor: EQU #BC38 ;Adresse von SET BORDER
BC3B          410          getbor: EQU #BC3B ;Adresse von GET BORDER
```

Preisausschreiben

30 Abos zu gewinnen!

Für unsere Knöbelfreunde bringen wir in diesem Heft ein Logical. Zu gewinnen gibt es diesmal 30 Jahresabonnements vom CPC Magazin. Teilnehmen kann jeder Leser. Einsendeschluß ist der 1. Februar 1986. Der Rechtsweg ist ausgeschlossen.

So, jetzt die Aufgabe:

Vor einer Ampel stehen 5 Autos und warten darauf, daß es grün wird. Ein Golf, ein Mercedes, ein Ford, ein Opel und ein 2 CV. Die Autos in den Farben rot, grün, weiß, gelb und blau haben alle ein anderes Baujahr und zwar 1978, 1985, 1973, 1980 und 1975. Die Insassen, eine Frau mit Hund, ein Soldat, ein Mann mit Glatze, ein Student und ein verliebtes Pärchen vertreiben sich die Zeit auf unterschiedliche Weise. Mit radiohören, Querflöte spielen, nach den Fußgängern schauen, mit küssen und mit Kassetten hören. Folgende Aussagen helfen Ihnen, die Reihenfolge vor der Ampel zu bestimmen, damit Sie am Schluß die 3 Fragen beantworten können.

1. Zwischen dem Mercedes und dem Opel steht der Ford.
2. Der 2 CV ist blau.
3. Der Ford wurde 1980 gebaut.
4. Der Student spielt auf der Querflöte.
5. Zwischen Baujahr 78 und Baujahr 80 steht ein grünes Auto.
6. Im Opel sitzt ein Soldat in Uniform.
7. Im Golf sitzt eine Frau mit Hund und hört Radio.
8. Der Soldat sieht vor sich den Studenten.
9. Der Opel, Baujahr 73, steht vor dem 2 CV.
10. Das Paar im blauen Auto bildet das Schlußlicht.
11. Der 2 CV wurde 1975 gebaut.
12. Der Mann mit Glatzkopf schaut sich die Fußgänger an.
13. Vor dem gelben steht ein weißes Auto.

Auf Ihrer Antwortkarte muß nun stehen, wer Kassetten hört, welche Farbe der Golf hat und wann der Mercedes gebaut wurde.

Viel Spaß beim Knobeln!

Senden Sie Ihre Lösung an das CPC-Magazin, Postfach 1640 in 7518 Bretten.

Sieben auf einen Streich – Teil 2

464

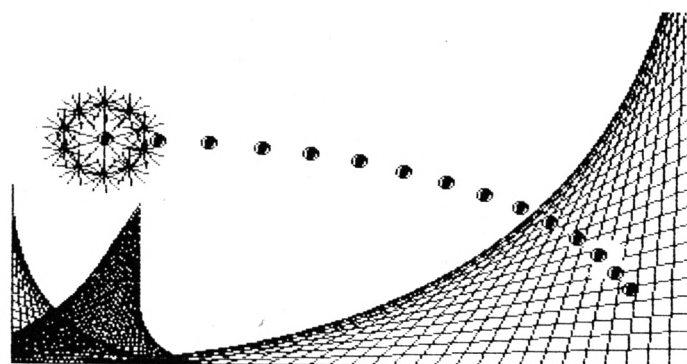
664

6128

Wie schon in der letzten Ausgabe, bringen wir an dieser Stelle wieder sieben kurze Listings, die eine Grafik aufbauen. In der letzten Folge wurden vor allem Titelgrafiken erstellt, die den Namen eines Programmes während dem Ladevorgang anzeigen.

Bei den sieben Listings in diesem Heft werden verschiedenartige Grafiken erstellt. Sie sollen vor allem Anregungen sein, die der Leser nach seinen eigenen Bedürfnissen ändern kann. Bei diesen sehr kurzen Programmen schadet es auch nichts, wenn man sie einfach nur abtippt und anschaut.

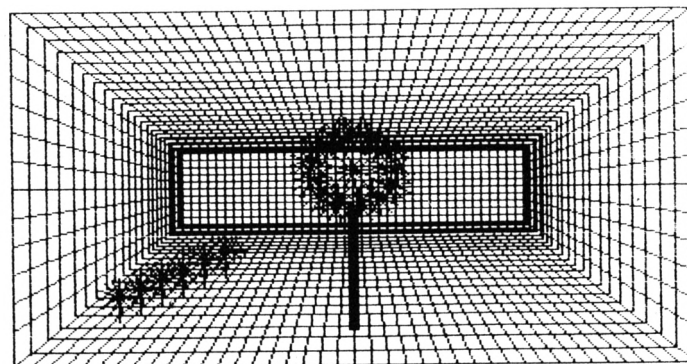
Eine Auswahl der Grafiken zeigen die abgedruckten Hardcopies. Sie sind zwar nur schwarzweiß, auf dem Bildschirm sieht man die Grafiken aber natürlich mehrfarbig.



Die Programme werden mit RUN gestartet. In Zeile 200 befindet sich jeweils eine Endlosschleife, damit ein READY die Grafik nicht zerstört. Um abzubrechen, drücken Sie zweimal ESCAPE (ESC). Vor dem Eintippen eines Listings sollten Sie den Rechner zurücksetzen, da bei den Listings

7 Kurzprogramme

davon ausgegangen wird, daß sich der Rechner im Einschaltzustand befindet. So werden nur die vom Einschaltzustand abweichenden Farben umdefiniert, um Platz und Aufwand zu sparen. Die Programme laufen auf allen Schneider-Heimcomputern.



1. Dieses Programm erstellt ein Grid (Gittermuster), durch das eine Blume wächst.
2. Auch hier wird ein Grid gemalt. Man sieht die aufgehende Sonne.
3. Dieses Programm malt eine Spinne mit Netz.
4. Dieses Programm zeichnet wieder ein Gittermuster. Dadurch wird ein dreidimensionaler Eindruck erweckt. Ein Gebilde fliegt in Kreisen auf den Betrachter zu.
5. Hier handelt es sich ebenfalls wieder um eine Blume mit Grid.
6. Programm 6 erstellt die Skyline einer Stadt, die sich im davor liegenden Meer spiegelt.
7. Dieses Programm erstellt einige Pyramiden. Auf Tastendruck wechseln die Farben. So werden verschiedene Tageszeiten, wie Mittagszeit und Dämmerung dargestellt.

Christoph Schillo + Andreas Zallmann

Teil 1

```

1 ' GRAFIKGAGS 2 - Teil 1
10 SYMBOL AFTER 230:SYMBOL 231,60,94,191
,191,189,189,122,60
20 MODE 1:INK 0,0:INK 1,13:INK 2,26:INK
3,18:BORDER 0
30 LOCATE 37,20:PEN 3:PRINT CHR$(231):PE
N 1
40 FOR a=0 TO 180 STEP 5:PLOT 120,a,1:DR
AW a+40-200,0:PLOT 120,a:DRAW 400-a-200,
0:NEXT
50 FOR a=0 TO 640 STEP 20:PLOT 640,a:DR
AW a,0:NEXT
60 FOR a=0 TO 200 STEP 10:PLOT 0,a:DRAW
200-a,0:NEXT
70 FOR a=100 TO 140 STEP 2:PLOT 544+RND*
5,a:DRAW 30-RND*5,0,0:NEXT
80 FOR a=0 TO 2*PI STEP PI/5:PLOT 86,252
,2:DRAW SIN(a)*40,COS(a)*40:FOR b=0 TO
2*PI STEP 0.6:PLOT 86+SIN(a)*40,252+COS(
a)*40:DRAW SIN(b)*20,COS(b)*20,RND*2+1:
NEXT b,a
90 TAG:FOR a=0 TO 1.4 STEP 0.1:PLOT SIN(
a)*500+80,COS(a)*200+60,3:PRINT CHR$(231
);:NEXT
200 GOTO 200

```

Teil 2

```

1 ' GRAFIKGAGS 2 - Teil 2
10 MODE 1:INK 0,0:BORDER 0:INK 1,13:INK
2,24:INK 3,7
20 ORIGIN 0,0,0,640,170,0
30 FOR a=-380 TO 1000 STEP 20
40 PLOT a,0,1:DRAW 320,300
50 NEXT
60 x=200:FOR a=1 TO 80:PLOT 0,201-x:DRAW
R 640,0:x=x/1.1:NEXT
70 ORIGIN 0,0,0,640,400,0

```

```

80 FOR a=1 TO 200:PLOT RND*640,RND*232+1
68,RND*2+1:NEXT
90 FOR a=PI TO PI*2 STEP PI/18.1
100 FOR b=0 TO 1:PLOT 320,172+b*2,2+b:DR
AWR COS(a)*40,-SIN(a)*40:NEXT b,a
200 GOTO 200

```

Teil 3

```

1 ' Grafikgags 2 - Teil 3
10 MODE 1:INK 0,11: BORDER 11:INK 1,0:INK
2,26:INK 3,3
20 FOR a=1 TO 5 STEP 0.4:PLOT 0,340-a*50
,2: DRAW 50*a,0: DRAW 50*a,50*a: DRAW 0,
50*a: NEXT
30 PLOT 0,340: DRAW 340,0: PLOT 0,340: DRAW
R 640,0: PLOT 0,0: DRAW 0,400: PLOT 0,340:
DRAW 80,60
40 FOR a=1 TO 7: LOCATE 30,a+10: FOR b=0 T
O 2: READ c: PRINT CHR$(c);: NEXT b,a
50 DATA 204,208,205,211,159,209,211,149,
209,205,171,204,217,207,219,40,32,41,181
,191,181
60 FOR a=0 TO 3: PLOT 468,180-6*a,1: DRAW
R -20,4: DRAW -20,20: PLOT 508,180-6*a: DRA
WR 20,4: DRAW 20,20: NEXT
70 PLOT 488,240,2: DRAW 0,100
200 GOTO 200

```

Teil 4

```

1 ' Grafikgags 2 - Teil 4
10 MODE 1:INK 0,13: BORDER 13:INK 1,0:INK
2,26:INK 3,7
20 FOR a=-380 TO 1000 STEP 20
30 ORIGIN 0,0,0,640,170,0: PLOT a,0,1: DRA
W 320,300
40 ORIGIN 0,0,0,640,400,230: PLOT a,400: D
RAW 320,100
50 NEXT
60 ORIGIN 0,0,0,640,400,0
70 x=200: FOR a=1 TO 80: PLOT 0,201-x: DRA
W 640,0: PLOT 0,201+x: DRAW 640,0: x=x/1.1
: NEXT
80 FOR a=0 TO 6.4 STEP 0.3: FOR c=0 TO 1
90 RESTORE: PLOT 300-(2*c-1)*SIN(a)*270+c
*10,220*c-(2*c-1)*COS(a)*80+94,2+c
100 FOR b=0 TO 7: READ x,y: DRAW 5*a*x,5*
-(2*c-1)*a*y/2
110 NEXT b,c,a
120 DATA 2,4,-2,0,2,-4,0,4,-1,2,-1,-2,0,
-4,2,0
200 GOTO 200

```

Teil 5

```

1 ' Grafiggags 2 - Teil 5
10 MODE 1: BORDER 2: INK 0,2: INK 1,14: INK
2,26: INK 3,13
20 FOR a=-160 TO 160 STEP 8
30 PLOT a*2+320,400,1: DRAW a+320,250: DRA
WR 0,-100: DRAW a*2+320,0: NEXT
40 FOR a=-80 TO 80 STEP 16
50 PLOT 0,a*2+200: DRAW 160,a/2+200: DRAW
R 320,0: DRAW 640,a*2+200: NEXT

```

```

60 x=20: FOR a=1 TO 53: y=180-x: PLOT y,y: D
RAW 640-2*y,0: DRAW 0,400-2*y: DRAW 2*y
-640,0: DRAW 0,2*y-400: x=x*1.1: NEXT
70 FOR a=316 TO 324 STEP 2: PLOT a,40,3: D
RAW 0,140: NEXT: PLOT 320,40,2: DRAW 0,14
0
80 FOR a=0 TO 2*PI STEP 0.5: PLOT 320,220
,2: DRAW SIN(a)*40,COS(a)*40: FOR b=0 TO
2*PI STEP 0.5: PLOT 320+SIN(a)*40,220+CO
S(a)*40: DRAW SIN(b)*20,COS(b)*20,INT(RND
*2+2): NEXT b,a
90 FOR b=100 TO 200 STEP 20: PLOT b,b/2,2
: DRAW 0,30: FOR a=0 TO 2*PI STEP 0.5: PLO
T b,b/2+30: DRAW SIN(a)*20,COS(a)*20,INT
(RND*2+2): NEXT a,b
200 GOTO 200

```

Teil 6

```

1 ' Grafikgags 2 - Teil 6
10 RANDOMIZE 1000
20 MODE 1: INK 0,1: BORDER 1: INK 1,0: INK 2
,24: INK 3,13
30 PEN 3: FOR a=11 TO 25: LOCATE 1,a: PRINT
STRING$(40,CHR$(207));: NEXT
40 FOR f=0 TO 1: x=0: WHILE x<640: w=RND*20
+20: y=RND*50+50: x=x+w+4: FOR a=1 TO w STE
P 2: PLOT a+x-w,292-f*50,1: DRAW 0,y: PLOT
a+x-w,184+f*50,3: DRAW 0,-y
50 FOR b=1 TO 3: q=RND*y: PLOT a+x-w,q+292
-f*50,2: PLOT a+x-w,184-q+f*50: NEXT b,a: P
LOT x-w,292-f*50,3: DRAW 0,y: DRAW w+2,0
: DRAW 0,-y: WEND: NEXT
60 FOR a=0 TO 200: PLOT RND*630,RND*240,0
: DRAW 6,2: PLOT RND*630,RND*240,3: DRAW
6,-2: NEXT
200 GOTO 200

```

Teil 7

```

1 ' Grafikgags 2 - Teil 7
10 RANDOMIZE 1000
20 MODE 1: INK 0,0: BORDER 0: INK 1,1: INK 2
,2: INK 3,14
30 x=10: FOR a=0 TO 640 STEP 2: PLOT a,240
,1: DRAW 0,x: x=x+RND*4-2-2*(x<3): PLOT RN
D*640,RND*120+290,RND*3+1: NEXT
40 FOR a=11 TO 25: LOCATE 1,a: PRINT STRIN
G$(40,CHR$(143));: NEXT
50 FOR b=0 TO 2: FOR a=1 TO 100+50*b STEP
2: PLOT 100+b*200-a/2,300-a-b*50,2: DRAW
a,0: DRAW a/6,a/5,3: NEXT a,b
60 FOR b=0 TO 1: FOR a=1 TO 100+50*b STEP
2: PLOT 50+b*175,100+a-b*150,0: DRAW a,0
: NEXT a,b
70 FOR a=0 TO PI*2 STEP 0.2: PLOT 472,360
,3: DRAW SIN(a)*30,COS(a)*30: NEXT
80 WHILE INKEY$="": WEND: INK 1,1: INK 2,2:
INK 3,14: WHILE INKEY$="": WEND: INK 1,3: IN
K 2,6: INK 3,16: WHILE INKEY$="": WEND: INK
1,15: INK 2,24: INK 3,25: GOTO 80
200 GOTO 200

```

Extended Basic

23 neue RSX-Befehle

464

664

6128

In dieser Serie wollen wir das Basic des Schneider Computers noch um einiges aufwerten. Es sollen pro Folge etwa 10 – 25 neue Befehle eingerichtet werden. Klar ist, daß wir diese Serie nicht in alle Ewigkeit fortsetzen können, denn irgendwann fallen einem ja keine neuen, sinnvollen Befehle mehr ein, die zudem von der Programmierung nicht zu umfangreich sein sollten. So werden wir diese Serie wohl auf 3 – 4 Teile beschränken. Wem noch sinnvolle Befehle zusätzlich einfallen, kann uns schreiben. Wir bringen dann die entsprechenden Befehlserweiterungen.

Neu an der Serie soll die vollständige Kompatibilität sein. Während die Befehle eines jeden Teiles auch unabhängig von den anderen Teilen laufen, ist es auch möglich, mit allen Teilen gleichzeitig zu arbeiten.

Jeder Teil hat einen gewissen Schwerpunkt. Diesmal geht es um die Hardware. Vieles was im Inneren des Rechners vorgeht, und was in seinem Speicher steht, kann durch diese 23 Befehle beeinflusst werden. Der nächste Teil beschäftigt sich dann vornehmlich mit Grafik: Zeichnen von Dreiecken, Vierecken und vieles andere.

Schwerpunkt Hardware

Die Befehle werden auf der Basis von RSX-Befehlen eingerichtet. Ihnen ist ein senkrechter Strich vorangestellt, den Sie mit der Tastatur über Shift + Klammeraffe @ erreichen. Den Befehlen können mehrere Parameter übergeben werden (durch Kommata voneinander getrennt). Beachten Sie dabei, daß schon bei einem Parameter direkt hinter dem Befehl ein Komma stehen muß. Beispiel **IPAUSE**, Zeit. Dem Befehl muß auch öfters der Speicherplatz einer Variablen mitgeteilt werden. Dies geschieht über den Klammeraffen. Z. B. ergibt **PRINT @i%** die Speicherzelle, ab der die Variable **i%** im Speicher abgelegt ist. Existiert eine Variable mit diesem Namen nicht, gibt der Rechner eine Fehlermeldung aus. Um das zu vermeiden, setzen Sie die Variable vorher auf einen beliebigen Wert (z. B. auf **i%=0**). Beachten Sie, daß das Prozentzeichen (%) eine Integervariable kennzeichnet.

In der Auflistung der neuen Befehle stehen verschiedene Parameter hinter den Befehlen. Dabei bedeuten **n1**, **n2**, **n3**,... direkte Zahlen. D. h., man kann dort eine direkte Zahl hinschreiben, wie etwa 10000 oder eine Variable (Integer oder Real) wie **i** oder **%**.

Zu beachten ist dabei, daß der Wert im Bereich von 0 bis +65535 liegen muß. Integervariablen haben allerdings nur den Wertebereich -32767 bis +32767. Um eine negative Zahl in eine positive umzurechnen, macht man folgendes: **PRINT 65536+i%** und schon kommt eine positive Zahl heraus. Für den Rechner ist es absolut egal, ob er mit positiven oder mit negativen Zahlen rechnet, für ihn sind sie identisch. Um eine positive Zahl in eine negative umzurechnen, so daß sie in eine Integervariable paßt, macht man folgendes: **i%=<zahl>-65536**.

Die zweite Art von Parametern, die übergeben werden, sind Variablen mit vorangestellten Klammeraffen. Hier dürfen nur Integervariablen stehen. Beispiel: **@i%**. Werden einem Befehl zuviel oder zuwenig Parameter übergeben, so kommt zwar keine Fehlermeldung, der Befehl wird aber nicht ausgeführt.

Für die Computer CPC 664 und 6128 müssen Sie zusätzlich folgende Zeile eingeben:

```
85 POKE &A5E3,&32:POKE &A5E4,&B6:POKE &A5E9,&32:POKE &A5EA,&B6:POKE &A5EF,&31:POKE &A5F0,&B6:POKE &A5F5,&31:POKE &A5F6,&B6
```

Tippen Sie das abgedruckte Listing sorgfältig ab und starten Sie es mit **RUN**. Der Rechner überprüft nun die Datenzeilen. Zuerst wird die Nummer der gerade überprüften Zeile gedruckt, dann ein Kommentar. Schreibt der Rechner **OK**, so ist die Zeile in Ordnung, die nächste wird untersucht. Drückt der Rechner **ERROR** oder eine Basicfehlermeldung wie **TYPE MISMATCH**, so ist die Zeile nicht korrekt. Diese müssen Sie dann überprüfen und verbessern und danach das Programm erneut mit **RUN** starten. Das tun Sie solange, bis alle Fehler eliminiert sind. Der Rechner schreibt nun **ACHTUNG: ABSPEICHERN**. Legen Sie eine Cassette oder Diskette ein und drücken Sie **REC & PLAY** (nur TAPE), anschließend eine Taste. Das Maschinenprogramm wird nun abgespeichert. Wollen Sie das alte Programm mit den Datas behalten, eventuell um jemand anderen eine Kopie zu überspielen, speichern Sie es mit **SAVE "EBAS1"** ab. Das Maschinenprogramm liegt nun unter dem Namen **EXTBAS1.BIN** vor.

Wollen Sie die Befehlserweiterung anwenden, müssen Sie folgendermaßen vorgehen:

1. Rechner zurücksetzen (ist nicht unbedingt notwendig).
2. **MEMORY 41899** – Speicherplatz für MC reservieren
3. **LOAD "EXTBAS1.BIN"** – MC-Programm laden
4. **CALL 41900** – Die neuen Befehle einrichten

Von nun an befinden sich die neuen Befehle im Speicher. Bei **CTRL+SHIFT+ESC** oder beim Ausschalten des Rechners gehen sie allerdings verloren, nicht jedoch bei **NEW**. Das Maschinenprogramm belegt den Speicher von 41619 bis 41900.

IBLOCKFILL,n1,n2,n3,: Der Speicher wird von Adresse **n1** mit **n2** Bytes vom Wert **n3** aufgefüllt (1-Byte-Wert).

IBYTECOUNT,n1,n2,n3,@a%: Ab Adresse **n1** werden **n2** Bytes auf den Wert **n3** untersucht. In der Variablen **a%** befindet sich nach Ausführung des Befehls die Anzahl der Bytes, die im angegebenen Bereich vorhanden waren.

ICAPSON: Der Capsmodus (Großbuchstaben) wird angeschlossen.

ICAPSOFF: Der Capsmodus (Großbuchstaben) wird abgestellt.

IDEEK,n1,@a%: **DEEK** steht für **DOUBLE PEEK**. Ab der Adresse **n1** werden zwei Bytes aus dem RAM gelesen und an die Variable **a%** übergeben. Dabei entspricht das High-

byte dem Wert der Adresse n1+1 und das Lowbyte dem Wert der Adresse n1.

IDOKE,n1,n2: DOKE steht für DOUBLE POKE. Ab Adresse n1 wird zuerst Low, dann Highbyte von n2 im Speicher abgelegt.

IDRUCKER,@a%: In der Variablen a% wird der Druckerstatus übergeben. Ist kein Drucker angeschlossen oder ein vorhandener Drucker nicht empfangsbereit, liefert der Befehl in a% eine 0, ist der Drucker empfangsbereit, ist a% gleich 1.

Extended Basic

```

10 REM =====
20 REM           Extended Basic Teil 1
30 REM =====
40 MEMORY 41899
50 pc=41900
60 FOR ze=10000 TO 10060 STEP 10:PRINT z
e;:sum=0:READ x$,s
70 FOR a=0 TO 99:a$=MID$(x$,a*2+1,2):POK
E pc,VAL("&"+a$):sum=sum+VAL("&"+a$):p
c=pc+1:NEXT a:IF sum<>s THEN PRINT "ERRO
R...":END
80 PRINT "OK":NEXT ze:PRINT "Achtung Abs
aven : SAVE 'EXTBAS1.BIN',b,41900,700"
90 SAVE "EXTBAS1.BIN",b,41900,700
10000 DATA "21CAA30122A43E16E5C5F5CDD1BC
F1C1E10303030303232323233D20EBC900000000
00000000000000000000000000000000000000
00000000000000000000000000000000000000
00000000000000000000000000000000000000
00000000000000000000000000000000000000
000000000000", 3297
10010 DATA "0000000000000000000000000000
0000000090A4C325A594A4C338A598A4C36EBC9F
A4C371BCA7A4C354A5B0A4C36EA5B9A4C397A5BC
A4C3ABA5C0A4C3B9A5C7A4C319BDCEA4C3CCA5D3
A4C3E0A5D9A4C3E6A5E0A4C3ECA5E7A4C3F2A5EF
A4C3F8A5F6A4", 14325
10020 DATA "C30EA6FDA4C329A604A5C337BD0D
A5C345A614A5C358A61BA5C300BB444F4BC54445
45CB4D4F544F524FCE4D4F544F524F46C6424C4F
434B46494CCC42595445434F554ED44C4FD74849
47C84C4F57484947C857414954464CD950415553
C5434150534F", 10495
10030 DATA "CE434150534F46C653484946544F
CE53484946544F46C6445255434B45D2524F4D50
4545CB524F4D444545CB56454B544F525245D357
4149544B45D9544553544B45D954415354524553
45D400FE02C0DD5E00DD5601DD6E02DD66037323
72C9FE02C0DD", 10089
10040 DATA "6E00DD6601DD5E02DD5603D5DDE1
DD7E007723DD7E0177C9FE03C0DD5600DD4E02DD
4603DD6E04DD660572230B78B120F9C9FE04C011
0000DD4E04DD4605DD6E06DD6607DD7E02BE2001
13230B78B120F3DD6E00DD6601732372C9FE02C0
DD6601DD6E00", 11007
10050 DATA "DD5E0273233600C9FE02C0DD6601
DD6E00DD560372233600C9FE02C0DD6601DD6E00
DD5E02DD5603722373C9FE01C218BDD5E00DD56
01CD19BD1B7AB320F8C93EFF32EBB4C93E0032EB
B4C93EFF32E7B4C93E0032E7B4C9FE01C0CD2EBD
3FCB17E601DD", 12281
10060 DATA "6601DD6E0077233600C9FE02C0DD
6603DD6E02CD00B9CD06B946DD6601DD6E007023
3600C9FE02C0DD6603DD6E02CD00B9CD06B94623
4EDD6601DD6E00702371C9FE01C218BBD18BBD
6E00DD660177233600C9FE01C218BBD4600CD18
BBB820FAC900", 11229
    
```

IHIGH,n1,@a%: Dieser Befehl liefert das Highbyte von n1 in der Variablen a%.

ILOW,n1,@a%: Dieser Befehl liefert das Lowbyte von n1 in der Variablen a%.

ILOWHIGH,n1,@a%: Das Lowbyte von n1 wird im Highbyte von a% übergeben, und das Highbyte von n1 wird im Lowbyte von a% übergeben.

IMOTORON: Der Motor des Cassettenteils wird angestellt.

IMOTOROFF: Der Motor des Cassettenteils wird abgestellt.

IPAUSE,n1: Der Rechner wartet n1/fünzigstel Sekunden. Um also eine Pause von einer Sekunde zu erreichen, wäre der Befehl IPAUSE,50 notwendig.

IROMDEEK,n1,@a%: Dieser Befehl wirkt ebenso, wie der Befehl IDEEK, bloß daß jeweils die Werte aus dem ROM gelesen werden.

IROMPEEK, n1, @a\$: Wie IROMDEEK, allerdings wird nur ein Byte aus der Adresse n1 ausgelesen und in das Lowbyte von a% übergeben.

ISHIFTON: Der Shiftmodus wird angestellt.

ISHIFTOFF: Der Shiftmodus wird abgestellt.

ITASTRESET: Die Tastatur wird zurückgesetzt. Dieses hat zur Folge:

1. Der Tastenbuffer wird geleert.
2. Die Tastenvektoren werden auf ihre Ursprungswerte gesetzt.
3. Daueranschlag und Wiederholungsgeschwindigkeit werden auf ihre Ursprungswerte gesetzt.
4. Die Tastendefinition wird zurückgesetzt.
5. CAPS und SHIFT sind aus.

ITESTKEY,n1: Der Rechner wartet solange, bis der Benutzer ein Zeichen mit dem ASCII-Code n1 eingetippt hat. Wird kein Parameter übergeben, wartet der Rechner nur auf den Druck irgendeiner Taste.

IVEKTORRES: Alle Vektoren werden auf ihre Ursprungswerte zurückgesetzt.

IWAITFLY: Dieser Befehl wartet auf den nächsten Rückstrahl des Elektronenstrahls des Monitors.

IWAITKEY,@a%: Wartet auf einen Tastendruck und übergibt den ASCII-Wert der Taste der Variablen a%. Ohne Parameter wird nur auf einen Tastendruck gewartet.

1. Seepferdchen/Apfelmann
mathematische Chaosgrafik
in Maschinensprache 20,- DM

2. Multistift/Screen Split
27 Farben, 3 Modes gleichzeitig
das älteste Programm dieser Art
jetzt bei NO-DATA 20,- DM

**3. Polygon-
Grafikbefehlsenerweiterung**
(kein Maiprogramm) bel. Vielecke,
Kreisbogen, Ellipsen in Schraglage, Flächen,
füllen und vieles mehr 20,- DM

4. Holodisc (in Vorbereitung)
alle 42 Spuren lesen, editieren, formatieren,
kopieren 20,- DM

**außergewöhnliche
Programme
für den CPC 464**
Softwareautoren gesucht

NO-DATA
Christine Widdel
Fraunhoferstraße 8
3000 Hannover 1



Alle Programme auf Diskette übertragbar
Alle Preise incl. Versand und Mehrwertsteuer

Schriften mischen mit dem CPC 464

Je nach Mode wird der Inhalt des Bildschirmspeichers vom CPC 464 ganz unterschiedlich interpretiert. Das ist eine unveränderliche Eigenschaft der Hardware und es ist ein erheblicher Software-Aufwand erforderlich, um den Bildschirm-RAM mit dem nötigen Inhalt zu versorgen, damit überhaupt lesbare Zeichen entstehen. Unter anderem muß deshalb bei jedem MODE-Kommando der Bildschirm völlig gelöscht werden. Die Software benutzt für die Anpassung sogenannte Bitmasken, die beim CPC 464 im RAM abgelegt sind, so daß wir mit POKEs den Vorgang beeinflussen können.

Es ist auf diesem Wege möglich, in MODE 2 und MODE 1 auch die breiteren Schrifttypen zu erzeugen. Schmalere Schrift scheidet an der Hardware, die ja eine bestimmte Pixelgröße vorgibt. Mit den zwei Unterprogrammzeilen in MODE 1 und den drei Unterprogrammzeilen in MODE 2 handelt es sich vermutlich um die einfachste Methode, das Erscheinungsbild einer Textausgabe zu verbessern. Etwas Geschick erfordert es, die richtige Ausgabeposition mit LOCATE-Befehlen herzustellen, denn die Bildschirmposition wird nach dem simulierten MODE berechnet. Auch die PEN- und PAPER-Masken sowie die WINDOW-Grenzen sind vom MODE abhängig, so daß Vorsicht angebracht ist.

Die eigentliche Textausgabe ist dagegen sehr einfach. Mit der entsprechenden Unterprogrammzeile muß man die Bitmasken für den gewünschten MODE erzeugen, den PRINT-Befehl geben und mit einem weiteren GOSUB die Bitmasken auf ihren Standardwert zurücksetzen. Falls bei Experimenten die Ready-Meldung unleserlich erscheint, hilft ein reguläres MODE-Kommando, das in diesem Falle „blind“ eingegeben wird. Die Unterprogrammzeilen, die ohne Probleme in jedes Basicprogramm eingebaut werden können, sind in REM-Zeilen erklärt. Die Anwendung ist aus dem Beispielprogramm ersichtlich.

Gerhard Knapieski

```

10 MODE 1
20 LOCATE 5,5:PRINT "Die interessanteste
n
30 GOSUB 800:LOCATE 5,10:PRINT "Tips";:G
OSUB 810:PEN 3:GOSUB 800:PRINT " & ";:GO
SUB 810:PEN 1:GOSUB 800:PRINT "Tricks":G
OSUB 810
40 LOCATE 14,15:PRINT "monatlich in:"
50 LOCATE 12,23:PRINT "<Taste druecken>"
60 CALL &BB18:MODE 2
70 GOSUB 900:LOCATE 4,5:PRINT "CPC - MAG
AZIN"
80 GOSUB 910:LOCATE 7,7:PRINT "fuer alle
Schneider Computer":GOSUB 920
90 LOCATE 13,10:PRINT "E i n e":GOSUB 91
0:LOCATE 14,10:PRINT "Zeitschrift,"
100 GOSUB 920:LOCATE 13,12:PRINT "b e i
d e r d i e":GOSUB 910:LOCATE 21,12:
PRINT "Leser"
110 LOCATE 7,14:PRINT "mitarbeiten":GOSU
B 920:LOCATE 37,14:PRINT "k o e n n e n
-"
120 CALL &BB18:GOTO 10
190 END
795 REM ***** Schriftbreite von MODE 0 b
ei Grund-MODE 1
800 POKE &B1C8,0: POKE &B1CF,&CC: POKE &
B1D0,&33: RETURN
805 REM ***** Ausgangszustand bei Grund-
MODE 1
810 POKE &B1C8,1: POKE &B1CF,&88: POKE &
B1D0,&44: POKE &B1D1,&22: POKE &B1D2,&11
: RETURN
895 REM ***** Schriftbreite von MODE 0 b
ei Grund-MODE 2
900 POKE &B1C8,0: POKE &B1CF,&FO: POKE &
B1D0,&F: RETURN
905 REM ***** Schriftbreite von MODE 1 b
ei Grund-MODE 2
910 POKE &B1C8,1: POKE &B1CF,&CO: POKE &
B1D0,&30: POKE &B1D1,&C: POKE &B1D2,3: R
ETURN
915 REM ***** Ausgangszustand bei Grund-
MODE 2
920 POKE &B1C8,2: POKE &B1CF,&80: POKE &
B1D0,&40: POKE &B1D1,&20: POKE &B1D2,&10
: POKE &B1D3,8: POKE &B1D4,4: POKE &B1D5
,2: POKE &B1D6,1: RETURN

```

CPC-Vokabi

Vokabeltrainer, für Sprachen des lateinischen Alphabetes

Vokabi, der ideale Partner für Schüler, Studenten und VHS'ler, die parallel zum Unterricht Vokabeln pauken müssen.

Nach Eingeben eines Kapitels fragt Sie **Vokabi** systematisch ab, bis Sie alle Vokabeln lesen und schreiben können.

Weitere Leistungsmerkmale:

- einfache Bedienung
- Fehleranzeige
- ständige Lehrstandskontrolle (abspeicherbar)
- zusätzliche Sonderzeichen in den Sprachen Französisch, Dänisch, Schwedisch, Italienisch, Spanisch
- Ausdrucken in Normal- und Spickzettelform (Epson)
- Suchen innerhalb eines Kapitels, u.v.m.

Vokabi, Kass. nur **48.- DM**, Disk nur **58.- DM**

Fa. Elfriede VAN DER ZALM – Softwareentwicklung & Vertrieb
Schieferstätte, 2949 Wangerland 3.

Für ganz Eilige: ☎ 064 82/42 37 – 044 61/7 17 19 ab 17.00 Uhr

haaga

Software
Roßstr. 4, 7080 Aalen
Tel. 07361/61981

Light Pen	98 DM
3-D-Voice Chess	48 DM
Alien 8	38 DM
American Football	38 DM
Decathlon	36 DM
Fighter Pilot	32 DM
Font 64	32 DM
Hunchback	28 DM
Hunter Killer	32 DM
Jet Set Willy	35 DM
Jump Jet	40 DM
Knight Lore	38 DM
Minder	32 DM
Pyjarama	32 DM
Slapshot	36 DM
Starion	32 DM
Nightshade	38 DM
Technican Ted	28 DM
The Hobbit	48 DM
Way of explod. Fist	38 DM

Eprom- Programmer

für 2716, 2732, 2764
und 27128

komplett mit Software,
eigener Stromversorgung,
ansprechendes Gehäuse

nur DM 198.-

zuzügl. Versandkosten: Nachnahme 6 DM; Vorkasse 2,50 DM. Katalog kostenlos.
Dipl.-Kfm. Peter Haaga, PF 13 23, 7080 Aalen, Tel. (07361) 61981

Der File-Examiner

464

664

6128

Tippen Sie Listing 1 ein und save Sie es mit GOTO 10000 ab. Starten Sie dann das Programm und laden Sie den Header eines zu untersuchenden Programmes in den Rechner. Es ist egal, um welchen Block es sich dabei handelt. Sobald der Rechner den Namen des Files anzeigt, drücken Sie zweimal ESCAPE. Nun wertet der Rechner das File aus. Er gibt Name, Länge, Startadresse, Autostart und Bemerkungen aus. So können alle wichtigen Informationen von einem auf Band abgespeicherten File herausgeholt werden. Diskettenprogramme kann Examiner nicht untersuchen.

Andreas Zallmann

Examiner

