

64'er
5. PROGRAMM
SONDERHEFT
 Programmierhilfen, Tips & Tricks für C 64
 ★ Neue Top-Listings zum Abtippen

64'er



Floppy

- ★ So programmiert man sie
- ★ Automatische Fehlersuche
- ★ Kopierschutz

Datasette

- ★ So macht man sie 10 mal schneller
- ★ Justierung

Tips & Tricks

- ★ Betriebssystem-Editor
- ★ Tastenbelegung ändern
- ★ Programme verkürzen
- ★ Mathematische Sonderzeichen
- ★ Sprachein- und ausgabe

Basic-Erweiterungen

- ★ Strukturiertes Basic
- ★ Grafik-Basic
- ★ Disk-Basic
- ★ Spiele-Basic



Alle Programme auch auf Diskette und Kasette erhältlich

Programme, die jeder braucht

Interessante Informationen und hervorragende Listings zu ausgesuchten Themen finden Sie in unserer 64'er Sonderheftreihe. In diesem Sonderheft haben wir bislang noch nicht veröffentlichte Listings und Tips & Tricks zu den Themen

- Datasette,
- Floppy
- Basic-Erweiterungen
- Programmier-Hilfen

zusammengetragen. Ob Anfänger oder Profi — für jeden ist etwas dabei.

Zwei Speichermedien sind es, die der C 64-Besitzer in erster Linie einsetzt: die Datasette und die Floppy 1541. Die Datasette trifft man hauptsächlich dort an, wo entweder die Finanzlage nicht allzu rosig aussieht, oder wo jemand erst einmal in die Computerwelt hineinschnuppern will. »Aufrüsten« ist ja immer noch möglich. Aber Berichte von Händlern und Umfragen zeigen es ganz deutlich: der Vormarsch der Floppy ist nicht zu bremsen. Und das mit Recht.

Schnelle Datasette

Datasetten-Benutzer haben oft Schwierigkeiten, Kassetten von Freunden oder Bekannten zu laden. Vielfach liegt es an einem verstellten Tonkopf. Wir geben Ihnen wertvolle Tips, wie man fremde Kassetten trotzdem nutzen kann. Das Laden und Speichern von Programmen ist bei der Datasette ein langwieriges Unternehmen. Mit unserem Quick-Tape wird sie auf einmal schneller als die Floppy.

Floppy- und Anwendungslistings

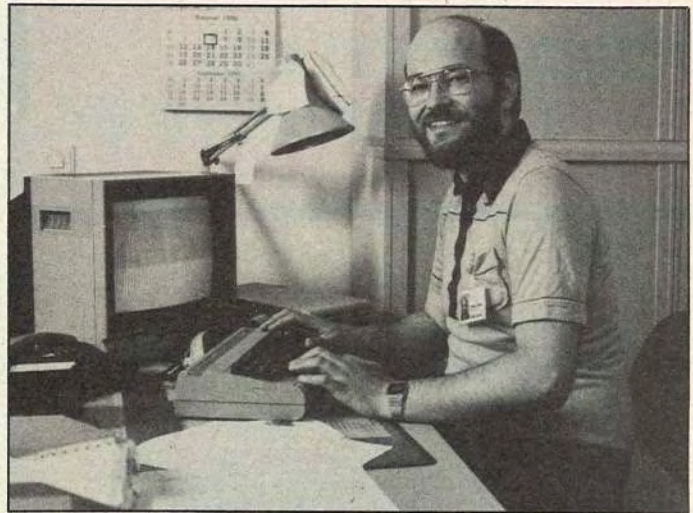
Unwahrscheinlich vielseitig ist der Komplex »Floppy«, für den wir einiges an Hilfsprogrammen und wertvollen Anwendungen anbieten. Eine schnelle Formatieroutine und ein sehr schnelles Kopierprogramm sind ebenso gefragt wie einige



Programme zur Manipulation des Directorys. Greifen Sie in das DOS der Floppy ein mit dem Floppy-Monitor! Vergessen Sie die manchmal sehr umständliche Bedienung des Floppy-Laufwerks! Arbeiten Sie mit »Disk-Master«! Eine komplette Dateiverwaltung, mit der unter anderem eigene Masken erstellt werden können, wird Sie begeistern!

Basic-Erweiterungen

Wir haben einige sehr interessante Basic-Erweiterungen für Sie herausgesucht, die alles bieten, was Sie sich vorstellen können. Sind Sie ein Liebhaber der strukturierten Programmierung? Dann ist »aufpoliertes Basic« das Richtige für Sie. Sogar Programmierer von Spielen dürfen sich auf eine spezielle Basic-Erweiterung für Spiele freuen. Und mit »Disc-Basic« bekommen Sie das Floppy-Laufwerk erst richtig in den Griff.



Tips und Tricks-Listings

Tips und Tricks sind immer wieder begehrt und kommen selbstverständlich nicht zu kurz. Darunter sind einige echte Leckerbissen, zum Beispiel die Cross-Reference-List, die Ihre Basic-Programme durchleuchtet. Komplett in Maschinencode geschrieben und deshalb sagenhaft schnell. Für Neugierige ist der Sourcecode gleich mit abgedruckt.

Alles in allem erhalten Sie in diesem Sonderheft rund 45 Listings, die es in sich haben. Alle Programme wurden mit Prüfsummen versehen, um Ihnen das Eintippen zu erleichtern. Sie finden dazu zwei Programme, den Checksummer 64 V3 (auf Seite 6 für Basic-Programme) und den MSE (für Maschinensprache-Listings auf Seite 8). Damit können Sie alle anderen Programme sicher und bequem abtippen.



Listing-Service auf Kassette und Diskette

Wer keine Zeit oder keine Lust hat, alle Programme selbst in mühevoller Kleinarbeit abzuschreiben, kann wieder auf den bewährten Leserservice zurückgreifen.

Alle Listings befinden sich auf

Diskette mit der Bestellnummer L6 85 S5D. Die Diskette kostet 29,90 Mark.

Kassette mit der Bestellnummer L6 85 S5K. Die Kassette kostet 19,90 Mark.

Bitte verwenden Sie für Ihre Bestellung die beigelegte Postscheck-Zahlkarte zur Überweisung des Rechnungsbetrags. Sie erleichtern uns die Auftragsabwicklung und Sie sparen sich die 3 Mark Versandkosten!

Vorwort

Programme, die jeder braucht 3

Prüfsummenlistings

Checksummer 64 V3 6
MSE — Eintippen sicher und leicht gemacht 8

Basic-Erweiterungen

Komfortable Befehls-erweiterung
42 neue Basic-Befehle erleichtern das Programmieren.
Disk-Basic
Speziell zur Programmierung von Dateiverwaltungen mit sequentiellen und relativen Dateien.
Record-Befehl für den C 64
Ein kleines Maschinenprogramm, das den RECORD-Befehl implementiert.
Aufpoliertes Basic
39 leistungsfähige Basic-Befehle.
Print-Using
Der C 64 kann Zahlen formatiert ausgeben.
Game-Basic
Eine Spezialerweiterung für Spiele.
Disk-Basic 64
Fast 50 neue Toolkit-Befehle für die 1541-Floppy.

Datsette

COP — Hilfe für die Datsette
Ein Directory auf jeder Kassette läßt ein Programm schnell finden.
So stellt man die Datsette ein
Eine einfache Anleitung zur Justierung des Tonkopfes von Datsetten.
Turbo Tape de Luxe — Datsette 10mal schneller
Das Hypra-Load für Kassette. Speichern und Laden geht nun mit der Datsette schneller als mit dem 1541-Laufwerk.

Floppy-DOS-Erweiterungen

Automatische Fehlersuche — Floppy-Fehler ohne Rätsel
Am Bildschirm erscheint automatisch eine eventuelle Fehlermeldung des Laufwerks.
Diskmaster — Beherrschen Sie das 1541-Laufwerk
Ein Toolkit für das Floppy-Laufwerk, das auch mit vielen kommerziellen Programmen zusammen verwendbar ist.
Entstörung des 1541-Laufwerks
Mit etwas Bastelaufwand können Sie Ihr Laufwerk gegen äußere Störungen abschirmen.
Schnelles Formatieren mit der 1541
In 11 Sekunden wird eine Diskette formatiert.
So programmiert man sie:
FMON 1541 — Betriebssystem-Monitor
Kein Diskettenmonitor, sondern ein DOS-Monitor. Mit ihm können Sie leicht eigene Programme in den Floppy-Speicher schreiben.
Directory-Editor
Mit dem Directory-Editor können Sie das Directory Ihrer Disketten ganz nach Ihren Wünschen gestalten.

Floppy

Programmschutz durch Autostart und Paßwort 59
Ein sehr guter Schutz für Ihre eigenen Programme.
Directory-Drucker 61
Die Directories von zwei Disketten werden tabellarisch nebeneinander gedruckt.
ProDat — Dateiverwaltung mit Raffinessen 64
Eine sehr universelle und schnelle Dateiverwaltung.
Hypra-Copy 69
Ein schnelles Kopierprogramm mit Hypra-Load und Hypra-Save-Routinen.
Programme leichter laden 72
Direktwahl aus dem Directory.
Disk-Füller 74
Bis zu 17 zusätzliche freie Blöcke generieren.
Schreibschutz per Software 75
Eleganter als das Hantieren mit Schreibschutz-Aufklebern ist dieser Software-Schreibschutz.
Disk-Tester 76
Ganze Disketten werden auf fehlerhafte Blöcke hin untersucht und diese als belegt gekennzeichnet.

Tips und Tricks

Betriebssystem selbstgemacht:
Betriebssystem-Editor 78
Änderungen am Betriebssystem des C 64 leichtgemacht.
Mehr Platz auf der Diskette 80
Ein File-Compactor kürzt Programme bis zu 55%.
Bildschirm-Effekte leichtgemacht 82
Tips für Anfänger 83
Aus ein nach zwei 84
Zwei Programme laufen gleichzeitig im C 64 ab.
Leichte Eingabe 86
Die Nachteile des INPUT-Befehls werden beseitigt.
Mathematische Sonderzeichen:
Formeln perfekt gedruckt 88
Bruchstrich, Wurzelzeichen etc. sind kein Problem mehr.
Schreiben Sie Ihre eigenen Fehlermeldungen! 92
Kopierschutz ohne Read Error 94
Tape Save — Maschinenprogramme einfach speichern 95
Linker 64 — Schluß mit dem Nachladen 96
Variablen Dump 98
Escape-Taste für Gänsefüßchen-Modus 99
Programme verkürzen — Sparen Sie Speicherplatz und Ladezeit 100
Ein ganz raffinierter Daten-Packer
Sprachein- und ausgabe: Der C 64 lernt sprechen 101
Sprechen kann der C 64 mit diesem Programm sofort. Damit er Sie auch versteht genügt ein Mikrofonadapter.
On Error Goto 103
Tastenbelegung ändern — Programmieren Sie Ihre Tastatur 105
Übersichtliches Listing 104
Komfortable Menüsteuerung 108
Anhalter — Programmstopp auf Tastendruck 109
Variable Funktionen 109
Geben Sie über den INPUT-Befehl ganze Gleichungen ein!
Cross-Ref 64 — Basic-Programme unter der Lupe 111

Rubriken

Leserservice 62
Impressum 122

Checksummer 64 — Neu

Der Checksummer 64 V3 überprüft jede Basic-Zeile direkt nach der Eingabe, erkennt Fehleingaben und auch Vertauschungen von Zahlen und Ziffern, und erspart deshalb eine aufwendige Fehlersuche.

Der Checksummer 64 V3 ist ein kleines Maschinenprogramm, das Sie sofort unterrichtet, ob Sie die jeweilige Programmzeile korrekt eingegeben haben.

So gehen Sie vor:

1. Programm abtippen und speichern.
2. Starten mit RUN
3. Nach kurzer Zeit sehen Sie am Bildschirm:
CHECKSUMMER 64, CHECKSUMMER AKTIVIERT,
AUSSCHALTEN MIT POKE 1,55, ANSCHALTEN MIT POKE
1,53, READY.

4. Anschalten des Checksummer 64 V3 mit POKE 1,53.

5. Test: Geben Sie in einer freien Zeile ein: »1 REM« und drücken die RETURN-Taste. Am Bildschirm oben links sollten Sie die Prüfsumme < 63 > sehen.

6. Geben Sie ein Listing aus unserem Heft ein. Nach jeder Zeile wird die Zahl, die im Listing in Klammern < > steht, in den Bildschirm eingeblendet. Stimmen die Zahlen nicht überein, so liegt vermutlich ein Eingabefehler vor. **Die Zahl in den Klammern, und auch die Klammern selbst, dürfen beim Abtippen nicht mit eingegeben werden!**

7. Dieser neue Checksummer 64 V3 bemerkt, im Gegensatz zu den bisherigen, auch Vertauschungen von Zahlen und Buchstaben.

8. Unsere Basic-Listings enthalten keine Steuerzeichen mehr. Diese werden ersetzt durch Klartext und stehen zwischen geschweiften Klammern. Deshalb sind weder die Klammern noch was dazwischen steht, abzutippen, sondern die in Tabelle 1 aufgeführten Tasten zu drücken. Auf Ihrem Bildschirm erhalten Sie dann wieder die entsprechenden Grafikzeichen (siehe Bild 1 und 2).

9. Alle Grafikzeichen werden ebenfalls ersetzt durch unterstrichene oder überstrichene Großbuchstaben. Unterstrichene Buchstaben bedeuten, daß Sie die SHIFT-Taste und den angegebenen Buchstaben drücken müssen, überstrichene jedoch die Commodore-Taste mit dem Buchstaben. Auch hier erhalten Sie am Bildschirm das entsprechende Grafikzeichen und nicht etwa das im Listing erkennbare Zeichen (siehe Bild 1 und 2).

Checksummer VC 20 V3

Der Checksummer VC 20 V3 ist im Prinzip genauso aufgebaut wie der Checksummer 64. Da beim VC 20 jedoch nicht die Möglichkeit besteht, das ROM softwaremäßig zu modifizieren, mußte ein anderer Weg als beim Commodore 64 gewählt werden, um die Checksumme zu generieren.

In ihrer Funktionsweise unterscheiden sich der Checksummer VC 20 und der Checksummer 64 nicht. Es gelten folgende Sonderregelungen bei der Benutzung des Checksummer VC 20:

— Da der Basic-Bereich nicht belegt werden soll, ist das Programm im Kassettenpuffer abgelegt.

— Angeschaltet wird der Checksummer VC 20 mit »SYS 955«.

CTRL steht für Control-Taste, so bedeutet [CTRL-A], daß Sie die Control-Taste und die Taste »A« drücken müssen. Im folgenden steht:

| | |
|-------------|--|
| {DOWN} | Taste neben rechtem Shift, Cursor unten |
| {UP} | Shift-Taste & Taste neben rechtem Shift; Cursor hoch |
| {CLR} | Shift-Taste & 2. Taste ganz rechts oben |
| {INST} | Shift-Taste & Taste ganz rechts oben |
| {HOME} | 2. Taste von ganz rechts oben |
| {DEL} | Taste ganz rechts oben |
| {RIGHT} | Taste ganz rechts unten |
| {LEFT} | Shift-Taste & Taste unten rechts |
| {SPACE} | Leertaste |
| {F1} | grauer Tastenblock rechts |
| {F3} | grauer Tastenblock rechts |
| {F5} | grauer Tastenblock rechts |
| {F7} | grauer Tastenblock rechts |
| {F2} | grauer Tastenblock rechts & Shift |
| {F4} | grauer Tastenblock rechts & Shift |
| {F6} | grauer Tastenblock rechts & Shift |
| {F8} | grauer Tastenblock rechts & Shift |
| {RETURN} | Shift-Taste & Return |
| {BLACK} | Control-Taste & 1 |
| {WHITE} | Control-Taste & 2 |
| {RED} | Control-Taste & 3 |
| {CYAN} | Control-Taste & 4 |
| {PURPLE} | Control-Taste & 5 |
| {GREEN} | Control-Taste & 6 |
| {BLUE} | Control-Taste & 7 |
| {YELLOW} | Control-Taste & 8 |
| {RVSON} | Control-Taste & 9 |
| {RVOFF} | Control-Taste & 0 |
| {ORANGE} | Commodore-Taste & 1 |
| {BROWN} | Commodore-Taste & 2 |
| {LIG.RED} | Commodore-Taste & 3 |
| {GREY 1} | Commodore-Taste & 4 |
| {GREY 2} | Commodore-Taste & 5 |
| {LIG.GREEN} | Commodore-Taste & 6 |
| {LIG.BLUE} | Commodore-Taste & 7 |
| {GREY 3} | Commodore-Taste & 8 |

Wenn Sie sich erst einmal an die in Klartext geschriebenen Steuerzeichen gewöhnt haben, werden Sie den Vorteil dieser Schreibweise erkennen. Der zu dem jeweiligen Steuerzeichen gehörende Klartext ist so verfaßt, daß Sie leicht die Taste beziehungsweise die Tastenkombination finden, die Sie drücken müssen.

Die Steuerbefehle im Klartext

— Abschaltung des Checksummer VC 20 wird mit »SYS 58459« vollzogen.

Achtung: Nehmen Sie keine Kassetten-Operationen vor, wenn der Checksummer VC 20 eingeschaltet ist. Da das Betriebssystem den Kassettenpuffer mit Daten belegt, kann der Checksummer VC 20 überschrieben werden, was zur Folge hat, daß sich der Computer bei aktiviertem Checksummer VC 20 »aufhängt«. Wollen Sie deshalb ein Programm auf (von) Kassette abspeichern (laden), so müssen Sie erst den Checksummer VC 20 abschalten (SYS 58459).

Daraufhin kann der Kassettenpuffer mit Daten überschrieben werden, ohne daß der Computer »aussteigt«.

Als Sicherung wird bei der Initialisierung geprüft, ob das zuletzt angesprochene Peripherie-Gerät der Kassettenrecorder war. Ist das der Fall, so werden die Betriebssystemroutinen LOAD und SAVE für die Benutzung gesperrt. Der Rechner meldet bei Aufruf einer dieser beiden Routinen READY, ohne weitere Aktionen durchzuführen. Diese Sicherung kann man nach der Tipparbeit aufheben, wenn man den Checksummer VC 20 mit SYS 58459 abschaltet. Dadurch wird der Kassettenpuffer für andere Daten freigemacht. Weiterhin wird dann durch gleichzeitiges Drücken der Tasten »Run-Stop & Restore« erreicht, daß die Betriebssystemroutinen LOAD und SAVE wieder eingerichtet werden.

— Bei Benutzung einer Diskettenstation brauchen Sie nicht darauf zu achten, daß bei LOAD beziehungsweise SAVE der

Checksummer VC 20 überschrieben wird, da der Kassettenpuffer für die Diskettenstation normalerweise nicht genutzt wird. Deshalb können Sie die beiden Routinen weiterhin normal nutzen, sofern der Rechner bei der Initialisierung des Checksummer VC 20 feststellt, daß das zuletzt angesprochene Peripherie-Gerät nicht der Kassettenrecorder war.

— Bedingt durch den anderen Aufbau des Checksummer VC 20 wird anders als beim Checksummer 64 nach der LOAD-Routine keine Checksumme ausgegeben.

— Wird eine Zeile gelöscht, also eine Zahl zwischen 0 und 65999 eingegeben, und danach Return gedrückt, so wird eine Checksumme ausgegeben, die aber keine Bedeutung hat.

Sie können die Programme auch weiterhin ohne den Checksummer eintippen. (F. Lonczewski/gk)

Hinweis: [13 SPACE] bedeutet 13mal die Leertaste drücken

```

1 REM *****
2 REM *
3 REM *      CHECKSUMMER 64 V3      *
4 REM *
5 REM *      WRITTEN MAERZ 1985 BY  *
6 REM *
7 REM *      FRANK LONCZEWSKI      *
8 REM *
9 REM *****
10 PRINT "{CLR,11SPACE,RVSON}CHECKSUMMER 64
    V3{RVOFF}"
11 PRINT "{2DOWN,9SPACE}EINEN MOMENT, BITTE
    ..."
12 FOR I=828 TO 864:READ A:POKE I,A:PS=PS+
    A+1:NEXT I
13 IF PS<>5802 THEN PRINT"PRUEFSUMMENFEHLE
    R IN ZEILEN 20-22":END
14 SYS 828:PS=0:FOR I=58464 TO 58583:READ
    A:POKE I,A:PS=PS+A+1:NEXT I
15 IF PS<>16267 THEN PRINT"PRUEFSUMMENFEHLE
    R IN ZEILEN 22-30":END
16 POKE 1,53:POKE 42289,96:POKE 42290,228
17 PRINT "{4DOWN,9SPACE}CHECKSUMMER AKTIVIE
    RT."
18 PRINT "{2DOWN}AUSSCHALTEN : POKE1,55"
19 PRINT "{DOWN}ANSCHALTEN{2SPACE}: POKE1,5
    3":NEW
20 DATA 169,0,133,254,162,1,189,93,3,133,2
    55,160,0,177,254
21 DATA 145,254,136,208,249,230,255,165,25
    5,221,95,3,208,238,202
22 DATA 16,230,96,160,224,192,0,160,2,169,
    0,170,133,254,177
23 DATA 95,240,40,201,32,208,3,200,208,245
    ,133,255,138,41,7
24 DATA 170,240,14,72,165,255,24,42,105,0,
    202,208,249,133,255
25 DATA 104,170,232,165,255,24,101,254,133
    ,254,76,111,228,192,4
26 DATA 48,219,198,214,165,214,72,162,3,16
    9,32,157,1,4,189
27 DATA 212,228,32,210,255,208,12,0,92,72,
    32,201,255,170,104
28 DATA 144,1,138,96,202,16,228,166,254,16
    9,0,32,205,189,169
29 DATA 62,32,210,255,104,133,214,32,108,2
    29,169,141,32,210,255
30 DATA 76,128,164,9,60,18,19
    
```

© 64'er

Dieser neue Checksummer 64 V3 erkennt auch Vertauschungen von Zahlen.

```

10 REM*****
11 REM*
12 REM* CHECKSUMMER *
13 REM*
14 REM* V3      VC20 *
15 REM*
16 REM*      WRITTEN *
17 REM* MAERZ 1985 *
18 REM*      BY *
19 REM*F. LONCZEWSKI*
20 REM*****
    
```

```

21 PRINT "{CLR,SPACE,RVSON}CHECKSUMMER V3 V
    C-20{RVOFF}"
22 PRINT "{2DOWN}EINEN MOMENT, BITTE..."
23 FOR I=827 TO 1019:READ A:POKE I,A
24 PS=PS+A+1:NEXT I
25 IF PS<>24464 THEN PRINT "{DOWN}PRUEFSUMM
    ENFEHLER !":END
26 SYS 981:PRINT"CHECKSUMMER AKTIVIERT."
27 PRINT"AN :SYS981"
28 PRINT "{DOWN}AUS:SYS58459, BEI CAS-{4SPA
    CE}SETTE ZUSAETZLICH{5SPACE}RUN/STOP &
    RESTORE"
29 PRINT "{DOWN}BEI AKTIVIERTEM CHECK-SUMME
    R KEIN";
30 PRINT" CASSETTEN-BETRIEB (LOAD, SAVE){2
    SPACE}ERLAUBT!":NEW
31 DATA 32,95,3,134,122,132,123,32,115,0,1
    70,240,243,162,255
32 DATA 134,58,144,10,162,0,134,255,32,121
    ,197,76,225,199,162
33 DATA 1,134,255,76,156,196,166,255,224,1
    ,240,3,76,96,197
34 DATA 160,2,169,0,170,133,254,177,95,240
    ,40,201,32,208,3
35 DATA 200,208,245,133,253,138,41,7,170,2
    40,14,72,165,253,24
36 DATA 42,105,0,202,208,249,133,253,104,1
    70,232,165,253,24,101
37 DATA 254,133,254,76,119,3,192,4,48,219,
    198,214,165,214,72
38 DATA 162,3,169,32,157,1,4,189,209,3,32,
    210,255,202,16
39 DATA 242,166,254,169,0,32,205,221,169,6
    2,32,210,255,104,133
40 DATA 214,32,135,229,169,141,32,210,255,
    162,0,134,255,240,148
41 DATA 9,60,18,19,169,59,141,2,3,169,3,14
    1,3,3,165
42 DATA 186,201,1,208,16,169,116,141,48,3,
    141,50,3,169,196
43 DATA 141,49,3,141,51,3,173,136,2,141,17
    0,3,96
    
```

64ER ONLINE

© 64'er

Der neue Checksummer VC 20 V3 erkennt auch Vertauschungen von Zahlen.

```

5 PRINT CHR$(14) <242>
10 PRINT "{CLR}" <254>
20 PRINT "QXXXXXXXXXXXXXXXXXXXXXXXXXXXX" <130>
30 PRINT "{4DOWN,2SPACE}TEST {SPACE,BLUE,6SP
    ACE}" <022>
40 PRINT "BBBBBBBBBBBBBBBBBBBBBBBBBBBBBB" <108>
    
```

© 64'er

Bild 1. So könnte ein Teil eines Listings abgedruckt sein. In Zeile 10 müssen Sie nach den Anführungsstrichen die CLEAR/HOME-Taste drücken und nicht die Klammern mit dem Wort CLR. In Zeile 20 drücken Sie nach den Anführungsstrichen die Commodore-Taste und den Buchstaben Q, gefolgt von mehreren SHIFT und Stern-Taste, und zum Schluß die Commodore-Taste und den Buchstaben W. In Zeile 30 ist es viermal die Cursor-nach-unten-Taste, gefolgt von zweimal die Leertaste, dann SHIFT und T und normal EST, zum Schluß noch einmal die Leertaste, die Farbtaste Blau (Control und 7) und sechsmal die Leertaste. Zeile 40 besteht lediglich aus mehreren Grafikzeichen, die mit der Commodore-Taste und B erzeugt werden.

```

5 PRINTCHR$(14)
10 PRINT"Q"
20 PRINT" |-----|"
30 PRINT"XXXXXXXXX | EST  "
40 PRINT"XXXXXXXXXXXXXXXXXXXXXXXXXXXX"
    
```

Bild 2. Auf dem Bildschirm oder Ihrem Drucker sieht das Listing (Bild 1) so aus.

MSE - Abtippen sicher und leicht gemacht

Ähnlich wie der »Checksummer« ist auch der MSE ein Hilfsmittel bei der Eingabe von Listings, diesmal jedoch bei reinen Maschinensprache-Programmen.

Im Gegensatz zum »Checksummer« aber ist die Eingabe nicht ohne den MSE möglich. Der MSE verringert die Tipparbeit um ein Drittel und schließt Fehleingaben vollkommen aus. Außerdem können Sie die Werte blind eingeben, ohne andauernd auf den Bildschirm schauen zu müssen. Dies wird durch akustische Meldungen realisiert.

MSE ist ein Maschinenspracheditor, mit dem ein Vertippen ausgeschlossen ist. Eine abgetippte Zeile wird nur angenommen, wenn sie richtig ist. Eine Checksumme am Ende jeder Zeile prüft, ob die richtigen Werte in der richtigen Zeile an der richtigen Stelle stehen. Wenn nicht, ertönt ein Warnsignal, und man beseitigt den Fehler.

War die Zeile korrekt, erklingt ein Gong, und die nächste Zeilennummer wird ausgegeben. Damit ist also auch »blindes« Eintippen möglich; Sie können sich voll auf den Text konzentrieren.

So arbeitet man mit MSE

Laden und starten Sie MSE. Zuerst wird der Programmname und die Start- und Endadresse erfragt. **Diese Angaben entnehmen Sie dem Kopf des jeweiligen abgedruckten Listings.** MSE meldet sich dann mit der Zeilennummer der ersten Zeile. Wenn Sie die Zeile richtig eingegeben haben, erscheint die nächste Zeilennummer und so weiter bis zum Ende. Zum Schluß wird das fertige Programm mit »CTRL-S« auf Diskette oder Kassette abgespeichert. Dazu sind keine

weiteren Angaben mehr erforderlich. Das Programm kann dann ganz normal wieder geladen und gestartet werden. Wenn Sie nicht alles auf einmal tippen wollen, können Sie jederzeit unterbrechen und den eingetippten Teil mit »CTRL-S« abspeichern. Wollen Sie weiterarbeiten, laden und starten Sie MSE wieder.

Geben Sie auf die Frage nach der Startadresse aber jetzt »L« ein, um Ihr Teilprogramm zu laden. Jetzt können Sie mit »CTRL-N« die Adresse eingeben, an der Sie weitertippen müssen. Wenn Sie sich nicht gemerkt haben, wie weit Sie gekommen sind, geben Sie nach dem Laden »CTRL-M« ein.

Auf die Frage nach der Startadresse antworten Sie mit der Anfangsadresse, die links in der Kopfzeile auf dem Bildschirm steht. Nun wird Ihr Programm aufgelistet. Mit »SPACE« wird das Listen fortgesetzt, mit »STOP« abgebrochen. Das Ende Ihres Programmteils erkennen Sie sehr einfach daran, daß nur noch der Wert »AA« in der Zeile steht. Die Adresse dieser Zeile müssen Sie anschließend mit »CTRL-N« eingeben. Das Programm ist nur mit »STOP/RESTORE« zu verlassen. Speichern Sie aber vorher unbedingt immer Ihren Text ab.

Hinweise zum Abtippen

Vor dem Abtippen oder späteren Wiederladen des MSE-Laders müssen Sie unbedingt folgende Zeile eingeben:
POKE 43,1: POKE 44,32: POKE 8192,0: NEW

Starten Sie das Programm mit RUN. Fehlerhafte Zeilen werden angezeigt und müssen korrigiert werden, bis der Lader zum »READY« durchläuft. Jetzt müssen Sie das fertige MSE-Programm abspeichern. Dazu brauchen Sie nur »RETURN« zu drücken, weil die erforderlichen Angaben schon auf dem Bildschirm stehen. (Kassettenbesitzer müssen in Zeile 343 die letzte Zahl in »1« abändern.) Ab jetzt können Sie »MSE V1.0« direkt, also ohne den DATA-Lader, benutzen. MSE V1.0 wird ganz normal mit »8« geladen (keine POKES notwendig).
(N. Mann / D. Weineck / gk)

MSE-Befehle:

| | |
|--------|--|
| DEL | löscht die letzte Eingabe. |
| CTRL-S | speichert das eingetippte Programm ab. |
| CTRL-L | lädt ein Programm. Start- und Endadresse werden automatisch ermittelt. |
| CTRL-M | listet den Speicherinhalt. Abbruch mit STOP-Taste, weiter mit Leertaste. |
| CTRL-N | erlaubt die Eingabe einer neuen Adresse zum Weitertippen. |
| CTRL-P | gibt ein MSE-Listing auf dem Drucker aus. |

```

100 REM ***** <091>
110 REM * <159>
120 REM * M S E LADER * <206>
130 REM * * <179>
220 REM ***** <211>
230 REM <036>
240 DIM H(75): FOR I=0 TO 9 <113>
250 H(48+I)=I: H(65+I)=I+10: NEXT <041>
260 FOR I=2048 TO 3755 : READ A# <198>
270 H=ASC(LEFT$(A#,1)): L=ASC(RIGHT$(A#,1)) <199>
280 D=H(H)*16+H(L): S=S+D: POKE I,D <219>
290 A=A+1: IF A<20 THEN NEXT: A=-1 <141>
300 PRINT " ZEILE: ";1000+Z; <011>
310 READ V : Z=Z+1: IF V=S THEN 330 <218>
320 PRINT "PRUEFSUMMENFEHLER !": STOP <138>
330 IF A<0 THEN 341 <221>
340 S=0: A=0: PRINT: NEXT <046>
341 PRINT "{CLR}P043,1:P044,8:P045,172:P046 <010>
,14
342 POKE 631,19: POKE 632,13: POKE 633,13: PO <249>
KE 198,3
343 PRINT "{3DOWN}SAVE"CHR$(34)"MSE V1.0"CH <171>
R$(34)",8
344 END <092>

```

```

350 REM : <111>
360 REM ***** <149>
370 REM * DATA * <078>
380 REM ***** <169>
390 REM : <151>
1000 DATA 00,0B,0B,0A,00,9E,32,30,36,31,00 <119>
,00,00,A2,0B,A9,36,85,A4,A9, 1247
1001 DATA 0B,85,A5,A9,00,85,A6,A9,B0,85,A7 <054>
,A0,00,B1,A4,91,A6,CB,D0,F9, 2888
1002 DATA E6,A5,E6,A7,CA,D0,F2,A9,36,85,01 <144>
,4C,00,B0,20,D1,B1,A9,06,8D, 2787
1003 DATA 21,D0,A9,03,8D,20,D0,8D,86,02,A0 <237>
,B3,A9,74,20,FF,B1,A0,B3,A9, 2667
1004 DATA B9,20,FF,B1,A0,00,20,CF,FF,99,01 <217>
,02,C8,C9,0D,D0,F5,8B,F0,D2, 2912
1005 DATA C0,0F,90,02,A0,0E,8C,00,02,20,EA <013>
,B1,A0,B3,A9,CF,20,FF,B1,20, 2323
1006 DATA 8E,B4,85,FC,85,62,20,8E,B4,85,FB <199>
,85,61,20,A7,B4,D0,20,A0,B3, 2864
1007 DATA A9,E5,20,FF,B1,20,8E,B4,85,60,20 <091>
,8E,B4,85,5F,20,A7,B4,D0,0A, 2624
1008 DATA A5,61,C5,5F,A5,62,E5,60,90,06,20 <167>
,43,B3,4C,3A,B0,A9,AA,A0,00, 2379
1009 DATA 91,FB,E6,FB,D0,02,E6,FC,20,3F,B2

```

Der MSE zum bequemen Abtippen von Assemblerprogrammen. Beachten Sie den Checksummer auf Seite 6.

| | | | | | |
|------|---------------------------------------|-------|--|-------|--|
| | ,90,EF,4C,FB,B4,A2,02,86,58, 3118 | <152> | 1050 DATA 45,49,4E,45,43,4B,00,0D,0D,0D,20 | | |
| 1010 | DATA A9,A6,A0,9D,20,F2,B1,20,E4,FF,F0 | <231> | ,20,20,50,52,4F,47,52,41,4D, 1102 | <117> | |
| | ,FB,C9,30,90,0C,C9,47,80,08, 2970 | | 1051 DATA 4D,4E,41,4D,45,20,3A,20,00,0D,0D | <095> | |
| 1011 | DATA C9,3A,90,0B,C9,41,80,07,C9,14,D0 | <121> | ,20,20,20,53,54,41,52,54,41, 1073 | <129> | |
| | ,0F,4C,0B,B1,20,D2,FF,4A,58, 2322 | | 1052 DATA 44,52,45,53,53,45,20,3A,20,24,00 | <095> | |
| 1012 | DATA 95,F7,C6,58,D0,D2,60,AE,8D,02,F0 | <057> | ,0D,0D,20,20,20,45,4E,44,41, 1014 | <129> | |
| | ,26,C9,0C,D0,03,4C,0B,86,C9, 2685 | | 1053 DATA 44,52,45,53,53,45,20,20,20,3A,20 | <217> | |
| 1013 | DATA 13,D0,03,4C,8B,85,C9,0D,D0,03,4C | <225> | ,24,00,92,05,20,50,52,4F,47, 1171 | <217> | |
| | ,BA,B4,C9,10,D0,03,4C,68,B5, 2282 | | 1054 DATA 52,41,4D,4D,20,3A,20,00,12,20,20 | <027> | |
| 1014 | DATA C9,0E,D0,06,20,5F,B4,4C,64,B1,4C | <208> | ,2A,2A,2A,20,46,41,4C,53,43, 1024 | <098> | |
| | ,92,80,A5,F9,20,02,B1,0A,0A, 2132 | | 1055 DATA 4B,45,20,45,49,4E,47,41,42,45,20 | <148> | |
| 1015 | DATA 0A,0A,85,F9,A5,F8,20,02,B1,05,F9 | <092> | ,2A,2A,2A,20,20,92,00,0D,0D, 1058 | <098> | |
| | ,60,C9,3A,90,02,69,08,29,0F, 1950 | | 1056 DATA 2A,2A,2A,20,45,0E,44,45,20,2A,2A | <148> | |
| 1016 | DATA 60,A6,59,E0,08,90,1F,A6,58,E0,02 | <188> | ,2A,00,13,05,20,20,12,44,92, 920 | <148> | |
| | ,B0,06,20,D2,FF,4C,8E,80,C6, 2509 | | 1057 DATA 49,53,4B,20,4F,44,45,52,20,12,54 | <035> | |
| 1017 | DATA 59,A0,14,A9,92,20,F2,B1,CA,D0,FA | <197> | ,92,41,50,45,0D,00,13,20,20, 1151 | <035> | |
| | ,84,57,68,68,4C,8B,B1,A6,D3, 2891 | | 1058 DATA 49,2F,4F,20,2D,20,46,45,48,4C,45 | <012> | |
| 1018 | DATA E0,08,80,03,4C,92,80,20,D2,FF,A6 | <049> | ,52,00,20,D1,B1,20,48,B2,A0, 1606 | <012> | |
| | ,58,E0,02,90,09,C6,59,20,D2, 2468 | | 1059 DATA B3,A9,CF,20,FF,B1,20,8E,B4,85,FC | <251> | |
| 1019 | DATA FF,C6,58,D0,F9,4C,8E,80,C6, 2509 | <035> | ,20,8E,B4,85,FB,C5,61,A5,FC, 3207 | <251> | |
| | ,4A,4A,20,59,B1,68,29,0F,C9, 2419 | | 1060 DATA E5,62,90,23,A5,FB,C5,5F,A5,FC,E5 | <112> | |
| 1020 | DATA 0A,90,02,69,06,69,30,4C,D2,FF,A2 | <073> | ,60,80,19,20,A7,B4,D0,14,60, 2860 | <112> | |
| | ,FC,9A,20,D1,B1,20,48,B2,20, 2261 | | 1061 DATA 20,A7,B4,F0,0C,85,F9,20,A7,B4,F0 | <088> | |
| 1021 | DATA EA,B1,20,9F,B2,A5,FC,20,4E,B1,A5 | <148> | ,05,85,FB,4C,EF,80,68,68,20, 2749 | <088> | |
| | ,FB,20,4E,B1,20,ED,B1,A9,3A, 2860 | | 1062 DATA 43,B3,4C,5F,B4,20,CF,FF,C9,4C,D0 | <046> | |
| 1022 | DATA A0,20,20,F2,B1,A9,00,85,59,20,8E | <233> | ,09,20,D1,B1,20,48,B2,4C,0B, 2372 | <046> | |
| | ,B0,20,ED,B1,A4,59,20,EF,B0, 2530 | | 1063 DATA B6,C9,0D,60,A9,00,85,5E,20,5F,B4 | <120> | |
| 1023 | DATA 91,FB,C8,84,59,C0,0B,90,EC,20,10 | <105> | ,20,EA,B1,20,0D,B5,24,5E,30, 2042 | <120> | |
| | ,B2,A9,12,20,D2,FF,20,8E,80, 2657 | | 1064 DATA 05,20,E4,FF,F0,FB,20,E1,FF,F0,26 | <198> | |
| 1024 | DATA 20,EF,B0,C5,FF,F0,0D,20,43,B3,A9 | <034> | ,20,9F,B2,24,5E,10,09,20,4E, 2435 | <198> | |
| | ,14,A0,14,20,F2,B1,4C,A2,B1, 2665 | | 1065 DATA B5,20,0D,B5,20,60,B5,20,33,B2,20 | <207> | |
| 1025 | DATA A9,92,20,D2,FF,20,33,B2,20,E0,B2 | <123> | ,3F,B2,90,D7,A0,B4,A9,28,20, 2190 | <207> | |
| | ,20,3F,B2,90,9F,4C,8B,85,A9, 2648 | | 1066 DATA FF,B1,20,E4,FF,C9,0D,D0,F9,A9,00 | <240> | |
| 1026 | DATA 93,20,D2,FF,A2,00,A9,03,9D,00,D8 | <237> | ,85,5E,A5,61,85,FB,A5,62,85, 3056 | <240> | |
| | ,9D,00,D9,9D,00,DA,9D,00,DB, 2476 | | 1067 DATA FC,20,E0,B2,4C,64,B1,A5,FC,20,4E | <221> | |
| 1027 | DATA E8,D0,EF,60,A9,0D,2C,A9,20,4C,D2 | <160> | ,B1,A5,FB,85,FF,20,4E,B1,A9, 3003 | <221> | |
| | ,FF,20,D2,FF,98,4C,D2,FF,20, 2965 | | 1068 DATA 20,A0,3A,20,F2,B1,A0,00,20,ED,B1 | <070> | |
| 1028 | DATA E4,FF,F0,FB,60,84,5D,85,5C,A0,00 | <077> | ,B1,FB,20,4E,B1,C8,C0,08,90, 2566 | <070> | |
| | ,B1,5C,F0,06,20,D2,FF,C8,D0, 3100 | | 1069 DATA F3,20,ED,B1,24,5E,30,03,A9,12,2C | <059> | |
| 1029 | DATA F6,60,A5,FB,85,5A,A0,00,84,5B,B1 | <156> | ,A9,20,20,D2,FF,20,10,B2,A5, 2190 | <059> | |
| | ,FB,18,65,5A,85,5A,90,02,E6, 2606 | | 1070 DATA FF,20,4E,B1,A9,92,20,D2,FF,4C,EA | <029> | |
| 1030 | DATA 5B,06,5A,26,5B,C8,C0,0B,90,EC,A5 | <219> | ,B1,A9,FF,85,88,85,B9,A9,04, 3073 | <029> | |
| | ,5A,65,5B,85,FF,60,18,A5,FB, 2467 | | 1071 DATA 85,BA,20,C0,FF,A2,FF,4C,C9,FF,20 | <189> | |
| 1031 | DATA 69,08,85,FB,90,02,E6,FC,60,A5,FB | <183> | ,CC,FF,A9,FF,4C,C3,FF,20,5F, 3315 | <189> | |
| | ,C5,5F,A5,FC,E5,60,60,A0,B3, 3106 | | 1072 DATA B4,A9,80,85,5E,20,4E,B5,20,48,B2 | <111> | |
| 1032 | DATA A9,FB,20,FF,B1,A0,01,B9,00,02,20 | <098> | ,A2,24,A9,2D,20,D2,FF,CA,D0, 2596 | <111> | |
| | ,D2,FF,CC,00,02,C8,90,F4,A9, 2692 | | 1073 DATA FA,20,EA,B1,20,EA,B1,20,60,B5,4C | <015> | |
| 1033 | DATA 10,ED,00,02,AA,20,ED,B1,CA,D0,FA | <236> | ,C1,B4,20,B8,B5,A6,5F,A4,60, 2812 | <015> | |
| | ,A5,62,20,4E,B1,A5,61,20,4E, 2453 | | 1074 DATA A9,61,20,D8,FF,B0,0A,20,B7,FF,29 | <201> | |
| 1034 | DATA B1,20,ED,B1,A5,60,20,4E,B1,A5,5F | <038> | ,BF,D0,03,4C,FB,B4,A9,01,20, 2577 | <201> | |
| | ,20,4E,B1,A9,9F,20,D2,FF,20, 2575 | | 1075 DATA C3,FF,20,68,B6,A0,B4,A9,4F,20,FF | <237> | |
| 1035 | DATA EA,B1,24,5E,10,01,60,A9,12,20,D2 | <161> | ,B1,20,F9,B1,4C,FB,B4,20,68, 2921 | <237> | |
| | ,FF,A2,28,20,ED,B1,CA,D0,FA, 2646 | | 1076 DATA B6,A9,37,A0,B4,20,FF,B1,20,F9,B1 | <213> | |
| 1036 | DATA A9,92,4C,D2,FF,A5,D6,C9,16,80,01 | <204> | ,A2,08,C9,44,F0,06,A2,01,C9, 2717 | <213> | |
| | ,60,A9,A0,85,A4,A9,78,85,A6, 2945 | | 1077 DATA 54,D0,F1,A9,01,AB,20,BA,FF,A0,00 | <101> | |
| 1037 | DATA A9,04,85,A5,85,A7,A2,13,A0,27,B1 | <208> | ,E0,01,F0,1A,A9,40,8D,20,02, 2403 | <101> | |
| | ,A4,91,A6,88,10,F9,CA,F0,19, 2671 | | 1078 DATA A9,3A,8D,21,02,B9,01,02,99,22,02 | <127> | |
| 1038 | DATA 18,A5,A4,69,28,85,A4,90,02,E6,A5 | <251> | ,C8,CC,00,02,90,F4,C8,C8,D0, 2182 | <127> | |
| | ,18,A5,A6,69,28,85,A6,90,E0, 2503 | | 1079 DATA 0C,B9,01,02,99,20,02,C8,CC,00,02 | <025> | |
| 1039 | DATA E6,A7,4C,B6,B2,A9,91,4C,D2,FF,A9 | <000> | ,D0,F4,98,A2,20,A0,02,4C,BD, 2018 | <025> | |
| | ,0F,8D,18,D4,A9,00,8D,05,D4, 2776 | | 1080 DATA FF,20,B8,B5,A5,BA,C9,08,90,33,A6 | <022> | |
| 1040 | DATA A9,F7,8D,06,D4,A9,11,8D,04,D4,A9 | <126> | ,B9,86,57,A9,01,20,C3,FF,A9, 2800 | <022> | |
| | ,32,8D,01,D4,A9,00,8D,00,D4, 2413 | | 1081 DATA 60,85,B9,20,C0,FF,B0,28,A5,BA,20 | <053> | |
| 1041 | DATA A0,80,20,0F,B3,A9,10,8D,04,D4,60 | <240> | ,B4,FF,A5,B9,20,96,FF,20,A5, 2911 | <053> | |
| | ,A2,FF,CA,D0,0D,88,D0,FB,60, 2914 | | 1082 DATA FF,85,61,A5,90,4A,4A,B0,13,20,A5 | <214> | |
| 1042 | DATA A9,0F,8D,18,D4,A9,2D,8D,05,D4,A9 | <119> | ,FF,85,62,20,AB,FF,A5,57,85, 2663 | <214> | |
| | ,A5,8D,06,D4,A9,21,8D,04,D4, 2385 | | 1083 DATA B9,A9,00,20,D5,FF,90,03,4C,A3,B5 | <131> | |
| 1043 | DATA A9,07,8D,01,D4,A9,05,8D,00,D4,A0 | <078> | ,86,5F,84,60,A5,BA,C9,01,D0, 2639 | <131> | |
| | ,FF,20,09,B3,A9,20,8D,04,D4, 2250 | | 1084 DATA 0A,AD,3D,03,85,61,AD,3E,03,85,62 | <120> | |
| 1044 | DATA A9,00,8D,01,D4,8D,00,D4,60,38,20 | <175> | ,4C,FB,B4,A9,13,20,D2,FF,A2, 2300 | <120> | |
| | ,F0,FF,8A,48,98,48,18,A0,06, 2179 | | 1085 DATA 1C,20,ED,B1,CA,D0,FA,60, 1230 | <214> | |
| 1045 | DATA A2,18,20,F0,FF,A0,B4,A9,0A,20,FF | <093> | | | |
| | ,B1,20,12,B3,20,E4,FF,F0,FB, 2931 | | | | |
| 1046 | DATA A2,1D,A9,14,20,D2,FF,CA,D0,FA,68 | <088> | | | |
| | ,A8,68,AA,18,4C,F0,FF,0D,0D, 2704 | | | | |
| 1047 | DATA 0D,20,20,20,20,20,20,4D,41,53 | <216> | | | |
| | ,43,48,49,4E,45,4E,53,50,52, 1144 | | | | |
| 1048 | DATA 41,43,48,45,20,2D,20,45,44,49,54 | <038> | | | |
| | ,4F,52,20,0D,0D,20,20,20, 1023 | | | | |
| 1049 | DATA 20,20,20,20,56,4F,4E,20,4E,2E,4D | <206> | | | |
| | ,41,4E,4E,20,26,20,44,2E,57, 1128 | | | | |

© 64'er

MSE (Schluß). Dieses Listing können Sie (müssen aber nicht) mit dem neuen Checksummer 64 V3 in diesem Heft eingeben.

Komfortable Befehls- erweiterung

»Basic-Expansion« ist ein universelles Programm, das den Befehlssatz des C 64 um 42 neue Kommandos erweitert, die das Arbeiten mit Floppy und Grafik enorm vereinfachen.

Das Besondere an dieser Erweiterung ist zudem, daß sie sich »unter« dem Basic-ROM befindet. Trotz der Länge von 6625 Byte des Maschinenprogramms werden nur 260 Byte des freien Basic-RAM benötigt.

Nachdem das Programm (siehe Listing) mit »SYS 40700« gestartet wurde, erscheint nach kurzer Zeit die Einschaltmeldung von »Basic-Expansion«. 42 neue Befehle (siehe Tabelle) können von nun an ganz normal verwendet werden.

Hier die neuen Befehle im Überblick:

Bei der Beschreibung gelten folgende Symbole:

ga: Geräteadresse

lfn: logische Filenummer

fa: Farbe (1 bis 16, 1=schwarz, 2=weiß, 3=rot, etc.); der Farbcode entspricht genau der Anordnung auf der Tastatur (CTRL 1 = schwarz, CTRL 2 = weiß etc.)

PLOT x,y,fa: Punkt mit den Koordinaten (x,y) und der Farbe fa setzen (nur im Grafikmodus)

LINE a, b, x, y, fa: Linie von Punkt (a,b) bis Punkt (x,y) in der Farbe fa zeichnen

SETARC Startwinkel, Endwinkel, Schrittweite, Mittelpunkt x, Mittelpunkt y, Radius in x-Richtung, Radius in y-Richtung, Farbe fa: Diese Routine zeichnet einen Kreisbogen (im Bogenmaß) um einen Mittelpunkt.

Beispiel: »SETARC 0,2 * 3.14,0.1,200,100,100,3« zeichnet einen vollen Kreis um den Mittelpunkt (100,100) mit dem Radius 100 in roter Farbe.

Diese Zeichenroutine kann jedoch nicht nur Punkte setzen, sondern auch wieder löschen. Dazu setzt man fa einfach auf Null.

Beispielsweise löscht »PLOT 100,100,0« den Punkt (100,100), weil der Farbcode gleich Null ist. Das gilt auch für LINE und SETARC.

GRAPHICS schaltet die hochauflösende Grafik ein (320x200 Punkte).

NORM ist der Gegenbefehl zu GRAPHICS und schaltet wieder zurück in den Textmodus.

GCLEAR löscht eine hochauflösende Grafik.

GCOL r, h setzt die Farbe r als Rahmenfarbe und h als Hintergrundfarbe für die hochauflösende Grafik.

GSAVE "PRG-NAME", ga speichert die hochauflösende Grafik auf Kassette oder Diskette.

GLOAD "PRG-NAME", ga lädt eine hochauflösende Grafik von Diskette oder Kassette. GSAVE und GLOAD speichern, beziehungsweise laden auch den Farbspeicher der hochauflösenden Grafik.

COLOUR r, h arbeitet im Textmodus und setzt die Farben für Rahmen (r) und Hintergrund (h).

HBLOCK l, fa zeichnet im Textmodus einen horizontalen Balken der Länge l in der Farbe fa ab der aktuellen Cursorposition und setzt den Cursor eine Zeile tiefer. l darf nur Werte zwischen 0 und 319 annehmen.

VBLOCK l, fa arbeitet wie HBLOCK, nur in vertikaler Richtung. Zugelassen sind Werte zwischen 0 und 199.

CURSCOL fa ändert die Cursorfarbe, das heißt man kann so die Schrittfarbe wählen.

SCROLL r, uz, oz scrollt den Bildschirm zwischen den Zeilen uz und oz in der Richtung, die der Parameter r angibt. Dabei bedeutet: 0 nach rechts; 1 nach oben; 2 nach links und 3 nach unten. uz ist die untere und oz die obere Zeile, die beide im Bereich zwischen 0 und 24 liegen.

DOKE a, b bedeutet ein »Doppelbyte-POKE«, das heißt, a und b sind 2-Byte-Zahlen. In die Speicherstelle a wird das Low-Byte von b gePOKEt, entsprechend wird der Speicherstelle a+1 das High-Byte von b zugewiesen.

SIZE gibt die Speicherplatzbelegung aus: gesamter Basic-Speicher, Programm, Variablen, Arrays, Strings, freier Basic-Speicher.

RESTORE n setzt den DATA-Zeiger auf die Zeile n oder auf die nächste darauffolgende Zeile. n muß dabei eine Zeilennummer sein. Der normale RESTORE-Befehl funktioniert natürlich auch noch.

ON...RESTORE a, b, c, d.. läuft ähnlich ab wie ON..GOTO. Je nach dem Ergebnis des Ausdrucks nach dem ON-Befehl wird der DATA-Zeiger auf die Zeile a, b, c,.. positioniert.

MERGE "PRG-NAME", ga hängt ein Basic-Programm von Diskette oder Kassette mit Geräteadresse ga an ein im Speicher befindliches Basic-Programm an und sortiert die Basic-Zeilen nach ihren Zeilennummern.

GEN "String" arbeitet nur im Programm-Modus. Dieser Befehl erzeugt eine neue Basic-Zeile in einem Basic-Programm. Der String muß am Anfang eine Zeilennummer und danach den Basic-Text enthalten. Zum Beispiel erzeugt der Befehl GEN "10 DATA 1,2,3" die Basic-Zeile »10 DATA 1,2,3« im Programm. Nach der Erzeugung fährt das Programm mit der auf den GEN-Befehl folgenden Basic-Zeile fort. Der Computer springt nach dem GEN-Befehl immer in die nächste Zeile, so daß Basic-Befehle in derselben Zeile nach dem GEN-Befehl nicht ausgeführt werden. Zu beachten ist aber, daß alle Variablenwerte gelöscht werden und daß der String nur 80 Zeichen umfassen darf. Und noch ein Hinweis: Enthält der String nur eine Zeilennummer, so wird die entsprechende Basic-Zeile gelöscht. In keinem Falle sollte der GEN-Befehl in einem Unterprogramm verwendet werden, da er den Basic-Stack manipuliert.

DUMP listet alle verwendeten Variablen tabellarisch auf. Lediglich die Arrays werden dabei nicht berücksichtigt.

ARRAY gibt sämtliche Arrays mit Werten aus. Die Ausgabe kann durch einmaligen Druck der SHIFT-Taste angehalten werden. Dann wartet das Programm so lange, bis die Commodore-Taste gedrückt wird. Dies gewährleistet, daß der Benutzer alle Arraywerte genauestens ablesen kann.

RENUM z, s numeriert ein Basic-Programm neu. RENUM ändert neben den Zeilennummern die Adressen bei GOTO, GO-SUB, THEN, RESTORE, RUN sowie bei sämtlichen ON..GOTO, ON..GOSUB oder ON..RESTORE-Ausdrücken. z gibt die erste neue Zeilennummer, s die Schrittweite an. RENUM berücksichtigt ebenfalls, ob möglicherweise zu große Zeilennummern erreicht werden. Bevor die Zeilennummern verändert werden, wird geprüft, ob die zu erzeugenden Zeilennummern auch im Bereich 0 bis 64000 liegen, andernfalls wird eine Fehlermeldung ausgegeben. Die Schrittweite 0 wird auch nicht akzeptiert. RENUM kann im Programm-Modus verwendet werden. Dabei werden alle Variablenwerte gelöscht. Wie beim GEN-Befehl springt der Computer nach dem RENUM-Befehl automatisch in die nächste Basic-Zeile. Der RENUM-Befehl kann zwei Fehlermeldungen ausgeben: »US-ERROR IN xx« bedeutet, daß im Programm eine Sprungadresse nicht vorhanden war. »US« steht für »UNDEFINED STATEMENT«. Der RENUM-Befehl unterbricht jetzt aber nicht einfach, sondern ersetzt die nicht-definierte Sprungadresse durch die Adresse

derjenigen Basic-Zeile, die als nächste auf die nicht-definierte folgen würde. Die zweite Fehlermeldung lautet: »SN-ERROR IN xx«, welche anzeigt, daß bei einer Sprungadresse eine zu große Zahl angegeben wurde. Hier verfährt der Befehl so, daß statt der fehlerhaften Adresse die Adresse der ersten Basic-Zeile im Programm eingesetzt wird. Dadurch wird verhindert, daß ein Programm durch die falsche Adresse möglicherweise abstürzt. Die Zeilennummern xx in den Fehlermeldungen gibt natürlich die neue Adresse der Zeile an, in der ein Fehler auftrat.

OLD ist der Gegenbefehl zu **NEW**. So können durch **NEW** oder **Reset** gelöschte Programme wieder »zurückgeholt« werden.

CURSOR s, z („**Drucktext**“) ist der Positionierbefehl für den Cursor. **s** gibt die Spalte (0 bis 39) und **z** die Zeile (0 bis 24) der gewünschten Position an. Es existieren zwei **CURSOR**-Varianten:

CURSOR s, z setzt nur den Cursor.

CURSOR s, z, „**String**“ dagegen positioniert den Cursor auf die Position (**s,z**) und druckt an diese Stelle den Text »String«. Wird also hinter der Zeilenangabe noch ein Komma gesetzt, so wird praktisch wie beim **PRINT**-Befehl fortgefahren.

Die nächsten Befehle beziehen sich nur auf das 1541-Floppy-Laufwerk. **gn** ist hierbei die Geräteadresse der Diskettenstation.

DIRgn listet das Directory der Diskette auf dem Bildschirm, ohne ein im Speicher befindliches Programm zu löschen. Der Benutzer kann während des Auflistens mit der gedrückten **SHIFT**-Taste das Listen so lange verzögern, bis die **SHIFT**-Taste wieder entriegelt wird. Die **Commodore**-Taste unterbricht das Auflisten des Inhaltsverzeichnisses ebenfalls, nur wird damit das Listen einfach abgeschlossen und der Computer verarbeitet den nächsten Befehl.

RECORD #, lfn, rn, (bp) ist der Positionierbefehl für relative Dateien. **lfn** bezeichnet die Filenummer des geöffneten relativen Files. **rn** stellt die Recordnummer (1-65535) und **bp** eventuell die Position innerhalb dieses Records (1-254) dar. Ohne Angabe von **bp** wird immer der Wert 1 angenommen.

DISC "Befehl", gn übermittelt dem 1541-Laufwerk **DOS**-Befehle wie zum Beispiel

DISC "IO", gn

Hier wird die Diskette in der Floppy-Station mit der Geräteadresse **gn** initialisiert. Mit **DISC** können alle **DOS**-Befehle, die an den Befehlskanal gehen, übertragen werden. Die Befehle müssen aber innerhalb der Anführungsstriche stehen.

DISCGET # lfn, Variable\$, (Länge l) ist ein erweiterter **INPUT #**-Befehl. **lfn** bezeichnet die logische Filenummer; danach muß eine Stringvariable angegeben werden. Diese Routine liest nun bis zum nächsten **RETURN**-Code Zeichen ein und speichert sie unter der Stringvariablen ab. Dabei können bis zu 255 Zeichen eingelesen werden. Wird zusätzlich noch eine Länge **l** mit angegeben, so liest der Befehl genau **l** Zeichen ein, so daß die Stringlänge der Variablen vorbestimmt wird. Dies erweist sich beispielsweise beim Einlesen von Records aus relativen Dateien als sehr nützlich, denn hier kann der Parameter **l** die Länge des Records darstellen.

PROTECT "File-Name", gn schützt ein File auf der Diskette gegen den **SCRATCH**-Befehl. Files, die so geschützt sind, werden im Directory gesondert gekennzeichnet. Mit dem **SCRATCH**-Befehl können diese Files nicht mehr gelöscht werden.

FREE "File-Name", gn ist der Gegenbefehl zum **PROTECT**-Befehl. Hiermit werden geschützte Files auf der Diskette wieder freigeben und können nun mit **SCRATCH** gelöscht werden. Für **PROTECT** und **FREE** gibt es auch die Möglichkeit, den Namen des Files mit dem Jokerzeichen "*" abzukürzen.

DOSTYPE "Stringzeichen", gn verändert das **DOS**-Flag der Floppy-Station. Das Zeichen »A« kennzeichnet das **DOS** der VC 1541. Wird jetzt **DOSTYPE "B"**, **gn** eingegeben, so hat der Benutzer das **DOS**-Flag verändert. Dies aber bedeutet,

daß man nichts mehr auf die Diskette schreiben kann. Bei jedem Schreibbefehl wird geprüft, ob auch das »richtige« **DOS**-Flag vorhanden ist (**DOS»A«**). Trifft dies nicht zu, so kann eine Diskette nicht mehr beschrieben werden. Durch »**DOSTYPE "A", gn**« erzeugt man wieder das richtige **DOS**-Flag, so daß die Diskette wieder beschrieben werden kann. Insofern bedeutet der **DOSTYPE**-Befehl eine Art Schreibschutz der Diskette. Anzumerken ist noch, daß die Routine immer nur das erste Zeichen im Stringausdruck berücksichtigt.

@: Dieser Befehl dient demjenigen, der mit dieser Basic-Erweiterung noch nicht zufrieden ist, als zusätzlicher Erweiterungsbefehl. Der **@**-Befehl bewirkt normalerweise nichts. Der Zeiger in der Sprungtabelle der Basic-Befehle ist auf ein **RETURN (RTS)** positioniert. Der Benutzer kann nun ein Maschinenprogramm mit dem Klammeraffen aufrufen. Dazu muß er die Werte in der Sprungtabelle für den **@**-Befehl ändern. Der Sprungvektor liegt in den Speicherstellen 40221 und 40222 (dezimal). Normalerweise steht dort der Vektor 40804, weil in der Speicherstelle 40805 ein **RTS**-Code enthalten ist. Der Sprungvektor kann vom Benutzer verändert werden (durch die **POKE**- oder **DOKE**-Befehle), indem der Vektor auf die Einsprungadresse gesetzt wird. Wenn ein Programmierer beispielsweise einen neuen Befehl als Maschinenprogramm ab Adresse 828 (Kassettenpuffer) angelegt hat, so **POKE**t er das Low-Byte von (828—1) in 40221 ein und speichert das High-Byte von (828—1) in 40222 ab. Nun kann die neue Routine mit **@** angesprochen werden. Durch Zuweisung des ursprünglichen Wertes von (40805—1) in die Speicherstellen der Sprungtabelle wird der neue Befehl wieder aufgehoben.

Soweit die neuen Befehle. Nun kommen wir zu den neuen Funktionen der Erweiterung:

RAM (xx) gibt den Inhalt der **RAM**-Speicherstelle **xx** an und ähnelt dadurch sehr der **PEEK**-Funktion. Doch die **RAM**-Funktion liest die Werte immer aus dem **RAM** aus. Zusätzlich wird im Bereich \$D000 bis \$DFFF (hexadezimal) der Charaktergenerator ausgelesen, so daß die Werte für die Zeichen aus dem Charakter-ROM aufgelistet werden können.

CVAL ("String") ähnelt der **VAL**-Funktion, kann jedoch auch Rechnungen auswerten. So ergibt »**CVAL("5*6+3")**« als Beispiel den Wert 33, wogegen »**VAL("5*6+3")**« gleich 5 ist. **CVAL** verarbeitet alle Funktionen, jedoch darf der String nicht länger als 80 Zeichen sein.

ROUND(x,y) rundet eine Zahl **x** auf **y** Nachkommastellen. Somit kann man eine Fließkommazahl auf eine ganze Zahl runden, wenn **y=0** ist.

&(xxxx) wandelt die dezimale Zahl **xxxx** (oder einen numerischen Integer-Ausdruck) in den entsprechenden hexadezimalen String, welcher zwei oder vier Zeichen umfaßt, je nachdem ob der Wert **xxxx** kleiner oder größer als 256 ist. Der Ausdruck **xxxx** muß eine positive Zwei-Byte-Zahl darstellen. Beispiel: »**PRINT &(15)**« ergibt »OF«.

!("yyyy") ist das Gegenstück zur **&**-Funktion. Sie wandelt einen hexadezimalen String in die entsprechende dezimale, positive Zwei-Byte-Zahl. Der String muß zwei oder vier Zeichen umfassen, je nachdem ob der Hex-Wert kleiner oder größer als "FF" ist.

TEST(x,y) fragt ab, ob in der hochauflösenden Grafik der Punkt mit den Koordinaten (**x,y**) gesetzt ist. Trifft dies zu, so erhält man das Ergebnis -1, sonst ergibt sich der Wert 0.

INSTR(1. String, 2. String) oder **INSTR(1. String, 2. String, a)** untersucht, ob der zweite String im ersten String enthalten ist und gibt die Position an, ab der der zweite String gefunden wird. Ist der zweite String nicht im ersten String enthalten, so wird eine Null ausgedruckt. Der Parameter **a**, der zusätzlich mit angegeben werden kann, gibt die Position im ersten String an, ab der getestet werden soll, ob der zweite String ein Teil des ersten ist. Wird **a** nicht mit angegeben, so wird er automatisch auf 1. gesetzt, das heißt es wird ab der ersten Position im ersten String untersucht.

Basic-Erweiterungen

DS\$(ga) liest den Fehlerkanal der Floppy-Station aus, wobei ga die Geräteadresse angibt.

DS(ga) ermittelt nur die Nummer der Fehlermeldung der Floppy-Station mit der Geräteadresse ga.

HEADBYT ("File-Name", ga, n) bezieht sich auf das n-te Byte im Eintrag eines Files im Inhaltsverzeichnis. Die Geräteadresse ist ga. Der String »File-Name« stellt den Namen des Files dar. Jeder Eintrag eines Files enthält 30 Byte mit wichtigsten Informationen über Länge des Files, File-Typ etc. Diese Bytes können nun einzeln mit der HEADBYT-Funktion ausgelesen werden. Beispielsweise stellt Byte 0 im Eintrag eines Files den File-Typ dar. Mit der HEADBYT-Funktion kann der Benutzer den File-Typ auslesen, wobei n=0 sein muß. Bei einer relativen Datei befindet sich im 21. Byte des Eintrags die Recordlänge, so daß mit HEADBYT die Recordlänge schnell festgestellt werden kann (bei n=21). Zu beachten ist, daß der Parameter n nur im Bereich 0 bis 29 liegen darf. Welche Bedeutung nun die einzelnen Bytes im Eintrag eines Files haben, läßt sich anhand des Bedienungshandbuches für die 1541 feststellen. Wichtig ist auch noch, daß bei der Angabe des File-Namens mit dem Jokerzeichen »*« gearbeitet werden kann.

DEEK(xxxx) stellt ein Doppelbyte-PEEK dar. DEEK (xxxx) ist die Gegenfunktion zu DOKE. Es werden die Inhalte der Speicherstellen xxxx und xxxx+1 als positive Zwei-Byte-Zahl dargestellt. xxxx muß die Adresse des Low-Bytes markieren; das Ergebnis ist eine Zahl zwischen 0 und 65535. Die DEEK-Funktion liest immer aus dem RAM aus. In den Bereichen des Basic-Interpreter-ROMs und dem Betriebssystem-ROMs wird ebenfalls immer das RAM ausgelesen. Das ROM kann nur mit PEEK ausgelesen werden. DEEK unterscheidet sich von der RAM-Funktion nur im Bereich \$D000 bis \$DFFF. Hier hat DEEK keinen Zugriff auf den Charaktergenerator.

Soweit sind nun alle neuen Befehle und Funktionen beschrieben worden. Eines sollte man jedoch unbedingt beachten: Bei der Benutzung des THEN-Befehls muß darauf geachtet werden, daß nach dem THEN ein Doppelpunkt »:« folgt, um danach die neuen Basic-Befehle benutzen zu können. Wird der Doppelpunkt weggelassen, so gibt der Computer »SYNTAX ERROR« aus, wenn nach THEN ein neuer Basic-Befehl verwendet wird.

Hinweise zur Eingabe:

Geben Sie das Programm mit dem MSE ein (siehe Seite 8). Geladen wird es dann mit »LOAD"...",8,1«. Danach geben Sie bitte NEW ein. Der Start erfolgt mit »SYS 40700«.

Sollte während der Benutzung der Erweiterung ein RESET ausgelöst werden, so erscheint wieder das gewohnte Bild des Basic V2, was jedoch nicht heißt, daß die Basic-Erweiterung gelöscht worden ist. Die Erweiterung ist nicht gegen RESETs des Computers geschützt. Um das zu erreichen, hätte der gesamte Bereich von hexadezimal \$8000 bis \$A000 vor Überschreiben geschützt werden müssen, was aber einem Basic-Speicherverlust von 8 KByte gleichkäme. Doch eben dieser Speicherverlust sollte vermieden werden. Dem Benutzer sollte noch möglichst viel Speicherplatz für Basic-Programme und Daten zur Verfügung stehen. Doch wenn der Benutzer nach einem RESET sofort »SYS 40700« eingibt, so ist die Basic-Erweiterung wieder aktiviert, wenn nicht schwerwiegende Programmfehler einen Teil der Erweiterung verändert haben.

(Bernd Stuke/ev)

| | |
|----------|-------------------|
| PLOT | DOSTYPE |
| LINE | DIR |
| SETARC | RECORD # |
| GRAPHICS | DISCGET # |
| NORM | DISC |
| GCLEAR | PROTECT |
| GCOL | FREE |
| COLOUR | CURSOR |
| GLOAD | @ (KLAMMERAFFE) |
| GSAVE | RAM |
| HBLOCK | CVAL |
| VBLOCK | ROUND |
| CURSCOL | ! (AUSRUFZEICHEN) |
| SCROLL | & (UND-ZEICHEN) |
| DOKE | TEST |
| SIZE | INSTR |
| MERGE | DSS |
| GEN | DS |
| DUMP | HEADBYT |
| ARRAY | DEEK |
| RENUM | RESTORE |
| OLD | ON RESTORE |

Tabelle 1. Liste aller neuen Befehle und Funktionen

```

programm : basic-ex/mse 8500 9f0c
-----
8500 : a9 fc a0 9e 85 5a 84 5b 17
8508 : a9 dd a0 b8 85 58 84 59 bf
8510 : 20 bf a3 4c fc 9e 00 e9 1b
8518 : 00 f6 54 78 a2 ff 9a d8 fe
8520 : 20 a3 fd 20 50 fd 20 15 35
8528 : fd 20 5b ff 58 a2 fc a0 dc
8530 : 9e 20 2d fe a9 00 8d 20 1b
8538 : d0 8d 21 d0 a9 05 8d 86 37
8540 : 02 20 53 e4 20 bf e3 a2 99
8548 : 07 bd 9c 9f 9d 04 03 ca e4
8550 : 10 f7 a9 73 85 22 a9 e4 0e
8558 : 85 23 a2 28 20 25 ab a9 4a
8560 : a4 85 22 a9 9f 85 23 a2 7d
8568 : 1c 20 25 ab a9 98 a0 e4 ff
8570 : 20 2d e4 4c 9d e3 20 8a 78
8578 : 9f 20 26 a2 08 48 a9 37 dd
8580 : 85 01 68 28 60 20 8a 9f 15
8588 : 4c e5 a1 20 8a 9f 20 55 04
8590 : a2 20 5d 9f 4c ae a7 20 a7
8598 : 8a 9f 20 1d a1 4c 5d 9f cf
85a0 : 8e f6 07 8c f7 07 20 96 62
85a8 : 9f 08 48 a9 36 d0 d1 8e e1
85b0 : f6 07 8c f7 07 20 5d 9f 72
85b8 : 6c f6 07 78 9f 66 9f 6c f5
85c0 : 9f 57 9f 20 20 20 20 20 bb
85c8 : 20 20 20 45 58 50 41 4e 53
85d0 : 44 45 44 20 42 59 20 42 c0
85d8 : 2e 20 53 54 55 4b 45 20 7b
85e0 : 5d 9f be 80 a0 4c 8a 9f a2
85e8 : 20 5d 9f 20 a2 bb 4c 8a f1
85f0 : 9f 20 5d 9f 20 d4 bb 4c 1b
85f8 : 8a 9f 20 5d 9f 20 67 b8 10
8600 : 4c 8a 9f 20 5d 9f 20 28 21
8608 : ba 4c 8a 9f 20 5d 9f b9 5e
8610 : 9d a0 4c 8a 9f 20 5d 9f 11
8618 : b9 9e a0 4c 8a 9f 31 55 e7
8620 : 33 34 35 50 4c 4f d4 4c f0
8628 : 49 4e c5 53 45 54 41 52 15
8630 : c3 47 52 41 50 48 49 43 47
8638 : d3 4e 4f 52 cd 47 43 4c 0d
8640 : 45 41 d2 47 43 4f cc 43 2c
8648 : 4f 4c 4f 55 d2 47 4c 4f 73
8650 : 41 c4 4f 53 41 56 c5 48 9e
8658 : 42 4c 4f 43 cb 56 42 4c 0e
8660 : 4f 43 cb 43 55 52 53 43 68
8668 : 4f cc 53 43 52 4f 4c cc 05
8670 : 44 4f 4b c5 53 49 5a c5 5c
8678 : 4d 45 52 47 c5 47 45 ce 2f
8680 : 44 55 4d d0 41 52 52 41 4f
8688 : d9 52 45 4e 55 cd 4f 4c 3f
8690 : c4 44 4f 53 54 59 50 c5 91
8698 : 44 49 d2 52 45 43 4f 52 d0
86a0 : 44 a3 44 49 53 43 47 45 e7
86a8 : 54 a3 44 49 53 c3 50 52 41
86b0 : 4f 54 45 43 d4 46 52 45 36
86b8 : c5 43 55 52 53 4f d2 c0 3b
86c0 : 52 41 cd 43 56 41 cc 52 d6
86c8 : 4f 55 4e c4 a6 a1 54 45 41
86d0 : 53 d4 49 4e 53 54 d2 44 55
86d8 : 53 a4 44 d3 48 45 41 44 45
86e0 : 42 59 d4 44 45 45 cb 00 3a
86e8 : 8c a6 98 a6 42 ab 4a a6 ba
86f0 : 70 a6 7b a7 a5 a7 af b0 3f
86f8 : bb a8 12 a9 ca b7 31 b8 62
8700 : c4 b0 74 a9 f9 b0 bd b1 ee
8708 : 69 b2 78 b2 ea b6 dc b5 a2
8710 : e0 a2 6e b0 06 ae 96 b3 8b
8718 : 10 b4 6f ad bb ac ba ae 7d
8720 : bd ae cf b0 64 9f 3b b1 d2
8728 : 5f b1 56 b4 b2 84 b8 b5 f6
8730 : 0c ab f1 b4 db ac d8 ac 05
8738 : 3a b0 19 b1 a6 7a a0 04 10
8740 : 84 0f bd 00 02 10 07 c9 0c
8748 : ff f0 41 e8 d0 f4 c9 20 49
8750 : f0 3a 85 08 c9 22 f0 59 e4
8758 : 24 0f 70 30 c9 3f d0 04 08
8760 : a9 99 d0 28 c9 30 90 04 77
8768 : c9 3c 90 20 84 71 a0 00 ce
8770 : 84 0b 88 86 7a ca c8 e8 60
8778 : bd 00 02 38 f9 04 a0 f0 e1
8780 : f5 c9 80 d0 33 a5 0b 18 51
    
```

Listing »Basic-Expansion«. Bitte mit dem MSE eingeben. Beachten Sie die Eingabehinweise auf Seite 8.

