

monatlich

Compute mit

COMMODORE & SCHNEIDER

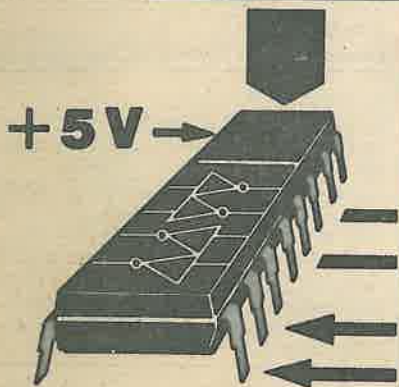
7/86

VC-64, VC-20, C-16/116, CPC-464

3,80 DM
33 öS
3,80 sFr

Unabhängiges Magazin für Anwender von Commodore- und Schneider-Computern

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



64 KB

für C-16/116
im Selbstbau

Seite 38

Commodore 64: Basic Extension

bindet Ihre Assembleroutinen mit einem Befehlsword in das Basic ein. Schicken Sie uns Ihre Assembleroutinen, für eine „Compute mit“-Bibliothek, damit sich unsere Leser ihr ganz persönliches Basic schaffen können!

Seite 22

Der neue Commodore 64 im Test!

Commodore stellte im Mai den neuen C-64 vor.

- Was ist eigentlich neu, außer dem Gehäuse?

Dem neuen C-64 legt Commodore ein neues Betriebssystem bei.

- Wie leistungsfähig kann ein Betriebssystem mit Fenster-technik auf dem C-64 sein?

Seite 3



Kleinanzeigen - Report - Software-Abo - Software-Service
Bücher - Werkstatt - Assembler-Kurs - Software-Reviews - Tips & Tricks
Schneider-Disketten-Archiv - Roadrunner - Break-Out - u. v. m.

monatlich

Compute mit

COMMODORE & SCHNEIDER

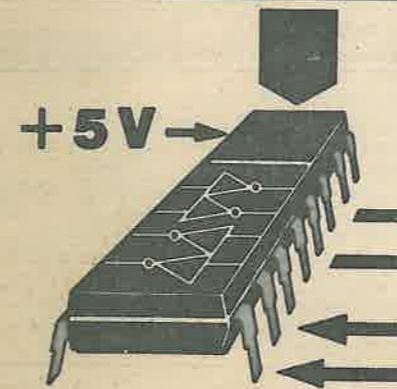
7/86

Unabhängiges Magazin für Anwender von Commodore- und Schneider-Computern

VC-64, VC-20, C-16/116, CPC-464

3,80 DM
33 öS
3,80 sFr

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



64 KB

für C-16/116 im Selbstbau

Seite 38

Commodore 64: Basic Extension

bindet Ihre Assembleroutinen mit einem Befehlswort in das Basic ein. Schicken Sie uns Ihre Assembleroutinen, für eine „Compute mit“-Bibliothek, damit sich unsere Leser ihr ganz persönliches Basic schaffen können!

Seite 22

Der neue Commodore 64 im Test!

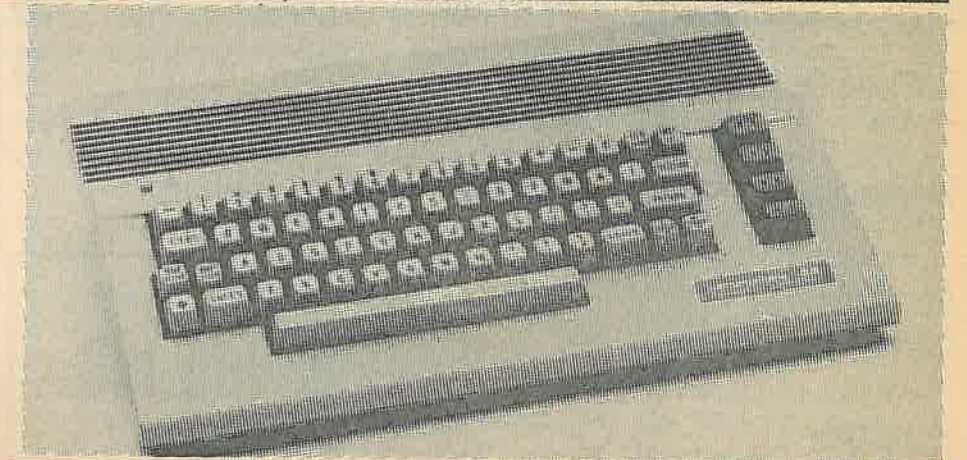
Commodore stellte im Mai den neuen C-64 vor.

- Was ist eigentlich neu, außer dem Gehäuse?

Dem neuen C-64 legt Commodore ein neues Betriebssystem bei.

- Wie leistungsfähig kann ein Betriebssystem mit Fenster-technik auf dem C-64 sein?

Seite 3



die neue Generation

der Computer-Zeitschriften

aktueller software markt

Nr. 5 Juni/Juli 1986 öS 50 sFr 6,- DM 6,-

Das aktuelle Software-Geschehen auf 100 Seiten! Was man wissen sollte.

★ Was hat GARFIELD, der berühmteste Kater der Welt, in der ASM zu suchen? Lesen Sie Seite 46

★ „LUCIFER'S REALM“ – das verbotene Spiel. ASM stellt es vor! Seite 88

★ „Volltreffer“ – Die High-Score-Liste Seite 23

Quiz:

115 Top-Games zu gewinnen!

++ Action-Games ++ Anwender ++ Adventures ++ Sound- und Lernprogramme ++
HOTLINE ... die Software-Hitparade

MASTERTRONIC
Preisrätsel:
50 x 10 Kassetten
sind zu gewinnen!



... die interessante Seite mit Tips + Lösungen

sie ist da...



- 100 Seiten Programme im Test
- der Software-Markt auf einen Blick
- speziell gesuchte Software kann direkt beim Hersteller bestellt werden
- großer Kleinanzeigenmarkt
- für alle Anwender von Heimcomputern

Tronic-Verlag GmbH · Am Stad 35 · 3440 Eschwege · Telefon: 0 56 51 / 3 00 11

Kleinanzeigen - Report - Software-Abo - Software-Service
Bücher - Werkstatt - Assembler-Kurs - Software-Reviews - Tips & Tricks
Schneider-Disketten-Archiv - Roadrunner - Break-Out - u. v. m.

Report

Der neue C-64 3

Bericht: Jugend forscht '86 4

Computer Camp 4

Was gibt es Neues auf dem Softwaremarkt 20

Bücher

Z-80 Maschinensprache 54

DFÜ für C-64 & 128 54

Der Computerdieb 55

Schneider-Floppy-Buch 55

Tips & Tricks

Scrolling für VC-20 10

INPUT und READ ERROR (C-64) 26

UPDATE für BEEP-Routine (C-16/116) 37

Software-Service

Kleinanzeigen 59

Assemblerkurs Teil 14 44

Werkstatt:

VC-20 Sound-Demo 12

Speichererweiterung für C-16/116 38

Diskettenarchiv für CPC 464, 664 und 6128 51

Software

Commodore

Mad Rush (VC-20) 5

Ghost-Hell (VC-20) 8

Roadrunner (C-64) 13

Basic Extension (C-64) 22

Meteoritensturm (C-64) 26

Checksummer OC für C-16/116 31

Airwolf (C-16/116) 34

Break-Out (C-16/116) 41

Schneider

Poker 46

Cave-Walker 57

Korrekturen

Klartext für C-16/116 30

Fehlerteufel 59

Die nächste Ausgabe von **Compute mit** liegt wieder ab **16. Juli** bei Ihrem Zeitschriftenhändler für Sie bereit.

Impressum

„Compute mit“ erscheint monatlich im Tronic-Verlag, 3440 Eschwege

Redaktion:

Axel Credé (verantwortlich)
 Chefredakteur: Hartmut Wendt
 Uwe Knierim, Manfred Kleimann, Bernd Zimmermann
 Frank Brall, Ottfried Schmidt, Siegfried Görk

Gesamtherstellung:

Druckhaus Dierichs Kassel, Frankfurter Str. 168, 3500 Kassel

Vertrieb:

Inland (Groß-, Einzel- und Bahnhofsbuchhandel) sowie Österreich und Schweiz
 Verlagsunion
 Friedrich-Bergius-Straße 20
 6200 Wiesbaden
 Telefon 0 61 21 / 26 60

Anfragen nicht an Vertrieb oder Druckerei, sondern nur an den Verlag!

Erscheinungsweise:

Erstverkaufstag von „Compute mit“ jeweils Mitte des Monats.

Urheberrecht:

Alle in „Compute mit“ veröffentlichten Beiträge sind urheberrechtlich geschützt. Alle Rechte, auch Übersetzungen, vorbehalten.

Reproduktionen jeder Art (Fotokopien, Microfilm, Erfassung in Datenverarbeitungsanlagen usw.) bedürfen der schriftlichen Genehmigung des Verlags. Alle veröffentlichte Software wurde von Mitarbeitern des Verlages oder von freien Mitarbeitern erstellt.

Aus Ihrer Veröffentlichung kann nicht geschlossen werden, daß die beschriebenen Lösungen oder Bezeichnungen frei von Schutzrechten sind.

Bezugspreis:

Einzelheft 3,80 DM
 Abonnement: Inland 42,- DM im Jahr (12 Ausgaben)
 Ausland: Europa 52,- DM
 ohne Kassettent!

Autoren, Manuskripte:

Der Verlag nimmt Manuskripte und Software zur Veröffentlichung gerne entgegen.

Sollte keine andere Vereinbarung getroffen sein, so gehen wir davon aus, daß Sie mit einem Honorar von 120,-DM pro abgedruckter Seite im Heft einverstanden sind.

Bei Zusendung von Manuskripten und Software erteilt der Autor dem Verlag die Genehmigung zum Abdruck und Versand der veröffentlichten Programme auf Datenträger.

Rücksendung erfolgt nur gegen Erstattung der Kosten. Zusendungen von Software zur Veröffentlichung sollten folgendes enthalten:

Kopierfähige Kassette oder Diskette mit dem Programm (Computer-Bezeichnung), von Drucker erstelltes Listing oder Serie von Bildschirmfotos (keine Schreibmaschinenlistings), evtl. Bildschirmfotos von einem Probelauf und ausführliche Programmbeschreibung (Erklärung der Variablenliste, Beschreibung des Bildschirmaufbaues, Farbe, Grafik usw.). Für eingesandte Programmunterlagen kann keinerlei Haftung übernommen werden.

Anzeigenpreise:

Bitte Mediaunterlagen anfordern.

Anzeigenverwaltung:

Tronic-Verlag GmbH,
 Am Stad 35, 3440 Eschwege
 Telefon 0 56 51 / 3 00 11
 Telefax 0 56 51 / 3 00 14

Anzeigenleitung:

Inland: H. Wendt
 Ausland: M. Kleimann

COMMODORE-NEUHEITEN

Endlich ist das Geheimnis gelüftet:
 Der neue C-64 ist da!



Überraschend hatte Commodore eine Veränderung des C-64 angekündigt. Neugierig haben wir uns gefragt, worin nun die eigentlichen Veränderungen liegen. Der erste Eindruck: der C-64 ist schlanker geworden. Das neue Design erinnert etwas an den PC 128, jedoch ohne 10'er-Tastatur. Das alte Gehäuse war ja häufiger Anlaß für Spott. Butterbrotschachtel hieß es, oder man sprach liebevoll von dem Käfereffekt. Nun sind die Zeiten vorbei: der C-64 hat sich das gängige „Slim-line“-Design der heutigen Computergeneration zu eigen gemacht. Allerdings: ganz so ergonomisch wie der PC 128 ist der neue C-64 nicht. Die Handballen müssen sich noch immer etwas krümmen, es fehlt vorne etwas Länge und die Tastatur ist immer noch etwas schräg, kleine Reminiscenzen an den alten C-64! Gibt es aber technische Neuerungen, ist die nächste Frage. Schraubt man das Gehäuse

auf, wird man von einer wunderschönen Originalplatine angelacht, wie wir sie bisher von allen unseren Oldtimermodellen her gewöhnt sind. Lediglich der VIC-Chip und der I/O-6526 tragen neue Seriennummern, ohne jedoch auf eine entschiedene Veränderung schließen zu lassen. Die einzige wirkliche Veränderung besteht in der wesentlich verbesserten Abschirmung der Platine. Weitere Veränderungen wurden nicht vorgenommen, um die Kompatilität zu den alten Geräten 100 %ig zu gewährleisten. Tatsächlich aber vermischen wir kleine Verbesserungen in Bereichen, die die Kompatilität in keiner Weise gefährdet hätten. So muß sich der User nach wie vor einen Reset-taster selbst stricken. Auch eine Betriebssystemumschaltplatine hätte integriert sein können. Der obligatorische Eigenbau bleibt auch hier nicht erspart, was jedoch durch das neue Gehäusekonzept

erheblich erschwert wird. Der Raum ist so eng, daß für eine Umschaltplatine zumindest ein Teil des Kühlblechs herausgesägt werden muß. Eine weitere wichtige Neuerung für den C-64-Käufer besteht in einer besonderen Beigabe. Commodore hat nun endgültig die Rechte an einem neuen Betriebssystem erworben, das erstmalig auf der letztjährigen CES-Messe in Las Vegas vorgestellt wurde, und legt dieses jedem neuen Gerät als Zugabe auf Diskette bei. Dieses neue Betriebssystem, GEOS genannt, hat 2 entscheidende Merkmale. Es bietet dem Benutzer eine grafische Benutzeroberfläche ähnlich dem AMIGA: Pull-Down-Menü mit Piktogrammen und Softwareauswahl per Joystick und Maus. Zweiter Vorteil von GEOS ist ein implementierter Floppybeschleuniger, der ungefähr 10-fache Geschwindigkeit erreicht. Der ist allerdings auch unbedingt erforderlich, da das Betriebssystem in Modulen vorliegt, die immer wieder von der Betriebssystemdiskette nachgeladen werden müssen. Tatsächlich bekommt der C-64 mit GEOS etwas von den Vorteilen der 16-Bitter eingehaucht und wird auch in einigen Funktionen erstaunlich schnell (vor allem wenn es darum geht, Grafik nachzuladen). Einige Nachteile sollen nicht verschwiegen werden: innerhalb von GEOS ist kein Basic-Editor integriert. Man muß GEOS verlassen, wenn man ins Basic gelangen will, kann jedoch durch einen einfachen Druck auf die RESTORE-Taste wieder in das GEOS gelangen. Ferner ist bekannt, daß die ersten Geräte nur mit der amerikanischen Version von GEOS ausgerüstet werden. Die deutsche Version ist noch in Arbeit. Insgesamt jedoch ist GEOS eine fantastische Erweiterung, und es kommt jetzt darauf an, daß genügend Pro-

grammierer sich damit beschäftigen – ein GEOS-Basic wäre dann auch denkbar. Auf der GEOS-Diskette befinden sich schon 2 GEOS-Programme: GEOWrite und GEOPaint, über die wir das nächste Mal berichten werden.

Als weitere Neuheiten kündigt Commodore sowohl für den C-64 als auch für den PC 128 Speichererweiterungen an. Der C-64 soll auf 512-KB und der PC 128 auf 640 KB aufgerüstet werden können.

JUGEND FORSCHT: Sieger ermittelt



Pascal Merle vor seiner Einrichtung zur Programmierung eines Zielprozessors (bester JUGEND-FORSCHT-Teilnehmer im Bereich Informatik) Foto: Ford-Werke AG

Vom 8. bis 14. Mai fand der 21. Bundeswettbewerb JUGEND FORSCHT in Saarlouis statt. Am Montag wurden bereits die Preisträger bekanntgegeben. Auffallend bei der Preis-

vergabe war die Vielzahl der Arbeiten, die sich des Computers als wissenschaftliches Hilfsmittel bedienen. Das beste Beispiel hierfür ist wohl die Arbeit des Bun-

desiegers für Biologie, Thomas Dick (17). In einer Computersimulation entwickelte er den Aufbau eines Enzyms, über dessen Molekularstruktur noch wenig bekannt ist. Für diese Arbeit, die bereits internationale wissenschaftliche Anerkennung gefunden hat, erhielt Thomas außerdem den Sonderpreis des Bundespräsidenten. Auch bei vielen anderen Arbeiten im Bereich Physik und Chemie spielten Computer eine herausragende Rolle. Um so erstaunlicher war es, daß im Bereich Mathematik/Informatik kein erster Preis vergeben wurde. Den 2. Preis erhielt der erst 14jährige Pascal Merle für ein Computerprogramm in der höheren Programmiersprache „C“, das ein anderes C-Programm in Maschinensprache übersetzt, die ein bestimmter Zielprozessor (TMS 99xxx) versteht.

Viele Arbeiten beschäftigten sich auch mit unserer natürlichen Umwelt und deren Schädigung. Dies veranlaßte Ford-Generaldirektor Daniel Goeudevert zu der Bemerkung: „Ich verstehe auch, daß viele junge Leute vom Aussterben ganzer Tiergattungen zum Beispiel mehr beunruhigt sind, als sie vom technologischen Fortschritt begeistert sind. Diesen Wertewandel müssen wir in unserer Produktentwicklung berücksichtigen. Denn technischer Fortschritt ist kein Selbstzweck, sondern dient stets den menschlichen Bedürfnissen und den gesellschaftlichen Notwendigkeiten.“

Wir schließen uns an dieser Stelle den zahlreichen Gratulanten an und wünschen den Siegern alles Gute für Ihre Zukunft.

Sommerferien im Computercamp

Für alle, die ihre Sommerferien mit einem Computerkurs verknüpfen wollen, bietet COMPUCAMP wieder Ferien im Computercamp an.

Das Angebot richtet sich an Jugendliche und auch an Kinder. Der Akzent dieses Angebots liegt dabei nicht einseitig auf „Lernen“, auch die Freizeit soll dabei nicht zu kurz kommen. Nicht Bildschirmfanatiker

sind also gefragt, sondern alle, die auch Spaß an Windsurfen, Wasserski und Tennis haben. Zum Programm gehören auch Ausflüge zu Firmen wie COMMODORE und Fischertechnik.

Die Computerkurse sind weit gestaffelt – von LOGO, über Basic bis zur Maschinensprache. Für jeden wird also etwas passendes angeboten. Camp finden statt in



Schloß Dankern (Emsland), Tönning (Nordsee) und Veltishof am Titisee im Schwarzwald.

CompuCamp GmbH, Goflerstr. 21, 2000 Hamburg 55, Tel.: 0 40/861255

Schnelles Radrennen für den VC 20

MAD RUSH

Dies ist ein wirklich schnelles Spiel für den VC 20

In bester Formel 1 - Art muß ein Radfahrer durch einen Straßenkurs gesteuert werden, ohne gegen die auftauchenden Hindernisse zu stoßen: Bäume, Ölfässer, Kakteen, Palmen und Sträucher. Außerdem

kann man auf Ölpfützen zur Seite wegrutschen. Das Spiel wird laufend schneller – wer 30 Meilen schafft ist schon ein Rennprofi!

Auf geht's!

```

0 REM*****
1 REM* MAD RUSH *
2 REM* *
3 REM* 1986 BY *
4 REM* HEINZ KROOS *
5 REM* NEUE REIHE 26 *
6 REM* 4795 BOKE *
7 REM*****
9 PRINTCHR$(14) "CLEAR SPACES" AD-USH":
PRINT "SPACE" "DOWN3 SPA
CE4-ATAS SPACE READING"
10 FORA=7000TO7679:READB:POKEA,B:C=C+B:W
EXT:PRINTCHR$(142) "CLEAR"
11 IFC<>61957THENPRINT"DATA SPACE ERROR"
:END
12 POKE198,1:POKE631,131
100 DATA162,0,189,0,31,157,22,31
101 DATA189,0,151,157,22,151,202,208
102 DATA241,162,0,189,0,30,157,22
103 DATA30,189,0,150,157,22,150,202
104 DATA208,241,189,255,29,141,44,30
105 DATA189,252,29,141,44,151,189,253
106 DATA29,141,44,150,189,254,29,141
107 DATA44,31,76,149,26,2,0,0
108 DATA1,0,32,1,216,0,15,0
109 DATA0,0,0,0,0,0,0,0
110 DATA0,0,0,0,0,0,0,0
111 DATA0,0,0,0,0,0,0,0
112 DATA0,0,0,0,96,96,96,96
113 DATA96,96,96,96,96,96,96,96
114 DATA96,96,96,96,96,96,96,96
115 DATA98,97,96,96,96,96,96,96
116 DATA96,96,96,96,96,96,96,96
117 DATA96,96,96,96,96,96,96,96
118 DATA96,96,96,96,96,96,4,96
119 DATA98,97,96,96,96,96,96,96
120 DATA96,96,96,96,96,96,96,140
121 DATA16,16,16,112,156,178,178,186
122 DATA254,124,56,24,24,16,16,16
123 DATA8,8,14,57,77,93
124 DATA127,62,28,24,24,8,8,8
125 DATA12,26,58,170,234,202,108,120
126 DATA120,58,29,13,13,13,5,2
127 DATA227,211,203,199,203,211,227,227
128 DATA255,255,255,255,255,255,255
    
```

```

129 DATA24,60,126,254,254,127,127,14
130 DATA60,126,255,255,254,124,124,48
131 DATA32,88,92,85,87,83,54,30
132 DATA30,92,184,176,176,176,160,64
133 DATA26,79,181,174,87,189,82,47
134 DATA224,80,168,104,164,216,116,168
135 DATA13,7,7,7,15,31,41
136 DATA176,224,192,192,192,224,240,88
137 DATA60,60,60,189,255,255,255,255
138 DATA189,60,60,60,60,60,60,126
139 DATA7,24,32,56,63,63,59,53
140 DATA224,24,4,28,252,252,220,92
141 DATA53,53,53,59,63,63,31,7
142 DATA92,92,92,76,124,252,248,224
143 DATA88,120,126,30,26,24,24,24
144 DATA48,56,248,248,251,255,191,63
145 DATA63,56,56,56,56,56,56,56
146 DATA0,0,0,0,0,0,48,127
147 DATA99,159,254,195,127,124,7,0
148 DATA240,159,255,49,255,135,255,120
149 DATA0,0,240,255,143,248,207,252
150 DATA0,0,0,0,240,255,248,143
151 DATA0,0,0,0,0,0,240,158
152 DATA7,0,0,0,0,0,0,0
153 DATA0,0,0,0,0,0,0,0
154 DATA240,127,7,0,0,0,0,0
155 DATA0,0,0,0,0,0,0,0
156 DATA223,253,199,127,7,0,0,0
157 DATA253,29,249,202,250,124,0,0
158 DATA1,7,5,26,111,90,245,174
159 DATA192,120,180,236,84,234,85,174
160 DATA173,82,61,86,93,181,106,13
161 DATA90,244,90,173,234,168,80,160
162 DATA15,15,7,7,7,7,7,7
163 DATA224,224,192,192,192,192,192,192
164 DATA7,7,15,15,31,63,82,0
165 DATA192,192,224,224,240,248,150,0
166 DATA53,19,73,54,75,37,31,1
167 DATA180,168,114,212,168,118,216,224
168 DATA0,0,0,2,10,21,19,75
169 DATA0,0,0,64,208,80,104,240
170 DATA16,8,6,114,31,15,7,15
171 DATA0,12,48,96,108,216,240,224
172 DATA25,48,32,64,0,0,0,0
173 DATA128,192,96,96,48,48,48,24
    
```

Zum Programmaufbau:

Das Programm besteht aus 5 Vorprogrammen und einem Hauptprogramm. Nachdem alle Vorprogramme einzeln abgetippt und abgespeichert wurden, gibt es die Möglichkeit, sie zu einem einzigen Vorprogramm zusammenzufassen:

1. Man lädt alle Vorprogramme einzeln und startet auch jedes für sich.
2. Dann gibt man ein: POKE 45,230: POKE 46,29: CLR
3. Man speichert es nun als ein Vorprogramm ab.

Beim Abtippen ist darauf zu achten, daß keine Veränderungen am Programm vorgenommen werden, da das Programm sonst aussteigen kann.

programme

```

174 DATA24,24,24,24,24,24,60,126 <72>
175 DATA1,3,1,3,7,1,3,7 <158>
176 DATA0,128,0,128,192,0,128,192 <245>
177 DATA15,3,7,15,31,7,15,31 <254>
178 DATA224,128,192,224,240,192,224,240 <173>
179 DATA63,15,31,63,127,3,3,3 <25>

```

```

0 REM***** <115>
1 REM* * * <98>
2 REM* MAD RUSH * <39>
3 REM* * * <100>
4 REM* TEIL 2 * <218>
5 REM* * * <102>
6 REM***** <121>
9 PRINTCHR$(14)"(CLEAR SPACE7)\AD-USH":
PRINT"(SPACE7)":PRINT"(DOWN3 SPA
CE4)-ATAS(SPACE)READING" <219>
10 FORA=6600T06999:READB:POKEA,B:C=C+B:N
EXT <29>
11 PRINT"(CLEAR)"CHR$(142):IFC<>43189THE
NPRINT"DATA(SPACE)ERROR":END <84>
12 POKE198,1:POKE631,131 <21>
100 DATA 169,127,141,34,145,189,32,145 <60>
101 DATA 75,128,201,64,240,3,76,89 <107>
102 DATA 26,76,169,26,96,169,6,141 <153>
103 DATA 4,30,169,6,141,14,30,169 <153>
104 DATA 0,141,4,150,169,0,141,14 <176>
105 DATA 150,173,49,24,141,224,25,173 <200>
106 DATA 49,24,105,9,141,229,25,255 <54>
107 DATA 248,25,255,242,25,173,49,24 <192>
108 DATA 141,234,25,173,49,24,105,9 <14>
109 DATA 141,239,25,255,6,26,255,12 <212>
110 DATA 26,76,238,26,96,169,0,141 <0>
111 DATA 28,27,96,96,96,96,96,96 <125>
112 DATA 96,96,96,96,169,4,141,150 <16>
113 DATA 27,255,151,27,173,151,27,41 <21>
114 DATA 2,240,23,206,155,26,206,185 <152>
115 DATA 26,206,193,26,206,203,26,206 <134>
116 DATA 198,26,169,0,141,151,27,206 <0>
117 DATA 152,26,76,181,26,96,96,96 <194>

```

```

0 REM***** <15>
1 REM@ @ @ <208>
2 REM@ MAD-RUSH @ <191>
3 REM@ @ @ <210>
4 REM@ TEIL 3 @ <189>
5 REM@ @ @ <212>
6 REM***** <21>
9 PRINTCHR$(14)"(CLEAR SPACE6)\AD-USH":
PRINT"(SPACE6)":PRINT"(DOWN3 SPA
CE4)-ATAS(SPACE)READING" <61>
10 FORA=6300T06599:READB:POKEA,B:C=C+B:N
EXT <230>
11 PRINT"(CLEAR)"CHR$(142):IFC<>21418THE
NPRINT"DATA(SPACE)ERROR":END <175>
12 POKE198,1:POKE631,131 <21>
100 DATA 6,6,6,5,5,5,5,4 <159>
101 DATA 4,4,4,4,4,5,5,5 <134>
102 DATA 6,6,6,6,6,6,6,6 <237>
103 DATA 7,7,8,8,8,8,8,8 <120>
104 DATA 9,9,9,8,8,8,8,7 <123>
105 DATA 7,7,7,6,5,5,5,5 <200>
106 DATA 5,5,5,5,5,5,5,5 <169>
107 DATA 5,5,5,5,5,5,5,5 <170>
108 DATA 5,5,5,5,5,5,5,5 <171>
109 DATA 5,5,5,5,5,5,5,5 <172>
110 DATA 5,5,5,5,5,5,5,5 <173>
111 DATA 5,5,5,5,5,5,5,5 <174>

```

```

180 DATA248,224,240,248,252,192,192,192 <39>
181 DATA00,0,0,0,0,0,0,0 <160>
182 DATA0,0,0,0,198,229,238,101 <223>
183 DATA0,255,0,255,32,53,49,32,53,14,32
,53,49,32,53,14 <14>

```

```

118 DATA 96,169,10,141,150,27,255,152 <188>
119 DATA 27,173,152,27,41,2,240,28 <43>
120 DATA 255,155,26,255,185,26,255,193 <12>
121 DATA 26,255,203,26,255,198,26,169 <144>
122 DATA 0,141,152,27,255,152,26,169 <17>
123 DATA 254,141,191,26,76,181,26,169 <33>
124 DATA 0,141,147,27,169,25,141,148 <232>
125 DATA 27,76,166,25,96,169,32,141 <134>
126 DATA 239,31,189,217,31,201,32,240 <231>
127 DATA 3,76,100,25,76,200,25,96 <211>
128 DATA 96,189,17,145,75,16,201,8 <220>
129 DATA 240,3,76,44,26,173,150,27 <189>
130 DATA 141,217,31,173,150,27,233,255 <190>
131 DATA 141,239,31,169,0,141,217,151 <115>
132 DATA 169,0,141,239,151,255,150,27 <244>
133 DATA 255,150,27,189,150,27,41,4 <197>
134 DATA 240,5,169,0,141,150,27,169 <55>
135 DATA 255,141,34,145,169,255,141,191 <86>
136 DATA 26,76,0,25,96,96,173,26 <194>
137 DATA 27,201,32,240,7,173,0,30 <112>
138 DATA 201,6,240,3,76,7,27,169 <146>
139 DATA 32,141,26,27,76,30,27,173 <243>
140 DATA 0,30,201,6,240,3,76,25 <205>
141 DATA 27,169,7,141,26,27,76,30 <222>
142 DATA 27,169,7,141,0,30,255,246 <163>
143 DATA 26,255,8,27,255,28,27,173 <149>
144 DATA 246,26,201,22,240,3,76,238 <240>
145 DATA 26,169,0,141,246,26,169,0 <196>
146 DATA 141,8,27,76,29,26,96,96 <157>
147 DATA 189,22,150,141,253,29,189,22 <130>
148 DATA 151,141,252,29,189,22,30,141 <142>
149 DATA 255,29,189,22,31,141,254,29 <185>

```

```

112 DATA 5,5,5,5,169,32,141,0 <191>
113 DATA 30,169,5,141,0,150,255,3 <216>
114 DATA 25,255,8,25,173,3,25,201 <238>
115 DATA 22,240,3,76,0,25,169,0 <183>
116 DATA 141,3,25,169,0,141,8,25 <114>
117 DATA 76,221,25,173,77,23,141,11 <42>
118 DATA 144,173,77,23,141,12,144,206 <80>
119 DATA 158,27,173,158,27,201,0,240 <97>
120 DATA 3,76,80,25,206,46,25,169 <95>
121 DATA 15,141,158,27,169,17,141,14 <98>
122 DATA 144,206,40,25,206,14,144,76 <83>
123 DATA 191,234,96,96,96,96,96,96 <171>
124 DATA 96,96,96,96,96,96,96,96 <175>
125 DATA 173,155,26,141,107,25,173,221 <49>
126 DATA 31,141,154,27,173,154,27,201 <137>
127 DATA 8,240,3,76,137,25,255,155 <60>
128 DATA 26,255,185,26,255,193,26,255 <17>
129 DATA 198,26,76,153,22,173,154,27 <194>
130 DATA 201,9,240,3,96,96,96,206 <186>
131 DATA 155,26,206,185,26,206,193,26 <140>
132 DATA 206,198,26,76,146,22,96,96 <174>
133 DATA 96,96,169,0,201,22,240,9 <46>
134 DATA 255,167,25,32,64,27,76,166 <7>
135 DATA 25,169,0,141,167,25,169,149 <151>
136 DATA 141,147,27,169,26,141,148,27 <191>
137 DATA 96,96,96,96 <198>

```

programme

```

0 REM* TEIL4 * <228>
9 PRINTCHR$(14)"(CLEAR SPACE6)\AD-USH":
PRINT"(SPACE6)":PRINT"(DOWN3 SPA
CE4)-ATAS(SPACE)READING" <61>
10 FORA=5765T05972:READB:POKEA,B:C=C+B:N
EXT <242>
11 FORA=6130T06299:READB:POKEA,B:C=C+B:N
EXT <209>
12 PRINT"(CLEAR)"CHR$(142):IFC<>36910THE
NPRINT"DATA(SPACE)ERROR":END <207>
13 POKE198,1:POKE631,131 <22>
100 DATA173,224,25,141,238,27,255,255 <101>
101 DATA27,76,64,27,96,206,203,26 <215>
102 DATA206,152,26,96,255,203,26,255 <91>
103 DATA152,26,96,96,169,0,141,14 <157>
104 DATA144,169,191,141,20,3,169,234 <221>
105 DATA141,21,3,96,169,39,141,20 <130>
106 DATA3,169,25,141,21,3,169,255 <231>
107 DATA141,40,25,160,255,141,46,25 <255>
108 DATA96,173,155,26,105,22,141,152 <28>
109 DATA26,76,135,26,96,173,224,25 <91>
110 DATA105,210,141,156,27,173,156,27 <43>
111 DATA141,155,26,173,156,27,141,185 <88>
112 DATA26,173,156,27,141,198,26,173 <221>
113 DATA156,27,105,22,141,193,26,173 <245>
114 DATA156,27,105,22,141,203,26,76 <12>
115 DATA198,22,96,227,223,227,223,227 <22>
116 DATA230,231,234,236,238,239,230,230 <147>
117 DATA234,230,234,231,231,230,230,227 <169>
118 DATA227,223,223,0,0,223,234,239 <195>

```

Teil 5:

```

0 POKE36869,255:POKE36879,24:PRINT"(CLEA
R)": <32>
9 PRINT"(GREEN)12" <202>
10 PRINT"3(BLUE)4(SPACE5 GREEN)LM(BLUE
11 PRINT"(SPACE)5(SPACE)G(SPACE)G(SPACE)
NO(SPACE2)G(SPACE3)GGG" <170>
12 PRINT"(SPACE2)G(SPACE)G(SPACE)G(SPACE
3)G(SPACE)G(SPACE2)G(SPACE)HG" <221>
13 PRINT"(SPACE2)G(SPACE3)G(SPACE3)GGG(S
PACE2)G(SPACE2)G(SPACE)GG" <252>
14 PRINT"(SPACE)G(SPACE3)H(SPACE)G(SPACE
)G(SPACE3)G(SPACE)G(SPACE2)G" <238>
15 PRINT"(SPACE)G(SPACE5)G(SPACE)G(SPACE
RED)Y(BLUE SPACE)G(SPACE)GGG" <150>
16 PRINT"(SPACE11 RED)Z\]^^(BLUE
17 PRINT"(SPACE12 RED)!#*(BLUE
18 PRINT"(SPACE5 GREEN)W(BLUE
20 PRINT"(SPACE)GGG(SPACE GREEN)X(BLUE)G
(SPACE3)G(SPACE2)GGG(SPACE2)G(SPACE)G" <193>
21 PRINT"(SPACE)G(SPACE2)G(SPACE)G(BLACK
)RS(BLUE SPACE)G(SPACE)G(SPACE5)G(SPACE)
G" <191>
22 PRINT"(SPACE)GGG(SPACE2)G(BLACK)TU(BL
UE SPACE)G(SPACE)G(SPACE5)G(SPACE)G" <138>
23 PRINT"(SPACE)GG(SPACE3)G(SPACE3)G(SPA
CE2)GGG(SPACE2)GGG" <208>
24 PRINT"(SPACE)G(SPACE)G(SPACE2)G(SPACE
3)G(SPACE5)G(SPACE)G(SPACE)G" <190>
25 PRINT"(SPACE)G(SPACE2)G(SPACE)G(SPACE
2)IG(GREEN)%&(BLUE SPACE3)G(SPACE)G(SPA
CE)G" <158>
26 PRINT"(SPACE)G(SPACE2)G(SPACE2)GGG(SP
ACE GREEN)'(BLUE)GGG(SPACE2)G(SPACE)G" <26>
27 PRINT"(GREEN)V(SPACE BLACK)/O(SPACEZ
RED)* <96>
28 PRINT"(SPACE2 BLACK)-(SPACEZ RED)+, <217>
29 PRINT"(SPACE18 GREEN)67 <229>
30 PRINT"(SPACE18 GREEN)89 <64>
31 PRINT"(SPACE18 GREEN):;(UP4 WHITE)" <199>