```

10 SYMBOL AFTER 255:SYMBOL 255,0,0,0,
0,0,0,66,126
20 MODE 1:PEN 1
30 PRINT "-----"
-----";:PEN 2:PRINT "          CPC 46
4 - HEADER READER"
40 PEN 1:PRINT "-----"
-----";:PEN 2:PRINT "          (C)
1985 by Andreas Zallmann."
50 PEN 1:PRINT "-----"
-----"
60 WINDOW 1,40,7,25
70 GOSUB 10000
80 PEN 1
90 ON BREAK GOSUB 5000
100 CAT
5000 CLS:PRINT:PRINT
5010 a=PEEK(47262):b=(a AND 14)/2
5020 IF b=0 THEN PRINT "Programm:";
5030 IF b=1 THEN PRINT "Bytes:";
5040 IF b=2 THEN PRINT "Screen:";
5050 IF b=3 THEN PRINT "ASCII:";
5060 n$=""
5070 FOR i=47244 TO 47259
5080 IF PEEK(i)=0 THEN POKE i,255
5090 n$=n$+CHR$(PEEK(i))
5100 NEXT i
5110 PRINT n$;
5120 PRINT " Block:";PEEK(47260)
5121 PRINT:PRINT:PRINT"Startadresse
: ";PEEK(47265)+256*PEEK(47266)
5130 PRINT:PRINT "Laenge des Blockes
: ";PEEK(47263)+256*PEEK(47264)
5140 PRINT:PRINT "Laenge des ganzen File
s : ";PEEK(47268)+256*PEEK(47269)
5145 PRINT:PRINT"Auto-Start
: ";PEEK(47270)+256*PEEK(47271)
5150 PRINT:PRINT:PRINT "Bemerkungen: ";
5160 IF PEEK(47261)<>0 AND PEEK(47267)<>
0 THEN PRINT "Der einzige Block":GOTO 51
80
5170 IF PEEK(47261)<>0 THEN PRINT "Der 1
etzte Block" ELSE IF PEEK(47267)<>0 THEN
PRINT "Der erste Block" ELSE PRINT "Wed
er erster noch","          letzter Bl
ock"

```

```

5180 c=a AND 1
5190 IF c=1 THEN PRINT "          Ges
chuetztes Programm" ELSE IF b=0 THEN PRI
NT "          Ungeschuetztes Programm
"
5200 LOCATE 12,19:PEN 2:PRINT "E-ENDE W
-WEITER":PEN 1
5210 i$=UPPER$(INKEY$)
5220 IF i$="E" THEN MODE 1:END ELSE IF i
$="W" THEN CLS:GOTO 100 ELSE 5210
10000 CLS:PRINT:PRINT "Druecken Sie eine
Taste um das Band zu Starten. Nach dem
Einlesen des Namens koennen sie 2 x >
ESCAPE< druecken, um alle Informatio
n ueber das File zu er-halten."
10010 PRINT:PRINT"Druecken Sie nach der
Ausgabe der In- formationen >W< um wei
terzulesen oder >E< um das Programm zu
beenden."

```

```

10020 LOCATE 7,19:PEN 2:PRINT "Bitte ein
e Taste druecken."
10030 WHILE INKEY$<>"":WEND:WHILE INKEY$
="" :WEND:CLS:RETURN

```

3 Zoll Disketten für den CPC 464/CPC 664/CPC 6128

Preis pro Stück bei		
1-9	10-99	ab 100
13.00 DM	12.50 DM	12.00 DM

Bestellungen unter 100.- DM + 5.- DM
Versandkosten?
Keine Programmabbrüche mehr durch
Disk-Fehler beim CPC 464 mit DDI-1!
Für Ihre eigenen Programme gibt es jetzt
einen Patch, der Disk-Fehler beherrsch-
bar macht. Zusätzlich kann die Meldung
über fehlende Disketten unterdrückt
werden. Man kann feststellen, in wel-
chen Laufwerken Disketten sind, den
Schreibschutz prüfen und die Zahl der
freien Blöcke abfragen.
Auf Kassette 30.- DM, Diskette 40.- DM!
Natürlich mit deutschem Handbuch. Es
werden keine Versandkosten berechnet!
Dietmar Brüggendiek Software, EDV-
Zubehör, Postfach 520119, 4600
Dortmund 50, ☎ 0231/736269.

ABACUS

SOFTWARE

BOUTIQUE

PROGRAMME, BÜCHER UND ZUBEHÖR
FÜR

COMMODORE 64/VC 20

SCHNEIDER CPC

ABACUS SOFTWARE-BOUTIQUE
VERTRIEBS-GMBH

ESCHERSHEIMER LANDSTR. 84 6000 FFM 1
(U1, 2, 3 GRÜNEBURGWEG) TEL.: 069/594019
GEÖFFNET: 11-18³⁰ SA 10-13

CPC 464 FORTH

Erleben auch Sie auf Ihrem CPC 464 die Geschwindigkeit eines schnellen FORTH-Compilers mit Turtlegrafik, Editor, Assembler, Tracer und De-Compiler. Dieses System ist im neuesten FORTH 83 Standard geschrieben und erzeugt kompakte Programme. Die Grafik ist um Kreis- und Füllbefehle erweitert, das System setzt ebenfalls Windows ein. Das Programm wird mit einem 180-seitigen deutschen Handbuch geliefert.

Preis auf Kassette
auf Disk 3", 5,25"

DM 148.-
DM 178.-

Ausführliches Prospektblatt bei:

FORTH-SYSTEME Angelika Flesch
Postfach 1226, 7820 Titisee-Neustadt
Telefon 07651 / 1665

XFormat, Hexamon, RAM/ROM-Überblick

Auch diesmal bringen wir wieder drei Programme für Sesam. Allerdings ist zum Verständnis die Einführung aus Heft 12/85 ab Seite 44 erforderlich. Wer das Heft nicht hat, kann es bei uns gegen 7 DM in Briefmarken anfordern (incl. Porto).

Wer mit seinen Diskettenexperimenten etwas weitergehen möchte, hat mit dem SESAM-Programm gute Chancen. Mit etwas Geschick können in Basic mit dem Erweiterungsbefehl XCALL eigene Diskettenformate erstellt, zwei zusätzliche Spuren genutzt und die Geheimnisse geschützter Diskettensoftware ergründet werden. Alle Anga-

9 Kilobyte mehr

ben gelten für die Schneider-Floppy (soweit bekannt auch für den 664), für andere Laufwerke müssen passende Einsprungadressen aus dem Controller-ROM ermittelt werden.

Fehlermeldungen einschalten:

```
IXCALL,&CA72,7,0
```

Fehlermeldungen ausschalten:

```
IXCALL,&CA72,7,1
```

Sektor lesen:

```
IXCALL,&C666,7,0,sec+off,256*track+drive,buffer
```

Sektor schreiben:

```
IXCALL,&C64E,7,0,sec+off,256*track+drive,buffer
```

Sektor-ID lesen:

```
IXCALL,&C55D,7,0,0,drive
```

Track formatieren:

```
IXCALL,&C652,7,0,sec+off,256*track+drive,formatbuffer
```

sec ist dabei die Nummer eines Sektors von 1 bis 9, off ist der vom Format abhängige Erkennungsoffset: 0 für IBM-Format, &40 für CP/M-Format und &C0 für das Datenformat. drive ist 0 für Laufwerk A und 1 für Laufwerk B, track ist die gewünschte Spurnummer von 0 bis 39 (&27) oder maximal 41 (&29), buffer ist die Adresse eines RAM-Bereiches für die jeweiligen Daten. Die Sektor-ID ist eine Erkennungsmarke von 4 Byte, der Reihe nach: Spurnummer (track), Kopfadresse (immer Null), Sektornummer (sec+off) und Sektorgröße (immer 2, entsprechend 512 Bytes).

Zum Formatieren einer Spur muß ein Formatbuffer mit den Sektor-IDs aller Sektoren (also $4*9=36$ Bytes) bestehen. Eine Schwierigkeit ist, daß die Sektoren in einer eigenwilligen Reihenfolge formatiert werden, um ein besseres Timing für die Schreib/Lesezugriffe zu erzielen. Bastler haben da ungeahnte Möglichkeiten, für einfache Anwendungen nimmt XFORMAT die Schwerarbeit ab. Das kleine Programm ergänzt bereits formatierte Disketten mit den beiden zusätzlich möglichen Spuren, die nötige Formattabelle wird aus den schon vorhandenen Spuren abgeleitet.

XFORMAT ist für das Datenformat geschrieben, für ein anderes Format muß der off-Wert in Zeile 10 verändert werden. Wir erkennen im einzelnen die Befehle für Fehlermeldungen wie Abschalten, Sektor lesen (damit eine definierte Spur eingestellt wird, der Inhalt des Sektors interes-

siert nicht weiter) und in einer FOR-NEXT-Schleife für Sektor-ID lesen. Die Sektor-ID wird an der festen RAM-Adresse &BE4F abgelegt, wo sie mit IXPEEK abgeholt werden kann. Ab Zeile 70 wird der richtige Tabellenanfang ermittelt (für Sektor 1), die neue Tracknummer gepoked und der neue Track formatiert. Das Ganze zweimal, anschließend wird sicherheitshalber mit einem Lesebefehl das Laufwerk auf eine normale Spur zurückgestellt und die Fehlermeldungen werden wieder zugelassen.

Mit diesem Programm können auch versteckte Sektoren ausfindig gemacht werden. Wir müssen nur in Zeile 30 track die Werte &28 und im zweiten Durchlauf &29 einsetzen. Zeile 70 wird ersetzt durch track=0:GOTO 180 (alles dazwischen kann auch weggelassen werden). Das Laufwerk wird vergeblich versuchen (deutlich hörbar), einen Sektor auf dieser Spur zu lesen. Im Formatbuffer stehen jedoch anschließend die ID-Werte aller doch vorhandenen Sektoren, die dann mit diesen Werten und dem Befehl für „Sektor Lesen“ ins RAM geholt werden können.

Zurück zu den selbsterzeugten zusätzlichen Spuren. Der CAT-Befehl zeigt weiterhin nur 178 K free, wo bleiben die zusätzlichen 9 K? Diese müssen erst der Diskettenverwaltung mitgeteilt werden. Dazu brauchen wir zunächst die RAM-Adressen, die der Controller benutzt. Bekanntlich

XFormat

```
5 MEMORY &BFFF
10 drive=0:sec=1:track=0:off=&C0:formatbuffer=&9000:buffer=&9200
20 :XCALL,&CA72,7,1
30 :XCALL,&C666,7,0,sec+off,256*track+drive,buffer
40 FOR t=1 TO 20
50 :XCALL,&C55D,7,0,0,drive:IXPEEK,&BE4F,&FF,formatbuffer,4
60 formatbuffer=formatbuffer+4:NEXT
70 formatbuffer=&9000
80 IF PEEK(formatbuffer+2)=sec+off THEN
100 ELSE formatbuffer=formatbuffer+4
90 IF formatbuffer<&9028 THEN 80 ELSE PRINT "Fehler"
100 FOR j=0 TO 8
110 POKE formatbuffer+4*j,&28
120 NEXT
130 :XCALL,&C652,7,0,sec+off,256*&28+drive,formatbuffer
140 FOR j=0 TO 8
150 POKE formatbuffer+4*j,&29
160 NEXT
170 :XCALL,&C652,7,0,sec+off,256*&29+drive,formatbuffer
180 :XCALL,&C666,7,0,sec+off,256*track+drive,buffer
190 :XCALL,&CA72,7,0
```

wird ein RAM-Bereich bei der Initialisierung (in der Regel beim Einschalten) zugeteilt und kann je nachdem, ob weitere Erweiterungen angeschlossen sind, verschoben werden. Die Floppy benutzt deshalb auch ein paar feste RAM-Adressen, die unter anderem auf den Variablen-Bereich zeigen.

```
floppyram = 256 * PEEK (&BE7E) + PEEK (&BE7D)
dpbram = 256 * PEEK (&BE43) + PEEK (&BE42)
```

In diesem Fall interessiert uns dpbram, die Adresse des Disk-Parameter-Blocks. floppyram ist der Basiswert des gesamten benutzten RAM-Bereiches (normalerweise &A700) von insgesamt &500 Bytes. Eine erweiterte Diskette können wir nun so benutzen: Zuerst ein CAT-Befehl, damit die sonstigen Formatwerte richtig eingestellt werden, danach muß mit POKE dpbram+&18,&FF das sogenannte Login abgeschaltet werden, damit nicht bei jedem Diskettenzugriff der Standardwert wieder hergestellt wird. Wenn jetzt mit

```
POKE dpbram + 5 + drive * &40, PEEK (dpbram + 5 + drive * &40) + 9
```

die Standardblockanzahl um 9 erhöht wird, stehen die 9 K-Byte zur Verfügung. Der CAT-Befehl zeigt jetzt 187 K free. Sobald wieder mit anders formatierten Disketten gearbeitet werden soll, kann mit POKE dpbram+&18,0 das Login und damit die automatische Parametereinstellung wieder eingeschaltet werden.

Gerhard Knapienski

HEXAMON mit SESAM

HEXAMON wurde als schnelle Bildschirmausgabe für ein umfassenderes Monitorprogramm entwickelt, das jeder in Basic selbst schreiben kann. Es läßt sich aber unabhängig davon auch für eine schnelle Textausgabe verwenden. Darüber hinaus ist HEXAMON bereits ein einfacher Ein-Befehl-Monitor. Voraussetzung ist lediglich, daß das Programm SESAM geladen ist, dann können mit dem SESAM-Befehl IXCALL insgesamt 5 Funktionen aufgerufen werden. HEXAMON ist für alle Fälle ebenfalls in 256-Byte-Schritten verschiebbar. Dazu muß nur nach dem Laden an eine Adresse &xx00 der Befehl CALL hmon+&290,hmon gegeben werden, wobei hmon die Ladeadresse ist.

Charakterausgabe

```
IXCALL,hmon+&235,&FF,zeichen,anzahl
```

Zeichen ist der ASCII-Wert eines beliebigen Zeichens, anzahl ist die gewünschte Anzahl bis 255 (0 entspricht 256). Bei der Befehlsfolge IXCALL, hmon + &200 ,&FF, 0, 256 * anzahl + min, adr ist eine RAM-Adresse, von der ab die genannte Anzahl Zeichen ausgegeben wird, min sollte 0 sein oder 32(&20) (siehe Text).

Hexadezimalausgabe

```
IXCALL,hmon+&210,&FF,0,byte
```

Der hexadezimale Wert von byte wird ausgegeben.

```
IXCALL,hmon+&220,&FF,anzahl,0,adr
```

Ab adr wird die angegebene Anzahl Bytes hexadezimal mit je einem Leerzeichen ausgegeben.

Pageausgabe

```
IXCALL,hmon+&240,&FF,anzahl,refadr,adr
```

Das ist das Hauptstück von HEXAMON. Mit diesem Befehl wird ein ganzer RAM-Block auf den Bildschirm gebracht (Achtung, nur für MODE 2 geeignet). Dabei ist adr wie gehabt die gewünschte RAM-Adresse und anzahl ist die Anzahl der Zeilen. In einer Zeile werden jeweils 16 Byte hexadezimal und daneben als Charakter dargestellt. Links wird zusätzlich die tatsächliche RAMadresse und rechts eine frei wählbare Referenzadresse (refadr) ausgegeben. Eine Standardpage (16*16 Byte) kommt damit in ca. 2 Sekunden auf den Bildschirm. Die Ausgabe erfolgt nur in WINDOW 0, mit POKE &B20C, windownummer läßt sich auch jedes andere Fenster erreichen.

Sesam öffne dich!

ROM-Pageausgabe

Im Page-Befehl ist der Wert &FF angegeben. Von der Konzeption der SESAM-Routine her bedeutet das zunächst, daß das auszuführende Programm im RAM liegt. Mit einem anderen Wert (0 für Basic-ROM, 7 für Floppy-ROM, &FE für Betriebssystem-ROM) wird aber eine ROM-Selektion vorgenommen, die auch bei der HEXAMON-Ausführung bestehen bleibt. Mit anderen Worten: Mit passendem adr-Wert und ROM-Nummer wird auch ein ROM-Inhalt direkt auf den Bildschirm gebracht.

Druckerausgabe

HEXAMON benutzt den Firmwareeinsprung TXT WR CHAR (&BB5D). Es ist relativ einfach, diesen Vektor auf die Druckerausgabe umzubiegen. Leider sind jedoch weitere Vorkehrungen notwendig. Zum einen dürfen die Steuerzeichen 0 bis 31, die mit HEXAMON als Sonderzeichen auf den Bildschirm gebracht werden, auf keinen Fall an den Drucker gegeben werden. Zum anderen gibt es Probleme mit den Grafiksonderzeichen. Als Beispiel: Das Zeichen 127 wird vom NLQ 401 nicht beachtet und würde dadurch das gesamte Druckbild durcheinanderbringen. Hier hilft nur ein Anpassungsprogramm wie LIST#8.

Hexamon

```
10 MEMORY &BFFF: p=&9D00
20 FOR k=0 TO 15: c=ASC(HEX$(k)): FOR l=
0 TO 15: d=ASC(HEX$(l))
30 q=p+32*k+2*l: POKE q,c: POKE q+1,d
40 NEXT: NEXT
50 FOR m=&9F00 TO &9FBF: READ a$: POKE m
,VAL("&"a$): NEXT
60 SAVE "HEXAMON.BIN",B,&9D00,&2C0
100 DATA 1A,13,C5,D5,B9,30,1,79,CD,5D,BB
,D1,C1,10,F1,C9
110 DATA 21,0,9D,9,9,7E,23,E5,CD,5D,BB,E
1,7E,C3,5D,BB
120 DATA EB,F5,6,0,4E,23,E5,CD,10,9F,3E,
20,CD,5D,BB,E1,F1,3D,20,ED,C9
130 DATA 41,F5,C5,CD,5D,BB,C1,F1,10,F7,C
9
140 DATA F5,C5,D5,6,0,4A,CD,10,9F,D1,D5,
6,0,4B,CD,10,9F,6,3,3E,20,CD,36,9F,D1,D5
,3E,10,CD,20,9F,6,2,3E,20,CD,36,9F
150 DATA D1,1,0,10,CD,0,9F,D5,6,3,3E,20,
CD,36,9F,E1,E3,E5,6,0,4C,CD,10,9F,21,10,
0,C1,9,E5,6,0,CD,10,9F,C1,D1,F1,3D,20,B1
,C9
160 DATA 3D,C0,DD,7E,1,67,24,24,2E,12,77
,2E,29,74,2E,48,74,2E,50,74,2E,57,74,2E,
5E,74,2E,65,74,2E,6C,74,2E,74,74,2E,7D,7
4,2E,88,74,C9,0,0,0,0,0,0
```

Die Unterdrückung der Steuerzeichen geschieht über HEXAMON. Aus der Befehlsliste kennen wir bereits den Wert min. Dabei handelt es sich um die niedrigste Zeichennummer, die zugelassen wird. Zweckmäßig sind dafür nur 0 (Bildschirmdarstellung, alle Zeichen zugelassen) und 32 (Druckerausgabe, Steuerzeichen unterdrückt). Für den Pageausgabebefehl muß min indirekt gesetzt werden: POKE hmon+268,min.

Neben einer vom Drucker abhängigen Anpassung wie LIST#8 sind dann trotz der langen Vorrede nur wenige POKEs erforderlich:

```
wrcharl=PEEK(&BB5E):wrcharh=PEEK(&BB5F)
```

```
Drucker: POKE hmon + &268,32: POKE &BB5D,
&C3: POKE &BB5E, &2B: POKE &BB5F,&BD
```

```
Bildschirm: POKE hmon + &268,0: POKE &BB5D, &CF:
POKE &BB5E, wrcharl:POKE &BB5F, wrcharh
```

Gerhard Knapienski

RAM/ROM Überblick mit Sesam

Mit dem SESAM-Befehl XPEEK kann man ein 16K-ROM direkt in den Bildschirmspeicher übertragen und damit sichtbar machen:

```
IXPEEK,0,&FE,&C000,&4000 (Betriebssystem-ROM)
```

```
IXPEEK,&C000,0,&C000,&4000 (Basic-ROM)
```

Der Aufbau des Bildschirmspeichers verhindert allerdings, daß wir daraus unmittelbar Informationen ablesen können. Um die ROM-Bytes in ihrer richtigen Reihenfolge darzustellen, brauchen wir das kleine Programm LOOK 1. Beim Basic-ROM sehen wir zwar noch keinen Unterschied, aber wer die Schneider-Floppy besitzt, kann im XPEEK-Befehl die 0 durch 7 ersetzen und es wird ein Streifen erkennbar, der uns zeigt, daß das Controller-ROM einen ungenutzten Bereich enthält. Ebenso kann man einen Überblick bekommen, welche RAM-Bereiche belegt sind, wenn i in Zeile 10 eine RAM-Adresse erhält.

Das Programm LOOK 2 ist erheblich langsamer. Die Bytes werden in diesem Fall nicht nebeneinander, sondern untereinander geschrieben. Das hat den Vorteil, daß auch

SYMBOL-Tabellen erkennbar werden. Mit den Beispielen wird der im Betriebssystem-ROM enthaltene Zeichensatz des CPC dargestellt. Der Startwert i=384 wurde gewählt, weil 384 Bytes des Bildschirmspeichers unsichtbar bleiben. In diesem Fall überspringen wir den ROM-Anfang, um das Ende vollständig zu sehen.

Vor jedem Programmablauf sollte ein MODE-Kommando gegeben werden, um den Bildschirmoffset zurückzusetzen.

Sesam-Nachlese

Zunächst gilt es, einen Fehler zu berichtigen. Im Hyperload-Programm können doch nicht ganz 4000 Baud erreicht werden. Der angegebene Wert bewirkt ca. 2700 Baud. Bis 3600 Baud sind möglich mit IXCALL,&BC68,&FF,&A,0,0,&5D. Hilfreich mag es sein, den Befehl name\$ = UPPER\$(name\$) einzufügen. Da die Schreibweise beim Schreiben und Lesen genau übereinstimmen muß, ist so die Gefahr von Fehlern geringer.

Dann war da noch die Rede von einer Anpassung für den CPC 664. Hier ist sie: Die letzten drei Werte in der DATA-Zeile 150 müssen geändert werden in CD,D6,B9.

Die Anwendungsbeispiele sind in der Regel nicht auf den 664 übertragbar. Nicht nur ROM-Routinen haben andere Adressen, sondern auch RAM-Tabellen haben eine veränderte Lage. Was gleich blieb, ist nur die Firmware-Einsprungtabelle. In der neuen SESAM-Anwendung HEXAMON wird z.B. ein POKE für WINDOW-Änderungen angegeben. Das ist ein typischer Wert, der beim 664 abweicht. Im übrigen läuft das Programm auf dem 664.

HEXAMON ist nebenbei auch ein Beispiel für eine weitere SESAM-Einsatz-Möglichkeit. Wer Programme in Maschinensprache selbst schreibt, hat immer einen lästigen Pflichtteil: die Übernahme der Parameter in die Z-80-Register. Diese Übergabe kann jedoch auch mit SESAM geschehen, so daß das Programm sich auf das Wesentliche beschränken kann. Bei HEXAMON können auf diese Weise auch alle Unterprogramme (wenn auch mit eigenwilliger Reihenfolge der Werte) genutzt werden. Weitere Tips und Informationen stehen in den Beschreibungen zu den jeweiligen SESAM-Programmen.

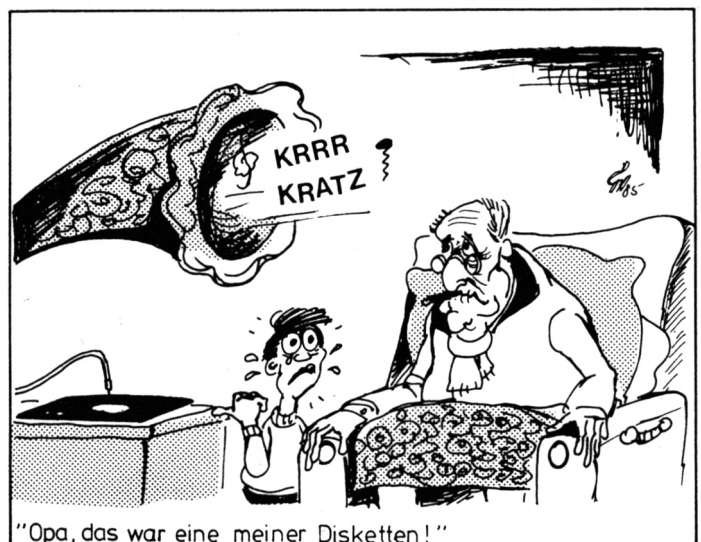
Gerhard Knapienski

Look 1

```
10 i=&C000:j=&C000
20 FOR z=0 TO 199
30 k=z MOD 8
40 IF z>0 AND k=0 THEN j=j+80
50 !XPEEK,i,0,j+k*&800,80
60 i=i+80
70 NEXT
```

Look 2

