



Das Insider Magazin

Nr. 6/85

3. Jahrgang

5,50 DM 6,00 sFr 48 öS

Computing Programming Using



Eine der zahlreichen Neuheiten auf der Hannover-Messe '85: Der MPF-PC/ET von Multitech



Hardware-Tests:

MSX-MPC 80
Spectrum Floppy:
„Discovery 1“

Software-Tests:

RGH-Basic = für Spectrum
Logo

User:

Reassembler = C-64
Maskengenerator

Videoprozessor des MSX

Tips & Tricks
für CPC 464 und Spectrum

UTILITY

TI-99: Multicolor – sprites

CPU-Game:

CPC 464: „Aladin“

C 64 * CPC 464 * VC-20 * Atari * Apple * TI-99 * ZX-Spectrum * MSX *

Noch mehr Infos und Utilities!

Schon gehört?! -- ++ - Schon gehört?! -- ++ - Schon gehört?! -- ++ -

Der TRONIC-VERLAG sucht in den Computerzeitschriften „COMPUTRONIC“ und „HOMECOMPUTER“ den

Software-Champion 1985!!!

Es warten tolle Preise auf Sie!

1. PREIS:

5000 DM in bar!

2. PREIS:

**Ein Farbmonitor
im Wert von 800,- DM!**

3. PREIS:

**Computer-Zubehör (wahlweise
im Gesamtwert von 300,- DM)**

Nutzen Sie Ihre Chance, beteiligen Sie sich mit einem Software-Programm an unserem Wettbewerb!

Jedes eingesandte Programm nimmt an der Auswahl teil!!!

Letzter Annahmetermin ist der 30. September 1985. Der Rechtsweg ist ausgeschlossen.

Zur Auswahl:

Wir stellen in jeder Ausgabe von COMPUTRONIC mindestens drei TOPPROGRAMME, in „HOMECOMPUTER“ 1 Topprogramm vor. Hierfür trifft die Redaktion die Entscheidung (unabhängig vom Computertyp).

Wird ein Programm zur Darstellung im Heft herangezogen, zahlt der Verlag ein Honorar von DM 120,- pro abgedruckte volle Seite. Zum Jahresende stellt die Redaktion noch einmal alle als TOPPROGRAMME ausgezeichneten Programme vor. Unsere Leser und die Redaktion werden dann in der Ausgabe Nov./Dez. '85 das absolute TOPPROGRAMM des JAHRES wählen.

Der Autor des Programmes wird in unserer Ausgabe vorgestellt und mit DM 5000,- prämiert werden.

Einzusenden sind:

++ Spielbeschreibung

++ Datenträger

++ Listing (nicht unbedingt erforderlich)

Einsenden an: TRONIC-VERLAG -

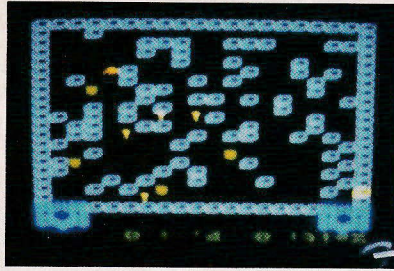
Postfach - 3444 Wehretal 1

KENNWORT: SUPER-SOFTWARE-
CHAMPION

**Wir wünschen allen Freizeitautoren
viel Spaß beim Mitmachen.**



Die Ce-Bit-Halle 4: Hier konzentrierte sich das Messageschehen im Bereich der Homecomputertechnologie



„Aladin“ – Ein Geschicklichkeitspiel für CPC 464-Fans!



Der größte Computer der Computer-Familie von Texas Instruments auf der Hannover-Messe '85 ▶

CPU-Utility

Atari	
Directory Printer	35
Monitorprogramm-Diskettenversion	32
VC-20	
Vic-Clock	40
ZX-Spectrum	
Fill-Programm	17
TI-99	
Multicolor	57

CPU-User

CPC-464	
Keyboard Toolkit	50
Farbdekodierung	70
TI-99	
Load Programm	22
Apple	
Disk Check	54
Dos Change	14
C-64	
Reassembler	27
Maskengenerator	25
ZX-Spectrum	
Copyit	
Mit Maschinensprache zum Bildschirm	9
MSX	
Videoprozessor	47

CPU-Tips und Tricks

Für CPC-464 und ZX-Spectrum	52/61
-----------------------------	-------

CPU-Markt

Neues vom Computermarkt	13/26
Das CPU-Team auf der Hannover-Messe	4

CPU-Game

CPC-464	
Aladin	62

CPU-Test

Hardware:	
Logo für den ZX-Spectrum	56
MSX-Computer	43
„Discovery 1“ – Ein neues Diskettenlaufwerk	30
Software:	
RGH-Basic von Wisosoft	11

Rubriken

CPU-Wörterbuch	8
Impressum	37
CPU-Happy	38
CPU-Bibliothek	75
Softwareservice	66
Reportage: Hannover-Messe '85	74
Computer-Messen '85	44
CPU-Basar	46
Computer-Club-Vorstellung	16

Hannover-Messe: Riesiges Angebot der Computerindustrie

Schon am ersten Tag der Hannover-Messe war der Tronic-Verlag mit einem Team auf dieser bedeutenden Industriemesse vertreten. Das für uns interessante Angebot (Personal- und Homecomputerbereich) konzentrierte sich in den Ce-Bit-Ausstellungshallen 1 und 4. Schon auf den ersten Eindruck besonders auffällig die Vielzahl an Neuheiten vor allem im Personalcomputerbereich. Das, was sich auf der Hobby-Tronic in Dortmund angekündigt hatte, wurde auf der Hannover-Messe durch die vertretenen Computerfirmen eindrucksvoll bestätigt. Auf der Hobby-Tronic konnten keine Neuvorstellungen notiert werden; im Gegensatz

zu Hannover, dort stellten beinahe die gesamten etablierten Unternehmen, unter anderem Atari, Commodore, Apple, ihre neuesten Entwicklungen vor, so daß es dem Besucher fast unmöglich war, den Überblick zu wahren. Besonders zu erwähnen Texas Instruments, der einzige Vertreter, der auf dieser Messe eine komplette Computer-Familie vorstellte. Im Gegensatz dazu glänzte Sinclair durch Abwesenheit. Das Team der CPU-Redaktion wird versuchen, in dieser und in den nächsten Ausgaben einen Überblick und eine ausführliche Vorstellung der wichtigsten Computerneuheiten zu geben.



Die Ce-Bit-Halle 4: Das Zentrum für den Homecomputerbereich.

Neu zur Hannover-Messe 1985

Philips MSX Home-Computer VG 8020

Voll-Tastatur ermöglicht komfortables Arbeiten

Nach der Markteinführung des ersten Philips MSX Home-Computers im Herbst 1984 folgt nun zur Hannover-Messe ein komfortabel ausgestattetes zweites Home-Computer-System: VG 8020. Eine Voll-Tastatur, die einschließlich Cursor-Kontrolltasten und 10 Funktionstasten insgesamt 73 Tasten umfaßt, erlaubt müheloses Arbeiten wie auf einer professionellen Schreibmaschinentastatur.

Die charakteristischen Daten und Ausstattungsmerkmale des VG 8020:

- 80 k RAM-Kapazität (einschließlich 16 k Video RAM)
- 32 k MSX Basic ROM
- Eingebautes Drucker-Interface
- Eingebautes Netzteil
- Verbessertes Daten-Recorder-Interface, das auch bei falscher Polarität der Kabel-/Recorder-Kombination einwandfrei arbeitet
- Anschlüsse für Joy-sticks (Handregler) auf der Vorderseite der Konsole

Das leistungsstarke, integrierte MSX Basic ermöglicht dem Benutzer ernsthaftes, schnell zum Ziel führendes Programmieren. Es können



16 Farben und 32 Sprites dargestellt werden. 254 unterschiedliche Zeichen und Symbole stehen zur Verfü-

gung. Mit dem „Soundprozessor“ sind Geräusche und Musik dreistimmig über 8 Oktaven programmierbar.

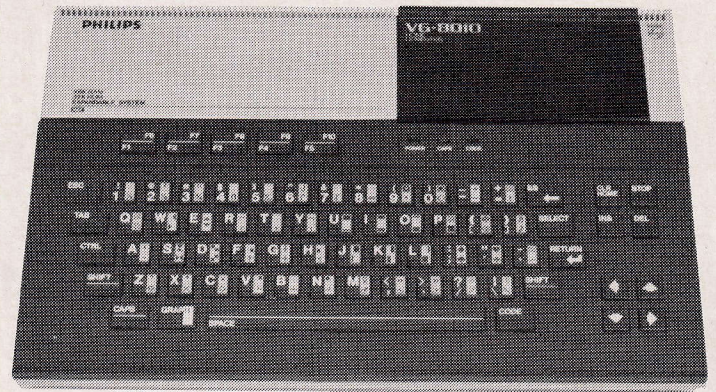
Austauschfähigkeit des RAM-Speichers über 128 k. Zusätzliche Bausteine sind Daten-Cassetten-Recorder, s/w- oder Farb-Monitore (Farbmonitore ab Mitte 1985), Drucker und Floppy-Disc-Speicher.

Technische Daten VG 8020:

Microprozessor:
Z80A mit 3,6 MHz Taktfrequenz

Speicher:
80 k RAM einschließlich 16 k Video RAM 32 k ROM

Schnittstellen:
HF-Ausgang z. FS-Gerät, Video-Ausgang z. Monitor, Centronics z. Drucker, Daten-Cassetten-Recorder, Handregler
2 Modul-Steckplätze (z. B. für Floppy-Disc, RAM Erweiterung, RS 232)
Zeichensatz:
254 alphanumerische und graphische Zeichen
Tastatur:
professionelle Schreibmaschinentastatur mit 73 Tasten einschließlich Cursor-Kontrolltasten und 10 Funktionstasten



Philips MSX-Home-Computer VG 8020

Hannover-Messe: Der PC-TANDY 1000

Der Markt der „Kompatiblen“:

Gründe für die Entwicklung des TANDY 1000:

Angesichts der Fülle von Variationen des IBM-PC, die sich augenblicklich auf dem Markt befinden, teilt sich der Sinn einer Neuentwicklung nicht unmittelbar mit. Es ist aber in jedem Falle interessant zu betrachten, welche Vorteile sich aus der Ausrichtung des Marktes auf den PC für den möglichen Anwender ergeben:

- Standardisierung des Betriebssystems
- dadurch Verbreiterung des Softwareangebots

- Austauschbarkeit von Datenträgern
- Große Peripherieauswahl
- breiteres Informationsangebot

Dieser Zustand kann sicherlich als befriedigend bezeichnet werden. Allerdings führt zum Beispiel das große Softwareangebot nicht nur zu einer guten Problemlösung, sondern oft genug zu großer Ratlosigkeit – insbesondere beim Microeinsteiger.

Was kann man verbessern und bieten?

- einen attraktiven Preis
- für ein „betriebsfertiges“ Gerät
- inklusive eines integrierten Softwarepaketes

Dies sind auch die Punkte, die der Konzeption des TANDY 1000 zugrunde liegen. Der Kunde erwirbt eine Sy-



stemzusammenstellung, die er direkt zum Einsatz bringen kann ohne erhebliche Summen in Zubehör und Software investieren zu müssen. Das mitgelieferte Programmpaket „DESKMATE“ beinhaltet alle wesentlichen Funktionen der populärsten Standardprogramme:

- Textverarbeitung
- Kalkulation
- Datenverwaltung
- Kommunikation
- Terminplanung

Texas Instruments stellt auf der Hannover-Messe komplette Computerfamilie vor

Nur wenige Wochen nach Ankündigung des tragbaren Professional Computers „Pro-Lite“ in USA und Europa stellt Texas Instruments mit Business-Pro das „Flaggschiff“ der bekannten TIPC-Familie vor.

Als derzeit einziger Hersteller von Mikrocomputern kann Texas Instruments eine „vollständige“ Produktpalette offerieren: PCs für mobilen und stationären Einsatz, netzwerkfähig in lokalen und öffentlichen Netzen für Datenübertragung, einige hundert verschiedene Anwendungen und Komplettierung der Systemfamilie durch eigenes Druckerprogramm.

In ca. zwei Jahren ist eine eigenständige neue Generation wirtschaftlicher „Com-

puterleistung“ bei Texas Instruments entwickelt worden, deren Anwendungsvielfalt mehr Effizienz und Entscheidungsparameter an den Arbeitsplatz bringen und die Wettbewerbssituation für kleine und mittlere Unternehmen verbessern hilft.

Business-Pro ergänzt die TI Professional Computer-Familie „nach oben“ und bietet bei erweiterter Kapazität für Arbeitsspeicher und Massenspeicher gleichzeitig eine höhere Verarbeitungsgeschwindigkeit durch Verwendung eines Hochleistungs-Prozessors 80286.

Das Leistungsspektrum von Business-Pro entspricht den steigenden Anforderungen

der Anwender nach Einbindung in vorhandene oder zu implementierende Computer-Netzwerke und nach Systemeigenschaften, die Mehrbenutzer-Betrieb ermöglichen.

Mit 14 Steckplätzen für Systemerweiterungen – Ausbau des Arbeitsspeichers 3 – 15 Mio. Byte, Schnittstellen für Datenübertragung, Steuereinheiten für integrierte oder externe Massenspeicher-Laufwerke (Disketten bis 1,2 MB, Winchester bis 72 MB); Steckmodulen für zusätzlich anschließbare Bildschirm-Arbeitsplätze usw. – läßt sich Business-Pro flexibel schon heute für Applikationen einsetzen, die in Zukunft an Umfang und Komplexität zunehmen werden.

Business-Pro besteht aus drei Systemkomponenten: einem hochauflösenden Bildschirm (monochrom oder Farbe), der ergonomischen Eingabetastatur (getrennte Funktionsblöcke) und der Systemeinheit mit allen Steuerfunktionen für das Gesamtsystem. Die Systemeinheit kann „platzsparend“ unter einem Schreibtisch untergebracht werden – Business-Pro paßt sich den räumlichen Gegebenheiten eines Arbeitsplatzes an.

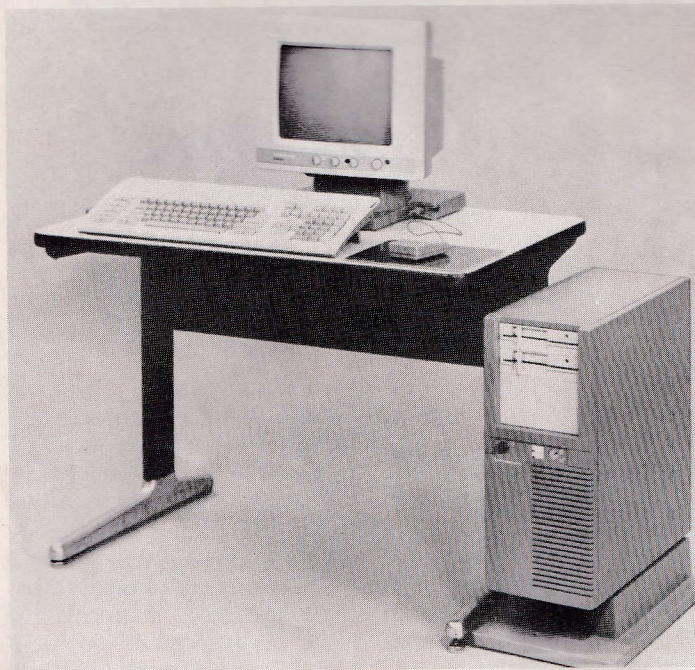
Business-Pro ist als TIPC- und IBM PC AT-Version verfügbar und unterstützt somit eine Vielzahl vorhandener Anwendungen. Das Betriebssystem MS-DOS 3.X enthält die Funktionen der vorhergehenden Version 2.X und die notwendigen Erweiterungen zur Unterstützung der Business-Pro-Hardware. Darüber hinaus sind die Möglichkeiten der Dateiverwaltung, Fehlerbeseitigung und -aufzeichnung verbes-

sert, virtuelle Speichermöglichkeiten gegeben und Tastaturanpassungen für nationale Zeichensätze vereinfacht. Das Betriebssystem XENIX 3 gestattet Mehrbenutzer-Betrieb (maximal acht) und wird im 4. Quartal 1985 für Business-Pro angeboten. Auslieferungen der TIPC-/IBM PC AT-Versionen sind für das 3. Quartal vorgesehen.

Technische Spezifikationen

Systemeinheit:

- 16-Bit-Prozessor 80286 (6MHz) – 16 Datenbits/24 Adressbits
- Co-Prozessor 80287 (Option)
- RAM-Speicher ab 512 KB (150 ns)
Ausbau bis 3 MB ohne Belegung eines Erweiterungs-Steckplatzes
maximaler Ausbau auf 15 MB
- Disketten-Laufwerke 360 KB/1,2 MB
- Winchester-Laufwerke 21/40/72 MB
- ¼-Zoll-Streamer-Tape 60 MB
- 8 plus 6 Erweiterungs-Steckplätze
- Druckeranschluß (Bit-parallel)
- Video-Steuereinheit (Monochrom/Farbe)
- Tastatur-Anschluß
- RS-232-C-(V.24-)Schnittstelle
- LAN-Schnittstelle
- 8/4-Kanal-Schnittstelle (Anschluß weiterer Bildschirm-Terminals)



Der größte Baustein der TI-Computerfamilie

**Fortsetzung
S. 74**

BILLIGER

wird auch Ihre Homecomputerzeitschrift

wenn Sie sich für ein Jahresabonnement
von
HOMECOMPUTER, COMPUTRONIC
und CPU als Kombination
entscheiden.

Für
6 Hefte von HOMECOMPUTER,
6 Hefte von COMPUTRONIC und
12 Hefte von CPU
zahlen Sie im Jahresabonnement
statt 141,- DM nur 100,- DM.
Sie sparen also sage und schreibe: 41,- DM.

Bei Lieferung in das europäische Ausland beträgt
das Jahresabonnement HOMECOMPUTER, COMPUTRONIC
und CPU 150,- DM, nach Übersee 200,- DM

Wir garantieren:

- ★ Sie erhalten HOMECOMPUTER, COMPUTRONIC und CPU ab der nächsterreichbaren Ausgabe.
- ★ Lieferung erfolgt frei Haus inkl. Mehrwertsteuer.
- ★ Die Zustellgebühren sind im günstigen Abonnementpreis enthalten
- ★ Es entstehen Ihnen keine weiteren Kosten.

Ausschneiden und einsenden an:

Bestellgutschein

Ja, ich möchte ins Computerleben einsteigen und bestelle deshalb:

- Jahresabonnement **Kombi: HC, Computronic und CPU** (24 Hefte) 100,- DM
- Jahresabonnement **HC und Computronic** (12 Hefte) 55,- DM
- Jahresabonnement **CPU** (12 Hefte) 55,- DM

Name/Vorname

Straße, Nr.

PLZ, Ort

Ich wünsche folgende Zahlungsweise:

- Bargeldlos durch Bankeinzug

Bankleitzahl

Geldinstitut

Konto-Nr.

- gegen Rechnung

Datum, Unterschrift

Abonnements-Kündigungen:

6 Wochen vor Ablauf des Jahresabonnements.

Außerdem kann ich diese Bestellung innerhalb von 10 Tagen beim Verlag widerrufen!

Zur Wahrung der Frist genügt

die rechtzeitige Absendung

des Widerrufs!

Datum, Unterschrift

Verlagsunion

Friedrich-Bergius-Straße 20
Postfach 57 07

6200 Wiesbaden



Fortsetzung des CPU-Wörterbuches

-D → → D →

Digitalisierung

Digitalisierung bedeutet die Umwandlung von analogen Formen (Kurven, Linien, Kreise) in digitale Zahlen. Diese Zahlen werden dann durch den Computer verarbeitet. Jede dieser Formen ist aus einzelnen Bildpunkten zusammengesetzt.

Dollar-Zeichen (\$)

In der Programmiersprache BASIC steht dieses Zeichen für eine Variable (= String). Bei bestimmten Computersystemen weist das \$-Zeichen darauf hin, daß die folgenden Zahlen in hexadezimaler Darstellung aufgeführt werden.

-E → → E →

Escape

Die ESCAPE-Funktionstaste ermöglicht das Aussteigen aus dem Programm nach erfolgter Fehlbedienung. Diese Taste arbeitet jedoch als Befehlscode-Umschaltungs-taste, in Verbindung mit anderen Tasten können bestimmte Betriebsarten des Computers abgerufen bzw. aufgehoben werden.

EPROM

Erasable Programmable Read Only Memory; zu deutsch = Festwertspeicher (löschar und programmierbar). EPROM besitzt ähnliche Eigenschaften wie ein PROM, kann aber im Gegensatz dazu den Speicherinhalt durch intensives ultraviolettes Licht löschen. Anschließend können neue Programme eingeladen werden. EPROMs sind durch ein rechteckiges Sichtfenster auf dem Chip erkennbar.

Erase

ERASE ist ein Löschbefehl aus der Programmiersprache BASIC. Bei Gebrauch dieses Befehls können Felder gelöscht werden.

Editieren

Bezeichnet das Eingeben von Zeichen über die Tastatur in den Computer. Diese Zeichen sind dann auf dem Bildschirm erkennbar. Für diesen Vorgang stehen dem Computer Editoren (Hilfsprogramme) zur Verfügung. Editoren sind meist schon als Unterprogramme im Betriebssystem integriert.

Elan

Dies ist eine höhere Programmiersprache, die in Deutschland zwischen 1974 und 1977 entwickelt wurde. Elan ist bei den Programmiersprachen Algol und Pascal anzusiedeln. Elan ist recht einfach erlernbar.

Endlosschleife

Bei der Verwendung einer Endlosschleife läuft das Programm nicht ununterbrochen durch, sondern wird an einer bestimmten Stelle gestoppt; von nun an läuft das Programm im Kreis. Endlosschleifen dienen zum wiederholbaren Abarbeiten von Anweisungen.

-F → → F →

Festwertspeicher

Englisch: Read Only Memory (ROM); Informationen, die im Festwertspeicher abgelegt sind, können nur noch gelesen werden. Im Gegensatz zu flüchtigen Speichern behält diese Speicherart auch nach Abschalten des Betriebsstroms die gesammelten Daten. Auf jede Speicherzelle eines Festwertspeichers kann durch eine Adressenangabe direkt zugegriffen werden.

File

Anderer Begriff für Datei, welcher auf einem Massenspeicher abgelegt werden kann.

Firmware

Dies sind Betriebsprogramme eines Computers, die fest im Computer integriert sind (ROM's, ERROM's).

Floppy

Kürzel für Diskettenlaufwerk (Massenspeicher).

Flußdiagramm

Dies ist die vereinfachte Darstellung eines Programmablaufes.

Form Feed

Seitenvorschub bei einem Drucker; aufgerufen durch den ASCII-Code 255 (\$FF).

Programm: „COPY IT“

Für den ZX-Spectrum 48k

Die Vorteile des Programms:

- 1.) Es können sowohl ‚normale‘, als auch Programme ohne Header (also ohne Briefkopf) kopiert werden.
- 2.) Es können so viele Programme geladen werden, bis der freie Speicher (ca. 40 000 Bytes) belegt ist, wobei
- 3.) auf Tastendruck eine Liste der im Speicher stehenden Programme abgerufen werden kann.
- 4.) Programme können in beliebiger Reihenfolge abgespeichert werden.
- 5.) Eventuelle Ladefehler werden abgefangen.
- 6.) Komfortable Menütechnik.

Abtippen:

Das gesamte Programm besteht aus zwei Teilen: Dem Basiclader und dem eigentlichen Kopierprogramm.

Natürlich ist der Maschinensprachenteil auch allein lauffähig.

Der Basiclader dient nur der Bequemlichkeit (man muß sich die Einsprungadresse nicht merken) und der Farbwahl.

Diese kann mittels des Basicladers leicht den eigenen Wünschen angepaßt werden.

Die Reihenfolge des Abtippens ist beliebig.

Wichtig ist nur, daß am Ende auf der Kassette zuerst Listing 1 (der Basiclader) steht. Gefolgt vom MC-Teil, der Listing 2 generiert wird.

Um das Abtippen der Datenwüste zu erleichtern, werden die Werte von je 40 Bytes zusammengezählt.

Stimmt die Summe nicht mit dem richtigen Wert überein, so gibt das Programm den Datenblock an, in dem der Tippfehler liegen muß.

Das Programm ist nur für die 48k Version geeignet.

Benutzen:

Wenn das Programm richtig abgetippt wurde und dann ausprobiert wird, meldet es sich mit folgenden Optionen:

- 1) Laden – normal
- 2) Laden – ohne Header
- 3) Abspeichern
- 4) Liste
- 5) Clear
- 6) Ende

zu 1): Hiermit werden Programme mit Header geladen. (Diese erkennt man daran, das zu Anfang des Ladevorgangs zweimal breite Streifen über den Bildschirm laufen.)

zu 2): Programme, die sich mit ‚1‘ nicht laden lassen, sind Programme ohne Header. (Hier laufen nur einmal breite Streifen über den Schirm.) Diese lassen sich mit ‚2‘ laden.

zu 3): Haben Sie jetzt alle Programme im Speicher und wollen diese abspeichern, so müssen Sie Taste ‚3‘ drücken. Nun erscheint die Art und der Programmname des zuerst geladenen Files und die Frage

SAVE (J/N/E).

Durch Drücken von ‚J‘ wird das Programm abgespeichert. (Es erscheint die Meldung ‚Start Tape then...‘ auf dem Bildschirm.)

Ist der Speichervorgang beendet, so wird das nächste Programm auf gleiche Weise abgefragt.

Wird ‚N‘ gedrückt, erscheint sofort der nächste Programmname mit der Aufforderung SAVE (J/..).

Durch Drücken von ‚E‘ (für Ende) gelangen Sie sofort wieder in das Hauptmenü.

zu 4): Hiermit werden Art, Name und Länge der im Speicher stehenden Files sichtbar gemacht.

zu 5): Durch ‚Clear‘ werden alle im Speicher stehenden Programme gelöscht.

Man befindet sich im Ausgangszustand, wie nach dem Laden von ‚COPY IT‘.

zu 6): ‚Ende‘ steht für ‚Programmende‘. Es erfolgt ein Sprung zum Anfang des ROMs – alles im RAM verschwindet; also auch das Kopierprogramm.

Man befindet sich wieder im Basic.

Sollte bei dem Laden von Programmen ein Fehler auftreten oder die BREAK-Taste gedrückt werden, so erscheint die Meldung ‚LADEFehler‘ auf dem Bildschirm.

Durch Drücken einer beliebigen anderen Taste kommen Sie zurück ins Hauptmenü.

Da Programme ohne Header keine Namen besitzen, werden sie durch ein Sternchen (‚★‘) und eine Ziffer (zur Unterscheidung) gekennzeichnet.

Ein typisches headerloses File sähe z. B. in der Liste so aus:

★0 03215 Bytes

Man muß also anhand der Ziffer nach dem ‚★‘ das Programm identifizieren können.

```
10 INK 6: PAPER 0: BORDER 0: CLS
20 PRINT AT 10,7;"LOADING 'COPY IT'"
30 INK 0
40 LOAD ""CODE
50 INK 6
60 RANDOMIZE USR 50294
```

Teil 2:

```
10 REM *****
20 REM *          COPY IT          *
30 REM *
40 REM *          URS MUELLER     *
50 REM *
60 REM *  SCHUERENKAMP 10        *
70 REM *
80 REM *  4 3 7 0  M A R L      *
90 REM *****
```

```
100 DATA 3913
110 DATA 221,33,66,97,221,34,37,96,
33,255,255,43,34,39,96,175,50,43,96,
62
120 DATA 47,50,64,97,195,30,95,205,
225,95,17,64,96,1,144,0,205,60,32,20
5
130 DATA 5031
140 DATA 51,95,205,253,95,254,49,20
2,101,94,254,52,202,114,95,254,51,40
,32,254
150 DATA 54,202,0,0,254,50,202,222,
93,254,53,32,225,205,225,95,17,3,97,
1
160 DATA 4609
170 DATA 19,0,205,60,32,205,253,95,
195,203,92,58,43,96,254,0,40,185,71,
33
```

180 DATA 255,255,43,43,43,229,35,35,197,229,213,205,225,95,1,15,0,17,239,96
 190 DATA 5578
 200 DATA 205,60,32,209,225,126,43,43,254,255,40,9,205,145,93,193,225,16,219,24
 210 DATA 142,6,10,94,35,86,213,213,205,234,95,225,221,225,205,253,95,254,106,40
 220 DATA 4052
 230 DATA 13,254,110,40,31,254,101,32,241,193,6,1,24,218,17,17,0,25,221,229
 240 DATA 229,17,22,97,1,19,0,205,60,32,225,221,225,205,112,9,24,193,94,35
 250 DATA 5393
 260 DATA 86,235,229,62,13,215,62,42,215,225,229,126,215,225,205,253,95,254,110,200
 270 DATA 254,106,40,7,254,101,32,242,193,24,194,229,17,22,97,1,19,0,205,60
 280 DATA 5170
 290 DATA 32,62,253,205,1,22,175,17,161,9,205,10,12,253,203,2,238,205,212,21
 300 DATA 62,255,225,35,94,35,86,35,229,221,225,205,194,4,201,205,225,95,17,224
 310 DATA 4269
 320 DATA 96,1,15,0,205,60,32,221,42,37,96,42,39,96,221,34,35,96,34,33
 330 DATA 96,175,119,221,229,209,43,114,43,115,43,221,35,221,35,221,35,221,229,209
 340 DATA 6304
 350 DATA 167,237,82,218,227,94,221,229,235,55,62,255,205,86,5,193,197,221,229,225
 360 DATA 167,237,66,32,17,33,18,0,229,235,167,237,82,235,225,221,229,193,9,229
 370 DATA 4911
 380 DATA 221,225,193,175,187,32,4,186,202,227,94,221,43,221,229,225,167,237,66,235
 390 DATA 33,64,97,52,126,42,37,96,119,35,115,35,114,221,34,37,96,33,39,96
 400 DATA 3455
 410 DATA 53,53,53,33,43,96,52,195,230,92,205,225,95,17,224,96,1,15,0,205
 420 DATA 60,32,221,42,37,96,42,39,96,6,221,34,35,96,34,33,96,17,20,0,221
 430 DATA 5818
 440 DATA 229,221,25,229,221,229,209,167,237,82,225,221,225,218,227,94,221,34,41,96

450 DATA 221,229,17,17,0,175,55,205,2,8,209,205,234,95,213,225,94,35,86,42
 460 DATA 5528
 470 DATA 39,96,229,221,229,193,3,3,3,167,237,66,237,82,225,218,227,94,221,229
 480 DATA 221,25,221,34,37,96,221,225,213,62,255,119,237,91,41,96,43,114,43,115
 490 DATA 3832
 500 DATA 43,34,39,96,209,55,205,2,8,33,43,96,52,195,230,92,17,43,97,1
 510 DATA 21,0,205,60,32,33,208,7,17,200,0,205,181,3,205,253,95,195,230,92
 520 DATA 3606
 530 DATA 42,33,96,221,42,35,96,34,39,96,221,34,37,96,17,200,0,33,200,0
 540 DATA 205,181,3,17,208,96,1,16,0,205,60,32,205,253,95,33,61,92,17,254
 550 DATA 4429
 560 DATA 91,115,35,114,235,249,59,17,251,94,115,35,114,195,230,92,229,213,42,39
 570 DATA 96,237,91,37,96,167,237,82,205,78,95,1,11,0,17,22,96,205,60,32
 580 DATA 4837
 590 DATA 209,225,201,17,16,39,62,47,167,60,237,82,48,251,25,215,1,0,0,229
 600 DATA 235,17,10,0,167,3,237,82,48,251,11,197,209,225,175,187,200,24,223,205
 610 DATA 3537
 620 DATA 225,95,17,44,96,1,20,0,205,60,32,58,43,96,254,0,40,86,71,33
 630 DATA 255,255,43,62,255,190,40,41,62,13,215,62,42,215,43,86,43,94,26,19
 640 DATA 4788
 650 DATA 197,229,213,215,1,2,0,17,40,97,205,60,32,62,32,215,225,94,35,86
 660 DATA 213,225,205,78,95,225,193,24,33,43,86,43,94,197,229,205,234,95,213,1
 670 DATA 4540
 680 DATA 2,0,17,40,97,205,60,32,62,32,215,225,94,35,86,213,225,205,78,95
 690 DATA 225,193,16,174,205,253,95,195,230,92,205,107,13,62,2,205,1,22,201,26
 700 DATA 3615
 710 DATA 245,213,17,192,9,205,10,12,209,19,1,10,0,205,60,32,241,201,175,50
 720 DATA 8,92,58,8,92,254,0,40,249,

```

201,11,96,16,39,232,3,100,0,10,0
730 DATA 1844
740 DATA 1,0,38,32,66,89,84,69,83,3
2,70,82,69,73,0,0,0,0,0,0
750 DATA 0,0,0,0,0,71,69,76,65,68,6
9,78,69,32,80,82,79,71,82,65
760 DATA 2009
770 DATA 77,77,69,58,13,22,0,9,32,6
7,32,79,32,80,32,89,32,32,73,32
780 DATA 84,13,13,13,49,46,41,32,76
,65,68,69,78,32,45,32,78,79,82,77
790 DATA 2370
800 DATA 65,76,13,50,46,41,32,76,65
,68,69,78,32,45,32,79,72,78,69,32
810 DATA 72,69,65,68,69,82,13,51,46
,41,32,65,66,83,80,69,73,67,72,69
820 DATA 2037
830 DATA 82,78,13,52,46,41,32,76,73
,83,84,69,13,53,46,41,32,67,76,69
840 DATA 65,82,13,54,46,41,32,69,78
,68,69,13,13,13,22,20,5,127,32,49
850 DATA 2195
860 DATA 57,56,52,32,66,89,32,85,82
,83,32,77,85,69,76,76,69,82,22,10
870 DATA 0,87,65,72,76,63,13,13,13,
22,10,8,32,76,65,68,69,70,69,72

```

```

880 DATA 2060
890 DATA 76,69,82,32,32,22,1,11,18,
1,76,79,65,68,73,78,71,18,0,13
900 DATA 22,1,9,83,97,118,101,40,74
,47,78,47,69,41,13,83,65,86,69,32
910 DATA 1792
920 DATA 22,11,9,20,1,83,80,69,73,6
7,72,69,82,32,32,70,82,69,73,22
930 DATA 1,9,32,32,18,1,83,65,86,73
,78,71,18,0,32,32,32,23,20,48
940 DATA 2764
950 DATA 22,11,7,20,1,80,82,79,71,6
5,77,77,32,90,85,32,76,65,78,71
960 DATA 33,47,33,0,192,17,203,92,1
,118,4,237,176,195,203,92,0,0,0,0
970 FOR i=49152 TO 50311 STEP 40
980 LET s=0: READ w
990 FOR x=0 TO 39
1000 READ a: LET s=s+a: POKE i+x,a
1010 NEXT x
1020 IF s<>w THEN LET l=(i-49152)/4
0: PRINT "Fehler in Zeilen ";100+30*
l;" bis ";120+30*l: PRINT s: STOP
1030 NEXT i
1040 PRINT "O.K."
1050 SAVE "COPYCODE"CODE 49152,1156
1060 VERIFY ""CODE

```

Bessere BASIC-Spiele programmieren? – Das RGH-BASIC von Wicosoft im Test! – Für den ZX-Spectrum

Sinclair bietet auf seinem Spectrum ein hervorragendes BASIC an, sowohl bezüglich der Bedienerfreundlichkeit als auch der Leistungsfähigkeit. Dennoch werden zahlreiche preiswerte Erweiterungen angeboten.

Das RGH stellt keine Erweiterung des Sinclair-BASIC's dar, sondern einen eigenen Interpreter. Das Programm wird auf Kassette geliefert; dazu gehört ein deutschsprachiges Handbuch.

Nachdem sich das Microdrive und andere Massenspeicher durchzusetzen beginnen, ist natürlich die Frage der Kompatibilität zu beantworten.

Das Programm belegt im Speicher genau den Bereich, den viele Massenspeicher für ihre Maps benötigen. Es kann jedoch mit einem Trick auf Cartidge gespeichert werden. Wie man das macht, steht im Handbuch. Allerdings stellt sich hier ernsthaft die Frage, ob es lohnt: selbst mit einer sehr schnellen Floppy ist kein übermäßiger Vorteil zu erzielen; der Interpreter ist sehr kurz.

Dieses Programm soll lt. in der Anleitung angegebener Benchmark-Tests um vieles schneller sein als das Sinclair-BASIC. Dies wird – wie im Handbuch zu entnehmen ist – dadurch erreicht, daß das

Programm nicht mit Ballast vollgestopft wurde. Die Einschränkungen im wesentlichen:

- Ganze Zahlen bis 32768
- Buchstabe und Buchstabe + Zahl als Variablenamen, jedoch wird Groß- und Kleinschreibung (anders als bei allen anderen Dialekten!) unterschieden
- keine Tokens oder Funktionstasten (die der Sinclair sonst nur verwendet!)
- einen absoluten Mini-Editor.

Die Einschränkung der Variablenamen fällt m. E. nicht so sehr ins Gewicht, ebenso die Begrenzung der Zahlen auf ganze Zahlen. Bei Spiel-

programmen – und dafür ist dieser Interpreter gedacht – ist es durchaus möglich, damit auszukommen.

Die ersten Gehversuche waren nicht ganz einfach. Ich habe allerdings dabei noch einen Vorteil: mir ist nicht nur das Sinclair-BASIC geläufig, sondern auch zahlreiche andere Versionen. Dabei fiel mir sehr schnell die wirklich langsame Tastaturabfrage unangenehm auf. Wer viel tippt, hat ein gewisses Tempo und wird hier leicht zu schnell für den „Editor“. Diese Geschwindigkeit der Abfrage kann allenfalls bei einem Kleinstrechner akzeptiert werden, bei größeren Com-

putern sollte man mehr erwarten dürfen.

Dieser Nachteil wurde noch dadurch verstärkt, daß entgegen der üblichen Spectrum-Art die BASIC-Wörter nun ausgeschrieben werden müssen. Das Eintippen eines längeren Listings kann wirklich zur Nervenprüfung werden. Eine ganze Reihe von Befehlen gibt es quasi zusätzlich. Neben dem bekannten BEEP-Kommando steht nun auch der SOUND-Befehl zur Verfügung. Mit 5 Parametern können damit alle möglichen Arten von Tönen erzeugt werden. Die Handhabung der Parameter ist etwas kompliziert.

Es gibt beispielsweise auch Spezialbefehle für die ganzzahlige Division ohne Rest, Erhöhung der Punktzahl oder anderer Variablenwerte (a MOD b, SCORED0, DECP1) und einiges mehr. Die Definition der frei belegbaren Zeichen (UDG's) ist vereinfacht. Der Befehl heißt einfach CHAR (code) = <Werteliste>.

Die Syntax einiger Befehle hat sich eindeutig verschlechtert. Im Spectrum-BASIC rechnet üblicherweise der

Computer die Farbwerte zusammen, die sich aus den Befehlen PAPER, INK, FLASH, BRIGHT ergeben. Dafür gibt es jetzt den Befehl COLOUR, gefolgt von einer Zahl, die alles auf einmal enthält. Es muß also gerechnet werden.

Ausgesprochen unfreundlich und vor allem für Anfänger sehr schwer zu überblicken ist die Stringablage. Zeichenketten werden am Ende des Programms mit in den Speicher geschrieben und mit Hilfe der Startadresse aufgerufen. Dies kann durch Benutzung von Variablen für diese Adressen zwar vereinfacht werden; jede zusätzliche Variable macht das Programm aber langsamer (grundsätzlich).

Meine Erwartungen wurden bisher weit unterboten. Aber wenn das Programm wirklich so schnell interpretiert, würde sich die erhebliche Umstellung ja lohnen.

Also habe ich eines der Beispiellistings aus dem Handbuch ausprobiert. Die Nachteile des „Spitzen-Editors“ kamen dabei trotz der Kürze des Programms voll zum Tragen. Eine fehlerhafte Zeile

muß vollständig neu geschrieben werden! Hoffentlich war sie kurz.

Bei der unvermeidlichen Fehlersuche (und seien es „nur“ Tippfehler) fiel die Möglichkeit eines Druckerlistings positiv auf. Normalpapierdrucker sollten auf den englischen Zeichensatz eingestellt sein. Vom Programm aus können ZX-Printer, Alphacom und Seikosha GP50S sowie alle Drucker über RS232 angesprochen werden. Centronics-Interfaces bzw. deren Ansteuerung sind nirgendwo erwähnt.

Nachdem das Programm endlich fehlerfrei war, konnte es erst einmal abgespeichert werden. Dafür steht nur der Kassettenrecorder zur Verfügung. Die Programme werden ohne Kopf abgelegt, können also ohne RGH-BASIC nur von Knack-Spezialisten geladen werden (beispielsweise um ein Druckerlisting zu erstellen, wenn der Drucker sich mit RGH-BASIC „beißt“).

Dann wollte ich gerne den Lohn für meine Mühe kassieren, d. h. mich beim Spielen so richtig austoben. Nun, auch hier wurde ich leider

enttäuscht. Die langsame Tastatur-Abfrage macht sich auch während des Spielens stark bemerkbar, so daß sich schnell Verkrampfungen einstellen.