```

Hauptprogramm:

```

0 POKE55,133:POKE56,22:Y=36876:POKEY-7,2
55:D=36903:E=40:F=7066:G=32:POKEF,32 <156>
1 POKED,50:I=7150:H=100:PRINT"(CLEAR)":P
OKEY+3,248:J=5765:K=1:GOSUB100 <65>
10 C=130:SYS5842:SYSJ:PRINT"(HOME RVSON
BLACK)"TAB(PEEK(I)+2)"MILE"INT(L/18) <254>
11 PRINT"(HOME)"TAB(PEEK(I)+25)"(RVSON)M
AN"K"(LEFT)." <19>
15 WAIT37137,32,32:SYS5809 <235>

```

```

16 PRINT " (HOME) "TAB (PEEK (I)+2) " (SPACE2) /
0 (SPACE4 DOWN LEFT8 SPACE2) - (SPACE3) " <249>
20 FOR=1 TO 6: SYSJ: FORA=0 TO C: NEXT: C=C-C/H
: IFPEEK (F) =G THEN NEXT <215>
21 PRINT " (HOME) "TAB (PEEK (I)) A$ (RND (1) *30
): L=L+1 <129>
22 IFPEEK (F) =GG TO 20 <137>
24 IFPEEK (F) =BORPEEK (F) =9 THEN POKEF, 32: GO
TO 20 <94>
25 SYS5793: POKEY, 0: POKEY-1, 0: POKEY+1, 220
: K=K+1 <97>
30 PRINT " (HOME DOWN12) "TAB (PEEK (6B11)-8)
" (BLACK) P (DOWN LEFT) Q"; <109>
40 FORA=15 TO 0 STEP -.05: POKEY+2, A: NEXT: POK
EY+1, 0: POKEF, 32: POKEY+2, 15 <113>
50 IFK <46 TO 10 <95>
55 PRINT " (HOME RVSON) "TAB (180) "
: PRINTTAB (4) " (RVSON) | (SPACE11) | " <139>
56 PRINTTAB (4) " (RVSON) | (SPACE11) | " <103>
57 PRINTTAB (4) " (RVSON) | (SPACE11) | ": PRINT
TAB (4) " (RVSON) " <242>
60 A$ (1) = "GAME" <250>
61 PRINT " (HOME RVSON) "TAB (204); : GOSUB 97:
A$ (1) = "OVER": PRINT " (SPACE) "; : GOSUB 97: PRI
NT: PRINT <111>
62 PRINTTAB (6) " (RVSON) "; : A$ (1) = STR$ (INT (
L/18)); : GOSUB 97 <116>
63 PRINT " (SPACE) "; : A$ (1) = "MILES": IF INT (L
/18) = 1 THEN A$ (1) = "MILE" <4>
64 GOSUB 97 <228>
90 WAIT 37137, 32, 32: RUN <81>
97 FORA=1 TO LEN (A$ (1)): POKEY, 240: A$ = MID$ (
A$ (1), A, 1) <251>
98 PRINT A$: : POKEY, 0 <0>
99 FORB=1 TO 100: NEXT: NEXT: RETURN <214>
    
```

```

100 DIMA$ (30): A$ (0) = " (RIGHT BLACK RIGHT2
) I" <13>
101 A$ (1) = " (BLACK RIGHT7) H" <215>
102 A$ (2) = " (RIGHT3 GREEN) LM (DOWN PURPLE
LEFT2) NO" <56>
103 A$ (3) = " (RIGHT7 GREEN) LM (DOWN LEFT2 P
URPLE) NO" <89>
104 A$ (4) = " (RIGHT9 BLACK) RS (DOWN LEFT2) T
U" <98>
105 A$ (5) = " (RIGHT BLACK) RS (DOWN LEFT2) TU
" <187>
106 A$ (6) = " (RIGHT GREEN) V" <242>
107 A$ (7) = " (RIGHT6 GREEN RIGHT) W (DOWN LE
FT) X" <116>
108 A$ (8) = " (RED) Y (DOWN LEFT) Z (\) ^ (DOWN L
EFT4) ! # $" <40>
109 A$ (9) = " (RIGHT8 GREEN) % & (DOWN LEFT2) '
(DOWN RED LEFT2) * (DOWN LEFT2) +, " <189>
110 A$ (10) = " (BLACK RIGHT3) / 0 (DOWN LEFT2)
- . " <79>
111 A$ (11) = " (RIGHT4 GREEN RIGHT) 12 (DOWN
LEFT2) 3 (PURPLE) 4 (DOWN LEFT2 RIGHT) 5" <209>
112 A$ (12) = " (GREEN RIGHT3) 67 (DOWN LEFT2)
89 (DOWN LEFT2) ; ; " <200>
113 A$ (13) = " (RIGHT2 GREEN) 12 (DOWN LEFT2)
3 (PURPLE) 4 (DOWN LEFT2 RIGHT) 5" <32>
114 A$ (14) = " (BLACK RIGHT5) / 0 (DOWN LEFT2)
- . " <196>
115 A$ (15) = " (BLUE RIGHT4) 67 (DOWN LEFT2) 8
9 (DOWN LEFT2) ; ; " <103>
116 A$ (16) = " (RIGHT7 GREEN) % & (DOWN LEFT2)
' (DOWN BLACK LEFT2) * (DOWN LEFT2) +, " : RE
TURN <201>
ENDE DES LISTINGS
    
```



Geisterhöhle für den VC-20

Ghost-Hell

Ziel des Spieles ist es, in der Geisterhöhle 8 Pillen aufzusammeln, was auch den Punktestand erhöht. Dies ist aber gar nicht so einfach, weil 3 Monster ihr Unwesen treiben.

Mit der „SPACE“-Taste springt Ihr Männchen hoch über die Monster. Ansonsten kann man Leitern rauf und runter gehen und sich fallen lassen – aber nicht zu tief! Andernfalls ist „GAME OVER“.

Wenn man das geschafft hat, erscheint ein Schlüssel. Hat man diesen genommen, geht oben links ein Tor auf. Dort muß man hinausgehen. Das nächste Bild beginnt ...

Das Spiel besteht aus 2 Teilen, die separat abgetippt und gespeichert werden müssen. Im Teil 2 ist eine Routine eingebaut,

die die „RUN-STOP“ und die „RESTORE“-Taste sperrt. Also: Erst abspeichern und dann spielen!!!!

Über die Tastatur wird gesteuert:

- A = rechts
- D = links
- F1 = Leiter rauf
- F7 = Leiter runter
- SPACE = Springen

```

0 REM***** * <145>
* *
1 REM*GHOST HELL BY* * <138>
* *
2 REM*ROLAND SCHOLZ * * <88>
* *
3 REM*MARTIN-LUTHER * * <144>
* *
4 REM*STRASSE 24 * * <48>
* *
5 REM*5354 WEILERSWIST * * <109>
6 REM***** * <133>
7 REM <150>
8 REM <151>
9 REM <152>
10 PRINT " (CLEAR) ": POKE56, 28 <223>
20 FORT=7168 TO 7248 <207>
30 READB <41>
40 IFB=-1 THEN 1000 <63>
50 POKET, B <253>
60 NEXT <190>
70 DATA126, 90, 126, 60, 24, 24, 36, 66 <168>
80 DATA126, 219, 255, 255, 231, 219, 126, 0 <52>
90 DATA31, 127, 255, 255, 170, 87, 82, 138 <20>
100 DATA255, 255, 255, 255, 238, 229, 180, 132 <115>
110 DATA248, 254, 255, 255, 238, 87, 82, 138 <139>
120 DATA129, 129, 255, 129, 129, 255, 129, 255 <128>
130 DATA64, 160, 191, 165, 65, 0, 0, 0 <148>
140 DATA0, 0, 0, 0, 0, 0, 0, 0 <99>
150 DATA255, 145, 145, 255, 255, 143, 137, 137 <210>
160 DATA0, 56, 124, 124, 124, 56, 0, 0, -1 <206>
1000 PRINT " (HOME RVSON RIGHT6) GHOST (SPAC
E) HELL " <72>
1010 PRINT " (DOWN) SIE (SPACE) MUESSEN (SPACE
) MIT (SPACE) IHREM (SPACE DOWN) MAENNCHEN (SP
ACE) ALLE (SPACE) PILLEN (SPACE DOWN) 'FRESSE
N' . " <144>
1020 PRINT " (DOWN) ABER (SPACE) PASSEN (SPACE
) SIE (SPACE) AUF, (SPACE2 DOWN) DIE (SPACE) MO
NSTER (SPACE) WOLLEN (SPACE4 DOWN) SIE (SPACE
) DARAN (SPACE) HINDERN. " <164>
1030 PRINT " (DOWN) UEBER (SPACE) DIE (SPACE) M
ONSTER (SPACE5 DOWN) KOENNEN (SPACE) SIE (SPA
CE) SPRINGEN (SPACE2 DOWN) MIT (SPACE) DER (SP
ACE) TASTE (SPACE RVSON) SPACE (RVSOFF) . " <199>
1040 PRINT " (DOWN2 RIGHT RVSON) TASTE ": POK
E198, 0: WAIT 198, 1: FORT=38 TO 160: POKE36881,
T: NEXT: PRINT " (CLEAR) " <106>
1051 PRINT " NUN (SPACE) ERSCHEINT (SPACE) EIN
(SPACE5 DOWN) SCHLUESSEL, DEN (SPACE) MUESSE
N (DOWN) SIE (SPACE) HOHLEN " <62>
1052 PRINT " (DOWN) DANN (SPACE) GEHT (SPACE) N
AEMLICH (SPACE) DAS (DOWN) TOR (SPACE) OBEN (SP
ACE) LINKS (SPACE) AUF (SPACE4 DOWN) DORT (SPA
CE) MUESSEN (SPACE) SIE (SPACE6 DOWN) RAUSGEH
EN " <222>
1053 PRINT " (DOWN) DANACH (SPACE) FAENGT (SPA
CE) DAS (SPACE5 DOWN) SPIEL (SPACE) VON (SPACE
) FORT (SPACE) AN. " <168>
1054 FORT=160 TO 38 STEP -1: POKE36881, T: NEXT
1055 PRINT " (DOWN2 RIGHT RVSON) TASTE (RVSO
FF) ": POKE198, 0: WAIT 198, 1: PRINT " (CLEAR) " <120>
1060 PRINT " (RIGHT3 SPACE3) STEUERUNG (DOWN
LEFT9) " <52>
1070 PRINT " LINKS (SPACE) : (SPACE) A " <197>
1075 PRINT <81>
<208>
    
```

```

1080 PRINT "RECHTS: (SPACE) D" <212>
1085 PRINT <218>
1090 PRINT "RAUF (SPACE2) : (SPACE) F1 " <156>
1095 PRINT <228>
1100 PRINT "RUNTER: (SPACE) F7 " <2>
1109 PRINT " (DOWN2) SPRINGEN: (SPACE3 RVSON
) SPACE (RVSOFF) " <46>
1110 PRINT " (DOWN2 RIGHT RVSON) TASTE ": POK
E198, 0: WAIT 198, 1: FORT=38 TO 160: POKE36881,
T: NEXT <178>
1120 PRINT " (CLEAR) LADEN (SPACE) SIE (SPACE)
BITTE (SPACE) DEN (SPACE RIGHT2 DOWN) 2. TEIL
(SPACE) EIN (SPACE) ! " <172>
1130 FORT=160 TO 38 STEP -1: POKE36881, T: NEXT
: NEW <6>
ENDE DES LISTINGS
    
```

Teil 2.

```

0 POKE37150, 2: POKE788, 194 <99>
5 POKE36878, 15: D1=1: R1=0: A=0: Q=0: W=0: E=0
: R=0 <123>
10 FORT=7680 TO 8186: POKET, 7: NEXT: POKE3686
9, 255: RESTORE <86>
11 FORT=7424 TO 7424+7: READS: POKET, S: NEXT <105>
12 DATA0, 0, 0, 0, 0, 0, 0, 0 <227>
13 POKE36879, 14: PRINT " (WHITE) " <17>
100 PRINT " (CLEAR) " <130>
110 PRINT <7>
115 PRINT <12>
120 PRINT " (SPACE13) I (SPACE3) E (SPACE) HH " <33>
130 PRINT " (SPACE) HH (SPACE) BCCCCCCCCCCCDE
(SPACE) HH " <42>
140 PRINT " (SPACE) HH (SPACE14) E (SPACE) HH " <13>
150 PRINT " (SPACE) HH (SPACE14) E (SPACE) HH " <23>
160 PRINT " (SPACE) HH (SPACE14) E (SPACE) HH " <33>
170 PRINT " (SPACE) HH (SPACE) I (SPACE9) E (SPA
CE2) E (SPACE) HH " <191>
180 PRINT " BCCCCCCCCCCCCDE (SPACE2) E (SPACE
) HH " <126>
190 PRINT " (SPACE6) HH (SPACE6) E (SPACE2) E (S
PACE) HH " <68>
200 PRINT " (SPACE6) HH (SPACE6) E (SPACE2) E (S
PACE) HH " <78>
210 PRINT " (SPACE) I (SPACE) E (SPACE2) HH (SPA
CE2) I (SPACE3) E (SPACE2) E (SPACE) HH " <239>
220 PRINT " (SPACE) BCED (SPACE) HH (SPACE2) BC
CCCCDE (SPACE) HH " <84>
230 PRINT " (SPACE) E (SPACE) E (SPACE2) HH (SPA
CE2) E (SPACE) HH " <105>
240 PRINT " BE (SPACE) E (SPACE2) HH (SPACE) B (S
PACE7) E (SPACE) HH " <113>
250 PRINT " (SPACE) E (SPACE) E (SPACE3) I (SPAC
E6) E (SPACE2) E IHH " <241>
260 PRINT " (SPACE) E (SPACE) E (SPACE3) BCCCCD
(SPACE) EBCCDHH " <20>
270 PRINT " (SPACE) E (SPACE) E (SPACE5) HH (SPA
CE3) E (SPACE4) HH " <19>
280 PRINT " (SPACE) E (SPACE) E (SPACE2) I (SPAC
E2) HH (SPACE3) E (SPACE4) HH " <142>
290 PRINT " (SPACE) E (SPACE) BCCCCDHH (SPACE3
) E (SPACE4) HH " <71>
    
```

```

300 PRINT" (SPACE)E (SPACE7)I (SPACE4)E (SPA
CE4)HH" <87>
310 PRINT"BBBBBBBBBBBBBBBBBBBBDDHH" <170>
311 B=7734:N=7845:M=8022:A=7728:POKEA+Q+
W+E+R,0 <14>
320 GETA$ <208>
330 IFA$="A"THEN500 <175>
340 IFA$="D"THEN500 <212>
350 IFA$="{F1}"THEN12100 <6>
360 IFA$="{F7}"THEN12000 <23>
370 IFA$="{SPACE}"THEN11000 <167>
380 POKEB+Z,7:POKEX+N,7:POKEV+M,7
390 Z=Z+1:X=X+1:V=V+1 <216>
400 IFZ>2THENZ=-5 <29>
410 IFX>2THENX=-6 <32>
430 IFV>0THENV=-4 <10>
440 IFZ+B=A+Q+W+E+RTHEN10000 <237>
450 IFX+N=A+Q+W+E+RTHEN10000 <35>
470 IFV+M=A+Q+W+E+RTHEN10000 <47>
480 POKEZ+B,1:POKEX+N,1:POKEV+M,1
490 FORT=0T0100:NEXT:OND1GOTO320,7000,11
000 <26>
500 POKEA+Q+W+E+R,7:Q=Q-1:GOSUB3000 <60>
510 IFS=8ORS=2ORS=4THENQ=Q+1 <157>
520 GETA$: IFA$="{SPACE}"THEN7000 <65>
530 POKEA+Q+W+E+R,0:POKE36875,225:POKE36
875,0 <142>
540 GOTO320 <99>
600 POKEA+Q+W+E+R,7:W=W+1:GOSUB3000 <68>
610 IFS=8ORS=2ORS=4THENW=W-1 <199>
620 GETA$: IFA$="{SPACE}"THEN7000 <165>
630 GOTO530 <196>
3000 S=PEEK (A+Q+W+E+R) <50>
3010 IFS=1THEN10000 <161>
3020 IFS=9THENGOSUB9000 <50>
3030 IFA+Q+W+E+R=7834THENGOTO5 <179>
3035 IFS=2ORS=4ORS=8THEN3060 <46>
3040 IFS=6THENGOSUB8000 <51>
3050 IFS=2ORS=4ORS=8THEN3060 <61>
3060 D=PEEK (A+Q+W+E+R+22) <52>
3070 IFD=7ORD=32THEN7000 <129>
3080 RETURN <162>
7000 D1=2:POKEA+Q+W+E+R,7 <238>
7010 E=E+22 <195>
7020 D2=PEEK (A+Q+W+E+R) <56>
7030 IFD2=7ORD2=32THEN7040 <82>
7035 D1=1:D3=0:E=E-22:POKEA+Q+W+E+R,0:GO

```

```

T0320 <2>
7040 POKEA+Q+W+E+R,0:POKE36875,225:POKE3
6875,0 <21>
7050 D3=D3+1 <199>
7060 IFD3>3THEN10000 <127>
7070 GOTO380 <16>
8000 SC=SC+100 <108>
8010 FORT=7747T07836STEP22:POKET,7:POKE3
6875,240:POKE36875,0:NEXT <112>
8020 FORT=7748T07837STEP22:POKET,7:POKE3
6875,200:POKE36875,0:NEXT <21>
8030 RETURN <11>
9000 FORT=200T0240:POKE36875,T:POKE36875
,0:NEXT <170>
9010 SC=SC+10:SP=SP+1:IFSP=8THENPOKE7933
,6:SP=0 <83>
9020 PRINT" (RVSON HOME)SCORE: ";SC <116>
9030 RETURN <247>
10000 PRINT" (RVSON HOME DOWN6 SPACE7)GAM
E (SPACE)OVER (SPACES)" <171>
10010 PRINT" (RVSON SPACE)PRESS (SPACE)A (S
PACE)KEY (SPACE) TO (SPACE)START (SPACES) THE
(SPACE)GAME (SPACE) AGAIN (SPACE4)":POKE198
,0:WAIT198,1:RUN <59>
11000 D1=3:POKEA+Q+W+E+R,7 <161>
11010 R1=R1+1 <135>
11020 IFR1>2THEN11050 <82>
11030 R=R-22:POKEA+Q+W+E+R,0 <233>
11040 GOTO380 <162>
11050 E=E+22 <155>
11060 A9=PEEK (A+Q+W+E+R) <27>
11070 IFR1=5THENE=E-22:POKEA+Q+W+E+R,0:R
1=0:D1=1:GOTO320 <157>
11090 POKEA+Q+W+E+R-22,7:POKEA+Q+W+E+R,0 <51>
11100 GOTO380 <222>
12000 POKEA+Q+W+E+R,5:E=E+22 <27>
12010 A2=PEEK (A+Q+W+E+R-22) <184>
12020 IFNOTA2=5THENE=E-22 <40>
12030 IFNOTPEEK (A+Q+W+E+R)=5THENE=E-22 <217>
12040 POKEQ+W+E+R+A,0:POKE36875,225:POKE
36875,0 <59>
12050 GOTO320 <134>
12100 POKEA+Q+W+E+R,5:R=R-22 <23>
12110 A2=PEEK (A+Q+W+E+R) <43>
12120 IFNOTA2=5THENR=R+22 <108>
12130 POKEA+Q+W+E+R,0:POKE36875,225:POKE
36875,0 <11>
12140 GOTO320 <224>

```

Tips & Tricks für den VC-20

SCROLLING TIPS

Scroll nach unten weg und zurück.

```

10 FOR X=25 TO130:POKE36881,X:NEXT:PRINT"clearscreen"
20 PRINT"neuer Text"
30 FOR X=130TO25STEP-1:POKE36881,X:NEXT

```

Diese Routine existiert in verschiedenen Varianten, die sich lediglich in den Werten für die X-Schleife unterscheiden.

Scroll nach links weg und zurück.

```

10 FORI=0TO30:POKE251,I:POKE252,16:FOR K=1TO130:NEXT

```

K,I:PRINT"clearscreen"

```

20 PRINT"neuer Text"
30 FORI=30TO1STEP-1:POKE251,I:POKE252,16:
FOR K=1TO10:NEXT K?

```

K.O.-Scroll.

Folgende Scrollroutine eignet sich besonders für Spiele, in denen die Spielfigur K.O.geschlagen wird - wie z. B. bei 'Kong'-Spielen:

```

10 FOR I=5TO55:POKE 36864,I:POKE36865,I+I:FOR K=1TO65:NEXT
K,I:PRINT"clearscreen"

```

```

20 PRINT"neuer Text"
30 POKE36864,12:POKE36865,38

```

Aus- und einblenden.

```

10 FOR I=23TO1 STEP-1:POKE36867,I+I:POKE36866,I-1:POKE36864,13:POKE36865,82-INT(I/2)*4:NEXT I
20 PRINT"clearscreen":PRINT"neuer Text"
30 FOR I=1TO23:POKE36867,I+I:POKE36866,I-1:POKE36864,13:POKE36865,82-
30 FOR I=1TO23:POKE36867,I+I:POKE36866,I-1:POKE36864,13:POKE36865,82-INT(I/2)*4:NEXT I:I=0

```

Seitwärts mit Text

```

10 BL$="19 leertasten": A$="text"
20 A$=BL$+A$
30 FOR I=1 TO LEN(A$): PRINT TAB(2)
MID$(A$,I,1)
40 PRINT "up up": FOR K=1TO150:
NEXT K,I (ggfls. :RETURN)

```

EINE Bildschirmzeile löschen.

```
POKE781,Zeilenummer:POKE782,0:POKE783,0:SYS60045
```

EINE Bildschirmzeile bis zur angegebenen Spalte löschen:

```
POKE781,Zeilenummer:POKE782,Spaltennummer:POKE783,0:SYS60047
```

Erdbeben

```

10 FOR Y=1TO20:FOR X=3 TO 7:POKE 36864,X:NEXT XY:POKE 36864,12
20 FOR Y=1TO20:FOR X=23TO27:POKE 36865,X:NEXT XY:POKE 36865,38
Die erste Zeile ist für ein horizontales, die zweite für ein vertikales Beben.

```

Scroll nach unten: Variante

```

10 A$="":FOR T= 0 TO 18:A$=
A$+CHR$(29):NEXT
11 A$=CHR$(19)+A$+CHR$(160)+CHR$(157)+CHR$(157)
12 A$=A$+CHR$(157)+CHR$(157)+CHR$(157)+CHR$(148)+CHR$(148)+CHR$(148)

```

Damit sind die einzusetzenden Strings definiert: der Aufruf erfolgt mit GOTO oder GOSUB folgende Zeile: nnnnn PRINT A\$:POKE217,158:POKE218,158

Schwarzes Loch

```

10 A=150:B=174
20 FOR C=1TO21:A=A-1:B=B-2:POKE 36866,A:POKE 36867,B:NEXT
30 PRINT CHR$(147);
40 FOR C=1TO21:A=A+1:B=B+2:POKE 36866,A:POKE 36867,B:NEXT
(50 RETURN) falls erforderlich

```

Wirbelsturm

```

10 A=36864:B=36865
20 FOR X=46 TO 0 STEP -2:POKEB,X:POKEA,51-X
30 FOR T=1TO5: NEXT T,X
40 ggfl. neues Bild erzeugen, sonst weglassen
50 FOR X= 0to 46Step 2:POKEB,X:POKEA,51-X
60 FOR T=1TO5: NEXT T,x
Zeilen 20 und 50 können als selbständige Unterroutinen aufgerufen werden, wenn die Zeile mit einem RETURN beendet wird.

```

Wieder auswärts

```

10 PRINT"clear 19(neunzehnmal) down"
20 FOR N=1TO LEN(A$):REM A$ ist der zu scrollende Text
30 PRINTTAB(11)MID$(A$,N,1)
40 FOR T=1TO175:NEXT T,N
50 FOR K=1TO500:NEXT:PRINTCHR$(147) (ggfls. :RETURN)

```

Scroll für eine Zeile nach unten (erste Zeile ausgenommen).

Diese Routine scrollt ab der zweiten Bildschirmzeile eine Zeile nach unten: PRINT"home crsrdown crsrleft inst":POKE218,158 In Verbindung mit einer Zählschleife, z. B. FOR I=1TO20, kann man den ganzen Bildschirm nach unten wegschrollen, und die erste Zeile als Statuszeile stehenlassen.

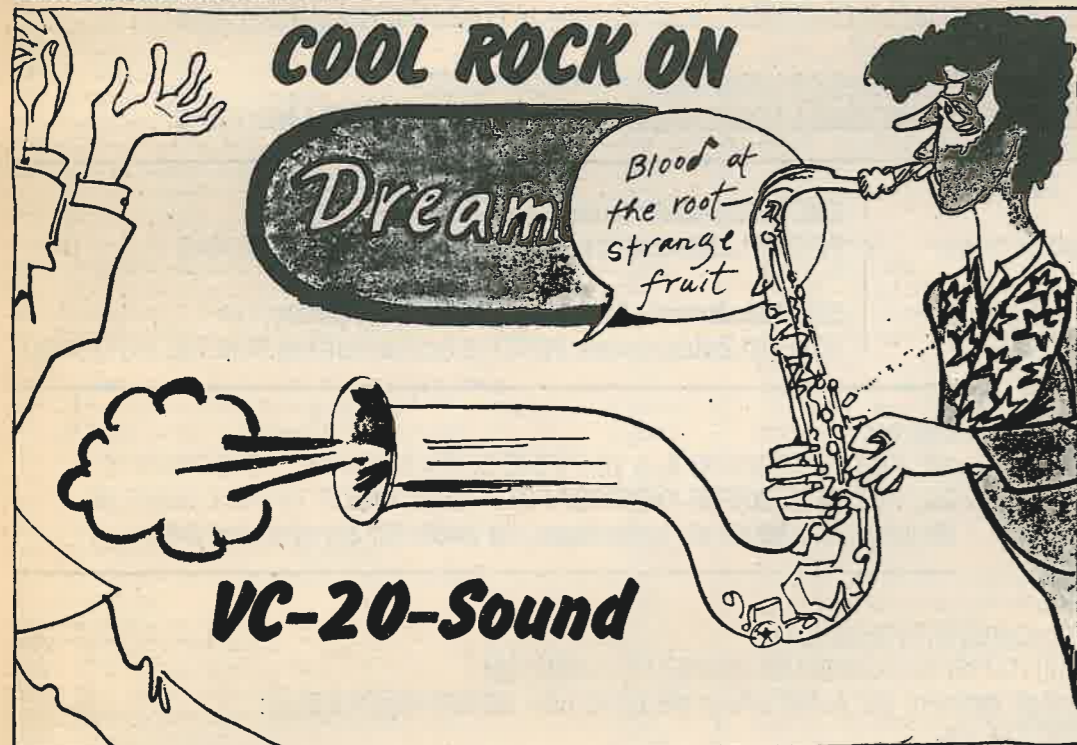
Langsam aufwärts

```

10 POKE 36879,8:PRINT" text"
20 FOR A=131TO0 STEP-1: POKE 36865,A: FOR B=1 TO45:
NEXT B,A: POKE 36865,24
Die Aufwärtsgeschwindigkeit wird durch die B Schleife kontrolliert; die durch diese Routine geänderten Farbwerte müssen ggfl. zurückgesetzt werden.

```

Werkstatt:



Dies ist eine außerordentlich kurze Soundroutine für den VC-20, die voll in Basic geschrieben ist und alle 4 Soundregister ausnutzt.

Mit diesem kleinen Programm läßt sich sehr schön experimentieren. Z.B. können die Werte in Zeile 120 geändert werden.

Versuchen Sie mal
250 AB\$="22324322"

Wir fanden, daß das Programm die Soundmöglichkeiten des VC-20 wirklich optimal ausnutzt.

```

160 POKE36879,253:REM COOL ROCK
170 PRINT" (CLEAR)"
180 PRINT" (RIGHT6) SILVER-SOFT"
191 PRINT" (DOWN RIGHTS BLACK)P (RED)R (CYA
N)E (PURPLE)S (GREEN)E (BLUE)N (YELLOW)T (BLA
CK):"
200 PRINTCHR$(144):PRINT" (C) '86 (SPACE) BY
(SPACE) THOMAS (SPACE) WELF"
201 PRINT" (RIGHT2 DOWN RIGHT4) COOL (SPACE
) ROCK"
210 B=36874:N=36875:M=36876:J=36877
220 R(1,1)=195:R(1,2)=207:R(1,3)=215:R(1
,4)=219:R(1,5)=221:R(1,6)=219:
221 R(1,7)=215:R(1,8)=207
230 R(2,1)=209:R(2,2)=219:R(2,3)=225:R(2
,4)=228:R(2,5)=229:R(2,6)=228:
231 R(2,7)=225:R(2,8)=219
240 R(3,1)=215:R(3,2)=223:R(3,3)=228:R(3
,4)=231:R(3,5)=232:R(3,6)=231:
241 R(3,7)=228:R(3,8)=223
250 AB$="11213211"
260 TA$="101202"
270 T(1,1)=207:T(2,1)=215:T(3,1)=225
280 T(1,2)=207:T(2,2)=215:T(3,2)=225
290 T(1,3)=209:T(2,3)=219:T(3,3)=225
300 T(1,4)=207:T(2,4)=215:T(3,4)=225
310 T(1,5)=215:T(2,5)=223:T(3,5)=228
320 T(1,6)=209:T(2,6)=219:T(3,6)=225
330 T(1,7)=207:T(2,7)=215:T(3,7)=225
340 T(1,8)=207:T(2,8)=215:T(3,8)=225
350 POKE36878,5
360 FORU=1TO6
370 FORZ=1TO8
380 RESTORE
390 AB=0:ZE=2:ZT=0
400 FORX=0TO23
<72>
<200>
<62>
<142>
<94>
<124>
<207>
<186>
<90>
<101>
<188>
<46>
<139>
<44>
<23>
<49>
<107>
<13>
<223>
<88>
<187>
<141>
<199>
<29>
<230>
<6>
<9>
<81>
<93>
410 READC:POKEM,T(C,Z)
420 ZE=ZE+1:IFZE=3THENZE=0:AB=AB+1
430 POKEN,R(VAL(MID$(AB$,Z,1)),AB)
440 ZT=ZT+1:IFZT=7THENZT=1
450 A=VAL(MID$(TA$,ZT,1)):IFA=1THENPOKEJ
,200
460 IFA=2THENPOKEJ,252
470 IFX=0ORX=11THENPOKEB,R(VAL(MID$(AB$,
Z,1)),1)
480 IFX=5ORX=17THENPOKEB,R(VAL(MID$(AB$,
Z,1)),3)
490 POKEJ,0
500 FORW=1TO15:NEXTW,X,Z
510 ONUGOSUB550,610,550,610,550,610
520 NEXTU:PRINT" (CLEAR)":END
530 DATA1,1,2,3,3,1,2,2,3,1,1,2,3,3,1,2,
2,3,3,3
540 DATA3,3,3,3
550 PRINT" (CLEAR DOWN2) DIESES (SPACE) MUSI
C-PROGRAMM (DOWN3 RIGHT2) *** IST (SPACE) COO
L (SPACE) ROCK ***"
560 PRINT
570 PRINT
580 PRINT
590 PRINT
600 RETURN
610 PRINT" (CLEAR DOWN4)":CHR$(144)
620 PRINT" DIE (SPACE) MELODIE (SPACE) BEINHA
LTET (DOWN2) 4 (SPACE) STIMMEN"
630 PRINT" (DOWN1) 3-TON (SPACE) MELODIE"
631 PRINT" (DOWN2) JAZZ-UNTERSTIMME"
632 PRINT" (DOWN3) WECHSELBASS"
640 PRINT" (DOWN4) SCHLAGZEUG":PRINTCHR$(
144)
650 PRINT
660 RETURN
<164>
<109>
<163>
<18>
<35>
<229>
<150>
<38>
<90>
<191>
<47>
<162>
<165>
<49>
<23>
<203>
<213>
<223>
<233>
<232>
<64>
<6>
<184>
<186>
<172>
<163>
<37>
<36>

```



ROADRUNNER

3 Spiele in einem Programm für den C-64

„Roadrunner“ ist ein Superspiel, ganz in Maschinensprache geschrieben und beinhaltet 3 verschiedene Spielfelder.

Level 1:

Im ersten Bild muß der Roadrunner von Sonnenaufgang bis Sonnenuntergang durch die Wüste laufen und dabei viele Hindernisse überwinden.

Joysticksteuerung:
nach oben = Sprung
nach oben und dann nach rechts verlängert den Sprung

Level 2:

Der Roadrunner hat sich aus der Wüste in die Berge gerettet. Nun muß er über Schluchten springen.

Joysticksteuerung:
Feuertaste: Roadrunner beginnt zu laufen
nach oben = Großer Sprung
nach rechts = kleiner Sprung
Sind alle 3 Level geschafft, geht das Spiel

mit erhöhter Geschwindigkeit von vorne los.

Level 3:

Roadrunner bleibt vom Pech verfolgt. Nun hat er sich in ein Indianerreservat verirrt und muß den Pfeilen der Indianer ausweichen.

Joysticksteuerung:
nach allen 4 Seiten

```

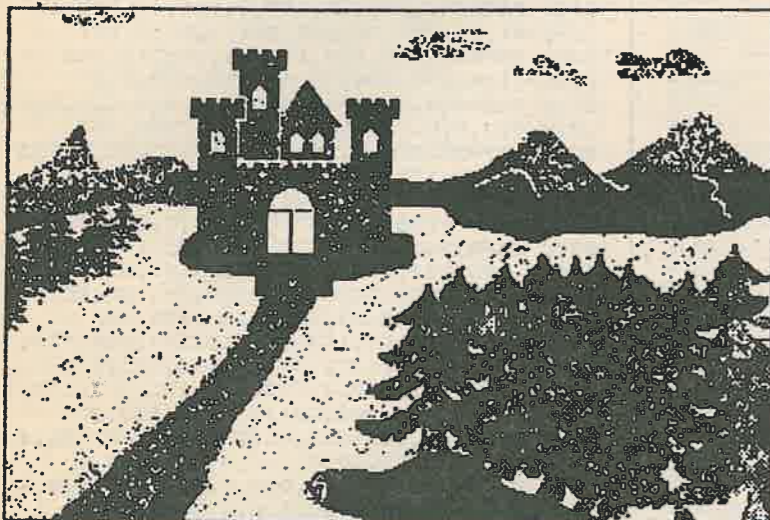
1000 REM DATA-LOADER FUER * ROADRUNNER *
1010 REM VOR DEM STARTEN ABSPEICHERN
1020 :
1030 :
1040 DATA0,55,8,194,7,158,53,49,50,49,58
,143,32,42,32,82,79,65,68,82,85,78,78
1050 DATA69,82,32,42,32,32,32,32,32,3
2,32,32,32,32,32,32,40,67,41,32,82
1060 DATA46,32,68,73,83,83,69,82,0,0,0,6
8,68,68,68,68,68,68,68,68,40,67
1070 DATA41,32,82,46,32,68,73,83,83,69,8
2,0,0,0,133,250,169,54,133,251,169,0
1080 DATA133,252,169,34,133,253,169,1,13
3,98,169,54,133,99,173,161,31,24,105
<3>
<216>
<57>
<68>
<39>
<41>
<90>
<94>
<9>
<3>
<234>
<210>
<26>
<177>
<171>
<136>
<247>

```


Was gibt's Neues auf dem Software-Markt

Hi-EDDI plus

Fantastisches Zeichenprogramm für den C-64



Mit Hi-EDDI plus stellt Markt & Technik ein Programm vor, das nicht wie üblich als Fertigprodukt aus einem großen Softwarehaus stammt, sondern von einem Anwender, Computerfan und Student der Elektrotechnik zunächst nur für eigene

Zwecke entwickelt wurde.