8788 : 69 cc a4 71 e8 c8 99 fb e2
8790 : 01 b9 fb 01 f0 64 38 e9 74
8798 : 3a f0 04 c9 49 d0 02 85 b3
87a0 : 0f 38 e9 55 d0 9c 85 08 08
87a8 : bd 00 02 f0 df c5 08 f0 32
87b0 : db c8 99 fb 01 e8 d0 f0 52
87b8 : a6 7a e6 0b c8 b9 03 a0 5e
87c0 : 10 fa b9 04 a0 d0 b1 a0 d5
87c8 : 00 84 0b 88 86 7a ca c8 d7
87d0 : e8 20 f6 9f 85 02 bd 00 d9
87d8 : 02 38 e5 02 f0 f1 c9 80 77
87e0 : d0 04 05 0b d0 a4 a6 7a 17
87e8 : e6 0b c8 20 ed 9f 10 fa 9c
87f0 : 20 f6 9f d0 dc bd 00 02 4d
87f8 : 10 90 99 fd 01 c6 7b a9 fe
8800 : ff 85 7a 60 10 08 c9 ff d5
8808 : f0 04 24 f0 10 0e a2 1a 16
8810 : 8e f6 07 a2 a7 8e f7 07 0c
8818 : aa 4c 96 9f c9 cc 90 ee a5
8820 : e9 cb aa 84 49 a0 ff ca 59
8828 : f0 08 c8 b9 04 a0 10 fa 01
8830 : 30 f5 c8 b9 04 a0 10 f0 58
8838 : a2 ef a0 a6 4c 90 9f 20 d7
8840 : d2 ff 4c 13 a2 a9 00 85 0a
8848 : 0d 20 73 00 c9 cc b0 0c 20
8850 : 68 68 20 79 00 a2 8d a0 b0
8858 : ae 4c 90 9f c9 ff f0 f0 87
8860 : c9 eb 90 1d 38 e9 c9 0a 01
8868 : a8 b9 ca a0 48 b9 c9 a0 6e
8870 : 48 4c 73 00 20 73 00 f0 3b
8878 : 21 c9 cc 90 0b c9 e9 90 93
8880 : e3 a2 08 a0 af 4c 90 9f a9
8888 : c9 91 f0 43 c9 8c f0 0b 99
8890 : 20 79 00 a2 ed a0 a7 4c dc
8898 : 90 9f 60 20 73 00 d0 07 9c
88a0 : a2 1d a0 a8 4c 81 9f 20 9e
88a8 : a2 a2 20 a9 a2 38 a5 5f 1a
88b0 : e9 01 a4 60 a2 24 8e f6 c2
88b8 : 07 a2 a8 8e f7 07 4c 96 22
88c0 : f7 a2 6b a0 a9 4c 81 9f e1
88c8 : a2 13 a0 a6 4c 81 9f 20 80
88d0 : 73 00 20 3c a6 48 c9 8c c0
88d8 : f0 0b c9 8d f0 07 c9 89 f3
88e0 : f0 03 4c 62 a2 c6 65 0d 49
88e8 : 0b 68 c9 8c f0 ad a2 ef 12
88f0 : 38 4c 76 a2 20 73 00 20 1e
88f8 : a2 a2 c9 2c f0 e7 68 60 94
8900 : 20 12 b1 8c c8 bf 8d c9 7b
8908 : bf 20 18 a6 20 12 b1 8c 25
8910 : ca bf 8d cb bf d0 03 98 56
8918 : f0 10 a9 ff aa 20 27 a5 0e
8920 : c9 fa 90 d0 d0 04 0e fa d3
8928 : 90 07 a2 48 a0 b2 4c 90 df
8930 : 9f 20 59 b3 a5 3a a6 39 e5
8938 : 20 27 a5 8e d2 bf 8d c3 30
8940 : bf 20 ec a5 20 ac a5 e6 cb
8948 : 7a d0 02 e6 7b a0 01 b1 ac
8950 : 7a d0 03 4c 99 a4 c8 b1 c2
8958 : 7a 8d ce bf c8 b1 7a 8d 63
8960 : cf bf 98 18 65 7a 85 7a 6d
8968 : 90 02 e6 7b 20 73 00 a0 01
8970 : 00 b1 7a f0 d2 c9 22 d0 ab
8978 : 15 85 07 e6 7a d0 02 e6 f2
8980 : 7b a2 d0 ad a9 20 81 9f 85
8988 : b1 7a d0 d6 f0 13 c9 8f 73
8990 : f0 08 c9 83 d0 0f a2 06 83
8998 : d0 02 a2 09 a0 a9 20 81 0e
89a0 : 9f 88 4c 43 a3 a2 05 dd 1e
89a8 : 1e a6 f0 0e ca d0 f8 c9 c2
89b0 : cb d0 b0 20 73 00 c9 a4 fd
89b8 : d0 b5 20 73 00 b0 b0 a6 6f
89c0 : 7a a4 7b 86 fb 84 fc 20 54
89c8 : a2 a2 b0 18 a9 53 20 d2 46
89d0 : ff a9 4e 20 f9 a4 20 73 68
89d8 : 00 90 fb ae c8 bf ad c9 ca
89e0 : bf 4c da a3 20 2b a5 b0 44
89e8 : 10 8e d0 bf 8d d1 bf a9 25
89f0 : 55 20 d2 ff a9 53 20 f9 b3
89f8 : a4 20 c0 a5 20 d8 a5 a5 3c
8a00 : 7a 38 e5 fb 85 02 a0 ff 7a
8a08 : c8 b9 00 01 d0 fa 98 38 85
8a10 : e5 02 f0 3e 90 67 85 02 59
8a18 : a5 7a 85 5f a5 7b 85 60 55
8a20 : a5 2d 85 5a 18 65 02 85 c8
8a28 : 58 a4 2e 84 5b 90 01 c8 be
8a30 : 84 59 a2 b8 8e f6 07 a2 22
8a38 : a3 8e f7 07 20 87 9f a5 09
8a40 : 31 85 2d a5 32 85 2e e0 fe
8a48 : 00 f0 07 20 78 b3 8a 4c 6e
8a50 : ce ac a0 ff c8 b9 00 01 f9
8a58 : f0 04 91 fb d0 f6 98 18 85
8a60 : 65 fb 85 7a a5 fc 69 00 5b
8a68 : 85 7b 20 f6 a5 20 ec a5 ec
8a70 : 20 79 00 c9 2c d0 03 4c 74
8a78 : 9b a3 4c 50 a3 49 ff 69 59
8a80 : 01 85 02 a5 7a 38 e5 02 7e
8a88 : 85 fd a5 7b e9 00 85 fe 97
8a90 : a4 02 a2 00 a5 2d 38 e5 4e

8a98 : 02 85 2d a5 2e e9 00 85 9a
8aa0 : 2e b1 fd 81 fd e6 fd d0 07
8aa8 : 02 e6 fe a5 2d c5 fd a5 d6
8ab0 : 2e e5 fe a5 2c 4c 33 a4 ee
8ab8 : a6 2b a5 2c 86 5f 85 60 1d
8ac0 : a0 01 b1 5f f0 2a c8 ad 18
8ac8 : c8 bf 91 5f c8 ad c9 bf 61
8ad0 : 91 5f 18 ad c8 bf 6d ca a2
8ad8 : bf 8d c8 bf ad c9 bf 6d 8b
8ae0 : cb bf 8d c9 bf a0 00 b1 8c
8ae8 : 5f aa c8 b1 5f 4c 9d a4 1d
8af0 : 20 78 b3 a4 3a c8 f0 19 ae
8af8 : ae d2 bf ac d3 bf e8 d0 15
8b00 : 01 c8 86 14 84 15 20 6d d6
8b08 : a5 38 a2 c5 a0 a8 4c 81 ae
8b10 : 9f a2 74 a0 a4 4c 90 9f 60
8b18 : 20 d2 ff a9 2d 20 7f ff f5
8b20 : a0 a3 a9 6b 20 b3 a5 a0 e1
8b28 : a3 a9 71 20 b3 a5 ad cf bf
8b30 : bf ae ce bf 20 27 a5 20 04
8b38 : 07 b4 a9 d0 2d d2 ff ae 9b
8b40 : d0 bf ad d1 bf 60 86 14 d7
8b48 : 85 15 20 6d a5 08 ad cb f6
8b50 : bf ae ca bf 20 c0 a5 a2 f5
8b58 : 57 a0 00 20 d2 9f ad cd 80
8b60 : bf ae cc bf 20 c0 a5 a9 93
8b68 : 57 a0 00 20 e4 9f a2 57 98
8b70 : a0 00 20 d2 9f ae c8 bf 85
8b78 : ad c9 bf 20 c0 a5 a9 57 8c
8b80 : a0 00 20 db 9f 20 15 b1 57
8b88 : a6 14 28 60 a9 00 8d cc b9
8b90 : bf 8d cd bf a5 2b a6 2c 28
8b98 : a0 01 85 5f 86 60 b1 5f f7
8ba0 : f0 27 c8 c8 a5 15 d1 5f 78
8ba8 : 90 20 f0 03 88 d0 09 a5 63
8bb0 : 14 88 d1 5f 90 14 f0 12 fa
8bb8 : 88 b1 5f aa 88 b1 5f ee b7
8bc0 : cc bf d0 03 ee cd bf b0 be
8bc8 : cf 18 60 a2 8e a0 a6 4c 31
8bd0 : 81 9f a2 1e 8e f6 07 a2 8f
8bd8 : ab 8e f7 07 4c 87 9f 85 34
8be0 : 62 86 63 a9 00 85 0d 38 64
8be8 : a2 90 a9 49 8d f6 07 a9 66
8bf0 : bc 8d f7 07 4c 87 9f a9 24
8bf8 : fd 8d f6 07 a9 bd 8d f7 eb
8c00 : 07 4c 87 9f a2 1d a0 a8 ea
8c08 : 4c 81 9f a2 33 a0 a7 1c b8
8c10 : 81 9f 20 e7 ff a5 37 a4 b9
8c18 : 38 85 33 84 34 a5 2d a4 df
8c20 : 2e 85 2f 84 30 85 31 84 6a
8c28 : 32 20 e5 a5 a2 19 86 16 d2
8c30 : a9 00 85 3e 85 10 60 a2 a2
8c38 : fd a0 ae 4c 81 9f 89 8d 11
8c40 : 8c 8a a7 a2 0c bd c8 bf a1
8c48 : 9d a1 02 ca 10 f7 60 a2 17
8c50 : 0c bd a1 02 9d c8 bf ca 98
8c58 : 10 f7 60 a2 9e a0 b7 4c 37
8c60 : 81 9f a2 f1 d0 f7 a2 eb c7
8c68 : d0 f3 a9 20 2d 11 d0 d0 e0
8c70 : 1e ad 11 d0 09 20 8d 11 ad
8c78 : d0 ad 00 dd 8d fe be 29 f9
8c80 : fc 8d 00 dd ad 18 d0 8d f9
8c88 : ff be a9 0b 8d 18 d0 60 50
8c90 : a9 20 2d 11 d0 f0 f8 ad 8b
8c98 : 11 d0 29 fd 8d 11 d0 ad 58
8ca0 : fe be 8d 00 dd ad ff be 29
8ca8 : 8d 18 d0 60 20 5f a9 20 65
8cb0 : e8 a6 20 92 a7 4c 36 a7 4b
8cb8 : 20 5f a9 8c 03 bf 8d 04 f0
8cc0 : bf 8e 05 bf 20 18 a6 20 9d
8cc8 : 5f a9 8c 00 bf 8d 01 bf 0b
8cd0 : 8e 02 bf 20 92 a7 ac 03 72
8cd8 : bf ad 04 bf ae 05 bf 20 b9
8ce0 : e8 a6 20 36 a7 ac 00 bf 4a
8ce8 : ad 01 bf ae 02 bf 4c 33 91
8cf0 : a8 20 47 a6 8a a8 a6 15 02
8cf8 : c0 c8 b0 0a a5 14 e0 01 0a
8d00 : 90 04 d0 02 c9 40 60 8c 40
8d08 : 03 bf 8d 04 bf 8e 05 bf d3
8d10 : 85 14 86 15 98 4a 4a 7d
8d18 : aa bd 39 a9 85 ad 8a 29 67
8d20 : 03 aa bd 53 a9 85 ac 98 fd
8d28 : 29 07 18 65 ac 85 ac a5 7d
8d30 : 14 29 f8 85 63 a9 e0 05 d9
8d38 : ad 85 ad 18 a5 ac 65 63 32
8d40 : 85 ac a5 ad 65 15 85 ad ab
8d48 : a5 14 29 07 a9 07 aa bd 16
8d50 : 57 a9 85 ab 60 00 00 08 6e
8d58 : 20 76 a7 a5 ab 24 97 30 ec
8d60 : 05 49 ff 31 ac 2c 11 ac fa
8d68 : 91 ac 24 97 10 1e a5 ac 2d
8d70 : 85 fd a5 ad 4a 66 fd 4a 77
8d78 : 66 fd 4a 66 fd 29 03 09 83
8d80 : c0 85 fe b1 fd 29 0f 0d 78
8d88 : 06 bf 91 fd a9 36 85 01 f6
8d90 : 58 28 a4 6f 60 a9 35 78 2c
8d98 : 85 01 60 a9 e0 85 fe a0 63
8da0 : 00 84 fd a2 20 98 91 fd bf

8da8 : c8 d0 fb e6 fe ca d0 f6 2b
8db0 : 60 20 43 a6 8a f0 0b ca b8
8db8 : 8a 0a 0a 0a 0a 8d 06 bf 0c
8dc0 : a2 80 86 97 60 20 3c a6 70
8dc8 : ca 8e 20 d0 20 43 a6 a9 06
8dd0 : c0 85 fe a0 00 84 fd ca d8
8dd8 : 8a 29 0f 8d 06 bf a2 04 5d
8de0 : b1 fd 29 f0 0d 06 bf 91 1b
8de8 : fd c8 d0 f4 e6 fe ca d0 4f
8df0 : ef 60 a5 ac 29 07 c9 07 0e
8dff : f0 05 38 a9 00 b0 04 a9 97
8e00 : 38 e6 ad 65 ac 85 ac a9 c0
8e08 : 00 65 ad 85 ad 60 30 e2 3b
8e10 : a5 ac 29 07 f0 05 18 a9 21
8e18 : ff 90 04 a9 c7 c6 ad 65 ca
8e20 : ac 85 ac a5 ad e9 00 85 a4
8e28 : ad 60 46 ab 90 0e 66 ab 77
8e30 : a5 ac c8 18 69 08 85 ac a7
8e38 : 90 02 e6 ad 60 10 eb 06 7b
8e40 : ab 90 0e 26 ab a5 ac 88 27
8e48 : 38 e9 08 85 ac b0 02 c6 0d
8e50 : ad 60 48 ad 05 bf 4a ad c8
8e58 : 04 bf 6a 4a 4a 85 6f 68 7f
8e60 : 48 38 ed 04 bf 08 8a ed 04
8e68 : 05 bf 85 6c b0 0a 68 a9 cb
8e70 : ff 69 01 48 a9 00 e5 6c 78
8e78 : 85 6a 85 6e 68 85 69 85 c5
8e80 : 6d 68 8d 04 bf 8e 05 bf 09
8e88 : 98 18 ed 03 bf 90 04 49 2b
8e90 : ff 69 fe 85 6b 8c 03 bf 5b
8e98 : 66 6c 38 e5 69 aa a9 ff 92
8ea0 : e5 6a 85 70 a4 6f b0 05 bc
8ea8 : 0a 20 1e a8 38 a5 6d 65 90
8eb0 : 6b 85 6d a5 6e e9 00 85 2f
8eb8 : 6e 84 6f 20 36 a7 e8 d0 2e
8ec0 : 05 e6 70 d0 01 60 a5 6c f1
8ec8 : b0 2e 20 ef a9 18 a5 6d 9a
8ed0 : 65 69 85 6d a5 6e 65 6a 31
8ed8 : 4c 98 a0 20 6e a9 a9 01 7b
8ee0 : 85 b9 20 d9 a8 a9 c1 8d 7f
8ee8 : 18 03 a0 20 a2 00 a9 00 7f
8ef0 : 20 d5 ff a9 47 8d 18 03 77
8ef8 : 78 c6 01 a9 00 85 62 85 0a
8f00 : 64 a9 e0 a0 20 a2 20 20 5d
8f08 : f8 a8 a9 c0 a0 a0 a2 04 75
8f10 : 20 f8 a8 e6 01 58 60 85 13
8f18 : 63 84 65 a0 00 b1 64 48 da
8f20 : b1 62 91 64 68 91 62 c8 21
8f28 : d0 f3 e6 63 e6 65 ca d0 7e
8f30 : ec 60 20 6e a9 a9 c1 8d 2c
8f38 : 18 03 20 9f a8 a2 00 a0 c6
8f40 : 44 a9 00 85 fd a9 20 85 c2
8f48 : fe a9 fd 85 b9 20 d8 ff 4b
8f50 : a9 47 8d 18 03 4c d9 a8 4f
8f58 : 00 01 02 03 05 06 07 08 67
8f60 : 0a 0b 0c 0d 0f 10 11 12 6e
8f68 : 14 15 16 17 19 1a 1b 1c 76
8f70 : 1e 1f 00 40 80 c0 01 02 3c
8f78 : 04 08 10 20 40 80 20 d2 b7
8f80 : a6 90 09 68 68 a2 f8 a0 7e
8f88 : a8 4c 81 9f 60 a2 d4 a0 5a
8f90 : e1 4c 81 9f 20 3c a6 8a 7f
8f98 : 29 03 4a b0 03 4c 03 aa df
8fa0 : f0 03 a9 ff 2c a9 01 85 9b
8fa8 : 02 20 d5 aa f0 fe 86 fc 9f
8fb0 : 84 fd 24 02 30 03 a6 fc 2c
8fb8 : 2c a6 fd a5 d1 48 a5 d2 07
8fc0 : 48 20 f4 a9 20 e3 e9 a9 a7
8fc8 : 00 85 d1 a9 bf 85 d2 a9 2b
8fd0 : 28 85 f3 a9 bf 85 f4 20 29
8fd8 : d2 e9 a5 ac 85 d1 a5 ad 77
8fe0 : 85 d2 20 24 ea 8a 18 65 89
8fe8 : 02 aa c6 fe 30 0b 20 f4 97
8ff0 : a9 20 e3 e9 20 d2 e9 30 80
8ff8 : e1 a9 00 85 ac a9 bf 85 81
9000 : ad a9 28 85 ae a9 bf 85 7f
9008 : af 20 d2 e9 68 85 d2 68 88
9010 : 85 d1 60 bd f0 ec 85 ac 33
9018 : b5 d9 29 03 0d 88 02 85 8d
9020 : ad 60 d0 01 38 08 20 d5 42
9028 : aa 28 08 90 03 c8 98 aa 29
9030 : bd a1 aa 85 62 bd bb aa 71
9038 : 85 61 28 08 90 08 e9 01 6c
9040 : 85 61 b0 02 c6 62 a5 62 bd
9048 : 29 03 0d 88 02 28 08 20 09
9050 : 5c aa 28 08 a5 61 90 0a c8
9058 : 69 27 85 61 90 0c e6 62 ac
9060 : b0 08 e9 27 85 61 b0 02 9e
9068 : c6 62 a5 62 29 03 09 d8 95
9070 : 28 08 20 5c aa c6 fe 10 2d
9078 : cd 28 60 85 62 90 03 4c 71
9080 : 81 aa a0 00 b1 61 aa a0 91
9088 : 27 b1 61 48 8a 91 61 68 75
9090 : aa 88 10 f5 18 a5 61 69 48
9098 : 28 85 61 90 02 e6 62 60 8f
90a0 : 38 a5 61 e9 28 85 61 b0 d6
90a8 : 02 c6 62 a0 28 b1 61 aa a5
90b0 : a0 01 b1 61 48 8a 91 61 4b

Basic-Erweiterungen

90b8 : 68 aa c8 c0 29 d0 f3 60 69
 90c0 : 04 04 04 04 04 04 04 05 c2
 90c8 : 05 05 05 05 05 06 06 d7
 90d0 : 06 06 06 06 07 07 07 ef
 90d8 : 07 07 00 28 50 78 a0 c8 45
 90e0 : f0 18 40 68 90 b8 e0 08 5c
 90e8 : 30 58 80 a8 d0 f8 20 48 5f
 90f0 : 70 98 c0 e8 20 43 a6 e0 72
 90f8 : 19 90 02 a2 18 86 fc 20 18
 9100 : 43 a6 e0 19 90 02 a2 18 c5
 9108 : 86 fd 8a a6 fc a4 fd 38 62
 9110 : e5 fc b0 08 a9 ff 01 dd
 9118 : a6 fd a4 fc 48 68 85 fe 61
 9120 : 60 a2 fa a0 ae a4 81 9f 37
 9128 : a2 f7 d0 f7 20 02 ab a2 ff
 9130 : 00 bd 03 bf 48 e8 e0 03 1d
 9138 : d0 f7 20 d2 a6 b0 20 48 17
 9140 : e8 a6 20 09 ab a2 03 68 51
 9148 : 9d 02 bf ca d0 f9 a0 00 8f
 9150 : a5 ab 31 ac f0 02 a9 ff 72
 9158 : a2 3c a0 bc 4c 81 9f 4c c0
 9160 : 0b a3 20 30 a6 b0 59 b3 83
 9168 : a2 a1 a0 02 86 ac 84 ad 7e
 9170 : a9 07 8d 09 bf d0 13 20 30
 9178 : 79 00 c9 2c f0 09 20 78 b2
 9180 : b3 20 24 a6 4c 62 a2 20 c4
 9188 : 73 00 20 b5 ac a5 66 30 ac
 9190 : ce a6 ac a4 ad 20 d2 9f d7
 9198 : a9 05 18 65 ac 85 ac 90 41
 91a0 : 02 e6 ad ce 09 bf d0 cf cc
 91a8 : 20 92 a7 a9 a5 a0 b1 20 97
 91b0 : c9 9f a2 e8 a0 c3 20 d2 5d
 91b8 : 9f a9 a6 a0 02 20 c9 9f 71
 91c0 : a9 ab a0 02 20 db 9f a2 4c
 91c8 : a6 a0 02 20 d2 9f a0 02 f4
 91d0 : a9 a1 20 c9 f7 a7 a6 a0 ae
 91d8 : 02 a2 5b 8e f6 07 d2 bc 80
 91e0 : 8e f7 07 20 87 9f c9 01 ce
 91e8 : d0 06 20 78 b3 4c 24 a6 4e
 91f0 : a2 fc a0 c3 20 d2 9f a2 0d
 91f8 : 64 a0 e2 20 81 9f a9 ba 9a
 9200 : a0 02 20 e4 9f a9 b0 a0 91
 9208 : 02 20 db 9f a2 f2 a0 c3 d1
 9210 : 20 d2 9f a9 fc a0 c3 20 da
 9218 : c9 9f a2 6b a0 e2 20 81 6b
 9220 : 9f a9 bf a0 02 20 e4 9f 8c
 9228 : a9 b5 a0 02 20 db 9f a2 b9
 9230 : f7 a0 c3 20 d2 9f a9 e8 0f
 9238 : a0 c3 20 78 ac 20 d5 a6 41
 9240 : b0 21 8d 04 bf 8e 05 bf 69
 9248 : 8c 03 bf a9 f2 a0 c3 20 fe
 9250 : 78 ac 20 d5 a6 b0 0c 8d 1c
 9258 : 01 bf 8e 02 bf 8c 00 bf fd
 9260 : 20 b7 a6 a9 f2 a0 c3 20 be
 9268 : c9 9f a2 e8 a0 c3 20 d2 15
 9270 : 9f a9 f7 a0 c3 20 c9 9f 99
 9278 : a2 ed a0 c3 20 d2 9f a9 1c
 9280 : ab a0 02 20 c9 9f a9 a1 83
 9288 : a0 02 20 db 9f a2 a1 a0 84
 9290 : 02 20 d2 9f a4 af ab 8d 57
 9298 : 00 bf 8c 01 bf 20 c9 9f 1e
 92a0 : a5 66 10 05 68 68 44 a1
 92a8 : ac 20 ae ac 85 15 84 14 e1
 92b0 : ac 01 bf 18 a9 05 60 00 48
 92b8 : bf 90 01 8c 20 c9 9f a5 33
 92c0 : 66 30 e1 20 ae ac c9 00 32
 92c8 : d0 da a6 65 60 a2 aa a0 63
 92d0 : b1 4c 81 9f a2 8a a0 ad 58
 92d8 : 4c 81 9f 20 4a ad 20 43 ea
 92e0 : a6 86 ba a9 6f 85 b0 20 f7
 92e8 : d5 ac b0 01 60 a2 3b a0 a9
 92f0 : a4 4c 90 9f 20 d5 f3 60 14
 92f8 : a0 00 2c a0 80 a5 02 48 85
 9300 : 84 02 20 02 ab 20 3c a6 c8
 9308 : a9 00 85 90 86 ba 8a 20 cd
 9310 : b1 ff 20 ae ff a5 90 f0 f0
 9318 : 05 20 07 f7 b0 cf 20 09 0a
 9320 : ab a5 ba 20 b4 ff a9 6f 21
 9328 : 85 b9 20 96 ff a0 00 20 aa
 9330 : a5 ff 99 0a bf c8 c9 0d 00
 9338 : d0 f5 88 84 fc 20 28 f5 13
 9340 : 24 02 30 2c ad a0 bf 29 73
 9348 : 0f a8 a9 00 c0 00 f0 06 f2
 9350 : 18 6f 0a 88 d0 fa 85 02 af
 9358 : ad 0b bf 29 0f 18 65 02 eb
 9360 : aa 68 85 02 a9 00 4c c0 2d
 9368 : a5 a2 57 a0 e2 4c 81 9f 1e
 9370 : a5 fc 20 6c ad a0 00 b9 7c
 9378 : 0a bf 91 62 c8 ca 61 d0 ec
 9380 : f6 68 85 02 a2 c0 a0 b4 b8
 9388 : 4c 81 9f a2 7d da f7 20 4f
 9390 : 3c a6 a9 1e 8d ff 07 a0 3b
 9398 : e1 20 84 9f 20 18 a6 a2 41
 93a0 : 8b a0 b0 20 81 9f 85 a9 69
 93a8 : 84 4a a2 a3 a0 20 81 b2
 93b0 : 9f 20 79 00 c9 b6 d0 3b 75
 93b8 : a5 62 48 a5 63 48 20 43 d5
 93c0 : a6 68 85 63 68 85 62 8a b9

93c8 : 20 cf ad a0 02 b9 61 00 c3
 93d0 : 91 49 88 10 f8 c8 20 a5 cc
 93d8 : ff 91 62 a5 90 d0 05 c8 22
 93e0 : c4 61 d0 f2 98 a0 00 91 99
 93e8 : 49 a6 9a 4c 40 f3 a2 75 ce
 93f0 : 4c 67 ad a0 00 20 a5 ff 07
 93f8 : 99 00 bf c9 0d f0 08 a5 7e
 9400 : 90 d0 04 c8 0d ef 88 98 f2
 9408 : 20 cf ad a0 02 b9 61 00 03
 9410 : 91 49 88 10 f8 a4 61 f0 86
 9418 : d0 88 b9 00 bf 91 62 88 be
 9420 : c0 ff d0 f6 f0 c3 20 4a 35
 9428 : ad a5 b7 d0 03 4c 62 a2 11
 9430 : a0 00 b1 bb 8d 00 bf a9 df
 9438 : 49 8d 01 bf 91 bb a9 01 20
 9440 : 85 b7 20 bf ac a9 23 a0 87
 9448 : 00 91 bb a9 62 20 6c ac d0
 9450 : a2 94 a0 ae a9 07 20 79 80
 9458 : ae a2 9b a0 ae a9 0c 20 fb
 9460 : 79 ae a2 a7 a0 ae a9 08 04
 9468 : 20 79 ae a2 00 b7 a9 9c
 9470 : 01 85 b7 a9 62 20 fd ae d1
 9478 : a2 af a0 ae a9 0c 20 79 5e
 9480 : ae a2 01 a0 bf a9 01 20 61
 9488 : 79 ae a9 62 85 b9 20 42 3a
 9490 : f6 a9 6f 85 b9 4c 42 f6 dc
 9498 : 85 b7 a9 6f 85 b9 86 bb 09
 94a0 : 84 bc a9 00 85 90 a5 ba d6
 94a8 : 20 b1 ff a5 b9 20 93 ff 40
 94b0 : 4c f6 f3 4d 2d 57 01 01 b1
 94b8 : 01 41 55 31 20 32 20 30 4a
 94c0 : 20 31 38 20 30 32 42 2d 73
 94c8 : 50 20 32 20 30 32 55 32 07
 94d0 : 20 32 20 30 20 31 38 20 c4
 94d8 : 30 30 a9 80 2c a9 00 85 b6
 94e0 : 02 20 45 af a9 32 8d 2e f8
 94e8 : af a0 02 20 fe ae 20 b4 bb
 94f0 : b8 24 02 30 03 29 bf 2c 12
 94f8 : 09 40 48 a0 02 20 fe ae c2
 9500 : 68 20 c9 b8 a9 00 8d cd 6e
 9508 : bf a8 20 fe ae ad cb bf 0a
 9510 : ac cc bf 20 14 b0 a9 62 48
 9518 : 85 b9 4c 6b ae 84 fe ad 61
 9520 : cd bf 0a 0a 0a 0a 18 da
 9528 : 65 fe a0 02 a2 2f 8e 38 ec
 9530 : f9 42 af b0 9f 48 8a 99 8b
 9538 : 3f af c8 60 72 af c0 70
 9540 : 03 d0 e9 a2 39 ad af a9 25
 9548 : 09 4c 79 ae 55 32 20 32 77
 9550 : 20 30 20 54 54 20 53 53 55
 9558 : 42 2d 50 20 32 20 43 43 01
 9560 : 43 64 0a 01 20 4a ad 20 c3
 9568 : 43 a6 86 ba a4 b7 d0 03 48
 9570 : 4c 62 a2 c0 10 90 02 a0 7d
 9578 : 10 84 b7 a0 00 b1 bb 99 7c
 9580 : 0a bf c8 c9 2a f0 0e c4 c1
 9588 : b7 d0 f2 c0 10 06 a9 70
 9590 : a0 99 0a bf c8 8c c8 bf 0b
 9598 : a9 49 8d 00 bf a9 01 85 a2
 95a0 : b7 a9 00 85 bb a9 bf 85 f0
 95a8 : bc 20 c4 ac a9 23 8d 00 25
 95b0 : bf a9 62 20 c6 ac a9 12 7d
 95b8 : 8d c9 bf 8d cb bf a9 01 2f
 95c0 : 8d ca bf 8d cb bf a9 31 28
 95c8 : 8d 2e af ad c9 bf ac ca f1
 95d0 : bf 8d cb bf 8c cd bf 20 af
 95d8 : 14 b0 a9 00 8d cd bf a8 46
 95e0 : 20 fe ae 20 b4 b8 8d c9 0a
 95e8 : bf 20 b4 be 8d ca bf a0 6b
 95f0 : 02 20 fe ae 20 b4 b8 c9 b6
 95f8 : 00 f0 16 a0 05 20 fe ae b5
 9600 : a0 00 20 b4 b8 d9 0a bf 41
 9608 : f0 1e b9 0a bf c9 2a f0 8c
 9610 : 21 ee cd bf ad cd bf c9 ef
 9618 : 08 d0 d4 ad c9 bf d0 ab a8
 9620 : 20 f7 ae a9 04 4c ce ac 54
 9628 : c9 a0 f0 06 c8 cc c8 bf d4
 9630 : d0 d0 60 20 30 b0 8e 34 b0
 9638 : af 8d 35 af 98 20 30 b0 9e
 9640 : 8e 37 af 8d 38 af a9 0c 7c
 9648 : a2 2d a0 af 4c 79 ae a2 2f
 9650 : 2f 38 ee a9 0a b0 bf 69 fb
 9658 : 3a 60 20 02 ab 20 4a ad 4b
 9660 : 20 43 a6 86 2a 20 43 a6 a3
 9668 : e0 1e 90 03 4c 0b a3 8e a5
 9670 : f0 bf 20 09 ab 20 4d af b9
 9678 : ac f0 bf c8 c8 20 fe ae 8c
 9680 : 20 b4 bf c8 20 f7 ae 68 7f
 9688 : aa a9 00 4c c0 a5 a5 2b b7
 9690 : a4 2c 85 22 84 23 a0 03 da
 9698 : c8 b1 22 d0 fb c8 98 18 74
 96a0 : 65 22 a0 00 91 2b a5 23 8e
 96a8 : 69 00 c8 91 2b 88 a2 03 fd
 96b0 : e6 22 d0 02 e6 23 b1 22 ae
 96b8 : d0 f4 ca d0 f3 a5 22 69 97
 96c0 : 01 85 2d a5 23 69 00 85 0c
 96c8 : 2e 20 f6 a5 4c f2 a4 20 a8
 96d0 : 3c a6 ca 8a 29 0f 8d 20 e5

96d8 : d0 20 43 a6 ca 8a 29 0f 22
 96e0 : 8d 21 d0 60 20 3c a6 ca 52
 96e8 : 8a 29 0f 8d 86 02 60 20 b7
 96f0 : 3c a6 e0 80 90 02 a2 27 ae
 96f8 : 86 02 20 43 a6 e0 19 90 e7
 9700 : 02 a2 18 a4 02 20 0c e5 0b
 9708 : 20 79 00 c9 2c d0 df 20 27
 9710 : 73 00 a2 a0 a0 aa 4c 81 d4
 9718 : 9f 20 12 b1 84 fc 85 fd c4
 9720 : 20 18 a6 20 12 b1 98 a0 4c
 9728 : 00 91 fc c8 a5 15 91 fc 8c
 9730 : 60 20 b5 ac a2 f7 4c 3e 3b
 9738 : a6 a5 14 48 a5 15 48 20 23
 9740 : 02 ab 20 12 b1 20 09 ab fa
 9748 : a0 00 b1 14 aa c8 b1 14 b7
 9750 : a8 68 85 15 68 85 14 98 65
 9758 : 4c c0 a5 a5 14 a8 a5 15 67
 9760 : 48 20 02 ab 20 12 b1 20 48
 9768 : 09 ab a9 31 78 85 01 a0 d8
 9770 : 00 b1 14 aa a9 36 85 01 08
 9778 : 58 4c 32 b1 4c 62 a2 20 5c
 9780 : 60 20 20 4a ad 20 09 ab 01
 9788 : a5 7a 48 a5 7b 48 a9 3f 50
 9790 : 8d 00 02 a4 7b f0 e5 c0 4f
 9798 : 51 90 05 a2 d0 4c 67 ad 2f
 97a0 : a2 a6 a0 b3 20 81 9f a9 14
 97a8 : 02 85 7b a4 b7 a9 00 85 b4
 97b0 : 7a 99 01 02 88 b1 bb 99 b0
 97b8 : 01 02 88 10 f8 20 1d a1 27
 97c0 : e6 7b a9 01 85 7a 20 b5 06
 97c8 : ac 68 85 7b 68 85 7a 60 d6
 97d0 : a2 aa 8e f6 07 a2 b6 8e c7
 97d8 : f7 07 4c 87 9f a2 0b 86 9f
 97e0 : fb a2 00 86 fc a6 fb bd 6e
 97e8 : 5e b2 a8 ca bd 5e b2 aa 12
 97f0 : b9 00 00 38 f5 00 85 63 ed
 97f8 : b9 01 00 f5 01 85 62 a4 00
 9800 : fc b9 16 b2 08 e6 fc 29 b3
 9808 : 7f 20 d2 ff 20 10 f0 a2 58
 9810 : d1 a0 bd 20 81 9f 98 38 8d
 9818 : e9 07 a8 b9 58 b2 d2 ff 32
 9820 : c8 d0 f8 a9 20 20 20 d2 07
 9828 : ff c8 c0 06 d0 f5 c6 fb 4c
 9830 : c6 f6 10 b1 60 4d 45 4d 4e
 9838 : 4f 52 59 3a 20 20 20 a0 13
 9840 : 50 52 4f 47 52 41 4d 3a 4f
 9848 : 20 20 a0 56 41 52 49 41 ba
 9850 : 42 4c 45 53 3a a0 41 52 c6
 9858 : 52 41 59 53 3a 20 20 20 71
 9860 : a0 53 54 52 49 4e 47 53 d4
 9868 : 3a 20 20 a0 46 52 45 a5 65
 9870 : 3a 20 20 a0 20 20 a0 42 d1
 9878 : 59 54 45 53 0d 31 33 33 45
 9880 : 3f 2f 31 2d 2f 2b 2d 2b 98
 9888 : 37 a9 00 85 0a 20 6e a9 f3
 9890 : a5 0a a6 2d a4 2e 20 d5 71
 9898 : ff b0 0d a5 90 29 bf f0 b1
 98a0 : 0e a2 9c a0 e1 4c 90 9f 3c
 98a8 : a2 f9 a0 e0 4c 90 9f 20 93
 98b0 : ec a5 a0 01 b1 2d f0 5a b4
 98b8 : c8 b1 2d 85 14 c8 b1 2d fd
 98c0 : 85 15 20 a9 a2 a0 03 b1 ac
 98c8 : 2d 99 00 bf 88 10 f8 a0 e8
 98d0 : 03 c8 b1 2d 99 00 bf d0 8d
 98d8 : f8 18 c8 84 0b a5 2d 85 34
 98e0 : 5a 65 0b 85 58 a4 2e 84 cd
 98e8 : 5b 90 01 c8 84 59 a2 b8 fd
 98f0 : 8e f6 07 a2 a3 8e f7 07 ac
 98f8 : 20 87 9f a5 31 a4 32 85 84
 9900 : 2d 84 2e a4 00 88 b9 00 6b
 9908 : bf 91 5f 88 c0 ff d0 f6 b6
 9910 : f0 9d 20 f3 a5 4c f2 a4 77
 9918 : a2 a6 a0 b3 20 81 9f 20 79
 9920 : 4a ad a4 b7 c0 51 90 03 40
 9928 : 4c 7c b1 88 30 43 a9 00 f3
 9930 : 99 01 02 85 7a b1 bb 99 d2
 9938 : 00 02 88 10 f8 a9 02 85 4d
 9940 : 7b 20 79 00 f0 2b b0 29 a7
 9948 : 20 a2 a2 20 59 b3 a2 9f 63
 9950 : a0 a4 20 81 9f a2 03 86 a3
 9958 : 02 20 f6 a5 a6 6d a4 6e 22
 9960 : e8 d0 01 c8 c6 02 f0 06 56
 9968 : 20 81 9f 4c 3a b3 20 78 6d
 9970 : b3 a6 39 a4 3a 4c df a4 28
 9978 : a9 c1 8d 18 03 a9 b3 8d d0
 9980 : 00 03 a9 e6 8d 01 03 a9 89
 9988 : 70 8d 2c 03 a9 ae 8d 2d cb
 9990 : 03 a9 34 8d 02 03 60 a9 34
 9998 : 83 8d 02 03 a9 2f 8d 2c 65
 99a0 : 03 a9 f3 8d 2d 03 a9 8b cf
 99a8 : 8d 00 03 a9 e3 8d 01 03 e0
 99b0 : a9 47 8d 18 03 60 20 3c 8f
 99b8 : a6 20 a0 b3 4c cb ac a9 36
 99c0 : 24 85 bf a9 f7 85 bb a9 09
 99c8 : 00 85 bc a9 01 85 b7 86 17
 99d0 : ba a9 60 85 b9 20 d5 f3 03
 99d8 : a5 ba 20 b4 ff a5 b9 20 cd
 99e0 : 96 ff a9 00 85 90 a0 03 46

```

99e0 : 84 fb 20 a5 ff 85 fc a4 90
99f0 : 90 d0 30 20 a5 ff a4 90 06
99f8 : d0 29 a4 fb 88 d0 e9 a6 09
9a00 : fc 20 07 b4 a9 20 20 d2 26
9a08 : ff 20 a5 ff a6 90 d0 13 d9
9a10 : aa d0 f3 a9 0d 20 d2 ff 71
9a18 : ad 8d 02 4a b0 fa 4a a0 a3
9a20 : 02 90 c5 4c 28 f5 a0 cd b6
9a28 : 8c f6 07 a0 bd 4c 84 9f 95
9a30 : 20 3c a6 20 1a b4 4c cb 2c
9a38 : ac 20 0f f3 f0 03 4c 01 91
9a40 : f7 20 1f f3 29 9f 8d 01 55
9a48 : bf 20 18 a6 20 12 b1 8c 65
9a50 : 02 bf 8d 03 bf 20 79 00 d8
9a58 : c9 2c d0 07 20 73 00 20 2a
9a60 : 3c a6 2c a2 01 8e 04 bf 63
9a68 : a9 50 8d 00 bf a2 00 a0 ef
9a70 : bf a9 05 4c 79 ae 20 02 60
9a78 : ab 20 b5 ac a5 66 48 a2 2a
9a80 : 1b a0 bc 20 81 9f a2 04 c6
9a88 : b5 61 48 ca 10 fa 20 43 39
9a90 : a6 8e f0 bf 8e f1 bf a2 6e
9a98 : 00 68 95 61 e8 e0 05 d0 a9
9aa0 : f8 bf 07 a2 e2 a0 ba 20 85
9aa8 : 81 9f ce f0 bf 10 f4 a2 60
9ab0 : 49 a0 b8 20 81 9f a2 cc b5
9ab8 : a0 bc 20 81 9f a2 a8 b4 57
9ac0 : a2 fe a0 ba 20 81 9f ce 8b
9ac8 : f1 bf 10 f4 68 85 66 4c 20
9ad0 : 09 ab 20 02 ab 20 12 b1 5f
9ad8 : 20 09 ab a9 02 a6 15 f0 28
9ae0 : 01 0a 20 6c ad a0 00 a5 a7
9ae8 : 15 f0 03 20 d7 b4 a5 14 1c
9af0 : 20 d7 b4 4c 65 ad 48 4a 2c
9af8 : 4a 4a 4a 20 e2 b4 68 29 c6
9b00 : 0f 18 69 f6 90 02 69 06 1f
9b08 : 69 3a 91 62 c8 60 4c 0b 16
9b10 : a3 20 02 ab 20 81 b5 20 de
9b18 : 88 b5 a5 64 48 a5 65 48 49
9b20 : 20 18 a6 20 81 b5 20 7a 35
9b28 : b5 f0 e3 85 9c 86 c1 84 0d
9b30 : c2 68 a8 68 20 b1 b1 f0 96
9b38 : d5 85 9b 86 b5 84 b6 a2 27
9b40 : 00 20 79 00 c9 2c d0 0a 04
9b48 : 20 73 00 20 3c a6 8a f0 2b
9b50 : bd ca 86 a5 20 09 ab a5 0d
9b58 : 9b 38 e5 9c 90 2e e5 a5 7a
9b60 : 90 2a 69 00 85 a7 a5 a5 d7
9b68 : 18 65 b5 85 b5 90 02 e6 07
9b70 : b6 a0 00 b1 b5 d1 c1 d0 3f
9b78 : 0d c8 c4 9c 90 f5 a6 a5 4d
9b80 : e8 a9 00 4c c0 a5 e6 a5 e7
9b88 : c6 a7 d0 04 a2 00 f0 f1 a8
9b90 : e6 b5 d0 dd e6 b6 4c 52 3b
9b98 : b5 a2 a3 a0 b6 4c 81 9f ae
9ba0 : a2 9e a0 ad 4c 81 9f a2 04

```

```

9ba8 : 8f d0 f7 20 02 ab 20 4a 34
9bb0 : ad 20 09 ab a0 00 8c f2 47
9bb8 : bf a5 b7 d0 03 4c 00 a3 58
9bc0 : 4a b0 fa c9 03 b0 f6 4a 80
9bc8 : b0 06 20 bb b5 8d f2 bf 0e
9bd0 : 20 bb b5 aa ad f2 bf 4c 9a
9bd8 : c0 a5 b1 bb 20 d2 b5 0a d2
9be0 : 0a 0a 0a 8d f3 bf c8 b1 e7
9be8 : bb 20 d2 b5 0d f3 bf c8 20
9bf0 : 60 c9 3a 08 29 0f 28 90 91
9bf8 : 02 b9 08 60 a6 30 a5 2f 9e
9c00 : 85 8b 86 8c e4 32 d0 02 a5
9c08 : c5 31 90 01 60 a0 04 69 98
9c10 : 05 90 01 e8 85 8d 86 8e b7
9c18 : b1 8b 0a a8 65 8d 90 01 2d
9c20 : e8 85 c1 86 c2 88 84 8f ae
9c28 : a9 00 99 05 02 88 10 fa 73
9c30 : 30 32 a4 8f 88 84 fd 98 6a
9c38 : aa fe 06 02 d0 03 fe 05 4e
9c40 : 02 b9 05 02 d1 8d 00 06 79
9c48 : c8 b9 05 02 d1 8d 90 14 62
9c50 : a9 00 a4 fd 99 05 02 99 df
9c58 : 06 02 88 10 d7 a5 c1 a6 82
9c60 : c2 4c e1 b5 20 c2 b6 a4 b4
9c68 : 8f a9 28 20 d2 ff b9 04 f6
9c70 : 02 be 05 02 84 fd 20 07 1a
9c78 : b4 a9 2c a4 fd 88 88 10 07
9c80 : ea a9 29 20 6a b7 a5 c1 0c
9c88 : a6 c2 85 22 86 23 a0 00 39
9c90 : b1 8b 10 07 20 71 b7 a9 ac
9c98 : 02 d0 11 c8 b1 8b 30 07 a6
9ca0 : 20 8b b7 a9 05 d0 05 20 d4
9ca8 : a1 b7 a9 03 18 65 c1 85 ae
9cb0 : c1 90 02 e6 c2 a2 d7 a0 f9
9cb8 : aa 20 81 9f 4c 13 b6 ad 5a
9cc0 : 8d 02 29 01 f0 14 2d 8d 38
9cc8 : 02 d0 fb 20 bb b6 ad 8d 78
9cd0 : 02 29 02 f0 f6 2d 8d 02 19
9cd8 : d0 fb a2 2c a0 a8 4c 81 57
9ce0 : 9f 20 a0 b6 a0 d2 00 b1 8b 76
9ce8 : aa 29 7f 20 d0 ff c8 b1 be
9cf0 : 8b a8 29 7f f0 03 20 d2 57
9cf8 : ff 8a 10 05 a9 25 4c d2 7c
9d00 : ff 98 10 05 a9 24 20 d2 d2
9d08 : ff 60 a5 2d a4 2e 85 14 e0
9d10 : 84 15 c4 30 d0 02 85 2f ae
9d18 : b0 18 69 02 90 01 c8 85 ae
9d20 : 22 84 23 20 31 b7 20 73 73
9d28 : b7 8a 10 07 20 71 b7 4c 0e
9d30 : 20 b7 60 98 30 06 20 8b 22
9d38 : b7 4c 20 b7 20 a1 b7 a9 56
9d40 : 0d 20 d2 ff a5 14 a4 15 ca
9d48 : 18 69 07 90 c1 c8 b0 be 8b
9d50 : a0 00 b1 1a aa 29 f7 20 11
9d58 : d2 ff c8 b1 14 a8 29 7f bc
9d60 : f0 03 20 d2 ff 8a 10 17 f7

```

```

9d68 : 98 30 10 a2 05 bd c5 b7 35
9d70 : 20 d2 ff ca 10 f7 68 68 86
9d78 : 4c 20 b7 a9 25 d0 62 98 8b
9d80 : 10 04 a9 24 d0 5b 60 a9 3e
9d88 : 20 20 d2 ff a9 3d d0 51 d7
9d90 : a0 00 b1 22 aa c8 b1 22 dd
9d98 : a8 a9 91 8d f6 07 a9 b3 e1
9da0 : 8d f7 07 8a 20 87 9f 4c 91
9da8 : 92 b7 a2 a6 a0 bb 20 81 ff
9db0 : 9f a2 d7 8e f6 07 a2 bd 16
9db8 : 8e f7 07 a0 01 4c 87 9f e8
9dc0 : 20 c0 b7 a0 02 b1 22 85 83
9dc8 : 25 88 b1 22 85 24 88 b1 e1
9dd0 : 22 85 26 f0 0a b1 24 20 5b
9dd8 : d2 ff c8 c4 26 d0 f6 a9 8d
9de0 : 22 4c d2 ff 4e 46 46 45 98
9de8 : 44 5f 20 47 a6 ca 86 24 f0
9df0 : a6 15 20 dd a6 b0 4f a5 5d
9df8 : 15 a6 24 0a 0a 0a 0a d8
9e00 : 85 23 a5 14 48 4a 4a 4a 97
9e08 : 18 65 23 65 03 48 a8 c5 f6
9e10 : d3 f0 17 c9 27 90 02 a0 9b
9e18 : 27 a9 a0 85 22 20 24 ea 76
9e20 : a5 22 20 1e ea 88 c4 d3 50
9e28 : 10 ef 68 a8 68 c0 28 b0 ee
9e30 : 10 29 07 aa bd 2a b8 a6 49
9e38 : 24 48 20 24 ea 68 20 1e bc
9e40 : ea a9 11 4c 16 e7 4c 0b b5
9e48 : a3 20 65 74 75 61 f6 ea f7
9e50 : e7 20 47 a6 a5 15 d0 ee 12
9e58 : ca 86 24 a5 14 c9 c8 b0 37
9e60 : e5 4a 4a 4a 85 23 a5 14 76
9e68 : 29 07 85 25 a5 d1 18 65 2f
9e70 : d3 85 fd a5 d2 69 00 85 bd
9e78 : fe a0 00 a6 23 f0 29 a5 45
9e80 : fd 85 f3 a5 fe 29 03 09 49
9e88 : d8 85 f4 a9 a0 91 fd a5 6f
9e90 : 24 91 f3 a5 fd 38 e9 28 c8
9e98 : 85 fd b0 08 c6 fe a5 fe 42
9ea0 : c9 04 90 1b c6 23 d0 d7 6b
9ea8 : a5 fd 85 f3 a5 fe 29 03 29
9eb0 : 09 d8 85 f4 a6 25 bd ac 09
9eb8 : b8 91 fd a5 24 91 f3 a5 57
9ec0 : d3 c5 fd a5 05 a9 1d 20 5c
9ec8 : 16 e7 60 20 64 6f 79 62 5a
9ed0 : f8 f7 e3 a5 ba 20 b4 ff f1
9ed8 : a9 62 85 b9 20 96 ff 20 42
9ee0 : a5 ff 48 20 ab ff 68 60 b8
9ee8 : 48 a5 ba 20 b1 ff a9 62 3c
9ef0 : 85 b9 20 93 ff 68 20 a8 e1
9ef8 : ff 4c 54 f6 78 a9 37 85 ce
9f00 : 01 a9 1b a0 85 85 8f 84 bc
9f08 : 60 4c 00 85 00 ff 00 ff 3f

```

Listing »Basic-Erweiterungen« (Schluß)

Disk-Basic — programmieren Sie Ihre eigene Daten- verwaltung!

Disk-Basic ergänzt das magere Basic V 2.0 um einige Befehle zum komfortablen Programmieren von sequentiellen und relativen Dateien.

Disk-Basic ist entstanden, um beispielsweise eine leistungsfähige Dateiverwaltung mit einfachen Befehlen zu programmieren. Das ist natürlich nur eines der denkbaren Anwendungsgebiete von Disk-Basic, denn sein umfangreicher Befehlsvorrat erleichtert generell das Arbeiten mit dem 1541-Floppy-Laufwerk. Disk-Basic hat das Ziel, nicht mehr mit spektakulären Kommandokombinationen die Laufwerke zum Arbeiten zu animieren, sondern mit Klartextbefehlen. Damit trägt es dem Prinzip einer Hochsprache wie Basic, komfortabel Daten zu verarbeiten, Rechnung.

Nach dem fehlerfreien Eingeben (verwenden Sie bitte den MSE) und Abspeichern des Programms (siehe Listing) kann Disk-Basic mit SYS 49152 gestartet werden. Die Befehlssyntax aller neuen Befehle richtet sich nach der altbewährten Methode: Sonderzeichenerkennung — Befehlsidentifizierung — Ausführung. Bevor nun die Wirkung der einzelnen Befehle erläutert wird, noch einige andere allgemeine Hinweise.

— Da, zumindest in Deutschland, das englische Pfund-Zeichen relativ wenig verwendet wird, wurde es als Einleitungszeichen für die neuen Befehle gewählt.

— Beim Start oder bei der Initialisierung werden zwei Kommandokanäle geöffnet:

OPEN 15,8,15 und OPEN 16,9,15

Es ist dabei ohne Belang, ob die Floppy-Laufwerke einge-

Basic-Erweiterungen

schaltet sind oder nicht. Da der Befehl RUN alle Kanäle schließt, muß am Programmanfang immer mit INIT initialisiert werden. Der Grund für diese Forderung ist, daß während des Arbeitens mit relativen Dateien an den Zuständen der Kanäle nichts verändert werden darf.

— Um auch bei den Sekundäradressen Ordnung zu schaffen, wird beim Öffnen einer Datei die Sekundäradresse gleich der Kanalnummer gesetzt (zum Beispiel OPEN 2,8,2). Aus diesem Grund darf die Kanalnummer nur zwischen 2 und 14 liegen.

— Das zweite Floppy-Laufwerk (nicht unbedingt notwendig) sollte die Geräteadresse 9 haben. Ist es bereits fest auf eine höhere Nummer eingestellt, muß das Programm geringfügig verändert werden. Ansonsten steht zur Veränderung der Geräteadresse ein gesonderter Befehl zur Verfügung.

Die 38 Befehle des Disk-Basic unterteilen sich in zwei Gruppen:

Die eigentlichen DOS-Befehle und allerlei Hilfsbefehle, zum Beispiel zur Aufbereitung von Datensätzen.

Folgendes zur Erklärung:

Steht »dr« in Klammern, so bedeutet das, daß die Geräte-Nummer weggelassen werden kann. Dabei wird angenommen, daß es sich um Gerät 8, das Floppy-Laufwerk handelt.

„...“ bedeutet, daß weitere Parameterkombinationen angehängt werden können.

Befehle für die Laufwerksbedienung

£CATALOG („Parameter“) (,dr)

Ausgabe des Directorys. Als Parameter sind Standardjoker (siehe 1541-Handbuch Seite 11) einsetzbar.

£COLLECT (,dr)

»Aufräumen« führt eine Disk-Garbage-Collection durch (entspricht dem Validate).

£COPY,„neufile=altfile,...“ (,dr)

Kopieren von Files

£CLALL

schließt alle Kanäle und öffnet die Kommandokanäle sofort wieder

£OPEN,fn,„filename,x,y“

öffnet Kanal fn.

x = s sequential

p program

u user

y = w write

r read

a append

m Lesen einer nicht geschlossenen Datei

fn muß zwischen 2 und 14 liegen

£ERROR,dr,ds (,ds\$) (,tr) (,se)

dr muß angegeben werden. Fehlerkanal wird gelesen

£FORMAT,„diskname,id“ (,dr)

formatieren einer Disk

£RENAME,„neuname=altname“ (,dr)

umbenennen von Dateien

£SCRATCH,„filename,...“ (,dr)

löschen von Dateien

£CHANGE,neue nr, alte nr

Die Gerätenummer des »alten« Floppy-Laufwerks wird umgeändert.

£INIT (,dr)

Das betreffende Laufwerk wird initialisiert

£CHAIN,„Programmname“ (,dr)

Aneinanderhängen von Programmen mit Variablenschutz; darf nur bei fertig entwickelten Programmen verwendet werden, da das Programm im Speicher nach Ende oder Abbruch nicht mehr ordnungsgemäß zu listen und zu starten ist.

£SYSTEM,„Programmname“ (,dr)

die Erweiterung wird gespeichert

£FETCH,fn,az,er\$,...

vom File fn werden az Zeichen in er\$ eingelesen

£TRANS, Sendedatei-Nr., Empfangsdatei-Nr.

beide Dateien müssen geöffnet sein. Die Daten können zwischen allen Peripheriegeräten ausgetauscht werden.

£RANDOM,fn,„Name“,le,az (,dr)

fn = Kanalnummer

le = Länge eines Records

az = die Anzahl der freigemachten Records (record az wird mit CHR\$(255) beschrieben)

£CLEAR

löscht den Datenpuffer für relative Dateien

£PUT,fn,rn,az (,dr)

schreibt az Byte des Puffers in den Record rn

£GET,fn,rn,az (,dr)

liest az Byte des Records rn in den Puffer

£WRITE,pos,x\$,az,...

schreibt az Zeichen des Strings x\$ ab pos (Position) in den Puffer.

Ist der String zu kurz, wird mit CHR\$(0) aufgefüllt.

£READ,Pos,az,x\$... liest ab Pos (Position) az: Zeichen des Puffers in x\$

Andere Befehle:

£COMP,a,a\$,...

wandelt die Zahl a (integer oder real) in den äquivalenten String. Real 5 Zeichen, Integer 2 Zeichen.

£RECOMP,a\$,a,...

Umkehrung von COMP

£BEEP

Ausgabe eines Signaltones

£CSET,x,y

Cursorpositionierung

x = Zeile, y = Spalte

£CREAD,x,y

Cursorabfrage

gibt aktuelle Cursorposition aus (x = Zeile, y = Spalte)

£RESTORE, Ausdruck

Positioniert den Lesezeiger für Daten auf Zeile »Ausdruck«

£INSTR\$a\$,b\$,Pos,erg,...

durchsucht a\$ ab Pos nach b\$. Ist erg = 0, dann nicht vorhanden

£STORE,adresse,„XXYYXXYYXXYY...“ ...

speichert Hexbytes ab Adresse (ideal für Sprites)

£STR\$,„XXYYXXYYXXYY“,a\$...

wandelt Hexbytes in ASCII-Zeichen und speichert sie in a\$

£STRING,a Zeichen,er\$...

erzeugt einen a Zeichen langen String aus

£COLOUR,f1,f2

verändert die Bildschirmfarben

£GOTO,Ausdruck

GOTO wird »berechenbar«

£HEX,z,h\$...

wandelt z in h\$ (hexadezimalzahl) um

£DEZ,h\$,z...

wandelt h\$ (hexadezimal) in z (dezimal) um

£LINPUT,a\$

Übernimmt eine ganze Zeile mit allen Zeichen

£SYS,adresse,akku,xreg,yreg...

SYS mit Parameterübergabe der 6510-Register

akku = Akkumulator

xveg = X-Register

yveg = y-Register

£QUIT

verlassen des Interpreters

(Harald Großbauer/aw)

programm : disk-basic c000 c8c4

```

c000 : 20 3e c0 a2 00 bd 11 c0 77
c008 : f0 06 20 d2 ff e8 d0 f5 d4
c010 : 60 0d 90 12 20 44 49 53 4d
c018 : 4b 2d 42 41 53 49 43 20 7f
c020 : 20 28 43 29 20 42 59 20 04
c028 : 47 52 4f 53 53 41 55 45 f6
c030 : 52 20 48 41 52 52 59 20 2a
c038 : 20 38 35 20 0d 00 a9 6e 1a
c040 : a0 c0 8d 08 03 8c 09 03 63
c048 : a9 00 48 20 bd ff a9 0f a8
c050 : a2 08 a8 20 ba ff 20 c0 d2
c058 : ff a9 ff 8d 8a 02 68 20 78
c060 : bd ff a9 10 a2 09 a0 0f 9c
c068 : 20 ba ff 4c c0 ff 20 73 e2
c070 : 00 c9 5c f0 03 4c e7 a7 0b
c078 : a0 00 84 a7 20 73 00 f0 ae
c080 : 0e c9 3a f0 0a c9 2c f0 a1
c088 : 06 99 c8 c9 c8 d0 ed a0 d2
c090 : 00 b9 c5 c0 f0 06 c8 d0 fa
c098 : f8 4c 08 af a2 00 e6 a7 c3
c0a0 : a9 28 c5 a7 f0 f3 c8 b9 09
c0a8 : c5 c0 f0 09 dd c8 c9 d0 17
c0b0 : e0 c8 e8 d0 f2 a6 a7 ca e1
c0b8 : bd 85 c1 85 ab bd ab c1 03
c0c0 : 85 a9 6c ab 00 00 43 41 da
c0c8 : 54 41 bc 00 43 4f 4c 4c 64
c0d0 : 45 43 54 00 54 52 41 4e 45
c0d8 : 53 00 43 4f 50 59 00 43 3c
c0e0 : 4c 41 4c 4c 00 9f 00 45 f1
c0e8 : 52 52 b0 00 52 af 4f 4d 0a
c0f0 : 00 81 4d 41 54 00 43 4c 17
c0f8 : 45 41 52 00 52 45 4e 41 7e
c100 : 4d 45 00 53 43 52 41 54 cf
c108 : 43 48 00 50 55 54 00 a1 b5
c110 : 00 43 48 41 4e 47 45 00 20
c118 : 57 52 49 54 45 00 87 00 e8
c120 : 49 4e 49 54 00 43 4f 4d 5f
c128 : 50 00 52 45 43 4f 4d 50 3a
c130 : 00 43 48 41 49 4e 00 9e 50
c138 : 54 45 4d 00 4e 45 54 43 e9
c140 : 48 00 42 45 45 50 00 43 1f
c148 : 53 45 54 00 8c 00 49 4e de
c150 : c4 00 53 a4 52 45 00 48 5e
c158 : 45 58 00 44 45 5a 00 4c 12
c160 : 85 00 89 00 9e 00 c4 00 45
c168 : 43 4f 4c 4f 55 52 00 53 de
c170 : 54 52 49 4e 47 00 51 55 6e
c178 : 49 54 00 43 87 00 00 0c cc
c180 : 00 00 00 00 00 22 97 e8 c2
c188 : e0 08 54 11 78 b1 de a4 fe
c190 : d9 ea 00 24 16 5d a7 8c 67
c198 : a3 e8 c6 92 c5 ea f8 8b 6e
c1a0 : ed 63 1b 16 3a 9d 35 8e 4b
c1a8 : 56 49 76 c3 c3 c3 c3 c4 ac
c1b0 : c4 c4 c3 c4 c3 c3 c4 83
c1b8 : c5 c4 c5 c5 c3 c5 c5 c5 16
c1c0 : c5 c6 c6 c6 c6 c7 c7 c8 ce
c1c8 : c8 c7 c7 c8 c8 c8 c7 c7 00
c1d0 : c7 20 df c1 98 a6 22 a4 68
c1d8 : 23 20 bd ff 4c e8 c1 20 ce
c1e0 : fd ae 20 9e ad 4c 82 b7 c7
c1e8 : a9 08 85 ba 20 79 00 c9 af
c1f0 : 2c d0 05 20 78 c2 86 ba f7
c1f8 : 60 20 cc ff 20 79 00 c9 fd
c200 : 3a d0 03 4c 07 ab 4c be 51
c208 : a7 20 fd ae 20 8a ad 20 62
c210 : f7 b7 a5 14 85 fb a5 15 c8
c218 : 85 fc 60 48 a6 ba e8 e8 f2
c220 : e8 e8 e8 e8 e8 86 ab 20 85
c228 : c9 ff 68 f0 0f 85 fd a0 7f
c230 : 00 b9 c8 c9 20 d2 ff c8 a2
c238 : c4 fd d0 f5 a9 0d 20 d2 17
c240 : ff 4c cc ff a9 50 8d c8 7d
c248 : c9 a5 a7 8d c9 c9 a5 fb f9
c250 : 8d ca c9 a5 fc 8d cb c9 68
c258 : a9 01 8d cc c9 a9 05 c4 15
c260 : 1b c2 20 78 c2 86 a7 20 33
c268 : 09 c2 20 78 c2 86 aa 20 35
c270 : e8 c1 20 44 c2 a6 a7 60 8a
c278 : 20 fd ae 4c 9e b7 20 fd f0
c280 : ae 20 8b b0 85 a9 84 4a 80
c288 : 20 a3 b6 a5 ab 20 75 b4 d7
c290 : a0 02 b9 61 00 91 49 88 8f
c298 : 10 f8 c8 60 20 fd ae 20 4f
c2a0 : 8b b0 85 ab 85 a9 84 a9 02
c2a8 : 84 4a a2 05 a5 0e c9 80 8e
c2b0 : d0 02 a2 02 86 ab 60 48 42
c2b8 : 98 48 20 9c c2 68 ab 68 f3
c2c0 : a2 00 86 d0 85 62 84 63 ea
c2c8 : a2 90 38 20 49 bc a5 ab 2d
c2d0 : c9 05 f0 03 4c c4 a9 4c e2

c2d8 : d0 bb b1 22 20 ef c2 0a d7
c2e0 : 0a 0a 0a 18 85 ab c8 b1 b1
c2e8 : 22 20 0f c2 05 ab 60 c9 31
c2f0 : 3a 08 29 0f 28 90 02 69 3c
c2f8 : 08 60 85 ab 4a 4a 4a 4a bc
c300 : 20 0b c3 c8 a5 ab 29 0f 2a
c308 : 4c 0b c3 18 69 f6 90 02 62
c310 : 69 06 69 3a 99 c8 c9 60 e6
c318 : 20 79 00 38 c9 2c f0 01 c0
c320 : 18 60 a9 08 85 ba 20 79 75
c328 : 00 c9 2c d0 06 20 d1 c1 5e
c330 : 4c 40 c3 a9 24 85 a7 a9 23
c338 : 01 a2 a7 a0 20 bd ff 80
c340 : a9 60 85 b9 20 d5 f3 a5 7e
c348 : ba 20 b4 ff a5 b9 20 96 15
c350 : ff a9 00 85 90 a0 03 84 f8
c358 : a8 20 a5 ff 85 a9 a4 90 d3
c360 : d0 2f 20 a5 ff a4 90 d0 8e
c368 : 28 a4 a8 88 d0 e9 a6 a9 68
c370 : 20 cd bd a9 20 20 d2 ff 6a
c378 : 20 a5 ff a6 90 d0 12 aa 6d
c380 : f0 06 20 d2 ff 4c 78 c3 a1
c388 : a9 0d 20 d2 ff a0 02 d0 c9
c390 : c6 20 42 f6 4c f9 c1 a9 c5
c398 : 56 8d c8 c9 20 e8 c1 a9 c4
c3a0 : 01 20 1b c2 4c f9 c1 20 ac
c3a8 : e7 ff 20 3e c0 a9 49 d0 7f
c3b0 : e8 a9 4e 8d c8 c9 a9 3a a8
c3b8 : 8d c9 c9 20 df c1 84 a7 0e
c3c0 : a0 00 b1 22 99 ca c9 c4 b2
c3c8 : a7 f0 03 c8 d0 f4 c8 c8 2b
c3d0 : 98 48 20 e8 c1 68 4c a1 85
c3d8 : c3 a9 53 8d c8 c9 d0 d6 c2
c3e0 : a9 43 d0 f7 a9 52 d0 f3 b6
c3e8 : 20 78 c2 20 c6 ff 20 78 d7
c3f0 : c2 20 c9 ff 20 cf ff 48 46
c3f8 : 20 b7 ff d0 07 68 20 d2 e8
c400 : ff 4c f4 c3 68 4c f9 c1 2f
c408 : 20 e7 ff 20 3e c0 4c f9 2f
c410 : c1 20 e8 c1 a9 00 20 1b a5
c418 : c2 20 fd ae a6 ab 20 ad e3
c420 : ab 4c f9 c1 20 78 c2 86 86
c428 : a7 20 e8 c1 a2 00 bd 4e 0f
c430 : c4 9d c8 c9 e8 e0 06 d0 7d
c438 : f5 18 a5 a7 69 20 8d ce e3
c440 : c9 18 a5 a7 69 20 8d ce e3
c448 : c9 a9 08 4c a1 c3 4d 2d 39
c450 : 57 77 00 02 20 78 c2 86 81
c458 : a7 20 df c1 84 aa 20 e8 2f
c460 : c1 a5 a7 ab a6 ba 20 ba 29
c468 : ff a5 aa a6 22 a4 23 20 cd
c470 : bd ff 20 c0 ff 4c f9 c1 1b
c478 : 20 78 c2 86 a7 20 df c1 d4
c480 : 84 aa a5 22 a6 23 85 ab f2
c488 : 86 a9 20 78 c2 86 ab 20 49
c490 : 09 c2 20 e8 c1 a5 a7 ab 59
c498 : a6 ba 20 ba ff a0 00 b1 63
c4a0 : a8 99 c8 c9 c4 aa f0 03 eb
c4a8 : c8 d0 f4 a9 2c 99 c8 c9 91
c4b0 : 99 ca c9 c8 a7 4c 99 c8 2f
c4b8 : c9 c8 c8 e6 ab a5 ab 99 be
c4c0 : c8 c9 c8 98 a2 c8 a0 c9 38
c4c8 : 20 bd ff 20 c0 ff 20 44 e0
c4d0 : c2 a6 a7 20 c9 ff a9 ff 16
c4d8 : 20 d2 ff 4c f9 c1 a0 00 1b
c4e0 : 98 99 c4 c8 c8 d0 fa 4c 27
c4e8 : f9 c1 20 62 c2 20 c9 ff 6b
c4f0 : a0 00 a9 c4 c8 20 d2 ff 70
c4f8 : c8 c4 ba d0 f5 4c f9 c1 14
c500 : 20 62 c2 20 c6 ff a0 00 f5
c508 : 20 cf ff 99 c4 c8 c8 c4 82
c510 : aa d0 f5 4c f9 c1 20 78 48
c518 : c2 86 a7 20 df c1 84 aa 7f
c520 : a5 22 a6 23 85 a8 86 a9 ef
c528 : 20 78 c2 86 ab e0 00 f0 a9
c530 : 24 a4 aa f0 20 a6 a7 a0 86
c538 : 00 b1 a8 9d c4 c8 e8 f0 07
c540 : 14 c8 c4 ab f0 0f c4 aa 4f
c548 : d0 ef a9 20 9d c4 c8 e8 73
c550 : c8 c4 ab d0 f7 20 18 c3 e8
c558 : b0 bc ac f9 c1 20 78 c2 3d
c560 : 86 a7 20 78 c2 86 ab 20 20
c568 : 7e c2 a6 a7 bd c4 c8 91 2e
c570 : 62 e8 f0 07 c8 c4 61 f0 7d
c578 : 0b d0 f1 a9 20 91 62 c8 47
c580 : c4 61 d0 f9 20 18 c3 b0 9b
c588 : d4 4c f9 c1 20 c9 c2 20 6b
c590 : 7e c2 b1 a8 91 62 c8 ca ca
c598 : ab d0 f7 20 18 c3 b0 ec e9
c5a0 : 4c f9 c1 20 df c1 84 aa d1
c5a8 : a6 22 a5 23 86 fb 85 fc 85
c5b0 : 20 9c c2 a0 00 b1 fb 91 84
c5b8 : a8 c8 c4 ab d0 f7 20 18 e8
c5c0 : c3 b0 e0 4c f9 c1 20 e7 9b

c5c8 : ff 20 d1 c1 a6 ba a9 01 6d
c5d0 : a8 20 ba ff a9 00 a2 c0 dd
c5d8 : 85 ab 86 a9 a2 c4 a0 c8 ec
c5e0 : a9 a8 20 d8 ff 4c f9 c1 ce
c5e8 : 20 26 b5 20 d1 c1 a9 01 60
c5f0 : a6 ba a8 20 ba ff a0 04 58
c5f8 : b6 30 96 68 88 d0 f9 38 e0
c600 : a5 69 e9 01 85 69 a5 6a 03
c608 : e9 00 85 6a 38 a5 6b e9 d2
c610 : 01 85 6b a5 6c e9 00 85 84
c618 : c6 b1 69 91 6b a5 2d c5 0d
c620 : 69 90 dc a5 2e c5 6a 90 99
c628 : d6 a9 00 20 d5 ff 38 a5 60
c630 : ae e5 2d 85 6d a5 af e5 5b
c638 : 2e 85 6e a5 ae 85 2d 85 50
c640 : 69 a5 af 85 2e 85 6a 38 42
c648 : a5 ae e5 6b a5 af e5 6c 73
c650 : 90 03 4c 35 a4 a5 2f 65 1b
c658 : 6d 85 2f a5 30 65 6e 85 fb
c660 : 30 18 a5 31 65 6d 85 31 66
c668 : a5 32 65 6e 85 32 a0 00 ba
c670 : b1 6b 91 69 e6 69 d0 02 69
c678 : e6 6a e6 6b d0 02 e6 6c 4c
c680 : a5 6b c5 33 90 ea a5 6c 82
c688 : c5 34 90 e4 20 8e a6 4c d2
c690 : ae a7 20 78 c2 20 c6 ff 71
c698 : 20 78 c2 86 ab 20 7e c2 b1
c6a0 : 20 cf ff 48 20 b7 ff d0 12
c6a8 : 0a 68 91 62 c8 c4 ab d0 9a
c6b0 : ef f0 0a 68 a9 00 91 62 4d
c6b8 : c8 c4 ab d0 f9 20 18 c3 70
c6c0 : b0 d0 4c f9 c1 a9 00 aa e9
c6c8 : 9d 00 d4 e8 e0 19 d0 f8 c3
c6d0 : a9 0f 8d 18 d4 a9 09 8d 41
c6d8 : 05 d4 a9 11 8d 04 d4 8d 3b
c6e0 : 00 d4 a9 64 8d 01 d4 4c 0e
c6e8 : f9 c1 20 78 c2 86 fd 20 72
c6f0 : 78 c2 18 a4 fd 20 f0 ff 09
c6f8 : 4c f9 c1 20 09 c2 a5 2b 49
c700 : a6 2c 20 17 a6 38 a5 5f 29
c708 : e9 01 a4 60 b0 01 88 85 e7
c710 : 41 84 42 4c f9 c1 20 cc 75
c718 : ff a0 00 20 cf ff c9 d0 aa
c720 : f0 06 99 c8 c9 c8 d0 f3 a1
c728 : 84 ab 20 7e c2 b9 c8 c9 0a
c730 : 91 62 c8 c4 61 d0 f6 4c ce
c738 : f9 c1 a2 00 86 14 86 15 08
c740 : 20 09 c2 20 a3 a8 4c f9 3e
c748 : c1 a9 e4 a2 a7 8d 08 03 78
c750 : 8e 09 03 4c 86 e3 20 78 a6
c758 : c2 86 ab 20 df c1 a0 00 db
c760 : b1 22 48 20 7e c2 68 91 fb
c768 : 62 c8 c4 61 d0 f9 20 18 19
c770 : c3 b0 e3 4c f9 c1 38 20 dc
c778 : f0 ff 86 fd a9 00 48 20 c5
c780 : b7 c2 68 a4 fd 20 b7 c2 8c
c788 : 4c f9 c1 20 df c1 84 aa b9
c790 : a0 00 b1 22 99 c8 c9 c8 7a
c798 : c4 aa d0 f6 20 df c1 a5 18
c7a0 : 22 a6 23 85 fb 86 fc 84 80
c7a8 : ab 20 78 c2 86 fd a0 00 b5
c7b0 : bd c8 c9 d1 fb f0 07 e8 b3
c7b8 : e4 aa d0 f4 f0 18 86 a7 fd
c7c0 : a0 00 bd c8 c9 d1 fb d0 a5
c7c8 : 08 e8 c8 c4 ab f0 0b d0 1f
c7d0 : f1 e8 a0 00 f0 da a0 ff c6
c7d8 : 84 a7 a4 a7 c8 a9 00 20 68
c7e0 : b7 c2 20 18 c3 b0 03 4c 6a
c7e8 : f9 c1 4c 8b c7 20 09 c2 6d
c7f0 : 20 df c1 84 aa a0 00 a2 f6
c7f8 : 20 d0 da c2 9d c8 c9 e8 30
c800 : c8 c4 aa d0 f4 86 fd a0 ac
c808 : 00 b9 c8 c9 91 fb c8 c4 f6
c810 : fd d0 f6 20 18 c3 b0 d5 45
c818 : 4c f9 c1 20 df c1 a0 00 64
c820 : 2c fa c2 48 c8 20 da c2 e5
c828 : a8 68 20 b7 c2 20 18 c3 18
c830 : b0 e9 4c f9 c1 20 df c1 47
c838 : 84 aa a0 00 a2 00 20 da 9a
c840 : c2 9d c8 c9 e8 c8 c4 aa 79
c848 : d0 f4 86 ab 20 7e c2 a0 ec
c850 : 00 b9 c8 c9 91 62 c8 c4 71
c858 : 61 d0 f6 20 18 c3 b0 d5 f1
c860 : 4c f9 c1 20 09 c2 a0 00 46
c868 : a5 fc 20 fa c2 c8 a5 fb f4
c870 : 20 fa c2 a0 04 84 ab 20 25
c878 : 7e c2 a0 00 b9 c8 c9 91 ac
c880 : 62 c8 c4 61 d0 f6 20 18 19
c888 : c3 b0 d8 4c f9 c1 20 78 82
c890 : c2 8e 20 d0 20 78 c2 8e a9
c898 : 21 d0 4c f9 c1 20 09 c2 3a
c8a0 : 20 78 c2 86 ab 20 78 c2 a1
c8a8 : 86 fd 20 78 c2 86 aa a9 a2
c8b0 : 4c 85 a7 a5 ab a6 fd a4 8e
c8b8 : aa 20 a7 00 20 18 c3 b0 8f
c8c0 : dc 4c f9 c1 00 00 ff 09 8b

```

Listing »Disk-Basic«. Bitte beachten Sie die Eingabehinweise auf Seite 8.

Record-Befehl für den C 64

Dieses Programm erleichtert Ihnen den Umgang mit relativen Files.

Bei relativen Files kann auf jeden Satz des Files direkt zugegriffen werden, ohne die vorherigen Sätze lesen zu müssen. Außerdem kann ein eröffnetes File nach Belieben beschrieben und gelesen werden, es ist also zum Beispiel möglich, Satz 12 zu beschreiben, auch wenn bereits höhere Sätze existieren. Ein relatives File wird durch folgenden OPEN-Befehl eingerichtet:

OPEN lfn,ga,sa,"filename,l"+chr\$(sl). Dabei bedeutet:

- lfn = logische Filenummer.
- ga = Geräteadresse (normalerweise 8)
- sa = Sekundäradresse (mindestens 2, höchstens 14)
- sl = maximale Satzlänge des relativen Files (mindestens 1, höchstens 254). Diese Angabe ist entscheidend für den späteren Diskettenplatzverbrauch des Files, da auch Sätze, die nicht mit voller Satzlänge beschrieben werden, den gesamten durch sl reservierten Platz verbrauchen. Ist ein relatives File erst einmal eingerichtet, so kann die Satzlänge nicht mehr verändert werden. Es genügt dann der folgende OPEN-Befehl:

OPEN lfn,ga,sa,"filename"
 Selbstverständlich muß ein relatives File auch mit CLOSE lfn geschlossen werden. Auf die einzelnen Sätze eines relativen Files wird normal mit PRINT #lfn, INPUT #lfn oder GET #lfn zugegriffen, allerdings muß vor jedem Zugriff auf den gewünschten Satz positioniert werden. Dies übernimmt die vorliegende Basic-Erweiterung.

Durch RECORD # lfn,sn,(p) wird auf den Satz mit der Nummer sn positioniert, wird zusätzlich noch p angegeben, so wird innerhalb des Satzes das p-te Zeichen angewählt. Der neue Befehl muß von THEN durch einen Doppelpunkt abgetrennt werden. Es können folgende Basic-Fehlermeldungen auftreten:

?SYNTAX ERROR: Schreibfehler im Befehl oder die Erweiterung war nicht eingeschaltet.
 ?ILLEGAL QUANTITY ERROR: Ein Argument lag außerhalb des erlaubten Bereichs.
 ?FILE NOT OPEN ERROR: Das File mit der logischen Filenummer lfn war nicht eröffnet.
 ?ILLEGAL DEVICE NUMBER ERROR: Dem File mit der Nummer lfn ist ein Gerät mit einer Adresse kleiner als 8 oder größer als 15 zugeordnet.

Es können außerdem folgende Floppy-Fehlermeldungen auftreten (Sie müssen durch den Fehlerkanal ausgelesen werden):

50,RECORD NOT PRESENT: Der Satz, auf den positioniert wurde, existiert nicht. Diese Meldung kann ignoriert werden, wenn der Satz beschrieben werden soll, denn durch das Beschreiben wird er eingerichtet. (Und alle nicht existierenden Sätze mit kleinerer Nummer ebenfalls.) Aus Geschwindigkeitsgründen empfiehlt es sich, bei der Einrichtung eines Files den höchsten Satz zuerst zu beschreiben.
 51,OVERFLOW IN RECORD: Die maximale Satzlänge sl wurde beim Schreiben überschritten. Zu beachten ist, daß das Carriage Return am Ende des Satzes mitzählt.
 52,FILE TOO LARGE: Die Diskette ist voll, der letzte Schreibzugriff kann nicht durchgeführt werden.

```

20 FOR I=49152 TO 49375:READ A:POKE I,A:S=
S+A:NEXT I <193>
30 IF S<>25982 THEN PRINT"PRUEFSUMMENFEHLE
R":STOP <103>
40 SYS 49152:NEW <225>
100 DATA 76,29,192,173,28,192,240,17,173,2
6,192,141,8,3,173,27,192,141,9,3 <071>
101 DATA 169,0,141,28,192,96,228,167,0,169
,255,141,28,192,173,8,3,201,75,240 <151>
102 DATA 3,141,26,192,173,9,3,201,192,240,
3,141,27,192,169,75,141,8,3,169 <045>
103 DATA 192,141,9,3,96,35,68,176,67,69,82
,35,68,176,8,165,122,72,165,123 <022>
104 DATA 72,32,115,0,201,100,240,15,160,6,
217,64,192,208,26,32,115,0,136,208 <102>
105 DATA 245,240,27,32,115,0,160,4,217,70,
192,208,8,32,115,0,136,208,245,240 <246>
106 DATA 9,104,133,123,104,133,122,108,26,
192,104,104,32,158,183,32,15,243 <096>
107 DATA 240,5,162,3,76,55,164,32,31,243,1
65,185,41,15,72,165,186,201,16,144 <130>
108 DATA 5,162,9,76,55,164,201,8,144,247,3
2,177,255,169,111,133,185,32,147 <244>
109 DATA 255,169,80,32,168,255,32,253,174,
32,138,173,32,247,183,104,32,168 <230>
110 DATA 255,165,20,32,168,255,165,21,32,1
68,255,32,121,0,240,7,32,241,183 <238>
111 DATA 138,32,168,255,32,174,255,76,174,
167 <152>
    
```

Listing 1. Basic-Lader »Record-Befehle«. Bitte beachten Sie die Eingabehinweise auf Seite 6.