```
10 i=384:j=&C000
20 FOR z=0 TO 15999
30 k=z MOD 8:l=(z\8) MOD 25
40 IF z>0 AND k=0 AND l=0 THEN j=j+1
50 !XPEEK,i,&FE,j+k*&800+1*80,1
60 i=i+1
70 NEXT
```



▶ HOTLINE 0211-6801403

CASSETTEN

A VIEW TO A KILL	39.00	GRAND PRIX RALLY II (NEU!)	29.90	3D STAR STRIKE	35.00
AIR TRAFFIC CONTROL	36.00	GREMLINS	39.00	STAR AVENGER	29.90
AMERICAN FOOTBALL	45.00	HARD HAT MACK	42.00	STARION	35.00
ANIMATED STRIP POKER	36.00	HARRIER ATTACK	35.00	THE HOBBIT	49.00
AIRWOLF (NEU!)	29.90	HACKER (NEU!)	39.90	THE WAY OF EXPL. FIST	36.00
ARCHON**	45.00	HIGHWAY ENCOUNTER	39.00	TRIPODS	42.00
ARTIST AND SPRITE GENERATOR	39.00	HOME ACCOUNTS	59.00	TURBO TAPE	29.90
ARTWORK	35.00	INTERDICTOR PILOT	59.00	TALES OF THE AR. NIGHTS	29.90
AZIMUTH	35.00	JACK AND THE BEANSTALK	35.00	3D CHESS (NEU!)	39.00
BASIC COMPILER**	55.00	JAMMIN	32.00	ALIEN 8 (NEU!)	39.00
BATTLE FOR MIDWAY	42.00	JETBOOT JACK	35.00	WIZARDS LAIR (NEU!)	45.00
BATTLE OF BRITAIN	39.00	JET SET WILLY	32.00	WORLD CUP	36.00
3D BOXING	39.00	JEWELS OF BABYLON	29.90	WORLD SERIES BASEBALL**	29.90
BATTLE BEYOND THE STARS	39.00	JUMP JET	38.00	SUPERPIPELINE II	32.00
BLAGGER	36.00	JUGGERNAUT	34.00	SUPERSTAR CHALLENGE	35.00
BOUNTY BOB STRIKES BACK**	39.00	KNIGHT LORE	39.00	EVERY ONE'S A WALLY (NEU!)	39.00
BRUCE LEE**	39.00	KONG STRIKES BACK	32.00	MASTER OF THE LAMPS (NEU!)	39.00
C-COMPILER (HISOFT)	98.00	KUNG FU IN LAS VEGAS	29.90	DEVAPAC (HISOFT ASSEMBLER)	89.00
CENTRE COURT	36.00	MACADAM BUMPER	39.00	TASCOPY	32.00
COMBAT LYNX	39.00	(PINBALL CONST.) (NEU!)	39.00	TASPRINT	32.00
CONFUZION	35.00	MANIC MINER	32.00	TASWORD	46.00
COVENANT	32.00	MASTERCHESS	35.00	LORDS OF MIDNIGHT (NEU!)	42.00
DAMBUSTERS**	39.00	MILLIONAIRE	35.00	SPY HUNTER	39.00
DARK STAR	39.00	3D MONSTER CHASE	29.90	ZAXXON	39.00
DECATHLON	32.00	MOON BUGGY	29.90	IMPOSSIBLE MISSION	49.00
DEFEND OR DIE	35.00	MUSIC COMPOSER	39.00	TAPPER	39.00
DUN DARACH	35.00	ONE ON ONE	39.00	WRITER (TEXTVERARBEITUNG)	89.00
DEVILS CROWN (NEU!)	39.00	ON THE RUN	39.00	THE NEVER ENDING STORY (NEU!)	39.00
DRAGONTORC OF AVALON	39.00	PAINT BOX	45.00	FIGHTING WARRIOR (NEU!)	39.00
DOPPELGÄNGER	39.00	PINBALL WIZARD	39.00	MATCHDAY	36.00
CODENAMEMAT II	39.00	POLE POSITION**	39.00		
FIGHTER PILOT	36.00	PROJECT VAL (DEUTSCHES ADV.)	39.00	DISK 464/664 : 5 1/4"	
FIRE ANT	32.00	PROTEUS (DT. HACKERADV.)	39.00	HOUSE OF USHER	39.00
FORMULA ONE	35.00	PYJAMARAMA	36.00	TASPRINT	46.00
FLIGHT PATH 737	29.90	QUILL (ADVENT. CONSTR. SET)	59.00	TASCOPY	46.00
FOOTBALLMANAGER	35.00	REALM OF IMPOSSIBILITY	39.00	TASWORD	66.00
FONT 464	36.00	RED COATS	35.00		
FOREST AT WORLDS END	29.90	RING OF DARKNESS	42.00	BASIC COMPILER	69.00
FORTH	45.00	ROCKY HORROR SHOW	36.00	ARIWOLF	49.00
FRANK BRUNO'S BOXING	35.00	ROBIN OF SHERWOOD	39.00	FIGHTER PILOT	46.00
FRANK N STEIN	36.00	SIR LANCELOT	32.00	JUMP JET	46.00
FRUITY FRANK	32.00	SLAPSHOT	32.00	FRANK BRUNO'S BOXING	49.00
GEMS OF STRATUS	29.90	SPY VS SPY	39.00	RED ARROWS	49.00
GHOSTBUSTERS	42.00	SNOOKER	32.00	SLAPSHOT	49.00
GRAND PRIX DRIVER	35.00	SOFTWARE STAR	35.00	SUPERPIPELINE II	45.00
3D GRAND PRIX (NEU!)	39.00	SORCERY	39.00	HOUSE OF USHER	39.00
		SPECIAL OPERATIONS	35.00	SNOOKER	45.00
		STAR COMMANDO	39.00	WIZARDS LAIR	46.00

** Programme bei Drucklegung noch nicht lieferbar.

Joysoft

5000 Köln 41 · Berrenratherstraße 159 · Telefon erfragen * 4000 Düsseldorf · Humboldtstraße 84 · Telefon 02 11 / 6801403

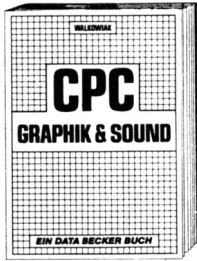


Computer Kontakt

CK Computer Kontakt

Weitere Berichte, Programme, Tips und Tricks für Ihren SCHNEIDER CPC finden Sie regelmäßig auf den Sonderseiten in Computer Kontakt.

**Walkowiak
CPC 464 Grafik & Sound**



220 Seiten
In diesem erstklassigen Buch wird gezeigt, wie man die außergewöhnlichen Grafik- und Soundmöglichkeiten des CPC 464 nutzt. Natürlich mit vielen interessanten Beispielen und nützlichen Hilfsprogrammen. Aus dem Inhalt: Grundlagen der Grafikprogrammierung, Sprites, Shapes und Strings, mehrfarbige Darstellungen, Koordinatentransformation, Verschiebungen, Drehungen, Rotation, 3-D-Funktionsplotter, CAD, Synthesizer, Miniorgel, Hüllkurven und vieles mehr.

Bestellnummer CPC 100 DM 39,-

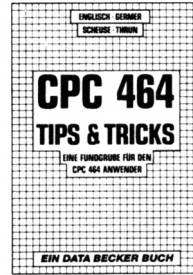
**Hans Lorenz Schneider
Arbeiten mit dem Schneider CPC 464**



288 Seiten
Das Buch befaßt sich mit der BASIC-Programmierung des Schneider CPC. Dabei werden anhand ausführlicher Beispielprogramme viele wichtige Programmiertechniken erläutert. Es werden ernsthafte Themen wie Sortieralgorithmen und Dateiverwaltung behandelt. Auch auf die Programmierung von Spielen wird eingegangen, wobei die Grafik- und Musikprogrammierung nicht zu kurz kommt. Das Buch wendet sich an Besitzer des Schneider CPC 464, die eigene BASIC-Programme erstellen und ihren Computer effektiver für Beruf und Hobby einsetzen möchten.

Bestellnummer CPC 103 DM 38,-

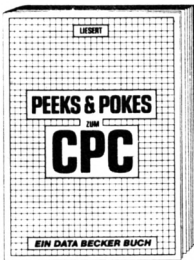
**Englisch/Germer/Scheuse/Thrun
CPC 464 Tips & Tricks
Eine Fundgrube für den CPC-Anwender**



263 Seiten,
Rund um den CPC 464 viele Anregungen und wichtige Hilfen. Von Hardwareaufbau, Betriebssystem, BASIC-Tokens, Zeichnen mit dem Joystick, Anwendungen der Windowtechnologie und sehr vielen interessanten Programmen wie einer umfangreichen Dateiverwaltung, Soundeditor, komfortablem Zeichengenerator bis zu kompletten Listings spannender Spiele bietet das Buch eine Fülle von Möglichkeiten.

Bestellnummer CPC 106 DM 49,-

**Liesert/Schieb
Peeks & Pokes zum CPC 464**



220 Seiten
Wer die wichtigen Peeks und Pokes zum CPC 464 kennen und anwenden will, der findet hier umfassende Information. Sie reicht vom Adreßbereich des Prozessors über Betriebssystem und Interpreter bis hin zur Einführung in die Maschinensprache. Dazu präzise Programmierhilfen, sinnvolle Routinen sowie reichlich Material zu den Themen: Grafikfunktionen, Massenspeicherung und Peripherie, Tricks und Formeln in BASIC, RAM-Pages.

Bestellnummer CPC 101 DM 29,-

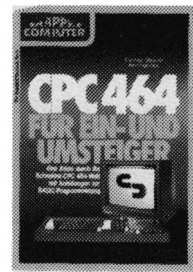
**Siegmar Wittig
BASIC-Brevier für den Schneider CPC 464**



224 Seiten
Dieses Buch zeigt, wie man auf dem Schneider CPC in Basic programmiert. Auch ohne Vorkenntnisse kann jeder nach kurzer Zeit seine eigenen Programme schreiben. Zahlreiche Aufgaben und Programmbeispiele tragen dazu bei, das Wissen zu festigen. Hier findet man fast alle Probleme wieder, die sich einem "Einsteiger" mit dem CPC 464 stellen. Dieses Buch würde sogar das Handbuch des Herstellers voll ersetzen.

Bestellnummer CPC 104 DM 29,80

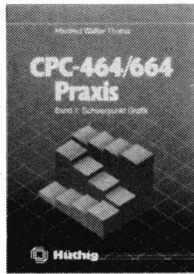
**C. Straush/H. Pick
CPC 464 für Ein- und Umsteiger**



260 Seiten
Über die hervorragende Qualität des CPC 464 sind sich alle Experten einig; die mitgelieferte Dokumentation läßt jedoch eine Menge Fragen offen. Der Computereuling, ebenso aber auch der "alte Hase", der bereits ein anderes System kennt, benötigt hier Hilfe, die ihm diese Buch gibt. Eine BASIC-Einführung (an den Beispielen Dateiverwaltung und Textverarbeitung!) ist selbstverständlich. Darüber hinaus konzentrieren sich die Autoren jedoch auf die besonderen Eigenschaften des 464ers, seine mächtigen Befehle, die andere BASICs ganz schön alt aussehen lassen und seine begeisternden Grafik- und Musikmöglichkeiten.

Bestellnummer CPC 109 DM 46,-

**Manfred Walter Thoma
CPC 464/664 Praxis Band 1:
Schwerpunkt Grafik**



192 Seiten
Der erste Band des zweibändigen Werkes beschäftigt sich schwerpunktmäßig mit den Grafikmöglichkeiten des Schneider CPC-464/664. Es ist besonders darauf geachtet worden, daß alle behandelten Probleme und Programmier-techniken einen konkreten Bezug haben. Neben unzähligen Beispielen findet der Benutzer viele Programme und Unterprogramme, die seine Arbeit mit dem Schneider CPC-464/664 erheblich erleichtern. Gezeigt werden Programme von der Multicolorgrafik bis hin zum kompletten CAD-System, vom Tortendiagramm bis zum 3D-Funktionsplotter, von der 3-dimensionalen Wertdarstellung bis zur Analoguhr.

Bestellnummer CPC 102 DM 34,-

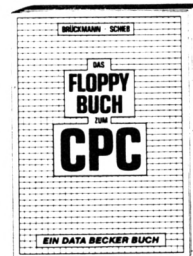
**Andreas Werminghoff
DuMont's Handbuch zum Schneider CPC**



234 Seiten
Dieses Buch zeigt auf verständliche Weise, was der Benutzer mit seinem Computer anfangen kann. Anhand von Programmbeispielen erklärt der Autor die Neuigkeiten des umfangreichen Schneiderbasics, die besonders im Bereich der Zeitgeber-Steuerung und der Tonabgabe liegen. Sogar die Möglichkeit, mehrere Arbeiten gleichzeitig zu erledigen (Multitasking), werden in diesem Buch ausführlich dargestellt. Es ist mehr als nur eine einfache Einführung.

Bestellnummer CPC 105 DM 24,80

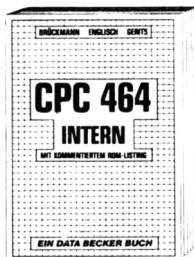
**Brückmann/Schieb
Das Floppy-Buch zum CPC 464**



250 Seiten,
Alles über Diskettenprogrammierung vom Einsteiger bis zum Profi. Natürlich mit ausführlichem ROM-Listing (Betriebssystem), einer äußerst komfortablen Dateiverwaltung, einem hilfreichen Disk-Monitor und einem ausgesprochen nützlichen Disk-Manager. Dazu eine Fundgrube verschiedener Programme und Hilfsroutinen, die das Buch für jeden Floppy-Anwender zur Pflichtlektüre machen.

Bestellnummer CPC 108 DM 49,-

**Brückmann/Englisch/Gerits
CPC 464 Intern
mit kommentiertem ROM-Listing**



548 Seiten
Wirklich alle Geheimnisse des CPC 464 lüftet dieses Standardwerk, das für den fortgeschrittenen BASIC-Programmierer unentbehrlich, für den Assembler-Programmierer ein absolutes Muß ist. Neben dem ausführlich dokumentierten und kommentierten BASIC-ROM-Listing enthält es umfangreiche Kapitel zu Speicheraufteilung, Prozessor, Besonderheiten des Z 80, Gate Array, Video-Controller und Video-Ram, Soundchip, Schnittstellen, Betriebssystem, Routinennutzung, Character-Generator, BASIC-Interpreter und mehr.

Bestellnummer CPC 107 DM 69,-

BUCH-BESTELLKARTE

Bitte liefern Sie mir folgende Bücher:

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

Name des Bestellers _____

Anschrift _____

PLZ/Ort _____

Telefon _____

Ich wünsche folgende Bezahlung:

Nachnahme (+ 5,70 DM Porto + Versandkosten)

Vorkasse (keine Versandkosten)

Bei Vorkasse bitte Scheck belegen oder auf Postscheckkonto Karlsruhe 43423-756 überweisen.

Datum/Unterschrift _____

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

Datenverwaltung für den Schneider

464

664

6128

Mit diesem Programm kann man eine komplette, komfortable Datenverwaltung durchführen. Eine Adressendatei ist ebenso möglich, wie die Verwaltung von Büchern, Kassetten oder Schallplatten. Für jeden Zweck kann hier der Benutzer seine eigene Maske definieren und abspeichern. In einer Dateiverwaltung trägt man z. B. Name, Vorname, Straße, Ort und Telefonnummer ein. Diese Kriterien nennt man Items, die zusammen die Maske ergeben. Da man all diese Items umdefinieren kann, ist so für jeden Zweck eine spezielle Maske möglich. Diese Maske kann man abspeichern und später wieder laden, so daß mehrere Dateierarten (z. B. eine Adressen- und eine Schallplattendatei) zusammen im Programm Platz finden.

Jede Maske darf bis zu 9 Items mit bis zu 25 Buchstaben enthalten. Eine Maske zur Adressenverwaltung befindet sich nach dem Laden des Programmes sofort im Speicher. Sie beinhaltet folgende Items: Name, Vorname, Straße und Nummer, PLZ und Ort, Telefonnummer, Anrede, Geburtstag, Bemerkungen 1 und Bemerkungen 2. Damit läßt sich schon eine recht gute Adressenverwaltung durchführen.

Selbstverständlich kann man mit dem Programm auch in allen denkbaren Formen nach Daten suchen, diese ändern, löschen, einfach nur in ihnen blättern, sie auf dem Bildschirm oder Drucker ausgeben, sie sortieren und vieles mehr.

Im Rechner können sich gleichzeitig bis zu 75 vollständige Datensätze mit bis zu 9 Items zu je 40 Buchstaben befinden. Das mag zwar auf den ersten Blick etwas wenig erscheinen, aber man muß davon ausgehen, daß im ungünstigsten Fall $75 \times 9 \times 40 = 27000$ Buchstaben, also 27000 Bytes anfallen. Dazu kommen dann noch die Stringdeskriptoren ($3 \times 75 \times 9$ Bytes) und einige Bytes zur Verwaltung der Arrays, die anzeigen, ob ein Datensatz frei ist. So kommt man sehr schnell über 30 K. Ich habe die Kapazität des Programmes etwas zurückgeschraubt, weil man ja damit rechnen muß, daß jemand die 40 Buchstaben überall voll ausnutzt. 75 Adressen sind aber doch schon ganz schön viel. Da muß man schon lange überlegen, um die alle vollzubekommen. Außerdem kann man ja noch ein zweites oder drittes Array abspeichern.

Darüber sollten Sie sich nicht wundern!

Wenn das Programm mal 1 oder 2 Minuten gar nichts macht, so wundern Sie sich nicht weiter, dann schlägt die Garbage Collection des CPCs zu. Jeder veränderte String wird im Rechner völlig neu angelegt. Das hat zur Folge, daß sich mit der Zeit viele alte, unbrauchbare Strings im Speicher ansammeln. Irgendwann wenn der Speicherplatz zu klein wird, muß der Rechner alle diese alten Strings rauschmeißen, um sich wieder Platz zu schaffen. Das kann im Höchstfall 1 bis 2 Minuten dauern. Mir ist es bisher bei diesem Programm noch nicht passiert, aber bei der großen Menge an Strings könnte es mal vorkommen und Sie sollten sich dann nicht wundern und denken, das Programm sei ausgestiegen. Übrigens ist der CPC nicht der einzige

Rechner, der eine Garbage Collection durchführt. Auch der Commodore 64 macht das manchmal und bei ihm kann das bis zu 20 Minuten dauern.

Vor jedem Lade- und Speichervorgang führt der Rechner ebenfalls eine Garbage Collection durch, auch wenn diese nicht nötig ist. Aber der CPC muß sich ja immerhin 2 K Speicherplatz für seinen Kassetten- bzw. Diskbuffer freischaufeln. Beim Start des Programms befinden sich noch keine großen Stringmengen im Speicher. Sie sollten dann, sofern erforderlich, zuerst die Maske laden, da diese auch nur wenig Strings beansprucht. Anschließend können Sie ohne Wartezeit die Daten laden. Jetzt ist der Speicher aber schon ziemlich voll, so daß beim Schließen der Datei der Rechner schon ein bißchen Zeit braucht, um seine Strings zu reorganisieren. Wollen Sie die Daten nach der Eingabe eines neuen Datensatzes wieder abspeichern, müssen Sie etwa 10-20 Sekunden warten, bis der Rechner mit dem Abspeichern beginnt.

Fehlermeldungen

Taucht irgendwann mal eine Fehlermeldung auf, z. B. ein Fehler beim Lesen von Kassette oder Diskette oder ein Syntax Error, so gelangen Sie durch GOTO 330 wieder ins Hauptmenü, ohne daß die Daten verlorengegangen sind. Jetzt können Sie einen erneuten Ladeversuch starten oder die Daten erst einmal abspeichern und anschließend den Syntax Error berichtigen. ACHTUNG: Wenn Sie das Programm verändern, um z. B. einen Syntax Error auszumerzen, verschieben sich die Strings, die nachher nur noch Kauderwelsch sind. Also: Erst abspeichern und dann berichtigen. Am besten ist es aber, wenn Sie nur ein oder zwei Datensätze eingeben und dann alle Funktionen des Programmes ausprobieren, um nach den Fehlern zu suchen.

Haben Sie einen Datensatz mit 9 Items und definieren Sie beispielsweise danach eine Maske mit nur 5 Items, so werden auch nur noch die 5 Items des Datensatzes angezeigt. Die Daten sind aber nicht verloren. Wenn Sie danach eine Maske mit 9 Items definieren, so werden die Daten wieder sichtbar. Speichern Sie die Datei allerdings mit einer Maske mit 5 Items ab, so werden auch nur die ersten 5 Items der Datensätze abgespeichert.

Wenn Sie eine Maske mit 5 Items haben und dann einen Datensatz mit einer anderen Anzahl an Items laden, so wird die Maske automatisch auf die neue Zahl erweitert bzw. gekürzt. Sie können dann aber wieder eine neue Maske definieren, um die gewünschte Anzahl an Items wiederherzustellen und können dann die Daten im neuen Format abspeichern.

Der Programmaufbau

Das Programm ist vollständig in Basic geschrieben, es wird aber durch geschickte Programmierung über Zeiger enorm schnell. Zum besseren Verständnis hier die verwendeten Variablen.

MASK gibt die Anzahl der Items (1-9) in einer Maske an. In M\$ (9) sind die Begriffe der Maske gespeichert, z. B. Name, Vorname usw. In D\$ (75,9) stehen die eigentlichen Daten. F% (75) gibt an, ob die entsprechende Stelle im Ar-

ray belegt oder frei ist. Ist F% (Nummer) gleich 1, so ist diese Stelle im Array besetzt.

Beim Sortieren oder Löschen müßte man normalerweise immer die ganzen Strings verschieben, was zwei Nachteile hätte. Erstens würde wieder jede Menge Stringmüll anfallen, was nachteilige Folgen hätte, und zweitens würde das viel Zeit kosten. Aus diesem Grunde arbeitet das Programm mit dem Zeigerarray Z% (75). Z% (1) zeigt auf das erste Element der Daten, Z% (2) auf das zweite usw. So müssen beim Sortieren und Löschen nur diese Arrays geändert werden.

NUMB gibt die nächste freie Stelle im Zeigerarray an. Die Daten im eigentlichen Array können wild durcheinander stehen, durch das Zeigerarray bleibt die Ordnung erhalten. Vielleicht möchten einige Anwender die abgespeicherten Daten in anderen Programmen verwenden. Hier die Angabe, in welchem Format diese abgespeichert sind: Zuerst kommen die Variablen MASK und NUMB, anschließend Z% (i), F% (i) und D\$(i, j), wobei i von 1 bis 75 und j von 1 bis MASK läuft.

Tippen Sie das gesamte Programm sorgfältig ab und lassen Sie nichts aus. Lediglich die Kommentare können wegfallen, da diese Zeilen nicht angesprungen werden. Sie sollten Sie aber dennoch übernehmen, weil das Programm

```

Datenverwaltung von H.Zallmann (C) 1985          Blättern          19:11:48
SPACE - Return
< - Zurueckblättern
> - Vorblättern

1 Name           : Zallmann
2 Vorname        : Andreas
3 Strasse & Nr.   : Eulenweg 5
4 PLZ & Ort       : 4923 Extertal
5 Telefonnummer   : 05262/2256
6 Adresse         : Herr
7 Geburtstag      : 01.06.1967
8 Bemerkungen 1   : Dies ist ein Beispiel fuer einen
9 Bemerkungen 2   : Datensatz mit der vorgegebenen Maske
  
```

Beispiel für einen Datensatz

dann übersichtlicher wird. Speichern Sie das Programm schließlich mit SAVE "DATENVER" ab.

Nach dem Einladen und dem Start des Programmes können Sie zuerst einmal entscheiden, ob eine Uhr mitlaufen soll. Ist das der Fall, so antworten Sie mit "j" für Ja. Danach müssen Sie die Uhr noch stellen.

Nun gelangen Sie ins Hauptmenü, wo Sie eine eingetragte Kopfzeile sehen. Hier ist zuerst der Titel des Programmes angegeben. Daneben befindet sich die Statusanzeige. Hier wird angegeben, was Sie im Augenblick machen. Wenn Sie mal nicht wissen, wo Sie sind, achten Sie auf diese Kopfzeile. Links daneben befindet sich die laufende Uhr, darunter die eigentlichen Menüpunkte des Hauptmenüs. Links davon sehen Sie eine Anzeige, die angibt, wieviel Datensätze noch frei sind.

Wenn Sie einen Menüpunkt anwählen, hinterfragt der Rechner Ihre Entscheidung mit "WIRKLICH (J/N)". Oben rechts sehen Sie in der Kopfzeile, welche Funktion Sie ausgewählt haben. Durch Drücken von "N" für NEIN gelangen Sie wieder ins Hauptmenü, durch Drücken von "J" wird die Funktion weiter ausgeführt.

Eingabe

Hier können Sie einen neuen Datensatz eingeben. Zuerst untersucht der Rechner, ob noch Speicherplatz frei ist.

Sollte das nicht der Fall sein, so gelangen Sie nach einer entsprechenden Meldung wieder zurück ins Hauptmenü. Ansonsten wird die Maske angezeigt, dahinter 40 Zeichen Striche. Auf diesen Strichen schreiben Sie jetzt ihre Eingaben und beschließen jede Eingabe eines Items mit ENTER. Nach der Eingabe der 1-9 Items gelangen Sie ins Hauptmenü.

Suchen

Geben Sie zuerst den zu suchenden String ein. Danach können Sie aus den vier Sucharten auswählen.

1. Der Suchstring soll ganz in den Daten enthalten sein. Es soll nichts mehr danach kommen und nichts davor. Da Sie den vollständigen Namen wissen, können Sie Punkt 1 wählen.

2. Der String soll irgendwo enthalten sein. Diese Funktion können Sie benutzen, um z. B. alle Datensätze zu suchen, in denen das Zeichen "?" vorkommt. Diese Funktion vereinigt die Punkte 1,3 und 4 miteinander.

3. Der String soll am Anfang stehen.

4. Der String soll am Ende stehen.

Anschließend geben Sie ein, von welchem bis zu welchem Item gesucht werden soll. Wird nur nach dem Namen und Vornamen durchsucht, wäre das 1 bis 2. Nun sucht der Rechner und gibt den ersten zutreffenden Datensatz aus und fragt "WEITERSUCHEN (J/N)". Wollen Sie nicht weitersuchen, gelangen Sie ins Hauptmenü, ansonsten sucht der Rechner weiter. Sind keine gesuchten Datensätze mehr vorhanden, meldet der Rechner "Nichts gefunden" und kehrt ins Hauptmenü zurück.

Ändern

Zuerst müssen Sie bei "Suchen" einen String und die Suchart eingeben. Die Datensätze, die diesen Suchkriterien entsprechen, können Sie dann ändern. Hat der Rechner einen Datensatz gefunden, gibt er ihn aus und fragt "ÄNDERN (j/n)". Antworten Sie mit "j", können Sie die Daten ändern. Das läuft genauso wie bei der Eingabe, nur daß die alten Daten schon auf dem Bildschirm stehen und Sie diese überschreiben können. Drücken Sie nur ENTER, wird die alte Datenzeile übernommen. Anschließend oder wenn Sie "n" gedrückt haben, fragt der Rechner "WEITERSUCHEN (j/n)". Sie können also weiter Ändern oder ins Hauptmenü zurückkehren. Sie werden dann aber wieder gefragt, ob Sie den nächsten Datensatz auch wirklich ändern wollen.

Löschen

Das Löschen läuft im Prinzip genauso wie das Ändern, nur fragt der Rechner nicht "ÄNDERN (j/n)", sondern "LÖSCHEN (j/n)". Nach Eingabe von "j" wird der Datensatz gelöscht. Am Anfang steht außerdem noch eine Abfrage, ob Sie alle Daten löschen wollen.

Blättern

Der Rechner gibt das erste Datenblatt aus. Mit "<" können Sie zurückblättern und mit ">" vorwärtsblättern. Gelangen Sie dabei außerhalb des belegten Bereiches, geht es auf der anderen Seite weiter. Mit SPACE gelangen Sie wieder ins Hauptmenü.

Sortieren

Mit dieser Funktion können Sie Ihre Daten alphabetisch sortieren. Dazu müssen Sie nur die Nummer des Items angeben, nach dem sortiert werden soll.

Ausgeben

Hier können Daten gezielt auf dem Drucker oder auf dem Bildschirm ausgegeben werden. Geben Sie dazu den Suchstring und die Suchart ein, um die auszugebenden Datensätze zu spezifizieren. Sollen alle Daten ausgegeben werden, drücken Sie beim Suchstring nur ENTER und wählen Suchart 2.

Tape/Disc Menü

Hier können Sie Daten und Masken laden und speichern. Zuerst geben Sie bitte den Namen der zu ladenden oder speichernden Daten oder Maske ein. Dieser darf nur acht Zeichen lang sein, um zu den Diskettenlaufwerkbesitzern kompatibel zu bleiben. An diesen Namen kommt dann noch ein 3 Zeichen langes Anhängsel: Bei den Masken msk und bei den daten dtv. So können Sie Maske und Daten den gleichen Namen geben. Anschließend wählen Sie eine der vier Funktionen.

Maske erstellen

Mit dieser Funktion können Sie eine neue Maske definieren. Zuerst geben Sie die Anzahl der Items (1-9) und anschließend die Bezeichnungen der Items ein.

Programm beenden

Diese Funktion spricht für sich selbst. Nachdem Sie aus dem Programm ausgestiegen sind, können Sie mit GOTO 330 wieder ins Hauptmenü gelangen.

Noch ein Tip zum Schluß: Geben Sie bei "Maske erstellen, Daten eingeben oder Ändern" nie ein Komma ein, da der Rechner dieses Zeichen als Trennzeichen behandelt und dann beim Laden und Abspeichern durcheinanderkommen würde. Aber keine Angst, wenn Sie ein Komma eingeben, merkt das der CPC und Sie müssen die Zeile noch einmal eingeben.

Andreas Zallmann
Eulenberg 5
4923 Extertal
Telefon 052 62 / 22 56

Datenverwaltung

```

100 '
110 ' Datenverwaltung - (C) 1985 by
120 '
130 '      Andreas Zallmann
140 '
150 MODE 2:INK 0,0:INK 1,26:BORDER 0:DIM
f%(75):DIM z%(75):DIM d$(75,9):DIM m$(9
):mask=9:numb=1
160 PAPER 1:CLS:PAPER 0:WINDOW #6,2,42,2
,2:CLS #6:PRINT #6," Datenverwaltung von
A.Zallmann (C) 1985";
170 WINDOW #6,70,79,2,2:CLS #6:WINDOW #6
,44,68,2,2:CLS #6:WINDOW #6,45,67,2,2
180 WINDOW #7,71,78,2,2:PRINT #7," :
";
190 WINDOW #0,2,79,4,24:CLS
200 n1$="":n2$="":FOR i=1 TO 40:n1$=n1$+
CHR$(95):IF i<26 THEN n2$=n2$+CHR$(95)
210 NEXT i
220 RESTORE 230:FOR i=1 TO 9:READ m$(i):
NEXT
230 DATA Name,Vorname,Strasse & Nr.,PLZ
& Ort,Telefonnummer,Anrede,Geburtstag,Be
merkungen 1,Bemerkungen 2
240 CLS:LOCATE 10,10:PRINT"Wuenschen Sie
eine Uhr (j/n)?"
250 GOSUB 2020:i$=LOWER$(i$):IF i$<>"j"
AND i$<>"n" THEN PRINT CHR$(7);:GOTO 250
260 IF i$="n" THEN 330
270 n$="Uhr stellen":GOSUB 2000
280 CLS
290 LOCATE 10,8:INPUT "Bitte Stunden, Mi
nuten, Sekunden eingeben ";st,mi,sek:st=
INT(st):mi=INT(mi):sek=INT(sek)
300 IF sek<0 OR sek>59 OR mi<0 OR st<0 O
R st>23 OR mi>59 THEN 290
310 EVERY 50 GOSUB 1940
320 'Mainmenue
330 n$="Hauptmenue":GOSUB 2000
340 CLS
350 RESTORE 420
360 FOR i=0 TO 9:READ a$:LOCATE 10,i+4:P
RINT i;" - ";a$:NEXT

```

```

370 LOCATE 11,15:PRINT "Bitte waehlen Si
e !"
380 LOCATE 55,4:PRINT "Frei :";76-numb
390 GOSUB 2020
400 IF i$<"0" OR i$>"9" THEN PRINT CHR$(
7);:GOTO 390
410 i=VAL(i$):IF i=0 THEN 440 ELSE ON i
GOTO 560,720,960,1180,1290,1370,1550,179
0,1890
420 DATA Eingabe,Suchen,Aendern,Loeschen
,Blaettern,Sotieren,Ausgeben,Tape/Disk M
enue,Maske erstellen,Programm beenden
430 'Eingabe
440 n$="Dateneingabe":GOSUB 2000:CLS
450 GOSUB 2040:IF ret=1 THEN 330
460 CLS
470 FOR n=1 TO 75:IF f%(n)=0 THEN 490 EL
SE NEXT n
480 LOCATE 10,10:PRINT "Kein Speicherpla
tz mehr !":GOSUB 2020:GOTO 330
490 GOSUB 2270
500 LOCATE 10,2:PRINT "Bitte geben Sie d
ie Daten ein (max. 40 Buchstaben)."
510 FOR i=1 TO mask
520 LOCATE 34,i+6:PRINT SPACE$(45);:LOCA
TE 34,i+6:PRINT n1$:LOCATE 34,i+6:LINE I
NPUT d$(n,i)
530 IF LEN(d$(n,i))>40 OR INSTR(d$(n,i),
",")<>0 THEN 520
540 NEXT i:f%(n)=1:z%(numb)=n:numb=numb+
1:GOTO 330
550 'Suchen
560 n$="Suchen":GOSUB 2000:CLS
570 GOSUB 2040:IF ret=1 THEN 330
580 GOSUB 2090
590 CLS:beg=1
600 LOCATE 10,8:INPUT "Durchsuchen ab It
em ";one
610 IF one<1 OR one>mask THEN 600
620 LOCATE 10,12:INPUT "Durchsuchen bis
Item ";two
630 IF two<one OR two>mask THEN 620
640 GOSUB 2270
650 GOSUB 2150
660 IF nr>=numb THEN CLS:LOCATE 10,10:PR
INT "Nicht gefunden.":GOSUB 2020:GOTO 33
0

```

```

670 nrrr=nr:nr=z%(nr):GOSUB 2280:nr=nrrr
680 LOCATE 10,20:PRINT "Weitersuchen (j/
n)"
690 GOSUB 2020:i#=LOWER$(i#):IF i#<>"j"
AND i#<>"n" THEN 690
700 IF i#="n" THEN 330 ELSE beg=nr+1:GOT
O 650
710 'Aendern
720 n#="Aendern":GOSUB 2000:CLS
730 GOSUB 2040:IF ret=1 THEN 330
740 GOSUB 2090:CLS:beg=1
750 LOCATE 10,8:INPUT "Durchsuchen ab It
em ";one
760 IF one<1 OR one>mask THEN 750
770 LOCATE 10,12:INPUT "Durchsuchen bis
Item ";two
780 IF two<one OR two>mask THEN 770
790 GOSUB 2270
800 GOSUB 2150
810 IF nr>=numb THEN CLS:LOCATE 10,10:PR
INT "Nicht gefunden.":GOSUB 2020:GOTO 33
0
820 nrrr=nr:nr=z%(nr):GOSUB 2280:nr=nrrr
830 LOCATE 10,20:PRINT "Aendern (j/n)
"
840 GOSUB 2020:i#=LOWER$(i#):IF i#<>"j"
AND i#<>"n" THEN PRINT CHR$(7);:GOTO 840
850 IF i#="j" THEN 890
860 LOCATE 10,20:PRINT "Weitersuchen (j/
n)"
870 GOSUB 2020:i#=LOWER$(i#):IF i#<>"j"
AND i#<>"n" THEN PRINT CHR$(7);:GOTO 870
880 IF i#="n" THEN 330 ELSE beg=nr+1:GOT
O 800
890 LOCATE 5,5:PRINT "ENTER um Zeile zu
uebernehmen"
900 FOR i=1 TO mask

910 LOCATE 34,i+6:LINE INPUT n#:IF n#=""
THEN 940
920 IF LEN(n#)>40 OR INSTR(n#,"")<>>0 TH
EN 910
930 d$(z%(nr),i)=n#
940 NEXT i:GOTO 860
950 'Loeschen
960 n#="Loeschen":GOSUB 2000:CLS
970 GOSUB 2040:IF ret=1 THEN 330
980 CLS:LOCATE 10,8:PRINT "1 - Alle Date
n loeschen":LOCATE 10,12:PRINT "2 - Spez
ielle Daten loeschen"
990 GOSUB 2020:i#=LOWER$(i#):IF i#<>"1"
AND i#<>"2" THEN PRINT CHR$(7);:GOTO 990
ELSE IF i#="2" THEN 1010 ELSE n#="Alle
Daten loeschen":GOSUB 2000:CLS:GOSUB 204
0:IF ret=1 THEN 330
1000 numb=1:FOR i=1 TO 75:f%(i)=0:NEXT:G
OTO 330
1010 GOSUB 2090:CLS:beg=1
1020 LOCATE 10,8:INPUT "Durchsuchen ab I
tem ";one
1030 IF one<1 OR one>mask THEN 1020
1040 LOCATE 10,12:INPUT "Durchsuchen bis
Item ";two
1050 IF two<one OR two>mask THEN 1040
1060 GOSUB 2270
1070 GOSUB 2150
1080 IF nr>=numb THEN CLS:LOCATE 10,10:P
RINT "Nicht gefunden.":GOSUB 2020:GOTO 3
30
1090 nrrr=nr:nr=z%(nr):GOSUB 2280:nr=nrr
r

1100 LOCATE 10,20:PRINT "Loeschen (j/n)
"
1110 GOSUB 2020:i#=LOWER$(i#):IF i#<>"j"
AND i#<>"n" THEN PRINT CHR$(7);:GOTO 11
10
1120 IF i#="j" THEN 1160
1130 LOCATE 10,20:PRINT "Weitersuchen (j
/n)"
1140 GOSUB 2020:i#=LOWER$(i#):IF i#<>"j"
AND i#<>"n" THEN PRINT CHR$(7);:GOTO 11
40
1150 IF i#="n" THEN 330 ELSE beg=nr+1:GO
TO 1070
1160 numb=numb-1:f%(z%(nr))=0:FOR i=nr T
O numb:z%(i)=z%(i+1):NEXT:GOTO 1130
1170 'Blaettern
1180 n#="Blaettern":GOSUB 2000:CLS
1190 IF numb=1 THEN 330
1200 GOSUB 2040:IF ret=1 THEN 330 ELSE C
LS
1210 GOSUB 2270:num=1:LOCATE 4,2:PRINT "
SPACE - Return":LOCATE 4,3:PRINT "< - Zu
rueckblaettern":LOCATE 4,4:PRINT "> - Vo
rblaettern"
1220 nr=z%(num):GOSUB 2280
1230 GOSUB 2020:IF i#<>"," AND i#<>". " A
ND i#<>" " THEN 1230
1240 IF i#="," THEN num=num-1:IF num=0 T
HEN num=numb-1
1250 IF i#="." THEN num=num+1:IF num=num
b THEN num=1
1260 IF i#=" " THEN 330
1270 GOTO 1220
1280 'Sotieren
1290 n#="Sotieren":GOSUB 2000:CLS

1300 GOSUB 2040:IF ret=1 THEN 330
1310 CLS:LOCATE 10,10:PRINT "Nach welche
m Item soll sotiert werden ( 1 -";mask;"
)";:INPUT it:IF it<1 OR it>mask THEN 131
0
1320 FOR a=1 TO numb-1
1330 FOR b=a+1 TO numb-1
1340 IF d$(z%(a),it)>d$(z%(b),it) THEN z
w=z%(a):z%(a)=z%(b):z%(b)=zw
1350 NEXT b:NEXT a:GOTO 330
1360 'Ausgeben
1370 n#="Ausgeben":GOSUB 2000:CLS
1380 GOSUB 2040:IF ret=1 THEN 330
1390 CLS:LOCATE 10,10:PRINT "1 - Bildsch
irm":LOCATE 10,12:PRINT "2 - Drucker":LO
CATE 10,8:PRINT "Ausgabe auf -"
1400 GOSUB 2020:IF i#<>"1" AND i#<>"2" T
HEN 1400
1410 IF i#="1" THEN k=0 ELSE k=8
1420 CLS
1430 GOSUB 2090
1440 beg=1
1450 CLS
1460 LOCATE 10,8:INPUT "Durchsuchen ab I
tem ";one
1470 IF one<1 OR one>mask THEN 1460
1480 LOCATE 10,12:INPUT "Durchsuchen bis
Item ";two
1490 IF two<one OR two>mask THEN 1480
1500 CLS
1510 GOSUB 2150:IF nr>=numb THEN PRINT:P
RINT"FERTIG":GOSUB 2020:GOTO 330
1520 FOR i=1 TO mask:PRINT #k,d$(z%(nr),
i):NEXT:PRINT #k
1530 beg=nr+1:GOTO 1510

```

```

1540 'Tape/Disk Menue
1550 n$="Tape/Disk Menue":GOSUB 2000:CLS
1560 CLS:WINDOW 10,70,10,20:PRINT "Bitte
  waehlen Sie die gewuenschte Option"
1570 PRINT:PRINT
1580 PRINT "1 - Daten speichern":PRINT "
  2 - Daten laden":PRINT "3 - Maske speich
  ern":PRINT "4 - Maske laden":PRINT "5 -
  Zurueck zum Hauptmenue"
1590 GOSUB 2020:IF i$<"1" OR i$>"5" THEN
  1590
1600 WINDOW 2,79,4,24:CLS:IF i$="5" THEN
  330
1610 LOCATE 10,8:PRINT "Bitte geben Sie
  den Namen ein (max. 8 Buchstaben)"
1620 LOCATE 10,12:LINE INPUT m$:IF LEN(m
  $)>8 THEN LOCATE 10,12:PRINT SPACE$(65):
  GOTD 1620
1630 ON VAL(i$) GOTD 1650,1690,1730,1760
1640 ' Daten speichern
1650 n$="Daten speichern":GOSUB 2000:CLS
:LOCATE 10,10:PRINT "Die Daten werden un
  ter dem Namen ";m$;".dtv abgespeichert."
1660 LOCATE 1,20
1670 OPENOUT m$+".dtv":PRINT #9,mask;num
  b:FOR i=1 TO 75:PRINT #9,f%(i);z%(i):FOR
  a=1 TO mask:PRINT #9,d$(i,a):NEXT a,i:C
  LOSEOUT:GOTO 330
1680 ' Daten laden
1690 n$="Daten laden":GOSUB 2000:CLS:LOC
  ATE 10,10:PRINT "Die Daten werden unter
  dem Namen ";m$;".dtv gesucht."
1700 LOCATE 1,20
1710 OPENIN m$+".dtv":INPUT #9,mask,numb
  :FOR i=1 TO 75:INPUT #9,f%(i);z%(i):FOR
  a=1 TO mask:INPUT #9,d$(i,a):NEXT a,i:CL
  OSEIN:GOTO 330
1720 ' Maske speichern
1730 n$="Maske speichern":GOSUB 2000:CLS
:LOCATE 10,10:PRINT "Die Maske wird unte
  r dem Namen ";m$;".msk abgespeichert."
1740 LOCATE 1,20:OPENOUT m$+".msk":PRINT
  #9,mask:FOR i=1 TO mask:PRINT #9,m$(i):
  NEXT:CLOSEOUT:GOTO 330
1750 ' Maske laden
1760 n$="Maske laden":GOSUB 2000:CLS:LOC
  ATE 10,10:PRINT "Die Maske wird unter de
  m Namen ";m$;".msk gesucht."
1770 LOCATE 1,20:OPENIN m$+".msk":INPUT
  #9,mask:FOR i=1 TO mask:INPUT #9,m$(i):N
  EXT:CLOSEIN:GOTO 330
1780 'Maske erstellen
1790 n$="Maske erstellen":GOSUB 2000:CLS
1800 GOSUB 2040:IF ret=1 THEN 330
1810 CLS
1820 LOCATE 10,10:INPUT "Wieviele Sparte
  n (1-9)";mask:IF mask<1 OR mask>9 THEN 1
  820
1830 CLS:LOCATE 10,3:PRINT "Bitte geben
  Sie nun die Spartenbezeichnungen ein (ma
  x. 25 Buchstaben)"
1840 FOR i=1 TO mask
1850 LOCATE 10,i+5:PRINT SPACE$(65):LOCA
  TE 10,i+5:PRINT i;n2$:LOCATE 13,i+5
1860 LINE INPUT m$(i):IF LEN (m$(i))>25
  OR INSTR(m$(i),"")<>0 THEN 1850
1870 NEXT:GOTO 330
1880 'Programm beenden
1890 n$="Programm beenden":GOSUB 2000
1900 CLS:LOCATE 10,10:PRINT "Soll das Pr
  ogramm wirklich beendet werden (j/n) ?"