Der erste Erfolg stellte sich ein, als das Programm in einer Computerzeitung veröffentlicht wurde. Seitdem meldeten sich bei dem Autor eine Vielzahl von Anwendern mit neuen Tips und auch Wünschen,

die Hans Haberl für eine Vielzahl von Verbesserungen und Erweiterungen verwendete. Das Ergebnis ist HI-EDDI plus, ein Malprogramm mit vielen ganz besonderen Eigenschaften. HI-EDDI + nutzt zunächst einmal konsequent den High-Resolutionmodus, um die größtmögliche Bildauflösung auf dem C-64 zu erreichen. Damit sind mit HI-EDDI technische Zeichnungen kein Problem mehr.

Technisches Zeichnen wird zudem noch unterstützt durch Konstruktion-Sets, die zum Beispiel Schaltbildsymbole enthalten, und der Möglichkeit, bis zu 6 Bildschirme zu einem Großbildschirm zusammenzufassen und nahtlos ausdrucken zu lassen. Wollten wir alle Vorzüge dieses Zeichenprogramms aufführen, wäre der Rahmen dieser Besprechung sicher gesprengt.

Auf zwei besondere Dinge sei jedoch hingewiesen. HI-EDDI + besticht vor allem durch seine exzellente Dokumentation in Buchform, die unserer Meinung nach der beste Schutz gegen unerlaubtes Kopieren ist. Die 48,- DM lohnen sich schon allein für die fantastische und sehr verständlich geschriebene Anleitung, die auch Wortindex, Befehlsübersicht und ausführliche Assemblerdokumentation beinhaltet, was man bei vielen Programmen vergeblich sucht. Besonders gefallen haben uns auch die Runtime-Module, mit denen man die fertigen Bilder auch in eigene Programme einbinden kann, auch in eigene Grafikadventure! HI-EDDI + besitzt Druckertreiber für die meisten gängigen Drucker und arbeitet mit Joystick und Koala-Pad als Eingabegeräte zusammen. (uk)

HI-EDDI plus, Markt & Technik,
8013 Haar, Preis: 48,- DM

NEU für den Schneider CPC

BASIC COMPILER V2.0

Das Softwareteam T. Brandt und F. Thilen hat eine verbesserte Version ihres Basic-Compilers auf den Markt gebracht. Die Version 2.0 vereinigt alle Features seines Vorgängers in sich, wurde aber in einer entscheidenden Sache verbessert: Speicherplatz ist nun auch für längere Programme vorhanden. Mit der neuen Version können nun auch eigenständige Programme entwickelt werden, wobei allerdings der Nachteil der vom Locomotive Basic abweichenden Syntax nach wie vor vorhanden ist. Ebenfalls besteht die Möglichkeit, fast jedes beliebige Textpro-

gramm zur Programmerstellung zu verwenden, allerdings fällt dann die Möglichkeit weg, die Syntax eines Programmes vor dessen Compilierung testen zu lassen. Alles in allem ein Programm, das hauptsächlich zu Optimierung und Geschwindigkeitssteigerung bereits bestehender Programme eingesetzt werden kann. Für den Software-Entwickler sicher nicht uninteressant.

Vertrieb: Software Team
Joachim Guenster
Mühlenstr. 12
5431 Boden

Das Soft- und Hardware-Angebot für C-16/116 wurde erweitert

In den letzten Wochen wurde das Soft- und Hardware Angebot der Fa. Kingsoft erheblich erweitert. So wurden die Spielprogramme

| | |
|------------------------|---------------|
| Bongo Construction Set | Preis 25,- DM |
| Galaxy | Preis 19,- DM |
| Ghost Town | Preis 19,- DM |
| Space Pilot | Preis 25,- DM |
| Tom | Preis 25,- DM |
| Winter Olympiade | Preis 29,- DM |

neu mit in das Liefersortiment aufgenommen. Die Programme Galaxy, Ghost Town, Grandmaster und Tom sind auch als Sammelkassette erhältlich und kosten im PLUS-Paket zusammen 39,- DM. Alle Programme sind mit eingebautem Turbotape auf Kassette abgespeichert, so daß die Ladezeit der einzelnen Programme deutlich verkürzt wird. Sie beträgt teilweise nur 1 Minute oder liegt noch darunter.

Das Programm Winter Olympiade lag der Redaktion leider nur als Demo-Version vor. Aber der kurze Einblick in dieses Programm, läßt auf einen absoluten Knüller hoffen. So wird das Spiel insgesamt 6 Disziplinen beinhalten (Abfahrt, Biathlon, Bobfahren, Eisschnelllauf, Skispringen, Slalom) und für 4 Mitspieler ausgelegt werden. Die Eröffnungszereemonie, die Namens eingabe, die Landesflaggen und die Hymnen werden selbstverständlich nicht fehlen. Der Aufbau ist also ähnlich wie bei den Winter Games angelegt.

Als weiteres sind auch einige Utilities in dem Programmangebot enthalten. So wird u. a. ein

Grafik-Designer Preis 19,- DM
(Zeichensatzeditor mit Animation)

Micro-Datei Preis 29,- DM
Dateiverwaltung in freiem Maskenaufbau

Micro Kalk Preis 29,- DM
(Tabellenkalkulation, einfach zu bedienen)

Micro Text Preis 29,- DM
Textverarbeitung u.a. mit Blocksatz)

Music Master Preis 29,- DM
(Musik-Pgm. mit Synthes. & Sequenzer)

Paintbox Preis 25,- DM
(Mal- u. Zeichenpgm. ähnlich Koala Paint)

Turbo-Tape Preis 19,- DM
(8mal schneller laden und speichern)

geliefert. Wer also einige nützliche Anwenderroutinen benötigt, sollte auf diese Angebote zurückgreifen.

Im Hardware-berich werden Joysticks, Joystick-Adapter, Speichererweiterungen u.v.m. angeboten, die zu einem sinnvollen Ausbau Ihres Systems beitragen.

Bezugsadresse: Kingsoft, Schnackebusch 4,
5106 Roetgen

Betrifft: Autorenrechte

Aus gegebenem Anlaß müssen wir auf eine besondere Form des Raubkopierens hinweisen. Im Verlag mußten wir wiederholt feststellen, daß unter den eingesandten Programmen unserer Leser Software war, deren Urheberrechte nicht oder nicht mehr bei den Einsendern lagen. In einigen Fällen hatten die Autoren ihr Programm schon in anderen Computerzeitungen veröffentlicht und Honorar kassiert. In einem anderen Fall hat ein Leser das kopierte Programm eines Bekannten eingeschickt und Autorenrechte geltend gemacht. Diese Form des Raubkopierens kann böse Folgen für die Einsender, bei Minderjährigen auch für die Eltern, haben. Nicht immer sind die eigentlichen Autoren bereit, einem außergerichtlichen Vergleich zuzustimmen. Sie werden sich jedoch in erster Linie an den Verlag wenden, der wiederum sich an dem Übeltäter schadlos hal-

ten muß. Anders als beim unerlaubten Kopieren von geschützter Software ist der Kopierer von Programmeinsendungen nicht durch Anonymität geschützt und muß in den allermeisten Fällen mit Strafverfolgung rechnen. Im Interesse unserer Leser und um unnötigen Ärger vor allem für Jugendliche zu vermeiden, bitten wir Programmeinsender, nur Programme zu schicken, die in allen wesentlichen Teilen aus eigener Feder stammen und uns dies in dem Coupon für Programmautoren mit einer separaten Unterschrift zu bestätigen.

Merke!

Keine Software-Unterlagen einsenden, wenn man nicht in besitz der Urheberrechte ist, um unnötigen Ärger zu vermeiden.

Achtung ...

Die nächste C-16/116-Sonderausgabe ist ab

16. Juni 1986

überall im Zeitschriftenhandel erhältlich.

Versäumen Sie nicht diese Ausgabe!

Es gibt eine Riesengewinn-Chance und wie immer jede Menge „Futter“ für Ihr System.

Schickt uns Eure Assembleroutinen!

Basic-Extension für Commodore 64

Tolles Programm, um Befehlsweiterungen selbst zu erstellen

Thomas Lais hat uns ein sehr gutes Programm geschickt, um kurze Assembler-routinen mit Basicbefehlswörtern aufzurufen. Unser Appell an unsere Leser: Schickt uns Eure kurzen Assemblerprogramme, damit sie von den anderen Lesern mit Basicextension in eine Befehlsweiterung eingebunden werden können. Im folgenden erklärt Thomas Lais selbst den Aufbau und die Bedienung seines Programms.

Ein großer Minuspunkt des C-64 ist sein recht mageres Basic.

Dem kann man abhelfen, indem man Basicerweiterungen verwendet - seien es käufliche oder in Zeitschriften veröffentlichte. Aber auch das ist nicht immer die beste Lösung, denn wahrscheinlich fehlt gerade der Befehl, den man für ein bestimmtes Programm braucht. Also bleibt nur eine Basicerweiterung selbst zu schreiben - und steht dann vor dem Problem, diese einzelnen Befehle aufzurufen. **SYS** Aufrufe zu verwenden, führt mit Sicherheit zu Fehlern, wenn 2 Adressen verwechselt werden. Am besten ist es doch, den Aufruf mittels eines Befehlswortes zu vollziehen. Und dieses Programm hilft dabei! Sie benötigen nur das oder die Assemblerprogramme, die dann mittels Befehlsworten aufgerufen werden sollen, die auf Diskette oder Kassette vorliegen müssen. Das Programm muß dabei bereits an der Stelle liegen, an der es auch in der Befehlsweiterung liegen soll. Nach Starten des Programms können Sie wählen zwischen der Neuerstellung einer Basicerweiterung und der Bearbeitung einer bereits vorhandenen Erweiterung. Zunächst zum Punkt Neuerstellung. Das Programm fragt die Startadresse und die

Endadresse der Erweiterung ab und überprüft die Eingaben. Erlaubt sind nur die Bereiche über 12800 (darunter liegt das Programm) bis 40959 (ab da beginnt der Basicinterpreter) oder von 49152 bis 53247. Danach wird das Assemblerprogramm, das die Dekodierung der Befehls-worte und den Aufruf der einzelnen Routinen vornimmt, in den Bereich ab der Startadresse gepoked. Ebenso die Datas für das fest installierte Befehlswort **OUT** und die Datas für diesen Befehl, der ein Ausschalten der Erweiterung bewirkt. Die Datas sind für eine Startadresse ab 49152 ausgelegt. Wurde eine andere Startadresse gewählt, wird das Assemblerprogramm noch entsprechend verändert. Nun ist die Befehlsweiterung initialisiert und Menü 2 wird angesprochen. Im Menü 2 stehen 5 Möglichkeiten zur Wahl: Anzeige bisheriger Befehle, Befehl löschen, Basicerweiterung abspeichern, Laden eines neuen Befehls und Ende. Anzeige bisheriger Befehle zeigt die bisher vorhandenen Befehle samt der Start-adressen an. Befehl löschen streicht einen Befehl aus der Befehlsliste. Der Befehl **OUT** ist fest eingebaut und kann nicht gelöscht werden. **Saven** der Befehlsweiterung sichert das gesamte Assemblerprogramm auf Diskette oder Kassette. Beim Laden eines neuen Befehls wird zunächst überprüft, ob noch Platz ist in der Erweiterung. Es können maximal 50 Befehle pro Befehlsweiterung verwendet werden, wobei die Gesamtzahl der Befehlsbuchstaben kleiner als 270 sein muß. Nach Eingabe des Namens und des Gerätes wird das Assemblerprogramm geladen. Die Startadresse und die Endadresse werden automatisch bestimmt. Falls diese beiden Adressen in einem falschen Bereich liegen, meldet das Basicprogramm

das. War aber alles in Ordnung, wird nun das neue Befehlswort eingegeben und eingebunden. **ACHTUNG:** Das Programm erkennt nicht, ob der neu geladene Befehl einen bereits vorhandenen ganz oder teilweise überschreibt. Das aber führt dazu, daß es bei einem Aufruf des überschriebenen Befehls zu einem Programmabsturz kommt. Vermeiden Sie dies, indem Sie Aufzeichnungen über die Länge und die Lage der einzelnen Assemblerprogramme führen. „Ende“ beendet das Programm. Wurde eine Veränderung am Assemblerprogramm vorgenommen, also ein Befehl gelöscht oder geladen, ohne das die geänderte Version gespeichert wurde, kommt es noch zu einer Sicherheitsabfrage ob das Programm auch wirklich beendet werden soll. Bei Eingabe von **N** wird zum Menü 2 gesprungen. Wird einer der 3 Punkte „Befehl löschen“, „Befehl laden“ oder „Saven“ versehentlich angewählt, kann durch die Eingabe von „←“ bei der ersten vom Programm erwarteten Eingabe zum Menü 2 zurückgesprungen werden. Wurde bei Menü 1 der Punkt „Bearbeiten einer Basicerweiterung“ gewählt, werden der Name der Erweiterung und das Gerät abgefragt, dann die Erweiterung geladen. Die Startadresse und die bisherige Endadresse werden automatisch bestimmt und ausgegeben; dann wird die Eingabe einer neuen Endadresse erwartet. Diese muß größer oder gleich der bisherigen Endadresse sein. Wird das Programm aus irgendwelchen Gründen unterbrochen, kann es mit **GOTO 3050** ohne Datenverlust fortgesetzt werden. Die abgespeicherte Befehlsweiterung wird mit **LOAD "NAME",8,1** oder **,1,1** geladen

und initialisiert mittels **SYS SA**, wobei **SA** die Startadresse ist, ab der das Programm liegt. Liegt die Erweiterung im Basicspeicher, also unterhalb 40959, so muß vor dem Laden der Basicspeicher entsprechend beschränkt werden.

```

10 REM ***** <15>
20 REM *** <17>
30 REM *** BASIC-EXTENSION *** <11>
40 REM *** ----- *** <39>
50 REM *** <47>
60 REM *** (C) 1986 BY *** <200>
70 REM *** ARTEXSOFT *** <157>
80 REM *** <77>
90 REM ***** <95>
100 IF A=0 THEN POKE 52,50:POKE56,50:CLR <91>
110 IF A=1 GOTO 7270 <18>
120 IF A=2 GOTO 2160 <243>
130 DEF FNL(X)=X-(INT(X/256)*256) <104>
140 DEF FNH(X)=INT(X/256) <195>
200 LL$="{SPACE39}" <100>
300 FOR I=710 TO 735 <88>
310 : READ P:POKE I,P <180>
320 NEXT <195>
350 DATA162,8,160,1,32,186,255,169,255,1 <54>
62,224,160,2,32,189,255,169,251,162 <20>
360 DATA0,160,0,32,216,255,96 <11>
700 REM ***** <172>
730 PRINTCHR$(147)
740 PRINT "{RVSON}****{SPACE}HAUPTMENUE{ <38>
SPACE}****{RVSOFF}"
750 PRINT:PRINT:PRINT"BITTE{SPACE}WAEHLE <200>
N:"
760 PRINT:PRINT:PRINT"(1){SPACE}NEUERSTE <126>
LLUNG{SPACE}BASICERWEITERUNG"
770 PRINT:PRINT"(2){SPACE}BEARBEITUNG{SP <71>
ACE}BASICERWEITERUNG"
780 GET A$:IF A$="2" GOTO 2000 <81>
790 IF A$<>"1" GOTO 780 <165>
1000 REM <122>
1010 REM ***** NEUERSTELLUNG ***** <223>
1020 REM <142>
1050 PRINTCHR$(147) <238>
1060 PRINT "{SPACE RVSON}***{SPACE}NEUERS <197>
TELLUNG{SPACE}BASICERWEITERUNG{SPACE}*** <239>
{RVSOFF}"
1070 PRINT:PRINT:PRINT <6>
1080 INPUT"STARTADRESSE";SA <30>
1090 PRINT:PRINT <19>
1100 INPUT"ENDADRESSE{SPACE2}";EA <132>
1110 IF SA>=EA GOTO 1180
1120 IF (SA>12800 AND SA<40960 AND EA<40 <244>
960) GOTO 1200
1130 IF (SA>49151 AND SA<13*4096 AND EA< <82>
13*4096) GOTO 1200
1180 PRINT:PRINT:PRINT"FALSCHER{SPACE}EIN <198>
GABEN.{SPACE}BITTE{SPACE}KORRIGIEREN" <172>
1190 FOR I=1 TO 2000:NEXT:GOTO 1000 <238>
1200 :
1210 FOR I=SA TO SA+143:READ P:POKEI,P:N <96>
EXT <47>
1220 DATA169,11,141,8,3,169,192,141,9,3, <189>
96,32,115,0,144,120,201,90,176,116 <112>
1230 DATA201,65,144,111,141,139,192,162, <189>
0,142,140,192,169,192,133,252,169 <112>
1240 DATA242,133,251,160,0,238,140,192,1 <189>
77,251,48,35,209,122,208,4,200,76 <112>
1250 DATA45,192,173,140,192,205,141,192, <189>
240,67,200,177,251,16,251,200,24,152

```



```

1260 DATA101,251,133,251,144,2,230,252,7 <125>
6,40,192,41,127,209,122,208,34,200 <116>
1270 DATA24,152,101,122,133,122,144,2,23 <79>
0,123,173,140,192,10,170,189,140,192 <212>
1280 DATA141,119,192,189,141,192,141,120 <9>
,192,32,0,194,76,234,167,173,140,192 <54>
1290 DATA205,141,192,208,194,173,139,192 <6>
,56,76,231,167,1,1,1,0,194 <208>
1300 FOR I=SA+242 TO SA+244:READ P:POKE <91>
I,P:NEXT <89>
1310 DATA79,85,212 <151>
1320 FOR I=SA+512 TO SA+522:READ P:POKE <228>
I,P:NEXT <195>
1330 DATA169,228,141,8,3,169,167,141,9,3 <170>
,96 <48>
1400 IF SA=49152 GOTO 1800 <240>
1500 POKE SA+1,FNL(SA+11):POKE SA+6,FNH( <7>
SA+11) <182>
1510 POKE SA+25,FNL(SA+139):POKE SA+26,F <60>
NH(SA+139) <154>
1520 POKE SA+30,FNL(SA+140):POKE SA+31,F <121>
NH(SA+140) <208>
1530 POKE SA+33,FNL(SA+242):POKE SA+37,F <91>
NH(SA+242) <89>
1540 POKE SA+43,FNL(SA+140):POKE SA+44,F <151>
NH(SA+140) <228>
1550 POKE SA+55,FNL(SA+45):POKE SA+56,FN <195>
H(SA+45) <170>
1560 POKE SA+58,FNL(SA+140):POKE SA+59,F <48>
NH(SA+140) <240>
1570 POKE SA+61,FNL(SA+141):POKE SA+62,F <7>
NH(SA+141) <182>
1580 POKE SA+82,FNL(SA+40):POKE SA+83,FN <60>
H(SA+40) <154>
1590 POKE SA+102,FNL(SA+140):POKE SA+103 <154>
,FNL(SA+140) <108>
1600 POKE SA+107,FNL(SA+140):POKE SA+108 <108>
,FNL(SA+140) <60>
1610 POKE SA+110,FNL(SA+119):POKE SA+111 <154>
,FNL(SA+119) <108>
1620 POKE SA+113,FNL(SA+141):POKE SA+114 <108>
,FNL(SA+141) <108>
1630 POKE SA+116,FNL(SA+120):POKE SA+117 <108>
,FNL(SA+120)

```

```

1640 POKE SA+125,FNL(SA+140):POKE SA+126
, FNH(SA+140) <218>
1650 POKE SA+128,FNL(SA+141):POKE SA+129
, FNH(SA+141) <124>
1660 POKE SA+133,FNL(SA+139):POKE SA+134
, FNH(SA+139) <117>
1670 POKE SA+142,FNL(SA+512):POKE SA+143
, FNH(SA+512) <104>
1800 PRINT:PRINT:PRINT <204>
1810 PRINT"(SPACE)DIE(SPACE)BASICERWEITE
RUNG(SPACE)IST(SPACE)INITIALISIERT" <121>
1900 PRINT:PRINTTAB(10)"ZUM(SPACE)MENUE(S
PACE)MIT(SPACE)RVSON(SPACE)SPACE(RVSOFF)" <206>
1910 GET A$:IF A$<>"(SPACE)" GOTO 1910 <132>
1920 GOTO 3050 <102>
2000 REM <102>
2010 REM ***** LADEN BASICERW ***** <132>
2020 REM <122>
2050 PRINTCHR$(147) <218>
2060 PRINT"(SPACE)RVSON)***** (SPACE)LAD
EN(SPACE)BASICERWEITERUNG(SPACE)***** (RV
SOFF)" <176>
2080 PRINT:PRINT:INPUT"NAME(SPACE)DER(S
PACE)ERWEITERUNG";NA$ <212>
2090 IF NA$="" GOTO 2050 <81>
2100 IF NA$="" GOTO 730 <13>
2110 PRINT:INPUT"(RVSON)K(RVSOFF)ASSETTE
/(RVSON)D(RVSOFF)ISKETTE";GE$ <238>
2120 IF GE$<>"D" AND GE$<>"K" GOTO 2050 <41>
2130 IF GE$="D" THEN GE=8 <208>
2140 IF GE$="K" THEN GE=1 <176>
2150 A=2:LOAD NA$,GE,1 <1>
2160 IFGE=1 THEN SA=PEEK(829)+PEEK(830)*25
6:EA=PEEK(831)+PEEK(832)*256-1:GOTO2210 <154>
2170 EA=PEEK(174)+PEEK(175)*256-1 <63>
2180 OPEN 2,B,2,NA$+",P,R" <200>
2190 GET#2,A$:GET#2,B$:CLOSE 2 <41>
2200 SA=ASC(A$+CHR$(0))+ASC(B$+CHR$(0))*
256 <165>
2210 PRINT:PRINT:PRINT"BASICERWEITERUNG(S
PACE)RVSON)NA$(RVSOFF)SPACE)GELADEN" <148>
2220 PRINT:PRINT"STARTADRESSE:"SA <247>
2230 PRINT:PRINT"BISHERIGE(SPACE)ENDADRE
SSE:"EA <170>
2240 PRINT:INPUT"NEUE(SPACE)ENDADRESSE";
FF <79>
2250 IF FF<EA GOTO 2240 <143>
2260 PRINT:PRINT"BASICERWEITERUNG(SPACE)
BETRIEBSBEREIT" <160>
2270 PRINT:PRINTTAB(10)"ZUM(SPACE)
MENUE(SPACE)MIT(SPACE)RVSON)SPACE(RVSOFF
)" <168>
2280 GET A$:IF A$="(SPACE)" GOTO 3050 <214>
2290 GOTO 2280 <77>
3000 REM <82>
3010 REM ***** BEARBEITEN ***** <247>
3020 REM <102>
3050 PRINTCHR$(147) <197>
3060 PRINT"(SPACE)RVSON)**** (SPACE)BEARB
EITUNG(SPACE)BASICERWEITERUNG(SPACE)****
(RVSOFF)" <122>
3070 PRINT:PRINT"BITTE(SPACE)WAELHEN:" <200>
3080 PRINT:PRINT <236>
3100 PRINT"(1) (SPACE)ANZEIGE(SPACE)BISHE
RIGE(SPACE)BEFEHLE" <120>
3110 PRINT:PRINT"(2) (SPACE)BEFEHL(SPACE)
LOESCHEN" <221>
3120 PRINT:PRINT"(3) (SPACE)BASICERWEITER
UNG(SPACE)SAVEN" <230>
3130 PRINT:PRINT"(4) (SPACE)LADEN(SPACE)N
EUEN(SPACE)BEFEHLS" <215>
3140 PRINT:PRINT"(5) (SPACE)ENDE" <48>
3200 GET A$:IF A$<"1" OR A$>"5" GOTO 320
0 <4>
3220 ON ASC(A$)-48 GOTO 4000,5000,6000,7
000,8000 <2>
4000 REM <62>
4010 REM ***** ANZEIGE ***** <187>
4020 REM <82>
4050 PRINTCHR$(147) <177>
4060 PRINT"(SPACE)RVSON)***** (SPACE)ANZE
IGEN(SPACE)BISHERIGE(SPACE)BEFEHLE(SPACE
)***** (RVSOFF)" <223>
4100 I=1:J=5:K=SA+242:L=SA+142 <217>
4110 POKE 214,J:PRINT:D$="" <158>
4120 D$=D$+CHR$(PEEK(K)AND127) <106>
4130 IF (PEEK(K)AND128)=0 THEN K=K+1:GOT
O 4120 <107>
4140 Z=PEEK(L)+PEEK(L+1)*256:K=K+1 <10>
4150 PRINTD$TAB(25)Z <134>
4170 I=I+1:J=J+1:L=L+2 <128>
4180 IF I>PEEK(SA+141) GOTO 4300 <47>
4190 IF J<20 GOTO 4110 <33>
4200 POKE 214,22:PRINT:PRINTTAB(8)"WEITE
R(SPACE)MIT(SPACE)RVSON)SPACE(RVSOFF)" <47>
4210 GET A$:IF A$<>"(SPACE)" GOTO 4210 <74>
4220 FOR T=5 TO 22:POKE 214,T:PRINT:PRIN
TLL$:NEXT <147>
4260 J=5:GOTO 4110 <93>
4300 POKE214,22:PRINT:PRINTTAB(10)"ZUM(S
PACE)MENUE(SPACE)MIT(SPACE)RVSON)SPACE(R
VSOFF)" <51>
4310 GET A$:IF A$<>"(SPACE)" GOTO 4310 <189>
4320 GOTO 3000 <31>
5000 REM <42>
5010 REM ***** BEFEHL LOESCHEN ***** <67>
5020 REM <62>
5050 PRINTCHR$(147) <157>
5060 PRINT"(SPACES)RVSON)***** (SPACE)BEF
EHL(SPACE)LOESCHEN(SPACE)***** (RVSOFF)" <0>
5100 PRINT:PRINT:INPUT"ZU(SPACE)LOESCHEN
DER(SPACE)BEFEHL";BE$ <255>
5130 IF BE$="" GOTO 5050 <76>
5140 IF BE$="" GOTO 3050 <80>
5170 IFBE$="OUT" THENPRINT:PRINT"GEHT(S
PACE)NICHT!";FORT=1TO3000:NEXT:GOTO5050 <126>
5180 BN$=LEFT$(BE$,LEN(BE$)-1)+CHR$(ASC(
RIGHT$(BE$,1)) OR 128) <182>
5200 I=1:J=SA+242:K=1 <131>
5210 L=J <252>
5220 IF ASC(MID$(BN$,I,1))=PEEK(J) GOTO
5400 <120>
5230 IF PEEK(SA+141)<K GOTO 5300 <29>
5240 IF (PEEK(J)AND128)=0 THEN J=J+1:GOT
O 5240 <253>
5250 J=J+1:I=1:K=K+1:GOTO 5210 <31>
5300 PRINT:PRINT:PRINT"DIESER(SPACE)BEFE
HL(SPACE)IST(SPACE)HIER(SPACE)NICHT(S
PACE)VORHANDEN!" <227>
5310 PRINT:PRINTTAB(10)"ZUM(SPACE)MENUE(S
PACE)MIT(SPACE)RVSON)SPACE(RVSOFF)" <45>
5320 GET A$:IF A$="(SPACE)" GOTO 3050 <194>
5330 GOTO 5320 <42>
5400 IF (PEEK(J)AND 128)=128 GOTO 5500 <84>
5420 J=J+1:I=I+1 <165>
5430 IF I<=LEN(BN$) GOTO 5220 <153>
5440 GOTO 5240 <158>
5500 IF K=PEEK(SA+141) GOTO 5650 <238>
5510 I=J+1:M=K <109>
5520 IF (PEEK(I)AND128)=0 THEN I=I+1:GOT
O 5520 <6>
5530 IF PEEK(SA+141)>M THEN M=M+1:I=I+1:
GOTO 5520 <240>
5540 FOR T=L TO I+L-J-1 <164>

```

```

5550 : POKE T,PEEK(T+J+1-L) <164>
5560 NEXT <79>
5570 FOR T=SA+140+2*K TO SA+139+2*PEEK(S
A+141) <48>
5580 : POKE T,PEEK(T+2) <167>
5590 NEXT <109>
5650 POKE SA+141,PEEK(SA+141)-1 <129>
5660 ZZ=1 <26>
5670 PRINT:PRINT"BEFEHL(SPACE)GELOESCHT.
":PRINT:PRINTTAB(10)"ZUM(SPACE)MENUE(S
PACE)MIT(SPACE)RVSON)SPACE(RVSOFF)" <83>
5680 GET A$:IF A$="(SPACE)" GOTO 3050 <44>
5690 GOTO 5680 <181>
6000 REM <22>
6010 REM ***** SAVEN ***** <67>
6020 REM <42>
6050 PRINTCHR$(147) <137>
6060 PRINT"(SPACE)RVSON)***** (SPACE)SAVE
N(SPACE)BEFEHLERWEITERUNG(SPACE)***** (R
VSOFF)" <9>
6090 PRINT:PRINT:INPUT"PROGRAMMNAME";PN$ <63>
6100 IF PN$="" GOTO 3050 <74>
6110 IF PN$="" OR LEN(PN$)>16 GOTO 6050 <197>
6120 PRINT:PRINT:INPUT"(RVSON)K(RVSOFF)A
SSETTE/(RVSON)D(RVSOFF)ISKETTE";GE$ <50>
6130 IF GE$<>"K" AND GE$<>"D" GOTO 6050 <243>
6140 IF GE$="K" THEN GE=1 <95>
6150 IF GE$="D" THEN GE=8 <148>
6200 POKE 711,GE <21>
6210 POKE 718,LEN(PN$) <193>
6220 POKE 251,FNL(SA):POKE 252,FNH(SA) <116>
6230 POKE 729,FNL(EA+1):POKE 731,FNH(EA+
1) <157>
6240 FOR T=1 TO LEN(PN$):POKE 735+T,ASC(
MID$(PN$,T,1)):NEXT <199>
6260 SYS 710 <235>
6320 ZZ=0 <172>
6500 PRINT:PRINT:PRINT"SAVEN(SPACE)BEEND
ET(SPACE)!" <171>
6510 PRINT:PRINT:PRINTTAB(8)"ZUM(SPACE)M
ENUE(SPACE)MIT(SPACE)RVSON)SPACE(RVSOFF)
" <189>
6520 GET A$:IF A$<>"(SPACE)" GOTO 6520 <178>
6530 GOTO 3050 <222>
7000 REM <2>
7010 REM ***** LADEN NEUER BEFEHL ***** <158>
7020 REM <22>
7050 PRINTCHR$(147) <117>
7060 PRINT"(SPACE)RVSON)***** (SPACE)LADE
N(SPACE)NEUER(SPACE)BEFEHL(SPACE)***** (R
VSOFF)" <71>
7090 IF PEEK(SA+141)=50 GOTO 7160 <122>
7100 J=SA+242 <252>
7110 FOR I=1 TO PEEK(SA+141) <113>
7120 IF (PEEK(J) AND 128)=0 THEN J=J+1:GO
TO7120 <65>
7130 J=J+1 <30>
7140 NEXT <129>
7150 IF SA+511-J>3 GOTO 7190 <116>
7160 PRINT:PRINT"KEIN(SPACE)PLATZ(SPACE)
MEHR!";PRINT:PRINT"ZUM(SPACE)MENUE(S
PACE)MIT(SPACE)RVSON)SPACE(RVSOFF)" <199>
7170 GET A$:IF A$="(SPACE)" GOTO 3050 <4>
7180 GOTO 7170 <126>
7190 PRINT:PRINT:INPUT"ZU(SPACE)LADENDER
(SPACE)BEFEHL";PN$ <53>
7200 IF PN$="" GOTO 3050 <155>
7210 IF PN$="" GOTO 7050 <189>
7220 PRINT:INPUT"(RVSON)K(RVSOFF)ASSETTE
/(RVSON)D(RVSOFF)ISKETTE";GE$ <248>
7230 IF GE$<>"D" AND GE$<>"K" GOTO 7220 <127>
7240 IF GE$="D" THEN GE=8:GOTO 7260 <43>
7250 GE=1 <25>
7260 A=1:LOAD PN$,GE,1 <111>
7270 IF GE=1 THENEN=PEEK(831)+PEEK(832)*
256:SS=PEEK(829)+PEEK(830)*256:GOTO7500 <131>
7280 OPEN 2,B,2,PN$+",P,R" <77>
7290 GET #2,A$:GET#2,B$ <125>
7300 CLOSE 2 <164>
7310 EN=PEEK(174)+PEEK(175)*256 <68>
7320 SS=ASC(A$+CHR$(0))+ASC(B$+CHR$(0))*
256 <221>
7500 IF SS>SA+522 AND EN<=EA GOTO 7600 <15>
7510 IF NOT((SS<SA AND EN<SA)OR SA>EA) G
OTO 7540 <162>
7520 PRINT:PRINT"PRG(SPACE)LIEGT(SPACE)A
USSERHALB(SPACE)DER(SPACE)ERWEITERUNG":G
OTO 7570 <20>
7540 IF SS>SA+522 AND EN>EA THEN PRINT:P
RINT"PRG(SPACE)IST(SPACE)ZU(SPACE)LANG!"
:GOTO 7470 <220>
7550 PRINT:PRINT"BEREICH(SPACE)SA(SPACE)
BIS(SPACE)SA+522(SPACE)WURDE(SPACE)BESCH
RIEBEN" <212>
7570 PRINT:PRINT:PRINTTAB(10)"ZUM(SPACE)
MENUE(SPACE)MIT(SPACE)RVSON)SPACE(RVSOFF
)" <113>
7580 GET A$:IF A$="(SPACE)" GOTO 3050 <159>
7590 GOTO 7580 <41>
7600 PRINT:PRINT:INPUT"BEFEHLSWORT";BW$ <46>
7610 IF SA+511-J<LEN(BW$) THEN PRINT:PRIN
T"ZU(SPACE)LANG":FOR T=1 TO 2000:NEXT:GO
TO 7600 <18>
7700 PP$=LEFT$(BW$,LEN(BW$)-1)+CHR$(ASC(
RIGHT$(BW$,1))OR128) <46>
7710 FOR I=1 TO LEN(PP$):POKE J+I-1,ASC(
MID$(PP$,I,1)):NEXT <99>
7720 POKE SA+142+PEEK(SA+141)*2,FNL(SS) <225>
7730 POKE SA+143+PEEK(SA+141)*2,FNH(SS) <154>
7740 POKE SA+141,PEEK(SA+141)+1 <161>
7750 PRINT:PRINT:PRINT"BEFEHL(SPACE)RVSD
N)BW$(RVSOFF)SPACE)IST(SPACE)EINGEBUN
D." <48>
7760 PRINT:PRINT"STARTADRESSE:"SS <25>
7770 ZZ=1 <96>
7780 PRINT:PRINT:PRINTTAB(10)"ZUM(SPACE)
MENUE(SPACE)MIT(SPACE)RVSON)SPACE(RVSOFF
)" <68>
7790 GET A$:IF A$="(SPACE)" GOTO 3050 <114>
7800 GOTO 7790 <6>
8000 REM <238>
8010 REM ***** ENDE ***** <83>
8020 REM <2>
8050 IF ZZ=0 GOTO 8150 <153>
8100 PRINT:PRINT:PRINT:PRINT <11>
8110 PRINT"AENDERUNGEN(SPACE)SIND(SPACE)
NICHT(SPACE)GESPEICHERT!" <99>
8120 PRINT"TRITZDEM(SPACE)PROGRAMM(S
PACE)BEENDEN(SPACE)(J/N)?" <132>
8130 GET A$:IF A$="N" GOTO 3000 <72>
8140 IF A$<>"J" GOTO 8130 <119>
8150 END <117>
9999 REM <197>

```

Wir hoffen, Ihnen eine nützliche Routine zur Befehlsweiterung vorgestellt zu haben und wünschen Ihnen viel Spaß bei der Anwendung.