Der Basic-Interpreter besitzt einen Zeiger (Adresse \$0308, im Listing »decod« genannt), der auf die Routine zur Befehlsinterpretation zeigt. Dieser Vektor wird nach dem Aufruf der Erweiterung zuerst gemerkt und danach auf eine eigene Routine zur Befehlsinterpretation gesetzt. Vor dem Merken wird noch geprüft, ob der Vektor schon auf die eigene Routine zeigt. Damit wird verhindert, daß der Computer bei einem versehentlichen

Aufbau des Programms

Der Basic-Interpreter besitzt einen Zeiger (Adresse \$0308, im Listing »decod« genannt), der auf die Routine zur Befehlsinterpretation zeigt. Dieser Vektor wird nach dem Aufruf der Erweiterung zuerst gemerkt und danach auf eine eigene Routine zur Befehlsinterpretation gesetzt. Vor dem Merken wird noch geprüft, ob der Vektor schon auf die eigene Routine zeigt. Damit wird verhindert, daß der Computer bei einem versehentlichen

Bild 2. Symbol-Tabelle

| | | | | | |
|------|----------|------|--------|------|--------|
| 0073 | chrget | ffb1 | listen | c036 | l2 |
| 0079 | chrgot | ff93 | seclis | c041 | tab1 |
| 007a | chrvek | ffa8 | iecout | c047 | tab2 |
| 0308 | decod | ffa | unlist | c04b | record |
| a437 | error | 0014 | integ | c05a | 13 |
| a7ae | schleife | 00b8 | filenr | c067 | simon |
| ad8a | getrec | 00b9 | secadr | c06c | 13s |
| aefd | chkcom | 00ba | ga | c079 | out |
| b79e | getbyt | c019 | endoff | c082 | ok |
| b7f1 | comget | c01a | vekt | c091 | 14 |
| b7f7 | fac16 | c01c | flag | c09f | illdev |
| f30f | search | c01d | on | c0a4 | 15 |
| f31f | parset | c02c | l1 | c0da | ende |

Bild 1. Befehlskodierung

| | | |
|-----------|----------|--------------------------|
| * = 49152 | | |
| chrget | = \$73 | :holt nächstes Zeichen |
| chrgot | = \$79 | :holt letztes Zeichen |
| chrvek | = \$7a | :Chrgetzeiger |
| decod | = \$0308 | :Vektor für Befehlsdec. |
| error | = \$a437 | :Fehlermeldung ausgeben |
| schleife | = \$a7ae | :Interpreterschleife |
| getrec | = \$ad8a | :Recordnummer holen |
| chkcom | = \$aefd | :prüft auf Komma |
| getbyt | = \$b79e | :holt Byte in x |
| comget | = \$b7f1 | :chkcom + getbyt |
| fac16 | = \$b7f7 | :FAC nach 16-Bit wandeln |
| search | = \$f30f | :sucht logische Filenr. |
| parset | = \$f31f | :setzt Fileparameter |
| listen | = \$ffb1 | :LISTEN senden |
| seclis | = \$ff93 | :Sekundärad. nach LISTEN |
| iecout | = \$ffa8 | :Ausgabe auf IEC-bus |
| unlist | = \$ffae | :UNLISTEN senden |
| integ | = \$14 | :Integer Wert |
| filenr | = \$b8 | :logische Filenummer |
| secadr | = \$b9 | :Sekundäradresse |
| ga | = \$ba | :Gerätenummer |

| | | | | | |
|--|---------------|-------------------------------|------------------------|--|--|
| * = 49152 ;Startadresse | | | | | |
| | jmp on | ;zum Einbinden | c000 4c 1d c0 | | |
| ;***** Erweiterung abschalten ***** | | | | | |
| | lda flag | ;schon abgeschaltet? | c003 ad 1c c0 | | |
| | beq endoff | | c006 f0 11 | | |
| | lda vekt | ;Vektor wieder her- | c008 ad 1a c0 | | |
| | sta decod | ;stellen | c00b 8d 08 03 | | |
| | lda vekt+1 | | c00e ad 1b c0 | | |
| | sta decod+1 | | c011 8d 09 03 | | |
| | lda #0 | ;Flag fuer Abge- | c014 a9 00 | | |
| | sta flag | ;schaltet setzen | c016 8d 1c c0 | | |
| endoff | rts | | c019 60 | | |
| | vekt | .word \$a7e4 | c01a e4 a7 | | |
| | flag | .byte \$00 | c01c 00 | | |
| ;***** Erweiterung einschalten ***** | | | | | |
| on | lda #\$ff | ;Flag fuer Einge- | c01d a9 ff | | |
| | sta flag | ;schaltet setzen | c01f 8d 1c c0 | | |
| | lda decod | | c022 ad 08 03 | | |
| | cmp #<record | | c025 c9 4b | | |
| | beq 11 | | c027 f0 03 | | |
| | sta vekt | | c029 8d 1a c0 | | |
| 11 | lda decod+1 | | c02c ad 09 03 | | |
| | cmp #>record | | c02f c9 c0 | | |
| | beq 12 | | c031 f0 03 | | |
| | sta vekt+1 | | c033 8d 1b c0 | | |
| 12 | lda #<record | | c036 a9 4b | | |
| | sta decod | | c038 8d 08 03 | | |
| | lda #>record | | c03b a9 c0 | | |
| | sta decod+1 | | c03d 8d 09 03 | | |
| | rts | | c040 60 | | |
| ;***** Interpreterdarstellung von **** | | | | | |
| ;***** RECORD# ***** | | | | | |
| tab1 | .byte | \$23,\$44,\$b0,\$43,\$45,\$52 | c041 23 44 b0 43 45 52 | | |
| tab2 | .byte | \$23,\$44,\$b0,\$08 | c047 23 44 b0 08 | | |
| ;***** Test auf RECORD# ***** | | | | | |
| record | lda chrvek | ;Chrgetzeiger | c04b a5 7a | | |
| | pha | ;retten | c04d 48 | | |
| | lda chrvek+1 | | c04e a5 7b | | |
| | pha | | c050 48 | | |
| | jsr chrget | | c051 20 73 00 | | |
| | cmp #\$64 | ;Simon's Basic | c054 c9 64 | | |
| | beq simon | ;TOKEN | c056 f0 0f | | |
| | ldy #6 | | c058 a0 06 | | |
| 13 | cmp tab1-1,y | | c05a d9 40 c0 | | |
| | bne out | | c05d d0 1a | | |
| | jsr chrget | | c05f 20 73 00 | | |
| | dey | | c062 88 | | |
| | bne 13 | | c063 d0 f5 | | |
| | beq ok | | c065 f0 1b | | |
| simon | jsr chrget | | c067 20 73 00 | | |
| | ldy #4 | | c06a a0 04 | | |
| 13s | cmp tab2-1,y | | c06c d9 46 c0 | | |
| | bne out | | c06f d0 08 | | |
| | jsr chrget | | c071 20 73 00 | | |
| | dey | | c074 88 | | |
| | bne 13s | | c075 d0 f5 | | |
| | beq ok | | c077 f0 09 | | |
| ;***** Weiter mit normalem Basic **** | | | | | |
| out | pla | | c079 68 | | |
| | sta chrvek+1 | | c07a 85 7b | | |
| | pla | | c07c 68 | | |
| | sta chrvek | | c07d 85 7a | | |
| | jmp (vekt) | | c07f 6c 1a c0 | | |
| ;***** eigentlicher RECORD#-Befehl *** | | | | | |
| ok | pla | ;Zeiger von Stack | c082 68 | | |
| | pla | | c083 68 | | |
| | jsr getbyt | ;holt Byte nach x | c084 20 9e b7 | | |
| | jsr search | ;sucht log. Filenr. | c087 20 0f f3 | | |
| | beq 14 | ;gefunden | c08a f0 05 | | |
| | ldx #3 | ;file not open | c08c a2 03 | | |
| | jmp error | ;ausgeben | c08e 4c 37 a4 | | |
| 14 | jsr parset | ;Fileparameter setzen | c091 20 1f f3 | | |
| | lda secadr | ;Sekundaeradresse | c094 a5 b9 | | |
| | and #\$0f | | c096 29 0f | | |
| | pha | ; = Kanalnummer | c098 48 | | |
| | lda ga | | c099 a5 ba | | |
| | cmp #16 | | c09b c9 10 | | |
| | bcc 15 | ;ga < 16 | c09d 90 05 | | |
| illdev | ldx #9 | ;illegal device nr. | c09f a2 09 | | |
| | jmp error | | c0a1 4c 37 a4 | | |
| 15 | cmp #8 | | c0a4 c9 08 | | |
| | bcc illdev | ;ga < 8 | c0a6 90 f7 | | |
| | jsr listen | | c0a8 20 b1 ff | | |
| | lda #\$6f | ;15 and \$60 | c0ab a9 6f | | |
| | sta secadr | | c0ad 85 b9 | | |
| | jsr seclis | | c0af 20 93 ff | | |
| | lda #'p | ;positionieren | c0b2 a9 50 | | |
| | jsr iecout | ;senden | c0b4 20 a8 ff | | |
| | jsr chkcom | | c0b7 20 fd ae | | |
| | jsr getrec | ;Recordnummer holen | c0ba 20 8a ad | | |
| | jsr fac16 | ;FAC nach 16-Bit | c0bd 20 f7 b7 | | |
| | pla | ;Kanalnummer | c0c0 68 | | |
| | jsr iecout | | c0c1 20 a8 ff | | |
| | lda integ | ;Recordnr. lo | c0c4 a5 14 | | |
| | jsr iecout | | c0c6 20 a8 ff | | |
| | lda integ+1 | ;Recordnr. hi | c0c9 a5 15 | | |
| | jsr iecout | | c0cb 20 a8 ff | | |
| | jsr chrget | ;letztes Zeichen | c0ce 20 79 00 | | |
| | beq ende | ;kein dritter Param.? | c0d1 f0 07 | | |
| | jsr comget | ;Komma und Parameter | c0d3 20 f1 b7 | | |
| | txa | | c0d6 8a | | |
| | jsr iecout | ;Position in Record | c0d7 20 a8 ff | | |
| ende | jsr unlist | | c0da 20 ae ff | | |
| | jmp schleife; | zur Interpreterschl. | c0dd 4c ae a7 | | |

Listing 2.
Source-Code von
»Record-Befehl«

chen Doppelaufwurf abstürzt. Die eigene Routine rettet zunächst den Zeiger, der auf das augenblickliche Zeichen im Basic-Text zeigt, und ruft dann die CHRGET-Routine (Befehls-codierung siehe Bild 1, Symboltabelle Bild 2) auf. Diese Routine holt das nächste Zeichen aus dem Basic-Text. Sie überliest automatisch Leerzeichen. Wenn das Ende einer Zeile oder ein Doppelpunkt erreicht wird, so wird die Z-Flag des Prozessors gesetzt. Ein Problem ergibt sich dadurch, daß in »RECORD #« der Simons Basic-Befehl »REC« vorkommt. Er wird bei Verwendung von Simons Basic also in eine interne Darstellung beginnend mit \$64 umgewandelt. Das Programm testet daher, ob \$64 vorliegt und verzweigt bei positiver Antwort. In beiden Fällen wird nun getestet, ob der Befehl »RECORD #« vorliegt. Unter dem Namen tab 1 (beziehungsweise tab 2 für Simons Basic) steht (rückwärts) die interne Darstellung des Befehls »RECORD«. Dabei ist zu beachten, daß »OR« wiederum nicht im ASCII-Code abgelegt ist, sondern als Token. Wird nur eine Abweichung gefunden, so wird der gemerkte CHRGET-Zeiger zurückgeholt und zur normalen Interpreterroutine gesprungen, anderenfalls tritt eine eigene Routine in Funktion (sie beginnt beim Label OK). Sie holt die Satznummer und, wenn angegeben, die Position im Satz aus dem Basic-Text und sendet die Informationen an die Floppy über die Sekundäradresse 15 (Kommandokanal). Die Floppy erwartet die Werte in folgender Form: »p« als Zeichen, das positioniert werden soll; die Sekundäradresse, mit der das relative File eröffnet wurde; die Satznummer in der Reihenfolge niederwertiges — höherwertiges Byte und, falls angegeben, die Position im Satz. Es werden folgende Routinen des Interpreters und des Betriebssystems verwendet:

- getbyt** : Holt einen beliebigen Ausdruck in das x-Register.
- search** : Sucht nach der logischen Filenummer in x.
- parset** : Holt logische Filenummer, Geräteadresse und Sekundäradresse in die Speicherzellen \$b8,\$ba und \$b9
- listen** : Teilt dem angesprochenen Gerät mit, daß es Daten empfangen soll.
- seclis** : Sendet die Sekundäradresse zum angesprochenen Gerät.
- iecout** : Sendet den Wert im Akkumulator zum angesprochenen Gerät.
- chkcom** : Prüft auf Komma im Basic-Text.
- getrec** : Holt einen numerischen Ausdruck in den Fließkomma-Akkumulator.
- fac16** : Wandelt den Wert im Fließkomma-Akkumulator in eine 16-Bit-Zahl und speichert sie in \$14 und \$15.
- comget** : =chkcom + getbyte.
- unlist** : Beendet die Übertragung an die Floppy.

Das Programm wurde mit dem Assembler »ASSI« übersetzt, eine Übernahme auf andere Assembler dürfte nicht schwierig sein. Der Basic-Lader (Listing 1, Source-Code: Listing 2) legt das Programm ab der Adresse 49152 (\$C000) ab, schaltet die Erweiterung ein und löscht sich dann selbst, er muß also vorher auf Diskette abgespeichert werden. Die Erweiterung wird durch SYS49152 ein- und durch SYS49155 ausgeschaltet. STOP/RESTORE schaltet die Erweiterung nicht ab.

(Bernward Bretthauer/rg)

Aufpoliertes Basic

39 zusätzliche und leistungsfähige Befehle stellt dieses Programm dem Programmierer zur Verfügung.

Das Programm (siehe Listing) belegt nur 8 KByte des Basic-RAMs. Der Restbedarf an Arbeitsspeicher wurde mit den Bereichen \$C000 bis \$CFFF und \$E000 bis \$FFFF abgedeckt.

Der zusätzliche Interpreter arbeitet nach demselben Prinzip wie der Commodore-Interpreter, das heißt jeder Befehl belegt nur genau 1 Byte. Die Erweiterung ist so geschrieben, daß Verzögerungen der Befehlserkennung kaum auftreten.

Das Programm wird mit SYS 64738 oder einem RESET-Schalter gestartet.

Die Basic-Erweiterung kann noch durch eigene Befehle ergänzt werden. Dazu hängt man die neuen Befehlsnamen an das Ende der Tabelle, die bei \$800A beginnt. Die Adresse der neuen Befehle wird an das Ende der Tabelle gehängt, die bei \$8275 beginnt. Hierbei muß der Wert »Adresse — 1« angegeben werden. Beispiel: Für eine Routine ab \$9E00 muß die Adresse \$9DFF (im Lo-Byte/Hi-Byte-Format) angegeben werden. Bei den Befehlen ist darauf zu achten, daß das letzte Zei-

| | | | |
|--------------------------|--|--------------------------|---|
| Die Befehle: | | HELP | — Gibt alle neuen Befehle aus |
| COLD | — Bewirkt einen Kaltstart, der den C 64 in den Startzustand bringt | HIRES X,Y | — Schaltet HiRes-Grafik ein, mit Vordergrundfarbe »X« und Zeichenfarbe »Y« (Grafik liegt bei \$E000) |
| REVIC | — Bewirkt, daß der Videocontroller in den Startzustand gebracht wird | TEXT | — Schaltet die HiRes-Grafik ab, der Bildschirm bleibt dabei vorhanden |
| DUMP | — Gibt alle benutzten Variablen aus | PLOT X,Y | — Setzt den Punkt mit den Koordinaten (X,Y) |
| DIRECTORY | — Gibt das Directory aus, wobei Programme im Speicher bleiben | CLOT X,Y | — Löscht den Punkt mit den Koordinaten (X,Y) |
| DERR | — Gibt den Fehlerkanal der Floppy auf dem Bildschirm aus | GRDI A,"NAME",GN | — Speichert beziehungsweise lädt ein Bild auf (von-) : GN=8: DISK GN=1:DATASETTE, wobei A=1 Abspeichern und A=0 Laden bedeutet |
| CURSET X,Y,'TEXT' | — Setzt Cursor und Position (X,Y) und druckt Text | LINE X1,Y1,X2,Y2 | — Zeichnet eine Linie von (X1,Y1) nach (X2,Y2) |
| RENEW | — Holt ein mit »NEW« gelöscht Programm in den Speicher zurück | REC X1,Y1,X2,Y2 | — Zeichnet ein Rechteck mit der linken oberen Ecke (X1,Y1) und der rechten unteren Ecke (X2,Y2) |
| AUTLIN X | — Beginnt automatische Zeilennummerierung nach der nächsten eingegebenen Zeilennummer mit der Schrittweite »X« | BIT X | — Bestimmt, ob beim folgenden Grafik-Kommando gezeichnet oder gelöscht wird; X=1 : Zeichnen ; X=0 : Löschen |
| DELETE X-Y | — Löscht die Zeilennummern von »X« bis »Y« | BLOCK X1,Y1,X2,Y2 | — Zeichnet ein ausgefülltes Rechteck mit der linken oberen Ecke (X1,Y1) und der rechten unteren Ecke (X2,Y2) |
| DELETE -X | — Löscht die Zeilennummern bis »X« | GCL | — Löscht die im Speicher befindliche Grafik |
| DELETE X- | — Löscht die Zeilennummern von »X« bis Programmende | INV | — Invertiert die im Speicher befindliche Grafik |
| SSTP | — Bewirkt eine Einzelschrittbearbeitung des Programmes, wobei nach jedem Druck auf die SHIFT-Taste ein Befehl ausgeführt wird | TRI | — Zeichnet ein Dreieck mit den Eckkoordinaten (E1,E2);(E3,E4);(E5,E6), zu beachten ist, daß (E5,E6) immer die Ecke ist, die auf dem Bildschirm unten rechts gezeichnet wird, da es ansonsten zu Fehlfunktionen kommen kann (eine Linie wird nicht gezeichnet) |
| OFF | — Schaltet sowohl die »SSTP«-als auch die »TRACE«-Anweisung ab | E1,E2,E3,E4,E5,E6 | — Zeichnet einen Kreis mit den Mittelpunktkoordinaten (MX,MY) und den X,Y Achsen-Radien RX,RY |
| HOLD 'X' | — Wartet, bis die Taste in '' gedrückt wird, wird als Taste ein '?' angegeben, so wird jede Taste akzeptiert | ANGL | — Zeichnet einen Winkel der Größe WI, vom Punkt (MX,MY) mit den entsprechenden X,Y Achsen-Radien DX,DY |
| RENUM X,Y,Z | — Numeriert das Programm von Zeile »X« bis Ende mit der Schrittweite »Z« neu und beginnt bei der Neunummerierung mit »Y« | PJOB SN,X,Y | — Setzt das Sprite mit der Nummer »SN« an den Punkt (X,Y) ('X' darf Werte von 0 - 344 enthalten) |
| FIND 'TEXT' | — Sucht im Programm nach »TEXT« und gibt gefundene Zeilen aus | SJOB | — Setzt die Parameter für ein Sprite; »SN«=Sprite.Nummer. »MC«=1:Multicolor; »MC«=0:Monochrom; (im Falle Multicolor müssen zwei weitere Parameter für die Farben angegeben werden) |
| FIND GOTO | — Sucht im Programm nach dem Befehl GOTO und gibt ebenfalls gefundene Zeilen aus, wobei zu beachten ist, daß, wenn man »neue« Befehle sucht, nicht die Bezeichnung des Befehls ausgegeben wird, sondern ein alter Befehl, was aber nicht weiter schlimm ist, da ja nur die Zeilennummer von Interesse ist und das Listing nicht beeinflußt wird | SN,MC,EX,CO,PR | »EX«=0:Keine Expansion; »EX«=1:X-Expansion; »EX«=3:Y-Expansion; »CO«:Farbe; »PR«:Sprite- Hintergrund Priorität |
| TRACE X | — Zeigt in der rechten oberen Ecke des Bildschirms die aktuelle Zeilennummer an und wartet auf die SHIFT-Taste. Da sich diese aber nicht als einzelnen Tastendruck abfragen läßt, wird mit »X« die Geschwindigkeit angegeben von »0« sehr langsam bis »255« sehr schnell. Während des Ablaufs ist die STOP-Taste nicht aktiv. Zum Abbruch erst die Commodore-Taste, dann die STOP-Taste drücken. | DJOB SN,BL | — »BL« gibt den Block an, der dem Sprite Nummer »SN« zugeordnet wird |
| INIT | — Initialisiert das Disk-Laufwerk | EMOB SN,A | — Wenn »A«=0, dann ist das Sprite mit der Nummer »SN« zu sehen, wenn »A«=1, dann ist das Sprite mit der Nummer »SN« nicht zu sehen |
| COLLECT | — Führt einen Validate auf der Diskette durch | SCR X,Y | — Setzt Bildschirm-Rahmenfarbe »X«; Hintergrundfarbe »Y« |
| SCRATCH "NAME" | — Löscht auf Disk das mit »NAME« bezeichnete Programm | | |
| HRDCPY | — Gibt auf dem Drucker eine Hardcopy aus (Drucker Adresse 4, auf fast allen Druckern lauffähig) | | |

Tabelle. Befehlsübersicht

chen geSHIFTet angegeben wird (MSB gesetzt), damit der Computer weiß, wo das Befehlswort endet. Alle Befehle lassen sich abkürzen, da sie in 1-Byte-Tokens umgesetzt werden.

Zusätzliche Befehle kann man risikolos von \$9875 unmittelbar nach dem JMP \$984F-Befehl bis \$9FFF (Basic-RAM-Ende) hinzufügen. (Markus Feldmann/rg)

Hinweise:

Wird während des Listens die SHIFT-Taste gedrückt, so hält der Computer so lange damit an, wie diese gedrückt bleibt; wird jedoch die »Commodore-Taste« gedrückt, so wird sehr langsam gelistet (auch nur solange die Taste gedrückt bleibt).

Wenn der C 64 sich längere Zeit (einige Sekunden) mit der Grafik beschäftigt (Abspeichern, Laden von Grafiken) sollte man ihn auf keinen Fall mit RUN/STOP-RESTORE abbrechen, da es sonst zum System-Absturz kommen kann (das Kern-ROM ist während des Zeichnens, und der Basic-Interpreter

während des Abspeicherns und Einladens von Grafiken abgeschaltet)!

Die Erweiterung ist darauf ausgelegt, auf ein EPROM gebrannt zu werden. Dieses ist daher ein recht einfacher Vorgang. Ist die Erweiterung nur als Programm vorhanden, läßt sie sich durch »POKE32776,0:COLD« abschalten.

Wichtig: Sollen die neuen Basic-Befehle nach »IF...THEN« benutzt werden, so müssen sie mit einem Doppelpunkt vom »THEN« getrennt werden. (Beispiel: IF A=0 THEN:PLOT X,Y)

```
programm : code free      8000 9901
8000 : 2c 83 00 83 c3 c2 cd 38 58
8008 : 30 0d 43 4f 4c c4 52 45 38
8010 : 56 49 c3 44 49 52 45 43 47
8018 : 54 4f 52 d9 44 45 52 d2 41
8020 : 43 55 52 53 45 d4 52 45 dc
8028 : 4e 45 d7 41 55 54 4c 49 f3
8030 : ce 44 45 4c 45 54 c5 53 b0
8038 : 53 54 d0 4f 46 c6 44 55 2a
8040 : 4d d0 48 4f 4c c4 52 45 b0
8048 : 4e 55 cd 46 49 4e c4 54 40
8050 : 52 41 43 c5 48 52 44 43 7b
8058 : 50 d9 53 43 52 41 54 43 d9
8060 : c8 49 4e 49 d4 43 4f 4c c7
8068 : 4c 45 43 d4 48 45 4c d0 44
8070 : 48 47 d2 54 45 58 d4 50 a6
8078 : 4c 4f d4 43 50 4c 4f d4 58
8080 : 47 52 44 c9 4c 49 4e c5 0e
8088 : 52 45 c3 42 49 d4 42 4c 93
8090 : 4f 43 cb 41 4e 47 cc 47 7d
8098 : 43 cc 49 4e d6 54 52 c9 4a
80a0 : 43 49 52 43 4c c5 50 4d 54
80a8 : 4f c2 53 4d 4f c2 44 4d 8e
80b0 : 4f c2 45 4d 4f c2 53 43 3a
80b8 : d2 00 00 00 00 00 00 00 8b
80c0 : 00 00 00 00 00 00 00 00 c1
80c8 : 00 00 00 00 00 00 00 00 c9
80d0 : 00 00 00 00 00 00 00 00 d1
80d8 : 00 00 00 00 00 00 00 00 d9
80e0 : 00 00 00 00 00 00 00 00 e1
80e8 : 00 00 00 00 00 00 00 00 e9
80f0 : 00 00 00 00 00 00 00 00 f1
80f8 : 00 00 00 00 00 00 00 00 f9
8100 : 00 00 00 00 00 00 00 00 01
8108 : 00 00 00 00 00 00 00 00 09
8110 : 00 00 00 00 00 00 00 00 11
8118 : 00 00 00 00 00 00 00 00 19
8120 : 00 00 00 00 00 00 00 00 21
8128 : 00 00 00 00 00 00 00 00 29
8130 : 00 00 00 00 00 00 00 00 31
8138 : 00 00 00 00 00 00 00 00 39
8140 : 00 00 00 00 00 00 00 00 41
8148 : 00 00 00 00 00 00 00 00 49
8150 : 00 00 00 00 00 00 a6 7a e0
8158 : a0 04 84 0f bd 00 02 10 01
8160 : 07 c9 ff f0 3e e8 d0 f4 c2
8168 : c9 20 f0 37 85 08 c9 22 68
8170 : f0 55 24 0f 70 2d c9 3f 0c
8178 : d0 04 a9 99 d0 25 c9 30 a6
8180 : 90 04 c9 3c 90 1d 84 71 f3
8188 : a0 00 84 0b 88 86 7a ca e7
8190 : c8 e8 bd 00 02 38 f9 9e 43
8198 : a0 f0 f5 c9 80 d0 2f 05 bc
81a0 : 0b a4 71 e8 c8 99 fb 01 c2
81a8 : c9 00 f0 c9 38 38 e9 3a f0 52
81b0 : 04 c9 49 d0 02 85 0f 38 fe
81b8 : e9 55 d0 a0 85 08 bd 00 24
81c0 : 02 f0 e0 c5 08 f0 dc c8 38
81c8 : 99 fb 01 e8 d0 f0 a6 7a e0
81d0 : e6 0b c8 b9 9d a0 10 fa ba
81d8 : b9 9e a0 d0 b5 f0 0f bd bd
81e0 : 00 02 10 bd 99 fd 01 c6 b8
81e8 : 7b a9 ff 85 7a 60 a0 00 16
81f0 : b9 0a 80 d0 02 c8 e8 bd 6e
81f8 : 00 02 38 f9 0a 80 f0 f5 9b
8200 : c9 80 d0 04 05 0b d0 99 dd
8208 : a6 7a e6 0b c8 b9 09 80 86
8210 : 10 fa b9 0a 80 d0 e0 f0 41
8218 : c6 10 0f 24 0f 30 0b c9 61
8220 : ff f0 07 c9 cc b0 06 4c 95
8228 : 24 a7 4c f3 a6 38 e9 cb 1d
8230 : aa 84 49 a0 ff ca f0 08 ad
8238 : c8 b9 0a 80 10 fa 30 f5 f5
8240 : c8 b9 0a 80 30 05 20 47 b2
8248 : ab d0 f5 4c ef a6 20 73 fe
8250 : 00 20 57 82 4c ae a7 c9 f3
8258 : cc 90 04 c9 ff 90 06 20 83
8260 : 79 00 4c ed a7 38 e9 cc 28
8268 : 0a aa bd 76 82 48 bd 75 52
8270 : 82 48 4c 73 00 e1 fc 3f 19
8278 : 84 7c 84 e5 84 01 85 18 af
8280 : 85 56 85 ac 85 48 86 6f bb
8288 : 86 7a 86 3a 87 ef 87 a1 8e
8290 : 8a 9f 8d 35 8d b7 8d 22 05
8298 : 8e 3b 8e 4c 8e 80 8e df d8
82a0 : 8e f2 8e ef 8e a9 8f 89 d0
82a8 : 90 35 92 f5 92 06 93 70 bf
82b0 : 93 f4 93 0d 94 33 94 0a 8d
82b8 : 95 52 98 d9 96 cd 97 95 57
82c0 : 97 1f 98 90 90 90 90 90 10
82c8 : 90 90 90 90 90 90 90 90 c7
82d0 : 90 90 90 90 90 90 90 90 cf
82d8 : 90 90 00 00 00 00 00 00 b1
82e0 : 00 00 00 00 00 00 00 00 e1
82e8 : 00 00 00 00 00 00 00 00 e9
82f0 : 00 00 00 00 00 00 00 00 f1
82f8 : 00 00 00 00 00 00 00 00 f9
8300 : 20 bc f6 20 e1 ff d0 21 e4
8308 : a9 01 8d 20 d0 8d 21 d0 39
8310 : 20 15 fd 20 a3 fd 20 18 19
8318 : e5 a9 06 8d 20 d0 8d 21 06
8320 : d0 a9 99 20 d2 ff 4c 10 ae
8328 : 84 4c 72 fe 8e 16 d0 20 6c
8330 : a3 fd 20 50 fd 20 15 fd 15
8338 : 20 5b ff 58 20 53 e4 20 81
8340 : bf e3 20 78 83 a2 fb 9a 7a
8348 : a2 05 bd 60 83 9d 04 03 23
8350 : ca 10 f7 a9 ff 85 37 a9 b2
8358 : 7f 85 38 a9 99 4c 68 83 82
8360 : 56 81 19 82 4e 82 00 00 06
8368 : 20 d2 ff a9 06 8d 20 d0 15
8370 : 8d 21 d0 a2 80 4c f1 83 4f
8378 : a0 83 a9 8a 20 1e ab a9 8a
8380 : 83 a2 16 8d 17 03 8e 16 7c
8388 : 03 60 99 20 2a 2a 2a 17 c7
8390 : 20 20 20 20 20 20 45 58 95
8398 : 50 41 4e 44 45 44 20 56 49
83a0 : 33 2e 35 20 42 41 53 49 4a
83a8 : 43 20 20 20 20 20 2a df
83b0 : 2a 2a 2a 0d 20 20 20 df
83b8 : 20 20 20 31 39 38 34 20 7d
83c0 : 20 20 20 20 42 59 20 4d 07
83c8 : 41 52 4b 55 53 20 46 45 8c
83d0 : 4c 44 4d 41 4e 4e 0d 0d 5f
83d8 : 33 30 37 31 39 20 42 41 37
83e0 : 53 49 43 20 42 59 54 45 78
83e8 : 53 20 46 52 45 45 0d 00 da
83f0 : 00 ea a9 00 a8 91 2b cb 25
83f8 : 91 2b a5 2c 18 69 02 85 cd
8400 : 2d a5 2c 69 00 85 2e 20 5d
8408 : 63 a6 a2 80 4c 27 84 00 87
8410 : a9 31 8d 14 03 a9 ea 8d 7c
8418 : 15 03 a9 83 8d 17 03 a9 7b
8420 : 16 8d 16 03 6c 02 a0 a9 90
8428 : 31 8d 14 03 a9 ea 8d 15 d8
8430 : 03 a9 83 8d 17 03 a9 16 f7
8438 : 8d 16 03 4c e0 87 00 00 65
8440 : a0 2f b9 4c 84 99 00 d0 26
8448 : 88 10 f7 60 00 00 00 00 e2
8450 : 00 00 00 00 00 00 1b 18 d1 35
8460 : 00 00 c8 00 15 79 f0 00 73
8468 : 00 00 00 00 f6 f6 f1 f2 3d
8470 : f3 f4 f0 f1 f2 f3 f4 f5 e6
8478 : f6 f7 fc ff a0 a9 24 85 92
8480 : fb a9 fb 85 bb a9 00 85 13
8488 : bc a9 01 85 b7 a9 08 85 fe
8490 : ba a9 60 85 b9 20 d5 f3 c3
8498 : a5 ba 20 b4 ff a5 b9 20 8d
84a0 : 96 ff a9 00 85 90 a0 03 06
84a8 : 84 fb 20 a5 ff 85 fc a4 50
84b0 : 90 d0 2f 20 a5 ff a4 90 86
84b8 : d0 28 a4 fb 88 d0 e9 a6 49
84c0 : fc 20 cd bd a9 20 20 d2 b9
84c8 : ff 20 a5 ff a6 90 d0 12 97
84d0 : aa f0 06 20 d2 ff 4c c9 6a
84d8 : 84 a9 0d 20 d2 ff a0 02 2c
84e0 : d0 c6 20 42 f6 60 a9 08 8d
84e8 : 85 ba 20 b4 ff a9 6f 85 7f
84f0 : b9 20 96 ff 20 a5 ff 20 ce
84f8 : d2 ff c9 d0 d0 f6 20 ab 7a
8500 : ff 60 20 9e b7 8a 48 20 3c
8508 : fd ae 20 9e b7 68 ab 18 ca
8510 : 20 f0 ff 20 fd ae 4c a4 7c
8518 : aa a5 2b a4 2c 85 22 84 75
8520 : 23 a0 03 c8 b1 22 d0 fb d5
8528 : c8 98 18 65 22 a0 90 91 39
8530 : 2b a5 23 69 00 c8 91 2b 07
8538 : 88 a2 03 e6 22 d0 02 e6 2d
8540 : 23 b1 22 d0 f4 ca d0 f3 af
8548 : a5 22 69 02 85 2d a5 23 38
8550 : 69 00 85 2e 4c 63 a6 20 9b
8558 : 9e b7 8e ff cf a9 6a 8d 84
8560 : 03 04 a9 85 8d 05 03 4c a6
8568 : ae a7 68 48 c9 a1 d0 3a 6e
8570 : 88 b1 7a d0 0d a9 56 8d 1c
8578 : 04 03 a9 81 8d 05 03 4c 3e
8580 : 74 a4 a5 14 a6 15 18 6d 81
8588 : ff cf 90 01 e8 85 63 86 08
8590 : 62 a2 90 38 20 49 bc 20 ee
8598 : df bd 20 87 b4 20 82 b7 15
85a0 : 84 c6 b1 22 99 77 02 88 a6
85a8 : 10 f8 4c 7c a5 20 79 00 18
85b0 : 90 06 f0 04 c9 ab d0 17 6b
85b8 : 20 6b a9 20 13 a6 a5 5f b8
85c0 : 85 19 a5 60 85 1a 20 79 e4
85c8 : 00 f0 04 c9 ab f0 05 a2 16
85d0 : 0b 4c 3a a4 20 73 00 20 02
85d8 : 6b a9 d0 f3 a5 14 05 15 04
85e0 : d0 08 a9 ff 85 14 85 15 58
85e8 : d0 06 e6 14 d0 02 e6 15 da
85f0 : 20 13 a6 a5 5f 85 24 a5 f6
85f8 : 60 85 25 38 a5 25 a5 19 b0
8600 : a5 25 e5 1a 90 c9 a5 2d 3d
8608 : e5 24 85 5f a5 2e a5 25 fa
8610 : 85 60 18 a5 19 65 5f 85 c5
8618 : 2d a5 1a 65 60 85 2e a0 77
8620 : 00 b1 24 91 19 e6 19 d0 03
8628 : 02 e6 1a e6 24 d0 02 e6 9f
8630 : 25 38 a5 5f a9 01 85 5f 42
8638 : a5 60 e9 00 85 60 10 e1 e7
8640 : 20 59 a6 20 33 a5 4c 91 6f
8648 : e3 a9 54 a2 86 8d 08 03 64
8650 : 8e 09 03 60 08 48 8a 48 ad
8658 : 98 48 a5 7b c9 08 30 07 99
8660 : ad 8d 02 29 01 f0 f9 68 ca
8668 : a8 68 aa 68 28 4c 8e 82 1f
8670 : a9 4e a2 82 8d 08 03 8e 7b
8678 : 09 03 60 a5 2d a4 2e 85 8b
8680 : 14 84 15 c4 30 d0 02 e5 d1
```

Listing »Basic-Erweiterung«. Bitte beachten Sie die Eingabehinweise auf Seite 8.

Basic-Erweiterungen

8688 : 2f b0 18 69 02 90 01 c8 7d
 8690 : 85 22 84 23 20 c1 86 20 16
 8698 : f5 86 8a 10 07 20 fe 86 ef
 86a0 : 4c b0 86 60 98 30 06 20 55
 86a8 : 0e 87 4c b0 86 20 17 87 78
 86b0 : a9 0d 20 d2 ff a5 14 a4 09
 86b8 : 15 18 69 07 90 c1 c8 b0 b0
 86c0 : be a0 00 b1 14 aa 29 7f 3f
 86c8 : 20 d2 ff c8 b1 14 a8 29 1b
 86d0 : 7f f0 03 20 d2 ff 8a 10 04
 86d8 : 11 98 30 0a a9 21 20 d2 4c
 86e0 : ff 68 68 4c b0 86 a9 25 e7
 86e8 : d0 4e 98 10 04 a9 24 d0 c7
 86f0 : 47 60 20 d2 ff a9 20 20 d8
 86f8 : d2 ff a9 3d 00 3a a0 00 3d
 8700 : b1 22 aa c8 b1 22 a8 8a 6a
 8708 : 20 95 b3 4c 11 87 20 a6 84
 8710 : bb 20 dd bd 4c 1e ab 20 af
 8718 : 36 87 a0 02 b1 22 85 25 07
 8720 : 88 b1 22 85 24 88 b1 22 4c
 8728 : 85 26 f0 0a b1 24 20 d2 a0
 8730 : ff c8 c4 26 d0 f6 a9 22 39
 8738 : 4c d2 ff 20 73 00 48 c9 d8
 8740 : 3f d0 09 20 e4 ff f0 3f 3b
 8748 : 68 4c 5c 87 20 e4 ff f0 e9
 8750 : f9 c9 03 f0 f3 ba dd 01 9b
 8758 : 01 d0 f1 68 20 73 00 20 29
 8760 : 73 00 4c ae a7 ea ea ea 0f
 8768 : ea ea 20 06 a7 18 98 65 19
 8770 : 7a 8d fa cf a5 7b 69 00 45
 8778 : 8d fb cf a5 39 8d fc cf 3f
 8780 : a5 3a 8d fa cf a9 52 8d 14
 8788 : fe cf 4c ae a7 20 8a ad 58
 8790 : a8 ad 4e cf c9 52 d0 29 8d
 8798 : 98 d0 17 ad fd cf 85 3a fd
 87a0 : ad fc cf 85 39 ad fb cf 00
 87a8 : 85 7b ad fa cf 85 7a 4c 61
 87b0 : ae a7 a0 0e a9 00 99 f0 ff
 87b8 : cf 88 c0 09 d0 f8 4c ae 80
 87c0 : a7 a9 ca 85 22 a9 87 4c c5
 87c8 : 45 a4 55 4e 54 49 4c 20 7f
 87d0 : 57 49 54 48 4f 55 54 20 1b
 87d8 : 52 45 50 45 41 d4 00 00 44
 87e0 : a9 ef a2 87 8d 00 03 8e 1c
 87e8 : 01 03 20 30 98 a2 90 4c b2
 87f0 : 8b e3 ea 20 8a ad 20 f7 b2
 87f8 : b7 a5 14 8d 3f 03 a5 15 05
 8800 : 8d 40 03 20 fd ae 20 eb 20
 8808 : b7 a5 14 8d 41 03 a5 15 36
 8810 : 8d 42 03 8e 3e 03 ad 3f 82
 8818 : 03 85 14 ad 40 03 85 15 f5
 8820 : 20 13 a6 b0 05 a2 11 4c cc
 8828 : 3a a4 a5 5f 85 f7 a5 60 79
 8830 : 85 f8 20 8e a6 a9 00 8d de
 8838 : 43 03 20 3a 8a aa 20 3a 3f
 8840 : 8a d0 06 8a d0 03 4c 42 00
 8848 : 8a 20 3a 8a 20 3a 8a 20 e0
 8850 : 3a 8a f0 e1 c9 3a f0 f7 6a
 8858 : c9 8f f0 20 c9 22 f0 28 eb
 8860 : c9 8f f0 2e c9 8d f0 4a 55
 8868 : c9 a7 f0 28 c9 91 f0 32 97
 8870 : c9 2c d0 db ad 43 03 f0 e2
 8878 : d6 4c b2 88 20 3a 8a f0 12
 8880 : b4 c9 3a d0 f7 4c 4f 88 f2
 8888 : 20 3a 8a f0 a8 c9 22 d0 89
 8890 : f7 4c 4f 88 20 3a 8a c9 24
 8898 : 30 90 b7 c9 3a b0 b3 4c c8
 88a0 : b5 88 20 3a 8a c9 8f 0f e8
 88a8 : 04 c9 8d d0 f5 a9 01 8d da
 88b0 : 43 03 20 3a 8a b8 a6 7a c2
 88b8 : 86 f9 a6 7b 86 fa 20 6b eb
 88c0 : a9 20 13 a6 b0 03 4c 25 b2
 88c8 : 88 a5 15 cd 40 03 90 09 92
 88d0 : d0 0d a5 14 cd 3f 03 b0 57
 88d8 : 06 20 3d 8a 4c 52 88 a5 54
 88e0 : f7 85 fb a5 f8 85 fc ad 58
 88e8 : 41 03 8d 3c 03 ad 42 03 42
 88f0 : 8d 3d 03 a0 02 b1 fb c5 1a
 88f8 : 14 d0 07 c8 b1 fb c5 15 8b
 8900 : f0 1e a0 00 b1 fb aa c8 5f
 8908 : b1 fb 85 fc 86 fb ad 3e 33
 8910 : 03 18 6d 3c 03 8d 3c 03 96
 8918 : 90 d9 ee 3d 03 4c f3 88 6c
 8920 : ad 3d 03 ac 3c 03 20 95 4a
 8928 : b3 20 dd bd a2 ff e8 bd 64
 8930 : 01 01 d0 fa 86 fd a0 ff 20
 8938 : c8 b1 f9 c9 20 f0 f9 c9 95
 8940 : 30 90 04 c9 3a 90 f1 98 14
 8948 : 85 fe c5 fd f0 05 90 2b 4d
 8950 : 4c dd 89 a5 f9 85 7a a5 a3
 8958 : fa 85 7b a0 00 a2 01 20 61
 8960 : 3d 8a bd 00 01 f0 09 91 31
 8968 : 7a e8 20 3a 8a 4c 62 89 4d
 8970 : a5 f9 85 7a a5 fa 85 7b 02
 8978 : 4c 4f 88 a5 2d 8d a7 89 34
 8980 : 8d aa 89 a6 2e 8e a8 89 a7
 8988 : 8e ab 89 a5 fd 38 e5 fe 3a
 8990 : 18 6d aa 89 8d aa 89 90 b0
 8998 : 03 ee ab 89 ad aa 89 85 90

89a0 : 2d ad ab 89 85 2e ad 06 4c
 89a8 : 08 8d 07 08 ad a7 89 c5 03
 89b0 : f9 d0 07 ad a8 89 c5 fa 6d
 89b8 : f0 1d ce a7 89 ad 07 89 77
 89c0 : c9 ff d0 03 ce a8 89 ce 13
 89c8 : aa 89 ad aa 89 c9 ff d0 80
 89d0 : d5 ce ab 89 4c a6 89 20 89
 89d8 : 33 a5 4c 53 89 a5 f9 8d 24
 89e0 : 0a 8a 8d 0d 8a a5 fa 8d 11
 89e8 : 0b 8a 8d 0e 8a a5 fd 18 5b
 89f0 : 6d 0d 8a 8d 0d 8a 90 03 a6
 89f8 : ee 0e 8a a5 fe 18 6d 0a bf
 8a00 : 8a 8d 0a 8a 90 03 ee 0b 18
 8a08 : 8a ad 40 08 8d 3f 08 ad c8
 8a10 : 0a 8a c5 2d f0 07 ad 0b 88
 8a18 : 8a c5 2e f0 13 ee 0a 8a 14
 8a20 : d0 03 ee 0b 8a ee 0d 8a f8
 8a28 : d0 df ee 0e 8a 4c 09 8a a9
 8a30 : 85 2e ad 0a 8a 85 2d 4c 9b
 8a38 : d7 89 20 73 00 a0 00 b1 b3
 8a40 : 7a 60 ad 41 03 8d 3c 03 11
 8a48 : ad 42 03 8d 3d 03 a0 03 fd
 8a50 : ad 3d 03 91 f7 88 ad 3c 82
 8a58 : 03 91 f7 88 b1 f7 aa 88 c9
 8a60 : b1 f7 a8 d0 04 8a d0 01 2b
 8a68 : 60 86 f8 84 f7 18 ad 3e 4d
 8a70 : 03 6d 3c 03 8d 3c 03 90 81
 8a78 : d5 ee 3d 03 4c 4e 8a 00 d5
 8a80 : ea ea ea ea ea ea ea ea 7f
 8a88 : ea ea ea ea ea ea ea ea 8f
 8a90 : ea ea ea ea ea ea ea ea 97
 8aa0 : ea ea 20 79 00 f0 7d c9 48
 8aa8 : 22 f0 11 a2 00 8e 03 c0 dd
 8ab0 : 9d 00 cf 20 73 00 0f 23 86
 8ab8 : e8 4c b0 8a a0 01 8c 03 8e
 8ac0 : c0 a2 ff b1 7a c9 22 f0 68
 8ac8 : 08 c8 e8 9d 00 cf 4c c3 59
 8ad0 : 8a 20 73 00 c9 22 d0 f9 2c
 8ad8 : 4c b3 8a e8 a9 00 9d 00 cf
 8ae0 : cf a0 00 b1 2b 8d 00 c0 d6
 8ae8 : c8 b1 2b 8d 01 c0 a5 2b 08
 8af0 : 85 5f a5 2c 85 60 4c 70 81
 8af8 : 8b ad 03 c8 8d 02 c0 a0 60
 8b00 : 00 b1 5f 8d 00 c0 c8 b1 ef
 8b08 : 5f 8d 01 c0 ad 00 c0 85 6f
 8b10 : 5f ad 01 c0 85 60 20 d0 1c
 8b18 : f8 a0 00 b1 5f d0 06 c8 bd
 8b20 : b1 5f d0 01 60 a0 03 a2 31
 8b28 : 00 c8 b1 5f f0 cb c9 22 bd
 8b30 : d0 08 a9 01 4d 02 c0 8d 92
 8b38 : 02 c0 ad 02 c0 d0 e8 b1 e0
 8b40 : 5f dd 00 cf d0 e1 e8 bd c3
 8b48 : 00 cf f0 03 4c 29 8b a0 4a
 8b50 : 00 84 0f c8 c8 b1 5f 85 12
 8b58 : 14 85 63 c8 b1 5f 85 15 77
 8b60 : 85 62 a9 0d 20 d2 ff 20 fb
 8b68 : 13 a6 20 84 8b 4c f9 8a 7f
 8b70 : ad 03 c0 8d 02 c0 4c 16 04
 8b78 : 8b ea 4c 08 af 00 00 00 87
 8b80 : 00 00 00 00 20 d1 bd a0 49
 8b88 : 03 a9 20 84 49 20 d2 ff d9
 8b90 : a4 49 b1 5f c9 22 d0 06 2e
 8b98 : a5 0f 49 ff 85 0f c8 84 14
 8ba0 : 49 b1 5f d0 01 60 c9 ff ee
 8ba8 : f0 e3 24 0f 30 df a5 49 a0
 8bb0 : 38 e9 01 18 65 5f 8d 05 b2
 8bb8 : c0 b0 8b a5 60 8d 06 c0 93
 8bc0 : 4c cb 08 a5 60 18 69 01 f8
 8bc8 : 8d 06 c0 ad 05 c0 cd 00 cc
 8bd0 : c0 d0 0d ad 06 c0 cd 01 91
 8bd8 : c0 d0 05 a9 12 20 d2 ff e4
 8be0 : a4 49 b1 5f 10 26 38 e9 68
 8be8 : 7f aa a0 ff ca f0 08 c8 8a
 8bf0 : b9 9e a0 10 fa 30 f5 c8 bd
 8bf8 : b9 9e a0 30 05 20 d2 ff cb
 8c00 : d0 f5 29 7f 20 d2 ff a9 f1
 8c08 : 92 4c 8d 8b a5 0f f0 05 36
 8c10 : b1 5f 4c 04 8c b1 5f 4c 71
 8c18 : 8d 8b a9 25 a2 8c 8d 08 4f
 8c20 : 03 8e 09 03 60 08 48 8a 8a
 8c28 : 48 98 48 a5 7b c9 08 30 0a
 8c30 : 03 4c 3d 8c 68 a8 68 aa fd
 8c38 : 68 28 4c 4e 82 a0 00 a9 12
 8c40 : 01 ea 99 23 08 c8 c0 05 62
 8c48 : d0 f8 a0 00 a9 01 ea 99 3e
 8c50 : 4b d8 c8 c0 05 d0 f8 a0 4d
 8c58 : 00 a9 01 ea 99 73 d8 c8 f5
 8c60 : c0 05 d0 f8 a0 00 a9 01 a9
 8c68 : ea 99 9b d8 c8 c0 05 d0 69
 8c70 : f8 a0 00 a9 01 ea 99 c3 43
 8c78 : d8 c8 c0 05 d0 f8 a0 00 cd
 8c80 : a9 20 2c 23 04 c8 c0 05 3c
 8c88 : d0 f8 a0 00 a9 20 2c 4b df
 8c90 : 04 c8 c0 05 d0 f8 a0 00 20
 8c98 : a9 20 2c 73 04 c8 c0 05 5e
 8ca0 : d0 f8 a0 00 a9 20 2c 9b 98
 8ca8 : 04 c8 c0 05 d0 f8 a0 00 38
 8cb0 : a9 20 2c c3 04 c8 c0 05 80

8cb8 : d0 f8 a5 39 a4 3a 84 62 88
 8cc0 : 85 63 a2 90 38 20 49 bc d5
 8cc8 : 20 dd bd a0 00 b9 4b 04 5d
 8cd0 : 99 23 04 c8 c0 05 d0 f5 78
 8cd8 : a0 00 b9 73 04 99 4b 04 97
 8ce0 : c8 c0 05 d0 f5 a0 00 b9 3c
 8ce8 : 9b 04 99 73 04 c8 c0 05 ee
 8cf0 : d0 f5 a0 00 b9 c3 04 99 e0
 8cf8 : 9b 04 c8 c0 05 d0 f5 a0 cf
 8d00 : 00 b9 01 01 f0 12 99 c3 cb
 8d08 : 04 c8 c0 05 d0 f3 20 aa c4
 8d10 : 8d 29 01 f0 14 4c 34 8c 1e
 8d18 : c0 05 f0 f2 a9 20 99 c3 7f
 8d20 : 04 c8 c0 05 d0 f8 4c 0e 7b
 8d28 : 8d ad 8d 02 29 02 f0 de 5a
 8d30 : 20 70 86 4c 34 8c a9 04 04
 8d38 : 85 ba a9 7e 85 b8 a9 00 19
 8d40 : a0 04 85 71 84 72 85 b7 d3
 8d48 : 85 b9 20 c0 ff a6 b8 20 22
 8d50 : c9 ff a2 19 a9 d0 20 d2 0e
 8d58 : ff 20 e1 ff f0 2e a0 00 e3
 8d60 : b1 71 85 67 29 3f 06 67 8c
 8d68 : 24 67 10 02 09 80 70 d2 df
 8d70 : 09 40 20 d2 ff c8 c0 28 95
 8d78 : d0 e6 98 18 65 71 85 71 bf
 8d80 : 90 02 e6 72 ca d0 c8 a9 d7
 8d88 : 0d 20 d2 ff 20 cc ff a9 16
 8d90 : 7e 4c c3 ff 00 00 00 00 25
 8d98 : 00 00 00 00 00 00 00 00 9f
 8da0 : 20 1a 8c 20 9e b7 8e ff d6
 8da8 : c4 60 ae ff c4 e6 8c d0 9f
 8db0 : fc e8 d0 f9 ad 8d 02 60 a4
 8db8 : a0 00 b1 7a c9 22 d0 39 77
 8dc0 : c8 b1 7a f0 34 a0 01 b1 cd
 8dc8 : 7a f0 08 99 01 c5 c8 c0 d2
 8dd0 : 11 d0 f4 a9 53 8d 00 c5 e9
 8dd8 : a9 3a 8d 01 c5 98 c8 a8 b7
 8de0 : a2 00 a0 c5 20 bd ff a9 a6
 8de8 : 0f a2 08 a0 0f 20 ba ff 3b
 8df0 : 20 c0 ff 20 e7 ff 4c 1d 5e
 8df8 : 8e a0 8e a9 03 20 1e ab b0
 8e00 : 4c 74 a4 0d 3f 4d 49 53 7b
 8e08 : 53 49 4e 47 20 46 49 4c 6e
 8e10 : 45 20 4e 41 4d 45 20 45 2b
 8e18 : 52 52 4f 52 00 20 e6 84 57
 8e20 : 4c 74 a4 a9 0f a2 08 a0 6c
 8e28 : 0f 20 ba ff a9 01 a2 11 45
 8e30 : a0 80 20 bd ff 20 c0 ff d4
 8e38 : 20 2f f3 60 a9 0f a2 08 a6
 8e40 : a0 0f 20 ba ff a9 01 a2 5e
 8e48 : 10 4c 30 8e 00 a0 00 b9 d5
 8e50 : 09 80 f0 10 29 80 f0 03 38
 8e58 : 4c 65 8e b9 09 80 20 d2 ec
 8e60 : ff c8 0b e8 b0 b9 09 80 6e
 8e68 : 38 e9 80 20 d2 ff 8c 00 18
 8e70 : cf a9 20 20 d2 ff a9 20 34
 8e78 : 20 d2 ff ac 00 cf 4c 61 09
 8e80 : 8e 4c b4 8e 8e 00 cf a9 af
 8e88 : 3b 8d 11 d0 a9 08 8d 18 29
 8e90 : d0 a9 00 8d 00 dd a9 0e 3e
 8e98 : 85 04 a9 00 85 03 a8 91 c0
 8ea0 : 03 c8 d0 fb e6 04 a9 00 f0
 8ea8 : c5 04 f0 05 a9 00 4c 9f 57
 8eb0 : 8e 4c cb 8e 20 9e b7 8e 1c
 8eb8 : 40 03 20 9b b7 8a 0a 0a 01
 8ec0 : 0a 0a 0d 40 03 8d 40 03 be
 8ec8 : 4c 87 8e ad 40 03 a0 00 d0
 8ed0 : 99 00 c0 99 00 c1 99 00 41
 8ed8 : c2 99 00 c3 c8 d0 f1 60 7b
 8ee0 : a9 03 8d 00 dd a9 15 8d 09
 8ee8 : 18 d0 a9 1b 8d 11 d0 60 9b
 8ef0 : a9 80 2c a9 00 85 97 20 e4
 8ef8 : eb b7 78 a0 34 84 01 e0 1e
 8f00 : c8 b0 67 a5 15 c9 01 90 74
 8f08 : 08 d0 5f a5 14 c9 40 b0 f7
 8f10 : 59 8a 4a 4a 4a 0a a8 b9 95
 8f18 : 70 8f 8d 02 c5 b9 71 8f 02
 8f20 : 8d 03 c5 8a 29 07 18 6d f8
 8f28 : 02 c5 8d 02 c5 a5 14 29 dd
 8f30 : f8 8d 00 c5 18 a9 00 6d 51
 8f38 : 02 c5 85 fd a9 e0 6d 03 9b
 8f40 : c5 85 fe 18 a5 fd 6d 00 8a
 8f48 : c5 85 fd a5 fe 65 15 85 7e
 8f50 : fe a5 14 29 07 a9 07 aa 77
 8f58 : bd a2 8f a0 00 24 97 10 fe
 8f60 : 05 49 ff 31 fd 2c 11 fd b1
 8f68 : 91 fd a2 37 86 01 58 60 1a
 8f70 : 00 00 40 01 80 02 c0 03 c2
 8f78 : 00 05 40 06 80 07 c0 08 1f
 8f80 : 00 0a 40 0b 80 0c c0 0d 7c
 8f88 : 00 0f 40 10 80 11 c0 12 da
 8f90 : 00 14 40 15 80 16 c0 17 37
 8f98 : 00 19 40 1a 80 1b c0 1c 94
 8fa0 : 00 1e 01 02 04 08 10 20 31
 8fa8 : 40 80 20 9e b7 e0 01 f0 6d
 8fb0 : 39 20 fd ae 20 d4 e1 a9 d2
 8fb8 : 01 85 b9 a9 00 20 a5 ff 78
 8fc0 : a9 3e 78 85 01 a9 a0 85 42
 8fc8 : 04 a0 00 84 03 84 05 a9 69

8fd0 : e0 85 06 b1 03 91 05 c8 8d
8fd8 : d0 f9 e6 04 e6 06 a9 c0 a6
8fe0 : c5 04 d0 ef a7 37 85 01 46
8fe8 : 58 60 a9 34 78 85 01 a9 6c
8ff0 : a0 85 04 a0 00 84 03 84 a1
8ff8 : 05 a9 e0 85 06 b1 05 91 e0
9000 : 03 c8 d0 f9 e6 04 e6 06 11
9008 : a9 c0 c5 04 d0 ef a9 37 a5
9010 : 85 01 20 fd ae 20 d4 e1 e1
9018 : a9 3e 85 01 a2 00 a0 c0 90
9020 : a9 00 85 fd a9 a0 85 fe 9e
9028 : a9 fd 85 b9 20 d8 ff a9 84
9030 : 37 85 01 58 60 a9 00 85 d4
9038 : 57 85 58 a0 08 18 26 5b 0d
9040 : 90 0d 18 a5 57 65 59 85 23
9048 : 57 a5 58 65 5a 85 58 88 79
9050 : f0 06 26 57 26 58 90 e6 ed
9058 : 60 a9 00 85 c5 85 5d a5 f0
9060 : 59 d0 07 a5 5a d0 03 4c 69
9068 : 8a bb a5 58 d0 04 a5 57 b7
9070 : f0 17 38 a5 57 e5 59 85 c4
9078 : 57 a5 58 e5 5a 85 58 90 c9
9080 : 08 e6 5c d0 e5 e6 5d d0 d9
9088 : e1 60 20 eb b7 8e 10 c5 db
9090 : a5 14 8d 11 c5 a5 15 8d be
9098 : 12 c5 20 fd ae 20 eb b7 60
90a0 : 8e 13 c5 a5 14 8d 14 c5 67
90a8 : a5 15 8d 15 c5 ae 14 c5 8b
90b0 : ad 15 c5 cd 12 c5 f0 02 2a
90b8 : b0 2b ec 11 c5 f0 6a b0 4a
90c0 : 24 ad 11 c5 ae 14 c5 8d 76
90c8 : 14 c5 8e 11 c5 ad 12 c5 22
90d0 : ae 15 c5 8d 15 c5 8e 12 0a
90d8 : c5 ad 10 c5 ae 13 c5 8d e6
90e0 : 13 c5 8e 10 c5 38 ad 14 78
90e8 : c5 ed 11 c5 8d 16 c5 ad 9d
90f0 : 15 c5 ed 12 c5 8d 17 c5 56
90f8 : a2 00 8e 18 c5 38 ad 13 3c
9100 : c5 ed 10 c5 f0 4f ee 18 ee
9108 : c5 8d 19 c5 b0 0b 49 ff 1b
9110 : 69 01 ca 8e 18 c5 8d 19 96
9118 : c5 ad 17 c5 d0 62 ad 16 35
9120 : c5 cd 19 c5 b0 5a 4c e1 9e
9128 : 91 ad 13 c5 cd 10 c5 b0 e3
9130 : 09 ae 10 c5 8e 13 c5 8d 01
9138 : 10 c5 ae 10 c5 8a 48 ad 06
9140 : 11 c5 85 14 ad 12 c5 85 a5
9148 : 15 20 fa 8e 68 aa e8 ee 57
9150 : 13 c5 90 e9 60 ad 11 c5 ea
9158 : 85 14 ad 12 c5 85 15 ae cf
9160 : 13 c5 20 fa 8e ee 11 c5 ed
9168 : d0 03 ee 12 c5 ad 15 c5 61
9170 : cd 12 c5 90 0a d0 de ad c8
9178 : 14 c5 cd 11 c5 b0 6d 60 02
9180 : a9 00 8d 1a c5 8d 1b c5 91
9188 : 18 ad 1a c5 85 59 6d 11 b1
9190 : c5 85 14 ad 1b c5 85 5a 7d
9198 : 6d 12 c5 85 15 ad 19 c5 df
91a0 : 85 5b 20 35 90 ad 16 c5 dc
91a8 : 85 59 ad 17 c5 85 5a 20 5a
91b0 : 59 90 ad 10 c5 2c 18 c5 68
91b8 : 30 05 18 65 c5 9c 03 38 e4
91c0 : e5 5c aa 20 fa 8e ee 1a 96
91c8 : c5 d0 03 ee 1b c5 ad 17 59
91d0 : c5 cd 1b c5 90 0a d0 b0 f9
91d8 : ad 16 c5 cd 1a c5 b0 a8 9f
91e0 : 60 a9 00 8d 1c c5 85 5b 83
91e8 : ad 10 c5 2c 18 c5 30 06 11
91f0 : 18 6d 1c c5 90 0a d0 ed 64
91f8 : 1c c5 48 ad 16 c5 85 59 17
9200 : ad 17 c5 85 5a 20 35 90 f7
9208 : ad 19 c5 85 59 a9 00 85 52
9210 : 5a 20 59 90 18 a5 5c 6d de
9218 : 11 c5 85 14 a5 5d 6d 12 0f
9220 : c5 85 15 68 aa 20 fa 8e af
9228 : ee 1c c5 ad 1c c5 cd 19 a5
9230 : c5 f0 b3 90 b1 60 20 eb e3
9238 : b7 8e 20 c5 a5 14 8d 22 6d
9240 : c5 a5 15 8d 23 c5 20 fd ac
9248 : ae 20 eb b7 8e 21 c5 a5 4c
9250 : 14 8d 24 c5 a5 15 8d 25 70
9258 : c5 ad 22 c5 8d 11 c5 ad 09
9260 : 23 c5 8d 12 c5 ad 20 c5 e1
9268 : 8d 10 c5 ad 24 c5 8d 14 f3
9270 : c5 ad 25 c5 8d 15 c5 ad 02
9278 : 20 c5 8d 13 c5 20 ad 90 76
9280 : ad 22 c5 8d 11 c5 ad 23 9e
9288 : c5 8d 12 c5 ad 21 c5 8d 67
9290 : 10 c5 ad 24 c5 8d 14 c5 17
9298 : ad 25 c5 8d 15 c5 ad 21 73
92a0 : c5 8d 13 c5 20 ad 90 ad b6
92a8 : 22 c5 8d 11 c5 ad 23 c5 14
92b0 : 8d 12 c5 ad 20 c5 8d 10 f4
92b8 : c5 ad 22 c5 8d 14 c5 ad 81
92c0 : 23 c5 8d 15 c5 ad 21 c5 a6
92c8 : 8d 13 c5 20 ad 90 ad 24 b3
92d0 : c5 8d 11 c5 ad 25 c5 8d 8f
92d8 : 12 c5 ad 20 c5 8d 10 c5 d1
92e0 : ad 24 c5 8d 14 c5 ad 25 33

92e8 : c5 8d 15 c5 ad 21 c5 8d 88
92f0 : 13 c5 20 ad 90 60 20 9e 6d
92f8 : b7 e0 01 d0 05 a9 00 85 22
9300 : 97 60 a9 80 85 97 60 20 19
9308 : eb b7 8e 20 c5 a5 14 8d 6b
9310 : 22 c5 a5 15 8d 23 c5 20 6a
9318 : fd ae 20 eb b7 8e 21 c5 f2
9320 : a5 14 8d 24 c5 a5 15 8d b0
9328 : 25 c5 ad 21 c5 cd 20 c5 96
9330 : b0 09 ae 20 c5 8e 21 c5 f5
9338 : 8d 20 c5 ae 20 c5 8e 26 d3
9340 : c5 ad 22 c5 8d 11 c5 ad f1
9348 : 23 c5 8d 12 c5 ad 26 c5 e1
9350 : 8d 10 c5 8d 13 c5 ad 24 67
9358 : c5 8d 14 c5 ad 25 c5 8d d8
9360 : 15 c5 20 ad 90 ee 26 c5 ba
9368 : ac 21 c5 cc 26 c5 b0 d1 a7
9370 : 60 20 eb b7 8e 10 c5 a5 9e
9378 : 14 8d 11 c5 a5 15 8d 12 ad
9380 : c5 20 fd ae 20 8a ad a0 f9
9388 : c5 a2 30 20 d4 bb 20 fd 56
9390 : ae 20 8a ad a0 c5 a2 40 ea
9398 : 20 d4 bb 20 fd ae 20 e8 bd
93a0 : 74 a0 c5 a2 50 20 d4 bb 1b
93a8 : 20 6b e2 a9 30 a0 c5 20 cb
93b0 : 28 ba a9 11 a0 bf 20 67 19
93b8 : b8 20 9b bc 18 a5 65 6d 1e
93c0 : 11 c5 8d 14 c5 a5 64 6d 90
93c8 : 12 c5 8d 15 c5 a9 50 a0 ef
93d0 : c5 20 a2 bb 20 64 e2 a9 c9
93d8 : 40 a0 c5 20 28 ba a9 11 ff
93e0 : a0 bf 20 67 b8 20 9b bc c9
93e8 : 18 a5 65 6d 10 c5 8d 13 65
93f0 : c5 20 ad 90 60 a9 e0 85 25
93f8 : 04 a9 00 85 03 a8 a9 00 9e
9400 : 91 03 c8 d0 fb e6 04 a9 b9
9408 : 00 c5 04 d0 f1 60 a2 34 1b
9410 : 78 86 01 a9 00 85 03 a9 cc
9418 : e0 85 04 a0 00 b1 03 a9 fc
9420 : ff 91 03 c8 d0 f7 e6 04 32
9428 : a9 00 c5 04 d0 ef a9 37 65
9430 : 85 01 58 60 20 eb b7 8e b5
9438 : 62 c5 a5 14 8d 60 c5 a5 a7
9440 : 15 8d 61 c5 20 fd ae 20 1a
9448 : eb b7 8e 65 c5 a5 14 8d 54
9450 : 63 c5 a5 15 8d 64 c5 20 f5
9458 : fd ae 20 eb b7 8e 68 c5 4f
9460 : a5 14 8d 64 c5 a5 8d 38
9468 : 67 c5 ad 60 c5 8d 11 c5 c2
9470 : ad 61 c5 8d 12 c5 ad 62 bc
9478 : c5 8d 10 c5 ad 63 c5 8d e9
9480 : 14 c5 ad 64 c5 8d 15 c5 17
9488 : ad 65 c5 8d 13 c5 20 ad 46
9490 : 90 ad 60 c5 8d 11 c5 ad 9b
9498 : 61 c5 8d 12 c5 ad 62 c5 60
94a0 : 8d 10 c5 ad 66 c5 8d 14 4f
94a8 : c5 ad 67 c5 8d 15 c5 ad ca
94b0 : 68 c5 8d 13 c5 20 ad 90 f6
94b8 : ad 63 c5 8d 11 c5 ad 64 f9
94c0 : c5 8d 12 c5 ad 65 c5 8d c1
94c8 : 10 c5 ad 66 c5 8d 14 c5 98
94d0 : ad 67 c5 8d 15 c5 ad 68 5b
94d8 : c5 8d 13 c5 20 ad 90 60 54
94e0 : 86 e5 2e e0 d1 00 00 00 1e
94e8 : 20 8a ad 20 0c bc a9 e0 cc
94f0 : a0 94 20 a2 bb a5 61 20 e5
94f8 : 12 bb 60 82 c9 0f da a1 14
9500 : 49 00 00 82 c9 0f da a1 5d
9508 : b6 00 00 20 8a ad a0 c5 e7
9510 : a2 70 20 d4 bb 20 fd ae 9f
9518 : 20 8a ad a0 c5 a2 80 20 b0
9520 : d4 bb 20 fd ae 20 8a ad 0b
9528 : a0 c5 a2 90 20 d4 bb 20 3d
9530 : fd ae 20 8a ad a0 c5 a2 1a
9538 : a0 20 d4 bb a0 c5 a9 70 55
9540 : 20 a2 bb a0 c5 a9 80 20 a0
9548 : 67 b8 20 0c bc a0 95 a9 0f
9550 : 0b 20 a2 bb a5 61 20 12 95
9558 : bb a0 94 a9 fb 20 28 ba 94
9560 : a0 95 a9 0b 20 28 ba a0 06
9568 : 94 a9 fb 20 8c ba a5 61 cc
9570 : 20 12 bb a0 c5 a2 b0 20 11
9578 : d4 bb a0 95 a9 03 20 a2 7d
9580 : bb a0 c5 a2 c0 20 d4 bb 29
9588 : a0 c5 a9 c0 20 a2 bb 20 d3
9590 : 64 e2 a0 c5 a9 80 20 28 b6
9598 : ba e0 c5 a9 a0 20 67 b8 63
95a0 : 20 9b bc a5 65 48 a0 c5 18
95a8 : a9 c0 20 a2 bb 20 6b e2 3e
95b0 : a0 c5 a9 70 20 28 ba a0 1b
95b8 : c5 a9 90 20 67 b8 20 9b 6e
95c0 : bc a5 65 85 14 a5 64 85 64
95c8 : 15 68 aa 20 07 96 a0 c5 f3
95d0 : a9 b0 20 a2 bb a0 c5 a9 59
95d8 : c0 20 67 b8 a0 c5 a2 c0 dd
95e0 : 20 d4 bb a0 c5 a9 c0 20 5a
95e8 : a2 bb a0 96 a9 00 20 5b 35
95f0 : bc 30 03 4c 88 95 60 00 c5
95f8 : 00 00 00 00 00 00 00 00 f9

9600 : 83 c9 0f da a1 b6 00 8e 74
9608 : d0 c5 a5 14 8d d1 c5 a5 70
9610 : 15 8d d1 c5 20 fa 8e ad c9
9618 : d2 c5 85 62 ad d1 c5 85 06
9620 : 63 a2 90 20 44 bc 20 0c bf
9628 : bc a0 c5 a9 90 20 a2 bb e7
9630 : a5 61 20 53 b8 a0 c5 a9 f3
9638 : 90 20 8c ba a5 61 20 53 df
9640 : b8 20 9b bc a5 65 85 14 4a
9648 : a5 64 85 15 ae d0 c5 20 9c
9650 : fa 8e 60 20 c9 97 e0 09 ec
9658 : b0 8e 43 03 a9 00 ca a3
9660 : f0 06 18 69 02 ca d0 fb 38
9668 : 85 fd a9 d0 85 fe 20 fd 3d
9670 : ae 20 eb b7 8e 42 03 a5 72
9678 : 15 c9 01 90 27 a5 14 c9 48
9680 : 59 b0 cf a0 01 ad 42 03 c6
9688 : 91 fd a5 15 c9 01 d0 32 70
9690 : ac 43 03 b9 c1 96 98 d0 23
9698 : 10 d0 8d 10 d0 a0 00 a5 d3
96a0 : 14 91 fd 60 ac 43 03 b9 6d
96a8 : c1 96 98 a9 ff 2d 10 d0 4f
96b0 : 8d 10 d0 a0 00 a5 14 91 2e
96b8 : fd a0 01 ad 42 03 91 fd 7a
96c0 : 60 01 02 04 08 10 20 40 a4
96c8 : 80 00 20 9e b7 c9 08 b0 6f
96d0 : 01 60 8e e0 c5 20 fd ae 74
96d8 : 20 9e b7 e0 02 b0 f2 8e e0
96e0 : e1 c5 20 fd ae 20 9e b7 41
96e8 : e0 04 b0 e5 8e e2 c5 20 0a
96f0 : fd ae 20 9e b7 e0 10 b0 44
96f8 : d8 8e e3 c5 20 fd ae 20 b6
9700 : 9e b7 e0 02 b0 cb 20 8f 5b
9708 : 97 ad e1 c5 f0 26 ac e0 5c
9710 : c5 b9 c1 96 0d 1c d0 8d 05
9718 : 1c d0 20 fd ae 20 9e b7 3a
9720 : e0 10 b0 ad 8e 25 d0 20 80
9728 : fd ae 20 9e b7 e0 10 b0 7c
9730 : a0 8e 26 d0 ad e2 c5 f0 ac
9738 : 3b c9 03 f0 07 c9 02 f0 df
9740 : 18 4c 68 97 ac e0 c5 b9 e9
9748 : c1 96 0d 17 d0 8d 17 d0 f2
9750 : 0d 1d d0 8d 1d d0 4c 74 54
9758 : 97 ac e0 5c b9 c1 96 0d 44
9760 : 17 d0 8d 17 d0 4c 74 97 96
9768 : ac e0 5c b9 c1 96 0d 1d 6c
9770 : d0 8d 1d d0 ad 83 c5 ac 3d
9778 : e0 c5 99 27 d0 ad e4 c5 20
9780 : f0 0c a9 e0 c5 b9 c1 96 1c
9788 : 0d 1b d0 8d 1b d0 60 8e df
9790 : e4 c5 ac e0 c5 b9 c1 96 4f
9798 : 49 ff 48 2d 1c d0 8d 1c 4c
97a0 : d0 68 48 2d 25 d0 8d 25 b5
97a8 : d0 68 48 2d 26 d0 8d 26 d0
97b0 : d0 68 48 2d 17 d0 8d 17 c8
97b8 : d0 68 48 2d 1d d0 8d 1d 3d
97c0 : d0 68 2d 1b d0 8d 1b d0 fd
97c8 : 60 20 9e b7 e0 60 20 9e 26
97d0 : b7 e0 08 b7 e0 8e 42 03 22
97d8 : 20 fd ae 20 9e b7 8a ae d6
97e0 : 42 03 9d f8 07 60 20 9e 5b
97e8 : b7 e0 08 b7 e0 8e 43 03 3f
97f0 : 20 fd ae 20 9e b7 e0 02 ee
97f8 : b0 eb 8e 42 03 ae 42 03 3e
9800 : e0 01 f0 0f ac 43 03 b9 e3
9808 : c1 96 49 ff 2d 15 d0 8d 40
9810 : 15 d0 60 ac 43 03 b9 c1 f2
9818 : 96 0d 15 d0 8d 15 d0 60 1a
9820 : 20 9e b7 e0 20 d0 fd 5a
9828 : ae 20 9e b7 8e 21 d0 60 7b
9830 : a2 3b a0 98 8e 06 03 8c e9
9838 : 07 03 60 08 48 8a 48 98 05
9840 : 48 ad 8d 02 29 01 d0 10 00
9848 : ad 8d 02 29 02 d0 13 68 25
9850 : a8 68 aa 68 28 4c 19 82 32
9858 : ad 8d 02 29 01 d0 f9 4c 89
9860 : 4f 98 ad 8d 02 29 02 f0 6c
9868 : e6 a2 e0 e6 8c d0 fc e8 b9
9870 : d0 f9 4c 4f 98 00 00 00 c3
9878 : 00 ff 00 00 ff 00 00 00 78
9880 : ff ff 00 00 ff ff 00 00 7f
9888 : ff ff 00 00 ff ff 00 00 87
9890 : ff ff 00 00 ff ff 00 00 8f
9898 : ff ff 00 00 ff ff 00 00 97
98a0 : ff ff 00 00 ff ff 00 00 9f
98a8 : ff ff 00 00 ff ff 00 00 a7
98b0 : ff ff 00 00 ff ff 00 00 af
98b8 : ff ff 00 00 ff ff 00 00 b7
98c0 : ff ff 00 00 ff ff 00 00 bf
98c8 : ff ff 00 00 ff ff 00 00 c7
98d0 : ff ff 00 00 ff ff 00 00 cf
98d8 : ff ff 00 00 ff ff 00 00 d7
98e0 : ff ff 00 00 ff ff 00 00 df
98e8 : ff ff 00 00 ff ff 00 00 ef
98f0 : ff ff 00 00 ff ff 00 00 ef
98f8 : ff ff 00 00 ff ff 00 00 f7
9900 : 00 ff ff ff ff ff ff ff 00

Listing »Basic-Erweiterung« (Schluß)

PRINT USING mit derUSR-Funktion

Haben Sie sich auch schon über die unformatierte Zahlenausgabe Ihres C 64 geärgert? Dann nehmen Sie in Zukunft die folgende USR-Funktion, um für Ihren C 64 ein PRINT USING zu erhalten.

Wie allgemein bekannt und vielfach bemängelt, bietet das Commodore-Basic keinen PRINT USING-Befehl. Doch gerade bei kommerziellen Problemen kann auf eine Druckaufbereitung von Zahlen nicht verzichtet werden. Eine Rechnung, in der nicht einmal die Dezimalpunkte untereinander stehen, ist eben unübersichtlich und keine Reklame. Für die Druckaufbereitung gibt es verschiedene Lösungen.

Die beste Lösung ist ein Maschinenprogramm. Interessierte Maschinenprogrammierer finden den Quellcode zu dieser Routine in Listing 1. Doch zunächst stellt sich die Frage, wie ein Maschinenprogramm für die Druckaufbereitung aufgerufen werden soll. Offensichtlich ist die USR-Funktion am geeignetsten, da sie sowohl in einer PRINT- als auch in einer PRINT #- oder einer LET-Anweisung verwendet werden kann. Der Aufruf der Funktion erfolgt durch USR(X),L,NK. Dabei ist X die Zahl, die aufbereitet werden soll, L die Gesamtlänge der aufbereiteten Zahl einschließlich Vorzeichen und Dezimalkomma und NK die Anzahl der darzustellenden Nachkommastellen. Die USR-Funktion wandelt zunächst die Zahl X in einen ASCII-String um und berechnet die Stringlänge und die Anzahl der Nachkommastellen. Wenn bei der Wandlung von X der Interpreter die Exponentialform wählt, dann wird die Exponentialdarstellung zunächst in die Fließkommadarstellung umgewandelt. Danach werden die Nachkommastellen aufbereitet. Fehlende Nachkommastellen werden durch angehängte Nullen ergänzt. Müssen Nachkommastellen abgeschnitten werden, dann wird die Zahl gerundet, wenn die erste abgeschnittene Dezimalstelle größer oder gleich 5 ist. Wenn die Anzahl der gewünschten Nachkommastellen null ist, dann wird die Zahl als ganze Zahl (Integer) ohne Dezimalkomma aufbereitet. Wegen der kaufmännischen Anwendung wird der Dezimalpunkt durch ein Dezimalkomma ersetzt. Nach der Aufbereitung der Nachkommastellen wird durch Voranstellen von Leerzeichen der String auf die erforderliche Länge gebracht. Ist der String nach der Aufbereitung der Nachkommastellen schon länger als gewünscht, dann wird er nicht mehr verändert, sondern in voller Länge ausgegeben, um einen Datenverlust zu verhindern.

Das hier vorgestellte Maschinenprogramm verwendet nur relative Sprünge — außer bei den Aufrufen der Betriebssystemroutinen. Daher kann sich jeder Anwender das Programm ohne Änderungen in den Speicherbereich laden, der ihm am geeignetsten erscheint. Als Stringpuffer wird der Bereich ab \$100 benutzt. Das hat zur Folge, daß die GETSTR-Routine diesen String nicht in den Stringbereich kopiert. Der Stringbereich wird also nicht unnötig belastet. Eine Wertzuweisung A\$=USR(X),L,NK ist dadurch aber auch nicht möglich, da die nächste Stringfunktion den Bereich ab \$100 wieder über-

schreibt. Wenn druckaufbereitete Werte einer Variablen zugewiesen werden sollen, dann muß die Anweisung A\$=" "+USR(X),L,NK oder A\$=(USR(X),L,NK)+" " lauten, da dann das Ergebnis der Stringverknüpfung in den Stringbereich kopiert wird und der Variablen A\$ dauerhaft zugewiesen ist.