Zum echten Vergleich gehört auch eine „Übersetzung“ des Programmes in Sinclair-BASIC. Wie die beiden Listings zeigen, habe ich beim Übersetzen den Spielablauf in keiner Weise geändert. Die Sinclair-Version war jedoch angenehmer zu spielen, weil die Tasten auf normalen Druck sofort reagierten. Außerdem war das Spiel erheblich schwieriger und damit reizvoller geworden.

Es ist natürlich durchaus möglich, daß ich das falsche Beispiel-Listing ausgewählt habe und die wirkliche Schnelligkeit des Interpreters erst bei anderen Programmen voll zum Tragen kommt. Die hervorragenden Editiermöglichkeiten werden aber vor allem bei längeren Programmen entscheidend zu Unlustgefühlen beitragen.

Für seine Leistungen ist dieses Programm – vor allem auch im Vergleich zu anderen Hilfsmitteln – entschieden zu teuer.

Beispiellisting: 1

```

10 REM MINI-INVADERS
20 REM RGH-BASIC
30 RANDOMIZE 99
40 BORDER 0: COLOUR 15: CLS
50 CLEAR -200
51 REM FUER STRINGABLAG
60 SCORE0: LET S=1, L=3
70 CHAR(144)=129,126,165,255,2
55,66,129,66
80 CHAR(145)=0,0,16,16,56,56,2
54,238
90 $(TOP)=" ";145;" "
100 LET $(TOR)="
"
105 LET S1=TOR, S2=TOR+40, $(S2)
)=$(S1)
110 FOR I=1 TO S
120 POKE TOR+5+RND(20),144
130 NEXT I
150 LET P1=0, P2=688
1000 POS P1: PRINT $(S2); $(S1)
1030 FOR I1=1 TO 2
1040 LET K=INKEY
1060 IF K='5' DEC P2: IF P2<672
LET P2=672
1065 BEEP 1,1;1,20
1070 IF K='8' INC P2: IF P2>701
THEN LET P2=701

```

```

1075 IF K='7' GOSUB 2000
1080 POS P2: PRINT $(TOP)
1085 POS 0: PRINT "SCORE:";#4;SCORE;
" WELLE:";#2;S;" LEBEN:";L
1090 NEXT I
1100 IF RND(2)=1 LET P1=P1+32
1200 IF P1>640 GOTO 3000
1210 IF $(S1)="
" GOTO 1500
1220 GOTO 1000
1500 CLS: INC S: GOTO 100
2000 PLOT P3,8: GRAPH ^160
2020 OVERON: PLOT P3,8: GRAPH ^1
60
2025 OVEROFF
2030 IF PEEK(P2-671+S1)=144 POKE
P2-671+S1,32: SCORE=SCORE+32-P1
/32: SOUND 1,1,2,5,200
2090 RETURN
3000 DEC L: IF L CLS: GOTO 100
3010 POS 96: PRINT " S P I E L E
N D E "
3020 PRINT : PRINT
3030 INPUT "NOCH EINMAL? ";$(S1)
3040 IF $(S1)="NEIN" STOP
3050 GOTO 10
10000 SAVE
10010 GOTO 10

```

Beispiellisting: 2

```

10 REM MINI-INVADERS
20 REM SINCLAIR-BASIC
30 RANDOMIZE
40 BORDER 0: PAPER 1: INK 7: C
LS
50 RESTORE 60
51 FOR I=USR "A" TO USR "B"+7
52 READ W: POKE I,W
53 NEXT I
60 DATA 129,126,165,255,255,66
,129,66,0,0,16,16,56,56,254,238
70 LET SCORE=0: LET S=1: LET L
=3
90 LET B$=" "+CHR$ 145+" "
100 LET A$="
"
105 LET C$=A$
110 FOR I=1 TO S
120 LET A$(5+(RND*20))=CHR$ 144
130 NEXT I
150 LET P1=0: LET P2=16
1000 PRINT AT P1,0;C$;A$
1030 FOR I=1 TO 2
1040 LET I$=INKEY$
1050 IF I$="5" THEN LET P2=P2-1:
IF P2<0 THEN LET P2=0
1055 BEEP .01,1: BEEP .01,20
1070 IF I$="8" THEN LET P2=P2+1:
IF P2>29 THEN LET P2=29
1075 IF I$="7" THEN GO SUB 2000

```

```

1080 PRINT AT 21,P2;B$
1085 PRINT AT 0,0;"SCORE: ";
"( TO 4-LEN STR$ SCORE);SCORE;"
WELLE: "; " ( TO LEN STR$ S);S
;" LEBEN: ";L
1090 NEXT I
1100 IF RND>=.5 THEN LET P1=P1+1
1200 IF P1>20 THEN GO TO 3000
1210 IF A$="
" THEN GO TO 1500
1220 GO TO 1000
1500 CLS: LET S=S+1: GO TO 100
2000 LET P3=8*P2+4+8
2010 PLOT P3,8: DRAW 0,160
2020 OVER 1: PLOT P3,8: DRAW 0,1
60
2030 IF A$(P2+2)+CHR$ 144 THEN L
ET A$(P2+2)=" ": LET SCORE=SCORE
+32+P1: BEEP .5,12
2090 RETURN
3000 LET L=L-1: IF L THEN CLS: G
O TO 100
3010 PRINT AT 3,0;" S P I E L E
N D E "
3020 PRINT : PRINT
3030 INPUT "NOCH EINMAL ";X$
3040 IF X$="NEIN" THEN STOP
3050 GO TO 70
9000 SAVE "INVADER" LINE 10

```

Sony stellt seinen kleinsten Compact-Disc-Player vor

Den kleinsten CD-Player der Welt stellte Sony in Deutschland vor. Er wurde erstmals auf der Audio Fair in Tokio Ende Oktober präsentiert und galt dort als „Revolution“ in der Weiterentwicklung der Digitaltechnik. Der D-50 wiegt nur 590 Gramm und ist kleiner als die Hülle einer Compact Disc: nämlich 12,7 cm x 3,7 cm x 13,3 cm (B x H x T). Die Wiedergabe kann wahlweise über Kopfhörer, Aktivboxen oder eine herkömmliche HiFi-Anlage erfolgen. Für den Netzbetrieb wird der Adapter AC-D 50 mitgeliefert. Im Auto kann der D-50 über den Zigarettenanzünder mit der Auto-Batterie betrieben werden. Durch ein netzunabhängiges Battery-Pack mit Tragetasche wird er sogar portabel und ist damit überall einsetzbar. Bezüglich seiner Klangqualität und Ausstattung kann sich

der CD-Winzling mit jedem „großen“ stationären CD-Player messen.

Der neue D-50 hat einen Dynamikbereich von 90 dB und einen sehr niedrigen Klirrfaktor von 0,008 Prozent. Damit ist der Mini-CD-Player von seiner Klangqualität her mit stationären Compact Disc-Playern durchaus vergleichbar.

Technische Daten D-50

Abtastung: berührungsfreie Optik (Laserdiode).
 Umdrehungen: 500-200 U/Min. (konstante lineare Geschwindigkeit).
 Fehlerkorrektur: Sony-Super Strategy (Cross-Interleave Reed Solomon-Code).
 Anzahl der Kanäle: 2.
 Quantisierung (D/A): 16 Bit linear.
 Frequenzgang: 20 Hz - 20 kHz (+1/-3 dB).

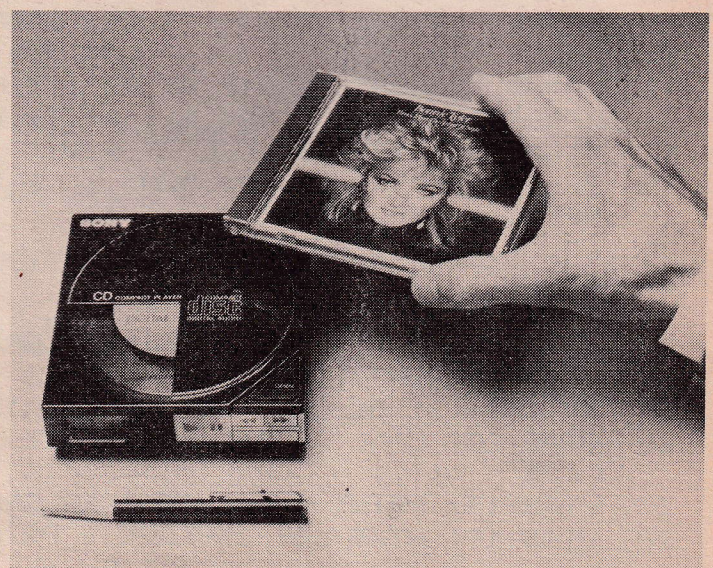
Kanaltrennung: über 85 dB (1 kHz).

Gleichlaufschwankungen: nicht meßbar.

Spieldauer: ca. 60 Minuten (max. 74 Minuten) (Software-abhängig).

Spannungsversorgung: DC 9 V

- sechs 1,5 V Babyzellen (in Verbindung mit dem auf Wunsch lieferbaren Batteriebehälter EBP-9 LC)



„DOS Change“

Für den Apple II

Das RAM des Apple II enthält bei Verwendung von Disketten-Laufwerken das Betriebssystem DOS. Das vorliegende Programm bietet die Möglichkeit, die DOS-Befehle des Apple umzubenennen.

So kann aus „LOAD“ „LADE“, aus „OPEN“ „OEFFNE“ werden. Man kann auch Abkürzungen wählen, für „CATALOG“ zum Beispiel „CAT“.

Wenn man das Programm startet, werden zunächst die momentan gültigen DOS-Befehle aus dem RAM gelesen. Diese werden dann auf dem Bildschirm ausgegeben.

Unter der Befehlsliste erscheint ein Menü. Dieses

bietet folgende Möglichkeiten:

– Ändern eines Befehls (Edit): Nach Angabe der Befehlsnummer (diese stehen vor den Befehlsworten) wird eine Linienfolge ausgegeben. Die Länge dieser Linie stimmt mit der maximalen Länge des neuen Befehls überein. Nun wird der neue Befehl eingegeben. Drückt man nur RETURN, so wird keine Änderung vorgenommen.

– Einpoken ins RAM (Sichern):

Diese Funktion speichert die neue Befehlsliste im DOS-RAM-Bereich ab. Ab sofort sind diese Befehle gültig.

Mit <E> (Edit) vorgenommene

Änderungen werden erst durch <S> (Sichern) wirksam.

– Einlesen aus dem RAM (Laden):

Es werden die momentan gültigen DOS-Befehle aus dem RAM gelesen. Anwenden kann man diese Funktion, um umfassende Änderungen rückgängig zu machen.

– Original-Befehle laden <O>: Nach Drücken von <O> werden die Original-Apple-DOS-Befehle aus den DATA-Zeilen gelesen. Nach <S> (Sichern) sind diese Befehle wieder gültig, da sie dann wieder ins RAM geschrieben sind.

– Informationen <?>:

<?> gibt auf dem Bildschirm

eine Kurzinformation über die Programmteile aus.

– Programmende (<Q> Quit): Um das Programm zu verlassen, drückt man vom Menü aus <Q>. Nach einer Sicherheitsabfrage wird das Programm beendet. Nun kann man die geänderten Befehle testen. Um die Änderungen auch nach dem Ausschalten des Computers zu bewahren, kann man nun eine Diskette mit dem DOS initialisieren.

Das Programm DOS CHANGE läuft auf dem Apple II mit 48/64 k Byte und mindestens einem Diskettenlaufwerk sowie DOS 3.3.

```

1  REM          DOS CHANGE

2  REM          COPYRIGHT
3  REM          BY
4  REM          OLIVER STEINMEIER

5  DIM CO$(27): REM  FUER 28 COMMANDS
6  LAENGE% = 132
7  TEXT : HOME : NORMAL
8  INVERSE : PRINT "          D O S   C H A N G E           ";; NORMAL
9  PRINT "  CO~YRIGHT 1984 BY OLIVER STEINMEIER  " : POKE 34,2
10 GOSUB 38: REM  BEFEHLE AUSLESEN
11 GOSUB 47: REM  BEFEHLSLISTE AUSGEBEN
12 VTAB 17: PRINT "<E> EDIT      -- <?> INFO"
13 PRINT "<S> SICHERN  -- <L> LADEN"
14 PRINT "<Q> QUIT      -- <O> ORIGINAL-BEFEHLE"
15 PRINT : PRINT "FREIE BYTES : ";LAENGE% - LE%
16 VTAB 23: PRINT TAB( 1);"BITTE WAEHLEN : ";; GET U$
17 VTAB 17: HTAB 1: CALL - 958
18 IF U$ = "Q" THEN 64: REM  ENDE
19 IF U$ = "S" THEN 54
20 IF U$ = "L" THEN RUN
21~ IF U$ = "O" THEN 71
22 IF U$ = "?" THEN 81
23 IF U$ < > "E" THEN PRINT CHR$( 7): GOTO 12
24 VTAB 20: INPUT "BEFEHLS-NUMMER : " ;CN%
25 IF CN% < 1 OR CN% > 28 THEN PRINT CHR$( 7): GOTO 24
26 VTAB 20: CALL - 868:LE% = LE% - LEN (CO$(CN% - 1)):SP% = LAENGE% -
  LE%
27 VTAB 17: PRINT "VERFUEGBARER SPEICHERPLATZ : ";SP%;" ZEICHEN"
28 VTAB 19: HTAB 10: INVERSE : PRINT CO$(CN% - 1): NORMAL

```

```

29 VTAB 22: CALL - 868: PRINT TAB( 1);"BEFEHL : ";
30 FOR I = 1 TO LAENGE% - LE%: PRINT CHR$( 95);: NEXT
31 VTAB 22: HTAB 10: INPUT "";C$
32 IF LEN (C$) = 0 THEN LE% = LE% + LEN (CO$(CN% - 1)): GOTO 11
33 IF LEN (C$) > SP% THEN PRINT CHR$( 7); CHR$( 7): GOTO 29
34 IF LEN (C$) = 1 THEN CO$(CN% - 1) = CHR$( ASC (C$) + 128): GOTO 36
35 CO$(CN% - 1) = LEFT$( C$, LEN (C$) - 1) + CHR$( ASC ( RIGHT$( C$,1)
    ) + 128)
36 LE% = LE% + LEN (C$): GOTO 11
37 END
38 REM  COMMANDS AUS RAM LESEN

39 VTAB 12: PRINT TAB( 15);"BITTE WARTEN"
40 BE% = 0:LE% = 0
41 FOR I = 43140 TO 43272
42 CH% = PEEK (I): IF CH% = 0 THEN RETURN
43 CO$(BE%) = CO$(BE%) + CHR$( CH%)
44 IF CH% > 127 THEN BE% = BE% + 1:LE% = LE% + LEN (CO$(BE% - 1))
45 NEXT
46 RETURN
47 REM  BEFEHLSLISTE AUSGEBEN

48 HOME : VTAB 4: PRINT "A K T U E L L E  D O S - B E F E H L E"
49 VTAB 6: FOR I = 0 TO 26 STEP 3
50 PRINT I + 1;": ";CO$(I); TAB( 14);I + 2;": ";CO$(I + 1); TAB( 28);I +
    3;": ";CO$(I + 2)
51 NEXT
52 PRINT "28: ";CO$(27)
53 RETURN
54 REM  BEFEHLE INS RAM POKEN

55 HOME : VTAB 12: PRINT TAB( 15);"BITTE WARTEN"
56 J = 43140
57 FOR I = 0 TO 27
58 FOR K = 1 TO LEN (CO$(I))
59 POKE J, ASC ( MID$( CO$(I),K,1))
60 J = J + 1
61 NEXT : NEXT
62 POKE J,0: REM  ENDE DES COMMAND-TABLES

63 GOTO 11
64 REM  ENDE

65 HOME : VTAB 12: PRINT "WOLLEN SIE DAS PROGRAMM VERLASSEN ? ";: GET U$
66 IF U$ < > "J" AND U$ < > "Y" THEN 11
67 HOME : TEXT : VTAB 10
68 PRINT "UM DIE NEUEN DOS-BEFEHLE WEITERHIN": PRINT "VERWENDEN ZU KOENN
    EN, SOLLTEN SIE NUN"
69 PRINT "EINE DISKETTE ZU INITIALISIEREN.": PRINT : PRINT "BENUTZEN SIE
    RUN, UM DIESES PROGRAMM": PRINT "ERNEUT ZU STARTEN !"
70 VTAB 20: END
71 REM  ORIGINAL-BEFEHLE EINLESEN

72 RESTORE
73 HOME : VTAB 12: PRINT TAB( 15);"BITTE WARTEN"

```

```

74 FOR I = 0 TO 27: READ U#
75 CO$(I) = LEFT$(U$, LEN (U$) - 1) + CHR$( ASC ( RIGHT$(U$,1)) + 12
  8)
76 NEXT :LE% = LAENGE%
77 GOTO 11
78 DATA INIT,LOAD,SAVE,RUN,CHAIN,DELETE,LOCK,UNLOCK,CLOSE,READ,EXEC,WRIT
  E
79 DATA POSITION,OPEN,APPEND,RENAME,CATALOG,MON,NOMON,PR#,IN#,MAXFILES,
  FP,INT
80 DATA BSAVE,BLOAD,BRUN,VERIFY
81 REM INFORMATIONEN

82 HOME : VTAB 5: PRINT "      I N F O R M A T I O N E N      "
83 PRINT : PRINT "<E> EDIT : VERAENDERN EINES DOS-BEFEHLS"
84 PRINT : PRINT "<S> SICHERN : EINPOKEN DER NEUEN DOS-BE-FEHLE INS DOS"

85 PRINT : PRINT "<L> LADEN : AUSLESEN DER BEFEHLE AUS DEMDOS"
86 PRINT : PRINT "<O> ORGINAL : EINLESEN DER ORGINAL-DOS- BEFEHLE"
87 PRINT : PRINT "<Q> QUIT : PROGRAMM BEENDEN (MOEGLICH- KEIT ZUM TESTE
  N DER BEFEHLE)
88 PRINT : PRINT "<?> INFO : ZEIGT DIESE INFORMATIONEN"
89 PRINT : PRINT : PRINT "DRUECKEN SIE EINE TASTE ! ";: GET U#
90 GOTO 11

```

CPU-Clubvorstellung: Sinclair-Benutzer-Club

Anschrift: Erika Hölscher - Ernst-August-Str. 5 - 2730 Zeven

**Nutzen Sie die kostenlose Möglichkeit zur Vorstellung Ihres
Computer-Clubs in der CPU!**

★ Monatlich erscheint unser Clubinfo mit Tips, Erfahrungsberichten, Kontakt-, Verkaufs- oder Kaufwünschen, Buchbesprechungen, Geräte- und Programmtests, Leserbriefen usw. Sollte ein Mitglied sein Info nicht bis Mitte des laufenden Monats vorliegen haben, so melde es sich bei der Clubverwaltung. Wir machen dies ehrenamtlich, also nach Feierabend, und da können solche Pannen leider mal passieren.

★ Die Clubmitglieder können mit dem Club oder auch untereinander Programme tauschen. **Wir weisen an dieser Stelle ausdrücklich darauf hin, daß urheberrechtlich geschützte Programme bei uns nicht getauscht werden und**

notfalls Schädiger mit uns zusammenarbeitender Firmen aus dem Club ausgeschlossen werden.

★ Die Bildung von Ortsgruppen ist geplant, sofern genügend Interesse besteht.

★ Es finden in regelmäßigen Abständen Clubtreffen statt. Für die von Zeven weiter entfernt Wohnenden gibt es teilweise die Möglichkeit, bei Treffen befreundeter Clubs zu Gast zu sein.

★ Der Club hat eine wachsende Leihbücherei. Diese Bücher können für maximal 4 Wochen ausgeliehen werden. Wir freuen uns natürlich auch, wenn ein Mitglied seine Bücher zur Verfügung stellen

möchte. Diese müssen nicht an die Clubleitung geschickt werden!

★ Ein Austritt aus dem Club ist jederzeit möglich. Monatsbeiträge, die schon im voraus entrichtet wurden, werden dann erstattet. Weitere Ansprüche bestehen nicht. Die Clubleitung behält sich vor, ein Mitglied vom Club auszuschließen, wenn dieses in grober Weise gegen die Satzung verstößt oder begründeter Verdacht besteht, daß es mit Raubkopien handelt.

★ Die Aufnahmegebühr beträgt 10,-, der Monatsbeitrag DM 5,-. Darin sind alle Clubleistungen enthalten außer Kopien von Programmbeschreibungen. Die Mitgliedschaft beginnt mit dem Eingang des Beitrages.

★ Der Club hat vergünstigte Einkaufsmöglichkeiten, die zum Teil meiner Tätigkeit als Journalistin zu verdanken sind, aber auch dem Einsatz eines Mitglieds, das unermüdlich Firmen anschreibt und prüft, ob diese Firma auch seriös ist. Die im Info angegebenen Adressen sind keine Reklame! Wir bekommen dafür kein Geld von irgendwem.

★ Die Mitgliedschaft beginnt mit Eintreffen der ersten Beitragszahlung. Das neue Mitglied erhält sofort unsere Clubsatzung, in der die genauen Verfahrensweisen für Programmtausch, Bücherverleih u. a. ausführlich erläutert sind.

CPC-464 Inside Out

Ein Buch für Programmierer

Von Winfried Huslik, erschienen im EDV-Buchverlag Augsburg
ISBN 3 - 925159 - 002 Preis: 59,- DM

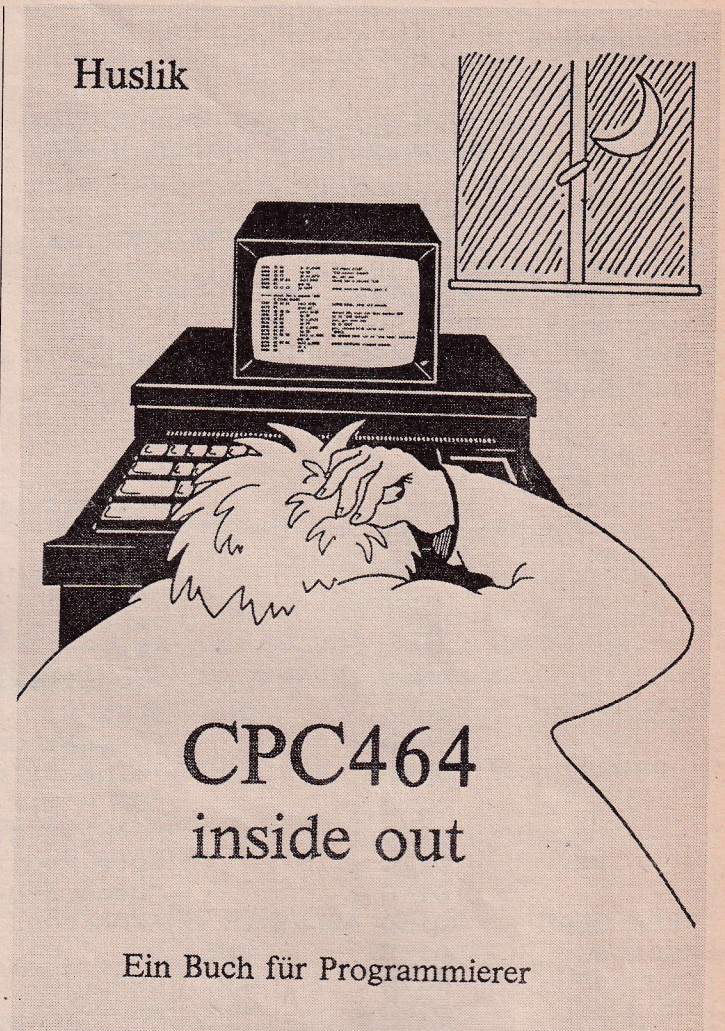
Endlich ein Buch, das die Geheimnisse des CPC-464 enthüllt. Neben den ausführlich dokumentierten ROM-Listings enthält es auch viele Tips, um noch mehr Möglichkeiten für den Schneider-Computer-Besitzer zu offenbaren. Einziges Manko: Leider sind die ROM-Listings in Englisch festgehalten, was nicht gerade zur Verständlichkeit beiträgt.

Mit Hilfe der Kommentare sollte es Ihnen möglich sein, die wichtigsten Abläufe und Funktionen im Betriebssystem und im Basic zu verstehen. Eine Vielzahl von Informationen, zum Beispiel die Syntax bei Basic-Anweisungen oder die Angabe der verwendeten Register, soll Sie dabei unterstützen. Die stack-orientierte Darstellung (das Ein- und Ausrücken bei Push- und Pop-Befehlen) macht die Programme lesbarer und betont die Stellen des Programms, an denen echte Aktionen stattfinden. Die bei jeder Routine angegebenen

Referenzadressen ermöglichen es, das Programm auch nach rückwärts zu verfolgen. Dieses Buch kann und soll weder ein Handbuch noch eine Hardware-Beschreibung ersetzen. Bei der Komplexität der Zusammenhänge ist es ohnehin ratsam, so viele Informationsquellen wie nur möglich zu Rate zu ziehen. Fazit: Trotz des stattlichen Preises ein empfehlenswertes Nachschlagewerk für ernsthafte Programmierer.

Einige Kapitel dieser Ausgabe:

Das RAM des CPC-464
Zugriff auf ROM und RAM
Das Basic
Das Basic des CPC-464
Funktion des Disassemblers
ROM-Listing des Betriebssystems
RAM-Listing AB80 - BFFF
ROM-Listing des Basic
Index
Disassembler Tables
Kleines Fachwörterbuch
Feed-Back Sheets
Tabellen innerhalb des



ROM-Listings
Error Messages
Arithmetische Funktionen
Basic-Funktionen ohne Argumente
Basic-Funktionen mit Argumenten

Basic-Kommando Adressen
Basic-Kommando Token für Interpreter
Direkte Basic-Kommandos
Adressen der Anfangsbuchstaben

„FILL“-Programm Für den ZX-Spectrum (48K)

Das Programm erweitert die Grafikfähigkeit des ZX Spectrums, es malt zusammenhängende Flächen auf dem Bildschirm aus. Das Programm ist vollständig in Maschinensprache geschrieben. Der MC-Code liegt in dem Adressbereich von 65119 bis 65279.

Der Aufruf geschieht durch „DRAW 0,0,USR 65153“. Die Funktionen INK, PAPER, FLASH und BRIGHT

können mitbenutzt werden. Die auszumalende Fläche wird durch den zuletzt geplotteten Punkt bestimmt. Zur Eingabe des MC-Programms wird zuerst die Eingaberoutine eingetippt. Der MC-Code wird in hexadezimaler Form in die Variable a\$ geschrieben (Achtung! Es werden nur Zahlen und Großbuchstaben richtig erkannt!). Das Programm sollte erst einmal gesaved werden,

bevor man es mit RUN startet. Falls man einen Fehler bei der Eingabe des Hex-Codes begangen hat, meldet sich das Programm mit „Fehler“. Andernfalls erscheint „Please start the tape“ um das MC-Programm abzuspeichern. Man kann nun den Vorspann (das Programm mit den vielen DRAW-Befehlen) eintippen und save. Der Vorspann wird zuerst geladen. Dieser schreibt in großen Buchsta-

ben „FILL“ auf den Bildschirm und lädt anschließend den MC-Code. Hiernach kann der Vorspann durch NEW gelöscht werden und die FILL-Routine ganz normal durch DRAW 0,0,USR 65153 aufgerufen werden. Es ist auch möglich den MC-Code ohne den Vorspann zu laden, man muß nur vorher CLEAR 65118 (Wichtig!) und anschließend LOAD „FILL“, CODE eingeben.

Eingaberoutine

```

10 CLEAR 65118
20 LET a$="3EAF92... Hier den MC-Code eingeben"
30 IF LEN a$<>322 THEN PRINT "Fehler": STOP
40 LET x=0
50 FOR i=65119 TO 65279
60 POKE i,16*(CODE a$-48-7*(CODE a$>=58))+CODE a$(2)-48-7*(CODE a$(2)>=58)
70 LET a$=a$(3 TO )
80 LET x=x+PEEK i
90 NEXT i
100 IF i<>18830 THEN PRINT "Fehler": STOP
110 SAVE "FILL"CODE 65119,161

```

	0010 ;		*****
	0020 ;		***
	0030 ;		***
	0040 ;		***
	0050 ;		***
	0060 ;		***
	0070 ;		*****
FE5F	0080	ORG	OFE5FH
	0090 ;		"Dieses Untrogramm rechnet x,y Koordinaten in "
	0100 ;		"eine Bildschirmadresse um "
	0110 ;		"DE = Y,X Pos. HL = Adr. B = Bit Nr. A=(HL) "
	0120 ;		"Z-Flag gesetzt falls alle Bits = 0 "
	0130 ;		"Die Routine wurde aus dem ROM uebernommen und "
	0140 ;		"ueberfluessige Teile gestrichen "
FE5F 3EAF	0150	ADRES LD	A,OAFH
FE61 92	0160	SUB	D
FE62 47	0170	LD	B,A
FE63 A7	0180	AND	A
FE64 1F	0190	RRA	
FE65 37	0200	SCF	
FE66 1F	0210	RRA	
FE67 A7	0220	AND	A
FE68 1F	0230	RRA	
FE69 A8	0240	XOR	B
FE6A E6F8	0250	AND	OFE8H
FE6C A8	0260	XOR	B
FE6D 67	0270	LD	H,A
FE6E 7B	0280	LD	A,E
FE6F 07	0290	RLCA	
FE70 07	0300	RLCA	
FE71 07	0310	RLCA	
FE72 A8	0320	XOR	B
FE73 E6C7	0330	AND	OC7H
FE75 A8	0340	XOR	B
FE76 07	0350	RLCA	
FE77 07	0360	RLCA	
FE78 6F	0370	LD	L,A
FE79 7B	0380	LD	A,E
FE7A E607	0390	AND	O7H
FE7C 47	0400	LD	B,A
FE7D 04	0410	INC	B

Vorspann bitte zuerst, vor Eingaberoutine, abspeichern!

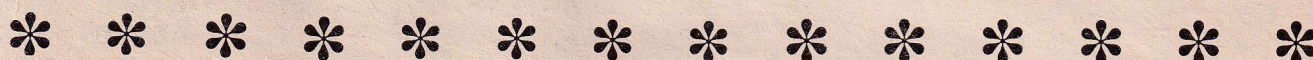
Vorspann

10 CLEAR 65118	230 DRAW 0,-72
20 PLOT 3,32	240 DRAW -8,0
30 DRAW 0,120	250 DRAW 0,-24
40 DRAW 64,0	260 PLOT 121,32
50 DRAW 0,-24	270 DRAW 64,0
60 DRAW -32,0	280 DRAW 0,24
70 DRAW 0,-24	290 DRAW -32,0
80 DRAW 32,0	300 DRAW 0,96
90 DRAW 0,-24	310 DRAW -32,0
100 DRAW -32,0	320 DRAW 0,-120
110 DRAW 0,-48	330 PLOT 188,32
120 DRAW -32,0	340 DRAW 64,0
130 PLOT 70,32	350 DRAW 0,24
140 DRAW 48,0	360 DRAW -32,0
150 DRAW 0,24	370 DRAW 0,96
160 DRAW -8,0	380 DRAW -32,0
170 DRAW 0,72	390 DRAW 0,-120
180 DRAW 8,0	400 PRINT AT 19,0;
190 DRAW 0,24	410 INK 7
200 DRAW -48,0	420 INK 7
210 DRAW 0,-24	430 LOAD "FILL"CODE
220 DRAW 8,0	440 INK 0
	450 PLOT 0,0
	460 PRINT AT 20,0;"
	470 DRAW 0,0,USR 65153

```

FE7E 7E      0420      LD   A,(HL)
FE7F A7      0430      AND  A
FE80 C9      0440      RET
              0450 ; "Hier Einstieg in die FILL-Routine
FE81 F3      0460 START DI
FE82 CD87FE  0470      CALL RES
FE85 FB      0480      EI
FE86 C9      0490      RET
              0500 ; "Das Start-Pixel (5C7DH der Systemvar.) loeschen"
FE87 ED5B7D5C 0510 RES  LD   DE,(5C7DH)
FE8B CD5FFE  0520      CALL ADRES
FE8E 3EFE    0530      LD   A,0FEH
FE90 0F      0540 R1   RRCA
FE91 10FD    0550      DJNZ R1
FE93 A6      0560      AND  (HL)
FE94 77      0570      LD   (HL),A
              0580 ; "X,Y Position (DE) und Zaehler (C) retten
FE95 D5      0590 FILL PUSH DE
FE96 C5      0600      PUSH BC
FE97 0E01    0610      LD   C,01H
              0620 ; "Dieser Programmteil sucht das ausserste linke
              0630 ; "gesetzte Pixel (X-Koordinate = Reg. E)
FE99 AF      0640 F1   XOR  A
FE9A BB      0650      CP   E
              0660 ; "X-Koordinate = 0 ... letzte Positin und Sprung
FE9B 2816    0670      JR   Z,F4
              0680 ; "X-Koordinate - 1
FE9D 1D      0690      DEC  E
FE9E CD5FFE  0700      CALL ADRES
              0710 ; "Pixel gesetzt ... Sprung
FEA1 2006    0720      JR   NZ,F2
              0730 ; "X-Koordinate - 7
FEA3 7B      0740      LD   A,E
FEA4 E6FB    0750      AND  0F8H
FEA6 5F      0760      LD   E,A
FEA7 18F0    0770      JR   F1
FEA9 07      0780 F2   RLCA
FEAA 10FD    0790      DJNZ F2
              0800 ; "Pixel nicht gesetzt ... Sprung
FEAC 30EB    0810      JR   NC,F1
FEAE 0D      0820      DEC  C
              0830 ; "Routine zum fuellen einer Zeile bis zum ersten
              0840 ; "rechten gesetztem Pixel
FEAF 0C      0850 F3   INC  C
              0860 ; "X-Koordinate + 1
FEB0 1C      0870      INC  E
              0880 ; "X-Koordinate = 255 ... Sprung
FEB1 2821    0890      JR   Z,F8
FEB3 CD5FFE  0900 F4   CALL ADRES
              0910 ; "Pixel gesetzt ... Sprung
FEB6 2011    0920      JR   NZ,F6
              0930 ; "B Pixel setzen
FEB8 3D      0940      DEC  A
              0950 ; "Reg. A = 255
FEB9 77      0960      LD   (HL),A
FEBA 3E07    0970      LD   A,07H
FEBC B3      0980      OR   E
FEBD 5F      0990      LD   E,A
              1000 ; "X-Koordinate + 7

```



FEBE 3E07	1010	LD	A,07H	
FECO 81	1020	ADD	C	
FEC1 4F	1030	LD	C,A	
	1040 ;			"Attribute (PAPER, INK, FLASH, BRIGHT) setzen"
FEC2 D5	1050 F5	PUSH	DE	
FEC3 CDD80B	1060	CALL	OBDBH	
FEC6 D1	1070	POP	DE	
FEC7 18E6	1080	JR	F3	
	1090 ;			"1 Pixel setzen u. pruefen ob ausserstes rechtes"
FEC9 3E01	1100 F6	LD	A,01H	
FECB 0F	1110 F7	RRCA		
FECC 10FD	1120	DJNZ	F7	
FECE 46	1130	LD	B,(HL)	
FECF 80	1140	OR	B	
FEDO 77	1150	LD	(HL),A	
FED1 B8	1160	CP	B	
	1170 ;			"Nicht ausserstes rechtes Pixel ... Sprung"
FED2 20EE	1180	JR	NZ,F5	
	1190 ;			"Reg. C und E korrigieren"
FED4 1D	1200 F8	DEC	E	
FED5 0D	1210	DEC	C	
FED6 AF	1220 SUCHE	XOR	A	
FED7 BA	1230	CP	D	
	1240 ;			"Y = 0 ... Sprung"
FED8 280D	1250	JR	Z,S3	
	1260 ;			"Y - 1"
FEDA 15	1270	DEC	D	
FEDB CD5FFE	1280	CALL	ADRES	
	1290 ;			"Kein Pixel gsetzt ... Sprung"
FEDE 2803	1300	JR	Z,S2	
FEE0 07	1310 S1	RLCA		
FEE1 10FD	1320	DJNZ	S1	
	1330 ;			"Pixel nicht gesetzt ... Sprung"
FEE3 D495FE	1340 S2	CALL	NC,FILL	
	1350 ;			"Y wieder herstellen (Y + 1)"
FEE6 14	1360	INC	D	
FEE7 3EAF	1370 S3	LD	A,OAFH	
FEE9 BA	1380	CP	D	
	1390 ;			"Y = 175 ... Sprung"
FEEA 280D	1400	JR	Z,S6	
	1410 ;			"Y + 1"
FEEC 14	1420	INC	D	
FEED CD5FFE	1430	CALL	ADRES	
	1440 ;			"Kein Pixel gsetzt ... Sprung"
FEF0 2803	1450	JR	Z,S5	
FEF2 07	1460 S4	RLCA		
FEF3 10FD	1470	DJNZ	S4	
	1480 ;			"Pixel nicht gesetzt ... Sprung"
FEF5 D495FE	1490 S5	CALL	NC,FILL	
	1500 ;			"Y wieder herstellen (Y - 1)"
FEF8 15	1510	DEC	D	
	1520 ;			"X - 1"
FEF9 1D	1530 S6	DEC	E	
	1540 ;			"Zaehler (C) - 1"
FEFA 0D	1550	DEC	C	
	1560 ;			"Zaehler nicht Null ... Sprung"
FEFB 20D9	1570	JR	NZ,SUCHE	
	1580 ;			"Position und Zaehler zurueckholen"
FEFD C1	1590	POP	BC	
FEFE D1	1600	POP	DE	
FEFF C9	1610	RET		
	1620 ENDE	END		

ENDE	FF00
S6	FEF9
S5	FEF5
S4	FEF2
S3	FEE7
S2	FEE3
S1	FEE0
SUCHE	FED6
F8	FED4
F7	FECB
F6	FEC9
F5	FEC2
F4	FEB3
F3	FEAF
F2	FEA9
F1	FE99
FILL	FE95
R1	FE90
RES	FE87
START	FE81
ADRES	FE5F
#	5DCE

+ An alle COMPUTER-CLUBS ***+***

Haben Sie Interesse an einer Vorstellung Ihres COMPUTER-CLUBS
in der neuen „CPU“?

Wenn ja, setzen Sie sich mit der Redaktion in Verbindung.

Geben Sie uns Informationen über Ihre besonderen

Club-Aktivitäten

z. B. Club-Treffen, Club-Magazine usw.

Wir sind an einer konstruktiven Zusammenarbeit zwischen
COMPUTER-CLUB und ZEITSCHRIFT „CPU“ interessiert.

Für Ihre Bemühungen eine attraktive Belohnung!

Unsere Aktion: **Gratis-Kleinanzeige**

Das SUPERANGEBOT für alle Mitglieder eines Computer-Clubs.

**Einmalige kostenlose Kleinanzeige in der
neuen „CPU“!!!**

Bitte kennzeichnen Sie Ihre Kleinanzeige mit dem Stichwort

„CLUB-ANZEIGE“

Schreiben Sie an: TRONIC-Verlag, Postfach, 3444 Wehretal 1

An alle COMPUTER-CLUBS +++ : +++ An alle COMPUTER-CLUBS

Load-Programm mit Diskettenkatalog und Generierung von Kleinbuchstaben in Picaschrift

Diskettenversion für den TI-99/4A

Benötigte Konfiguration: TI-99/4A, Ext.-Basic, Diskettenstation

Mit diesem Programm werden alle auf der Diskette befindlichen Programme angezeigt und gestartet.

Alle Texte in den so gestarteten Programmen, werden in Picaschrift, einschließlich ä, ö, ü, ß dargestellt. Voraussetzung ist, daß dieses Programm unter LOAD auf der Diskette abgespeichert wird und alle auf der Diskette befindlichen Programme vom Katalog aus gestartet werden. Entweder automatisch durch

die Wahl von Ext.-Basic oder durch RUN „DSKI.LOAD“. Danach die entsprechende Programm-Nr. aus der Programmübersicht und ENTER.

Die Umlaute werden durch gleichzeitiges Betätigen der Tasten CTRL&A=ä, CTRL&B=ö, CTRL&C=ü, CTRL&D=ß geschrieben. In den Zeilen bis 120 werden die Picaschrift und Zeichen definiert. Wenn man das Programm startet nachdem bis Zeile 120 eingegeben wurde,

kann im weiteren Verlauf sofort mit den neuen Zeichen gearbeitet werden.

In den Zeilen 150-290 werden alle auf der Diskette befindlichen Programme eingetragen, evtl. mit kurzen Erklärungen zum Programm oder es werden Leerzeichen für noch einzutragende Programme freigelassen. In diesen Zeilen entsprechen die Buchstaben ‚E‘ und ‚O‘ CTRL&E bzw. CTRL&O. Ab den Zeilen 370 werden die

in der Programmübersicht angezeigten Programme gestartet.