Tips und Tricks für den C-64

Zwei nützliche Routinen :

Verbesserter INPUT-Befehl und READ ERROR

Der verbesserte Input-Befehl

Bei den meisten Programmen, die den Input-Befehl verwenden, ist es sehr nützlich und nervenschonend, den Cursor und die Home Taste abzuschalten. Eine Möglichkeit ist die Verwendung von Pokes: **Poke 198.1:Poke 631.34** setzt ein Anführungszeichen in den Tastaturpuffer, Cursor und Home werden dann als inverse Zeichen dargestellt, das ist aber meistens auch nicht erwünscht.

Das folgende Maschinenprogramm bin-

```

50 FOR X=1 TO 53:READA:POKE 709+X,A:B=B+
A:NEXT:IF B=6212 THEN END <218>
70 PRINT "DATA{SPACE}ERROR" <50>
100 DATA 169,230,141,143,2,169,2,141,144
,2 <84>
110 DATA 173,229,2,73,1,141,229,2,208,10
120 DATA 169,72,141,143,2,169,235,141,14
4,2 <203>
130 DATA 96,0,165,203,201,2,240,8,201,7
140 DATA 240,4,201,51,208,4,169,64,133,2
03 <81>
150 DATA 76,72,235 <185>
    
```

det sich in die Tastatur-Dekodieroutine ein und unterdrückt den Cursor und die Hometaste vollkommen.

Mit **SYS 710** wird Cursor und Home abgeschaltet, und nach dem Abschalten auch wieder aktiviert. **SYS 710** sollte also vor und nach jedem Input-Befehl stehen.

```

50 FOR X=1 TO 70:READ A:POKE 827+X,A:NEX
T:IF B=9101 THEN END <247>
100 DATA 169,71,141,143,2,169,3,141,144,
2 <8>
110 DATA 96,173,141,2,201,4,208,49,165,2
03 <72>
120 DATA 201,54,208,43,169,64,133,203,16
9,0 <108>
130 DATA 141,14,220,169,8,32,180,255,169
,111 <186>
140 DATA 133,185,32,150,255,169,0,133,14
4,32 <224>
150 DATA 165,255,32,210,255,165,144,240,
246,32 <185>
160 DATA 171,255,169,1,141,14,220,76,72,
235 <155>
    
```

READ ERROR

ist ein kurzes Maschinenspracheprogramm, das den Fehlerkanal der Floppy liest, wenn man CTRL und die Pfeil-nach-oben-Taste drückt. Durch

SYS 828

wird die Routine in die Tastaturdekodierung eingebunden und mit RUN/STOP-Res-tore wieder abgehängt.

Mit dem Commodore-64 im Meteoritensturm

Weltraumabenteuer

mit tückischen Gefahren und enormen Flugbehinderungen

Sie sind in einem Raumschiff unterwegs, um neue Welten zu entdecken. Plötzlich geraten Sie in einen Meteoritensturm. Ihre Aufgabe ist es nun, das Raumschiff mög-

lichst geschickt durch diese Gefahren zu manövrieren. Stößt der Spieler an einen Meteoriten, ist das Spiel beendet. Je länger die Kapsel im

Weltall unbeschadet bleibt, desto höher der Punktestand.

Gespielt wird mit einem Joystick (Port 2).

Achtung!

Hebel nach links bewirkt, daß die Kapsel sich nach rechts bewegt und umgekehrt. Oben und unten sind allerdings nicht vertauscht.

Nach dem Abtippen des Programms bitte erst absaven und dann normal mit **RUN** starten.

Bei einem **?DATA ERROR** wurden die DATA's falsch abgetippt. Sind die DATA's richtig, erscheint auf dem Bildschirm die **SYS-Adresse**, mit der das Programm gestartet wird, die dann nur noch mit der **RETURN-Taste** bestätigt werden muß. Es erscheint die Abfrage der Spielstärke, die mit den Tasten **F1, F3** und **F5** beantwortet wird.

Jeder weitere Tastendruck startet das Spiel.

```

100 REM ***** <105>
110 REM * * * * * <207>
120 REM * METEORITENSTURM * <196>
130 REM * * * * * <227>
140 REM * COPYRIGHT '86 TRONIC-VERLAG * <172>
150 REM * * * * * <247>
160 REM * 21.03.1986 SWEN OTTMANN * <225>
170 REM * * * * * <11>
180 REM * $C000 BIS $C4C0 MASPRG. * <248>
190 REM * * * * * <31>
200 REM ***** <205>
210 : <12>
220 DATA 3, 255, 0, 63, 255, 192, 255, 2
55, 240, 221 <186>
230 DATA 254, 188, 213, 122, 172, 213, 1
22, 175, 213, 250 <128>
240 DATA 171, 213, 234, 171, 213, 234, 1
71, 213, 234, 171 <207>
250 DATA 245, 250, 171, 53, 126, 175, 53
, 94, 172, 61 <201>
260 DATA 126, 172, 15, 254, 188, 3, 255,
240, 3, 255 <69>
270 DATA 192, 0, 255, 0, 0, 0, 0, 0, 0,
0 <42>
280 DATA 0, 0, 0, 0, 0, 0, 0, 1, 0, 0 <80>
290 DATA 5, 64, 0, 23, 84, 0, 23, 213, 6
4, 95 <226>
300 DATA 253, 80, 95, 175, 212, 94, 170,
245, 94, 170 <60>
310 DATA 245, 95, 175, 212, 95, 253, 80,
23, 213, 64 <106>
320 DATA 23, 84, 0, 5, 64, 0, 1, 0, 0, 0
<124>
330 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
<114>
340 DATA 0, 0, 0, 0, 0, 0, 0, 0, 32, 68
<18>
350 DATA 229, 162, 14, 160, 11, 169, 1,
142, 32, 208 <114>
360 DATA 140, 33, 208, 141, 134, 2, 169,
0, 141, 30 <247>
370 DATA 208, 169, 3, 141, 0, 16, 169, 4
8, 141, 0 <32>
380 DATA 32, 141, 1, 32, 141, 2, 32, 141
    
```

```

, 3, 32 <126>
390 DATA 141, 4, 32, 141, 5, 32, 234, 23
4, 234, 160 <215>
400 DATA 0, 185, 0, 192, 153, 64, 3, 200
, 192, 128 <253>
410 DATA 208, 245, 162, 0, 160, 1, 169,
15, 142, 39 <197>
420 DATA 208, 142, 40, 208, 140, 37, 208
, 141, 38, 208 <202>
430 DATA 169, 3, 141, 21, 208, 162, 13,
160, 14, 142 <58>
440 DATA 248, 7, 140, 249, 7, 162, 170,
160, 190, 142 <106>
450 DATA 2, 208, 140, 3, 208, 169, 255,
141, 28, 208 <101>
460 DATA 165, 203, 201, 64, 208, 3, 76,
240, 192, 32 <119>
470 DATA 68, 229, 162, 60, 160, 140, 142
, 2, 208, 140 <52>
480 DATA 3, 208, 234, 234, 234, 169, 13,
141, 250, 7 <141>
490 DATA 141, 251, 7, 141, 252, 7, 141,
253, 7, 141 <109>
500 DATA 254, 7, 141, 255, 7, 169, 255,
141, 21, 208 <24>
510 DATA 169, 0, 141, 41, 208, 141, 42,
208, 141, 43 <246>
520 DATA 208, 141, 44, 208, 141, 45, 208
, 141, 46, 208 <45>
530 DATA 162, 0, 160, 255, 169, 23, 142,
134, 2, 140 <214>
540 DATA 27, 208, 141, 17, 208, 169, 1,
133, 199, 162 <39>
550 DATA 0, 160, 0, 185, 224, 195, 32, 2
10, 255, 200 <247>
560 DATA 192, 40, 208, 245, 232, 224, 24
, 208, 238, 169 <238>
570 DATA 0, 141, 32, 208, 32, 240, 193,
76, 130, 193 <118>
580 DATA 72, 134, 251, 132, 252, 165, 25
3, 133, 2, 160 <80>
590 DATA 0, 200, 192, 0, 208, 251, 198,
2, 208, 245 <222>
600 DATA 166, 251, 164, 252, 104, 96, 20
6, 0, 208, 206 <255>
610 DATA 4, 208, 206, 6, 208, 206, 8, 20
8, 206, 10 <75>
620 DATA 208, 206, 12, 208, 206, 14, 208
, 32, 104, 193 <235>
630 DATA 173, 0, 220, 41, 1, 201, 0, 240
, 30, 173 <111>
640 DATA 0, 220, 41, 2, 201, 0, 240, 30,
173, 0 <45>
650 DATA 220, 41, 4, 201, 0, 240, 30, 17
3, 0, 220 <39>
660 DATA 41, 8, 201, 0, 240, 30, 76, 229
, 193, 206 <95>
670 DATA 3, 208, 206, 3, 208, 76, 190, 1
93, 238, 3 <15>
680 DATA 208, 238, 3, 208, 76, 190, 193,
238, 2, 208 <137>
690 DATA 238, 2, 208, 76, 190, 193, 206,
2, 208, 206 <187>
700 DATA 2, 208, 76, 190, 193, 32, 64, 1
94, 169, 0 <88>
710 DATA 141, 30, 208, 76, 101, 193, 162
, 170, 160, 170 <151>
720 DATA 142, 0, 208, 140, 1, 208, 162,
255, 160, 130 <125>
730 DATA 142, 4, 208, 140, 5, 208, 162,
205, 160, 107 <235>
740 DATA 142, 6, 208, 140, 7, 208, 162,
255, 160, 220 <20>
750 DATA 142, 8, 208, 140, 9, 208, 162,
    
```

75, 160, 90 <144>
 760 DATA 142, 10, 208, 140, 11, 208, 162 <84>
 , 99, 160, 190 <13>
 770 DATA 142, 12, 208, 140, 13, 208, 162 <124>
 , 111, 160, 120 <129>
 780 DATA 142, 14, 208, 140, 15, 208, 94, <205>
 234, 234, 234 <138>
 790 DATA 234, 234, 234, 234, 234, 234, 1 <227>
 20, 72, 173, 30 <177>
 800 DATA 208, 201, 0, 240, 3, 74, 239, 1 <118>
 94, 32, 86 <79>
 810 DATA 194, 104, 96, 104, 206, 32, 208 <181>
 , 96, 238, 1 <11>
 820 DATA 208, 238, 5, 208, 238, 7, 208, <73>
 238, 9, 208 <208>
 830 DATA 238, 11, 208, 238, 13, 208, 238 <204>
 , 15, 208, 169 <50>
 840 DATA 21, 141, 24, 212, 160, 255, 162 <15>
 , 111, 140, 5 <85>
 850 DATA 212, 142, 6, 212, 162, 6, 160, <61>
 255, 142, 1 <55>
 860 DATA 212, 140, 4, 212, 169, 33, 141, <212>
 4, 212, 238 <125>
 870 DATA 0, 32, 173, 0, 32, 201, 58, 240 <128>
 , 1, 96 <0>
 880 DATA 169, 48, 141, 0, 32, 238, 1, 32 <112>
 , 173, 1 <138>
 890 DATA 32, 201, 58, 240, 1, 96, 169, 4 <11>
 8, 141, 1 <30>
 900 DATA 32, 238, 2, 32, 173, 2, 32, 201 <237>
 , 58, 240 <225>
 910 DATA 1, 96, 169, 48, 141, 2, 32, 238 <94>
 , 3, 32 <127>
 920 DATA 173, 3, 32, 201, 58, 240, 1, 96 <211>
 , 169, 48 <21>
 930 DATA 141, 3, 32, 238, 4, 32, 173, 4, <1090 DATA 169, 4, 76, 92, 195, 234, 234, <224>
 32, 201 <11>
 940 DATA 58, 240, 1, 96, 169, 48, 141, 0 <224>
 , 32, 141 <162>
 950 DATA 1, 32, 141, 2, 32, 141, 3, 32, <25>
 141, 4 <1120 DATA 82, 73, 84, 69, 78, 83, 84, 85

, 82, 77 <99>
 1130 DATA 13, 13, 32, 32, 32, 32, 32, 32 <199>
 , 32, 32 <28>
 1140 DATA 32, 32, 32, 49, 57, 56, 54, 32 <8>
 , 83, 87 <131>
 1150 DATA 69, 78, 32, 79, 84, 84, 77, 65 <88>
 , 78, 78 <171>
 1160 DATA 13, 13, 13, 70, 49, 32, 45, 32 <148>
 , 65, 78 <7>
 1170 DATA 70, 65, 69, 78, 71, 69, 82, 13 <63>
 , 70, 51 <14>
 1180 DATA 32, 45, 32, 70, 79, 82, 84, 71 <24>
 , 69, 83 <175>
 1190 DATA 46, 13, 70, 53, 32, 45, 32, 80 <97>
 , 82, 79 <235>
 1200 DATA 70, 73, 83, 73, 83, 234, 234, <21>
 234, 234, 234 <186>
 1210 DATA 234, 234, 29, 29, 29, 29, 29, <130>
 29, 29, 29 <154>
 1220 DATA 29, 29, 29, 29, 29, 29, 29, 29 <232>
 , 29, 29 <52>
 1230 DATA 29, 29, 29, 29, 29, 29, 29, 29 <209>
 , 29, 29 <52>
 1240 DATA 160, 160, 160, 160, 160, 160, <254>
 160, 160, 160, 160 <80>
 1250 DATA 160, 160, 234, 234, 234, 234, <101>
 234, 234, 234, 234 <183>
 1260 DATA 169, 0, 141, 21, 208, 32, 51, <160>
 196, 234, 234 <163>
 1270 DATA 234, 234, 234, 169, 128, 133, <144>
 253, 234, 234, 234 <177>
 1280 DATA 234, 234, 160, 10, 32, 104, 19 <173>
 3, 136, 208, 250 <223>
 1290 DATA 76, 32, 195, 234, 234, 234, 17 <78>
 3, 5, 32, 141 <150>
 1300 DATA 235, 5, 173, 4, 32, 141, 236, <1>
 5, 173, 3 <1460 PRINT "{DOWN}DATEN{SPACE}SIND{SPACE}<169>
 1310 DATA 32, 141, 237, 5, 173, 2, 32, 1 <167>
 41, 238, 5 <167>
 1320 DATA 173, 1, 32, 141, 239, 5, 173, <80>
 0, 32, 141 <223>
 1330 DATA 240, 5, 96, 234, 234, 234, 234 <78>
 , 234, 234, 255 <150>
 1340 DATA 169, 0, 141, 21, 208, 162, 170 <1690 PRINT "{DOWN}START{SPACE}MIT{SPACE}R<167>
 , 160, 170, 142 <80>
 1350 DATA 0, 208, 140, 1, 208, 162, 255, <80>
 160, 130, 142 <223>
 1360 DATA 4, 208, 140, 5, 208, 162, 205, <78>
 160, 107, 142 <150>
 1370 DATA 6, 208, 140, 7, 208, 162, 255, <1>
 160, 220, 142 <1690 PRINT "{DOWN}START{SPACE}MIT{SPACE}R<167>
 1380 DATA 8, 208, 140, 9, 208, 162, 75, <80>
 160, 90, 142 <223>
 1390 DATA 10, 208, 140, 11, 208, 162, 99 <78>
 , 160, 190, 142 <150>
 1400 DATA 12, 208, 140, 13, 208, 162, 11 <1690 PRINT "{DOWN}START{SPACE}MIT{SPACE}R<167>
 1, 160, 120, 142 <80>
 1410 DATA 14, 208, 140, 15, 208, 169, 0, <223>
 141, 30, 208 <78>
 1420 DATA 141, 31, 208, 76, 128, 192, 0, <150>
 255, 0, 255 <1690 PRINT "{DOWN}START{SPACE}MIT{SPACE}R<167>
 1430 DATA 0, 255, 0, 255, 0, 255, 0, 255 <80>
 , 0, 999 <223>
 1440 : <78>
 1450 CLR: X=49152: E=144827 <150>
 1460 READY: IF Y=999 THEN 1480 <1690 PRINT "{DOWN}START{SPACE}MIT{SPACE}R<167>
 1470 POKEX, Y: P=P+Y: X=X+1: GOTO 1460 <80>
 1480 IF P<>ETHEN PRINT "?DATA{SPACE}ERROR": <119>
 END <1690 PRINT "{DOWN}START{SPACE}MIT{SPACE}R<167>
 1490 PRINT "{DOWN}DATEN{SPACE}SIND{SPACE} <80>
 OK." <223>
 1500 PRINT "{DOWN}START{SPACE}MIT{SPACE}R <78>
 VSON}SYS49952{RVSOFF}" <150>
 1510 PRINT "{DOWN2}SYS49952{UP3}": END <119>

Compute mit

öS 55 sfr 6,50 DM 6,50

COMMODORE

C-16

Heimcomputer

C-16/plus 4

SONDER-

Ausgabe



50 mal können Sie diese Programm-Kassette gewinnen!



alle Programme mit Checksummen

Auch in dieser Ausgabe wieder interessante Beiträge zu den Themen:

- Werkstatt
- Bücher
- Tips & Tricks
- Hardware
- Software-Reviews
- u. v. m.

Sound-Machine - Key-Caller - Führung von Haushaltskonten - Textprogramm
 Pac-Mac - Jäger des verlorenen Schatzes - Oil of Texas - Skateboarding
 Mimi - Jäger der verlorenen Möhre - Soundhelfer - Auto-Starter - Ton-Clock

Neu: Sonderzeichen werden durch Klartext ersetzt!

Eingabe-Hinweise für Commodore C-16/116-Benutzer!

Um Ihnen die Eingabe unserer Listings zu erleichtern, werden zukünftig all C-16/116-Listings ohne Steuerzeichen abgedruckt. Alle Steuerzeichen werden in unseren LISTINGS durch Klartext, welcher die Taste kennzeichnet, ersetzt. Eine Tabelle der Tastenfunktionen finden Sie in jedem Tronic-Magazin.

Tastenbezeichnungen für Steuercodes

(TRONIC-NORM)

| | | | | | |
|-------|----------------|---|-----------|-------------|--|
| DOWN | CURSOR UNTEN | TASTE NEBEN RECHTEM SHIFT | CYAN | TUERKIS | CONTROL-TASTE & 4 |
| UP | CURSOR HOCH | SHIFT-TASTE & TASTE NEBEN RECHTEM SHIFT | PURPLE | PURPUR | CONTROL-TASTE & 5 |
| CLR | CLEAR SCHIRM | SHIFT-TASTE & 2. TASTE GANZ RECHTS OBEN | GREEN | GRUEN | CONTROL-TASTE & 6 |
| INST | EINFUEGEN | SHIFT-TASTE & TASTE GANZ RECHTS OBEN | BLUE | BLAU | CONTROL-TASTE & 7 |
| HOME | CURSOR IN ECKE | 2. TASTE VON GANZ RECHTS OBEN | YELLOW | GELB | CONTROL-TASTE & 8 |
| DEL | DELETE | TASTE GANZ RECHTS OBEN | RVSON | INVERSE EIN | CONTROL-TASTE & 9 |
| RIGHT | CURSOR RECHTS | TASTE GANZ RECHTS UNTEN | RVSOFF | INVERSE AUS | CONTROL-TASTE & 0 |
| LEFT | CURSOR LINKS | SHIFT-TASTE & TASTE UNTEN RECHTS | ORANGE | ORANGE | COMMODORE-TASTE & 1 |
| SPACE | LEERZEICHEN | LEERTASTE (GROESSTE TASTE) | BROWN | BRAUN | COMMODORE-TASTE & 2 |
| F1 | FUNKTIONSTASTE | FUNKTIONSTASTE F1 | LIG.RED | HELLROT | COMMODORE-TASTE & 3 |
| F3 | FUNKTIONSTASTE | FUNKTIONSTASTE F3 | DGREY | DUNKELGRAU | COMMODORE-TASTE & 4 |
| F5 | FUNKTIONSTASTE | FUNKTIONSTASTE F5 | MGREY | MITTELGRAU | COMMODORE-TASTE & 5 |
| F7 | FUNKTIONSTASTE | FUNKTIONSTASTE F7 | LIG.GREEN | HELLGRUEN | COMMODORE-TASTE & 6 |
| F2 | FUNKTIONSTASTE | FUNKTIONSTASTE F2 | LIG.BLUE | HELLBLAU | COMMODORE-TASTE & 7 |
| F4 | FUNKTIONSTASTE | FUNKTIONSTASTE F4 | HGREY | HELLGRAU | COMMODORE-TASTE & 8 |
| F6 | FUNKTIONSTASTE | FUNKTIONSTASTE F6 | CTRL | CONTROL | CONTROL-TASTE ZUSAMMEN MIT DEM NACHFOLGENDEN ZEICHEN. (Z.B. CTRLA ENTSPRICHT CTRL & A) |
| F8 | FUNKTIONSTASTE | FUNKTIONSTASTE F8 | | | |
| BLACK | SCHWARZ | CONTROL-TASTE & 1 | FLASH ON | BLINKEN EIN | CONTROL-TASTE & |
| WHITE | WEISS | CONTROL-TASTE & 2 | FLASH OFF | BLINKEN AUS | CONTROL-TASTE & |
| RED | ROT | CONTROL-TASTE & 3 | | | |

Achtung: Einige Listings enthalten das Zeichen "-", dieses Zeichen entspricht dem Pfeil nach links (Taste ganz links oben) das Zeichen "^" steht für den Pfeil nach oben, (SHIFT/Null) das Zeichen "~" steht für das (engl.) Pfund-Symbol (Taste neben Restore)

Folgendes Beispiel demonstriert die Arbeitsweise unseres Systems:

| | |
|--|-------|
| STANDARD AUSDRUCK: | |
| 10 A\$="TEST":PRINT"LEERZEICHEN ALLOBERLEUTE !":PRINT"TRONIC VERLAG" | |
| 20 PRINT"LEERZEICHEN" | |
| UNSER NEUER AUSDRUCK DER GLEICHEN ZEILEN: | |
| 10 A\$="TEST":PRINT" (CLEAR RIGHT3 DOWN3 R ED)HALLO (RIGHT2 BLUE)LEUTE (SPACE) !":PRINT" (TRONIC (SPACE) VERLAG" | <34> |
| 20 PRINT" (CYAN PURPLE GREEN)ENDE" | <117> |

Achtung!!!!

Kurz vor Redaktionsschluss bemerkten wir, daß die Klartext-Tabelle mit den Schlüsselwörtern beim C-16/116 nicht ganz zutrifft. In erster Linie sind hier die Farben betroffen. Zwar stimmt die Tastenbelegung genau mit den abgedruckten Codes überein, jedoch erscheint bei einigen Tastenkombinationen eine andere Farbe auf dem Bildschirm. Die nachfolgende Gegenüberstellung klärt hier auf:

| Abgedruckter Klartext | Richtige Farbe |
|-----------------------|----------------|
| LIG.RED | Gelb-Grün |
| DGREY | Rosa |
| MGREY | Blau-Grün |
| LIG.GREEN | Hellblau |
| LIG.BLUE | Dunkelblau |
| HGREY | Hellgrün |

Ebenfalls stimmen einige Tastenbeschreibungen nicht ganz überein, da der C-16/116 die Cursorstasten sowie CLR und DEL anders angeordnet hat. Da diese Tasten jedoch beschriftet sind, dürfte es hier keine Probleme geben. In zukünftigen Ausgaben werden wir eine vollständig neue Tabelle abdrucken.

Wie in dem Beispiel zu erkennen ist, wird das Herz-Symbol durch das Wort CLEAR ersetzt. Dies bedeutet, Sie müssen die Taste CLEAR drücken, um das Herz-Symbol zu erzeugen. Sollten Sie einmal nicht wissen, welche Taste gemeint ist, hilft Ihnen ein Blick in unsere Tastentabelle bestimmt weiter.

Um im Listing Tastenbezeichnungen von Grafik oder normalen Texten zu unterscheiden, werden alle Tastenkennzeichnungen in geschweifte Klammern gesetzt. Diese dürfen selbstverständlich nicht eingegeben werden. Auch Leerzeichen innerhalb geschweifeter Klammern

dienen nur zur Trennung einzelner Tastenfunktionen und dürfen ebenfalls nicht eingegeben werden. Um die Tastenfunktionen noch besser hervorzuheben, werden diese in unterstrichener Kursivschrift (Schrägschrift) abgedruckt.

Steht hinter einer Tastenfunktion eine Zahl, welche ebenfalls unterstrichen ist, bedeutet dies, daß die letzte Tastenfunktion mehrmals betätigt werden muß. Die in unserem Beispiel abgedruckte Funktion RIGHT3 bedeutet, daß die Taste RIGHT (Cursor rechts) 3 mal hintereinander betätigt werden muß. Auch einzelne oder mehrere Leer-

zeichen innerhalb von Anführungszeichen werden auf diese Art gekennzeichnet. Das bislang übliche Abzählen einzelner Zeichen entfällt somit völlig.

Alle Zeichen außerhalb von den geschweiften Klammern werden normal abgedruckt und auch eingegeben.

Auf den ersten Blick hört sich das sicher etwas kompliziert an, ist jedoch in der Praxis ganz einfach. Wenn man sich erst an die in Klartext geschriebenen Steuerzeichen gewöhnt hat, wird man den großen Vorteil dieser Schreibweise erkennen.

Checksummer OC Version 1.0 für C-16/116

Die Arbeitsweise des C-16/116 Checksummers entspricht weitgehend der VC-20- und C-64-Version und braucht deshalb nicht näher erklärt zu werden. Da auch der C-16/116 nicht den größten Speicher besitzt, wurde hier die Methode der VC-20-Version gewählt und das Maschinenprogramm in den Kassettenpuffer gelegt. Dies hat den Vorteil, daß kein Basic-Speicher verlorengeht. Der Nachteil besteht darin, daß nach der Aktivierung des Checksummers keine Kassettenoperationen durchgeführt werden dürfen, da diese unweigerlich zum Systemabsturz führen. Um die Kassettenoperationen wieder zuzulassen, muß der Checksummer durch die beiden folgenden SYS-Aufrufe abgeschaltet werden:

SYS 62158 : SYS 33047

Aktiviert wird der Checksummer mit **SYS 818**. Die Funktion und Arbeitsweise entnehmen Sie bitte der C64-Checksummer-Version. Eine Ausnahme bei C-16/116 sind

die zusätzlichen Steuerzeichen 'FLASHON' (CTRL.) und 'FLASHOFF' (CTRL.) welche ebenfalls durch Klartext ersetzt werden (s. Tabelle Tastenbezeichnungen).

Arbeitsweise und Aufbau unseres Checksummers:

Unser Checksummer besteht aus einem kleinen Maschinenprogramm, welches als Basic-Loader abgedruckt ist.

Tippen Sie diesen Loader ein und speichern ihn auf Kassette oder Diskette, denn Sie können ihn zukünftig immer wieder benutzen.

Der Start erfolgt durch den Befehl „RUN“. Nach kurzer Zeit meldet sich der Rechner mit der Meldung „TRONIC ...“. Der Checksummer ist nun aktiv und man kann ein beliebiges Tronic-Listing eingeben. Nachdem eine Zeile mit RETURN abgeschlossen wird, erscheint links oben auf dem

Schirm eine Prüfzahl. Vergleichen Sie diese mit der Zahl im Heft hinter der Zeile. Stimmt die Zahl überein, ist die Zeile richtig eingegeben.

Auf diese Weise können Sie das gesamte Listing schnell und fehlerfrei eingeben.

Interessant ist auch, daß bei der Eingabe von Zeilen die üblichen Abkürzungen benutzt werden können, ohne die Checksumme zu verändern. Leerzeichen außerhalb von Anführungszeichen werden ignoriert, da diese auf die Ausführung der einzelnen Befehle keinen Einfluß haben.

Für alle, die nicht gerne abtippen, ist der Checksummer unter folgender Bestellnummer zu beziehen:

OV 10 K Kassette 10 DM
OV 10 D Diskette 15 DM

```

1 REM ***** <4>
2 REM * * * * * <99>
3 REM * C16-CHECKSUMMER * * * * * <140>
4 REM * * * * * <101>
5 REM * COPYRIGHT BY FRANK BRALL * * * * * <182>
6 REM * * * * * <103>
7 REM * (C) 18.04.86 * * * * * <181>
8 REM * * * * * <105>
9 REM ***** <12>
10 PRINT" (CLEAR DOWN SPACE2) ♦♦♦♦ (SPACE2)
CHECKSUMMER (SPACE) OC (SPACE) 1.0 (SPACE2) ♦♦
♦♦ (DOWN) " <103>
11 PRINT" (SPACE2) ♦♦ (SPACE2) COPYRIGHT (SPA
CE) FRANK (SPACE) BRALL (SPACE3) ♦♦ (DOWN) " <110>
12 PRINT" (SPACE8) FUER (SPACE) TRONIC-VERLA
G" <196>
60000 DIMH(75):FORI=0TO9 <123>
60010 H(48+I)=I:H(65+I)=I+10:NEXT <145>
60020 FORI= 818 TO 1010 :READA$ <151>
60030 H=ASC(LEFT$(A$,1)) <234>
60040 L=ASC(RIGHT$(A$,1)) <253>
60050 D=H(H)*16+H(L):S=S+D:POKEI,D <13>
60060 A=A+1:IFA<20THENNEXT:A=-1 <97>
60070 READV:Z=Z+1:IFV=STHEN60085 <177>
60080 PRINT"DATAFEHLER (SPACE) IN (SPACE) ZE
ILE (SPACE) : ";60200+Z:END <39>
60085 IFA<0THEN60100 <116>
60090 S=0:A=0:NEXT <246>
60100 PRINT" (DOWN2) " <172>
60110 PRINT"CHECKSUMMER (SPACE) EIN (SPACE)
=(SPACE) SYS (SPACE) 818 (DOWN) " <140>
60120 PRINT"CHECKSUMMER (SPACE) AUS (SPACE)
=(SPACE) SYS (SPACE) 62158+SYS (SPACE) 33047 (
DOWN) " <122>
60130 PRINT" (DOWN) SAVE/LOAD (SPACE) IST (SP
ACE) WAEREND (SPACE) DER (SPACE) CHECK-" <61>
60140 PRINT"SUMMER (SPACE) AKTIV (SPACE) IST
, (SPACE) NICHT (SPACE) MOEGLICH (SPACE) !" <221>
60150 PRINT" (DOWN) BEACHTEN (SPACE) SIE (SPA
CE) DIE (SPACE) HINWEISE (SPACE) IN (SPACE) DEN
" <189>
60170 PRINT"HEFTEN (SPACE) COMPUTE (SPACE) M
IT (SPACE) UND (SPACE) COMPUTRONIC" <234>
60180 PRINT" (DOWN) VIEL (SPACE) SPASS (SPACE
)! (SPACE) (AUTOR:F. BRALL/6443SONTRA) <116>
60190 POKE B14,147:POKEB15,251:POKE B16,
147:POKE B17,251 <109>
60195 SYS B18:END <177>
60201 DATA A9,3D,8D,02,03,A9,03,8D,03,03
,60,A2,FF,86,3A,20,5A,8B,86,3B, 1851 <56>
60202 DATA B4,3C,20,73,04,AA,F0,EF,90,09
,20,53,89,20,79,04,4C,D9,8B,20, 2018 <28>
60203 DATA 3E,8E,20,53,89,84,0B,A9,00,8D
,E4,03,8D,E5,03,8D,E6,03,18,A5, 2076 <40>
60204 DATA 14,65,15,8D,E6,03,A0,FF,CB,B1
,3B,F0,2C,C9,22,D0,0A,AD,E4,03, 2508 <18>
60205 DATA 49,01,8D,E4,03,B1,3B,AE,E4,03
,D0,04,C9,20,F0,E4,EE,E5,03,AE, 2644 <87>
60206 DATA E5,03,18,B1,3B,6D,E6,03,8D,E6
,03,CA,D0,F4,4C,76,03,3B,20,39, 2204 <107>
60207 DATA DB,8E,E2,03,8C,E3,03,A2,00,A0
,00,1B,20,39,DB,A9,5B,20,D2,FF, 2365 <115>
60208 DATA A9,00,AE,E6,03,20,5F,A4,A9,5D
,20,D2,FF,20,0B,FE,20,0B,FB,AC, 2380 <169>
60209 DATA E3,03,AE,E2,03,18,20,39,DB,A2
,00,B6,FF,4C,36,87,00,00,00,00, 1778 <133>
60210 DATA 00,00,00,00,00,00,00,00,00,00
,00,00,00, 0 <104>

```


Speichererweiterung für den C-16/116

64K

Sensationell:

- für weniger als 50 DM
- voll kompatibel zum Urzustand durch Umschalter

Die Erweiterung des Speichers auf 64 K-Byte wird am Beispiel des C-116 besprochen. Da aber die Nummern und Anschlüsse der Chips beim C-16 gleich sind und nur die Lage der Bauelemente anders ist, kann auch dieses Gerät nach dieser Anleitung erweitert werden.

Obwohl der Umbau des Rechners an sich einfach ist, sollte sich dennoch jeder Umbauwillige fragen, ob er auch genügend Erfahrung im Löten feinsten Leiterbahnen hat. Abgelöste, verbrannte oder falsch durchtrennte Leiterbahnen versprechen Ärger. Eine doppelseitige Platine zu reparieren, gleicht einem Geduldspiel.

Für Unsichere gibt es einen einfachen und billigen Weg ihre Fähigkeiten zu testen. Löten Sie z. B. einen 7400 Baustein für ca. 2,- DM auf ein Stück Experimentierplatine. Damit haben Sie geübt und Sie wissen sich und Ihre Fähigkeiten einzuschätzen. So sind vier Voraussetzungen zum erfolgreichen Basteln unerlässlich:

1. Ein qualifizierter Umrüster
2. Gutes Werkzeug
3. Ausreichend Zeit
4. Noch mehr Geduld

Folgende Werkzeuge sind zu beschaffen:

- Ein kleiner Elektronik-LötKolben mit feiner Spitze. Leistung ca. 16 Watt
- Elektronik-Lot mit 60% Zinn-Anteil und einem Durchmesser kleiner als 1,5 mm
- Ein kleiner Seitenschneider, möglichst mit planen Schneiden

- Ein feiner Uhrmacher-Schraubenzieher oder ein spitzes scharfes Messer zum Durchtrennen zweier Leiterbahnen
- Ein kleiner Kreuzschlitz-Schraubenzieher
- Ein Bohrer für den Schalterdurchgang
- Eine Entlötpumpe; zur Not tuts auch Entlötlitze

Benötigte Bauteile:

1. Die neuen Speicherelemente: 4 x 64 K dynamische RAMs z. B. von Texas Instruments TMS 4464-15. Dieses Bauteil wird 2 x benötigt.
 2. Ein 2fach UM-Schalter. Vom Einbaugerät sollte es ein Micro-Schalter mit wenig Platzbedarf sein.
 3. Feine Litze für die Verbindungen.
- Die genannten Bauteiletypen sind keineswegs ein Muß. Es wurden lediglich mit diesen Typen Erfahrungen gesammelt. Eine mögliche Bezugsquelle findet sich am Schluß. Der Preis für die beiden Speicherbausteine und den Microschalter liegt unter 50,- DM.

Vorbereitungen:

Bevor Sie anfangen, sich mit dem Innenleben Ihres Rechners zu beschäftigen, sollten Sie stets an einen geerdeten Gegenstand gefaßt haben, um elektrostatische Aufladungen abzuführen. Blitze vom Synthetikpulli sind das Ende des Computers. Mit dem Kreuzschlitz-Schraubenzieher werden nun die vier Gehäuseschrauben

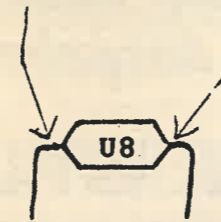
an der Vorder- bzw. Rückseite des Rechners entfernt. Beachten Sie, daß die längeren Schrauben später wieder zum Verschrauben der Rückseite verwendet werden. Heben Sie jetzt vorsichtig das Gehäuse-Oberteil ab. Sie müssen dabei einen kleinen Klemmwiderrstand überwinden. Sie sehen jetzt ein mit Leiterbahnen versehenes Kunststoffband von der Tastatur im Deckel zu einer Buchse in der Platine reichen. Ziehen Sie das Band (vorsichtig!!) aus der Buchse indem Sie beide Hände auf das Abschirmblech auflegen, mit Daumen und Zeigefinger das Band greifen und nach oben herausziehen. Es geht anfangs schwer, aber bitte trotzdem nicht zu heftig reißen.