Vor dem ersten Aufruf der USR-Funktion muß jetzt noch in Adresse 785 (Low-Byte) und 786 (High-Byte) die Startadresse der USR-Funktion hinterlegt werden. Listing 2 zeigt das Basic-Ladeprogramm für die USR-Funktion. Die Ladeadresse können Sie selbst bestimmen. Das Ladeprogramm setzt die Startadresse der USR-Funktion in den Speicherstellen 785 und 786 entsprechend. Zur Verdeutlichung der Anwendung der USR-Funktion enthält das Ladeprogramm verschiedene Druckaufbereitungen der Zahl π . Das Ergebnis des Beispiels ist in Bild 1 wiedergegeben.

(Dr. Michael Irskens/ah)

| | | |
|------------|--------------|---------------|
| 3 | 3,1 | 3,14 |
| 31 | 31,4 | 31,42 |
| 314 | 314,2 | 314,16 |
| 3142 | 3141,6 | 3141,59 |
| 31416 | 31415,9 | 31415,93 |
| 314159 | 314159,3 | 314159,27 |
| 3141593 | 3141592,7 | 3141592,65 |
| 31415927 | 31415926,5 | 31415926,50 |
| 314159265 | 314159265,0 | 314159265,00 |
| 3141592650 | 3141592650,0 | 3141592650,00 |

Bild 1. Beispiele für verschiedene Druckaufbereitungen

```

1000 ; formatroutine
1010 ;
1020 ;
1030 ; dr.m.irskens
1040 ; leveser allee 13
1050 ; 3061 hespe
1060 ;
1070 ;
1080 space equ 32 ; ' '
1090 komma equ 44 ; ','
1100 e equ 69 ; 'e'
1110 punkt equ 46 ; '.'
1120 null equ 48 ; '0'
1130 minus equ 45 ; '-'
1140 ckcom equ $aeFd ; prueft auf komma
1150 getstr equ $b487 ; string aus stringpuffer in stringbereich
1160 numtest equ $ad8d ; ergebnis auf numerisch pruefen
1170 facstr equ $bddd ; fac in string wandeln
1180 getbyt equ $b79e ; byte holen
1190 string equ $100 ; string-puffer
1200 temp equ $02 ; temporaerer speicher
1210 nk equ $57 ; fließkomma-akku#3
1220 flen equ $58
1230 *equ $c000 ; startadresse
1240 fo10 jsr numtest ; auf numerisch pruefen
1250 jsr facstr ; umwandeln numerisch->alpha
1260 jsr ckcom ; auf komma pruefen
1270 jsr getbyt ; feldlaenge holen
1280 stx flen
1290 jsr ckcom ; auf komma pruefen
1300 jsr getbyt ; nachkommastellenanzahl holen
1310 stx nk
1320 pla ; rucksprungadresse vom stack entfernen,
1330 pla ; damit kein numtest durchgefuehrt wird
1340 ;
1350 ; pruefen auf darstellung im e-format
1360 ; x-register equ stringlaenge
1370 ; y-register equ anzahl vorkommastellen
1380 ; y-register equ 0, wenn keine nachkommastellen gefunden
1390 fo19 ldx #$ff
1400 ldy #0
1410 fo20 inx
1420 lda string,x ; zeichen aus puffer laden
1430 beq fo30 ; 0->stringende
1440 cmp #e ; vergleiche auf 'e'
1450 beq fo201 ; e-format umwandeln
1460 cmp #punkt ; vergleiche auf '.'
1470 bne fo20
1480 txa
1490 tay ; anzahl vorkommastellen in y-register
1500 bne fo20
1510 ;
1520 ; umwandeln e-format
1530 ;
1540 fo201 lda string+2
1550 cmp #punkt
1560 bne fo22
1570 ;
1580 ; dezimalpunkt entfernen
1590 ;
1600 dex
1610 ldy #1
1620 fo21 iny
1630 lda string+1,y
1640 sta string,y
1650 bne fo21
    
```

Listing 1. PRINT USING-Quellprogramm

```

1660 ;
1670 ; stellenverschiebung errechnen
1680 ;
1690 fo22 lda string+2,x ; zehnerstelle exponent
1700 and #0f ; ziffer errechnen
1710 asl a ; *2
1720 sta temp
1730 asl a ; *4
1740 asl a ; *8
1750 adc temp ; 8* + 2* equ 10*
1760 adc string+3,x ; teinerstelle ascii
1770 sbc #47 ; ascii 0 abziehen
1780 ldy string+1,x ; vorzeichen exponent
1790 cpy #minus
1800 beq fo24
1810 adc #3
1820 stx temp
1830 sbc temp
1840 tay
1850 lda #null
1860 fo23 sta string,x ; string mit nullen erweitern fuer + exponent
1870 inx
1880 dey
1890 bne fo23
1900 lda #0
1910 sta string,x ; neues stringende
1920 beq fo19
1930 ;
1940 ; exponent<0: vornullen ergaenzen
1950 ; string um a-reg.+1 stellen verschieben
1960 fo24 sta temp
1970 lda #0
1980 sta string,x ; "e" durch 0 ersetzen
1990 txa
2000 clc
2010 adc temp
2020 tay
2030 fo25 lda string,x
2040 beq fo26 ; stringende uebertragen
2050 cmp #30 ; vergleich auf ziffer
2060 bcs fo26 ; ziffer
2070 lda #null
2080 bne fo27
2090 dex
2100 fo26 sta string,y
2110 dey
2120 bne fo25
2130 lda #punkt
2140 sta string+1
2150 bne fo19
2160 ;
2170 ; dezimalpunkt durch komma ersetzen
2180 ; y-register: anzahl vorkommazeichen
2190 ;
2200 fo30 tya
2210 beq fo50 ; nachkommastellen ergaenzen
2220 lda nk
2230 bne fo45 ; nachkommastellen
2240 tya
2250 tax
2260 lda string+1,x
2270 bne fo91 ; keine nachkommastellen, aber runden
2280 fo45 lda #komma ; '.' gefunden
2290 sta string,y ; durch ',' ersetzen
2300 bne fo71
2310 fo50 cpy nk ; nachkommastellen ergaenzen
2320 beq fo100
2330 lda #komma
2340 sta string,x
2350 inx
2360 bne fo79
2370 fo71 sty temp ; anzahl nachkommastellen errechnen
2380 sec
2390 txa
2400 sbc temp
2410 sec
2420 sbc #1 ; 1 abziehen fuer dezimalpunkt
2430 cmp nk ; vergleichen mit sollanzahl
2440 beq fo100 ; anzahl gleich
2450 bcs fo90 ; anzahl > soll
2460 tay ; anzahl < soll

2470 fo79 lda #null
2480 fo80 sta string,x ; string mit nullen ergaenzen
2490 inx
2500 iny
2510 cpy nk
2520 bne fo80
2530 fo85 lda #0
2540 sta string,x
2550 ;
2560 ; nachkommastellen aufbereitet
2570 ; gesamtfeldlaenge korrigieren
2580 ;
2590 ; x-reg feldlaenge aktuell
2600 ; fien gesamtlange soll
2610 fo100 lda string+1
2620 cmp #30
2630 bcs fo109 ; 1.zeichen ist eine ziffer
2640 inx
2650 txa
2660 tay
2670 fo105 lda string-1,y
2680 sta string,y
2690 dey
2700 bne fo105
2710 lda #null ; null vor komma ergaenzen
2720 sta string+1
2730 fo109 cpx fien
2740 bcs fo130 ; feldueberlauf oder feldlaenge gleich
2750 ldy fien
2760 fo110 lda string,x ; string verschieben
2770 sta string,y
2780 dey
2790 dex
2800 bpl fo110
2810 lda #space ; leerzeichen
2820 fo120 sta string,y ; rest mit leerzeichen fuellen
2830 dey
2840 bpl fo120
2850 fo130 lda #<string ; startadresse string laden in a/y
2860 ldy #>string
2870 jmp getstr
2880 fo90 sec
2890 sbc nk
2900 sta temp ; anzahl ueberfluessiger nachkommastellen
2910 txa
2920 sec
2930 sbc temp ; neue stringlaenge
2940 tax
2950 y
2960 ; beim abschneiden von nachkommastellen runden
2970 ;
2980 lda string,x
2990 fo901 cmp #35
3000 bcc fo85
3010 txa
3020 tay ; index letztes zeichen
3030 fo91 dey
3040 beq fo92 ; string eine 1 voranstellen
3050 lda string,y
3060 cmp #30 ; vergl < "0"
3070 bcc fo91
3080 clc
3090 adc #1 ; ziffer um eins erhoehen
3100 cmp #3a ; ueberlauf abfragen
3110 sta string,y ; veraendertes zeichen speichern
3120 bne fo85 ; kein uebertrag
3130 lda #null ; null laden
3140 sta string,y
3150 bne fo91 ; naechste ziffer erhoehen
3160 fo92 txa ; string um eine stelle verschieben fuer "1"
3170 tay
3180 fo94 lda string,y
3190 sta string+1,y
3200 dey
3210 bne fo94
3220 lda #31 ; "1" laden
3230 sta string+1
3240 inx ; feldlaenge erhoehen
3250 bne fo85

```

Listing 1. PRINT USING-Quellprogramm (Schluß)

```

100 S=0: INPUT "STARTADRESSE";B <182>
110 FOR I=B TO B+335 <000>
120 READ A <160>
130 S=S+A:POKE I,A <062>
140 NEXT I <224>
150 IF S<>36986 THEN PRINT "{RVSON,SPACE}F
EHLER IN DEN DATAZEILEN{SPACE,RVOFF}";
STOP <119>
160 POKE 785,B-256*INT(B/256):POKE 786,B/2
56 <040>
170 PRINT " ALLES OK. " <036>
199 REM TESTBEISPIELE <050>
200 FOR E=0 TO 9 <238>
210 F=t*10↑E <176>
220 A$=(USR(F),11,0)+" <178>
230 PRINT A$;USR(F),13,1;USR(F),14,2 <024>
240 NEXT E <036>
250 END <252>
299 REM DATAZEILEN <190>
300 DATA 32,141,173,32,221,189,32,253,174,
32,158,183,134,88,32,253,174,32 <212>
301 DATA 158,183,134,87,104,104,162,255,16
0,0,232,189,0,1,240,117,201,69,240 <057>
302 DATA 8,201,46,208,242,138,168,208,238,
173,2,1,201,46,208,12,202,160,1 <250>
303 DATA 200,185,1,1,153,0,1,208,247,189,2
,1,41,15,10,133,2,10,10,101,2,125 <139>
304 DATA 3,1,233,47,188,1,1,192,45,240,23,
105,3,134,2,229,2,168,169,48,157 <106>
305 DATA 0,1,232,136,208,249,169,0,157,0,1
,240,168,133,2,169,0,157,0,1,138 <207>
306 DATA 24,101,2,168,189,0,1,240,8,201,48
,176,4,169,48,208,1,202,153,0,1 <048>
307 DATA 136,208,236,169,46,141,1,1,208,12
9,152,240,18,165,87,208,7,152,170 <211>
308 DATA 189,1,1,208,119,169,44,153,0,1,20
8,12,196,87,240,40,169,44,157,0 <159>
309 DATA 1,232,208,16,132,2,56,138,229,2,5
6,233,1,197,87,240,19,176,72,168 <011>
310 DATA 169,48,157,0,1,232,200,196,87,208
,247,169,0,157,0,1,173,1,1,201,48 <115>
311 DATA 176,17,232,138,168,185,255,0,153,
0,1,136,208,247,169,48,141,1,1,228 <091>
312 DATA 88,176,20,164,88,189,0,1,153,0,1,
136,202,16,246,169,32,153,0,1,136 <163>
313 DATA 16,250,169,0,160,1,76,135,180,56,
229,87,133,2,138,56,229,2,170,189 <206>
314 DATA 0,1,201,53,144,179,138,168,136,24
0,24,185,0,1,201,48,144,246,24,105 <086>
315 DATA 1,201,58,153,0,1,208,157,169,48,1
53,0,1,208,229,138,168,185,0,1,153 <028>
316 DATA 1,1,136,208,247,169,49,141,1,1,23
2,208,131 <030>
@ 64'er

```

Listing 2. PRINT USING-Basic-Lader. Bitte beachten Sie die Eingabehinweise auf Seite 6

Game-Basic

Mit dieser Befehlserweiterung lassen sich mit wenig Aufwand sehr effektvolle Spiele schreiben. Es wird dabei der Umgang mit Sprites und Sonderzeichen und die Bildschirmverwaltung (Scrolling) unterstützt.

Da ich einen Großteil meiner eigenen Programme in Basic schreibe und mir dabei die nicht sehr komfortable »POKE«-erei auf die Nerven geht, habe ich nun eine Basic-Erweiterung geschrieben die, wie ich meine, diese Vorgänge um Einiges erleichtert.

Eingabe des Programms

- 1) Das Programm »Game-Basic-C000« mit dem MSE-V 1.0 eingeben (siehe Listing 1) und speichern.
- 2) Programm mit »SYS 49152« starten. Das Programm kopiert sich nun selbständig an den Basic-Start ab \$0800.
- 3) Das Programm nun mit SAVE "GAME-BASIC",8 (oder bei Kassette: ,1) speichern.

Das Game-Basic kann nun ganz normal geladen und mit »RUN« gestartet werden.

Die Speicherbelegung

Alle wichtigen Adressen (Bildschirm, Farbspeicher, Sprites, SID, Timer etc.) bleiben unverändert. Der Bereich \$0800-\$0FFF ist für 32 Sprites reserviert, die dort abgelegt werden. Die Erweiterung selbst liegt im Bereich \$1000-\$17B0. Der Bereich \$17B1-\$1FFF bleibt frei und kann für eigene Maschinenprogramme genutzt werden. Ab der Adresse \$2000 bis \$27FF liegt ein frei definierbarer Zeichensatz. Ab \$2800 beginnt der Basic-Speicher (!). Dieser dehnt sich wie gewohnt bis zur Adresse \$9FFF. Das RAM im Bereich \$A000-\$CFFF und von \$E000-\$FFFF bleibt frei und kann für eigene Zwecke verwendet werden.

Die Befehle

Der Befehl »RESET« bewirkt einen Neustart der Basic-Erweiterung, wobei ein vorhandenes Basic-Programm nicht gelöscht wird.

Da es gerade bei Basic-Erweiterungen leicht vorkommen kann, daß man einzelne Befehle vergißt und nicht immer eine Anleitung zur Hand hat, habe ich den Befehl HELP mit eingebaut. Er listet alle zusätzlichen Kommandos auf.

Auf HiRes-Befehle habe ich grundsätzlich verzichtet, da meine Erweiterung ja hauptsächlich für die Verwendung im Spiele-Bereich vorgesehen ist. Normalerweise ist es nicht üblich, ein Spielprogramm in HiRes-Grafik zu erstellen (außer Loderunner und ein paar anderen), sondern mit Hilfe von Sonderzeichen. Außerdem wäre das Programm dann um einiges länger. (Es genügen schon zwei KByte, wenn man sie abtippen muß.)

Alle reinen Basic-Programme laufen natürlich auch mit der Erweiterung. Will man die Erweiterung trotzdem abschalten, ohne den Computer ausschalten zu müssen, so kann man das mittels des Befehls »BASIC« tun. Die Erweiterung kann dann mit SYS 4096 wieder aktiviert werden, falls sie nicht aus irgendeinem Grund überschrieben wurde.

Wer hat sich nicht auch schon über das lästige POKE 53281,xx:POKE 53280,xx:POKE 646,xx... geärgert? Mit dem Befehl »COLOR« lassen sich Bild-, Rahmen- und Zeichenfarbe ganz einfach einstellen. Zum Beispiel blauer Rahmen, weißer Hintergrund und grüne Schrift: COLOR6,1,5

Steuerzeichen entfallen

Damit wären wir auch schon bei den Befehlen, die den Bildschirm betreffen. Der Befehl »SCROFF« schaltet den Bildschirm aus; zum Beispiel während vom Programm ein Bild aufgebaut wird. Mit »SCRON« wird er wieder eingeschaltet. Die komplizierte »POKE«-erei im VIC hat nun ein Ende.

Mit »PAINT« wird der Farbspeicher für den Bildschirm mit der angegebenen Farbe gefüllt. Ein beschriebener Bildschirm kann natürlich auch nachträglich umgefärbt werden. Man kann damit ganz hübsche Effekte erzielen.

»REVERS« invertiert den Bildschirminhalt.

Auch die lästigen Steuerzeichen entfallen mit der Erweiterung bis auf RVSON und RVSOFF. (Aber auch die kann man durch POKE 199,1 beziehungsweise POKE 199,0 umgehen.) Der Bildschirm wird nicht mehr mit PRINTCHR\$(147), sondern einfach mit »CLS« gelöscht.

Mit dem Befehl »CLEAR« kann man sogar einzelne Zeilen beziehungsweise Zeilenblöcke löschen. Zum Beispiel »CLEAR 12,24« löscht den Bildschirm ab der 13. Zeile (die erste Zeile ist Zeile 0).

Das Editieren von Programmen wird mit meiner Erweiterung fast nicht unterstützt, da die Aufgabe der Expansion ja darin liegen soll, Spielprogramme in Basic schneller ablaufen zu lassen. Mit dem Befehl »TAB« kann man zumindest die Übersicht in einem Listing etwas erhöhen. Es wird jeder Befehl in eine eigene Zeile geschrieben, was, die Übersicht betreffend, auch nicht zu verachten ist. Bevor man dann eine Zeile wieder ändern will, muß unbedingt durch einen erneuten »TAB«-Befehl diese Darstellung wieder abgeschaltet werden.

Die Befehle INC und DEC, also Inkrement (Erhöhen) und Dekrement (Erniedrigen) sind eine Abwandlung des POKE-Befehls. Es wird nicht wie bei POKE eine Zahl in eine Adresse geschrieben, sondern eine in einer Adresse vorhandene Zahl um den angegebenen Wert erhöht beziehungsweise erniedrigt.

Sollte ein Programm für eine Sekunde anhalten, mußte man bisher eine umständliche Zeitschleife verwenden (FOR T = 1 TO 1000 : NEXT). Diese wird nun durch den Befehl PAUSE ersetzt. 1 Sekunde Wartezeit = PAUSE 500.

Zeichensatz verändern

Jedes professionelle Spiel kommt nicht ohne Sonderzeichen aus. Daher war es von vornherein klar, daß meine Erweiterung Befehle zur Erstellung eines eigenen Zeichensatzes enthalten sollte. Dabei ist die Darstellungsweise in Form von DATA-Zeichen die übersichtlichste. Ich habe dazu den Befehl »CODE« gewählt (siehe Beispiel »Ö« im folgenden Text).

In meiner Erweiterung gibt es den Befehl COPY: Er kopiert den Zeichensatz ins RAM, wo er dann mittels CREATE und CODE verändert werden kann. Mit CREATE wird das zu verändernde Zeichen festgelegt. Wir wollen zum Beispiel aus dem O einen Umlaut Ö machen. So schreiben wir folgendes: CREATE 1,15.

Die 1 bewirkt, daß wir ein Zeichen (0 wäre für Sprites) verändern und die 15 ist der POKE-Code für den Buchstaben O. Der Befehl CODE bestimmt nun das Bitmuster für das mit CREATE ausgewählte Zeichen. Unser Ö würde nun im Programm so aussehen:

Alle neuen Befehle auf einen Blick

RESET Neustart der Erweiterung; bereits vorhandene Basic-Programme werden nicht gelöscht.

HELP Auflistung aller Befehle von Game-Basic.

BASIC Abschalten der Erweiterung; auch hier werden Programme nicht gelöscht.

COLOR rf,hf,zf Rahmen- (rf), Hintergrund- (hf) und Zeichenfarbe (zf) setzen.

REVERS Invertiert den gesamten Bildschirminhalt.

PAINT f Färbt Bildschirminhalt mit Farbe f.

CLS Bildschirm löschen.

SCROFF Bildschirm ausschalten.

SCRON Bildschirm wieder einschalten.

CLEAR an,en Bildschirm wird von Zeile an bis Zeile en gelöscht.

COPY Zeichensatz ins RAM kopieren.

TAB Formatierten LIST-Modus ein- beziehungsweise ausschalten.

PAUSE t Das Programm wird t*0,002 Sekunden lang angehalten.

INC ad,eh Addiert eh zum Inhalt der Adresse ad.

DEC ad,eh Subtrahiert eh vom Inhalt der Adresse ad.

SET s,z,f,"text" Schreibt »text« in die Zeile z, Spalte s mit der Farbe f.

SCROLL az,ez,s,z,f,r Scrollt Bildschirmbereich von Anfangszeile az bis Endzeile ez in Richtung r (0=nach rechts, 1=nach links). Dabei werden Zeichen mit dem Code z in der Farbe f nachgeschrollt.

SON s Schaltet Sprite Nummer s ein.

SOFF s Schaltet Sprite Nummer s aus.

SPRITE s,bl,ye,xe,pr,mc,f
Bestimmt Sprite Nummer s:
bl = Block (32-64)
ye = doppelte Höhe (0=nein, 1=ja)
xe = doppelte Breite (0=nein, 1=ja)
pr = Priorität vor (=0) oder hinter (=1) Bildschirmzeichen
mc = Multicolor an (=1) oder aus (=0)
f = Farbe des Sprites

SSET s,x,y Setzt Sprite Nummer s an Position x (0-511!), y (0-255).

SRUN s,ri Sprite s bewegt sich in angegebene Richtung:

0 7 6 5 4 3 2 1

SSTOP s Sprite s hält wieder an.

MULTI c,z1,z2,s1,s2 Multicolor-Mode ein- und ausschalten:
c = 0 = Multicolor für Zeichen an
1 = Multicolor für Zeichen aus
2 = Angabe von Zusatzfarben
Bei Verwendung von 0 oder 1 brauchen die Zusatzfarben nicht mit angegeben werden.
z1 = Zusatzfarbe 1 für Zeichen
z2 = Zusatzfarbe 2 für Zeichen
s1 = Zusatzfarbe 1 für Sprites
s2 = Zusatzfarbe 2 für Sprites

CREATE m,b1 Erstellen eines neuen Zeichens/Sprites:
m = Mode (0=Sprite, 1=Zeichen)
b1 = Spriteblock oder Zeichencode

CODE bits Nur in Zusammenhang mit dem CREATE-Befehl.
Bei Sprites 24 Bits, bei Zeichen 8 Bits.
. = Hintergrundfarbe.
1 = Farbe des Zeichens/Sprites (bei Multicolor-Mode Zusatzfarbe 1)
2 = Zusatzfarbe 2
3 = Farbe des Zeichens/Sprites bei Multicolor-Mode

| | | |
|------------------|------|------------------|
| 10 CREATE 1,15 | oder | 10 CREATE 1,15 |
| 20 CODE .11..11. | | 20 CODE .11..11. |
| 30 CODE | | 30 CODE ..1111.. |
| 40 CODE ..1111.. | | 40 CODE |
| 50 CODE .11..11. | | 50 FORI=1TO4 |
| 60 CODE .11..11. | | 60 CODE .11..11. |
| 70 CODE .11..11. | | 70 NEXT I |
| 80 CODE .11..11. | | 80 CODE ..1111.. |
| 90 CODE ..1111.. | | |

Es würde auch schon folgendes genügen, da die restlichen Bitmuster mit dem O übereinstimmen:

| |
|------------------|
| 10 CREATE 1,15 |
| 20 CODE .11..11. |
| 30 CODE |
| 40 CODE ..1111.. |

Der Zeichensatz kann natürlich auch durch die altbewährte POKE-Methode verändert werden, da er jetzt »mitten« im RAM-Bereich liegt (ab \$2000 dez. 8192).

Auch der Multicolor-Modus wird in meiner Erweiterung sowohl für Zeichen als auch für Sprites voll unterstützt:

Aktiviert wird dieser mit dem Befehl MULTI 1 beziehungsweise ausgeschaltet mit MULTI 0. Die Multicolor-Zusatzfarben (zwei an der Zahl) bestimmt man ebenfalls mit MULTI.

»MULTI 2, 5, 2, 6, 7« bedeutet zum Beispiel, daß ab jetzt Zeichen in den Farben Grün (5) und Rot (2), Sprites in den Farben Blau (6) und Gelb (7) dargestellt werden. MULTI 2,... bedeutet: Multicolor-Modus mit den folgenden Farben einschalten. Bei den Bitmustern ändern sich jetzt allerdings die Codes: 1 = Zusatzfarbe 1, 2 = Zusatzfarbe 2, 3 = normale Zusatzfarbe, . = leer

Die bunte Version unseres O sähe nun so aus:

| | |
|---------------|-----------------------------------|
| CODE ..11..11 | |
| CODE ..11..11 | entspricht grün |
| CODE | |
| CODE ..222222 | |
| CODE ..22..22 | |
| CODE ..22..22 | entspricht rot |
| CODE ..33..33 | |
| CODE ..333333 | entspricht aktueller Zeichenfarbe |

Es ist darauf zu achten, daß die Zahlen der CODEs immer paarweise angegeben werden (Multicolor). Bei der Erstellung von Sprites wird genauso vorgegangen, bis auf folgende Änderungen:

1) »CREATE 0, Spriteblock (32-63)«. Statt dem Zeichensatz wird der gewählte Block angegeben (32-63). Die 0 teilt dem Computer mit, daß ein Sprite und kein Zeichen verändert beziehungsweise erstellt werden soll.

2) Das Bitmuster beim CODE-Befehl hat jetzt 24 Stellen anstatt acht bei Zeichen.

Bildschirmverwaltung

Der Befehl SET ist ein erweiterter PRINT-Befehl. Der Unterschied zum PRINT-Befehl liegt darin, daß die Steuerzeichen entfallen (wie anfangs erwähnt).

Zum Beispiel SET 18,12,1,"TEXT"

Dieser Befehl schreibt das Wort TEXT in die zwölfte Zeile ab der 18. Spalte in weißen Zeichen. Mit dem normalen PRINT-Befehl würde man dazu eine Unmenge Steuerzeichen benötigen; außerdem ist der SET-Befehl schneller.

Der Befehl SCROLL verschiebt beziehungsweise verändert beliebige Bildschirmbereiche. Eine genauere Erklärung möchte ich an dieser Stelle unterlassen, da man die Möglichkeiten durch eigene Versuche am besten sehen kann.

Die restlichen sechs der insgesamt 26 Befehle meiner Erweiterung unterstützen ausschließlich den Umgang mit

Basic-Erweiterungen

Sprites. Mit SON wird ein Sprite ein- beziehungsweise mit SOFF wieder ausgeschaltet.

SSET positioniert ein Sprite an eine beliebige Stelle im (oder außerhalb vom) Bildschirm. Das achte Bit im VIC-Register 16 für die Sprite X-Position braucht dabei nicht beachtet zu werden. Mit SPRITE wird ein Sprite definiert. Folgende Parameter werden hier angegeben:

Spritenummer, Block (Aussehen des Sprites — 32-63), Y-Expanded (ja oder nein), X-Expanded (ja oder nein), vor oder hinter Bildschirmzeichen, Multicolor (ja oder nein), Grundfarbe.

Ein ganz spezieller Befehl ist SRUN. Mit SRUN kann ein beliebiges Sprite (oder auch mehrere) in eine von acht möglichen Richtungen bewegt werden. Das Programm wird dadurch aber nicht aufgehalten, da die Sprites vom Interrupt gesteuert wer-

den. Dadurch ist es zum Beispiel möglich, daß ein Sprite, während es bewegt wird, seine Gestalt verändern oder sogar das Programm gelistet werden kann. Man kann sich so zum Beispiel ein Sprite in Bewegung ansehen, während man es editiert. Aber auch hier sind natürlich eigenen Versuchen keine Grenzen gesetzt. Der letzte Befehl SSTOP ist der Gegensatz zu SRUN, das heißt er stoppt die mit SRUN gestarteten Sprites. Die Positionen der Sprites können wie bisher mit dem normalen PEEK ausgelesen werden.

Eine Befehlsübersicht gibt Tabelle 1. Das Beispielprogramm (Listing 2) zeigt, wie man innerhalb von Spielen diese neuen Befehle einsetzt. Bitte zur Eingabe von Listing 2 nicht den Checksummer verwenden, da dieser nicht mit Basic-Erweiterungen zusammenarbeitet.

(Ronald Mayer/tr)

```

programm : gamebasic-c000 c000 c815
c000 : a9 01 85 2b 85 2d a9 08 69
c008 : 85 2c 85 ff a9 10 85 2e 92
c010 : a9 c0 85 fd a9 30 a2 00 e1
c018 : 85 fc 86 fe a0 00 b1 fc 67
c020 : 91 fe c8 d0 f9 e6 fd e6 19
c028 : ff a5 ff c9 10 90 ef 60 39
c030 : 00 15 08 c1 07 9e 28 32 60
c038 : 30 37 31 29 20 3a 20 a2 0f
c040 : 20 20 20 00 00 00 00 a9 d0
c048 : 08 85 fd a9 10 85 ff a9 48
c050 : 3d a2 00 85 fc 86 fe a0 d0
c058 : 00 b1 fc 91 fe c8 d0 f9 0f
c060 : e6 fd e6 ff a5 ff c9 18 b0
c068 : 90 ef 4c 00 10 20 a0 e5 53
c070 : a9 7c a0 16 20 1e ab a2 29
c078 : 0a ca bd 99 17 9d 00 03 ee
c080 : 8a d0 f6 a9 28 85 2c a9 18
c088 : 00 8d 00 28 60 a6 7a a0 ba
c090 : 04 84 0f bd 00 02 10 07 b0
c098 : c9 ff f0 3e e8 d0 f4 c9 e1
c0a0 : 20 f0 37 85 08 c9 22 f0 f0
c0a8 : 55 24 0f 70 2d c9 3f d0 a1
c0b0 : 04 a9 99 d0 25 c9 30 90 8c
c0b8 : 04 c9 3c 90 1d 84 71 a0 bf
c0c0 : 00 84 0b 88 86 7a ca c8 cf
c0c8 : e8 bd 00 02 38 f9 9e a0 de
c0d0 : f0 f5 c9 80 d0 2f 05 0b ee
c0d8 : a4 71 e8 c8 99 fb 01 c9 99
c0e0 : 00 f0 38 38 e9 3a f0 04 aa
c0e8 : c9 49 d0 02 85 0f 38 e9 50
c0f0 : 55 d0 a0 85 08 bd 00 02 f9
c0f8 : f0 e0 c5 08 f0 dc c8 99 17
c100 : fb 01 e8 d0 f0 a6 7a e6 cc
c108 : 0b c8 b9 9d a0 10 fa b9 83
c110 : 9e a0 d0 b5 f0 0f bd 00 68
c118 : 02 10 bd 99 fd 01 c6 7b bf
c120 : a9 ff 85 7a 60 a0 00 b9 f8
c128 : a8 16 d0 02 c8 e8 bd 00 1a
c130 : 02 38 f9 a8 16 f0 c5 c9 36
c138 : 80 d0 04 05 0b d0 99 a6 ad
c140 : 7a e6 0b c8 b9 a7 16 10 5b
c148 : fa b9 a8 16 d0 e0 f0 c6 71
c150 : a6 d3 e0 06 b0 04 a2 06 9a
c158 : 86 d3 aa 10 03 4c 07 11 45
c160 : c9 3a d0 0d a5 0f 29 01 96
c168 : d0 05 a9 3a 4c 04 11 a9 e9
c170 : 0d 4c f3 a6 10 0f 24 0f 9d
c178 : 30 0b c9 ff f0 07 c9 cc a8
c180 : b0 06 4c 24 a7 4c f3 a6 c5
c188 : 38 e9 cb aa 84 a9 a0 ff 12
c190 : ca f0 08 c8 b9 a8 16 10 47
c198 : fa 30 f5 c8 b9 a8 16 30 da
c1a0 : 05 20 47 00 d0 f5 4c ef ca
c1a8 : a6 20 73 ab 20 45 11 4c 44
c1b0 : ae a7 c9 cc 90 04 c9 ff 8e
c1b8 : 90 06 20 79 00 4c ed a7 ec
c1c0 : 38 e9 cc 0a aa bd 46 17 41
c1c8 : 48 bd 45 17 48 4c 73 00 d8
c1d0 : 98 48 8a 4a a8 b9 1e 17 77
c1d8 : 4d 10 d0 8d 10 d0 68 a8 8e
c1e0 : 60 b9 27 17 a8 a6 02 b9 05
c1e8 : 10 d0 1d 1e 17 99 10 d0 8c
c1f0 : 60 b9 27 17 a8 a6 02 b9 15
c1f8 : 10 d0 49 ff 1d 1e 17 49 74
c200 : ff 99 10 d0 60 8a 10 03 8b
c208 : 4c 74 a4 c9 1f bd 03 4c 0d
c210 : 3a a4 8a 0a aa bd 32 16 0e
c218 : 85 22 bd 33 16 85 23 4c 37
c220 : 47 a4 20 fd ae 20 9e b7 57
c228 : 60 a0 00 84 02 a2 08 a9 12
c230 : 2e d1 7a d0 05 06 02 4c 21
c238 : fd 11 a9 31 d1 7a d0 07 91
c240 : 06 02 e6 02 4c fd 11 a9 8d
c248 : 32 d1 7a d0 0d 06 02 06 31
c250 : 02 e6 02 20 73 00 ca 4c 45
c258 : fd 11 a9 33 d1 7a d0 11 05
c260 : 06 02 e6 02 06 02 20 73 39
c268 : 00 ca 20 73 00 ca d0 bf 5d
c270 : 60 a2 24 4c 98 11 18 85 31
c278 : 8e a0 00 84 8f 06 8e 90 6b
c280 : 02 e6 8f ca f0 05 06 8f a1
c288 : 4c 10 12 60 20 9e b7 e0 05
c290 : 09 90 05 a2 1f 4c 98 11 50
c298 : ad 15 d0 1d 1e 17 8d 15 a3
c2a0 : d0 60 20 9e b7 e0 09 b0 84
c2a8 : ea ad 15 d0 49 ff 1d 1e 0d
c2b0 : 17 49 ff 8d 15 d0 60 20 b7
c2b8 : 9e b7 8e 20 d0 20 b5 11 e0
c2c0 : 8e 21 d0 20 b5 11 8e 86 42
c2c8 : 02 60 a9 e8 a2 03 85 fe d8
c2d0 : 86 ff a0 00 b1 fe 49 80 b7
c2d8 : 91 fe c8 d0 f7 e6 ff a5 36
c2e0 : ff c9 07 90 ef a0 e8 9e 57
c2e8 : b7 8a 48 20 b5 11 e8 86 8f
c2f0 : 02 68 aa e4 02 b0 07 20 6f
c2f8 : ff e9 e8 4c 86 12 20 66 ff
c300 : e5 60 20 9e b7 86 d3 20 30
c308 : b5 11 86 d6 20 6c e5 20 ff
c310 : b5 11 8e 86 02 20 fd ae 39
c318 : 4c a4 aa 20 8a ad 20 f7 eb
c320 : b7 aa e8 c8 a9 ff 85 ff 30
c328 : a5 cb c9 3c f0 0a c6 ff 27
c330 : d0 fc 88 d0 ef ca d0 ec 2d
c338 : 60 20 f5 12 20 9e b7 e0 00
c340 : 08 90 05 a2 1f 4c 98 11 ff
c348 : a9 01 86 02 9d 81 17 20 d6
c350 : b5 11 e0 08 90 05 a2 20 c3
c358 : 4c 98 11 8a a6 02 9d 79 6a
c360 : 17 60 78 a9 02 8d 14 03 dd
c368 : a9 13 8d 15 03 58 60 a2 5b
c370 : 00 bd 81 17 f0 41 bd 79 95
c378 : 17 0a a8 8a 0a aa b9 89 00
c380 : 17 f0 1f c9 02 d0 0e bd 6b
c388 : 00 d0 d0 03 20 63 11 de a4
c390 : 00 d0 4c 35 13 bd 00 d0 73
c398 : c9 ff d0 03 20 63 11 fe 55
c3a0 : 00 d0 b9 8a 17 f0 0d c9 89
c3a8 : 02 d0 06 de 01 d0 4c 47 c6
c3b0 : 13 fe 01 d0 8a 4a aa e8 14
c3b8 : e0 08 d0 b5 4c 31 ea 20 c1
c3c0 : 9e b7 e0 08 f0 d0 90 05 37
c3c8 : a2 1f 4c 98 11 a9 00 9d ba
c3d0 : 81 17 60 a2 d0 8a 9d 81 17
c3d8 : 17 e8 e0 08 d0 f8 60 20 33
c3e0 : 9e b7 86 02 a9 00 a2 d8 13
c3e8 : 85 fe 86 ff a0 00 a5 02 32
c3f0 : 91 fe c8 d0 fb e6 ff a6 91
c3f8 : ff e0 dc 90 f3 60 ad 06 b5
c400 : 03 c9 e3 d0 07 a2 07 a0 de
c408 : 11 4c a3 13 a2 e3 a0 10 76
c410 : 8e 06 03 8c 07 03 60 20 3e
c418 : 53 e4 78 a9 ea a2 31 8d d4
c420 : 15 03 8e 14 03 58 60 20 92
c428 : 44 e5 a2 00 bd a8 16 f0 63
c430 : 16 85 02 0a 4a 20 d2 ff bc
c438 : a5 02 10 07 86 09 20 db ac
c440 : 13 a6 09 e8 4c bf 13 60 d6
c448 : a5 d3 c9 14 b0 08 a9 14 e6
c450 : 85 d3 20 c6 e5 60 20 d7 e6
c458 : aa 60 20 9e b7 e0 08 90 d2
c460 : 05 a2 1f 4c 98 11 86 02 38
c468 : 20 b5 11 8a a6 02 9d f8 db
c470 : 07 a0 00 84 ff 20 b5 11 52
c478 : a4 ff 8a f0 06 20 74 11 32
c480 : 4c 19 14 20 84 11 e6 ff ce
c488 : a5 ff c9 04 90 27 20 b5 54
c490 : 11 8a a6 02 9d 27 d0 60 e7
c498 : 20 9e b7 e0 19 90 05 a2 81
c4a0 : 0e 4c 98 11 8e a8 17 20 e7
c4a8 : b5 11 e0 19 b0 f1 e8 8e 9c
c4b0 : a3 17 20 b5 11 e0 28 b0 b8
c4b8 : e6 e8 8e a5 17 20 b5 11 d6
c4c0 : 8e a6 17 20 b5 11 8e a7 d9
c4c8 : 17 20 b5 11 8e a4 17 ae 47
c4d0 : a8 17 bd f0 e8 85 fc 85 8b
c4d8 : fe ad 88 02 18 7d 2c 17 5c
c4e0 : 85 fd 69 d4 85 ff ad a4 b1
c4e8 : 17 f0 06 20 a3 15 4c 87 20
c4f0 : 14 20 c2 15 e8 ec a3 17 1a
c4f8 : 90 d8 ce a5 17 d0 d0 60 59
c500 : 20 9e b7 e0 08 90 05 a2 d8
c508 : 1f 4c 98 11 8a 0a 85 02 a9
c510 : 20 fd ae 20 eb b7 8a a6 d2
c518 : 02 9d 01 d0 a5 14 a6 02 dd
c520 : 9d 00 d0 a0 04 a5 02 4a 10
c528 : 85 02 a5 15 f0 0c c9 02 55
c530 : 90 05 a2 0e 4c 98 11 4c 14
c538 : 74 11 4c 84 11 20 9e b7 d4
c540 : e0 02 90 05 a2 23 4c 98 8c
c548 : 11 8a 48 20 b5 11 68 d0 dc
c550 : 1c e0 20 90 04 e0 40 90 60
c558 : 05 a2 21 4c 98 11 8a a2 02
c560 : 03 86 8d a2 15 86 8c a2 5b
c568 : 06 20 09 12 60 8a a2 01 ea
c570 : 86 8d a2 08 86 8c a2 03 c4
c578 : 20 09 12 18 a5 8f 69 20 61
c580 : 85 8f 60 a5 8c d0 05 a2 42
c588 : 22 4c 98 11 6c 8c a9 00 90
c590 : 85 8b 20 bc 11 a5 02 a0 02
c598 : 00 91 8e e6 8e 8e 02 e6 26
c5a0 : 8f e6 8b a4 8b c4 8d d0 d0
c5a8 : e9 4c 79 00 78 a9 33 85 c2
c5b0 : 01 a9 00 85 fe 85 fc a9 9a
c5b8 : d0 85 ff a9 20 85 fd a0 e7
c5c0 : 00 b1 fe 91 fc c8 d0 f9 d8
c5c8 : e6 ff e6 fd a5 ff c9 d8 5a
c5d0 : 90 ef a9 37 85 01 58 a9 bc
c5d8 : 19 8d 18 d0 a9 ff 8d 91 cc
c5e0 : 02 60 20 9e b7 8a d0 06 0d
c5e8 : a0 c8 8c 16 d0 60 e0 02 6a
c5f0 : b0 05 a0 d8 4c 7d 15 20 ab
c5f8 : b5 11 8e 22 d0 20 b5 11 25
c600 : 8e 23 d0 20 b5 11 8e 25 c0
c608 : d0 20 b5 11 8e 26 d0 60 96
c610 : a0 26 b1 fe 48 b1 fc c8 a7
c618 : 91 fc 68 91 fe 88 88 c0 4b
c620 : ff d0 ef c8 ad a7 17 91 34
c628 : fe ad a6 17 91 fc 60 a0 4d
c630 : 01 b1 fe 48 b1 fc 88 91 1b
c638 : fc 68 91 fe c8 c8 c0 28 d3
c640 : d0 ef 88 ad a7 17 91 fe 57
c648 : ad a6 17 91 fc 60 20 44 1c
c650 : e5 60 20 eb b7 8e f1 15 cd
c658 : a0 00 b1 14 18 69 00 91 d7
c660 : 14 60 20 eb b7 8e 01 16 4a
c668 : a0 00 b1 14 38 e9 00 91 ed
c670 : 14 60 ad 11 d0 09 10 8d f3
c678 : 11 d0 60 ad 11 d0 29 ef db
c680 : 4c 0a 16 49 4c 4c 45 47 4b
c688 : 41 4c 20 53 50 52 49 54 c7
c690 : 45 4e 55 4d 42 45 d2 49 28
c698 : 4c 4c 45 47 41 4c 20 44 c4

```

Listing 1. Game-Basic. Bitte mit dem MSE (Seite 8) eingeben.

| | | |
|-----------------------------------|-----------------------------------|-----------------------------------|
| c6a0 : 49 52 45 43 54 49 4f ce 37 | c728 : 56 45 52 d3 43 4c 45 41 5e | c7a0 : 01 01 01 01 01 01 02 02 a7 |
| c6a8 : 50 41 47 45 20 49 53 20 ed | c730 : d2 53 45 d4 50 41 55 53 a3 | c7a8 : 02 02 02 02 02 03 03 03 b7 |
| c6b0 : 55 53 45 c4 43 4f 44 45 e3 | c738 : c5 53 52 55 ce 53 53 54 64 | c7b0 : 03 03 ff 0f 1e 12 34 12 7e |
| c6b8 : 20 4e 4f 54 20 44 45 46 24 | c740 : 4f d0 50 41 49 4e d4 54 37 | c7b8 : 49 12 5c 12 78 12 94 12 f2 |
| c6c0 : 49 4e 45 c4 49 4c 4c 45 cd | c748 : 41 c2 42 41 53 49 c3 48 c2 | c7c0 : ad 12 cb 12 51 13 71 13 45 |
| c6c8 : 47 41 4c 20 4d 4f 44 c5 b3 | c750 : 45 4c d0 53 50 52 49 54 bf | c7c8 : 90 13 a9 13 b9 13 ec 13 bd |
| c6d0 : 43 52 45 41 54 49 4e 47 0d | c758 : c5 53 43 52 4f 4c cc 53 13 | c7d0 : 2a 14 92 14 cf 14 15 15 48 |
| c6d8 : 20 43 4f 44 c5 16 16 2a b0 | c760 : 53 45 d4 43 52 45 41 54 f0 | c7d8 : 3e 15 74 15 e0 15 e4 15 d5 |
| c6e0 : 16 3b 16 47 16 57 16 63 3e | c768 : c5 43 4f 44 c5 43 4f 50 80 | c7e0 : f4 15 04 16 0d 16 00 00 a4 |
| c6e8 : 16 93 0d 9a 20 2a 2a 2a af | c770 : d9 4d 55 4c 54 c9 43 4c 08 | c7e8 : 00 00 00 00 00 00 00 00 e9 |
| c6f0 : 47 41 4d 45 42 41 53 49 e2 | c778 : d3 49 4e c3 44 45 c3 53 20 | c7f0 : 00 00 00 00 00 00 00 02 f5 |
| c6f8 : 43 20 56 31 2e 30 20 20 2c | c780 : 43 52 4f ce 53 43 52 4f d1 | c7f8 : 01 02 01 00 01 01 00 01 55 |
| c700 : 28 43 29 20 31 39 38 35 40 | c788 : 46 c6 00 01 02 04 08 10 d2 | c800 : 02 01 02 00 02 02 9b 11 b8 |
| c708 : 20 4d 52 2d 53 4f 46 54 7b | c790 : 20 40 80 ff 07 0d 0c 0b 0f | c808 : 83 a4 20 10 e3 10 3c 11 b9 |
| c710 : 2a 2a 2a 0d 00 52 45 53 ca | c798 : 00 00 00 00 00 00 00 00 99 | c810 : 00 00 00 00 00 ff 00 ff 10 |
| c718 : 45 d4 53 4f ce 53 4f 46 d7 | | |
| c720 : c6 43 4f 4c 4f d2 52 45 45 | | |

Listing 1. Game-Basic. (Schluß)

```

10 REM*****
20 REM* GAME-BASIC DEMO *
30 REM*-----*
40 REM* (C) 1985 MR-SOFT *
50 REM* RONALD MAYER *
60 REM* THENING 24 *
70 REM* A-4062 THENING *
80 REM* A U S T R I A *
90 REM*****
95 A$="* GAME-BASIC DEMOPROGRAMM ****
  $ 1985 BY MR-SOFT RONALD MAYER ****"
100 COLOR9,15,6:CLS:COPY
110 SET15,11,6,"GAMEBASIC":SET16,13,2,"D
  E M O"
120 FORI=0TO24:SCROLL0,24,0,32,15,0:SCRO
  LL0,24,0,32,15,1:NEXT:PAUSE500
125 FORI=1TO10:REVERS:PAUSE100:NEXT
126 SCROLL0,10,39,160,9,1
127 SCROLL14,24,39,160,9,0:PAUSE500
128 FORI=1TO40:SCROLL11,13,0,160,12,1:PA
  USE40:NEXT
130 FORI=0TO39:SCROLL0,24,0,160,I,0:NEXT
140 FORI=0TO24:SCROLLI,24,1,160,I,1:NEXT
  :PAINT11
150 CREATE1,160:FORI=1TO8
160 CODE.....:PAUSE200:NEXT
161 CREATE1,160:FORI=1TO8
162 CODE11111111:PAUSE200:NEXTI
170 SCROLL0,24,19,160,15,0
180 COLOR9,0,6
190 SCROLL0,24,19,32,0,1
200 CREATE0,32
210 CODE.....333333.....
220 CODE....3333333333.....
230 CODE..33333333333333.....
240 CODE..33333333333333.....
250 CODE3333333333333333.....
260 CODE3333333333..33333333.....
270 CODE33333333.....331111111111
280 CODE33333333.....1111111111
290 CODE33333333.....11111111..
300 CODE33333333.....111111.....
310 CODE33333333.....
320 CODE33333333.....222222.....
330 CODE33333333.....22222222..
340 CODE33333333.....2222222222
350 CODE33333333.....332222222222
360 CODE3333333333..33333333.....
370 CODE33333333333333333333.....
380 CODE..3333333333333333.....
390 CODE..3333333333333333.....
400 CODE....3333333333.....
410 CODE.....333333.....
420 SSTOP8:SOFF8:MULTI2,0,0,10,6:FORI=0T
  07:SSETI,40+I,138-I:NEXT
430 FORI=7TO0STEP-1:SPRITEI,32,0,0,1,1,I
  +1:SONI:PAUSE300:NEXT
440 FORJ=1TO2
450 FORI=0TO7:SRUNI,1:PAUSE80:NEXT
460 FORI=0TO7:SRUNI,5:PAUSE80:NEXT
465 FORI=0TO7:SRUNI,3:PAUSE80:NEXT
466 FORI=0TO7:SRUNI,7:PAUSE80:NEXTI,J
470 FORI=0TO7:SOFFI:SSTOPI:PAUSE100:NEXT
  I
480 SCROLL5,19,39,45,15,1
490 FORI=0TO7:SSETI,172,139:NEXT
500 SCROLL10,14,39,64,14,1
501 SCROLL0,4,9,32,0,0
502 SCROLL20,24,9,32,0,0
510 SON8:FORI=0TO7:SRUNI,I:NEXT:PAUSE200
  0
520 CREATE0,32
521 CODE.....33333333333333.....
522 CODE....3333333333333333....
523 CODE....3333333333333333....
524 CODE..333333.....333333..
525 CODE..333333.....333333..
526 CODE..3333..22222222..3333...:PAUSE50
  0
527 CODE333333..22222222..333333
528 CODE3333..112222222222..3333
529 CODE3333..111122222222..3333
530 CODE3333..111122222222..3333
531 CODE3333..111111222222..3333
532 CODE3333..111111112222..3333
533 CODE3333..111111112222..3333
534 CODE3333..111111111122..3333
535 CODE333333..11111111..333333:PAUSE50
  0
536 CODE..3333..11111111..3333..
537 CODE..333333.....333333..
538 CODE..333333.....333333..
539 CODE....3333333333333333....
540 CODE....3333333333333333....
541 CODE.....333333333333.....
550 CREATE1,0
560 CODE..1111..
570 CODE.1....1.
580 CODE1..11..1
590 CODE1.1....1
600 CODE1.1....1
610 CODE1..11..1
620 CODE.1....1.
630 CODE..1111..
640 FORI=1TO2:FORJ=1TO69:PAUSE40
649 B$=MID$(A$,J,1)
650 SCROLL12,12,0,ASC(B$)ANDNOT64,15,0:N
  EXTJ,I:PAUSE1000
660 FORI=0TO39:SCROLL0,24,0,77+RND(1)*2,
  9,1:NEXT:PAUSE3000:GOTO126
READY.

```

Listing 2. Beispielprogramm zu Game-Basic

Disk-Basic 64

Disk-Basic 64 ist eine besonders leistungsfähige diskettenorientierte Basic-Erweiterung, die mit fast 50 neuen Befehlen das Arbeiten mit dem Diskettenlaufwerk enorm erleichtert.

Wer kennt sie nicht, die unzähligen OPENS und PRINTs, die notwendig sind, wenn man Befehle an das 1541-Laufwerk senden will. Auch beim Laden und Speichern ist das normale Commodore-Basic nicht gerade komfortabel. Abhilfe schafft die Basic-Erweiterung »Disk-Basic 64«. Sie enthält unter anderem fast alle diskettenorientierten Befehle des Basic 4.0 von Commodore, das der C 128 ebenfalls kennt. Viele Funktionen die sich bisher nur mit Spezialprogrammen oder einem Diskmonitor realisieren ließen, werden nun auf einen Basic-Befehl hin ausgeführt und sind somit auch programmierbar.

Zur Eingabe von »Disk-Basic 64« verwenden Sie bitte den MSE. Nach dem Speichern kann die Basic-Erweiterung dann jederzeit mit »LOAD "DISK-BASIC",8« geladen und mit »RUN« gestartet werden. Danach steht »Disk-Basic 64« im Speicherbereich von \$9200 bis \$9DFF. Dieser Bereich wird für Basic automatisch gesperrt, so daß nur noch 35327 Bytes, also 3584 Bytes weniger, für Basic-Programme zur Verfügung stehen. Der \$C-Bereich bleibt für eigene Maschinenprogramme oder einen Monitor frei.

Doch jetzt zur Beschreibung der neuen Befehle:

DLOAD "name"

Das Programm »name« wird von der Diskette an den Basic-Anfang geladen. Entspricht »LOAD "name",8«.

DSAVE "name"

Das momentan im Speicher befindliche Basic-Programm wird auf der Diskette gespeichert. Entspricht »SAVE "name",8«.

DVERFIY "name"

Das momentan im Speicher befindliche Basic-Programm wird mit dem auf Diskette gespeicherten Programm »name« verglichen. Entspricht »VERIFY "name",8«.

REPLACE "name"

Ersetzt das Programm »name« auf der Diskette durch das im Speicher befindliche. Entspricht dem Basic-Befehl »SAVE "@:name",8«.

SCRATCH "name"

Das File »name« wird gelöscht. Im Namen dürfen auch die Jokerzeichen »?« und »*« verwendet werden. Es werden dann sämtliche Files, die dem Namen entsprechen, gelöscht. Entspricht dem Floppy-Befehl »S:«.

RENAME "altname" TO "neuname"

Das File mit dem Namen »altname« wird in »neuname« umbenannt. Entspricht dem Floppy-Befehl »R:«.

COPY "altfile" TO "neufile"

Das File »altfile« wird auf dieselbe Diskette als »neufile« kopiert. So können Sicherheitskopien von Files hergestellt werden. Mit diesem Befehl kann nicht von einer Diskette auf eine andere kopiert werden. Entspricht dem Floppy-Befehl »C:«.

HEADER "name,id"

Die im Laufwerk befindliche Diskette wird mit dem Namen »name« und der ID »id« formatiert. Wird »id« weggelassen, so wird die Diskette nur gelöscht. Entspricht dem Floppy-Befehl »N:«.

COLLECT

Die Diskette wird aufgeräumt, das heißt, daß alle Blöcke, die nicht von Files belegt sind, freigegeben werden und sämtliche nicht ordnungsgemäß geschlossenen Files gelöscht werden. Entspricht dem Floppy-Befehl »V«.

INIT

Die Diskette wird initialisiert. Entspricht dem Floppy-Befehl »I«.

CHECK

Der Fehlerkanal des Laufwerks wird auf dem Bildschirm ausgegeben (siehe auch DS\$).

CATALOG

Das vollständige Directory wird auf dem Bildschirm ausgegeben. Dabei verlangsamt die CTRL-Taste den Ausdruck, die Commodore-Taste hält die Ausgabe an und die Stop-Taste beendet den Befehl.

DIR "\$:auswahl"

Die dem Ausdruck entsprechenden Teile des Directory werden auf dem Bildschirm ausgegeben. »DIR "\$:* =P"« druckt beispielsweise nur die Programme, »DIR "\$:?=R"« alle einbuchstabigen, relativen Files. Ansonsten gelten die Kontrollmöglichkeiten von CATALOG.

BSAVE "name",start,ende+1

Mit diesem Befehl kann der Speicherbereich von start bis ende auf Diskette gespeichert werden. Dies können Maschinenprogramme, Grafikbilder oder sonstiges sein.

BLOAD "name" <,adr>

Das File »name« wird an die angegebene Speicheradresse geladen. Wird keine Adresse angegeben, so wird das File an die Adresse geladen, an der es beim Speichern stand. Da keine Basic-Vektoren verändert werden, tritt kein »OUT OF MEMORY ERROR« mehr auf, wenn in den \$C-Bereich geladen wurde. Ebenso wird bei einem BLOAD in einem Basic-Programm das Programm nicht neu gestartet, wie bei LOAD, sondern direkt hinter dem BLOAD fortgesetzt.

CHAIN "name" <,zn>

Das Basic-Programm »name« wird geladen und bei der Zeilennummer zn gestartet. Fehlt die Zeilennummer, so wird bei der ersten Zeile gestartet. Wird eine nicht existierende Zeilennummer angegeben, so gibt der Computer einen »UNDEF'D STATEMENT ERROR« aus. Achtung! Bei CHAIN gehen die Inhalte der Variablen wie bei LOAD verloren! Dies muß man beachten, wenn man CHAIN zum Nachladen von Programmteilen in Basic-Programmen einsetzen will.

PASS "kommando"

Mit diesem Befehl lassen sich Kommandos an das Laufwerk senden, die nicht direkt von Disk-Basic unterstützt werden, so zum Beispiel die »U«- oder »B«-Kommandos.

DPOKE adresse,wert

Dieser Befehl entspricht dem normalen POKE mit der Ausnahme, daß der Wert in der Adresse des Laufwerk-RAMs gespeichert wird. Er ist nur im Adressenbereich von 0 bis 2047 sinnvoll.

DPEEK(adresse)

Diese Funktion entspricht dem PEEK. Der Inhalt einer Speicherstelle des Laufwerkes kann mit »?DPEEK(adresse)« ausgedruckt oder mit »variable = DPEEK(adresse)« in eine Variable übergeben werden.

DSYS(adresse)

Mit diesem Befehl können Maschinenprogramme im Laufwerk gestartet werden. Entspricht dem Basic-Befehl SYS. Ein Beispiel: DSYS (60064) löst in dem angeschlossenen Floppy-Laufwerk einen Reset aus.

Basic-Erweiterungen

programm : disk-basic 0801 1641

```

0801 : 17 08 c1 07 9e 20 32 30 82
0809 : 37 33 20 44 49 53 4b 2d 21
0811 : 42 41 53 49 43 00 00 20 26
0819 : a9 41 85 fb a9 08 85 fc 2e
0821 : a9 00 85 fd a9 92 85 fe 2f
0829 : a0 00 a2 0e b1 fb 91 fd 71
0831 : 88 d0 f9 e6 fc e6 fe ca 15
0839 : d0 f2 4c 00 92 ea ea ea 97
0841 : 4c ff 9d 4c 06 9e a6 7a 63
0849 : a0 04 84 0f bd 00 02 10 f2
0851 : 07 c9 ff f0 3e e8 d0 f4 b3
0859 : c9 20 0f 37 85 08 c9 22 59
0861 : f0 56 24 0f 70 2d c9 3f 7d
0869 : d0 04 a9 99 d0 25 c9 30 97
0871 : 90 04 c9 3c 90 1d 84 71 e4
0879 : a0 00 84 0b 88 86 7a ca d8
0881 : c8 e8 bd 00 02 38 f9 9e 34
0889 : a0 f0 f5 c9 80 d0 30 05 b1
0891 : 0b a4 71 e8 c8 99 fb 01 b3
0899 : b9 fb 01 09 5a 38 e9 3a 32
08a1 : f0 04 c9 49 d0 02 85 0f 80
08a9 : 38 e9 55 d0 9f 85 08 bd 07
08b1 : 00 02 f0 df c5 08 f0 db 02
08b9 : c8 99 fb 01 e8 d0 f0 a6 93
08c1 : 7a e6 0b c8 b9 9d a0 10 b5
08c9 : fa b9 9e a0 d0 b4 a0 00 91
08d1 : f0 02 c8 e8 bd 00 02 38 66
08d9 : f9 92 9e f0 f5 c9 80 d0 32
08e1 : 02 f0 ac a6 7a e6 0b c8 f8
08e9 : b9 91 9e 10 fa b9 92 9e 19
08f1 : d0 e2 bd 00 02 10 9a 99 e0
08f9 : fd 01 c6 7b a9 ff 85 7a 3d
0901 : 60 10 12 c9 ff f0 0e 24 2f
0909 : 0f 30 0a 38 e9 7f c9 4d 16
0911 : b0 06 4c 27 a4 4c f3 a6 b6
0919 : 84 49 e9 4c aa a0 ff ca 8b
0921 : f0 08 c8 b9 92 9e 10 fa d3
0929 : 30 f5 c8 b9 92 9e 30 06 a8
0931 : 20 47 ab 4c ea 9e 42 ac ef bd
0939 : a6 20 73 00 20 02 93 4c c5
0941 : ae a7 f0 3d c9 8b f0 3a d8
0949 : c9 9b f0 6a 20 79 00 e9 0b
0951 : 80 90 07 c9 23 b0 d0 4c 9f
0959 : f7 a7 4c a5 a9 c9 3a f0 99
0961 : d8 4c 08 af c9 4c b0 03 1f
0969 : 4c 0e a8 c9 73 90 03 4c 80
0971 : 0e a8 e9 4b 0a a8 b9 8d 9f
0979 : 9f 48 b9 8c 9f 48 4c 73 91
0981 : 00 60 20 73 00 20 9e ad ff
0989 : 20 79 00 c9 89 f0 05 a9 27
0991 : a7 20 ff ae a5 61 d0 05 d1
0999 : 20 09 a9 f0 0d 68 68 20 bc
09a1 : 79 00 b0 03 4c a0 a8 4c ac
09a9 : fc 92 4c fb a8 20 79 00 f2
09b1 : 4c 9c a6 4c 08 99 20 73 33
09b9 : 00 f0 f2 90 f0 20 57 e2 33
09c1 : 85 24 a5 24 f0 ed a2 f2 35
09c9 : 20 54 94 a5 bb 85 22 a5 a9
09d1 : bc 85 23 20 45 94 a9 2c 15
09d9 : 20 dd ed a9 53 20 dd ed 22
09e1 : a9 2c 20 dd ed a9 57 20 2e
09e9 : dd ed 20 54 f6 a2 62 20 9e
09f1 : 54 94 a9 d1 8d 00 03 a9 6c
09f9 : 93 8d 01 03 a5 9a 48 a5 8f
0a01 : 02 85 9a 20 19 95 d0 03 f8
0a09 : 20 73 00 20 79 00 20 9c 38
0a11 : a6 a9 76 a0 a3 20 1e ab 48
0a19 : a9 8b 8d 00 03 a9 e3 8d 13
0a21 : 01 03 68 85 9a 20 54 f6 58
0a29 : a2 e2 20 54 94 4c 54 f6 ba
0a31 : a9 00 85 0d 20 73 00 c9 0f
0a39 : f3 90 21 c9 fa b0 1d 48 30
0a41 : 20 73 00 68 c9 f6 48 90 bf
0a49 : 03 20 f1 ae 68 38 e9 f3 86
0a51 : a8 b9 e1 9f 85 55 b9 da e2
0a59 : 9f 4c de af 20 79 00 4c 32
0a61 : 8d ae 20 57 e2 a9 00 85 bf
0a69 : b9 a5 02 85 ba 4c 59 e1 5d
0a71 : a9 01 2c a9 00 85 0a 20 70
0a79 : 57 e2 a9 00 85 b9 a5 02 6c
0a81 : 85 ba 4c 6f e1 a0 00 b1 eb
0a89 : 22 20 dd ed c8 c4 24 d0 d5
0a91 : ff 60 a2 ff a5 02 2c a2 c0
0a99 : f6 48 a9 00 85 90 68 20 e5
0aa1 : 0c ed 8a 20 b9 ed a5 90 0d
0aa9 : 10 e7 20 07 f7 aa 4c 37 0a
0ab1 : a4 20 52 94 20 54 f6 a5 58
0ab9 : 02 20 09 ed a9 6f 20 c7 f1
0ac1 : ed 4c 66 94 20 0e 99 20 1a
0ac9 : 52 94 a9 53 20 dd ed a9 36
0ad1 : 3a 20 dd ed 20 45 94 4c 67
0ad9 : 54 f6 a0 52 2c a0 43 84 f9
0ae1 : 92 20 0e 99 20 ca 94 a9 38
0ae9 : a4 20 ff ae 20 0e 99 20 8c
0af1 : 52 94 a5 92 20 dd ed a9 45
0af9 : 3a 20 dd ed 20 45 94 a9 4a

```

```

0b01 : 3d 20 dd ed 20 da 94 4c 47
0b09 : 54 f6 8d a7 02 a0 00 b1 b9
0b11 : 22 99 a8 02 c8 c4 24 d0 4f
0b19 : f6 60 ad a7 02 85 24 a9 d0
0b21 : 02 85 23 a9 a8 85 22 4c bc
0b29 : 05 94 20 0e 99 20 52 94 8f
0b31 : a9 4e 20 dd ed a9 3a 20 1a
0b39 : dd ed 20 45 94 20 19 95 97
0b41 : d0 13 20 73 00 20 9e ad e8
0b49 : 20 a6 b6 f0 08 a9 2c 20 47
0b51 : dd ed 20 45 94 4c 54 f6 c0
0b59 : 60 a0 00 b1 7a c9 2c 60 a7
0b61 : a0 56 2c a0 49 20 52 94 54
0b69 : 98 20 dd ed 4c 54 f6 20 ca
0b71 : 71 94 20 13 ee 48 20 d2 ee
0b79 : ff 68 c9 0d d0 f4 4c ef 86
0b81 : ed a2 f0 20 54 94 a9 24 d8
0b89 : 20 a8 ff 20 ae ff a5 02 87
0b91 : 20 b4 ff a9 60 20 96 ff a2
0b99 : a2 05 2c a2 03 20 87 95 98
0ba1 : ca d0 fa aa ad 8d 02 c9 ca
0ba9 : 02 f0 f9 20 e1 ff d0 03 0d
0bb1 : 4c 91 95 20 d7 aa 20 87 92
0bb9 : 95 20 cd bd 20 87 95 aa 73
0bc1 : f0 d9 20 d2 ff d0 f5 20 9f
0bc9 : a5 ff a4 90 d0 01 60 68 10
0bd1 : 68 20 ab ff a2 e0 20 54 8e
0bd9 : 94 4c ae ff a2 f0 20 54 1a
0be1 : 94 20 9e ad 20 a6 b6 85 00
0be9 : 24 a0 00 b1 22 c9 24 d0 36
0bf1 : 06 20 45 94 4c 4b 95 4c f9
0bf9 : 08 af 20 57 e2 a5 02 85 3a
0c01 : ba 20 8d 95 a5 14 85 c1 49
0c09 : a5 15 85 c2 20 d8 95 a6 5f
0c11 : 14 a4 15 a9 c1 4c 5f e1 b2
0c19 : 20 fd ae 20 8a ad 4c f7 1f
0c21 : b7 20 52 94 a5 9a 48 a5 ab
0c29 : 02 85 9a 20 a0 aa 68 85 a4
0c31 : 9a 4c 54 f6 20 57 e2 a5 79
0c39 : 02 85 ba a9 00 85 b9 a9 48
0c41 : 00 a6 2b a4 2c 20 d5 ff 0f
0c49 : b0 25 20 b7 ff 29 bf f0 b5
0c51 : 05 a2 1d 4c 37 a4 86 2d 85
0c59 : 84 2e 20 33 a5 20 79 00 a4
0c61 : c9 2c d0 06 20 73 00 4c 6b
0c69 : 71 a8 a9 00 4c 71 a8 4c 24
0c71 : d1 e1 20 57 e2 a5 02 85 94
0c79 : ba a9 01 85 b9 a9 19 95 25
0c81 : d0 0b 20 d8 95 a9 00 85 ac
0c89 : b9 a6 14 a4 15 a9 00 20 0e
0c91 : d5 ff b0 db 20 b7 ff 29 1f
0c99 : bf f0 05 a2 1d 4c 37 a4 c0
0ca1 : 60 20 eb b7 86 92 20 52 25
0ca9 : 94 20 15 9b a9 57 20 dd 98
0cb1 : ed a5 14 20 dd ed a5 15 88
0cb9 : 20 dd ed a9 01 20 dd ed dd
0cc1 : a5 92 20 dd ed 4c 54 f6 f4
0cc9 : 20 52 94 a5 15 48 a5 14 3e
0cd1 : 48 20 f7 b7 20 03 9b a5 f2
0cd9 : 14 20 dd ed a5 15 20 dd 72
0ce1 : ed 20 54 f6 20 71 94 20 f2
0ce9 : 13 ee a8 20 ef ed 68 85 bd
0cf1 : 14 68 85 15 4c a2 b3 20 26
0cf9 : 9e b7 20 0f f3 20 1f f3 01
0d01 : 20 52 94 a9 50 20 dd ed fe
0d09 : a5 b9 20 dd 20 fd ae 84
0d11 : 20 8a ad 20 f7 b7 a2 01 af
0d19 : 20 19 95 d0 03 20 9b b7 54
0d21 : a5 14 20 dd ed a5 15 20 35
0d29 : dd ed 8a 20 dd ed 4c 54 cb
0d31 : f6 20 71 94 a0 00 c8 20 94
0d39 : 13 ee 99 a6 02 c9 d0 d0 43
0d41 : f5 88 98 85 92 20 7d b4 db
0d49 : a0 00 b9 a7 02 91 62 c8 14
0d51 : c4 92 d0 f6 20 ef ed 68 7b
0d59 : 68 4c ca b4 20 71 94 a0 52
0d61 : 00 c8 20 13 ee 99 a6 02 8a
0d69 : c9 0d d0 f5 a5 7b 48 a5 4e
0d71 : 7a 48 a9 02 85 7b a9 a6 e2
0d79 : 85 7a 20 73 00 20 f3 bc fc
0d81 : 68 85 7a 68 85 7b 4c ef 9d
0d89 : ed 20 9e b7 20 1e e1 20 e0
0d91 : fd ae 20 9e b7 8a 48 20 f2
0d99 : fd ae 20 8b b0 85 49 84 cc
0da1 : 4a 20 a3 b6 68 20 75 b4 82
0da9 : a0 02 9f 61 00 91 49 88 a8
0db1 : 10 f8 c8 20 12 e1 91 62 af
0db9 : c8 c4 61 d0 f6 20 cc ff f9
0dc1 : 60 a9 90 85 a7 a9 11 d0 78
0dc9 : 06 a9 a2 85 a7 a9 06 85 e8
0dd1 : a8 4c 3a 9a f0 03 4c d0 7b
0dd9 : 9c 20 23 95 a9 58 85 92 99
0de1 : 4c d6 97 20 8a ad 20 f7 09
0de9 : b7 20 52 94 20 15 9b a9 44
0df1 : a5 20 dd ed a5 14 20 dd b3
0df9 : ed a5 15 20 dd ed 4c 54 29
0e01 : f6 f0 03 4c cd 9c 20 23 42
0e09 : 95 a9 41 85 92 a9 01 85 f9
0e11 : 15 85 14 20 66 96 a9 02 b8

```

```

0e19 : 85 14 a9 07 85 15 20 66 42
0e21 : 96 a9 07 85 14 a9 ef 85 58
0e29 : 15 20 a9 97 4c 23 95 20 20
0e31 : 9e 07 87 93 a9 a4 20 ff ff
0e39 : ae 20 9e b7 8a 18 09 40 a4
0e41 : 85 92 a9 00 85 15 a9 78 12
0e49 : 85 14 a5 93 20 57 94 20 03
0e51 : 69 96 a9 77 85 14 a5 93 15
0e59 : 20 57 94 a5 92 29 bf 09 82
0e61 : 20 85 92 4c 69 96 20 57 ec
0e69 : e2 a4 2e 38 a5 2d e9 02 9f
0e71 : b0 01 88 aa a5 02 85 ba 0f
0e79 : a9 00 85 b9 4c 75 e1 20 f3
0e81 : 9e b7 8a 20 57 94 a9 55 0d
0e89 : 20 dd ed a9 3a 4c 5c 9c f9
0e91 : 20 9e b7 8a 48 20 57 94 4c
0e99 : 68 85 02 4c 54 f6 a2 f2 3b
0ea1 : 20 54 94 a9 23 20 dd ed cc
0ea9 : 20 54 f6 20 9e b7 a0 00 df
0eb1 : 20 9c 9c 20 52 94 a9 4d 55
0eb9 : 20 dd ed a9 2d 20 dd ed 9f
0ec1 : a9 45 20 dd ed a9 a3 20 cc
0ec9 : dd ed a9 fd 20 dd ed 4c 08
0ed1 : 54 f6 a2 30 38 e9 0a 90 6b
0ed9 : 03 e8 b0 f9 69 3a 60 20 e6
0ee1 : 9e b7 86 b8 20 fd ae 20 00
0ee9 : 57 e2 20 0f f3 d0 03 4c 06
0ef1 : fe f6 a6 98 e0 0a 90 03 ce
0ef9 : 4c fb f6 e6 98 a5 b8 9d b2
0f01 : 59 02 29 0f 09 60 9d 6d 6c
0f09 : 02 a5 02 9d 63 02 a5 b7 5e
0f11 : f0 36 a9 00 85 90 a5 b8 6c
0f19 : 09 f0 aa 20 54 94 a5 90 eb
0f21 : 10 03 4c 07 f7 a0 00 b1 8f
0f29 : bb 20 dd ed c8 c4 b7 d0 5c
0f31 : f6 a9 2c 20 dd ed a9 53 a6
0f39 : 20 dd ed a9 2c 20 dd ed 0f
0f41 : a9 41 20 dd ed 4c 54 f6 cf
0f49 : 20 10 f7 4c 6d 94 20 9e 32
0f51 : ad 20 a6 b6 f0 f2 85 24 94
0f59 : 60 20 9e b7 86 b8 8a 29 12
0f61 : 0f 85 b9 a5 02 85 ba 20 cd
0f69 : fd ae 20 57 e2 20 19 95 6f
0f71 : d0 4f a0 00 b1 bb 99 a7 c0
0f79 : 02 c8 c4 b7 d0 f6 a9 77 c2
0f81 : 85 bb a9 02 85 bc 20 79 4b
0f89 : 00 99 a7 02 c8 20 73 00 db
0f91 : c9 4c f0 12 c9 57 f0 2c 72
0f99 : 99 a7 02 99 a7 02 c8 20 a8
0fa1 : 73 00 f0 1b d0 f5 99 a7 26
0fa9 : 02 c8 a9 2c 99 a7 02 c8 70
0fb1 : 84 92 20 9b b7 8a a4 92 81
0fb9 : 99 a7 02 c8 4c 7f 99 84 f0
0fc1 : b7 4c c1 e1 a9 53 99 a7 36
0fc9 : 02 c8 a9 2c 99 a7 02 c8 90
0fd1 : a9 57 99 a7 02 c8 20 73 4f
0fd9 : 00 4c f7 99 d0 1e a0 7f 92
0fe1 : 98 84 92 20 14 f3 d0 0d ae
0fe9 : 20 1f f3 a5 ba c5 02 d0 c2
0ff1 : 04 8a 20 9c f2 a4 92 88 86
0ff9 : d0 e6 18 60 a9 23 20 ff 83
1001 : ae 4c c7 e1 a0 00 b9 92 19
1009 : 9e 30 09 f0 14 20 d2 ff ad
1011 : c8 4c c6 99 29 f7 20 d2 99
1019 : ff a9 0d 20 d2 ff c8 d0 26
1021 : e5 a9 0d 4c d2 ff 20 0e 71
1029 : 99 20 ca 94 a9 26 20 ff 64
1031 : ae ac a7 02 a9 2c c8 99 b2
1039 : a7 02 8c a7 02 20 c8 99 86
1041 : ae a7 02 a0 00 18 6d a7 1d
1049 : 02 8d a7 02 b1 22 9d a8 30
1051 : 02 c8 e8 c4 24 d0 f5 a9 7e
1059 : a4 20 ff ae 20 0e 99 20 fc
1061 : 52 94 a9 43 20 dd ed a9 cc
1069 : 3a 20 dd ed 20 45 94 a9 ba
1071 : 3d 20 dd ed 20 da 94 4c b7
1079 : 54 f6 a0 10 a9 20 99 a6 c2
1081 : 02 88 d0 fa 20 9e ad 20 49
1089 : a6 b6 f0 4e c5 a8 b0 45 7f
1091 : 85 24 a0 00 b1 22 99 a7 32
1099 : 02 c8 c4 24 d0 f6 20 52 9f
10a1 : 94 20 15 9b a9 57 20 dd 90
10a9 : ed a5 a7 20 dd ed a9 07 59
10b1 : 20 dd ed c6 a8 a5 a8 20 af
10b9 : dd ed a0 00 b9 a7 02 20 d6
10c1 : dd ed c8 c4 a8 d0 f5 20 89
10c9 : 54 f6 a9 07 85 14 a9 ef 63
10d1 : 85 15 4c a9 97 a2 17 4c ad
10d9 : 37 a4 4c 08 99 20 53 e4 28
10e1 : a9 a0 85 38 20 60 a6 4c 7b
10e9 : 74 a4 a9 40 8d a7 02 a9 93
10f1 : 3a 8d a8 02 20 0e 99 70 76
10f9 : 00 b1 22 99 a9 02 c8 c4 e5
1101 : 24 d0 f6 c8 c8 84 b7 a9 47
1109 : a7 85 bb a9 02 85 bc 4c 6f
1111 : 25 94 20 23 95 a0 fc 20 7f
1119 : f1 9a 20 0b 9b 84 62 a0 69
1121 : fa 20 f1 9a 20 0b 9b 84 cd
1129 : 63 a5 62 a2 90 38 4c 49 da

```

| | | |
|-----------------------------------|-----------------------------------|-----------------------------------|
| 1131 : bc 20 52 94 20 03 9b 98 de | 12e9 : 8e ef 9f 8d f0 9f 98 20 b7 | 14a1 : 33 2e 32 20 2a 2a 2a 2a 6d |
| 1139 : 20 dd ed a9 02 20 dd ed 6d | 12f1 : 92 98 8e f2 9f 8d f3 9f 47 | 14a9 : 0d 0d 20 33 35 33 32 37 cf |
| 1141 : 4c 54 f6 20 15 9b a9 52 f2 | 12f9 : 20 52 94 a9 e8 85 22 a9 33 | 14b1 : 20 42 59 54 45 53 20 46 cf |
| 1149 : 4c dd ed 20 71 94 20 13 66 | 1301 : 9f 85 23 a9 0c 85 24 20 1f | 14b9 : 52 45 45 20 28 43 29 20 85 |
| 1151 : ee a8 4c ef ed a9 4d 20 46 | 1309 : 45 94 4c 54 f6 a9 01 2c 4f | 14c1 : 33 2f 31 39 38 35 20 42 32 |
| 1159 : dd ed a9 2d 4c dd ed 20 e8 | 1311 : a9 00 85 93 20 c5 9b a5 78 | 14c9 : 59 20 41 2e 46 52 49 4e 01 |
| 1161 : 82 b7 d0 08 20 10 f7 a2 9c | 1319 : 92 0a 0a 0a 0a 18 69 98 | 14d1 : 4b 0d 44 53 41 56 c5 44 85 |
| 1169 : 08 4c 37 a4 a6 22 a4 23 4e | 1321 : 02 48 20 74 9c 20 6a 9c 8c | 14d9 : 56 45 52 49 46 d9 44 4c 6c |
| 1171 : 20 f9 fd a0 00 20 60 9b db | 1329 : 20 13 ee 48 20 ef ed 68 a1 | 14e1 : 4f 41 c4 53 43 52 41 54 e1 |
| 1179 : 20 52 94 20 54 f6 a5 02 83 | 1331 : 29 bf a4 93 d0 02 09 40 97 | 14e9 : 43 c8 52 45 4e 41 4d c5 7d |
| 1181 : 20 09 ed a9 60 20 c7 ed d8 | 1339 : 85 a9 68 20 74 9c a2 62 2c | 14f1 : 43 4f 50 d9 48 45 41 44 67 |
| 1189 : 20 13 ee 85 63 20 13 ee 00 | 1341 : 20 54 94 a5 a9 20 dd ed 54 | 14f9 : 45 d2 43 4f 4c 4c 45 43 25 |
| 1191 : 85 62 20 ef ed a9 7f 20 b8 | 1349 : 20 54 f6 a6 a7 a4 aa 20 b0 | 1501 : d4 49 4e 49 d4 43 48 45 4a |
| 1199 : c3 ff a2 90 38 4c 49 bc 9b | 1351 : 9f 9c 4c 62 9c 20 c5 9b b7 | 1509 : 43 cb 43 41 54 41 4c 4f 4a |
| 11a1 : a9 7f a6 02 20 00 fe 4c 8a | 1359 : 20 fd ae 20 9e b7 8a 18 29 | 1511 : c7 44 49 d2 42 53 41 56 17 |
| 11a9 : c0 ff 20 57 e2 a0 0e 20 07 | 1361 : 69 80 85 a9 a5 92 0a 0a cc | 1519 : c5 50 41 53 d3 43 48 41 bc |
| 11b1 : 60 9b a9 2e 8d 2c 03 a9 a9 | 1369 : 0a 0a 0a 18 69 02 8d fb 1d | 1521 : 49 ce 42 4c 4f 41 c4 44 86 |
| 11b9 : 93 8d 00 03 a9 9b 8d 01 23 | 1371 : 9c 4c 20 95 20 82 b7 00 85 | 1529 : 53 59 d3 44 50 4f 4b c5 de |
| 11c1 : 03 a9 b7 8d 24 03 a9 9b 71 | 1379 : 03 4c 9a 9a 85 24 20 c8 28 | 1531 : 52 45 43 4f 52 44 a3 46 43 |
| 11c9 : 8d 25 03 a2 7f 20 c6 ff 12 | 1381 : 9b a5 92 0a 0a 0a 0a 02 | 1539 : 45 54 43 48 a3 4e 41 4d ce |
| 11d1 : 4c 7b a4 a9 8b 8d 00 03 64 | 1389 : 18 69 1e 20 74 9c 20 6a 63 | 1541 : c5 49 c4 50 52 4f 54 45 61 |
| 11d9 : a9 e3 8d 01 03 a9 57 8d ed | 1391 : 9c 20 13 ee 85 63 20 13 fa | 1549 : 43 d4 52 45 50 52 4f 54 b1 |
| 11e1 : 24 03 a9 f1 8d 25 03 a9 91 | 1399 : ee 85 62 20 ef ed 20 62 9a | 1551 : 45 43 d4 43 48 41 4e 47 2c |
| 11e9 : 2f 8d 2c 03 a9 7f 20 c3 e9 | 13a1 : 9c 4c 5a 9b 20 a6 b6 85 8a | 1559 : c5 4d 45 52 47 c5 52 45 d7 |
| 11f1 : ff 20 e7 ff 4c 74 a4 a5 40 | 13a9 : 24 a8 d0 03 4c 9a 9a 85 c5 | 1561 : 53 45 d4 44 45 56 49 43 c7 |
| 11f9 : 90 f0 04 a9 0d 18 60 20 8b | 13b1 : 24 20 c8 9b a5 92 0a 0a b6 | 1569 : c5 44 45 53 54 52 4f d9 d5 |
| 1201 : 13 ee 4c d2 ff 20 0e 99 65 | 13b9 : 0a 0a 0a 18 69 02 8d fb ba | 1571 : 41 50 50 45 4e 44 c4 44 b5 |
| 1209 : a0 00 b1 22 99 a7 02 c8 ca | 13c1 : 02 20 74 9c 20 6a 9c a2 91 | 1579 : 4f 50 45 4e a3 51 55 49 b8 |
| 1211 : c4 24 d0 f6 a9 a0 99 a7 50 | 13c9 : 00 20 13 ee 9d a7 02 e8 6d | 1581 : d4 44 43 4c 4f 53 c5 43 ff |
| 1219 : 02 c8 c0 10 90 f8 20 55 ad | 13d1 : e0 1e d0 f5 20 ef ed 20 2d | 1589 : 4f 4d 4d 41 4e 44 d3 43 d7 |
| 1221 : 9c a9 12 85 a7 85 ab a9 70 | 13d9 : 62 9c a5 a7 8d c5 02 a5 42 | 1591 : 4f 4e 43 41 d4 52 45 50 96 |
| 1229 : 01 85 a8 85 aa a6 a7 a4 8f | 13e1 : aa 8d c6 02 a9 21 20 7d 63 | 1599 : 4c 41 43 c5 45 4e 54 45 b2 |
| 1231 : a8 20 9c 9c a5 a8 85 aa af | 13e9 : b4 a0 20 b9 a7 02 91 62 c2 | 15a1 : d2 52 45 53 43 55 c5 57 fd |
| 1239 : a5 ab 85 a7 a9 00 20 74 0e | 13f1 : 88 10 f8 68 68 4c ca b4 4a | 15a9 : 52 49 54 c5 44 53 a4 44 68 |
| 1241 : 9c 20 6a 9c 20 13 ee 85 7d | 13f9 : 20 0e 99 c9 21 f0 03 4c fe | 15b1 : d3 42 4c 4f 43 4b d3 44 09 |
| 1249 : ab 20 13 ee 85 a8 20 ef a5 | 1401 : 08 af a5 22 85 a7 a5 23 01 | 15b9 : 50 45 45 cb 53 49 5a c5 eb |
| 1251 : ed a9 07 85 92 a5 92 0a 3a | 1409 : 85 a8 20 55 9c a0 1e b1 40 | 15c1 : 45 4e 54 52 59 a4 53 54 3d |
| 1259 : 0a 0a 0a 0a 18 69 05 20 4d | 1411 : a7 aa c8 b1 a7 a8 20 9c ef | 15c9 : 41 52 d4 00 21 94 2f 94 05 |
| 1261 : 74 9c 20 6a 9c a0 00 20 88 | 1419 : 9c a0 20 b1 a7 20 74 9c ca | 15d1 : 32 94 83 94 99 94 9c 94 9b |
| 1269 : 13 ee 99 b7 02 c8 c0 10 da | 1421 : a2 62 20 54 94 a0 00 b1 39 | 15d9 : e9 94 1f 95 22 95 2e 95 3a |
| 1271 : d0 f5 20 ef ed a0 00 b9 99 | 1429 : a7 20 dd ed c8 c0 1e d0 c2 | 15e1 : 40 95 9b 95 b9 95 0e 95 7c |
| 1279 : b7 02 d9 a7 02 d0 06 c8 ed | 1431 : f6 20 54 f6 a9 32 8d e9 61 | 15e9 : f3 95 31 96 a2 97 60 96 5c |
| 1281 : c0 10 d0 f3 60 c6 92 10 a2 | 1439 : 9f 20 88 9c 4c 62 9c a0 36 | 15f1 : b6 96 48 97 80 97 88 97 0d |
| 1289 : cc a5 ab d0 a0 20 62 9c fb | 1441 : 92 84 38 20 44 a6 20 18 52 | 15f9 : 93 97 c0 97 ee 97 25 98 ec |
| 1291 : a2 04 4c 37 a4 a2 f2 20 9b | 1449 : e5 a2 00 a0 00 bd 3e 9e b8 | 1601 : 3e 98 4f 98 5d 98 9e 98 b8 |
| 1299 : 54 94 a9 23 20 dd ed 4c 47 | 1451 : 20 d2 ff 98 5d 3e 9e a8 81 | 1609 : 18 99 9c 9a 9b 99 c3 99 31 |
| 12a1 : 54 f6 20 ab ff a2 e2 4c 27 | 1459 : e8 e0 54 d0 f0 07 10 5d | 1611 : e5 99 a9 9a 69 9b 14 9d 7f |
| 12a9 : 96 95 a5 02 20 09 ed a9 09 | 1461 : 12 a0 07 b9 36 9e 94 04 83 | 1619 : b7 9d 96 97 9a 96 9d 9d 47 |
| 12b1 : 62 4c c7 ed 20 92 98 8d fd | 1469 : 03 88 10 f7 a9 08 85 02 a8 | 1621 : 9b f1 1c d2 88 34 64 1f 10 |
| 12b9 : fc 9f 8a 38 e9 30 20 92 f4 | 1471 : 4c 74 a4 6c ff 06 92 bb | 1629 : 55 31 20 32 20 30 20 00 69 |
| 12c1 : 98 8e fa 9f 8d fb 9f a9 dd | 1479 : c1 92 f9 92 f0 93 93 8e 6b | 1631 : 00 20 00 00 42 2d 50 3a 85 |
| 12c9 : f4 85 22 a9 9f 85 23 a9 44 | 1481 : 0d 20 2a 2a 2a 20 43 69 | 1639 : 32 20 00 53 41 56 45 45 4c |
| 12d1 : 09 85 24 20 52 94 20 45 7f | 1489 : 4f 4d 4d 4f 44 4f 52 45 4f | |
| 12d9 : 94 4c 54 f6 a9 31 2c a9 af | 1491 : 20 36 34 20 44 49 53 4b 50 | |
| 12e1 : 32 8d e9 9f 8a 20 92 98 6d | 1499 : 2d 42 41 53 49 43 28 56 7e | |

Listing »Disk-Basic«. Beachten Sie die Eingabehinweise auf Seite 8.