1910 GOSUB 2020:i$=LOWER$(i$):IF i$<>"j"
  AND i$<>"n" THEN PRINT CHR$(7);:GOTO 19
  10
1920 IF i$="j" THEN LOCATE 10,13:PRINT "
  OK. Warmstart : GOTO 330":LOCATE 1,20:EN
  D ELSE 330
1930 'Uhr
1940 sek=sek+1:IF sek=60 THEN sek=0:mi=m
  i+1:IF mi=60 THEN mi=0:st=st+1:IF st=24
  THEN st=0
1950 LOCATE #7,1,1:PRINT #7,USING "##";s
  t;
1960 LOCATE #7,4,1:PRINT #7,USING "##";m
  i;
1970 LOCATE #7,7,1:PRINT #7,USING "##";s
  ek;
1980 RETURN
1990 'Modusausgabe
2000 CLS #6:l=LEN(n$):LOCATE #6,12-INT(l
  /2),1:PRINT #6,n$:RETURN
2010 'Tastenabfrage
2020 i$="":WHILE i$="":i$=INKEY$:WEND:RE
  TURN
2030 'wirklich
2040 CLS:LOCATE 10,10:PRINT "Wirklich (j
  /n)"
2050 GOSUB 2020:i$=LOWER$(i$):IF i$<>"j"
  AND i$<>"n" THEN PRINT CHR$(7);:GOTO 20
  50
2060 IF i$="n" THEN ret=1 ELSE ret=0
2070 RETURN
2080 'ST$-Stringeingabe,Suchart SUCH
2090 CLS:LOCATE 5,6:INPUT "Geben Sie den
  zu suchenden String ein ";st$
2100 WINDOW 10,50,13,24:PRINT "Welche Ar
  t des Suchens wuenschen Sie ?"
2110 PRINT:PRINT:PRINT"1 - String soll g
  anz enthalten sein":PRINT"2 - String sol
  l irgendwo enthalten sein":PRINT"3 - Str
  ing soll am Anfang stehen":PRINT"4 - Str
  ing soll am Ende stehen"
2120 GOSUB 2020:IF i$<"1" OR i$>"4" THEN
  PRINT CHR$(7);:GOTO 2120
2130 WINDOW #0,2,79,4,24:such=VAL(i$):RE
  TURN
2140 'Suchen von String ST$,von beg nach
  Art such:(von one bis two)liefert nr
2150 FOR nr=beg TO numb
2160 nrr=z%(nr)
2170 IF f%(nrr)=0 THEN 2250
2180 FOR i=one TO two
2190 ON such GOTD 2200,2210,2220,2230
2200 IF d$(nrr,i)=st$ THEN RETURN ELSE
  2240
2210 IF INSTR(d$(nrr,i),st$)<>0 THEN RE
  TURN ELSE 2240
2220 IF LEFT$(d$(nrr,i),LEN(st$))=st$ T
  HEN RETURN ELSE 2240
2230 IF RIGHT$(d$(nrr,i),LEN(st$))=st$
  THEN RETURN ELSE 2240
2240 NEXT i
2250 NEXT nr:RETURN
2260 'Bildschirmausgabe
2270 CLS:FOR i=1 TO mask:LOCATE 3,i+6:PR
  INT i;n2$:LOCATE 6,i+6:PRINT m$(i):LOCAT
  E 32,i+6:PRINT "":NEXT:RETURN
2280 FOR i=1 TO mask
2290 LOCATE 34,i+6:PRINT SPACE$(45);:LOC
  ATE 34,i+6:PRINT n1$:LOCATE 34,i+6:PRINT
  d$(nr,i)
2300 NEXT i:RETURN

```

Machen Sie Ihre Disketten fit

Mit dem Diskdoktor für alle CPC Computer

Mit dem Programm DISKDOKTOR können Sie gelöschte Files wieder aktivieren, Files verstecken und wieder auffinden, Files mit einem Schreibschutz versehen und diesen wieder entfernen. Sie bekommen wichtige Informationen aus der Directory, haben Einblick in jeden Sektor, können Files kopieren und jedes Byte auf der Diskette ändern.

Zuerst aber etwas Theorie. Der CPC unterscheidet vier verschiedene Diskettenformate: Das AMSDOS CP/M Format, das Vendor Format, das Daten Format und das IBM CP/M Format. Alle diese Diskettenformate haben folgendes gemeinsam:

1. Die Disketten haben 40 Spuren, durchnummeriert von 0 bis 39, wobei 0 außen und 39 innen ist.
2. Diese Spuren sind in 9, bzw. 8 Sektoren unterteilt, die jeweils 512 Byte groß sind.
3. In der Directory (Inhaltsverzeichnis der Diskette) können höchstens 64 Einträge stehen.

Das AMSDOS CP/M Format

Dieses Format werden Benutzer des Diskettenlaufwerks wohl am häufigsten verwenden. Jede Spur hat 9 Sektoren zu je 512 Byte. Die Spuren 0 und 1 sind für CP/M reserviert und zwar in folgender Belegung:

Spur 0, Sektor 1: BOOT SEKTOR
 Spur 0, Sektor 2: CONFIGURATIONSEKTOR
 Spur 0, Sektor 8 und 9 sowie
 Spur 1, Sektor 1 bis 9: CCP und BDOS

Es verbleiben also 38 Spuren zu 9 Sektoren mit je 512 Byte, wovon 4 Sektoren für die Directory abgehen. Damit ergibt sich eine Speicherkapazität von exakt 169 K.

Die 4 Diskettenformate

Das Vendor Format

Das Vendor Format ist prinzipiell das Gleiche wie das AMSDOS CP/M Format, nur werden die Spuren 0 und 1 nicht benutzt, d.h., das CP/M wird nicht kopiert. Es wird dadurch aber kein Speicherplatz gewonnen, da die Spuren einfach nicht benutzt werden. Aus rechtlichen Gründen darf das CP/M selbst nicht kopiert werden. Deshalb werden CP/M Programme im Vendor Format verkauft und der Benutzer muß sich mit dem CP/M Hilfsfile SYSGEN.COM selbst die Systemspuren einrichten.

Das Daten Format

CP/M wird auch hier nicht überspielt, allerdings werden die Spuren 0 und 1 mitbenutzt, was einen Speicherplatzgewinn von 2 Spuren x 9 Sektoren x 512 Byte = 9 K ergibt. Es stehen also 178 K Speicherplatz zur Verfügung.

Das IBM CP/M Format

Dieses Format ist identisch mit dem IBM PC Format unter CP/M. Jede Spur ist in 8 Sektoren zu je 512 Byte unterteilt. Als CP/M Spur wird nur die Spur 0 verwendet, so daß sich ein Speicherplatz von 154 K ergibt.

Diese verschiedenen Formatierungsarten erreichen Sie durch folgende CP/M-Befehle:

1. FORMAT,S – AMSDOS SYSTEM Format (CP/M und BASIC)
2. FORMAT,V – AMSDOS SYSTEM Format ohne CP/M
3. FORMAT,D – AMSDOS DATEN Format
4. FORMAT,I – IBM PC CP/M Format

Die ersten CP/M Rechner hatten ihre Sektoren auf 128 Byte formatiert. Diese 128 Byte nennt man einen Record. Um kompatibel zu bleiben, wurde beim CPC softwaremäßig die Möglichkeit geschaffen, auch auf Records zuzugreifen, obwohl die kleinste erreichbare Einheit eigentlich aus einem Sektor (512 Byte) besteht.

Für ein 10 K Programm, müßte man z. B. in der Directory alle Records eintragen, die Teile dieses Programmes enthalten. Bei einem 10 K Programm wären das 80 (!) Einträge, was sehr viel ist. Um diesen Speicheraufwand zu verringern, wurden jeweils 8 Records oder 2 Sektoren zu Blöcken zusammengefaßt. Ein Block ist also ein Kilobyte groß. Das erklärt auch, warum die kleinsten Programme mindestens 1 K lang sind: Immer der ganze Block wird in der Directory reserviert. Er kann also nicht mehr anderweitig verwendet werden.

Die ersten beiden Sektoren der ersten freien Spur sind Block 0, die Sektoren 3 und 4 der ersten freien Spur sind Block 1, usw. Daraus läßt sich folgende mathematische Beziehung herleiten, mit der man aus der Blockzahl den Sektor und die Spur ausrechnen kann, wo sich ein File auf der Diskette befindet. Das ist deshalb notwendig, da in der Directory nur die Blöcke eingetragen werden.

$$\text{SPUR} = \text{INT}(\text{BLOCK} * 2 / \text{BLOCKANZAHL})$$

$$+ \text{SYSTEMSPUREN}$$

$$\text{SEKTOR} = \text{BLOCK} * 2 - \text{SPUR} * \text{BLOCKANZAHL} + \text{SYSTEMSPUREN} * \text{BLOCKANZAHL}$$

Die Variablen BLOCKANZAHL und SYSTEMSPUREN hängen allerdings vom Diskettenformat ab:

Format	Blockanzahl	Systemspuren
AMSDOS CP/M	9	2
Vendor	9	2
Daten	9	0
IBM	8	1

Saven Sie einmal folgendes File ab: SAVE "Name",b,0,1024. Es wird jetzt genau 1 K abgespeichert. Aber wenn Sie CAT tippen, werden 2 K angezeigt. Das kommt daher, da bei jedem Filetyp (außer ASCII-Dateien) als erster Record (128 Byte) ein Headerteil abgelegt wird, der Informationen über das File enthält, was bei ASCII-Dateien allerdings fehlt. Aber wie unterscheidet AMSDOS zwischen ASCII-Dateien und anderen Files? Die ersten 66 Byte des ersten Records werden zusammengezählt. Entsprechen Sie der Prüfsumme in den Bytes 67 und 68, dann handelt es sich um keine ASCII-Datei. Stimmt die Prüfsumme zufällig doch mal, so ist das Pech, da AMSDOS die Datei nun für eine nicht ASCII-Datei hält.

Der Aufbau der Directory

Die Directory ist jeweils in der ersten freien Spur in den Sektoren 1-4 untergebracht:

AMSDOS CP/M Format: Spur 2, Sektor 1-4
 Vendor Format: Spur 2, Sektor 1-4
 Daten Format: Spur 0, Sektor 1-4
 IBM CP/M Format: Spur 1, Sektor 1-4

Pro Eintrag sind 32 Bytes vorgesehen. Sehen wir uns doch mal ein Beispiel an:

```
000 00 43 48 48 44 49 53 43 20 43 4F 4D 00 00 00 13 .CHKDISC COM...
010 44 45 46 00 00 00 00 00 00 00 00 00 00 00 00 DEF.....
```

Dieser Eintrag befindet sich auf Spur 2, Sektor 2 der CP/M Diskette. Das erste Byte ist eine 0. Dieses Byte stellt die Usernummer dar. Insgesamt kann man 16 verschiedene Directoryebenen anwählen (0-15). In diesem Falle ist das File im Directory 0 enthalten. Steht im ersten Byte ein Wert größer als 15 dez oder 0F hex, dann ist das File für AMSDOS nicht mehr ladbar und erscheint auch in keinem Directory mehr. Beim Löschen mit ERA wird also nur dieses Byte auf einen Wert größer als 15 gesetzt. Wollen Sie dieses File wieder reaktivieren, müssen Sie in dieses Byte einen Wert von 0 bis 15 schreiben, was mit dem abgedruckten Programm ohne Schwierigkeiten möglich ist. Allerdings dürfen die belegten Blöcke nicht schon anderweitig beschrieben worden sein.

Damit läßt sich's leben

Die nächsten 13 Bytes beziehen sich auf den Namen. Wie Sie sofort sehen, wird der Punkt nicht mehr abgespeichert und der Name mit Spaces aufgefüllt. Anschließend folgen drei 0 Bytes und ein Byte, das die Anzahl der Records angibt, die das File belegt (hier 13). Nun sind noch 16 Bytes frei. Hier werden die belegten Blöcke eingetragen. Was passiert aber, wenn das File größer als 16 K ist, denn dann ist kein Platz mehr in der Directory. In einem solchen Falle richtet AMSDOS einfach noch einen zweiten Eintrag ein, in dem dann die restlichen Blöcke stehen. Bei ganz langen Files werden sogar drei Einträge gebraucht. Daß noch Einträge folgen, merkt AMSDOS an der Anzahl der Records. Das 13. Byte erhält jetzt doch noch eine Funktion: Beim ersten Eintrag eines Files ist es 0, beim zweiten 1, beim dritten 2. So kann der Rechner bei jedem File unterscheiden, um den wievielten Eintrag es sich handelt.

Noch zwei wichtige Dinge

Die letzten 3 Buchstaben des Namens sind ja die sogenannte Extension.

1. Wird Bit 7 des ersten Buchstabens der Extension, also das 10. Byte des Directoryeintrages gesetzt, erhält das File die Bemerkung READ ONLY, d.h., es kann unter AMSDOS oder CP/M nicht mehr gelöscht oder umbenannt werden. Wollen Sie den READ ONLY Status setzen, müssen Sie zum Code des ersten Buchstabens der Extension 128 addieren und den Sektor neu abspeichern. Wollen Sie den READ ONLY Status eines Files beseitigen, subtrahieren Sie 128 vom Code des ersten Buchstabens und speichern Sie den entsprechenden Directorysektor neu ab.
2. Ist Bit 7 des zweiten Buchstabens der Extension gesetzt, also das 11. Byte des Directoryeintrages, wird das File „unsichtbar“, d.h., es kann noch geladen wer-

den, taucht aber unter AMSDOS und CP/M in keiner Directory mehr auf. Das Programm DISC DOKTOR macht natürlich auch solche Files sichtbar.

Die Bedienung

Tippen Sie das Programm sorgfältig ab und speichern Sie es mit GOTO 10000 unter dem Namen "DISKDR" ab. Äußerste Sorgfalt ist deshalb geboten, weil schon ein falsches Byte Daten auf der Diskette zerstören kann. Nehmen Sie deshalb z.B. Ihre Backup Kopie der LOGO-Diskette zum Ausprobieren.

Der Bildschirm ist in drei Fenster unterteilt: Der obere, gelbe Teil enthält Informationen zu Laufwerk, Sektor, Spur und Format. Im zweiten, hellblauen Fenster sind 64 Bytes hexadezimal dargestellt. Dahinter jeweils die entsprechenden Charaktere im ASCII-Format. Nicht darstellbare Zeichen werden durch einen Punkt (.) angezeigt. 512 Bytes werden eingelesen, die man sich in 8 Teilen zu je 64 Bytes ansehen kann. Die Anzeige „BLOCK“ im oberen Fenster zeigt den augenblicklich angewählten Teil. Das dritte, weiße Fenster zeigt das Menü oder eine Meldung.

Das Hauptmenü

Hier haben Sie folgende Funktionen zur Verfügung:

T – TEXTEINGABE

Ab einer Adresse, die Sie in Hex eingeben müssen, wird ein String abgelegt, den Sie ebenfalls eingeben müssen. So können Sie z.B. Programmnamen ändern. Beachten Sie, daß der Punkt für eines von vielen, nicht darstellbaren Zeichen steht.

E – EDIT

Geben Sie eine Adresse ein. Diese muß allerdings durch acht teilbar sein. Ab dieser Adresse können Sie acht Bytes verändern. Die Originalbytes sind oben abgedruckt. Wenn Sie welche übernehmen wollen, geschieht dies mit dem COPYCURSOR. Es müssen aber immer acht Bytes auf einmal geändert werden. Vergessen Sie nicht, den Block wieder abzuspeichern, wenn Sie die Änderungen vorgenommen haben.

D – DRIVE

Damit können Sie zwischen Laufwerk A und B hin- und herschalten.

T – TRACK

Damit können Sie eine Spur von 0-39 auswählen.

S – SEKTOR

Damit wählen Sie einen Sektor von 1-9 oder von 1-8 (IBM Format) aus.

B – BLOCK

Damit werden die nächsten 64 Bytes auf dem Bildschirm ausgegeben. Ist das Ende erreicht, wird wieder von vorne angefangen.

F – FORMAT

Hier können Sie eines der drei Formate AMSDOS CP/M, AMSDOS Daten, IBM CP/M wählen. Für Vendor wählen Sie AMSDOS CP/M.

CTRL+C – COPY TO PRINTER

Der im Augenblick im Speicher befindliche 512 Bytes lange Block wird auf dem Drucker ausgegeben. Die Angaben Track und Sektor, die mit ausgedruckt werden, bezie-

hen sich auf den angewählten Track/Sektor. Dieser ist noch nicht im Speicher, wenn er nicht zuvor mit CTRL + L geladen wurde.

CTRL+L – LOAD BLOCK

Der augenblicklich durch Track/Sektor festgelegte Block wird von der Diskette geladen. Gibt es Probleme beim Laden, kommt die übliche Fehlermeldung: "RETRY, IGNORE or CANCEL". Mit I oder C kommen Sie ins Menü zurück, mit R versuchen Sie es nochmal. (Anm.: Wenn Sie ein Format gewählt haben, das auf der Diskette nicht vorhanden ist, kommt es zu eben dieser Fehlermeldung. Sie müssen es dann mit einem anderen Format versuchen. Es kann aber auch wirklich ein Lesefehler sein.)

CTRL+S – SAVE BLOCK

Der im Augenblick im Speicher befindliche Block wird in die durch Track/Sektor festgelegte Spur und in den festgelegten Sektor geschrieben. Vorsicht: Wenn Sie Track oder Sektor nach dem Laden ändern, wird der Block auf einen anderen Diskettenbereich geschrieben, was aber auch beabsichtigt sein kann.

WICHTIG: Sammeln Sie erst mit einer BACKUP-Diskette gründliche Erfahrungen, bevor Sie sich an Ihre wichtigen Disketten wagen, denn ehe man sich versieht, sind Daten dahin.

Bei Fragen wenden Sie sich direkt an mich:

Andreas Zallmann
Eulenweg 5
4923 Extertal 1
Tel.: 05262/2256 (ab 16 Uhr)

Discdoktor

```

100 REM =====
110 REM
120 REM     DISC DOCTOR- (C) 1985 by
130 REM
140 REM     Andreas Zallmann
150 REM     Eulenweg 5
160 REM     4923 Extertal 1
170 REM     Tel.: 05262/2256
180 REM           (ab 16 Uhr)
190 REM
200 REM     Nur fuer Besitzer des Dis-
210 REM     kettenlaufwerks !!!
220 REM
230 REM     EXCLUSIV FUER CK'
240 REM
250 REM =====
260 REM     INITIALISIERUNG
270 REM =====
280 MEMORY 39999
290 FOR i=41000 TO 41029:READ a:POKE i,a
: NEXT
300 track=0:sektor=1:drive=0:format=0:f1
=&X1000000:msek=9:DIM f$(2):f$(0)="AMSDOS
CP/M":f$(1)="AMSDOS DATEN":f$(2)="IBM
CP/M"
310 block=1
320 FOR I=40000 TO 40511:POKE I,0:NEXT
330 MODE 1:INK 0,26:INK 1,0:INK 2,25:INK
3,23:BORDER 26
340 PAPER 1:CLS
350 WINDOW #1,2,39,2,5:PEN #1,1:PAPER #1
,2:CLS #1:WINDOW #1,3,38,3,4
360 WINDOW #2,2,39,7,16:PEN #2,1:PAPER #
2,3:CLS #2:WINDOW #2,3,38,8,15
370 WINDOW #0,2,39,18,24:PEN 1:PAPER 0:C
LS:WINDOW #0,3,38,19,23
380 GOSUB 1720:GOSUB 1620:GOSUB 1530
390 REM =====
400 REM     TASTATURABFRAGE IM HAUPTMENUE
410 REM =====
420 V=0:WHILE V=0
430 IF NOT INKEY(23) THEN V=1
440 IF NOT INKEY(63) THEN V=1
450 IF NOT INKEY(58) THEN V=2
460 IF NOT INKEY(61) THEN V=3
470 IF NOT INKEY(51) THEN V=4
480 IF NOT INKEY(60) THEN V=5
490 IF NOT INKEY(54) THEN V=6
500 IF NOT INKEY(53) THEN V=10
510 GOTO 580
520 REM =====
530 REM     CTRL GEDRUECKT
540 REM =====
550 IF NOT INKEY(36) THEN V=9
560 IF NOT INKEY(60) THEN V=8
570 IF NOT INKEY(62) THEN V=7
580 WEND
590 ON V GOSUB 750,610,1480,1420,1360,11
90,1010,790,910,1240
600 GOTO 420
610 REM =====
620 REM     8 BYTES EDITIEREN
630 REM =====
640 GOSUB 1780:IF adr/8 <> INT(adr/8) TH
EN PRINT:PRINT"Adresse nicht durch 8 tei
lbar.":WHILE INKEY$<>"":WEND:CALL &BB18:
GOTO 640
650 CLS:PRINT:PRINT " ";
660 FOR i=0 TO 7:PRINT HEX$(PEEK(40000+i
+adr),2);:NEXT
670 PRINT:LINE INPUT "BYTES: ";t$
680 t$=UPPER$(t$)
690 IF LEN(t$)<>16 THEN CLS:PRINT:PRINT"
8 Bytes muessen eingegeben werden.":WHIL
E INKEY$<>"":WEND:CALL &BB18:GOTO 650
700 FOR i=1 TO 16:a$=MID$(t$,i,1):IF a$<
"0" OR a$>"F" OR (a$>"9" AND a$<"A") THE
N CLS:PRINT:PRINT "Nur Hexzahlen eingebe
n.":WHILE INKEY$<>"":WEND:CALL &BB18:GOT
O 650
710 NEXT i
720 FOR i=0 TO 7:a$=MID$(t$,i*2+1,2)
730 POKE 40000+i+adr,VAL ("%"+a$):NEXT i
740 GOSUB 1620:GOTO 1530
750 REM =====
760 REM     TEXT EINGEBEN
770 REM =====
780 GOSUB 1780:PRINT:LINE INPUT "TEXT ?"
:T$:FOR i=1 TO LEN(T$):POKE 40000+adr+i-
1,ASC(MID$(t$,i,1)):NEXT i:GOSUB 1530:GO
TO 1620
790 REM =====
800 REM     BLOCK SPEICHERN
810 REM =====
820 CLS:PRINT:PRINT:PRINT"BLOCK WIRD GES
PEICHERT."
830 POKE &A036,DRIVE
840 POKE &A038,TRACK
850 POKE &A03A,(SEKTOR OR f1)
860 POKE &A042,&85
870 CALL 41000
880 POKE &A042,&84

```

```

890 block=1
900 GOTO 1620
910 REM =====
920 REM          BLOCK LADEN
930 REM =====
940 block=1
950 CLS:PRINT:PRINT:PRINT"BLOCK WIRD GEL
ADEN."
960 POKE &A036,DRIVE
970 POKE &A038,TRACK
980 POKE &A03A,(SEKTOR OR f1)
990 CALL 41000
1000 GOSUB 1720:GOSUB 1620:GOTO 1530
1010 REM =====
1020 REM BLOCK AUF DEM DRUCKER AUSGEBEN
1030 REM =====
1040 CLS:PRINT:PRINT"DRUCKEN - BITTE WAR
TEN.":PRINT #8,"      DISC DOCTOR BY ANDR
EAS ZALLMANN"
1050 PRINT #8:PRINT #8,"      SEKTOR :";S
EKTOR
1060 PRINT #8,"      TRACK  :";TRACK
1070 PRINT #8,"      FORMAT : ";F$(FORMAT
)
1080 PRINT #8
1090 FOR A=40000 TO 40511 STEP 16
1100 PRINT #8,"      ";HEX$(A-40000,3);"
";
1110 FOR B=0 TO 15
1120 PRINT #8,HEX$(PEEK(A+B),2);" ";:NEX
T B
1130 PRINT #8," ";
1140 FOR B=0 TO 15
1150 CHAR=PEEK(A+B):IF CHAR<32 OR char>1
22 OR (char<97 AND char>90)THEN CHAR=46
1160 PRINT #8,CHR$(CHAR);:NEXT:PRINT #8:
NEXT
1170 PRINT #8,"      READY."
1180 GOTO 1620
1190 REM =====
1200 REM          NAECHSTEN BLOCK AUSGEBEN
1210 REM =====
1220 block=block+1:IF block=9 THEN block
=1
1230 GOSUB 1720:GOTO 1530
1240 REM =====
1250 REM          FORMAT AUSWAEGHLEN
1260 REM =====
1270 CLS:PRINT "FORMAT AUSWAEGHLEN":PRINT
1280 PRINT "0 ..... AMSDOS CP/M FORMAT"
1290 PRINT "1 ..... AMSDOS DATEN FORMAT"
1300 PRINT "2 ..... IBM CP/M FORMAT"
1310 FORMAT=255:WHILE FORMAT=255
1320 IF NOT INKEY(64) THEN FORMAT=1:f1=&
X11000000:msek=9
1330 IF NOT INKEY(65) THEN FORMAT=2:f1=0
:msek=8
1340 IF NOT INKEY(32) THEN FORMAT=0:f1=&
X1000000:msek=9
1350 WEND:GOSUB 1620:GOTO 1720
1360 REM =====
1370 REM          SEKTOR AUSWAEGHLEN
1380 REM =====
1390 WHILE INKEY#<>"":WEND
1400 CLS:PRINT:PRINT"SEKTOR AUSWAEGHLEN (
1 -";msek;") ";:INPUT sektor
1410 IF SEKTOR<1 OR SEKTOR>msek THEN 140
0 ELSE GOSUB 1720:GOTO 1620
1420 REM =====
1430 REM          TRACK AUSWAEGHLEN
1440 REM =====
1450 WHILE INKEY#<>"":WEND
1460 CLS:PRINT:INPUT "TRACK AUSWAEGHLEN (
0-39) ";TRACK
1470 IF TRACK<0 OR TRACK>39 THEN 1460 E
LSE GOSUB 1720:GOTO 1620
1480 REM =====
1490 REM          DRIVE AUSWAEGHLEN
1500 REM =====
1510 IF DRIVE=1 THEN DRIVE=0 ELSE DRIVE=
1
1520 GOTO 1720
1530 REM =====
1540 REM          BLOCK AUSGEBEN
1550 REM =====
1560 CLS #2:FOR A=40000+BLOCK*64-64 TO 4
0000+BLOCK*64-5 STEP 8
1570 PRINT #2,HEX$(A-40000,3);" ";
1580 FOR B=0 TO 7:PRINT #2,HEX$(PEEK(B+A
),2);" ";:NEXT
1590 FOR B=0 TO 7:CHAR=PEEK(A+B):IF CHAR
<32 THEN CHAR=46
1600 PRINT #2,CHR$(CHAR);:NEXT B,A
1610 RETURN
1620 REM =====
1630 REM          MENUE ANZEIGEN
1640 REM =====
1650 CLS
1660 PRINT "E-EDIT          CTRL+L-BLOCK LA
DEN"
1670 PRINT "X-TEXT EING. CTRL+S-BLOCK SP
EICHERN"
1680 PRINT "D-DRIVE A/B  CTRL+C-COPY TO
PRINTER"
1690 PRINT "T-TRACK          B-BLOCK WA
EHLLEN"
1700 PRINT "S-SEKTOR        F-FORMAT A
NWAEGHLEN";
1710 RETURN
1720 REM =====
1730 REM          PARAMETER ANZEIGEN
1740 REM =====
1750 CLS #1:PRINT #1,"DRIVE: ";CHR$(driv
e+65);"      SEKTOR:";sektor;"      TRACK:";tra
ck;
1760 PRINT #1,"BLOCK:";block;"      FORMAT:
";f$(format);
1770 RETURN
1780 REM =====
1790 REM          HEZZAHLEINGABE
1800 REM =====
1810 WHILE INKEY#<>"":WEND
1820 CLS:PRINT:PRINT "Bitte geben Sie di
e Adresse in HEX":PRINT "ein (3 Ziffern)
.";
1830 LINE INPUT t$:IF LEN(t#)<>3 THEN 18
20
1840 t#=UPPER$(t#)
1850 FOR i=1 TO 3:a#=MID$(t#,i,1):IF a#<
"0" OR a#>"F" OR (a#<"A" AND a#>"9") THE
N 1820 ELSE NEXT
1860 adr=VAL ("%"+t#):CLS:RETURN
1870 REM =====
1880 REM          DATEN FUER MASCHINENPROGRAMM
1890 REM =====
1900 DATA 33,66,160,205,212,188,34,67,16
0,121,50,69,160,30,0,22,0,14,0,33,64,156
,223,67,160,201,132,0,0,0
9970 REM =====
9980 REM          SAVE-ROUTINE
9990 REM =====
10000 SAVE "DISKDR"

```

Topprogramm des Monats:

Showdown in Tombstone City

Die Sonne steht senkrecht über Tombstone City. Es ist High Noon! Die Luft flimmert vor Hitze. Unbeweglich stehen sich zwei Cowboys auf der leergefegten Straße gegenüber: Der stahlharte Held, der Sheriff dieser Stadt (mit anderen Worten der Spieler) und einer der zahlreichen Gesetzeslosen, die immer wieder die unbescholtenen Bürger dieser Stadt bedrohen. Beide wissen: Wer zuletzt zieht, stirbt zuerst. Der Job ist gefährlich, aber er lohnt sich, denn auf die Köpfe der Banditen sind hohe Belohnungen ausgesetzt.

Nach dem Laden des Programmes erscheint die Titelseite der Zeitung von Tombstone City, in der berichtet wird, daß der alte Marshall Mat Dillon von Billy the Kid erschossen wurde. Gleichzeitig wird ein neuer Marshall gesucht. Dieser Job erfordert eine ruhige Hand, viel Konzentration und Nerven wie Stahlseile. Wenn Sie sich wider Erwarten zu diesem Job befähigt fühlen, drücken Sie "Y" (YEAH), ansonsten "N" (NOP). Haben Sie die Aufgabe übernommen, sehen Sie den Steckbrief eines Mannes, der kurz darauf vor ihnen auftaucht. Jetzt heißt es: Sie oder er. Nur einer wird die Straße aufrecht gehend wieder verlassen. Sie stehen sich gegenüber und warten. Plötzlich zieht der Schurke seinen Colt.

Wie er, haben auch Sie sechs Kugeln in Ihrem schweren Revolver. Sie müssen nun versuchen, den Gegner zu treffen. Das heißt, das Fadenkreuz auf den Kopf oder auf den Brustkorb des Gegners zu lenken und abzudrücken (Feuer). Ihr Gegner hat am Anfang eine Trefferrate von 50%, durchschnittlich trifft also jeder zweite Schuß. Treffen Sie den Schußarm Ihres Gegners, so sinkt diese Prozentzahl um 25%, beim anderen Arm um 15%, bei der Hose um 20% und bei den Stiefeln um 5%. Sinkt die Prozentzahl unter null, so ist der Gegner so schwer verletzt, daß er völlig kampfunfähig ist. Um sich eines Gegners zu entledigen, können Sie ihn auch dreimal am Arm treffen. Haben beide Gegner ihre sechs Patronen verschossen, so ist das Duell unentschieden und nach einer Zeit zum Nachladen, stehen Sie Ihrem Gegner erneut gegenüber.

Haben Sie verloren, erscheint eine Zeitung mit der Meldung, daß Sie soeben von einem Gangster erschossen wurden (mit dem Bild des gemeinen Schurken). Es wird nun auch wieder ein neuer Sheriff gesucht. Wollen Sie sich erneut der Herausforderung stellen?

Haben Sie gewonnen, wird abgerechnet. Sie erhalten die ausgesetzte Belohnung und müssen jede verschossene Kugel ersetzen. Eine Kugel kostet 10 Dollar, da Sie nur Silberkugeln benutzen. Also gehen Sie sparsam mit Ihrer Munition um.

Kaum haben Sie Ihren Colt wieder geladen, wartet ein neuer, noch gefährlicherer Gegner auf Sie. Haben Sie tat-



sächlich alle 29 verschiedenen Gangster bezwungen, beginnt das Spiel von vorne, allerdings jetzt mit einer höheren Trefferwahrscheinlichkeit.

Wenn Sie zweimal ESCAPE drücken (BREAK), dann kommen Sie ins Menü.

S-STOP. Das Spiel wird abgebrochen.

D-DEFINE ITEMS. Hier können Sie eigene Parameter eingeben. Zuerst werden Sie nach der Steuerung gefragt. Diese ist standardmäßig auf Joystick 0 eingestellt, die entsprechenden Tastencodes sind:

hoch	- 72
runter	- 73
links	- 74
rechts	- 75
Feuer	- 76

Wollen Sie eine andere Steuerung wählen, so müssen Sie bei der entsprechenden Richtung den TASTENCODE der von Ihnen gewünschten Taste eingeben. Die Tastencodes finden Sie im Benutzerhandbuch des 464 Anhang III, Seite 16. Anschließend geben Sie noch die prozentuale Trefferquote ein, bei der Ihr Gegner startet (normalerweise 50%). Anschließend geben Sie noch die Prozentzahlen ein, um die die Trefferquote bei diversen Treffern verringert wird.

I-RESET ITEMS. Die veränderten Werte werden wieder auf ihren Ursprungswert gesetzt.

Z-ZURÜCK. Kehrt zum Spiel zurück.

Das Programm besteht aus fünf Listings. Tippen Sie zuerst Listing 1 ein, das ein Titelbild erstellt und die anderen Programmteile nachlädt. Achtung: CPC 664 und 6128 Anwender müssen Zeile 220 im Listing 1 wie folgt ändern:

```
220 matr = PEEK (&B736) + 256 * PEEK (&B737)
```

Saven Sie Listing 1 mit GOTO 10000 auf eine leere Kasette ab. Starten Sie das Programm aber noch nicht.

Listing 2 erstellt eine in hochauflösender, farbiger Grafik gehaltene Hintergrundszene, nämlich eine Straße in Tombstone City. Nach dem äußerst sorgfältigen Abtippen starten Sie dieses Programm mit RUN. Ist der Screen erstellt,

wird er nach Adresse 20000 umgeladen. Das Programm speichert den Screen nach Tastendruck automatisch ab. Sorgen Sie dafür, daß der Screen hinter Listing 1 abgespeichert wird.

Listing 3 erstellt die Grafiks. Tippen Sie dieses Listing sorgfältig ein und starten Sie wieder mit RUN. Die Grafiks, festgehalten in den DATA-Zeilen, werden nun in den Speicher gelesen. Fehlerhaft eingetippte Zeilen erkennt der Rechner. Nach einer Fehlermeldung müssen Sie die entsprechende Zeile berichtigen und erneut mit RUN starten. Sind alle Daten ohne Fehler, werden die Grafiks nach Tastendruck hinter dem Screen abgespeichert. Listing 3 benötigen Sie wie schon Listing 2 nun nicht mehr.

Genauso verfahren Sie bitte mit Listing 4, nur daß hier das Maschinenprogramm in DATAs festgehalten ist.

Listing 5 schließlich ist das Kontrollbasicprogramm. Tippen Sie es sorgfältig ein und saveen Sie es mit GOTO 10000 hinter das Maschinenprogramm (aus Listing 4). Starten Sie dieses Programm nicht!

Nun spulen Sie bitte die Kassette zurück und laden bzw. starten Sie Listing 1 mit RUN "SHOWDOWN". Diskettenbesitzer verhalten sich genauso, nur daß bei Ihnen kein Tastendruck zum Abspeichern notwendig ist und das Spulen entfällt.

Viel Spaß wünschen Andreas Zallmann und Cristoph Schillo