Hier darf nach RUN „DSKI.“ für x nur der Name eingetragen werden, unter dem das Programm abgespeichert ist. **In allen Programmen, die sich auf dieser Diskette befinden, brauchen also die Buchstaben und Umlaute nicht noch einmal definiert werden. Sie belegen somit in diesen Programmen keinen zusätzlichen Speicherplatz!**

```

10 ! Leadprogramm zum Erzeugen v.Kleinbuchstaben
20 ! Markus Spannkebel,Ffm.
30 FOR I=97 TO 144:: READ M$
40 CALL CHAR(I,M$) :: NEXT I
50 FOR I=1 TO 9 :: READ S,M$
60 CALL CHAR(S,M$) :: NEXT I
70 DATA 00000038043C443A,0040404078444438,0000003C40404
  03C,000404043C44443A,000000384478403C
80 DATA 0018242070202020,0000003C443C0438,0040407844444
  444,001000181010103C,0008000808084830
90 DATA 0040405060504848,00302020202078,0000007854545
  454,0000005864444444,0000003C44444438
100 DATA 0000003844784040,00000038443C0404,000000582420
  2020
110 DATA 116,0020207020202018,117,000000444444443A,121,

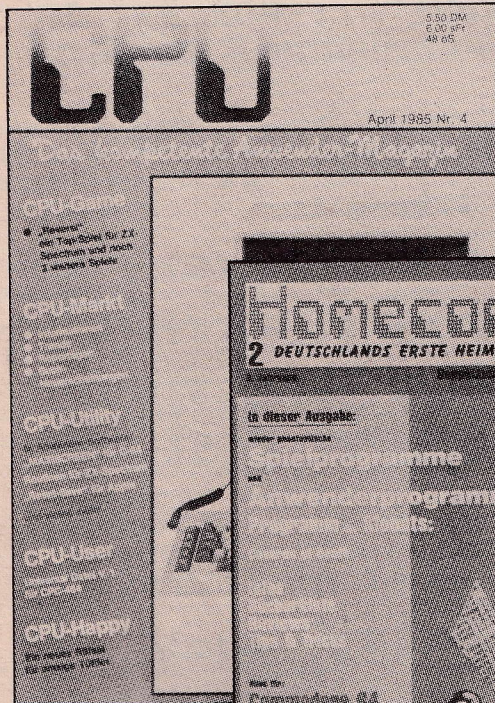
```

```

oooooo44443Co438,129,oo44oo38o43C443A,13o,oo44oo3C4
4444438
12o DATA 131,oo44oo444444443A,132,oo1824282424282o,133,
ooo8o4o2FFo2o4o8,143,1o1o1o1o1o9254381o
1o3 CALL CLEAR
14o DISPLAY AT(9,5):"Programmübersicht" :: FOR I=1 TO 5
oo :: NEXT I
15o DISPLAY AT(1,2):"1 'E' " :: DISPLAY AT(2,26):"I" !
'E' entspricht Tasten CTRL&E drücken
16o DISPLAY AT(3,2):"2 'E' " :: DISPLAY AT(4,26):"H"
17o DISPLAY AT(5,2):"3 'E' " :: DISPLAY AT(6,26):"R"
18o DISPLAY AT(7,2):"4 'E' " :: DISPLAY AT(8,26):"E"
19o DISPLAY AT(9,2):"5 'E' " :: DISPLAY AT(12,26):"W"
2oo DISPLAY AT(11,2):"6 'E' " :: DISPLAY AT(14,26):"A"
21o DISPLAY AT(13,2):"7 'E' " :: DISPLAY AT(16,26):"H"
22o DISPLAY AT(15,2):"8 'E' " :: DISPLAY AT(18,26):"L"
23o DISPLAY AT(17,2):"9 'E' " :: DISPLAY AT(20,26):"?"
24o DISPLAY AT(19,1):"1o 'E'" :: DISPLAY AT(22,26):"'0'"
25o DISPLAY AT(21,1):"11 'E'"
300 DISPLAY AT(23,1):"16 'E' .....E N D E....."
31o ACCEPT AT(24,26)VALIDATE(DIGIT)SIZE(2)BEEP:W
36o ON W GOTO 37o,38o,39o,4oo,41o,42o,43o,44o,45o,46o,4
7o,52o
37o RUN "DSK1.xxxxxxxxxx" ! Für x gültigen Programmnamen
einsetzen.
38o RUN "DSK1.xxxxxxxxxx"
39o RUN "DSK1.xxxxxxxxxx"
40o RUN "DSK1.xxxxxxxxxx"
41o RUN "DSK1.xxxxxxxxxx"
42o RUN "DSK1.xxxxxxxxxx"
43o RUN DSK1.xxxxxxxxxx"
44o RUN DSK1.xxxxxxxxxx"
45o RUN DSK1.xxxxxxxxxx"
46o RUN DSK1.xxxxxxxxxx"
47o RUN DSK1.xxxxxxxxxx"
52o END

```

Ihr Erfolgsprogramm – aus dem Tronic-Verlag



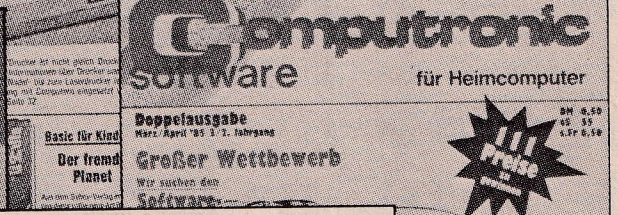
CPU
monatlich
Format A4
5,50 DM



Compute mit
monatlich
Format A4
2,80 DM



Homecomputer
zweimonatlich
Format A4
6,- DM



Computronic
zweimonatlich
Format A4
6,50 DM

Spiele-Kassetten
jetzt auch über den
Zeitschriftenhandel



Spiele-Kassetten:
Commodore 64
VC-20, Atari,
TI-99, ZX-Spectrum,
Laser 2001,
Schneider CPC.464
Je 19,90 DM

Spitzenprodukte

für den wachsenden Markt
der Heimcomputer-Anwender

Vertrieb: Verlagsunion, Friedrich-Bergius-Str. 20, 6200 Wiesbaden, Ruf (0 61 21) 26 60
Informationen: Tronic-Verlag, Landstr. 29, Postfach 41, 3444 Wehretal 1, Ruf (0 56 51) 4 06 93

„Maskengenerator“-Programm

Für den Commodore 64

Das Programm erstellt, wie der Name es bereits ausdrückt, auf komfortable Weise eine Bildschirmmaske, wie sie bei jeglicher Art von Basicprogrammen nötig ist, sei es als ‚Titelbild‘, ‚Spielfeld‘, ‚Eingabemaske‘, ‚Menue‘, ... Dabei wurden bei der Entwicklung folgende Ziele beachtet:

- (1) Das Programm sollte möglichst kurz sein, um auch für Leute mit ‚nur Datasette‘ durch die relativ kurze Ladezeit eine bessere Verwendungsmöglichkeit zu bieten.
- (2) Für die ‚nur Datasette-Besitzer‘ wurde auf die Möglichkeit verzichtet per ‚OPEN 1,8,15,“...P,W“‘ direkt auf Floppy zu speichern.
- (3) Auf die Möglichkeit, die

Maske per relativer oder gar sequentieller Datei zu speichern, wurde verzichtet.

(4) Als beste Lösung für die Maskenerstellung wurde daher das Generieren der Maske als Basicprogramm erachtet.

Vorteile der Lösung:

– leichteste Handhabung, da die Maske per Mergebefehl leicht an ein bereits vorhandenes Programm angefügt werden kann, bzw. die Maske bildet den Ausgangspunkt für ein neu zu schreibendes Basicprogramm.

Arbeiten mit „Maskengenerator“:

Nach Starten des Programms erscheint ein Menue, das es erlaubt, die Maske zu editieren (erstellen):

Dabei bleiben die Cursorfunktionen links, rechts, hoch, tief vorhanden. – Die Maske kann erstellt werden. – Nach Beendigung wird die ‚Pfeil links‘-Taste gedrückt und man kehrt zurück ins Menue.

Jetzt kann nach Belieben (a) die erstellte Maske gelöscht werden (Taste ‚E‘) oder (b) die Maske kann als Basicprogramm generiert werden (Taste ‚M‘).

Es können beliebig viele Masken erstellt werden!!!

Dann wählt man die Anfangszeilennummer des späteren Basicprogramms vor. – Sobald die Bildschirmmaske generiert ist, hat man die Wahl, entweder eine weitere Maske zu erstellen oder

„Maskengenerator sich selbst löschen zu lassen“.

Zum Schluß muß dann nur noch die Schlußzeile der erstellten Maske um ein Zeichen gekürzt werden, um beim späteren Aufruf ein ‚Scrollen‘ des Bildschirms zu verhindern.

VARIABLENDEF:

- A berechnet ASCII-Wert von A\$
- A\$ Menuvariable
- A0 Peek-Wert
- C Poke-Wert des Cursorzeichens
- I/J Laufvariablen
- N Zeilennummer
- NR Maskennummer
- P Pokeadresse
- ZN endgültige Zeilennummer

```

10 POKE53280,0:POKE53281,0:POKE650,128:PRINT"███"
11 POKE211,10:PRINT"███'84 BY G.SOFT":PRINT"* OLIVER GAUSS*7518 BRETTE *FALKEN
STR.88"
12 POKE211,10:PRINT"███MASKEN-GENERATOR":PRINT"███EDIT":PRINT"███MASKENERSTELLU
NG"
14 GETA$:IFA$=""THEN14
16 IFA$="E"THEN22
18 IFA$="M"THEN88
20 GOTO14
22 PRINT"███":P=1025:A0=PEEK(1025):C=160:POKEP,C
24 GETA$:IFA$=""THEN24
26 IFA$="←"THENPOKEP,A0:FORI=1024TO2023:POKEI+48128,PEEK(I):POKEI,81:NEXT:GOTO10
28 A=ASC(A$):IFA$="███"ANDP<2022THENPOKEP,A0:A0=PEEK(P+1):P=P+1:POKEP,C:GOTO24
30 IFA$="███"ANDP>1024THENPOKEP,A0:A0=PEEK(P-1):P=P-1:POKEP,C:GOTO24
32 IFA$="███"ANDP<1983THENPOKEP,A0:A0=PEEK(P+40):P=P+40:POKEP,C:GOTO24
34 IFA$="███"ANDP>1064THENPOKEP,A0:A0=PEEK(P-40):P=P-40:POKEP,C:GOTO24
36 IFA>191THENA=A-128:GOTO44
38 IFA>160THENA=A-64:GOTO44
40 IFA>95THENA=A-32:GOTO44
42 IFA>63THENA=A-64:GOTO44
44 IFP<2023THENPOKEP,A:A0=PEEK(P+1):P=P+1:POKEP,C
46 GOTO24
48 J=-1
50 IFJ<24THENJ=J+1:GOTO54
52 GOTO70
54 POKE631,13:POKE632,13:POKE198,2
56 PRINT"███":PRINTZN;"███"CHR$(34);:FORI=0TO39
58 P=49152+I+J*40:A=PEEK(P)
60 IFA>96THENA=A+64:GOTO68
62 IFA>63THENA=A+32:GOTO68
64 IFA>31THEN68
66 A=A+64

```

```

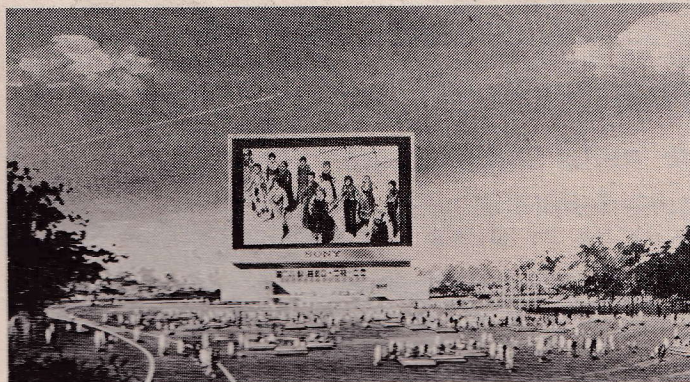
68 PRINTCHR$(A);:NEXT:PRINTCHR$(34);":PRINT"ZN="ZN+1":J="J":GOTO78":END
70 INPUT"WEITERE MASKE(J/N)";A$:IFA$="J"THENRUM
72 POKE830,8:POKE831,72
74 IFPEEK(830)<PEEK(831)THENA=PEEK(830)+2:POKE830,A:GOTO78
76 GOTO82
78 POKE631,13:POKE632,13:POKE198,2
80 PRINT"74";PEEK(830):PRINT"74";:END
82 POKE631,13:POKE632,13:POKE633,13:POKE634,13:POKE635,13:POKE636,13:POKE637,13:
POKE638,13
84 POKE639,13:POKE640,13:POKE198,10:PRINT"72":PRINT"74":PRINT"76":PRINT"78":P
RINT"80":PRINT"82"
86 PRINT"84":PRINT"86":PRINT"88":PRINT"?CHR$(147)";:END
88 INPUT"MASKENNUMMER(0-...)";NR:INPUT"ZEILENNUMMER(>255 )";N:ZN=N+NR*25:GOTO4
8

```

READY.

„JumboTRON“: 100mal größer als ein normaler Fernsehschirm

Rekorde und Höchstleistungen fallen nicht nur im Sport. Absolut rekordverdächtig ist eine gigantische Fernsehwand, die Sony 1985 anlässlich der Internationalen Messe im japanischen Tsukuba präsentiert: „JumboTRON“. Die technische Neuheit in Zahlen: Wenn der Bildschirm in seiner kompletten Größe erstrahlt, ist er 100mal größer als ein normaler 20-Zoll-Fernsehschirm. Der überdimensionale Schirm ist aus insgesamt 6300 Elementen zusammengesteckt. Jedes Element besteht aus 24 Zellen („Trini-Lites“), die ihrerseits



jeweils Licht in den drei Grundfarben ausstrahlen können. Der Bildaufbau ergibt demnach ungefähr 450 000 Bildpunkte und ist

vergleichbar mit der Punktauflösung derzeitiger Fernseher, die, je nach Betriebsart, eine Auflösung zwischen 300 000 und 700 000 Punkten

erbringen. Die Leistungsaufnahme pro Element beträgt etwa 8 Watt. Mit 7,5 Watt werden mehr als 90 Prozent der Leistung in Licht umgesetzt.

Daß das „JumboTRON“-Projektionssystem noch lange nicht an die Grenzen seiner Möglichkeiten angekommen ist, glauben Sonys Entwicklungs-Ingenieure. Wenn sich in Tsukuba die Messetore öffnen, weist der Riesens-Bildschirm eine Größe von 45 x 25 m auf – bei einem Gewicht von 36 Tonnen. Damit wächst natürlich auch der Betrachtungsabstand.

Neue Mikrodiskette von Wabash Datatech

Das zu den fünf bedeutendsten Datenträger-Herstellern der Welt zählende Unternehmen Wabash-Datatech hat jetzt eine 3,5"-Mikrodiskette auf den Markt gebracht. Als erstes Wabash-Produkt besitzt diese neue Diskette eine neuartige, mit Kobalt angereicherte Oxidbeschichtung, die eine besonders hohe Aufzeichnungsdichte erlaubt.

Die Verteilung beträgt nominell $1,77 \times 10^3 \mu\text{m}$ und ergibt eine hohe Gleichförmigkeit der Oberfläche mit gleichbleibender Signalqualität und -amplitude. So beträgt die Amplitude bei 1f auf Spur 00 weniger als 125% und macht bei 2f auf Spur 79 mehr als 80% aus. Die nominelle Koerzitivität von 675 Oerstedt erbringt eine verbesserte Bit-

auflösung und damit einen erhöhten Rauschspannungsabstand.

Für den Endanwender bedeutet all dies, daß Wabash Datatech aufgrund der genannten Verbesserungen ein Produkt entwickelt hat, das den gegenwärtigen und kommenden Normen für Mikrofloppies von 135 Spuren pro Zoll und mehr als 8000 Bit pro

Zoll entspricht. Die Kapazität wird mit 0,5 MByte je Seite angegeben, wobei das Aufzeichnungsformat dem der 5¼"-Floppies mit doppelter Aufzeichnungsdichte entspricht. Die aus Hartplastik gefertigte Schutzhülle der 3,5"-Mikrofloppies verleiht den Datenträgern den Schutz, der auf dem hier gegebenen

Fortsetzung S. 27

Leistungsniveau unverzichtbar ist.

Bei Wabash ist man davon überzeugt, daß in der 3,5"-Mikrofloppy eine große Marktchance liegt; zwar werde die 5¼"-Floppy noch einige Zeit sehr große Verbreitung behalten, stückzahlmäßig allerdings auf lange Sicht die Führungsposition an die Mikrofloppy abgeben müssen.

Die Lebensdauer der neuen Datenträger wird von Wabash mit minimal 5 Mio. Durchläufen angegeben; die Aufzeichnungsfläche jeder Diskette wird außerdem vor

dem Verlassen des Fertigungswerks im amerikanischen Bundesstaat Illinois genau geprüft. Eine weitere Qualitätsprüfung schließt sich an, um Transportschäden oder Fehler an der Verpackung zu ermitteln.

Die neue Mikrodiskette kann in allen Computersystemen mit einem Sony/Shugart-Mikrofloppy-Laufwerk eingesetzt werden. Hierzu gehören momentan Apricot, Apple Machintosh, Lisa, der Touchscreen-PC Hewlett-Packard 150 und die Textverarbeitungssysteme der Serie 35.

NACHTRAG zum Listschutz aus dem Programm FAST-LOAD für den C-64

Durch „£“ ist kein sicherer Listschutz gewährleistet. Dieser Listschutz ist durch die Befehlsfolge –
POKE 2052,0
POKE 2078,0
0 <CR>

1 <CR>
<CR> = CARRIAGE RETURN – zu knacken!
Vielen Dank an unseren aufmerksamen Leser, dem wir diese „Aufklärung“ zu verdanken haben!

Reassembler-Programm für den C-64

Der Reassembler 64 ist ein Toolkit für den Maschinensprachprogrammierer des C-64

Nach dem Laden des BASIC-Loaders und dessen Start befindet er sich im Speicher des 64'er von \$C 800 bis \$D 000 und kann mit einem Monitor absolut abgespeichert werden (z. B. mit ASMON). Der Reassembler 64 wird mit SYS 51200 gestartet. Danach fragt er nach der Anfangsadresse

des Maschinenprogramms, das er wieder in Assemblerquellcode umwandeln soll, die man als 2-Byte-Zahl eingeben muß. Daraufhin die Endadresse derselben (auch als 2-Byte). Es folgt der Zeilenabstand (als 1-Byte) und die Zeilennummer, bei der der Assemblerquellcode be-

ginnen soll (die Zeilennummer muß größer sein als die letzte Zahl eines eventuell noch vorhandenen BASIC-Programms). Die Zeilennummer wird auch als 2-Byte-Zahl eingegeben.

Nun beginnt der Reassembler 64 das Maschinenprogramm in Quellcode zurückzuwan-

deln und meldet sich, wenn dies beendet ist, wieder mit ‚Ready‘. Der so erstellte Assemblerquellcode kann mit jedem guten Assembler verarbeitet werden.

Das Programm ist mit Prüfsommen versehen, so daß Eingabebefehle ausgeschlossen werden.

```

1 REM *****
2 REM *
3 REM * REASSEMBLER 64
4 REM *
5 REM * COPYRIGHT BY STEPHAN BARTOSCH*
6 REM *
7 REM * (C) TRONIC-SOFT
8 REM *
9 REM *****
60000 DIMH(75):FORI=0TO9
60010 H(48+I)=I:H(65+I)=I+10:NEXT
60020 FORI= 51200 TO 53250 :READA$
60030 H=ASC(LEFT$(A$,1))
60040 L=ASC(RIGHT$(A$,1))
60050 D=H(H)*16+H(L):S=S+D:POKEI,D
60060 A=A+1:IFAC<20THENNEXT:A=-1
60070 READY:Z=Z+1:IFV=STHEN60085
60080 PRINT"DATAFEHLER IN ZEILE ":";60200+Z:END
60085 IFA<0THEN60100
60090 S=0:A=0:NEXT
60100 END
60201 DATA A9,34,A0,CC,20,1E,AB,20,61,C8,8D,37,CD,85,FA,20,61,C8,8D,36, 2455

```



```

60262 DATA 52,45,53,53,45,20,3A,00,0D,1D,1D,1D,1D,1D,45,4E,44,41, 972
60263 DATA 44,44,52,45,53,53,45,20,3A,00,0D,1D,1D,1D,1D,5A,45,49,4C, 1078
60264 DATA 45,4E,41,42,53,54,41,4E,44,20,3A,00,0D,1D,1D,1D,1D,5A, 1023
60265 DATA 45,49,4C,45,4E,4E,55,4D,4D,45,52,20,3A,00,0D,0D,11,11,11,1D, 1029
60266 DATA 20,20,20,20,1D,1D,1D,1D,1D,12,20,20,42,49,54,54,45,20,57,41, 915
60267 DATA 52,54,45,4E,20,20,A1,06,01,50,00,17,0C,0A,00,00,00,C0,01,EE, 1101
60268 DATA 8F,FF,FF,FF,00,00,00,00,00,00,00,00,FF,10,00,00,00,00,00, 1179
60269 DATA 00,00,00,00,00,00,00,00,00,00,10,00,00,00,10,00,00,00,00, 32
60270 DATA 00,00,00,00,00,00,00,00,00,00,FF,00,00,00,00,00,00,00,00, 255
60271 DATA 00,00,00,00,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF, 4080
60272 DATA FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF, 5100
60273 DATA FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF, 5100
60274 DATA FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,00,00,00,00,00,00, 3060
60275 DATA FF,10,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,10,00, 287
60276 DATA 00,00,10,00,00,00,00,00,00,00,00,00,00,00,00,00,00,FF,00, 271
60277 DATA 00,00,00,00,00,00,00,00,00,00,00,00,FF,FF,FF,FF,FF,FF,FF, 2040
60278 DATA FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF, 5100
60279 DATA FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,FF,42,43,43, 4158
60280 DATA 53,42,45,51,42,4E,45,42,4D,49,42,50,4C,42,56,43,42,56,53,42, 1470
60281 DATA 52,48,43,4C,43,53,45,43,43,4C,44,53,45,44,43,4C,49,53,45,49, 1458
60282 DATA 43,4C,56,49,4E,58,49,4E,59,44,45,58,44,45,59,49,4E,43,44,45, 1514
60283 DATA 43,4E,4F,50,50,48,41,50,4C,41,50,48,50,50,4C,50,54,53,58,54, 1549
60284 DATA 58,53,54,41,59,54,59,41,54,41,58,54,58,41,4C,44,41,4C,44,59, 1563
60285 DATA 4C,44,58,53,54,41,53,54,59,53,54,58,52,54,53,52,54,49,4A,4D, 1614
60286 DATA 50,4A,53,52,45,4F,52,4F,52,41,41,4E,44,42,49,54,4C,53,52,41, 1515
60287 DATA 53,4C,52,4F,52,52,4F,4C,53,42,43,41,44,43,43,4D,50,43,50,58, 1514
60288 DATA 43,50,59,90,18,B0,18,F0,18,D0,18,30,18,10,18,50,18,70,18,00, 1452
60289 DATA 19,18,19,38,19,D8,19,F8,19,58,19,78,19,B8,19,E8,19,C8,19,CA, 1820
60290 DATA 19,88,19,EE,00,E6,01,F6,02,FE,14,CE,00,C6,01,D6,02,DE,14,EA, 2274
60291 DATA 19,48,19,68,19,08,19,28,19,BA,19,9A,19,A8,19,98,19,AA,19,8A, 1442
60292 DATA 19,A9,0C,AD,00,A5,01,B5,02,BD,04,B9,05,A1,06,B1,17,A0,0C,AC, 1822
60293 DATA 00,A4,01,B4,02,BC,14,A2,0C,AE,00,A6,01,B6,03,BE,15,8D,00,85, 1740
60294 DATA 01,95,02,9D,04,99,05,81,06,91,17,8C,00,84,01,94,12,8E,00,86, 1489
60295 DATA 01,96,03,9E,15,60,19,40,19,4C,00,6C,1B,20,10,49,0C,4D,00,45, 1033
60296 DATA 01,55,02,5D,04,59,05,41,06,51,17,09,0C,0D,00,05,01,15,02,1D, 346
60297 DATA 04,19,05,01,06,11,17,29,0C,2D,00,25,01,35,02,3D,04,39,05,21, 432
60298 DATA 06,31,17,2C,00,24,11,4E,00,46,01,56,02,5E,04,4A,1A,0E,00,06, 630
60299 DATA 01,16,02,1E,04,0A,1A,6E,00,66,01,76,02,7E,04,6A,1A,2E,00,26, 774
60300 DATA 01,36,02,3E,04,2A,1A,E9,0C,ED,00,E5,01,F5,02,FD,04,F9,05,E1, 1886
60301 DATA 06,F1,17,69,0C,6D,00,65,01,75,02,7D,04,79,05,61,06,71,17,C9, 1412
60302 DATA 0C,CD,00,C5,01,D5,02,DD,04,D9,05,C1,06,D1,17,E0,0C,EC,00,E4, 2208
60303 DATA 11,00,0C,CC,00,C4,11,FF,00,00,00, 893

```

READY.

**Jetzt überall
im
Zeitschriften-
handel:**

Computronic
software für Heimcomputer

Das Softwaremagazin

Spectrum Hardwaretest: Ein neues Diskettenlaufwerk

Wer die Wahl hat, hat die Qual!

Nachdem es für den Spectrum schon 2 Diskettenlaufwerke gibt, die Timex-Floppy und die Beta Disk, erreichte uns gerade die neueste Nachricht aus England über eine 3. Floppy. Dabei geht es um das Diskettenlaufwerk der Firma Opus mit dem klangvollen Namen „Discovery 1“. Da uns diese Floppy sehr interessant erschien, haben wir uns ein Vorserienmodell besorgt und für Sie getestet. Vielleicht können wir Ihnen mit unserem folgenden Bericht die Auswahl Ihres Diskettenlaufwerks leichter machen.

Eines aber schon vorweg: Wir wurden von der Discovery 1 nicht enttäuscht!

Schon die äußere Aufmachung der Floppy läßt auf ein gut durchdachtes und ausge-

feiltes System schließen. Die Discovery 1 sieht recht professionell aus und hat ein robustes, schwarzes Stahlblechgehäuse, das sehr schön zum Spectrum und auch zum Spectrum + paßt.

Eine tolle Sache ist natürlich auch, daß man die Floppy-Einheit auf den Spectrum-Bus aufsteckt und beide Geräte zusammen verschraubt. Zum ersten läuft somit die Stromversorgung beider Geräte über die Floppy, zum zweiten ist die Gefahr eines Wackelkontaktes, der zuweilen sehr ärgerlich sein kann, so gut wie ausgeschlossen.

Das Handbuch, das der Floppy demnächst in deutscher Übersetzung beiliegt, ist sehr übersichtlich und bietet dem Anfänger eine sehr gute Einführung. Bei weiteren Lieferungen wird es auch eine Ein-

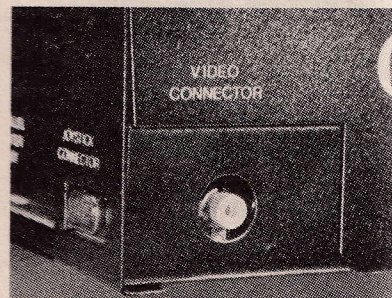
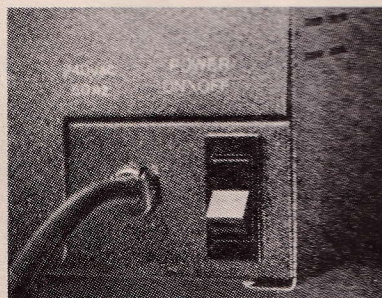
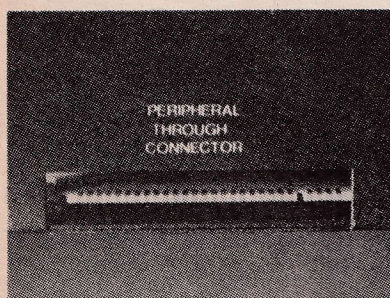
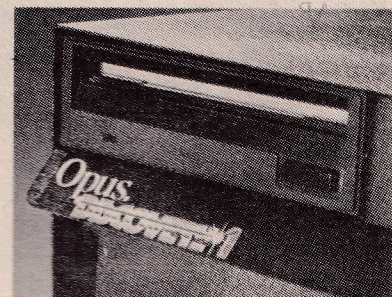
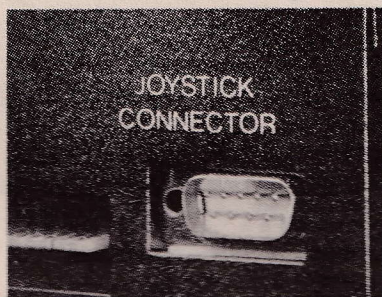
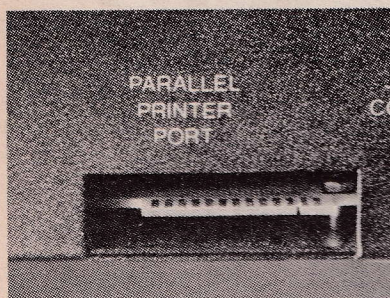
führungsdiskette mit einem kleinen Adreßverwaltungsprogramm, für die ersten Gehversuche mit dem Laufwerk, geben.

Liest man sich nun das Handbuch ein wenig durch und schaut sich das Gehäuse etwas näher an, so bemerkt man, daß die Opus-Floppy noch einige Trümpfe auszuspielen hat!

In der Gehäuserückwand der Discovery 1 befindet sich zunächst einmal ein monochromes Monitorinterface, das Ihnen ein flimmerfreies Bild liefert. Auf der rechten Seite wurde ein Kempston-kompatibler Joystick-Port in das Gehäuse integriert. Ein Stück weiter vorne stoßen wir auf ein bidirektionales Centronics-Interface, das zum einen den Drucker bedient und zum anderen eine Kommuni-

kation mit anderen Rechnern ermöglicht. Um die Sache abzurunden, findet sich dann auch noch, ganz vorne, ein komplett durchgeführter Datenbus für den Anschluß weiterer Peripheriegeräte. Lediglich solche Geräte, die die NMI-Leitung benötigen, können nicht angeschlossen werden. Daß die Opus-Floppy das Ergebnis einer sorgfältigen Planung ist, zeigt sich auch daran, daß an einen Netzschalter gedacht wurde. Den üblichen Kontaktproblemen mit den vielen verschiedenen Interfaces wird durch diese Komplettlösung endlich ein Ende bereitet!

Wenn sie nun denken, daß das alles war, müssen wir Sie (allerdings angenehm) enttäuschen! Das größte Plus der Discovery 1 ist das Betriebssystem.



Anschlußmöglichkeiten für Discovery 1 – Ein neues Diskettenlaufwerk für den ZX-Spectrum

Der erste wichtige Punkt im Betriebssystem ist die volle Microdrive-Kompatibilität des Befehlsaufbaus (siehe auch Befehlsliste). Sie können nun Basic-Programme, die für den Microdrive momentan vermehrt auf dem Markt angeboten werden, ohne irgendwelche Änderungen auf die Floppy bringen.

Dies bietet Ihnen kein anderes Laufwerk für den Spectrum!

Des weiteren wurde das Betriebssystem noch um einige nützliche Befehle erweitert:

A) Serielle Files können wieder eröffnet, vergrößert und neu beschrieben werden.

B) Random-Dateien können durch einfache Basic-Befehle erstellt, gelesen und vergrößert werden.

C) Das Betriebssystem ermöglicht die Definition einer

RAM-Disk im Speicher. Auch das geschieht durch einen einfachen Basic-Befehl. Um die Microdrive-Kompatibilität zu überprüfen, haben wir einige Basic-Programme auf Kassette gebracht, um sie mit der Discovery 1 zu testen. Alle Basic-Programme, die wir untersuchten, liefen auf dem Laufwerk einwandfrei. So haben wir z. B. eine microdrivefähige Version von Tasword II auf die Diskette gebracht. Da auch der Maschinencode-Teil problemlos arbeitete, war der nächste logische Schritt, die Lauffähigkeit von anderen Maschinencode-Programmen auszutesten. So z. B. Beta Basic, das, bis auf kleine Fehlercode-Verschiebungen, problemlos lief. Maschinencode-Programme, die direkt auf das Microdrive-Betriebssystem

zugreifen, wie z. B. der C-Compiler von Hisoft, funktionieren leider nicht.

Nach Auskunft des Importeurs der Discovery ist auch schon deutsche Software auf Diskette lieferbar. Außerdem wird es in Kürze auch Hardcopy-Software für die Centronics-Schnittstelle geben.

Hier noch ein kurzer Überblick der technischen Daten des Opus-Laufwerks:

Diskettenlaufwerk: 3,5 Zoll Industriestandard

40 Track, single-sided, double-density

Gesamtkapazität 250 KB, unformatiert

Formatierte Kapazität 180 KB

Track to Track Time 3 ms

Eingebautes Netzteil

Eigenes DOS auf Eprom; es wird kein Spectrum-RAM belegt.

Zusätzliche Features:

Durchgeführter Datenbus Parallel-Druckerinterface (bidirektional)

Joystick-Interface (Kempston-kompatibel)

Video Monitorausgang (monochrom)

Preis:

Discovery 1 (Single-Laufwerk) DM 998,-

Discovery 2 (Doppellaufwerk) DM 1498,-

Discovery + (2. Laufwerk für Discovery 1) DM 559,-

Garantie: 12 Monate ab Verkaufsdatum

Nach unserer Meinung stellt die Discovery eine klare Alternative zu anderen Systemen auf dem Markt dar.

Unser Urteil lautet daher nach erfolgreichem Test: empfehlenswert!

Befehls-Directory zur Discovery 1

CAT <Laufwerk> Directory
 CAT # <Strom>; <Laufwerk> Directory auf einen Strom ausgeben
 CLEAR # Setzt alle geöffneten Ströme zurück
 CLEAR # <Strom> Setzt einen bestimmten Strom zurück (mit Verlust der Daten im Puffer)
 CLOSE # <Strom> Schließt Strom; Daten, die sich im Puffer befinden, werden gesendet
 CLS # BORDER 0 : INK 0 : Paper 7 : CLS
 LOAD*„m“; 1; „NAME“ DATA <ARRAYNAME> 0; CODE <Startadresse>; <BYTES>; LINE <AUTOSTART-ZEILE>
 SAVE*„m“; 1; „NAME“ Löschen von Programmen etc.
 VERIFY*„m“; 1; „NAME“
 ERASE*„m“; 1; „NAME“

Die Microdrive Syntax wurde voll übernommen. Dieselben Befehle in einer zusätzlichen Kurzform:
 LOAD*1; „NAME“, SAVE*1; „NAME“, VERIFY*1; „NAME“, ERASE 1; „NAME“
 Alle anderen Parameter sind wie bei Microdrive.

FORMAT „j“; <Status> Zur Steuerung des Joystick-Ports
 FORMAT „m“; <Laufwerk>; „<DISK-NAME>“
 FORMAT <Laufwerk>; „<DISK-Name>“
 Formatieren einer Diskette

INKEY\$ # <Strom> Einzelnes Zeichen aus einem Strom lesen
 INPUT # <Strom>; Variable1; Variable2; Werte oder eine Folge von Werten aus einem Strom lesen
 LPRINT # <Strom>; Variable1, Variable2; ... Gibt Werte der Variablen auf Strom aus
 OPEN # <Strom>; <Kanal> <Zugriffsart> Verbindet den Strom mit dem Kanal
 Folgende Zugriffsarten sind möglich:
 IN Für Input
 OUT Für Output
 EXP Vergrößerung eines sequentiellen Files
 RND Für Random-Dateien (Input & Output)
 POINTE # <Strom>; <Datensatz Nummer> Setzt bei Random Files den Pointer auf den entsprechenden Datensatz
 Zusammen mit LOAD*, SAVE* VERIFY*
 SCREEN \$ Diskette komprimieren
 MOVE „d“;1 TO „d“; 1 Diskette kopieren bei Einzeldrives
 MOVE „d“;1 TO „d“; 3 Kopieren bei Doppellaufwerk
 MOVES „d“;1 TO „d“; 2
 FORMAT 5; „NAME“ Ramdisk definieren

Monitorprogramm für den Atari 400/600 XL/800/800 XL

Cassettenversion

Nach der Vorstellung des Monitorprogramms in der CPU-Ausgabe April haben wir zahlreiche Reaktionen durch unsere Leser erhalten. Aus diesem Grund hat sich das CPU-Team entschlossen, das Monitorprogramm auch in der Cassettenversion zu veröffentlichen. Dadurch sollen eventuell aufgetretene Mißverständnisse aufgeklärt werden.

Programmbeschreibung:

Ist das Programm vollständig eingegeben, so wird es mit RUN gestartet. Nun geschieht für den Betrachter ca. 3 Minuten nichts. In Wirklichkeit aber beginnt das BASIC-Programm das in HEX-Code abgelegte Maschinenprogramm des Monitorprogramms in einen dafür vorher reservierten Pufferbereich (BUF\$) zu speichern. Die Cassettenversion wird nach dem Signal für die Ausgabe auf Cassette und dem RETURN in einem BOOT-Format abgespeichert. Das Programm kann nicht im BASIC mit CLOAD geladen werden, sondern beim Einschalten des Computers muß durch Drücken der START-Taste der BOOT-Modus aufgerufen werden. Die Cassettenversion läßt sich auch ohne Modul starten. In einige Module, so z. B. PAC-MAN, kann man mit der Cassettenversion eindringen und sich den Speicherbereich ansehen oder kopieren.

Zur Anwendung:

Dieses Programm ermöglicht dem Anfänger den ersten Blick ins „Reich der Maschinsprache des 6502-Prozessors“. Besonders geeignet erscheint mir hierfür der implementierte Disassembler so-

wie die Möglichkeit, Speicherinhalte als HEX- und ASCII-Codezeichen darzustellen und zu verändern. Damit wird dem Anfänger auch die Möglichkeit gegeben, Programme selbst zu erstellen und zu testen. In diesem Zusammenhang sei noch erwähnt, daß jedes selbstgeschriebene Programm am Ende auf jeden Fall wieder auf den Anfang des Monitorprogramms gerichtet sein sollte (\$700 oder \$1F00).

Aber auch dem Fortgeschrittenen werden einige nützliche Befehle an die Hand gegeben. So existiert z. B. die Möglichkeit, einzelne Sektoren auf einer einfach formatierten Diskette zu lesen oder Daten auf sie zu schreiben. Dies kann dann von Nutzen sein, wenn man Daten von einer Diskette retten will, deren Inhaltsverzeichnis defekt ist oder, wenn man sich die Daten von BOOT-Programmen einmal genauer ansehen möchte. Günstig dürfte auch die gleichzeitige Darstellung von HEX- und ASCII-Codezeichen sein.

Beschreibung des Monitorprogramms:

Es handelt sich hierbei um ein Urladerprogramm, das nur benötigt wird, um das Maschinenprogramm, das in den Zeilen 300 bis 1150 abgelegt ist, auf Cassette abzuspeichern. Die Cassettenversion wird dann durch das Drücken der Starttaste beim Einschalten „gebootet“ und automatisch gestartet.

Das Monitorprogramm stellt eine autonome Programmiersprache dar, die über die folgenden Befehle verfügt:

1. Der Befehl „Z“ listet einen Speicherbereich in Hexadezi-

malzahlen und in ASCII-Code auf. Der dazu verwendete Befehl lautet: „Zxxxx,yyyy“. Dabei stellen die vier x-Buchstaben die Anfangsadresse und die y-Buchstaben die Endadresse dar.

2. Der implementierte Disassembler läßt sich mit „Dxxxx,yyyy“ aufrufen.

3. Der Speicher kann mit dem Eintragebefehl verändert werden. Dazu empfiehlt sich folgende Vorgehensweise:

a) Den zu verändernden Speicherbereich mit der Z-Funktion auflisten.

b) Um die HEX-Zahlen zu verändern, muß „E“ eingegeben werden. Zur Manipulation der ASCII-Code-Zeichen muß „EA“ eingegeben werden.

c) In der vorliegenden Maske kann nun jeweils entsprechend der HEX-Wert oder der ASCII-Code eingegeben werden. Die Daten werden in den Speicher übernommen, sobald Return gedrückt wurde. Um aus diesem Modus auszusteigen, muß die BREAK-TASTE gedrückt werden.

4. Der Goto-Befehl springt in ein beliebiges Maschinenprogramm im Speicherbereich. Er hat folgende Schreibweise: „Gxxxx“.

5. Der Ausdruck auf dem Drucker, anstatt auf dem Bildschirm (Z- oder D-Funktion), wird durch die Eingabe von „P“ ermöglicht. Ausgeschaltet wird diese Funktion durch „B“.

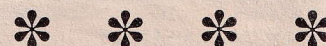
6. Programme können von Cassette geladen oder abgespeichert werden. Zum Laden wird „L“ eingegeben. Die Abspeicherung erfolgt durch die Eingabe von „Sxxxx,yyyy“.

7. Daten können frei im Speicher verschoben werden. Der zu verwendende Befehl lautet „Mxxxx,yyyy,zzzz“. Hierbei stellt „xxxx“ die Adresse dar, ab die der Bereich „yyyy“ bis „zzzz“ neu abgespeichert werden soll.

8. Mit „F“ wird das Sektorlese- bzw. Sektorschreibprogramm aufgerufen, das als erstes die Anzahl der zu übertragenden Sektoren erwartet. Da bei einem Durchgang maximal nur 255 (\$FF) Sektoren gelesen/geschrieben werden können, ist die Angabe über die Anzahl auf zwei Ziffern begrenzt. Als nächstes wird nach dem Sektor gefragt, ab dem gelesen/geschrieben werden soll. Der Bereich erstreckt sich von \$0001 bis \$02D0. Deshalb werden die Angaben durch vier Ziffern getätigt. Nun wird die Adresse eingegeben, ab die der erste Sektor gelesen/geschrieben wird. Alle anderen Sektoren werden im Abstand von 128 Bytes gelesen/geschrieben. Im letzten Schritt kann entschieden werden, ob in den Speicherbereich gelesen („L“) oder der Speicherbereich auf Diskette geschrieben werden soll („S“).