NAME "name"

Der 16 Zeichen lange Name einer Diskette wird umgeändert. Hat der Stringname mehr als 16 Zeichen, wird ein »STRING TOO LONG ERROR« ausgegeben.

ID "id"

Mit diesem Befehl kann die Id sowie das Formatkennzeichen in der ersten Zeile des Directory geändert werden. Der String »id« kann maximal 5 Buchstaben haben und überschreibt dann die ID, das Formatkennzeichen (normalerweise »2A«) und das Leerzeichen zwischen den beiden. Achtung! Diese Änderungen beziehen sich nur auf das Directory. Die tatsächliche ID und Formatkennzeichen werden nicht verändert.

PROTECT <"name">

Mit dem Befehl PROTECT wird eine ganze Diskette schreibgeschützt, indem das echte Formatkennzeichen von »A« auf »X« geändert wird. Es sind dann keinerlei Schreibzugriffe mehr auf diese Diskette möglich.

Wird beim PROTECT-Befehl ein Name angegeben, so wird nur das File mit diesem Namen gegen Löschen und Überschreiben geschützt. Der Schreibschutz ist am Kleinerzeichen hinter dem Filetyp im Directory erkennbar. Es dürfen keine Joker im Namen angegeben werden.

REPROTECT <"name">

Mit diesem Befehl lassen sich der Disketten- und File-Schreibschutz wieder rückgängig machen. Siehe PROTECT.

CHANGE dev.alt TO dev.neu

Hiermit lassen sich softwaremäßig die Gerätenummern des 1541-Laufwerks verändern. So ändert »CHANGE 8 TO 9« die Gerätenummer des gerade eingeschalteten Laufwerkes auf 9. So lassen sich mehrere Laufwerke gleichzeitig betreiben.

DEVICE dev

Mit diesem Befehl läßt sich einstellen, auf welche Gerätenummer sich die Befehle des Disk-Basics beziehen. Nach »DEVICE 9« werden sämtliche Befehle am Laufwerk unter der Gerätenummer 9 ausgeführt. Dieser Wert wird in der Speicherstelle 2 gespeichert.

RESET d

Das Laufwerk mit der Gerätenummer d wird in den Einschaltzustand versetzt. Aus Sicherheitsgründen muß immer die Gerätenummer angegeben werden. Bitte warten Sie einige Sekunden bevor Sie dieses Laufwerk wieder ansprechen.

MERGE "name"

Das Programm »name« wird von der Diskette geladen und an das im Speicher befindliche angehängt. Achten Sie darauf, daß das zweite Programm höhere Zeilennummern als das erste hat.

CONCAT "altfile1" & "altfile2" TO "neufile"

Mit diesem Befehl können zwei sequentielle Files zu einem verkettet werden. Dies funktioniert nicht mit Programmen!

Basic-Erweiterungen

DS\$ und DS

In diesen beiden Variablen ist stets die aktuelle Fehlermeldung (DS\$) beziehungsweise die aktuelle Fehlernummer (DS) des Laufwerkes. Sie dürfen deswegen nicht in Zuweisungen wie »DS\$= "HALLO"« verwendet werden.

DOPEN # lfn, "name", p1, p2

Mit DOPEN lassen sich alle Arten von Files auf dem Laufwerk eröffnen. Die Filenummer darf von 2 bis 255 gehen. Empfehlenswert sind Nummern von 2 bis 14, da die Sekundäradresse des Files immer gleich der Filenummer logisch »UND«-verknüpft mit 15 ist, also immer im Bereich von 0 bis 15 liegt. Für den Fileparameter p1 sind die Buchstaben »S«, »U«, »P« und »L« erlaubt. Die ersten drei stehen für sequentielle, User- und Programm-Files. Der Parameter p2 ist dann entweder »W« oder »R« für Schreiben oder Lesen.

Lautet der erste Parameter »L« so wird ein relatives File zum Schreib/Lese-Zugriff eröffnet. Die Länge eines Datensatzes (Record) wird direkt an das »L« angehängt. Der zweite Parameter entfällt. Ein Beispiel: »DOPEN 2, "name", L40« eröffnet ein relatives File mit der Record-Länge 40.

DCLOSE < # lfn >

Dieser Befehl schließt das File mit der Nummer lfn. Wird keine Filenummer angegeben, werden alle geöffneten Files ordnungsgemäß geschlossen.

APPEND # lfn, "name"

Ein sequentielles File wird zum Erweitern (Anhängen von Datensätzen) geöffnet. Die Sekundäradresse errechnet sich wie bei DOPEN.

FETCH # lfn, len, xx\$

Mit diesem Befehl werden aus dem File mit der Nummer lfn eine festgelegte Anzahl (len) von Zeichen in eine beliebige Stringvariable (xx\$) übertragen. Die Vorteile gegenüber dem INPUT #-Befehl sind: Datensätze sind nicht mehr auf 88 Zeichen beschränkt, sondern dürfen bis zu 255 Zeichen haben. Es werden auch Kommata und Doppelpunkte übernommen. Der Befehl kann auch im Direktmodus eingesetzt werden.

RECORD # lfn, rec < , pos >

Mit diesem Befehl wird die Bearbeitung von relativen Dateien zum Kinderspiel. Der Zeiger auf die Datei mit der Filenummer lfn wird auf den Datensatz mit der Nummer rec positioniert. Zusätzlich kann noch die Position des Datei-Zeigers innerhalb dieses Datensatzes angegeben werden. Wird dieser Parameter weggelassen, wird immer auf das erste Zeichen positioniert. Dies entspricht der Befehlsfolge:

```
»PRINT #15, "P";CHR$(lfn);CHR$(recl);CHR$(rech);CHR$(pos)«
```

LIST "name" < , start-ende >

Der LIST-Befehl wurde erweitert, so daß bei Angabe eines Namens das Listing im ASCII-Code als sequentielles File auf der Diskette abgelegt wird. Sollen nur Teile abgespeichert werden, so kann die normale LIST-Syntax für Bereichsmarkierungen angehängt werden. Basic-Programme können somit in Textverarbeitungsprogramme übernommen werden.

ENTER "name"

Mit ENTER kann ein auf Diskette gelistetes Programm wieder in den Speicher übernommen werden. Die Zeichen, die gerade gelesen werden, werden zusätzlich auf dem Bildschirm ausgegeben. Eine Anwendungsmöglichkeit ist ein »echtes« MERGE, bei dem das zweite Programm nicht einfach angehängt, sondern die beiden entsprechend der Zeilennummern zusammengemischt werden. Achtung! Es dürfen keine doppelten Zeilennummern vorkommen!

BLOCKS

Dies ist eine weitere Basic-Funktion. Mit ihr erhält man die

Anzahl der freien Blöcke einer Diskette. Diese Funktion kann wahlweise in PRINT-Befehlen oder Wertzuweisungen verwendet werden.

START("name")

Mit dieser Funktion läßt sich die Startadresse des Programms »name« ermitteln. Auch diese Funktion kann in PRINT-Befehlen und Wertzuweisungen verwendet werden. Ist dieses Programm nicht vorhanden, wird als Startadresse der Wert 51143 ermittelt.

SIZE("name")

Mit dieser Funktion läßt sich die Länge eines Files in Blocks ermitteln. Hier gilt dasselbe wie bei den beiden anderen beschriebenen Funktionen.

RESCUE "name", ft

Mit diesem Befehl kann ein soeben gelöscht File wieder gerettet werden beziehungsweise der Filetyp eines Files geändert werden. Für den Parameter ft sind die Zahlen 1 bis 4 erlaubt. Sie bedeuten:

```
0 = DELETED  
1 = SEQUENTIAL  
2 = PROGRAM  
3 = USER  
4 = RELATIVE
```

Der Filetyp im Directory wird angepaßt und anschließend ein VALIDATE ausgeführt, um die Blöcke dieses Files wieder zu belegen. Achtung! Wurde zwischen dem Löschen und dem Rettungsversuch auf die Diskette geschrieben (SAVE oder ähnliches), dann läßt sich mit hoher Wahrscheinlichkeit das gelöschte Programm nicht mehr retten!

ENTRY\$("name")

Mit dieser Funktion lassen sich die kompletten 30 Bytes eines Directory-Eintrags in einen String ablegen: »xx\$=ENTRY\$("name")«. Die Verwendung im PRINT-Befehl ist zwar möglich aber sinnlos. Achtung! An den 30 Byte langen Fileeintrag werden drei Bytes angehängt, die Spur, Sektor und Position des gelesenen Eintrages darstellen. Diese drei Bytes dürfen nicht verändert werden, da sonst beim Zurückschreiben mit WRITE das Directory zerstört wird!

WRITE xx\$

Mit diesem Befehl kann ein manipulierter Directory-Eintrag zurückgeschrieben werden. Zusammen mit der ENTRY\$-Funktion läßt sich beispielsweise ein Directory-Sortierprogramm aufbauen. Siehe ENTRY\$.

DESTROY n

Der Track mit der Nummer n wird endgültig zerstört. Beim Zugriff auf diesen Track erhält man nur den 21 READ ERROR. Die Diskette kann nur noch durch komplettes Neuformatieren normalisiert werden. Nach der Ausführung dieses Befehls muß das Laufwerk kurz ausgeschaltet werden.

QUIT

Mit QUIT wird Disk-Basic verlassen, ohne daß das im Speicher befindliche Basic-Programm gelöscht wird. Allerdings gehen die Variablen-Inhalte verloren.

COMMANDS

Auf dieses Kommando hin werden alle zusätzlichen Befehle von Disk-Basic auf dem Bildschirm ausgegeben.

Noch einige kleine Anmerkungen: Anders als bei vielen Basic-Erweiterungen muß zwischen dem THEN-Befehl und einem Disk-Basic-Kommando kein Doppelpunkt stehen! Bei allen Befehlen wird, ist das durch DEVICE festgelegte Laufwerk nicht angeschlossen, ein »DEVICE NOT PRESENT ERROR« gemeldet. Bei fehlendem Filenamen gibt es einen »MISSING FILENAME ERROR«.
(Alexander Frink/bs)

COP — Hilfe für Datasette

COP ist die Abkürzung für »Cassette Operations Program«. Es erleichtert den Umgang mit der eigenen Programmsammlung auf Kassetten.

COP ist ein Hilfsprogramm zur Verwaltung von Programmsammlungen. Mit ihm können Sie Programme auf einer Datasette leichter und schneller finden und bei Bedarf ein Inhaltsverzeichnis einer Kassettenseite ausdrucken.

COP besteht aus zwei Teilen: Einem Hauptprogramm, das auf einer separaten Kassette gespeichert werden kann, und den Datensätzen für die einzelnen Kassettenseiten, die zur einfacheren Handhabung am Anfang jeder Kassettenseite abgelegt werden sollten.

Nach dem Starten des Programms mit »RUN« erscheint ein Menü auf dem Bildschirm, dessen einzelne Punkte hier erklärt werden:

1) Anfänge ermitteln

In diesem Programmpunkt können Sie die Programme einer Kassette katalogisieren. Zunächst geben Sie bitte den Namen und die Seite der Kassette ein. Danach wird nach den Titeln der einzelnen Programme gefragt. Hier können Sie auch persönliche Zusätze angeben wie zum Beispiel Zählerstand oder Fast-Tape, da der eingegebene Titel nicht mit dem Namen des Programms auf der Kassette übereinstimmen muß. Haben Sie alle Titel eingegeben, tippen Sie bitte ein »*« als Endekennzeichen ein.

Nun wird die zu katalogisierende Kassette im zurückgespulten Zustand eingelegt und immer bis zum jeweiligen Programm vorgespult. Dann drücken Sie die STOP-Taste. Zu diesem Zweck sollte man sich vorher den Stand des Zählwertes der einzelnen Programme notiert haben. Dies geschieht fortgesetzt bis zum letzten Programm, dann wird wieder zum Eingangs-menü zurückgesprungen.

2) Programm suchen

Nach der Anzeige des Kassettennamens und der Seite können die Programmnamen mit der Leertaste durchgeblättert werden. Hat man das zu suchende Programm gefunden, drückt man die Taste »RETURN«. Nun spult man die Kassette vor, der Computer stoppt den Spulvorgang automatisch kurz vor dem Programm. Drücken Sie nun »Stop« und »Play« an der Datasette, und das gewünschte Programm wird automatisch geladen. COP wird dabei gelöscht.

3) Zeiten abspeichern

In diesem Programmteil werden alle erstellten Daten (Kassetten- und Programmnamen und Zeit zum Stoppen) in einem Datensatz gespeichert. Dies kann zum Beispiel am Anfang der zu bearbeitenden Kassette geschehen. Dann sollte man das erste Programm allerdings nicht direkt am Anfang des Bandes speichern, da es dann den Datensatz überschreibt oder umgekehrt. Am praktikabelsten hat sich ein Vorspulen bis zum Zählwerksstand »30« erwiesen, bevor man das erste Programm speichert.

4) Zeiten laden

Hier werden die unter Punkt 3 abgespeicherten Datensätze wieder geladen.

5) Verzeichnis ausdrucken

Zunächst erscheinen die ersten 14 der auf der Kassette befindlichen Programme auf dem Bildschirm. Mit der Leertaste kann zu den nächsten 14 weitergeblättert werden. Wurden alle Programme aufgelistet, kann man sich das komplette Ver-

zeichnis auf dem Plotter 1520 ausgeben lassen. Eine Anpassung an andere Drucker dürfte überhaupt kein Problem sein, da keine druckerspezifischen Befehle verwendet werden. Einzig und allein der OPEN-Befehl in Zeile 5160 muß entsprechend des vorhandenen Druckers geändert werden.

Noch einige Hinweise: Das Programm ist auf 30 Titel pro Kassette beschränkt, mehr sind auf einer Kassette kaum sinnvoll. »Turbo-Tape«-Benutzer müssen noch die Zeile 2200 ändern. Dort wird das Eintippen der Tastenkombination »SHIFT/LOAD« simuliert, um das angewählte Programm zu laden, was mit manchen »Turbo-Tape«-Versionen nicht funktioniert. Am allereinfachsten wird dort nur der END-Befehl eingesetzt und der LOAD-Befehl dann von Hand eingegeben.

Für diejenigen, die das Programm verändern wollen, hier noch eine Variablenliste: (Hubertus Luhmann/bs)

Variablenliste:

| | |
|-----------|--|
| A\$ | Eingabe über GET |
| C\$ | Kassettensnummer |
| CS\$ | Kassettenseite |
| PP\$(Q) | laufende Programmtitel |
| Q | Zählvariable |
| P\$/P1\$ | Play/Play löschen |
| F\$/F1\$ | FFwd/FFwd löschen |
| S\$/S1\$ | Stop/Stop löschen |
| R\$/R1\$ | Record/Record löschen |
| Z\$/Z1\$ | Rewind/Rewind löschen |
| PR\$/PL\$ | Anzeige »Programm:«/PR\$ löschen |
| Q1 | Höchste Anzahl Programme |
| Q2 | Höchste Anzahl Programme und Sternchen (Q2=Q1+1) |
| TI\$(Q) | Zeitvariable |
| T1\$(Q) | Zwischenspeicher der Zeitvariablen |

```

10 DIM TI$(30):DIM PP$(30):DIM T1$(30) <058>
20 Q=0:T1$=TI$ <075>
30 PRINT CHR$(147):POKE 53280,0:POKE 53281,0 <047>
40 REM ***** MENUE ***** <050>
50 PRINT TAB(8)CHR$(18) " CASSETTEN - DIREC <173>
    TORY ";CHR$(146)
60 PRINT:PRINT:PRINT:PRINT:PRINT"AUS <037>
    WAHLMOEGlichkeiten:"
70 PRINT:PRINT:PRINT:PRINT"1 - ANFAENGE (SS <014>
    PACE)ERMITTELN - 1"
80 PRINT:PRINT"2 - PROGRAMM (8SPACE)SUCHEN <038>
    - 2"
90 PRINT:PRINT"3 - ZEITEN (5SPACE)ABSPEICHE <151>
    RN - 3"
100 PRINT:PRINT"4 - ZEITEN (8SPACE)EINLADEN <137>
    - 4"
110 PRINT:PRINT"5 - VERZEICHNIS AUSDRUCKEN <026>
    - 5"
120 PRINT:PRINT:PRINT:PRINT"BITTE TASTE DR <041>
    UECKEN !"
130 GET A$:IF A$="" THEN 130 <195>
140 ON VAL (A$)GOTO 1000,2000,3000,4000,500 <052>
    0 <126>
150 GOTO 130 <040>
200 REM *** UNTERPROGRAMME ***** <233>
210 REM * TASTATUR ZEICHNEN * <083>
220 PRINT "
230 PRINT" {CLR,SPACE} ***** <064>
    *****"
240 FOR Q=1 TO 13 <197>
250 PRINT" _{36SPACE}_ " <145>
260 NEXT Q <154>
270 PRINT" *****R*****R*****R*****R <138>
    *****"
280 PRINT" _ RECORD_PLAY_REWIND_FFWD_STOP_ <087>
    EJECT _
290 PRINT" ***** <117>
    *****"
300 FOR Q=1 TO 4 <082>
310 PRINT" _{7SPACE}__{4SPACE}__{6SPACE}__{4S <187>
    PACE}__{4SPACE}__{6SPACE}_
320 NEXT Q <214>

```

Das Listing zu »COP«. Bitte beachten Sie die Eingabehinweise auf Seite 6.

```

330 PRINT " *****E*****E*****E*****E*****E"
      *****X"
340 R$=" {HOME,19DOWN,3RIGHT}>↑↑↑↑↑↑"
350 R1$=" {HOME,19DOWN,3RIGHT,6SPACE}"
360 P$=" {HOME,19DOWN,10RIGHT}>↑↑↑↑↑"
370 P1$=" {HOME,19DOWN,10RIGHT,4SPACE}"
374 Z$=" {HOME,19DOWN,15RIGHT}>↑↑↑↑↑↑"
376 Z1$=" {HOME,19DOWN,15RIGHT,6SPACE}"
380 F$=" {HOME,19DOWN,22RIGHT}>↑↑↑↑↑"
390 F1$=" {HOME,19DOWN,22RIGHT,4SPACE}"
400 S$=" {HOME,19DOWN,27RIGHT}>↑↑↑↑↑"
410 S1$=" {HOME,19DOWN,27RIGHT,4SPACE}"
420 PRINT " {HOME,2DOWN,3RIGHT}>CASSETTE: ";C$
430 PRINT " {HOME,3DOWN,3RIGHT,3SPACE}>SEITE:
      ";CS$
440 PR$=" {HOME,6DOWN,3RIGHT}>PROGRAMM:"
445 PL$=" {HOME,6DOWN,3RIGHT,35SPACE}"
450 RETURN
460 REM UNTERPROGRAMM ZEIT
470 TI$=T1$
480 RETURN
500 T1$=TI$
510 RETURN
1000 REM *** ANFAENGE ERMITTELN *****
1010 PRINT CHR$(18)" {CLR,SPACE}>ANFAENGE ER
      MITTELN ";CHR$(146)
1020 PRINT:PRINT:INPUT"WELCHE CASSET
      TE ";C$
1030 INPUT"WELCHE {4SPACE}>SEITE ";CS$
1040 PRINT:PRINT"BITTE GEBEN SIE DIE PROGR
      AMME EIN:"
1050 Q=Q+1
1060 PRINT"PROGRAMM";Q;"="::INPUT PP$(Q)
1070 IF PP$(Q)="*"THEN 1090
1080 GOTO 1050
1090 REM *** ZEIT ERMITTELN *****
1200 PRINT " {CLR}>":Q1=Q-1
1210 GOSUB 200
1220 IF (PEEK(1) AND 16)=16 THEN 1250
1230 IF (PEEK(1) AND 16)=0 THEN PRINT S$
1240 WAIT 1,16,255-16:PRINT S1$
1250 FOR Q=1 TO Q1
1260 PRINT PL$:PRINT PR$:PP$(Q)
1270 PRINT F$
1275 GET A$
1280 IF (PEEK(1) AND 16)=0 THEN GOTO 1290
1285 IF A$=""THEN 1275
1290 IF Q=1 THEN TI$="000000":IF Q=1 THEN
      TI$="000000"
1291 GOSUB 460
1292 PRINT F1$:PRINT S$
1295 GET A$
1300 IF (PEEK(1) AND 16)=16 THEN 1310
1305 IF A$=""THEN 1295
1310 GOSUB 500
1315 TI$(Q)=T1$
1320 PRINT S1$:NEXT Q
1330 PRINT Z$:PRINT PL$:PRINT PR$: "ENDE"
1340 GET A$
1350 IF (PEEK(1) AND 16)=0 THEN GOTO 1365
1360 IF A$=""THEN 1340
1365 PRINT Z1$:PRINT S$
1370 GET A$
1380 IF (PEEK(1) AND 16)=16 THEN 1400
1390 IF A$=""THEN 1370
1400 GOTO 30
2000 REM *** PROGRAMM SUCHEN *****
2010 GOSUB 200:Q=0
2020 Q=Q+1:IF Q=Q1+1 THEN 2075
2030 PRINT PL$:PRINT PR$:PP$(Q)
2040 GET A$:IF A$=""THEN 2040
2050 IF A$=CHR$(32)THEN 2020
2060 IF A$=CHR$(13)THEN 2080
2070 GOTO 2040
2075 Q=0:GOTO 2020
2080 PRINT F$
2090 GET A$
2100 IF (PEEK(1) AND 16)=0 THEN 2115
2110 IF A$=""THEN 2090
2115 TI$(Q)=TI$(Q)
2120 TI$="000000":PRINT F1$
2130 IF TI$(Q)<>TI$THEN 2130
2150 POKE 192,1:POKE 1,PEEK(1) OR 32
2160 PRINT S$
2170 GET A$
2180 IF (PEEK(1) AND 16)=16 THEN 2200
2190 IF A$=""THEN 2170
2200 PRINT CHR$(147):POKE 198,1:POKE 631,1
      31:END
3000 REM *** ZEITEN ABSPEICHERN *****
3010 GOSUB 200
3020 PRINT R$:PRINT K$
3030 PRINT P$
3040 PRINT:PRINT:Q=0:Q2=Q1+1
3050 OPEN 1,1,1,"DIRECTORY"
3055 PRINT#1,C$:PRINT#1,CS$:PRINT#1,Q2:PRI
      NT#1,Q1
3060 FOR Q=Q+1 TO Q2
3070 PRINT#1,PP$(Q)
3080 PRINT#1,TI$(Q)
3090 NEXT Q
3100 CLOSE 1
3110 PRINT S$:PRINT P1$:PRINT R1$
3120 GET A$
3130 IF (PEEK(1) AND 16)=16 THEN 3150
3140 IF A$=""THEN 3120
3150 GOTO 30
4000 REM *** ZEITEN EINLADEN *****
4010 GOSUB 200
4030 PRINT P$
4040 PRINT:Q=0
4050 OPEN 1,1,0,"DIRECTORY"
4055 INPUT#1,C$:INPUT#1,CS$:INPUT#1,Q2:INP
      UT#1,Q1
4060 Q=Q+1
4065 IF Q=Q2 THEN 4100
4070 INPUT#1,PP$(Q)
4080 INPUT#1,TI$(Q)
4090 GOTO 4060
4100 CLOSE 1
4110 PRINT S$:PRINT P1$
4120 GET A$
4130 IF (PEEK(1) AND 16)=16 THEN 4150
4140 IF A$=""THEN 4120
4150 GOTO 30
5000 REM *** VERZEICHNIS AUSDRUCKEN ***
5010 PRINT CHR$(147):Q=0
5020 PRINT TAB(10)CHR$(18)" PROGRAMMVERZEI
      CHNIS ";CHR$(146)
5030 PRINT:PRINT"CASSETTE: ";C$
5040 PRINT "{3SPACE}>SEITE: ";CS$:PRINT
      Q=Q+1:IF Q=Q1+1 THEN 5100
5055 IF Q=15 THEN GOTO 5080
5060 PRINT"PROGRAMM ";Q;"=":PP$(Q)
5070 GOTO 5050
5080 GET A$:IF A$=""THEN 5080
5090 IF A$=CHR$(32)THEN PRINT CHR$(19):PRI
      NT:PRINT:PRINT:PRINT:Q=15:GOTO
      5060
5100 PRINT"*****"
5110 PRINT"A = AUSDRUCK {6SPACE}>M = MENUE"
5120 GET A$:IF A$=""THEN 5120
5130 IF A$="M"THEN 30
5140 IF A$="A"THEN 5160
5150 GOTO 5120
5160 OPEN 1,6
5170 PRINT#1,"***** V E R Z E I C H N I
      S *****"
5180 Q=0:PRINT#1:PRINT#1
5190 PRINT#1,"CASSETTE: ";C$
5200 PRINT#1," {3SPACE}>SEITE: ";CS$
5210 PRINT#1
5220 Q=Q+1:IF Q=Q1+1 THEN 5250
5230 PRINT#1,"PROGRAMM ";Q;"=":PP$(Q)
5240 GOTO 5220
5250 CLOSE 1
5260 GOTO 30

```

So stellt man die Datasette ein!

Die Fehlermeldung »LOAD ERROR« weist meistens auf eine defekte Kassette oder einen dejustierten Tonkopf hin. Dieser läßt sich allerdings mit einem Schraubenzieher einstellen.

Wer mit seiner Datasette Programme eines anderen Datenset-Besitzers laden will, der erlebt nach Eingabe des LOAD-Befehls oft die bizarrsten Dinge. Neben einem lapidaren »LOAD ERROR«, kann man andere Fehlermeldungen wie »OUT OF MEMORY ERROR« erhalten, aber auch Meldungen wie »FOUND A!.\$Y&«, (meist folgt dann innerhalb von Sekunden ein »LOAD ERROR«). Wenn es ganz schlimm kommt, stürzt der Computer auch noch ab und läßt sich nur mit einem Reset zu einem weiteren Ladeversuch bewegen, der dann allerdings ähnlich effektiv endet.

Hier gibt es mehrere mögliche Fehlerursachen: die Kassette ist defekt, der Tonkopf ist verstellt oder die elektromagnetischen Störstrahlungen eines Fernsehers oder Monitors verhindern einen einwandfreien Betrieb. Das letzte Problem läßt sich lösen, wenn man die Datasette bis auf maximale Kabellänge vom Fernseher während dem Laden entfernt. Sollte die Kassette defekt sein, hilft nur eine Sicherheitskopie. Am häufigsten tritt allerdings der zweite Grund, ein dejustierter Tonkopf, auf. Im folgenden soll beschrieben werden, wie man seinen Tonkopf kurzzeitig auf ein anderes Programm einstellen kann. Eine komplette, einwandfreie Einstellung läßt sich allerdings nur mit einem kleineren Bastelaufwand erreichen. Wir werden im 64'er, Ausgabe 10/85, eine Schaltung vorstellen, mit der die Datasette auf jede Kassette optimal justiert werden kann. Will man aber nur ein Programm laden, das man aufgrund einer falschen Einstellung leider nicht laden kann, reicht das folgende Verfahren aus:

Versuchen Sie, das Programm zu laden. Sollte der Computer überhaupt nichts auf der Kassette finden (Bildschirm bleibt hellblau, Datasette stoppt nicht, keine »FOUND ...«-Meldung), dann sollten Sie vorerst lieber auf das Programm verzichten. Die Tonkopfeinstellung, mit der dieses Programm gespeichert wurde, und die ihrer Datasette unterscheiden sich dann so erheblich, daß es einerseits sehr lange dauert, den Tonkopf ohne Zusatzhardware einzustellen, Sie andererseits aber mindestens die gleiche Zeit brauchen, ihre eigene Tonkopfeinstellung wiederzufinden.

Sollte der Computer etwas auf dem Band finden, dann läßt sich die Dejustierung des Kopfes in drei Klassen einteilen:

- Der Computer meldet sich mit »FOUND filename«, lädt einige Sekunden bis Minuten und meldet dann einen »LOAD ERROR«: Entweder ist der Tonkopf nur sehr leicht verstellt oder die Kassette ist defekt.
- Der Computer meldet sich mit »FOUND filename« und sofort mit »LOAD ERROR« oder »OUT OF MEMORY ERROR« oder einer anderen Fehlermeldung: Der Tonkopf ist etwas mehr verstellt als im ersten Fall, allerdings immer noch nicht sehr.

— Der Computer meldet sich mit »FOUND irgendwas«, wobei »irgendwas« meist aus Grafikzeichen und Farbumschaltungen besteht, meldet sofort oder später einen Fehler oder stürzt irgendwann ab: Der Tonkopf ist mittelstark verstellt und dürfte sich ohne Zusatzhardware noch richten lassen.

Bei der Tonkopfeinstellung tut man sich erheblich leichter, wenn man die Datasette aufschraubt. Lösen Sie also den Stecker der Datasette vom Computer, nehmen Sie eine in der Datasette liegende Kassette heraus und lösen Sie die vier Schrauben an der Unterseite.

Nehmen Sie dann das Oberteil des Gehäuses ab. Ziemlich genau in der Mitte der Mechanik befindet sich der Tonkopf. Er ist mit zwei Schrauben an der Mechanik befestigt (siehe Bild). Mit der linken Schraube läßt sich die Einstellung des Tonkopfes vornehmen. Drehen Sie niemals an der rechten Schraube!

Bevor Sie nun wild an der linken Schraube herumkurbeln, sollten Sie sich die jetzige Position merken. Die Schraube hat einen langen und einen kurzen Schlitz. Merken Sie sich die Position des langen Schlitzes am besten wie die Zeiger einer Uhr. Steht der beispielsweise senkrecht, kann man das als zwölf Uhr interpretieren. Vergessen Sie diese Einstellung nicht, sonst müssen Sie nachher stundenlang herumprobieren, bis Sie Ihre eigenen Kassetten laden können.

Legen Sie nun die Kassette mit dem Programm, das Sie laden wollen, ein. Verdrehen Sie die linke Kopfschraube ein wenig (maximal um ein bis zwei »Stunden«). Es ist egal in welche Richtung Sie drehen, Sie sollten sie sich nur merken. Machen Sie wieder einen Ladeversuch. Es gibt jetzt drei Möglichkeiten:

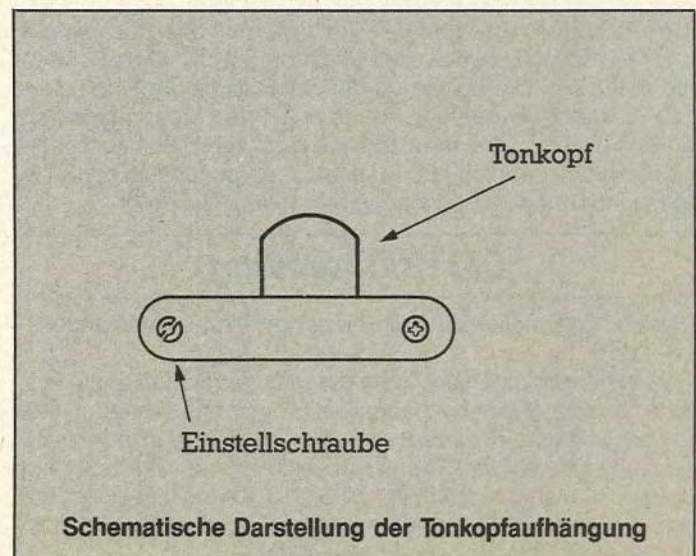
— Es hat sich überhaupt nichts geändert: Verdrehen Sie die Schraube wieder um ein bis zwei »Stunden« in dieselbe Richtung und machen Sie einen erneuten Ladeversuch.

— Der Fehler ist, gemäß der oben gemachten Abstufungen, kleiner geworden: Verdrehen Sie die Schraube wieder um ein bis zwei »Stunden« in dieselbe Richtung und machen Sie einen erneuten Ladeversuch.

— Der Fehler ist, gemäß der oben gemachten Abstufung, größer geworden oder der Computer findet gar nichts mehr: Drehen Sie in die Ausgangsstellung zurück und probieren Sie es nochmal in der anderen Richtung.

Sollten Sie irgendwann das Programm fehlerfrei laden können: Herzlichen Glückwunsch! Drehen Sie nun die Schraube wieder in ihre Ausgangsstellung zurück und speichern Sie das Programm.

Ein letztes Mal sei hier auf die kleine Schaltung in Ausgabe 10/85 verwiesen, mit der sich jede Datasette exakt auf jede noch so verstellt aufgenommene Kassette in wenigen Minuten einstellen läßt. (bs)



Turbo Tape de Luxe

Machen Sie Ihrer Datasette Beine: Mit diesem Programm wird sie sogar etwas schneller als das Floppy-Laufwerk.

Datasettenbesitzer müssen eigentlich gemütliche Leute sein. Bis zu zwanzig Minuten kann es nämlich dauern, wenn man ein umfangreiches Programm laden will. Viele Leute haben deswegen schon schnelle Lade- und Speicherroutinen geschrieben, die die Datasette um Faktoren zwischen 5 und 15 beschleunigen. Am bekanntesten und verbreitetsten ist das Programm »Turbo Tape«.

Das hier abgedruckte Programm »Turbo Tape de Luxe« (siehe Listing) zeichnet sich vor anderen Schnelladesystemen durch folgende Punkte aus:

— Es ist kompatibel zu »Turbo Tape«, kann also mit »Turbo Tape« gespeicherte Programme lesen. Daraus folgt, daß der Kassettenbetrieb um den Faktor 10 (wie bei »Turbo Tape«) beschleunigt wurde. Das ist immerhin sogar etwas schneller als das Floppy-Laufwerk 1541 (ohne Hypra-Load).

— Es arbeitet nicht mit neuen Basic-Befehlen, sondern mit einem Eingabemenü, so daß keine Konflikte mit Basic-Erweiterungen entstehen können.

— Das Basic-ROM kann während des Betriebs abgeschaltet werden, so daß auch Programme, die teilweise unter dem Basic-ROM liegen (lange Spiele und ähnliches), gespeichert werden können.

— Das Programm schützt sich selbst vor dem Überladen durch andere Programme.

— Es können beliebige RAM-Bereiche gespeichert werden.

— Programme können codiert und decodiert werden, damit kein Unbefugter Zugriff auf die Programme hat.

— Mit Hilfe des SMON kann das Programm in beliebige Speicherbereiche geschoben und sogar in ein EPROM gebrannt werden.

Nach dem Abtippen mit dem MSE und Speichern des Programms, kann es mit »SYS 49152« gestartet werden. Vorher sollten Sie jedoch NEW eingeben. Das Hauptmenü erscheint dann sofort und wartet auf eine Eingabe. Mit der Taste »Q« kann man dieses Menü wieder verlassen und dann später mit dem Befehl »ET« wieder ins Menü zurückkehren. Damit dies funktioniert, wird allerdings ein Basic-Vektor verbogen, der von vielen Basic-Erweiterungen benutzt wird und somit eine Zusammenarbeit mit »Turbo Tape de Luxe« völlig unmöglich macht. Soll dieser Vektor nicht verbogen werden, muß »Turbo Tape de Luxe« mit »SYS 49164« gestartet werden. An den Menüfunktionen hat sich nichts geändert, allerdings kann nach dem Verlassen mit »Q« nicht mehr mit »ET« ins Menü zurückgesprungen werden, sondern nur mit dem angegebenen SYS-Befehl.

Die Kommandos

Doch nun zur Erklärung der einzelnen Menü-Kommandos:

L — Load

Nach dem Drücken der Taste »L« wird vom Benutzer die Eingabe des Namens des zu ladenden Programmes verlangt. Soll das nächstbeste Programm geladen werden, darf der Name weggelassen werden. Ebenso ist eine Angabe mit Abkürzung (Joker) (*«) erlaubt (Beispiel: »TUR*«). Da bei fast jeder Programmoption ein Filename angegeben werden muß, wird im einzelnen nicht mehr darauf hingewiesen.

Nachdem das Programm gefunden wurde, werden Name und Adressen des Programmes angezeigt und auf einen Druck auf die Commodore-Taste gewartet. Wird diese Taste nicht gedrückt, wird nach zirka acht Sekunden der Ladevorgang fortgesetzt.

P — Proload

Mit dieser Funktion können sowohl Basic-, wie auch Maschinensprache-Programme geladen und automatisch gestartet werden. Ist das eingeladene Programm ein Basic-Programm, wird der »RUN«-Befehl simuliert. Ist das geladene Programm hingegen ein Maschinen-Programm, wird es nach dem Laden an der Anfangsadresse gestartet. Dies funktioniert nicht bei allen Maschinen-Programmen!

V — Verify

Mit Verify kann ein abgespeichertes Programm mit dem im Speicher befindlichen verglichen werden.

S — Save

Speichert das im Speicher vorhandene Basic-Programm.

A — Allsave

Speichert beliebige RAM-Bereiche ab. Die Start- und Endadresse müssen in hexadezimaler Form angegeben werden. Beim Laden wird das Programm wieder an diese Adressen zurückgeladen.

O — Original

Speichert RAM-Bereiche so ab, daß sie später an andere Adressen geladen werden. Als erstes müssen Sie die Adressen angeben, an die das Programm später geladen werden soll. Danach geben Sie die Adressen an, an denen sich das Programm jetzt befindet. Zum Schluß folgt, wie bei allen Save-Befehlen, der Filename.

R — Renew

Holt ein durch »NEW« oder einen Reset gelöscht Basic-Programm wieder zurück.

K — Kill

Zerstört »Turbo Tape de Luxe« und »biegt« die Vektoren wieder gerade. Zum Neuaufwurf muß es erneut geladen werden.

M — Monitor

Löscht den Bildschirm und startet einen Maschinensprache-Monitor, sofern dieser im Speicher vorhanden ist. Damit individuelle Monitore verwendet werden können, muß die Einsprungadresse in \$C52C/\$C52D angepaßt werden. Ist kein Monitor vorhanden, kann das Programm abstürzen!

C — Code

Codiert einen Speicherbereich nach Adressenangabe. Das so codierte Programm kann gespeichert und später wieder geladen und decodiert werden.

D — Decode

Decodiert einen bestimmten Speicherbereich.

Wer sich einen individuellen Code erstellen möchte, damit außer ihm niemand Zugriff auf die Programme hat, muß folgende zwei Speicherstellenpaare ändern:

CODE1 : C591 und C59C

CODE2 : C594 und C59A

In den beiden Speicherstellen eines Paares muß immer dasselbe Byte stehen!

Q — Quit

Verläßt das »Turbo Tape de Luxe«-Menü und springt zurück in Basic.

programm : t tape de luxe c000 c9c0

c000 : ad 8d c0 8d 08 03 ad 8e c2
c008 : c0 8d 09 03 a9 00 8d 20 43
c010 : d0 8d 21 d0 20 68 c2 20 9a
c018 : e4 ff f0 fb 48 20 76 c0 98
c020 : a9 9e 20 d2 ff 68 20 d2 e4
c028 : ff a2 10 dd 5d c6 f0 06 14
c030 : ca 10 f8 4c 0c 08 b6 b4 14
c038 : 8a 0a 65 b4 aa bd 48 c0 f2
c040 : 48 bd 47 c0 48 60 2c a3 d0
c048 : c2 c2 96 c3 2c ab c3 2c c6
c050 : b8 c3 2c bb c3 2c 25 c5 2a
c058 : 2c 2d c5 2c 3e c5 2c 44 5d
c060 : c5 2c b1 c5 2c b8 c5 2c 58
c068 : bf c5 2c ca c5 2c 0f c6 f6
c070 : 2c 00 00 2c 00 00 a0 11 c7
c078 : 2c a0 02 2c a0 0f 2c a0 6f
c080 : 16 a2 09 4c 0c e5 a2 14 56
c088 : a0 06 d0 f7 2c 8f c0 20 e1
c090 : 73 00 f0 04 c9 5c 0f 03 09
c098 : 4c e7 a7 20 73 00 c9 54 cd
c0a0 : f0 03 4c 08 af 4c 0c c0 35
c0a8 : 49 ff 4c 0c e1 a2 00 bd 34
c0b0 : 6f c7 f0 06 20 a8 c0 e8 1c
c0b8 : d0 f5 bd 6f c8 f0 06 20 4c
c0c0 : a8 c0 e8 d0 f5 bd 6f c9 bb
c0c8 : f0 06 20 a8 c0 e8 d0 f5 5b
c0d0 : 60 8a 48 20 86 c0 a9 9e de
c0d8 : 20 d2 ff 68 aa bd 73 c6 62
c0e0 : f0 06 20 a8 c0 e8 d0 f5 73
c0e8 : 60 a2 38 20 d1 c0 a9 02 79
c0f0 : 8d 8b 02 a5 cb c9 3c f0 56
c0f8 : 02 d0 f3 4c 0c a0 20 70 1b
c100 : 2c a2 8c a9 c4 48 20 d1 88
c108 : c0 68 aa 4c 70 c1 a2 8c e9
c110 : a9 a8 d0 f1 a9 08 85 a1 b4
c118 : 20 e0 e4 4c e9 c0 a2 e0 5c
c120 : 20 d1 c0 4c ee c0 85 02 f1
c128 : 4a 4a 4a 4a 20 33 c1 a5 61
c130 : 02 29 0f c9 0a 30 06 69 d1
c138 : 36 20 ca f1 60 69 30 d0 23
c140 : f8 20 79 c0 a5 40 20 26 e8
c148 : c1 a5 3f 20 26 c1 a9 2d 21
c150 : 20 ca f1 a5 42 20 26 c1 48
c158 : a5 41 4c 26 c1 20 7f c0 12
c160 : a2 00 bd 41 03 20 ca f1 da
c168 : e8 e0 10 d0 f5 4c 41 c1 29
c170 : 20 d1 c0 4c 14 c1 00 00 82
c178 : 00 00 00 00 00 2c c0 02 e1
c180 : ad 7e c1 85 bb ad 7f c1 38
c188 : 85 bc 60 20 79 c0 a2 00 b0
c190 : a9 20 20 ca f1 e8 e0 09 a7
c198 : d0 f8 60 20 80 c1 a9 a4 06
c1a0 : 20 ca f1 a0 00 84 b7 20 f9
c1a8 : 3e f1 f0 fb c9 93 f0 f7 87
c1b0 : c9 13 f0 f3 c9 11 f0 ef 86
c1b8 : c9 91 f0 eb c9 9d f0 e7 20
c1c0 : c9 1d f0 e3 c9 14 f0 1e 0e
c1c8 : c9 0f 02 2c a4 b7 91 bb 9f
c1d0 : 48 a9 9d 20 ca f1 68 20 76
c1d8 : ca f1 c4 ba f0 c7 a9 a4 60
c1e0 : 20 ca f1 c8 d0 bf a2 00 90
c1e8 : bd 6d c6 f0 06 20 ca f1 9c
c1f0 : e8 d0 f5 c6 b7 4c a7 c1 96
c1f8 : a4 b7 c4 ba f0 0a a9 9d 41
c200 : 20 ca f1 a9 20 20 ca f1 49
c208 : 60 20 80 c1 a0 00 a9 20 c2
c210 : 91 bb c8 c0 10 d0 f9 60 f9
c218 : 20 79 c0 a9 08 85 ba d0 93
c220 : 07 20 7f c0 a9 0f 85 ba ce
c228 : 20 09 c2 4c 9b c1 20 3f ce
c230 : c2 0a 0a 0a 0a 85 02 c8 22
c238 : 20 3f c2 18 65 02 60 b1 f7
c240 : bb c9 3a 30 04 38 e9 37 8d
c248 : 60 38 e9 30 60 a0 00 20 90
c250 : 2e c2 85 42 c8 20 2e c2 55
c258 : 85 41 c8 c8 20 2e c2 85 53
c260 : 44 c8 20 2e c2 85 43 00 fc
c268 : 20 ad c0 20 7c c0 a9 01 09
c270 : 2c 01 00 f0 05 a9 31 4c 36
c278 : ca f1 a9 30 d0 f9 a9 00 2f
c280 : a8 aa 4c ba ff a2 04 b5 78
c288 : 40 95 ab ca d0 f9 60 20 76
c290 : 18 c2 4c 4d c2 20 8f c2 b7
c298 : 20 85 c2 20 8b c1 20 8f 96
c2a0 : c2 4c 21 c2 a2 05 86 ab ed
c2a8 : 20 9e c2 20 85 c2 20 7e b8
c2b0 : c2 a2 1c 20 d1 c0 20 d0 14
c2b8 : f8 20 2e f8 d0 f5 20 83 af
c2c0 : c3 20 63 c3 a5 b9 18 69 40
c2c8 : 01 ca 20 35 c3 a2 08 b9 c2
c2d0 : ac 00 20 35 c3 a2 06 c8 26
c2d8 : c0 05 d0 f3 a0 00 a2 04 6a
c2e0 : b1 bb c4 b7 90 03 a9 20 9f
c2e8 : ca 20 35 c3 a2 05 c8 c0 7f
c2f0 : bb d0 ed a9 02 85 ab 20 ff
c2f8 : 63 c3 98 20 35 c3 84 27 9a

c300 : a2 07 b1 41 20 35 c3 a2 ba
c308 : c3 e6 41 d0 04 e6 42 ca ff
c310 : ca a5 41 c5 43 a5 42 ea ec
c318 : 44 90 e7 a5 d7 20 35 c3 2e
c320 : a2 07 88 d0 f6 c8 84 c0 cb
c328 : 58 18 a9 00 8d a0 02 20 1d
c330 : 93 fc 4c e9 c0 85 bd 45 4b
c338 : d7 85 d7 a9 08 85 a3 06 44
c340 : bd a5 01 29 f7 20 56 c3 97
c348 : a2 11 09 08 20 56 c3 a2 bf
c350 : 0e c6 a3 d0 ea 60 ca d0 43
c358 : fd 90 05 a2 0b ca d0 fd 79
c360 : 85 01 60 a0 00 a9 02 20 28
c368 : 35 c3 a2 07 88 c0 09 d0 5d
c370 : f4 a2 05 c6 ab d0 ee 98 fe
c378 : 20 35 c3 a2 07 88 d0 f7 60
c380 : ca ca 60 a0 00 84 c0 ad 5e
c388 : 11 d0 29 ef 8d 11 d0 ca 84
c390 : d0 fd 88 d0 fa 78 60 a2 d5
c398 : 05 86 ab a2 04 b5 2a 95 e1
c3a0 : ab 95 40 ca d0 f7 20 21 0f
c3a8 : c2 4c ae c2 a2 05 86 ab 58
c3b0 : 20 95 c2 4c ae c2 4c ae 64
c3b8 : c2 a2 00 2c a2 01 86 0a b1
c3c0 : 86 93 20 21 c2 20 ee c3 ac
c3c8 : a5 c3 85 2d a5 c4 85 2e 49
c3d0 : a5 0a f0 0d 20 b7 ff 29 6a
c3d8 : bf d0 03 4c 0e c1 4c 01 6c
c3e0 : c1 20 b7 ff 29 bf f0 03 f9
c3e8 : 4c fe c0 4c e9 c0 20 76 7f
c3f0 : c4 a5 ab c9 02 f0 04 c9 f6
c3f8 : 01 d0 f3 ad 3c 03 85 c3 8d
c400 : 85 3f ad 3d 03 85 c4 85 b2
c408 : 40 ad 3e 03 85 a1 ad 3f a6
c410 : c3 85 42 20 5d c1 2c 00 ff
c418 : c0 2c ff cf ad 14 c4 c0 12
c420 : 3e 03 ad 15 c4 ed 3f 03 ad
c428 : b0 16 ad 17 c4 cd 3c 03 e3
c430 : ad 18 c4 ed 3d 03 90 c8 16
c438 : a2 54 20 d1 c0 4c 14 c1 89
c440 : a9 10 85 a1 20 e0 e4 a5 6f
c448 : cb c9 3f d0 03 4c c0 c0 26
c450 : a4 b7 f0 0f 88 b1 bb c9 86
c458 : 2a f0 05 d9 a1 03 d0 8e 03
c460 : 98 d0 f1 84 90 20 8c 33
c468 : a5 bd 45 d7 05 90 f0 04 d9
c470 : a5 ff 85 90 18 60 20 c6 1b
c478 : c4 c9 00 f0 f9 85 ab 20 fa
c480 : f4 c4 91 b2 c8 c0 c0 d0 c8
c488 : f6 f0 2e ea 20 c6 c4 20 6b
c490 : f4 c4 c4 93 d0 02 91 c3 75
c498 : d1 c3 f0 02 86 90 45 d7 79
c4a0 : 85 d7 e6 c3 d0 02 e6 c4 85
c4a8 : a5 c3 c5 41 a5 c4 e5 42 65
c4b0 : 90 dd 20 f4 c4 20 83 c3 b8
c4b8 : c8 84 c0 18 58 a9 00 8d e3
c4c0 : a0 02 4c 93 fc ea 20 1b c5
c4c8 : c5 20 83 c3 84 d7 a9 07 b2
c4d0 : 8d 06 dd a2 01 20 04 c5 d9
c4d8 : 26 bd a5 bd c9 02 d0 f5 da
c4e0 : a0 09 20 f4 c4 c9 02 f0 30
c4e8 : f9 c4 bd d0 e8 20 f4 c4 ba
c4f0 : 88 d0 f6 60 a9 08 85 a3 e2
c4f8 : 20 04 c5 26 bd c6 a3 d0 93
c500 : f7 a5 bd 60 a9 10 2c 0d 2b
c508 : dc f0 fb ad d0 dd 8e 07 19
c510 : dd 48 a9 19 8d 0f dd 68 38
c518 : 4a 4a 60 a2 00 20 d1 c0 be
c520 : 20 2e f8 d0 fb 60 a9 93 40
c528 : 20 ca f1 4c 00 90 a2 36 2f
c530 : a9 30 86 01 48 20 7c c0 ac
c538 : 68 20 ca f1 4c 17 c0 a2 67
c540 : 37 a9 31 d0 ed a5 2b a4 b4
c548 : 2c 85 22 84 23 a0 03 c8 25
c550 : b1 22 d0 fb c8 98 18 65 42
c558 : 22 a0 00 91 2b a5 23 69 3c
c560 : c0 c8 91 2b 88 a2 03 e6 06
c568 : 22 d0 02 e6 23 b1 22 d0 3a
c570 : f4 ca d0 f3 a5 22 69 02 91
c578 : 85 2d a5 23 69 00 85 2e 6b
c580 : 4c 1e c1 20 8f c2 a0 00 e1
c588 : b1 41 a6 ba 00 70 f0 08 40
c590 : 49 ff 18 69 20 4c 9d c5 72
c598 : 38 e9 20 49 ff 91 41 e6 55
c5a0 : 41 d0 02 e6 42 a5 43 c5 91
c5a8 : 41 a5 44 e5 42 b0 d7 4c 2b
c5b0 : 1e c1 a9 00 85 ba 4c 83 80
c5b8 : c5 a9 70 85 ba 4c 83 c5 c6
c5c0 : a9 93 20 ca f1 a2 f6 9a d9
c5c8 : 6c 00 03 ad 09 03 cd 8e a8
c5d0 : c0 d0 0a a9 e4 8d 08 03 91
c5d8 : a9 a7 8d 09 03 2c 00 c0 ec
c5e0 : 2c bf c5 ad dc c5 85 41 c7
c5e8 : ad df c5 85 42 ad e1 c5 4b
c5f0 : 85 43 ad e2 c5 85 44 a9 cb
c5f8 : 00 a0 00 91 41 e6 41 d0 6d

c600 : 02 e6 42 a5 43 c5 41 a5 6d
c608 : 44 e5 42 b0 ea 4c c0 c5 85
c610 : a2 00 2c a2 01 86 0a 86 8b
c618 : 93 20 21 c2 20 ee c3 a5 30
c620 : c3 85 2d a5 c4 85 2e a5 22
c628 : 0a f0 0d 20 b7 ff 29 bf 91
c630 : d0 03 4c 0e c1 4c 01 c1 5d
c638 : 20 b7 ff 29 bf f0 03 4c 81
c640 : fe c0 a9 93 20 ca f1 a5 e6
c648 : 3f a6 40 c5 2b d0 04 e4 b6
c650 : 2c f0 03 6c 3f 00 20 59 6a
c658 : a6 4c ae a7 ff 41 53 4f bb
c660 : 4c 56 4d 2d 2b 52 43 44 ab
c668 : 51 4b 50 00 00 9d 20 9d 1c
c670 : 9d a4 00 df df df df 98
c678 : af ad ba ac ac df af b3 32
c680 : be a6 df b0 b1 df ab be e5
c688 : af ba df df df df 00 af e4
c690 : ad ba ac ac df ad ba bc 2b
c698 : b0 ad bb df d9 df af b3 cc
c6a0 : be a6 df b0 b1 df ab be 05
c6a8 : af ba 00 df df df df ed
c6b0 : df df df af ad ba ac 29
c6b8 : df ac af be bc ba df df 92
c6c0 : df df df df df df 00 df 3f
c6c8 : df df df df b3 b0 be bb be
c6d0 : df b2 ba b2 b0 ad a6 df e0
c6d8 : ba ad ad b0 ad df df df 03
c6e0 : df df 00 df df df df e7
c6e8 : df df df df df b3 b0 be 86
c6f0 : bb df df df df df df cb
c6f8 : df df df df df df df 77
c700 : df df df df df df df a9 92
c708 : ba ad b6 b9 a6 df df df 26
c710 : df df df df df df df 0f
c718 : df df 00 e2 e2 e2 e2 da
c720 : e2 e2 e2 e2 e2 e2 e2 1f
c728 : e2 df b0 b4 df df df df f9
c730 : df df df df df df 00 e2 b5
c738 : e2 e2 e2 e2 e2 e2 e2 37
c740 : e2 e2 e2 e2 e2 df ba ad 1c
c748 : ad b0 ad df df df df df f1
c750 : df df 00 df df df df bb 0f
c758 : b0 b1 ba df d1 d1 d1 df 3e
c760 : af ad ba ac ac df ac af 06
c768 : be bc ba df df df 00 6c 05
c770 : 60 ed df df df df df df 26
c778 : aa df ad df df df df df d6
c780 : df ab df be df af df ba 75
c788 : df df bb df ba df df b3 d3
c790 : df aa df a7 df ba df df c4
c798 : df df df df df df df df 97
c7a0 : df df df df df df df df 9f
c7a8 : df df df df df df df df a7
c7b0 : df df df df df df df df af
c7b8 : df df df df df df df df b7
c7c0 : df df 61 df df df df df 1f
c7c8 : df a8 ad b6 ab ab ba b1 a4
c7d0 : df bd a6 df b5 b0 b7 be 71
c7d8 : b1 b1 ba ac df b8 b0 b3 94
c7e0 : b0 b2 bd ba b4 df df df 39
c7e8 : df df df df df df df df e7
c7f0 : df df df df b1 ba aa b2 aa 0e
c7f8 : ba b1 ac ab ba ad d0 bd 03
c800 : ad be aa b1 ac bc b7 af ce
c808 : ba b6 b8 df df df df df 84
c810 : df df df df f2 f2 f2 ac ac 57
c818 : df df be bb ad ba ac ac cb
c820 : ba df df 6d df df ed bd 9f
c828 : ad df bd ba 6d df df ed bd
c830 : df df df df df b9 b6 b3 01
c838 : ba b1 be b2 ba 4f df df bb
c840 : df df f2 df 4f 3c 3c 08
c848 : 3c 3c 3c 3c 3c 3c 51 df e4
c850 : df 4f 3c 4d 3c 51 df df 1d
c858 : 4f 3c 3c 3c 3c 3c 3c 6b
c860 : 3c 3c 3c 3c 3c 3c 3c 60
c868 : 3c 51 df df 3d df df df 53
c870 : df df df df df df df df e4
c878 : df 3d df 3d df 3d df df bc
c880 : 3d df df df df df df df dd
c888 : df df df df df df df df 87
c890 : df 3d df df 52 3c 3c 3c 72
c898 : 3c 3c 3c 3c 3c 3c 42 df f7
c8a0 : df 52 3c 4e 3c 42 df df 96
c8a8 : 52 3c 3c 3c 3c 3c 3c 6b
c8b0 : 3c 3c 3c 3c 3c 3c 3c b0
c8b8 : 3c 42 f2 f2 f2 c3 b3 6d
c8c0 : c1 b0 be bb df c5 df c3 33
c8c8 : a9 c1 ba ad b6 b9 a6 df 4a
c8d0 : df c5 df c3 af c1 ad b0 23
c8d8 : b3 b0 be bb df c5 df c3 3d

Listing zu »Turbo Tape de Luxe«. Beachten Sie bitte die Eingabehinweise auf Seite 8.

```

c8e0 : ae c1 aa b6 ab df c3 ac 12
c8e8 : c1 be a9 ba df c5 df c3 fd
c8f0 : be c1 b3 b3 ac be a9 ba cf
c8f8 : df c5 df c3 b0 c1 ad b6 67
c900 : b8 b1 be b3 df c5 df c3 ea
c908 : b4 c1 b6 b3 b3 df c3 bc 84
c910 : c1 b0 bb ba df c5 df c3 a3
c918 : bb c1 ba bc b0 bb ba df 8d
c920 : df c5 df c3 b2 c1 b0 b1 b1
c928 : b6 ab b0 ad df c5 df c3 c9
c930 : ad c1 ba b1 ba a8 f2 df 1f
c938 : df df df df ed df df df 18
c940 : df df ac a6 ac ab ba b2 87
c948 : df b0 ad bb ba ad ac d0 cf
c950 : ba ad ad b0 ad ac df df e2
c958 : df df df f2 df df df df b9
c960 : df 4f 3c 3c 3c 3c 3c 8d
c968 : 3c 3c 3c 3c 3c 3c 3c 68
c970 : 3c 3c 3c 3c 3c 3c 3c 70
c978 : 3c 3c 3c 3c 3c 3c 51 ee 32
c980 : 62 3d 6e f2 df df df df b7
c988 : df 3d f2 df df df df fb
c990 : 52 3c 3c 3c 3c 3c 3c a6
c998 : 3c 3c 3c 3c 3c 3c 3c 98
c9a0 : 3c 3c 3c 3c 3c 3c 3c a0
c9a8 : 3c 3c 3c 3c 42 00 00 6f
c9b0 : 00 ff 00 ff 00 ff 00 ff b0
c9b8 : 00 ff 00 ff 00 ff 00 ff b8
    
```

**Listing
»Turbo Tape
de Luxe«
(Schluß).
Beachten
Sie bitte die
Eingabeinweise
auf Seite 8.**

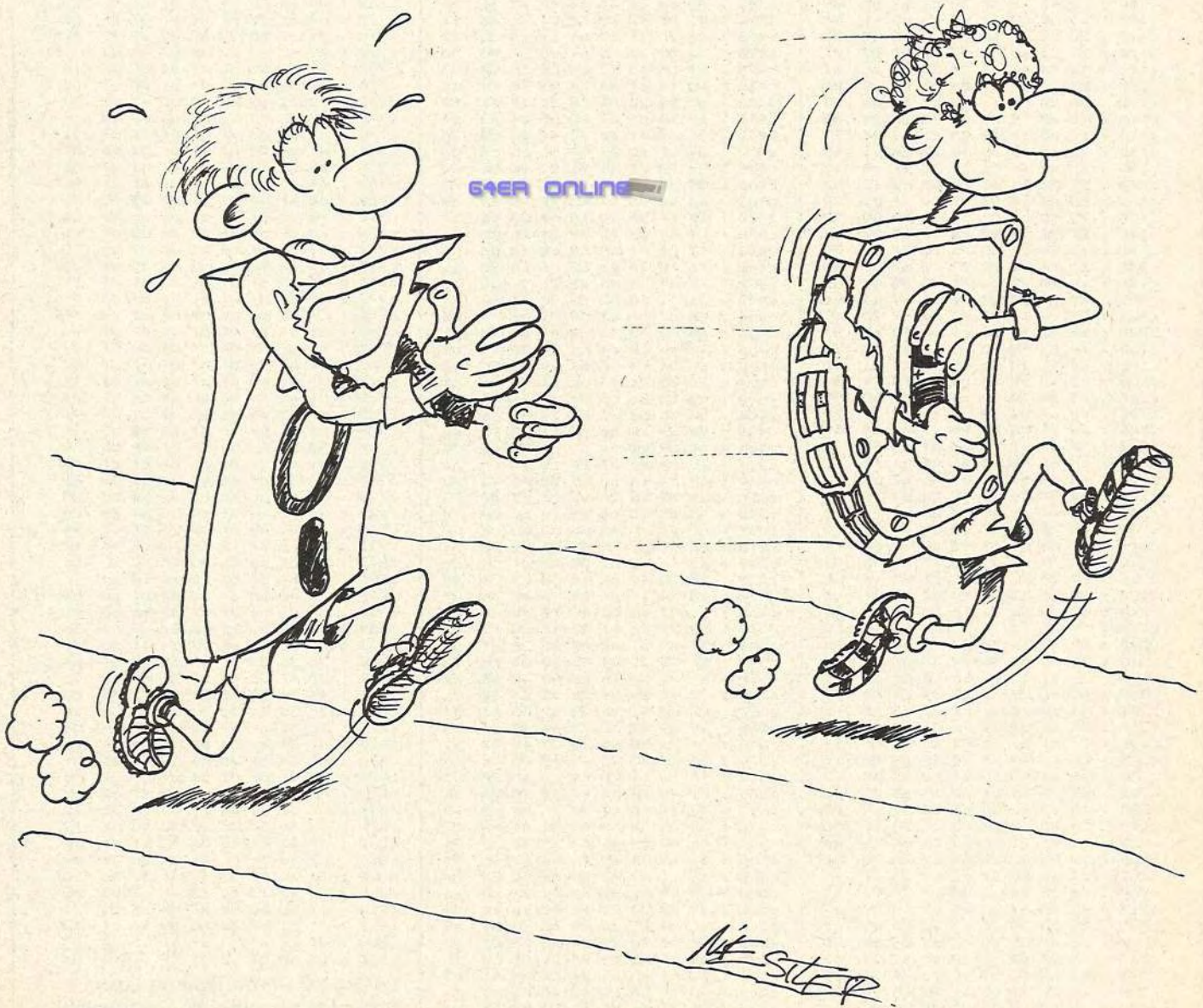
± — Basic-ROM ein-/ausschalten

Im »Turbo Tape de Luxe«-Menü kann das Basic-ROM (\$A000-\$BFFF) abgeschaltet werden, um Programme, Grafikbilder und anderes im RAM unter dem ROM aus diesem Bereich speichern zu können. Der derzeitige Zustand des Basic-ROMs wird in dem Kästchen »BR« angezeigt.

Die abgedruckte Version von »Turbo Tape de Luxe« liegt im Speicherbereich von \$C000 bis \$C9B0. Sie kann mit dem SMON und seinem »C«-Befehl in beliebige Speicherbereiche geschoben werden, da alle verwendeten Sprungtabellen als BIT-Befehle getarnt sind. Genaueres zur Verwendung des »C«-Befehls beim SMON entnehmen Sie bitte der Anleitung zum SMON selbst (64'er, Ausgabe 11/84 bis 2/85).

Wer will, kann »Turbo Tape de Luxe« nach dem Verschieben sogar in ein EPROM brennen.

Sollte sich »Turbo Tape de Luxe« einmal mit »Load Memory Error« melden, dann haben Sie versucht, ein Programm zu laden, das »Turbo Tape de Luxe« teilweise überschrieben und gelöscht und somit einen sauberen Systemabsturz heraufbeschworen hätte. Diese Sicherheitsabfrage funktioniert auch in etwaigen verschobenen Versionen. (J. Golombek/bs)



Floppy-Fehler ohne Rätsel

Wer kennt ihn nicht, den bekannten Vierzeiler, mit dem der Fehlerkanal des Disketten-Laufwerks abgefragt wird? Mit diesem Programm zeigt der Computer automatisch an, welcher Fehler aufgetreten ist.

Das Eintippen jenes berühmten Vierzeilers ist zwar ein einfaches, aber zeitraubendes Verfahren:

```
10 OPEN 15,8,15
20 INPUT # 15,A1,A2$,A3,A4
30 PRINT A1,A2$,A3,A4
40 END
```

Immerhin sind ungefähr 56 Tasten zu drücken. Meistens verbißt der Anwender außerdem die Zeilen wieder zu löschen. Eine Alternative besteht darin, das DOS 5.1 der Demodiskette zu laden. Doch stellte auch dies keine befriedigende Lösung dar. Es sind immer noch einige Tasten zu drücken. Eine bessere Lösung ist ein entsprechendes Maschinenprogramm (Listing). Der Fehlerkanal der Floppy wird selbständig vom Computer überwacht und bei Bedarf vollautomatisch eine Fehlermeldung ausgegeben.

Das Programm ist ganz in Maschinensprache geschrieben. Nach dem Betätigen einer beliebigen Taste werden Sie aufgefordert, die Startadresse einzugeben. Das Programm ist frei verschiebbar, die entsprechenden Änderungen werden vom Programm selbst durchgeführt. Zunächst werden Sie noch nichts bemerken, außer, daß sich der Computer mit einem READY wiedermeldet. Versuchen Sie aber nun einmal eine Fehlermeldung der Floppy zu erzeugen. Als Beispiel können Sie ein Programm von der Diskette laden, das es nicht gibt. Auf dem Bildschirm erhalten Sie sofort die entsprechende Fehlermeldung und die Aufforderung, eine Taste zu drücken. Die Funktionen des Computers sind um eine wichtige und nützliche Fähigkeit erweitert worden. Außerdem arbeitet das Programm innerhalb von Basic-Programmen ebenfalls zuverlässig. Bei einem Fehler wird ein Programm nicht unterbrochen. Als weitere Besonderheit ist zu erwähnen, daß das Programm auch mit anderen Hilfsprogrammen zusammenarbeitet. Sie können zum Beispiel Hypra-Load oder das DOS 5.1 weiterhin benutzen. Das Programm wird durch Drücken der Run/Stop/Restore-Taste unterbrochen. Ein erneuter Start ist jederzeit mit SYS (Startadresse) möglich.

Funktionsweise des Programmes

Zuerst werden die IRQ-Vektoren geändert. Der Computer arbeitet dann bei jedem Interrupt das Programm ab. Dann findet eine Verzögerung statt, damit der Fehlerkanal nicht bei jedem Interrupt abgefragt wird. Danach wird geprüft, ob die Diskettenstation arbeitet und wenn ja, wird die Routine verlassen. Als nächstes wird der Fehlerkanal abgefragt und geprüft, ob ein Fehler vorliegt. Ist dies der Fall, dann wird die augenblickliche Cursorposition zwischengespeichert und der Computer merkt sich den Inhalt der oberen drei Zeilen. Diese Zeilen werden dann gelöscht, der Cursor abgeschaltet, eine neue Schriftfarbe gewählt und die Fehlermeldung in reverser Schrift ausgegeben.

Anschließend wird die Aufforderung, eine Taste zu drücken, auf den Bildschirm geschrieben. Der Computer wartet dann so lange, bis eine Taste gedrückt wird. Dann wird der ursprüngli-

che Bildschirminhalt wieder hergestellt, die alte Bildschirmfarbe ins Register gePOKEt und der Cursor auf den alten Platz gebracht. Das Programm benutzt insgesamt elf Unterroutinen des Betriebssystems. Die Farbe, in der die Fehlermeldung ausgegeben werden soll, ist durch den Befehl POKE Startadresse+123, Farbcode veränderbar. Dem Programmierer ist mit diesem Programm ein sehr gutes Hilfsmittel in die Hand gegeben. (Georg Kramer/law)

```
0 REM ***** <131>
1 REM ** AUTOMATISCHE FEHLERMELDUNG ** <044>
2 REM ** VON DER FLOPPY DISK 1541 ** <055>
3 REM **-----** <046>
4 REM ** WRITTEN BY GEORG KRAMER ** <150>
5 REM ** AUF DEM MOORE 15 ** <234>
6 REM ** 4783 ANROECHTE ** <141>
7 REM ** TEL.: 02947/3511 ** <122>
9 REM ***** <140>
10 : <242>
20 PRINT CHR$(147) <049>
30 PRINT SPC(81)"CCCCCCCCCCCCCCCCCCCCCCCC <051>
CCCCCCCCCCCCI"
40 PRINT " B AUTO. FEHLERMELDUNG VON DER FL <133>
OPPY B"
50 PRINT " JCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC <112>
CCCCCK"
60 PRINT SPC(88)"WRITTEN BY GEORG KRAMER" <065>
70 PRINT SPC(93)"(C) JUNI 1985":PRINT <069>
80 FOR I=1 TO 40:PRINT"-";:NEXT <210>
90 PRINT SPC(44)"{RVSON,SPACE}ACHTUNG:{SPA <088>
CE,RVOFF,SPACE}FLOPPY EINSCHALTEN !"
100 PRINT SPC(94)"<< TASTE >>" <051>
110 POKE 198,0 <018>
120 GET A$:IF A$=""THEN 120 <121>
130 : <106>
140 PRINT CHR$(147):PRINT:PRINT <237>
150 INPUT "{SPSPACE,RVSON}STARTADRESSE : {SP <074>
ACE,RVOFF,SPACE}49152{7LEFT}";S
160 FOR I=0 TO 247:READ A:X=X+A <165>
170 POKE S+I,A:NEXT <095>
180 IF X<>32204 THEN PRINT"DATEN-FEHLER !" <028>
:END <047>
185 REM *** PROGRAMM ANPASSEN ***
190 POKE S+7,(S+13)/256 <013>
200 POKE S+2,(S+13)-PEEK(S+7)*256 <228>
202 POKE S+154,(S+218)/256 <152>
204 POKE S+153,(S+218)-PEEK(S+154)*256 <122>
210 SYS(S) <231>
220 PRINT CHR$(147):PRINT SPC(8)"{RVSON,3S <004>
PACE}PROGRAMM AKTIVIERT{4SPACE}"
230 PRINT SPC(8)"{RVSON,SPACE}START:{3SPAC <106>
E}SYS"S"{LEFT,6SPACE}"
240 PRINT SPC(8)"{RVSON,SPACE}ENDE : RUN-S <105>
TOP/RESTORE "
250 END <252>
1000 REM *** MASCHINENPROGRAMM *** <036>
1001 DATA 120,169,13,141,20,3,169,192,141, <109>
21,3,88,96,230,182,240,3,76,49,234
1002 DATA 165,144,201,66,240,8,165,144,201 <230>
,128,240,2,208,239,169,255,162,8
1003 DATA 160,15,32,186,255,169,0,32,189,2 <178>
55,32,192,255,162,255,32,198,255
1004 DATA 32,207,255,201,48,208,7,32,47,24 <142>
3,88,76,49,234,72,165,211,141,168
1005 DATA 2,165,214,141,178,2,162,0,189,0, <223>
4,157,132,3,232,224,120,208,245,162
1006 DATA 0,169,32,157,0,4,232,224,120,208 <160>
,246,32,102,229,169,207,240,252,169
1007 DATA 1,133,204,173,134,2,133,189,169, <194>
1,141,134,2,169,1,133,199,104,32
1008 DATA 210,255,32,207,255,32,210,255,16 <113>
5,144,201,64,208,244,32,47,243,162
1009 DATA 0,189,218,192,32,210,255,232,224 <120>
,30,208,245,32,135,234,165,203,201
1010 DATA 64,240,247,165,189,141,134,2,162 <084>
,0,162,0,189,132,3,157,0,4,232,224
1011 DATA 120,208,245,165,189,157,0,216,23 <153>
2,224,120,208,246,173,168,2,133,211
1012 DATA 173,178,2,133,214,32,16,229,76,4 <084>
9,234,18,32,60,60,32,66,73,84,84
1013 DATA 69,32,84,65,83,84,69,32,68,82,85 <015>
,69,67,75,69,78,32,62,62,32,146
```

Listing. »Automatische Fehlermeldung«. Bitte beachten Sie die Eingabhinweise auf Seite 6.