```

100 REM =====
==
110 REM
120 REM     SHOWDOWN - (C) 1985 by
130 REM
140 REM     Andreas Zallmann
150 REM     Eulenweg 5
160 REM     4923 Extertal 1
170 REM     Tel.:05262/2256
180 REM
190 REM =====
200 MODE 1
210 SYMBOL AFTER 32
220 matr=PEEK(&B296)+256*PEEK(&B297)
230 INK 0,12:INK 1,3:INK 2,0:INK 3,13:BO
RDER 12
240 SYMBOL AFTER 240
250 WINDOW 4,36,2,16:WINDOW#1,1,40,1,25:
WINDOW#2,16,25,1,40
260 PAPER #2,1:PEN #1,3:CLS #2:PAPER 1:P
EN 2:CLS:PAPER #1,0
270 FOR a=16 TO 48 STEP 2:PLOT a,382,1:D
RAW 48,144:PLOT 560+a,144:DRAW 576,382:N
EXT
280 PLOT 15,384,2:DRAWR 562,0:DRAW 608,1
42:DRAWR -562,0:DRAW 15,384
290 PLOT 240,0:DRAWR 0,140:PLOT 400,0:DR
AWR 0,140:PLOT 240,384:DRAWR 0,15:PLOT 4
00,384:DRAWR 0,15
300 a$="00000111110011100100010111100000
0000000010001000101101101000100000000000
0100010001011111011110000000000000100010
00101010101000100000000000001000011100100
01011111"
310 b$="00011100111110011100100010111110
001000000010001000101100101000000000111
0000100010001010101011100000000001000100
01000101001101000000000111100001000011100
10001011111"
320 LOCATE 1,2:FOR a=1 TO LEN(a$):IF MID
$(a$,a,1)="1" THEN PRINT CHR$(143); ELSE
PRINT " ";
330 NEXT
340 LOCATE 1,9:FOR a=1 TO LEN(b$):IF MID
$(b$,a,1)="1" THEN PRINT CHR$(143); ELSE
PRINT " ";
350 NEXT
360 SYMBOL 253,16,48,110,126,255,127,60,
8
370 SYMBOL 254,20,62,126,255,255,126,56,
96
380 SYMBOL 255,12,62,127,127,126,219,14,
4
390 c$="11110100100110010001000000000000
010000100101001010001000000000000100001
00101001010101010000000000001111011110100
1010101000000000000010100101001011111
00000000"
400 d$="000000000000001110001100100010100
1000000000000010010100101000101101000000
0000000100101001010101011010000000000000
100101001010101010110000000000001001010
0101111101011000000000000011100011000101
001001"
410 PEN 0
420 LOCATE 1,2:PRINT SPACE$(200):LOCATE
1,2:FOR a=1 TO LEN(c$):IF MID$(c$,a,1)="
1" THEN FOR b=1 TO 100 STEP 7:SOUND 1,b*
10,1,15:BORDER RND*26:NEXT:BORDER 12:PRI
NT CHR$(INT(RND*3+253)); ELSE PRINT " ";
430 NEXT
440 LOCATE 1,9:PRINT SPACE$(200):LOCATE
1,9:FOR a=1 TO LEN(d$):IF MID$(d$,a,1)="
1" THEN FOR b=1 TO 100 STEP 7:SOUND 1,b*
10,1,15:BORDER RND*26:NEXT:BORDER 12:PRI
NT CHR$(INT(RND*3+253)); ELSE PRINT " ";
450 NEXT
460 PEN 3
470 LOCATE 24,1:PRINT"von"
480 a$="A.Zallmann &":LOCATE 22,3:GOSUB
580:a$="C.Schillo":LOCATE 22,6:GOSUB 580
490 a$="Exklusiv":LOCATE 2,9:GOSUB 580:a
$="fuer CK":LOCATE 2,12:GOSUB 580
500 WINDOW SWAP 0,1:PEN 2
510 a$="Stand by -":LOCATE 2,20:GOSUB 58
0:a$="Program loading":LOCATE 1,22:GOSUB
580
520 a$="SHOWDOWN":LOCATE 29,18:GOSUB 580
:a$="in":LOCATE 32,20:GOSUB 580:a$="Tomb
stone":LOCATE 29,22:GOSUB 580:a$="city":
LOCATE 31,24:GOSUB 580
530 MEMORY 19999
540 LOAD"!showscr"
550 LOAD"!showgr"
560 LOAD"!showmc"
570 RUN"!showbas"
580 FOR a=1 TO LEN(a$)
590 FOR b=0 TO 7
600 adr=PEEK(matr+(ASC(MID$(a$,a,1))-32)
*8+b)
610 A(b)=adr:NEXT:SYMBOL 254,A(0),A(0),A
(1),A(1),A(2),A(2),A(3),A(3):SYMBOL 255,
A(4),A(4),A(5),A(5),A(6),A(6),A(7),A(7)
620 PRINT CHR$(254);CHR$(10);CHR$(8);CHR
$(255);CHR$(11);
630 NEXT
640 RETURN
10000 SAVE "SHOWDOWN"

```

Listing 2

```

100 REM =====
==
110 REM
120 REM   SHOWCITY - (C) 1985 by
130 REM
140 REM   Chrsitoph (Psy) Schillo
150 REM   Bruchweg 4
160 REM   4923 Extertal 1
170 REM   Tel.: 05262/835
180 REM
190 REM =====
200 MEMORY 19999:RANDOMIZE 100000
210 MODE 1:INK 0,12:INK 1,0:INK 2,3:BORD
ER 12
220 FOR a=0 TO 38 STEP 2:PLOT 52,48+2*a:
DRAW 320/(40-a),800/(40-a),2:NEXT
230 fd=1.3:f=0:FOR a=0 TO 400 STEP 2:PLO
T 518+f,a:DRAWR 64*fd,0,1 XOR INT(RND*1.
5):DRAWR 0,-2,1:f=f-0.45:fd=fd-0.0038:NE
XT
240 fd=1.3:f=32:FOR a=0 TO 210 STEP 2:PL
OT f,a:DRAWR 64*fd,0,1 XOR INT(RND*1.5):
DRAWR 0,-2,1:f=f+0.75:fd=fd-0.0038:NEXT
250 SYMBOL 240,0,0,0,24,60,60,44:SYMBOL
L 241,44,44,44,44,44,44,126:SYMBOL 24
2,86,89,151,109,180,234,158,85:SYMBOL 24
3,132,21,64,88,10,193,36,74
260 FOR a=1 TO 3:READ x,y:PLOT x,y:DRAW
320,800:NEXT
270 PAPER 2:FOR a=6 TO 22:LOCATE 8,a:PRI
NT CHR$(242);CHR$(243):LOCATE 33,a+3:PRI
NT CHR$(242);CHR$(243):NEXT
280 TAG:FOR a=62 TO 318 STEP 16:MOVE 136
,a:PRINT " ";:NEXT:TAGOFF
290 fd=1.38:f=0:FOR a=16 TO 400 STEP 2:P
LOT f,a,1:DRAWR -64*fd,80*fd:IF RND>0.3
THEN DRAWR 0,140,INT(RND*2)+1
300 f=f+0.816:fd=fd-0.0041:NEXT
310 FOR a=-16 TO 16 STEP 6:PLOT 0,a:DRA
W 320,800:NEXT
320 DRIGIN 0,0,136,640,400,0:fd=1.2012:f
=51.5:FOR a=50 TO 400 STEP 2:PLOT f,a:DR
AWR 64*fd,0,1 XOR INT(RND*1.5):DRAWR 0,-
2,1:f=f+0.75:fd=fd-0.0038:NEXT
330 DRIGIN 0,0,0,640,400,0
340 FOR a=1 TO 12:READ x,y:PLOT x,y:DRAW
320,800:NEXT
350 fd=1.3:f=0:FOR a=0 TO 400 STEP 2:PLO
T 630+f,a:DRAWR 64*fd,0,INT(RND*2)+1:f=f
-0.77:fd=fd-0.0038:NEXT
360 DATA 512,0,520,0,522,0,128,0,120,0,1
18,0,0,20,0,256,0,260,640,144,640,170,64
0,200,614,0,628,0,32,0
370 FOR a=616 TO 626 STEP 2:PLOT a,0:DRA
W 320,800,2:NEXT
380 fd=1.3:f=0:FOR a=0 TO 400 STEP 80:z=
128*fd:PLOT 630+f-2,a-4:DRAWR 0,z,1:DRAW
R 10,-20,1:DRAWR 0,-z,1:FOR b=0 TO 5:PLO
T 630+f+2*b,a-4*b:DRAWR 0,z,0:NEXT:DRAWR
2,-2,1:DRAWR 0,-z:DRAWR 2,-2:DRAWR 0,z,
0:DRAWR 2,-2:DRAWR 0,-z,1:f=f-30.8:fd=fd
-0.152:NEXT
390 FOR a=0 TO 2 STEP 2:PLOT 576-a,400:D
RAWR 0,-56:DRAW 616-a,272:DRAWR 0,128:PL
OT 596-a,304:DRAWR 0,80:PLOT 616,330-a:D
RAWR -40,54:NEXT
400 FOR a=0 TO 32 STEP 2:PLOT 640-a/2,20
2+a:DRAWR 0,200,2:NEXT:PLOT 623,236:DRAW
R 0,200,1
410 PEN 0:LOCATE 40,6:PRINT CHR$(237);
420 PAPER 0:PEN 1
430 FOR a=1 TO 200:PLOT RND*440+100,RND*
400,RND*2:NEXT
440 FOR a=0 TO 1:FOR c=0 TO 1:PEN 2-c:LO
CATE 35,1+23*a+c:FOR b=1 TO 6:PRINT CHR$(
240+c);:NEXT:NEXT:NEXT
450 FOR a=220 TO 280 STEP 2:PLOT 38,a:DR
AW 320,800,0:NEXT:PLOT 38,220:DRAWR 0,60
,2:DRAWR 4,0:DRAWR 0,-60:PLOT 42,280:DRA
W 320,800
460 FOR a=320 TO 400 STEP 2:PLOT 20,a:DR
AW 320,800,0:NEXT:FOR a=0 TO 2 STEP 2:PL
OT 20+a,320:DRAWR 0,81,1:DRAWR 26,0:DRAW
R 0,-34:PLOT 20+a,320:DRAW 320,800:PLOT
20,360+a:DRAW 320,800:NEXT
470 FOR a=64 TO 144 STEP 2:PLOT 154,a:DR
AWR 12,0,1:DRAWR 6,8,2:PLOT 192,192+a*0.
8:DRAWR 10,0,1:DRAWR 4,6,2:NEXT
480 FOR b=1 TO 16 STEP 2:FOR a=130 TO 14
0:PLOT 152+b,a:DRAWR 48-b/4,192,RND*2:NE
XT:NEXT
490 PLOT 50,252:DRAWR 8,18,1:DRAWR 0,14:
DRAWR -8,-18:DRAWR 0,14:DRAWR 8,18:PLOT
64,284:DRAWR 0,28:PLOT 64,298:DRAWR 8,18
:PLOT 72,300:DRAWR 0,28
500 PLOT 80,312:DRAWR 0,28:DRAWR 12,26:P
LOT 92,336:DRAWR -12,-24:DRAWR 0,16:DRAW
R 8,16
510 PLOT 98,348:DRAWR 0,26:DRAWR 8,16:DR
AWR 0,-14:DRAWR -8,-14:DRAWR 10,6
520 PLOT 114,382:DRAWR 0,18:PLOT 122,398
:PLOT 122,400
530 FOR a=1 TO 40:READ a$:FOR b=1 TO LEN
(a$):PLOT 272+b*2,80-a*2,VAL(MID$(a$,b,1
)):NEXT:NEXT
540 FOR i=0 TO 11:READ a:POKE 20000+i,a:
NEXT:CALL 20000
550 FOR i=0 TO 11:READ a:POKE 20000+i,a:
NEXT
560 CLS:PRINT "ACHTUNG ABSAVEN"
570 SAVE "showscr",b,20000,16384+12
580 DATA "000000000000000000001000000000
01
590 DATA "000000000000000000001110000001
101
600 DATA "0000000000000000000011110001111
111
610 DATA "000000000000000000001111111011111
111
620 DATA "000000000000000000001111111111111
111
630 DATA "000000000000000000001111111111111
111
640 DATA "0000011000000000111111111111111
111
650 DATA "0000111110000000111111111111111
111
660 DATA "0000111111111100111111111111111
11100000001111
670 DATA "0001111111111100122211111111111
211000001111111
680 DATA "0001111111111101222111111111222
2111000111111111
690 DATA "0001111111111111222222222222222
2111001111111111
700 DATA "0001111111111111222222222222222
2111011111111111
710 DATA "0000111111111101122222222222222
2111111111111111

```

```

720 DATA "000011111111101112222222222221
111101111111111111
730 DATA "000001111111101111111111111111
111011111111111111
740 DATA "000001111111101111111111111111
110111111111111111
750 DATA "000000111111110111111111111111
001111111111111111
760 DATA "000000111111111000101111111111
011111111111111111
770 DATA "000000011111111100000000000001
111111111111111111
780 DATA "000000001111111111111111111111
111111111111111111
790 DATA "000000000111111111111111111111
111111111111111111
800 DATA "000000000011111111111111111111
111111111111111111
810 DATA "000000000001111111111111111111
111111111111111111
820 DATA "000000000000111111111111111111
111111111111111111
830 DATA "000000000000110101011111111111
11
840 DATA "0000000000010101110101110101010
111
850 DATA "0000000000022110110111011011011
1122
860 DATA "0000000000022110110111011011011
0122
870 DATA "0000000000022010111111011010110
0122
880 DATA "0000000000022010111110111110110
1122
890 DATA "0000000000001110110101110110110
101
900 DATA "0000000000001011110101110101101
101
910 DATA "0000000000000010101010101010101
1
920 DATA "0000000000000000110101010111011
930 DATA "00000000000000001111111111111111
940 DATA "000000000000000012222222222221
950 DATA "00111111111111111111111111111111
111111111111111111
960 DATA "01111111111111111111111111111111
111111111111111111
970 DATA "11111111111111111111111111111111
111111111111111111
980 DATA 1,0,64,33,0,192,17,44,78,237,17
6,201,1,0,64,17,0,192,33,44,78,237,176,2
01

```

Listing 3

```

10 REM =====
==
20 REM          LISTING 3 - GRAFIK
30 REM =====
40 MEMORY 19999
50 pc=39983
60 FOR ze=10000 TO 10210 STEP 10:PRINT z
e;:sum=0:READ x$,s
70 FOR a=0 TO 99:a$=MID$(x$,a*2+1,2):POK
E pc,VAL("&"+a$):sum=sum+VAL("&"+a$):p
c=pc+1:NEXT a:IF sum<>s THEN PRINT "ERRO
R...":END
80 PRINT "OK":NEXT ze:PRINT "Achtung Abs
aven : SAVE 'SHOWGR',b,39983,2145"
90 SAVE "SHOWGR",b,39983,2145

```

```

95 END
1000 DATA "F0F096969696871E871E96969696
F0F0F0F03C3C2D1EE11E3C3C3C1E3C96F0F00000
00000000000030F071F5F2FAF5F50000308072C8
71E4F0F0F5F5FAFAF5F5000000000000000F0C0
F5E4FAEBF5E472FA30F00001000100000000000
0000FAC0F000", 11935
10010 DATA "0800080000000000000000000000
000000000000000000000000F000000000000000
70E070E063ACD37CF0F0000000000000000000000
00000000F00010F000F00000000000000000000000
0000F080F000000000000000000000000000000000
000000000000", 3358
10020 DATA "0000008B00F500720000224064E4
EAF4E4F5FACB870F8F3E000000000000000001100
FA00E400C800003000000000000000000000000000
000080000000000000000000000000000000000000
0F0F69691E0F1E0F0F4B3486030CF0F0F0F09090
F0F01E0F0F0F", 6356
10030 DATA "34C2030CF0F1F0F09696F0F0F0F0
70E070E030C00000000000000000000000000000000
0000003000F010F010F030F030F030F070F00000
000000000000000000100010001070F070F0F0B0
F0B0F030F030E010E000000000F0107810871078
008000000000", 8825
10040 DATA "E0E040F080F080F148D100D100D1
00D100000000000000000000000000000000000030
00F010F010F030F030F030F070F0000000000000000
0000000000100010001070F070F0F0B0F0B0F030
F030E0D0E05000210021004300A500D2002C00C0
0000E0E048F0", 8813
10050 DATA "80F080F040D100D100D100D10000
0000000000000010007000F030F0003000F030F0
70F0F0F0F0F0F0F0F0F030F030F030E030C030C0
30C030C03078F0F0E07000300030003000300010
0010212C215A21B4104800A4000000000000010E0
00F100F100F1", 9784
10060 DATA "00D100D100D100D100000000000000
00000000000000100030003000F010F030F070F0
F0F0F0F0F0F0007000F010F010F010F000F00070
0030F0F0F030E0108010D010F090C3D04B580000
000000000000000000000000000000000000A5A496F052F0
20F000D100D1", 9101
10070 DATA "00D100D10000000000000000000000
000000100030003000F010F030F070F0F0F0F0F0
F0F0007000F010F010F010F000F000700030F0F0
F030C0108010D010B490D2D04BF00010000000000
000000000000000000002D7896A452F020F000D1
00D100D100D1", 8952
10080 DATA "000000000000000000000000000010
0010003000F010F030F030F070F0F0F0F0F00030
0030003000300030002000200010F0B0F010E010
C010C01040104010C01000210030002100100000
00000000000058F090F058F080F000D100D100D1
00D100000000", 6854
10090 DATA "0000000000000000001000100030
00F010F030F030F070F0F0F0F0F0E030302080C0
D0302040F05020204040F0B0F010205040502090
90101050809050109000A0505060301060C0C0B0
9010507010B050B040B0205170D1C0D120D1FCF3
FCF3FCF3FCF3", 11104
10100 DATA "FCF3F0F0FCF3FCF3FCF3FCF3FCF3
FCF3FCF33CC3CC3CF0F0F1F079D279D279D23D87
79D2F1F0F1F0F1F0F5F4F9F2F2F8F4F4F4F4F8F2
F0E0FDFDF2FAF6FCF5F8FFF0F487FCF0F8F0F0F0
F078F0F0F078F0F0F078F0F0F0F0E000F000F000
F000F080F080", 21277
10110 DATA "F0C0F0E0B0E0B0E090F090F090F0
90F080F000F0800E80969052DC52CC52CC60CC80
CC004E5FAEAFF0F0EEFFEEFFEEFFEEFFEEFFEEFF0033
0033003300330033003300330033EE77EE77EE77

```

```

EE77EE77EE77EE77EE77CC00CC00CC00CC00CC00
CC00CC00CC00", 14307
10120 DATA "0020000200200002002000120020
0010A4504A05A4504A05A4504A05025048054800
B4004800B4004800B4005A00A5840020005000A1
001A00A4000000000000A4504A7084000B000000
0000000000005A4A70E000000000000000000000
000000020022", 3862
10130 DATA "0002002200130001004600AB4E05
AE554E05AE554E05AE574626AA558C004C008C00
4C008C00080026005F0011572327134E00000000
0000000000008260000000000000000000000000
0000AB885D4C278C000000000000000000000032
003200230032", 3562
10140 DATA "00230032003200FCE2746A756A47
2E656A652E65E274E274E200E200E2006A00C400
6A00E200E20033F074F177EE0000000000000000
0000C432CC11000000000000000000000000F100
F88B748833000000000000000000000000000000
000000000000", 6252
10150 DATA "00400030C000E000F080B0007000
E0002000000000000080004040100000000040201
0000000001000040800010C030E070E070F0000
00010204080400000808020000022000F0F096F0
D2F030F0D200000000030F0F078F0E1F0F0F0F0
001000300030", 5624
10160 DATA "F0F0961E2D964B1E8768F0E0F0F0
F0F0E000F0C078F0F0F070F00070000000000400
00008000C000E000E000F000F0000000000000000
0060009000900090009000510033007700FF11FF
91FFB3FFB3FFDDEEDDDFFDDFFDDFFDDFFDDFF99FF
887700300010", 12221
10170 DATA "0021CC21EE10EE00FC80CB60F000
960096002C00E06010A0606040A0008000800070
00400030004000400030F0EE80E6804000C09000
5000600080000033001000000000000000000000
000080100008000400030800040003000109040
20404080B000", 6637
10180 DATA "8000E00040008000000000000000
0000000000100030007000000000000000000000
0C000E000E000000000000000000000000000070C
0F0E00000000000000000000000000600050000070
007000300010000000000000000077F0F0F0E1F0C3
F0E1F0E17610", 3835
10190 DATA "EC00EC00F0F00FF04B789678F03C
7FBCF7FEF7FF8000F000F080F0C0F0C0F0C0F0C0
F0C04080204020301080004000C000A0007030FF
C0F700F70073007300E6F0000000FE00FE00FEF0
EE80887000800080007073FE62006201E2126200
800080000000", 12063
10200 DATA "F0805A80A5005A00000000000000
000000108000108000108000F0F000F0F0001080
0010800010800000C00000C00000C00070F08070
F08000C00000C00000C000006000006000006000
30F0C030F0C00060000060000060000030000030
0000300010F0", 6713
10210 DATA "E010F0E000300000300000300000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
000000000000", 848
50 pc=38000
60 FOR ze=10000 TO 10170 STEP 10:PRINT z
e::sum=0:READ x$,s
70 FOR a=0 TO 99:a$=MID$(x$,a*2+1,2):POK
E pc,VAL("&"a$):sum=sum+VAL("&"a$):p
c=pc+1:NEXT a:IF sum<>s THEN PRINT "ERRO
R...":END
80 PRINT "OK":NEXT ze:PRINT "Achtung Abs
aven : SAVE 'SHOWMC',b,38000,1800"
90 SAVE "SHOWMC",b,38000,1800
95 END
10000 DATA "1830F5E5219494CD7D94E1F1C9E5
F5D5C516007A4ECD34BD23147AFE0E20F4C1D1F1
E1C900001F0100000F2D001000B80B09DDE5FDE1
FD7E00DD21DFA0116000CD2298011307CDF9970C
CDF9970CCDF997040D0DCDF9970CCDF9970CCDF9
97FD7E06DD21", 12496
10010 DATA "0FA0112000CD2298011403CDF997
04CDF997FD7E04DD213F9D111000CD2298011402
CDF997FD7E02DD214F9C115000CD2298011301CD
F9970CCDF9970CCDF9970D0D04CDF9970C0CCDF9
97FD7E08327CA4FD7E0A327EA4327FA43E013280
A4DD216FA001", 10970
10020 DATA "1503CDF99704CDF99704CDF9970D
DD219FA0CDF9970D04CDF9970CCDF9970CCDF997
3E00CD4597CD48984721D0072B7CB520FB10F63E
01CD45973E06327AA4327BA43A7AA4FE00200B3A
7BA4FE002004327DA4C93A75A44F3A76A447C5CD
2B98C1F5C5CD", 12333
10030 DATA "6B97E178BCC2249679BD207C11C2
011B7AB320FBF1CB67C23996217EA43520BE3A7F
A4773A80A43CFE0620533A7AA4FE0028AB3A7AA4
FE0028A43D327AA4DD2183A4012200814FCDF997
3E06CD4597CD72941150C31B7AB320FBCD4898FE
6530F9473A7C", 11861
10040 DATA "A4B8DA1996011718DD212F9CCDF9
970CCDF9973E01327DA4C93E053280A4CD4597C3
7A95C5E5C1CD9297C1783276A4793275A4CD9297
C3B0953A7BA4FE00CAB6953D327BA4DD2183A401
2217814FCDF997CD729421204E2B7CB520FB3A75
A4C603CB3FCB", 12811
10050 DATA "3FCB3F4F3A76A4C603CB3FCB3FCB
3F4779FE10202178FE022808FE032804FE042014
3E02327DA4DD212F9C011700CDF9970CCDF997C9
79FE11201178FE032808FE042804FE052004160F
185879FE0E2804FE0F201178FE032808FE042804
FE0520041619", 9048
10060 DATA "183E78FE072804FE08201179FE0F
2808FE102804FE1120041605182479FE10200978
FE0520041614181678FE06C2B69579FE0F2809FE
102805FE11C2B6951614DD212F9CD5011700CDF9
970CCDF9971130751B7AB320FBD13A7CA4B79A30
063E02327DA4", 9458
10070 DATA "C9327CA4011700DD2183A4CDF997
0CDD2183A4CDF997C3B695DD216F9D116000CD22
98011203CDF9970CCDF9970D04CDF9970CCDF997
0D04CDF9970CCDF99757CB5A28070C79FEFF2001
0DCB5228040D20010CCB42280405200104CB4ACB
0478FEBED806", 11281
10080 DATA "BEC92108C078CB3FCB3FCB3F1150
00F5FE002804193D20FCF1CB27CB27CB275778B7
9A110008FE002804193D20FC060079CB39CB3909
E603DD2110A4111800FE002805DD193D20FB0608
DD7E00AE7723DD7E01AE7723DD7E02AE772B2BCD
26BCDD23DD23", 10045
10090 DATA "DD2310E2C92100C0C578FE002806
1150001910FDCB2109060811FF07DD7E007723DD
23DD7E0077DD231910F0C1C9FE00C8DD193D20FB
C9060516001E01DD2170A4DD7E00DD23CD1EBB28
037AB357CB2310EF7AC9DF4C98C94F98FCE5C5D5
DDE52A77A406", 11290

```

Listing 4

```

10 REM =====
==
20 REM          LISTING 4 - MCODE
30 REM =====
40 MEMORY 19999

```

```

10100 DATA "003A79A44F093005010040ED4222
77A4463A79A4A83279A4DDE1D1C1E1C9DDE5FDE1
011308DD214F9C115000FD7E00CD2298CDF9970C
CDF9970CCDF9970D0D04CDF9970C0CCDF9970D11
1000DD213F9DFD7E02CD2298CDF997040DD217F
9DCDF9970CDD", 12064
10110 DATA "216FA00CCDF9970D112000DD210F
A0FD7E04CD2298CDF9970C9DD4E00DD4601DD21FF
A1CDF997040CCDF9970CCDF9970CCDF9970CCDF9
970D0D0D041603D5CDF9970CCDF9970CCDF9970C
CDF9970CCDF9970D0D0D0D04D11520E3C9DD4601
DD4E000CDD21", 12287
10120 DATA "3FA3CDF9970CCDF9970CCDF9970D
OD0D041602D5CDF9970CCDF9970CCDF9970CCDF9
970CCDF9970D0D0D0D04D11520E3C9DF997031300DFC
1C6865616D6D6F3A6465667309310D061D6D6561
6D6D6F3A6465667309310D101D70726F7A656E74
3A6465667309", 10275
10130 DATA "310D1A1D666C61673A6465667309
310D241D61726D7761693A6465667309310D2E1D
61726D64656663A6465667309310D381D61726D6E
723A6465667309310D421D6465667309320D4C1D
6C6565723A646566730931360D166A72097A2C6A
6F79320D4416", 7371
10140 DATA "6C6409612C640D4E166F7209650D
58166C6409642C610D62166A6F79323A736C6109
650D6C16646A6E7A096A6F79310D76166C640961
2C640D80167265740D8A163B0D94163B20524E44
20526F7574696E650D9E163B0D8A16726E643A72
737409233138", 7417
10150 DATA "0DB21664656677097461626C650D
BC167265740DC6167461626C653A646566770972
6F7574696E0DD01664656662093235320DDA1672
6F7574696E3A7075736809686CODE41670757368
0962630DEE16707573680964650DF81670757368
0969780D0217", 8943
10160 DATA "6C6409686C2C28616C74686C290D
0C176C6409622C300D16176C6409612C28616C74
726E64290D20176C6409632C610D2A1761646409
686C2C62630D34176A72096E632C726E64320D3E
176C640962632C31363338340D48177362630968
6C2C62630D52", 6785
10170 DATA "17726E64323A6C640928616C7468
6C292C686C0D5C176C6409622C28686C290D6617
6C6409612C28616C74726E64290D7017786F7209
620D7A176C640928616C74726E64292C610D8417
706F700969780D8E17706F700964650D9817706F
700962630DA2", 7750
250 bu=0
260 GOSUB 1180:GOSUB 670
270 flag=PEEK(42109)
280 IF flag=0 THEN CLS:PEN 1:LOCATE 13,1
3:PRINT "UNENTSCHIEDEN!":LOCATE 9,20:PRI
NT"PRESS ANY KEY OR FIRE.":bu=bu+(6-PEEK
(40107)):WHILE INKEY#<>"":WEND:CALL &BB1
8:GOTO 260
290 IF flag=1 THEN 600
300 REM =====
310 REM          GEGNER GETROFFEN
320 REM =====
330 WINDOW #7,19,22,1,10:CLS #7
340 CALL dead1,17:FOR i=1 TO 500:NEXT:WI
NDOW #7,18,23,1,5:CLS #7
350 CALL dead1,17+512:FOR i=1 TO 500:NEX
T:CLS #7
360 CALL dead2,18+4*256:WHILE INKEY#<>""
:WEND:CALL &BB18
370 REM =====
380 REM          ABRECHUNG
390 REM =====
400 MODE 1:INK 0,26:BORDER 26:INK 1,0:IN
K 2,6:PEN 1:PAPER 0
410 PRINT "YOU EARNED:"
420 PRINT:PRINT"YOU PAID  :"
430 PRINT:PRINT"YOU HAVE  :"
440 PRINT:PRINT:PRINT"TOTAL      :"
450 LOCATE 19,1:PRINT "$."
460 LOCATE 19,3:PRINT "$ FOR BULLETS."
470 LOCATE 19,5:PRINT "$."
480 LOCATE 19,8:PRINT "$."
490 LOCATE 9,20:PRINT "PRESS ANY KEY OR
FIRE."
500 PLOT 192,312:DRAWR 128,0,1
510 bu=bu+6-PEEK(42107)
520 PEN 2:LOCATE 12,1:PRINT dol%(cow)
530 LOCATE 12,3:PRINT bu*10
540 LOCATE 12,5:PRINT sc
550 sc=sc+dol%(cow)-bu*10
560 LOCATE 12,8:PRINT sc
570 WHILE INKEY#<>"":WEND:CALL &BB18
580 cow=cow+1:IF cow=30 THEN cow=1:pr=pr
+25:IF pr>100 THEN pr=100
590 GOTO 240
600 REM =====
610 REM          SPIELER GETROFFEN
620 REM =====
630 WINDOW #7,16,23,21,25:CLS #7
640 WHILE INKEY#<>"":WEND:CALL &BB18
650 MODE 1:IF sc>=hsc THEN hsc=sc:h$=nn$
660 tot=cow:GOSUB 1310:GOTO 230
670 REM =====
680 REM          BILD AUFBAUEN
690 REM =====
700 MODE 1:PAPER 0:PEN 1:BORDER 12:inc=1
2:GOSUB 1950
710 CALL 20000
720 a=VAL (MID$(dat$(cow),1,1))-1
730 b=VAL (MID$(dat$(cow),2,1))-1
740 c=VAL (MID$(dat$(cow),3,1))-1
750 d=VAL (MID$(dat$(cow),4,1))-1
760 PEN 1:LOCATE 32,25:PRINT CHR$(255)
770 LOCATE 1,25:PRINT "REWARD:":PEN 2:P
RINT dol%(cow):PEN 1:PRINT "$"
780 LOCATE 1,24:PRINT n$(cow,1);" ";n$(c
ow,2)
790 INK 1,0:INK 0,12:INK 2,3:INK 3,6:farb%
(cow)
800 CALL mc,sp%(cow),prozent,a,b,c,d:RET
URN

```

Listing 5

```

100 REM =====
==
110 REM
120 REM          SHOWDOWN - LISTING 5
130 REM
140 REM =====
150 MEMORY 19999
160 cowdruck=&9876:mc=38000:dead1=&98D5:
dead2=&9917:bild=20000
170 ON BREAK GOSUB 810
180 DIM n$(29,2):DIM SP%(29):DIM dol%(29
):DIM farb%(29):DIM dat$(29):DIM t$(5):D
IM t$(5)
190 hsc=1000:h$="MATT DILLON"
200 GOSUB 1990
210 tot=29:nn$="MATT DILLON":sc=1000:GOS
UB 1310
220 GOSUB 1130
230 cow=1:prozent=pr:sc=0
240 GOSUB 1670

```

```

810 REM =====
820 REM          BREAK PATCH
830 REM =====
840 BORDER 0:MODE 1:INK 0,0:INK 1,23:INK
  2,26,15
850 PEN 1:PAPER 0
860 LOCATE 3,7:PRINT "S - STOP"
870 LOCATE 3,10:PRINT "D - Items neu def
  inieren"
880 LOCATE 3,13:PRINT "I - Alte Items wi
  ederherstellen"
890 LOCATE 3,16:PRINT "Z - Zurueck zum S
  piel"
900 i=0:WHILE i=0
910 IF NOT INKEY(60) THEN i=1
920 IF NOT INKEY(61) THEN i=2
930 IF NOT INKEY(35) THEN i=3
940 IF NOT INKEY(71) THEN i=4
950 WEND
960 IF i=1 THEN STOP
970 IF i=3 THEN GOSUB 1130:PEN 2:LOCATE
  5,20:PRINT "OK.":WHILE INKEY<>"":WEND:C
  ALL &BB18:GOTO 810
980 IF i=4 THEN INK 0,12:INK 1,0:INK 2,3
  :MODE 1:BORDER 12:GOSUB 1180:RETURN
990 WHILE INKEY<>"":WEND
1000 CLS
1010 LOCATE 1,3:PRINT "Bitte geben Sie n
  un die ";:PEN 2:PRINT"Tastencodes":PEN 1
  :PRINT"Ihrer gewuenschten Steuertasten.
  Sie finden diese im Benutzerhandbuch
  Anhang III, Seite 16."
1020 WINDOW 1,40,11,11:CLS
1030 FOR i=1 TO 5
1040 PRINT "Tastencode fuer ";:PEN 2:PRI
  NT t$(i);" ";:PEN 1:INPUT t%(i)
1050 IF t%(i)<0 OR t%(i)>79 THEN 1040 EL
  SE NEXT
1060 MODE 1:CLS:INPUT "Treffsicherheit (
  1-100%) ";pr
1070 IF pr<1 OR pr>100 OR INT(pr)<>pr TH
  EN 1060
1080 CLS:INPUT "Prozentabzug beim Schuss
  arm ";armr:IF armr<0 OR armr>100 THEN 10
  80
1090 CLS:INPUT "Prozentabzug beim andere
  n Arm ";arml:IF arml<0 OR arml>100 THEN
  1090
1100 CLS:INPUT "Prozentabzug Hose ";hose
  :IF hose<0 OR hose>100 THEN 1100
1110 CLS:INPUT "Prozentabzug Stiefel ";s
  t:IF st<0 OR st>100 THEN 1110
1120 prozent=pr:GOTO 810

1130 REM =====
1140 REM          URSPRUNGSWERTE SETZEN
1150 REM =====
1160 FOR i=1 TO 5:t$(i)=i+71:NEXT
1170 prozent=50:pr=50:st=5:armr=25:arml=
  15:hose=20
1180 REM =====
1190 REM          WERTE EINPOKEN
1200 REM =====

1210 FOR i=1 TO 5:POKE 42095+i,t$(i):NEX
  T:FOR i=0 TO 15:POKE i+42115,0:NEXT
1220 POKE &96AD,arml
1230 POKE &96C7,armr
1240 POKE &96E1,st
1250 POKE &96EF,hose
1260 POKE 42105,INT(RND*256)
1270 h1=INT(RND*15000)

1280 POKE 42103,(h1 MOD 256)
1290 POKE 42114,INT(h1/256)
1300 POKE 42101,216:POKE 42102,192:RETUR
  N
1310 REM =====
1320 REM          ZEITUNG ERSTELLEN
1330 REM =====
1340 inc=24:GOSUB 1950:BORDER 24:MODE 1:
  PAPER 3:PEN 1
1350 WINDOW 5,36,2,25:CLS
1360 LOCATE 5,2:PRINT "TOWN IN FEAR AND
  PANIC !"
1370 LOCATE 6,4:PRINT "MARSHAL
  OF"
1380 LOCATE 3,5:PRINT "TOMBSTONE CITY SH
  OT DOWN BY"
1390 LOCATE 6,6:PRINT "CRUEL, BRUTAL BAN
  DIT !"
1400 LOCATE 4,14:PRINT "MARSHAL
  LEAVES"
1410 LOCATE 5,15:PRINT "A WIFE, A COW AN
  D";:PEN 2:PRINT sc;:PEN 1:PRINT"$."
1420 LOCATE 2,20:PRINT "NEW MARSHAL WANT
  ED. GOOD WAGES"
1430 LOCATE 3,17:PRINT "BEST MARSHAL YET
  ";:PEN 2:PRINT H$:PEN 1
1440 LOCATE 8,18:PRINT "WHO EARNED";:PEN
  2:PRINT hsc;:PEN 1:PRINT"$."
1450 LOCATE 2,22:PRINT "ARE YOU BRAVE EN
  OUGH TO ACCEPT"
1460 LOCATE 2,23:PRINT "THIS DANGEROUS J
  OB (";:PEN 2:PRINT"YEAH";:PEN 1:PRINT"/"
  ;:PEN 2:PRINT"NDP";:PEN 1:PRINT")?";
1470 PEN 2:LOCATE 14,4:PRINT nn$:LOCATE
  12,14:PRINT nn$
1480 t$=n$(tot,1)+" "+n$(tot,2)
1490 LOCATE 16-LEN(t$)/2,12:PRINT t$
1500 PLOT 64,384:DRAWR 512,0:DRAWR 0,-38
  4:DRAWR -512,0:DRAWR 0,384
1510 FOR i=0 TO 4:PLOT 128,348-i:DRAWR 3
  84,0:NEXT
1520 PLOT 300,274:DRAWR 54,0,1:DRAWR 0,-
  54,1:DRAWR -54,0,1:DRAWR 0,54,1
1530 a=VAL(MID$(dat$(tot),1,1))
1540 b=VAL(MID$(dat$(tot),2,1))
1550 c=VAL(MID$(dat$(tot),3,1))
1560 CALL cowdruck,a-1,b-1,c-1
1570 WINDOW 1,40,1,25
1580 INK 0,24:INK 1,0:INK 2,6:INK 3,26
1590 GOSUB 2680
1600 i$="":WHILE i$="":i$=INKEY$:WEND
1610 IF i$="n" OR i$="N" THEN MODE 1:PAP
  ER 0:PEN 1:END ELSE IF i$<>"Y" AND i$<>"
  y" THEN 1600
1620 CLS:PEN 1:BORDER 26
1630 LINE INPUT "TELL ME YOUR NAME MARSH
  AL ?",nn$
1640 IF LEN(nn$)>11 THEN LOCATE 1,3:PRIN
  T "Nur 11 Buchstaben.":LOCATE 1,1:PRINT
  SPACE$(80):LOCATE 1,1:GOTO 1630
1650 IF LEN(nn$)=0 THEN nn$="XXX"
1660 nn$=UPPER$(nn$):RETURN
1670 REM =====
1680 REM          STECKBRIEF ANZEIGEN
1690 REM =====
1700 inc=24:GOSUB 1950:BORDER 24:MODE 1:
  PAPER 0:PEN 1
1710 FOR i=0 TO 400 STEP 35:PLOT 16,i:DR
  AW 639,i,1:NEXT
1720 WINDOW #1,16,26,5,18:PAPER #1,3:PEN
  #1,1:CLS #1

```