Anmerkung:

Natürlich müssen alle Befehle mit RETURN abgeschlossen werden. Zusätzlich ist noch zu beachten, daß alle Anfangs- und Endadressen mit vier Ziffern dargestellt werden (also z. B. \$700 als 0700).



```

0 REM (C)          BY MICHAEL SASSIN
1 REM MONITORPROGRAMM-CASSETTENVERSION
2 REM DIE DATEN WERDEN NACH DEM SUMMZEICHEN AUF CASSETTE UEBERTRAGEN
3 DIM H$(1),B$(2),A$(60),BUF$(2560)
4 I=1
5 GOTO 100
10 IF H$>="0" AND H$<="9" THEN D=ASC(H$)-48:RETURN
15 D=ASC(H$)-55:RETURN
20 H$=B$(1,1):GOSUB 10:B=D:H$=B$(2,2):GOSUB 10:B=B*16+D:RETURN
100 READ A$:IF A$="END" THEN GOTO 150
110 FOR X=1 TO 30:B$=A$(2*X-1,2*X):GOSUB 20:BUF$(I,I)=CHR$(B):I=I+1:NEXT X:GOTO
100
150 CLOSE #1:OPEN #1,8,128,"C"
160 IOCB=832+16
170 POKE IOCB+2,11
180 BUF=ADR(BUF$):POKE IOCB+4,BUF-(INT(BUF/256)*256):POKE IOCB+5,INT(BUF/256)
190 BUFLN=2560:POKE IOCB+8,BUFLN-(INT(BUFLN/256)*256):POKE IOCB+9,INT(BUFLN/
256)
200 Z=USR(ADR("hhh*LVd"),16):REM BEACHTE DIE INVERSEN ZEICHEN * UND d
210 CLOSE #1:END
300 DATA 0014E106FF06A93C8D02D3A9018509A9008D4402A2089DFFD1CAD0FAEAEA
310 DATA EADBA2FF9AA9108D0602A90BBD0702A900855220DE077D9B202020202020
320 DATA 20202020204D4F4E49544F5250524F4752414D4D9B9B2020202020284329
330 DATA 2031393834204259204D49434841454C2053415353494E9BEA58BEAEAEAEA
340 DATA 20FB0720B507AD0001C947D0034CA008C945D0034C4809C95AD0034CB008
350 DATA C953D0034CBF09C94CD0034C2D0AC944D0034CC10BC950D0034C070FC942
360 DATA D0034C160FC946D0034C250FC94DD00A4C4D10EAEAEAEAEAEAEAE2007084C
370 DATA 5407A200A9059D4203A9009D4403A9019D4503A9269D4803A9009D490320
380 DATA 56E4981007C089F0034C4608606885B06885B1E6B0D002E6B1A000B1B0C9
390 DATA EAF006208B084CE4076CB00020DE079B52454144599BEA6020DE079B4552
400 DATA 524F52209BEA60BD0001202F080A0A0A0A99B200E8BD0001202F0819B200
410 DATA 99B20060C9303011C93A300AC9413009C947B0056909290F6068686868C0
420 DATA 80F0062007084C540720DE079B425245414B9BEA4C5407A000484A4A4A4A
430 DATA C90AB0086930208B084C78086936208B0868290FC90AB0066930208B0860
440 DATA 6936208B08602095089810034C460860AAAD47034BAD460348BA60A001A2
450 DATA 01201408A000E82014086CB20020BD0D203609A90085B8A5B3206008A5B2
460 DATA 206008A920208B08A920208B08A000B1B2206008A920208B0820460DE6B8
470 DATA 203609A5B8C908904BE68B38A5B2E5B885B2A5B3E90085B3A90085B8B1B2
480 DATA 48297FC97DB004C920B00968A920208B084C150968208B0820460DE6B820
490 DATA 3609A5B8C90890DB20460DA93C208B08A99B208B084CB6084CCE08A5B2C5
500 DATA B4D008A5B3C5B5D00568684C54076020FB07AD0101C941D0034C8B0920B5
510 DATA 07A001A20086B8201408E8A000201408E8E8E8A003201408A5B5A00091B2
520 DATA 20460DE6B8A5B8C908900620B5074C5809E8E84C6A0920B507A001A20086
530 DATA B8201408E8A000201408A91E85B4A90185B5A000B1B491B220460D20CF10
540 DATA E6B8A5B8C90890EC20B5074CBE0920BD0D20900A20F10AA270A9039D4203
550 DATA A98D9D4403A90A9D4503A90B9D4A03A9809D4B032056E498100620030B4C
560 DATA 4C08A90B9D4203A5B29D4403A5B39D4503A5B99D4803A5BA9D49032056E4
570 DATA 98100620030B4C4C08A270A90C9D42032056E498100620030B4C4C082003
580 DATA 0B4C5407A980853EE64B207DE4207AE4A203BD00049D4002CA10F7AD4202
590 DATA B504AD43028505A006B900049104C810FB4C650AA07FB9000491048B10FB
600 DATA 18A50469808504A50569008505CE4102F00BEE0A03207AE410DC4C4C0820
610 DATA 7AE4A93C8D02D34C5407433A9B38A5B4E5B285B9A5B5E5B385BA38A5B2E9
620 DATA 0685B2A5B3E90085B318A5B9690685B9A5BA690085BA18A5B9F02AC98010
630 DATA 1218A5BA0A90034C4608690185B1A98085B960A5BA1869010A90034C4608
640 DATA 85B1E6BAA90085B960A5BA0A90034C460885B160A000B1B2993001B9B000
650 DATA 91B2C8C00630F160A000B9300191B2C8C00630F660A2FF9A4C540742524B
660 DATA 525449525453504850434C43504C50534543504841434C49504C41534549
670 DATA 444559545941544159434C56494E59434C44494E5853454441534C524F4C
680 DATA 4C5352524F525458415458535441585453584445584E4F5042504C424D49
690 DATA 425643425653424343424353424E454245514C44414C44594C4458535441
700 DATA 535459535458434D504350594350584F5241414E44454F52414443534243
710 DATA 424954494E434445434A53524A4D503F3F3F20BD0DA99B208B08A000B1B2

```

```

720 DATA 85B8D9D80DF005C8C097D0F69848A5B3206008A5B2206008A920208B0820
730 DATA 620D68A8B96F0E85B0A90B85B19848A90085BBA4B8B1B0208B0BE6B8A5B8
740 DATA C90390F1A920208B086BABC01DB0034C680CC025B0034CFD0CC030B0034C
750 DATA 6B0CC038B0034C760CC040B0034CB30CC055B0034C900CC065B0034C960C
760 DATA C067B0034CA20CC07EB0034CAE0CC08DB0034CB40CC096B0034CC00CC096
770 DATA D0034CC0C4C590DA923208B0820E80C4C530D20E30C20DE072C5829EA4C
780 DATA 530D20E30C20DE07292C59EA4C530D20E80C4C530D20E80C20DE072C58EA
790 DATA 4C530D20E80C20DE072C59EA4C530D20DC0C4C4D0D20DC0C20DE072C58EA
800 DATA 4C4D0D20DC0C20DE072C59EA4C4D0DA928208B0820DC0CA929208B084C4D
810 DATA 0DA90285B84CEC0CA928208B08A90185B8A924208B08A4B8B1B2206008C6
820 DATA B8D0F560A001B1B2301018A5B271B285B0A5B3690085B14C270D49FF1869
830 DATA 0185B038A5B2E5B085B0A5B3E90085B118A5B0690285B0A5B1690085B1A9
840 DATA 24208B08A5B1206008A5B02060084C530DE6B2D002E6B36020460D203609
850 DATA 20460D20360920460D2036094CC40BA201A5B8C900F01AC940F016C960F0
860 DATA 12A203C920F00C291FC919F006290FABBEAD0DA00098488A48B1B2206008
870 DATA A920208B0868AA68ABCBCAD0EA984820DE07202020EA68ABC8C004D0F060
880 DATA 02020201020202010102010103030301A001A201201408A000EB201408A0
890 DATA 03E8EB201408A002EB20140860004060081B283848586B788898A8B8C8D8
900 DATA E8F80A2A4A6A8A9AAABACAEA1030507090B0D0F0A0C0E0A209294969A9C9
910 DATA E90121416181A1C1E11131517191B1D1F12484A4C4E40525456585A5C5E5
920 DATA 0626466686A6C6E694B41535557595B5D5F516365676D6F696B6202C4C8C
930 DATA ACCCEC0D2D4D6D8DADDCDE0E2E4E6E8EAECCEEEBC1D3D5D7D9DBDDDFD1E3E
940 DATA 5E7EDEFE1939597999B9D9F9BE6C16191C1F2225282B2E3134373A3D4043
950 DATA 46494C4F5255585B5E6164676A6D707376797C7F82889A9D8BA0A3A6A985
960 DATA 97ACA0A3A6A98E8597ACA0A3A6A98E8597ACAF91889A9DA0A3A6A98E8597
970 DATA AC4F525558948BB5B29188A0A3A6A98E8597AC4F525558B5B2948BB8AFBB
980 DATA 918B9A9DA0A3A6A98E8597AC4F525558948BB5B288A0A3A6A98E8597AC4F
990 DATA 525558B5B2A0A3A6A98E8597AC8BBBEBEAD36E48D4603AD37E48D47034C54
1000 DATA 07AD06E48D4603AD07E48D47034C540720DE075749455649454C45205345
1010 DATA 43544F52454E20284D41582E2024464629203F9BEA20B507A200A0062014
1020 DATA 0820DE0741422057454C4348454D2053454B544F52203F9BEA203B10A5B2
1030 DATA 8D0A03A5B38D0B0320DE0741422057454C4348454D205350454943484552
1040 DATA 504C41545A203F9BEA203B10A5B28591A5B3859220DE07CC4553454E204F
1050 DATA 44455220D3434852454942454E203F9BEA20B507AD0001C953F00FC94CD0
1060 DATA D7A9528590A94085934CDD0FA9578590A9808593A9808D0803A9008D0903
1070 DATA A9318D0003A9018D0103A90FB0603A5918D0403A592BD0503A5908D0203
1080 DATA A5938D03032059E49810034C4C08C6BBF02118AD0A0369018D0A03AD0B03
1090 DATA 69008D0B0318A59169808591A592690085924CDD0F4C540720B507A200A0
1100 DATA 01201408A202A0002014086020BD0DA005EBEB201408A004E8201408A5B3
1110 DATA C5B5300EF0034C8310A5B2C5B430034C8310A000B1B491B220BD1020460D
1120 DATA 20CF104C711038A5B6E5B485B9A5B7E5B585BA18A5B965B285B2A5BA65B3
1130 DATA 85B3A000B1B691B220BD10C6B6A5B6C9FFD002C6B7C6B2A5B2C9FFD002C6
1140 DATA B34C9D10A5B6C5B4D00BA5B7C5B5D00568684C540760E6B4D002E6B56000
1150 DATA END
    
```

Apple II-BASIC-Wegweiser

Von Ekkehard Kaier; erschienen im Vieweg-Verlag, Braunschweig/Wiesbaden, ISBN 3-528-04259-1

Das vorliegende Wegweiser-Buch weist Wege zum erfolgreichen Einsatz von Mikrocomputern der Apple II-Familie wie Apple IIe, Apple II-Plus und sprachgleicher Systeme.

Der Autor vermittelt aktuelles Grundlagenwissen zur Datenverarbeitung:

- Was ist Hardware, Software und Firmenware?
- Was sind Großcomputer

- und Mikrocomputer?
- Was sind Datenstrukturen und Programmstrukturen?
- Was sind Betriebssysteme und Anwenderprogramme?
- Was heißt „fertige Programm-Pakete einsetzen“?
- Was beinhaltet das eigene Programmieren?

Das Wegweiser-Buch gibt eine erste Benutzungsanleitung:

- Wie bedient man den Apple II?

- Wie erstellt man das erste eigene Anwenderprogramm?

- Wie setzt man die verfügbaren Systemprogramme ein?

Außerdem enthält diese Ausgabe einen kompletten Programmierkurs in der Programmiersprache Applesoft-BASIC, der Wege zu grundlegenden Anwendungsmöglichkeiten weist:

- Programme mit Schleifen und Unterprogrammen.
- Text-, Tabellen- und Grafikverarbeitung.
- Formen der Tastatureingabe und Druckausgabe.
- Maschinennahe Programmierung in Assembler.
- Suchen, Sortieren, Mischen und Gruppieren von Daten.
- Sequentielle, direkte/random, index-sequentielle und verkettete Organisation einer Datei.

- Datei mit zeigerverketteter Liste und binärem Baum. Dieses Buch soll die vom Hersteller gelieferten System-Handbücher keinesfalls ersetzen, sondern ergänzen: In den Apple-Handbüchern werden Programmiersprachen beschrieben (z. B. Applesoft-BASIC Programmierhandbuch), Betriebssysteme (z. B. DOS-Handbuch), technische Eigenschaften (z. B. Apple IIe Benutzer-Handbuch) oder spezielle Geräte (z. B. Grafik Tablett Handbuch) und Software (z. B. Apple Writer).

Hier hingegen erfolgt eine Beschreibung der Grundlagen der Datenverarbeitung, um sie an zahlreichen Anwendungsmöglichkeiten für den Apple II zu demonstrieren und zu veranschaulichen.

Im Wegweiser-Buch sind 80 Programm-Beispiele als Codierung in Applesoft-BASIC (List) und als Ausführung (Run) wiedergegeben sowie vollständig beschrieben.

Da auf Programmiertricks verzichtet wurde, können die Programme leicht an andere BASIC-Dialekte angepaßt werden.

Die Abschnitte 2 und 3 bauen aufeinander auf und sollten in dieser Abfolge gelesen werden. Abschnitt 1 hingegen kann parallel hierzu bearbeitet werden.

Abschnitt 1:

COMPUTER
ALLGEMEIN

Abschnitt 2:

ARBEITSWEISE DES
APPLE II

Abschnitt 3:

PROGRAMMIERUNG
IN APPLESOFT-BASIC

Dieses Buch läßt sich auch als Nachschlagewerk benutzen. Aus diesem Grunde wurden das Inhaltsverzeichnis wie auch das Sachwortverzeichnis sehr detailliert aufgeli-

Lotus: System der Ursprungsfeststellung gegen Software-Piraterie

Die Lotus Development Corporation wird verstärkt den grauen Markt ihrer Produkte bekämpfen. In den USA wurde jetzt ein System vorgestellt, das mittels Strichcode-Scanner und Symphony-Software den Ursprung der Produkte auf jeder Ebene des Vertriebs feststellen und somit gegen Software-Piraterie eingesetzt werden kann. Dieses System wird zunächst ab April in den USA und später in der Bundesrepublik Deutschland eingeführt. Das System der Ursprung-

feststellung ist ebenso elegant wie einfach. Alle Lotus-Produkte werden mit einer Seriennummer in Strichcodierung versehen. Bei Lieferung der Produkte an den Distributor liest der Scanner die Seriennummer und vergleicht sie mit dem einmaligen Distributor-Code. Diese Information wird in einem IBM PC-XT eingelesen und täglich auf den VAX des Unternehmens eingegeben. Bei der Lieferung an die Händler wenden die Distributoren dasselbe Verfahren

an. Die Seriennummern werden mit den Codes der autorisierten Händler verglichen. Diese Daten werden dann an Lotus zurückübermittelt und gehen in die Hauptliste im VAX ein. Wird ein Produkt durch einen nicht autorisierten Verkaufspunkt angeboten, kann Lotus die Quelle des Lieferumschlags durch den Strichcode ganz leicht feststellen. Mit diesem System soll das umfassende Netz autorisierter Händler geschützt werden.

Directoryprinter

für Atari 400/800 mit min. 32 K
Atari 810

Option: Printer (Epson, MX, RX ...)

Dieses Programm ermöglicht es, die Directory der Diskette zu lesen. Desweiteren gibt es den Zustand des Files aus und gibt den Startsektor und die Länge des Files an.

Laden des Programms

1. Programmdiskette einlegen
2. LOAD „D:DIRPRINT.BAS“ eintippen
3. RUN eintippen
4. Programmdiskette herausnehmen und die zu bearbeitende Diskette einlegen.

Nun fragt der Computer, ob die Ausgabe auch auf dem Drucker ausgegeben werden soll.

Bei Druckerbenutzung „Y“ eingeben, wenn nicht, dann „N“.

Wenn Sie den Druckerbetrieb ausgeschaltet haben, dann erscheint nun der Disketteninhalt.

In der ersten Spalte steht der PROGRAMMNAME, EXTENSION, Zahl der SEKTOREN und der STARTSEKTOR. Zum Schluß wird dann noch die Zahl der freien Sektoren ausgegeben.

Dann drücken Sie die START-Taste, um wieder zum Programmanfang zu kommen.

Wenn Sie dagegen den Druckerbetrieb eingeschaltet ha-

ben, fragt der Computer nach der Zahl der Leerzeilen. Anschließend nach dem Diskettenamen. Er darf maximal 29 Buchstaben lang sein. Dann fragt er, ob Sie die Inhaltsangabe der Diskette anfertigen wollen. Entweder „Y“ oder „N“ eingeben. „Y“ ist nur notwendig, wenn Sie Ihre Disketten dokumentieren wollen und bei Disketten, bei denen die Directory nicht gelesen werden kann (Bootsdisketten, spezielle Datendisketten usw.).

Eingabebeispiel:

* BOOTDISK BOT 707 4
Anschließend fragt er, ob Sie

die Directory noch dazu ausgeben wollen oder nicht. „Y“ oder „N“ eingeben. Sie können aber auch „A“ oder „W“ eingeben. Beides führt zum wiederholten Abfragen der Eingaben (siehe Eingabebeispiel).

Bei der Abfrage nach Inhaltsangabe ist wie oben erwähnt mit „Y“ oder „N“ zu antwor-

ten. „N“ führt zur Kopie des Bildschirmes. Wenn nun alle Fileparts eingegeben sind, drücken Sie die START-Taste. Dies führt zu einem Neustart des Programms.

Falls ein Fehler auftritt, START drücken. Wenn dies nicht hilft, dann SYSTEM RESET drücken und RUN

eintippen. Wenn auch das nicht zum Erfolg führt, dann Rechner ausschalten und neu laden (siehe LADEN DES PROGRAMMS).

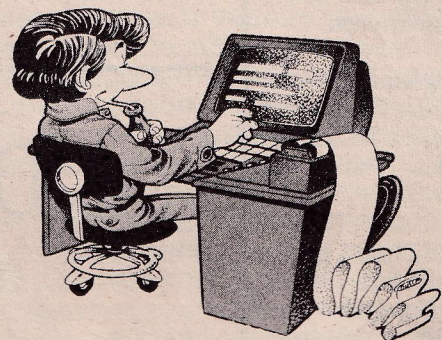
Beispiel für das Directory-Printer-Programm:

Testdiskette Nr. 1/A/II

Filename	Ext.	Sec.	Start-sec.
* DOS	SYS	39	4
* DUP	SYS	42	43
* SPACEINV	OBJ	35	85
* CHICKEN	BAS	69	270
* MINER	OBJ	132	120
* FROGGER	OBJ	159	435
* BUGOFF	OBJ	126	594
* RACE	OBJ	40	371
FREE SECTOR: 20			

```

1 REM
2 REM
3 REM
4 REM
5 PRI=0
6 TRAP 20000
8 POKE 764,255:POKE 752,1
9 T=360
10 DIM A$(1100),B$(128),C$(1100),D$(16),E$(20),F$(50)
11 GOSUB 2000
12 GOSUB 15000
13 IF PRI=1 THEN GOSUB 2000:GOSUB 2005
14 GOSUB 2000
15 GOSUB 10000
20 FOR I=203 TO 207
21 READ A:POKE I,A:NEXT I
22 DATA 104,32,83,228,96
30 B$(1)=" ":B$(128)=" "
40 ADDR=ADR(B$)
50 ADHI=INT(ADDR/256)
60 ADLO=ADDR-(ADHI*256)
70 DMS=768
80 POKE DMS+1,1
90 POKE DMS+2,82
100 POKE 773,ADHI
110 POKE 772,ADLO
120 T=T+1
121 IF T>368 THEN GOTO 144
130 POKE 779,INT(T/256)
131 POKE 778,T-(PEEK(779)*256)
140 I=USR(203)
141 A$(LEN(A$)+1,LEN(A$)+129)=B$
143 GOTO 120
144 GOSUB 21000
145 Z=0:GOTO 148
146 Z=Z+1
147 IF Z>(LEN(A$)/16)-1 THEN 160
148 D$=A$(Z*16+1,Z*16+16)
150 IF D$(1,1)="b" OR D$(1,1)="B" THEN C$(LEN(C$)+1)=D$
151 GOTO 146
160 E=LEN(C$)/16:X=0:GOTO 163
161 X=X+1
162 IF X>E-1 THEN 170
163 D$=C$(X*16+1,X*16+16)
164 GOSUB 1000
166 GOTO 161
170 ? :? "FREE SECTORS :";FREESECTOR
180 IF PRI=0 THEN 200
190 LPRINT "FREE SECTOR :";FREESECTOR
    
```



Hallo, liebe Rätselfreunde!

Die CPU-Redaktion prüft diesmal Ihr Computerwissen. Sehen Sie sich die Fragen an und schreiben Sie sich die Zahl vor der richtigen Antwort auf.

Am Ende werden alle Zahlen addiert.

Die so errechnete Zahl schreiben Sie bitte auf eine Postkarte und schicken diese an:

Tronic Verlag GmbH
Postfach 1
3444 Wehretal 1
Kennwort CPU-Happy

Vergessen Sie bitte nicht die Angabe Ihres Computersystems!

1. Preis

10 Kassetten aus unserem Software-Service

2. Preis

3 Kassetten aus unserem Software-Service

3. Preis

2 Kassetten aus unserem Software-Service

4.-30. Preis

1 Kassette aus unserem Software-Service

Welche Aufgabe hat ein Videocontroller?

- 1) Steuerung eines Videorecorders
- 2) Interface für Floppy-Laufwerk
- 3) Bildaufbau in einem Computer

Bedeutung von CP/M?

- 4) Neuer Computer der Firma Commodore
- 5) Diskettenbetriebssystem
- 6) Programmiersprache

Was ist ein Sprite?

- 7) Durch Hardware belegte Figur

8) Defektes Bit im Controller

9) Spezieller Sound-Effekt

Welche Funktion erfüllt ein Paddle?

- 10) Drehregler für analoge Eingaben
- 11) Andere Bezeichnung für Joysticks
- 12) Schnelle Sortieroutine

Welche Computersprache verwendet zum Zeichnen eine „Schildkröte“?

- 13) BASIC
- 14) LOGO
- 15) PASCAL

Wozu dient ein RESET?

- 16) Zum Löschen eines Graphikpunktes
- 17) Zum Rücksetzen des Computers
- 18) Zum Aufruf einer Interruptroutine

Welcher Teil eines Computers wird mit Akkumulator bezeichnet?

- 19) Pufferbatterie zur Datensicherung
- 20) Hauptregister des Prozessors
- 21) Speicherbereich für Daten

Was ist BIOS?

- 22) Ein Programm zur Ausgabe des Biorhythmus
- 23) Teil eines bekannten Betriebssystems
- 24) Befehl zur Ausgabe von Binärzahlen

Ein Netzwerk ist?

- 25) Die Zusammenschaltung von mehreren Computern
- 26) Verschachtelung von FOR NEXT-Schleifen
- 27) Die Benutzung vieler Programmebenen

Alle richtigen Lösungen nehmen an der Verlosung der Programmkassetten teil. Mitmachen darf jeder, mit Ausnahme der Mitarbeiter des Tronic-Verlages sowie deren Angehörige. Einsendeschluß ist der 10. 6. 1985.

Der Rechtsweg bleibt wie üblich ausgeschlossen.

ZX-Spectrum: Mit Maschinensprache zum Bildschirm

Dieser Kurs setzt voraus, daß die Grundzüge der Z 80-Assemblerprogrammierung bekannt sind.

Wer schon einmal versucht hat, Spiele oder andere Programme in Maschinensprache zu programmieren, der wird schnell feststellen, daß der Bildspeicher des Spectrum etwas merkwürdig aufgebaut ist. Wer zum Beispiel vorher mit einem ZX 81 gearbeitet hat, wird sich jetzt fragen, wie er etwas auf dem Bildschirm anzeigen kann. (Objekte, Graphik, Zahlen usw.). Die dazu nötigen Kenntnisse zu vermitteln, das ist das Ziel dieses Kurses. Um etwas auf den Bildschirm zu bringen, könnte man jetzt aus Zeile, Spalte und Punktreihe die Bildschirmadresse errechnen. Keine Angst, das haben wir nicht vor. Denn wer sich intensiv mit dem ROM des Spectrum befaßt, findet dort alles, was er braucht.

Nehmen wir also an, daß der Buchstabe A angezeigt wer-

den soll. Wie gehen wir vor? Dazu muß man wissen, daß der Spectrum die Möglichkeit hat, mit verschiedenen Datenkanälen zu arbeiten. Die Kanalnummer des Bildschirms ist 2, dies müssen wir dem Computer vor der Ausführung eines MC-Programmes mitteilen. Aber wie? Ganz einfach. Es ist eine Routine vorhanden, die das Setzen beliebiger Kanalnummern erlaubt.

Diese Routine steht bei 1601H 5633dez.

Bei Aufruf muß das Register A mit dem Wert des Kanals geladen sein, in diesem Falle mit 2. Also lautet die Befehlsfolge, um den Bildkanal zu setzen:

```
3 E 02      LD A 2
CD 01 16    CALL 1601
```

Von jetzt an erfolgen alle Ausgaben auf dem Bildschirm. Um nun das A auf den Schirm zu bringen, brauchen wir eine andere Routine. Diese wird mit einem RST-Befehl aufgerufen und steht bei 10H 16dez.

Um diese Routine benutzen zu können, muß Register A mit dem ASCII-Code des Zeichens geladen werden, im Falle des A mit dem Wert 65. Demnach lautet die Fortsetzung unseres Kurzprogrammes:

```
3E 41      LD A 41
D7         RST 10
C9         RET
```

Dieses kurze Programm können Sie an eine beliebige Stelle im RAM setzen, nach dem Start erscheint das A an der letzten PRINT-Position.

Nun wollen wir das A aber an eine bestimmte Stelle des Bildschirms setzen, auch dafür gibt es ein Programm im ROM.

Es befindet sich an der Adresse 0DD9H 3545dez und funktioniert wie die PRINT AT-Funktion. Die Register B und C sind mit den Koordinaten zu laden. Aber Vorsicht! Diese Werte entsprechen nicht der üblichen Nummerierung! 24 ist die oberste, 1 die unterste Zeile. Dieser

Wert wird in das B-Register geladen. Register C enthält die Spaltennummer, wobei 32 die linke und 2 die rechte Kante bezeichnet.

Um also das A an der Position 0,0 erscheinen zu lassen, ist folgendes Programm nötig:

```
3E 02      LD A 2
CD 01 16    CALL 1601
06 18      LD B 18
0E 21      LD C 21
CD D9 0D    CALL 0DD9
3E 41      LD A 41
D7         RST 10
C9         RET
```

Nach Ausführung des Programms erscheint das A an Position 0,0, und die PRINT-Position, wo das nächste Zeichen gedruckt werden würde, ist 0,1. Das Programm läuft folgendermaßen ab:

Die ersten beiden Zeilen setzen die Kanalnummer. Die nächsten 3 Zeilen setzen die Koordinaten. Die 3 Zeilen am Schluß bewirken die Ausgabe des Zeichens und den Rücksprung zum Basic.

Die Gewinner unseres Preisrätsels aus dem Monat April!

Verehrte Teilnehmer und Gewinner! Ein Lob vom CPU-Team – dieses Mal waren fehlende Computersystem-Angaben wirklich die Ausnahme.

Wenn man die Gewinnerliste betrachtet, fällt die internationale Beteiligung auf. Unsere Redaktion bekam zahlreiche Zuschriften aus Holland, Belgien, der Schweiz, Österreich, Luxemburg und Frankreich.

CPU-HAPPY als internationaler Wettbewerb!

1. Preis:

10 Kassetten aus unserem Software-Service.
Thomas Breitter, Bern (Schweiz).

2. Preis:

3 Kassetten aus unserem Software-Service.
Heinz Gödde, Brilon.

3. Preis:

2 Kassetten aus unserem Software-Service.
Lothar Maluche, Lindenberg im Allgäu.

4. – 30. Preis:

Kai-Uwe Frindt, Bad Oeynhaus. Ulrich Höfer, Burgkirchen/Alz. Klaus Barnert, Seesen. Heinz Kostka, Berlin. Gunnar Bargatzky, Mülheim/Ruhr. Thomas und Peter Küker, Bedburg. Siegfried Engelmann, Taunusstein. Rudolf Radecke, Minden. Rolf Genz, Bonn. J. Ruijs, SN Helmond (Holland). Hans Schöffeler, Kevelaer. Gerald Pfennich, Köflach (Austria). Axel Trocha, Düren-Birgel. Hans-Jürgen Henning, Marl-

Hils. Frau G. De Bruin, KH Eindhoven (Holland). Georg Vollweiter, Essen. Herbert Glaesner, Hof. Walter Eisheuer, St. Augustin. Leopold Luschnitz, Wien (Austria). Timo Plokarz, Schwanewede. Bernhard Ammerl, Ebermannstadt. Karl Bruckner, Krems (Austria). David Pfändler, Riniken (Schweiz). Marco Schwaiger, Reutte (Tirol). Markus Koojmann, Duisburg. Ralf Lutz, Stuttgart. Alexander Spörl, Eichstätt.

VIC-Clock für den VC-20

VIC-CLOCK ist ein Programm für den VC-20, das es ermöglicht, während des Programmierens die Uhrzeit in einer zusätzlichen Zeile anzuzeigen. Das Programm läuft auf allen Versionen, nur bei Version GV+3K ist die Einschränkung zu machen, daß die Erweiterung für BASIC unbrauchbar wird. Besitzer dieser und der GV-Version müssen vor dem Eintippen und Einladen den VC-20 der Erweiterungsversion anglei-

chen. Das geschieht mit:
POKE 648,16:SYS 64821
POKE 44,18:POKE 4608,0:
POKE 56,32:NEW

Wer zusätzlich 8K RAM in A000-BFFF besitzt, kann die CLOCK-Routine dort ablegen. Die Routine besteht aus einer Initialisierungsroutine, mit der die Uhr aufgerufen wird. Diese Routine muß auch nach Betätigung von STOP + RESTORE gestartet werden. Der zweite Teil dient

zum Anzeigen der Uhrzeit. Er wird 60mal pro Sekunde von der Interrupt-Routine aufgerufen.

Noch ein Hinweis zu VIC-CLOCK: Wenn man das Programm mit dem vom Ladeprogramm angegebenen SYS-Befehl gestartet hat, erscheint auf dem Bildschirm NEW. Die Routine muß durch RETURN ausgeführt werden, jedoch nur einmal. Falls VIC-CLOCK durch STOP + RE-

STORE unterbrochen wird, so muß es durch den entsprechenden SYS-Befehl wieder gestartet werden. Der NEW-Befehl wird jedoch nicht mehr benötigt. Er sollte umgangen werden, um das BASIC-Programm nicht zu zerstören.

Für den etwas erfahreneren Programmierer ist das Programm auch als Assembler-Listing abgedruckt.

Teil 1: BASIC-Loader

```

Ø REM    VIC CLOCK                VON          H. SPIEGEL
1 DATA169,160,162,73,141,21,3,142,20,3,169, 48,141,3,144,169,17,133,43,169
2 DATA 0,141,16,18,162,22,169,160,157,249, 17,173,134,2,157,249,149,202,208,242
3 DATA169,186,141,9,18,141,12,18,169,78,162, 69,141,119,2,142,120,2,169,13
4 DATA141,122,2,169,3,133,198,169,87,141,121,2,96,165,160,166,161,164,162,133
5 DATA251,134,252,132,253,162,255,232,56,165,253,233,192,133,253,165,252,233,75
6 DATA133,252,165,251,233,3,133,251,16,234,134,,24,165,253,105,192,133,253,165,2
52
7 DATA105,75,133,252,165,251,105,3,133,251,162,255,232,56,165,253,233,16,133,253
8 DATA165,252,233,14,133,252,165,251,233,,133,251,16,234,134,1,24,165,253,105
9 DATA16,133,253,165,252,105,14,133,252,162,255,232,56,165,253,233,60,133,253,16
5
10 DATA252,233,,133,252,16,240,134,2,162,2,181,,160,255,200,56,233,10,16
11 DATA250,105,10,149,,152,149,251,202,16,236,162,2,181,,24,105,176,72,181
12 DATA251,105,176,72,202,16,242,104,141,7,18,104,141,8,18,104,141,10,18,104
13 DATA141,11,18,104,141,13,18,104,141,14,18,76,191,234
50 PRINT"***** VIC - CLOCK *****"
60 PRINT"Ø VON H. SPIEGEL"
61 FORI=ØTO253:READZ:C=C+Z:NEXT:IFC=33661GOTO63
62 PRINT"Ø EINGABEBEFehler IN DEN DATA ZEILEN":END
63 RESTORE
65 IFPEEK(648)<>16THENPRINT"Ø FUNKTIONIERT NICHT AUF DIESER VERSION":END
68 PRINT"Ø STELLEN SIE DIE UHR HHMMSS" ; INPUTA$:TI$=RIGHT$(A$,6)
70 PRINT"Ø HABEN SIE 8K RAM IN A000-BFFF FREI ZUR VERFUEGUNG ? (J/N)"
75 GETA$:IFA$="J"THEN200
78 IFA$="N"THEN100
80 GOTO75
100 POKE56,PEEK(56)-1
110 A=PEEK(56)*256
120 FORI=ATO A+253:READJ:POKEI,J:NEXT
130 POKEA+1,PEEK(56)
140 PRINT"ØØ STARTEN MIT Ø SYS "A
190 END
200 FORI=40960TO40960+253:READJ:POKEI,J:NEXT
210 PRINT"ØØ STARTEN MIT Ø SYS 40960"

```

READY.

Teil 2: Assembler-Listing

```

B*
  PC  IRQ  AC  XR  YR  SP  N  V  #  B  D  I  Z  C
>; 9068 9364 00 90 00 F6 - - ↑ ↑ - - ↑ -
>
>, 7F00 A9 7F      LDA #$7F
>, 7F02 A2 49      LDX #$49
>, 7F04 8D 15 03 STA $0315
>, 7F07 8E 14 03 STX $0314
>, 7F0A A9 30      LDA #$30
>, 7F0C 8D 03 90 STA $9003
>, 7F0F A9 11      LDA #$11
>, 7F11 85 2B      STA $2B
>, 7F13 A9 00      LDA #$00
>, 7F15 8D 10 12 STA $1210
>, 7F18 A2 16      LDX #$16
>, 7F1A A9 A0      LDA #$A0
>, 7F1C 9D F9 11 STA $11F9,X
>, 7F1F AD 86 02 LDA $0286
>, 7F22 9D F9 95 STA $35F9,X
>, 7F25 CA          DEX
>, 7F26 D0 F2      BNE $7F1A
>, 7F28 A9 BA      LDA #$BA
>, 7F2A 8D 09 12 STA $1209
>, 7F2D 8D 0C 12 STA $120C
>, 7F30 A9 4E      LDA #$4E
>, 7F32 A2 45      LDX #$45
>, 7F34 8D 77 02 STA $0277
>, 7F37 8E 78 02 STX $0278
>, 7F3A A9 0D      LDA #$0D
>, 7F3C 8D 7A 02 STA $027A
>, 7F3F A9 04      LDA #$04
>, 7F41 85 C6      STA $C6
>, 7F43 A9 57      LDA #$57
>, 7F45 8D 79 02 STA $0279
>, 7F48 60          RTS
>, 7F49 A5 A0      LDA $A0
>, 7F4B A6 A1      LDX $A1
>, 7F4D A4 A2      LDY $A2
>, 7F4F 85 FB      STA $FB
>, 7F51 86 FC      STX $FC
>, 7F53 84 FD      STY $FD
>, 7F55 A2 FF      LDX #$FF
>, 7F57 E8          INX
>, 7F58 38          SEC
>, 7F59 A5 FD      LDA $FD
>, 7F5B E9 C0      SBC #$C0
>, 7F5D 85 FD      STA $FD
>, 7F5F A5 FC      LDA $FC
>, 7F61 E9 4B      SBC #$4B
>, 7F63 85 FC      STA $FC
>, 7F65 A5 FB      LDA $FB
>, 7F67 E9 03      SBC #$03
>, 7F69 85 FB      STA $FB
>, 7F6B 10 EA      BPL $7F57
>, 7F6D 86 00      STX $00
>, 7F6F 18          CLC
>, 7F70 A5 FD      LDA $FD
>, 7F72 69 C0      ADC #$C0
>, 7F74 85 FD      STA $FD
>, 7F76 A5 FC      LDA $FC
>, 7F78 69 4B      ADC #$4B
>, 7F7A 85 FC      STA $FC
>, 7F7C A5 FB      LDA $FB
>, 7F7E 69 03      ADC #$03
>, 7F80 85 FB      STA $FB
>, 7F82 A2 FF      LDX #$FF
>, 7F84 E8          INX
>, 7F85 38          SEC
>, 7F86 A5 FD      LDA $FD
>, 7F88 E9 10      SBC #$10
>, 7F8A 85 FD      STA $FD
>, 7F8C A5 FC      LDA $FC
>, 7F8E E9 0E      SBC #$0E
>, 7F90 85 FC      STA $FC
>, 7F92 A5 FB      LDA $FB

```

>, 7F94 E9 00	SBC #\$00	>, 7FC9 69 0A	ADC #\$0A
>, 7F96 85 FB	STA \$FB	>, 7FCB 95 00	STA \$00,X
>, 7F98 10 EA	BPL \$7F84	>, 7FCD 98	TYA
>, 7F9A 86 01	STX \$01	>, 7FCE 95 FB	STA \$FB,X
>, 7F9C 18	CLC	>, 7FD0 CA	DEX
>, 7F9D A5 FD	LDA \$FD	>, 7FD1 10 EC	BPL \$7FBF
>, 7F9F 69 10	ADC #\$10	>, 7FD3 A2 02	LDX #\$02
>, 7FA1 85 FD	STA \$FD	>, 7FD5 B5 00	LDA \$00,X
>, 7FA3 A5 FC	LDA \$FC	>, 7FD7 18	CLC
>, 7FA5 69 0E	ADC #\$0E	>, 7FD8 69 B0	ADC #\$B0
>, 7FA7 85 FC	STA \$FC	>, 7FDA 48	PHA
>, 7FA9 A2 FF	LDX #\$FF	>, 7FDB B5 FB	LDA \$FB,X
>, 7FAB E8	INX	>, 7FDD 69 B0	ADC #\$B0
>, 7FAC 38	SEC	>, 7FDF 48	PHA
>, 7FAD A5 FD	LDA \$FD	>, 7FE0 CA	DEX
>, 7FAF E9 3C	SBC #\$3C	>, 7FE1 10 F2	BPL \$7FD5
>, 7FB1 85 FD	STA \$FD	>, 7FE3 68	PLA
>, 7FB3 A5 FC	LDA \$FC	>, 7FE4 8D 07 12	STA \$1207
>, 7FB5 E9 00	SBC #\$00	>, 7FE7 68	PLA
>, 7FB7 85 FC	STA \$FC	>, 7FE8 8D 08 12	STA \$1208
>, 7FB9 10 F0	BPL \$7FAB	>, 7FEB 68	PLA
>, 7FBB 86 02	STX \$02	>, 7FEC 8D 0A 12	STA \$120A
>, 7FBD A2 02	LDX #\$02	>, 7FEF 68	PLA
>, 7FBF B5 00	LDA \$00,X	>, 7FF0 8D 0B 12	STA \$120B
>, 7FC1 A0 FF	LDY #\$FF	>, 7FF3 68	PLA
>, 7FC3 C8	INY	>, 7FF4 8D 0D 12	STA \$120D
>, 7FC4 38	SEC	>, 7FF7 68	PLA
>, 7FC5 E9 0A	SBC #\$0A	>, 7FF8 8D 0E 12	STA \$120E
>, 7FC7 10 FA	BPL \$7FC3	>, 7FFB 4C BF EA	JMP \$EABF
		>	

Auflösung des Rätsels aus der Mai-Ausgabe der CPU

Dies ist die richtige Lösung:
– CPU, das kompetente Anwendermagazin –

Die Codierung sah wie folgt aus: Byte = 3; A = ASC II A-64. Das bedeutete: Byte = 8; 8 - 3 = 5; es handelte sich dabei

um eine 5-Bit-Darstellung. A = ASC II A-64; ASC II A = 65; 65 - 64 = 1; A besaß also

den Wert 1. Mit Hilfe der Zahlen 5 und 1 war es dann möglich, den Lösungssatz zu decodieren.

Computertest: CE-TEC MPC-80

Nach dem Auspacken präsentiert sich der CE-TEC MPC-80 in einem formschönen, farblich gut abgestimmten Gehäuse. Ebenfalls positiv zu vermerken ist, daß das Netzteil im Gehäuse integriert wurde.

Die Inbetriebnahme des Computers war relativ problemlos. Antennenkabel anschließen und einschalten. Beim Testen der Tastatur und beim Eintippen des ersten Programmes fiel uns einiges Erwähnenswertes auf. Der MSX verfügt über eine deutsche Tastatur, d. h. aus ae wird ä, aus ue wird ü usw. Als Commodore- und Sinclair-Benutzer tippten wir anfangs beim Z und beim Y des Öfteren daneben, man gewöhnt sich jedoch schnell an die neue Anordnung.