Nun werden die Halteschrauben für das Abschirmblech herausgedreht. Achten Sie beim Abnehmen des Bleches darauf, daß die heruntergebogene Lasche nicht verbogen wird. Sie dient der Kühlung des TED-Chips. Bei schlechter Auflage sollte die Lasche evtl. mit etwas Wärmeleitpaste bestrichen werden. Nun schrauben Sie noch die Schrauben zur Platinenhalterung heraus. Jetzt haben Sie Ihren Einplatinen-Computer in der Hand.

Der Umbau:

Sehen Sie nun nach vorne rechts. Dort sind zwei längere IC's (18-polig) mit den Nummern U5 und U6. Dies sind die beiden Speicherbausteine, die ausgewechselt werden müssen. Beim Herausnehmen sollte nicht an eine Erhaltung der Bausteine

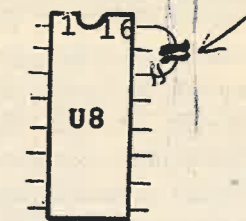
HIER DURCHTRENNEN



Ansicht von vorne

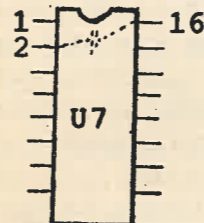
Bild 1

LEITERBAHN UNTERBRECHEN



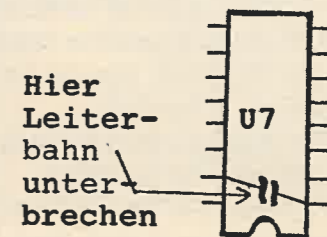
Ansicht von oben (Bauteileseite)

Bild 2



Ansicht von oben (Bauteileseite)

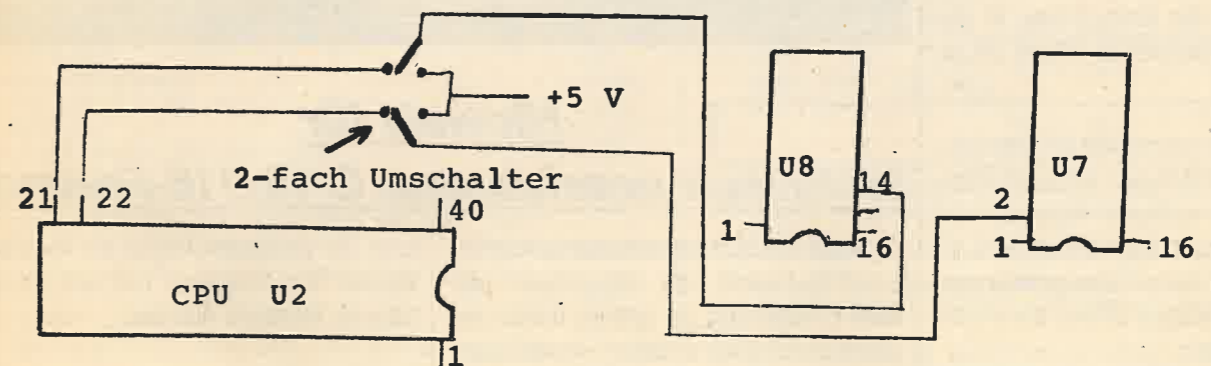
Bild 3a



Hier Leiterbahn unterbrechen

Ansicht von unten (Leiterbahnseite)

Bild 3b



Ansicht von unten (Leiterbahnseite) - Schalter auf 64-K-Stellung

Bild 4

ne gedacht werden. Erstens werden sie nicht mehr benötigt, und zweitens ist die Erhaltung der Platine höchstes Gebot. So knipsen Sie bei den IC's wie auf Bild 1 gezeigt am Gehäuse alle Beinchen ab. Dann werden die Beinchen vorsichtig nach und nach herausgelötet und jeder Lötspunkt anschließend mit der Entlötpumpe oder der Entlötlitze von Lotresten gereinigt. Beim Einlöten brauchen die neuen Bausteine übrigens nur von unten gelötet zu werden, da das Lot nach oben durchdringt. Die Lage der Kerbe im IC muß unbedingt dem auf der Platine aufgedruckten Umriß entsprechen, damit der Speicher richtig gepolt wird.

Nun müssen zwei Verbindungen unterbrochen werden. An U8 ist auf der Bauteil-Seite der Platine die Verbindung zwischen dem Pol 16 und dem Pol 14, wie in Bild 2 dargestellt, zu unterbrechen. An U7 muß auf der Platine-Unterseite die Verbindung zwischen Pol 2 und Pol 16 getrennt werden. Siehe Bilder 3a und 3b. Achtung: Beim Umdrehen ist die Zählrichtung der Anschlüsse gespiegelt (Bild 3b). Verbinden sie jetzt noch entsprechend Bild 4 die CPU (U2) an den Polen 22 und 21 mit dem Schalter und den Polen 2 und 14 von U7 und U8. Den nun noch benötigten Pluspol können Sie beispielsweise an einer Leiterbahn anschließen, die mit Pin 16 von U7 oder U8 verbunden ist.

Es bleibt im Wesentlichen dem Geschmack eines jeden einzelnen überlassen, wo er den Schalter einbaut. Um jedoch übermäßige Kabellängen zu vermeiden, ist es am einfachsten den Schalter direkt gegenüber der Einbuchtung in der Platine bei den Bausteinen U7/U8 einzupassen.

Sollten Sie daran interessiert sein, ein Eingabe/Ausgabe-Interface anzuschließen, wie es in einer späteren Ausgabe von COMPUTE MIT noch besprochen wird, so sollten noch zwei kurze Kabel gelötet werden, um ein unnötiges Öffnen des Rechners zu vermeiden.

Achtung: Der Baustein, an dem die Kabel angelötet werden sollen, hat im C-116 die Nummer U101 und im C-16 die Nummer U16!!!

Verbunden werden muß der Pin 15 von U101 (nicht benutzt) mit Pin Z des Expansion-Ports. Wenn der Rechner mit der Bauteile-Seite nach unten und mit dem Expansionport zu Ihnen auf dem Tisch liegt ist Pin Z der vierte Pin von rechts der Reihe vorne zum Platinenrand. Weiterhin sollte Pin 16 des gleichen Bausteines für eventuelle Erweiterungen auf Pin AA, das ist der dritte Pin von rechts der gleichen Reihe, gelegt werden.

Vor dem Zusammenbau sollten Sie Netz- und Antennen- bzw. Videokabel anschließen und den Rechner einschalten. Der Text sollte jetzt... 60671 BYTES FREE lauten. Wenn nicht: Keine Panik! Legen Sie den neuen Schalter um und drücken Sie die Reset-Taste. Erscheint immer noch nicht das Gewünschte oder etwa ein Wert von ca. 28 kbyte, so muß die Verkabelung überprüft werden. Wenn Ihr Rechner einwandfrei funktioniert, bauen Sie ihn in umgekehrter Reihenfolge wieder zusammen. Vorsicht beim Folienleiterband der Tastatur.

Damit ist der Umbau an der Platine des C-116 beendet.

Ein Hinweis für Insider:

Im Graphic-Modus liegt in der 12kbyte Version das Programm VOR dem Graphikbereich, in der 60kbyte Version wird beim Graphik-Aufruf alles HINTER den Graphikbereich transferiert, auch Maschinenprogramme. In dieser Tatsache liegt der

Bezugsquelle für die Bauteile: Fa. Elektronik Richter, 6200 Wiesbaden, Rheinstraße 85, Tel. 0 61 21 / 37 71 17.

Hinweis für technisch unerfahrene C-16/116-Freunde

Um auch den in Hardware unerfahreneren C-16/116-Freund die Möglichkeit der RAM-Erweiterung zu geben, haben wir uns noch mit einer weiteren Firma zusammengetan. Diese Firma übernimmt für Sie den Umbau Ihres Systems, wenn Sie Ihren Computer einsenden, zu einem Preis von 79,— DM inclusive Bauteile und Arbeits-

In der nächsten Ausgabe:

VERSITALE INTERFACE

(Eingabe - Ausgabe) im Selbstbau

Grund für die Notwendigkeit eines Umschalters auf Originalzustand, z. B. bei einigen Spielen. Um noch einmal darauf hinzuweisen: Es gibt Programme, die mit keiner noch so guten POKE-Kombination zum Laufen gebracht werden können, wenn der Rechner auf 60kbyte-free umgerüstet ist und KEINEN Schalter besitzt, um auf 12kbyte-free umzuschalten. Sollten Sie den Wunsch haben, an Ihrem „Gummi“-Computer eine andere Tastatur anzuschließen, so ist das Parallel zur Originaltastatur möglich. Wenn dafür Interesse besteht, so schreiben Sie uns.

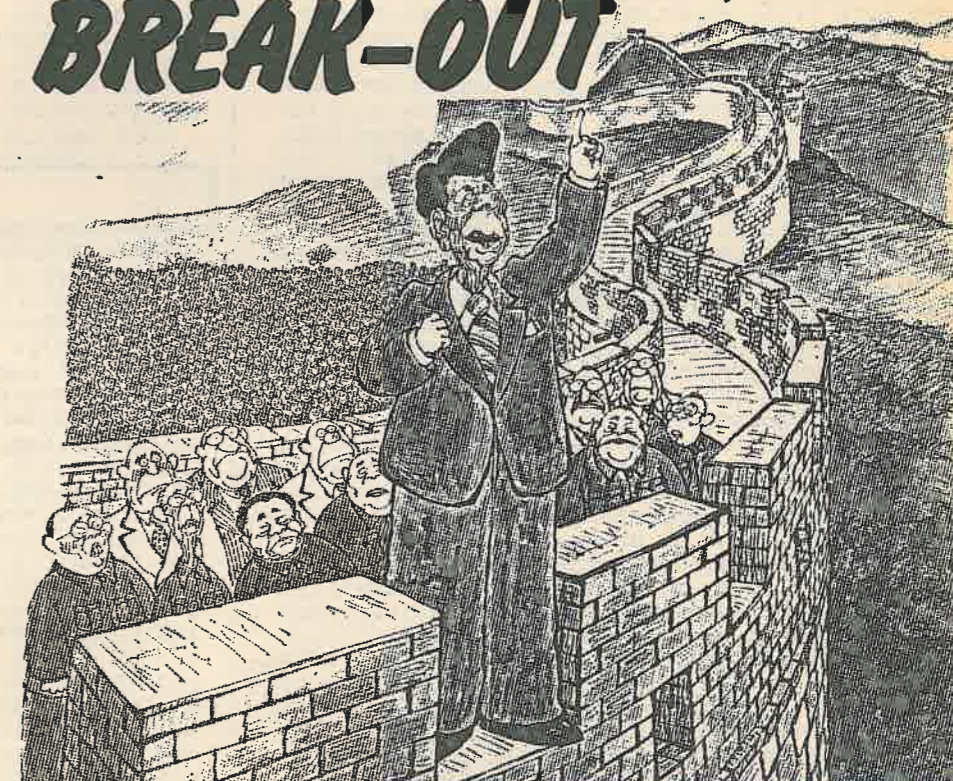
Michael Lausch

lohn. Die Umbauzeit liegt in der Regel bei 1 Woche. Wer Interesse hat, wendet sich bitte an folgende Adresse:

Wolf-Dieter Herr
Bischofsgasse 9
6940 Weinheim
Tel.: 0 62 01 / 6 91 89

Speed ist Trumpf auf dem C-16/116

BREAK-OUT



Bei diesem Spiel ist Geschicklichkeit gefragt. Schlagen Sie mit Hilfe eines Schlägers den Ball gegen eine Mauer aus Steinen. Trifft der Ball einen dieser Steine, werden Ihnen 5 Punkte gutgeschrieben. Bei einem Bonus-Stein sind es sogar 100 Punkte. Trifft der Ball aber das grüne Herz, so springt das Spiel in das nächste Level, in dem die Spielgeschwindigkeit erhöht wird. Insgesamt müssen Sie sich durch 5 Level schlagen. Das Spiel ist vollkommen in Maschinensprache geschrieben. Gespielt wird mit „Z“ = links und „/“ = rechts oder mit Joystick.

Eingabehinweise:

Zunächst Teil 1 abtippen und speichern, dann Teil 2 abtippen und speichern usw. Die Teile möglichst kurz, d. h. ohne Leerzeichen zwischen den Befehlen abtippen. Zum Starten Teil 1 einladen und starten, Teil 2 und Teil 3 werden automatisch nachgeladen und gestartet.

```

1 REM ----- <250>
2 REM -- BREAK-OUT ( C 1 6 ) -- <62>
3 REM ----- <252>
4 REM -- GESCHRIEBEN VON : -- <151>
5 REM -- CARSTEN REINKE -- <167>
6 REM -- 1000 BERLIN 21 -- <89>
7 REM ----- <0>
8 : <66>
10 GRAPHIC 0,1 <144>
20 FORA=1TO18:READB#:B=DEC(B#):FE=FE+B:P
OKEB31+A,B:NEXTA <122>
30 DATA A2,00,BD,00,DO,9D,00,30,BD,00,D1
,9D,00,31,CA,DO,F1,60 <38>
40 IFFE<>2115THENPRINT"DATA-FEHLER<SPACE
>30":STOP:ELSE SYS832 <90>
50 FOR A = 1 TO 17 <234>
60 READ PD <80>
70 PS=12288+8*PD <213>
80 FOR AN = 0 TO 7 <78>
90 READ BY <112>
100 POKE PS+AN,BY:PR=PR+BY <194>
110 NEXT AN,A <81>
120 DATA 0,215,215,85,85,85,85,215,215 <173>
140 DATA 64,171,171,171,255,207,255,243,
255 <112>
150 DATA 65,171,171,171,255,234,234,234,
255 <44>
160 DATA 66,255,119,87,87,84,87,223,207 <142>
170 DATA 67,255,195,195,235,235,215,215,
255 <176>
180 DATA 68,213,90,90,213,213,90,90,213 <77>
190 DATA 69,85,170,170,85,85,150,150,85 <170>
200 DATA 70,87,165,165,87,87,165,165,87 <199>
210 DATA 71,255,252,255,255,85,85,215,21

```

```

5 <158>
220 DATA 72,0,0,0,7,15,31,31,31 <18>
230 DATA 73,0,0,0,255,255,255,255,255 <239>
240 DATA 74,0,0,0,224,240,248,248,248 <255>
250 DATA 75,31,31,31,31,31,31,31,31 <114>
260 DATA 76,248,248,248,248,248,248,248,
248 <72>
270 DATA 79,248,248,248,240,224,0,0,0 <197>
280 DATA 78,255,255,255,255,0,0,0 <81>
290 DATA 77,31,31,31,15,7,0,0,0 <212>
300 IF PR<>19847 THEN PRINT"DATA-FEHLER<
SPACE>120-290":STOP <126>
310 SCNCLR <31>
320 PRINT"{DOWN2}LOAD" <4>
330 PRINT"{HOME}"; <16>
340 POKE 1319,13:POKE1320,82:POKE1321,21
3:POKE 1322,13:POKE 239,4:END <208>
ENDE DES LISTINGS

```

Teil 2:

```

10 POKE52,31:POKE56,31:CLR:SCNCLR <218>
20 FORA=B192TO9200STEP23:FE=0:FORB=0TO22
:READC#:C=DEC(C#):FE=FE+CAND255 <201>
30 POKEA+B,C:NEXTB:READF#:F=DEC(F#):IFF<
>FETHEN40:ELSENEXTA:GOTO50 <132>
40 PRINT"DATA-FEHLER";60+INT((A-B192)/24
):STOP <0>
50 PRINT"{CLEAR DOWN2}LOAD{HOME}";:POKE1
319,13:POKE1320,82:POKE1321,213:POKE1322
,13:POKE239,4:END <121>
60 DATA A9,0B,85,D1,A9,07,85,D3,A9,0B,85
,DO,85,D2,AE,3D,03,E0,00,DO,01,60,A5,E3 <147>
70 DATA DO,18,69,28,90,04,E6,D1,E6,D3,85
,DO,85,D2,CA,4C,11,20,CE,3C,03,60,EE,CB <150>

```


6502-Assembler-Kurs

Teil 14

STA - STORE ACCUMULATOR IN MEMORY (Speichere Akkumulator in Speicher)

Dies ist das Gegenstück zum LDA-Befehl. Er speichert den Inhalt des Akkumulators an einer bestimmten Speicherstelle ab. Wir haben diesen Befehl schon mehrmals in unseren Beispielen verwendet; da er sicherlich neben LDA der wichtigste Befehl des 6502 ist. Er bietet die gleichen Speicher-Adressierungsmöglichkeiten wie der ADC-Befehl mit der Ausnahme, daß eine unmittelbare Adressier-Möglichkeit nicht vorhanden ist.

| Hex | Adressierung | Bytes | Syntax | |
|-----|--------------------------------|-------|--------------|-----------------|
| 81 | Indirekt, vor-indiziert mit X | 2 | STA (addr),x | |
| 85 | Null-Seite (direkt) | 2 | STA addr | |
| 89 | Nicht verwendet | | | |
| 8D | Absolut (direkt) | 3 | STA addr | |
| 91 | Indirekt, nach indiziert mit Y | 2 | STA (addr),y | |
| 95 | Null-Seite indiziert mit X | 2 | STA addr,x | 7 6 5 4 3 2 1 0 |
| 99 | Absolut indiziert mit Y | 3 | STA addr,y | N V B D I Z C |
| 9D | Absolut indiziert mit X | 3 | STA addr,x | nicht beinflußt |

STX - Store Index Register X in Memory (Speichere X-Register in Speicher)

Auch diesen Befehl haben wir schon häufiger in unseren Beispielen angewandt. Er stellt das Gegenstück zu LDX da, und arbeitet so mit genauso wie STA, mit dem Unterschied, daß hier das X-Register im Speicher abgelegt wird.

Beachten Sie, daß es hier keine indizierten Adressier-Arten unter Verwendung des Indexregisters X gibt. Es gibt auch keine absolute indizierte Adressierung. STX und LDX sind die einzigen Befehle, die die indizierte Nullseiten-Adressierung mit Indexregister Y verwendet. Auch hier werden keine Status-Bits beeinflusst.

| Hex | Adressierung | Bytes | Syntax | |
|-----|----------------------------|-------|------------|-----------------|
| 86 | Null-Seite (direkt) | 2 | STX addr | |
| 8E | Absolut (direkt) | 3 | STX addr | 7 6 5 4 3 2 1 0 |
| 96 | Null Seite indiziert mit Y | 2 | STX addr,y | N V B D I Z C |
| 9E | Nicht verwendet | | | nicht beinflußt |

STY - Store Index Register Y in Memory (Speichere Indexregister Y in den Speicher)

Dieser Befehl dient ebenfalls zum Ablegen eines Registers. Er benutzt die gleichen Adressierungsmöglichkeiten wie der STX-Befehl, legt jedoch den Inhalt des Y-Registers im Speicher ab. Die Status-Flags werden auch hier nicht beeinflusst.

| Hex | Adressierung | Bytes | Syntax | |
|-----|----------------------------|-------|------------|-----------------|
| 84 | Null-Seite (direkt) | 2 | STY addr | |
| 8C | Absolut (direkt) | 3 | STY addr | 7 6 5 4 3 2 1 0 |
| 94 | Null Seite indiziert mit X | 2 | STY addr,x | N V B D I Z C |
| 9C | Nicht verwendet | | | nicht beinflußt |

TAX Transfer Accumulator to Index-Register X (Übertrage Inhalt von Akku nach X-Register)

Neben den bekannten Registern „LADEBEFEHLEN“ gibt es beim 6502, sowie beim 6510 noch die sogenannten Transfer-Anweisungen. Diese veranlassen den Prozessor, den Inhalt eines Registers in ein anderes zu übertragen. Das Ursprungsregister wird dadurch nicht beeinflusst. Der TAX-Befehl überträgt den Inhalt des Akkumulators in das X-Register. Transfer-Befehle haben den Vorteil, daß diese keinen Operanten benötigen und dadurch nur 1 Byte belegen. Auch die Ausführungsgeschwindigkeit der Befehle macht sich der geschulte Programmierer oft zu Nutze. In der Praxis werden Transfer-Befehle häufig verwandt um ein Register zwischenspeichernd, vorallem wird hier der Akkumulator gerettet, da er als einziger Arithmetische Operationen durchführen kann.

| Hex | Adressierung | Bytes | Syntax | 7 6 5 4 3 2 1 0 |
|-----|--------------|-------|--------|-----------------|
| AA | Impliziert | 1 | TAX | N V B D I Z C |
| | | | | x x |

TAY Transfer Accumulator to Index-Register Y (Übertrage Akkumulator in Y-Register)

Dieser Befehl arbeitet auf die gleich Weise wie der TAX-Befehl, jedoch wird hier als Ziel das Y-Register verwendet.

| Hex | Adressierung | Bytes | Syntax | 7 6 5 4 3 2 1 0 |
|-----|--------------|-------|--------|-----------------|
| A8 | Impliziert | 1 | TAY | N V B D I Z C |
| | | | | x x |

TSX Transfer Stackpointer to Index Register X (Übertrage Stackzeiger in X-Register)

Auch für den Stackpointer gibt es einen Transfer Befehl. Dieser überträgt den Zeiger, welcher auf den Stack zeigt, in das X-Register. Dadurch läßt sich durch die indizierte Adressierung direkt auf den Stack zugreifen. Die Programmierertechnik sollte jedoch nur von etwas erfahreneren Programmierern genutzt werden, da falsche Stackmanipulationen schnell zum Systemabsturz führen können.

| Hex | Adressierung | Bytes | Syntax | 7 6 5 4 3 2 1 0 |
|-----|--------------|-------|--------|-----------------|
| BA | Impliziert | 1 | TSX | N V B D I Z C |
| | | | | x x |

TXA Transfer Index-Register X to Accumulator (Übertrage X-Register in Akkumulator)

Natürlich besitzt der Prozessor auch Transfer-Befehle in umgekehrter Richtung. Der TXA-Befehl überträgt den Inhalt des X-Register's in den Akkumulator.

| Hex | Adressierung | Bytes | Syntax | 7 6 5 4 3 2 1 0 |
|-----|--------------|-------|--------|-----------------|
| 8A | Impliziert | 1 | TXA | N V B D I Z C |
| | | | | x x |

TXS Transfer Index-Register X to Stackpointer (Übertrage X-Register in Stackzeiger)

Genauso ist die Umkehrung für TSX möglich. Der TXS-Befehl lädt den Stackpointer mit dem Inhalt des X-Registers. Dadurch lassen sich beispielsweise Rücksprungadressen manipulieren oder löschen. Auch dieser Befehl sollte mit Vorsicht verwendet werden.

| Hex | Adressierung | Bytes | Syntax | 7 6 5 4 3 2 1 0 |
|-----|--------------|-------|--------|-----------------|
| 9A | Impliziert | 1 | TXS | N V B D I Z C |
| | | | | nicht beinflußt |

Literatur:

- 6502 Microcomputer-Programmierung,
Peter Heuer, Hofacker Verlag

- 6502 Programmierung in Assembler,

Lance A. Leventhal
- 64 Intern, Angershausen,
Becker, Englisch, Gerits,
Data-Becker Buch

- 6502/65c02, Christian Persson,
Heinz Heise Verlag
- Butterfield's Lehrbuch,
J. Butterfield, Hanser-Verlag

"Poker"



mit dem Schneider CPC 464/664 und 6128

In fast jedem Western oder Kriminalfilm hat dieses Spiel seinen festen Platz und erfreut sich großer Beliebtheit, spannungsgeladene Szenen zu untermauern. Pokern gehört einfach dazu, wenn es darum geht, Nervenkitzel zu erzeugen.

Dieses Nervenkitzeln können Sie nun zusammen mit anderen Mitspielern oder aber allein mit Ihrem Computer erleben. Poker macht aus Ihrer „Hackerstube“ eine „Spielhöhle“.

Gespielt wird Poker ausschließlich mit dem Joystick oder den Cursortasten. Joystick nach oben und unten oder Cursor nach oben und unten wählt aus, ob man mit 52 oder mit 32 Karten spielt. Bei 52 Karten werden generell die Gewinne verdoppelt.

Das bedeuten die Copy- und die Cursortasten:

- Bet:**
 - ▲▼ Einsatz im Bereich von 1-50 verändern
 - Karten werden ausgegeben
- Hold:**
 - ▶◀ zum Steuern des Balkens
 - ▲▼ Karte wird gesetzt bzw. wieder freigegeben
 - freie Karten werden ersetzt
- Next turn:**
 - führt zurück zu 'Bet'
- GEWINN:**
 - Gewinn wird 1-mal ausgegeben
 - ▶◀ führt zu Risiko 1 bzw. Risiko 2
- Risk I:**
 - ▶◀ wählt die Farbe; bei entsprechender Farbe der Zufallskarte wird der Gewinn 2mal ausgegeben, wenn nicht, geht er verloren
- Risk II:**
 - ◀▶ steuert den Balken unter die gewünschte Farbe
 - zieht eine Karte; bei richtiger Wahl wird der Gewinn 4mal ausgegeben, ansonsten geht er verloren

Eine Besonderheit stellt die gute Grafik dieses Programmes dar. Dem Autor ist hier eine gute Version des Pokerspiels geglückt.

```

10 REM *****
15 REM ** SCHNEIDER CPC 464 **
20 REM **-----**
25 REM ** P O K E R (plus) **
30 REM **-----**
35 REM ** W r i t t e n   b y : **
40 REM **
45 REM ** Serge Hommel **
50 REM ** 19,rte.de Wilwerdange **
55 REM ** 9952 Drinklange **
60 REM **
65 REM ** G.D. LUXEMBOURG **
70 REM **-----**
75 REM ** Tel.: (00352)/98701 **
80 REM *****
100
110 REM *****
120 REM ** Titelmelodie - Titelmelodie **
130 REM *****
140 BORDER 0:INK 0,0:INK 4,2:INK 5,26:INK
K 6,6:INK 7,7:INK 8,13:INK 9,9:MODE 0
150 PEN 7:PRINT" HOMULUS - SOFTWARE":LOC
ATE 7,3:PRINT"presents":LOCATE 5,23:PEN
B:PRINT"published by",,"C O M P U T E M
    
```

```

I T";
160 PEN 6:LOCATE 9,19:PRINT STRING$(4,14
3):PEN 5:LOCATE 9,20:PRINT STRING$(4,143
):PEN 4:LOCATE 9,21:PRINT STRING$(4,143)
170 ENT 1,1,-120,1,1,120,1
180 RESTORE 190:FOR t=1 TO 44:READ ton:S
OUND 1,ton,15,15,,1:NEXT
190 DATA 169,179,169,159,169,179,169,213
,169,179,169,142,169,179,169,213,169,179
,169,142,169,179,169,126,169,179,169,142
,169,179,169,213,169,179,169,213,169,179
,169,142,169,179,169,213
200 FOR i=1 TO 500:NEXT
210 FOR i=1 TO 31:INK 1,i
220 PLOT 140,130:DRAW 140,330:DRAW 500,3
30:DRAW 500,270:DRAW 160,270:DRAW 160,13
0:DRAW 140,130:MOVE 160,290:DRAW 160,310
:DRAW 480,310:DRAW 480,290:DRAW 160,290:
MOVE 200,130:DRAW 200,210:DRAW 260,210:D
RAW 260,130:DRAW 200,130:MOVE 220,150
230 DRAW 220,190:DRAW 240,190:DRAW 240,1
50:DRAW 220,150:MOVE 280,130:DRAW 280,25
0:DRAW 300,250:DRAW 300,170:DRAW 340,210
    
```

```

: DRAW 340,190:DRAW 320,170:DRAW 340,150:
DRAW 340,130:DRAW 310,160:DRAW 300,150:D
RAW 300,130:DRAW 280,130:MOVE 360,130
240 DRAW 360,210:DRAW 420,210:DRAW 420,1
70:DRAW 380,170:DRAW 380,150:DRAW 420,15
0:DRAW 420,130:DRAW 360,130:MOVE 380,180
: DRAW 380,200:DRAW 400,200:DRAW 400,180:
DRAW 380,180:MOVE 440,130:DRAW 440,210:D
RAW 460,210:DRAW 460,190:DRAW 500,210
250 DRAW 500,190:DRAW 460,170:DRAW 460,1
30:DRAW 440,130:PEN 1:LOCATE 13,10:PRINT
"PLUS":LOCATE 17,5:PRINT CHR$(64)
260 FOR y=3 TO 14 STEP 2:SOUND 1,(y*i),3
,15:NEXT:NEXT:FOR i=1 TO 1500:NEXT
270 REM *****
280 REM ** Symboldefinition **
290 REM *****
300 SYMBOL AFTER 159:RESTORE 310:FOR i=1
60 TO 238:READ a,b,c,d,e,f,g,h:SYMBOL i,
a,b,c,d,e,f,g,h:NEXT i
310 DATA 204,204,51,51,204,204,51,51,16,
56,124,254,254,16,56,,,108,254,254,254,1
24,56,16,56,56,16,214,254,214,16,56,16,5
6,124,254,124,56,16,,,,,1,3,,,16,16,5
6,124,255,255,,,,,128,15,31,63,63,63,
31,7,3,224,240,248,248,248,240,192,128
320 DATA 1,,,1,3,,,255,214,16,56,255,25
5,,,,,128,,,,,3,7,15,31,31,63,,,131,1
99,239,239,255,255,,,128,224,240,240,24
8,31,31,15,7,1,,,255,255,255,255,255,25
4,254,124,240,240,224,192,,,,,124,56,56,
16,16,,,,,24,60,126,255,255,126,6,15
330 DATA 31,63,63,31,15,6,60,24,153,255,
255,153,24,24,96,240,248,252,252,248,240
,96,,,,,1,,,60,60,60,126,255,255,,,,,16
,16,56,56,124,254,1,3,15,63,15,3,1,,255,
255,255,255,255,255,254,,128,224,248
,224,128,,,8,28,62,127,127,8,28,,,54
340 DATA 127,127,127,62,28,8,28,28,8,107
,127,107,8,28,8,28,62,127,62,28,8,,,121,
59,31,15,11,8,15,,195,231,255,255,102,,2
55,,158,220,248,240,208,16,240,15,8,41,7
4,89,40,8,40,255,,153,219,153,24,24,66,2
40,16,148,210,154,20,16,20,74,82,36,1
350 DATA 24,60,126,223,60,129,153,24,36,
,,129,82,74,36,128,24,60,126,251,143,199
,163,161,176,176,168,172,255,255,255,255
,126,24,24,126,241,227,197,133,13,13,21,
53,172,170,234,234,235,251,251,251,90,12
6,126,90,126,,,117,85,87,215,215,223
360 DATA 223,223,7,15,31,31,31,30,60,59,
255,255,223,143,7,1,,28,128,192,224,224,
240,240,240,112,56,124,116,244,244,244,2
46,242,64,64,64,96,80,,136,113,120,124,1
26,126,126,111,111,199,242,242,227,224,2
36,223,233,18,3,6,24,244,,195,255,255
370 DATA 7,7,7,7,51,251,251,75,36,72,127
,143,135,135,199,255,153,153,165,255,66,
126,68,124,36,18,254,241,225,225,227,255
,255,251,245,251,245,234,245,234,100,60,
36,189,173,185,169,185,255,223,175,223,1
75,87,175,87,,,135,255,73,41,38,16,,124
380 DATA 255,131,69,69,56,,,,,194,254,36,
    
```

```

40,200,16,17,15,8,19,20,19,48,32,255,139
,10,10,136,10,12,8,16,224,160,160,128,16
0,192,128,64,224,32,120,112,64,113,31,12
,4,29,21,52,101,194,,208,80,80,80,64,80,
32,,,63,127,145,245,253,255,124,,7,255
390 DATA 253,85,85,223,249,,224,240,120,
126,253,248,240,62,31,143,207,39,227,17,
240,3,7,143,143,223,254,252,248,224,195,
140,159,32,63,64,127
400 FOR i=1 TO 50:PRINT:NEXT
410 REM *****
420 REM ** Bildschirmaufbau **
430 REM *****
440 as=65:te$="10":hs$=STRING$(3,143)+CH
R$(128)+STRING$(3,143):h$="H O L D":hd$=
"
450 DATA Royal flush,250,Straight flush,
150,Four of a kind,75,Full house,40,Flus
h,20,Straight,8,Three of a kind,3,Two pa
irs,1
460 RESTORE 450:FOR i=1 TO 8:READ typ$(i
),w(i):NEXT
470 MODE 1:INK 1,23:INK 2,6:INK 3,13:PEN
1:sc=100:bet=1:DIM contr(54):DIM f(15)
480 PEN 1:LOCATE 5,10:PRINT CHR$(240);":
As-K-D-B-10-9-8-7-6-5-4-3-2":LOCATE 5,1
2:PRINT CHR$(241);":As-K-D-B-10-9-8-7"
490 IF JOY(0)=1 OR INKEY(0)=0 THEN RAM=1
:g=2:GOTO 520
500 IF JOY(0)=2 OR INKEY(2)=0 THEN RAM=6
:g=1:GOTO 520
510 GOTO 490
520 CLS:WINDOW#6,1,39,11,21:PAPER#6,3:CL
S#6:WINDOW#7,2,38,24,24:PEN#7,1:CLS#7
530 PEN 3:PRINT CHR$(150)STRING$(21,154)
CHR$(156)SPACE$(3)CHR$(150)STRING$(11,15
4)CHR$(156),CHR$(149)SPACE$(21)CHR$(149)
);:PEN 2:PRINT CHR$(150)CHR$(154)CHR$(156
);:PEN 3:PRINT CHR$(149)SPACE$(11)CHR$(1
49)
540 PRINT CHR$(149)SPACE$(21)CHR$(149);:
PEN 2:PRINT CHR$(149)"P"CHR$(149);:PEN 3
:PRINT CHR$(151)STRING$(11,154)CHR$(157)
,CHR$(149)SPACE$(21)CHR$(149);:PEN 2:PRI
NT CHR$(149)"O"CHR$(149);:PEN 3:PRINT CH
R$(149)SPACE$(11)CHR$(149)
550 PRINT CHR$(149)SPACE$(21)CHR$(149);:
PEN 2:PRINT CHR$(149)"K"CHR$(149);:PEN 3
:PRINT CHR$(147)STRING$(11,154)CHR$(153)
,CHR$(149)SPACE$(21)CHR$(149);:PEN 2:PRI
NT CHR$(149)"E"CHR$(149)SPACE$(2);:PEN 3
:PRINT CHR$(150);
560 PRINT STRING$(7,154)CHR$(156),CHR$(1
49)SPACE$(21)CHR$(149);:PEN 2:PRINT CHR$(
149)"R"CHR$(149)SPACE$(2);:PEN 3:PRINT
CHR$(149)SPACE$(7)CHR$(149),CHR$(149)SPA
CE$(21)CHR$(149);:PEN 2:PRINT CHR$(147)C
HR$(154)CHR$(153)SPACE$(2);
570 PEN 3:PRINT CHR$(151)STRING$(7,154)C
HR$(157),CHR$(149)SPACE$(21)CHR$(149)SPA
CE$(5)CHR$(149)SPACE$(7)CHR$(149),CHR$(1
47)STRING$(21,154)CHR$(153)SPACE$(5)CHR$(
147)STRING$(7,154)CHR$(153)
    
```

```

580 LOCATE 1,23:PRINT CHR$(150)STRING$(3
7,154)CHR$(156),CHR$(149)SPACE$(37)CHR$(
149),CHR$(147)STRING$(37,154)CHR$(153);:
FOR i=1 TO 8:LOCATE 2,i+1:PRINT typ$(i):
LOCATE 17,i+1:PRINT":":NEXT
590 PEN 2:LOCATE 28,2:PRINT" S C O R E "
:LOCATE 30,7:PRINT" B E T "
600 PEN 1:LOCATE 30,4:PRINT USING "#####"
#:sc:LOCATE 33,9:PRINT USING "##";bet
610 WINDOW#1,1,7,12,20:WINDOW#2,9,15,12,
20:WINDOW#3,17,23,12,20:WINDOW#4,25,31,1
2,20:WINDOW#5,33,39,12,20
620 REM *****
630 REM ** Steuerprogramm **
640 REM *****
650 FOR i=1 TO 54:contr(i)=0:NEXT:FOR i=
1 TO 5:hold(i)=0:NEXT:hold=0:FOR i=1 TO
33 STEP 8:LOCATE i,22:PRINT hd$:NEXT:CLS
#7
660 FOR i=1 TO 5:PAPER#i,1:PEN#i,3:SOUND
2,240,10,13,0,0,1:CLS#i:PRINT#i,STRING$(
63,127);:NEXT
670 GOSUB 910
680 FOR w=1 TO 5
690 IF hold(w)=1 THEN 780
700 RANDOMIZE TIME
710 ra=INT(RND(6)*5)
720 IF ra<1 OR ra>4 THEN GOTO 710
730 ran=INT(RND(6)*14)
740 IF ran<ran OR ran>13 THEN GOTO 730
750 IF contr(ran*ra)=1 THEN GOTO 710
760 GOSUB 2270
770 contr(ran*ra)=1
780 NEXT
790 IF hold=1 THEN 820
800 GOSUB 1090
810 IF hold=1 THEN 680
820 GOSUB 1280
830 GOSUB 1540
840 IF sc=0 THEN FOR i=1 TO 3000:NEXT:GO
TO 2780
850 CLS#7:PEN#7,1:SOUND 1,1200,30,15:LOC
ATE #7,15,1:PRINT#7,"NEXT TURN"
860 IF JOY(0)<>16 AND INKEY(9)=-1 THEN 8
60
870 IF JOY(0)=16 OR INKEY(9)=0 THEN 650
880 REM *****
890 REM ** Einsatz **
900 REM *****
910 PEN#7,1:LOCATE#7,15,1:PRINT#7,"B E
T"
920 IF bet>=sc THEN bet=sc:sc=0:FOR i=1
TO 10:SOUND 1,40,10,15:SOUND 1,30,5,15:N
EXT:GOTO 1040
930 sc=sc-bet:GOTO 1040
940 PEN 1:LOCATE 33,9:PRINT USING "##";b
et:LOCATE 30,4:PRINT USING "#####";sc
950 PEN 3:FOR i=1 TO 8:LOCATE 18,i+1:PRI
NT USING "#####";k(i):NEXT
960 IF JOY(0)=16 OR INKEY(9)=0 THEN RETU
RN
970 IF bet>=50 OR sc=0 THEN GOTO 990
980 IF JOY(0)=1 OR INKEY(0)=0 THEN bet=b