Diskmaster — Beherrschen Sie das 1541-Laufwerk!

Dieses Programm macht die Bedienung des Floppy-Laufwerks 1541 sicher und komfortabel. Alle Befehle, die man sich bislang mühsam aus dem Handbuch suchen mußte, stehen auf Knopfdruck parat. Außerdem gibt es die Möglichkeit, eine Reihe von kommerziellen Programmen mit eigenen Basic-Programmen zu kombinieren.

Das Programm Diskmaster (siehe Listing) ist ein Maschinensprache-Programm, das für den Commodore 64 geschrieben wurde, damit sich Floppy-geplagte Commodore-Besitzer nicht länger die Finger wund schreiben müssen, wenn sie zum Beispiel die Floppy initialisieren oder ein Programm laden wollen. Diskmaster muß mit dem MSE eingegeben werden.

Das Hauptmenü

Nachdem Diskmaster mit SYS 49152 gestartet wurde, meldet es sich mit dem Hauptmenü. Hier gibt es fünf Wahlmöglichkeiten:

1. Programm laden
2. Programm speichern
3. Diskette verwalten
4. Inhalt zeigen
5. Zurück zu Basic

Man kann nun einen der Menüpunkte anwählen, indem man eine der Tasten 1 bis 5 tippt. Der angewählte Menüpunkt erscheint dann revers auf dem Bildschirm. Die Wahl wird nun durch Drücken der RETURN-Taste bestätigt.

Programm laden

Das erste Untermenü bietet insgesamt vier Menüpunkte:

1. laden
2. laden und starten
3. merge (zusammenfügen)
4. zurück

Es gibt hier nun vier verschiedene Möglichkeiten, ein Programm zu laden. Unter Menüpunkt 1 wird ein Programm genauso geladen, wie man das vom Basic her bereits kennt, allerdings geht es hier etwas komfortabler. Man gibt einfach den Namen des gewünschten Programms ein, schließt die Eingabe mit RETURN ab, und der C 64 lädt das Programm. Vom Hauptmenü aus kann man dann wieder in das Basic zurück und das Programm RUN neu starten.

Menüpunkt 1 bietet im Prinzip nichts Neues. Die Punkte 2 und 3 hingegen stellen eine Erweiterung der Bedienungsmöglichkeiten gegenüber dem Standard dar. Unter Punkt 3 wird das angegebene Programm an das bereits im Speicher befindliche Basic-Programm angehängt. Man kann also zwei (oder mehr) Programme miteinander verknüpfen und erspart sich so

eine Menge Schreibarbeit. Hierbei ist allerdings unbedingt darauf zu achten, daß die Zeilennummern des zweiten Programms höher sind als die des ersten.

Programm speichern

Der Menüpunkt 2 des Hauptmenüs ermöglicht das Speichern von Programmen, die sich im Basic-Speicherbereich befinden. Dazu wird einfach der Name des Programms eingegeben. Falls ein Programm gleichen Namens bereits existiert, wird eine entsprechende Fehlermeldung in der untersten Zeile des Bildschirms ausgegeben.

Diskette verwalten

Alle Befehle, die der Floppy über den Kommando-Kanal übermittelt werden, können unter Menüpunkt 3 des Hauptmenüs übermittelt werden. Es erscheint folgendes Untermenü:

1. Initialisieren
2. Reorganisieren
3. Name ändern
4. File löschen
5. File kopieren
6. Name der Disk ändern
7. Formatieren
8. zurück

Wählt man Punkt 1 an, so wird die Floppy initialisiert.

Unter 2 wird ein Validate ausgeführt, das heißt, alle unbenutzten Blocks auf der Diskette freigegeben.

Mit Punkt 3 kann der Name eines Files geändert werden. Diskmaster fordert Sie hier auf, den alten und den neuen Namen des betreffenden Files einzugeben. Anschließend wird der Name geändert.

Menüpunkt 4 wird angewählt, wenn man ein File (Programm oder Datei) löschen möchte. Hier muß der Name des betreffenden Files eingegeben werden.

Unter Menüpunkt 5 können Files auf der Diskette kopiert werden. Auch hier müssen der Name des alten und des neuen Files angegeben werden.

Will man den Namen der Diskette ändern, so muß man den Menüpunkt 5 anwählen. Man kann dann den neuen Namen sowie die neue ID der Diskette eingeben.

Anschließend wird der Name geändert. Soll eine Diskette hingegen total gelöscht werden, ist Menüpunkt sieben an der Reihe. Auch hier werden Name und, wenn eine vollkommene Neuformatierung der Diskette verlangt wird, ID der Diskette eingegeben. Punkt acht führt Sie zurück ins Hauptmenü.

Inhalt zeigen

Vom Hauptmenü aus kann der Inhalt der Diskette gezeigt werden. Dazu wird der Menüpunkt 4 gewählt. Es erscheint dann, nach kurzer Diskettenaktivität, ein kleines Fenster auf dem Bildschirm, in dem die erste »Seite« des Directorys ausgegeben wird. Die jeweils nächste Seite wird durch Drücken der Taste F3 aufgerufen. Mit F1 verläßt man diesen Menüpunkt wieder.

Der Editor

Diskmaster bietet zur komfortablen Eingabe der Namen einen kleinen Eingabe-Editor. Dieser bietet die üblichen Funktionen des C 64 Basic-Editors, bis auf eine Änderung: DEL löscht den Buchstaben unter und hinter dem Cursor.

Zeigen der Directory während der Eingabe

Das Directory kann nicht nur vom Hauptmenü aus angesehen werden, sondern auch immer dann, wenn eine Eingabe, zum Beispiel eines Filenamens, erfolgen soll. Dank dieser Funktion kann man sich zum Beispiel davon überzeugen, daß das Programm, das man laden möchte, auch wirklich existiert, ohne ins Hauptmenü gehen zu müssen. Will man das Directory aufrufen, so muß man F3 drücken. Mit F1 verläßt man das Directory wieder.

Verlassen eines Menüpunktes

Jeder Menüpunkt kann verlassen werden, ohne daß er ausgeführt wird, in dem man F1 drückt. Es erscheint dann wieder

das jeweilige Menü und es kann ein anderer Menüpunkt gewählt werden.

Die Fehlermeldungen

Es erscheinen die üblichen Fehlermeldungen der Floppy, wenn durch einen Eingabefehler eine Fehlerbedingung entstanden sein sollte. Mit F1 wird die Fehleroutine wieder verlassen.

Verträglichkeit mit anderen Programmen

Diskmaster kann mit einer Reihe von bekannten Programmen benutzt werden, zum Beispiel Vizawrite. Diskmaster funktioniert, wenn man eine kleine Einschränkung macht, auch mit Hypra-Load. Man muß dann allerdings auf das Einlesen des Directorys verzichten. Stellt sich das Directory als lebenswichtig heraus, so muß man mit POKE 1,55 das Hypra-Load inaktivieren. Als Alternative bietet sich außerdem an, das Directory als Basic-Programm zu laden.

Einige technische Details

Diskmaster belegt den Speicherplatz von \$C000 bis \$D000. Außerdem wird der Speicherbereich von \$D000 bis \$FFFF zum Teil zur Zwischenspeicherung des Directorys benutzt. Daher kann kein Directory eingelesen werden, solange Hypra-Load aktiv ist. Ist Hypra-Load aktiviert, so erscheint nach dem Laden eines Programms die Warmstartmeldung der Floppy.

Das Directory wird bei Diskmaster auf eine etwas ungewöhnliche Art eingelesen: Diskmaster initialisiert zunächst die Diskette und prüft anschließend anhand des Namens und der ID, ob das Directory der betreffenden Diskette bereits eingelesen wurde. Wenn ja, so wird das Directory nicht erneut eingelesen. Das spart Zeit, besonders bei sehr vollen Disketten.

Selbstverständlich kann die Diskette vom Basic aus wie üblich benutzt werden. (Volker Köhne/aw)

```

programm : disk.ma      c000 c1c
-----
c000 : a9 01 8d c0 cd 20 7a ca 03
c008 : a9 00 8d 43 03 a9 10 85 46
c010 : f9 a2 00 86 fc 20 cb c3 b3
c018 : a9 31 85 fd a9 36 85 fe db
c020 : 20 c6 c9 a2 05 86 fb 20 1f
c028 : 17 c9 c9 0d d0 f2 a9 01 85
c030 : 85 cc a2 00 e4 fc d0 03 43
c038 : 4c 56 c0 e8 e4 fc d0 03 7c
c040 : 4c 3f c1 e8 e4 fc d0 03 39
c048 : 4c 96 c0 e8 e4 fc d0 03 ac
c050 : 4c 56 c1 4c c5 c1 20 7a a1
c058 : ca a2 00 86 fc 20 f5 c3 74
c060 : a9 31 85 fd a9 35 85 fe 1b
c068 : 20 de c9 a2 04 86 fb 20 63
c070 : 17 c9 c9 0d d0 f2 a9 01 cd
c078 : 85 cc a2 00 e4 fc d0 03 8b
c080 : 4c f6 c0 e8 e4 fc d0 03 14
c088 : 4c 0e c1 e8 e4 fc d0 03 e8
c090 : 4c 26 c1 4c 05 c0 20 7a b5
c098 : ca a2 00 86 fc 20 e7 c3 7c
c0a0 : a9 31 85 fd a9 39 85 fe 7b
c0a8 : 20 f6 c9 a2 08 86 fb 20 ef
c0b0 : 17 c9 c9 0d d0 f2 a9 01 0d
c0b8 : 85 cc a2 00 e4 fc d0 03 cb
c0c0 : 4c 5f c1 e8 e4 fc d0 03 c9
c0c8 : 4c 65 c1 e8 e4 fc d0 03 d4
c0d0 : 4c 6b c1 e8 e4 fc d0 03 df
c0d8 : 4c 82 c1 e8 e4 fc d0 03 72
c0e0 : 4c 99 c1 e8 e4 fc d0 03 06
c0e8 : 4c bd c1 e8 e4 fc d0 03 20
c0f0 : 4c b0 c1 4c 05 c0 20 27 b3
c0f8 : c2 c0 00 d0 03 4c 56 c0 a2
c100 : 8c 62 cd 20 de c1 ad bf 67
c108 : cd d0 0a 4c 05 c0 20 27 6f
c110 : c2 c0 00 d0 03 4c 56 c0 ba
c118 : 8c 62 cd 20 de c1 ad bf 7f
c120 : cd d0 f2 4c 2e c2 20 27 63
c128 : c2 c0 00 d0 03 4c 56 c0 d2
c130 : 8c 62 cd 20 de c2 20 de 9e
c138 : c1 20 61 c2 0c 4c 56 c0 75
c140 : d9 c3 20 27 c2 c0 00 d0 bc
c148 : 03 4c 05 c0 8c 62 cd 20 1e
c150 : 6a c2 4c 05 c0 60 20 f1 43
c158 : c6 20 f6 c7 4c 05 c0 20 15
c160 : e5 c4 4c 96 c0 20 0e c5 5e
c168 : 4c 96 c0 a9 02 8d 3e 03 f0
c170 : a9 52 20 e0 c2 c0 00 f0 7b
c178 : 06 20 30 c3 20 3b c5 4c 9e
c180 : 96 c0 a9 01 8d 3e 03 a9 2b
c188 : 53 20 ff c2 c0 00 f0 06 1f
c190 : 20 68 c3 20 3b c5 4c 96 19
c198 : c0 a9 02 8d 3e 03 a9 43 88
c1a0 : 20 e0 c2 c0 c0 0f 06 20 d9
c1a8 : 30 c3 20 3b c5 4c 96 c0 c4
c1b0 : a9 03 8d 3e 03 a9 4e 20 fd
c1b8 : b8 c2 4c 96 c0 a9 41 20 5d
c1c0 : b8 c2 4c 96 c0 a9 06 8d 4c
c1c8 : 21 d0 a9 0e 8d 20 d0 db
c1d0 : 44 e5 a9 9a 20 d2 ff a2 a2
c1d8 : 80 86 9d 4c ab e1 20 7a cb
c1e0 : ca 20 f1 cd a9 e4 8d 8b ff
c1e8 : c9 a9 cc 8d 8c c9 a9 64 f1
c1f0 : 8d 86 c9 a9 cd 8d 87 c9 63
c1f8 : 20 0e ca 20 9d c4 20 c2 f2
c200 : c4 a2 08 a0 00 20 ba ff 17

c208 : ad 62 cd a2 9d a0 cd 20 04
c210 : bd ff a9 00 85 9d a9 00 23
c218 : a6 2b a4 2c 20 9e f4 86 da
c220 : 2d 84 2e 20 9e c6 60 20 01
c228 : 76 c4 20 21 c3 60 a9 92 38
c230 : 20 d2 ff a9 04 85 c6 a9 c9
c238 : 52 8d 77 02 a9 55 8d 78 db
c240 : 02 a9 4e 8d 79 02 a9 0d c4
c248 : 8d 7a 02 4c c5 c1 a5 2d 78
c250 : 85 2b a5 2e 85 2c a5 2b 41
c258 : e9 02 85 2b b0 02 c6 2c 98
c260 : 60 a9 08 85 2c a9 01 85 67
c268 : 2b 60 20 7a ca 20 f1 cd 2c
c270 : a9 eb 8d 8b c9 a9 cc 8d 1c
c278 : 8c c9 a9 65 8d 86 c9 a9 87
c280 : cd 8d 87 c9 20 0e ca 20 0d
c288 : 9d c4 20 cd c4 a2 08 20 0b
c290 : ba ff ad 62 cd a2 9d a0 ab
c298 : cd 20 bd ff a9 00 65 9d d0
c2a0 : 20 10 e5 a6 2d a4 2e a9 1a
c2a8 : 2b 20 d2 f5 20 9e c6 20 6c
c2b0 : ff c4 a9 01 8d c0 cd 60 73
c2b8 : 85 fc 20 03 c4 20 21 c3 7d
c2c0 : c0 00 d0 01 60 8c 62 cd 64
c2c8 : 20 18 c4 a9 02 85 f9 20 cf
c2d0 : 12 c3 a9 10 85 f9 8c 63 51
c2d8 : cd 20 8f c3 20 3b c5 60 c5
c2e0 : 85 fc 20 42 c4 20 21 c3 8d
c2e8 : c0 00 d0 01 60 8c 62 cd 8c
c2f0 : 20 57 c4 20 12 c3 c0 00 33
c2f8 : d0 01 60 8c 63 cd 60 85 24
c300 : fc 20 2d c4 20 21 c3 c0 8c
c308 : 00 d0 01 60 8c 62 cd 4c 68
c310 : fe c2 a9 16 20 6c c4 20 55
c318 : 83 c9 20 9a c4 20 9a c5 5e
c320 : 60 a9 15 20 6c c4 20 83 13
c328 : c9 20 8b c4 20 9a c5 60 2b
c330 : 20 f1 cd a9 27 8d 8b c9 92
c338 : a9 cd 8d 8c c9 a9 67 8d 5f
c340 : 86 c9 a9 cd 8d 87 c9 20 4b
c348 : 0e ca 20 9d c4 a9 20 20 d1
c350 : d2 ff a9 3d 20 d2 ff a9 20
c358 : 3e 20 d2 ff a9 20 20 d2 1d
c360 : ff 20 b5 c4 20 cd c4 60 b9
c368 : 20 f1 cd a9 29 8d 8b c9 ea
c370 : a9 cd 8d 8c c9 a9 55 8d 4f
c378 : 86 c9 a9 cd 8d 87 c9 20 83
c380 : 0e ca a5 fc c9 41 f0 36 d3
c388 : 20 9d c4 20 cd c4 60 20 71
c390 : f1 cd a5 fc c9 41 f0 26 27
c398 : a9 33 8d 8b c9 a9 cd 8d ec
c3a0 : 8c c9 a9 68 8d 86 c9 a9 10
c3a8 : cd 8d 87 c9 20 0e ca 20 35
c3b0 : 9d c4 a9 2c 20 d2 ff 20 78
c3b8 : b5 c4 20 cd c4 60 a9 40 08
c3c0 : 8d 8b c9 a9 cd 8d 8c c9 c9
c3c8 : 4c a2 c3 a9 a7 8d ac c9 b8
c3d0 : a9 ca 8d ad c9 20 a4 c9 bb
c3d8 : 60 a9 fd 8d ac c9 a9 ca 93
c3e0 : 8d ad c9 20 a4 c9 60 a9 28
c3e8 : 28 8d ac c9 a9 cb 8d ad c6
c3f0 : c9 20 a4 c9 60 a9 d2 8d e5
c3f8 : ac c9 a9 ca 8d ad c9 20 fa
c400 : a4 c9 60 a9 8c 8d 8b c9 91
c408 : a9 cd 8d 8c c9 a9 5f 8d 8f
c410 : 86 c9 a9 cd 8d 87 c9 60 9c
c418 : a9 d6 8d 8b c9 a9 cd 8d 39
c420 : 8c c9 a9 5f 8d 86 c9 a9 6f

c428 : cd 8d 87 c9 60 a9 bf 8d 44
c430 : 8b c9 a9 cc 8d 8c c9 a9 5b
c438 : 61 8d 86 c9 a9 cd 8d 87 89
c440 : c9 60 a9 a1 8d 8b c9 a9 87
c448 : cc 8d 8c c9 a9 60 8d 86 18
c450 : c9 a9 cd 8d 87 c9 60 a9 ae
c458 : b0 8d 8b c9 a9 cc 8d 8c 3b
c460 : c9 a9 60 8d 86 c9 a9 cd c1
c468 : 8d 87 c9 60 85 d6 a9 01 ef
c470 : 85 d3 20 10 e5 60 a9 93 18
c478 : 8d 8b c9 a9 cc 8d 8c c9 71
c480 : a9 5f 8d 86 c9 a9 cd 8d 49
c488 : 87 c9 60 a9 9d 85 fd a9 92
c490 : cd 85 fe 60 a9 ad 85 fd 06
c498 : a9 cd 85 fe 60 a9 9d 8d 4e
c4a0 : 8b c9 a9 cd 8d 8c c9 a9 ec
c4a8 : 62 8d 86 c9 a9 cd 8d 87 fa
c4b0 : c9 20 83 c9 60 a9 ad 8d c8
c4b8 : 8b c9 a9 cd 8d 8c c9 a9 04
c4c0 : 63 8d 86 c9 a9 cd 8d 87 13
c4c8 : c9 20 83 c9 60 a4 d3 a9 89
c4d0 : 20 20 d2 ff c8 c0 27 90 05
c4d8 : f8 a5 d6 c9 18 d0 05 a9 01
c4e0 : a0 8d e7 07 60 a9 f7 8d 70
c4e8 : 8b c9 a9 cc 8d 8c c9 a9 13
c4f0 : 66 8d 86 c9 a9 cd 8d 87 46
c4f8 : c9 20 0e ca 20 cd c4 20 72
c500 : 04 c9 a9 49 20 d2 ff 20 55
c508 : cc ff 20 fe c8 60 a9 0f 10
c510 : 8d 8b c9 a9 cd 8d 8c c9 19
c518 : a9 66 8d 86 c9 a9 cd 8d 65
c520 : 87 c9 20 0e ca 20 cd c4 c4 8c
c528 : a9 09 8d 21 d0 8d 20 d0 79
c530 : 20 04 c9 a9 56 20 d2 ff ac
c538 : 4c 70 c5 20 7a ca a9 01 d8
c540 : 8d c0 cd 20 04 c9 a5 fc c4 91
c548 : c9 41 d0 03 4c 41 ce 20 c4
c550 : d2 ff ad 3e 03 a9 3a 20 fc
c558 : d2 ff ad 3e 03 c9 01 f0 c1
c560 : 0c c9 03 f0 18 20 b5 c4 13
c568 : a9 3d 20 d2 ff 20 9d c4 13
c570 : 20 cc ff 20 9e c6 20 fe 99
c578 : c8 20 ff c4 60 20 9d c4 f0
c580 : a9 2c 20 d2 ff 20 b5 c4 03
c588 : 4c 70 c5 a0 10 a9 20 91 84
c590 : fd 20 d2 ff 88 d0 f8 91 68
c598 : fd 60 20 8b c5 a5 d3 e9 eb
c5a0 : 0f 85 d3 a0 00 a9 00 85 d3
c5a8 : cc 98 48 20 e4 ff aa 68 a0
c5b0 : a8 8a f0 f1 a6 cf d0 fc 3e
c5b8 : a2 01 86 cc c9 0d d0 03 64
c5c0 : 4c 8c c6 c9 13 d0 0d 8c 42
c5c8 : bd cd a5 d3 ed bd cd 85 5f
c5d0 : d3 c4 9a c5 c9 1d d0 0a 05
c5d8 : c4 f9 b0 c9 c8 e6 d3 4c aa
c5e0 : a5 c5 c9 9d d0 0a c0 00 ee
c5e8 : f0 bb 88 c6 d3 4c a5 c5 72
c5f0 : c9 94 d0 33 8c bf cd 20 dc
c5f8 : 8c c6 c4 f9 b0 23 a9 94 4c
c600 : 20 d2 ff a9 00 85 d8 a4 97
c608 : f9 cc bf cd f0 0c 90 0a d7
c610 : 88 b1 fd c8 91 fd 88 4c cd
c618 : 09 c6 ac bf cd a9 20 91 75
c620 : fd ac bf cd c4 a5 c5 c9 b9
c628 : 14 d0 22 a5 d3 48 98 48 54
c630 : c4 f9 b0 d0 c8 b1 fd 88 e2
c638 : 91 fd 20 d2 ff c8 4c 30 02
c640 : c6 a9 20 91 fd 68 a8 68 ac
    
```

Listing »Diskmaster«. Bitte mit dem MSE eingeben. Beachten Sie die Eingabehinweise auf Seite 8

Floppy-DOS-Erweiterungen

```

c648 : 85 d3 4c a5 c5 c9 85 d0 e1
c650 : 0b 20 8b c5 a9 00 8d 43 5e
c658 : 03 4c 8c c6 c9 86 d0 13 b7
c660 : a5 d6 48 20 f1 c6 20 f6 4a
c668 : c7 68 20 6c c4 20 83 c9 e8
c670 : 4c 9a c5 c9 87 90 04 c9 54
c678 : a0 90 0e c9 20 90 0a c4 55
c680 : f9 b0 0e 91 fd c8 20 d2 d1
c688 : ff 4c a5 c5 a4 f9 b1 fd ac
c690 : c9 20 d0 08 c0 00 f0 05 78
c698 : 88 4c 8e c6 c8 60 a9 18 29
c6a0 : 85 d6 a9 00 85 d3 20 10 92
c6a8 : e5 a9 08 85 ba 8d 3f 03 30
c6b0 : 20 b4 ff a9 6f 85 b8 20 a6
c6b8 : 96 ff 20 a5 ff c9 30 d0 bb
c6c0 : 0f 20 a5 ff c9 30 d0 08 ba
c6c8 : a9 00 8d bf cd 4c ed c6 51
c6d0 : 20 d2 ff 20 a5 ff c9 d0 f9
c6d8 : d0 f6 20 ef ed a5 b9 20 5c
c6e0 : c3 ff 20 cd c4 20 07 cf 6e
c6e8 : a9 01 8d bf cd 20 ef ed e7
c6f0 : 60 20 89 c7 ad c0 cd d0 75
c6f8 : 03 4c 83 c7 20 b3 c8 a9 11
c700 : 69 8d 3c 03 a9 cd 8d 3d 59
c708 : 03 a9 02 85 b9 a9 60 85 87
c710 : b8 a2 01 20 e5 c8 a9 08 b9
c718 : 20 b4 ff a9 60 20 96 ff 29
c720 : a9 00 85 90 a0 03 84 fb 69
c728 : a9 00 85 fa 20 a5 ff 8d dc
c730 : bd cd a4 90 d0 42 20 a5 fa
c738 : ff a4 90 d0 3b a4 fb 88 a1
c740 : d0 e4 ae bd cd 20 c2 c8 60
c748 : 8a 20 c2 c8 20 a5 ff a6 28
c750 : 90 d0 25 aa f0 19 c9 22 2a
c758 : d0 02 e6 fa a6 fa e0 01 0a
c760 : d0 03 e4 69 c7 c9 20 f0 1f
c768 : 03 20 c2 c8 4c 4c c7 a9 df
c770 : 0d 20 c2 c8 a0 02 d0 ae 12
c778 : 20 ef ed 20 fe c8 a9 0c 04
c780 : 20 c2 c8 a9 00 8d c0 cd 74
c788 : 60 ad c0 cd d0 67 20 ff 71
c790 : c4 20 b3 c8 20 04 c9 a2 f9
c798 : 00 bd 6b cd 20 d2 ff e8 76
c7a0 : e0 06 90 f5 20 cc ff a9 22
c7a8 : 08 20 b4 ff a9 6f 20 96 b1
c7b0 : ff a2 04 86 f7 8e bd cd 59
c7b8 : 20 ce c8 8d bf cd 20 a5 59
c7c0 : ff ae bd cd e0 14 d0 05 3b
c7c8 : 20 a5 ff a2 14 c9 a0 f0 03
c7d0 : 14 cd bf cd f0 0f ad bf 32
c7d8 : cd c9 22 f0 08 a9 01 8d 1d
c7e0 : c0 cd 4c ef c7 e8 e0 17 0d
c7e8 : 90 cb a9 00 8d c0 cd 20 1f
c7f0 : ef ed 20 fe c8 60 20 c1 51
c7f8 : cd 20 52 ca a9 05 20 d2 ac
c800 : ff 20 b3 c8 a0 00 8c be cf
c808 : cd a9 00 a8 8d 3f 03 85 a9
c810 : d3 a9 04 85 d6 20 10 e5 e4
c818 : c8 8c bd cd cc be cd 90 6a
c820 : 03 8c be cd 20 ce c8 c9 02
c828 : 0c f0 5a 18 ac 41 03 c0 a8
c830 : b2 b0 0a 48 ad 41 03 69 8a
c838 : 08 8d 41 03 68 48 20 ce 9e
c840 : c8 aa 68 20 cd bd a9 20 2d
c848 : 20 d2 ff a6 d3 e0 05 90 1f
c850 : f7 20 ce c8 20 d2 ff c9 50
c858 : 22 d0 0f ad 3f 03 49 01 8f
c860 : 8d 3f 03 c9 00 d0 ea 20 f9
c868 : a7 c8 c9 0d d0 14 ac bd 63
c870 : cd c0 0c 90 0a 20 0f cf 30
c878 : c9 85 f0 25 4c 09 c8 4c ad
c880 : 18 c8 4c 51 c8 18 a6 d6 cf
c888 : ca a0 10 20 f0 ff 20 a7 89
c890 : c8 20 0f cf c9 85 f0 09 c5
c898 : 20 7a ca 20 f1 cd 4c f6 59
c8a0 : c7 a9 9e 20 d2 ff 60 a9 e9
c8a8 : 20 20 d2 ff a6 d3 e0 19 4c
c8b0 : 90 f7 60 a9 00 85 f7 8d b0
c8b8 : c3 c8 a9 e0 85 f8 8d c4 46
c8c0 : c8 60 8d 00 e0 ee c3 c8 42
c8c8 : d0 03 ee c4 c8 60 78 a9 33
c8d0 : 34 85 01 a2 00 a1 f7 48 d9
c8d8 : e6 f7 d0 02 e6 f8 a9 37 79
c8e0 : 85 01 58 68 60 8a 48 a5 d0
c8e8 : b9 a2 08 a4 b8 20 ba ff 00
c8f0 : 68 ae 3c 03 ac 3d 03 20 0c
c8f8 : bd ff 20 c0 ff 60 a9 02 83
c900 : 20 c3 ff 60 a9 02 85 b9 42
c908 : a9 0f 85 b8 a2 00 e5 28
c910 : c8 a2 02 20 c9 ff 60 a9 1f
c918 : 00 aa 18 6d 8b c9 8d 8b 75
c920 : c9 a9 00 6d 8c c9 8d 8c d2
c928 : c9 e4 fc d0 05 a9 12 20 e3
c930 : d2 ff 20 83 c9 20 9b c9 1a

c938 : a9 92 20 d2 ff 18 a9 01 f6
c940 : 6d 86 c9 8d 86 c9 a9 00 72
c948 : 6d 87 c9 8d 87 c9 98 e8 98
c950 : e4 fb d0 c6 a0 00 b9 53 d6
c958 : cb 20 d2 ff c8 cc 4d cd ac
c960 : d0 f4 a2 00 86 cc 20 e4 6c
c968 : ff c9 0d f0 15 c5 fd 90 46
c970 : f5 c5 fe b0 f1 a2 01 86 63
c978 : cc e9 30 85 fc 69 30 20 12
c980 : d2 ff 60 a0 00 cc 4e cd b9
c988 : f0 0a b7 65 cb 20 d2 ff a1
c990 : c8 4c 85 c9 60 a9 0d 20 e1
c998 : d2 ff 60 a9 0d 20 d2 ff d4
c9a0 : 20 d2 ff 60 a0 00 a7 0e 02
c9a8 : 20 d2 ff b9 a7 ca 20 d2 5f
c9b0 : ff c8 c0 2a d0 f5 20 9b fd
c9b8 : c9 60 20 66 e5 a9 0d 20 a6
c9c0 : d2 ff 20 d2 ff 60 20 ba ed
c9c8 : c9 a9 cb 8d 8c c9 a9 65 93
c9d0 : 8d 8b c9 a9 cd 8d 87 c9 c5
c9d8 : a9 4e 8d 86 c9 60 20 ba 72
c9e0 : c9 a9 cb 8d 8c c9 a9 cd 7c
c9e8 : 8d 8b c9 a9 cd 8d 87 c9 dd
c9f0 : a9 53 8d 86 c9 60 20 ba 0d
c9f8 : c9 a9 cd 8d 8c c9 a9 01 3a
ca00 : 8d 8b c9 a9 cd 8d 87 c9 f5
ca08 : a9 57 8d 86 c9 60 a9 18 07
ca10 : 85 d6 a9 00 85 d3 20 10 02
ca18 : e5 20 83 c9 60 ad 19 d0 a1
ca20 : 8d 19 d0 29 01 d0 07 ad a1
ca28 : 0d dc 58 4c 31 ea ad 12 88
ca30 : d0 cd 41 03 b0 0b a9 00 a2
ca38 : 8d 21 d0 ad 41 03 4c 4c 36
ca40 : ca a9 06 8d 21 d0 8d 20 21
ca48 : d0 ad 40 03 8d 12 d0 4c a5
ca50 : bc fe 78 a9 1d 8d 14 03 73
ca58 : a9 ca 8d 15 03 ad 40 03 11
ca60 : 8d 12 d0 ad 11 d0 29 7f 1b
ca68 : 8d 11 d0 a9 81 8d 1a d0 76
ca70 : 58 a9 52 8d 40 03 8d 41 b8
ca78 : 03 60 78 a9 31 8d 14 03 d4
ca80 : a9 ea 8d 15 03 ad 40 03 49
ca88 : 8d 12 d0 ad 11 d0 29 7f 43
ca90 : 8d 11 d0 a9 f0 8d 1a d0 95
ca98 : 58 a9 06 8d 21 d0 8d 20 07
caa0 : d0 a9 9e 20 d2 ff 60 93 c6
caa8 : 12 20 20 20 20 20 20 9a
cab0 : 20 20 20 20 20 20 20 c4 fa
cab8 : 49 53 4b 2d 4d 41 53 54 f8
cac0 : 45 52 20 20 20 20 20 fe
cac8 : 20 20 20 20 20 20 20 c8
cad0 : 20 92 93 12 20 20 20 24
cad8 : 20 20 20 20 20 20 20 d8
cae0 : d0 52 4f 47 52 41 4d 4d 95
cae8 : 20 4c 41 44 45 4e 20 8f
caf0 : 20 20 20 20 20 20 20 f0
caf8 : 20 20 20 20 92 93 12 20 83
cb00 : 20 20 20 20 20 20 20 00
cb08 : 20 d0 52 4f 47 52 41 4d b5
cb10 : 4d 20 53 50 45 49 43 48 88
cb18 : 45 52 4e 20 20 20 20 e2
cb20 : 20 20 20 20 20 20 20 92 05
cb28 : 93 12 20 20 20 20 20 94
cb30 : 20 20 20 20 c4 49 53 4b e7
cb38 : 45 54 54 45 20 56 45 52 d4
cb40 : 57 41 4c 54 45 4e 20 5d
cb48 : 20 20 20 20 20 20 20 48
cb50 : 20 20 92 11 12 20 c9 48 21
cb58 : 52 45 20 d7 41 48 4c 20 18
cb60 : 92 3d 3e 20 9d 1d 31 2e 08
cb68 : 2e 2e d0 52 4f 47 52 41 27
cb70 : 4d 4d 20 4c 41 44 45 4e dd
cb78 : 1d 32 2e 2e 2e d0 52 4f 51
cb80 : 47 52 41 4d 4d 20 53 50 ae
cb88 : 45 49 43 48 45 52 4e 1d a6
cb90 : 33 2e 2e 2e c4 49 53 4b a6
cb98 : 45 54 54 45 20 56 45 52 34
cba0 : 57 41 4c 54 45 4e 1d 34 d9
cba8 : 2e 2e 2e c9 4e 48 41 4c 77
cbb0 : 54 20 5a 45 49 47 45 4e d4
cbb8 : 1d 35 2e 2e 2e 5a 55 52 71
cbc0 : 55 45 43 4b 20 5a 55 20 5d
cbc8 : c2 c1 d3 c9 c3 1d 31 2e df
cbd0 : 2e 2e cc 41 44 45 4e 1d 52
cbd8 : 32 2e 2e 2e cc 41 44 45 e5
cbe0 : 4e 20 26 20 d3 54 41 52 55
cbe8 : 54 45 4e 1d 33 2e 2e 2e d0
cbf0 : cd 45 52 47 45 1d 34 2e 48
cbf8 : 2e 2e 5a 55 52 55 45 43 ea
cc00 : 4b 91 1d 31 2e 2e c9 22
cc08 : 4e 49 54 49 41 4c 49 53 7b
cc10 : 49 45 52 45 4e 1d 32 2e 2c
cc18 : 2e 2e d2 45 4f 52 47 41 e2
cc20 : 4e 49 53 49 45 52 45 4e a9

cc28 : 1d 33 2e 2e 2e ce 41 4d 29
cc30 : 45 20 41 45 4e 44 45 52 3f
cc38 : 4e 1d 34 2e 2e 2e c6 49 ea
cc40 : 4c 45 20 4c 4f 45 53 43 b3
cc48 : 48 45 4e 1d 35 2e 2e 2e 44
cc50 : c6 49 4c 45 20 4b 4f 50 b1
cc58 : 49 45 52 45 4e 1d 36 2e 84
cc60 : 2e 2e ce 41 4d 45 20 44 89
cc68 : 45 52 20 c4 49 53 4b 20 14
cc70 : 41 45 4e 44 45 52 4e 1d ca
cc78 : 37 2e 2e 2e c6 4f 52 4d e2
cc80 : 41 54 49 45 52 45 4e 1d a9
cc88 : 38 2e 2e 2e da 55 52 55 75
cc90 : 45 43 4b d0 52 4f 47 52 c5
cc98 : 41 4d 4d 4e 41 4d 45 3a a5
cca0 : 20 41 4c 54 45 52 20 c6 f3
cca8 : 49 4c 45 4e 41 4d 45 3a 3b
ccb0 : 4e 45 55 45 52 20 c6 49 73
ccb8 : 4c 45 4e 41 4d 45 3a c6 d8
ccc0 : 49 4c 45 4e 41 4d 45 3a 53
ccc8 : ce 41 4d 45 20 44 45 52 11
ccd0 : 20 c4 49 53 4b 3a 20 20 56
ccd8 : 20 20 20 20 20 20 20 d8
cce0 : 20 c9 c4 3a 12 20 4c 41 33
cce8 : 44 45 20 12 20 53 50 45 82
ccf0 : 49 43 48 45 52 45 20 12 8a
ccf8 : 20 49 4e 49 54 49 41 4c a7
cd00 : 49 53 49 45 52 45 20 c4 47
cd08 : 49 53 4b 45 54 54 45 12 97
cd10 : 20 52 45 4f 52 47 41 4e 96
cd18 : 49 53 49 45 52 45 20 c4 5f
cd20 : 49 53 4b 45 54 54 45 12 af
cd28 : 20 12 20 4c 4f 45 53 43 d6
cd30 : 48 45 20 12 20 46 4f 52 7b
cd38 : 4d 41 54 49 45 52 45 20 a0
cd40 : 12 20 41 45 4e 44 45 52 1c
cd48 : 45 20 49 4e 20 13 13 17 cf
cd50 : 17 12 15 0a 14 0a 0c 14 e1
cd58 : 13 11 12 12 1a 10 0c 0e 29
cd60 : 0f 09 00 00 07 0c 18 02 29
cd68 : 0d 24 23 4d 2d 52 90 07 b0
cd70 : 14 42 2d 5d 20 33 20 31 79
cd78 : 34 34 42 2d 50 20 33 20 10
cd80 : 31 36 32 55 32 3a 20 33 df
cd88 : 20 30 20 31 38 20 30 55 df
cd90 : 31 3a 20 33 20 30 20 31 b3
cd98 : 38 20 30 09 0c 20 20 20 90
cda0 : 20 20 20 20 20 20 20 a0
cda8 : 20 20 20 20 20 20 20 a8
cdb0 : 20 20 20 20 20 20 20 b0
cdb8 : 20 20 20 20 20 00 00 00 f7
cdc0 : 00 ad 43 03 d0 25 a9 f0 87
cdc8 : 8d d9 cd a9 04 8d d6 cd 8e
cdd0 : a2 04 a0 00 b9 00 04 99 7b
cdd8 : 00 f0 c8 d0 f7 ee d6 cd 8a
cde0 : ee d9 cd ca d0 ec a9 01 a5
cde8 : 8d 43 03 a9 0e 20 2b ce 39
cdf0 : 60 ad 43 03 f0 34 78 a9 3e
cdf8 : 34 85 01 a9 f0 8d 0b ce aa
ce00 : a9 04 8d 0e ce a2 04 a0 24
ce08 : 00 b9 00 f0 99 00 04 c8 3e
ce10 : d0 f7 ee 0b ce ee 0e ce 33
ce18 : ca d0 ec a9 37 85 01 58 0f
ce20 : a9 07 20 2b ce a9 00 8d 10
ce28 : 43 03 60 a2 8d 8e 36 ce d2
ce30 : a2 04 a0 00 99 00 8d c8 8b
ce38 : d0 fa ee 36 ce ca d0 f2 74
ce40 : 60 a9 10 8d 62 cd a9 03 6c
ce48 : 85 b9 85 b8 a9 6a 8d 3c bf
ce50 : 03 a9 cd 8d 3d 03 a2 01 c5
ce58 : 20 e5 c8 20 e4 ce a9 71 ef
ce60 : 8d 8b c9 a9 cd 8d 8c c9 69
ce68 : 20 b6 ce a2 02 20 c9 ff 34
ce70 : 20 83 c9 20 cc ff a2 03 26
ce78 : 20 c9 ff 20 9d c4 20 c1 85
ce80 : ce ad 63 cd f0 28 a9 02 b2
ce88 : 8d 63 cd a9 7a 8d 8b c9 45
ce90 : a9 cd 8d 8c c9 20 b6 ce 2b
ce98 : a2 02 20 c9 ff 20 83 c9 1f
cea0 : 20 cc ff a2 03 20 c9 ff d3
cea8 : 20 b5 c4 20 c1 ce a9 03 17
ceb0 : 20 c3 ff 4c 70 c5 a9 9b 4e
ceb8 : 8d 86 c9 a9 cd 8d 87 c9 2b
cec0 : 60 20 cc ff a9 83 8d 8b 67
cec8 : c9 a9 cd 8d 8c c9 a9 9c 82
ced0 : 8d 86 c9 a9 cd 8d 87 c9 43
ced8 : a2 02 20 c9 ff 20 83 c9 5f
cee0 : 20 cc ff 60 20 cc ff a9 2e
cee8 : 8f 8d 8b c9 a9 cd 8d 8c b2
cef0 : c9 a9 9c 8d 86 c9 a9 cd 60
cef8 : 8d 87 c9 a2 02 20 c9 ff 58
cf00 : 20 83 c9 20 cc ff 60 20 e7
cf08 : e4 ff c9 85 f0 f4 60 20 ad
cf10 : e4 ff c9 85 f0 f4 c9 86 7a
cf18 : d0 f5 60 00 ff ff ff ff fb

```

Listing »Diskmaster« (Schluß). Bitte mit dem MSE eingeben.

Entstörung des 1541-Laufwerks

Bisher war es fast unmöglich, den Monitor oder Fernseher in die unmittelbare Nähe des Diskettenlaufwerks zu stellen. Durch die elektromagnetischen Felder ist kaum noch ein störungsfreier Zugriff möglich. Abhilfe schafft eine selbstgebastelte Abschirmung.

Eine Computeranlage braucht schon ihren Platz: Computer, Monitor, Diskettenstation und Drucker benötigen schon fast den gesamten Arbeitstisch. Warum also nicht platzsparend das Floppy-Laufwerk unter den Fernseher oder zumindest näher an ihn heran stellen. Doch unterschreitet man einen gewissen Mindestabstand, führt jeder Zugriff auf eine Diskette nur noch zu Lesefehlern. In einem Monitor und Fernsehgerät wird mit sehr starken elektromagnetischen Feldern gearbeitet, die die ungenügend abgeschirmte 1541 bei geringem Abstand aus dem Takt bringen können. Wer wenig Platz hat, und das Laufwerk am oder unter dem Fernseher stehen haben möchte, der muß die Abschirmung entscheidend verbessern. Das ist aber sehr viel einfacher als es sich anhört. Sie brauchen noch nicht einmal Ihren Lötkolben anzuheizen, ein Schraubenzieher ist schon Werkzeug genug.

Sie schrauben einfach ihre Floppy auf und nehmen den Deckel ab, natürlich erst, nachdem Sie den Netzstecker gezogen haben! In den Deckel kleben Sie nun eine möglichst starke Metallfolie, also keine einfache Alu-Frischhaltefolie. Die Folie muß aus einem Stück sein. Achten Sie auch darauf, daß Sie nur die vorderen zwei Drittel des Deckels bekleben, so daß die Lüftungsschlitze hinten frei bleiben, sonst stirbt Ihr Laufwerk recht bald den Wärmetod.

Die Metallfolie muß nun noch leitend mit dem Laufwerk-Chassis verbunden werden, damit der gewünschte Abschirm-

effekt eintritt. Zu diesem Zweck benötigen Sie eine Kontaktfeder, die Sie sich sehr einfach aus einem Blech nach unserer Vorlage (siehe Bild) basteln können. Diese Kontaktfeder kann bei Laufwerken mit kurzer Platine einfach an einem der beiden vorderen Befestigungswinkel festgeschraubt werden. Bei Laufwerken mit langer Platine stehen diese beiden Befestigungswinkel nicht mehr zur freien Verfügung. Isolieren Sie einfach die Unterseite der Kontaktfeder, damit keine Kurzschlüsse auf der Platine entstehen können. Dann darf die Kontaktfeder auf der Platine am Befestigungswinkel festgeschraubt werden. Wird der Deckel wieder aufgesetzt, sollte die Kontaktfeder gegen die Metallfolie drücken und somit die Verbindung herstellen.

Wer sein Laufwerk offen, das heißt ohne Deckel, betreibt, hat es sogar noch einfacher. Sie besorgen sich dann ein dünnes Metallblech im DIN-A3-Format, das Sie halbkreisförmig über das offene Laufwerk stülpen. An den unteren Befestigungswinkeln, die das Chassis im Gehäuse halten, müssen Sie nun einfach eine leitende Verbindung zwischen Blech und Chassis herstellen. Dies geht zum Beispiel durch Zwischenklemmen von zwei Pfennig-Stücken. Da das Blech nur seitlich, aber nicht nach vorne und hinten abschließt, ist eine mehr als ausreichende Kühlung gewährleistet.

Nach diesen kleinen Umbauten sollten Sie in der Lage sein, das Laufwerk ohne Schwierigkeiten in unmittelbarer Nähe von Fernseher oder Monitor zu betreiben. (Uwe Gerlach/bs)

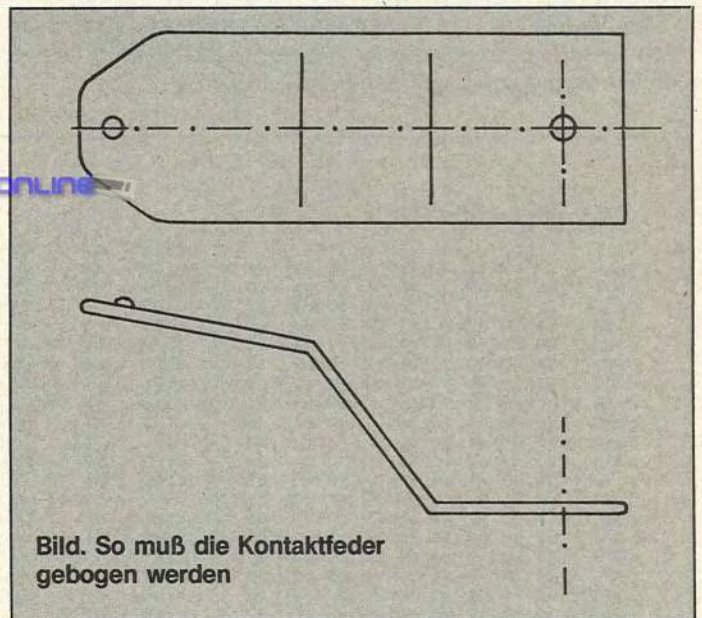


Bild. So muß die Kontaktfeder gebogen werden



Schnelles Formatieren mit der 1541

Das 1541-Floppy-Laufwerk gehört nicht nur beim Laden, sondern auch beim Formatieren einer Diskette nicht gerade zu den schnellsten seiner Gattung. »Fast Format« ändert das.

Wer bislang mit einer Datasette gearbeitet hat, wird erstaunt feststellen, daß eine Diskette erst formatiert werden muß, bevor man etwas mit ihr anfangen kann. Zu allem Unglück dauert dieser Vorgang, bei dem die Diskette in Spuren und Sektoren eingeteilt wird, fast genau 80 Sekunden. Wenn man beispielsweise ein ganzes Paket (zehn Disketten) nacheinander formatieren möchte, sitzt man fast 15 Minuten vor dem Computer. Fast Format reduziert diesen Vorgang auf nur 11 Sekunden pro Diskette, oder, bei zehn Disketten, auf unter drei Minuten.

Das eigentliche Fast Format ist als Maschinenprogramm im \$C000 Bereich abgespeichert (Listing 1). Die neue Formatier-Routine wird durch folgenden Befehl aufgerufen:

```

0 POKE 53280,0:POKE 53281,0:PRINT" {CLR,GRE
  EN}" <067>
1 PRINT SPC(12)"FASTFORMAT 11.0{DOWN}" <001>
2 PRINT SPC(13)"VON JAN KUSCH{DOWN}" <237>
10 INPUT" {DOWN}DISK-NAME {3SPACE}";N$ <228>
20 INPUT" {DOWN}DISK-ID {5SPACE}";I$ <168>
30 INPUT" {DOWN}LAUFWERK-NR.";D <009>
35 IF D<>8 AND D<>9 THEN 30 <239>
40 SYS 49152,N$,I$,D <096>
    
```

Listing 2. Dieses kleine Hilfsprogramm vereinfacht das Formatieren mit »Fast Format«. Bitte beachten Sie die Eingabehinweise auf Seite 6.

SYS 49152,N\$,I\$,D

Dabei ist N\$ der neue Diskettenname, I\$ die genau zwei Zeichen lange ID und D die Geräteadresse des betreffenden Floppy-Laufwerks. Der Fast Format-Befehl kann genauso wie der normale Formatierbefehl (N:Name, ID) verwendet werden. Etwas einfacher geht es mit dem kleinen Basic-Unterprogramm (Listing 2). Es fragt alle Parameter ab und startet die Format-Routine. Bitte vergessen Sie nicht, vor dem Start des Programms eine neue Diskette einzulegen, denn sonst ist Ihr gerade eingegebenes Programm wieder verloren.

Eine kleine Einschränkung noch zum Schluß. In der Regel gibt es keine Probleme mit schnell formatierten Disketten, bei älteren und verstellten Laufwerken sind allerdings Fehler möglich. Verwenden Sie deshalb für ganz besonders wichtige Daten-Disketten den normalen Formatier-Befehl.

(Jan Kusch/aw)

programm : fast format c000 c3fc

```

c000 : 20 e8 c0 20 9b b7 86 ba cf
c008 : ea ea a9 05 a2 00 a0 c1 a3
c010 : 20 44 c0 a9 06 a2 e8 a0 12
c018 : c1 20 44 c0 a9 07 a2 da 26
c020 : a0 c2 20 44 c0 a9 01 a2 55
c028 : c0 a0 00 20 ae c0 ad a7 33
c030 : 02 20 a8 ff 20 ae ff a2 29
c038 : 00 a0 05 20 b9 c0 20 d7 9f
c040 : c0 4c ae ff 8d 77 c0 86 76
c048 : fb 84 fc a0 00 8c 76 c0 98
c050 : 20 d7 c0 a2 00 bd 73 c0 1d
c058 : 20 a8 ff e8 00 06 d0 f5 57
c060 : a2 20 b1 fb 20 a8 ff c8 d7
c068 : ca d0 f7 20 ae ff c0 00 8a
c070 : d0 db 60 4d 2d 57 e0 07 0f
c078 : 20 20 a4 c0 a9 52 20 8a ac
c080 : c0 20 ae ff 20 e1 c0 a0 51
c088 : 00 60 8d a0 c0 20 d7 c0 1e
c090 : a0 00 b9 9e c0 20 a8 ff 22
c098 : c8 c0 06 d0 f5 60 4d 2d 4e
c0a0 : 57 c0 00 01 8d a3 c0 8e 8e
c0a8 : a1 c0 8c a2 c0 60 20 a4 fa
c0b0 : c0 a9 57 20 8a c0 a0 00 50
c0b8 : 60 8e d5 c0 8c d6 c0 20 af
c0c0 : d7 c0 a0 00 b9 d2 c0 20 95
c0c8 : a8 ff c8 c0 05 d0 f5 4c 01
c0d0 : ae ff 4d 2d 45 00 05 a5 2b
c0d8 : ba 20 b1 fd a9 6f 4c 93 7d
c0e0 : ff ea 20 fd ae 4c 57 e2 8d
c0e8 : a2 10 a9 a0 9d 67 c1 ca c2
c0f0 : 10 fa 20 e2 c0 c9 11 90 a1
c0f8 : 02 a9 10 4c d8 c3 ea ea 8a
c100 : ad 79 05 85 12 ad 7a 05 de
c108 : 85 13 a9 01 85 22 a9 0a c5
c110 : 85 ba 78 ad 00 1c 09 04 d3
c118 : 8d 00 1c a9 2d 85 4a 20 4a
c120 : 93 05 c6 4a d0 f9 a2 00 98
c128 : 20 a0 05 20 bd 05 a9 ee 66
c130 : 8d 0c 1c 20 00 06 85 c0 96
c138 : ad 00 1c 29 fb 8d 00 1c 76
c140 : a9 ec 8d 0c 1c 58 90 01 0d
c148 : 60 20 94 07 a9 12 85 06 0c
c150 : a9 00 85 07 20 c8 07 20 e0
c158 : be 07 a9 ff 8d 01 03 e6 bf
c160 : 07 20 c8 07 4c 05 d0 54 63
c168 : 45 53 54 4c 41 55 46 94 f6
c170 : a0 a0 a0 a0 a0 a0 a0 6f
c178 : a0 30 31 a0 32 41 a0 a0 82
c180 : a0 a0 a0 02 ae 00 1c e8 06
c188 : 20 a0 05 88 d0 f6 e6 22 ef
c190 : 4c bd 05 a0 02 ae 00 1c de
c198 : ca 20 a0 05 88 d0 f6 60 e7
c1a0 : 8a 29 03 85 bb ad 00 1c 92
c1a8 : 29 fc 05 bb 8d 00 1c a9 a5
c1b0 : 04 85 bb a2 00 ca d0 fd 50
c1b8 : c6 bb d0 f9 60 a5 22 20 cb
c1c0 : 4b f2 8a 0a 0a 0a 0a 95
c1c8 : 85 44 ad 00 1c 29 9f 05 6e
c1d0 : 44 8d 00 1c 60 ad 0c 1c 3a
c1d8 : 29 1f 09 c0 8d 0c 1c a9 e8
c1e0 : ff 8d 03 1c 8d 01 1c 60 fc
c1e8 : a5 22 20 4b f2 85 43 20 b8
c1f0 : d5 05 a9 ff 8d 01 1c a9 57
c1f8 : 00 85 bc aa a8 a5 39 99 0f
c200 : 00 03 a5 bc 99 02 03 a5 84
c208 : 22 99 03 03 a5 13 99 04 79
c210 : 03 a5 12 99 05 03 a9 0f cb
c218 : 99 06 03 99 07 03 a9 0d d8
c220 : 59 02 03 59 03 03 59 04 1c
c228 : 03 59 05 03 99 01 03 18 57
c230 : 98 69 08 a8 e6 bc a5 bc f8
c238 : c5 43 90 c1 98 48 e8 8a 80
c240 : 9d 00 04 e8 d0 fa a9 4b 1e
c248 : 8d 00 04 a9 03 85 31 20 6d
c250 : 30 fe 68 a8 88 20 e5 fd 4c
c258 : 20 f5 fd a9 04 85 31 20 99
c260 : e9 f5 85 3a 20 8f f7 a9 9e
c268 : 00 85 32 a9 ff 8d 01 1c 95
c270 : a2 05 50 fe b8 ca d0 fa a4
c278 : a2 0a a4 32 50 fe b8 b9 e2
c280 : 00 03 8d 01 1c c8 ca d0 5a
c288 : f3 84 32 a2 08 50 fe b8 0f
c290 : a9 55 8d 01 1c ca d0 f5 af
c298 : a9 ff a2 05 50 fe b8 8d 85
c2a0 : 01 1c ca d0 f7 a2 bb 50 a0
c2a8 : fe b8 bd 00 01 8d 01 1c 2a
c2b0 : e8 d0 f4 a0 00 50 fe b8 41
c2b8 : b1 30 8d 01 1c c8 d0 f5 3c
c2c0 : a9 55 a2 08 50 fe b8 8d b8
c2c8 : 01 1c ca d0 f7 c6 bc d0 ee
c2d0 : 9a 50 fe b8 50 fe b8 4c e1
c2d8 : 00 07 20 00 fe a5 c0 d0 26
c2e0 : 03 4c 67 07 a9 c8 85 bd 36
c2e8 : a5 43 85 bc a9 00 85 32 3d
c2f0 : 20 77 07 a2 0a a4 32 50 11
c2f8 : fe b8 ad 01 1c d9 00 03 74
c300 : d0 30 c8 e6 32 ca d0 ef 94
c308 : 20 77 07 a0 bb 50 fe b8 65
c310 : ad 01 1c d9 00 01 d0 1a 00
c318 : c8 d0 f2 a2 fc 50 fe b8 19
c320 : ad 01 1c d1 30 d0 0b c8 d6
c328 : ca d0 f2 c6 bc d0 c1 4c e2
c330 : 67 07 c6 bd d0 b2 c6 ba b7
c338 : f0 03 4c 00 06 a9 03 38 e7
c340 : 60 a5 22 c9 23 f0 06 20 47
c348 : 82 05 4c 00 06 a9 01 18 42
c350 : 60 a9 d0 8d 05 18 a9 03 28
c358 : 2c 05 18 10 0c 2c 00 1c 69
c360 : 30 f6 ad 01 1c b8 a0 00 a1
c368 : 60 68 68 4c 58 07 20 be 5c
c370 : 07 a9 03 85 6e 20 b7 ee 62
c378 : a0 1b b9 67 05 99 90 03 66
c380 : 88 10 f7 a9 41 8d 02 03 d2
c388 : a9 2a 8d 03 03 a9 11 8d e7
c390 : 48 03 a9 fc 8d 49 03 60 54
c398 : a0 00 98 99 00 03 c8 d0 6e
c3a0 : fa 60 a9 90 85 00 a5 00 36
c3a8 : 30 fc 60 00 00 ad 00 04 e4
c3b0 : 85 06 a9 00 85 07 85 1c 82
c3b8 : a9 80 85 00 a5 00 30 fc 18
c3c0 : a5 16 85 12 a5 17 85 13 63
c3c8 : a9 00 85 be 85 c0 ad 01 c2
c3d0 : 04 20 4b f2 85 bf 60 ea c3
c3d8 : a8 88 f0 a4 30 08 b1 bb c3
c3e0 : 99 67 c1 88 10 f8 20 e2 bd
c3e8 : c0 c9 02 f0 03 4c 48 2b 45
c3f0 : a8 88 b1 bb 99 79 c1 88 3e
c3f8 : 10 f8 60 ea 8d ad c9 20 a7
    
```

Listing 1. »Fast Format 1.0«. Bitte beachten Sie die Eingabehinweise auf Seite 8.

64'er

PROGRAMM-SERVICE

Bestellungen aus anderen Ländern bitte per Auslandspost-anweisung! Achtung: Nicht die eingehaftete Zahlkarte verwenden.

Bestellungen aus der Schweiz bitte direkt an: Markt & Technik Vertriebs AG, Kollerstr. 3, CH-8300 Zug, Tel.: 042/223155.

Bestellungen aus Österreich bitte direkt an: Bücherzentrum Meidling, Schönbrunnerstr. 261, 1120 Wien. Microcomput-ique Erhard Schiller Fasangasse 21 1030 Wien

Programme aus früheren Ausgaben

Sonderheft: Grafik

Bestell-Nr. L685 S4 A DM 29,90*

Ausgabe 9/85

Bestell-Nr. L 6 8509 A DM 29,90*

Ausgabe 8/85

Bestell-Nr. L 6 8508 A DM 29,90*

Sonderheft: Tips & Tricks (2. ü. Auflage)

Floppy-Utilities CB 023 DM 29,90*

Hilfsprogramme CB 024 DM 29,90*

Sonderheft: Spiele

Beide Disketten in einem Paket! Verwenden Sie nur diese Bestell-Nr.: L 6 85S3 A DM 34,90*

Ausgabe 7/85

Bestell-Nr. L 6 8507 A DM 29,90*

Commodore 64
Haushaltsbuch (AdM) S. 57
Terminalprogramm S. 152
Centron S. 80
Editor S. 151
Ein-/Ausgaberroutine S. 77
Fenster (C 16) S. 84
File-Compactor S. 82
Hypra-Assembler (LdM) S. 66
IEEE-Basic S. 46
Logik S. 144
Merkzettel S. 83

Modulator S. 46
REM-Killer S. 75
Sound Editor S. 136
Startgenerator S. 74

Ausgabe 6/85

Bestell-Nr. L 6 8506A DM 29,90*

Commodore 64
MSE S. 54
HI-EDDI/MPS 801 S. 69
Prost S. 76
E-Routine 64 S. 148
GCR-HEX S. 117
HEX-GCR S. 118
Samurai S. 72
Scroll-Machine (LdM) S. 61
Crossreferenz S. 155
Heapsort S. 126

C 16

F-Plotter S. 68

Sonderheft: Abenteuerspiele

Bestell-Nr.: L685 S2 DM 34,90*

Ausgabe 5/85

Bestell-Nr. L 6 8505 A DM 29,90*

Commodore 64
Checksum. Schnell S. 54
MSE Lader S. 55
MPS 802 S. 31
Format-System S. 147
VIC S. 175
6510 I S. 71
Sternenhimmel (AdM) S. 57

Assemblerkurs S. 144
Direktory-Sorter S. 77
Trick.OBJ S. 65
3D-Movie-Maker (LdM) S. 65
Modulator (Heft 4) S. 155
VC 20
Checksummer S. 54
Minigrafik S. 69
Longscreen S. 83
C 16
Help & Trace S. 84

Ausgabe 4/85

Bestell-Nr. L 6 8504A DM 29,90*

Commodore 64
Checksum. Schnell S. 54
MSE Lader S. 58
S-MON Teil 5 S. 67
Window 64 S. 88
XBASIC S. 59
ASS. Kurs Teil 8 S. 138
Hypra-Perfect S. 76
11 neue Einzeiler S. 153
Print-List S. 79
Netzgrafik S. 71
ROT-DEMO S. 73
X.Y-DEMO 1 u. 2 S. 73
Ostereier S. 51
Befehlserweiterung VC 20 S. 23
Checksummer S. 56
Supergrafik III S. 73
Prg. Schutz S. 83

Ausgabe 3/85

Bestell-Nr. L 6 8503A DM 29,90*

Commodore 64
Checksum. Schnell S. 70
MSE Lader S. 79
Hires-3/Teil 2 S. 136
Intellisearch\$C S. 153
Son of Destroyer S. 42
Ligatab (LdM) S. 63
Fl. Basicloader S. 82
Floppyliester S. 82
E.Zeichnen 1 und 2 S. 72
Meisterschütze (AdM) S. 53
Saver S. 89
Screensave S. 88
Pic-Loader S. 90
VC 20
Checksummer S. 70
IRQ-Clock S. 156
Autos... S. 159

Ausgabe 2/85

Bestell-Nr. L 6 8502A DM 29,90*

Commodore 64
Checksummer 64 S. 65
MSE S. 68
SMON (Teil 4) S. 72

Grab des Pharaos (LdM) S. 56
Basic-Lader S. 91
RAM-Floppy S. 92
Notlandung S. 156
Hires-3 S. 123
Benchmark-Test S. 37
Format.Stringkur S. 148
Versch. Test.USR S. 140
Cursorsteuerung S. 86
Rolling-Data S. 88
Super-Memory S. 81
VC 20
Checksummer VC 20 S. 65
Familienplanung (AdM) S. 53
Super 8-Steuerung S. 70
Q+Bert (3K) S. 78
Joypaint S. 143
Bitmapping S. 145
Multicolor-Demo S. 146

Ausgabe 1/85

Bestell-Nr. L 6 8501A DM 29,90*

Commodore 64
Checksummer 64 S. 72
Handballtrainer (AdM) S. 53
SMON (Teil 3) S. 69
Hi-Eddi (LdM) S. 57
Hypra-Load mal vier S. 82
Tips und Tricks S. 87
Provic 64 S. 76
Eingabe (UPB) S. 156
VC 20
Checksummer VC 20 S. 72

Ausgabe 12/84

Bestell-Nr. CB 022 DM 29,90*

Commodore 64
Synthesizer (AdM) S. 51
SMON (2. Teil) S. 60
3D-Vier gewinnt S. 96
Trace S. 76
Stringy S. 88
Lader S. 92
Auto S. 84
Listenschutz S. 85
Simons Axo (SB) S. 64
Kreuzworträtsel S. 150

VC 20

Mathematik Basic (8K >) (LdM) S. 55
Fast Tape S. 80

Ausgabe 11/84

Bestell-Nr. CB 020 DM 29,90*

Ausgabe 10/84

Bestell-Nr. CB 019 DM 29,90*

Ausgabe 9/84

Bestell-Nr. CB 014 DM 29,90*

Bedeutung der Abkürzungen

*LdM = Listing des Monats
*AdM = Anwendung des Monats
*SB = Simons Basic
*GV = Grundversion
*GV > = alle Speicherversionen können

verwendet werden (einschließlich GV)

*3K = 3-KByte-Speichererweiterung wird benötigt
*8K > = Speichererweiterung größer als 8 KByte wird benötigt
*UPB = Unterprogrammibliothek

* Alle Preise inklusive Mehrwertsteuer.

Achtung!

Bitte verwenden Sie für Ihre Bestellung nur die eingehaftete Postscheck-Zahlkarte zur Überweisung des Rechnungsbetrags.

Fehlende Hefte erhalten Sie bei: Markt & Technik

Vertrieb 64'er
Hans-Pinsel-Str. 2,
8013 Haar

FMON 1541 — das Werkzeug für werdende Floppy- Spezialisten

Zur Programmierung des 1541-Floppy-Laufwerks braucht man gutes Werkzeug. Mit dem FMON 1541 können Sie Programme für das Floppy-Laufwerk schreiben oder sich einfach ein wenig im Speicher umsehen.

FMON 1541 ist eine große Hilfe für alle, die Ihr 1541-Laufwerk programmieren wollen. Er erlaubt einfaches Assemblieren und Disassemblieren. Außerdem sind noch viele andere Sonderfunktionen eingebaut. In der folgenden Anleitung werden wir Ihnen die Funktionen und Befehle des FMON 1541 erläutern.

Tippen Sie zunächst den FMON 1541 (Listing 1) mit dem MSE ab und speichern Sie ihn. Nach dem Laden mit LOAD "FMON 1541",8,1 wird der Monitor mit SYS 32768 oder SYS 8*4096 gestartet. Folgende Befehle bietet Ihnen der FMON 1541:

A Direktassembler

Format :A aaaa bbb ccccc

Der Befehl bbb mit dem Operanden ccccc wird an die Adresse aaaa geschrieben (6502-Schreibweise). Alle Angaben erfolgen hexadezimal. Ein \$ muß nicht vorgestellt werden. Bei der Art Immediate sind achtstellige Binärzahlen (% vorgestellt) und ASCII-Angaben (in " (CHR\$(34)) erlaubt. Beispiel: AND #%10101100 oder LDA #"A", wobei der zweite Anführungsstrich weggelassen werden kann (LDA #"A). Bei Accu-Befehlen kann das nachfolgende A weggelassen werden. Einzelne Bytes können mit einem ».« eingegeben werden (.A9). Bei Branch-Befehlen wird die Zieladresse in Form einer zweistelligen Hexzahl angegeben. Und jetzt eine »Spezialität« des FMON 1541: Alle undefinierten Opcodes (vorgestellt im 64'er, Ausgabe 3/85) werden verarbeitet! (Tabelle 1)

D Disassemblieren von Speicherbereichen

Format: D aaaa, bbbb

Disassembliert den Speicherbereich aaaa bis bbbb. Wenn bbbb fehlt, wird ab der Adresse aaaa disassembliert. Wenn beide Adressen fehlen, dann wird ab 0300 disassembliert. Ändern von Bytes ist möglich. Fahren Sie an die Stelle und ändern Sie die oder das Byte und drücken »RETURN«.

X Exit

Springt ins Basic zurück

Z Exit II

Springt ins Basic zurück (wie RUN-STOP / RESTORE)

G GO

Format: G aaaa

Springt zur Speicheradresse aaaa. Das Programm muß mit RTS enden.

M Memory Dump

Format: M aaaa, bbbb

Gibt den Speicherbereich aaaa bis bbbb mit acht Byte pro Zeile und ASCII-Ausgabe aus. Wenn bbbb fehlt, wird wie beim D-Befehl gehandelt. Wenn aaaa fehlt, dann wird der Bereich \$0300 bis 03FF gelistet. Ändern von Byte möglich.

Ändern der Floppyadresse.

zum Beispiel: #9. Alle Befehle wirken auf das Floppy-Laufwerk mit der Geräteadresse 9.

> Ändern der Druckeradresse

zum Beispiel: >5

Das Floppy-Laufwerk ist auf acht und der Drucker auf vier vor-eingestellt.

@ Senden von Befehlen in den Befehls-Kanal

Format: @a: bbbbbb

Sendet den Befehl a mit dem Text bbbbbb an das Floppy-Laufwerk.

* Liest Fehlerkanal aus.

\$ Listet Directory.

C Kaltstart des Floppy-Laufwerks. Alle Puffer werden gelöscht!.

I Initialisieren des Floppy-Laufwerks.

R Lesen eines Blocks.

Format: R tt ss

Liest den Track tt, Sector ss in Puffer Null (\$0300 bis 03FF). Mit M kann er angezeigt und verändert werden.

W Schreiben eines Blocks auf Diskette (\$0300 bis 03FF).

Format: Wie bei Read.

Read und Write werden mit Jobbefehlen gehandhabt. Ein eventuell auftretender Fehler steht in Adresse \$0000 (Fehlermeldungen siehe Tabelle 2). Meiden Sie ILLEGAL BLOCK OR SECTOR!

J Senden eines Jobbefehls für Puffer Null.

Format: J aa, bb cc

Der Jobbefehl aa wird mit den Parametern bb und cc in Speicheradresse \$0000, 0006/0007 geschrieben.

P ASCII-Text an eine bestimmte Stelle schreiben.

Format: P aaaa, tttttttt

Der Text tttttttt wird an die Adresse aaaa und folgende gesetzt.

S Sichern von Speicherbereichen.

Format: S "tttttttt", aaaa, bbbb

Der Bereich aaaa bis bbbb wird in die sequentielle Datei tttttttt zwischengespeichert. Da sequentielle Daten Puffer benutzen, wird der Speicherbereich aaaa bis bbbb erst in den Bereich ab \$C000 im Computer zwischengespeichert, dann erst in die sequentielle Datei geschrieben. Dieses Verfahren dauert seine Zeit (der FMON ist nicht abgestürzt!). Wenn Sie Ihr Programm nach dem Speichern starten wollen, müssen Sie es erst mit »L« neu laden, da die Puffer benutzt werden.

L Laden von Speicherbereichen.

Format L "tttttttt"

Das Programm tttttttt wird in die Floppy geladen. Auch der L-Befehl braucht seine Zeit!

Sichern und laden Sie keine Speicherbereiche, die länger als 4 KByte sind (ist ja auch gar nicht notwendig).

F Füllen von Bereichen.

Format F aaaa, bbbb, cc

Der Bereich aaaa bis bbbb wird mit dem Byte cc vollgeschrieben.

Q Beschleunigen des Schreib-Lesekopfs (64'er Ausgabe 4/85).

Format: Q oder Qaa

Nur »Q« beschleunigt und »Qaa« schaltet wieder auf normal um, wobei aa ein beliebiges Zeichen sein kann.

N Drückt eine Datei.

Format: N "tttttttt, X"

Druckt die Datei tttttttt (Dateityp X).

(M. Köhler/P. Baumann/aw)

Absturzbefehle werden durch ABS gekennzeichnet.
 No operation (1 Byte) werden bis auf NOP mit NO1 gekennzeichnet.
 No operation (2 Byte) werden mit NO2 bezeichnet.
 No operation (3 Byte) werden mit NO3 angezeigt.
 Wenn man diese Befehle beim A-Befehl benutzt, wird der erste
 jeweils mögliche Befehl genommen (bei ABS: 02).

Besondere Schreibweisen:

```
ASL:ORA    ASO
ROL:AND    RAN
LSR:EOR    LSE
ROR:ADC    RAD
DEC:CMP    DEM
INC:SBC    INB
```

Ansonsten wird die übliche 6502-Schreibweise benutzt.
 Hinter den Opcodes steht der Name des Befehls und danach die
 Adressierungsart: ZO = ZEROPAGE; ZOX = ZEROPAGE, X; ZOY =
 ZEROPAGE, Y; IMM = IMMEDIATE; INX = INDIRECT, X; INY = INDI-
 RECT, Y; ABS = ABSOLUT; ABSX = ABSOLUT, X; ABSY = ABSO-
 LUT, Y;

```
0B =ANN (ZO)
2B =ANN (ZO)
4B =ANL (IMM)
6B =DAR (IMM)
83 =AXX (INX)
87 =AXR (ZO)
8B =TAN (IMM)
8F =AAX (ABS)
93 =AXY (INY)
97 =AXI (ZOX)
9B =AXS (ABSY)
9C =YXA (ABSX)

9E =XYA (ABSY)
9F =AYY (ABSY)
A3 =LDT (INX)
A7 =LTA (ZO)
AB =??? (Wirkung noch
nicht ermittelt)
AF =LTX (ABS)
B3 =LYT (INY)
B7 =LAX (ZOY)
BB =TSA (ABSY)
BF =LYX (ABSY)
CB =XAS (IMM)
EB =SBC (IMM (wie der
normale Befehl
SBC))
```

Tabelle 1. Eingabehinweise und besondere Schreibweisen des FMON 1541

Rückmeldungen in der Adresse 0000 bei R/W/J (Fehlermeldungen)

```
$01 = Fehlerfreie Durchführung (00, OK)
$02 = Blockheader nicht gefunden (20, Read Error)
$03 = Sync-Markierung nicht gefunden (21, Read Error)
$04 = Datenblock nicht gefunden (22, Read Error)
$05 = Datenprüfsumme ist falsch (23, Read Error)
$07 = Fehler nach einem Verify (25, Write Error)
$08 = Diskette ist schreibgeschützt (26, Write Protect on)
$09 = Prüfsumme im Header falsch (27, Read Error)
$0A = Datenblock auf Diskette zu lang (28, Write Error)
$0B = Falsche ID im Block Header (29, Disk ID Mismatch)
$0F = Keine Diskette im Laufwerk (74, Drive not Ready)
$10 = Fehler bei Decodierung (24, Read Error)
```

Tabelle 2. Die Fehlermeldungen in der Adresse 0000 werden mit dieser Tabelle entschlüsselt

programm : f-mon 1541 8000 94b4

```
8000 : 20 44 e5 a2 04 86 7a a2 b4
8008 : 08 86 02 a2 00 8e 86 02 bb
8010 : a2 0d 8e 21 d0 a2 05 8e 54
8018 : 20 d0 4c 2e 82 a9 24 a0 c0
8020 : 80 4c 1e ab 0d 20 20 20 56
8028 : 20 20 20 20 20 20 2a 2a 64
8030 : 2a 2a 20 46 2d 4d 4f 4e 57
8038 : 20 31 35 34 31 20 2a 2a d6
8040 : 2a 2a 0d 20 20 20 20 28 9b
8048 : 43 29 20 31 39 38 35 20 b9
8050 : 42 59 20 4d 2e 4b 4f 45 f6
8058 : 48 4c 45 52 20 50 2e 42 24
8060 : 41 55 4d 41 4e 4e 0d 00 53
8068 : 48 a2 08 0a 48 a9 30 90 f8
8070 : 02 a9 31 c8 68 ca d0 f3 b4
8078 : c8 68 48 4a 4a 4a 20 30
8080 : 83 80 68 29 0f c9 0a 90 0b
8088 : 02 69 06 69 30 4c d2 ff 9e
8090 : 48 a9 20 20 d2 ff 68 60 48
8098 : 18 69 30 20 d2 ff 60 48 b4
80a0 : a9 2e 20 d2 ff 68 60 48 18
80a8 : a9 3b 20 d2 ff 68 60 48 a7
80b0 : a9 2c 20 d2 ff 68 60 48 27
80b8 : a9 3a 20 d2 ff 68 60 48 36
80c0 : a9 20 20 d2 ff a9 3f 20 66
80c8 : d2 ff 68 60 c9 30 90 05 2a
80d0 : c9 47 b0 01 60 68 68 20 b4
80d8 : bf 80 4c 31 82 c9 3a 08 80
80e0 : 29 0f 28 90 02 69 08 60 f9
80e8 : 0a 0a 0a 0a 60 48 a9 0d c4
80f0 : 20 d2 ff 68 60 20 e4 ff 21
80f8 : f0 fb 60 a9 01 a6 02 a0 c2
8100 : 0f 20 ba ff a9 00 20 bd 64
8108 : ff 4c c0 ff a2 01 4c c9 54
8110 : ff 20 cc ff a9 01 4c c3 ae
8118 : ff a2 01 4c c6 ff 4d 2d 2e
8120 : 45 00 4d 2d 52 00 4d 2d 13
8128 : 57 00 55 4a 49 55 31 3a 96
8130 : 00 55 32 00 23 85 fb 84 bf
8138 : fc a0 00 b1 fb f0 07 20 5e
8140 : d2 ff c8 4c 3b 81 60 a5 5a
8148 : d0 f0 07 20 cf ff c9 0d 94
8150 : d0 04 a9 20 18 60 38 60 b7
8158 : 48 a5 22 20 d2 ff a5 23 09
8160 : 20 d2 ff 68 60 a5 23 cd 52
8168 : 35 03 b0 01 60 a5 22 cd c2
8170 : 34 03 b0 01 60 68 68 4c f6
8178 : bb 82 a5 cb c9 3f f0 03 b7
8180 : 4c e9 82 20 11 81 4c 31 16
8188 : 82 48 ad 8d 02 c9 01 f0 a0
8190 : f9 68 60 20 cf ff c9 2c 56
8198 : d0 01 60 4c bb 82 20 2c 33
81a0 : 50 2c 57 00 20 2c 50 2c d9
81a8 : 52 00 a9 91 20 d2 ff 60 f0
81b0 : 20 cf ff c9 30 90 e4 c9 a0
81b8 : 39 b0 e0 38 e9 30 60 20 6a
81c0 : cf ff c9 30 90 d5 c9 31 48
81c8 : b0 d1 4c bb 81 a2 ff a0 5a
81d0 : ff ca d0 fd 88 d0 fa 60 e4
81d8 : a9 22 a0 81 20 35 81 20 dd
81e0 : 58 81 a9 0d 4c d2 ff 20 a0
81e8 : cf ff c9 20 d0 01 60 68 95
81f0 : 68 4c bb 82 4d 2d 57 00 59
81f8 : 00 02 00 00 0d 5f 85 fb d3
8200 : 84 fc a0 00 b1 fb c9 5f 0b
8208 : f0 06 20 a8 ff c8 d0 f4 8c
8210 : 60 48 8a 20 28 82 20 e8 24
8218 : 80 8d 34 03 68 20 28 82 fa
8220 : 18 6d 34 03 8d 34 03 60 a4
8228 : 20 cc 80 4c dd 80 20 1d f5
8230 : 80 20 ed 80 20 9f 80 20 8d
8238 : cf ff c9 0d f0 f3 c9 2e 4d
8240 : f0 f5 c9 20 f0 f1 a2 00 ca
8248 : 85 b6 bd e4 8c c5 b6 f0 e8
8250 : 0e c9 00 f0 04 e8 4c 4a ae
8258 : 82 20 bf 80 4c 31 82 8a 58
8260 : 0a aa bd ff 8c 85 22 bd 28
8268 : 00 8d 85 23 6c 22 00 20 0d
8270 : cf ff c9 20 f0 f9 20 28 65
8278 : 82 20 e8 80 8d 34 03 20 1b
8280 : cf ff 20 28 82 18 6d 34 63
8288 : 03 8d 34 03 60 20 6f 82 89
8290 : 8d 35 03 20 6f 82 20 fb 00
8298 : 80 20 0c 81 a9 1e a0 81 6d
```

Listing 1. Der FMON 1541 muß mit dem MSE eingegeben werden. Bitte beachten Sie die Eingabehinweise auf Seite 8.