Was passiert nach dem Einschalten?

Die Firma Microsoft präsentiert sich in weißer Schrift auf blauem Grund – und dies gleich zweimal! Denn erst nachdem das Titelbild auf dem Screen einige Sekunden zu erkennen war, erscheint die OK-Meldung mit der Angabe 28815 BYTES FREE. In der untersten Bildschirmzeile tauchen nun folgende Begriffe auf: Color – auto – goto – list – run. Mit diesen Funktionen sind die Funktionstasten F1 – F5 belegt.

Nach dem Drücken der SHIFT-Taste verändert dies sich wie folgt: Color (womit Color 15, 4, 4 gemeint ist) – CLOAD – cont – list – run. Hier handelt es sich um die Funktionen, mit denen die Funktionstasten F6 – F10 belegt sind. Die Funktionen der Tasten F6 – F10 werden bis auf CLOAD – sofort ausgeführt, ohne daß man dabei die Return-Taste betätigen muß.

Der COLOR-Befehl auf der F6-Taste ermöglicht es, nach einem Spiel oder Programm in den Einschaltfarbmodus zu gelangen.

Besonders erwähnenswert sind die vier großen Cursor-Tasten, dadurch ist es möglich, den Cursor in acht verschiedene Richtungen zu steuern!

Was bietet das BASIC?

Der Befehlssatz ist sehr umfangreich. Die Grafik- und Soundeigenschaften des MSX-Computers werden durch das BASIC intensiv unterstützt. Variablennamen können mehrere Zeichen lang sein, wovon aber nur die ersten beiden Zeichen für die Auswertung relevant sind.

Einige BASIC-Befehle:

AUTO (Anfangszeile, Schriftweite) bewirkt eine automatische Zeilenummerierung.

KEY LIST zeigt die Belegung der Funktionstasten.

ERASE (Feldvariable, Feldvariable ...) löscht mit DIM definierte Variablen.

SWAP (Variable, Variable) vertauscht zwei Variablen miteinander; z. B. A = 5, B = 10, SWAP A, B; danach hat A den Wert 10 und B den Wert 5.

WIDTH (Wert) bestimmt im Textmodus die Anzahl der angezeigten Zeichen pro Zeile.

Key	{	ON	Anzeigen
		OFF	oder Löschen
			der Funktions-
			tasten-
			Belegung
			(auf dem
			Bildschirm)

MAX FILES (Ausdruck) bestimmt die Anzahl der Dateien, die von einem Programm eröffnet werden können.

MOTOR	{	ON	schaltet
		OFF	den Motor
			des
			Cassetten-
			recorders
			ein oder
			aus

VPOKE (Adresse, Ausdruck) poked in das Video-RAM.

Positiv zu bewerten ist die Arbeitsgeschwindigkeit des Computers, so daß auch komplexe Programme, z. B. Spielprogramme mit angemessener Geschwindigkeit ablaufen.

Das BASIC ermöglicht die Programmierung von INTERRUPTS, d. h., der Programmablauf wird unterbrochen und zu einem Unterprogramm verzweigt. Nach einmaliger Definition des Interrupts ist ein weiterer Programmaufruf unnötig.

Die Funktions-, Cursor-, Edittier- und Stoptaste sind von der übrigen Tastatur abgesetzt. Die Tasten selbst sind bis zu 5mal belegt, zum Umschalten sind 3 Umschalt-Tasten vorhanden. Der MSX-Computer ist mit einem Bildschirm-Editor ausgestattet, der mit dem des C-64 vergleichbar ist.

Über welche Anschlußmöglichkeiten verfügt der CE-TEC MPC-80?

Es ist ein Modulschacht vorhanden, in den Spiel- und Erweiterungsmodule eingesetzt werden können, außerdem ein Expansions-Bus, der mit dem Modulschacht identisch ist.

Von Nachteil ist der eingebaute Centronicsanschluß, an den z. B. Epson- oder Seikosha-Drucker nur mittels Adapter anschließbar sind. Diese Adapter sind zur Zeit im Handel nur schwer erhältlich, so daß man normalerweise auf die MSX-Drucker angewiesen ist. Die Video- und Audiosignale stehen durch das Vorhandensein von zwei Chinchbuchsen zur Verfügung. Als erfreulich anzumerken ist die Berücksichtigung von zwei allgemein gebräuchlichen Joystickports, was nicht der Regelfall ist.

Das sichere Tastaturgefühl wird durch ein Klicken aus dem Fernsehlautsprecher unterstützt. Die 10 Funktionstasten lassen sich mit je bis zu 15 Zeichen langen Ausdrücken belegen.

Computer-Handbuch:

Es ist mittlerweile keine Ausnahme mehr, daß die Computer-Anleitungen oft eine schlechte Qualität aufweisen oder nur in englischer Sprache vorhanden sind. Beim MSX-Computer konnten wir bisher noch kein Handbuch erhalten! Nach Auskunft des Herstellers ist eine Anleitung in Arbeit.

Wir mußten also diesen Computer ohne jegliche Informationen testen.

Cassettenrecorder:

Der Umgang mit dem Cassettenrecorder gestaltete sich am Anfang etwas schwierig, da eine präzise Lautstärkeneinstellung aufgrund der fehlenden Ladekontrolle kaum möglich ist. Lesefehler werden nicht erkannt! Nach Beginn des Ladevorganges erscheint, vorausgesetzt die Lautstärke ist korrekt, FOUND „Programmname“. Danach geschieht bis zum Ende der Ladezeit nichts mehr. Am Anfang passierte es uns des öfteren, daß ein Programm zwar geladen, aber nicht ausgeführt werden konnte, da es voller Lesefehler war. Eine Empfehlung für den ernsthaften Anwender: Die Anschaffung einer Floppystation lohnt sich!

Bei konsequenter Ausnutzung sämtlicher sich bietender Möglichkeiten lassen sich Programme schreiben, die über die normale Geschwindigkeit des Microsoft-BASIC hinausgehen.

Fazit:

Mit dem CE-TEC MPC-80 erhält man einen Computer mit einem relativ günstigen Preis-Leistungs-Verhältnis. Grafik, Sound und BASIC können überzeugen. Negativ festzuhalten bleibt das fehlende Handbuch und das unsichere Arbeiten mit dem Cassettenrecorder. Ob sich der MSX-Computer als Standard auf dem Markt durchsetzen wird, werden die nächsten Monate zeigen.

Kassettenservice

Tronic-Verlag
Landstraße 29
3444 Wehretal 1

Für unsere Leser, die Überraschungen lieben, hat sich unsere Redaktion etwas einfallen lassen. Überzeugen Sie sich selbst. Wir haben Ihnen einige **Super-Angebote** zusammengestellt:

**Unser
Paketservice**

Spitzen-Programme zum Spitzen-Preis:

Bestell-Nr. 100	kleines Programm-Paket	3 bespielte Kassetten DM 27,50
Bestell-Nr. 110	großes Programm-Paket	3 bespielte Disketten DM 48,-
Bestell-Nr. 200	exklusive Disketten-Box	8 bespielte Kassetten DM 64,50
Bestell-Nr. 210	exklusive Disketten-Box	8 bespielte Disketten DM 99,-
		inkl. 8 bespielte Disketten
		zum Preis von DM 148,-
		inkl. 10 Leerdisketten
		zum Preis von DM 99,-

Dieses Angebot gilt nur für Commodore 64, Atari, TI-99 und Apple!

Computer-Messen '85

22. 5.-24. 5. 85	Komm 85/Berlin
4. 6.- 8. 6. 85	Internationale Computer-Schau/Köln
30. 8.- 8. 9. 85	Internationale Funkaus- stellung Berlin
24. 9.-27. 9. 85	CAMP/Computer Graphics/Berlin
9. 10.-13. 10. 85	Hobby-Elektronik 85/ Stuttgart
9. 10.-14. 10. 85	Frankfurter Buchmesse
17. 10.-20. 10. 85	CFA - Commodore Fachausstellung/Frankfurt
28. 10.- 1. 11. 85	SYSTEMS 85/München
12. 11.-16. 11. 85	PRODUCTRONICA Intern. Fachmesse für die Fertigung in der Elektronik/München

Computermessen im Ausland

1. 6.- 4. 6. 85	CES - Consumer Electronic- Show/Chicago, USA
11. 6.-15. 6. 85	Nor-Com - Internationale Konferenz und Ausstellung für Kommunikation/Oslo
Juni 85	PET World Show/London
September 85	SICOB/Paris
18. 9.-22. 9. 85	PCW/London
Oktober 85	COMDEX/Las Vegas, USA
November 85	ELKOM - Internationale Elektronik-Ausstellung/Hel- sinki

Die Home-Computer-Serie MZ-800 von Sharp mit erweiterter Grafikfähigkeit, Musikfunktion, erweiterter Peripherie:

Mit der MZ-800-Serie präsentiert Sharp ein zusätzliches Homecomputer-System. Aufbauend auf die erfolgreiche MZ-700-Serie hat man mit der MZ-800-Serie eine Programmerweiterung geschaffen, die dem Wunsch nach einer weiteren Funktionsverbesserung z. B. der Grafikfähigkeit berücksichtigt.

Die Auflösung liegt nun bei 640 x 200 Punkten bei 16 möglichen Farben, die alternativ auf einem PAL-Fernseher oder einem 14"-Farb-Monitor dargestellt werden können. Für den Farb-Ausdruck bietet Sharp in der umfangreichen Peripherie einen 80-Farben-Punktmatrix-Drucker.



Ebenfalls interessant ist das Diskettenlaufwerk „Disk-Drive/Quick-Disk“ mit 2 x 64 KB, das in der Lage ist, das Betriebssystem von 42 KB innerhalb von 5 Sekunden und eine Seite der 2,8"-Minidiskette mit 64 KB in 8 Sekunden einzuladen und abzuspeichern. Das Diskettenlaufwerk kann alternativ zu dem serienmäßig integrierten Cassettenrecorder genutzt werden, wobei Programme vom Cassettenrecorder auf die Mini-Disketten überspielt werden können. Für MZ-700-Anwender ist dieses Laufwerk ebenfalls anwendbar. Dies gilt übrigens für die gesamte Peripherie und das gesamte Software-Angebot, das für beide Serien kompatibel ist.

In der Grundausstattung bietet die MZ-800-Serie eine RAM-Speicherkapazität von 64 KB, die erweiterbar ist auf

128 KB und 16-KB-ROM-Speicher. Die Grundeinheit des MZ-821 umfaßt den Z-80A-Mikroprozessor, einen 4-KB-Monitor, ein Keyboard mit ASCII-Standard-Tastatur und zahlreichen Funktionstasten, einen Cassettenrecorder mit 1200 Baud Transferrgeschwindigkeit, ei-

nen Musikgenerator mit einem Umfang von 3 Harmonien und 6 Oktaven, einen Joy-Stick-Anschluß sowie zahlreiche Anschluß-Interfaces für Drucker, Bildschirme und ein zusätzliches 5 1/4"-Floppy-Disk-Laufwerk für max. 4 Mini-Floppy à 320 KB.

Die Schnittstelle RS-232C kann sowohl vom Quick-Disk als auch von der Mini-Floppy gesteuert werden.

Das umfangreiche Peripherieangebot umfaßt, neben den Mini-Floppy-Disks und der Quick-Disk-Farb-Monitore, einen 12"-Farb-Monitor mit grüner Darstellung, Farb-Matrix-Drucker und Vierfarb-Plotter.

Der MZ-800 kostet in der Version MZ-821 ab DM 1198,- inkl. MwSt.

Maxell's the disk that many drive manufacturers trust to put new equipment through its paces. It's that bug-free. So you can drive a bargain. But in accelerated tests, Maxell floppys lead the industry in error-free performance and durability. Proving that if you can't stand the heat you don't stand a chance.

maxell
IT'S WORTH IT.

NEU Der NEWMAN Beratungs-Katalog

Rund 1.000 Angebote. Alles von COMMO-DORE, Sinclair, Dragon, Sharp, Spectra-video und anderen mit Original-Werks-Garantie. SOFORT LIEFERBAR. Ob Hardware, Peripherie, Bücher, Programme oder Zubehör, Sie erhalten alles aus einer Hand.

Sofort GRATIS anfordern

Teilzahlung, technischer Service, BERATUNG.

Gutschein für 1 Katalog
Ausfüllen, ausschneiden, auf Postkarte kleben MU11 und absenden.

Name/Vorname _____
Straße/Nr. _____
PLZ/Ort _____

100 Seiten dick

ALLES FÜR IHREN HOME-COMPUTER

NEWMAN Computer-Versand Postfach 50 11 26,
2000 Hamburg 50, Tel. 040/850 60 71

Wie funktioniert der Videoprozessor bei den MSX Computern?

Im MSX BASIC existieren Befehle wie VPOKE, VPEEK, BASE und VDP, womit die meisten User nichts anfangen können, da im Handbuch hierzu nur sehr unzureichende Informationen zu finden sind.

Aber gerade mit diesen Befehlen eröffnen sich Möglichkeiten, von denen die meisten MSX Benutzer keine Ahnung haben.

So ist zum Beispiel der zweite Textmodus (24 Zeilen mit 32 Zeichen) des MSX BASICs eigentlich ein Grafikmodus, worin Grafiken in 16 Farben erzeugt werden können. Auch die Sprites können in diesem Modus angesprochen werden.

Die Grafiken in diesem Quasi Textmodus können mit PRINT geschrieben und somit sehr schnell bewegt werden. Ferner sind bis zu 14 Grafikseiten möglich, worin „geblättert“ werden kann.

In den meisten MSX Handbüchern wird dies jedoch vollkommen verschwiegen. Wir zeigen deshalb, wie man diese Grafiken erzeugt.

Zuerst müssen wir uns die Struktur des bei MSX verwendeten Videoprozessors ansehen:

Der MSX Standard verwendet den TMS9929A Video Display Processor (VDP) von Texas Instruments.

Dieser Videoprozessor verwaltet 16 k Videoram. Dieser Videoram ist mit dem Videoprozessor verdrahtet und kann deshalb von der CPU nicht angesprochen werden. MSX hat deshalb die zusätzlichen Befehle VPOKE und VPEEK, die über den VDP gesteuert werden.

Eine MSX Konsole mit 64 kB RAM hat also in Wirklichkeit 64 + 16 kB RAM, also 80 kBytes.

Der TMS9929A VDP hat 9 Register.

Davon sind die ersten 8 Register nur Schreibregister, d. h. die Werte in diesen Registern können nicht gelesen werden. MSX legt jedoch eine Kopie der Registerwerte ab, so daß der Benutzer auch den Inhalt dieser Register erfahren kann.

Das neunte Register ist das sogenannte „Status“ Register. Dieses Register kann nur gelesen werden. Es enthält wichtige Informationen über den Zustand der Sprites.

Wir numerieren die Register 0 bis 7 als Schreibregister und das Statusregister als Nummer 8. Diese Nummern entsprechen dann genau den Wertzuweisungen der VDP Anweisung im MSX BASIC.

Wir schalten den Textmodus mit SCREEN 1 ein. Unsere MSX Konsole ist jetzt auf den Textmodus mit 24 mal 32 Zeichen und einer Spritegröße von 8 mal 8 Punkten eingestellt.

Zuerst ist jetzt das VDP Register 1 wichtig, denn darin ist die Spritegröße gespeichert.

Registerinhalt &hE0 = Sprites in der 8 x 8 Matrix unvergrößert,

Registerinhalt &hE1 = Sprites in der 16 x 16 Matrix unvergrößert,

Registerinhalt &hE2 = Sprites in der 8 x 8 Matrix vergrößert,

Registerinhalt &hE3 = Sprites in der 16 x 16 Matrix vergrößert.

Die Spritegröße wird in MSX BASIC normalerweise mittels der SCREEN Anweisung gewählt, was jedoch eine komplette Neuinitialisierung des entsprechenden Videoram-Bereiches hervorruft.

Will man also in einem laufenden Programm die Spritegröße verändern, so muß das Sprite anschließend wieder neu definiert werden.

Durch einen Eintrag in VDP Register 1 kann jedoch die Spritegröße beliebig verändert werden, ohne daß hierbei die Spritecharakteren verloren gehen.

Wollen Sie z. B. auf Sprites mit einer 16x16 Matrix vergrößert umschalten, so geschieht dies mittels der Anweisung:

VDP(1) = &hE3

Es sollten jedoch nur die vier zuvor angegebenen Werte in das Register 1 eingeschrieben werden, da andere Werte den Bildschirmstatus verändern.

VDP Register 2 gibt die Startadresse der Bildschirmabbildung wieder. Der Wert in Register 2 wird mit &h400 multipliziert, um diese Adresse zu erhalten.

Es sind Werte zwischen &h00 und &h0F zulässig.

In unserem 24x32 Zeichen Modus liegt dieser Wert standardmäßig bei &h06, was Startadresse &h1800 in Video Ram entspricht.

Wird der Wert verändert, so wird auf eine Abbildung umgeschaltet, die der neuen Startadresse entspricht.

Man kann mit der Anweisung: VPOKE VDP(2) * &h400 + 32 * Zeile + Spalte, Ascii-Wert immer das entsprechende Zeichen an der gewünschten Bildschirmposition ausgeben, egal, wo die Abbildung gerade liegt.

Wollen Sie z. B. das Zeichen „A“ in Zeile 5 und Spalte 8 ausgeben, so sieht die Anweisung wie folgt aus: VPOKE VDP(2) * &h400 + 32 * 5 + 8,65

und das „A“ steht exakt an der gewünschten Position, gleichgültig, in welchem Bereich des Videorams sich die Abbildung gerade befindet.

Haben Sie auf diese Weise mehrere Abbildungen in den Videoram gelegt, so können Sie durch eine Eintragung in das VDP Register 2 zwischen diesen Abbildungen umschalten:

VDP(2) = &h00 schaltet auf die Abbildung ab Adresse &h000,

VDP(2) = &h01 schaltet auf die Abbildung ab Adresse &h400,

VDP(2) = &h02 schaltet auf die Abbildung ab Adresse &h800.

usw.

VDP Register 3 gibt die Startadresse der Farbtabelle an. Dieser Wert in Register 3 wird mit &h40 multipliziert, um die Startadresse zu finden. Der Wert darf im Bereich &h00 bis &hFF liegen. Im 24x32 Zeichen Textmodus ist die Farbtabelle 32 Bytes lang. Die ASCII Tabelle wird hierbei in 32 Blöcke zu je 8 Zeichen eingeteilt, worin jedem Block jeweils eine Vordergrund- und eine Hintergrundfarbe zugeordnet wird. Die Farben entnehmen Sie bitte Ihrem MSX Handbuch, wobei die Farbe 0 transparent nachfolgend mit &h0 und die Farbe 15 weiß nachfolgend mit &hF bezeichnet wird.

Fortsetzung S. 48

Die Farbtabelle ist wie folgt aufgebaut:

Byte Nr.	Zeichen Nr.
0	0 bis 7
1	8 bis 15
2	16 bis 23
3	24 bis 31
4	32 bis 39
5	40 bis 47
6	48 bis 55
7	56 bis 63
8	64 bis 71
9	72 bis 79
10	80 bis 87
11	88 bis 95
12	96 bis 103
13	104 bis 111
14	112 bis 119
15	120 bis 127

Byte Nr.	Zeichen Nr.
16	128 bis 135
17	136 bis 143
18	144 bis 151
19	152 bis 159
20	160 bis 167
21	168 bis 175
22	176 bis 183
23	184 bis 191
24	192 bis 199
25	200 bis 207
26	208 bis 215
27	216 bis 223
28	224 bis 231
29	232 bis 239
30	240 bis 247
31	248 bis 255

Jedem Block von je 8 Zeichen kann jetzt eine Vordergrund- und Hintergrundfarbe zugeordnet werden, wobei in den 4 höherwertigen Bits die Vordergrund-, und in den 4 niederwertigen Bits die Hintergrundfarbe steht.

Die Anweisung zum Farbeintrag sieht wie folgt aus:

VPOKE VDP(3) * &h40 + Byte Nr., Farbe

Wollen Sie zum Beispiel dem Zeichenblock 80 bis 87 die Farbe Mittelrot auf schwarzem Hintergrund zuordnen, so sieht die Anweisung wie folgt aus: VPOKE VDP(3) * &h40 + 10, &h81

Die Farbe Mittelrot ist &h8 und die Farbe Schwarz ist &h1.

Versuchen Sie einmal folgende Anweisung im Direkt-

gabemodus: VPOKE VDP(3) * &h40 + 31, &hA8

Sie werden sehen, daß jetzt der Cursor Rot erscheint und das entsprechende Zeichen im Cursor Gelb.

Dies funktioniert natürlich nur, wenn Sie sich im 24x32 Zeichen Modus (SCREEN 1) befinden.

Sie sehen also, daß im zweiten Textmodus (SCREEN 1) des MSX BASICs alle 16 Farben möglich sind.

Register 4 des Videoprocessors gibt die Startadresse des Zeichengenerators im Videoram an. Der Inhalt des Registers wird mit &h800 multipliziert und muß im Bereich &h00 bis &h07 liegen. Die Startadressen des Zeichengenerators sind:

Registerwert	Startadresse
&h00	&h0000
&h01	&h0800
&h02	&h1000
&h03	&h1800

Registerwert	Startadresse
&h04	&h2000
&h05	&h2800
&h06	&h3000
&h07	&h3800

Die Startadresse des Zeichengenerators gibt die erste Punktzeile des ASCII Zeichens 0 an.

Der Zeichengenerator besteht aus 8*256 Bytes und enthält die Bitmuster der darstellbaren Zeichen.

Durch Einträge in den Zeichengenerator lassen sich Zeichen verändern oder selbst definieren. Zur Definition sehen wir uns den Aufbau eines Zeichens einmal genau an:

Ein Zeichen besteht aus 8 Zeilen je 8 Punkten. Wir bezeichnen die erste Zeile eines Zeichens als Zeile 0 und die letzte Zeile des Zeichens als Zeile 7.

Um ein Zeichen zu definieren, sind 8 VPOKE Anweisungen erforderlich, und zwar

für jede Zeile des Zeichens eine Anweisung, geben Sie z. B. folgende Anweisung:

VPOKE VDP(4) * &h800 + 32 * 8 + 3, &hFF so wird im Leerzeichen jetzt in der 3. Zeile ein schwarzer horizontaler Strich erscheinen.

Die Anweisung ist wie folgt aufgebaut:

VPOKE VDP(4) * &h800 + 8 * Zeichennummer + Zeichenzeile, Umbelegungscode. Wie Sie den Umbelegungscode ermitteln, erfahren Sie in Ihrem MSX Handbuch unter der Anweisung SPRITES\$.

Es ist auf diese Weise möglich, z. B. einen zweiten Zeichensatz zu definieren und durch Änderung des Registerinhaltes in VDP Register 4 zwischen dem normalen Zeichensatz und dem selbst erstellten Zeichensatz umzuschalten. Dazu sind jedoch bereits sehr leistungsfähige Programme auf dem Markt, die diese Arbeit der Zeichendefinition hervorragend bewältigen.

Sie sehen also, daß mittels der Zeichendefinition und der Farbcodetabelle im 24x32 Zeichen Textmodus ohne weiteres Grafiken erzeugt werden können.

Das Register 5 des Videoprocessors gibt die Startadresse der Spriteattributliste wieder. Diese Spriteattributliste enthält die Positionen, Farben und Charakter der auf dem Bildschirm sichtbaren Sprites.

Um die Startadresse zu erfahren, ist der Registerwert mit &h80 zu multiplizieren.

Die Spriteattributliste beinhaltet 32*4 Bytes, wobei jedem Sprite 4 Bytes zugeordnet sind.

In den 4 Bytes für jedes Sprite sind abgelegt:

Byte 1: Zeile des Sprites,
Byte 2: Spalte des Sprites,
Byte 3: Charakternummer des Sprites,
Byte 4: Farbe des Sprites.

Einträge in die Spriteattributliste machen ein Sprite sofort auf dem Bildschirm sichtbar.

Die Adressen errechnen sich wie folgt:

VPOKE VDP(5) * &h80 + 4 * Spritenr, Zeile,
VPOKE VDP(5) * &h80 + 4 * Spritenr + 1, Spalte,
VPOKE VDP(5) * &h80 + 4 * Spritenr + 2, Charakter,
VPOKE VDP(5) * &h80 + 4 * Spritenr + 3, Farbe.

Der Charakter des MSX BASICs muß bei 16x16 Sprites mit 4 multipliziert werden, da hierzu immer 4 Charaktere zu einem Sprite zusammengefaßt werden.

Es funktionieren jedoch auch im 24x32 Zeichen Textmodus die PUT SPRITE-Befehle.

Interessant ist eigentlich nur das 3. Byte eines jeden Sprites in der Spriteattributliste, da man hier durch ständige Veränderung des Charaktereintrages das Sprite in sich bewegen kann. Das geht wesentlich schneller als mittels der PUT SPRITE Anweisung, womit ja alle 4 Werte des Sprites geschrieben werden.

Das VDP Register 6 enthält die Startadresse des Spritegenerators. Dieses Register ist genau so zu behandeln wie das Register 4 für den Zeichengenerator. Wir wollen deshalb hier nicht näher darauf eingehen. Interessant ist jedoch die Anweisung: VDP(6) + VDP(4) = welche den Spritegenerator dem Zeichengenerator überlagert. Nach dieser Anweisung werden nämlich die Spritecharaktere aus der Zeichengeneratorliste gelesen.

Im letzten Schreibregister des Videoprocessors sind die Bildschirmfarben abgelegt.

Im 24x40 Zeichen Textmodus geben die 4 höherwertigen Bits die Vordergrundfarbe und die 4 niederwertigen Bits die Hintergrundfarbe an. Im 24x32 Zeichen Modus hingegen haben die 4 höherwertigen Bits keine Bedeutung, aber in den 4 niederwertigen Bits ist die Randfarbe des Bildschirms angegeben. Ist einem Charakterblock die Hintergrundfarbe transpa-

rent (0) zugeordnet, so erscheint die Randfarbe als Hintergrundfarbe.

Die Anweisung: VDP(6) = &h8 schaltet die Randfarbe auf Mittelrot um. Dies wirkt sich auch auf alle Charakterblöcke mit der Hintergrundfarbe transparent aus, die jetzt ebenfalls Rot als Hintergrundfarbe erhalten.

Mittels der BASE Anweisung lassen sich ebenfalls die Bereiche der jeweiligen VDP Tabellen verändern.

Auf die Adressen der BASE Anweisung wird jedoch von MSX BASIC zugegriffen. Der richtige Zugriff nach Veränderung der BASE Adressen erfolgt jedoch erst, nachdem der Bildschirmmodus mittels der SCREEN Anweisung neu initialisiert wurde.

Ein Beispiel:

Die Abbildung beginnt im 24x32 Zeichen Modus standardmäßig bei &h1800. Der zugehörige Wert in VDP Register 2 ist &h06.

Durch die Veränderung von Register 2 auf &h07 wird zwar auf die Abbildung ab Adresse &h1c00 umgeschal-

tet, aber die PRINT Anweisung hat keine Wirkung.

Wenn wir jedoch die Anweisung BASE(5) = &h1C00 und danach die Anweisung SCREEN 1 geben, so wird ebenfalls auf die Abbildung ab &h1C00 umgeschaltet. Fragen wir jetzt mit PRINT VDP(2) das Register ab, so erhalten wir den Wert &h07. Wir sehen also, daß der Registerwert sich mit geändert hat.

Probieren wir jetzt die PRINT-Anweisung aus, so stellen wir fest, daß auch diese Anweisung ordnungsgemäß funktioniert.

SCREEN 0 hat also die Bereiche des Rechners auf die Adressen belegt, die wir vorher mittels der BASE Anweisung angegeben haben.

Das Statusregister des Videoprocessors (VDP Register 8). Es ist unmöglich, in dieses Register einen Wert hineinzuschreiben. Das Statusregister enthält jedoch wichtige Informationen über den Zustand von Sprites.

Das Register ist wie folgt aufgebaut:

Bit	7	6	5	4	3	2	1	0
	F	5. Sprite	C	Nummer 5. Sprite				

Bit 7 ist das Interrupt Flag, welches für unsere Zwecke uninteressant ist.

Bit 6 ist das 5. Sprite Flag, welches an (1) ist, sobald sich mehr als 4 Sprites in einer Zeile befinden, da der Videoprocessor nur 4 Sprites pro Zeile gleichzeitig darstellen kann. Alle weiteren Sprites in der gleichen Zeile sind nicht sichtbar, obwohl sie vorhanden sind.

Bit 5 ist das Flag, welches an (1) ist, wenn sich 2 oder mehrere Sprites berühren.

In den Bits 0 bis 4 steht die Nummer des 5. Sprites, wenn sich mehr als 4 Sprites in einer Punktzeile befinden.

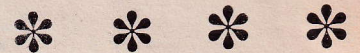
Wir können jetzt mittels der Anweisung: IF (VDP(8) AND 32) = 32 then STOP kontrollieren, ob sich 2 oder mehrere Sprites berühren.

Ist das der Fall, dann hält das Programm an. Im Prinzip ist dies dasselbe, wie die Sprite Interrupt Routine, nur das

hier eine gezielte Spriteberührung kontrolliert werden kann.

Die Anweisung: IF (VDP(8) AND 64) = 64 then PRINT "SPRITE"; (VDP(8) AND 31) gibt die Sritennummer des 5. Sprites auf dem Bildschirm aus, wenn sich mehr als 4 Sprites gleichzeitig in einer Bildschirmzeile befinden. Dadurch ist es möglich, die „Unsichtbarkeit“ von Sprites zu vermeiden.

Allen Lesern, die sich näher mit dem Videoprocessor vertraut machen wollen, sei hier das TMS9918A / TMS9928A / TMS9929A Video Display Processors Data Manual empfohlen; hier sind alle notwendigen Informationen aufgeführt.

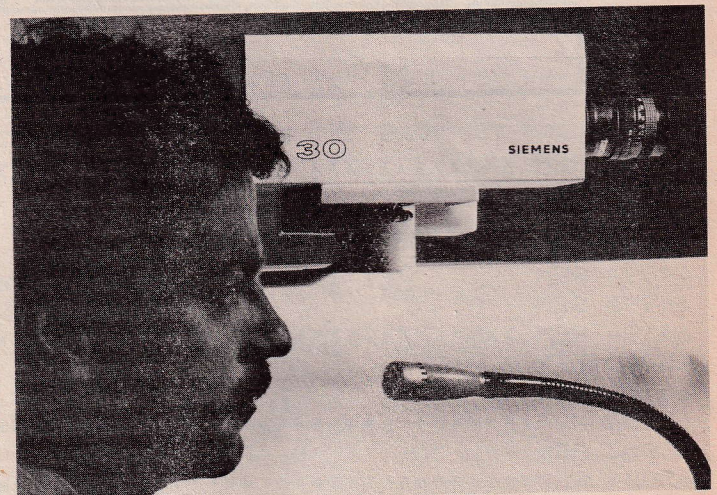


Menschliche Sprache steuert Fernsehkamera

Eine sprachgesteuerte Kamera führt Siemens erstmals auf der Hannover-Messe '85 vor.

Die verschiedenen Steuerfunktionen für Brennweite, Schärfe, Neigung, Schwenken, Ein- und Ausschalten usw. werden dabei nur durch die menschliche Sprache, d. h. durch ein vorgesprochenes Wort für einen bestimmten Befehl ausgeführt. Das jeweilige Wort des gewünschten Befehls für die auszuführende Steuerfunktion wird direkt über ein Mikrofon ohne sonstiges manuelles Bedienen in das sprachgesteuerte System gesprochen.

Das Spracheingabesystem, mit dem die Kamera gesteuert wird, ist lernfähig. Eingebene Befehle merkt sich das System. Zugleich werden auch die Sprechweise und charakteristischen Merkmale des Sprechers vermerkt und abgespeichert. Die einzelnen Befehle einer Befehlskette wie ein, aus, nah, rechts, scharf, stop usw. werden in gleicher Reihenfolge dreimal hintereinander in das System gesprochen und abgespeichert. Die Ausführung der Befehle wird zuvor softwareseitig definiert, damit die Befehle entsprechend ausgeführt werden können.



Fortsetzung S. 50

CPU 49

Nach dem Einspeichern ist das Gerät betriebsbereit, identifiziert die jeweils aufgerufenen Befehle und führt diese aus. Die Befehle lassen sich jederzeit löschen und auf vorgeschriebene Weise von z. B. einem anderen Sprecher, wieder neu eingeben. Je nach Speicherplatzausbau kann eine bestimmte Anzahl von Wörtern gespeichert werden. Ähnliche Wortverbindungen wie „aus“ und „raus“ müssen vermieden werden.

Das System läßt sich überall dort sinnvoll verwenden, wo die Hände für andere Aufgaben gebraucht werden. Das sprachgesteuerte System kann zur Erleichterung der Arbeit, z. B. auch in Behindertenarbeitsplätze, eingebaut werden.

KEYBOARD TOOLKIT (V.1.0) für den CPC 464

Mit diesem Programm ist es möglich, auf dem Keyboard des Schneider CPC 464 22 Befehle auf Tastendruck sowie annähernd den kompletten Zeichensatz des Schneider-Computers auf Tastendruck zu erreichen.

Es wird hierbei im Programm zwischen zwei Modi unterschieden.

1. Der Befehlsmodus

Im Befehlsmodus erhalten alle Alphatasten sowie die Tasten 0, 1, 2, 3, 4, 5, 6 u. 7

eine Option zur Aufnahme von Befehlswörtern.

Einzige Ausnahme bilden hierbei die Tasten X und V. Auf diesen Tasten bleibt die von Haus aus vorgesehene CTRL-Funktion (Invers, Transparent) erhalten. Alle anderen Zeichentasten bekommen ein Grafikzeichen zugeteilt.

2. Der Grafikmodus

Im Grafikmodus wird allen Zeichentasten des Keyboards eines der Grafikzeichen des Zeichensatzes zugeteilt. Die

Sonderfunktion der CTRL-x- und CTRL-V-Tasten bleibt hiervon unberührt.

Besonderheiten: Die Belegung des Befehlscodes 128 darf in keinem Fall geändert werden, da sie in beiden Modi den Aufruf zur Umschaltung in den jeweils anderen Code enthält.

Inbetriebnahme: Nach dem Laden wird mit RUN gestartet. Der Rechner meldet sich dann mit READY. Damit ist der Rechner im Befehlsmodus. Zur Umschaltung zwischen den Modi wird CTRL-0 (Zehnerblock) und danach ENTER gedrückt. Der Rechner meldet sich wiederum mit READY.

Befindet man sich beim Programmieren in der automatischen Zeilenummerierung, so muß vor dem Ändern der Modi die AUTO-Funktion ausgeschaltet werden.

Wenn es stört, daß nach einer Modusänderung jedesmal der Bildschirm gelöscht wird, bzw. wenn der automatisch eingestellte 80-Zeichen-Bildschirm stört, der kann die entsprechende Programmzeile einfach auslassen.

Wenn Sie Ihr Programm zu Ende geschrieben haben, dann löschen Sie einfach den Toolkit mit DELETE 64990-.



```

64990 REM XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
64991 REM
64992 REM      K E Y B O A R D T O O L K I T      (V.1.0)
64993 REM
64994 REM      KEY 1985, Andreas Barufe, Salinenstr.30, 6350 Bad Nauheim
64995 REM
64996 REM XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
65000 REM
65001 REM
65002 REM*****
65003 REM      B E F E H L S B E L E G U N G
65004 REM*****
65005 REM Belegungsdaten des Tastenfeldes und des Zehnerblocks mit ASCII-Codes
        im      B e f e h l s m o d u s
    
```

```

65006 REM*****
65007 DATA 64,1,33,49,160
65008 DATA 65,1,34,50,161
65009 DATA 57,1,35,51,126
65010 DATA 56,1,36,52,162
65011 DATA 49,1,37,53,163
65012 DATA 48,1,38,54,164
65013 DATA 41,1,39,55,165
65014 DATA 40,1,40,56,166
65015 DATA 33,1,41,57,167
65016 DATA 32,1,95,48,168
65017 DATA 25,1,61,45,169
65018 DATA 24,1,163,94,170
65019 DATA 68,1,171,172,173
65020 DATA 67,1,81,113,138
65021 DATA 59,1,87,119,139
65022 DATA 58,1,69,101,140
65023 DATA 50,1,82,114,141
65024 DATA 51,1,84,116,142
65025 DATA 43,1,89,121,143
65026 DATA 42,1,85,117,144
65027 DATA 35,1,73,105,145
65028 DATA 34,1,79,111,146
65029 DATA 27,1,80,112,147
65030 DATA 26,1,124,64,174
65031 DATA 17,1,123,91,175
65032 DATA 69,1,65,97,148
65033 DATA 60,1,83,115,149
65034 DATA 61,1,68,100,150
65035 DATA 53,1,70,102,151
65036 DATA 52,1,71,103,152
65037 DATA 44,1,72,104,153
65038 DATA 45,1,74,106,154
65039 DATA 37,1,75,107,155
65040 DATA 36,1,76,108,156
65041 DATA 29,1,42,58,176
65042 DATA 28,1,43,59,177
65043 DATA 19,1,125,93,178
65044 DATA 71,1,90,122,157
65045 DATA 63,1,88,120,24
65046 DATA 62,1,67,99,158
65047 DATA 55,1,86,118,22
65048 DATA 54,1,66,98,159
65049 DATA 46,1,78,110,136
65050 DATA 38,1,77,109,137
65051 DATA 39,1,60,44,179
65052 DATA 31,1,62,46,180
65053 DATA 30,1,63,47,181
65054 DATA 22,1,96,92,182
65055 DATA 10,1,192,55,135
65056 DATA 11,1,193,56,227
65057 DATA 3,1,194,57,228
65058 DATA 20,1,189,52,132
65059 DATA 12,1,190,53,133
65060 DATA 4,1,191,54,134
65061 DATA 13,1,186,49,129
65062 DATA 14,1,187,50,130
65063 DATA 5,1,188,51,131
65064 DATA 15,1,183,48,128
65065 DATA 7,1,184,46,185
65066 DATA 6,0,13,13,13
65067 :
65068 :
65069 INK 0,0:INK 1,13:BORDER 3:MODE 2:C
LS
65070 REM
65071 REM belegung der befehle

```

```

65072 REM
65073 RESTORE 65007
65074 FOR a=1 TO 60:READ keynr:READ wied
erholung:READ shift:READ normal:READ ctr
l
65075 KEY DEF keynr,wiederholung,normal,
shift,ctrl
65076 NEXT a
65077 REM
65078 REM belegung ascii-code
65079 REM
65080 REM 1.Zahl=Befehlscode 2,"xxx"=Be
fehl (Gesamt max. 100 Buchstaben)
65081 REM Befehl 128 darf nicht geaender
t werden.
65082 DATA 32
65083 DATA 128,"goto 65206"
65084 DATA 129,"list"
65085 DATA 130,"edit "
65086 DATA 131,"auto"
65087 DATA 132,"4"
65088 DATA 133,"5"
65089 DATA 134,"6"
65090 DATA 135,"7"
65091 DATA 136,"next"
65092 DATA 137,"mode"
65093 DATA 138,"q"
65094 DATA 139,"window"
65095 DATA 140,"else"
65096 DATA 141,"read"
65097 DATA 142,"then"
65098 DATA 143,"if"
65099 DATA 144,"using"
65100 DATA 145,"input"
65101 DATA 146,"paper"
65102 DATA 147,"pen"
65103 DATA 148,"a"
65104 DATA 149,"string$("
65105 DATA 150,"dim"
65106 DATA 151,"for"
65107 DATA 152,"goto"
65108 DATA 153,"h"
65109 DATA 154,"ink"
65110 DATA 155,"k"
65111 DATA 156,"locate"
65112 DATA 157,"return"
65113 DATA 158,"cls"
65114 DATA 159,"b"
65115 RESTORE 65082:READ a
65116 FOR b=1 TO a:READ keynr:READ keybe
z$:KEY keynr,keybez$
65117 IF keybez$="q" THEN KEY keynr,CHR$
(208)
65118 IF keybez$="a" THEN KEY keynr,CHR$
(211)
65119 IF keybez$="h" THEN KEY keynr,CHR$
(212)
65120 IF keybez$="k" THEN KEY keynr,CHR$
(213)
65121 IF keybez$="b" THEN KEY keynr,CHR$
(221)
65122 IF keybez$="4" THEN KEY keynr,CHR$
(223)
65123 IF keybez$="6" THEN KEY keynr,CHR$
(238)
65124 IF keybez$="7" THEN KEY keynr,CHR$
(226)
65125 IF keybez$="5" THEN KEY keynr,CHR$
(229)

```

```

65126 NEXT
65127 CLS:END
65128 :
65129 :
65130 REM
65131 REM grafikbelegung
65132 REM
65133 REM belegunsdaten
65134 REM*****
*****
65135 DATA 64,1,33,49,195
65136 DATA 65,1,34,50,196
65137 DATA 57,1,35,51,197
65138 DATA 56,1,36,52,198
65139 DATA 49,1,37,53,199
65140 DATA 48,1,38,54,200
65141 DATA 41,1,39,55,201
65142 DATA 40,1,40,56,202
65143 DATA 33,1,41,57,203
65144 DATA 32,1,95,48,204
65145 DATA 25,1,61,45,205
65146 DATA 24,1,163,94,206
65147 DATA 68,1,171,172,207
65148 DATA 67,1,81,113,138
65149 DATA 59,1,87,119,139
65150 DATA 58,1,69,101,140
65151 DATA 50,1,82,114,141
65152 DATA 51,1,84,116,142
65153 DATA 43,1,89,121,143
65154 DATA 42,1,85,117,144
65155 DATA 35,1,73,105,145
65156 DATA 34,1,79,111,146
65157 DATA 27,1,80,112,147
65158 DATA 26,1,124,64,209
65159 DATA 17,1,123,91,210
65160 DATA 69,1,65,97,148
65161 DATA 60,1,83,115,149
65162 DATA 61,1,68,100,150
65163 DATA 53,1,70,102,151
65164 DATA 52,1,71,103,152
65165 DATA 44,1,72,104,153
65166 DATA 45,1,74,106,154
65167 DATA 37,1,75,107,155
65168 DATA 36,1,76,108,156
65169 DATA 29,1,42,58,214
65170 DATA 28,1,43,59,215
65171 DATA 19,1,125,93,216
65172 DATA 71,1,90,122,157
65173 DATA 63,1,88,120,24
65174 DATA 62,1,67,99,158

65175 DATA 55,1,86,118,22
65176 DATA 54,1,66,98,159
65177 DATA 46,1,78,110,136
65178 DATA 38,1,77,109,137
65179 DATA 39,1,60,44,217
65180 DATA 31,1,62,46,218
65181 DATA 30,1,63,47,219
65182 DATA 22,1,96,92,220
65183 DATA 10,1,192,55,135
65184 DATA 11,1,193,56,233
65185 DATA 3,1,194,57,236
65186 DATA 20,1,189,52,132
65187 DATA 12,1,190,53,133
65188 DATA 4,1,191,54,134
65189 DATA 13,1,186,49,129
65190 DATA 14,1,187,50,130
65191 DATA 5,1,188,51,131
65192 DATA 15,1,183,48,128
65193 DATA 7,1,184,46,222
65194 DATA 6,0,13,13,13
65195 :
65196 :
65197 :
65198 :
65199 :
65200 :
65201 :
65202 :
65203 REM
65204 REM ascii-code grafikastatur
65205 REM
65206 RESTORE 65135
65207 FOR a=1 TO 60:READ keynr:READ wied
erholung:READ shift:READ normal:READ ctr
l
65208 KEY DEF keynr,wiederholung,normal,
shift,ctrl
65209 NEXT a
65210 REM
65211 REM ascii-codes 128 bis 159
65212 REM*****
*****
65213 REM
65214 KEY 128,"goto 65071"
65215 :FOR a=129 TO 159:KEY a,CHR$(a):NE
XT
65216 CLS:END

```



Tips und Tricks für den Schneider CPC 464

Turbotape

Viele Anwender wissen nicht, daß sich die Geschwindigkeit des Kassettenrecorders noch um fast das doppelte steigern läßt. Das unten abgedruckte Demo erlaubt die direkte

Eingabe einer Baudrate, welche zwischen 1400 und 3600 Baud liegen kann.