```

1730 LOCATE #1,3,12:PRINT #1,"REWARD:":P
RINT #1," #";dol%(cow)
1740 a=VAL (MID$(dat$(cow),1,1))
1750 b=VAL (MID$(dat$(cow),2,1))
1760 c=VAL (MID$(dat$(cow),3,1))
1770 CALL cowdruck,a-1,b-1,c-1
1780 LOCATE #1,6-LEN(n$(cow,1))/2,9:PRIN
T #1,n$(cow,1)
1790 LOCATE #1,6-LEN(n$(cow,2))/2,10:PRI
NT #1,n$(cow,2)
1800 PLOT 300,274:DRAWR 54,0,1:DRAWR 0,-
54,1:DRAWR -54,0,1:DRAWR 0,54,1
1810 PLOT 240,336:DRAWR 176,0,1:DRAWR 0,
-224,1:DRAWR -176,0,1:DRAWR 0,224,1
1820 RESTORE 1840:FOR i=1 TO 4:READ x,y:
PLOT x,y:DRAWR 4,0:DRAWR 2,-2:DRAWR 0,-4
:DRAWR -2,-2:DRAWR -4,0:DRAWR -2,2:DRAWR
0,4:DRAWR 2,2
1830 NEXT i
1840 DATA 246,332,406,332,246,124,406,12
4
1850 ORIGIN 256,316
1860 PLOT 0,0:DRAWR 8,-32:DRAWR 8,16:DRA
WR 8,-16:DRAWR 8,32
1870 PLOT 32,-32:DRAWR 8,32:DRAWR 4,-16:
DRAWR -4,0:DRAWR 4,0:DRAWR 4,-16
1880 PLOT 56,-32:DRAWR 0,32:DRAWR 16,-32
:DRAWR 0,32
1890 PLOT 88,-32:DRAWR 0,32:DRAWR -8,0:D
RAWR 0,-4:DRAWR 0,4:DRAWR 16,0:DRAWR 0,-
4
1900 PLOT 120,-28:DRAWR 0,-4:DRAWR -16,0
:DRAWR 0,16:DRAWR 8,0:DRAWR 0,4:DRAWR 0,
-8:DRAWR 0,4:DRAWR -8,0:DRAWR 0,16:DRAWR
16,0:DRAWR 0,-4
1910 PLOT 128,-32:DRAWR 0,32:DRAWR 12,0:
DRAWR 4,-4:DRAWR 0,-24:DRAWR -4,-4:DRAWR
-12,0
1920 LOCATE 11,24:PRINT "PRESS ANY KEY O
R FIRE"
1930 INK 0,24:INK 1,0:INK 2,6:INK 3,26
1940 WHILE INKEY$<>"":WEND:CALL &BB18:RE
TURN
1950 REM =====
1960 REM          INK's LOESCHEN
1970 REM =====
1980 FOR i=0 TO 3:INK i,inc:NEXT:RETURN
1990 REM =====
2000 REM  DATEN IN ARRAY's EINLESEN
2010 REM =====
2020 RESTORE 1990
2030 FOR i=1 TO 29:READ n$(i,1):READ N$(
I,2):NEXT
2040 FOR i=1 TO 29:READ SP$(i):READ dol%
(i):READ farb$(i):READ dat$(i):NEXT
2050 FOR i=1 TO 5:READ T$(i):NEXT
2060 FOR i=0 TO 15:POKE i+42113,0:NEXT
2070 SYMBOL 255,24,24,24,255,255,24,24,2
4
2080 RETURN
2090 DATA DRUNKEN,HARRY
2100 DATA ROY,RANDOM
2110 DATA PSYCHO,THE I.
2120 DATA IAN,HEUTEUFEL
2130 DATA PIZZA,PASTA
2140 DATA PETROL,DOLLAR
2150 DATA HANK,HANDICAP
2160 DATA BRUND,BRUTAL
2170 DATA PINK,PAT
2180 DATA RONALD,RAYGUN
2190 DATA PATRICK,PACKEIS
2200 DATA JUMPING,JACK
2210 DATA CRAZY,CARL
2220 DATA GLASS,JOE
2230 DATA SLIME,SAM
2240 DATA TOM,STONE
2250 DATA SAD,BARNY
2260 DATA GUN,GLIDDER
2270 DATA SAD,SANCHO
2280 DATA DOC,PHIL
2290 DATA UGLY,ADAM
2300 DATA HUNTER,MASTER
2310 DATA FAST,EDDIE
2320 DATA DARE,DAVE
2330 DATA MAD,MAX
2340 DATA GUN,SLINGER
2350 DATA MARC,MCGUN
2360 DATA DON,DIABOLO
2370 DATA BILLY,THE KID
2380 DATA 255,15,11,"1121"
2390 DATA 246,25,2,"2111"
2400 DATA 238,50,15,"1121"
2410 DATA 229,75,10,"3111"
2420 DATA 221,105,18,"1122"
2430 DATA 212,175,12,"3112"
2440 DATA 204,200,0,"2112"
2450 DATA 195,250,27,"3111"
2460 DATA 187,0,8,"2122"
2470 DATA 178,300,7,"1221"
2480 DATA 170,450,11,"2231"
2490 DATA 161,500,18,"2232"
2500 DATA 153,550,9,"1232"
2510 DATA 144,620,14,"2231"
2520 DATA 136,650,12,"2232"
2530 DATA 127,700,22,"1222"
2540 DATA 119,780,0,"1222"
2550 DATA 110,830,26,"1221"
2560 DATA 102,900,10,"1212"
2570 DATA 93,970,2,"1222"
2580 DATA 84,1100,21,"3311"
2590 DATA 76,1180,20,"3322"
2600 DATA 67,200,26,"3311"
2610 DATA 59,1230,13,"3312"
2620 DATA 50,1500,25,"3311"
2630 DATA 42,1700,10,"3332"
2640 DATA 35,1730,0,"2332"
2650 DATA 30,1890,26,"2311"
2660 DATA 25,2000,0,"2332"
2670 DATA hoch,runter,links,rechts,Feuer
2680 REM =====
2690 REM          SOUND
2700 REM =====
2710 ENV 1,10,-1,2
2720 ENT 1,5,1,1,10,-1,1,10,1,1,5,-1,1
2730 RESTORE 2740: FOR a=1 TO 38:READ b,
c:SOUND 1,c/2^3,b*12.5,0,1,1:SOUND 2,c/2
^-1,b*12.5,15:SOUND 4,c/2^2,b*12.5,0,1,1
:NEXT
2740 DATA 2,568,4,478,2,426,3,379,1,358,
2,379,4,426,2,506,3,638,1,568,2,506,4,47
8,2,568,3,568,1,602,2,568,4,506,2,602,4,
758,2,568
2750 DATA 4,478,2,426,3,379,1,358,2,379,
4,426,2,506,3,638,1,526,2,506,3,478,1,50
6,2,568,3,606,1,677,2,638,5,568,4,568,0,
0
2760 RETURN
9970 REM =====
9980 REM          SAVEROUTINE
9990 REM =====
10000 SAVE "SHOWBAS"

```

Lottoglück mit dem Schneider CPC

Läuft nur mit Diskettenlaufwerk

464

664

6128

Das Programm LOTTO.BAS erlaubt die Erstellung und Gewinnkontrolle jeweils eines Service-Scheines des Spiels „6 aus 49“ und „Spiel 77“ für mehrere Wochen.

Ein Service-Schein der Lottogesellschaften erlaubt es, nur einen Tippschein mit bis zu acht Tippzeilen abzugeben, dessen Zahlen maximal fünf Wochen Gültigkeit haben, einschließlich „Spiel 77“. Die Bearbeitungsgebühr wird nur einmal erhoben. Kleingewinne werden von den Gesellschaften akkumuliert und am Ende der Laufzeit zusammen ausgezahlt. Großgewinne bleiben davon unberührt. Mit dem vorliegenden für die Samstagziehung geschriebenen Programm kann man sich auf komfortable Weise den Tippzettel per Zufallsgenerator erstellen. Die erzeugten Zahlen werden auf Diskette gespeichert und stehen daher dem Programm nach der Ziehung zur Gewinnkontrolle wieder zur Verfügung.

Nach dem Abtippen des Programms und der Beseitigung des SYNTAX-TERRORs sehen Sie bitte mit CAT nach, ob sich in das Inhaltsverzeichnis Ihrer Diskette Dateien mit den Namen LOTTOSVC.DAT oder LOTTOSVC.BAK eingeschlichen haben. Da sich in ihnen, bedingt durch das Entwanzen, Unsinn befinden könnte, sollten Sie sie killen. Am bequemsten mit dem Programm KILLER.BAS, ansonsten mit der Befehlsfolge: A\$="LOTTOSVC.*": ERA, A\$.

CPC 664 und 6128 Besitzer müssen in Zeile 970 den Befehl IRESET (im Listing :RESET) weglassen.

Starten Sie das Programm nun mit RUN. Es bricht mit der Fehlermeldung „Datei nicht gefunden“ ab. Das ist richtig so. Es zeigt Ihnen, daß Ihr Rechner den richtigen Dateinamen in der richtigen Variablen hat. Geben Sie nun einfach ohne vorher etwas anderes zu tun das Kommando: GOTO 920 ein. Auf der Diskette wird nun eine sequentielle Leerdatei mit dem Inhalt (nicht der Länge!) null eingerichtet. Dieser Vorgang muß nur einmal ablaufen, wenn das Programm neu installiert wird, und natürlich auch dann, wenn Sie es auf eine andere Diskette kopieren.

Danach ist Ihr Werk syntaktisch richtig. Sie sollten jetzt, bevor Sie ernsthaft tippen, erst einige Kontroll- und Probelaufe machen, um auch die richtige Funktion zu prüfen.

So arbeitet das Programm

Nach dem Start bittet Sie der Rechner um die Anzahl der gewünschten Tippzeilen (1 bis 8; bei 0 erfolgt ENDE), um die Anzahl der Wochen (1 bis 5) und um die „Spiel 77“ Zahl Ihres Tippscheines. Hier können Sie mit ENTER antworten, wenn Sie das Spiel nicht mitmachen. In diesem Falle wird bei der Gewinnkontrolle die für „Spiel 77“ gezogene Zahl nicht abgefragt. Sonst verlangt der Rechner eine genaue 7-stellige Eingabe. Nach einer kurzen Pause von ca. drei Sekunden bei acht Zeilen werden Sie gefragt, ob die Ausgabe auf dem Drucker oder dem Bildschirm erfolgen soll. Nur die Buchstaben D oder B (groß oder klein) werden an-

genommen. Wenn Ihr Drucker bei Eingabe von D unklar oder nicht angeschlossen ist, bleibt das Programm hängen. Wenn Sie den Drucker in Ordnung bringen, geht es weiter, ansonsten ist ein Neustart notwendig. Die vom Rechner getippten Zahlen erscheinen der Größe nach sortiert in Kolonnen untereinander, jede Kolonne ist dabei eine Tippreihe. Nach dem Speichern der Werte setzt sich der Rechner mit READY zur Ruhe und Sie können die Werte auf Ihren Tippzettel übertragen.

Beim erneuten Start verlangt der Rechner von Ihnen die Eingabe der gezogenen Zahlen, der Zusatzzahl, und falls gespielt, der Gewinnzahl aus „Spiel 77“. Auf der rechten Seite erscheinen dann Ihre Tippzeilen und eventuell die 77-Zahl Ihres Lottoscheines. Links unten werden die Treffer pro Reihe aufgelistet, wenn Sie Glück haben mit dem Vermerk „+ Zusatzzahl“. Rechts daneben sehen Sie die Richtigen des „Spiel 77“. Darunter steht, für wieviel Spiele Ihr Schein noch gültig ist. Nach der letzten Ziehung werden Sie gefragt, ob Sie einen neuen Schein tippen wollen. Sie können mit J(a) oder N(ein) antworten. Bei N wird wieder eine 0-Datei abgespeichert. Bei allen Fragen, bei denen eine mögliche Antwort in spitzen Klammern angegeben wird, ist diese Antwort bereits voreingestellt. Sie wird auch bei ENTER oder SPACE usw. vom Rechner übernommen.

Aus Vorstehendem ergibt sich, daß Sie beim ernsthaften Tippen, also wenn es um Geld geht, das Programm nicht unter der Woche nur mal so „zum Spaß“ aufrufen dürfen. Der Rechner versteht keinen Spaß und zieht Ihnen prompt eine Woche Ihrer Spielzeit ab. Ist es doch einmal passiert, so sollten Sie mit 2*ESC abrechnen, ohne Zahlen eingegeben zu haben. Bei der ersten Prüfung der Funktionen achten Sie bitte darauf, daß beim Tippen alle Zahlen von 1 bis 49 erscheinen. Bei der Gewinnkontrolle muß der Rechner alle Treffer in jeder Reihe finden, auch mit Zusatzzahl und „Spiel 77“.

Zum Programm

Bei der Erstellung des Tippzettels arbeitet der Rechner nach der Methode „Ziehen aus geordneter Menge“, wodurch Doppelziehungen vermieden werden. Für jede Reihe wird ein Vektor mit den Zahlen von 1 bis 49 erstellt, der in jeder Zeile die eigene Nummer enthält. Die durch den Zufallsgenerator errechnete Zahl gilt als Zellennummer. Der Inhalt dieser Zeile wird als Tipp gespeichert und das dadurch entstandene „Loch“ im Vektor mit dem Inhalt der jeweils letzten Zelle gestopft. Anschließend wird die Obergrenze des Zufallsgenerators um 1 dekrementiert, so daß die vorher letzte Zelle nicht mehr „drankommt“. Die gezogenen Zahlen werden gleich nach der Ziehung der Größe nach eingeordnet. Es handelt sich hier um ein Verfahren ähnlich Bubblesort.

Die Gewinnkontrolle erfolgt durch einfaches lineares Suchen und Mitzählen. Bei den kleinen Zahlenmengen sind aufwendigere Verfahren ineffektiv. Die Zahlen des „Spiel 77“ werden als String gespeichert und bei der Kontrolle rückwärts gelesen und verglichen. Richtige Ziffern werden

gezählt. Bei der ersten falschen Ziffer wird die Schleife durch Nullsetzen der Laufvariablen abgebrochen.

Interessant ist noch die Variable zr\$ (zr = zurück). Sie enthält nur vier Bytes mit den Werten 4, 11, 11 und 13, also nur Steuercodes mit den auf den Seiten K9/S2-4 des Handbuches beschriebenen Funktionen. Wenn diese Variable geprintet wird, so ertönt ein Piepser und der Cursor rutscht um zwei Zeilen nach oben auf den Zeilenanfang, vorausgesetzt, der Printbefehl wurde mit einem Semikolon abgeschlossen. Andernfalls resultiert daraus nur ein Hochsetzen um eine Zeile. Mit dieser einfachen Konstruktion kann man die lästige Wiederholung des Ausdrucks auf dem Schirm verhindern, wenn eine Frage falsch beantwortet wurde und wiederholt werden muß. Die falsche Antwort bleibt unter einem durchscheinenden Cursor sichtbar.

Das Programm läuft bei mir seit einigen Monaten zur vollsten Zufriedenheit. Ich bin aber immer noch nicht Millionär geworden.

Prof. W. Tosberg

```

10 'LOTTO.BAS ----->
   Programm-Name auf Diskette
20 'LOTTOSVC.DAT -----> Zugehoeriger Dat
   einame
30 '----- Vorbereitung -----

40 MODE 2: CLEAR : DEFINT a-z: DIM tip(8
,6),v(49)
50 zr$=CHR$(7)+CHR$(11)+CHR$(11)+CHR$(13
)
60 RANDOMIZE TIME
70 PRINT"Zahlenlotto - Service-Schein"
80 PRINT STRING$(11,"*"): PRINT: A$="L
OTTOSVC.DAT"
90 OPENIN ""+A$
100 INPUT#9,w,r,sp77$
110 CLOSEIN
120 IF w THEN 450
130 '----- Neu Tippen -----

140 INPUT "Wieviel Reihen (8) : ",r: IF
r=0 THEN 980
150 IF r>8 THEN SOUND 1,90:PRINT zr$: GO
TO 140
160 INPUT"Wieviel Wochen (5) : ",w
170 IF w=0 THEN SOUND 1,90:PRINT zr$:GOT
O 160
180 IF w>5 THEN SOUND 1,90:PRINT zr$: GO
TO 160
190 INPUT "Spiel 77 : ",sp77$:
PRINT
200 IF sp77$="" THEN 220
210 IF LEN(sp77$)<>7 THEN SOUND 1,90: PR
INT zr$: GOTO 190
220 PRINT "Bitte warten.": PRINT
230 FOR i=1 TO r
240 FOR j=1 TO 49: v(j)=j: NEXT j: e=49
250 FOR j=1 TO 6: h=INT(e*RND+1)
260 tip(i,j)=v(h):v(h)=v(e):e=e-1: NEXT
j
270 FOR j=1 TO 5: FOR k=j+1 TO 6
280 IF tip(i,j)<=tip(i,k) THEN 300
290 h=tip(i,j):tip(i,j)=tip(i,k):tip(i,k
)=h
300 NEXT k,j
310 NEXT i:PRINT
320 '----- Ausdruck -----

330 PRINT"Ausdruck auf Drucker oder B) i
dschirm ? ";
340 h$=UPPER$(INKEY$): IF h$="" THEN 340
350 h=INSTR(1,"BD",h$)-1
360 IF h<0 THEN PRINT b$: GOTO 340
370 PRINT h$:PRINT: IF h THEN h=8
380 FOR j=1 TO 6: FOR i=1 TO r
390 PRINT#h,USING "#####":tip(i,j);
400 NEXT i: PRINT#h: NEXT j

```

```

410 PRINT:PRINT"Allles okay ? <J> N ";
420 h$=UPPER$(INKEY$): IF h$="" THEN 420
430 IF h$="N" THEN 880 ELSE 910
440 '----- Tip-Kontrolle -----

450 PRINT"Bitte die gezogenen Zahlen:"
460 FOR i=1 TO 6:PRINT USING "##."): ";i
;:INPUT "",g(i)
470 NEXT i
480 PRINT: INPUT "Zusatzzahl : ",zsz
490 IF sp77$="" THEN PRINT:PRINT: GOTO 5
20
500 INPUT "Spiel 77 : ",sp$: PRINT
510 IF LEN(sp$)<>7 THEN SOUND 1,90:PRINT
zr$: GOTO 500
520 PRINT"Allles richtig ? <J> N"
530 h$=UPPER$(INKEY$): IF h$="" THEN 530
540 IF h$="N" THEN LOCATE 1,4:PRINT CHR$
(20);:GOTO 450
550 LOCATE 1,14
560 OPENIN ""+A$
570 INPUT#9,w,r,sp77$
580 FOR i=1 TO r: FOR j=1 TO 6: INPUT#9,
tip(i,j)
590 NEXT j,i
600 CLOSEIN
610 WINDOW#1,30,80,5,17
620 FOR j=1 TO 6: FOR i=1 TO r
630 PRINT#1,USING"#####":tip(i,j);
640 NEXT i:PRINT#1:NEXT j
650 LOCATE#1,4,9:PRINT#1,sp77$
660 PRINT: FOR i=1 TO r: q=0
670 FOR j=1 TO 6
680 FOR k=1 TO 6: IF tip(i,j)=g(k) THEN
q=q+1
690 NEXT k,j :PRINT "Richtig in Reihe";i
: " : ";q;
700 IF q<5 THEN PRINT: GOTO 740
710 y=0: FOR j=1 TO 6: IF zsz=tip(i,j) T
HEN y=1:j=6
720 NEXT j
730 IF y THEN PRINT "+ Zusatzzahl" ELSE
PRINT
740 NEXT i
750 WINDOW#2,43,80,15,20
760 IF sp77$="" THEN 830
770 q=0
780 FOR i=7 TO 1 STEP -1
790 IF MID$(sp77$,i,1)=MID$(sp$,i,1) THE
N q=q+1 ELSE i=0
800 NEXT i
810 PRINT#2,"Spiel 77: ";q;" richtig.":P
RINT#2: w=w-1
820 IF w>0 THEN PRINT#2,"Noch";w;" mal Z
iehung.": GOTO 910
830 PRINT#2: w=w-1: IF w>0 THEN PRINT#2,
"Noch";w;" mal Ziehung.": GOTO 910
840 PRINT#2,"Letzte Ziehung !":PRINT#2,"
Neues Spiel? <J> N "
850 h$=UPPER$(INKEY$): IF h$="" THEN 850
860 IF h$="N" THEN 880 ELSE q=1000
870 '----- Tip loeschen -----

880 w=0: r=0: sp77$=""
890 FOR i=1 TO 8: FOR j=1 TO 6: tip(i,j)
=0: NEXT j,i
900 '----- Abspeichern -----

910 !ERA,@a$
920 OPENOUT ""+A$
930 WRITE#9,w,r,sp77$
940 FOR i=1 TO 8: FOR j=1 TO 6: WRITE#9,
tip(i,j)
950 NEXT j,i
960 CLOSEOUT
970 !RESET : IF q=1000 THEN RUN
980 PRINT
990 END
1000 '*****