Floppy-DOS-Erweiterung

```

82a0 : 20 35 81 ad 34 03 20 d2 f2
82a8 : ff ad 35 03 20 d2 ff 20 04
82b0 : ed 80 20 cc ff 20 11 81 c7
82b8 : 4c 31 82 4c d5 80 20 72 8e
82c0 : 82 85 23 20 6f 82 85 22 37
82c8 : 20 cf ff c9 2c d0 0c 20 c3
82d0 : 6f 82 8d 35 03 20 6f 82 7e
82d8 : 4c e3 82 a9 ff 8d 34 03 2f
82e0 : 8d 35 03 20 65 81 20 fb a8
82e8 : 80 20 0c 81 a9 22 a0 81 dd
82f0 : 20 35 81 20 58 81 a9 08 57
82f8 : 20 d2 ff 20 cc ff 20 19 05
8300 : 81 20 ed 80 20 9f 80 20 5e
8308 : b7 80 a5 23 20 68 80 a5 60
8310 : 22 20 68 80 20 70 80 a2 3a
8318 : 08 8a 48 20 e4 ff 85 07 ee
8320 : 20 68 80 68 48 aa a5 07 20
8328 : 9d 3c 03 20 90 80 68 aa ac
8330 : ca d0 e6 20 bf 8d 18 a5 34
8338 : 22 6f 08 85 22 a5 23 69 70
8340 : 00 85 23 a5 23 cd 35 03 fc
8348 : 90 0d f0 02 b0 0f a5 22 3a
8350 : cd 34 03 90 02 b0 06 20 08
8358 : 89 81 4c 7a 81 20 11 81 65
8360 : 4c 31 82 20 fb 80 20 0c 46
8368 : 81 20 6f 82 85 23 20 6f f6
8370 : 82 85 22 a9 26 a0 81 20 20
8378 : 35 81 a5 22 20 d2 ff a5 ff
8380 : 23 20 d2 ff a9 08 20 d2 69
8388 : ff a2 08 20 6f 82 9d 3c d8
8390 : 03 20 d2 ff ca d0 f4 a9 b2
8398 : 0d 20 d2 ff 20 11 81 20 3b
83a0 : 90 80 20 bf 8d 4c 31 82 75
83a8 : 20 6f 82 85 02 4c 31 82 1d
83b0 : 20 fb 80 20 0c 81 20 cf df
83b8 : ff 20 d2 ff c9 0d d0 f6 b2
83c0 : 20 11 81 4c cf 83 20 aa 42
83c8 : 81 20 11 81 4c bb 82 20 bb
83d0 : 9f 80 20 cc ff 20 fb 80 43
83d8 : 20 19 81 20 cf ff 20 d2 0c
83e0 : ff a5 90 c9 40 d0 f4 20 ae
83e8 : 11 81 20 aa 81 4c 31 82 5b
83f0 : 20 ed 80 4c cf 83 20 ed 26
83f8 : 80 20 9f 80 a5 bb a9 fb 57
8400 : 85 bb a9 24 85 fb a9 00 31
8408 : 85 bc a9 08 85 ba a9 01 2e
8410 : 85 b7 a9 60 85 b9 20 d5 3a
8418 : f3 a5 ba 20 b4 ff a5 b9 e6
8420 : 20 96 ff a9 00 85 90 a0 70
8428 : 03 84 fb 20 a5 ff 85 fc db
8430 : a4 90 d0 37 20 a5 ff a4 b0
8438 : 90 d0 30 a4 fb 88 d0 e9 ec
8440 : a6 fc 20 cd bd a9 20 20 10
8448 : d2 ff 20 a5 ff a6 90 d0 f0
8450 : 1a aa f0 05 20 d2 ff d0 d6
8458 : f1 a9 0d 20 d2 ff 20 9f 52
8460 : 80 20 e4 ff d0 05 ea a0 4b
8468 : 02 d0 be 20 42 f6 20 aa 38
8470 : 81 4c 31 82 a9 02 20 c3 67
8478 : ff a9 02 a6 02 a0 0f 20 43
8480 : ba ff a9 02 a2 a0 81 e5
8488 : 20 bd ff 20 c0 ff 20 00 17
8490 : 8e 20 e7 ff 4c 31 82 a9 d4
8498 : 02 20 c3 ff a9 02 a6 02 e4
84a0 : a0 0f 20 ba ff a9 01 a2 be
84a8 : 2c a0 81 20 bd ff 20 c0 67
84b0 : ff 4c 31 82 20 6f 82 8d 15
84b8 : fa 81 20 6f 82 8d fb 81 f0
84c0 : 20 a6 8d a5 02 20 b1 ff 33
84c8 : a9 6f 20 93 ff a9 06 8d 24
84d0 : ff 81 a9 00 8d fb 81 a9 ec
84d8 : f4 a0 81 20 fe 81 a5 02 17
84e0 : 20 ae ff 60 20 b4 84 a5 68
84e8 : 02 20 b1 ff a9 6f 20 93 24
84f0 : ff a9 80 8d a3 8d a9 9d 1e
84f8 : a0 8d 20 fe 81 a5 02 20 d4
8500 : ae ff 20 11 81 4c 31 82 1c
8508 : 20 b4 84 a5 02 20 b1 ff 40
8510 : a9 6f 20 93 ff a9 90 8d 96
8518 : a3 8d 4c f6 84 20 6f 82 80
8520 : 8d a3 8d 20 93 81 20 b4 15
8528 : 84 a5 02 20 b1 ff a9 6f a4
8530 : 20 93 ff 4c f6 84 4c bb df
8538 : 82 20 cf ff c9 22 f0 09 42
8540 : 20 72 82 a8 a9 01 4c 3c 9b
8548 : 87 20 cf ff 4c 43 85 4c 61
8550 : 31 82 20 6f 82 85 3e 20 46
8558 : 6f 82 85 3d 20 54 8e a0 31
8560 : 00 20 cf ff c9 20 f0 f9 b9
8568 : c9 2e f0 cd c9 0d f0 df c7
8570 : 99 3c 03 c8 20 cf ff c9 15
8578 : 20 f0 f9 c9 24 f0 15 99 9d
8580 : 3c 03 c9 0d f0 03 c8 d0 3e
8588 : eb a0 03 b9 3c 03 c9 23 05
8590 : f0 12 c9 28 f0 5c c9 0d 34
8598 : f0 2a c9 41 f0 4f 20 cc db
85a0 : 80 4c 1a 86 a0 04 b9 3c 27

```

```

85a8 : 03 c9 22 f0 09 c9 25 f0 8c
85b0 : 10 a9 0c 4c 4b 87 c8 b9 a9
85b8 : 3c 03 85 4e a9 0c 4c 01 cf
85c0 : 87 4c 5f 8e ad 3c 03 c9 73
85c8 : 41 f0 14 c9 4c f0 1e c9 18
85d0 : 52 d0 07 ad 3d 03 c9 54 be
85d8 : d0 13 a9 00 4c 01 87 ad e2
85e0 : 3d 03 c9 53 d0 e5 ad 3e eb
85e8 : 03 c9 4c d0 de a9 01 4c d5
85f0 : 01 87 a0 07 b9 3c 03 c9 db
85f8 : 58 f0 10 a0 08 b9 3c 03 26
8600 : c9 59 f0 0c c9 29 f0 0d f7
8608 : 4c bb 82 a9 05 4c 4b 87 f7
8610 : a9 04 4c 4b 87 a9 02 4c 9e
8618 : 54 87 a0 01 b7 3c 03 c9 95
8620 : 49 f0 08 88 b9 3c 03 c9 12
8628 : 42 f0 12 4c 96 86 a0 04 19
8630 : b9 3c 03 48 88 b9 3c 03 1e
8638 : aa 68 4c 11 82 20 2e 86 3b
8640 : 85 a0 a0 06 20 30 86 85 77
8648 : 3f 18 a5 3d 6f 02 85 41 e4
8650 : a5 3e 69 00 85 42 a5 40 f0
8658 : c5 42 90 1f f0 02 b0 06 34
8660 : a5 3f c5 41 90 15 38 a5 1c
8668 : 3f e5 41 85 4e a5 40 e5 7a
8670 : 42 18 65 4e 85 4e a9 03 59
8678 : 4c 01 87 38 a5 41 e5 3f a8
8680 : 85 4e a5 42 e5 40 18 65 69
8688 : 4e 85 4e a9 ff 88 e5 4e 58
8690 : 18 69 01 4c 74 86 a0 06 31
8698 : b9 3c 03 c9 59 f0 19 c9 7e
86a0 : 58 f0 19 88 b9 3c 03 c9 e5
86a8 : 0d f0 15 a0 07 b9 3c 03 bc
86b0 : c9 0d d0 10 a9 0b d0 3b e3
86b8 : a9 06 d0 2d a9 07 d0 29 a6
86c0 : a9 08 d0 25 a0 05 b9 3c d8
86c8 : 03 c9 0d d0 04 a9 08 d0 5d
86d0 : 18 a0 08 b9 3c 03 c9 58 25
86d8 : f0 07 c9 59 f0 07 4c 08 72
86e0 : 86 a9 0a d0 0e a9 09 d0 cb
86e8 : 0a 48 20 2e 86 85 4e 68 83
86f0 : 4c 01 87 48 20 2e 86 85 40
86f8 : 4f a0 06 20 30 86 85 4e 07
8700 : 68 85 57 a0 a0 02 00 b9 9d
8708 : a5 9d d0 3c 03 f0 06 c8 56
8710 : d0 f5 4c bb 82 e8 b9 a5 07
8718 : 91 dd 3c 03 08 ca 28 d0 20
8720 : ee e8 e8 b9 a5 92 dd 3c d2
8728 : 03 08 ca ca 28 d0 e0 b9 3b
8730 : a5 93 c5 57 d0 d9 a6 57 20
8738 : bd a5 94 aa 48 98 85 4d 3c
8740 : 8d 01 02 68 48 8d 00 02 d0
8748 : 4c 67 87 48 a0 05 20 30 46
8750 : 86 4c ed 86 48 a0 05 20 26
8758 : 30 86 85 4f a0 06 20 30 32
8760 : 86 85 4e 68 4c 01 87 a5 80
8768 : 4e 8d 02 02 a5 4f 8d 03 4f
8770 : 02 20 9b 87 68 85 22 18 c6
8778 : a5 3d 65 22 85 3d a5 3e af
8780 : 69 00 85 3e a9 4f a0 8e c7
8788 : 20 35 81 a5 3e 20 68 80 df
8790 : a5 3d 20 68 80 20 90 a0 35
8798 : 4c 5c 85 20 fb 80 a5 3d 4c
87a0 : c9 5f f0 2c 8d 35 8e a5 e2
87a8 : 3e 8d 36 8e ad 00 02 8d 0a
87b0 : 37 8e 20 0c 81 a9 32 a0 27
87b8 : 8e 20 29 90 a0 00 b9 01 a6
87c0 : 02 20 d2 ff c8 cc 00 02 7e
87c8 : d0 f4 20 ed 80 4c 11 81 8a
87d0 : a9 5e a4 3e 20 9b 8e 48 43
87d8 : a9 5e a4 3e 8d 35 8e 8c 77
87e0 : 36 8e ae 00 02 e8 8e 37 19
87e8 : 8e 20 0c 81 a9 32 a0 8e 85
87f0 : 20 29 90 68 20 d2 ff 4c 07
87f8 : bc 87 a9 00 85 53 20 cf f5
8800 : ff c9 0d f0 1e 20 72 82 f7
8808 : 85 3e 20 6f 82 85 3d 20 2c
8810 : cf ff c9 2c d0 15 20 6f ec
8818 : 82 85 3c 20 6f 82 85 3b 08
8820 : 4c 31 88 a9 00 85 3d a9 d1
8828 : 03 85 3e a9 ff 85 3b 85 d7
8830 : 3c 20 87 8e 20 ed 80 20 e4
8838 : 9f 80 20 af 80 a5 3e 20 84
8840 : 68 80 a5 3d 20 68 80 20 81
8848 : 90 80 a5 3d a4 3e 20 9b 1d
8850 : 8e 85 4e 20 68 80 e6 3d d9
8858 : d0 02 e6 3e a5 4e a8 b9 8e
8860 : a5 90 8d 3c 03 b9 a5 91 f0
8868 : 8d 3d 03 b9 a5 92 8d 3e 2d
8870 : 03 b9 a5 93 85 50 aa bd 2d
8878 : a5 94 85 4f c9 01 f0 26 67
8880 : a5 3d a4 3e 20 9b 8e 8d e9
8888 : 3f 03 e6 3d d0 02 e6 3e df
8890 : a5 4f c9 02 f0 10 a5 3d 30
8898 : a4 3e 20 9b 8e 8d 40 03 33
88a0 : e6 3d d0 02 e6 3e a5 4f 2f
88a8 : c9 01 f0 18 20 70 80 ad 15

```

```

88b0 : 3f 03 20 68 80 a5 4f c9 8c
88b8 : 02 f0 09 20 90 80 ad 40 bd
88c0 : 03 20 68 80 20 c3 8e ad b3
88c8 : 3c 03 20 d2 ff ad 3d 03 51
88d0 : 20 d2 ff ad 3e 03 20 d2 31
88d8 : ff 20 90 80 20 cb 8e a5 01
88e0 : 4f c9 01 f0 1c c9 02 d0 2c
88e8 : 09 ad 3f 03 20 68 80 4c d8
88f0 : fe 88 ad 40 03 20 68 80 7a
88f8 : ad 3f 03 20 68 80 20 18 45
8900 : 8f ad 8d 02 c9 01 f0 f9 66
8908 : a5 cb c9 3f f0 15 a5 3e b8
8910 : c5 3c 90 c0 f0 02 b0 0b 91
8918 : a5 3d c5 3b f0 02 b0 03 1c
8920 : 4c 34 88 20 11 81 4c 31 5d
8928 : 82 a9 80 85 53 20 cf ff c5
8930 : c9 2c f0 f5 20 72 82 85 b5
8938 : 3e 20 6f 82 85 3d 20 cf 15
8940 : ff c9 0d d0 03 4c 31 82 de
8948 : 20 72 82 85 4e 8d 01 02 4c
8950 : a8 b9 a5 90 8d 3c 03 b9 8a
8958 : a5 91 8d 3d 03 b9 a5 92 8b
8960 : 8d 3e 03 b9 a5 93 85 50 b2
8968 : aa bd a5 94 85 4f 8d 00 f6
8970 : 02 c9 01 f0 18 20 6f 82 fb
8978 : 8d 3f 03 8d 02 02 a5 4f 7d
8980 : c9 02 f0 09 20 6f 82 8d 4a
8988 : 40 03 8d 03 02 20 c3 8e 5b
8990 : ad 3c 03 20 d2 ff ad 3d 7f
8998 : 03 20 d2 ff ad 3e 03 20 79
89a0 : d2 ff 20 90 80 20 cb 8e e1
89a8 : a5 4f c9 01 f0 13 c9 02 5a
89b0 : f0 06 ad 40 03 20 68 80 eb
89b8 : ad 3f 03 20 68 80 20 18 05
89c0 : 8f 20 9b 87 a2 0a 20 90 53
89c8 : 80 ca d0 fa 20 ed 80 20 f4
89d0 : 9f 80 20 af 80 a5 3d 18 08
89d8 : 6d 00 02 85 3d a5 3e 69 43
89e0 : 00 85 3e 20 68 80 a5 3d d2
89e8 : 20 68 80 20 90 80 4c 3e 1b
89f0 : 89 20 3c 90 a0 00 91 a7 4a
89f8 : 4c 23 8a 20 94 8f 20 d5 70
8a00 : 8f 8a 10 20 93 81 20 7f 8f
8a08 : 8f 20 fb 80 a2 00 86 a7 4a
8a10 : a9 c0 85 a8 a5 22 a4 23 d4
8a18 : c9 5f f0 d5 20 9b 8e a0 e2
8a20 : 00 91 a7 e6 a7 d0 02 e6 86
8a28 : a8 a5 23 c5 25 90 06 a5 5f
8a30 : 22 c5 24 b0 08 e6 22 d0 36
8a38 : db e6 23 d0 d7 20 11 81 2f
8a40 : a9 c0 85 a8 a9 00 85 a7 c0
8a48 : 4c 60 8a a9 01 a6 02 a0 2b
8a50 : 62 20 ba ff a5 10 a2 3c 4f
8a58 : a0 03 20 bd ff 4c c0 ff 9f
8a60 : 20 4b 8a 20 0c 81 a5 05 3a
8a68 : 20 d2 ff a5 06 20 d2 ff 53
8a70 : a0 00 b1 a7 20 d2 ff e6 d8
8a78 : a7 d0 02 e6 a8 24 05 c5 40
8a80 : 25 90 06 a5 05 c5 24 b0 94
8a88 : 08 e6 05 d0 e3 e6 06 d0 8e
8a90 : df 20 cc ff 20 11 81 20 83
8a98 : ed 80 20 9f 80 4c 0f 83 f7
8aa0 : 20 b4 8f 20 82 8f 84 10 df
8aa8 : a9 00 85 a7 a9 c0 85 a8 b0
8ab0 : 20 4b 8a 20 19 81 20 e4 0a
8ab8 : ff 85 22 20 e4 ff 85 23 b1
8ac0 : 20 e4 ff a0 00 91 a7 e6 5f
8ac8 : a7 d0 02 e6 a8 24 90 50 c3
8ad0 : ef 20 11 81 20 fb 80 a9 7b
8ad8 : 00 85 24 a9 c0 85 25 a0 e7
8ae0 : 00 b1 24 a4 23 a6 22 a0 08
8ae8 : 5f f0 20 13 90 a5 25 62
8af0 : c5 a8 90 06 a5 24 c5 a7 d0
8af8 : b0 0e e6 24 d0 02 e6 25 f0
8b00 : e6 22 d0 db e6 23 d0 d7 2f
8b08 : 4c 94 8a 20 65 90 4c ee 21
8b10 : 8a 20 f7 8f 20 93 81 20 7f
8b18 : 6f 82 85 07 20 fb 80 20 2f
8b20 : 0c 81 a5 22 c9 5f f0 33 5c
8b28 : 8d 2c 8e a5 23 8d 2d 8e 94
8b30 : a5 07 8d 2f 8e a9 29 a0 be
8b38 : 8e 20 29 90 20 cc ff a5 e6
8b40 : 23 c5 25 90 0c a5 22 c5 a3
8b48 : 24 90 06 20 11 81 4c 31 eb
8b50 : 82 e6 22 d0 ca e6 23 d0 fa
8b58 : c6 f0 f0 a9 5e a4 23 20 df
8b60 : 9b 8e 48 a9 5e 8d 35 8e ce
8b68 : a5 23 8d 36 8e a9 02 8d 22
8b70 : 37 8e 20 0c 81 a9 32 a0 e7
8b78 : 8e 20 29 90 68 20 d2 ff 4c
8b80 : a5 07 20 d2 ff 20 ed 80 c5
8b88 : a9 60 4c 28 8b 20 cf ff 72
8b90 : c9 0d d0 22 a9 0f 8d 2f 00
8b98 : 8e a9 07 8d 2c 8e a9 1c 84
8ba0 : 8d 2d 8e 20 fb 80 20 0c c8
8ba8 : 81 a9 29 a0 8e 20 29 90 0c
8bb0 : 20 11 81 4c 31 82 a9 3a 85

```

| Stichwort | Titel | Seite | Ausgabe | |
|--|--|--------------------|---------|-------|
| Kurse | | | | |
| Assembler | Assembler ist keine Alchimie (Teil 1) | 138 | 9/84 | |
| | Assembler ist keine Alchimie (Teil 2) | 150 | 10/84 | |
| | Assembler ist keine Alchimie (Teil 3) | 121 | 11/84 | |
| | Assembler ist keine Alchimie (Teil 4) | 134 | 12/84 | |
| | Assembler ist keine Alchimie (Teil 5) | 142 | 1/85 | |
| | Assembler ist keine Alchimie (Teil 6) | 134 | 2/85 | |
| Codes | Alle Tasten, Zeichen- und Steuer-codes (Teil 1) | 114 | 4/84 | |
| | Alle Tasten, Zeichen- und Steuer-codes (Teil 2) | 104 | 5/84 | |
| | Alle Tasten, Zeichen- und Steuer-codes (Teil 3) | 146 | 7/84 | |
| | Alle Tasten, Zeichen- und Steuer-codes (Teil 4) | 151 | 8/84 | |
| Comal | Comal - Eine Einführung (Teil 1) | 44 | 11/84 | |
| | Comal - Eine Einführung (Teil 2) | 145 | 12/84 | |
| Eff. Prog. | Comal - Eine Einführung (Teil 3) | 130 | 2/85 | |
| | Finden mit System - Eine neuartige Suchmethode (Teil 3) | 148 | 3/85 | |
| Floppy | Müllabfuhr im Computer: Die Garbage Collection (Teil 1) | 122 | 1/85 | |
| | Stringprogrammierung in Maschinensprache (Teil 2) | 147 | 2/85 | |
| Grafik | In die Geheimnisse der Floppy eingetaucht (Teil 1) | 153 | 10/84 | |
| | In die Geheimnisse der Floppy eingetaucht (Teil 2) | 117 | 11/84 | |
| | In die Geheimnisse der Floppy eingetaucht (Teil 3) | 139 | 12/84 | |
| | In die Geheimnisse der Floppy eingetaucht (Teil 4) | 148 | 1/85 | |
| | In die Geheimnisse der Floppy eingetaucht (Teil 5) | 130 | 3/85 | |
| Grundlagen | Hires 3 (Teil 1) | 123 | 2/85 | |
| | Hires 3 (Teil 2) | 136 | 3/85 | |
| Musik | Reise durch die Wunderwelt der Grafik (Teil 1) | 119 | 4/84 | |
| | Reise durch die Wunderwelt der Grafik (Teil 2) | 109 | 5/84 | |
| | Reise durch die Wunderwelt der Grafik (Teil 3) | 132 | 6/84 | |
| | Reise durch die Wunderwelt der Grafik (Teil 4) | 162 | 7/84 | |
| | Reise durch die Wunderwelt der Grafik (Teil 5) | 142 | 8/84 | |
| | Reise durch die Wunderwelt der Grafik (Teil 6) | 144 | 9/84 | |
| Precompiler | Reise durch die Wunderwelt der Grafik (Teil 7) | 146 | 10/84 | |
| | Geschwindigkeit durch Maschinencode - so arbeiten Compiler | 39 | 2/85 | |
| Speicher | Dem Klang auf der Spur (Teil 1) | 131 | 12/84 | |
| | Dem Klang auf der Spur (Teil 2) | 136 | 1/85 | |
| | Dem Klang auf der Spur (Teil 3) | 152 | 2/85 | |
| | Strubs - ein Precompiler für Basic-Programme (Teil 1) | 110 | 4/84 | |
| VC 20 | Strubs - ein Precompiler für Basic-Programme (Teil 2) | 116 | 5/84 | |
| | Strubs - ein Precompiler für Basic-Programme (Teil 3) | 128 | 6/84 | |
| | Strubs - ein Precompiler für Basic-Programme (Teil 4) | 154 | 7/84 | |
| | Memory Map mit Wandervorschlägen (Teil 1) | 133 | 11/84 | |
| | Memory Map mit Wandervorschlägen (Teil 2) | 132 | 12/84 | |
| Grundlagen | Memory Map mit Wandervorschlägen (Teil 3) | 127 | 1/85 | |
| | Memory Map mit Wandervorschlägen (Teil 4) | 150 | 2/85 | |
| | Memory Map mit Wandervorschlägen (Teil 5) | 144 | 3/85 | |
| | Der gläserne VC 20 (Teil 1) | 155 | 9/84 | |
| | Der gläserne VC 20 (Teil 2) | 157 | 10/84 | |
| | Der gläserne VC 20 (Teil 3) | 126 | 11/84 | |
| | Der gläserne VC 20 (Teil 4) | 130 | 1/85 | |
| | Der gläserne VC 20 (Teil 5) | 141 | 2/85 | |
| | Der gläserne VC 20 (Teil 6) | 155 | 3/85 | |
| | Software-Test | Abenteuer | | |
| Abenteuer selbst gemacht - Adventure Creator | | 43 | 11/84 | |
| Abenteuer | Die Lösung von Hobbit | 49 | 2/85 | |
| | Exodus - Ultima III | 42 | 11/84 | |
| | Gordon Saga | 48 | 2/85 | |
| | Gruds in Space | 137 | 8/84 | |
| | House of Usher | 37 | 10/84 | |
| | Lösung von Dallas Quest | 92 | 1/85 | |
| | Lösung von Enchanter | 44 | 3/85 | |
| | Lösung von The Blade of Blackpool | 34 | 10/84 | |
| | The Quest | 47 | 1/85 | |
| | Action | Dino Eggs | 57 | 6/84 |
| | | Flip and Flop | 48 | 4/84 |
| | | Impossible Mission | 46 | 2/85 |
| | | Loderunner | 68 | 7/84 |
| | | OX 3, Catastrophes | 48 | 12/84 |
| | | Raingame | 52 | 8/84 |
| Save New York and Survivor | | 46 | 4/84 | |
| Tom + Zaga | | 48 | 1/85 | |
| Wizard | | 49 | 12/84 | |
| Zaxxon | | 68 | 7/84 | |
| Arcade | AE - ein Action-Spiel | 56 | 6/84 | |
| | Fire Galaxy (VC 20) | 37 | 10/84 | |
| Flipper | Schnellboot - Rettung aus der grünen Hölle | 109 | 9/84 | |
| | David's Midnight Magic | 60 | 5/84 | |
| | Night Mission | 61 | 5/84 | |
| Grundlagen | Slamball - der ellenlange Flipper | 105 | 9/84 | |
| | Fantasy-Spiele | 106 | 9/84 | |
| Simulation | Flight II - fast wie richtiges Fliegen | 69 | 7/84 | |
| | One on One | 136 | 8/84 | |
| Sport | Spiel des Jahres: International Soccer | 46 | 12/84 | |
| | Summer Games - Los Angeles läßt grünen | 138 | 8/84 | |
| Taktik | Taktik- und Strategiespiele | 46 | 3/85 | |
| | | | | |
| So machen's andere | | | | |
| Amateurfunk | Funkende Computer | 132 | 4/84 | |
| | Klein aber oho - der VC 20 | 136 | 4/84 | |
| Datenbank | Geregelter Zahlungsverkehr | 164 | 9/84 | |
| | Der VC 20 als Fotolehring | 138 | 6/84 | |
| Finanzen | Mode-Fotos mit Bits und Bytes | 144 | 11/84 | |
| | Der Computer im Kunstsal | 156 | 8/84 | |
| Landwirtschaft | Mit vier Baud über den Balkon | 166 | 10/84 | |
| | Computer bringen den Kreislauf in Schwung | 170 | 7/84 | |
| Medizin | Welche Hausnummer hat der Kölner Dom? | 128 | 5/84 | |
| | Wohn mit dem Heimcomputer? | 136 | 5/84 | |
| Orientierung | | | | |
| | | | | |
| Platzproblem | | | | |
| | | | | |
| Software-Test | | | | |
| Assembler | Assembler im Test (AS-64, MAE, TEX-AS, ASSI/M) | 34 | 1/85 | |
| | Assembler im Test: Mastercode, Profimat, Profisoft, Maschine 64 | 30 | 2/85 | |
| Basic-Extension | Assembler? Assembler! | 32 | 1/85 | |
| | Exbasic Level II - eine tolle Sache | 42 | 6/84 | |
| Büro | GBasic | 28 | 1/85 | |
| | Simons Basic (Teil 1) | 40 | 4/84 | |
| CP/M | Simons Basic (Teil 2) | 42 | 5/84 | |
| | Magic Desk - Bildschirm statt Schreibbüsch | 62 | 7/84 | |
| Compiler | Erste Erfahrungen mit dem CP/M-Modul | 18 | 4/84 | |
| | Basic-Programme auf Trab gebracht: Austro-Speed, BASS, Exbasic, Petspeed | 34 | 2/85 | |
| DFÜ | Teleterm - Die Verbindung zum Modem | 20 | 7/84 | |
| | Terminal 64 - Schwer auf Draht | 24 | 2/85 | |
| Datenbank | Terminal 64 - Schwer auf Draht | 24 | 2/85 | |
| | Daten gut im Griff (Datamag, Multidata, Datenmanager) | 52 | 5/84 | |
| Finanzen | ISM 64 - ohne Fleiß kein Preis | 117 | 9/84 | |
| | ISM 64 - professionelle Datenverwaltung | 59 | 5/84 | |
| Floppy | Mandart 64 | 58 | 5/84 | |
| | Superbase 64 | 46 | 5/84 | |
| Grafik | Lohnsteuerjahresausgleich leicht gemacht | 46 | 10/84 | |
| | Ex-DOS und Disk Doctor | 48 | 10/84 | |
| Grafik & Drucker | Quickcopy - das schnelle Kopierprogramm | 28 | 9/84 | |
| | Elektronische Aquarelle: Paint Magic | 114 | 8/84 | |
| | Grafik hoch zwei - das Extended Graphic System | 37 | 11/84 | |
| | Grafiktafeln: KoalaPad und SuperSketch | 34 | 11/84 | |
| Graphics-Basic (HES) | 38 | 12/84 | | |

| Stichwort | Titel | Seite | Ausgabe |
|---|---|-------|---------|
| Lernprogramme | Wie Super ist die Supergrafik? (Supergrafik 64) | 30 | 11/84 |
| | Melodienschreiber und Musik-Synththesizer | 43 | 12/84 |
| Mathematik | Nachhilfe auf Knopfdruck (Mathematik) | 26 | 2/85 |
| | SoftLearning - die weiche Welle des Lernens | 40 | 1/85 |
| | Vokaltraining mit dem Computer | 39 | 3/85 |
| | Was bringt die Lern-Software? | 42 | 12/84 |
| | Nachhilfe auf Knopfdruck (Mathematik) | 26 | 2/85 |
| | Gute Noten für gute Noten: Extended Synththesizer | 24 | 9/84 |
| Programmgenerator | System | 43 | 12/84 |
| | Melodienschreiber und Musik-Synththesizer | 50 | 10/84 |
| | Music-Composer - Komponieren leicht gemacht | 42 | 9/84 |
| | Musicalo - oder was wirklich im C 64 steckt | 29 | 9/84 |
| | Synthimat - Das Piano für den Aktenkoffer | 38 | 9/84 |
| | Basic Bar - Ein Programmgenerator | 65 | 7/84 |
| Sprachen | Die Turbo-Pascal-Story | 40 | 12/84 |
| | Forth ohne Floppy (C 64 und VC 20) | 10/84 | |
| | HES 64 Forth - komfortabler als Basic | 66 | 7/84 |
| | Oxford-Pascal für den Commodore | 39 | 12/84 |
| | Pascal - leistungsfähiger und eleganter als Basic (Teil 2) | 44 | 8/84 |
| | Sechs Pascal-Versionen im Vergleich | 50 | 8/84 |
| Tabellenkalkulation | Multiplan: Viel zu schade, um nur damit zu kalkulieren | 32 | 11/84 |
| | Calc Result - Dreidimensionale Kalkulation | 21 | 9/84 |
| Textverarbeitung | VizStar - Ein Stern wird geboren | 38 | 11/84 |
| | Blitztext - schnell wie der Blitz? | 64 | 6/84 |
| | Homeword - Textverarbeitung zu Hause | 36 | 3/85 |
| | SM-Text 64 - die professionelle Textverarbeitung | 48 | 6/84 |
| | Textomat - Büroanwendung zum kleinen Preis | 34 | 9/84 |
| | Tot-Text - Flexibilität ist Trumpf | 38 | 3/85 |
| Vokabeln | Vizawrite 64 - Der C 64 wird zum PC | 43 | 10/84 |
| | Wordpro 3 plus - Von den Großen auf den Kleinen | 52 | 6/84 |
| Software | Vokaltraining mit dem Computer | 39 | 3/85 |
| | | | |
| Adressenvergleich | Adressenvergleich VC 20-C 64 | 52 | 7/84 |
| | | | |
| Basic | Debugging - Fehlersuche in Basic-Programmen | 46 | 7/84 |
| | Fehlersuche in Basic-Programmen (Teil 2) | 67 | 9/84 |
| CP/M | CP/M-Software vom Apple auf den C 64 | 36 | 7/84 |
| | Geschwindigkeit durch Maschinencode - so arbeiten Compiler | 39 | 2/85 |
| DFÜ | Begriffe aus der DFÜ | 27 | 7/84 |
| | Datex-P und ausländische Netzwerke | 59 | 10/84 |
| EPROM | Electronic Mail - die neue Form der Postbeförderung | 22 | 7/84 |
| | Mailboxprogramm für den C 64 | 114 | 9/84 |
| Floppy | Terminalprogramm für den C 64 | 24 | 7/84 |
| | Wie bedient man eine Mailbox? | 29 | 7/84 |
| Grundlagen | Datenbrenner: Wie programmiere ich EPROMs? | 162 | 9/84 |
| | 22 Read Error - Theorie und Praxis | 41 | 3/85 |
| Grafik | Neues vom Video-Chip beim VC 20 | 56 | 8/84 |
| | Daten im (relativen) Direktzugriff | 58 | 7/84 |
| Grundlagen | Datenkreislauf: Die sequentielle Datenspeicherung | 63 | 8/84 |
| | Die index-sequentielle Datei | 54 | 9/84 |
| Musik | Flußdiagramm | 20 | 9/84 |
| | So macht man Basic-Programme schneller (Teil 2) | 44 | 12/84 |
| Sprachen | Strukturiertes Programmieren (Teil 1) | 33 | 8/84 |
| | Strukturiertes Programmieren (Teil 2) | 37 | 8/84 |
| Textverarbeitung | Tips für den Umgang mit Sinnbildern (Flußdiagrammen) | 14 | 9/84 |
| | Tips für sauberes Programmieren | 38 | 4/84 |
| Wettbewerbe | Hard und Soft: eine kleine Marktübersicht | 58 | 9/84 |
| | Klangsynthese und Synthesizer-technik | 62 | 9/84 |
| Musik | Marktübersicht der Musikprogramme | 27 | 9/84 |
| | Basic ist out - Es lebe Forth | 43 | 1/85 |
| Sprachen | Forth - die etwas andere Programmiersprache | 33 | 7/84 |
| | Pascal - leistungsfähiger und eleganter als Basic | 40 | 7/84 |
| Textverarbeitung | Pascal - leistungsfähiger und eleganter als Basic | 44 | 8/84 |
| | Was ist Comal? | 41 | 8/84 |
| Tips | Textverarbeitungsprogramme - Marktübersicht | 47 | 6/84 |
| | Von der Schreibmaschine zum Textsystem (Auswahlhilfe) | 34 | 3/85 |
| Wettbewerbe | DOS 5.1 (Teil 1) | 40 | 5/84 |
| | DOS 5.1 (Teil 2) | 16 | 9/84 |
| Einzeiler | Die Top 10 (Einzeiler) | 158 | 11/84 |
| | Schnell-Wettbewerb: Die nächsten 14 Kreuzworträtsel selber machen | 151 | 12/84 |
| Kreuzworträtsel | Das schönste Spritze - Auflösung | 174 | 7/84 |
| | Formatierte Eingabe | 156 | 1/85 |
| Unterprogramm | Export - Sortieren mit Komfort | 154 | 11/84 |
| | Sieger mit Maske - Maskenerstellungsprogramm | 172 | 10/84 |
| Alle Beiträge sind in der Regel für den C 64, sofern nicht anders gekennzeichnet (VC 20). | | | |
| Folgende Abkürzungen wurden verwendet: LDM = Listings des Monats. | | | |
| AdM = Anwendung des Monats; SB = Simons Basic | | | |

Auch die bisher erschienenen Sonderhefte können Sie jetzt direkt bestellen:



TIPS & TRICKS
(1. Programm-Sonderheft)
Eine wahre Fundgrube an Ideen und Programmen für Computer-Profis und alle, die es werden wollen.
BESTELLCODE: Tips & Tips



ABENTEUERSPIELE
(2. Programm-Sonderheft)
Auf mehr als 100 Seiten viele interessante Adventures, die Sie lange Zeit fesseln werden. Mit abgeschlossenem Kurs zur Programmierung eigener Abenteuerspiele und zahlreichen Lösungen professioneller Adventures.
BESTELLCODE: Abenteuerspiele



SPIELE
(3. Programm-Sonderheft)
Heiße Listings für alle Spiele-Fans: Sportspiele, Schießspiele, Denkspiele, Spielgeneratoren, Abenteuerspiele, Brettspiele, Taktikspiele, Geschicklichkeitsspiele und eine Marktübersicht aller in Deutschland erhältlichen professionellen Spiele bringen alles, was das Herz der Spiele-Fans höher schlagen läßt.
BESTELLCODE: Spiele



GRAFIK & DRUCKER
(4. Programm-Sonderheft)
Randvoll mit Informationen: Rund 28 Listings der Spitzenklasse. Darunter Top-Listings zur räumlichen Darstellung von Körpern aus beliebigen Betrachtungsrichtungen.
Weiters: Prüfsummenlistings, Drucker-Anwendung, Basic-Erweiterung, Hardcopy-Routinen, Zeichengenerator, Grundlagen, Tips & Tricks.
BESTELLCODE: Grafik & Drucker

Neu: die 64'er-Sammelbox!

Für alle Leser, die »64'er« regelmäßig kaufen, sammeln oder im Abonnement beziehen, gibt es jetzt ein interessantes Service-Angebot: die 64'er-Sammelbox!

Mit dieser Sammelbox bringen Sie nicht nur Ordnung in Ihre wertvollen Hefte, sondern schaffen sich gleichzeitig ein interessantes und attraktives Nachschlagewerk.

Übrigens: Die Sammelbox ist nicht nur ein praktisches Aufbewahrungsmittel: Sie eignet sich auch hervorragend als Geschenk für Freunde und Bekannte zu vielen Anlässen.

Ein kompletter Jahrgang (12 Hefte) paßt in die praktische Sammel-Box! Am besten gleich bestellen!



Directory-Editor mit Komfort

Mit diesem Programm gestalten Sie das Disketten-Directory ganz nach Ihren Wünschen.

Mit dem Directory-Editor kann man das Inhaltsverzeichnis einer Diskette auf komfortable Art und Weise ändern. Das Programm ist lauffähig auf dem Commodore 64 sowie seinen »großen Brüdern« CBM 2001 bis 8001. Es erkennt Computer- und Floppy-Laufwerk-Typ und sollte ohne Schwierigkeiten auch an andere und zukünftige Commodore-Computer und -Floppy-Laufwerke anzupassen sein.

Das Programm wird mit RUN gestartet. Es dauert einige Sekunden, bis Computer- und Floppy-Laufwerk-Typ erkannt und die Variablen entsprechend gesetzt sind.

Befehlsbeschreibung

Tabelle 1 zeigt eine Übersicht über alle Befehle des Directory-Editors. Die Befehle im einzelnen: Durch »A« oder »SPACE« (Leertaste) erhöht man eine zweiseitige Kurzanleitung mit einer Befehlsübersicht (weiter mit beliebiger Taste).

Mit »0« und »1« kann man bei einem angeschlossenen Doppelfloppy-Laufwerk zwischen den beiden Laufwerken wählen. Falls ein Doppelfloppy-Laufwerk angeschlossen ist, wird beim Start des Programms automatisch Laufwerk 1 gewählt, so daß sich folgende Laufwerksaufteilung anbietet: Laufwerk 0 mit Systemdiskette und Laufwerk 1 mit der zu bearbeitenden Diskette.

»L« dient zur Umschaltung zwischen Groß- und Kleinschreibung. Mit »X« kann das Programm jederzeit verlassen werden. Wird gerade eine Diskette bearbeitet, so wird zunächst automatisch der Befehl »E« ausgeführt.

Der eigentliche Editor wird mit »I« initialisiert. Die Aufforderung zum Einlegen der zu bearbeitenden Diskette bestätigt

| Befehl | Mnemonic | Bedeutung |
|--------|----------|---|
| A/SPC | Anl | Anleitung zeigen |
| B | Block | blockweises Weiterlesen der Directory |
| D | DEL | File als gelöscht kennzeichnen |
| E | Ende | Bearbeitung der Directory beenden |
| F | Floppy | Floppykommando eingeben |
| I | Init | Editor initialisieren, Diskettenamen und -ID ändern |
| L | Letter | Groß-/Kleinschrift umschalten |
| N | Name | Filenamen neu eingeben |
| P | PRG | als Programmfile kennzeichnen |
| R | REL | *) als relatives File kennzeichnen |
| S | SEQ | als sequentielles File kennzeichnen |
| U | USR | als User-File kennzeichnen |
| V | Valid | Block-Überprüfung erzwingen |
| X | Exit | Programm beenden |
| -- | | nächsten Fileeintrag anzeigen |
| <RET> | | einen Fileeintrag zurückgehen |
| * | *) | geöffnetes File schließen |
| + | *) | Löschschrift aktivieren |
| - | *) | Löschschrift abschalten |
| 0/1 | | auf entsprechendes Laufwerk schalten |

*) nicht für 2040/3040-Laufwerke

Tabelle 1. Die Befehle des Directory-Editors

man mit einem beliebigen Tastendruck. Nun wird die eingelegte Diskette initialisiert und nachgeprüft, ob sie dem Format des Floppy-Laufwerks entspricht. Jetzt werden nacheinander Diskettenname und Identität (ID) angezeigt, die man durch einfaches Überschreiben ändern kann. Die ID darf übrigens bis zu fünf (bei CBM 2040/3040 vier) Zeichen lang sein. Mit der RETURN-Taste kommt man zum Inhaltsverzeichnis.

Das Programm zeigt jetzt den ersten Eintrag an. Im ersten Feld wird die Länge des Files in Blöcken angezeigt. Das zweite Feld enthält den Filenamen, der nach dem Drücken von »N« frei geändert werden kann.

Im dritten Feld werden drei Fileparameter angezeigt, die in dem Directory auf der Diskette alle in einem Byte untergebracht sind: Wenn ein Stern vor dem Filetyp angezeigt wird, so wurde das File nach dem Schreiben nicht geschlossen, was es normalerweise unmöglich macht, auf dieses File wieder zuzugreifen. Glücklicherweise geschieht dies recht selten, da so etwas nur durch Stromausfall, versehentliches Ziehen des Netzsteckers oder durch Unterbrechen eines Basic-Programms mitten im Schreibvorgang zustandekommen kann. Trotzdem ist es ärgerlich, wenn es einmal passiert. Mit dem Directory-Editor können solche Files wieder lesefähig gemacht werden, indem man »*« eingibt. Dabei wird die erste Spur/Sektor-Kennzeichnung ausgelesen und das File bis zum Ende verfolgt. Danach wird die errechnete Blockzahl in das Directory geschrieben und ein Validate der Diskette nach dem Programmablauf veranlaßt (ab Zeile 6000). Der Stern vor dem Filetyp verschwindet, das File ist wieder in Ordnung. Es kann wohl vorkommen, daß die letzten (maximal 254) Bytes verlorengehen, wenn diese bei dem Zwischenfall noch im Floppy-puffer zwischengespeichert waren; dafür ist der Rest aber gerettet.

Als zweite Information in diesem dritten Feld wird der Filetyp angegeben. Dieser kann durch Drücken des Anfangsbuchstaben des gewünschten Filetyps geändert beziehungsweise das File mit »D« gelöscht werden. Auch gelöschte Files werden mit angegeben und können (leider nicht bei den Floppy-Laufwerken CBM 2040/3040 wegen der völligen Zerstörung der Block-Zeiger) wieder eingetragen werden. Bei der Benutzung ist aber Vorsicht geboten, denn wenn nach dem Löschen schon wieder auf die Diskette geschrieben wurde, können Teile des Files oder Programms zerstört sein. Werden nach einem Validate zu viele freie Blöcke ausgegeben, dann kann man nichts mehr retten und das File sollte wieder gelöscht und die Diskette mit dem Validate-Befehl nochmals überprüft werden.

Die dritte Information, die nur ab DOS-Version 2 vorhanden ist (also nicht für 2040/3040), ist der Löschschrift. Er wird durch »+« ein- und durch »-« wieder ausgeschaltet und schützt vor dem Scratch-Befehl. Der New-Befehl beachtet solch einen softwaremäßigen Schutz nicht, hier hilft nur der physikalische Löschschrift, das Ankleben einer Schreibschutzmarke. Der softwaremäßige Löschschrift wird mit »<« hinter dem Filetyp in der Directory angezeigt.

Die letzten beiden Felder der Zeile zeigen den Startsektor, also den Anfang des Files, auf der Diskette. Bei 2040/3040-Floppy-Laufwerken sieht man, daß mit »Scratch« gelöschte Files eine Null in der Anzeige der Spur anzeigen und somit nicht mehr reparabel sind.

Mit der RETURN-Taste, »-« oder »=« kann man den nächsten Eintrag in die Anzeige holen. Sucht man nur ein bestimmtes File, um etwas zu ändern, so kann man mit »B« gleich bis zum Ende des Directory-Blocks »vorspulen«. Dies hat den Vorteil, daß es schneller geht. Trotzdem kann man immer mit »f« oder »@« ein File zurückgehen, falls es in demselben Block liegt.

Wenn ein File wieder in die Directory zurückgeholt oder gelöscht wird, so muß die Diskette später zur korrekten Blockbelegung überprüft werden. Will man diesen Vorgang erzwingen,

so kann man »V« drücken. Ob überprüft wird oder nicht, steht unten in der vorletzten Textzeile hinter »V«. Ein »+« bedeutet, daß erst später überprüft wird.

Mit »E« kann die Bearbeitung jederzeit unterbrochen und zum Ende gesprungen werden. Am Ende des Directory, beziehungsweise wenn »E« eingegeben wurde, wird — falls erforderlich durch Löschen, Wiedereintragen oder Schließen eines Files — die Diskette mit dem »Validate«-Befehl überprüft. Dies kann eine Minute dauern, falls sich viele Files auf der bearbeiteten Diskette in ungelöschtem Zustand befinden. Danach wird die Anzahl der benutzten und der freien Blöcke ausgegeben. In Klammern erscheint die Anzahl der freien Blöcke, die in dem Directory abgelegt sind. Kommt es zu Unterschieden, so sollte, wenn das Directory bis zum Ende durchgeblättert wurde, mit »Validate« überprüft werden. Wenn mit »E« oder »X« abgebrochen wurde, dann sind wahrscheinlich nicht alle belegten Blöcke mitgezählt worden und die Zahlen, die nicht in Klammern stehen, sind nicht relevant. Nach Anzeige der Blockzahlen ist das Programm bereit, eine neue Diskette zu bearbeiten.

Eingebaute Geräteanpassung

Am Anfang des Programms wird in eine Unterroutine ab Zeile 7 500 gesprungen, die durch Vergleichen eines Bytes mit einer Tabelle herausfindet, in welchem Computer sich das Programm im Augenblick befindet, und danach alle systemspezifischen Adressen, die vom Hauptprogramm benötigt werden, in Variablen einliest. Dieses Vorgehen ermöglicht es, jeden Commodore-Computer, der mindestens einen 40 Zeichen breiten Bildschirm besitzt, zu benutzen, ohne das Programm vollständig verändern zu müssen. Außerdem können auch die

| Variable Bedeutung | Computer-Name | | | | |
|---|---------------|-------------|-------------|-------------|------|
| | CBM 2001 | CBM 3001 | CBM 4001 | CBM 8001 | C 64 |
| C%(0) PEEK(58453) | 136 | 152 | 32 | 76 | 189 |
| WI Zeilenlänge des Bildschirms | 40 | 40 | 40 | 80 | 40 |
| C0 Länge des Tastaturpuffers | 525 | 158 | 158 | 158 | 198 |
| C1 =0: Cursor eingeschaltet | 532 | 167 | 167 | 167 | 204 |
| C2 Zähler für Cursorblinken | 533 | 168 | 168 | 168 | 205 |
| C3 Zeiger auf Start der BS-Zeile | 224 | 196 | 196 | 196 | 209 |
| C4 augenblickl. Cursorspalte | 226 | 198 | 198 | 198 | 211 |
| C5 =1: Hochkommamodus (*) | 234 | 205 | 205 | 205 | 212 |
| C6 Länge der Bildschirmzeile | 242 | 213 | 213 | 213 | 213 |
| C7 augenblickliche Cursorzeile | 245 | 216 | 216 | 216 | 214 |
| C8 Anzahl der Inserts | 251 | 220 | 220 | 220 | 216 |
| C9 =128: Repeat auf allen Tasten bei C9=0: kein Repeat-Flag | --- | --- | --- | 228 | 650 |

Tabelle 2. Adressenerklärung zur Computeranpassung

großen Commodore CBM 600/700 oder die kleinen C 16/C 116 und der Plus/4 mit dem gleichen Programm laufen, wenn die Abfragen entsprechend erweitert werden. Zur eigenen Anpassung des Programms sind in Tabelle 2 die benötigten Systemadressen aufgeführt. Sie müssen in dieser Reihenfolge nach dem PEEK-Wert und dem Namen des Computers ab Zeile 7 560 in DATAs abgelegt werden.

Für alle Commodore-Computer gilt, daß mindestens 16 KByte RAM vorhanden sein müssen. Hiervon sollten 12 KByte zur freien Verfügung übrigbleiben, denn der Directory-Editor belegt 10 KByte und benötigt noch einmal 2 KByte für Variablen.

Auch die Floppy-Laufwerke besitzen unterschiedliche Eigenschaften und Eigenarten, die es nötig machen, den Geräte-Typ durch das Programm erkennen zu lassen, um ein zufriedenstellendes Arbeiten des Directory-Editors mit allen Laufwerken zu gewährleisten. Diese Routine (ab Zeile 7 000)

wurde aus dem Programm »Disk Addr Change« der 1541-Test/Demo-Diskette entnommen und für den Directory-Editor erweitert. Auch dieses Unterprogramm kann leicht an neue Floppy-Laufwerke angepaßt werden. Es werden mit »M-R« einzelne Bytes aus dem Floppy-ROM gelesen, die charakteristisch für bestimmte Floppy-Laufwerke sind. Anpassungen an alle gängigen Commodore-Floppy-Laufwerke sind eingebaut (2/4031, 1540/1, 2/3040, 4040, 8050, 8250, wobei CBM 4040/8050 entweder mit DOS 2.1/2.5 oder DOS 2.7 ausgestattet sind). Die Erweiterung auf neue Commodore-Laufwerke (zum Beispiel 1571 und SFD 1001/1002) ist natürlich möglich. In Tabelle 3 werden alle erforderlichen Variablen erklärt und am Beispiel der eingebauten Anpassung demonstriert. Tabelle 4 zeigt die Bedeutung wichtiger Variablen, die bei verschiedenen Laufwerks-Typen unterschiedlich sein kann.

Die Unterroutinen

Der Programmablauf ist in Tabelle 5 erläutert. Einige interessante, auch für andere Programme verwendbare Unterroutinen sind enthalten. Das Unterprogramm ab Zeile 8 000 stellt eine neue Eingabe-Routine dar, die die Möglichkeit bietet, die Cursorsteuerung sowie Insert/Delete zu erlauben, und trotzdem aufzupassen, daß nicht über den Rand des Eingabefeldes hinausgefahren wird. Außerdem sind zwei Tasten etwas anders belegt: Die HOME-Taste setzt den Cursor auf den Anfang des Eingabe-Feldes und CLR löscht nur das Eingabe-Feld, so daß der Bildschirmaufbau nicht aus Versehen einmal zerstört werden kann. Zeile 8 021 wurde nur speziell für den Directory-Editor eingebaut: sie füllt den Rest des Strings bis zur Maximallänge IL mit SHIFT-Leerzeichen auf. Das wird zum

| Variable | Bedeutung | Floppy-Name und DOS-Version | | | | |
|----------|---------------------------|-----------------------------|---------|--------|---------|------|
| | | 2/3040 | 4040 | 2/4031 | 8050 | 8250 |
| F\$ | | | 2.1/2.7 | 1540/1 | 2.5/2.7 | /LP |
| F%(2) | Floppy-PEEK(65535) *) | 226 | 213/198 | 254 | 242/198 | 198 |
| F%(1) | Floppy-PEEK(16618) | -99 | -99/0 | -99 | -99/1 | 1 |
| F%(2) | Floppy-PEEK(16556) | -99 | -99/-99 | -99 | -99/1 | 2 |
| DR | Anzahl der Laufwerke -1 | 1 | | 0 | 1 | |
| BF | Puffer #0 (High-Byte) | 17 | | 3 | 17 | |
| TY | Floppy-Typ (siehe Tab. 4) | 1 | 2 | | 3 | 4 |

*) -99: Nicht zur Unterscheidung benötigt, daher auf »0« gesetzt

Tabelle 3. Variablenerklärung zur Floppy-Anpassung

| Variable | Bedeutung | TY= 1 | 2 | 3 | 4 |
|----------|--|----------|-------|------|-------|
| FB | Anzahl der freien Blöcke | 670 | 664 | 2052 | 4133 |
| DM\$ | Formatkennzeichen auf Disk | CHR\$(1) | "a" | "c" | "c" |
| TT,TS | Directory-Header-Block | | 18,00 | | 39,00 |
| TD,SD | erster Directory-Block | | 18,01 | | 39,01 |
| NB | Beginn des Namen (Byte #) | | 144 | | 6 |
| AT | Anzahl der Tracks +1 | | 36 | | 78 |
| DT | Anzahl der Dateitypen | 4 | | 5 | |
| FT\$ | Dateitypen (im Augenblick: DEL, SEQ, PRG, USR, REL) | | | | |
| Typ 1: | 2040/3040 V1.0 | | | | |
| Typ 2: | 4040V2.1/2.7, 1540/1 V2.6, 4031/2031 V2.6, (SFS 491) | | | | |
| Typ 3: | 8050 V2.5/2.7 | | | | |
| Typ 4: | 8250/LP V2.7, (SFD 1001/1002) | | | | |

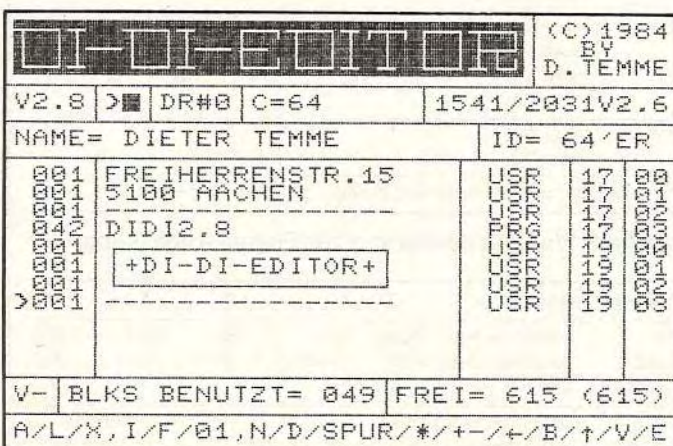
Tabelle 4. Variablenerklärung für verschiedene Floppy-Laufwerke

| | |
|-----------|---|
| 1000-1999 | Initialisierung der Variablen, Aufruf von 7000 und 7500 |
| 2000-2499 | Eingabe-Warteschleife und Verarbeitung der Befehle |
| 2500-2999 | Unterprogramm: Schließen eines Files |
| 2800-2999 | Unterprogramm: Umschaltung Groß-/Kleinschreibung |
| 3000-3499 | Unterprogramm: Ausgabe der Anleitung |
| 3500-3999 | Unterprogramm: Ein- und Weitergabe eines Floppy-Kommandos |
| 4000-4999 | Unterprogramm: Initialisierung des Editors |
| 5000-5999 | Unterprogramm: Anzeige des nächsten Eintrags |
| 6000-6999 | Unterprogramm: Ende des Directorys, Anzeige der Blockzahl |
| 7000-7099 | Unterprogramm: Erkennung des Floppy-Typs |
| 7100-7199 | Vorbereitung von 2040/3040 |
| 7200-7299 | Data: Floppy-Laufwerk-Erkennung (Tabelle 3) |
| 7300-7399 | Data: Merkmale gleicher Typen (Tabelle 4) |
| 7400-7499 | Data: erweitertes M-R für 2/3040 |
| 7500-7599 | Unterprogramm: Erkennung des Computer-Typs |
| 7600-7699 | Data: computerspezifische Variablen (Tabelle 2) |
| 8000-8999 | Unterprogramme: 8000 Eingabe-Routine 8030 Inget-Routine 8040 Bildschirmroll-Routine 8050 Diskettenfehler behandeln 8060 Taste holen 8070 Ausgabe mit Steuerzeichen 8080 Cursor auf Zeile LN stellen |
| 9000-9099 | Data: Bildschirmaufbau |
| 9100-9199 | Data: Änderungen des Aufbaus für die Anleitung |
| 9500-9999 | Data: Anleitung |

Tabelle 5. Programmaufbau des Directory-Editors

Löschen eines möglichen Reststrings in der Directory benötigt und kann bei eigener Anwendung der Eingabe-Routine gegebenenfalls weggelassen werden. Die Eingabe von SHIFT-RETURN bewirkt, daß der Rest ab der Cursor-Position gelöscht und die Eingabe wie bei RETURN beendet wird.

Die Routine ab Zeile 8040 scrollt einen beliebigen Bildschirmbereich von Zeile X1 bis X2 aufwärts, falls der Cursor in Zeile X2 steht. Dies geschieht durch »INPUT« vom Bildschirm und Ausgabe der eingegebenen Strings je eine Zeile höher.



Das Hauptmenü des Disk-Directory-Editors

Ab Zeile 8060 steht die Warteschleife bis zum Drücken einer Taste. Dabei wird der Cursor eingeschaltet und an eine vordefinierte Stelle auf den Bildschirm gesetzt. In PR\$ ist das Promptzeichen enthalten. Wenn das Unterprogramm von Zeile 2000 her angesprungen wird, sieht man ein »>«. Aus einem anderen Unterprogramm heraus (zum Beispiel Anleitung) zeigt sich ein »<«. Daran erkennt man, in welchem Modus man sich gerade befindet. Die gedrückte Taste wird direkt daneben angezeigt.

Ab Zeile 8070 befindet sich eine Routine, die einen Text inklusive aller Steuerzeichen ausdrückt. Auch ein sonst »unverdauliches« RETURN-Zeichen (CHR\$(13)) wird mitgedruckt. Da Sie mit einem fehlerhaft arbeitendem Directory-Editor einigen Schaden anrichten können, sollten Sie erst nach dem sorgfältigen Austesten des Programms mit Disketten, von denen ein Doppel existiert, die Arbeit mit dem Directory-Editor beginnen.
(Dieter Temme/ev)

```

1000 REM" *** DI-DI-EDITOR V2.8 *** " <251>
1010 REM DIETER TEMME, -03.01.1985- <198>
1020 REM FREIHERRENSTR. 15 <001>
1030 REM 5100 AACHEN <176>
1040 REM TEL. 0241/13415 <162>
1050 REM ----- <127>
1060 FD=B:REM >> FLOPPY DEVICE << <091>
1070 FT$="DEL SEQ PRG USR REL":REM FILETYP
EN <020>
1080 LB=256:B2=128:SC=64:BT=7:C$=CHR$(13):
DE$=CHR$(20):ZE$=CHR$(0):TW$=CHR$(2) <159>
1090 BR$="UA 2 ":BW$="UB 2 ":TE$="U0":MR$="
M-R":MW$="M-W" <177>
1100 CH$="{HOME}":CC$="{CLR}":CR$="{RIGHT}
":CL$="{LEFT}":CD$="{DOWN}":CU$="{UP}
":RV$="{RVSON}":RO$="{ROFF}" <008>
1110 BE$=CHR$(7):PP=59468:LE=0:PRINT CH$CH
$CC$CHR$(142)CHR$(8)"BITTE WARTEN!" <240>
1120 D$=CH$:FOR I=1 TO 25:D$=D$+CD$:S$=S$+
" (2SPACE)":Z$=Z$+CL$+CL$:FR=FRE(0):NE
XT <167>
1130 Z$=BE$+LEFT$(S$,36)+LEFT$(Z$,36):U$=C
HR$(160):T$=U$ <045>
1140 OPEN 1,FD,15,TE$ <249>
1150 GOSUB 7000:BF$=CHR$(BF):DR$=CHR$(DR+4
B) <134>
1160 GOSUB 7500:T$=T$+LEFT$(S$, (WI-40)/2):
T=LEN(T$)+1 <220>
1170 READ R$:IF R$<>">S"GOTO 1170 <201>
1180 DIM M$(18),B$(14):FOR I=1 TO 18:READ
M$(I):NEXT:PRINT CC$: <126>
1190 FOR I=1 TO 9:PRINT T$M$(I):NEXT <160>
1200 FOR I=1 TO 11:PRINT T$M$(10):NEXT <253>
1210 FOR I=11 TO 15:PRINT T$M$(I):NEXT <213>
1220 IF ER THEN PRINT LEFT$(D$,24)TAB(T)RV
$Z$ER$CH$CD$CD$CD$:IF ER<3 THEN END <101>
1230 IF ER THEN CLOSE 1:OPEN 1,FD,15:REM R
E-INIT <016>
1240 PRINT LEFT$(D$,6)TAB(T+11)DR$TAB(T+13
)CT$TAB(T+23)F$ <152>
1250 BZ$=T$+M$(14):IF C9 THEN POKE C9,128 <133>
2000 REM $$$ EINGABE $$$ <222>
2010 PR$=">":GOSUB 8060:G$=K$:PR$="<" <006>
2020 IF BZ THEN LN=PEEK(C7):PRINT LEFT$(D$
,24)BZ$:GOSUB 8080:BZ=0 <233>
2030 IF G$="A"OR G$=" " THEN GOSUB 3000 <230>
2040 IF G$="L" THEN GOSUB 2800 <129>
2050 IF G$="X" THEN G$="E":EX=1 <019>
2060 IF OP=0 GOTO 2220 <206>
2070 IF G$="B" THEN GB=1:G$=C$:IF GZ=1 THEN
GZ=0:GB=0:G$="E":EN=1 <051>
2080 IF G$="N"OR G$=CR$ THEN PRINT TAB(T+5)
;:IL=16:GOSUB 8000:N$=IX$:PRINT CU$: <036>
2090 IF G$="*"OR G$=":" THEN IF FT THEN IF
PT$>" THEN IF C AND FT<>4 THEN GOSUB
2500:PR=1 <175>
2100 IF G$="+"OR G$=":"OR G$="-" THEN IF TY
>1 THEN S$="(G$<>")*SC:PR=Z <162>
2110 IF G$="↑"OR G$="@" THEN IF ZZ>0 THEN P
RINT TAB(T)" "CU$:NG=NG+2:ZR=1:ZZ=ZZ-
2:G$="←" <002>
2120 IF G$="←"OR G$=C$OR G$="=" THEN IF (EN
OR ER)=0 THEN GOSUB 5000:IF GB THEN I
F ZZ<7 GOTO 2120 <073>
2130 IF EN=1 AND GB=1 THEN GZ=1:PT$=" ":PR
INT CU$TAB(T)">":CU$:IF ZZ THEN ZZ=ZZ-
1:EN=0 <207>
2140 IF GZ=1 THEN GB=0:IF ER=0 GOTO 2000 <112>
2150 IF EN=1 GOTO 2250 <003>
2160 IF G$="V" THEN LN=PEEK(C7):VA=1:VO=1:P
RINT LEFT$(D$,22)TAB(T+1)"+" :GOSUB 80
80 <249>

```

Listing »Disk-Directory-Editor«. Beachten Sie bitte die Eingabehinweise auf Seite 6.

```

2170 IF G$="E" THEN EN=1:GOSUB 5000:GOTO 22
50 <053>
2180 IF PT$=""GOTO 2210 <113>
2190 FOR I=0 TO DT-1:IF G$=MID$(FT$,I*4+1,
1) THEN FT=I:I=DT:C=OC:PR=1:IF FT=0 TH
EN C=B2 <168>
2200 NEXT <178>
2210 : <154>
2220 IF OP=1 GOTO 2260 <190>
2230 IF G$="I" THEN GOSUB 4000:ON-(ER=0)GOS
UB 5240:IF ER GOTO 1220 <074>
2240 IF G$="F" THEN GOSUB 3500 <032>
2250 IF G$="0"OR G$="1" THEN IF VAL(G$)<=DR
THEN DR$=G$:PRINT LEFT$(D$,6)TAB(T+1
1)G$ <253>
2260 IF EN THEN GOSUB 6000 <226>
2270 IF PR=1 THEN GOSUB 5190:PR=0 <085>
2280 IF EX=1 THEN ER=2:ER$=">>> PROGRAMMEN
DE <086>
2290 IF ER GOTO 1220 <167>
2300 GB=0:GOTO 2000 <145>
2500 REM $$$ FILE SCHLIESSEN $$$ <189>
2510 PRINT#1,MR$CHR$(K+1)BF$TW$:GET#1,H$,I
$:IF H$=""OR FT=4 THEN RETURN <131>
2520 PRINT TAB(T)"*CU$:OPEN 3,FD,3,"#1":B
G$=CHR$(BF+1):BK=0 <030>
2530 RT=ASC(H$):RS=ASC(I$+ZE$):PRINT#1,"UA
"3;DR$RT;RS:BK=BK+1 <189>
2540 PRINT#1,MR$CHR$(3)BG$TW$:GET#1,H$,I$:
IF H$<CHR$(AT)AND H$>""GOTO 2530 <243>
2550 PRINT#1,MW$CHR$(K+28)BG$TW$CHR$(BK-3
2768)AND(LB-1))CHR$(BK/LB) <105>
2560 IF H$>"" THEN PRINT#1,MW$ZE$BG$TW$ZE$Z
E$:PRINT#1,"UB"3;DR$RT;RS <128>
2570 VA=1:C=0:OC=C:CLOSE 3:FR=FRE(0):RETUR
N <167>
2800 REM $$$ LETTER ON/OFF $$$ <117>
2810 IF RIGHT$(CT$,3)="001" THEN POKE PP,(P
EEK(PP)AND 253)OR(2-LE) <211>
2820 IF RIGHT$(CT$,3)<>"001" THEN PRINT CHR
$(14-128*(LE>0)); <142>
2830 LE=2-LE:RETURN <243>
3000 REM $$$ ANLEITUNG $$$ <116>
3010 LN=PEEK(C7):PRINT LEFT$(D$,7):OPEN 3,
3:FOR I=0 TO 14:INPUT#3,B$(I):PRINT C
L$ <203>
3020 NEXT:PRINT LEFT$(D$,7)T$M$(16):FOR I=
0 TO 14:PRINT T$M$(17):NEXT:PRINT T$M
$(18) <246>
3030 IF AN=1 THEN RESTORE:PRINT LEFT$(D$,8
)TAB(T)RV$Z$TAB(T+12)"BITTE WARTEN <161>
3040 READ R$:IF R$<>"A"GOTO 3040 <215>
3050 PRINT LEFT$(D$,7):FOR I=0 TO 4:READ B
$:PRINT TAB(T)Z$B$:NEXT <217>
3060 PRINT LEFT$(D$,12):FOR I=0 TO 9:READ
B$:PRINT TAB(T)Z$B$:NEXT:GOSUB 8060 <093>
3070 READ B$:IF B$<>"*GOTO 3060 <111>
3080 PRINT LEFT$(D$,7)T$M$(7):FOR I=0 TO 1
4:PRINT B$(I):NEXT <088>
3090 PRINT T$M$(13):GOSUB 8080 <125>
3100 AN=1:POKE C0,0:FOR I=0 TO 14:B$(I)=""
:NEXT:FR=FRE(0):CLOSE 3:RETURN <243>
3500 REM $$$ DISK-KOMMANDO $$$ <033>
3510 PRINT LEFT$(D$,24)T$M$(17)CU$C$TAB(T)
"@":IL=35:IJ=1:GOSUB 8000:IJ=0:BZ=1 <231>
3520 IF LEFT$(IX$,1)<>"U" THEN PRINT#1,IX$:
GOTO 8050 <220>
3530 RETURN <030>
4000 REM $$$ START $$$ <254>
4010 BR$=LEFT$(BR$,5)+DR$+"":BW$=LEFT$(BW
$,5)+DR$+"", <067>
4020 PRINT LEFT$(D$,8)T$M$(8)CD$:FOR I=1 T
O 11:PRINT T$M$(10):NEXT:PRINT CD$T$M
$(12) <107>
4030 PRINT LEFT$(D$,24)TAB(T)RV$Z$" {6SPACE
}BITTE DISKETTE ENLEGEN":GOSUB 8060 <186>
4040 PRINT#1,"I"+DR$:VA=0:X1=10:X2=20:GOSU
B 8050:IF ER THEN RETURN <084>
4050 PRINT CU$BZ$:OPEN 2,FD,2,"#0":PRINT#1
,BR$TT;TS:GOSUB 8050:IF ER THEN RETUR
N <203>
4060 PRINT#1,MR$TW$BF$CHR$(1):GET#1,DO$:IF
DO$<>DM$GOTO 4200 <239>
4070 PRINT#1,MR$CHR$(NB)BF$CHR$(16):GOSUB
8030:PRINT LEFT$(D$,8)TAB(T+6); <149>
4080 GOSUB 8070:PRINT C$CU$TAB(T+6);:IL=16
:GOSUB 8000:N$=IX$ <215>
4090 PRINT#1,MR$CHR$(NB+18)BF$CHR$(5+(TY=1
)):GOSUB 8030:PRINT CU$TAB(T+30); <052>
4100 GOSUB 8070:PRINT C$CU$TAB(T+30);:IL=5
+(TY=1):GOSUB 8000:PRINT <134>
4110 PRINT#1,MW$CHR$(NB)BF$CHR$(23+(TY=1))
N$"--"IX$:PRINT#1,BW$TT;TS:ZZ=0:OP=1 <046>
4120 ZB=0:VA=0:VO=0:LN=PEEK(C7):PRINT LEFT
$(D$,22)TAB(T+1)"-":ZT=TD:ZS=SD <160>
4130 IL=16:GOTO 8080 <065>
4200 ER=3:ER$=">>> FALSCHES DISK-FORMAT":R
ETURN <089>
5000 REM $$$ NAECHSTER EINTRAG $$$ <178>
5010 IF EN=1 GOTO 5080 <080>
5020 PRINT TAB(T)" ":PRINT#1,MW$CHR$(K+3)B
F$CHR$(16)N$:IF ZR THEN PRINT CU$CU$;
:ZR=0 <155>
5030 PRINT#1,MW$CHR$(K)BF$CHR$(1)CHR$(S OR
(B2-C)OR FT) <000>
5040 IF (OF=0 AND FT>0)OR(OF>0 AND FT=0)THE
N VA=1:LN=PEEK(C7) <080>
5050 IF VO=0 AND VA=1 THEN LN=PEEK(C7):PRI
NT LEFT$(D$,22)TAB(T+1)"+":GOSUB 8080
:VO=1 <161>
5060 IF ZZ<7 THEN PRINT#1,MR$CHR$(ZZ*32+39
)BF$CHR$(1):GET#1,NF$ <168>
5070 IF (NT>0 OR(I<7 AND NF$>"")) THEN GOSUB
8040 <164>
5080 IF EN=1 THEN ZZ=7 <064>
5090 ZZ=ZZ+1:IF ZZ>7 GOTO 5220 <151>
5100 K=ZZ*32+2:PRINT#1,MR$CHR$(K+3)BF$CHR$(
16):GOSUB 8030:IF H$="" THEN EN=1:RET
URN <123>
5110 N$=I$:PRINT#1,MR$CHR$(K)BF$CHR$(1):GE
T#1,I$:FT=ASC(I$+ZE$) <069>
5120 S=SC AND FT:C=B2-(B2 AND FT):FT=(BT)A
ND FT:OF=FT:OC=C:IF FT=0 THEN OC=0 <191>
5130 PRINT TAB(T+5);:IF NG=0 THEN I$=N$:GO
SUB 8070 <091>
5140 PRINT#1,MR$CHR$(K+28)BF$TW$:GET#1,K$,
L$:BK=ASC(K$+ZE$)+LB*ASC(L$+ZE$) <169>
5150 IF NG=0 THEN IF FT<>0 THEN ZB=ZB+BK <177>
5160 PRINT C$CU$TAB(T)">"RIGHT$(STR$(1000+
BK),3);:IF NG THEN NG=NG-1 <230>
5170 PRINT#1,MR$CHR$(K+1)BF$TW$:GET#1,PT$,
K$:PRINT TAB(T+31); <176>
5180 PRINT RIGHT$(STR$(100+ASC(PT$+ZE$)),2
)CR$RIGHT$(STR$(100+ASC(K$+ZE$)),2); <110>
5190 PRINT C$CU$TAB(T+25)" ";:IF C THEN IF
FT THEN PRINT CL$"*"; <216>
5200 PRINT MID$(FT$,FT*4+1,3)" ";:IF S>0 A
ND TY>1 THEN PRINT CL$"<"; <074>
5210 PRINT C$CU$;:FR=FRE(0):RETURN <168>
5220 PRINT#1,BW$ZT;ZS:IF EN=0 THEN ZS=ASC(
BL$+ZE$):ZT=NT:IF ZT=0 THEN EN=1 <194>
5230 IF EN=1 THEN RETURN <213>
5240 PRINT#1,BR$ZT;ZS:PRINT#1,MR$ZE$BF$TW$
:GET#1,TR$,BL$:NT=ASC(TR$+ZE$):ZZ=0 <056>
5250 GOTO 5100 <158>
6000 REM $$$ ENDE $$$ <252>
6010 PRINT LEFT$(D$,24);:IF VA=0 GOTO 6040 <091>
6020 PRINT TAB(T)Z$" {5SPACE}UEBERPRUEFE BL
OCKBELEGUNG"CU$:PRINT#1,"V"DR$:GOSUB
8050 <214>
6030 IF ER THEN RETURN <208>
6040 PRINT BZ$C$CU$CU$CU$TAB(T+17)RIGHT$(S
TR$(1000+ZB),3); <184>
6050 PRINT TAB(T+27)RIGHT$(STR$(1000+FB-ZB
),3)CR$CR$; <199>
6060 CLOSE 2:OPEN 2,FD,0,"$"+DR$+"":+ZE$:G
ET#2,B$,B$,B$,B$,B$,B$ <135>
6070 GET#2,B$:IF B$>"GOTO 6070 <230>
6080 GET#2,B$,B$,K$,L$ <005>
6090 PRINT RIGHT$(STR$(1000+ASC(K$+ZE$)+LB
*ASC(L$+ZE$)),3) <238>
6100 CLOSE 2:OP=0:EN=0:FR=FRE(0):RETURN <242>
7000 REM $$$ FLOPPY-TYP $$$ <058>
7001 READ N:DIM F$(N):FOR I=0 TO N:READ X:
PRINT#1,MR$CHR$(X-32768)AND 255)CHR$(
X/256) <200>
7002 GET#1,I$:F$(I)=ASC(I$+ZE$):NEXT:ER=0:
TY=0:PRINT#1,TE$ <227>
7003 J=0:FOR I=0 TO N:READ X:IF X>-1 THEN
IF X<>F$(I) THEN J=1 <127>
7004 NEXT:READ F$:IF F$<>"***" THEN READ DR
,BF,TY:IF J GOTO 7003 <042>

```

Listing »Disk-Directory-Editor« (Fortsetzung)


```

9008 DATA" _NAME={20SPACE}_ID={7SPACE}_ <019>
9009 DATA" @*****RE***** <128>
9010 DATA" _{4SPACE}_ {19SPACE}_ {5SPACE}_ {25 <044>
PAGE}_ {2SPACE}_
9011 DATA" @*****R*****E*****F*****E <185>
*****
9012 DATA" _V_ _BLKS BENUTZT={4SPACE}_ _FREI={ <201>
SPACE}_ ({3SPACE}) _
9013 DATA" @*****E***** <117>
*****
9014 DATA" _A/L/X, I/F/01, N/D/SPUR*/+/-/+/B/ <219>
+/V/E_
9015 DATA" @***** <157>
*****
9100 REM +++ DATA: RAHMEN (ANL) +++ <027>
9101 DATA" @***** <030>
*****
9102 DATA" _ {36SPACE}_ <139>
9103 DATA" @***** <117>
*****
9499 DATA" >A":REM ERKENNUNG <038>
9500 REM +++ DATA: ANLEITUNG +++ <065>
9501 DATA" @***** <040>
*****
9502 DATA" _ {6SPACE}_ DISK-DIRECTORY-EDITOR {7 <049>
SPACE}_
9503 DATA" _ {4SPACE}_ K U R Z A N L E I T U N <082>
G {5SPACE}_
9504 DATA" @***** <172>
*****
9505 DATA" <138>
9506 DATA" <A/SPC> {3SPACE} AUSGABE DIESER AN <106>
LEITUNG
9507 DATA" <L> {7SPACE} GROSS- ODER KLEINSCHR <236>
IFT
9508 DATA" <I> {7SPACE} START DES EDITORS <004>
9509 DATA" <F> {7SPACE} EINGABE FLOPPY-KOMMAN <165>
DO
9510 DATA" <0/1> {5SPACE} LAUFWERKS-NUMMER FE <066>
STLEGEN
9511 DATA" {11SPACE} {FALLS DOPPEL-LAUFWERK} <044>
9512 DATA" <N/RIGHT> NEUEN NAMEN EINGEBEN <178>
9513 DATA" <S/P/U/R> FILETYP NEU FESTLEGEN <038>
9514 DATA" <D> {7SPACE} FILE LOESCHEN <004>
9515 DATA, + <150>
9516 DATA" <*> {7SPACE} OFFENES FILE SCHLIESS <221>
EN
9517 DATA" <+/-> {5SPACE} FILESCHUTZ SETZEN/A <083>
UFHEBEN
9518 DATA" <+/= /RET> EINTRAG WEITERGEHEN <251>
9519 DATA" <B> {7SPACE} BLOCK WEITERGEHEN <044>
9520 DATA" <+/@> {5SPACE} EINTRAG ZURUECKGEHE <119>
N
9521 DATA" <V> {7SPACE} SPAETER 'VALIDATE' <028>
9522 DATA" <E> {7SPACE} AUSSTIEG AUS EDITOR <024>
9523 DATA" <X> {7SPACE} BEENDEN DES PROGRAMMS <113>
9524 DATA, , * <244>
10000 OPEN 3, 8, 0, " $: $ $ <248>
10001 GET#3, A$: PRINT ASC (A$+CHR$(0)) A$: GOT <074>
O 10001
@ 64'er
Listing »Disk-Directory-Editor« (Schluß)

```

Programm- schutz durch Autostart und Paßwort

Ein Programm, das nach dem Laden sofort gestartet wird und nicht mehr unterbrochen werden kann, ist eine Art Programmschutz. Richtig raffiniert wird es aber, wenn noch zusätzlich ein Paßwort nötig ist. »Autostart C 64 Plus« macht es Ihnen einfach, Ihre Programme so zu bearbeiten.