Es ist ratsam, bei diesen höheren Schreibgeschwindig-

keiten hochwertigeres Bandmaterial zu benutzen, da es sonst zu Ladefehlern kommen kann.

Ist die Baudrate einmal eingestellt, so kann das Pro-

gramm gelöscht werden. Da der CPC beim Einladen die Baudrate feststellt, können auch andersartig abgespeicherte Programme ohne Turbotape eingeladen werden.

Listing Turbotape

```

10 REM *** TURBOTAPE 464 ***
20 REM
30 MEMORY HIMEM-10:SPEICHER =HIMEM
40 REM
50 CLS:MODE 1:RESTORE
60 PRINT"*** TURBOTAPE ***":PRINT:PRINT
70 PRINT"BAUDRATE MIN.1400 MAX.3600":PRINT
80 FOR ZAEHLER = 1 TO 9
90 READ CODE:POKE ZAEHLER+SPEICHER, CODE
100 NEXT ZAEHLER
110 REM
120 INPUT "BAUDRATE ";BAUD
130 HL=INT((333333/BAUD)+0.5)
140 IF HL<1 OR HL>255 THEN 50
150 POKE SPEICHER+2,HL:REM SPEED
160 CALL SPEICHER+1
170 CLS:PRINT BAUD"BAUD SIND EINGESTELLT !"
180 REM
190 REM ** MASCHINENCODE **
200 DATA 33,93,0,62,10,205,104,188,201

```

Wenn Sie in der Lage sind, auch andere an Ihren Ideen teilhaben zu lassen, dann setzen Sie sich mit dem TRONIC-Verlag in Verbindung!

Wir suchen ständig neue Beiträge, z. B.:

- **Tips und Tricks für die verschiedenen Computersysteme**
- **Kleine Maschinen-Routinen (Assembler-Listing)**
- **BASIC-Unterprogramme**
- **Hardware-Bauanleitungen**
- **und andere Beiträge**

Wir brauchen von Ihnen:

- 1. Ausführliche Beschreibung des eingesandten Beitrages**
- 2. Datenträger (Kassette oder Diskette)**
- 3. Komplettes Listing (wenn zur Beschreibung erforderlich)**

All dies senden Sie bitte an den TRONIC-Verlag, Postfach, 3444 Wehretal 1

Wir zahlen: 120,- DM pro abgedruckte Seite.

„DISK CHECK“ - FÜR APPLE II

Das Programm DISK CHECK überprüft Disketten auf Lesefehler. Es besteht aus zwei Teilen, dem Basic-Programm (Listing 1) und dem Maschinen-Programm (Listing 2).

Die eigentliche Lese-Routine liegt im Maschinen-Programmteil. Die erforderlichen Ein- und Ausgaben werden vom Applesoft-Programmteil erledigt.

Wenn man das Basic-Programm startet, wird zunächst die Maschinen-Routine von der Diskette gelesen.

Nachdem der Ladevorgang abgeschlossen ist, erscheint auf dem gelöschten Bildschirm eine Tabelle der Tracks von 0-34.

Dann kommen verschiedene Eingaben bezüglich der zu testenden Tracks.

Nachdem die Eingaben erfolgt sind, wird der Benutzer aufgefordert, seine Diskette in das spezifische Laufwerk einzulegen und <RETURN> zu drücken.

In Zeile 270 erfolgt dann die Übergabe der Werte an das Maschinen-Programm. Dann

verzweigt das Basic-Programm zum Maschinen-Programm.

Wie schon oben erwähnt, untersucht das Binär-Programm die Diskette auf Lesefehler. Dabei werden sämtliche Sektoren der vorher gewählten Tracks gelesen.

Nach jedem gelesenen Track erscheint auf dem Bildschirm unter der jeweiligen Track-Nummer die Anzahl der Lesefehler auf diesem Track. Dies geschieht in Hexadezimal-Form, für 15 unlesbare Sektoren erscheint dann also ein „F“ unter der Track-Nr. Falls der ganze Track (16 Sektoren) nicht lesbar sein sollte, erscheint ein „X“ in inverser Darstellung.

Nachdem sämtliche Tracks gelesen worden sind, verzweigt das Maschinen-Programm zurück zum Haupt-Programm.

Das Hauptprogramm bringt nun ein Menue auf den Bildschirm.

Durch das Menue hat man 3 Wahlmöglichkeiten:

<1> Ausgabe einer genauen Fehlerliste (Track/Sektor-Liste)

<2> Neuer Start (Untersuchen einer weiteren Diskette)

<3> Beenden des Programms

Wählt man die erste Möglichkeit, wird der Bildschirm gelöscht und man muß den Output-Slot (Ausgabe-Slot) eingeben.

Wenn man „0“ eingibt, wird die Fehlerliste auf dem Bildschirm ausgegeben, wählt man einen anderen Slot, so kann sie auch auf einem Drucker ausgegeben werden.

Die Liste hat folgende Form:

<1> Track-Nummer

<2> Nummer des unlesbaren Sektors

Nachdem die Liste vollständig ausgegeben wurde, springt

das Programm nach Eingabe von RETURN zurück zum Menue (s.o.).

Wenn man im Menue die zweite Möglichkeit wählt, wird das Programm wieder gestartet.

Falls man das Programm verlassen will, wählt man vom Menue aus die dritte Möglichkeit.

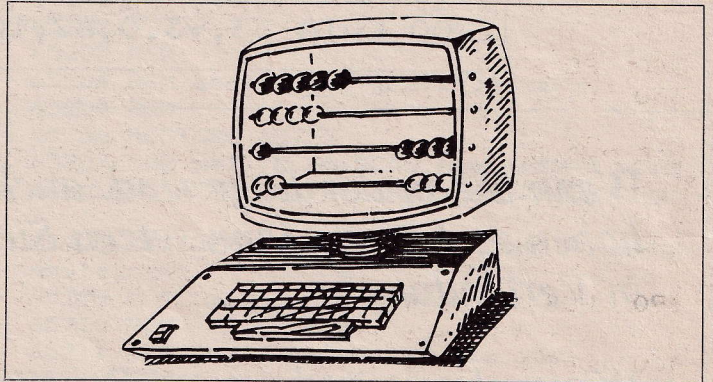
Tips zur Eingabe des Programms:

<1> CALL -151

<2> 8000: A9 3F 8D 47 80 A9 ...

<3> BSAVE DISK CHECK.OBJ,A\$8000,L\$8F

oder
BSAVE DISK CHECK.OBJ,A16384,L143



```

10 REM *****
20 REM ***      DISK CHECK JL      ***
30 REM ***      COPYRIGHT BY      ***
40 REM ***      OLIVER STEINMEIER  ***
50 REM ***      MAUERSTR. 1        ***
60 REM ***      4934 HORN-BAD MBG.1 ***
70 REM ***
80 REM *****
90 PRINT CHR$(4);"BLOAD DISK CHECK.OBJ"
95 REM AUSGABE DER TRACK-LISTE
100 TEXT : NORMAL : HOME
110 VTAB 3: PRINT "DISK CHECK JL (C) 1983 BY O. STEINMEIER"
120 VTAB 7: PRINT "D -----"
130 PRINT "E 0000000000111111111222222222233333"
140 PRINT "C 01234567890123456789012345678901234"
150 PRINT "-" -----"
160 PRINT "H 0000000000000000011111111111111222"
170 PRINT "E 0123456789ABCDEF0123456789ABCDEF012"
180 PRINT "X -----"
190 POKE 34,13
195 REM EINGABE

```

```

200 VTAB 19: INVERSE : INPUT "FIRST TRACK (00 - 34 <DEFAULT 00>) ? ";T$:
:F = INT ( ABS ( VAL ( T$ ))): IF F > 34 THEN 200
210 VTAB 21: INPUT "LAST TRACK (00 - 34 <DEFAULT 34>) ? ";T$:L = ABS
( VAL ( T$ )): IF L > 34 OR L < F THEN 210
220 IF LEN ( T$ ) = 0 OR VAL ( LEF\$( T$,1)) = 0 THEN L = 34
240 VTAB 23: INPUT "DRIVE (01 - 02 <DEFAULT 01>) ? ";T$:D = ABS
( INT ( VAL ( T$ ))): IF D > 2 THEN 240
250 IF D = 0 THEN D = 1
260 NORMAL : HOME : VTAB 20: PRINT " INSERT DISK IN DRIVE ";D;" -->
<CR>": WAIT - 16384,128: HOME : HTAB F + 4
265 REM UEBERGABE DER WERTE
266 REM LESEN DER TRACKS
270 POKE 32803,F: POK32871,L: POKE 32779,D: CALL 32768:ERR = PEEK (76
9) * 256 + PEEK (768)
275 REM MENUE
280 HTAB 26: VTAB 18: FLASH : PRINT ERR;" DISK ERROR";
290 IF ERR < > 1 THEN PRINT "S"
300 NORMAL : HTAB 1: VTAB 18: PRINT "1 PRINT BAD SECTORS": PRINT : PRINT
"2 RESTART": PRINT : PRINT "3 EXIT": PRINT : GET T$:X = ABS ( INT
O( VAL ( T$ ))): IF X > 3 OR X = 0 THEN 300
310 IF X = 2 THEN HOME : GOTO 190
320 IF X = 3 THEN 420
325 REM FEHLER-AUSGABE
330 IF E~R = 0 THEN HOME : HTAB 15: VTAB 19: INVERSE : PRINT "NO ERRORS
!": WAIT - 16384,128: HOME : GOTO 300
340 POKE 34,16: HOME
350 VTAB 19: INVERSE : INPUT "OUTPUT-SLOT (0-7 <~EFAULT 0> ) ? ";S$:S
= ABS ( INT ( VAL ( S$ ))): IF S > 7 THEN 350
360 HOME :: HTAB 17: VTAB 19: PRINT "PRESS RETURN !": WAIT - 16384,128:
NORMAL
370 PRINT CHR$( 4);"PR#";S
380 HOME : PRINT : PRINT "DISK ERRORS: ";ERR: PRINT : FOR I = 1 TO 750: NEXT
390 FOR I = 1 TO ERR: PRINT "TRACK "; PEEK ( I + 16383),"SECTOR "; PEEK (
I + 16944): NEXT
410 PRINT CHR$( 4);"PR#0"
410 PRINT : PRINT : INVERSE : PRINT "PRESS RETURN !": POKE - 16368,0: WAIT
- 16384,128: HOME : GOTO 300
415 REM ENDE
420 HOME : INVERSE : HTAB 18: VTAB 19: PRINT "EXIT ? ": WAIT - 16384,12
8: GET T$: IF T$ = "N" THEN HOME : GOTO 300
430 NORMAL : TEXT : HOME : NEW

```

1

*8000.80BF

```

8000- A9 3F 8D 47 80 A9 42 8D
8008- 4D 80 A9 01 8D EA B7 8D
8010- F4 B7 A9 00 8D EB B7 8D
8018- 00 03 01 03 85 FF 8D
8020- ED B7 A9 00 8D EC B7 20
8028- E3 03 20 D9 03 90 1F E6
8030- FF EE 00 03 D0 09 EE 01

```

```

8038- 03 EE 47 80 EE 4D 80 AE
8040- 00 03 AD EC B7 9D FF 3F
8048- AD ED B7 9D 30 42 EE ED
8050- B7 A9 10 CD ED B7 D0 CF
8058- A9 00 8D ED B7 A5 FF C9
8060- 10 F0 1E 20 E3 FD A9 22
8068- CD EC B7 F0 0A EE EC B7
8070- A9 00 85 FF 4C 27 80 20
8078- E4 FB 20 E4 FB 20 E4 FB
8080- 60 20 80 FE A9 D8 20 ED
8088- FD 20 84 FE 4C 66 80 FF

```

*

Test-Bericht: Logo-Programm für den 48K-Byte Spectrum

Sinclair hat für den 48K-Byte Spectrum jetzt Logo auf den Markt gebracht. Da man überall von Logo spricht, sollte man natürlich wissen, was es damit auf sich hat.

Um die Qualität eines solchen Interpreters beurteilen zu können, muß man erst einmal in Erfahrung bringen, was sich die „Erfinder“ von Logo dabei gedacht haben. Logo ist keine kommerzielle Sprache, sondern für den Bildungsbereich konzipiert. Es sollte ein einfach zu bedienendes Programmierwerkzeug werden, so daß vor allem auch Kinder damit zurecht kommen können. Dabei ist Logo aber keineswegs eine Kindersprache geworden. Den Erfindern ist ein äußerst leistungsfähiges Programm gelungen, das dem interessierten Programmierer jeder Wissensstufe noch etwas bieten kann!

Das Konzept von Logo ist einfach und – vielleicht deshalb – ungemein wirkungsvoll: Programmiert wird nicht mehr mit Zeilennummern (BASIC) oder formal sehr strengen Befehlsfolgen (Pascal), sondern einfach mit Procedures. Diese sind im Grunde eigenständige Programme: ein Assembler-Programmierer könnte sie mit den bei guten Assemblern möglichen Macros (zusammengesetzten Befehlen) vergleichen.

Dieses Konzept fördert zwei Eigenschaften beim (angehenden) Programmierer, die das übliche BASIC zwar nicht verbietet, aber auch nicht erzwingt: Das Denken in bezug auf die Funktion des Programms und das Top-Down-Prinzip (von Grob und Fein) beim Programmieren.

Wieviel davon kann das Sinclair-Logo? Insbesondere von Commodore ist ja bekannt, daß es zwar eine Programmiersprache gibt, diese aber nur sehr eingeschränkt implementiert (angepaßt) werden konnte.

Bei meinen ersten Logo-Gehversuchen fiel mir sofort auf, daß jeder, der nur etwas Englisch kann, keinerlei Probleme mit der Lesbarkeit der Programme haben wird. Sie lesen sich fast wie Umgangssprache (was sonst nur COBOL hat, aber das ist eine kaufmännische Sprache mit vielen formalen Bedingungen). Aber auch ohne Englisch kommt man problemlos zurecht. Dafür sorgen schon die zwei ausgezeichneten Handbücher: einfach eintippen und warten, was dann passiert.

Wer kein Englisch beherrscht, kann sich auch einfach die Procedures auf Deutsch übersetzen: Er definiert einfach einen neuen Befehl mit dem Namen DRUCKE, der dann seinerseits den Befehl PRINT aufruft. Dies hat seine Grenzen im Speicherbereich. Überraschend war für mich jedoch die Erfahrung, daß auch kleinere Kinder ohne Englisch und Zusatzinformationen sehr schnell zurechtkamen. Probieren hilft hier eigentlich immer.

Der Befehlsvorrat ist sehr groß. Viele sehr häufig gebrauchte Befehle (PRINT z. B.) können abgekürzt werden.

Der Editor ist bildschirmorientiert und mit einer guten Cursorsteuerung ausgestattet. Dazu gibt es – wie beim Sinclair gewohnt – einen Ton für

eine akzeptierte Taste, der sehr unterschiedlich hoch ist für Groß- und Kleinbuchstaben. Befehle können sowohl in Klein- als auch in Großbuchstaben eingegeben werden. Sie werden automatisch in Großbuchstaben umgewandelt.

Wie ein Logo-Programm aussehen kann, zeigen die kurzen Listings. Jedes ist als Prozedur zu verstehen, die ihrerseits wieder Procedures aufruft. Eine Prozedur kann sich auch selbst aufrufen (rekursiv), bis eine bestimmte Abbruchbedingung erfüllt ist. Die hier gezeigten Programme benutzen nur lokale Variable. Es ist aber auch möglich, globale Variable zu definieren. Procedures können auch so angelegt werden, daß eine echte Parameterübergabe erfolgt.

Das Programm kann natürlich auch Drucker ansprechen. Diese sollten aber entweder den Spectrum-Schriftsatz (=ASCII-Standard) benutzen oder aber umschaltbar sein auf den Standard-Schriftsatz. Logo benötigt sehr oft die [und], die im Listing sonst als Ä und Ü erscheinen würden. Der Einsatz der Microdrives ist vorgesehen, ebenso der Druck über die serielle Schnittstelle des Interface 1. Bei längeren Programmen weiß der Programmierer einen Drucker immer zu schätzen.

Aber Logo kann noch mehr: von Sinclair eingeplant ist auch der Einsatz einer „echten“ Schildkröte, also ein Roboter. Auch diese Befehle sind bereits eingebaut. Gerade diese Option ist für Kinder besonders reizvoll, obwohl – ich habe auch meinen Spaß

daran, wenn so ein Roboter einfach malt, wenn ich nur sage „FORWARD 50“ (50 Einheiten vorwärts).

Neben den unbestreitbaren Vorteilen, die Logo für den absoluten Anfänger bietet (s. Grafik „Spirale“, man programmiere das in BASIC!), kann Logo auch für andere Problemlösungen eingesetzt werden. Eine Dateiverwaltung ist ebenso denkbar wie ein Mathematik-Programm für Schüler oder schlicht ein Spiel. Es stehen Tonbefehle und mathematische Funktionen zur Verfügung, dazu ausgesprochen einfache Grafik-Befehle (Winkelangaben!) und einiges mehr.

Die Handbücher tun das ihre: der 1. Band erklärt „nur“ einen Teil der Befehle, das aber mit guten Beispielen und einfachen Worten (notfalls hilft hier tatsächlich ein normales Wörterbuch oder Ausprobieren). Der 2. Band zählt alle Befehle auf und gibt dazu die Erklärung. Dies reicht auch, wenn man den ersten Band durchgearbeitet hat. Im übrigen – für Logo gibt es genauso Lehrbücher wie für alle anderen Programmiersprachen auch.

Wer sich wirklich für diese Sprache interessiert, kann davon ausgehen, daß das Sinclair-Logo einem Vergleich mit den Logo-Versionen auf anderen Computern standhält. Es wird der von den Erfindern gesetzten Zielvorstellung vollständig gerecht, was auf einem Heimcomputer nicht selbstverständlich ist. Es kann auf Microdrive abgespeichert werden (Anleitung) und auch Files dort ablegen.

Procedures siehe S. 68

KIRCHNER-Elektronik – Innovative Techniken

In der Automobil-Branche kennt man sie schon seit langem, die „Tuner“, die Serienfahrzeuge um- und aufrüsten, um mehr Leistung aus ihnen herauszuholen.

Auch im Computergeschäft haben sich inzwischen einige Firmen darauf spezialisiert, die elektronischen Geräte auf Höchstleistung zu trimmen. Einer der ersten Computer-Tuner in Deutschland ist der Diplom-Ingenieur Anton Kirchner aus Duisburg-Neu-

dorf. An der Duisburger Universität hat sich Kirchner mit dem „tunen“ eines Epson QX-10 einen Namen gemacht. Er schaffte es, die Speicherkapazität des ohnehin schon hochqualifizierten Systems zu verdoppeln, um damit auch den gehobenen Ansprüchen in Forschung und Verwaltung gerecht zu werden.

Gut drei Viertel der von Kirchner-Elektronik verkauf-

ten Computer sind getunte Serienprodukte oder Spezialanfertigungen nach Kundenwunsch.

Für den transportablen Einsatz konstruierte Kirchner einen Computer mit kleinstmöglichen Abmessungen. Dieser vollwertige CP/M-Computer wiegt nur knapp 2 kg, mißt 70 x 110 x 170 mm und speichert 800 KB auf seiner Diskette (bei einem Preis von ca. 2000,- DM).

Neben der umfangreichen Hardware-Veränderung erstellt Kirchner selbst eine ganze Reihe kleiner, nützlicher Nebenprogramme, die die Routinearbeiten erleichtern helfen.

Alle Programme haben eines gemeinsam: Sie machen Computer zuverlässiger, anwenderfreundlicher und schneller.

Dennoch sind sie im Preisgefüge für jeden Anwender erschwinglich.

Multicolor-Grafik für den TI-99

Nur wenigen TI-Spezialisten ist bekannt, daß der im TI-99 eingebaute Video-Prozessor TMS9918 (TI-99/4) bzw. TMS9918A (TI-99/4A) neben dem normalen Grafikmodus und der hochauflösenden Grafik auch noch den sog. Multicolor-Modus beherrscht. Das folgende Maschinenprogramm für Extended Basic, 32K-Speichererweiterung und Diskettenlaufwerk macht diese Betriebsart nun auch dem Basic-Programmierer mit vielen einfachen Grafikbefehlen zugänglich.

Im Multicolor-Modus bestehen alle Zeichen nur noch aus einer 4★4-Matrix. Somit ist der Bildschirm in 48 Zeilen zu je 64 Spalten aufgeteilt. In jeder dieser 3072 Bildschirmpositionen kann eine der 16 möglichen Farben angezeigt werden – Sie haben also fast unbegrenzte Farbsetzungsmöglichkeiten!

Vorher müssen Sie aber noch den Basic-Loader für das Maschinenprogramm abtippen. Um Ihnen dies möglichst einfach zu machen, berechnet das Programm aus jeder DATA-Zeile die Prüfsumme und zeigt eventuelle Tippfehler sofort an. In diesem Fall

wird die Zeilennummer ausgegeben und mittels einiger POKEs, die den NUMBER-Modus simulieren, wird die fehlerhafte Zeile angezeigt. Haben Sie das Programm abgetippt und auf Diskette gespeichert, dann sollten Sie sich als Lohn für die Anstrengung die Grafikbeispiele ansehen, die in den Basic-Loader integriert sind. Wollen Sie nun selber Grafiken programmieren, so stehen Ihnen folgende Befehle zur Verfügung:

– CALL LINK(„HELP“) gibt einen Katalog der Grafikbefehle aus, die im folgenden erklärt werden. Dazu liest der Computer die Namenstabelle von 16280 bis 16383 und zeigt die Einträge geordnet auf dem Bildschirm an.

– CALL LINK(„MULTI“, schirmfarbe) ist der wichtigste Befehl, denn er schaltet den Video-Chip in den Multicolor-Modus um und färbt den Bildschirm entsprechend ein. Natürlich können Sie auch weiterhin dazu CALL SCREEN verwenden. Beim Aufruf dieses Programms wird auch gleich der Bildschirm initialisiert, d. h. mit den entsprechenden Zeichen beschrieben, und die gesamte

Farbtabelle wird auf „transparent“ gesetzt.

– CALL LINK(„CLEAR“) ersetzt den Befehl CALL CLEAR, der nicht mehr funktioniert, und löscht den Bildschirm.

– CALL LINK(„POINT“, zeile,spalte,farbe) setzt einen Punkt mit der bereits bekannten 4★4-Matrix in der angegebenen Farbe an die gewünschte Bildschirmposition. Hierbei muß der Zeilenwert zwischen 1 und 48, der Spaltenwert zwischen 1 und 64 liegen. Wird ein Farbcode größer als 16 gewählt, so wird der danebenliegende Punkt mit eingefärbt.

– CALL LINK(„HLINE“, zeile,spalte,länge,farbe) zieht eine horizontale Linie, beginnend an der gegebenen Zeile/Spalte-Position mit der angegebenen Länge in der entsprechenden Farbe.

– CALL LINK(„VLINE“, zeile,spalte,länge,farbe) macht genau dasselbe – nur eben vertikal.

– CALL LINK(„LDIAG“, zeile,spalte,länge,farbe) zieht eine Diagonale, die bildlich gesehen von links oben nach rechts unten verläuft.

– CALL LINK(„RDIAG“, zeile,spalte,länge,farbe) zieht

ebenfalls eine Diagonale, nur beginnt diese rechts oben und geht nach links unten.

– CALL LINK(„BLOCK“, zeile,spalte,rechts,runter,farbe) zeichnet einen Block, dessen linke obere Ecke durch Zeile und Spalte bestimmt ist und der die angegebenen Einheiten nach rechts und nach unten geht.

– CALL LINK(„RECT“, zeile,spalte,rechts,runter,farbe) zeichnet ein Rechteck. Der Befehl ist genauso aufgebaut wie BLOCK.

– CALL LINK(„INVERT“) invertiert alle Farben des Multicolor-Bildschirms bis auf die SCREEN-Farbe; so wird z. B. aus schwarz weiß und umgekehrt.

– CALL LINK(„QUIT“, schirmfarbe) verläßt den Multicolor-Modus wieder und schaltet in den normalen Grafikmodus zurück. Die Bildschirmfarbe wird entsprechend der Angabe geändert.

– CALL LINK(„RECALL“, schirmfarbe) ruft den zuletzt bearbeiteten Multicolor-Bildschirm zurück, sofern dieser mit CALL LINK(„QUIT“) verlassen wurde. Dabei wird auch das SCREEN entsprechend eingefärbt.

```

100 ! *****
110 ! *****
120 ! * *
130 ! * MULTICOLOR *
140 ! * *
150 ! *****
160 ! *****
170 !
180 ! (C) Martin Kotulla 84
190 ! -----
200 !
210 CALL CHAR(42,"AA55AA55AA55AA55",91,"7E8199A1A199817E")
220 CALL CHAR(45,RPT$("0",8)&"FF")
230 CALL CHAR(128,"DFBF60C0919B959191808080C060BFD")
240 CALL CHAR(130,"FBFD060301010149516151490306FDF")
250 CALL CLEAR :: CALL HCHAR(2,4,42,26)
260 CALL VCHAR(3,4,42,5):: CALL VCHAR(3,29,42,5)
270 CALL MAGNIFY(3):: CALL SPRITE(#1,128,2,28,45)
280 DISPLAY AT(5,8)SIZE(17):"MULTICOLOR-GRAFIK"
290 CALL HCHAR(8,4,42,26)
300 DISPLAY AT(12,4):CHR$(91)&" MARTIN KOTULLA 1984"
310 CALL HCHAR(13,6,45,22):: DISPLAY AT(17,7):"DATAS PRUEFEN? J"
320 ACCEPT AT(17,22)BEEP SIZE(-1)VALIDATE("JN"):JN$
330 IF JN$="N" THEN 530
340 FOR I=1 TO 64 :: READ A :: X=X+A :: NEXT I
350 DISPLAY AT(20,1):"PRUEFSUMMEN-DATA:"
360 IF X(>)72304 THEN DISPLAY AT(20,19):"FALSCH!" :: END
370 DISPLAY AT(20,19):"KORREKT!"
380 DIM CHECK(64):: RESTORE 670
390 DISPLAY AT(22,1):"PROGRAMM-DATA:"
400 FOR I=1 TO 64 :: READ CHECK(I):: NEXT I
410 FOR I=1 TO 64 :: SUMME=0
420 ZEILE=I*10+750 :: IF ZEILE>1320 THEN ZEILE=ZEILE+30
430 FOR J=1 TO 16 :: READ A :: SUMME=SUMME+A :: NEXT J
440 IF SUMME=CHECK(I)THEN 510
450 CALL SOUND(150,220,0):: DISPLAY AT(22,15):"FEHLER IN";ZEILE
460 Z1=INT(ZEILE/256) | 480 CALL LOAD(-31986,0,10)
470 Z2=ZEILE-Z1*256 | 490 CALL LOAD(-31980,Z1,Z2)
500 CALL LOAD(-31931,1):: END | 520 NEXT I
510 DISPLAY AT(22,15)BEEP;ZEILE;"KORREKT!" | 530 CALL INIT :: RESTORE 760
531 DISPLAY AT(24,1):"ALLES RICHTIG! BITTE WARTEN!"
540 FOR I=9460 TO 10370 STEP 4
550 READ A,B,C,D :: CALL LOAD(I,A,B,C,D):: NEXT I
560 FOR I=16280 TO 16383 STEP 4 :: READ A,B,C,D :: CALL LOAD(I,A,B,C,D)
570 NEXT I :: CALL LOAD(8194,40,130,63,152)
580 CALL HCHAR(17,1,32,256)
590 DISPLAY AT(17,1):"WOLLEN SIE DIE GRAFIK-DEMOS": "SEHEN? J"
600 ACCEPT AT(19,8)BEEP SIZE(-1)VALIDATE("JN"):JA$
610 IF JA$="J" THEN 1460
620 CALL DELSPRITE(ALL):: CALL CLEAR
630 STOP | 650 ! PRUEFSUMMEN-DATAS:
640 ! | 660 !
670 DATA 0309,0553,1778,1932,1682,0826,1642,1231,1571,0781,1540,1415
680 DATA 0999,1164,0563,1125,1268,0909,1252,1304,1058,1252,1058,1293
690 DATA 1175,1168,1273,1535,1107,1306,0965,1402,1032,0738,1173,0780
700 DATA 0884,0343,1916,0922,1103,0918,1144,0724,1299,1257,0904,1332
710 DATA 1340,1408,1041,0799,1053,1316,1185,1330,0621,1030,1324,1133
720 DATA 1024,0962,1287,0546
730 !

```

```

740 i PROGRAM-DATA#
750 i
760 DATA 000,000,000,002,000,008,000,248,032,000,004,007,000,002,000,006
770 DATA 003,000,000,000,000,042,128,000,064,000,037,068,000,000,037,174
780 DATA 039,134,035,253,038,184,040,183,041,182,061,221,144,000,044,179
790 DATA 045,194,144,000,004,216,061,221,060,191,061,190,062,192,094,197
800 DATA 000,027,000,000,036,254,038,112,212,000,173,181,172,180,169,128
810 DATA 192,000,000,046,000,002,000,053,000,012,000,009,083,203,002,224
820 DATA 036,254,006,160,039,106,002,224,036,254,006,160,039,124,004,192
830 DATA 004,224,037,068,002,009,000,006,192,046,037,068,032,036,252
840 DATA 037,068,193,001,006,196,192,192,002,035,000,128,192,129,006,194
850 DATA 002,034,000,032,004,032,032,032,006,193,005,129,128,129,022,001
860 DATA 192,068,006,193,005,128,128,192,017,245,006,009,022,229,004,096
870 DATA 039,092,002,224,036,254,006,160,039,106,004,096,039,092,002,224
880 DATA 036,254,002,001,000,001,006,160,040,036,192,032,131,074,002,032
890 DATA 006,255,004,032,032,048,002,000,224,000,216,000,131,212,002,000
900 DATA 001,224,004,032,032,048,002,000,004,000,004,032,032,048,004,096
910 DATA 09,092,002,224,036,254,002,001,000,001,006,160,040,036,200,032
920 DATA 131,074,037,070,006,160,040,036,200,032,131,074,037,074,006,160
930 DATA 040,036,200,032,131,074,037,078,006,032,037,078,004,032,040,052
940 DATA 004,096,039,092,002,224,036,254,006,160,039,172,004,032,040,052
950 DATA 005,160,037,074,006,004,021,250,004,096,039,092,002,224,036,254
960 DATA 006,160,039,172,004,032,040,052,005,160,037,070,006,004,021,250
970 DATA 004,096,039,092,002,224,036,254,006,160,039,172,004,032,040,052
980 DATA 005,160,037,070,005,160,037,074,006,004,021,248,004,096,039,092
990 DATA 002,224,036,254,006,160,039,172,004,032,040,052,005,160,037,070
1000 DATA 006,032,037,074,006,004,021,248,004,096,039,092,002,224,036,254
1010 DATA 006,160,039,222,193,132,004,032,040,052,005,160,037,074,006,006
1020 DATA 021,250,005,160,037,070,200,032,037,076,037,074,006,005,021,242
1030 DATA 004,096,039,092,002,224,036,254,006,160,039,222,193,132,004,032
1040 DATA 040,052,005,160,037,074,006,006,021,250,000,032,037,076,037,074
1050 DATA 193,197,004,032,040,052,005,160,037,070,006,007,021,250,200,032
1060 DATA 037,076,037,074,193,132,004,032,040,052,005,160,037,074,006,006
1070 DATA 021,250,200,032,037,072,037,070,193,197,004,032,040,052,005,160
1080 DATA 037,070,006,007,024,250,016,048,002,224,036,254,002,000,056,000
1090 DATA 004,193,004,032,032,040,005,065,004,032,032,032,005,128,002,128
1100 DATA 062,008,017,247,004,096,039,092,002,224,036,254,004,032,032,024
1110 DATA 000,038,002,007,063,152,002,008,037,062,002,009,000,006,209,183
1120 DATA 002,038,000,222,006,006,009,022,250,002,000,002,226,002,001
1130 DATA 037,062,002,002,000,006,004,032,032,036,004,032,024,000,038
1140 DATA 005,199,002,135,063,255,017,231,216,032,036,244,131,124,002,224
1150 DATA 131,224,004,096,000,112,002,000,056,000,004,193,004,032,032,032
1160 DATA 005,128,002,128,062,008,017,250,194,139,002,001,000,001,006,160
1170 DATA 040,036,192,032,131,074,002,032,006,255,004,032,02,048,002,000
1180 DATA 232,000,216,000,131,212,002,000,001,232,004,032,032,048,002,000
1190 DATA 004,007,004,032,032,048,004,090,194,139,002,001,000,001,006,160
1200 DATA 040,036,200,032,131,074,037,070,006,160,040,036,200,032,131,074
1210 DATA 037,074,006,160,040,036,193,032,131,074,006,160,040,036,200,032
1220 DATA 131,074,037,078,006,032,037,078,004,090,194,139,002,001,000,001
1230 DATA 006,160,040,036,200,032,131,074,037,070,200,032,131,074,037,072
1240 DATA 006,160,040,036,200,032,131,074,037,074,200,032,131,074,037,076
1250 DATA 006,160,040,036,193,032,131,074,006,160,040,036,193,096,131,074
1260 DATA 006,160,040,036,200,032,131,074,037,078,006,032,037,078,004,090
1270 DATA 004,192,004,032,032,012,004,032,032,012,004,032,032,018,184,005,129,004,091
1280 DATA 037,030,040,056,193,160,037,074,008,022,010,054,193,032,037,070
1290 DATA 161,132,002,038,056,000,008,052,057,032,036,250,161,133,192,006
1300 DATA 193,032,037,078,006,196,004,193,004,032,032,040,195,032,037,074
1310 DATA 004,203,062,224,036,246,195,012,019,003,002,065,240,000,016,003
1320 DATA 002,065,015,000,010,068,160,006,196,004,193,004,032,032,003,128,002,000

```

```

1330 i
1340 i NAMENSTABELLE:
1350 i
1360 DATA 072,069,076,080,032,032,039,028,081,085,073,084,032,032,037,178
1370 DATA 082,069,067,065,076,037,090,073,078,086,069,082,084,038,252
1380 DATA 082,069,067,084,032,032,038,168,066,076,079,067,075,032,038,128
1390 DATA 082,068,073,065,071,032,038,100,076,068,073,065,071,032,038,072
1400 DATA 086,076,073,078,069,032,038,048,072,076,073,078,069,032,038,024
1410 DATA 080,079,073,078,084,032,037,230,067,076,069,065,082,032,037,166
1420 DATA 077,085,076,084,073,032,037,082,000,000,000,000,000,000,000,000
1430 i
1440 i MULTICOLOR-DEMO:
1450 i
1460 ON WARNING NEXT :: CALL DELSPRITE(#1)
1470 FOR P=1 TO 11
1480 ON P GOSUB 1520,1580,1630,1670,1720,1770,1830,2000,2050,2100,2140
1490 CALL SOUND(150,440,0)
1500 CALL KEY(0,X1,X2):: IF X2=0 THEN 1500
1510 NEXT P :: CALL CLEAR :: END
1520 i *****
1530 CALL LINK("MULTI",2):: FOR I=0 TO 9.2 STEP .3
1540 CALL LINK("POINT",INT(30*ATN(I))*I*7,14)
1550 CALL LINK("POINT",SGR(ABS(6-1^2))*5,I*7,7)
1560 CALL LINK("POINT",INT((SIN(I)+1)*20)*I*7,13)
1570 NEXT I :: RETURN
1580 i *****
1590 CALL LINK("MULTI",2):: FOR I=PI+PI/40 TO 3*PI STEP PI/40
1600 A=ABS(ATN(COS(I)*TAN(I))*20)+13 :: B=I*7-15
1610 CALL LINK("POINT",A,B,14):: CALL LINK("POINT",48-A,B,13)
1620 NEXT I :: RETURN
1630 i *****
1640 CALL LINK("MULTI",2):: FOR I=1 TO 48
1650 J=J+1 :: IF J>16 THEN J=3
1660 CALL LINK("HLINE",I,2,I,J):: NEXT I :: RETURN
1670 i *****
1680 CALL LINK("MULTI",2):: FOR I=1 TO 48
1690 J=J+1 :: IF J>13 THEN J=3
1700 CALL LINK("HLINE",I,2,I,J):: CALL LINK("HLINE",I,64-I,64-I,16-J)
1710 NEXT I :: RETURN
1720 i *****
1730 CALL LINK("MULTI",2):: FOR I=1 TO 48
1740 J=J+1 :: IF J>13 THEN J=3
1750 CALL LINK("VLINE",I,I,47,J):: CALL LINK("HLINE",I,1,60,16-J)
1760 NEXT I :: RETURN
1770 i *****
1780 CALL LINK("MULTI",2):: FOR L=1 TO 4
1790 FOR I=1 TO 38
1800 J=J+1 :: IF J>16 THEN J=3
1810 CALL LINK("LDIAG",I,38-I,48-I,J):: NEXT I
1820 J=J-2 :: NEXT L :: RETURN
1830 i *****
1840 CALL LINK("MULTI",2):: FOR I=28 TO -30 STEP -1
1850 F=F+1 :: IF F>16 THEN F=3
1860 CALL LINK("RDIAG",18,I,28,F):: NEXT I
1870 FOR I=-27 TO 32
1880 F=F+1 :: IF F>16 THEN F=3
1890 CALL LINK("LDIAG",18,I,28,F):: NEXT I

```

```

1900 FOR I=1 TO 20 :: CALL LINK("HLINE",I,I,40,F)
1910 F=F+1 :: IF F>16 THEN F=3
1920 NEXT I
1930 FOR I=1 TO 30 :: CALL LINK("RDIAG",1,I,20,F)
1940 F=F+1 :: IF F>16 THEN F=3
1950 NEXT I
1960 FOR I=1 TO 30 :: CALL LINK("VLINE",I,I,15,F)
1970 CALL LINK("VLINE",32-I,64-I,15,16-F)
1980 F=F+1 :: IF F>13 THEN F=3
1990 NEXT I :: RETURN
2000 ! *****
2010 CALL LINK("MULTI",2):: FOR I=40 TO 3 STEP -1
2020 F=F+1 :: IF F>16 THEN F=3
2030 CALL LINK("BLOCK",I,I,5,5,F):: CALL LINK("BLOCK",I,56-I,5,5,F)
2040 NEXT I :: RETURN
2050 ! *****
2060 CALL LINK("MULTI",2):: FOR I=6 TO 40 STEP 2
2070 F=F+1 :: IF F>16 THEN F=3
2080 CALL LINK("RECT",5,10,I,I,F)
2090 NEXT I :: RETURN
2100 ! *****
2110 CALL LINK("MULTI",2):: FOR I=6 TO 30
2120 J=J+I :: IF J>16 THEN J=3
2130 CALL LINK("RECT",5,I,I,I,J):: NEXT I :: RETURN
2140 ! *****
2150 F=3 :: S=6 :: CALL LINK("MULTI",2):: FOR I=1 TO 25
2160 CALL LINK("LDIAG",1,S,8,F):: CALL LINK("RDIAG",9,S+6,7,F)
2170 S=S+2 :: F=F+1 :: IF F>16 THEN F=3
2180 NEXT I
2190 F=3 :: S=S-1 :: FOR I=1 TO 25
2200 CALL LINK("RDIAG",16,S,8,F):: CALL LINK("LDIAG",24,S-6,7,F)
2210 S=S-2 :: F=F+1 :: IF F>16 THEN F=3
2220 NEXT I
2230 S=S+1 :: FOR I=1 TO 25
2240 CALL LINK("LDIAG",31,S,8,F):: CALL LINK("RDIAG",39,S+6,7,F)
2250 S=S+2 :: F=F+1 :: IF F>16 THEN F=3
2260 NEXT I
2270 FOR I=1 TO 10 :: CALL LINK("INVERT"):: NEXT I
2280 RETURN
2290 END ! *****

```

Tips und Tricks für den ZX-Spectrum

Spectrum-Programme, die zahlreiche Farbbefehle enthalten, lassen sich beschleunigen, wenn man die Farbwerte durch einen Pokebefehl eingibt. Um dies zu erreichen, gebraucht man die Systemvariable ATTRP. Poked man anschließend eine Zahl zwischen 0-255 ein, so erfol-

gen die weiteren Ausgaben in der gewählten Farbe. Die Adresse von ATTRP lautet 23693.