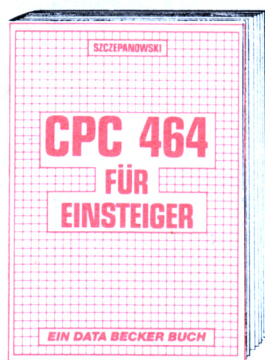
1010 ' Prof. Walter Tosberg, 4900 Herfor
d,Lessingstr.1
1020 ' 25.07.85 Letzte Aende
rung: 07.10.85
1030 '*****

```

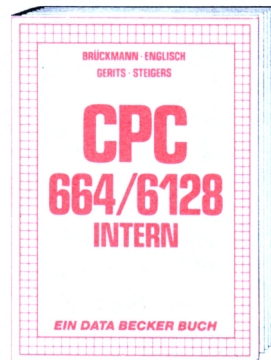
DATA BECKER Hits zu



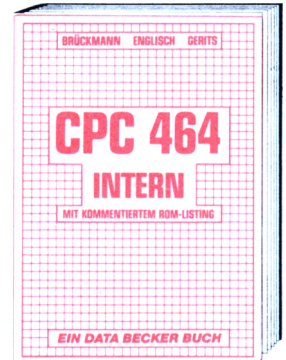
Das sollte Ihr erstes Buch zum CPC-6128 sein! CPC-6128 für Einsteiger ist eine sehr leicht verständliche Einführung in Handhabung und Einsatz des CPC-6128, die keinerlei Vorkenntnisse voraussetzt. Dazu eine Einführung in BASIC, wobei Sie eine komplette Adressverwaltung erstellen, die Sie anschließend nutzen können. Unentbehrlich für jeden CPC-6128 Einsteiger!
CPC-6128 für Einsteiger, 215 Seiten, DM 29,-



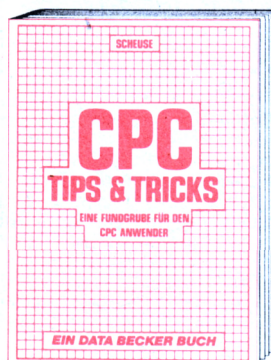
Wer sich für den Schneider CPC 464 entschieden hat, der findet hier den optimalen Einstieg! Neben den wichtigsten Hinweisen zu Handhabung und Anschlußmöglichkeiten werden Sie schrittweise in die Programmierung eingeführt. Zahlreiche Abbildungen und Bildschirmfotos ergänzen den Text. Das ideale Buch für jeden, der mit dem CPC 464 das Computern beginnen will!
CPC 464 für Einsteiger, 206 Seiten, DM 29,-



Ein Muß für jeden, der sich professionell mit dem CPC 664 oder dem CPC 6128 beschäftigt. Einführung in das System, den Prozessor, das Gate Array, den Video-Controller, den Schnittstellenbaustein 8255, den Soundchip, die Schnittstellen. Mit Disassembler und ausführlichen Kommentaren zu den Routinen von Interpreter und Betriebssystem. Ein Superbuch, wie alle Titel der INTERN-Reihe!
CPC 664/6128 INTERN, 456 Seiten, DM 69,-



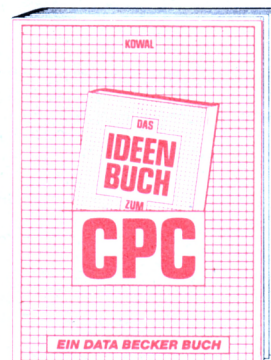
Wirklich alle Geheimnisse des CPC 464 liefert dieses Standardwerk! Neben dem kommentierten BASIC-ROM-Listing enthält es Kapitel zu Speicheraufteilung, Prozessor, Besonderheiten des Z80, Gate Array, Video-Controller und Video-Ram, Soundchip, Schnittstellen, Betriebssystem, Routinenutzung, Character-Generator, u.v.m. Für den fortgeschrittenen BASIC-Programmierer unentbehrlich, für den Assembler-Programmierer ein absolutes Muß!
CPC 464 Intern mit kommentiertem ROM-Listing, 548 Seiten, DM 69,-



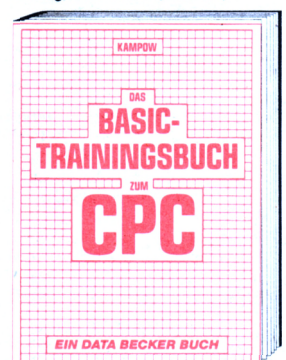
Rund um den CPC 464 viele Anregungen und wichtige Hilfen! Von Hardwareaufbau, Betriebssystem, BASIC-Tokens, Anwendungen der Windowtechnik und sehr vielen interessanten Programmen bis zu einer umfangreichen Dateiverwaltung, Soundeditor, komfortablem Zeichengenerator und kompletten Listings spannender Spiele bietet dieses Buch eine Fülle von Möglichkeiten. Diese Tips kommen von den DATA BECKER Spezialisten!
CPC 464 Tips & Tricks, 271 Seiten, DM 39,-



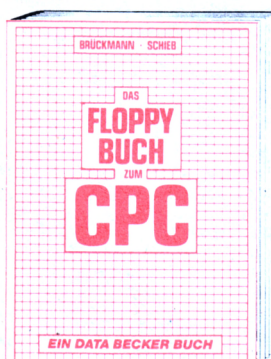
Tips & Tricks für alle CPC Benutzer! Menügenerator, Maskengenerator, BASIC-Befehlsweiterungen, Programmierhilfen wie Dump, BASIC-Zeile von BASIC aus erzeugen, wichtige Systemroutinen und deren Nutzung, Beschleunigung von Programmen u. v. m.
CPC Tips and Tricks Band II, ca. 250 Seiten, DM 39,-



Nützliche und pfiffige Ideen rund um Freizeit und Alltag! Denn auch hier ist der CPC ein kleines Genie. Das zeigen Programme mit Beschreibung und Beispielen zu: Lotto – Benzinverbrauch – Geld und Kredit – Schreiben und Verwalten – Staat und Steuer – Haushaltsorganisation – Stricken, Fußball, Blumenpflege – Kinder und Schule – u.v.m. Dazu im Anhang wichtige BASIC-Anweisungen und Fehlermeldungen!
Das Ideenbuch zum CPC 464, 664 & 6128, 294 Seiten, DM 39,-



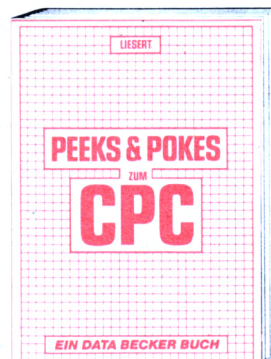
CPC 464 BASIC? Kein Problem! Mit diesem Trainingsbuch lernen Sie von Grund auf nicht nur die einzelnen Befehle und ihre Anwendungen, sondern auch einen richtig sauberen Programmierstil. Von der Problemanalyse über den Datenflußplan bis zum fertigen Programm. Dazu viele Übungsaufgaben mit Lösungen und zahlreichen Beispielen. Schlichtweg unentbehrlich!
Das BASIC-Trainingsbuch zum CPC 464, 285 Seiten, DM 39,-



Alles über Floppyprogrammierung vom Einsteiger bis zum Profi. Natürlich mit ausführlichem ROM-Listing, einer äußerst komfortablen Dateiverwaltung, einem hilfreichen Disk-Manager. Dazu eine Fundgrube verschiedener Programme und Hilfsroutinen, die das Buch für jeden Floppy-Anwender zur Pflichtlektüre machen!
Das Floppybuch zum CPC, 353 Seiten, DM 49,-



Endlich CP/M beherrschen! Von grundsätzlichen Erklärungen zu Speicherung von Zahlen, Schreibschutz oder ASCII, Schnittstellen und Anwendung von CP/M-Hilfsprogrammen. Für Fortgeschrittene: Fremde Diskettenformate lesen, Erstellen von Submit-Dateien u.v.m. Dieses Buch berücksichtigt die Versionen CP/M 2.2 und 3.0 für Schneider 464, 664 und 6128.
Das CP/M-Trainingsbuch zum CPC, 260 Seiten, DM 49,-



Wer PEEKS und POKES zum CPC 464 kennen und anwenden will, der findet hier umfassende Information! Sie reicht vom Adressbereich des Prozessors über Betriebssystem und Interpreter bis hin zur Einführung in die Maschinensprache. Dazu Programmierhilfen, Routinen sowie reichlich Material zu den Themen Grafikfunktionen, Massenspeicherung und Peripherie, Tricks und Formeln in BASIC und RAM-Pages!
Peeks & Pokes zum CPC, 180 Seiten, DM 29,-



Von den Grundlagen der Maschinenspracheprogrammierung über die Arbeitsweise des Z80-Prozessors und einer genauen Beschreibung seiner Befehle bis zur Benutzung von Systemroutinen ist alles ausführlich und mit vielen Beispielen erklärt. Im Buch enthalten sind Assembler, Disassembler, Einzelschritt-Simulator und Monitor als komplette Anwenderprogramme. So wird der Einstieg in die Maschinensprache leichtgemacht!
Das Maschinensprachebuch zum CPC ca. 300 Seiten, DM 39,-

Aus Deutschlands größtem Verlag für Computerbücher

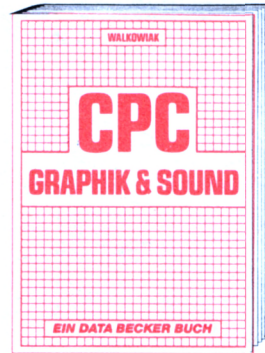
CPC 464, 664 & 6128



Speziell für den Hobbyelektroniker, der mehr aus seinem CPC machen möchte! Von nützlichen Tips zur Platinenherstellung über Adreßdecodierung, Adapterkarten und Interfaces bis zu EPROM-Programmierboard und -Programmierzettel oder Motorsteuerung für Gleich- und Schritt-schaltmotoren werden machbare Erweiterungen ausführlich und praxisnah beschrieben. Am besten gleich anfangen! **CPC Hardware-Erweiterungen, 445 Seiten, DM 49,-**



Das Superbuch zum Z80 Prozessor! Systemarchitektur, Pinbeschreibung, Register, Befehlsausführung, Flags, CPU-Software, Anschluß von Systembausteinen, serielle/parallele Datenübertragung, Zähler/Timerbaustein Z80-CTC und Befehlsatz. Alles ausführlich beschrieben und mit vielen Abbildungen! Als Lehrbuch und Nachschlagewerk für jeden Maschinenspracheprogrammierer unentbehrlich! **Das Prozessorbuch zum Z80, 560 Seiten, DM 59,-**



Nutzen Sie die außergewöhnlichen Grafik- und Soundmöglichkeiten des CPC 464! Natürlich mit vielen interessanten Beispielen und Programmen: Grafikgrundlagen, Sprites, Shapes, Strings, mehrfarbige Darstellungen, Koordinationstransformation, Verschiebungen, Drehungen, Rotation, 3-D-Funktionsplotter, CAD, Synthesizer, Miniorgel, Hüllkurven u.v.m. Dieses Buch wird Sie begeistern! **CPC Graphik & Sound, 220 Seiten, DM 39,-**



DFÜ für Jedermann mit dem CPC bietet eine ausführliche und verständliche Einführung in das Gebiet der Datenfernübertragung: was ist DFÜ, BTX, DATEX, Mailbox, alles über Modems und Koppler. Begriffs-erklärung: Originate, Answer, Half-Duplex usw., eine serielle Schnittstelle am CPC, RS-232C/V.24 simuliert, Mailboxsoftware – selbstgestrickt, Postbestimmungen u. v. m. Steigen Sie mit diesem Buch in die Welt der Datenetze und Datenfernübertragung ein! **DFÜ für Jedermann zum CPC, über 250 Seiten, DM 39,- (Erscheint ca. Dez.)**

Textomat

Deutschlands meistgekauft Textverarbeitung bietet Profileistung zum Hobbypreis! TEXTOMAT in Stichworten:
– Diskettenprogramm durchgehend menuegesteuert – deutscher/amerikanischer Zeichensatz – Rechenfunktionen für alle Grundrechenarten – über 17.000 Zeichen pro Text im Speicher – beliebig lange Texte durch Verknüpfung – 80 Zeichen pro Zeile – läuft mit ein oder zwei Floppys – 27 Farben für Rahmen-Hintergrund-Bildschirmfarbe – es können Trennvorschläge gemacht werden – Word-wrap – Tabulatoren – Seitenummerierung – Proportional-schrift auf entsprechendem Drucker – Zuweisungstabelle für ASCII-Code – frei definierbare Steuerzeichen, z.B. für Indices, Schriftarten, Unterstreichen, Formate – umfangreiche Formularanpassungen – Blockoperationen, „Suchen und Ersetzen“ – Serienbriefherstellung mit DATAMAT – formatierte Ausgabe auf dem Bildschirm – Anpassung an fast jeden Drucker – ausführliches Handbuch mit Übungslektion – **Komplett nur DM 148,- für CPC 464, 664 und 6128, die richtige Version wird automatisch geladen.**

Textomat Plus

Neues Textverarbeitungsprogramm der Superlative. Erheblich erweiterte, leistungsstärkere TEXTOMAT-Version. Bietet alle Möglichkeiten von TEXTOMAT und zusätzlich:
+ ergonomische, schreibmaschinenähnliche Texteingabe arbeitet grundsätzlich im 80 Zeichenmodus + 2 dynamisch verwaltete Textbereiche im Speicher. Zwischen beiden Texten kann beliebig hin- und hergeschaltet sowie kopiert werden. Wahlweise Menuesteuerung oder schnelle Direktwahl der Funktionen. 10 Floskelkasten für häufig wiederkehrende Worte oder Redewendungen. Sehr komfortable Cursorsteuerung (vor/zurück – Zeichen/Wort/Satz/Absatz) + Trennvorschläge nach deutscher Grammatik + Kopf- und Fußzeilen während des Textes änderbar + bedingter Seitenwechsel + BASIC-Programme können eingelesen, editiert und abgespeichert werden, dabei automatisch ASCII Um- und Rückwandlung + Suchen und Ersetzen mit vielen Optionen und Joker (vor/rückwärts – Klein-/Großschreibung – ganze Wörter) + komplettes Terminalprogramm zum problemlosen Senden und Empfangen von Texten sowohl zum Halb- als auch Voll duplexbetrieb.
TEXTOMAT PLUS für CPC 6128 kostet DM 198,- Lieferbar ab ca. Dezember

Datamat

Deutschlands meistgekauft Dateiverwaltung bietet eines, was in dieser Preisklasse bisher unvorstellbar schien:
– menuegesteuertes Diskettenprogramm, dadurch extrem einfach zu bedienen – für jede Art von Daten – völlig frei

gestaltbare Eingabemaske – 80 Zeichen pro Zeile – Hardcopy – 50 Felder pro Datensatz – 512 Zeichen pro Datensatz – bis zu 4000 Datensätze pro Datei je nach Umfang – 27 Farben für Rand, Hintergrund und Buchstaben – Schnittstelle zu TEXTOMAT – Benutzung von Rechenfeldern – Anzeige des Disketteninhaltes – läuft mit ein oder zwei Floppys – komplett in Maschinensprache, dadurch extrem schnell – deutscher/amerikanischer Zeichensatz – fast jeder Drucker ist anschließbar – duplizieren der Datendiskette – gute Benutzerführung – Hauptprogramm komplett im Speicher – kein lästiges Nachladen – deutsches Handbuch mit Übungslexikon – Sie können: jeden Datensatz in wenigen Sekunden suchen – nach beliebigen Feldern selektieren – nach allen Feldern, auf- oder absteigend sortieren – Listen in völlig freiem Format drucken – Etiketten drucken.
Komplett nur DM 148,-. Für CPC 464, 664 und 6128, die richtige Version wird automatisch geladen.

Profimat CPC

Zur Programmierung in Maschinensprache benötigt man einen Assembler. Doch Assembler ist nicht gleich Assembler. Deshalb gibt es PROFIMAT nun auch für die SCHNEIDER-Rechner. Durch den integrierten Editor wird das Arbeiten mit PROFIMAT zum Vergnügen. Verketteten von Quelltexten für besonders lange Assemblerprogramme ist selbstverständlich möglich. PROFIMAT für den SCHNEIDER ist aber mehr als nur ein Assembler, er ist gleichzeitig auch Monitor. Der absolute Clou dieses Assemblers ist die Möglichkeit, die frisch assemblierten Programme im TRACE-Modus (Einzelschritt-) laufen zu lassen und so jede Änderung an den CPU-Registern verfolgen zu können. PROFIMAT ist frei verschiebbar und kann somit nie in Konflikt mit Ihren eigenen Maschinenprogrammen kommen. Einfache Handhabung durch den komfortablen Editor auch für Anfänger garantiert. Selbstverständlich „beherrscht“ der Assembler auch die sogenannten Pseudo-Ops, die bedingtes Assemblieren möglich machen.
PROFIMAT CPC DM 99,-. Lieferbar für SCHNEIDER CPC 464, 664 und 6128

Mathemat CPC

MATHEMAT CPC ist ein unentbehrliches Hilfsmittel für Schule, Studium und Beruf. Mit MATHEMAT CPC erhalten Sie die Möglichkeit, Probleme der Algebra, Geometrie und der Kurvendiskussion besser und schneller zu lösen. Zudem können Sie mit dem eingebauten Taschenrechner Aufgaben aus den Gebieten Primzahl-, Prozent- oder auch der Dreisatzrechnung spielend lösen. Zu jedem Zeitpunkt besteht die Möglichkeit eine Hardcopy auf einem Drucker auszugeben.
Für CPC 464, 664, 6128. MATHEMAT CPC DM 99,-

Profi-Painter CPC

PROFI PAINTER, ein sensationelles Programm zum Malen, Entwerfen und Zeichnen auf CPC Computern. Den berühmten Vorbildern der 32-bit-Welt steht PROFI PAINTER kaum nach und übertrifft diese sogar in manchen Punkten. Zum Erstellen, Korrigieren, Sichern und Drucken von Grafiken. **PROFI PAINTER** beinhaltet eine **grafikorientierte Benutzerschnittstelle**. Direkt am Bildschirm und mit Hilfe von deutlichen Symbolen kann jeder anfangen, Bilder, Grafiken oder technische Zeichnungen zu erstellen. Mit dem Joystick bewegen Sie den Zeiger an jede beliebige Stelle auf dem Bildschirm, ein Klick und die Funktion wird ausgeführt. Folgende Hilfsmittel stehen zur Verfügung:
– der **Bleistift**, mit dem Sie feine Linien zeichnen oder löschen
– der **Pinself**, in verschiedenen Größen und Formen, mit dem Sie malen
– die **Sprühdose**, mit der Sie Graffiti erstellen
– der **Farbeimer**, mit dem Sie beliebige Flächen ausfüllen
– der **Radiergummi**, mit dem Sie bestimmte Stellen wieder löschen
– das **Lineal**, mit dem Sie beliebige Linienzüge zeichnen
– das **Rechteck**, mit abgerundeten Ecken
– der **Kreis/Ellipse**
– die **Polygone**
– das **Lasso**, mit dem Sie Bildausschnitte einfangen können
– der **Markierungsrahmen**, um Bildschirmbereiche zwecks weiterer Bearbeitung zu definieren
– die **Textmarke**, ab der Sie Text eingeben können
– aus einem der fünf Zeichensätze
– in einer der drei Zeichengrößen
– in einer der fünf Schriftarten
Folgende Optionen können Sie anwählen:
– den **Vergrößerungsmodus**
– das **Ganze Seite Zeigen**
– vorhandene **Muster umdefinieren**
Das professionelle deutsche Spitzenprogramm, komplett mit ausführlichem Handbuch, für CPC 464, 664 und 6128. PROFI PAINTER CPC DM 198,-.

BESTELL-COUPON
Einsenden an: DATA BECKER · Merowingerstr. 30 · 4000 Düsseldorf 1
Zzgl. DM 5,- Versandkosten
 per Nachnahme Verrechnungsscheck liegt bei
Name und Adresse bitte deutlich schreiben

DATA BECKER

Merowingerstr. 30 · 4000 Düsseldorf · Tel. (02 11) 31 00 10

CPC-Orgel

Damit machen Sie aus Ihrem Schneider-Computer eine Heimorgel



Mit diesem Programm für den 464, 664 und 6128 kann jeder aus dem Computer eine Heimorgel mit fast allen Schikanen machen. Nach dem Start des Programms wird ein Bild aufgebaut, das in drei Abschnitten folgendes anzeigt:

1. Klaviaturbelegung der Tasten mit Noten und Sonderfunktionen (Tongag und Schlagzeugbreaks).
2. Belegung der numerischen Sondertasten mit Ton-, Klang- und Rhythmusvariationen und der Anzeige der aktuellen Einstellung.
3. Form der vordefinierten Hüllkurven für Lautstärke und Ton als grafische Kurvendarstellung.

Nun kann mit den numerischen Sondertasten die gewünschte Klang-/Rhythmuskombination eingestellt werden. Diese beginnt nach der Einstellung sofort mit dem Abspielen der Begleitung. Auf den oberen beiden Tastenreihen kann man nun die Melodie spielen. Die Sonderfunktion Tongag und Schlagzeugbreak lassen sich durch Tastendruck an der gewünschten Stelle in der Darbietung plazieren, wobei während des Tongags der Rhythmus weiterläuft. Die Unterbrechung des Programmes ist mit ESC möglich, sollte aber nur bei ausgeschaltetem Tonrepeat versucht werden, da andernfalls weitere Eingaben nur mit erheblichen Schwierigkeiten möglich sind.

Einstellungen

(0) Tonhöhe

Hier wird die Lage der gesamten Klaviatur geändert. Jeder Tastendruck erhöht die Tonlage der Klaviatur um eine Oktave. Ist Oktave 4 eingestellt, führt ein weiterer Tastendruck zur Oktave 1.

(1)-(4) Hüllkurven

Für den Melodie- und den Oberton können hier getrennt die Ton- und Lautstärkenhüllkurven eingestellt werden. Dies ermöglicht enorm viele Klangvariationen im Melodiespiel.

(5) Rhythmus

Es kann zwischen normalem Rhythmus (2-, 3- und 4-Viertel-Takt) und einigen Rhythmusvariationen gewählt werden, die in dieser Fassung alle auf dem 4-Viertel-Takt aufgebaut sind.

(6) Tempo

Regelt die Geschwindigkeit der eingestellten Rhythmusvariante in Zweierschritten zwischen 16 und 40.

(7) Rhythmus-Lautstärke

Regelt die Lautstärke der Begleitung. Da die Melodie immer mit festeingestellter Lautstärke spielt, kann damit das Verhältnis zwischen Melodie und Begleitung festgelegt werden.

(8) Tondauer

Hier wird die maximale Tondauer zwischen 30 und 80 Hundertstel Sekunden variiert. Das Drücken einer neuen Tontaste führt jedoch zum Abbruch des eben gespielten Tones.

Variablenliste

oct	= aktuelle Melodieoktave
ev	= Melodielautstärkenkurve
evo	= Obertonlautstärkenkurve
Rhythmus	= eingestellte Begleitung
Td	= Tondauer
h	= Höhe des Melodietones
l	= Rhythmuslautstärke
et	= Melodietonkurve
eto	= Oberton-Tonkurve
p	= Taktlänge
repeat	= Tonwiederholung

(9) Tonrepeat

Ist dieser Modus eingestellt, wird bis zum Loslassen der gedrückten Tontaste der Ton wiederholt.

Zum Programmaufbau

Hüllkurven: In den Zeilen 1040-1100 werden die Ton- und Lautstärkehüllkurven definiert. Wer mit diesen angebotenen Klangvariationen nicht zufrieden ist, kann diese Definitionen abändern oder auch zusätzliche Kurven festlegen. Dabei ist aber zu bedenken, daß die Grafik der Hüllkurven eventuell ebenfalls geändert werden sollte. Werden mehr als 8 Kurven definiert, muß dies in den Zeilen 1270-1310 in den Variablen et, ev, eto und evo berücksichtigt werden.

Melodiespiel: In der Zeile 1000 ist das Abspielen des Melodietones festgelegt. In der vorliegenden Fassung wird der gespielte Ton durch einen Oberton ergänzt, der eine Oktave höher liegt. Änderungen in dieser Zeile können im wesentlichen nur den Oberton betreffen:

SOUND 130,h/1.5,Td,0,evo,eto führt z.B. zum Abspielen des Obertones in der Quinte statt Oktave.

SOUND 130,h/2,Td/2,0,evo,eto limitiert die Länge des Obertones auf die Hälfte des Grundtones.

Außerdem kann der Oberton natürlich vollkommen weggelassen werden.

Rhythmus: Es kann sein, daß die vorgegebenen Rhythmen von manchem Organisten als nicht ausreichend angesehen werden. Die Zeilen 530, 550, 570 und 590 sind die Einsprungadressen für die Definition der Rhythmen 5-8. Da die Tonschlangen des CPC 464 jeweils 5 Töne aufnehmen können, wurden jeweils 5 Töne pro Takt festgelegt, wobei die Zeilen jeweils nach einer Taktlänge Timer-gesteuert angesprungen werden. Die o.g. Zeilen können natürlich den eigenen Vorstellungen entsprechend abgeändert werden. Auch ein Aufteilen eines Taktes in zwei mal fünf Töne erscheint möglich, ist aber nicht ganz unkompliziert.

Schlagzeugbreaks: In den Zeilen 1140-1230 werden die Breaks abgearbeitet. Auch diese kann man ändern. Weiterhin lassen sich natürlich auch zusätzliche Breaks einarbeiten. Dazu ist lediglich in den Zeilen 640-980 eine weitere Tastenabfrage mit Einsprungadresse etwa wie in den Zeilen 650-670 notwendig.

Beim Melodiespiel entspricht die Belegung der oberen beiden Tastenreihen dem Aufbau einer normalen Orgelklaviatur, wobei die oberste Reihe die "schwarzen" Tasten, also die Tasten mit den Halbtonschritten trägt. Die Spannweite der Klaviatur liegt dabei von g bis zu dem etwa 1 1/2 Oktaven höheren d und ist damit für die meisten Musikstücke ausreichend. Wer schon mehrfach mit Tasteninstrumenten gespielt hat, wird sich rasch an die CPC-Klaviatur gewöhnen.

Wolfgang Klement



DM 1000

für das Schneider CPC Topprogramm

Angesprochen sind alle Schneider-CPC-Programmierer, denn ab der nächsten Ausgabe gibt es das Toplisting im CPC-Magazin. Und so läuft es: Beteiligen können sich alle, die für einen Schneider-CPC Programme schreiben. Schickt diese Programme auf einem geeigneten Datenträger samt Beschreibung und Listing an die Redaktion. Ein Originallisting muß nicht unbedingt beiliegen. Wer noch keinen Drucker hat, kann auch nur den Datenträger und eine gut lesbare Beschreibung einsenden. Die Redaktion wählt dann das beste Programm zum Topprogramm. Autoren, deren Programme zwar nicht prämiert aber trotzdem veröffentlicht werden, erhalten je nach Qualität und Umfang des Programmes DM 100.- bis DM 500.- Honorar.

So Leute, nun ran an die Tasten Eurer Keyboards und los geht's. Die Chancen, veröffentlicht zu werden oder gar zu gewinnen, stehen gut.

Schickt Eure Programme an das

CPC-MAGAZIN, Verlag Rätz-Eberle, Abteilung Toplisting, Postfach 1640, D-7518 Bretten

CPC-Orgel

```

10 REM *****
*****
**
20 REM ***          C P C -
  O R G E L          ***
30 REM ***          -----
                      ***
40 REM ***          Copyright
  1985 by            ***
50 REM ***          Wolfgang
  Klement           ***
60 REM ***          ***

70 REM *****
*****
80 REM
90 REM >-----Bildschi
rmaufbau-----
100 REM
110 MODE 2:PRINT:PRINT"
*> C P C - O R G E L <*:PRINT:PRINT
120 PRINT" 1 2 3 4 5 6 7
  B 9 0 - ↑":PRINT
130 PRINT"      g i s b      c i s d i s
f i s g i s b      c i s      -Noten":PRINT
140 PRINT:PRINT"  Q  W  E  R  T
Y  U  I  O  P  @  [ ":PRINT
150 PRINT"  g  a  h  c  d  e  f
  g  a  h  c  d      -Noten":PRINT
160 FOR i=16 TO 368 STEP 32:MOVE i,312:D
RAWR 24,0:DRAWR 0,32:DRAWR -24,0
170 DRAWR 0,-32:NEXT:FOR i=24 TO 376 STE
P 32:MOVE i,232:DRAWR 24,0:DRAWR 0,32
180 DRAWR -24,0:DRAWR 0,-32:NEXT:PRINT"
(b)= Break 1 / (n)= Break 2";
190 PRINT" / (m)= Break 3 / (g)= Gag
":MOVE 1,399:DRAWR 638,0:DRAWR 0,-230
200 DRAWR -638,0:DRAWR 0,230:SPEED KEY 1
0,3:WINDOW#1,1,80,16,25:CLS#1
210 MOVE 0,366:DRAW 639,366:MOVE 0,200:D
RAW 639,200:BORDER 21:INK 0,15:INK 1,0
220 REM
230 REM >-----Parameter
-Anzeige-----
240 REM
250 PRINT#1,"(0)= Tonhoehe (Oktave) ....
..... 3";
260 PRINT#1,"          HUELLKURVEN:"
270 PRINT#1,"(1)= Lautstaerkenhuellkurve
Melodie ..... 1";
280 PRINT#1,"  Lautstaerke:  Ton:"

290 PRINT#1,"(2)= Lautstaerkenhuellkurve
Oberton ..... 1"
300 PRINT#1,"(3)= Tonhuellkurve
Melodie ..... 1"
310 PRINT#1,"(4)= Tonhuellkurve
Oberton ..... 1"
320 PRINT#1,"(5)= Rhythmus (2-4/4-Takt,5
-8 Rhythmen,1=aus) ... 1"
330 PRINT#1,"(6)= Tempo (16-40) .....
..... 30"

340 PRINT#1,"(7)= Rhythmuslautstaerke ..
..... 6"
350 PRINT#1,"(8)= Tondauer .....
..... 50"
360 PRINT#1,"(9)= Tonrepeat (0=aus,1=ein
) ..... 0"
370 oct=3:ev=1:et=1:evo=1:eto=1:Rhythmus
=1:p=30:l=6:Td=50:repeat=0:GOSUB 1040
380 KEY DEF 13,0,201:KEY DEF 14,0,202:KE
Y DEF 5,0,203:KEY DEF 15,0,200
390 KEY DEF 20,0,204:KEY DEF 12,0,205:KE
Y DEF 4,0,206:SPEED KEY 50,2
400 KEY DEF 10,0,207:KEY DEF 11,0,208:KE
Y DEF 3,0,209:GOSUB 1430
410 REM
420 REM >-----Rhythmus
initialisieren-----
430 REM
440 xx=REMAIN(1):xx=REMAIN(2):IF Rhythmu
s=1 THEN 640
450 IF Rhythmus<5 THEN EVERY p,2 GOSUB 5
10:EVERY Rhythmus*p,1 GOSUB 520
460 IF Rhythmus=5 THEN EVERY 2*p,1 GOSUB
530
470 IF Rhythmus=6 THEN EVERY 2*p,1 GOSUB
550
480 IF Rhythmus=7 THEN EVERY 2*p,1 GOSUB
570
490 IF Rhythmus=8 THEN EVERY 2*p,1 GOSUB
590
500 GOTO 640
510 SOUND 4,0,5,1,0,0,1:RETURN
520 SOUND 4,3822,15,0,9:RETURN
530 SOUND 132,3822,p,0,9:SOUND 4,0,p,0,9
,0,1:SOUND 4,3822,p/2,0,9
540 SOUND 4,2551,p/2,0,9:SOUND 4,0,p,0,9
,0,25:RETURN
550 SOUND 132,3822,p,0,9:SOUND 4,0,p/2,0
,9,0,31:SOUND 4,0,p/2,0,9,0,1
560 SOUND 4,3034,p,0,9:SOUND 4,2551,p,0,
9:RETURN
570 SOUND 132,3822,p,0,9:SOUND 4,0,p/2,0
,9,0,1:SOUND 4,0,p/2,0,9,0,1
580 SOUND 4,2551,p,0,9:SOUND 4,0,p,0,9,0
,31:RETURN
590 SOUND 132,3822,p,0,9:SOUND 4,0,p,0,9
,0,1:SOUND 4,2551,p,0,9
600 SOUND 4,0,p/2,0,9,0,31:SOUND 4,0,p/2
,0,1,0,15:RETURN
610 REM
620 REM >-----Tastatur
abfragen (Notenwerte+Aenderungen)-----
630 REM
640 a$=INKEY$:IF a$="" THEN 640
650 IF a$="b" THEN GOTO 1140
660 IF a$="m" THEN GOTO 1180
670 IF a$="n" THEN GOTO 1150
680 IF a$=CHR$(200) THEN oct=oct+1:GOSUB
1270
690 IF a$="g" THEN GOSUB 1220
700 IF a$=CHR$(201) THEN ev=ev+1:GOSUB 1
270
710 IF a$=CHR$(202) THEN evo=evo+1:GOSUB
1270
720 IF a$=CHR$(203) THEN et=et+1:GOSUB 1
270
730 IF a$=CHR$(204) THEN eto=eto+1:GOSUB
1270
740 IF a$=CHR$(205) THEN rhythmus=rhythm
us+1:GOSUB 1270

```