```

```

et+1:sc=sc-1:SOUND 1,(1040-bet*10),10,15
:GOTO 1040
990 IF bet<=1 THEN GOTO 960
1000 IF JOY(0)=2 OR INKEY(2)=0 THEN bet=
bet-1:sc=sc+1:SOUND 1,(1040-bet*10),10,1
5:GOTO 1040
1010 IF JOY(0)<>1 AND JOY(0)<>2 OR INKEY
(0)+INKEY(2)=-2 THEN GOTO 940
1020 IF bet >50 THEN bet=50
1030 IF bet <1 THEN bet=1
1040 FOR i=1 TO 8:k(i)=w(i)*bet#g:NEXT
1050 GOTO 940
1060 REM *****
1070 REM ** Karten halten **
1080 REM *****
1090 CLS#7:PEN#7,1:PRINT#7,"
H O L D":l=1:hold=1
1100 IF JOY(0)=16 OR INKEY(9)=0 THEN 119
0
1110 PEN#6,0:LOCATE#6,1,11:PRINT#6,hs$;
1120 IF l>25 THEN l=33:GOTO 1140
1130 IF JOY(0)=8 OR INKEY(1)=0 THEN FOR
f=1 TO 100:NEXT:SOUND 1,100,3,15:LOCATE#
6,1,11:PRINT#6,hd$;:l=l+8:GOTO 1110
1140 IF l<8 THEN l=1:GOTO 1160
1150 IF JOY(0)=4 OR INKEY(8)=0 THEN FOR
f=1 TO 100:NEXT:SOUND 1,100,3,15:LOCATE#
6,1,11:PRINT#6,hd$;:l=l-8:GOTO 1110
1160 IF JOY(0)=1 OR INKEY(0)=0 THEN PEN
2:SOUND 1,200,10,15:LOCATE 1,22:PRINT h$
:hold((1+7)/8)=1
1170 IF JOY(0)=2 OR INKEY(2)=0 THEN LOCA
TE 1,22:SOUND 1,400,10,15:PRINT hd$:hold
((1+7)/8)=0
1180 GOTO 1100
1190 LOCATE#6,1,11:PRINT#6,hd$;
1200 GOSUB 1220
1210 RETURN
1220 FOR i=1 TO 5:IF hold(i)=0 THEN PEN#
i,3:CLS#i:SOUND 2,240,10,13,0,0,1:PRINT#
i,STRING$(63,127);
1230 NEXT
1240 RETURN
1250 REM *****
1260 REM ** Auswertung **
1270 REM *****
1280 aa=aa(1):ab=aa(2):ac=aa(3):ad=aa(4)
:ae=aa(5):ba=bb(1):bb=bb(2):bc=bb(3):bd=
bb(4):be=bb(5):sa=aa+ab+ac+ad+ae:maa=MAX
(aa,ab,ac,ad,ae):mia=MIN(aa,ab,ac,ad,ae)
1290 IF ba=bb AND bb=bc AND bc=bd AND bd
=be THEN GOTO 1400
1300 IF (aa=ab AND ab=ac AND ac=ad) OR (
aa=ab AND ab=ac AND ac=ae) OR (aa=ab AND
ab=ad AND ad=ae) OR (aa=ac AND ac=ad AN
D ad=ae) OR (ab=ac AND ac=ad AND ad=ae)
THEN typ=3:GOTO 1430
1310 IF (aa=ab AND ac=ad AND ad=ae) OR (
aa=ac AND ab=ad AND ad=ae) OR (aa=ad AND
ab=ac AND ac=ae) OR (aa=ae AND ab=ac AN
D ac=ad) OR (aa=ac AND ac=ae AND ab=ad)
OR (aa=ac AND ac=ad AND ab=ae) THEN typ=
4:GOTO 1430

```

```

1320 IF (ab=ac AND aa=ad AND ad=ae) OR (
aa=ab AND ab=ae AND ac=ad) OR (aa=ab AND
ab=ad AND ac=ae) OR (aa=ab AND ab=ac AN
D ad=ae) THEN typ=4:GOTO 1430
1330 IF (aa=ab AND (ab=ac OR ab=ad OR ab
=ae)) OR (aa=ac AND (ac=ad OR ac=ae)) OR
(ad=ae AND (ad=aa OR ad=ab OR ad=ac)) O
R (ab=ac AND (ac=ad OR ac=ae)) THEN typ=
7:GOTO 1430
1340 IF aa<> ab AND aa<>ac AND aa<>ad AN
D aa<>ae THEN GOTO 1360
1350 IF (aa=ab AND (ac=ad OR ac=ae OR ad
=ae)) OR (aa=ac AND (ab=ad OR ab=ae OR a
d=ae)) OR (aa=ad AND (ab=ac OR ab=ae OR
ac=ae)) OR (aa=ae AND (ab=ac OR ab=ad OR
ac=ad)) THEN typ=8:GOTO 1430
1360 IF (ab=ac AND ad=ae) OR (ab=ad AND
ac=ae) OR (ab=ae AND ac=ad) THEN typ=8:G
OTO 1430
1370 IF aa=ab OR aa=ac OR aa=ad OR aa=ae
OR ab=ac OR ab=ad OR ab=ae OR ac=ad OR
ac=ae OR ad=ae OR (maa-mia)<>4 THEN GOTO
1390
1380 IF sa>10 AND sa<60 AND (sa/5-INT(sa
/5))=0 THEN typ=6
1390 GOTO 1430
1400 IF sa=55 AND maa-mia=4 THEN typ=1:G
OTO 1430
1410 IF sa>10 AND sa<55 AND sa/5-INT(sa/
5)=0 AND maa-mia=4 THEN typ=2:GOTO 1430
1420 typ=5
1430 IF typ=0 THEN 1500
1440 RESTORE 1450:FOR t=1 TO 12:READ ton
:SOUND 1,ton,10,15:NEXT
1450 DATA 268,225,1,225,1,253,225,201,22
5,253,225,225
1460 a$=typ$(typ)
1470 win=k(typ)
1480 PEN 2:LOCATE 2,(typ+1):PRINT a$:LOC
ATE 18,typ+1:PRINT USING "#####";win
1490 GOSUB 1720
1500 RETURN
1510 REM *****
1520 REM ** Gewinn ausgeben **
1530 REM *****
1540 IF tim=0 THEN GOTO 1670
1550 IF tim=1 THEN GOTO 1580
1560 DATA 478,15,,5,358,10,,20,358,20,,3
,301,10,,20,301,20,,5,239,35,,10,301,40
1570 RESTORE 1560:FOR y=1 TO 13:READ i,1
:SOUND 1,i,1,15:NEXT
1580 PEN 2:LOCATE 18,typ+1:PRINT USING "
#####";win:LOCATE 17,1:PRINT USING "#";t
im:FOR i=1 TO 1250:NEXT
1590 ti=tim-1
1600 FOR y=1 TO tim
1610 wi=win
1620 FOR p=1 TO 100:NEXT
1630 FOR i=bet TO wi STEP bet
1640 wi=wi-bet:sc=sc+bet:SOUND 1,45,3,15
1650 PEN 2:LOCATE 18,typ+1:PRINT USING "
#####";wi:LOCATE 17,1:PRINT USING "#";ti
:PEN 1:LOCATE 30,4:PRINT USING "#####";

```

```

sc
1660 NEXT:ti=ti-1:NEXT
1670 PEN 3:LOCATE 2,typ+1:PRINT a$;
1680 IF typ=0 THEN 1700
1690 LOCATE 18,typ+1:PRINT USING "#####"
;k(typ)
1700 LOCATE 17,1:PRINT CHR$(154);:tim=0:
typ=0:a$=""
1710 RETURN
1720 rr=0:CLS#7:PEN#7,1:PRINT#7,"Take Wi
n (X), Double "CHR$(242)" or Quadruple "
CHR$(243);
1730 IF JOY(0)<4 AND INKEY(8)<>0 AND INK
EY(1)<>0 AND INKEY(9)<>0 THEN 1730
1740 IF JOY(0)=16 OR INKEY(9)=0 THEN tim
=1:GOSUB 1540:GOTO 1780
1750 IF JOY(0)=4 OR INKEY(8)=0 THEN rr=2
:GOTO 1770
1760 IF JOY(0)=8 OR INKEY(1)=0 THEN rr=4
:GOTO 1770
1770 GOSUB 1820
1780 RETURN
1790 REM *****
1800 REM ** Risiko **
1810 REM *****
1820 CLS#6:CLS#3:PEN#3,3:SOUND 2,240,10,
13,0,0,1 :PRINT#3,STRING$(63,127);
1830 FOR i=1 TO 33 STEP 8:LOCATE i,22:PR
INT hd$:NEXT
1840 RESTORE 1850:FOR i=1 TO 7:READ t,1:
SOUND 1,t,1,15:NEXT
1850 DATA 638,10,,20,638,15,,5,638,30,,2
,758,60
1860 IF rr=2 THEN GOSUB 1920
1870 IF rr=4 THEN GOSUB 2030
1880 RETURN
1890 REM *****
1900 REM ** Risiko I **
1910 REM *****
1920 CLS#7:PEN#7,2:PRINT#7,"
R I S K I"
1930 CLS#2:PEN#2,0:PRINT#2,STRING$(63,14
3);:CLS#4:PEN#4,2:PRINT#4,STRING$(63,143
);
1940 IF (JOY(0)<4 OR JOY(0)>8) AND INKEY
(8)<>0 AND INKEY(1)<>0 THEN 1940
1950 IF JOY(0)=4 OR INKEY(8)=0 THEN rp=0
:GOTO 1970
1960 IF JOY(0)=8 OR INKEY(1)=0 THEN rp=2
:GOTO 1970
1970 GOSUB 2210
1980 IF rp=p THEN tim=2 ELSE tim=0:win=0
:FOR i=80 TO 1500 STEP 40:SOUND 1,i,5,15
:NEXT
1990 RETURN
2000 REM *****
2010 REM ** Risiko II **
2020 REM *****
2030 CLS#7:PEN#7,2:PRINT#7,"
R I S K II":as=128
2040 ran=13:ra=0:FOR w=1 TO 5:ra=ra+1:ON
w GOSUB 2270,2270,2170,2270,2270:NEXT
2050 PEN 2:l=17:rb=3:as=65

```

```

2060 IF JOY(0)=16 OR INKEY(9)=0 THEN GOT
0 2130
2070 LOCATE 1,22:PRINT hs$
2080 IF 1>25 THEN 1=33:GOTO 2100
2090 IF JOY(0)=8 OR INKEY(1)=0 THEN FOR
f=1 TO 100:NEXT:LOCATE 1,22:PRINT hd$;:1
=1+8:rb=rb+1:GOTO 2060
2100 IF 1<8 THEN 1=1:GOTO 2060
2110 IF JOY(0)=4 OR INKEY(8)=0 THEN FOR
f=1 TO 100:NEXT:LOCATE 1,22:PRINT hd$;:1
=1-8:rb=rb-1:GOTO 2060
2120 GOTO 2060
2130 IF 1=17 THEN 2060
2140 IF rb>3 THEN rb=rb-1
2150 GOSUB 2210
2160 IF rb=bb(3) THEN tim=4 ELSE tim=0:wi
n=0:FOR i=80 TO 1500 STEP 40:SOUND 1,i,5
,15:NEXT:RETURN
2170 ra=ra-1:RETURN
2180 REM *****
2190 REM ** Karten auswahlen **
2200 REM *****
2210 RANDOMIZE TIME
2220 ra=INT(RND(6)*5)
2230 IF ra<1 OR ra>4 THEN GOTO 2220
2240 ran=INT(RND(6)*14)
2250 IF ran<ran OR ran>13 THEN GOTO 2240
2260 w=3:GOSUB 2270:RETURN
2270 bb(w)=ra:aa(w)=ran:s=160+bb(w):z=18
9+bb(w)
2280 IF (bb(w)/2)-INT(bb(w)/2)=0 THEN p=
2 ELSE p=0
2290 CLS#w:PEN#w,p:SOUND 2,240,10,13,0,0
,1
2300 ON ran GOSUB 2600,2600,2620,2620,26
20,2640,2640,2420,2430,2660,2670,2680,26
90
2310 RETURN
2320 REM *****
2330 REM ** Matrice **
2340 REM *****
2350 REM ** 2,3 **
2360 PRINT#w,CHR$(v) SPACE$(5) CHR$(v) SPAC
E$(10) CHR$(s) ,SPACE$(10) CHR$(y) ,SPACE$(1
0) CHR$(s) SPACE$(4) ,CHR$(v) SPACE$(5) CHR$(
v);:RETURN
2370 REM ** 4,5,6 **
2380 PRINT#w,CHR$(v) SPACE$(5) CHR$(v) SPAC
E$(8) CHR$(s) SPACE$(3) CHR$(s) SPACE$(8) CHR
$(y) CHR$(128) CHR$(x) CHR$(128) CHR$(y) SPAC
E$(8) CHR$(s) SPACE$(3) CHR$(s) SPACE$(7) CHR
$(v) SPACE$(5) CHR$(v);:RETURN
2390 REM ** 7,8 **
2400 PRINT#w,CHR$(v) SPACE$(5) CHR$(v) SPAC
E$(8) CHR$(s) SPACE$(3) CHR$(s) SPACE$(3) CHR
$(s) ,CHR$(128) CHR$(s) SPACE$(3) CHR$(s) ,SP
ACE$(3) CHR$(y) ,CHR$(128) CHR$(s) SPACE$(3)
CHR$(s) SPACE$(7) CHR$(v) SPACE$(5) CHR$(v);
:RETURN
2410 REM ** 9 **
2420 PRINT#w,CHR$(57);SPACE$(5);CHR$(57)
;CHR$(128);CHR$(s);SPACE$(3);CHR$(s),SPA
CE$(8);CHR$(s);SPACE$(3);CHR$(s),SPACE$(

```

```

3);CHR$(s),CHR$(128);CHR$(s);SPACE$(3);C
HR$(s),SPACE$(8);CHR$(s);SPACE$(3);CHR$(
s),CHR$(57);SPACE$(5);CHR$(57);:RETURN
2430 REM ** 10 **
2440 PRINT#w,te$;SPACE$(3);te$;CHR$(128)
CHR$(s)SPACE$(3)CHR$(s),SPACE$(3)CHR$(s)
,CHR$(128)CHR$(s)SPACE$(3)CHR$(s),SPACE$(
8)CHR$(s)SPACE$(3)CHR$(s),SPACE$(3)CHR$(
s),CHR$(128)CHR$(s)SPACE$(3)CHR$(s),te$
;SPACE$(3);te$;:RETURN
2450 REM ** A **
2460 PRINT#w,CHR$(as)SPACE$(5)CHR$(as),S
PACE$(2)CHR$(a(1))CHR$(a(2))CHR$(a(3)),S
PACE$(2)CHR$(a(4))CHR$(a(5))CHR$(a(6)),S
PACE$(2)CHR$(a(7))CHR$(a(8))CHR$(a(9))SP
ACE$(14)CHR$(as)SPACE$(5)CHR$(as);:RETUR
N
2470 REM ** B,D,K **
2480 PRINT#w,CHR$(b)CHR$(128)CHR$(s)CHR$(
128)CHR$(z)CHR$(128)CHR$(b):PRINT#w,CHR
$(z)CHR$(128)CHR$(f(1))CHR$(f(2))CHR$(f(
3))CHR$(128)CHR$(s)SPACE$(2)CHR$(f(4))CH
R$(f(5))CHR$(f(6)),CHR$(z)CHR$(128)CHR$(
f(7))CHR$(f(8));
2490 PRINT#w,CHR$(f(9))CHR$(128)CHR$(s)S
PACE$(2)CHR$(f(10))CHR$(f(11))CHR$(f(12)
),CHR$(z)CHR$(128)CHR$(f(13))CHR$(f(14))
CHR$(f(15))CHR$(128)CHR$(s):PRINT#w,CHR$(
b)CHR$(128)CHR$(s)CHR$(128)CHR$(z)CHR$(
128)CHR$(b);:RETURN
2500 REM *****
2510 REM ** Variablen fuer Matrice **
2520 REM *****
2530 DATA 165,166,167,168,143,169,170,17
1,172
2540 DATA 173,174,175,176,177,178,128,17
9,128
2550 DATA 128,180,128,181,182,183,184,18
5,172
2560 DATA 128,186,128,187,188,189,128,17
9,128
2570 DATA 224,225,226,227,228,229,230,23
1,232,233,234,235,236,237,238
2580 DATA 209,210,211,212,213,214,215,21
6,217,218,219,220,221,222,223
2590 DATA 194,195,196,197,198,199,200,20
1,202,203,204,205,206,207,208
2600 IF aa(w)=1 THEN v=50:y=128 ELSE v=5
1:y=s
2610 GOSUB 2360:RETURN
2620 IF aa(w)=3 THEN v=52:x=128:y=128 EL
SE IF aa(w)=4 THEN v=53:x=s:y=128 ELSE v
=54:y=s:x=128
2630 GOSUB 2380:RETURN
2640 IF aa(w)=6 THEN v=55:y=128 ELSE v=5
6:y=s
2650 GOSUB 2400:RETURN
2660 RESTORE 2570:FOR d=1 TO 15:READ f(d
):NEXT:b=66:GOSUB 2480:RETURN
2670 RESTORE 2580:FOR d=1 TO 15:READ f(d
):NEXT:b=68:GOSUB 2480:RETURN
2680 RESTORE 2590:FOR d=1 TO 15:READ f(d
):NEXT:b=75:GOSUB 2480:RETURN

```

```

2690 ON ra GOSUB 2710,2720,2730,2740
2700 RETURN
2710 RESTORE 2530:FOR d=1 TO 9:READ a(d)
:NEXT:GOSUB 2460:GOTO 2700
2720 RESTORE 2540:FOR d=1 TO 9:READ a(d)
:NEXT:GOSUB 2460:GOTO 2700
2730 RESTORE 2550:FOR d=1 TO 9:READ a(d)
:NEXT:GOSUB 2460:GOTO 2700
2740 RESTORE 2560:FOR d=1 TO 9:READ a(d)
:NEXT:GOSUB 2460:GOTO 2700
2750 REM *****
2760 REM ** Programmende **
2770 REM *****
2780 MODE 0:INK 0,0:INK 6,0:INK 7,0:INK
8,0:PEN 6
2790 FOR y=1 TO 13
2800 FOR i=1 TO 10:PRINT CHR$(128);CHR$(
143);:NEXT

```

```

2810 FOR i=1 TO 10:PRINT CHR$(143);CHR$(
128);:NEXT
2820 NEXT
2830 INK 0,7,15:INK 6,9,1:INK 7,1,3
2840 WINDOW#7,5,16,11,15:PAPER#7,8:CLS#7
:PEN#7,7:PRINT#7:PRINT#7:PRINT#7," YOU B
USTED"
2850 FOR i=20 TO 1000 STEP 5:SOUND 1,i,5
,15:NEXT:FOR i=1 TO 500:NEXT
2860 MODE 1:INK 0,0:CLS
2870 LOCATE 9,10:PRINT"Do you want a new
game?"
2880 LOCATE 18,14:PRINT"[Y/N]"
2890 yn$=INKEY$:IF yn$="" THEN GOTO 2890
2900 IF UPPER$(yn$)="Y" THEN RUN 440
2910 IF UPPER$(yn$)="N" THEN INK 1,24:PE
N 1:CLS:END
2920 GOTO 2890

```

| ZEILENNR.: | | SUMMEN | | | | | | | | | |
|------------|------|--------|-------|-------|--------|-------|-------|--------|-------|-------|--------|
| 10- | 15: | 44541 | 570- | 580: | 27738 | 1210- | 1220: | 35274 | 1850- | 1860: | 50314 |
| 20- | 25: | 47820 | 590- | 600: | 56382 | 1230- | 1240: | 377 | 1870- | 1880: | 13882 |
| 30- | 35: | 46305 | 610- | 620: | 56347 | 1250- | 1260: | 20676 | 1890- | 1900: | 13908 |
| 40- | 45: | 23864 | 630- | 640: | 31860 | 1270- | 1280: | 54668 | 1910- | 1920: | 26353 |
| 50- | 55: | 63185 | 650- | 660: | 116764 | 1290- | 1300: | 79807 | 1930- | 1940: | 41355 |
| 60- | 65: | 27239 | 670- | 680: | 7226 | 1310- | 1320: | 116940 | 1950- | 1960: | 100222 |
| 70- | 75: | 43057 | 690- | 700: | 19879 | 1330- | 1340: | 94794 | 1970- | 1980: | 34360 |
| 80- | 100: | 24018 | 710- | 720: | 46888 | 1350- | 1360: | 92345 | 1990- | 2000: | 6698 |
| 110- | 120: | 64812 | 730- | 740: | 71384 | 1370- | 1380: | 83255 | 2010- | 2020: | 15728 |
| 130- | 140: | 76044 | 750- | 760: | 53645 | 1390- | 1400: | 23912 | 2030- | 2040: | 78716 |
| 150- | 160: | 16800 | 770- | 780: | 29369 | 1410- | 1420: | 29537 | 2050- | 2060: | 62065 |
| 170- | 180: | 21163 | 790- | 800: | 11993 | 1430- | 1440: | 18556 | 2070- | 2080: | 31170 |
| 190- | 200: | 21191 | 810- | 820: | 13573 | 1450- | 1460: | 67550 | 2090- | 2100: | 40615 |
| 210- | 220: | 46096 | 830- | 840: | 47727 | 1470- | 1480: | 68919 | 2110- | 2120: | 23437 |
| 230- | 240: | 32180 | 850- | 860: | 94963 | 1490- | 1500: | 1216 | 2130- | 2140: | 38358 |
| 250- | 260: | 79648 | 870- | 880: | 31683 | 1510- | 1520: | 33256 | 2150- | 2160: | 3647 |
| 270- | 280: | 38207 | 890- | 900: | 13594 | 1530- | 1540: | 25407 | 2170- | 2180: | 26619 |
| 290- | 300: | 14990 | 910- | 920: | 79015 | 1550- | 1560: | 69409 | 2190- | 2200: | 40247 |
| 310- | 320: | 114511 | 930- | 940: | 36039 | 1570- | 1580: | 16122 | 2210- | 2220: | 15822 |
| 330- | 340: | 106444 | 950- | 960: | 51272 | 1590- | 1600: | 25170 | 2230- | 2240: | 48834 |
| 350- | 360: | 126909 | 970- | 980: | 77465 | 1610- | 1620: | 18424 | 2250- | 2260: | 62909 |
| 370- | 380: | 32342 | 990- | 1000: | 54281 | 1630- | 1640: | 59839 | 2270- | 2280: | 103016 |
| 390- | 400: | 59668 | 1010- | 1020: | 49646 | 1650- | 1660: | 35627 | 2290- | 2300: | 59797 |
| 410- | 420: | 37750 | 1030- | 1040: | 55452 | 1670- | 1680: | 33749 | 2310- | 2320: | 5312 |
| 430- | 440: | 29494 | 1050- | 1060: | 10726 | 1690- | 1700: | 55539 | 2330- | 2340: | 13497 |
| 450- | 460: | 45917 | 1070- | 1080: | 27174 | 1710- | 1720: | 22125 | 2350- | 2360: | 53892 |
| 470- | 480: | 56455 | 1090- | 1100: | 46780 | 1730- | 1740: | 23751 | 2370- | 2380: | 6206 |
| 490- | 500: | 127778 | 1110- | 1120: | 46398 | 1750- | 1760: | 104184 | 2390- | 2400: | 54808 |
| 510- | 520: | 59544 | 1130- | 1140: | 22463 | 1770- | 1780: | 597 | 2410- | 2420: | 7452 |
| 530- | 540: | 68838 | 1150- | 1160: | 16862 | 1790- | 1800: | 11489 | 2430- | 2440: | 4351 |
| 550- | 560: | 16359 | 1170- | 1180: | 4405 | 1810- | 1820: | 5675 | 2450- | 2460: | 28775 |
| | | | 1190- | 1200: | 16454 | 1830- | 1840: | 54124 | 2470- | 2480: | 67687 |

Checksummen

| | | |
|-------------------|-------|--------|
| 2490- | 2500: | 46588 |
| 2510- | 2520: | 61132 |
| 2530- | 2540: | 70709 |
| 2550- | 2560: | 70946 |
| 2570- | 2580: | 55235 |
| 2590- | 2600: | 77153 |
| 2610- | 2620: | 9358 |
| 2630- | 2640: | 52025 |
| 2650- | 2660: | 24858 |
| 2670- | 2680: | 46116 |
| 2690- | 2700: | 21364 |
| 2710- | 2720: | 130780 |
| 2730- | 2740: | 129921 |
| 2750- | 2760: | 25426 |
| 2770- | 2780: | 29608 |
| 2790- | 2800: | 45208 |
| 2810- | 2820: | 38966 |
| 2830- | 2840: | 34207 |
| 2850- | 2860: | 35424 |
| 2870- | 2880: | 51690 |
| 2890- | 2900: | 52393 |
| 2910- | 2920: | 34567 |
| GESAMTCHECKSUMME: | | 25519 |

Werkstatt:

DISKETTEN-ARCHIV für Schneider CPC 464,664 und 6128

DISKETTEN-ARCHIV ist ein Programm zur Verwaltung Ihrer Programm- und Datendisketten. Mit DISKETTEN-ARCHIV können Sie sich, ohne lästigen Diskettenwechsel, die Inhalte Ihrer Disketten ansehen, oder aber gezielt nach einem bestimmten File suchen. Bis zu 19 Disketten können verwaltet werden. Nach dem Start des Programms wird Ihnen zunächst das auf die letzte Aktualisierung folgende Datum angezeigt. Haben Sie das Datum bestätigt, bzw. das korrekte Datum eingegeben, gelangen Sie in das Menü von DISKETTEN-ARCHIV. Es stehen fünf Menüpunkte zur Auswahl!

DIREKTORY ANSEHEN: Über diesen Menüpunkt können Sie sich, nachdem Sie die gewünschte Diskettennummer und-Seite eingegeben haben, die Directory ausgegeben lassen. In der Kopfzeile werden neben Disketten-

AKTUALISIEREN: Dient zum Auslesen der Directory einer Diskettenseite. Nach dem Auslesen werden Ihnen die Diskettennummer und -Seite angezeigt. Hatte die

Diskette eine ungültige oder keine Nummer, so wird eine entsprechende Meldung ausgegeben.

nummer und -Seite das Datum der letzten Aktualisierung und das Format der Diskette ausgegeben. (Das sehr selten benutzte IBM-Format wird nicht berücksichtigt.) Anschließend werden alphabetisch geordnet die Filenamen mit der jeweiligen Filelänge ausgegeben. (Das zur Nummerierung der Diskette dienende File wird nicht angezeigt.) Abschließend wird die Anzahl der freien Kbytes der Diskettenseite, sowie die Anzahl der Einträge, abzüglich des Files für die Nummerierung, angezeigt. Auf Wunsch kann die Directory auf einem Drucker ausgegeben werden.

EINTRAG SUCHEN: Diese Option dient, wie der Name schon sagt, zum Suchen eines Files. Ist das File vorhanden, gibt DISKETTEN-ARCHIV den Filenamen, Filelänge, Diskettennummer und -Seite, sowie das Datum, an dem das File zuletzt erfaßt wurde, aus. Konnte das File nicht gefunden werden, so wird Ihnen dies mitgeteilt. Neben der Suche nach einem bestimmten

File, können Sie sich über diese Option auch Listen von Filetypen (z. B. alle Basicfiles durch Eingabe von .BAS) oder bestimmten Filenamen erstellen lassen. Die Suchroutine ist in Maschinensprache geschrieben, so daß die Suche sehr rasch geschieht.

Bei der Eingabe des Suchbegriffs ist darauf zu achten, daß dieser zwischen 2 und 12 Zeichen lang sein muß. Der Suchbegriff kann in Klein- oder Großbuchstaben eingegeben werden.

Die Ausgabe kann wahlweise auf dem Bildschirm oder einem Drucker erfolgen.

Disketten numerieren: Voraussetzung für das Funktionieren von DISKETTEN-ARCHIV ist, daß sich auf den Disketten ein File befindet, das zur Identifizierung der Diskette dient. Über diesen Menüpunkt wird ein File auf Ihren Disketten angelegt, aus dessen Name sich die Diskettennummer und aus dem Filetyp die Seite der Diskette ergibt. Obwohl das Nummerierungsfile nur 1 Byte lang ist, belegt es unter AMSDOS 1

Kilobyte. Sollten Sie der Meinung sein, daß hierdurch der Ihnen auf der Diskette zur Verfügung stehende Speicherplatz zu sehr verringert wird, können Sie natürlich auch ein vorhandenes File so umbenennen, daß es zur Identifizierung der Diskette herangezogen werden kann.

Programm beenden: Bevor DISKETTEN-ARCHIV beendet wird, können Sie entscheiden, ob Sie die Daten speichern möchten oder nicht.

Haben Sie sich dafür entschieden die Daten abzuspeichern, werden Sie aufgefordert die Diskette, auf der sich DISKETTEN-ARCHIV befindet einzulegen. Ist dies geschehen, werden die Daten nach einer kurzen Verzögerung von ca. 2 bis 5 Sekunden, je nach Anzahl der Einträge, abgespeichert.