Errechnet man vor Beginn der Programmierung die Farbwerte, so können die gesamten INK-, PAPER-, BRIGHT- und FLASHBEFEHLE durch jeweils einen

POKE-Befehl ersetzt werden, dies bewirkt nicht nur eine höhere Ablaufgeschwindigkeit, sondern auch eine Speicherplatzersparnis.

Die zu pokenden Werte errechnen sich wie folgt:
Wert für INK + 8 x Wert für PAPER + 64 für BRIGHT 1 + 128 für FLASH 1.

Ein helles, blinkendes Zeichen - gelb auf blauem Grund - besitzt den Wert 206. Also POKE 23693,206. Der Wert 206 ergibt sich aus: $6 + 8 + 64 + 128 = 206$. Ein Zeichen BRIGHT 0: FLASH 0: INK 4: PAPER 7 besitzt den Wert: $4 + (7 \times 8) + 0 + 60$.

Geschicklichkeitsspiel: „Aladin“ – Schneider CPC 464 –

In diesem Spiel soll Aladin die im Verlies versteckte Lampe finden. Leider ist es im Verlies sehr dunkel, so daß er nur wenig sehen kann. Auf seinem Weg trifft Aladin auf folgende Gegenstände:

– Steinblöcke, die sichtbar werden, wenn er dagegen läuft.

– Geister, die immer kurz aufblincken. Ein Dagegenlaufen ist absolut tödlich.

– Kerzen: Wenn er eine solche Kerze trifft, erhellt sich der Raum kurz und man sieht alles.

– Die Lampe: Es gibt 250 Punkte.

Aladin startet am linken Tor. Hat er die Lampe gefunden, geht das rechte Tor auf; Aladin sollte schleunigst dorthin laufen, denn er hat insgesamt nur 30 Sekunden Zeit. Hat er es geschafft, bekommt er 500 Punkte und ein weiteres Spiel.

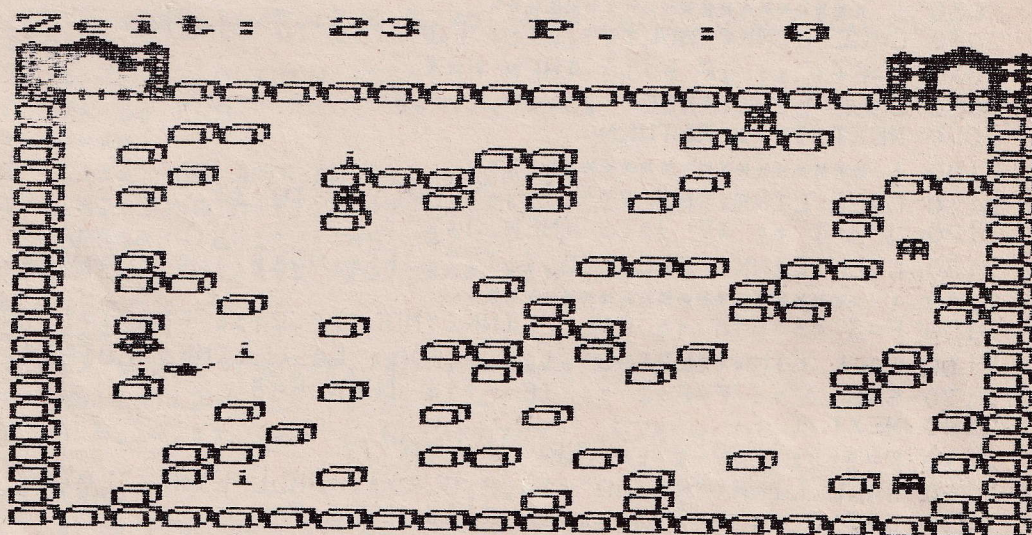
Die Steuerung erfolgt mit den Cursortasten, wobei durch Drücken zweier Tasten diagonal gegangen werden kann, oder mit dem Joystick.

Zum Programm:

Das Programm ist in Vor- und Hauptprogramm gegliedert. Tippen Sie zuerst das Vorprogramm ein, saveen Sie es ab unter „ALADIN“; danach das Hauptprogramm,

das mit „ALADIN2“ abgesavet werden muß. Wollen Sie das Programm starten, starten Sie zuerst das Vorprogramm – das Hauptprogramm wird dann automatisch nachgeladen.

Aladin-Hardcopy



```

10 * *****
20 * Aladin (c) 1985 *
30 * by JeBaSoft *
40 * *****
50 SYMBOL AFTER 128
60 SYMBOL 128,60,126,90,255,255,102,60,24
70 SYMBOL 129,60,126,90,219,255,102,90,126
80 SYMBOL 130,60,126,90,255,219,66,60,24
90 SYMBOL 131,63,67,253,133,133,133,134,252
100 SYMBOL 132,28,62,42,62,62,62,42,42
110 SYMBOL 133,0,0,17,250,188,248,56,0
120 SYMBOL 134,0,0,32,113,255,255,247,163
130 SYMBOL 135,24,60,126,255,255,255,255,231
140 SYMBOL 136,0,0,4,14,255,255,239,197
150 SYMBOL 137,247,255,191,247,247,163,227,227
160 SYMBOL 138,195,129,0,0,0,0,0,0
170 SYMBOL 139,239,255,253,239,239,197,199,199
180 SYMBOL 140,247,255,191,247,247,171,255,235
190 SYMBOL 141,195,165,255,36,36,255,36,36
200 SYMBOL 142,239,255,253,239,239,213,255,215
210 SYMBOL 143,0,0,8,0,8,8,8,28
220 MODE 1
230 BORDER 8
240 INK 0,B
250 PAPER 0:INK 1,8:PEN 1
260 PRINT CHR$(24) "

```

A L A D I N

"CHR\$(24)

```

270 PRINT:PRINT "Im Spiel Aladin musst du eine im Verliesversteckte Dellampe fin
den."
280 PRINT "Leider ist es im Verlies so dunkel, dassdu nichts siehst."
290 PRINT "Nun gibt es aber im Verlies mehrere Ker-zen, die, wenn man sie findet
, kurz das Verlies aufleuchten lassen."
300 PRINT "Aber es gibt im Verlies auch 3 Geister, die absolut toedlich sind..."
310 PRINT "Zudem liegen im Verlies eine Menge Steine im Wege herum, die man
erst bei Beruehrung sieht."
320 PRINT "Wenn man die Dellampe gefunden hat, wirdder Ausgang geoeffnet und man
muss auf dem schnellsten Wege dorthin gelangen."
330 PRINT:PRINT "Dein Aladin wird mit dem JOYSTICK oder der Tastatur geste
uert"
340 INK 1,0
350 INK 2,6,26
360 a#=INKEY#:IF a#="" THEN 360
370 PEN 2:PRINT:PRINT" LADEN DES 2. TEILS ..."
380 RUN "!ALADIN2"

```

Teil 2

```

10 DIM b(20,25)
20 MODE 0:BORDER 0:INK 0,0:INK 1,9:INK 2,0:INK 3,11:INK 4,0:INK 5,0:INK 6,13:INK
7,0
30 zeit=30
40 PAPER 0:CLS
50 PEN 3:PRINT:PRINT CHR$(134)CHR$(135)CHR$(136):LOCATE 18,2:PRINT CHR$(134)CHR$
(135)CHR$(136)
60 LOCATE 1,3:PRINT CHR$(137)CHR$(138)CHR$(139):LOCATE 18,3:PRINT CHR$(137)CHR$(
138)CHR$(139)
70 LOCATE 1,4:PRINT CHR$(140)CHR$(141)CHR$(142);:PEN 6:PRINT STRING$(14,131);:PE
N 3:PRINT CHR$(140)CHR$(141)CHR$(142);
80 FOR i=5 TO 24:LOCATE 1,i:PEN 6:PRINT CHR$(131):LOCATE 20,i:PRINT CHR$(131):NE
XT i
90 LOCATE 1,24:PRINT STRING$(20,131)
100 x=2:y=5:LOCATE x,y:PEN 1:PRINT CHR$(128)
110 b(x,y)=255
120 FOR i= 1 TO 75
130 xr=INT(RND(1)*18)+2:yr=INT(RND(1)*18)+6
140 IF b(xr,yr)<>0 THEN 130
150 b(xr,yr)=1
160 PEN 2:LOCATE xr,yr:PRINT CHR$(131):NEXT
170 FOR i=1 TO 4
180 xr=INT(RND(1)*18)+2:yr=INT(RND(1)*18)+5
190 IF b(xr,yr)<>0 THEN 180
200 b(xr,yr)=4
210 PEN 7:LOCATE xr,yr:PRINT CHR$(143):NEXT
220 FOR i=1 TO 4
230 xr=INT(RND(1)*18)+2:yr=INT(RND(1)*18)+5
240 IF b(xr,yr)<>0 THEN 230
250 b(xr,yr)=3
260 PEN 5:LOCATE xr,yr:PRINT CHR$(132):NEXT
270 xr=INT(RND(1)*18)+2:yr=INT(RND(1)*7)+14
280 IF b(xr,yr)<>0 THEN 270
290 b(xr,yr)=2
300 PEN 4:LOCATE xr,yr:PRINT CHR$(133)
310 b(2,5)=0
320 EI
330 REM ##### hauptprogramm #####
340 EVERY 100,3 GOSUB 890
350 EVERY 50,2 GOSUB 950
360 rx=0:ry=0:GOSUB 620:IF jo=0 THEN 360

```

```

370 IF jo AND 1 THEN ry=-1
380 IF jo AND 2 THEN ry=1
390 IF jo AND 4 THEN rx=-1
400 IF jo AND 8 THEN rx=1
410 IF rx=0 AND ry=0 THEN 360
420 x2=x+rx:y2=y+ry
430 pe=b(x2,y2)
440 IF pe = 3 THEN 780
450 IF pe = 4 THEN GOSUB 700
460 IF pe = 1 THEN GOSUB 560:GOTO 360
470 IF pe = 2 THEN GOSUB 1020
480 IF f1=1 AND x2=19 AND y2=4 THEN 1090
490 IF x2<2 THEN 360
500 IF x2>19 THEN 360
510 IF y2<5 THEN 360
520 IF y2>23 THEN 360
530 FOR i=1 TO 20:NEXT
540 LOCATE x,y:PRINT CHR$(32):x=x2:y=y2:LOCATE x,y:PRINT CHR$(128)
550 GOTO 360
560 REM Stein
570 PEN 1:LOCATE x,y:PRINT CHR$(129)
580 LOCATE x2,y2:PEN 6:PRINT CHR$(131)
590 SOUND 7,0,1,4,,,1
600 PEN 1
610 RETURN
620 REM Steuerung
630 jo=JOY(0):IF jo<>0 THEN RETURN
640 jo=0
650 IF INKEY(0)=0 THEN jo=jo+1
660 IF INKEY(2)=0 THEN jo=jo+2
670 IF INKEY(8)=0 THEN jo=jo+4
680 IF INKEY(1)=0 THEN jo=jo+8
690 RETURN
700 REM Kerze
710 ENV 1,5,3,1,15,-1,1
720 INK 2,13:INK 4,15:INK 5,12:INK 7,24
730 SOUND 7,284,20,4,1
740 FOR i=1 TO 150:NEXT
750 INK 2,0:INK 4,0:INK 5,0:INK 7,0
760 b(x2,y2)=0
770 RETURN
780 REM Verloren
790 AFTER 0,3 GOSUB 610:AFTER 0,2 GOSUB 610
800 INK 5,12:LOCATE x,y:PRINT CHR$(32)
810 INK 2,13:INK 4,15:INK 5,12:INK 7,24
820 FOR i=119 TO 239:SOUND 7,i,5,6
830 NEXT
840 CLS
850 PEN 1:PRINT"VERLOREN..."
860 PRINT:PRINT"Punkte "p
870 FOR k=1 TO 2000:NEXT k
880 RUN
890 INK 5,12:LOCATE 1,1:PRINT CHR$(7);
900 xpo=POS (#0):ypo=VPOS (#0)
910 FOR k=1 TO 40:NEXT
920 INK 5,0
930 LOCATE xpo,ypo
940 RETURN
950 REM Zeitabzug
960 zeit = zeit -1
970 xpo=POS (#0):ypo=VPOS (#0)
980 LOCATE 1,1:PEN 1:PRINT"Zeit:"zeit" P. : "p
990 IF zeit<=0 THEN 780

```



```

1000 LOCATE xpo,ypo
1010 RETURN
1020 REM Lampe
1030 f1=1
1040 GOSUB 700
1050 LOCATE 1,25:PRINT"DAS TOR IST OFFEN";
1060 LOCATE 19,4:PRINT CHR$(32)
1070 p=p+250
1080 GOTO 540
1090 REM Gewonnen
1100 AFTER 0,3 GOSUB 610:AFTER 0,2 GOSUB 610
1110 INK 2,13:INK 4,15:INK 5,12:INK 7,24
1120 p=p+500
1130 LOCATE x,y:PRINT" "
1140 LOCATE x2,y2:PRINT CHR$(130)
1150 FOR i=7 TO 0 STEP -1
1160 SOUND 7,239,40,i
1170 NEXT
1180 CLS:PEN 1:PRINT"GEWONNEN..."
1190 PRINT:PRINT"Punkte "p
1200 FOR k=1 TO 2000:NEXT k
1210 ERASE b:f1=0
1220 GOTO 10

```



HEYNE COMPUTER-BÜCHER
 Manfred S. Schmidt
DER MIKROCOMPUTER
IM BERUF
 Heyne-Buch Nr. 15/4, DM 9,80
 Originalausgabe

Variablenliste: Aladin

b : Bild-Variable zur Speicherung des Bildschirmes
 zeit : Zeit-Variable
 i,k : Schleifen
 x,y : Koordinaten des Aladin
 xr,yr : Zufalls-Variablen
 rx,ry : Richtung ,in die Aladin gehen soll
 x2,y2 : Position ,auf die Aladin gehen soll
 pe : Hilfsvariable
 f1 : Flag fuer Tor offen
 jo : Joystick/Tastatur - Variable
 xpo,ypo : Hilfsvariablen
 p : Punkte - Stand

10 - 320 : Printen des Spielfeldes
 330 - 350 : Festlegen der Interrupts
 360 - 420 : Steuerung
 430 - 550 : Erkennung der Zeichen
 560 - 610 : Laufen gegen einen Stein
 620 - 690 : Abfrage des Joysticks bzw. der Tastatur
 700 - 770 : Gehen auf eine Kerze
 780 - 880 : Verloren
 890 - 940 : Geisterblinken
 950 - 1010 : Zeitabzug
 1020 - 1080 : Finden der Lampe
 1090 - 1220 : Gewonnen

TRONIC-SOFTWARE-SERVICE

Atari, Mastermind/Schlangenkrieg, Kassetten 10,50 DM, Diskette 19,50 DM, Bestell-Nr. 141

Tank-Battle/Oil Panic, Kassetten 12,50 DM, Bestell-Nr. 151

Startup/Zeilen-Split/Tomstone-City, Diskette 19,50 DM, Bestell-Nr. 171

Painter/Hardcopy, Kassetten 14,- DM, Diskette 19,50 DM, Bestell-Nr. 181

The Big Quest/Fünf gewinnt, Kassetten 14,- DM, Diskette 19,50 DM, Bestell-Nr. 191

Splitt, Kassetten 11,- DM, Diskette 17,50 DM, Bestell-Nr. 1101

Ski/Mutation, Kassetten 14,50 DM, Diskette 19,50 DM, Bestell-Nr. 1121

Super Miner/Diamonds, Kassetten 14,50 DM, Diskette 19,50 DM, Bestell-Nr. 122

Donkey Kong/Kerzenheinz, Kassetten 16,50 DM, Diskette 19,50 DM, Bestell-Nr. 132.

TI-99, Drei-Kronen-Spiel/Zahlenputzen, Kassetten 8,50 DM, Bestell-Nr. T 41

Karl der Käfer/Alien-Landing, Kassetten 14,50 DM, Bestell-Nr. T 51

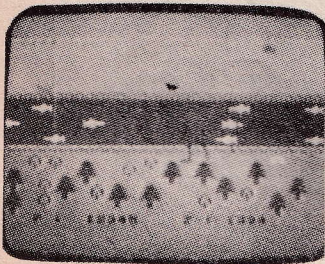
Jack the Digger I/Noah 2099, Kassetten 14,50 DM, Bestell-Nr. T 61

Lift Bär/ASC II DEF Teil 1, Kassetten 14,- DM, Bestell-Nr. T 71

Maya/ASC II DEF Teil 2, Kassetten 14,50 DM, Bestell-Nr. T 81

ASC II DEF Teil 1 + 2, Diskette 19,50 DM, Bestell-Nr. T 881

Transfer/Silverspar, Kassetten 14,50 DM, Diskette 19,50 DM, Bestell-Nr. T 91



Mother Duck/Screen Designer, Kassetten 14,50 DM, Diskette 19,50 DM, Bestell-Nr. T 101

Cave Man, Kassetten 8,- DM, Diskette 11,50 DM, Bestell-Nr. T 121

Moon Race/Frogger/Slicks, Kassetten 19,50 DM, Diskette 25,50 DM, Bestell-Nr. T 22

Panzerschlacht, Kassetten 8,- DM, Diskette 11,50 DM, Bestell-Nr. T 32.

Comodore 64, Mauern/Widerstand, Kassetten 8,- DM, Diskette 15,- DM, Bestell-Nr. C 41

Space-Comets/Erdspalte/Sprite-Data, Kassetten 15,- DM, Diskette 23,50 DM, Bestell-Nr. C 51

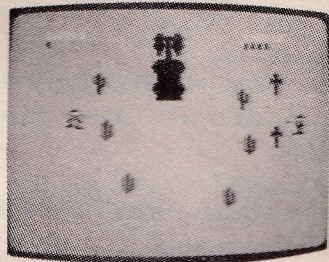
Autostart/Bestellschein/Roadpainter, Kassetten 16,50 DM, Diskette 23,50 DM, Bestell-Nr. C 61

Hardcopy/Space-Fighter/Data-Generator, Kassetten 15,50 DM, Diskette 19,50 DM, Bestell-Nr. C 71

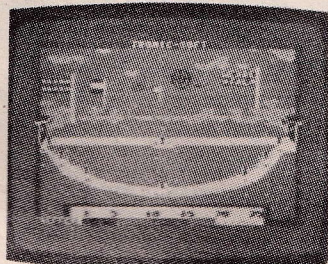
Monster-Attack/Block-Painter/Epson-Drucker, Kassetten 16,50 DM, Diskette 23,50 DM, Bestell-Nr. C 81

Projekt/Datenbank, Kassetten 16,- DM, Diskette 23,50 DM, Bestell-Nr. C 91

Spiders/The Basic, Kassetten 16,50 DM, Diskette 23,50 DM, Bestell-Nr. C 101



High Noon/Skeet/Grafik-Designer, Kassetten 17,50 DM, Diskette 23,50 DM, Bestell-Nr. C 121



Painter/Star-Baddle/Editor, Kassetten 17,50 DM, Diskette 23,50 DM, Bestell-Nr. C 22

Wüstenrallye/Jet-Pac/Black Moore Castle Kassetten 17,50 DM, Diskette 23,50 DM, Bestell-Nr. C 32.

VC-20, Bestellschein/Glücksky, Kassetten 8,50 DM, Diskette 15,- DM, Bestell-Nr. V 61

Multigraph/All Rammer, Kassetten 11,- DM, Diskette 15,50 DM, Bestell-Nr. V 71

Zyklus/Meteorit, Kassetten 11,- DM, Diskette 19,50 DM, Bestell-Nr. V 81

Garten/Schloß Gruselstein, Kassetten 14,- DM, Diskette 19,50 DM, Bestell-Nr. V 91

Fressman/Outlaw, Kassetten 14,- DM, Diskette 19,50 DM, Bestell-Nr. V 101

Prost/Bufalo Bill, Kassetten 14,- DM, Diskette 19,50 DM, Bestell-Nr. V 121

Joy Man/Powerpack, Kassetten 14,- DM, Diskette 19,50 DM, Bestell-Nr. V 22

Der rasende Malocher/Frankie goes to Pharao, Kassetten 14,- DM, Diskette 19,50 DM, Bestell-Nr. V 32.

Apple II, Wilder Westen/Karabollage/Maskengenerator, Diskette 19,50 DM, Bestell-Nr. A 41

Music-Maker/Mission: Adler/Disk-Katalog, Diskette 19,50 DM, Bestell-Nr. A 51

Snake/Super Datei/Shapeables, Diskette 19,50 DM, Bestell-Nr. A 61

Library/Fight, Diskette 19,50 DM, Bestell-Nr. A 71

Reversal/Disk-Menue-Generator, Diskette 19,50 DM, Bestell-Nr. A 81

Diamonds/Hilfsprogramm, Diskette 19,50 DM, Bestell-Nr. A 91

Tic-Tac-Toe/Jumper, Diskette 19,50 DM, Bestell-Nr. A 101

Donovan/Basic-Konverter, Diskette 19,50 DM, Bestell-Nr. A 121

Funktionstasten/Painter/Bowling, Diskette 19,50 DM, Bestell-Nr. A 22

Thunder/Castle of Doom, Diskette 19,50 DM, Bestell-Nr. A 32.

Dragon 32, Blizzard, Kassetten 8,- DM, Bestell-Nr. D 41

Space-Flight/Geosoft, Kassetten 10,- DM, Bestell-Nr. D 51

Waregames, Kassetten 8,50 DM, Bestell-Nr. D 61

Laser-Attack, Kassetten 8,50 DM, Bestell-Nr. D 71

Hardcopy, Kassetten 13,- DM, Bestell-Nr. D 81

Anwenderprogramm, Kassetten 10,- DM, Bestell-Nr. D 91

Dragon Paint, Kassetten 14,- DM, Bestell-Nr. D 101.

ZX-Spectrum, Inventur, Kassetten 12,- DM, Bestell-Nr. S 41

Missile-Comment, Kassetten 8,50 DM, Bestell-Nr. S 51

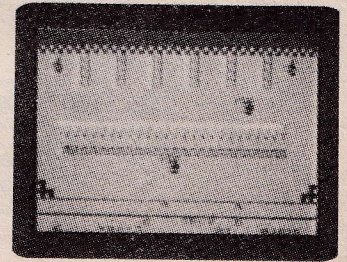
Defender/Lui der Wurm/Alternativer Zeichensatz, Kassetten 13,50 DM, Bestell-Nr. S 61

Matheprogramm/Bongo-Beeatcher, Kassetten 12,50 DM, Bestell-Nr. S 71

Solitaire/Superstat/Kleinste gem. Vielfache, Kassetten 14,50 DM, Bestell-Nr. S 81

Jump about, Kassetten 14,50 DM, Bestell-Nr. S 91

Pac-Man/Oil Panic, Kassetten 14,50 DM, Bestell-Nr. S 101



Frogger, Kassetten 16,- DM, Bestell-Nr. S 121

Jump, Kassetten 14,50 DM, Bestell-Nr. S 22

Jet Set Freddie, Kassetten 8,- DM, Bestell-Nr. S 32.

ZX-81, Go-Ball/Grand-Prix, Kassetten 10,- DM, Bestell-Nr. Z 51

Moon-Crash/ZX-Draw, Diskette 10,- DM, Bestell-Nr. Z 61

Tonprogramm/Aldebaran, Kassetten 10,- DM, Bestell-Nr. Z 71

Reversi, Kassetten 10,- DM, Bestell-Nr. Z 91

Panik Labyrinth, Kassetten 10,- DM, Bestell-Nr. Z 101

Expedition, Kassetten 10,- DM, Bestell-Nr. Z 121

Spinnen, Kassetten 14,50 DM, Bestell-Nr. Z 22

Spukhaus, Kassetten 14,50 DM, Bestell-Nr. Z 32.

CPU-SOFTWARE-SERVICE

Commodore 64, Multi-Key/S-Tool 64/Interrupt-Programme, Kassette 16,- DM, Diskette 21,- DM, Bestell-Nr. UC 2/85.

Spritehilfe/Diskloader/Directory, Diskette 21,- DM, Bestell-Nr. UC 3/85.

Tape-Directory/Asmon/Data-Generator, Kassette 15,- DM, Diskette 21,- DM, Bestell-Nr. UC 3/85.

Fast-Load, Kassette 10,- DM, Bestell-Nr. UC 4/85.

Diskmonitor/Zeichensatz, Diskette 20,- DM, Bestell-Nr. UC 5/85.

Reassembler/Maskengenerator, Kassette 15,- DM, Diskette 20,- DM, Bestell-Nr. UC 6/85.

Schneider CPC-464, Autoren, Kassette 11,- DM, Bestell-Nr. US 3/85.

Universal-Datei V1, Kassette 9,- DM, Bestell-Nr. US 4/85.

Computerschrift/Symbol-Swap, Kassette 10,- DM, Bestell-Nr. US 5/85.

Keyboard Toolkit/Farbdekodierung, Kassette 11,- DM, Bestell-Nr. US 6/85.

Aladin, Kassette 9,- DM, Bestell-Nr. US 6/85.

VC-20, Sprites/Space-Battle, Kassette 13,- DM, Diskette 19,- DM, Bestell-Nr. UV 2/85.

Grafik-Painter, Kassette 10,- DM, Bestell-Nr. UV 5/85.

Vic-Clock, Kassette 9,- DM, Bestell-Nr. UV 6/85.

Apple, Asteroids, Diskette 18,- DM, Bestell-Nr. UA 2/85.

Dateiverwaltung, Diskette 18,- DM, Bestell-Nr. UA 3/85.

Digitizer/Reset-Saver, Diskette 18,- DM, Bestell-Nr. UA 4/85.

Assamblor-Monitor, Diskette 17,- DM, Bestell-Nr. UA 5/85.

Disk Check IC/Dos Change, Diskette 18,- DM, Bestell-Nr. UA 6/85.

TI-99, Olympics, Kassette 11,- DM, Bestell-Nr. UT 2/85.

Funktionsanalyse/Hardcopy, Kassette 11,- DM, Bestell-Nr. UT 3/85.

Invaded by the Empire, Kassette 11,- DM, Bestell-Nr. UT 4/85.

Multicolor/Load-Programm, Kassette 14,- DM, Diskette 19,- DM, Bestell-Nr. UT 6/85.

ZX-81, The Castle of Questionmark, Kassette 10,- DM, Bestell-Nr. UX 2/85.

ZX-Spectrum, Panik Sam/Cour Change/Clock, Kassette 13,- DM, Bestell-Nr. UZ 2/85.

MC-Routinen/Data-Base/Pixel, Kassette 11,- DM, Bestell-Nr. UZ 3/85.

Reversi/UDG-Editor, Kassette 15,- DM, Bestell-Nr. UZ 4/85.

Star-Basic, Kassette 13,- DM, Bestell-Nr. UZ 5/85.

Fill-Programm/Copy it, Kassette 14,- DM, Bestell-Nr. UZ 6/85.

Atari, Mauerflucht/Lunar-Simulation, Kassette 13,- DM, Diskette 19,- DM, Bestell-Nr. UI 2/85.

Ram-Tester/Cherry-Harry, Kassette 13,- DM, Diskette 19,- DM, Bestell-Nr. UI 3/85.

Crazy Egon, Kassette 13,- DM, Diskette 19,- DM, Bestell-Nr. UI 5/85.

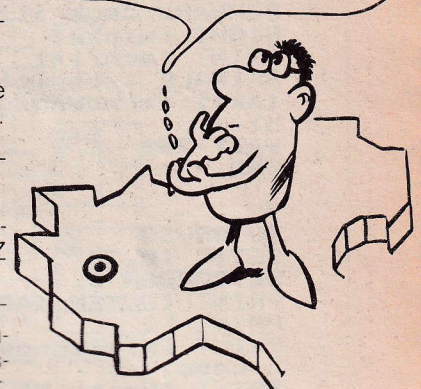
Monitorprogramm, Kassette 11,- DM, Bestell-Nr. UI 6/85.

MSX, Zeichensatzkonverter, Kassette 10,- DM, Bestell-Nr. UM 3/85.

Zirkus, Kassette 13,- DM, Bestell-Nr. UM 5/85.

Pipeline, Kassette 13,- DM, Bestell-Nr. UM 4/85.

Wie komme ich an Programme?



Ganz einfach! - Nutzen Sie den CPU-Software-Service!

Bestellung per Telefon:

Wenn es schnell gehen soll ... rufen Sie uns an. Wir nehmen Ihre Bestellung gern entgegen.

Tel.-Nr. 0 56 51 / 4 06 93 oder 0 56 51 / 4 06 43

Tronic-Verlag
Postfach 41
3444 Wehretal 1

CPU Bestellkarte-Softwareservice

Alle im Heft abgedruckten Programme können als zusätzlicher Service über den Verlag bezogen werden. (Ausland nur gegen Vorkasse)

Die Zustellung erfolgt: gegen **Vorkasse**

oder Inland per **Nachnahme**
+ Versandkosten

innerhalb von 1 Woche

Entnehmen Sie bitte aus unseren Preislisten die notwendigen Angaben für Ihre Bestellung:

Bitte liefern Sie mir:

Cassette für

_____ System

Anzahl

Diskette für

_____ System

Anzahl

zum Preis von gesamt

_____ DM

Name/Vorname: _____

Straße, Nr.: _____

PLZ/Ort: _____

Datum, Unterschrift _____

Proceduren für das Logo-Programm vom ZX-Spectrum

```
?
TO SPIRALE :schritt :winkel :erh
  oehung
  IF :schritt > 100 [stop]
  FORWARD :schritt
  RIGHT :winkel
  PRINT :schritt
  SPIRALE :schritt + :erhoehung :w
  inkel :erhoehung
HT
WAIT 200
END
```

```
?
TO GRUSS
  SETTC [0 7]
  SETBORDER 0
  PRINT [GUTEN TAG]
  PRINT []
  PRINT [Wie schoen, dass ich jetz
  t Logo]
  PRINT [kann. Du wirst es auch se
  hr]
  PRINT [schnell lernen, glaube es
  mir!]
  PRINT []
  PRINT [Ich zeige Dir jetzt ein w
  enig]
  PRINT [Grafik, ja?]
  WAIT 500
  END
```

?
Die Procedure 'Gruss'.

```
?
TO GRUSS
  WINDOW
  SETBR 2
  SETTC [2 6]
  BRIGHT 1
  SETCUR [0 10]
  PR [TAG AUCH, ICH BIN DIE KROETE
  .]
  SETTC [1 5]
  SETCUR [0 12]
  PR [SIEH NUR WAS ICH ALLES KANN!
  ]
  SETTC [7 0]
  SOUND [0.2 12]
  SOUND [0.2 16]
  SOUND [0.5 19]
  CT
  RAND
  SETBG 4
  SETPC 1
  SPIRALE 1 91 1
  WAIT 500
  END
```

?
Sie ruft einige andere Proce -
?
duren auf, z.B. 'Spirale'.

```
?
TO SPIRALE :STEP :ANGLE :INC
  IF :STEP > 200 [STOP]
  FORWARD :STEP
  RIGHT :ANGLE
  SOUND [.001 40]
  SPIRALE :STEP + :INC :ANGLE :INC
  END
```

?
Oder 'Rand' und 'Hintergrund'.

```
?
TO RAND
  PENUP
  SETPOS [-128 87]
  PENDOWN
```

```
SETPOS [127 87]
SETPOS [127 -88]
SETPOS [-128 -88]
SETPOS [-128 87]
PENUP
SETPOS [0 0]
PENDOWN
END
```

```
TO HINTERGRUND :P :I :B
  SETBG :P
  SETPC :I
  SETBR :B
  END
```

?

```
POALL
PO
POTS
SAVEALL
SAVESCR
.BSAVE
SAVED
.RESERVE
.CALL
.EXAMINE
.SERIALIN
ERASEFILE
CATALOG
?
```

```
PONS
POPS
SAVE
LOAD
LOADSCR
.BLOAD
LOADD
.RESERVED
.DEPOSIT
.SETSERIAL
.SERIALOUT
SETDRIVE
```

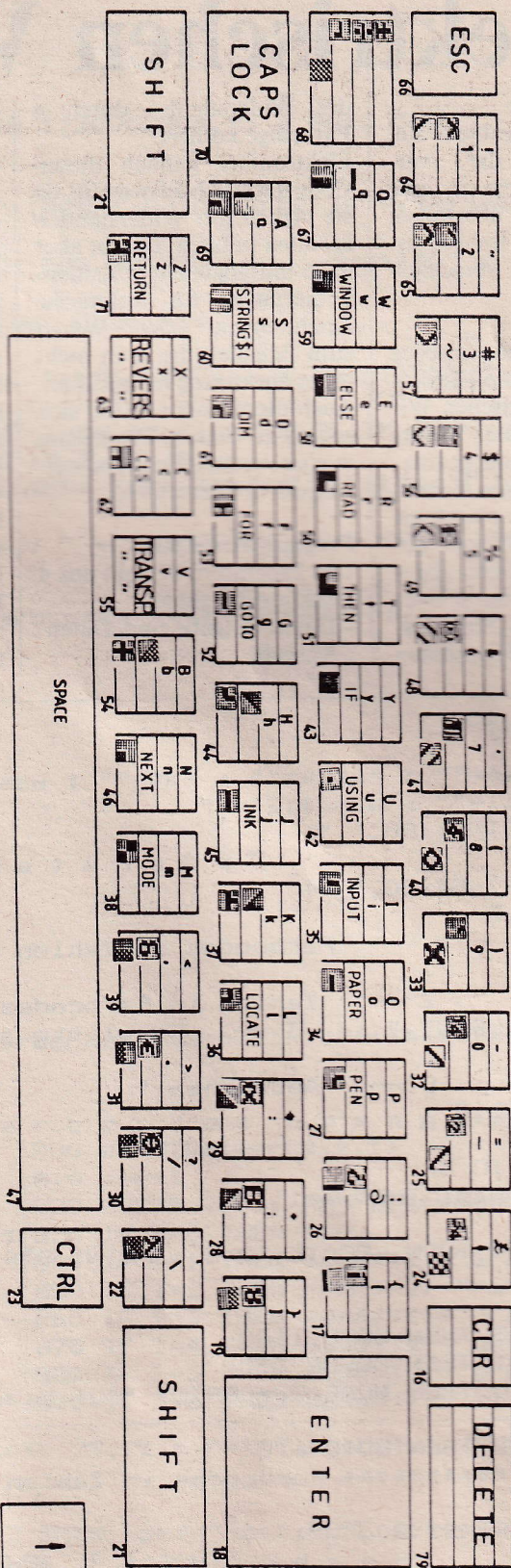
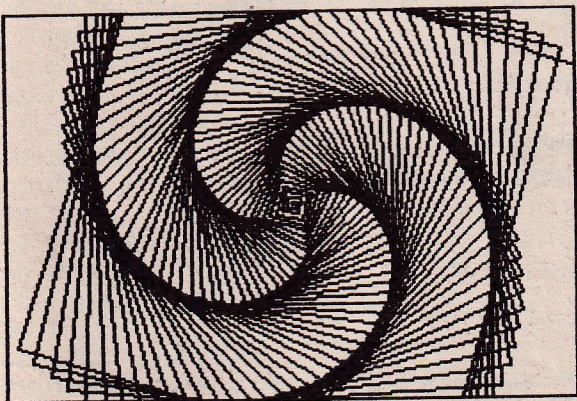
```
?
END
FALSE
(
OUTPUT
```

```
TO
EDIT
DEFINE
TEXT
THING
EQUALP
LISTP
NUMBERP
PRIMITIVEP
NOT
OR
IF
REPEAT
HIDETURTLE
SHOWTURTLE
RT
SETHEADING
HEADING
SETSCRUNCH
XCOR
CLEAN
CLEARSCREEN
FD
BK
LT
SETBG
CLEARTEXT
SETCURSOR
CURSOR
TC
BRIGHT
PC
BG
PD
PU
PE
PX
FENCE
WINDOW
```

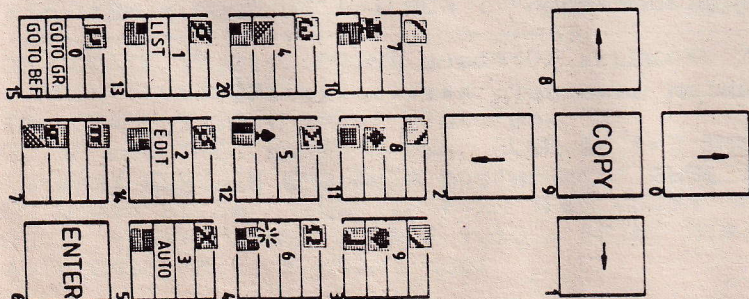
```
TRUE
)
OP
STOP
TOPLEVEL
ED
EDNS
COPYDEF
MAKE
=
EMPTYP
MEMBERP
WORDP
DEFINEDP
AND
NAMEP
RUN
HT
ST
COPYSCREEN
RIGHT
SETH
SETSCR
YCOR
SCRUNCH
HOME
CS
FORWARD
BACK
LEFT
SETPC
CT
SETCUR
SETTC
TEXTCOLOUR
OVER
PENCOLOUR
BACKGROUND
PENDOWN
PENUP
PENERASE
PENREVERSE
WRAP
SETX
```

Tastaturbelegung des Keyboard Toolkit

- | | |
|------------|--------------|
| SETY | TS |
| TEXTSCREEN | SETPOS |
| POSITION | POS |
| DOT | TOWARDS |
| SETBORDER | SETBR |
| SHOWNP | STARTROBOT |
| STOPROBOT | ASCII |
| BUTFIRST | BF |
| BUTLAST | BL |
| CHAR | COUNT |
| FIRST | FPUT |
| ITEM | LAST |
| LIST | LPUT |
| SENTENCE | SE |
| WORD | + |
| SUM | - |
| * | PRODUCT |
| / | DIV |
| RANDOM | REMAINDER |
| SQRT | > |
| < | ROUND |
| INT | ARCSIN |
| ARCCOS | ARCTANGENT |
| ARCTAN | ARCCOTANGENT |
| ARCCOT | SINE |
| SIN | TANGENT |
| TAN | COTANGENT |
| COT | COSINE |
| COS | READLIST |
| RL | READCHAR |
| KEYP | RC |
| BYE | .PRIMITIVES |
| .CONTENTS | NORMAL |
| INVERSE | FLASH |
| PRINTON | PRINTOFF |
| PRINT | PR |
| SHOW | SOUND |
| WAIT | TYPE |
| NODES | RECYCLE |
| ERNS | ERALL |
| ERPS | ERASE |
| ER | ERN |



IN BEIDEN MODI ERREICHBAR
IM BEFEHLSMODUS ERREICHBAR MIT CTRL
IM GRAFIKMODUS ERREICHBAR MIT CTRL
TASTENNUMMER



Die Cursortasten sind
direkt neben der
Tastaturbelegung angeordnet

CPC 464: Farbdekodierung von elektrischen Widerständen

Aufgabenstellung:

Häufig tritt das Problem auf, Informationen, die verschlüsselt vorliegen, in eine lesbare Form zu übertragen oder, umgekehrt, lesbare Informationen zu verschlüsseln.

Ein einfaches, aber typisches Beispiel dafür ist die Farbkodierung von elektrischen Widerständen. Da die Widerstände oft sehr kleine geometrische Abmessungen haben, ist man von der vor noch einigen Jahren üblichen Beschriftung mit Ziffern abgekommen, da diese nur schwer zu erkennen waren. Statt dessen wurden die Kenndaten

des Widerstandes durch 4 Farbringe kodiert.

Elektroniker kennen diesen Code natürlich auswendig, da sie täglich mit Widerständen arbeiten müssen. Es gibt aber auch umfangreichere Zahlenkodierungen für technische Zeichnungen oder Waren zum Beispiel, die man nicht mehr auswendig lernen kann. Auch diese Codes lassen sich analog zu diesem Programm in eine verständliche Darstellungsform übersetzen.