```

750 IF a$=CHR$(206) THEN p=p+2:GOSUB 127
0
760 IF a$=CHR$(207) THEN l=l+1:GOSUB 127
0
770 IF a$=CHR$(208) THEN Td=Td+5:GOSUB 1
270
780 IF a$=CHR$(209) THEN repeat=repeat+1
:GOSUB 1270
790 IF a$="q" THEN h=638/oct
800 IF a$="2" THEN h=602/oct
810 IF a$="w" THEN h=568/oct
820 IF a$="3" THEN h=536/oct
830 IF a$="e" THEN h=506/oct
840 IF a$="r" THEN h=478/oct
850 IF a$="5" THEN h=451/oct
860 IF a$="t" THEN h=426/oct
870 IF a$="6" THEN h=402/oct
880 IF a$="y" THEN h=379/oct
890 IF a$="u" THEN h=358/oct
900 IF a$="8" THEN h=338/oct
910 IF a$="i" THEN h=319/oct
920 IF a$="9" THEN h=301/oct
930 IF a$="o" THEN h=284/oct
940 IF a$="0" THEN h=268/oct
950 IF a$="p" THEN h=253/oct
960 IF a$="@" THEN h=239/oct
970 IF a$="^" THEN h=225/oct
980 IF a$="[" THEN h=213/oct
990 IF ander=1 THEN ander=0:GOSUB 1040:G
OTO 440
1000 SOUND 129,h,Td,0,ev,et:SOUND 130,h/
2,Td,0,evo,eto:GOTO 640
1010 REM
1020 REM >-----Huellkur
ven generieren-----
1030 REM
1040 ENV 1,1,12,1,12,-1,5:ENV 2,6,2,1,4,
-1,3,10,0,1,8,-1,5:ENV 3,6,2,1,5,-2,8
1050 ENV 4,12,1,1,15,0,1,5,-2,8:ENV 5,12
,1,3,12,-1,3:ENV 6,12,1,5
1060 ENV 7,12,1,1,6,-1,1,20,0,1,4,1,1,8,
-1,4:ENV 9,1,2,1,5,0,1,1,-2,2
1070 ENV 8,12,1,1,10,-1,2,8,1,1,6,-1,2,4
,1,8:ENT -1,2,1,2,4,-1,2,2,1,2
1080 ENT -2,4,1,1,8,-1,1,4,1,1:ENT -3,2,
1,4,4,-1,4,2,1,4
1090 ENT -4,4,1,2,8,-1,2,4,1,2:ENT -5,8,
1,1,16,-1,1,8,1,1:ENT 6,4,2,1,8,-1,2
1100 ENT -7,4,1,1,8,-1,1,12,1,1,16,-1,1,
8,1,1:ENT 8,4,-4,1,16,1,1:RETURN
1110 REM
1120 REM >-----Abarbei
tung der Breaks-----
1130 REM
1140 x=REMAIN(1):x=REMAIN(2):FOR i=1 TO
4:SOUND 4,3822,55,2,6,0,1:NEXT:GOTO 440
1150 x=REMAIN(1):x=REMAIN(2):FOR i=1 TO
4:SOUND 4,0,10,1,0,0,31
1160 SOUND 4,0,5,1,0,0,1:SOUND 4,0,15,0:
NEXT:SOUND 4,3822,50,0,6,0,15
1170 SOUND 4,3822,40,0,1,0,31:GOTO 440
1180 x=REMAIN(1):x=REMAIN(2)
1190 FOR i=1 TO 21 STEP 5:SOUND 4,0,5,1,
0,0,i:SOUND 4,0,2,0:SOUND 4,0,5,1,0,0,i
1200 SOUND 4,0,16,0:NEXT:FOR i=1 TO 16:S
OUND 4,0,3,1,0,0,31:SOUND 4,0,2,0:NEXT
1210 FOR i=24 TO 1 STEP -1:SOUND 4,0,3,1
,0,0,i:NEXT:GOTO 440
1220 FOR k=1 TO 9:FOR n=60 TO 119 STEP I
NT(RND(1)*11)+3:SOUND 1,n,2,12:NEXT
1230 NEXT:SOUND 1,119,0,0,4:RETURN
1240 REM
1250 REM >-----Abarbeit
ung der Aenderungen-----
1260 REM
1270 IF et=9 THEN et=1
1280 IF oct=5 THEN oct=1
1290 IF ev=9 THEN ev=1
1300 IF eto=9 THEN eto=1
1310 IF evo=9 THEN evo=1
1320 IF rhythmus=9 THEN rhythmus=1
1330 IF p>40 THEN p=16
1340 IF l=8 THEN l=3
1350 IF Td>80 THEN Td=20
1360 IF repeat=2 THEN repeat=0:Td=50:SP
EED KEY 50,2
1370 LOCATE 50,17:PRINT ev:LOCATE 50,18:
PRINT evo:LOCATE 50,19:PRINT et
1380 LOCATE 50,20:PRINT eto:LOCATE 50,21
:PRINT Rhythmus:LOCATE 50,22:PRINT p
1390 IF repeat=1 THEN Td=15:SPEED KEY 1,
1
1400 LOCATE 50,23:PRINT l:LOCATE 50,24:P
RINT Td:LOCATE 50,25:PRINT repeat
1410 LOCATE 50,16:PRINT oct:PLOT 420,1:D
RAW 420,165:PLOT 421,1:DRAW 421,165
1420 ander=1:RETURN
1430 REM
1440 REM >-----Huellkurv
engraphik-----
1450 REM
1460 PLOT 420,1:DRAW 420,165:PLOT 421,1:
DRAW 421,165
1470 FOR i=18 TO 25:LOCATE 67,i:PRINT i-
17:NEXT
1480 PLOT 530,1:DRAW 450,1:DRAW 470,10:D
RAW 490,3:DRAW 505,7:DRAW 515,3
1490 DRAW 530,5:PLOT 530,17:DRAW 450,17:
DRAW 470,27:DRAW 480,20:DRAW 505,20
1500 DRAW 515,25:DRAW 530,17:PLOT 530,33
:DRAW 450,33:DRAW 530,45
1510 PLOT 530,49:DRAW 450,49:DRAW 490,61
:DRAW 530,49
1520 PLOT 530,65:DRAW 450,65:DRAW 470,77
:DRAW 490,77:DRAW 530,68
1530 PLOT 530,81:DRAW 450,81:DRAW 470,93
:DRAW 530,84
1540 PLOT 530,97:DRAW 450,97:DRAW 462,10
9:DRAW 470,105:DRAW 500,105:DRAW 530,97
1550 PLOT 530,113:DRAW 450,113:DRAW 458,
125:DRAW 530,113
1560 MOVE 550,120:FOR i=550 TO 630:DRAW
i,120+3*SIN((i-550)/3):NEXT i
1570 MOVE 550,104:FOR i=550 TO 630:DRAW
i,104+6*SIN((i-550)/3):NEXT i
1580 MOVE 550,88:FOR i=550 TO 630:DRAW i
,88+3*SIN((i-550)/6):NEXT i
1590 MOVE 550,72:FOR i=550 TO 630:DRAW i
,72+6*SIN((i-550)/6):NEXT i
1600 MOVE 550,56:FOR i=550 TO 630:DRAW i
,56+12*SIN((i-550)/6):NEXT i
1610 PLOT 630,35:DRAW 550,35:DRAW 565,45
:DRAW 580,35:PLOT 634,24:DRAW 550,24
1620 DRAW 556,28:DRAW 568,20:DRAW 586,30
:DRAW 610,18:DRAW 628,28:DRAW 634,24
1630 PLOT 630,14:DRAW 550,14:DRAW 562,1:
DRAW 590,14:RETURN
1640 REM
1650 REM >-----

```

Leserfragen – Leserfragen – Leserfragen

Kann ich mit meinem Schneider IBM-Disketten lesen und dessen Programme benutzen?

Da beim IBM immer nur 5 1/4" Disketten verwendet werden, ist das Lesen von IBM-Disketten auch nur mit einem Laufwerk dieser Größe möglich. Außerdem ist es angebracht, ein Doppelkopflaufwerk zu nehmen, da dies beim IBM ebenfalls so ist. Es würde sich also ein Vortexlaufwerk anbieten. Allerdings ist der IBM ein 16 Bit Computer, so daß Programme nur bedingt verwendet werden können. Es gibt für den IBM aber Karten, mit denen er unter CP/M 2.2 arbeiten kann. Demzufolge werden auch irgendwo IBM Disketten mit Programmen unter CP/M 2.2 vorhanden sein. Diese Programme laufen nach Installation auf dem CPC, allerdings nur mit vermindertem Speicherplatz. Hier würde eine Vortexspeichererweiterung Abhilfe schaffen. Ob sich der Aufwand jedoch lohnt, muß jeder selbst entscheiden.

Oft finden sich in Zeitungen Programme, die RSX-Befehle einrichten. Ist es möglich, mehrere dieser Befehlsweiterungen gleichzeitig zu benutzen?

Es ist leider oft nicht möglich, RSX-Kommandos aus verschiedenen Programmen gleichzeitig zu verwenden, da sich die Speicherbereiche oft überlappen. Meistens sind diese Programme oberhalb der Adresse 40000 angelegt und lassen sich nur äußerst schwer verschieben. Das CPC-Magazin bringt aber ab sofort eine Serie mit

RSX-Befehlen. Diese besteht aus 3-4 Teilen, von denen jeder 10-20 neue Basicbefehle einrichtet. Diese 3-4 Teile können dann alle zusammen aber auch jeder für sich betrieben werden. Die Erweiterungen sollen auf allen Schneider Computern (bis auf den JOYCE) mit und ohne Floppy laufen.

Was hat es mit dem Befehl mod auf sich?

$a \text{ mod } b$ berechnet den Rest beim Teilen von a durch b . $1000 \text{ mod } 33$ ergibt z. B. 10 ($1000/33$ ist 30 mit Rest 10).

Wie legt man Dateien an (Openin, Openout)?

Um Dateien auf Band oder Disc zu schreiben, muß man sie zuerst mit OPENOUT "Name" eröffnen. Dann kann man mit PRINT #9, <Variable> Strings, Real- und Integerzahlen auf das angeschlossene Speichermedium speichern und mit CLOSEOUT muß die Datei nach Übernahme aller Werte geschlossen werden.

Beim Lesen beginnt man mit OPENIN "Name". Dies hat zur Folge, daß eine Datei mit diesem Namen in einen Pufferspeicher geladen wird. Mit INPUT #9, <Variable> kann man die Variablen wieder auslesen und zwar in der gleichen Reihenfolge, wie sie abgespeichert wurden. Mit CLOSEOUT wird die Datei zum Lesen geschlossen. Es kann immer nur eine Datei zum Lesen und eine Datei zum Schreiben eröffnet werden.

Wie legt man RSX-Befehle an?

Die Beantwortung dieser Frage ist ziemlich umfangreich. Wir verweisen deshalb auf folgende Literatur:

1. CPC-Magazin Nov./Dez. 85 unter der Rubrik Assembler-tips.
2. CPC 464 Intern von DATA BECKER.

Bei den CPC-Computern 664 + 6128 gibt es einige neue Befehle. Manche davon, wie z. B. FRAME oder GCHAR, lassen sich auf dem CPC 464 durch CALLs erreichen. Können auch die anderen Befehle (z. B. FILL) auf dem CPC 464 durch CALLs aufgerufen werden?

Der CPC 6128 und der CPC 664 haben ein völlig neues ROM erhalten. Dieses enthält viele zusätzliche Befehle, davon kann man allerdings nur sehr wenige durch CALLs auf dem CPC 464 ersetzen. Den Befehl FILL gibt es auf dem CPC 464 keinen auf jeden Fall.

Leserforum

Wer sich intensiv mit seinem Computer beschäftigt, kennt bestimmt die Situation: Ein Problem ist aufgetaucht, das Handbuch gibt keine Auskunft und der Freak aus dem Freundeskreis hat einen Commodore. Kurz gesagt, es fehlt ein Retter in der Not.

Hier soll in Zukunft das Leserforum des CPC-Magazins Abhilfe schaffen. Unsere Schneider-Spezialisten stehen für Sie bereit, um alle auftauchenden Fragen schnell und präzise zu beantworten. Ob es um Schwierigkeiten bei der Programmierung oder um Hardwareprobleme geht, niemand braucht mehr zu

verzweifeln, denn es gibt immer jemanden, der weiterhelfen kann. Anfragen, die nicht in unserer Redaktion direkt beantwortet werden können oder deren Inhalt für viele CPC-User von Interesse ist, sollen auf dieser Seite veröffentlicht werden.

Wer also Fragen gleich welcher Art hat, kann diese ab sofort aufschreiben und zusammen mit einem frankierten und adressierten Rückumschlag an uns einschicken. Für eine schnelle Erledigung werden wir sorgen.

Unsere Anschrift:
CPC-Magazin
Postfach 1640
7518 Bretten

User-Clubs

Schneider Computer-Club Saar

Dieser Club für saarländische User hat zur Zeit etwa 25 Mitglieder. Darunter sind auch erfahrene EDV-Fachleute. Die Mitglieder treffen sich jeweils zweiwöchentlich in Saarbrücken. Info durch:

Schneider-Computerclub Saar
Michael Mathieu
Im Fuchstälchen
6600 Saarbrücken 11
Tel: 06 81 / 3 02 20 98

CPC 464 Club Lahr

Dieser Club besteht seit ca. 3 Monaten und hat schon viele Mitglieder. Der Clubbeitrag be-

trägt pro halbes Jahr 10 DM. Es gibt regelmäßig eine Clubzeitschrift für alle Mitglieder und außerdem kostengünstige Software- und Hardwareangebote für den CPC 464. Jedes neue Mitglied ist willkommen.

Frank Forst
Im Münchtal 31/1
7630 Lahr

Wir wollen einen Club gründen!

Ich suche Kontakt zu Usern aus dem Raum Zürich/Schweiz, um einen Club zu gründen, Alter möglichst 14-20 Jahre. Ruft an unter der Nummer 01 / 2 52 03 31 (abends).

Bendix Freutel
Dolderstraße 9
CH-8032 Zürich

Ihr direkter
Draht zur
Redaktion:
☎ 07252/42948

Amstrad-Schneider User-Club-Aachen

Nun ist es soweit: Wir haben den A.S.U.C.A. gegründet und suchen weitere Mitglieder, nicht nur aus dem Rheinland. Wir wollen mit minimalem Aufwand (Beitrag) und viel Engagement neue Wege gehen.

Das sind unsere Schwerpunkte:

- reger Erfahrungsaustausch
- Starthilfen für Einsteiger und System-Umsteiger
- Erstellung einer regelmäßig erscheinenden Clubzeitschrift mit vielen Tips + Tricks + Listings
- Aufbau einer Club-Programmbibliothek
- preisgünstige Beschaffung von Hard- und Software durch "Sammelbestellungen"

Schreibt (bitte nur mit frankiertem Rückumschlag) an unsere Kontaktadresse:

ASUCA
Richard Cloots
Zeißstraße 7
5132 Übach-Palenberg

Ortenauer Computer Club

Unser Club, der jetzt ca. 5 Monate alt ist, hat schon 40 Mitglieder. Teilweise gibt es aber auch Abteilungen für andere Computer. Wir machen regionale wöchentliche Clubabende, die in der Presse angekündigt werden. Wer sich für unsere Clubarbeit interessiert, kann nähere Unterlagen anfordern.

Ortenauer Computer Club
Wolfgang A. Jaeger
Postfach 1731
7600 Offenburg

Neuer Berliner Schneider Club

Wir suchen noch Mitglieder, auch Mädchen, egal welches Alter. Also Anfänger, Profis, Hacker meldet euch. Wir tauschen auch Informationen aus.

Berliner User Club
Wolfgang Windorpski
Gritznerstraße 38
1000 Berlin 41

Kontakt gesucht!

Ich suche Kontakt zu einem User-Club im Raum Hessen (wenn möglich in Frankfurt).

Roman Diehl
In den Biegen 19
6000 Frankfurt 56
Tel. 0 61 01 / 4 12 83

Ich will in der Region Darmstadt/Dieburg einen Schneider Club gründen und suche noch Leute, die da mitmachen wollen.

Ralf Willner
Breslauer Straße 38
6114 Groß-Umstadt
Tel. 0 60 78 / 44 46

Für die Städte Ludwigshafen, Mannheim und Umgebung suche ich Leute, die mit mir einen Club gründen.

Gerhard Scholz
Friedensstraße 107
6700 Ludwigshafen

Spiele-Ecke

Ob Action oder Adventure, es gibt immer wieder Programme, die einem zum Verzweifeln bringen. Sei es eine schier unüberwindliche Hürde im 3. Level, 7. Screen, oder eine knifflige Situation in einem Abenteuer, niemand ist vor solchen Problemen sicher.

Auf der anderen Seite gibt es aber auch immer wieder Freaks, die sich Tage und Nächte um die Ohren schlagen, um die tiefsten Geheimnisse eines Programms zu ergründen (manchmal hilft auch einfach der Zufall). Unsere Spiele-Ecke, ab sofort eine ständige Einrichtung, soll allen Spielern Hilfestellungen geben, also die Wisenden und die Hilfesuchenden zusammenführen. Wer Lösungshinweise für Adventure oder Unsterblichkeits-Pokes für Spiele hat, wer Tips + Tricks kennt, um Anwenderprogramme besser in den Griff zu bekommen, kann diese einschicken. Jeder Hinweis, gleich welcher Art, der anderen CPC-Benutzern hilfreich sein kann, wird gebraucht. Und natürlich wird auch jeder von uns veröffentlichte Tip honoriert.

Also nichts wie ran an die Schreibwerkzeuge und her mit den Informationen. Die CPC Magazin-Leser werden es danken.

Spieletips für den CPC 464

Kontonummer für GHOSTBUSTERS

Kein Name, nur ENTER
00 30 62 27 für 970100\$

Pokes für Pyjamarama

Verluste der "Schnarchenergie" werden verhindert durch: POKE &23E4,0 :POKE &23F8,0

Pokes für Knight Lore

Die Zahl der Spielexistenzen wird mit POKE &2570, &80: POKE &258C, 0 erhöht. Der zweite POKE ist nötig, da Knight Lore erstaunlicherweise gegen diese Lebenshilfe abgesichert ist. Die Anzahl der Leben steigt auf 128, was hexadezimal als 7F angezeigt wird. Da das Knight Lore Ladeprogramm in Maschinensprache ist, gibt es keine Poke-Ergänzung in Basic. Das Hauptprogramm muß geladen (Startadresse &2000, Länge &8000), verändert und die neue Fassung dann an dieselbe Stelle abgespeichert werden.

Gerhard Knapienski

Wer weiß mehr?

Ich habe zwei Fragen zu Adventures, weil ich da nicht weiterkomme.

1. Jewels of Babylon: Wie komme ich an ein Pulverfaß, mit dem ich den Löwen töten kann?
2. Message from Andromeda: Was muß ich im Computerraum tun, nachdem der Alien Commander besiegt ist?

Dennis Pfofe, Gerlingen

Wie findet man bei dem Adventure "Der Diamant von Rabenfels" den Drachen und die Feen?

Oliver Gerstlauer, Öhringen

Wer hier weiter weiß, kann uns seinen Tip zusenden. Wir veröffentlichen diese Tips und Hinweise dann in den nächsten Ausgaben.

Noch ein Tip für Listings

Das Kommando WIDTH 255 verhindert unerwünschten Zeilenvorschub.

Gerhard Knapienski

Das CPC-Magazin schafft Kontakte!

So schnell sich die Schneider-Computer auf dem deutschen Markt durchgesetzt haben, so schnell haben sich auch User-Clubs und Benutzergruppen gebildet. Diese Clubanschriften, Neugründungen, Termine, Nachrichten und andere Informationen aus der Szene wollen wir an unsere Leser über diese Kontaktseite weitergeben. Ausführliche Clubvorstellungen sind ebenso möglich, wie Kurzinfos, Änderungen oder Kontaktgesuche.

Wer also einen User-Club leitet oder gründen will, wer Kontakte zu anderen Computereeks sucht oder wer besondere Aktivitäten melden kann, sollte uns schreiben.

Unsere Anschrift:
CPC-Magazin
Postfach 1640
7518 Bretten

ABO

n n e m e n t

Abo-Bestellschein

Ich möchte das CPC-Magazin in Zukunft regelmäßig zugeschickt bekommen. Meine Abo-Bestellung gilt ab der nächsten Ausgabe. Die Abodauer beträgt 12/6 Ausgaben und kann bis spätestens 4 Wochen vor Aboende wieder gekündigt werden. Der Abonnementspreis beträgt 66.- DM einschließlich Mehrwertsteuer und Versandkosten. Für Bestellungen aus dem europäischen Ausland wird es aber nur ein wenig teurer: Hier kostet das Abo 75.- DM.

- Ich wünsche ein Jahresabo mit 12 Ausgaben
- Ich wünsche ein Halbjahresabo mit 6 Ausgaben zum halben Preis (33.- DM/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 das CPC-Magazin, Postfach 1640, 7518 Bretten schicken.

CPC **Magazin**
Für alle Schneider Computer

● CPC 464 ● CPC 464 ● CPC 464 ●
Einsatz in Manhattan

Super Text-Adventure ● über 160 verschiedene Räume orig. nach Manhattan ● >39 K ● Für Farb- und Grünmonitor ● Schlüpfen Sie in die Rolle des Kojaks ● echt toll!!! Kassette nur DM 20.–/Disk (3") nur DM 30.–. Bei: D. Sander/Oderstr. 12, 2842 Lohne

Verkaufe CPC 464-Farbe + 3" Floppy + Bücher + jede Menge guter Software + Zubehör für 2199.– DM, nagelneu. ☎ 05553 / 775

SUCHE KONTAKT zu anderen CPC-USERN im Kreis Cochem-Zell. Bitte melden bei J. Schneider, Hauptstr. 15, 5591 Greimersburg, ☎ 026 71 / 15 08

Verkaufe CPC 464 mit Farbmonitor und Textverarbeitung. PHASE 4. VB 1100.–. Orig.-Progr. (50% unter NP): Focus, Centre Court, Colour Star, Tascopy, Amskey, Basic 1 uva. ☎ 040/7017641. Floppy DD1, DM 680.–, ☎ 040/7017641

CPC 464. Tausche Software (Kass.), Spiele und Utilities. Listen an: Rüdiger Möller, Werner-Sombart-Str. 2, oder ☎ 075 31 / 5 41 51, 7750 Konstanz

Elbe-Statistikprogramm

RVO-EGO Kassen für alle Arztpraxen mit Schneider CPC-464-664-6128 und Vortex-Format auf Kass./Disk 3"/5 1/4" Info: Dr. I. Apul, Großpölinger-Heerstr. 155, 2800 Bremen 21, ☎ 0421 / 6 16 46 00. Anfragen mit DM 1.90 Porto.

CPC SOFT + HARD = z. B. Painter. Das Grafikprogramm mit Cursor-Steuerung CPC SOFT + HARD = z. B. umschaltbarer Joystick, Y-Adapter und und und... Info: J. Ruhr, Am Anger 2, 4020 Mettmann

BACKUP: Disketten-Kopierprogramm für CPC 464 mit Vortex 512 K Erweiterung. Wie DISCOPY, jedoch ohne CP/M und in nur einem Durchgang. Für nur 20.– DM (Brief) inkl. Anleitung Porto und Verpackung von W. Spütz, Eyller-Str. 225, 4132 Kamp-Lintfort

●●●●● CPC 464 ●●●●●

Deutsche Umlaute auf Ihrem CPC 464. Buchstaben-Aufkleber für die Tastatur liegen bei! Kassette kommt gegen 10-Mark-Schein von: S. Üren, Merkenicher Hauptstr. 139, 5000 Köln 71

!!! Dringend gesucht !!!

Fußballmanager – deutsch mit Bundesligamannschaften. Abgespeichert auf Kassette für CPC 464. Adressenangabe mit Preis an Schärer, Sebastian, Seesteg 18, CH-8437 Zurzach/Schweiz

GESUCHT: Drucker NLQ 401. ☎ 022 41 / 33 44 74 (Meffert) und Softwareaustauschpartner.

Preisgünstige Programme aus Eigenentwicklung für Ihren Schneider-Computer erhalten Sie bei Friedrich Neuper, Postfach 72, 8473 Pfreimd. Einfach das Gratisinfo anfordern.

CPC-Einsteiger aufgepaßt!!!

15 Originalprogramme auf Kassette zu verkaufen. DM 15.– Stück, DM 200.– gesamt. ☎ 08 21 / 41 37 68, tagsüber.

CPC 464 + 664:

Sicherungskopie einer Diskettenseite in 8 Minuten auf Kassette. 10 Diskettenseiten auf einer C90 Kass. sichern! Geschw. von 1000 bis 3600 Baud. Backups von Laufwerk A + B, auch von 5.25 Laufwerk (40 Spuren). Kassette 14.–DM (Scheck). Dipl.-Ing. W. Renziehausen, Hauptstraße 32, 3360 Osterode 1

● 1 Jahr ●

Herzlichen Glückwunsch, COMPI-CLUB! Zum Jubiläum urd den über 210 Mitgliedern gratulieren die Leiter der User-Gruppen Nord, Süd und Westfalen.

Suche und tausche Programme für CPC 664 (464/6128). Karsten Kammler, Heinrich-Hertz-Str. 4, 4280 Borken-Gemen

Hallo Adventure-Fans!

Jetzt gibt es Treasure Hunt, ein deutsches Adventure mit über 90 (!) Räumen. Jagen Sie den Räuber des Schatzes durch die Pyramide! Disk 35.– DM, Kass. 20.–DM. ☎ 0 48 41 / 6 59 06, Mathias verlangen!

● CPC 464 ● CPC 464 ● CPC 464 ● Softwaretausch! K/3" D. + 1.– DM Porto an: S. Basler, Zur Vogelstange 3h, 4400 Münster

Finden Sie Ihren Drucker auch zu langsam! Dann sollten Sie sich dieses Programm holen: DRUCKERPUFFER. Sie können die Puffergröße frei wählen. Info gegen Rückumschlag bei Sönke Gaekel, Virchowring 107, 2359 Henstedt-Ulzburg, ☎ 0 41 93 / 13 99

DMP1 MATRIX-DRUCKER mit Kabel für CPC 464, neuwertig, englische Anleitung, VB 400.–DM, ☎ 07 31 / 7 99 06

Achtung Fußball-Freunde!

18-fach menügesteuert. Fußballbundesliga-Verwaltungsprogramm. Aktuelle Tabellen, Tabellen aller Spiele seit 1963, Ergebnisabspeicherung und vieles mehr. Info gegen Freiumschlag. Kassette DM 29.–. Klaus Wehrle, Saalburg-ring 40, 6054 Rodgau 2

CPC 464 LOTTO-STATISTIK

●●● 6 aus 49 ●●●

Wollen Sie alle Ziehungen im Lotto 6 aus 49 blitzartig durchsuchen? Z. B. feststellen, welche Gewinnklassen Sie mit Ihren Stammzahlen erreicht hätten, wenn Sie von Anfang an (1955) gespielt hätten? Mit diesem Programm kein Problem: Auswertung beliebig. Bank- und/oder Auswahlzahlen in ca. 12 Sek.! (MC). Kass. 45.– DM, Disk 3" 58.– DM. (Info 1.– DM Briefm.) Nur die Ziehungen als Datenbestand: Kass. 12.– DM, Disk 3" 25.– DM. (Auf der eigenen Disk abzügl. 15.– DM). Günter Blatt, Wiesenstr. 10, 5509 Schillingen

Suche deutsche Adventures für 664 auf Disk. Tausche auch gegen D. B. Bücher! ☎ 0 73 21 / 4 13 57

● CPC ● CPC ● CPC ● CPC ● LOHN- u. EINKOMMENSTEUER

Super Jahresausgleich, Monatslohnsteuer, Steuerklassenwahl, jährl. Aktualisierung ohne Neukauf. Kass. 60.– DM, Disk 75.– DM. Info gegen Rückporto. Horst Ichen, Niederfelderstraße 44, 8072 Manching, ☎ 0 84 59 / 16 69

Textverarbeitung und Dateiverwaltung auf einer Kassette für nur 20.–DM. Klaus Ziehr, Ilxetwiete 1, 2000 Hamburg 74

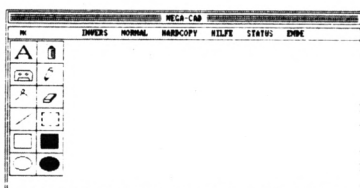
Commodore 64, Schneider CPC 464. Verkaufe Software führender Hersteller. Liste anfordern bei: R. Wilhelm, Hochstadtstr. 12, 6967 Buchen/Odw.

Schneider CPC

Biete Software zu Superpreisen. Über 50 Programme vorhanden. Liste gegen Rückporto bei: Frank Schulze, Oberhausener Str. 68, 4330 Mülheim/Ruhr.

KENNEN SIE DIESES SYSTEM ?

Es läuft auch auf Ihrem Schneider CPC!



MEGA-CAD.

Das Grafiksystem der Superlative

- 640-200 Punkte Auflösung
- Super-komfortable Bedienung in PC-Qualität
- Technische Zeichnungen, Schaltpläne, Platinen layouts, Bilder, usw. können mit geringem Aufwand erstellt werden.
- Zeichnungen können geladen und gespeichert werden
- Problemloser Ausdruck mittels 'Hardcopy' möglich
- Funktionen wie 'Spray', 'Text', 'Freihand-Zeichnen', 'Radieren', 'Kreis', 'Rechteck', usw.
- Bedienung erfüllt zu 100 % interaktiv, Hilfstexte können eingeblendet werden, daher keine Spezialkenntnisse erforderlich
- Mehrere Demo-Grafiken enthalten
- Ausführliches deutsches Handbuch
- Die Lieferung erfolgt auf Kassette

GRATIS-INFO anfordern!
79,-
incl. MwSt.

Computertechnik
Z. Zaporowski
Vinckestr. 4
D-5800 Hagen 1
Tel.: 02331/14344

RIESENAUSWAHL
an Hard- u. Software
für CPC.
Händleranfragen erwünscht



Wir sind die Verbindung zwischen Mensch und Computer!

Inhaber Hans-Jürgen Pirreck

Pizie-Data

Mittelstraße 61

4322 Sprockhövel 2

Telefon: 02339/7191

Schneider CPC 464 mit Grün-Monitor	768.– DM
Schneider CPC 464 mit Farb-Monitor	1268.– DM
Schneider CPC 664 mit Grün-Monitor	1368.– DM
Schneider CPC 664 mit Farb-Monitor	1868.– DM
Schneider CPC 6128 mit Grün-Monitor	1568.– DM
Schneider CPC 6128 mit Farb-Monitor	2068.– DM
Joyce PCW 8256 mit Textverarbeitung und Drucker	2490.– DM
Diskettenlaufwerk "DDI-1"	788.– DM
Diskettenlaufwerk "FD-1"	588.– DM
Vortex F1-F 5 1/4" Diskettenstation 700 KB	1169.– DM
Vortex F1-D 5 1/4" Diskettendoppelstation 1,4 MB	1669.– DM
Vortex SP 64 KB – Speicherkarte, aufrüstbar	269.– DM
Mannesmann Tally-Drucker MT 80+	869.– DM
Wordstar, dBase, Multiplan je	199.– DM
Textpack professionelle Textverarbeitung	198.– DM
RH-DAT Datenbankprogramm, nur Randomzugriff	89.– DM
EASY-CAD 2-D-Zeichenprogramm für CPC 664/6128	89.– DM
5er Pack Schneider 3"-Disketten	59.– DM

Alle Preise inkl. MwSt.
Versand per Vorkasse zuzügl. DM 5.– Versandkosten.
Info gegen DM 2.– in Briefmarken.
Händleranfragen erwünscht.

Bezugsquellenverzeichnis

Was hilft ein Testbericht, wenn keine Bezugsquelle angegeben ist. Deshalb hier die entsprechenden Lieferanschriften. Dies ist allerdings nur eine Auswahl, die meisten Produkte gibt es auch bei anderen Lieferfirmen.

● Drucker-Interface Typ 29000 – Wiesemann Mikrocomputertechnik, Winchenbachstraße 3a, 5600 Wuppertal 2 ● Diskettenwerkzeug – Dagmar Schmidt Software, Wallensteinstraße 37, 7000 Stuttgart 40 ● Schnittstellenvervielfacher Typ 2C/C/0 – Wiesemann Mikrocomputertechnik, Winchenbachstraße 3a, 5600 Wuppertal 2 ● Macadam Bumper, Hacker, Robin von Sherwood – Profisoft GmbH, Sutthausen Straße 50-52, 4500 Osnabrück ● Deathpit, Dragontorc – verschiedene Händler ● Diagramme mit dem CPC 464 – Dynamics GmbH, Große Bäckerstraße 11, 2000 Hamburg 1 ● Teleterminal 300 S – Rushware GmbH, An der Gumpgesbrücke 24, 4044 Kaarst 2 ● ZEN-Assembler – keine deutsche Bezugsquelle bekannt ● CPC-Text/Adress – Markt + Technik Software, Hans-Pinsel-Straße 2, 8013 Haar ● dBase II – verschiedene Händler oder auch CPC-Magazin, Postfach 1640, 7518 Bretten ● Schneider Joyce PCW 8256 – verschiedene Händler oder Schneider Rundfunkwerke, 8939 Türkheim ● PIO-Interface und Epromer – Maximini und Lichter, Hauptstraße 1a, 5511 Mannebach ● Textmaster – Heim Verlag, Heidelberger Landstraße 194, 6100 Darmstadt ● Speichererweiterung – Vortex GmbH, Klingenberg 13, 7106 Neuenstadt.

Schneider **Sinclair**
COMPUTER DIVISION

multihafte Auswahl

ständig das **NEUESTE AUS ENGLAND**

ca. 120 Schneider- und
ca. 200 Sinclair-Programme
ständig auf Lager

Seit dem 2. November
sind wir in den neuen Verkaufsräumen

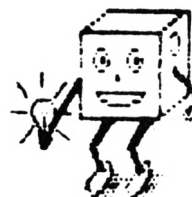
MULTIDATA

Super Allzweckdatei. Voll Maschinencode. Deutscher Zeichensatz, 64 Zeichen (Schneider 80) je Zeile. Verknüpfbare Such- oder Sortierkriterien, gezieltes Suchen einzelner Wörter, Zeichen etc. innerhalb eines Feldes möglich. Maske frei definierbar, 2 frei wählbare Druckerformatierungen, ca. 33 K frei für Daten. Vollkommen menügesteuert. Einfache Handhabung.

Für Spectrum
Für Schneider Kass.
Disk

DM 69.-
DM 89.-
DM 99.-

MULTI-SOFT ROLF STRECKER



Berrenrather Str. 354

5000 Köln 41

☎ 02 21 / 41 77 89

M_N Michael Naujoks

Airwulf	DM 38.-	Lords of Midnight	DM 39.-
Battle of Britain	DM 39.-	Marsport	DM 39.-
Boulder Dash	DM 39.-	Master of Lamps	DM 39.-
Brian Jacks Superstar	DM 33.-	On the Run	DM 33.-
Centre Court	DM 35.-	Pinball Wizzard	DM 33.-
Doppleganger	DM 35.-	Project Future	DM 38.-
Everyone's a Wally	DM 39.-	Robin of Sherwood	DM 39.-
Formula One	DM 33.-	Slapshot	DM 36.-
Frank Bruno's Boxing	DM 38.-	Southern Belle	DM 33.-
Hacker	DM 39.-	Starion	DM 39.-
Highway Encounter	DM 39.-	Way of explod. Fist	DM 39.-
Juggernaut	DM 33.-	Wizzards Lair	DM 39.-

Hardware

SCHNEIDER
CPC

Software

dk'tronics Stereo		Programme auf	
Speech Synthesizer	DM 129.-	3"-Disk	
dk'tronics Light Pen	DM 98.-	Red Arrows	DM 49.-
Teleterminal 300S mit		Slapshot	DM 49.-
RS 232-I/F-Kabel	DM 169.-	Fighter Pilot	DM 49.-
Tonkopf-Justier-Tape	DM 36.-	St. Davis Snooker	DM 49.-
Tasword 464 (dt. Hdb.)	DM 69.-	Tasword 464 (dt.)	DM 99.-
Tasprint 464 (dt. Hdb.)	DM 39.-	Tasprint 464 (dt.)	DM 69.-
Tascopy 464 (dt. Hdb.)	DM 39.-	Tascopy 464 (dt.)	DM 69.-
Multidata (dt. Hdb.)	DM 89.-	Multidata (dt.)	DM 99.-

Kostenlosen Katalog anfordern!

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

Hotline:
(06221) 46885

HP-SOFT

Hindenburg-Allee 3, 8420 Berchtesgaden
Hot-Line 086 52 / 6 30 61

● Super News ●
● Spitzen-Preise ●

★ CPC Kassetten – Software ★

★ CPC Disc – Software ★

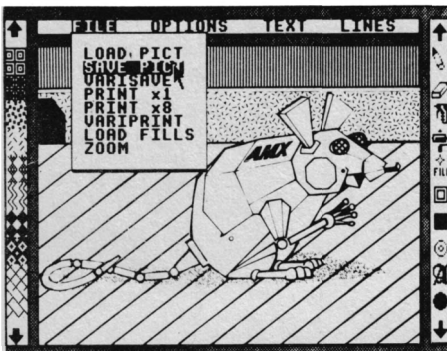
Hacker (super!!!)	39.90 DM	Airwolf (erstklassig!!!)	53.90 DM
Marsport	39.90 DM	Codename Matt II	53.90 DM
Starion (brandneu!!!)	39.90 DM	Fighter Pilot	
They sold a Million	39.90 DM	(Flugsimulator)	53.90 DM
Codename Mat II	39.90 DM	Frank Bruno's Boxing	53.90 DM
Sabre Wulf (v. Ultimate)	39.90 DM	Quill/The Illustrator	43.90 DM
Match Day (Super Fußball)	38.90 DM	Red Arrows	53.90 DM
Spy vs Spy		Slapshot	
(aus dem Comic MAT)	39.90 DM	(Eishockey)	53.90 DM
Southern Bell		Sorcery +	53.90 DM
(464,664,6128)	35.90 DM	Steve Davis Snooker	53.90 DM
Battle of Britain	39.90 DM	CPC Dictionary E/D	69.90 DM
Dynamite Dan	35.90 DM	CPC Dictionary D/E	69.90 DM
Dragontorc of Avalon		Power Conaktion (Profi-Kunden-	
(464, 664)	36.90 DM	dateiverwaltung für ca. 3000 Kunden.)	
Highway Encounter		Zum Superpreis	
(464, 664)	39.90 DM	von NUR!!!	79.90 DM
Slapshot (Eishockey)	39.90 DM	Satellit Warrior	53.90 DM
●●●● DER RENNER ●●●●		Wizzards Lair Disc	49.90 DM
WINTERGAMES!!!	39.90		

VORSCHAU

Das neue CPC-Magazin gibt es ab dem 27. 1. am Kiosk

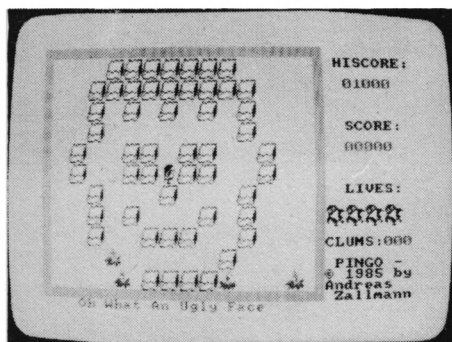
Maus für CPC

In der nächsten Ausgabe stellen wir die neue AMX-Maus für den CPC vor. Was kann man damit machen? Handelt es sich mehr um ein Spielzeug oder ist die Maus ein hilfreiches Arbeitsinstrument?



Pingo

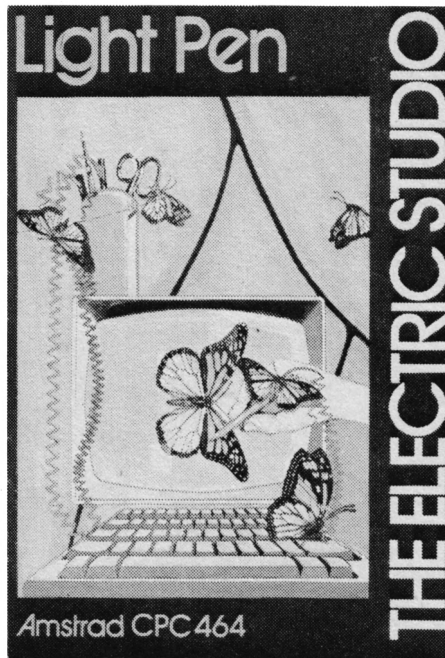
Das altbekannte Pinguin-Spiel mit den Eisblöcken gibt es jetzt auch für den CPC. Andreas Zallmann hat es



für uns programmiert. Passen Sie auf und lassen Sie sich von den Clumbies nicht erwischen.

2 neue Lightpen

Im nächsten Heft stellen wir den dk'tronics Graphic Lightpen und den Electric Studio Lightpen vor. Welcher ist der bessere?



Weitere Themen

Im nächsten Heft gibt es dann noch folgende interessante Themen zu lesen: Die Assemblerecke mit einigen ROM-Routinen speziell für Plot und Draw. Als Anwendungsbeispiel wird eine Fillroutine erstellt.

Beim Z 80 Kurs wird das schon bekannte Programm in Bezug auf Speicherplatz und Geschwindigkeit nochmal verbessert. Auch wollen wir mit dem Abdruck der angesprochenen Z 80 Tabelle beginnen.

Beim Floppykurs setzen wir die bisherigen Kenntnisse in die Praxis um und erstellen ein richtiges Dateiprogramm.

Weiter geht es auch bei unserem Pascal Kurs. Hier werden neue Befehle und Schlüsselwörter vorgestellt und ihre Anwendung erklärt.

Und natürlich bringen wir wie immer allerlei Tips und Tricks, Listings und natürlich wieder ein Topprogramm des Monats. Lassen Sie sich überraschen!

Inserentenverzeichnis

Abacus-Software	53
Brüggendiek	53
Computer Team	28, 79
Data Becker	78, 79
Data Service	6
Data Service Bonn	15
Dobbertin	21
Flesch	53
G-Computerstore	15
Haaga	52
HP-Soft	88
Imfo Imbeck	6
Integral Hydraulik	8
Joysoft	57
Kunz	89
Media Plast	92
Naujoks	88
Pizie-Data	87
Rolf Strecker	88
Schuster	5
SFK	7
Signum Verlag	5
Softwareagentur Heyns	32
Unicom	27
Unicon	3
Valc	3
van der Zalm	52
Vortex	13
Waldeck	91
Widdel	50
Wiesemann	27
Zaporowski	87
ZS-Soft	2

Impressum

Herausgeber	Thomas Eberle Werner Rätz
Chefredakteur	Thomas Eberle
Techn. Redaktion	Werner Rätz
Ständige freie Mitarbeiter	Andreas Zallmann Manfred W. Thoma Rolf Knorre Markus Pisters Hans-Peter Schwaneck Hans Joachim Janke Gerhard Knapieski Prof. Walter Tosberg
Versandservice	Elvira Rätz
Layout	Jürgen Seefeld
Titelbild	Rainer Grinda
Anzeigen	Arno Weiß Es gilt die Anzeigenpreisliste Nr. 2 vom 24. 9. 1985.
Satz	Druckerei Sprenger 7143 Vaihingen/Enz
Druck	Südd. Zeitungsdruck GmbH Druckerei- u. Verlags-GmbH 7080 Aalen
Vertrieb	Verlagsunion 6200 Wiesbaden
Anschrift des Verlages	Verlag Rätz-Eberle Postfach 1640 7518 Bretten Telefon 0 72 52 / 4 29 48

Manuskript- und Programmeinsendungen: Manuskripte und Programm Listings werden gerne von der Redaktion angenommen. Sie müssen frei von Rechten Dritter sein. Sollten sie auch an anderer Stelle zur Veröffentlichung oder gewerbliche Nutzung angeboten worden sein, muß dies angegeben werden. Mit der Einsendung von Manuskripten und Listings gibt der Verfasser die Zustimmung zum Abdruck in den vom Verlag Rätz-Eberle herausgegebenen Publikationen und zur Vervielfältigung der Programme auf Datenträgern. Für unverlangt eingesandte Manuskripte und Listings wird keine Haftung übernommen. Eine Gewähr für die Richtigkeit der Veröffentlichungen kann trotz sorgfältiger Prüfung durch die Redaktion nicht übernommen werden. Die gewerbliche Nutzung, insbesondere der Schaltpläne und Programme, ist nur mit schriftlicher Genehmigung des Herausgebers zulässig.

Das "CPC-Magazin" erscheint monatlich jeweils am letzten Montag des Vormonats. Das Einzelheft kostet DM 5.50.

Schneider CPC Software-Schnell-Versand

Winter Games
Der Hit von EPYX
jetzt für den SCHNEIDER
KASSETTE
DM 36,90

BRUCE LEE

20 KAMMERN!
Kassette 35,90
Disc 49,-

ELITE
endlich lieferbar

DISC 3" 69,90
KASSETTE 58,90

AMX mouse
MAUS, GRAFIKPAKET
und BASIC-ERWEITERUNG
für alle CPC's **298,- DM**


ExBasic 50 neue Befehle für den 464
Grapaper Reset.on Scrolldown Tprint Wait
Grapen Rightscroll Scrollup Unlock Zip
Beep Breakoff Bscrollup Cmode Copychar Deek Fill Hmirror Lock Romdeek Rompeek Video Window
Blank Breakon Circle Colour.off Crash Doke Frame Inkscroll Pause Reset.off Setcolour Vmirror Zrscroll
Box Bscrolldown Clw Colour.on Cursor Disc Get Leftscroll Picture Ppicture Sound Square Vreset Zlscroll
Kassette 49,- Disc 59,-



KASSETTE

A View to a kill	47,90
Action Biker	9,95
Airwulf	29,00
Alien 8	35,90
Arnheim	39,00
Assembler Kurs (Sybex)	64,00
Battle of Midway	36,90
Battle of Britain	34,90
Beach Head	37,90
Boulderdash	38,90
Brain Jacks Superstar	29,90
Centre Court	31,90
Chiller	9,95
Code Name Mat 2	49,00
Colour Star 2	29,90
Combat Lynx	29,90
Confuzion	29,00
Copy Star 2	39,90
Cyrus 2 Schach	39,00
Danger Mouse	38,90
Dethlon	29,00
Designer Star	59,90
Devils Crown	29,90
Devpac	129,00
Doppelgänger -	29,90
Dragonator	34,90
Dun Darach	37,90
Dynamite Dan	29,90
Eric the Viking	36,90
Everyone's a Wally	33,90
Fantastic Voyage	34,90
Fighter Pilot	34,90
Fighting Warrior	33,90
Finders Keepers	9,95

Forest at the Worlds End	27,90
Formula One	29,90
Formula One Simulator	9,95
Frank Brunos Boxing	29,00
Genesis	39,00
Ghostbusters	38,90
Gremlins	36,90
Hacker	38,90
Hard Hat Mac	37,90
Herberts Duminy Run	34,90
Hexenküche	29,00
Highway Encounter	35,90
Hypersports	39,00
Kaiser	59,00
Knight Lore	35,90
Locomotion	9,95
Lords of Midnight	39,00
Macadam Bumper	29,90
Marsport	35,90
Masterfile	99,00
Master of the Lamps	37,90
Match Point	29,00
Message from Andromeda	27,90
Moonbuggy	29,90
Mr Freeze	9,95
Multidata	69,00
Multivokabel	49,00
Neverending Story	34,90
Nightshade	34,90
Nonterraqueous	9,95
One Man & Droid	9,95
On the Run	29,90
Pinball Wizard	27,90
Project Future	34,90
Raid !! (over Moscow)	38,90
Red Arrows	35,90

Red Moon	28,90
Return to Eden	37,90
Robin of Sherwood	37,90
Rocky Horror Show	29,90
Shorts Füse	9,95
Slapshot	31,90
Sorcery	33,00
Soul of a Robot	9,95
Souls of Darkon	32,90
Southern Belle	29,90
Space Hunter	9,95
Spy vs Spy	34,90
Starion	34,90
Star - Moon	59,00
Statistic - Star	59,90
Strip Poker	34,90
Strong Man	35,90
Tales of Arabian Nights	27,90
Tascopy deutsch	29,90
Tasprint deutsch	29,90
Tasword deutsch	49,90
The Hobbit	46,90
The Quill	59,00
They Sold a Million	36,90
Torremolinos	29,00
Warlord	29,90
Way of Exploding Fist	36,90
Wild Bunch	9,95
Winter Sports	36,90
Winter Games	36,90
Wizards Lair	29,90
Word Cup	34,90
Word Series Baseball	29,00
Yie ave Kung Fu	34,90
Zapp	49,90

DISC 3"

Airwolf	49,00
A View to a Kill	59,00
Com Pack	798,00
Code Name Mat 2	49,00
Cyrus 2 Schach	49,00
Datamat	139,00
Datei Star	98,00
dBase 2	199,00
Devpac	145,00
3 D Stuntrider	49,00
3 D Grand Prix	49,00
Fantastic Voyage	49,00
Fibu Star	98,00
Fighter Pilot	49,00
Frank Brunos Boxing	49,00
Grafik Master	89,00
Lager Star	89,00
Lords of Midnight	49,00
Masterfile	99,00
Multiplan	199,00
Multivokabel	59,00
Profi Painter	179,00
Red Arrows	49,00
Satelite Warrior	49,00
Slapshot	49,00
Sorcery plus	49,00
Spy vs Spy	49,00
Star Writer 2	189,00
Star Moon	89,00
Statistic Star	79,00
Supercalc 2	198,00
Tasword 6128	96,00
Tasword-D	96,00
Tex Pack	198,00
Textomat	139,00
Textomat plus	198,00
Wizard Lair	49,00
Wordstar	199,00

DRUCKER

Melchers CP 80	698,00
Melchers CPA 80 P	698,00
Neu-Speziell für Schneider	749,00
Melchers CPA 80 GS	1149,00
Taxan KP 810	1449,00
Taxan KP 910	798,00
Schneider NLQ 401	798,00
Panasonic KX-P 1090	998,00
Panasonic KX-P 1091	998,00
Panasonic KX-P 1092	1249,00

HARDWARE

CPC 464 grün	798,00
CPC 464 farbe	1298,00
CPC 6128 grün	1598,00
CPC 6128 farbe	2098,00
Joyce PCW 8256	2490,00
Floppy DDI-1	798,00
Vortex F1-S	1198,00
Vortex F1-D	1598,00
3" Zweitlaufwerk	399,00
5,25" Zweitlaufwerk	598,00
Joystick PRO 5000 Mic.	59,00
Joystick Y-Adapter	47,90
800 Kbyte Zweitlaufwerk für CPC 464	659,00
800 Kbyte Zweitlaufwerk für CPC 664/6128	799,00
Joycard	24,95
Sprachsynthesizer	139,00
Lightpen	99,00



Heimcomputer-Shop

Waldeck-Automaten Vertriebsgesellschaft mbH

Bahnhofstraße 10 Telefon
2870 Delmenhorst (04221) 16464

Auf 1 Postkarte kleben oder in Briefmarken stecken und ab geht die Post!
Ja, schicken Sie mir umgehend folgende Artikel aus Ihrem Angebot

V-Scheck anbei per Nachnahme (zzgl. 5,- DM Versandkosten)

Stück Bargeld anbei DM
 Stück DM
 Stück DM

Versandskosten _____ DM
 Gesamtsomme _____ DM

Absender (deutsch schreiben)