Innerhalb der kurzen Zeitspanne zwischen der Bestätigung und dem Abspeichern werden die Daten komprimiert, so daß sie immer möglichst wenig Platz auf der Diskette benötigen.

```

100 *****
*****
110 *** Diskettenarchiv fuer Schneider
CPC ***
120 *** Basic-Lader fuer Maschinenrouti
nen ***
130 *****
*****
140
150 MODE 2:LOCATE 10,10:PRINT "Bitte war
ten..."
160 MEMORY 7994:adr=&A410
170 WHILE a$<>"ende"
180 READ a$ : a=VAL("&"+a$)
190 POKE adr,a : sum=sum+a : adr=adr+1
200 WEND
210 RESTORE 540
220 FOR i=0 TO 6
230 READ a$ : a=VAL("&"+a$)
240 POKE 7995+i,a : sum=sum+a
250 NEXT
260 CLS:LOCATE 10,10
270 IF sum<>44314 THEN PRINT "Fehler in
Data's !":END
280
290 PRINT "Bitte legen Sie Ihre 'Archiv'
-Diskette ein und"
300 LOCATE 11,12
310 PRINT "und druecken Sie anschliessen
d eine Taste !"
320 CALL &BB18
330 SAVE "ARCHIV",B,&A410,&167:SAVE "ARC
HIV.DAT",B,7995,7

```

```

340
350 DATA 21,5A,8B,7E,36,C9,F5,E5,11,00,9
C,CD,9B,BC,E1,F1,77,C9,21,0E
360 DATA 9C,ED,5B,D7,A4,FD,21,D9,A4,FD,3
6,00,00,3E,FF,BE,20,1B,23,FD
370 DATA 34,00,01,08,00,ED,B0,3E,2E,12,1
3,01,03,00,ED,B0,7E,CB,FF,12
380 DATA 23,13,23,18,E0,3E,FE,12,C9,06,0
B,11,00,C0,21,DC,A4,CD,77,BC
390 DATA C5,21,3B,1F,E5,CD,B3,BC,CD,7A,B
C,E1,C1,09,2B,E5,01,3E,03,3E
400 DATA FF,ED,B9,23,22,DA,A4,11,41,1F,A
1,ED,52,E1,DB,E5,2A,DA,A4,E5
410 DATA 23,5E,23,56,C1,E1,A1,E5,ED,42,1
9,E5,C1,E1,7E,02,36,00,FE,FF
420 DATA 0B,2B,20,F6,2A,DA,A4,C3,6E,A4,C
9,3E,FE,21,40,1F,01,74,9A,ED
430 DATA B1,22,DA,A4,3E,FF,01,74,9A,ED,B
1,2B,11,74,9A,E5,A1,ED,52,E1
440 DATA D0,ED,5B,DA,A4,7E,12,36,00,13,2
3,FE,FE,20,F6,EB,C3,B1,A4,C3
450 DATA 5A,8B,ED,43,41,52,43,48,49,56,2
0,2E,44,41,54,21,40,1F,36,00
460 DATA 11,41,1F,01,00,7D,ED,B0,C3,55,A
4,CD,A7,A4,06,0B,21,DC,A4,11
470 DATA 00,C0,CD,8C,BC,3E,02,11,3B,1F,2
A,DA,A4,A1,ED,52,EB,E5,C1,CD
480 DATA 9B,BC,C3,8F,BC,2A,D7,A4,36,00,E
5,D1,13,01,2C,03,ED,B0,C3,22
490 DATA A4,FD,21,D9,A4,AF,32,77,A5,2A,7
A,A5,3A,7D,A5,23,BE,28,0B,EB
500 DATA 21,FF,9B,A7,ED,52,DB,EB,18,F1,2
2,7B,A5,3A,7C,A5,47,05,11,7D

```

```

510 DATA A5,23,13,1A,BE,20,1B,10,FB,32,7
7,A5,FD,34,00,23,CB,7E,28,FB
520 DATA 11,0C,00,A7,ED,52,22,7B,A5,23,2
2,7A,A5,C9,2A,7B,A5,18,BD,ende
530
540 DATA 01,01,56,FF,FF,FF,FE

```

Checksummen Teil 1:

| ZEILENR.: | SUMMEN | | | |
|-----------|--------|-------------------------|------|-------|
| 100- 110: | 53382 | 340- | 350: | 36751 |
| 120- 130: | 61924 | 360- | 370: | 71237 |
| 140- 150: | 35761 | 380- | 390: | 71318 |
| 160- 170: | 20371 | 400- | 410: | 73353 |
| 180- 190: | 77926 | 420- | 430: | 72455 |
| 200- 210: | 624 | 440- | 450: | 65331 |
| 220- 230: | 32807 | 460- | 470: | 73240 |
| 240- 250: | 56750 | 480- | 490: | 70412 |
| 260- 270: | 2008 | 500- | 510: | 69277 |
| 280- 290: | 41526 | 520- | 530: | 53580 |
| 300- 310: | 29310 | 540- | | 14926 |
| 320- 330: | 2491 | GESAMTCHECKSUMME: 23258 | | |

```

100 *****
*****
110 ***
***
120 *** Disketten-Archiv fuer Schneide
r CPC ***
130 ***
***
140 *** (C) Wolfgang Boenigk Osnabru
eck ***
150 ***
***
160 *****
*****
170
180 REM Initialisierung
190
200 MODE 2:MEMORY &A000:WIDTH 80:KEY DEF
66,0,1,1,1
210 LOCATE 20,10:PRINT "Bitte warten..."
220 LOAD "ARCHIV.BIN"
230 MEMORY 7990
240 CALL &A4E7
250 ts$="druecken Sie anschliessend eine
Taste !"
260 uts$="und " + ts$
270 REM *** Datum ***
280 tg=PEEK(7995) : tg=tg+1
290 mn=PEEK(7996) : jr=PEEK(7997)
300 IF (tg=29 AND mn=2 AND jr MOD 4) THE
N 350
310 IF tg=30 AND mn=2 AND NOT jr MOD 4.T
HEN 350
320 IF tg=31 AND (mn=4 OR mn=6 OR mn=9 O
R mn=11) THEN 350
330 IF tg<>32 THEN 350
340 tg=1 : mn=mn+1 : IF mn=13 THEN mn=1
: jr=jr+1
350 CLS:LOCATE 10,8 : PRINT "Ist heute d
er ";
360 PRINT USING "##";tg; : PRINT ".";
370 PRINT USING "##";mn; : PRINT ".";
380 PRINT USING "##";jr;
390 PRINT " J/N ?"
400 a$=INKEY$ : IF a$="" THEN 400

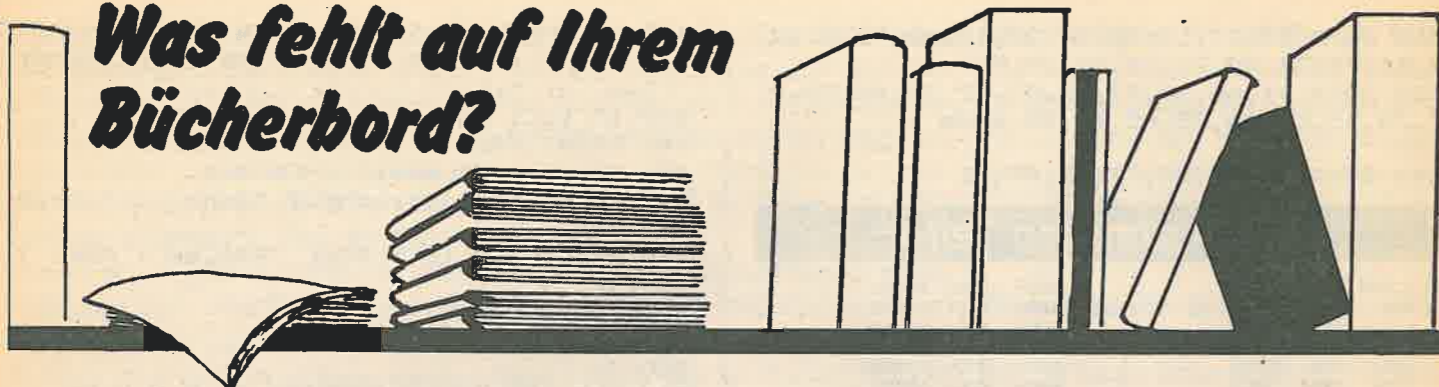
```

```

410 IF UPPER$(a$)<>"N" THEN 470
420 PRINT : PRINT :PRINT TAB (10);:INPUT
"Tag : ";tg
430 IF tg<1 OR tg > 31 THEN 420
440 PRINT TAB(10);:INPUT "Monat : ";mn
450 IF mn<1 OR mn>12 THEN 440
460 PRINT TAB(10);:INPUT "Jahr : ";jr:G
OTO 350
470 POKE 7995,tg : POKE 7996,mn : POKE 7
997,jr
480
490 REM Menue
500
510 CLS:LOCATE 24,5:PRINT "M E N U E"
520 LOCATE 20,8:PRINT "1 Aktualisieren"
530 LOCATE 20,10:PRINT "2 Directory ans
ehen"
540 LOCATE 20,12:PRINT "3 Eintrag suche
n"
550 LOCATE 20,14:PRINT "4 Disketten num
merieren"
560 LOCATE 20,16:PRINT "5 Programm been
den"
570 LOCATE 11,20:PRINT "Geben Sie bitte
die gewuenschte Nummer ein !"
580 a$=INKEY$:IF a$="" THEN 580
590 IF a$<"1" OR a$>"5" THEN 580
600 CLS:ON VAL(a$) GOTO 610,970,1160,162
0,1830
610
620 REM AKTUALISIEREN
630
640 LOCATE 10,10:PRINT "Soll wirklich ak
tualisiert werden J/N ?"
650 a$=INKEY$:IF a$="" THEN 650 ELSE IF
UPPER$(a$)<>"J" THEN 480
660 CLS:LOCATE 10,7:PRINT "Bitte legen S
ie eine Diskette ein und"
670 LOCATE 10,8:PRINT ts$
680 CALL &BB18 : CALL &A410
690 nr=PEEK(&9C01)-&30
700 IF nr>9 OR nr<1 THEN 920
710 IF nr<>1 THEN 740 ELSE IF PEEK(&9C02
)=32 THEN 740
720 b=PEEK(&9C02)-&30
730 IF b>9 THEN 920 ELSE nr=10+b
740 pg=PEEK(&9C09)-&30
750 IF pg<>1 AND pg<>2 THEN 920
760 LOCATE 10,10:PRINT "Disc : ";nr;TAB(2
3);"Seite : ";pg
770 ab=((nr-1)*2+pg-1)*830+8000
780 POKE ab,&FF : POKE ab+3,nr
790 POKE ab+4,pg : POKE ab+7,tg
800 POKE ab+8,mn : POKE ab+9,jr
810 abh=INT((ab+10)/256)
820 abl=ab+10-abh*256
830 POKE &A4D7,abl:POKE &A4DB,abh
840 abh=INT(ab/256) : abl=ab-abh*256
850 POKE ab+1,abl : POKE ab+2,abh
860 CALL &A519
870 POKE ab+5,PEEK(&A4D9)
880 POKE ab+6,PEEK(&A89F)
890 LOCATE 10,12:PRINT "Soll eine weiter
e Diskette aktualisiert werden J/N ?"

```

Was fehlt auf Ihrem Bücherbord?



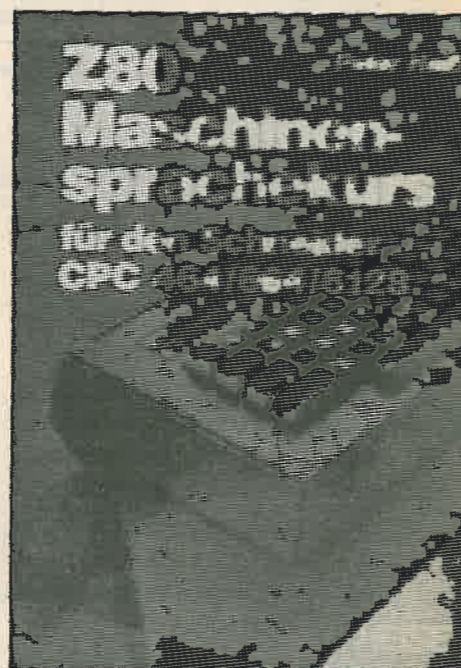
Z80 Maschinensprachekurs für den Schneider CPC 464 / 664 / 6128

von Peter Heiß

Wer seinen Computer richtig einsetzen will, dem reicht früher oder später die Arbeitsgeschwindigkeit eines BASIC-Interpreters nicht mehr aus, so daß er auf die wesentlich schnellere Maschinensprache zurückgreifen muß. Doch spätestens jetzt fangen die Probleme an. Bücher, die die Hard- und Softwareeigenschaften der Schneider-Computer beschreiben, sind zwar schon vorhanden, aber die nützen einem wenig, wenn man nicht weiß, wie der Z80-Prozessor zu programmieren ist. Genau in diese Lücke stößt das Buch „Z80 Maschinensprachekurs“ von Peter Heiß. Praktisch jeder Befehl wird anhand von Übungsprogrammen erklärt, was den Vorteil hat, daß der Leser ziemlich schnell zu

einem Erfolgserlebnis kommt und die Bedeutung und Wirkung der Befehle fast spielerisch begreift. Nach jedem Kapitel werden Übungsaufgaben gestellt, die mit der Zeit schwieriger werden, aber von jedem leicht gelöst werden können, der das Buch aufmerksam durcharbeitet. Am Ende müßte eigentlich jeder in der Lage sein, eigene Programme zu erstellen, so daß das Lernziel relativ schnell erreicht wird. Zum Anhang des Buches gehört neben einer vollständigen Befehlsübersicht auch das Listing eines Direktassemblers, so daß auch selbstentwickelte Programme leicht eingegeben werden können.

Fazit: Empfehlenswert
Z80 Maschinensprachekurs für den Schneider CPC 464 / 664 / 6128
 Heise-Verlag, Hannover



DFÜ für Jedermann

Rainer Severin

Datenfernübertragung ist in der Userszene so etwas wie ein Zauberwort. Nicht zuletzt deshalb, weil es ein Kommunikationsmittel darstellt, das ohne Rücksicht auf die Systemvielfalt Gleichgesinnte miteinander verbindet. Ursprünglich war es der Reiz am Hacken (sprich: häcken), also das Eindringen in fremde Datenbanken, was viele User zum Akustikkoppler greifen ließ. Heute jedoch sind alle wichtigen Datenbanken so kompliziert abgesichert, daß dieser Aspekt kaum noch eine wichti-

ge Bedeutung hat. Dennoch erfreut sich die DFÜ immer größerer Beliebtheit. Elektronischer Briefkasten, Mailboxen mit Kleinanzeigemarkt und Programmübertragung sind nur die populärsten Schlagworte der DFÜ.

Wer sich allerdings als Anfänger an die DFÜ wagt, wird unversehens mit einer Vielzahl von neuen Begriffen, gesetzlichen Bestimmungen und elektronischen Daten konfrontiert.



Der Computerdieb

von Emil Zopfi



Rainer Severins Buch setzt genau an dieser Stelle an und führt den Leser in die Feinheiten und Besonderheiten der DFÜ ein. Mit der den Databeckerbüchern selbstverständlichen Gründlichkeit werden alle wichtigen Aspekte der DFÜ behandelt und verständliche Anleitungen für das eigene Arbeiten mit der DFÜ geliefert. Neben einer Reihe von Hardwaretips, Adressen und Mailboxen und einer Softwareübersicht, wird auch ein eigenes Terminalprogramm zum Abtippen angeboten und eine Bauanleitung für eine Autoanwer-Einrichtung präsentiert. Eine wirklich sehr gelungene Einführung in die Datenfernübertragung, allerdings vermissen wir am Ende des Buchs ein Sachwortregister (in der nächsten Auflage, ja?). (uk)

Databecker-Verlag, Düsseldorf
 Preis: 39,-- DM

Das hier vorgestellte Buch stellt kein Fachbuch zum Thema „Computer“ im üblichen Sinne dar. Es ist vielmehr ein spannendes Unterhaltungsbuch, das sich mit dem Thema von ganz anderer Seite beschäftigt.

Marc's Traum und seine größte Leidenschaft ist der Besitz eines Computers. Er träumt täglich davon, den Golden Arrow,

den besten Computer der Welt zu besitzen. Doch leider hat er kein Geld! Kurz entschlossen beschließt er, während einer Messe seinen Computer zu stehlen. Den Diebstahl plant er wie ein Computerprogramm. Doch wie jedes Programm am Anfang, so hat auch dieser Plan Fehler. Zum Beispiel wußte Marc nicht, daß die Computerfirma eine Sperre gegen Dieb-

stahl in den Computer eingebaut hat, so daß Marc den Computer überhaupt nicht benutzen kann.

In einem Freizeitzentrum trifft Marc Andreas, einen ehemaligen Computerspezialisten. Obwohl der nie wieder etwas mit Computern zu tun haben wollte, reizt es ihn doch, die Sperre zu knacken. So beschließt er zusammen mit Marc gegen den Computer und die Polizei, die Marc mittlerweile auf der Spur ist, zu kämpfen. Wir wollen nicht zu viel über den Inhalt verraten, denn ein wenig Spannung soll schon bleiben. Eine positive Begleitscheinung des Inhaltes dieses Buches ist die Weitergabe von Fachwissen aus dem Bereich der Programmierung. Der Autor, selbst Computerfachmann, hat mit großem Einfühlungsvermögen und Sachverstand eine Geschichte über die Faszination der Computer geschrieben.

Benzingerverlag Zürich, Köln ca. 144 S., 16,80 DM, ISBN 3 545 32260 2

Neu für den Schneider:

Floppybuch für den CPC



Typisch für Databeckerbücher ist, daß sie den Leser tief in das Innere der technischen Geheimnisse blicken lassen. So nicht anders bei dem Floppybuch für den Schneider CPC. Neben einer grundlegenden Einführung in das Diskettenbetriebssystem unter CP/M und in das Arbeiten mit AMSDOS werden alle Filetypen genauestens erläutert und in Programmbeispielen demonstriert. Für das fortgeschrittene Programmieren werden die DOS-Vektoren beschrieben. Das komplette DOS-Listing gehört ebenso zum Umfang des Buches wie ein Diskmonitor und ein

Diskettenmanager. Da die Programmbeispiele sehr umfangreich sind, wird eine Diskette zum Buch für 29,-- DM angeboten.

Das Buch hat schon beste Dienste in unserer Programmierabteilung geleistet und wird auch von unseren Programmierern wärmstens weiterempfohlen.

Brückmann/Schieb, Das große Floppybuch, Diskettenprogrammierung für Einsteiger, Fortgeschrittene und Profis mit dem CPC
 Databecker, Düsseldorf, 49,-- DM

```

900 a$=INKEY$: IF a$="" THEN 900
910 IF UPPER$(a$)="N" THEN 480 ELSE GOTO 660
920 REM *** Ungueltige Discnummer ***
930 PRINT:PRINT TAB(10);"Die eingelegte Diskette hat keine gueltige Nummer !"
940 PRINT : PRINT
950 LOCATE 25,25:PRINT "Weiter auf Taste ndruck !"
960 CALL &BB18 : GOTO 480
970
980 REM Directory ausgeben
990
1000 LOCATE 1,5
1010 PRINT TAB(10);:INPUT "Gewuenschte D isc-Nummer : ";nr
1020 IF nr <1 OR nr >19 THEN 1010 ELSE PR INT
1030 PRINT TAB(10);:INPUT "Gewuenschte S eite : ";pg
1040 IF pg <>1 AND pg <>2 THEN 1030
1050 ab=(nr-1)*2+pg-1)*830+8009
1060 abz=ab
1070 IF PEEK(ab-9)<>&FF THEN PRINT:PRINT :PRINT TAB(10);"Fuer Disc";nr;" Seite";p g;"liegen keine Eintraege vor!":GOTO 950
1080 aus=0 : GOSUB 1950
1090 PRINT : PRINT
1100 PRINT TAB(10);"Soll die Directory a uf den Drucker ausgegeben werden J/N ?"
1110 a$=INKEY$: IF a$="" THEN 1110 ELSE I F UPPER$(a$)<>"J" THEN 480
1120 PRINT:PRINT TAB(10);"Schalten Sie b itte Ihren Drucker ein und"
1130 PRINT TAB(10);ts$
1140 CALL &BB18 : aus=8
1150 ab=abz:GOSUB 1950:GOTO 480
1160
1170 REM Suchen
1180
1190 LOCATE 6,7:L$="":R$=""
1200 PRINT TAB(6);:LINE INPUT "Suchbegri ff : ";s$
1210 IF s$="" OR LEN(s$)=1 OR LEN(s$)>12 THEN 1200
1220 s$=UPPER$(s$)
1230 IF LEN(s$)<5 OR LEN(s$)=12 THEN 131 0
1240 IF MID$(s$,LEN(s$)-3,1)<>"." THEN 1 310
1250 REM Blanks einfuegen
1260 l$=LEFT$(s$,LEN(s$)-4) : r$=RIGHT$( s$,4)
1270 FOR i=LEN(l$) TO 7
1280 l$=l$+" "
1290 NEXT i
1300 s$=l$+r$
1310 POKE &A57C,LEN(s$) : POKE &A4D9,0
1320 POKE &A57A,&40 : POKE &A57B,&1F
1330 FOR i=1 TO LEN(s$)
1340 POKE &A57C+i,ASC(MID$(s$,i,1))
1350 NEXT i
1360 PRINT:PRINT:PRINT TAB(6);"Ausgabe a uf >B<ildschirm oder >D<rucker ?"

```

```

1370 a$=INKEY$: IF a$="" THEN 1370
1380 IF UPPER$(a$)="D" THEN aus=8 ELSE a us=0
1390 IF aus=8 THEN PRINT:PRINT TAB(6);"S chalten Sie bitte Ihren Drucker ein und ":PRINT TAB(6);ts$:CALL &BB18
1400 CLS:LOCATE 1,5
1410 CALL &A529
1420 IF PEEK(&A4D9) THEN 1460
1430 LOCATE 20,7:PRINT s$;" ist nicht vo rhanden !"
1440 LOCATE 25,25:PRINT "Weiter auf Tast endruck !"
1450 CALL &BB18 : GOTO 480
1460 IF PEEK(&A577)=0 THEN PRINT:GOTO 14 40
1470 ad=PEEK(&A578)+PEEK(&A579)*256
1480 gr=INT((ad-8000)/830)*830+8000
1490 PRINT#aus,TAB(5);
1500 FOR i=ad TO ad+11
1510 PRINT#aus,CHR$(PEEK(i));
1520 NEXT i
1530 PRINT#aus,TAB(17);USING "####";PEEK (ad+12)-&80;
1540 PRINT#aus,"K";TAB(28);"Disc : ";USI NG "##";PEEK(gr+3);
1550 PRINT#aus,TAB(42);"Seite : ";USING "##";PEEK(gr+4);
1560 PRINT#aus,TAB(58);"Stand : ";USING "##";PEEK(gr+7);
1570 PRINT#aus,".";USING "##";PEEK(gr+8) ;
1580 PRINT#aus,".";USING "##";PEEK(gr+9)
1590 adh=INT((ad+12)/256):ad1=ad+12-adh* 256
1600 POKE &A57A,ad1:POKE &A57B,adh
1610 GOTO 1410
1620
1630 REM Disketten nummerieren
1640
1650 LOCATE 10,10:PRINT "Sollen wirklich Disketten nummeriert werden J/N ?"
1660 a$=INKEY$: IF a$="" THEN 1660 ELSE I F UPPER$(a$)<>"J" THEN 480
1670 CLS : LOCATE 1,7
1680 PRINT TAB(10);:INPUT "Welche Nummer soll die Diskette erhalten ? ",nr
1690 IF nr<1 OR nr >19 THEN 1680
1700 nr$=STR$(nr)
1710 PRINT : PRINT
1720 PRINT TAB(10);"Legen Sie bitte Seit e 1 der Diskette ein"
1730 PRINT TAB(10);uts$:CALL &BB18
1740 SAVE nr$+".1",b,1,1
1750 PRINT : PRINT
1760 PRINT TAB(10);"Legen Sie bitte Seit e 2 der Diskette ein"
1770 PRINT TAB(10);uts$:CALL &BB18
1780 SAVE nr$+".2",b,1,1
1790 PRINT : PRINT
1800 PRINT TAB(10);"Soll eine weitere Di skette nummeriert werden J/N ?"
1810 a$=INKEY$: IF a$="" THEN 1810
1820 IF UPPER$(a$)<>"J" THEN 480 ELSE 16

```

```

70
1830
1840 REM Programm beenden
1850
1860 LOCATE 5,7:PRINT "Soll das Programm wirklich beendet werden J/N ?"
1870 a$=INKEY$: IF a$="" THEN 1870 ELSE I F UPPER$(a$)<>"J" THEN 480
1880 LOCATE 5,10:PRINT "Sollen die Daten gespeichert werden J/N ?"
1890 a$=INKEY$: IF a$="" THEN 1890 ELSE I F UPPER$(a$)="N" THEN 1930
1900 LOCATE 5,10:PRINT "Bitte legen Sie die Programmdiskette ein "
1910 LOCATE 5,11:PRINT uts$
1920 CALL &BB18 : CALL &A4F7
1930 CALL 0
1940
1950 REM Unterprogramm Directory ausgebe n
1960
1970 CLS:PRINT#aus," Disc : ";USING "##";PEEK(ab-6);
1980 PRINT#aus,TAB(18);"Seite : ";USING "##";PEEK(ab-5);
1990 PRINT#aus,TAB(38);"FORMAT : ";
2000 IF PEEK(ab-3)=&41 THEN PRINT#aus,"C PM"; ELSE PRINT#aus,"DATA";

```

```

2010 IF PEEK(ab-3)=&41 THEN fr=169 ELSE fr=178
2020 PRINT#aus,TAB(58);"Stand : ";
2030 PRINT#aus,USING "##";PEEK(ab-2);:PR INT#aus,".";
2040 PRINT#aus,USING "##";PEEK(ab-1);:PR INT#aus,".";
2050 PRINT#aus,USING "##";PEEK(ab):PRINT #aus
2060 FOR i=1 TO PEEK(ab-4)
2070 FOR b=1 TO 12
2080 PRINT#aus,CHR$(PEEK(ab+b));
2090 NEXT b
2100 PRINT#aus,USING "####";PEEK(ab+13)- &80;
2110 PRINT#aus,"K ";
2120 ab=ab+13
2130 NEXT i
2140 PRINT#aus:PRINT#aus
2150 FOR i=1 TO PEEK(abz-4)
2160 fr=fr-(PEEK(abz+13*i)-&80)
2170 NEXT i
2180 PRINT#aus,fr-1;"K free";TAB(20);PE EK(abz-4);" Eintraege"
2190 IF aus=8 THEN PRINT#8:PRINT#8:PRINT #8
2200 RETURN

```

| CHECK V1: PRUEFSUMMEN: | 480- 490: 3176 | 960- 970: 6109 | 1420- 1430: 67562 | 1880- 1890: 60336 |
|------------------------|------------------|--------------------|-------------------|--------------------|
| 100- 110: 54320 | 500- 510: 11262 | 980- 990: 21345 | 1440- 1450: 59143 | 1900- 1910: 60838 |
| 120- 130: 24138 | 520- 530: 74941 | 1000- 1010: 10754 | 1460- 1470: 49548 | 1920- 1930: 8009 |
| 140- 150: 15812 | 540- 550: 92276 | 1020- 1030: 100511 | 1480- 1490: 54362 | 1940- 1950: 59053 |
| 160- 170: 48020 | 560- 570: 104390 | 1040- 1050: 83122 | 1500- 1510: 41902 | 1960- 1970: 65555 |
| 180- 190: 16695 | 580- 590: 45913 | 1060- 1070: 14201 | 1520- 1530: 66814 | 1980- 1990: 32103 |
| 200- 210: 47354 | 600- 610: 29641 | 1080- 1090: 10198 | 1540- 1550: 29536 | 2000- 2010: 48698 |
| 220- 230: 7375 | 620- 630: 9390 | 1100- 1110: 80278 | 1560- 1570: 51540 | 2020- 2030: 87658 |
| 240- 250: 31970 | 640- 650: 63115 | 1120- 1130: 56655 | 1580- 1590: 87765 | 2040- 2050: 109738 |
| 260- 270: 27172 | 660- 670: 46612 | 1140- 1150: 34084 | 1600- 1610: 27348 | 2060- 2070: 28179 |
| 280- 290: 68213 | 680- 690: 17764 | 1160- 1170: 4005 | 1620- 1630: 28699 | 2080- 2090: 34298 |
| 300- 310: 123242 | 700- 710: 82092 | 1180- 1190: 16215 | 1640- 1650: 27964 | 2100- 2110: 55456 |
| 320- 330: 19619 | 720- 730: 50394 | 1200- 1210: 45704 | 1660- 1670: 22156 | 2120- 2130: 11560 |
| 340- 350: 57031 | 740- 750: 39947 | 1220- 1230: 44523 | 1680- 1690: 51869 | 2140- 2150: 38452 |
| 360- 370: 30591 | 760- 770: 67138 | 1240- 1250: 51766 | 1700- 1710: 12144 | 2160- 2170: 62247 |
| 380- 390: 11101 | 780- 790: 74411 | 1260- 1270: 16961 | 1720- 1730: 52864 | 2180- 2190: 29512 |
| 400- 410: 40580 | 800- 810: 66639 | 1280- 1290: 9153 | 1740- 1750: 14248 | 2200- : 201 |
| 420- 430: 62683 | 820- 830: 59499 | 1300- 1310: 31561 | 1760- 1770: 51637 | |
| 440- 450: 51646 | 840- 850: 68552 | 1320- 1330: 26925 | 1780- 1790: 15985 | |
| 460- 470: 75233 | 860- 870: 16605 | 1340- 1350: 30262 | 1800- 1810: 35631 | |
| | 880- 890: 57581 | 1360- 1370: 76147 | 1820- 1830: 25051 | |
| | 900- 910: 53006 | 1380- 1390: 72210 | 1840- 1850: 16946 | |
| | 920- 930: 79943 | 1400- 1410: 2323 | 1860- 1870: 22082 | |
| | 940- 950: 54568 | | | |

GESAMTCHECKSUMME: 43630

CAVE-WALKER

Schneider CPC 464

10 SYMBOLBESTIMMUNG
20 SYMBOL AFTER 0
30 SYMBOL 200,127,119,187,187,183,127,54,119
40 SYMBOL 201,54,27,51,51,99,99
50 SYMBOL 202,0,0,0,0,0,142,95,31

Ziel des Spiels ist es, „Jonni“ den Archäologen durch neun Höhlen zu steuern und in der letzten ein Schwert einzusammeln. „Jonni“ wird dabei von „Steino“, dem Urzeitmensch, einem fliegenden Totenkopf, herabfallenden Steinen und brodelnden Lavaquellen behindert. Falls eine Perle in der Höhle liegt, sollte sie eingesammelt

werden, denn das bringt Punkte und die werden im weiteren Verlauf des Spieles benötigt. Wenn nämlich in der letzten Höhle der Score unter 800 liegt, findet sich „Jonni“ plötzlich in der 5. Höhle wieder. Vorsicht! Verschiedene Höhlen sind nicht zu bewältigen. Dann muß „Jonni“ zurück gehen.

60 SYMBOL 203,100,62,79,254,56,124,24,24
70 SYMBOL 204,30,119,187,187,183,127,54,119
80 SYMBOL 229,0,32,223,255,199,64
90 SYMBOL 230,0,0,0,0,0,0,112,250
100 SYMBOL 231,124,242,127,28,30,24,24,6

```

3
110 SYMBOL 232,127,223,111,222,222,222,2
23,15
120 SYMBOL 233,27,27,27,27,27
130 SYMBOL 234,56,84,254,238,124,40,56,1
6
140 SYMBOL 235,16,56,16,84,254,84,16,16
150 SYMBOL 236,16,16,16,16,16,16,16
160 SYMBOL 237,56,124,78,87,9,24,24
170 SYMBOL 238,60,118,239,239,246,124,56
,108
180 SYMBOL 239,108,198,195,195,195,231,0
,0
190 SYMBOL 240,0,0,128,192,96,48,96,192
200 SYMBOL 241,255,255,127,127,63,15,31,
7
210 SYMBOL 242,255,255,254,254,252,240,2
48,224
220 SYMBOL 243,255,255,126,60,48,56,48,1
6
230 SYMBOL 244,0,0,0,0,0,60,126,255
240 SYMBOL 245,31,5,3,30,124,80
250 SYMBOL 246,16,56,108,62,56,8,8,0
260 SYMBOL 248,171,182,236,172,120,120,4
8,48
270 SYMBOL 249,138,149,75,83,147,171,215
,175
280 SYMBOL 250,8,28,62,126,126,255,255,2
55
290 SYMBOL 251,108,68,130,130,68,68,40,2
0
300 SYMBOL 252,0,0,0,0,16,16,56,56
310 SYMBOL 253,28,56,16,56,120,56,64,16
320 TITELBILD
330 MODE 1:INK 0,0:BORDER 1:INK 1,18:INK
2,13:INK 3,2:PAPER 0:PEN 1:CLS
340 PEN 1:A$=CHR$(234):LOCATE 1,1:PRINT
A$;A$;A$;" ";A$;" ";A$;" ";A$;" ";A$;A
$;A$;
350 A$=CHR$(234):LOCATE 1,2:PRINT A$;"
";A$;" ";A$;" ";A$;" ";A$;" ";A$
360 A$=CHR$(234):LOCATE 1,3:PRINT A$;"
";A$;" ";A$;" ";A$;" ";A$;" ";A$;A$
370 A$=CHR$(234):LOCATE 1,4:PRINT A$;"
";A$;A$;A$;" ";A$;" ";A$;" ";A$
380 A$=CHR$(234):LOCATE 1,5:PRINT A$;A$;
A$;" ";A$;" ";A$;" ";A$;" ";A$;A$;A$
390 ASCII$="-WALKER"
400 POKE &B338,219
410 ORIGIN 260,340
420 FOR K=1 TO LEN (ASCII$)
430 ASCII=ASC (MID$(ASCII$,K,1))
440 CHARADR=HIMEM+ASCII*8+8
450 FOR I=1 TO 8
460 A=PEEK (CHARADR)
470 BINAER$=BIN$(A,B)
480 FOR J=1 TO 8
490 SEGMENT$=MID$(BINAER$,J,1)
500 IF SEGMENT$="0" THEN 550
510 PLOT X+J*4,I*4
520 PLOT X+2+J*4,I*4
530 PLOT X+J*4,I*4+5
540 PLOT X+2+J*4,I*4+2
550 NEXT J:CHARADR=CHARADR-1:NEXT I

```

```

560 X=X+40
570 NEXT K
580 PEN 2:FOR A=1 TO 39:LOCATE A,7:FOR K
=1 TO 60:NEXT K:SOUND 1,25,1,7:PRINT CH
R$(234):NEXT
590 PEN 2:LOCATE 20,9:PRINT "MENUE:":LOC
ATE 1,12:PRINT "Erklaerung
----- (1):PRINT:PRINT "Start
----- (2)"
600 PEN 3:LOCATE 1,19:PRINT STRING$(39,"
*"):LOCATE 1,25:PRINT STRING$(39,"*"):FO
R A=20 TO 24:LOCATE 1,A:PRINT "*":LOCATE
39,A:PRINT "*":NEXT PEN 1:LOCATE 3,21:P
RINT "(C) 1986 By Michael Bussmann":LOCA
TE 8,23:PRINT "and Lutz Klein.":PEN 2
610 ENV 1,2,7,1,1,-1,1,12,-1,10:ENV 2,1,
15,1,15,-1,10:ENV 3,11,1,1,11,-1,6
620 READ TONE:IF TONE=1 THEN 670
630 IF TONE=80 THEN PAUSE=60 ELSE PAUSE=
30
640 SOUND 2,TONE,PAUSE,0,1,1:SOUND 4,TON
E*2,PAUSE,0,2,1:SOUND 1,TONE/2,PAUSE,0,3
,1
650 GOTO 620
660 DATA 159,119,119,119,127,119,106,106
,106,119,106,95,95,95,106,95,89,89,89,95
,89,80,80,95,106,119,106,106,106,119,127
,119,119,119,119,1
670 LOCATE 1,17:PRINT "Ihre Wahl bitte..
...":
680 O$=INKEY$:IF O$="" THEN 680
690 IF O$<>"1" AND O$<>"2" THEN 680
700 IF O$="2" THEN leben=4:GOTO 830
710 MODE 1:BORDER 0:INK 0,0:INK 1,2:INK
2,9:INK 3,6,2
720 PEN 1:FOR A=1 TO 40:LOCATE A,1:PRINT
CHR$(241);SOUND 1,A*15,1,0,1,1:LOCATE
A,25:PRINT CHR$(241);NEXT A:FOR A=1 TO
25:SOUND 4,A*10,1,0,1,1:LOCATE 1,A:PRINT
CHR$(241);LOCATE 40,A:PRINT CHR$(241);
:NEXT A
730 :POKE &B1CB,0:POKE &B1CF,&CC:POKE &B
1D0,&33:
740 BORDER 3:LOCATE 5,3:PEN 3:PRINT"CAVE
-WALKER":PEN 2:POKE &B1CB,1:POKE &B1CF,&
88:POKE &B1D0,&44:POKE &B1D1,&22:POKE &B
1D2,&11:
750 PEN 2:LOCATE 8,4:PRINT "-----
-----"
760 LOCATE 3,6:PRINT"Das Spiel laeuft in
einer Hoehle ab.":LOCATE 3,8:PRINT"Sie
muessen versuchen,die in der":LOCATE 3,1
0:PRINT"Hoehle liegenden Perlen zu samme
ln.":LOCATE 3,12:PRINT"Dabei behindern S
ie Lavaquellen,"
770 LOCATE 3,16:PRINT"herrabfallende Fel
sbrocken.":LOCATE 3,18:PRINT "Um aus der
Hoehle zu kommen,muessen":LOCATE 3,20:P
RINT "Sie ein Zauberschwert einsammeln."
780 LOCATE 3,14:PRINT "Hoehlenbewohner,g
rosse Steine und"
790 LOCATE 3,23:PEN 3:PRINT "Bitte druec
ken Sie [TAB]!"
800 WHILE INKEY (68) <>0:WEND

```

Korrekturen!!!

The little Puck goes home für C-64 in Ausgabe 6

Die Ladeanweisungen sind mißverständlich erklärt gewesen. So starten Sie den LIT-LEPUCK richtig: 1. Abtippen des Gesamtprogramms; 2. abspeichern; 3. die angegebenen Pokebefehle im Direktmodus eingeben; 4. das Programm neu laden und starten; 5. das Programm mit LOAD"PUCK(16400)", (Geräteadresse),1 laden und mit SYS16400 starten.