Programmstruktur:

Das Programm besteht aus 4 Funktionsblöcken:

- Daten-Eingabe und Daten-Ausgabe

- Tabelle mit dem Farbkodierungs-Schlüssel

- Entscheidungs-Tabelle zur Verknüpfung der Daten

- Grafische Darstellung

Die Verknüpfung der Daten erfolgt mit String-Verarbeitungs-Befehlen, da Text- und Zahleninformationen miteinander zu verknüpfen sind.

Mit geringen Änderungen läßt sich das Programm auch zur Farbdekodierung von elektrischen Kondensatoren verwenden.

Farbkodierung der elektrischen Widerstände:

Der Farbring, der dem einen Ende des Widerstandes am nächsten ist, wird als 1. Farbring bezeichnet, die benach-

barten Farbringe werden als 2., 3. und 4. Ring bezeichnet. Die ersten beiden Ringe stellen jeweils eine der Ziffern 0, 1, 2, 3, 4, 5, 6, 7, 8 oder 9 dar. Der 3. Ring gibt die Komma-stellen an. Das Komma kann entweder direkt hinter der zweiten Ziffer stehen oder um bis zu 6 Stellen nach rechts verschoben sein. Durch Multiplikation mit 0.01 oder 0.1 kann das Komma auch um 2 oder um 1 Stelle nach links verschoben werden.

Der 4. Farbring gibt die Toleranz des Widerstandes in Prozent vom Nennwert an. Die Farbkodierungs-Tabelle ist am Ende des Programms dargestellt.

```

10 PLOT 1,1,13
20 MODE 2
30 fak=1 : a$="" : ausd$="" : fa$="" : we=0 : w$="" : we$="" : o=0 : ausd$(0)=
   "" : ausd$(1)="" : ausd$(2)=""
40 PRINT STRING$(80,"_")
50 PRINT "                               W i d e r s t a n d s c o d i e r u n g                               "
60 PRINT STRING$(80,"_")
70 LOCATE 1,8
80 PRINT "      1 --> Farbcodes in Zahlen umwandeln."
90 PRINT
100 PRINT "      2 --> Zahlen in Farbcodes umwandeln."
110 PRINT : PRINT "      3 --> Farbliste ausdrucken." : PRINT
120 PRINT
130 PRINT "      0 --> Abbrechen."
140 LOCATE 50,8
   ";
160 FOR i=0 TO 3
170   IF i=wahl THEN a=1
180 NEXT i
190   IF a=0 THEN LOCATE 50,8 : PRINT STRING$(30," ") : SOUND 1,200,200,15 : GO
   TO 140
200 a=0
210 IF wahl=3 THEN GOTO 1180
220 IF wahl=1 THEN GOSUB 250
230 IF wahl=0 THEN MODE 2 : PRINT "Auf Wiedersehen" : FOR i=1 TO 3000 : NEXT i :
   CALL bd06
240 IF wahl=2 THEN GOTO 670
250 REM ***** Farbcodes in Zahlen umwandeln *****
260 CLS
270 PRINT STRING$(80,"_")
280 PRINT "      F a r b c o d e s   i n   Z a h l e n   u m w a n d e l n"
290 PRINT STRING$(80,"_")
300 GOSUB 1020
310 LOCATE 1,8
320 FOR i=1 TO 3
340   IF i=1 THEN LOCATE 15,21 : INPUT farb$(1)
350   IF i=2 THEN LOCATE 29,21 : INPUT farb$(2)

```

```

360 IF i=3 THEN LOCATE 41,21 : INPUT farb$(3)
370 IF i=1 AND UPPER$(farb$(1))="SCHWARZ" THEN SOUND 1,200,200,15 : GOTO 340
380 REM ***** Vergleichen mit den Farbcodes *****
390 farb$(i)=UPPER$(farb$(i))
400 RESTORE
410 READ fa$,we
420 fa%=UPPER$(fa$)
430 IF fa$=" " : " THEN RESTORE : SOUND 1,2000,200,15 : GOTO 340
440 IF i<>3 THEN 470
450 IF farb$(i)="SCHWARZ" THEN 490
455 IF fa%=farb$(i) AND fa%="SILBER" THEN fak=100 ELSE IF fa%=farb$(i) AND fa%="GOLD" THEN fak=10
460 IF fa%=farb$(i) AND we<>0 THEN FOR j=1 TO we : a%=a$+"0" : NEXT j : GOTO 490 ELSE 410
470 IF fa%=farb$(i) THEN a%=a$+STR$(we) ELSE 410
480 NEXT I
490 LOCATE 1,8 : PRINT "Widerstandswert : ";VAL(a$)/fak : LOCATE 55,21 : INPUT farb$(4)
500 fa%=UPPER$(farb$(4)) : IF fa%="BRAUN" THEN T$="1 %" ELSE IF fa%="ROT" THEN T$="2 %" ELSE IF fa%="GRUEN" THEN T$="0.5 %" ELSE IF fa%="GOLD" THEN T$="5 %" ELSE IF fa%="SILBER" THEN T$="10 %" ELSE IF fa%="" THEN T$="20 %" ELSE T$="Nicht vorhanden"
510 LOCATE 1,10 : PRINT "Toleranz " : LOCATE 24,10 : PRINT T$
"
530 IF INKEY$=CHR$(13) THEN 20 ELSE 530
540 REM ***** Datenzeilen fuer die Farbcodes *****
550 DATA braun,1
560 DATA schwarz,0
570 braun,1
580 DATA rot,2
590 DATA orange,3
600 DATA gelb,4
610 DATA gruen,5
620 DATA blau,6
630 DATA Violett,7
640 DATA grau,8
650 DATA weiss,9
655 DATA silber,0.01
657 DATA gold,0.1
660 DATA " : ",0
670 REM ***** Zahlen in Farbcodes umwandeln *****
680 CLS
690 PRINT STRING$(80,"_")
700 PRINT " Zahlen in Farbcodes umwandeln"
710 PRINT STRING$(80,"_")
720 GOSUB 1020
730 LOCATE 1,8
740 INPUT "Wert des Widerstandes ";wert$
750 l=LEN(wert$)-2
760 w$=LEFT$(wert$,1)
770 p=p+1
780 GOSUB 970
790 o=o+1
800 w%=MID$(wert$,2,1)
810 p=p+1
820 GOSUB 970 : REM Bestimmen der Farben
830 IF l=0 THEN 900
840 IF l=1 THEN ausd$(2)="braun"
850 IF l=2 THEN ausd$(2)="rot"
860 IF l=3 THEN ausd$(2)="orange"
870 IF l=4 THEN ausd$(2)="gelb"
880 IF l=5 THEN ausd$(2)="gruen"
890 IF l=6 THEN ausd$(2)="blau"

900 LOCATE 17,21 : PRINT ausd$(0) : LOCATE 29,21 : PRINT ausd$(1) : LOCATE 42,21 : PRINT ausd$(2)
910 LOCATE 1,10 : INPUT "Toleranz ";t
920 IF t=1 THEN spe$="Braun" ELSE IF t=2 THEN spe$="Rot" ELSE IF t=0.5 THEN spe$="Gruen" ELSE IF t=5 THEN spe$="Gold" ELSE IF t=10 THEN spe$="Silber" ELSE IF t=20 THEN spe$="Keine Angabe / Standart"
930 IF spe$="" THEN LOCATE 1,10 : PRINT STRING$(80," ") : LOCATE 1,10 : GOTO 910
940 LOCATE 55,21 : PRINT spe$
"
960 IF INKEY$<>CHR$(13) THEN 960 ELSE 20
970 REM ***** Bestimmen der Farben *****

```

CPU-User

```

980 RESTORE
990   READ fa$,we
1000   IF we=VAL(w$) THEN ausd$(o)=fa$ ELSE 990
1010 RETURN
1020 REM ***** Widerstand zeichnen *****
1030 ORIGIN 100,200 : DRAWR 400,0 : DRAWR 0,2 : DRAWR -400,0
1040   DRAWR 0,-100 : DRAWR 400,0 : DRAWR 0,-2 : DRAWR -400,0
1050 ORIGIN 500,200 : DRAWR 0,-100 : DRAWR 2,0 : DRAWR 0,102
1060 ORIGIN 100,100 : DRAWR 0,100 : DRAWR 2,0 : DRAWR 0,-102
1070 FOR i=1 TO 40
1080   ORIGIN 120+i,100 : DRAWR 0,100
1090   ORIGIN 220+i,100 : DRAWR 0,100
1100   ORIGIN 320+i,100 : DRAWR 0,100
1110   ORIGIN 430+i,100 : DRAWR 0,100
1120 NEXT i
1130 FOR i=1 TO 5
1140   ORIGIN 50,147+i : DRAWR 50,0
1150   ORIGIN 500,147+i : DRAWR 50,0
1160 NEXT i
1170 RETURN
1180 REM ***** Tabelle ausdrucken *****
1190 MODE 2
1200 PRINT STRING$(80,"_")
1210 PRINT "          Farbtabelle "
1220 PRINT STRING$(80,"_")
1230 LOCATE 1,8
1240 PRINT "Kennfarbe :      1.Ziffer :      2.Ziffer :      Multiplikator :      To
leranz : (%)"
1250 PRINT STRING$(80,"_")
1260 PRINT "Schwarz          -          0          -          -
_"
1270 PRINT "Braun           1          1          0          +-
1"
1210 PRINT "          Farbtabelle "
1220 PRINT STRING$(80,"_")
1230 LOCATE 1,8
1240 PRINT "Kennfarbe :      1.Ziffer :      2.Ziffer :      Multiplikator :      To
leranz : (%)"
1250 PRINT STRING$(80,"_")
1260 PRINT "Schwarz          -          0          -          -
_"
1270 PRINT "Braun           1          1          0          +-
1"
1280 PRINT "Rot               2          2          00          +-
2"
1290 PRINT "Orange           3          3          000          -
_"
1300 PRINT "Gelb             4          4          0000          -
_"
1310 PRINT "Gruen            5          5          00000          +-
0.5"
1320 PRINT "Blau             6          6          000000          "
1330 PRINT "Violett         7          7          -          -
_"
1340 PRINT "Grau             8          8          -          -
_"
1350 PRINT "Weiss           9          9          -          -
_"
1360 PRINT "Gold            -          -          * 0.1          +-
5"
1370 PRINT "Silber          -          -          * 0.01          +-
10"
1380 PRINT "ohne           -          -          -          +-
20"
"
1400 IF INKEY$<>CHR$(13) THEN 1400
1410 GOTO 20

```

Ab sofort wieder bei Ihrem Zeitschriftenhändler erhältlich!

monatlich

Compute mit

COMMODORE & SCHNEIDER

5/85

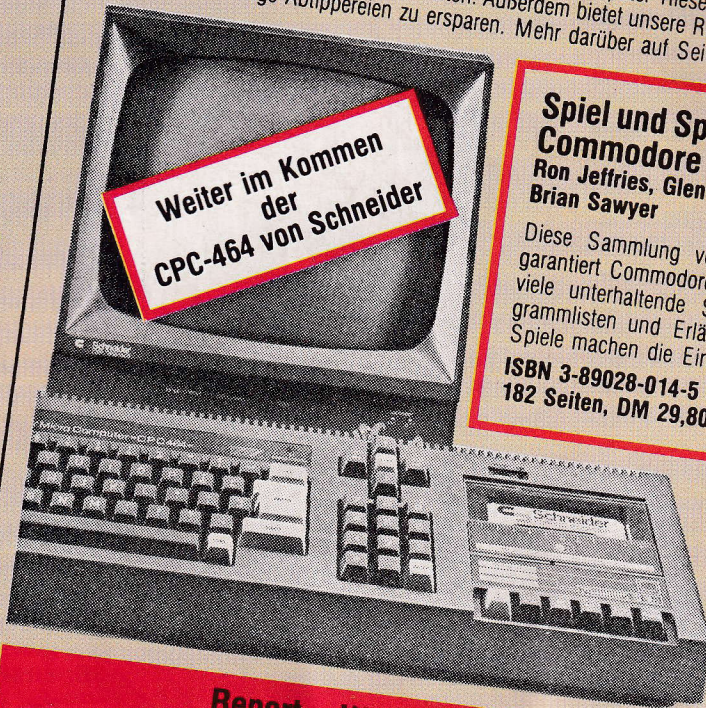
VC-64, VC-20, C-16, CPC-464

Unabhängiges Magazin für Anwender von Commodore- und Schneider-Computern

2,80 DM
24 öS
2,80 sFr

Aktuell +++ Aktuell +++ Aktuell +++ Aktuell

„COMPUTE MIT...“ haben wir, lieber Leser, speziell auf die Commodore- und Schneider- (hier der CPC 464)-Computer zugeschnitten. Nachdem sich schon im Herbst 1984 auf dem deutschen Computer-Markt abzeichnen schien, daß der CPC 464 nach einem nahezu kometenhaften Aufstieg gute Chancen besitzt, dem Giganten Commodore ein ernstzunehmender Konkurrent zu werden, wurde unser Team nahezu gezwungen, diesen Trend aufzunehmen und auch weiterhin zu verfolgen. D. h., für beide populären Homecomputer-Riesen werden wir in jeder Ausgabe von „COMPUTE MIT...“ ein attraktives Angebot bereithalten. Außerdem bietet unsere Redaktion einen Super-Service an, um Ihnen eventuelle schwierige Abtippereien zu ersparen. Mehr darüber auf Seite 11!



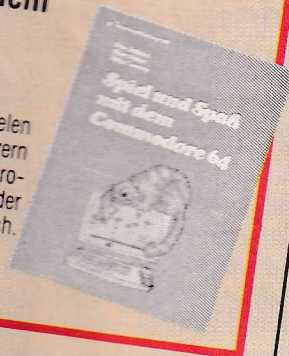
Weiter im Kommen der CPC-464 von Schneider

Spiel und Spaß mit dem Commodore 64

Ron Jeffries, Glen Fisher & Brian Sawyer

Diese Sammlung von 35 Spielen garantiert Commodore 64-Benutzern viele unterhaltende Stunden. Programmlisten und Erläuterungen der Spiele machen die Eingabe einfach.

ISBN 3-89028-014-5
182 Seiten, DM 29,80



Tips & Tricks für CPC-464

ROM-Calls ohne Einsprungsbedingungen
mehr darüber auf Seite 55

Report – Werkstatt – Hardware – Bücher –
Software – Tips & Tricks – Assemblerkurs Teil 1 –
Jungle – Sequenzer – Olympia – Cobra – Krümi – Q-Bert u. v. mehr

STOP + + + Neue Ausgabe + + + STOP

Tastatur:

- Funktionstasten frei programmierbar
- Cursor-Steuerung
- Numerisches Tastenfeld
- Neigungswinkel 5 - 15 Grad
- Feststell-Tasten mit optischer Anzeige

Bildschirme:

- TI-Modus
720 x 300 Bildpunkte
Kapazität 2000 Zeichen
(25 x 80)

- 8 Farben bzw. Grünstufen Grafische Darstellungen
- PC AT-Modus
640 x 200 Bildpunkte (2 Farben)
- 320 x 200 Bildpunkte (4 Farben)
- Kapazität 2000 Zeichen Grafische Darstellungen
- Monochrom und Farb-Bildschirm anschließbar an Video-Steuereinheit für TI-Modus, PC AT-Modus oder beide

- Bildschirm neig- und schwenkbar
- je zwei Anschlüsse für Tastatur/Maus

Betriebssysteme:

- MS-DOS 3.X für Einplatz-Betrieb
- XENIX 3 für Mehrbenutzer-Betrieb (max. acht) in Vorbereitung

Programmiersprache:

- Assembler, Basic, Cobol, Pascal, Fortran, C, Lisp

Software für Datenübertragung:

- Novell Netware/E-TI LAN mit File- und Record locking
- Ethernet LAN
- TTY, 3101
- 3780 Stapel-Verarbeitung
- 327 x BSC/SNA
- 931 Bildschirm-Emulator (TI Bürocomputer BS 300A/600A/800A)

Der PC-D von Siemens auf der Hannover-Messe

Während der diesjährigen Hannover-Messe zeigte Siemens auf seinem Messestand in der CeBIT-Halle 1 (Nr. B 5102/5202) in einem eigens eingerichteten PC-Präsentationsforum sein derzeitiges Angebot an Personalcomputern. Mehrmals täglich werden dort in einer großen Präsentation von einem Moderator die Personalcomputer PC-D (MS-DOS), PC-X und PC-MX (beide Sinix/Unix) vorgestellt. Beispielhaft für die breiten Einsatzgebiete dieser PC sind nachfolgend die Vorführungen auf dem PC-D erläutert.

An einem **Sekretariatsarbeitsplatz** wurde die Textbed- und Textverarbeitung mit dem Programm MS-Word gezeigt. Ein weiteres Textver-

arbeitungsprogramm ist Text-Auss-Window, das als eines der ersten deutschen Programme mit der neuen Fenster-technik arbeitet und zusätzliche Bürofunktionen bietet. Mit diesen Funktionen verwaltet die Sekretärin auf dem PC-D Termine und kann mit der Taschenrechnerfunktion z.B. Reisekosten ausrechnen. Abrufen, Sortieren und Pflegen einer Adres- sendatei wird mit dem Datenbanksystem dBase II erledigt, während Preiskalkulationen oder auch Umsatzplanungen mit dem Programm Multiplan sekundenschnell erstellt werden.

Auf einem weiteren PC-D wurde die Fähigkeit zur **Kommunikation** demonstriert. So hat der Personalcomputer



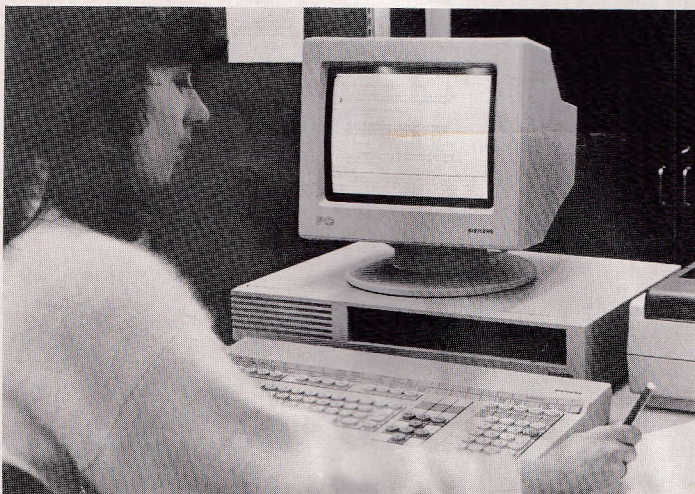
z.B. Zugang zum internationalen Teletex-Netz oder kann über eine Datenfernübertragungseinrichtung auf Dateien in Siemens- und auch IBM- Computern zugreifen. Dazu emuliert der PC-D die Datensichtgeräte 9750 oder 3270.

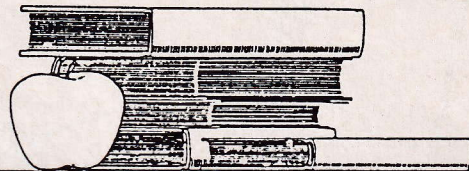
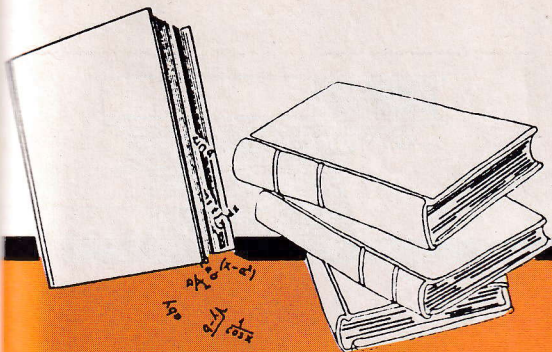
Ferner wurde mit dem PC-D demonstriert, wie vielseitig **Bürografik** verwendet werden kann. Über das Softwarepaket Open Access lassen sich Berechnungen, Texte und Grafiken integrieren. Die Texte werden dabei mit den Programmkomponenten MS Word Graphik, die Graphiken mit MS Chart Graphik und die Berechnungen mit der Programmiersprache GW Basic bearbeitet.

Klassisches Einsatzfeld für Personalcomputer sind die **betriebswirtschaftlichen Auf-**

gabenbereiche der Unternehmen. In Zusammenarbeit mit Software-Partnern wurden auf den PC-D in Hannover entsprechende Anwendungen wie Auftragsabwicklung, Finanzbuchhaltung und Lohn- und Gehaltsabrechnung vorgestellt.

Ergänzt wurden diese Demonstrationsbeispiele durch einige **Branchensoftwarepakete**. So zeigte Siemens Software für Handwerker, die durch den PC-Einsatz wesentlich rationellere Verwaltungsarbeit in Maler- und Dachdeckerbetrieben erreichen. Ein anderes Beispiel ist die Hausverwaltung mit ihrer Vielzahl von Abrechnungen für die Wohnungseigentümer und Mieter; diese Aufgaben werden mit dem Programm Cyrus auf dem PC-D abgewickelt.





C-64 – Akustik und Grafik

Von John J. Anderson, te-wi-Verlag München – ISBN 3-921803-31

Wer je die Umständlichkeit des C-64 beim Programmieren von Grafiken und Tonergebnissen beklagt hat, findet hier Abhilfe.

Dieser Titel zeigt alle Möglichkeiten, die in BASIC2, in Structured Basic und in Simon's Basis für Grafik und Akustik am C-64 zur Verfügung steht.

Dieses Buch nun soll zeigen, daß die Beschäftigung mit Ton- und Grafikerzeugung des C-64 nicht unbedingt zur Frustration führen muß.

Hierzu werden zwei BASIC-Erweiterungen zum C-64 vorgestellt, die unter anderem das Programmieren von Grafik- und Klangeffekten erst ermöglichen – es wäre falsch zu sagen „vereinfachen“, denn diese Sprachen machen Grafik und Ton überhaupt erst zugänglich. Sicherlich lassen sich diese Einrichtungen des C-64 auch mit dem Standard-BASIC programmieren. Genauso wie jeder Anwender stieß jedoch auch ein Engländer namens

Andrew Simon auf die Schwierigkeiten mit dem konventionellen BASIC. Er war sich bald darüber klar, daß eine Programmiersprache gewisse Bedingungen erfüllen mußte, um ein nützliches Hilfsmittel sein zu können. Das wichtigste ist ein Befehlssatz mit leicht verständlichen Anweisungen, die dem Anwender die unkomplizierteste Formulierung seiner Aufgabe ermöglicht.

Dieser Mr. Simon machte sich also daran, ein neues und leistungsfähigeres BASIC zu entwickeln, dem er seinen Namen gab. Und als er diese Arbeit beendet hatte, war er gerade 16 Jahre alt.

Das Ergebnis der Arbeit dieses Mr. Simon ist ein Programmpaket, das zu den nützlichsten Hilfsmitteln gehört, die für die Grafik- und Tonerzeugung auf dem C-64 überhaupt angeboten werden. Es gibt dieses Programm als preisgünstiges Steckmodul, das in die Rückseite des Computers eingesteckt wird. Es verleiht dem C-64 ein völlig neues BASIC, das sich unter anderem durch besondere Grafik- und Klanganweisung auszeichnet. Es ist sicherlich nicht übertrieben zu sagen, daß erst eine BASIC-Erweiterung den Zugriff auf alle Fähigkeiten des C-64 ermöglicht.

Darüber hinaus befaßt sich dieses Buch auch mit den Eigenschaften, die das Structured BASIC für den Grafik- und Tonerzeugung bietet. Die Herren Standke und Hartwig waren ebenfalls mit den Möglichkeiten des Standard-BASIC nicht zufrieden. Anstatt jedoch aufzugeben und in Zu-

kunft Computer Computer sein zu lassen, machten sich beide daran, eine BASIC-Erweiterung zu schaffen, die äußerst anspruchsvolle Möglichkeiten bietet. Wie der Name schon andeutet, ermöglicht dieses BASIC die strukturierte Programmierung, die besonders bei umfangreichen Programmen eine gesteigerte Übersichtlichkeit ergibt. Die Eigenschaften von Structured BASIC, die der allgemeinen Programmierertechnik nützen, sind jedoch im vorliegenden Werk nur am Rande erwähnt, da dieses Buch eben unter dem Motto der Grafik- und Tonerzeugung steht. In den entsprechenden Kapiteln findet sich stets eine gemeinsame Überschrift wie z. B. „Kreiserzeugung“, unter der für Structured BASIC und Simon's BASIC die nötigen Anweisungen erläutert sind. Steht Structured BASIC oder Simon's BASIC nicht zur Verfügung, so bedeutet dies eine im Grunde genommen völlig unnötige Beschränkung in bezug auf die Möglichkeiten des Computers. Der Grund liegt nicht nur in der dürftigen Dokumentation, die dem C-64 vom Hersteller beigelegt wird. Auch das in der Maschine in einem Festwertspeicher (ROM) enthaltene Standard-BASIC erweist sich gerade für den Anfänger als ausgesprochen unhandlich. Zur Programmierung von Grafik- oder Klangeffekten sind zahlreiche POKE-Befehle nötig. Das heißt, daß vom BASIC aus Befehle in Maschinensprache eingegeben werden müssen. Nun un-

Fortsetzung S. 76

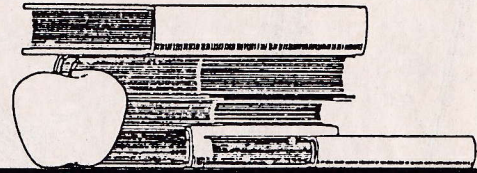
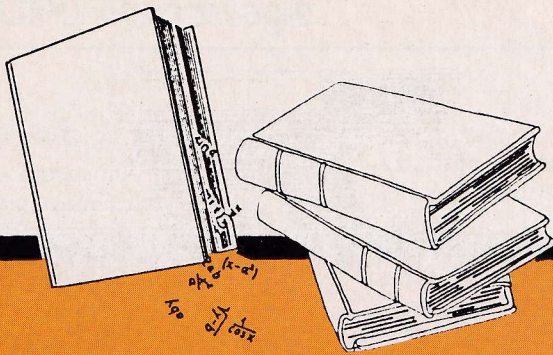
Programme
in
STRUCTURED - SIMONS'S
UND VERSION-2-BASIC

C64 AKUSTIK UND GRAPHIK



John J. Anderson

te-wi



terscheiden sich die POKE-Befehle untereinander nur durch ihre Argumente (das sind die Zahlen, die dem jeweiligen Befehl nachgestellt sind) und enthalten somit keinerlei Hinweise auf die Wirkungsweise. Ohne intensives Studieren der internen Strukturen des C-64 sind diese Befehle deshalb schlichtweg unverständlich.

Grundsätzlich kann man also sagen, daß beide hier angesprochenen BASIC-Erweiterungen eine feine Sache sind. Sie sind es wert, daß Sie ihnen mit Hilfe dieses Buches auf den Grund gehen, um die Leistungsfähigkeit des C-64 ergründen zu können. Bereits nach kurzer Zeit werden Grafik, Klänge und sogar Musik erzeugbar.

Vom C-Programmierer wird größte Disziplin und genaueste Kenntnis der Sprache verlangt. Nur so läßt sich erfolgreich programmieren, nur so lassen sich lesbare, sichere und portable Programme schreiben.

Auf diese Ausgabe wurde schon in der CPU Nr. 5 ausführlich eingegangen.

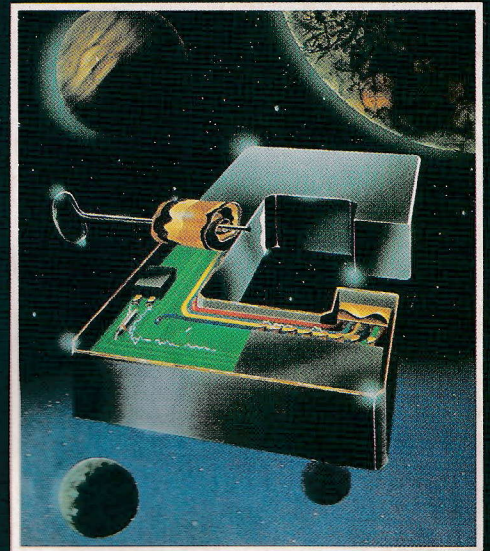
„Erfolgreiches Programmieren mit C“

von J. A. Illik

Das von Bell Laboratories entwickelte Betriebssystem UNIX hat sich zum Standardbetriebssystem für Mini- und Mikrocomputer der gehobenen Leistungsklasse entwickelt. Gleichzeitig eroberte sich in einem wahren Siegeszug die Programmiersprache C die Welt der Computer.

C ist eine universell verwendbare Sprache und ein wenig vom Hauch der Allmächtigkeit umgeben. Leistungsattribute wie Kompaktheit, Effizienz und Schnelligkeit werden meist in einem Atemzug mit der Sprache C erwähnt. Reizvoll ist auch die Möglichkeit, Hardware-Einrichtungen explizit kontrollieren zu können. Tatsächlich verbindet die Sprache die Fähigkeiten von Assemblersprachen mit dem Komfort von Hochsprachen. Man muß sich allerdings bewußt sein, daß die Leistungsfähigkeit und das hohe Maß an Freiheit auch ihren Preis haben.

Erfolgreich
programmieren
mit C



Spielen, lernen und arbeiten mit dem ATARI

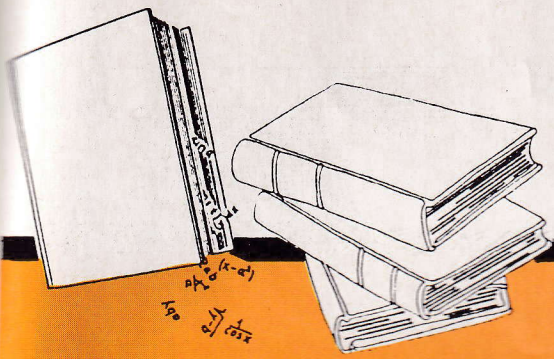
Erschienen im Sybex-Verlag, Düsseldorf; von Georg-Peter Raabe und Klaus-Jürgen Schmidt ISBN 3-88745-082-5

Dieses Buch hat die Zielsetzung, theoretische und praktische Kenntnisse für die Anwendung des Homecomputers Atari zu vermitteln. Die große Vielfalt der Einsatzmöglichkeiten eines solchen Rechners kann besonders für den Anfänger in der „Home-

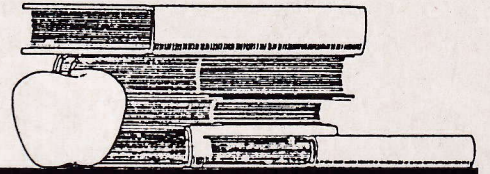
computerei“ wegen der Unüberschaubarkeit des Gebiets leicht zu einer gewissen Hilflosigkeit führen. Als gar nicht seltene Folge führt dies schließlich dazu, daß dieses preiswerte, aber doch nicht ganz billige Gerät nach kurzer Zeit nicht mehr benutzt wird.

Damit dies nicht geschieht, sollte der Rechner von Anfang an schrittweise aufbauend, immer besser kennengelernt und beherrscht werden. Die Absicht dieses Buches ist es, einen Weg für die Eroberung des Atari zu zeigen, der in angenehmer, spielerischer

Art verläuft und zugleich angemessenes, solides EDV-Handwerk vermittelt. Über die Phasen SPIELEN, LERNEN und ARBEITEN sollen Beispiele und Anregungen zu Fähigkeiten verhelfen, die bei der Programmierung und Anwendung des



CPU Bibliothek



Atari direkt benötigt werden und die es ermöglichen, auch größere Programmkomplexe anderer Programmierer und Hersteller weiterer BASIC-Computer zu verstehen.

Die wichtigsten Kapitel aus dem Inhalt:

Kapitel 1: EINFÜHRUNG
Beschreibung des ATARI
Ergänzungen und Erweiterungen

ungen des ATARI-Systems
Kapitel 2: SPIELEN
Grundlagen (Toolbox)
Die Programmierung von Pausen und Verzögerungen
Spiele mit dem Zufall
Bestimmung der Kreiszahl π mittels Zufallszahlen
Geläufigkeitsübung für die Tastatur des ATARI
Die Gestaltung eigener Zeichen

Der Sound des ATARI
Maschinenprogramm für die Soundsteuerung
Erzeugen einer Tonleiter
ATARI-Orgel
Kapitel 3: LERNEN
Bits, Bytes und deren Anwendung
Der Aufbau des ATARI
Programmierung
Kapitel 4: ARBEITEN
Dateneingabe und Menü-

technik
Kalender Unterprogramme
Biorhythmus
Balkendiagramme
Piecharts
Kurvendarstellung: Zykloiden
Programme für private Anwendungen
Neutrales Hauptmenü und noch weitere Gliederungspunkte

MSX - Der erste Heimcomputer-Standard

Einführung in Technik, Programmierung, Anwendung

Autor: Dietmar Eirich, Heyne-Verlag München - ISBN-3-453-47052-4, Preis 12,80 DM

Mit den Geräten der großen MSX-Familie ist das Computern vielfältiger, noch einfacher. Die neue Technik ermöglicht den Einstieg in die Computerwelt auf besonders einfache Weise. Das vorliegende Buch erklärt:

- Was heißt MSX - welche Technik steckt dahinter?
- Wie steht es mit der Software?
- Welche Anwendungen gibt es - welche sind wirklich sinnvoll?

Dieses Buch ist eines der ersten, das sich intensiv mit dem neuen Standard befaßt. Es zeigt die Hintergründe über MSX auf, macht mit den Geräten, dem Programmieren und den speziellen Anwendungen vertraut. Nebenbei werden Informationen über Markt und Technik der MSX-Computer gegeben, die

den etablierten Heimcomputer-Anbietern wie Atari oder Commodore schnell Marktanteile abnehmen können. Ist das Argument - MSX-Computer besitzen eine veraltete Technik - richtig oder falsch?

Soviel läßt sich immerhin schon sagen: Vor allem die Kombinationen mit anderen elektronischen Geräten bringen völlig neue und einfach zu bedienende Computeranwendungen, Möglichkeiten, die von den etablierten Anbietern bislang übersehen wurden, wohl weil sie davon ausgingen, daß ihre Geräte überwiegend von technikbegeisterten jugendlichen Freaks gekauft werden.

Hier noch einige weitere Kapitel aus dem Inhalt dieses Titels:



Spielen, lernen, arbeiten mit dem ATARI



G.-P. Raabe/K.-J. Schmidt

- Markt und Technik eines neuen Standards
- Firmen und Geräte: Hintergrundinformationen und Marktübersichten
- Peripherie und Zusatzgeräte
- Programmieren mit MSX-BASIC.

Der Riteman F+-Matrixdrucker – vorgestellt auf der Hannover-Messe

Der Riteman F+-Matrixdrucker besitzt eine kompakte Bauweise.

Mit seiner geringen Stellfläche und der gut durchdachten Papierablage unter dem Drucker bringt der Riteman F+ seine Leistung dort, wo Sie sie brauchen – an Ihrem Arbeitsplatz. Ein Personal-Printer – kompatibel zu Epson FX-80 – der Ihnen zeigt, wie einfach drucken ist: in Schönschrift, mit 9 internationalen Zeichensätzen und Puffer.

Technische Daten:

Druckbreite
80 Stellen bei 10 cpi

Druckgeschwindigkeit
105 Zeichen/s bei 10 Zeichen/Zoll

RAM-Speicher
Standard: 2 KB, Option:
8 KB

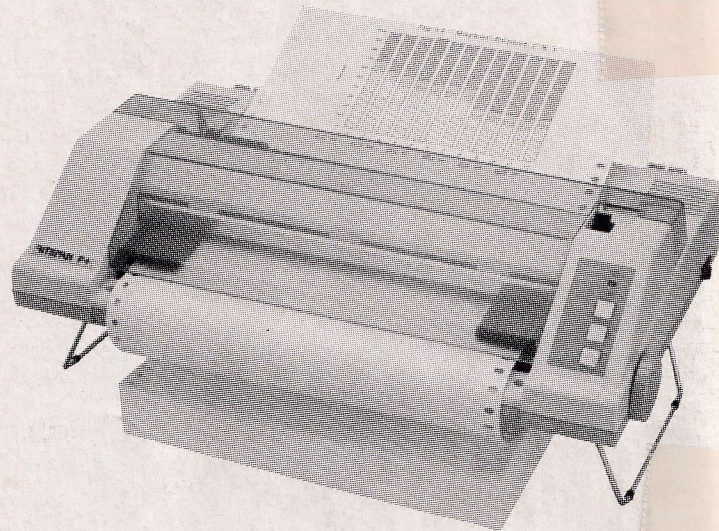
Schnittstellen
Standard: 8-bit Parallel
Centronics; Opt: RS-232
C

Abmessungen
395 x 270 x 105
(B x T x H) mm

Druckkopf
9 Nadeln

Matrix
9 x 9

Schriftarten
Pica, Elite, komprim.,
Proportional, Breitschrift-
kombinationen, Super-/
Subscript, Unterstreichen/
Unterlängen



Grafik
Einzelnadel-Ansteuerung
Druckarten
Standard, Fettdruck, Doppeldruck, Schönschrift

Zeichensatz
96 ASCII-, 96 Italic-Zeichen, 9 internat. Zeichensätze, 32 Grafik-Symbole, freiladbarer Zeichensatz

Siemens: MS-DOS für SMP-SYS900

Das Betriebssystem MS-DOS erfreut sich weltweit einer starken Nachfrage. Die Vielzahl der inzwischen verfügbaren Anwendungsprogramme, die ständig in allen wichtigen Bereichen erweitert werden, machen den Einsatz verlockend. Ab Mitte 1985 wird Siemens deshalb sein neues Mikrocomputersystem für OEM (SMP-SYS900) auch mit diesem Betriebssystem bisher nur CP/M 85 und 86) anbieten.

Das SMP-SYS900 ist das Grundmodell einer neuen Generation von Komponenten, mit denen mikrocompu-

tergesteuerte Applikationen schnell und effektiv verwirklicht werden können. Gegenüber dem modularen Mikrocomputersystem SMP (Europaformat) bietet das SYS900 vorbereitete Software-Komponenten, gegenüber dem tragbaren Entwicklungssystem PMS ist die Grundausstattung sparsam auf den kleinsten gemeinsamen Nenner für alle Anwendungsfälle gebracht: OEM-Zubehör wie Bildschirm und Tastatur, Programmierereinrichtung, Emulator, Winchester-Laufwerk, Drucker oder SMP-Baugruppen kann der An-

wender individuell nach Maß anschließen. Einsatzbereiche: Steuereinheiten, Test- und Prüfgeräte, Entwicklungssysteme für Hard- und Software.

Das nun zusätzlich angebotene Betriebssystem MS-DOS (V2.11) ist sehr benutzerfreundlich. Zu leicht verständlichen Systemantworten und Fehlermeldungen kommt die Schnelligkeit, die vor allem auf einer effizienten Disketteneinteilung beruht und einen wahlfreien Zugriff erlaubt. Bedeutend für viele Anwender dürfte auch die Kompatibilität zu

CP/M und der Einsatz von UNIX/XENIX-ähnlichen Strukturen im MS-DOS sein. Dazu gehören die hierarchische Struktur des Inhaltsverzeichnisses, die Unterstützung elementarer XENIX-Dateiprozeduren und Systemaufrufe. Bisher ausgelieferte SMP-SYS900 können mit dem Systembausatz SMP-SYBI88-A1 das neue Betriebssystem ohne weiteres einsetzen. Der Bausatz enthält in der Version MS-DOS V2.11 als Software Makroassembler, Linker und Locater, Bibliotheksverwalter, Texteditor und Software-Debugger.

magna

HOME-COMPUTER CASSETTEN + DISKETTEN

CASSETTEN

Präzise Cassetten-Mechanik
Hohe Speicherdichte
Für alle Data-Recorder

DATA-DISK DISKETTEN

Extreme Lebensdauer durch
zusätzliche Oxygenbeschichtung
Zuverlässige Datensicherheit
durch mehr als 70 chemische,
magnetische und elektrische
Qualitäts-Tests



TONTRÄGER

magna tonträger vertriebs gmbh

Bunzlauer Straße 3 · Postfach 400340 · 5000 Köln 40
Telefon (02234) 74054 · Telex 889975

DOPPELT GEWINNEN MIT "ELEPHANT"!



Jetzt gibt es neben dem 1. Gewinn, der überragenden Qualität der "ELEPHANT"-Floppy-Disk, noch einen triftigen Grund mehr, schnell Ihren "ELEPHANT"-Händler aufzusuchen: ein Puzzle-Spiel, das Ihnen einen schönen Extra-Gewinn einbringen kann. Was es zu gewinnen gibt, steht auf dem fertigen Puzzle! Also nichts wie hin zu Ihrem "ELEPHANT"-Spezialisten.

Dennison

ELEPHANT™ NEVER FORGETS.

MARCOM Computerzubehör GmbH

Podbielskistraße 321, 3000 Hannover 51, Tel. (05 11) 64 74 20

Frankreich: Soroclass, 8, Rue Montgolfier - 93115, Rosny-Sous-Bois, Tel.: 16 (1) 855-73-70

Großbritannien: Dennison Mfg. Co. Ltd., Colonial Way, Watford WD2 4JY, Tel.: 0923 41244, Telex: 923321

Weiteres Ausland: Dennison International Company, 4006 Erkrath 1, Matthias-Claudius-Straße 9, Telex: 858 6600