Tips und Tricks für den VC-20 in Ausgabe 5, S. 22

In der Programmzeile 90 befindet sich ein Dreher! Richtig lautet sie: 90 POKE A+4609:PEEK(A+4609)+18

```

930 LOCATE PERLE,17:PRINT CHR$(246)
940 GRUB1=INT(RND*30+7):GRUB=INT(RND*30+
7):LOCATE X,17:PEN 3:PRINT CHR$(244);CHR
$(244):LOCATE X1,17:PRINT CHR$(244);CHR$
(244):PEN 2:LOCATE GRUB1,17:PRINT CHR$(2
50);CHR$(250);CHR$(250):LOCATE GRUB,17:P
RINT CHR$(250);CHR$(250);CHR$(250)
950 PRINT CHR$(22)CHR$(1)
960 PRINT CHR$(22);CHR$(1):LOCATE 1,1:PE
N 2:PRINT "Lives: ";LEBEN:LOCATE 20,1:PRI
weiter Seite 62

```

```

810 BORDER 1:RUN
820 MAINROUTINE
830 OT=TIME:farb=INT(RND*25)+1:IF farb=3
OR farb=1 THEN 830 ELSE INK 1,3:INK 2,1
:INK 3,6,26:SPEED INK 10,5:MODE 1:INK 0,
farb:BORDER 0:
840 IF SCHWERT=1 THEN LOCATE SCHWERTP,10
:PRINT CHR$(235):LOCATE SCHWERTP,11:PRIN
T CHR$(236)
850 FOR A=1 TO 40:FOR B=1 TO 3:PAPER 1:L
OCATE A,B:PRINT " ";NEXT B,A
860 ZUF=INT(RND*3):IF ZUF=0 THEN A$=CHR$(
241):O$=CHR$(242) ELSE IF ZUF=1 THEN A$
=CHR$(243):O$=CHR$(241) ELSE IF ZUF=2 TH
EN A$=CHR$(242):O$=CHR$(243)
870 PAPER 0:PEN 1:FOR A=1 TO 40 STEP 2
880 LOCATE a,4:PRINT a$;o$;
890 NEXT a:
900 PAPER 1:FOR a=1 TO 40:FOR b=25 TO 18
STEP -1:LOCATE a,b:PRINT " ";NEXT b,a:
PAPER 0:PEN 1:FOR a=1 TO 40:LOCATE a,17:
PRINT CHR$(244);NEXT a
910 x=INT(RND*28)+7:X1=INT(RND*28)+7:XM=
1:YM=14:IF X=X1 THEN 910
920 PERLE=INT(RND*37)+1:IF PERLE=X OR PE
RLE=X+1 OR PERLE=X1 OR PERLE=X1+1 THEN 9
20

```

★ Kleinanzeigen ★ Kleinanzeigen ★ Kleinanzeigen ★

Auftrag für Gelegenheitsanzeigen in „Compute mit ...“

An „Compute mit ...“
Tronic-Verlag
Postfach
3440 Eschwege

Unter der Rubrik „Kleinanzeigen“-
veröffentlichen wir Gelegenheits-
anzeigen für Verkaufsangebote,
Kauf- und Tauschgesuche, Kon-
taktanfrage bzw. Erfahrungsaustausch usw.
Unser Preis für „Kleinanzeigen“:
Private Kleinanzeigen
bis 5 Zeilen nur DM 5,- DM
bis 10 Zeilen nur DM 10,- DM
Jede gewerbliche Gelegenheits-
anzeige
bis 10 Zeilen nur 15,- DM.
(dürfen nicht unter Chiffre er-
scheinen)
Eine Veröffentlichung erfolgt nur
gegen Vorkasse!

**Kleinanzeigen jetzt
noch preiswerter**

Name und Adresse _____

Unterschrift _____ Datum _____

Ich wünsche folgenden Text zu veröffentlichen:

_____ 5,-

_____ 10,-

Zutreffendes ankreuzen

suche Software suche Hardware Tausch Versch.
biete an Software biete an Hardware Kontakte Chiffre

Biete Software

Über 1300 Spiele und Anwenderprogramme für Ihren C-64 gegen Unkosten abzugeben. Liste mit Preisangabe kostenlos. A. Wittgen, Postlagernd 073861 C, 5060 Bergisch Gladbach 1

Verkaufe original „PROTEXT“-Textverarbeitung für Commodore 128 mit Handbuch für 40,- DM. Tel. 05 61 / 89 86 33.

C16/116/4 PLUS/VC-20/C-64 Fibu, VE/EK, Foto-, Film-, Video-Musikarchive, Lotto, Adress, YSI Tabello, Digitalo, Spielepakete, Kartei, Biblio, KFZ, Denktraining Lernprogramme, Handbücher, Staubschutzhaube ab 9,90 DM. Katalog gegen 80 Pf-Marke. Computerservice T. Hofstede, A. d. Windmühle 8, 5010 Bergheim 5.

C-64! Nur 1 DM je Programm! C-64! Clubsoftware (Anwender-, Spiel-, Lern-, Denk- und Musikprogramme gegen Kopiergebühr von nur 1 DM abzugeben. Z. B. Vokabeltrainer, Astrologieprogr. mit Bildschirm- u. Drukerausgabe, Synthesizer, Biorhythmus, Actionspiele und Hilfsroutinen. Kein Schund! Info (Rückporto). **TEXASSOFT**, Herthastr. 26, 8 München 19

★ **C16/116 ★ Grafikprogramme**
★ Mini-CAD, sichern, laden
★ und weiterbearb. d. Grafi-
★ ken; Info gegen Porto; M. ★
★ Rätzel, Ulvenbergstr. 6, D-★
★ 6100 Darmstadt ★ ★ ★ ★ ★

Hallo C16 Fans und 116 Plus 4 Ich habe für Euch den Automaten Super 7 programmiert. Cassette + Listing DM 10,- Schein Stichwort S7. Rolf Leesemann, Feldstr. 40, Postfach 3172, 2300 Kiel 1. Lieferung: 5 Tage.

Über 1300 Spiele und Anwenderprogramme für Ihren C-64 gegen Unkosten abzugeben. Liste mit Preisangabe kostenlos. A. Wittgen, Postlagernd 073861 C, 5060 Bergisch Gladbach 1

C-16/C-116 Superspiele! Info gegen Rückporto bei Hannes Kaltenbach, Prielmayerstr. 16, 7990 Friedrichshafen 1.

C-16/116 ★ Plus 4 ★ C-16/116 Software ? - na klar!
- Grafik
- Sound
- und für die Schule

Eine ausführliche Info gibts gegen Rückporto bei: B. Lauer, Handgasse 10, 8700 Würzburg.

● ● ● ● ● **C-16/C-116** ● ● ● ● ●
Spiele, Anwender-Progr. Softw.-Schutz(Broschüre), Turbo-Tape (3600 Baud) zu verkaufen. Alle Programme auf Cassette. A. Lämmerzahl, Saarstr. 85, 4100 Duisburg 17. Ruf: 02136/113994.

VC-20 Spitzenprogramme für 6V, 8K und 16K. Liste für nur 50 Pfr-Briefmarke bei M. Platt, Carl-Orff-Str. 4, 8721 Hambach! Jetzt preiswert: Zentrax (16K). Ein S.F.-Abenteurer mit herrlichen 3D-Bildern. Nur 180 DM. 3 Actionspiele für die 6V, nur 14 DM! Super Spiele - super billig!

C16/116 ★ C16/116 ★ C16/116 Das erste Gaphic-Adventure „Die Jünger des Todeslichtes“ für den C-16/116 ist da. 14 Seq. a 12KB = 168 KB. Preis? Ganze 20,- DM. Bestellen bei: Jörg Kowalski, Bickernstr. 161, 4650 Gelsenkirchen. Oder Info gegen Rückporto. Nur als Cassette lieferbar!

Für alle Schneider-Besitzer Verkaufte ich viele Spielprogramme. Preisliste u. Beschreibung gegen Briefmarke. Keine Raubkopien! K. Ley, Postfach 2124, 2390 Flensburg.

VC-20 ★ ★ ★ ★ ★ C-64 Neue Liste Spiel + Anwendersoftware. Info bei: Volker Coert, Söllingstr. 4-8, 4300 Essen 1. Bitte Rückporto!

CPC 464 ★ CPC 664 ★ CPC 6128 Verkaufte super CHEMIEPROGRAMM 10 DM bei Cas., 25 DM bei Disc. Tausche auch Programme. Liste an: Holger Groß, Am Ginsterberg 13, 6630 Saarlouis 4

CPC ● TOP-SOFTWARE ● CPC
● ● Biorhythmus-Total ● ●
Läßt vergleichbare Programme erblissen. Exklusiver Bildschirm, doppelte Grafik, Vergrößerung, sehr komfortabler Ausdruck - Ein Monat ca. 8 DIN-A4-Seiten mit sehr ausführlichen Erläuterungen und Hardcopy für Grafiken. Zusätzlich können Partnerschaftsanalysen erstellt werden, ebenfalls mit ausführlichem Ausdruck. Interessante Nebenverdienstmöglichkeiten bietet dieses Programm, das sich auf rein wissenschaftliche Grundlagen und Erfahrungen stützt (49K). Bestellungen unter Nr. 0961 gegen Vorkasse bei Hilterscheid, Ludwig-Rosenberg-Ring 47, 2050 Hamburg-80, Tel. 040/7211974, Cass. DM 30,-, Disk DM 40,-

C-16/C-116/plus 4
Musikcompiler für Interruptgesteuerte Musik; Sonderzeichen für Ihre eigenen Programme; Vokabelprogramm mit französischen Zeichen; Spiele; Lernprogramme etc. Liste anfordern! BILLIG PREISE! Tel. 025 81 - 67 15, M. Suer, Walgernweg 22, 4410 Warendorf 1

★ **C-64 ★ Spitzensoftware** ★ ★ ★ ★ ★
Adressenverw. (D) 12,95
Videokass.-Verw. (D) 12,95
★ Jed. Prg. auf Disk + Handbuch. Info geg. 80 Pf. Preis per NN + Porto bei: Michael Rau, Hüheweg 46, 8580 Bayreuth. ★ ★ ★ ★ ★

C-16/116 ★ C-16/116 ACTION
Bei mir gibt es Topspiele für nur 20,- DM. Diesen Monat sind es: PENGO, TERMINATOR, SHAO-LIN, BOMBER, DALLAS. Den Bestellungen bitte Schein oder Scheck beilegen. Bestellungen bitte an: Roland Fuchs, Blanckenagelweg 9, 4770 Soest/Westf. P.S. Tausche auch. Angaben ob Diskette oder Kassette. ★ ★ ★ ★ ★

★ **C16/116 ★ Grafikprogramme**
★ Mini-CAD, sichern, laden
★ und weiterbearb. d. Grafi-
★ ken; Info gegen Porto; M. ★
★ Rätzel, Ulvenbergstr. 6, D-★
★ 6100 Darmstadt ★ ★ ★ ★ ★

Tausch-Biete C-16/116 Software
C-16/116
C-16/116
Tausche - Biete C-16/C-116 Software. Habe Utilities, Textverarbeitung, Plotterfunktionen, Mathematik, Knochelein u. a. nützliche (!!!) Programme: Info bei St. Sedlaczek, Florastr. 11, 5063 Overath

Biete Hardware

★ ★ ★ **C-16 Hardware** ★ ★ ★
★ 16K Ram Erweiterung 89,- ★
★ 64K Ram Erw. (nur C16) 169,- ★
★ Joystickadapter 10,- ★
★ ★ ★ **C-16 Software** ★ ★ ★
★ Favourite 4 (Sammlung) 30,- ★
★ Plus Paket (Sammlung) 30,- ★
★ Turbo Tape 15,90 ★
★ Tel. Best. 073 07 / 62 30 ★
★ Comp. Service PF1304 7913 Senden ★

★ **Commodore Hardware** ★
★ Drucker Comrex CR220 349,- ★
★ Drucker Star Gemini 10x 578,- ★
★ Oki 182 798,- ★
★ Schutzhaub. C64/C16 ab 12,90 ★
★ Schutzhauben C128 15,90 ★
★ Joystick Quickshot II 19,90 ★
★ Disketten Select 2D 27,90 ★
★ Nur solange Vorrat reicht. ★
★ ★ ★ ★ ★ Tel. Bestellung: 073 07 / 62 30 ★
★ ★ ★ ★ ★ Comp. Service PF1304 7913 Senden ★

Für VC-20: Biete 24KB Erweiterungsmodule an. Schaltbar von 3KB bis 24 KByte. Preis: 90,- DM. Informationen gegen Rückporto. Friedel, Heinz, An der Halde 43, 4330 Mülheim-Ruhr.

Commodore C-16 + Datensette + Basic-Lehrbuch und Kassette incl. Garantie bis 30.9.1986 nur DM 189,-!!! ★ ★ ★ Commodore C16 Speichererweiterung nur DM 79,- incl. Einbau und Funktionsgarantie auf 60671 Bytes free!! = Selbstkostenpreis. 128 K für C-16 in Kürze!! Info: Tel.: 062 01/69189 ★ ★ ★ Anwender- u. Spiele-Software vorhanden.

★ **Hallo C-16 Besitzer** ★
★ Wie wäre es, wenn Sie ihr
★ C-16 nach dem Einschalten so
★ begrüßt
★ ★ ★ **60671 BYTES FREE** ★ ★ ★
★ Info gegen Freiumschlag
★ Markus Antel, Buchenweg 14
★ 8205 Kiefersfelden
★ ★ ★ ★ ★

ACHTUNG C16/116 Besitzer
Speichererweiterung 64K (60671) Bytes free) zum Superpreis von 59, 50 DM + 5 DM f. Porto u. Verpackung. (Muß eingebaut werden, Bauanleitung liegt bei). Martin Brüggemann, Bachstr. 10, 5940 Lennestadt 17, Tel. Ab 16 Uhr 027 21/828 27.

Verkaufe wegen Systemwechsel Commodore C-16 mit Datensette u. Joy-Adap. v. verschiedene Programme. (DM 250) - 5 Monate alt). An Jan Linek, Blumenthalstr. 9, 8500 Nürnberg 80, Tel. 09 11 / 26 32 86.

Verkaufe Schneider CPC 464. Mit grün Monitor und 5 Programmen wegen Systemwechsel zu verkaufen. Preis: VB. Frank Siegel, ☎ 070 21 / 67 24

★ ★ ★ ★ ★
★ **ACHTUNG C16/C116 Be-**
★ **sitzer!!** Endlich ist er da, ★
★ der Professionelle Joy-★
★ stick-Adapter, für nur sage ★
★ und schreibe 9,95 DM und ★
★ wo: Fa. H.P.T. Electronic & ★
★ Co, Am Rain 23-27, 8024 ★
★ Deisenhöfen, Tel. 089 / ★
★ 613 41 20 ★
★ Nur solange Vorrat reicht. ★
★ ★ ★ ★ ★

Verkaufe Commodore 128 PC, Floppy 1571, Monitor 1901, Drucker MPS 803 komplett 2.000,- Neuwert 3.700,- alles erst 2 Monate alt. Abgabe wegen Bundeswehr. Dazu gratis 2 Joystick + 40 Disketten mit Box. Programme wie Nord Star, Datamat, Protex, Winter Games usw. Markus Jung, Schmachendorfer Str. 158, 4200 Oberhausen 11, ☎ 02 08 / 68 23 67

Verkaufe VC-20, VC-20 + Datensette + 16K. Erw. 9 Kassetten (Superspiele) z. B.: Chopflifer, Spuckhaus, Schaelt, Fire Galaxy. Preis: 260,- (VHB). Tel. 065 43 / 29 84 von 16 bis 20 Uhr.

Verkaufe Commodore CBM 8032, DIN-Tastatur 8250 Dual Floppy, Matrixdrucker Itho 1550 incl. Interface + D-P-Kabel super günstig! Schmeding, Sachsenstr. 5, 4980 Bünde, Tel. 052 23 / 109 71

★ **VC-20 ★ SUPER ★ C-64 ★**
★ VC-20 - Speichererweite-★
★ rung 32 KB; Insges. ä. B. 27 ★
★ 5K für Basic, 11K für MC-★
★ Prg's. 1MCL. 3K-Bereich!!! ★
★ Der Preis: nur 70 DM! Au-★
★ Berdem Relais-Interface ★
★ 8xUM für C-64 oder VC-★
★ 20! Mit Resettaster nur 50 ★
★ DM! ★ Melden bei: Sven ★
★ Rakers, Feldweg 10, 4425 ★
★ Billerbeck, Tel. 025 43 / ★
★ 4238 ★

★ **C16/C116 Ram-Erweiterung!**
C16/116 60671 Bytes free für DM 97,80 - Alle Ports frei - keine Einschränkungen - incl. Umbau ★ Computer ohne Zubehör mit 100,- DM (2,20 für die Post) an: **Elektronik-Technik, Tannenweg 9, D-2351 Trappenkamp**

Verkaufe Drucker Seikosha GP-100 VC für Commodore 150,- DM und dazu Eprom mit deutschem Zeichensatz auf Sekundäradresse 7 für 30,- DM. Tel.: 05 61 / 89 86 33.

SUPER ANGEBOT!!! Biete den Schneider CPC 464 und Diskettenlaufwerk für 2000,- DM (Festpreis) an. Mit Anleitungsbüchern für Computer und Floppy. Geeignet für Anfänger: Meldet euch bei: Oliver Ilns, Rispenweg 21c, 2000 Hamburg 53, Tel. 040 / 84 24 05. Eilt!!!! Zwecks Systemwechsels.

Suche

+++ Suche VC-20 +++
+ möglichst mit Cassetten-+
+ laufwerk ohne Joystick. +
+ K.H. Wendelstein, Her-+
+ mann-Löns-Straße 31, Tel.: +
+ 07123/14513, 7430 Met-+
+ zingen. +
+++++

Suche für den VC-20 „POLE POSITION, FLUCH DES PHARAO.“ Außerdem viele Action- und Starspiele. Tausche auch. Biete für original Modul „Pole Position“ bis 30 DM. Brauche auch Anwendungsprogramme, Floppy, Akustikkoppler, Drucker und Videokamera zu vernünftigen Preisen. Tel.: 084 59 / 15 36

C-16: Suche Kassetten-Anwenderprogramme (keine Spiele!), 64 K-Ram realisiert. Suche auch jegliche Tips zu Hardware-Erweiterungen. Freundliche Angebote bitte mit kurzer Beschreibung an: Ulrich Vollmer, Prökelmoor 1, 2000 Hamburg 71.

Suche Diskettenlaufwerk für C-16. Tel. 045 31 / 32 13, Arno Fenske, Berliner Ring 35, 2060 Bad Oldesloh.

Suche Joysticks oder Adapter + Joysticks, um Commodore C16 bedienen zu können.

Suche wenig gebrauchten Drucker für Commodore C-16/116 + 64 und Joysticks für C-16/116. Klaus Michel, 3565 Breidenbach.

C-16 ★ C-16 ★ C-16
Suche ein gebrauchtes, nicht zu teures Diskettenlaufwerk für den Commodore 16. Angebote bitte an die Tel.-Nr. A-077 58 / 25 05. Es EILT!!!! Verkaufte Plotter/Printer 1520/Neu SCHNELL!!!!

Club-Nachrichten

Geschenkt ist das nun wirklich schon! Für nur 5,- monatlich könnt ihr Mitglied im VC-20-Club „VC-Freaks“ werden. Davon wird die umfangreiche Clubzeitung mit massenhaft Listings, Tips und Tricks usw. finanziert. Zusätzlich erhält jedes Mitglied monatlich 6 Computerprogramme, Überraschungspakete selbstredend. Keinen Schund, sondern harte Actionspiele in Maschinensprache!!! Jedes Mitglied hat ebenfalls Zugriff zu den circa 800 Computerprogrammen der VC-20-Clubs, und es werden täglich mehr. Für einen lächerlichen Unkostenbeitrag von 5 DM bekommt ihr ein Informationspaket auf Band, sowie massenhaft Spiel- und Anwenderprogramme!! Erstgenanntes überwiegend! Also Freaks, stürzt Euch auf die Kulis und schreibt an A. Spind-

ler, Moschstr. 10, 6522 Osthofen, Telefon 062 42 / 33 90
AUCH NICHTMITGLIEDER KÖNNEN MIT UNS TAUSCHEN!

Der C-64/VC-20-mini-club besteht seit April 1984. Ursprünglich war er für C-64 und VC-20 Anwender gedacht, doch inzwischen haben wir auch C-128 und neuerdings auch C-16 User unter unseren Mitgliedern. Unsere Mitglieder sind über die gesamte Bundesrepublik verteilt. Kontakte zu anderen Computer-Clubs betreiben wir ebenfalls. Es wäre also auch schön, wenn andere Clubs mit uns Verbindung aufnehmen würden. Das Alter unserer Mitglieder geht von 12 bis 65 Jahren, ist also breit gefächert. Unser Beitrag beträgt 2,50 DM im Monat. Dafür bekommt man vorerst alle zwei Monate unsere Clubzeitschrift bzw. in zeitschriftlosen Monaten unsere Clubnachrichten. Eine Mailbox ist in Arbeit. Weitere Themen sind unser Club Funk, in dem man auch mal akustischen Kontakt der anderen Mitglieder mitbekommt. Weitere Fragen oder sonstiges werden beantwortet unter der Nummer 02151 / 73 59 46. Info (kostenlos) erhält man beim C-64/VC-20 mini-club, Postfach 1314 in 415 Krefeld 29.

Verschiedenes

★ **C-64 ★ Spitzensoftware** ★ ★ ★ ★ ★
Adressenverw. (D) 12,95
Videokass.-Verw. (D) 12,95
★ Jed. Prg. auf Disk + Handbuch. Info geg. 80 Pf. Preis per NN + Porto bei: Michael Rau, Hüheweg 46, 8580 Bayreuth. ★ ★ ★ ★ ★

C16/116 ★ C16/116 ★ C16/116 Das erste Gaphic-Adventure „Die Jünger des Todeslichtes“ für den C-16/116 ist da. 14 Seq. a 12KB = 168 KB. Preis? Ganze 20,- DM. Bestellen bei: Jörg Kowalski, Bickernstr. 161, 4650 Gelsenkirchen. Oder Info gegen Rückporto. Nur als Cassette lieferbar!!


```

NT "Score: "; SCORE: LOCATE 1,3: PRINT "Score: "; HOEHLE
970
980 LOCATE XM,14: PRINT CHR$(230): LOCATE
XM,15: PRINT CHR$(231): LOCATE XM,16: PRINT
CHR$(232); CHR$(240): LOCATE XM,17: PRINT
CHR$(233): LOCATE XM,18:
990 STEIN=INT(RND*2): STEINO=INT(RND*2): W
URF=INT(RND*2)
1000 IF STEINO=1 THEN LOCATE 38,15: PEN 1
: PRINT CHR$(237): PEN 2: LOCATE 38,16: PRIN
T CHR$(238): PEN 1: LOCATE 38,17: PRINT CHR
$(239) ELSE LOCATE 38,15: PRINT " ": LOCAT
E 38,16: PRINT " ": LOCATE 38,17: PRINT " "
1010 PEN 3: IF steino=1 AND wurf=1 THEN L
OCATE 37,16: PEN 1: PRINT CHR$(229): FOR KO
PF=36 TO 3 STEP -1: LOCATE KOPF,16: PRINT
CHR$(234): CALL &BD18: PRINT CHR$(22): CHR$(
0): SOUND 1,15,1,2: LOCATE 37,16: PRINT "
": IF KOPF=XM AND YM=14 THEN WURF=0: STEIN
0=0: GOTO 1160
1020 PRINT CHR$(22); CHR$(0): PEN 1: IF STE
IN=1 THEN STEINK=XM: FOR STEINP=6 TO 16: L
OCATE STEINK,STEINP: PRINT CHR$(242): CALL
&BD18: SOUND 4,0,2,6,0,0,30
1030 IF INKEY(75)=0 THEN GOTO 1270
1040 IF INKEY(76)=0 THEN GOTO 1350
1050 IF INKEY(74)=0 THEN GOTO 1310
1060 IF TIME-OT>5700 THEN GOTO 1460
1070 IF XM=PERLE THEN SCORE=SCORE+100: FO
R SOUNTE=1000 TO 1 STEP -15: SOUND 2,SOUN
TE,1,7,0,0,20: NEXT SOUNTE: PERLE=1
1080 IF XM=X OR XM=X1 OR XM=X+1 OR XM=X1
+1 THEN 1480
1090 IF XM=GRUB OR XM=GRUB+1 OR XM=GRUB1
OR XM=GRUB1+1 OR XM=GRUB+2 OR XM=GRUB1+
2 THEN 1160
1100 IF STEIN=1 AND STEINK=XM AND STEINP
>14 AND STEINP<16 THEN 1160
1110 IF XM=PERLE THEN SCORE=SCORE+100: FO
R SOUNTE=1000 TO 1 STEP -13: SOUND 2,SOU
NTED,1,6,0,0,1: NEXT
1120 IF steinp<17 AND STEIN=1 THEN LOCAT
E STEINK,STEINP: PRINT " ": NEXT STEINP
1130 IF STEIN=0 AND STEINO=1 AND WURF=1
THEN LOCATE KOPF,16: PRINT " ": NEXT KOPF
1140 GOTO 990
1150 'LEBEN VERLOREN
1160 INK 0,0:
1170 FOR WAITED=1 TO 50
1180 BORDER INT(RND*26)
1190 SOUND 1,2000,1,7,0,0,20
1200 NEXT WAITED
1210 BORDER 0: INK 0,ZUF
1220 LEBEN=LEBEN-1:
1230 IF HOEHLE>B THEN HOEHLE=5: SCHWERT=0
: SCHWERTP=0
1240 IF LEBEN=0 THEN INK 0,11: INK 1,26: I
NK 2,6: PEN 2: PAPER 0: MODE 0: PRINT: PRINT
"Leider verloren!!!!": PRINT: PEN 1: PRINT
"Score: "; score: PRINT: PEN 14: PRINT "Press
[Tab] to": PRINT: PRINT "restart or 'E' t
o": PRINT: PRINT "end!" ELSE 830
1250 IF INKEY$="" THEN 1250

```

```

1260 IF INKEY(68)=0 THEN RUN ELSE a$=LOW
ER$(INKEY$): IF a$="" THEN 1260 ELSE IF a
$="e" THEN CALL &BBFF: CALL &BB4E: END ELS
E 1260
1270 LOCATE XM,14: PRINT " ": LOCATE XM,15
: PRINT " ": LOCATE XM,16: PRINT " ": LOCAT
E XM,17: PRINT CHR$(244): SOUND 1,0,2,3,0,
0,20
1280 XM=XM+1: LOCATE XM,14: PEN 2: PRINT CH
R$(230): LOCATE XM,15: PEN 1: PRINT CHR$(23
1): LOCATE XM,16: PEN 2: PRINT CHR$(232); CH
R$(240): LOCATE XM,17: PEN 1: PRINT CHR$(23
3): CALL &BD18: LOCATE XM,17: PRINT CHR$(20
1)
1290 IF XM>37 THEN HOEHLE=HOEHLE+1: GOTO
1430
1300 GOTO 1040
1310 LOCATE XM,14: PRINT " ": LOCATE XM,15
: PRINT " ": LOCATE XM,16: PRINT " ": LOCAT
E XM,17: PRINT CHR$(244): SOUND 1,0,2,3,0,
0,20
1320 XM=XM-1: IF XM<3 THEN HOEHLE=HOEHLE-
1: GOTO 1430
1330 PEN 2: LOCATE XM,14: PRINT CHR$(202):
PEN 1: LOCATE XM,15: PRINT CHR$(203): PEN 2
: LOCATE XM,16: PRINT CHR$(204): LOCATE XM,
17: PEN 1: PRINT CHR$(233): CALL &BD18: LOCA
TE XM,17: PRINT CHR$(201)
1340 GOTO 1060
1350 LOCATE XM,14: PRINT " ": LOCATE XM,15
: PRINT " ": LOCATE XM,16: PRINT " ": LOCAT
E XM,17: PRINT CHR$(244)
1360 IF XM=SCHWERTP THEN GOTO 1520
1370 XM=XM+5: IF XM>37 THEN HOEHLE=HOEHLE
+1: GOTO 1430
1380 LOCATE XM,10: PRINT CHR$(230): LOCATE
XM,11: PRINT CHR$(231): LOCATE XM,12: PRIN
T CHR$(232); CHR$(240): LOCATE XM,13: PRINT
CHR$(245)
1390 FOR SOUNTE=1 TO 1000 STEP 20: SOUND
4,SOUNTE,1,7: NEXT SOUNTE:
1400 LOCATE XM,10: PRINT " ": LOCATE XM,11
: PRINT " ": LOCATE XM,12: PRINT " ": LOCAT
E XM,13: PRINT " "
1410 LOCATE XM,14: PRINT CHR$(230): LOCATE
XM,15: PRINT CHR$(231): LOCATE XM,16: PRIN
T CHR$(232); CHR$(240): LOCATE XM,17: PRINT
CHR$(233):
1420 GOTO 1050
1430 BORDER 6,1: FOR SOUNTE=1000 TO 1 ST
EP -5: INK 0,INT(RND*26): SOUND 2,SOUNTE,
1,7: NEXT SOUNTE:
1440 IF HOEHLE<1 THEN HOEHLE=1
1450 IF HOEHLE<>9 THEN SCHWERTP=0: SCHWER
T=0: GOTO 830 ELSE IF SCORE>800 THEN 1510
ELSE 50000
1460
1470 LOCATE XM,14: PRINT " ": LOCATE XM,15
: PRINT " ": LOCATE XM,16: PRINT " ": LOCAT
E XM,17: PRINT CHR$(244): LOCATE XM,16: PRI
NT CHR$(248): LOCATE XM,17: PRINT CHR$(249
): FOR WAITED=1 TO 500: NEXT WAITED: GOTO 1
160
1480 PEN 1: LOCATE XM,14: PRINT " ": LOCATE

```

```

XM,15: PRINT CHR$(253): PEN 3: LOCATE XM,1
6: PRINT CHR$(252); " ": LOCATE XM,17: PRINT
CHR$(251)
1490 FOR SOUNTE=2000 TO 1000 STEP -20: S
OUND 2,SOUNTE,1,7,0,0,1: NEXT SOUNTE
1500 FOR WAITED=1 TO 500: NEXT WAITED: GOT
O 1160
1510 SCHWERTP=INT(RND*30+7): SCHWERT=1: MO
DE 0: PEN 3: BORDER 24,2: INK 0,1: INK 4,2,2
4: PRINT " ACHTUNG !!!
": PEN 4: PRINT "Letzte Hoehle !!!": FOR WAI
TED=1 TO 1500: NEXT WAITED: MODE 1: PEN 3: G
OTO 830
1520 IF SCORE<700 THEN HOEHLE=5: GOTO 830
1530 'ALLE HOEHLLEN DURCHQUERT
1540 RESTORE 1600
1550 tempo=12
1560 status=1
1570 WHILE status<>0 AND NOT brk
1580 READ status,tone,duration
1590 SOUND status,tone,INT(tempo*duratio
n*0.8334),13
1600 WEND
1610 DATA 49,159,2,42,190,2,28,239,2
1620 DATA 17,190,1,10,239,1,49,179,2,42,
213,2,28,239,2,49,159,4,42,190,4,28,239,
4,49,159,2,42,190,2,28,239,2,49,142,2,42
,179,2,28,239,2,49,159,2,42,190,2,28,239
,2,49,142,2,42,179,2
1630 DATA 28,239,2,49,134,2,42,159,2,28,
201,2,17,159,1,10,201,1,49,150,1,42,179,
1,28,201,1,49,134,4,42,159,4,28,201,4,49
,134,2,42,159,2,28,201,2,49,119,2,42,150
,2,28,201,2,49,134,2,42,159,2
1640 DATA 28,201,2,49,119,2,42,150,2,28,
201,2,49,113,4,42,134,4,28,169,4,49,113,
1,42,134,1,28,169,1,49,134,1,42,169,1,28
,239,1,49,113,2,42,134,2,28,169,2,49,106
,4,42,134,4,28,179,4
1650 DATA 49,106,1,42,134,1,28,179,1,49,
134,1,42,179,1,28,213,1,49,106,2,42,134,
2,28,179,2,49,106,4,42,127,4,28,159,4,49
,106,1,42,127,1,28,159,1,49,127,1,42,159
,1,28,213,1
1660 DATA 49,106,2,42,127,2,28,159,2,49,
100,4,42,127,4,28,169,4,49,100,1,42,127,
1,28,169,1,49,127,1,42,169,1,28,201,1,49
,100,2,42,127,2,28,169,2,49,95,10,42,127

```

```

,10,28,150,10,0,0,0
1670 MODE 1: INK 0,1: BORDER 13: PAPER 0: IN
K 1,6: INK 2,18: INK 3,2,1
1680 PEN 3: PRINT "Herzlichen Glueckwunsch
!!!
": PEN 2: PRINT "Sie haben es geschafft 10
Hoehlen zu
": PRINT "durchqueren."
1690 LOCATE 1,7: PRINT "Sie erreichten": P
EN 1: LOCATE 16,7: PRINT score: PEN 2: LOCAT
E 22,7: PRINT "Punkte.": PEN 1: PRINT "====
=====
"
1700 IF SQ(1)<>4 THEN 1700 ELSE CLEAR
1710 FOR A=1 TO 360
1720 MOVE 300,100
1730 PLOT 300+100*SIN(A),100+90*COS(A),I
NT(RND*4)
1740 DEG: NEXT
1750 MOVE 230,160: DRAW 295,270,2: MOVE 37
6,160: DRAW 305,270
1760 MOVE 250,20: DRAW 295,100: MOVE 350,2
0: DRAW 305,100
1770 LOCATE 19,24: PRINT CHR$(233): LOCATE
19,23: PRINT CHR$(232); CHR$(240): LOCATE
19,22: PRINT CHR$(231): LOCATE 19,21: PRINT
CHR$(230)
1780 FOR A=1 TO 360
1790 DEG
1800 MOVE 300,100:
1810 PLOT 300+10*COS(A),100+10*SIN(A),1
1820 NEXT
1830 FOR DUR=1 TO 150: FOR P=1 TO 3: FARB=
(RND*25)+2
1840 INK P,FARB: NEXT P,DUR
1850 LOCATE 1,10: PRINT "Nocheinmal <j/n>
?"
1860 A$=UPPER$(INKEY$): IF A$="" THEN 186
0
1870 IF A$<>"J" AND A$<>"N" THEN 1860
1880 IF A$="J" THEN RUN ELSE CALL &BBFF:
CALL &BB4E: MODE 1: END
50000 'ZUWENIG PUNKTE FUER LETZTE HOEHLE
50001 MODE 1: BORDER 6,3: INK 0,2: PEN 1: PR
INT "Ihre Punkte reichen nicht aus,
": PRINT "deshalb koennen Sie ab der 5.
": PRINT "Hoehle weitere Perlen einsammel
n": FOR waited=1 TO 2000: NEXT waited: hoeh
le=5: GOTO 820

```

| CHECK V1: PRUEFSUMMEN: | 350- 360: 48494 | 770- 780: 65001 | 1190- 1200: 11413 | 1610- 1620: 34014 |
|------------------------|-----------------|-------------------|--------------------|--------------------|
| | 370- 380: 81184 | 790- 800: 67579 | 1210- 1220: 24245 | 1630- 1640: 29920 |
| | 390- 400: 16957 | 810- 820: 8127 | 1230- 1240: 93130 | 1650- 1660: 62907 |
| | 410- 420: 22383 | 830- 840: 84255 | 1250- 1260: 13111 | 1670- 1680: 71680 |
| | 430- 440: 70152 | 850- 860: 65936 | 1270- 1280: 100254 | 1690- 1700: 56202 |
| | 450- 460: 20528 | 870- 880: 35004 | 1290- 1300: 63658 | 1710- 1720: 8858 |
| | 470- 480: 22387 | 890- 900: 46466 | 1310- 1320: 108674 | 1730- 1740: 15647 |
| | 490- 500: 48259 | 910- 920: 111428 | 1330- 1340: 29145 | 1750- 1760: 56713 |
| | 510- 520: 40738 | 930- 940: 50648 | 1350- 1360: 47449 | 1770- 1780: 68829 |
| | 530- 540: 52276 | 950- 960: 63379 | 1370- 1380: 59715 | 1790- 1800: 1826 |
| | 550- 560: 59846 | 970- 980: 62700 | 1390- 1400: 74933 | 1810- 1820: 44292 |
| | 570- 580: 26914 | 990- 1000: 14522 | 1410- 1420: 10769 | 1830- 1840: 54499 |
| | 590- 600: 78843 | 1010- 1020: 29313 | 1430- 1440: 46620 | 1850- 1860: 57514 |
| | 610- 620: 55808 | 1030- 1040: 25541 | 1450- 1460: 23158 | 1870- 1880: 74957 |
| | 630- 640: 77584 | 1050- 1060: 31185 | 1470- 1480: 80773 | 50000-50001: 69760 |
| | 650- 660: 38502 | 1070- 1080: 78345 | 1490- 1500: 57428 | |
| | 670- 680: 56232 | 1090- 1100: 95034 | 1510- 1520: 79011 | |
| | 690- 700: 60025 | 1110- 1120: 68497 | 1530- 1540: 23871 | |
| | 710- 720: 21858 | 1130- 1140: 31204 | 1550- 1560: 14990 | |
| | 730- 740: 37825 | 1150- 1160: 11543 | 1570- 1580: 76469 | |
| | 750- 760: 55414 | 1170- 1180: 21968 | 1590- 1600: 14570 | |

BESAMTCHECKSUMME: 2061