Das Programm »Autostart C 64 Plus« ist eine besonders komfortable und nützliche Art der Programmsicherung für den C 64 und ein Diskettenlaufwerk. Das Autostartprogramm lädt das zu bearbeitende Programm, das bis zu 230 Blöcke lang sein darf, in den Speicher des C 64, verschlüsselt es, fügt noch einige Modifizierungen ein und speichert es wieder ab. Der so eingefügte Programmschutz kann nur noch mit großem Aufwand rückgängig gemacht werden. Sicherlich kann jeder Programmschutz auch wieder entfernt werden, aber um dies so schwer wie möglich zu machen, soll hier nichts weiteres darüber verraten werden. Die Dauer des gesamten Bearbeitungsvorgangs richtet sich ganz nach der Länge des zu bearbeitenden Programms. Die Eingabe aller Parameter ist dabei komfortabel durch Menüpunkte und Abfragen realisiert wor-

den. Ein so gespeichertes Programm startet sich nach dem Laden automatisch, wenn Sie das entsprechende Paßwort wissen.

Geben Sie zunächst das Programm Autostart C 64 Plus (siehe Listing) mit dem Checksummer ein. Nach dem Starten des Programms ist es notwendig, einige Parameter einzugeben, denn es wird zwischen Basic- und Maschinenprogrammen unterschieden. Bei Maschinenprogrammen benötigt Autostart C 64 Plus zusätzlich die Startadresse. Mit zwei weiteren Menüpunkten läßt sich jedes Programm gegen eine Unterbrechung durch RUN/STOP-RESTORE absichern. Zuletzt wird das jeweilige Paßwort eingegeben. Ein Paßwort ist eine Art Geheimzahl, das aber auch Buchstaben enthalten darf. Es darf bis zu 10 Zeichen lang sein. Falls kein Paßwort eingegeben wird, startet das bearbeitete Programm später ohne jegliche Abfrage. Bevor Autostart C 64 Plus nun mit der Modifizierung des Programms beginnt, wird nochmals nach der Richtigkeit aller Angaben gefragt.

Während dem Laden ist jedes so modifizierte Programm gegen einen Reset geschützt. Dieser Schutz wird jedoch nach dem Laden abgeschaltet, da sonst ein Speicherplatzverlust von 8 KByte eintreten würde. Das bearbeitete Programm wird wie ein Maschinenprogramm mit LOAD »Name«, 8, 1 eingeladen. Danach erscheinen auf dem Bildschirm die normalen Betriebssystem-Meldungen — soweit nichts Ungewöhnliches. Nun muß man aber, und das ist der Trick, unaufgefordert das zuvor festgelegte Paßwort über die Tastatur eingeben (ohne anschließendes RETURN). Falls das Paßwort richtig war, blinkt der Bildschirm kurz auf und der Ladevorgang wird normal fortgesetzt. War das Paßwort aber falsch, so läuft das Diskettenlaufwerk ohne jede Fehlermeldung bis ans Ende ihrer Tage weiter. Falls Sie sich vertippt haben, ist es am einfachsten, eine Taste zu drücken, die nicht im Paßwort vorhanden ist, und dann nochmals das Paßwort einzugeben. Dadurch, daß keinerlei Fehler- oder Rückmeldungen ausgegeben werden, ist es für jemanden, der das Paßwort nicht kennt, fast unmöglich, Ihre Programme zu verwenden.

Ein Tip noch zum Schluß: Schreiben Sie sich Ihre Paßwörter auf, oder wenden Sie immer das gleiche Schema an (etwa jeder zweite Buchstabe des Programmnamens), denn sonst ist auch für Sie selbst der Zugang versperrt. (Jan Kusch/aw)

```

1 REM ***** <132>
2 REM ***** <133>
3 REM ** AUTOSTART C64 PLUS ** <120>
4 REM ----- <047>
5 REM ** WRITTEN 1984 BY JAN KUSCH ** <123>
7 REM ***** <138>
8 REM ***** <139>
9 REM ***** <071>
100 POKE 643,PEEK(45):POKE 55,PEEK(45):POK
    E 644,PEEK(46)+3:POKE 56,PEEK(46)+3:CL
    R <120>
101 PRINT "{CTRL-I,CTRL-N,CTRL-H,CLR,RVSON,
    DOWN,2SPACE}*{2SPACE}*{2SPACE}*{2SPACE}
    }AUTOSTART C64 PLUS {2SPACE}*{2SPACE}*{
    2SPACE}*{2SPACE}" <115>
102 C$="{38SPACE}":O$=CHR$(0) <025>
103 PRINT"DATA CHECKING {12LEFT}";:
    GOSUB 142 <143>
104 RESTORE:OPEN 15,8,15:FOR X=0 TO 112:RE
    AD A:POKE 828+X,A:NEXT <122>
105 A$="ENTER SOURCEFILENAME":GOSUB 133:SF
    $=T$ <165>
106 A$="ENTER TARGETFILENAME":GOSUB 133:TF
    $=T$ <188>
107 PRINT {DOWN}:A$="BASICPROGRAMM ":GOSU
    B 138:BA$=T$ <035>
108 IF BA$="J"THEN PRINT {DOWN}C$:PRINT C
    $:SA=42926:NF$="N":GOTO 111 <254>
109 PRINT:A$="STARTADRESSE ":GOSUB 140:SA=
    T:IF SA>65535 THEN PRINT {3UP}:GOTO 1
    09 <188>
110 A$="NEWFLAG":GOSUB 138:NF$=T$ <116>
111 PRINT:A$="RESTOREPROTECT":GOSUB 138:RS
    $=T$ <004>
112 A$="STOPPROTECT {3SPACE}":GOSUB 138:SS$
    =T$ <204>
113 PRINT {CODEWORD {12SPACE}}:;CW$="":INPUT
    CW$ <092>
114 IF LEN(CW$)>10 THEN PRINT {2UP}:GOTO
    113 <107>
115 PRINT {DOWN}:A$="ENTERY CORRECT":GOSU
    B 138:IF T$="N"THEN PRINT {HOME,2DOWN}
    ":GOTO 105 <080>
116 PRINT:FOR X=1 TO 40:PRINT"-":NEXT:INP
    UT"INSERT SOURCE-DISK";T$ <014>
117 PRINT {UP}C$:PRINT {UP}LOADING:;:OPE
    N 1,8,2,SF$+",P,R":GET#1,L1$,L2$ <110>
118 GOSUB 135:SYS 831:GOSUB 135:CLOSE 1:IN
    PUT {RVOFF}INSERT TARGET-DISK";T$ <044>
119 PRINT {UP}C$:PRINT {UP}SAVING:;:OPE
    N 1,8,2,"@:"+TF$+",P,W":GOSUB 135 <093>
120 FOR X=0 TO 257:READ A:PRINT#1,CHR$(A);
    :NEXT <192>
121 S1=INT(SA/256):S2=SA-S1*256:PRINT#1,CH
    R$(S2);CHR$(S1); <241>
122 A0=ASC(L1$+O$)+ASC(L2$+O$)*256:A1=PEEK
    (251)+PEEK(252)*256 <092>
123 A2=1+PEEK(643)+PEEK(644)*256:A3=(A1-A2
    )+A0:P1=INT(A3/256):P2=A3-P1*256 <190>
124 PRINT#1,CHR$(P2);CHR$(P1); <168>
125 A=237:IF RS$="J"THEN A=225 <134>
126 PRINT#1,CHR$(A);:A=49:IF SS$="J"THEN A
    =52 <234>
127 PRINT#1,CHR$(A);:A=0:IF NF$="J"THEN A=
    1 <018>
128 PRINT#1,CHR$(A);:IF CW$=""THEN PRINT#1
    ,CHR$(0)"1234567890":GOTO 130 <017>
129 PRINT#1,CW$CHR$(0);:IF LEN(CW$)<10 THE
    N PRINT#1,LEFT$(C$,10-LEN(CW$)); <188>
130 SYS 828:GOSUB 135:CLOSE 1:PRINT:A$="{D
    OWN}RESTART PROGRAMM":GOSUB 138:CLOSE
    15 <051>
131 IF T$="J"THEN RUN <051>
132 SYS 64738 <192>
133 T$="":PRINT A$;:INPUT T$:IF T$=""OR LE
    N(T$)>16 THEN PRINT {2UP}:GOTO 133 <207>
134 RETURN <192>
135 INPUT#15,A,B$,C,D:PRINT {RVSON}"A;B$;C
    ;D; <252>
136 IF A<2 THEN RETURN <113>
137 FOR X=1 TO 15:CLOSE 1:NEXT:PRINT:PRINT
    "?DISK-ERROR";:FOR X=1 TO 5000:NEXT:RU
    N <245>
138 T$="":PRINT A$ (J/N);:INPUT T$:IF T$
    <>"J"AND T$<>"N"THEN PRINT {2UP}:GOTO
    138 <184>
139 RETURN <197>
140 T$="":PRINT A$;:INPUT T$:T=VAL(T$):IF
    T$=""OR T=0 THEN PRINT {2UP}:GOTO 140 <242>
141 RETURN <199>
142 P1=15862:S=0:FOR X=0 TO 112:READ A:S=S
    +A:NEXT:IF S<>P1 THEN 144 <133>
143 P2=32966:P=1:S=0:FOR X=0 TO 257:READ A
    :S=S+A:NEXT:IF S=P2 THEN RETURN <052>
144 PRINT"?DATAERROR"P+1,"PRUEFSUMME:"S
    <188>
145 DATA 76, 121, 3, 173, 131, 2, 24, 105,
    1 <031>
146 DATA 133, 251, 173, 132, 2, 105, 0, 13
    3, 252, 162, 1 <056>
147 DATA 32, 198, 255, 169, 253, 37, 144,
    133, 144, 32, 225 <170>
148 DATA 255, 32, 207, 255, 160, 48, 120,
    132, 1, 160, 0 <107>
149 DATA 145, 251, 160, 55, 132, 1, 88, 36
    , 144, 112, 57 <179>
150 DATA 230, 251, 208, 222, 230, 252, 208
    , 218, 162, 1, 32 <084>
151 DATA 201, 255, 160, 48, 120, 132, 1, 1
    60, 0, 177, 251 <057>
152 DATA 160, 55, 132, 1, 88, 73, 255, 32,
    210, 255, 32 <055>
153 DATA 225, 255, 165, 251, 208, 2, 198,
    252, 198, 251, 165 <000>
154 DATA 251, 205, 131, 2, 208, 219, 165,
    252, 205, 132, 2 <250>
155 DATA 208, 212, 76, 204, 255 <066>
156 REM <218>
157 REM <219>
158 DATA 168, 2, 238, 32, 208, 76, 168, 2,
    174, 170, 3 <034>
159 DATA 134, 174, 172, 171, 3, 132, 175,
    134, 45, 132, 46 <218>
160 DATA 32, 142, 3, 144, 251, 32, 239, 23
    7, 32, 66, 246 <138>
161 DATA 169, 0, 141, 6, 128, 32, 21, 253,
    32, 83, 228 <126>
162 DATA 173, 173, 3, 141, 20, 3, 173, 172
    , 3, 141, 40 <017>
163 DATA 3, 173, 174, 3, 240, 3, 32, 68, 1
    66, 169, 0 <026>
164 DATA 76, 152, 3, 3, 255, 165, 174, 201
    , 186, 240, 65 <010>
165 DATA 96, 83, 80, 83, 32, 83, 79, 70, 8
    4, 87, 65 <073>
166 DATA 82, 69, 139, 227, 131, 164, 124,
    165, 26, 167, 228 <011>
167 DATA 167, 134, 174, 1, 2, 3, 50, 76, 7
    2, 178, 0
168 DATA 49, 234, 102, 254, 168, 2, 74, 24
    3, 145, 242, 14 <073>
169 DATA 242, 80, 242, 51, 243, 87, 241, 2
    02, 241, 237, 2 <002>
170 DATA 62, 241, 47, 243, 102, 254, 165,
    244, 237, 245, 162 <190>
171 DATA 10, 189, 143, 3, 157, 0, 128, 202
    , 208, 247, 169 <061>
172 DATA 0, 133, 153, 170, 134, 198, 134,
    251, 189, 175, 3 <157>
173 DATA 240, 17, 32, 62, 241, 240, 251, 1
    66, 251, 221, 175 <020>
174 DATA 3, 208, 230, 232, 230, 251, 208,
    234, 170, 160, 245 <153>
175 DATA 238, 32, 208, 238, 33, 208, 232,
    208, 247, 200, 208 <196>
176 DATA 244, 76, 174, 2, 169, 253, 37, 14
    4, 133, 144, 32 <119>
177 DATA 19, 238, 73, 255, 160, 0, 145, 17
    4, 165, 174, 208 <212>
178 DATA 2, 198, 175, 198, 174, 36, 144, 8
    0, 2, 56, 36 <029>
179 DATA 24, 96, 168, 2, 168, 2, 195, 194,
    205, 56, 48 <230>
180 DATA 141, 0, 8, 32, 113, 168, 32, 51,
    165, 32, 167 <069>
181 DATA 3, 76, 84, 168, 76 <198>

```

© 64'er

Listing »Autostartgenerator C 64 Plus«. Verwenden Sie zur Eingabe bitte den Checksummer. Beachten Sie die Eingabehinweise auf Seite 6.

Zu zweit geht's besser

Dieses Programm für C 64, Floppy und Drucker beseitigt ein lästiges und papieraufwendiges Problem: Drucker, die keinen Randsteller haben, können zwei Directories nebeneinander ausgeben.

Nachdem eine Diskette eingelegt und eine Taste gedrückt wurde, lädt das Programm die Directory in den Speicher. Nun ist die zweite Directory an der Reihe. Sobald diese geladen ist, werden die Directories nebeneinander ausgedruckt. Hierbei wird nach und vor ihnen eine Leerzeile eingefügt, damit in einer Serie von Directories nicht alle aneinanderkleben. Nach dem Ausdrucken können weitere Disketteninhalte bearbeitet werden.

Vorsicht! Es werden keinerlei Druckereinstellungen vorgenommen. Also müssen alle Einstellungen vor dem Start des Programms vorgenommen werden (Zeilenabstand etc.). Die voreingestellte Sekundäradresse ist 0. Sie kann aber mit POKE 2144, neue Sekundäradresse auf einen beliebigen Wert gesetzt werden. Die Geräteadresse kann mit POKE 2142,5 zum Beispiel von 4 auf 5 gesetzt werden. (Übrigens: da das Programm am Basic-Anfang liegt und weil ihm ein SYS-Befehl vorgelagert ist, kann es mit LOAD »name«,8 (,1 bei Kassette) geladen und mit RUN gestartet werden.)

(Claus Brunzema/rg)

```

0  "CHECKSUM.SCHNELL" PRG
5  "CHECKSUMMER VC20" PRG
7  "MSE LADER" PRG
29 "MSE V1.0" PRG
7  "-----" DEL
0  "MPS802.LISTING2" PRG
2  "MPS802.LISTING1" PRG
6  "-----" DEL
0  "FORMAT-SYSTEM" PRG
5  "-----" DEL
0  "VIC" PRG
36 "-----" DEL
0  "6510 I" PRG
101 "-----" DEL
0  "STERNENHIMMEL" PRG
57 "-----" DEL
0  "ASS.-9/PROG.1" PRG
1  "ASS.-9/PROG.2" PRG
5  "ASS.-9/PROG.3" PRG
1  "ASS.-9/PROG.4" PRG
1  "VERSCHIEBEN 9" PRG
1  "-----" DEL
0  "DIRECTORY-SORTER" PRG
19 "-----" DEL
0  "TRICK.OBJ" PRG
3  "3D-MOVIE-MAKER" PRG
28 "XDEMO-1.GRA" SEQ
68 "DEMO-2.GRA" SEQ
14 "DEMO-3.GRA" SEQ
67 "DEMO-4.GRA" SEQ
91 "DEMO-5.GRA" SEQ
18 "----- VC 20 ----" PRG
1  "MINIGRAFIK" PRG
5  "MINIGRAFIK DEMO" PRG
2  "LONGSCREEN" PRG
9  "----- C 16 ----" PRG
1  "HELP & TRACE" PRG
8  "MODULATOR" PRG
4  "MODULATOR.TEST" PRG
13 "-----" PRG
43 BLOCKS FREE.
    
```

```

0  "MSE V1.0" PRG
7  "MSE LADER" PRG
29 "-----" DEL
0  "CHECKSUM.SCHNELL" PRG
5  "CHECKSUMMER VC20" PRG
7  "-----" DEL
0  "HI-EDDI/MPS 801" PRG
7  "HI-PRINT /801" PRG
2  "MPS802/DATA" PRG
11 "HI-PRINT /802" PRG
3  "-----" DEL
0  "UPD.49309" PRG
1  "PROST" PRG
25 "-----" DEL
0  "E-ROUTINE 64" PRG
2  "-----" DEL
0  "GCR-HEX" PRG
4  "HEX-GCR" PRG
5  "-----" DEL
0  "SAMURAI" PRG
41 "-----" DEL
0  "SCROLL-MACHINE" PRG
13 "SCROLL-EDITOR" PRG
43 "DEMO I" PRG
10 "PSEUDO I" PRG
45 "ZEICHENSATZ" PRG
9  "SPRITES" PRG
5  "DEMO II" PRG
4  "PSEUDO II" PRG
33 "SPIEL" PRG
13 "-----" DEL
0  "F-PLOTTER C16" PRG
7  "CROSSREFERENZ" PRG
20 "-----" DEL
0  "HEAPSORT" PRG
3  "-----" PRG
310 BLOCKS FREE.
    
```

Beispielausdruck

programm : directory 0801 09bd

```

0801 : 1c 08 0a 00 9e 32 30 38 51
0809 : 30 3a 8f 20 53 4f 46 54 b0
0811 : 49 45 2d 53 4f 46 54 57 da
0819 : 41 52 45 00 00 00 24 a9 b9
0821 : 00 a9 10 a2 00 8d 3d 03 b6
0829 : 8e 3c 03 20 fd 08 8e 3e 71
0831 : 03 8c 3f 03 98 20 fd 08 3d
0839 : a9 00 85 c6 a0 10 20 37 96
0841 : a5 ad 3e 03 ac 3f 03 20 be
0849 : 37 a5 a9 01 8d 43 03 8d f7
0851 : 41 03 20 e7 ff a9 00 20 a6
0859 : bd ff a9 01 a2 04 a0 00 6d
0861 : 20 ba ff 20 c0 ff b0 65 7c
0869 : a2 01 20 c9 ff 20 68 09 82
0871 : ad 3c 03 ae 3d 03 20 26 8c
0879 : 09 8d 40 03 ad 43 03 f0 9c
0881 : 06 ce 43 03 ce 42 03 8c 44
0889 : 3c 03 8e 3d 03 ac 42 03 37
0891 : 20 63 09 c8 c0 28 d0 f8 41
    
```

```

0899 : ad 41 03 d0 06 20 68 09 d7
08a1 : 4c b9 08 ad 3e 03 ae 3f b7
08a9 : 03 20 26 09 8d 41 03 8c 6f
08b1 : 3e 03 8e 3f 03 20 68 09 e1
08b9 : ad 40 03 d0 b3 ad 41 03 15
08c1 : f0 13 a2 29 20 63 09 ca df
08c9 : d0 fa 4c 99 08 48 20 f4 8a
08d1 : 08 68 4c 04 e1 20 68 09 74
08d9 : 20 f4 08 a9 a0 a0 09 20 1e
08e1 : 1e ab 20 6d 09 c9 4a f0 75
08e9 : 07 c9 4e f0 06 4c e3 08 e9
08f1 : 4c 20 08 20 cc ff a9 01 c9
08f9 : 20 c3 ff 60 48 8a 48 a9 54
0901 : 79 a0 09 20 1e ab 20 6d ab
0909 : 09 a9 01 a2 1f a0 08 20 d3
0911 : bd ff a2 08 a0 00 20 ba 77
0919 : ff 68 aa 68 a8 a9 00 20 1c
0921 : d5 ff b0 a9 60 85 fb 86 86
0929 : fc a0 00 b1 fb 85 fd c8 21
0931 : b1 fb 85 fe c8 b1 fb aa 80
0939 : c8 b1 fe 20 cd bd 8c 42 5e
0941 : 03 20 63 09 a0 04 b1 fb 37
    
```

```

0949 : f0 0a ee 42 03 20 d2 ff bf
0951 : c8 4c 47 09 a9 92 20 d2 88
0959 : ff a0 00 b1 fd a4 fd a6 29
0961 : fe 60 a9 20 4c d2 ff a9 ac
0969 : 0d 4c d2 ff a5 c6 f0 fc 9f
0971 : ad 77 02 a2 00 86 c6 60 bf
0979 : 42 49 54 54 45 20 44 49 c8
0981 : 45 20 4e 41 45 43 48 53 f8
0989 : 54 45 20 44 49 53 4b 45 f7
0991 : 54 54 45 20 45 49 4e 4c d5
0999 : 45 47 45 4e 0d 0d 00 4e 73
09a1 : 4f 43 48 20 45 49 4e 45 0a
09a9 : 20 44 49 53 4b 45 54 54 81
09b1 : 45 20 28 4a 2f 4e 29 20 a4
09b9 : 3f 0d 0d 00 a9 00 85 59 26
    
```

Listing »Directory-Drucker«. Bitte beachten Sie die Eingabe-hinweise auf Seite 8

Ergänzen Sie jetzt Ihre

GATEWAY-Sammlung!

Schaffen Sie sich ein interessantes Nachschlagewerk und gleichzeitig ein wertvolles Archiv!

Kennen Sie alle Ausgaben von 64'er? Suchen Sie einen ganz bestimmten Testbericht? Oder haben Sie einen Teil eines interessanten Kurses versäumt? Suchen Sie nach einer speziellen Anwendung?

Damit Sie jetzt fehlende Hefte mit »Ihrem« Artikel nachbestellen können, finden Sie auf diesen Seiten eine Zusammenstellung aller wesentlichen Artikel von Ausgabe 4/84 bis Ausgabe 3/85.

Und so kommen Sie schnell an die noch lieferbaren Ausgaben: Prüfen Sie, welche Ausgabe in Ihrer Sammlung noch fehlt, oder welches Thema Sie interessiert. Tragen Sie die Nummer dieser Ausgabe und das Erscheinungsjahr (z.B. 2/85) auf dem Bestellabschnitt der hier eingehafteten Bestell-Zahlkarte ein. Die ausgefüllte Zahlkarte einfach heraustrennen und Rechnungsbetrag beim nächsten Postamt einzahlen. Ihre Bestellung wird nach Zahlungseingang umgehend zur Auslieferung gebracht.

| Stichwort | Titel | Seite | Ausgabe |
|------------------------------|--|-------|---------|
| Aktuell | | | |
| Computer | Die neuen - 264 und 364 (von der CES in Las Vegas) | 9 | 4/84 |
| | Heiße Messen in der Wüste: CES (PC 128, PC 10, Commodore LCD) | 8 | 3/85 |
| DFU | Datex-P und ausländische Netzwerke | 59 | 10/84 |
| | Informationen zur Datenfernübertragung | 8 | 11/84 |
| | Interessant bis brisant - die elektronischen Briefkästen | 10 | 12/84 |
| | Internationaler Chaos Communication Congress | 15 | 3/85 |
| | Kreatives Chaos (Interview mit dem CCC) | 12 | 10/84 |
| | MC1 Mail: die schnelle Post | 8 | 2/85 |
| Floppy | News 145: Laufwerk | 14 | 3/85 |
| | SFD 1002 | 8 | 9/84 |
| Messen | Commodore Fachausstellung in Frankfurt | 10 | 11/84 |
| | Consumer Electronics Show in Chicago | 10 | 8/84 |
| | Die Neuigkeiten von der Hannover Messe '84 | 8 | 6/84 |
| Musik | Musikneigkeiten aus den USA - MIDI | 44 | 9/84 |
| Raubkopierer | Der Sumpf | 12 | 6/84 |
| Listings zum Abtippen | | | |
| Anwendung | | | |
| Abtippen | Checksummer (C 64 und VC 20) | 72 | 1/85 |
| | Checksummer (C 64 und VC 20) | 65 | 2/85 |
| | MSE - Abtippen sicher und leicht gemacht | 98 | 2/85 |
| | MSE - Abtippen sicher und leicht gemacht | 78 | 3/85 |
| | Neuer Checksummer 64 - blitzschnell und kürzer | 68 | 3/85 |
| DFU | Mailboxprogramm für den C 64 | 114 | 9/84 |
| | Terminalprogramm für den C 64 | 24 | 7/84 |
| Datei | Adreß- und Telefonregister | 64 | 5/84 |
| | Relativ Programm-Datei (VC 20) | 69 | 5/84 |
| EPROM | Datenbrenner: Wie programmiere ich EPROMs? | 162 | 9/84 |
| Familie | Familienplanung mit dem VC 20 (AdM) | 52 | 2/85 |
| Finanzen | Abgerechnet wird mit dem C 64 (AdM) | 68 | 9/84 |
| | Menügesteuerte Finanzmathematik (AdM) | 68 | 10/84 |
| Floppy | Der Softwarekatalog für Ihre Programme | 72 | 7/84 |
| | Drucker/Floppy ein- oder ausgeschaltet? | 77 | 8/84 |
| | Hyper-Load: Schnelles Laden von Diskette (LdM) | 67 | 10/84 |
| Kalender | Elektronisches Notizbuch (VC 20) | 50 | 4/84 |
| Lehrer | Lehrerkalender (AdM) | 64 | 6/84 |
| Lernprogramm | Russische Vokabeln (nur auf Disk) | 76 | 7/84 |
| Mathe | Bildschirmmasken schnell erstellt | 78 | 9/84 |
| Mathematik | Mathematical-Basic: Das Super-Basic für den VC 20 (LdM) | 50 | 12/84 |
| Monitor | Ohne gutes Werkzeug geht es nicht: SMON (Teil 1) | 60 | 11/84 |
| | Ohne gutes Werkzeug geht es nicht: SMON (Teil 2) | 61 | 12/84 |
| | Ohne gutes Werkzeug geht es nicht: SMON (Teil 3) | 69 | 1/85 |
| | Ohne gutes Werkzeug geht es nicht: SMON (Teil 4) | 72 | 2/85 |
| Morsen | Kurz - lang - kurz: kein Problem | 72 | 6/84 |
| Musik | Die Musik macht der C 64: Elektronikorgel (AdM) | 70 | 9/84 |
| | Musik, Musik, Musik: Synthesizer (AdM) | 51 | 12/84 |
| Schach | Schachmeister (AdM) | 50 | 11/84 |
| Sport | Computer und Sport - Auswertung von DNM-Ereignissen | 56 | 4/84 |
| | Der C 64 als Handballtrainer (AdM) | 52 | 1/85 |
| | Gut Ziel mit dem C 64 (AdM) | 52 | 3/85 |
| | Ohne Organisation kein Tor: Ligabat (LdM) | 50 | 3/85 |
| Strubs | Das erste Strubs-Listing (VC 20) | 85 | 7/84 |
| Super 8 | VC 20 steuert Super 8-Kamera | 70 | 2/85 |
| Textverarbeitung | Quicktext - Textverarbeitung zum Abtippen | 60 | 6/84 |
| User-Port | Analoger Meßwert rein - analoger Stellwert raus | 78 | 8/84 |
| | Blumen gießen mit dem C 64 (AdM) | 82 | 7/84 |
| | Kopplung über den User-Port (VC 20) | 73 | 8/84 |
| Video | Video-Vorspann mit dem VC 20 | 80 | 10/84 |
| Vokabeln | Supervoc | 69 | 6/84 |
| Zeichensatz | Deutscher Zeichensatz für den VC 20 | 79 | 9/84 |
| Grafik | | | |
| Algorithmus | Ein schneller Drawline-Algorithmus | 65 | 4/84 |
| Auflösung | Supergrafik II (VC 20) | 73 | 11/84 |
| | Supergrafik ohne Erweiterungsmodul | 81 | 5/84 |
| Axiometrie | Von allen Seiten betrachtet (SB) | 69 | 12/84 |
| Befehlsweiterung | Screen Change | 94 | 9/84 |
| Diagramme | Grafische Darstellung statt Zahlenriedhof (SB) | 82 | 6/84 |
| Elektrotechnik | Elektrotechnisches Zeichnen mit dem VC 20 | 71 | 3/85 |
| Funktionen | Grafik leicht gemacht | 70 | 11/84 |
| | Kudiplo erfüllt Schülerträume (Kurvendiskussion auf dem VC 20) | 80 | 8/84 |

ACHTUNG: Die Ausgaben 5, 6 und 11/85 sind vergriffen und nicht mehr lieferbar!

| Stichwort | Titel | Seite | Ausgabe |
|----------------------|--|-------|---------|
| | Hydra-Load mal vier | 82 | 1/85 |
| | Kopieren mit Komfort: Super Copy | 102 | 10/84 |
| | Kopierprogramm für relative Files | 132 | 7/84 |
| | Maschinenprogramme auf Diskette speichern | 91 | 2/85 |
| | View B&M | 99 | 8/84 |
| Funktionen | Kudiplo auf für den C 64 (Kurvendiskussion) | 91 | 10/84 |
| Generator | Dem Springvogel auf die Sprünge geholfen | 82 | 11/84 |
| Grafik | Tips und Erweiterungen zu Hi-Eddi und Simons Basic | 88 | 3/85 |
| | Vom Bildschirm auf Kassetten/Diskette | 135 | 7/84 |
| Hex-Data | Unterbrechen Sie mich bitte! | 81 | 11/84 |
| Interrupt | Cursorsteuerung leicht gemacht (mit Joystick) | 84 | 11/84 |
| Joystick | Joystick-Abfrage in Theorie und Praxis | 83 | 11/84 |
| | Der große Überblick: formatiertes Listing | 90 | 10/84 |
| Listing | Fehleruche leicht gemacht: LISTSTOP | 97 | 9/84 |
| | Programmiertes LISTing: LIST XY | 100 | 10/84 |
| | List- und Löserschutz leicht gemacht | 85 | 12/84 |
| Listenschutz | Maschinenprogramme auf Tastendruck | 80 | 12/84 |
| Maschinen-sprache | DTP-Wandler | 94 | 10/84 |
| Merge | Kleben per Software - Merge | 102 | 9/84 |
| Module | Steckmodule auf Kassetten (VC 20) | 107 | 6/84 |
| Monitor | Besseres Monitorbild beim C 64 | 90 | 2/85 |
| Opocodes | Hex-eieren: undefinierte Opocodes beim 6502 | 84 | 3/85 |
| POKES | Durch POKES zum Erfolg - Die Spiele-Trickkiste | 83 | 3/85 |
| | POKE mal wieder: diverse POKES | 91 | 10/84 |
| Parameter | Parameterübergabe an Programme in Maschinen-sprache | 88 | 1/85 |
| Reset | Reset für den VC 20 | 116 | 6/84 |
| | Resetschalter am C 64 | 34 | 8/84 |
| Restore | Restore für Unterprogramme | 90 | 1/85 |
| Retten | Erste Hilfe (VC 20) | 88 | 4/84 |
| | Erste Hilfe für den C 64: RENEW | 102 | 9/84 |
| SYS | Mehr über SYS | 131 | 7/84 |
| Schnittstellen | Die RS232-Schnittstelle am VC 20 | 100 | 9/84 |
| | Komfortables Treiberprogramm für Centronics-Drucker | 110 | 7/84 |
| | Verbindungs-freundlich (VC 20) | 91 | 3/85 |
| Scrollen | Alle Bilder laufen lassen - (Scrollen) | 86 | 2/85 |
| Simons | Befehlsweiterung für Simons Basic | 90 | 10/84 |
| Basic | Haben Sie den Bogen raus? (ARC bei Simons Basic) | 98 | 9/84 |
| | Simons Basic: Befehle die nicht im Handbuch stehen | 103 | 9/84 |
| Speicher | RAM-Floppy | 92 | 2/85 |
| Synthetische | Die Suche nach den Synthetischen | 104 | 9/84 |
| | Synthetische Steuerzeichen | 114 | 6/84 |
| | Synthetische Steuerzeichen (Teil 2) | 136 | 7/84 |
| Tastatur | Yusef-Tastatur (Zehnerastatur) | 93 | 10/84 |
| Tips & Tricks | Diverses | 89 | 10/84 |
| | Diverse Tips & Tricks für den C 64 | 86 | 5/84 |
| | Lösung von Dallas Quest | 90 | 1/85 |
| | Tips & Tricks für den C 64 | 108 | 4/84 |
| | Tips & Tricks für den VC 20 | 118 | 6/84 |
| Trace | Trace und Single Step für Maschinenprogramme | 76 | 12/84 |
| Transfer | Adressenvergleich VC 20-C 64 | 52 | 7/84 |
| | Der C 64 als PET | 67 | 1/85 |
| | Die Software-Vielzahl der CBMs für den C 64 nutzen | 102 | 8/84 |
| | Von den Kleinen auf die Großen (C 64 - CBM) | 96 | 8/84 |
| User-Port | User-Port-Display | 97 | 8/84 |
| | User-Port-Tastatur (Zehnerastatur) | 93 | 10/84 |
| Hardware-Test | | | |
| 80-Zeichenkarten | Mehr Übersicht am Bildschirm (VC 20) | 20 | 10/84 |
| Computer | C 16: Der Nachfolger? | 20 | 6/84 |
| | Ein Wolf im Schafspelz - der 264 (Plus/4) | 14 | 5/84 |
| | Generationswechsel - Test C 16 | 6 | 1/85 |
| | Plus und Minus beim Plus/4 | 14 | 2/85 |
| | SX 64 im Test | 27 | 4/84 |
| Drucker | Adcomp X100 - farbig plotten und drucken | 22 | 10/84 |
| | Brother HR-SC: fast nicht zu hören | 24 | 10/84 |
| | Commode-Drucker unter 1000 Mark | 20 | 4/84 |
| | Der Petal MA20 - kleiner Name, großer Drucker | 24 | 11/84 |
| | Die farbige druckende Kreissäge GP-700A | 24 | 5/84 |
| | Drucksymptome (BMC EX100) | 26 | 11/84 |
| | Ein Drucker für alle Fälle: Epson FX-80 | 23 | 10/84 |
| | Ein Star der es in sich hat (delta-10) | 25 | 10/84 |
| | Ein starkes Stück (Itoh 8510) | 22 | 11/84 |
| | Olympic electronic compact 2: Schreibmaschine für den C 64 | 28 | 10/84 |
| | Roland DXY-101 - ein Flachbettplotter im DIN-A3-Format | 27 | 10/84 |
| | Seikosha Größter: Test GP-550A | 26 | 10/84 |
| EPROM | EPROM-Brenner: Vergleichstest | 36 | 8/84 |
| Erweiterungen | KFC-Super - ein Super-Toolkit? | 20 | 5/84 |
| Expansions | Expansions über alle Grenzen hinaus | 34 | 4/84 |
| Floppy | Floppy mit Nachbrennen (Speeddos, Turboaccess, Computing) | 26 | 12/84 |
| | M5D-Super-Disk-Drive | 14 | 11/84 |
| Grafik | Grafikpakete: KoalaPad und SuperSketch | 34 | 11/84 |
| Interface | Das macht den kleinen größer (64-KByte-Karten) (VC 20) | 112 | 9/84 |
| | Digitalisierte Bilder mit dem C 64 | 24 | 1/85 |
| | Speicherumtun für den C 64: MR 64 | 26 | 1/85 |
| Joystick | Joystick-Vielfalt (20 Joysticks im Test) | 34 | 12/84 |
| Monitore | Die Schorfmarkter (Cable, Taxan, BMC) | 20 | 12/84 |
| | Monitor Kontra Fernseher (Taxan, 1701, Zenith, Sharp, Panasonic) | 16 | 11/84 |
| Schnittstellen | Card/Print + G - Das Allround-Interface (Centronics) | 20 | 3/85 |
| | Centronics-Schnittstellen: Vergleichstest | 12 | 7/84 |
| | Das Interface mit Watlbick (WW92000/G: Centronics) | 18 | 3/85 |
| | Hardware-Interface ganz weich: EC-64 | 23 | 1/85 |
| | Print 64 - das universelle Interface (Centronics) | 24 | 12/84 |
| Sprachausgabe | Die Stimme des Meisters: Voice Master | 19 | 2/85 |
| | Spuchausgabe mit dem SDP 120 | 22 | 8/84 |
| Hardware | | | |
| Bauanleitung | 16-KByte-Erweiterung umschaltbar (VC 20) | 20 | 2/85 |
| | Commode im neuen Kleid | 30 | 8/84 |
| | Das 30-Mark-Interface (RS232) | 29 | 3/85 |
| | Ihr Akustikkoppler wird zum Modem: Automodem | 114 | 9/84 |
| | Joystick im Selbstbau | 33 | 3/85 |
| | Resetschalter am C 64 | 34 | 8/84 |
| | Rechtig verbunden - Video/Audio-Kabel für den C 64 | 22 | 2/85 |
| DFU | Akustikkoppler und Modem: Marktübersicht | 28 | 8/84 |
| Datensatz | Rund um die Datensätze | 30 | 6/84 |
| Drucker | MPS 801 - Ein Erfahrungsbericht | 20 | 8/84 |
| | Marktübersicht: Drucker (Teil 1) | 29 | 10/84 |
| | Marktübersicht: Drucker (Teil 2) | 21 | 11/84 |
| | Nichts ist ewig (ROM-Change, verbessertes Betriebssystem) | 30 | 12/84 |
| EPROM | Expansions - Marktübersicht | 18 | 7/84 |
| Expansion | Der serielle Bus des VC 20 und C 64 | 28 | 5/84 |
| Grundlagen | Richtig verbunden - Video/Audio-Kabel für den C 64 | 22 | 2/85 |
| Monitor | Marktübersicht: Schwarzweiß- und Farbmonitore | 19 | 11/84 |
| | MIDI - Glanz und Elend eines Interfaces | 46 | 9/84 |
| Musik | Geheimnissen auf der Spur: IS41 reparieren | 24 | 8/84 |
| Reparatur | Centronics-Schnittstelle für Epson FX-80 (mit Kabel) | 24 | 6/84 |
| Schnittstellen | Erst ein IEC-Bus öffnet Tür und Tor (Marktübersicht und Test) | 24 | 3/85 |
| | Gute Connections (RS232: Centronics-Marktübersicht) | 21 | 3/85 |
| | Kopplung zwischen zwei VC 20 | 24 | 6/84 |

| Stichwort | Titel | Seite | Ausgabe | | |
|---|--|--|---|-------|------|
| Kurse | | | | | |
| Assembler | Assembler ist keine Alchimie (Teil 1) | 138 | 9/84 | | |
| | Assembler ist keine Alchimie (Teil 2) | 150 | 10/84 | | |
| | Assembler ist keine Alchimie (Teil 3) | 121 | 11/84 | | |
| | Assembler ist keine Alchimie (Teil 4) | 134 | 12/84 | | |
| | Assembler ist keine Alchimie (Teil 5) | 142 | 1/85 | | |
| | Assembler ist keine Alchimie (Teil 6) | 134 | 2/85 | | |
| | Assembler ist keine Alchimie (Teil 7) | 124 | 3/85 | | |
| Codes | Alle Tasten, Zeichen- und Steuerodes (Teil 1) | 114 | 4/84 | | |
| | Alle Tasten, Zeichen- und Steuerodes (Teil 2) | 104 | 5/84 | | |
| | Alle Tasten, Zeichen- und Steuerodes (Teil 3) | 146 | 7/84 | | |
| | Alle Tasten, Zeichen- und Steuerodes (Teil 4) | 151 | 8/84 | | |
| Comal | Comal - Eine Einführung (Teil 1) | 44 | 11/84 | | |
| | Comal - Eine Einführung (Teil 2) | 145 | 12/84 | | |
| Eff. Prog. | Comal - Eine Einführung (Teil 3) | 130 | 2/85 | | |
| | Finden mit System - Eine neuartige Suchmethode (Teil 3) | 142 | 3/85 | | |
| Floppy | Müllabfuhr im Computer: Die Garbage Collection (Teil 1) | 128 | 1/85 | | |
| | Stringprogrammierung in Maschinensprache (Teil 2) | 147 | 2/85 | | |
| | In die Geheimnisse der Floppy eingetaucht (Teil 1) | 153 | 10/84 | | |
| | In die Geheimnisse der Floppy eingetaucht (Teil 2) | 117 | 11/84 | | |
| | In die Geheimnisse der Floppy eingetaucht (Teil 3) | 139 | 12/84 | | |
| | In die Geheimnisse der Floppy eingetaucht (Teil 4) | 148 | 1/85 | | |
| Grafik | In die Geheimnisse der Floppy eingetaucht (Teil 5) | 130 | 3/85 | | |
| | Hires 3 (Teil 1) | 123 | 2/85 | | |
| | Hires 3 (Teil 2) | 136 | 3/85 | | |
| | Reise durch die Wunderwelt der Grafik (Teil 1) | 119 | 4/84 | | |
| | Reise durch die Wunderwelt der Grafik (Teil 2) | 109 | 5/84 | | |
| Grundlagen | Reise durch die Wunderwelt der Grafik (Teil 3) | 132 | 6/84 | | |
| | Reise durch die Wunderwelt der Grafik (Teil 4) | 162 | 7/84 | | |
| | Reise durch die Wunderwelt der Grafik (Teil 5) | 142 | 8/84 | | |
| | Reise durch die Wunderwelt der Grafik (Teil 6) | 144 | 9/84 | | |
| | Reise durch die Wunderwelt der Grafik (Teil 7) | 146 | 10/84 | | |
| | Geschwindigkeit durch Maschinencode - so arbeiten Compiler | 39 | 2/85 | | |
| | Musik | Dem Klang auf der Spur (Teil 1) | 131 | 12/84 | |
| | | Dem Klang auf der Spur (Teil 2) | 136 | 1/85 | |
| | | Dem Klang auf der Spur (Teil 3) | 152 | 2/85 | |
| | Precompiler | Strubs - ein Precompiler für Basic-Programme (Teil 1) | 110 | 4/84 | |
| Strubs - ein Precompiler für Basic-Programme (Teil 2) | | 116 | 5/84 | | |
| Strubs - ein Precompiler für Basic-Programme (Teil 3) | | 128 | 6/84 | | |
| Speicher | Strubs - ein Precompiler für Basic-Programme (Teil 4) | 154 | 7/84 | | |
| | Memory Map mit Wandervorschlägen (Teil 1) | 133 | 11/84 | | |
| | Memory Map mit Wandervorschlägen (Teil 2) | 132 | 12/84 | | |
| | Memory Map mit Wandervorschlägen (Teil 3) | 127 | 1/85 | | |
| VC 20 | Memory Map mit Wandervorschlägen (Teil 4) | 150 | 2/85 | | |
| | Memory Map mit Wandervorschlägen (Teil 5) | 144 | 3/85 | | |
| | Der gläserne VC 20 (Teil 1) | 155 | 9/84 | | |
| | Der gläserne VC 20 (Teil 2) | 157 | 10/84 | | |
| | Der gläserne VC 20 (Teil 3) | 126 | 11/84 | | |
| Spiele-Test | Der gläserne VC 20 (Teil 4) | 130 | 1/85 | | |
| | Der gläserne VC 20 (Teil 5) | 141 | 2/85 | | |
| | Der gläserne VC 20 (Teil 6) | 155 | 3/85 | | |
| | Abenteuer | Abenteuer selbst gemacht - Adventure Creator | 43 | 11/84 | |
| | Die Lösung von Hobbit | 49 | 2/85 | | |
| | Exodus - Ultima III | 42 | 11/84 | | |
| | Gordon Saga | 48 | 2/85 | | |
| | Gruds in Space | 137 | 8/84 | | |
| | House of Usher | 37 | 10/84 | | |
| | Lösung von Dallas Quest | 96 | 1/85 | | |
| | Lösung von Exchanter | 44 | 3/85 | | |
| Action | Lösung von The Blade of Blackpool | 34 | 10/84 | | |
| | The Quest | 47 | 1/85 | | |
| | Dino Eggs | 57 | 6/84 | | |
| | Flip and Flop | 48 | 4/84 | | |
| | Impossible Mission | 46 | 2/85 | | |
| | Loderunner | 68 | 7/84 | | |
| | OK 9. Catastrophes | 48 | 12/84 | | |
| | Raingame | 62 | 5/84 | | |
| | Save New York und Survivor | 46 | 4/84 | | |
| | Tom + Zaga | 48 | 1/85 | | |
| | Arcade | Wizard | 49 | 12/84 | |
| Zaxxon | | 68 | 7/84 | | |
| AE - ein Action-Spiel | | 56 | 6/84 | | |
| Fire Galaxy (VC 20) | | 37 | 10/84 | | |
| Flipper | Schnellboot - Rettung aus der grünen Höhle | 109 | 9/84 | | |
| | David's Midnight Magic | 60 | 5/84 | | |
| | Night Mission | 61 | 5/84 | | |
| | Slamball - der ellenlange Flipper | 105 | 9/84 | | |
| | Fantasy-Spiele | 106 | 9/84 | | |
| Grundlagen | Flight II - fast wie richtiges Fliegen | 69 | 7/84 | | |
| | One on One | 136 | 8/84 | | |
| Simulation | Spiel des Jahres: International Soccer | 46 | 12/84 | | |
| | Summer Games - Los Angeles läßt grüßen | 138 | 8/84 | | |
| Sport | Taktik- und Strategiespiele | 46 | 3/85 | | |
| | Taktik | | | | |
| So machen's andere | Amateurfunk | Funkende Computer | 132 | 4/84 | |
| | Datenbank | Klein aber oho - der VC 20 | 136 | 4/84 | |
| | Finanzen | Geregelter Zahlungsverkehr | 164 | 9/84 | |
| | Fotografie | Der VC 20 als Fotolehrling | 138 | 6/84 | |
| | Landwirtschaft | Mode-Fotos mit Bits und Bytes | 144 | 11/84 | |
| | Lichttelefon | Der Computer im Kuhstall | 156 | 8/84 | |
| | Medizin | Mit vier Baud über den Balkon | 156 | 10/84 | |
| | Orientierung | Computer bringen den Kreislauf in Schwung | 170 | 7/84 | |
| | Platzproblem | Welche Hausnummer hat der Kölner Dom? | 128 | 5/84 | |
| | | Wohn mit dem Heimcomputer? | 136 | 5/84 | |
| | Software-Test | Assembler | Assembler im Test (AS-64, MAE, TEX AS, ASSI/M) | 34 | 1/85 |
| | | Basic-Erweiterung | Assembler im Test: Mastercode, Profimat, Profisoft, Maschine 64 | 30 | 2/85 |
| | | | Assembler? Assembler! | 32 | 1/85 |
| | | | Exbasic Level II - eine tolle Sache | 42 | 6/84 |
| | | | GBasic | 28 | 1/85 |
| | | Büro | Simons Basic (Teil 1) | 40 | 4/84 |
| | | | Simons Basic (Teil 2) | 42 | 5/84 |
| Magic Desk - Bildschirm statt Schreibtisch | | | 62 | 7/84 | |
| CP/M | | | Erste Erfahrungen mit dem CP/M-Modul | 184 | 4/84 |
| Compiler | | Basic-Programme auf Trab gebracht: Austro-Speed, BASS, Exbasic, Popspeed | 34 | 2/85 | |
| | | DFÜ | Teleterm - Die Verbindung zum Modem | 20 | 7/84 |
| Datenbank | | Terminal 64 - Schwer auf Draht | 24 | 2/85 | |
| | Daten gut im Griff (Datamat, Multidata, Datenmanager) | 52 | 5/84 | | |
| | ISM 64 - ohne Fleiß kein Preis | 117 | 8/84 | | |
| Finanzen | ISM 64 - professionelle Datenverwaltung | 59 | 5/84 | | |
| | Mandat 64 | 56 | 5/84 | | |
| | Superbase 64 | 46 | 5/84 | | |
| | Floppy | Lohnsteuerjahresausgleich leicht gemacht | 46 | 10/84 | |
| Grafik | Ex-DOS und Disk Doctor | 48 | 10/84 | | |
| | Quickcopy - das schnelle Kopierprogramm | 28 | 9/84 | | |
| | Elektronische Aquarelle: Paint Magic | 114 | 8/84 | | |
| | Grafik hoch zwei - das Extended Graphic System | 37 | 11/84 | | |
| | Grafiktablets: KoalaPad und SuperSketch | 34 | 11/84 | | |
| | Graphics-Basic (HES) | 38 | 12/84 | | |

| Stichwort | Titel | Seite | Ausgabe | |
|---|--|---|--|-------|
| Lernprogramme | Wie Super ist die Supergrafik? (Supergrafik 64) | 30 | 11/84 | |
| | Melodienschreiber und Musik-Synthesizer | 43 | 12/84 | |
| Mathematik | Nachhilfe auf Knopfdruck (Mathematik) | 26 | 2/85 | |
| | SoftLearning - die weiche Welle des Lernens | 40 | 1/85 | |
| | Vokaltraining mit dem Computer | 39 | 3/85 | |
| | Was bringt die Lern-Software? | 42 | 12/84 | |
| | Nachhilfe auf Knopfdruck (Mathematik) | 26 | 2/85 | |
| Musik | Gute Noten für gute Noten: Extender Synthesizer System | 24 | 9/84 | |
| | Melodienschreiber und Musik-Synthesizer | 43 | 12/84 | |
| | MUSIC-COMPOSER - Komponieren leicht gemacht | 42 | 9/84 | |
| | Musicalc - oder was wirklich im C 64 steckt | 29 | 9/84 | |
| | Synthimat - Das Piano für den Aktenkoffer | 38 | 9/84 | |
| | Basic Bär - Ein Programmgenerator | 65 | 7/84 | |
| | Die Turbo-Pascal-Story | 40 | 12/84 | |
| | Forth ohne Floppy (C 64 und VC 20) | 50 | 10/84 | |
| | HES 64 Forth - Komfortabler als Basic | 66 | 7/84 | |
| | Oxford-Pascal für den Commodore | 39 | 12/84 | |
| Programmgenerator | Pascal - leistungsfähiger und eleganter als Basic (Teil 2) | 44 | 8/84 | |
| | Sechs Pascal-Versionen im Vergleich | 50 | 8/84 | |
| Sprachen | Multiplan: Viel zu schade, um nur damit zu kalkulieren | 32 | 11/84 | |
| | Calc Result - Dreidimensionale Kalkulation | 21 | 9/84 | |
| Tabellenkalkulation | VivaStar - Ein Stern wird geboren | 38 | 11/84 | |
| | Elstext - schnell wie der Blitz? | 54 | 6/84 | |
| | Homeward - Textverarbeitung zu Hause | 35 | 3/85 | |
| | SM-Text 64 - die professionelle Textverarbeitung | 48 | 6/84 | |
| Textverarbeitung | Textomat - Büroanwendung zum kleinen Preis | 34 | 9/84 | |
| | TotText - Flexibilität ist Trumpf | 38 | 3/85 | |
| | Vizawrite 64 - Der C 64 wird zum PC | 43 | 10/84 | |
| | Wordpro 3 plus - Von den Großen auf den Kleinen | 52 | 6/84 | |
| | Vokaltraining mit dem Computer | 39 | 3/85 | |
| Software | Adressenvergleich VC 20-C 64 | 52 | 7/84 | |
| | Adressenvergleich | Basic | 46 | 7/84 |
| | | Fehlersuche in Basic-Programmen (Teil 2) | 67 | 9/84 |
| | | CP/M | 37 | 7/84 |
| | | Compiler | Geschwindigkeit durch Maschinencode - so arbeiten Compiler | 39 |
| | DFÜ | Begriffe aus der DFÜ | 27 | 7/84 |
| | | Datex-P und ausländische Netzwerke | 59 | 10/84 |
| | | Electronic Mail - die neue Form der Postbeförderung | 22 | 7/84 |
| | | Mailboxprogramm für den C 64 | 114 | 9/84 |
| | EPROM | Fehlersuche in Basic-Programmen (Teil 1) | 24 | 7/84 |
| | | Wie bedient man eine Mailbox? | 144 | 3/85 |
| | Floppy | Datenbrenner: Wie programmiere ich EPROMs? | 162 | 9/84 |
| | | 22 Read Error - Theorie und Praxis | 41 | 3/85 |
| | Grafik | Neues vom Video-Chip beim VC 20 | 56 | 8/84 |
| | | Daten im (relativen) Direktzugriff | 58 | 7/84 |
| | Grundlagen | Datenkreislauf: Die sequentielle Datenspeicherung | 63 | 8/84 |
| | | Die index-sequentielle Datei | 54 | 9/84 |
| Flußdiagramme | | 20 | 9/84 | |
| So macht man Basic-Programme schneller (Teil 2) | | 44 | 12/84 | |
| Musik | Strukturiertes Programmieren (Teil 1) | 33 | 5/84 | |
| | Strukturiertes Programmieren (Teil 2) | 37 | 6/84 | |
| | Tips für den Umgang mit Sinnbildern (Flußdiagrammen) | 14 | 9/84 | |
| | Tips für sauberes Programmieren | 38 | 4/84 | |
| | Hard und Soft: eine kleine Markttüber- | 58 | 9/84 | |
| | Klangsynthese und Synthesizerchnik | 62 | 9/84 | |
| | Marktübersicht der Musikprogramme | 27 | 9/84 | |
| | Basic ist out - Es lebe Forth | 43 | 1/85 | |
| | Forth - die etwas andere Programmiersprache | 33 | 7/84 | |
| | Pascal - leistungsfähiger und eleganter als Basic | 40 | 7/84 | |
| Pascal - leistungsfähiger und eleganter als Basic | 44 | 8/84 | | |
| Sprachen | Was ist Comal? | 41 | 8/84 | |
| | Textverarbeitungsprogramme - Marktübersicht | 47 | 6/84 | |
| Textverarbeitung | Von der Schreibmaschine zum Textsystem (Auswahlhilfe) | 34 | 3/85 | |
| | DOS 3.1 (Teil 1) | 40 | 5/84 | |
| | DOS 3.1 (Teil 2) | 16 | 9/84 | |
| Wettbewerbe | Einzeiler | Die Top 10 (Einzeiler) | 158 | 11/84 |
| | Kreuzworträtsel | Einzeiler-Wettbewerb: Die nächsten 14 Kreuzworträtsel selber machen | 157 | 1/85 |
| | Sprits | Das schönste Sprite - Auflösung | 174 | 7/84 |
| | Unterprogramm | Formatierte Eingabe | 155 | 1/85 |
| | Sieger mit Maske - Maskenerstellungsprogramm | 154 | 11/84 | |
| | Sieger mit Maske - Maskenerstellungsprogramm | 172 | 10/84 | |

Alle Beiträge sind in der Regel für den C 64, sofern nicht anders gekennzeichnet (VC 20).
Folgende Abkürzungen wurden verwendet: LDM = Listings des Monats, AdM = Anwendung des Monats, SB = Simons Basic

Neu: die 64'er-Sammelbox!

Für alle Leser, die 64'er regelmäßig kaufen, sammeln oder im Abonnement beziehen, gibt es jetzt ein interessantes Service-Angebot: die 64'er-Sammelbox!

Mit dieser Sammelbox bringen Sie nicht nur Ordnung in Ihre wertvollen Hefte, sondern schaffen sich gleichzeitig ein interessantes und attraktives Nachschlagewerk.

Übrigens: Die Sammelbox ist nicht nur ein praktisches Aufbewahrungsmittel: Sie eignet sich auch hervorragend als Geschenk für Freunde und Bekannte zu vielen Anlässen.

Ein kompletter Jahrgang (12 Hefte) paßt in die praktische Sammel-Box! Am besten gleich bestellen!

Auch die bisher erschienenen Sonderhefte können Sie jetzt direkt bestellen:



TIPS & TRICKS
(1. Programm-Sonderheft)
Eine wahre Fundgrube an Ideen und Programmen für Computer-Profis und alle, die es werden wollen.
BESTELLCODE: Tips & Tips



ABENTEUERSPIELE
(2. Programm-Sonderheft)
Auf mehr als 100 Seiten viele interessante Adventures, die Sie lange Zeit fesseln werden. Mit abgeschlossenem Kurs zur Programmierung eigener Abenteuerspiele und zahlreichen Lösungen professioneller Adventures.
BESTELLCODE: Abenteuerspiele



SPIELE
(3. Programm-Sonderheft)
Heiße Listings für alle Spiele-Fans: Sportspiele, Schießspiele, Denkspiele, Spielegeneratoren, Abenteuerspiele, Brettspiele, Taktikspiele, Geschicklichkeitsspiele und eine Marktübersicht aller in Deutschland erhältlichen professionellen Spiele bringen alles, was das Herz der Spiele-Fans höher schlagen läßt.
BESTELLCODE: Spiele



GRAFIK & DRUCKER
(4. Programm-Sonderheft)
Randvoll mit Informationen: Rund 28 Listings der Spitzenklasse. Darunter Top-Listings zur räumlichen Darstellung von Körpern aus beliebigen Betrachtungsrichtungen. Weiters: Prüfsummenlistings, Drucker-Anwendung, Basic-Erweiterung, Hardcopy-Routinen, Zeichengenerator, Grundlagen, Tips & Tricks.
BESTELLCODE: Grafik & Drucker

ProDat — Dateiverwal- tung mit Raffinessen

Das Besondere an »ProDat« ist, daß man die Eingabemaske wie bei den teuren kommerziellen Programmen selbst definieren kann. Außerdem ist »ProDat« durch verschiedene Maschinenprogramme besonders schnell und einfach zu bedienen.

Wenn man das Listing vollständig und fehlerfrei eingetippt hat, darf man erst mal aufatmen. Elf KByte Basic sind schließlich kein Honiglecken. Im Bereich von \$0334 (820) bis \$03cf (1020) stehen verschiedene Maschinenprogramme. Da dies aber der Kassettenpuffer ist, läßt sich »ProDat« nicht ohne weiteres für Datasette verwenden.

Die Daten werden von Maschinenroutinen geladen und gespeichert. Auch das Kopieren der Bildschirmmaske der aktuellen Datei in den Bildschirmspeicher geschieht in Maschinensprache. Da die INPUT-Routine des Betriebssystems geringfügig geändert wurde, bereitet bei der Maskenerstellung auch die Eingabe von Anführungszeichen, Doppelpunkten oder Kommata keinerlei Schwierigkeiten. Um bei der Verarbeitung der Daten der gefürchteten Garbage-Collection aus dem Weg zu gehen, wurde eine etwas geänderte Version des Stringvertauschers von Boris Schneider aus der 64'er, Ausgabe 1/85 verwendet. Für Interessierte zeigt Tabelle 1 eine dokumentierte Liste der wichtigsten im Programm verwendeten Variablen.

Hinweise zum Abtippen:

Manche Zeilen haben Überlänge, das heißt, sie sind länger als die vom Basic-Interpreter erlaubten 80 Zeichen. Diese Zeilen können Sie nur mit den im Anhang D des Commodore-Handbuchs vermerkten Abkürzungen der Basic-Befehle eingeben. Das gilt zum Beispiel für die Zeile 360. Leerzeichen vor und nach den einzelnen Basic-Befehlen dienen nur der Übersichtlichkeit und können beim Eintippen ersatzlos weggelassen werden.

Programmbeschreibung:

Menü 1, Hauptmenü

- 1 — Aufruf Menü 2
- 2 — Neue Datei definieren

Zuerst muß die Eingabemaske definiert werden. Wie das aussehen kann, zeigt Bild 1. Sie können frei mit den Cursor-Tasten auf dem Bildschirm umherwandern und Ihr persönliches Eingabeformular erstellen. Dabei sind alle Tasten und Zeichen erlaubt. (Vorsicht vor SHIFT-CLR!) Zahlen zum Beachten: Maximale Felderanzahl (in der Sprache der Dateiverwaltung wird jede Stelle in der Maske, an der Sie später eine Eingabe machen, als Feld bezeichnet): 1 500 pro Datei, 40 pro Datensatz (ein Datensatz ist eine mit Daten gefüllte Eingabemaske). Maximale Feldlänge: 80 Zeichen.

Wie das Beispiel Bild 1 zeigt, werden Felder mit Hochpfeilen markiert. Wird der abschließende Pfeil weggelassen, so nimmt das Programm hierfür die maximale Feldlänge von 80 Zeichen.

Wird später bei der Eingabe der Schlußpfeil erreicht, so springt der Cursor automatisch zum nächsten Feld weiter. Die Pfeile selbst tauchen später nicht mit in der Eingabemaske auf (siehe Bild 2). Mit dem Pfeil nach links können Sie eine Datensatznummeranzeige programmieren. Das heißt, daß an dieser Stelle später die Nummer des Datensatzes innerhalb der Datei ausgedruckt wird. Weiterhin ist wichtig, daß Sie die letzte Bildschirmzeile nicht verwenden dürfen. Ein Feld in der vorletzten Zeile darf also nur 40 Zeichen lang sein! Nachdem die Eingabemaske definiert ist, müssen noch die Druckparameter eingegeben werden (siehe nächster Menüpunkt).

3 — Druckparameter festlegen/ändern

Nun müssen Sie festlegen, welche Felder in welcher Reihenfolge später auf dem Drucker ausgedruckt werden sollen. Ein Feld wird mit der »eckigen Klammer auf« markiert (siehe Bild 3). Sie können natürlich auch erklärenden Text oder ähnliches ausdrücken lassen. Alles, was zwischen den »eckigen Klammern zu« steht, wird später im Breitschrift-Modus gedruckt. Durch Änderung der Zeilen 2 570 und 2 580 hat man auch andere Möglichkeiten, zum Beispiel Kursiv-Druck oder dergleichen. Es können maximal 60 Druckzeilen definiert werden. Zu beachten ist, daß ein angesprochenes Feld auch wirklich vorhanden ist, da dies nicht überprüft wird. Sie können ein Feld auch mehrmals ausgeben lassen, das liegt ganz bei Ihnen. Eine Druckzeile kann maximal 80 Zeichen fassen, wodurch beispielsweise auch kleinere Formbriefe möglich sind.

4 — Directory laden

5 — File löschen

6 — Validate

7 — Disk formatieren

8 — File umbenennen

Die Menüpunkte 4 bis 8 erklären sich von selbst, beziehungsweise sind im Floppy-Handbuch genauer erklärt.

| | |
|----------|--|
| a%(x,y) | Cursorposition für Maske |
| a1% | Zeile für Datensatznummeranzeige |
| a2% | Spalte für Datensatznummeranzeige |
| d, d\$ | Anzahl Felder je Datei |
| ds | aktuelles Feld |
| df | Anzahl Druckfelder |
| f | Anzahl Felder je Datensatz |
| f1,f2,f3 | Für Fehlermeldung |
| fe\$(x) | Feld |
| fl | Flag für alles mögliche |
| n\$,na\$ | Name der Datei |
| pz\$(x) | Druckzeile |
| ud | Flag für unveränderte, geänderte oder nicht im Speicher stehende Datei |

Tabelle 1. Die wichtigsten Variablen.

| | |
|----|----------------------|
| f1 | Ein Feld zurück |
| f3 | Feld löschen |
| f5 | Anzeige »Bytes free« |
| f7 | Eingabe beenden |

Tabelle 2. Tastenbelegung im Anzeigemodus.

| | |
|----|---|
| + | Vorwärtsblättern |
| - | Rückwärtsblättern |
| l | Löschen angezeigter Datensatz |
| s | Suchen |
| f1 | Angezeigten Datensatz formatiert ausgeben |
| f3 | Hardcopy erstellen |
| e | Datei erweitern (Eingabemodus) |
| a | Datensatz ändern |
| ↑ | Beenden |

Tabelle 3. Tastenbelegung im Eingabemodus.

```

┌──────────────────┐
│ ADRESSE NR.: ←   │
└──────────────────┘

Vorname: ↑
Name : ↑
Strasse: ↑
Telefon: ↑
Postltz: ↑   ↑Ort: ↑

Bemerkung: ↑
↑

```

Bild 1. So definiert man eine Eingabemaske.

```

┌──────────────────┐
│ ADRESSE NR.:     │
└──────────────────┘

Vorname:
Name :
Strasse:
Telefon:
Postltz:      Ort:

Bemerkung:

```

Bild 2. Beim Arbeiten mit der Datei sieht die Maske so aus.

```

┌──────────────────┐
│ Neue Datei definieren │
└──────────────────┘

Name der Datei? Adressen
Druckparameter eingeben!
 1. Druckzeile:
[01 [02
 2. Druckzeile:
[03
 3. Druckzeile:
[05 [06
 4. Druckzeile:

 5. Druckzeile:
]Telefon:] [04
 6. Druckzeile:
←
alles richtig (j/n)? j

```

Bild 3. So könnten die Druckparameter für unsere Beispielmaste aussehen.

```

┌──────────────────┐
│ ADRESSE NR.: 1   │
└──────────────────┘

Vorname: Wolfgang
Name : Hahn
Strasse: An der Stadtmauer 12
Telefon: 09253/1744
Postltz: 8671 Ort: Weissenstadt

```

Bild 4. Ein ausgefüllter Datensatz.

Menü 2, Datei verwalten

1 — Datei laden

Sollte eine erweiterte oder geänderte Datei noch im Speicher stehen, so wird sicherheitshalber erst gefragt, ob Sie nicht erst speichern möchten. Die vorhandene Datei wird sonst gelöscht.

2 — Datei pflegen/ändern/ergänzen

Nach Wahl dieses Punktes sind Sie zunächst im Anzeigemodus, der wie Bild 4 aussehen könnte. In diesem Modus sind einige Tasten mit Sonderfunktionen belegt. (Siehe Tabellen 2 und 3)

3 — Datei abspeichern

4 — Datei sortieren

Es besteht jetzt die Möglichkeit, entweder alphabetisch nach einem bestimmten Feld zu sortieren, oder man läßt sich bestimmte Daten herausuchen, zum Beispiel alle Adressen eines bestimmten Postleitzahlbereichs. Die kann man auf dem Drucker ausgeben lassen, oder speichert sie gleich in einer neuen Datei ab.

5 — Alle angegebenen Felder der Datei können aufaddiert werden, zum Beispiel bei einer Haushaltsbuchhaltung alle Ausgaben für Essen, etc.

6 — Aufruf Menü 3

Menü 3, Drucker Menü

1 — Alle Datensätze mit den vorher definierten Druckparametern auf dem Drucker ausgeben.

2 — Gesamte Datei ausdrucken.

3 — Entspricht Punkt 1, aber nur in Verbindung mit der Sortieroutine.

4 — Entspricht Punkt 2, aber nur in Verbindung mit der Sortieroutine.

Allgemeine Hinweise:

Sollte der Cursor einmal an einer Stelle stehenbleiben, und auch keine Taste mehr annehmen, so ist das kein Grund zur Panik. Der Computer braucht ab einer gewissen Datenmenge einfach Zeit, um das Eingeebene sinnvoll im Speicher unterzubringen. Wenn Sie »ProDat« als Textverarbeitungsprogramm verwenden wollen, so definieren Sie die Eingabefelder einfach über den gesamten Bildschirm und lassen die Datensatz-Nummeranzeige als zum Beispiel Seitennummer des Textes arbeiten. (Wolfgang Hahn/tr)

```

10 REM *****
20 REM *
30 REM * P R O D A T *
40 REM *
50 REM * AUTOR: WOLFGANG HAHN *
60 REM * AN DER STADTMAUER 12 *
70 REM * 8687 WEISSENSTADT *
80 REM * TELEFON: 09253/1744 *
90 REM *
100 REM *****
110 :
120 PRINT CHR$(14):GOSUB 2980:PRINT" {6DOWN
   >4 SEK.":GOSUB 3820
130 POKE 90,0:POKE 91,192:POKE 88,0:POKE 8
   9,192:POKE 780,0:POKE 781,160:SYS 820
140 POKE 90,0:POKE 91,0:POKE 88,0:POKE 89,
   0:POKE 780,0:POKE 781,224:SYS 820:POKE
   1,53
150 S=44136:FOR I=S TO S+8 STEP 4:POKE I,0
   :NEXT:S=43991:POKE S,3:POKE S+1,172:S=
   42349
160 POKE S,228:POKE S+1,180:FOR I=S+4 TO S
   +8:POKE I,234:NEXT:POKE 59012,96
170 POKE 1,55:CLR:GOSUB 3980
180 REM H A U P T M E N U
190 GOSUB 2980:PRINT"-1- DEFINIERTE DATEI
   BENUTZEN"
200 PRINT" {DOWN}-2- NEUE DATEI DEFINIEREN"
210 PRINT" {DOWN}-3- DRUCKPARAMETER AENDERN
   "

```

Listing »ProDat«. Beachten Sie bitte die Eingabehinweise auf Seite 6.

```

220 E=2:GOSUB 2970 <211>
230 PRINT"-4- DATEI LADEN" <209>
240 PRINT"<DOWN>-5- DATEI LOESCHEN" <214>
250 PRINT"<DOWN>-6- DATEI VALIDATE" <098>
260 PRINT"<DOWN>-7- DISK FORMATIEREN" <195>
270 PRINT"<DOWN>-8- DATEI UMBENENNEN" <236>
280 GOSUB 3010:IF E<1 OR E>8 THEN 280 <078>
290 ON E GOSUB 690,330,320,3200,3420,3500, <081>
3550,3610 <118>
300 GOTO 190 <053>
310 REM NEUE DATEI DEF. <249>
320 X=-1:IF D=0 THEN RETURN
330 GOSUB 2950:PRINT"(HOME,DOWN,BRIGHT,SPACE)NEUE DATEI DEFINIEREN(SPACE,2DOWN)":IF X=-1 GOTO 540 <084>
340 PRINT"<DOWN>MIT HARDCOPY (J/N/↑=ZURUECK)":GOSUB 3010:IF E$="↑"THEN RETURN <007>
350 PRINT"<CLR>" <084>
360 PRINT"(HOME)":POKE 1,53:POKE 58879,133:POKE 59639,24:POKE 780,0:SYS 58826:POKE 1,55:SYS 895 <013>
370 IF E$<>"J"THEN 390 <095>
380 GOSUB 2670:SYS 932:IF L=2 GOTO 380 <121>
390 CLR:GOSUB 3980:F=1:A1%=-1 <200>
400 FOR I=1024 TO 1983:IF PEEK(I)<>30 GOTO 450 <185>
410 AZ(F,Y)=(I-1024)/40:POKE I,32:FOR Z=0 TO 79:IF PEEK(I+Z)<>30 THEN NEXT Z <004>
420 Y=Y+1:AZ(F,Y)=I-(AZ(F,Y-1)*40+1024):Y=Y+1:AZ(F,Y)=Z:Y=0:I=I+Z:IF Z=80 THEN I=I-1 <188>
430 IF Z<80 THEN POKE I,32 <102>
440 F=F+1:IF F=40 GOTO 490 <025>
450 IF PEEK(I)=31 THEN A1%=(I-1024)/40:A2%=I-(A1%*40+1025):POKE I,32 <022>
460 NEXT I:IF F=1 THEN F=40:GOTO 490 <079>
470 FOR I=1984 TO 2023:IF PEEK(I)=30 OR PEEK(I)=31 THEN POKE I,32 <191>
480 NEXT I:IF AZ(F-1,0)=23 AND AZ(F-1,2)>40 THEN AZ(F-1,2)=40 <069>
490 F=F-1:IF F=39 THEN PRINT"<CLR,2DOWN>FELDERANZAHL PRUEFEN!!<DOWN>",&T$:GOSUB 3010:SYS 932:GOTO 360 <008>
500 SYS 895:IF A2%<0 THEN A2%=0 <061>
510 GOSUB 2950:PRINT"(HOME,DOWN,BRIGHT,SPACE)NEUE DATEI DEFINIEREN(SPACE,2DOWN)":IF X=-1 GOTO 540 <035>
520 IF X=-1 GOTO 540 <009>
530 INPUT"NAME DER DATEI":NA$:IF RIGHT$(NA$,1)="$*"GOTO 530 <196>
540 PRINT"DRUCKPARAMETER EINGEBEN!":POKE 180,81:POKE 129,1:POKE 1,53:POKE 58879,13 <038>
550 FOR I=0 TO 59:PRINT I+1"<LEFT> DRUCKZ EILE: "&E$="" <129>
560 INPUT E$ <182>
570 IF E$=""THEN PZ$(I)="0":GOTO 600 <067>
580 PZ$(I)=E$ <084>
590 IF E$="←"GOTO 610 <018>
600 NEXT I:PZ$(60)="←" <008>
610 POKE 129,32:POKE 1,55:INPUT"ALLES RICHTIG (J/N)":E$:UD=1 <161>
620 IF E$="N"GOTO 510 <164>
630 Y=0:DF=I:E$="M/"+LEFT$(NA$,14) <197>
640 IF X=-1 THEN X=0:RETURN <105>
650 GOSUB 3060 <050>
660 D$="0":D=0:GOSUB 1650:IF F1<>0 THEN Y=Y+1:IF Y<2 GOTO 650 <112>
670 GOTO 190 <234>
680 REM HILFSMENUE <145>
690 GOSUB 2980:PRINT"<2DOWN>-1- DATEI LADEN" <007>
700 PRINT"<DOWN>-2- DATEI PFLEGEN" <161>
710 PRINT"<DOWN>-3- DATEI ABSPEICHERN" <049>
720 PRINT"<DOWN>-4- SORTIERROUTINE" <161>
730 PRINT"<DOWN>-5- ADDITIONSROUTINE" <157>
740 PRINT"<DOWN>-6- DRUCKER-MENUE" <103>
750 PRINT"<DOWN>-↑- ZURUECK" <098>
760 GOSUB 3010:IF E$="↑"THEN RETURN <045>
770 IF E<1 OR E>6 GOTO 760 <244>
780 ON E GOSUB 870,1110,1640,2070,3090,2790 <137>
790 GOTO 690 <140>
800 REM TEST FLOPPY <013>
810 POKE 768,61:OPEN 1,8,15,"I":CLOSE 1 <119>
820 IF ST<>-128 THEN POKE 768,139:RETURN <079>
830 GOSUB 840:GOTO 810 <048>
840 PRINT"<CLR,5DOWN>":E=1:GOSUB 2970:PRINT"<5SPACE>PERIPHERIE EINSCHALTEN!!":PRINT T$ <087>
850 GOSUB 2970:GOSUB 3010:RETURN <234>
860 REM LOAD FILE <072>
870 GOSUB 2950:PRINT"(HOME,DOWN)"SPC(13)"DATEI LADEN " <130>
880 IF UD=1 AND D<>0 THEN PRINT"<4DOWN,SPACE>DATEI ABSPEICHERN?! (J/N)<DOWN>":PRINT T$:GOSUB 3010:IF E$<>"N"THEN RETURN <006>
890 CLR:GOSUB 3980 <155>
900 E$="":PRINT"(HOME,5DOWN)"A$<DOWN>MIT WELCHER DATEI WOLLEN SIE":INPUT"ARBEITEN";E$ <187>
910 IF RIGHT$(E$,1)="$*"GOTO 870 <178>
920 IF E$="↑"GOTO 1090 <088>
930 IF E$=""THEN 900 <139>
940 NA$=E$ <240>
950 UD=0 <100>
960 E$="M/"+LEFT$(NA$,14):SYS 827(E$)E$:E$=PEEK(141)*256+PEEK(140):POKE 139,PEEK(E) <079>
970 POKE 140,PEEK(E+1):POKE 141,PEEK(E+2):GOSUB 810:SYS 969:GOSUB 3710:IF F1 GOTO 870 <152>
980 OPEN 2,8,2,NA$+",S,R":POKE 180,81 <180>
990 INPUT#2,F:INPUT#2,A1%:INPUT#2,A2% <107>
1000 FOR I=1 TO F:FOR X=0 TO 2:INPUT#2,AZ(I,X):NEXT X,I <177>
1010 INPUT#2,DF:POKE 1,54:FOR I=0 TO DF:INPUT#2,PZ$(I):IF PZ$(I)=""←"GOTO 1030 <058>
1020 NEXT <012>
1030 INPUT#2,D <251>
1040 IF D=0 GOTO 1080 <243>
1050 POKE 129,255:FOR I=1 TO D <214>
1060 INPUT#2,FE$(I) <026>
1070 NEXT I:POKE 129,32 <114>
1080 POKE 1,55:GOSUB 3710:CLOSE 2:IF F1 GOTO 870 <083>
1090 E=1:GOTO 290 <098>
1100 REM ANZEIGE <145>
1110 IF UD=3 THEN RETURN <253>
1120 DS=0:IF D=0 GOTO 1340 <234>
1130 FL=7:L=0:IF DS<0 OR DS=D GOTO 3040 <193>
1140 PRINT"<CLR>":FOR I=1 TO 24:PRINT"<LIG.BLUE,40SPACE>";:NEXT <109>
1150 PRINT"<LIG.BLUE,39SPACE,HOME>"; <045>
1160 SYS 932:IF A1%<-1 THEN GOSUB 1320 <113>
1170 FOR I=1 TO F:GOSUB 1310:PRINT FE$(I+D S);:NEXT <139>
1180 IF L=2 THEN GOSUB 2680 <083>
1190 IF FL=8 GOTO 2040 <221>
1200 GOSUB 3010 <188>
1210 IF E$="←"AND DS<D THEN DS=DS+F:GOTO 130 <208>
1220 IF E$="←"AND DS>=0 THEN DS=DS-F:GOTO 130 <207>
1230 IF E$="{F1}"AND FL<>9 THEN Y=0:GOSUB 2460:GOTO 1130 <011>
1240 IF E$="{F3}"AND FL<>9 THEN GOSUB 2670:IF L=2 GOTO 1160 <167>
1250 IF E$="L"AND FL<>9 GOTO 1800 <233>
1260 IF E$="S"AND FL<>9 GOTO 1890 <154>
1270 IF E$="A"AND FL<>9 GOTO 1350 <129>
1280 IF E$="E"THEN DS=D:FL=0:GOTO 1340 <186>
1290 IF E$="↑"THEN RETURN <102>
1300 GOTO 1200 <002>
1310 POKE 214,AZ(I,0):POKE 211,AZ(I,1):SYS 58640:RETURN <161>
1320 POKE 214,A1%:POKE 211,A2%:SYS 58640:PRINT DS/F+1:RETURN <221>
1330 REM INPUT-ROUT. <243>
1340 PRINT"<CLR>":SYS 932 <126>
1350 OPEN 3,3:POKE 204,0:UD=1 <128>
1360 IF A1%<-1 THEN GOSUB 1320 <065>
1370 IF DS+F>1499 THEN GOSUB 2420:GOTO 1460 <194>
1380 FOR I=1 TO F:POKE 1,53 <162>
1390 Y=0:X=LEN(FE$(DS+I)):GOSUB 1310 <117>
1400 SYS 861 <070>

```

Listing »ProDat« (Fortsetzung)

```

1410 GOSUB 3010 <144>
1420 IF E$=CHR$(13) THEN PRINT E$;:GOTO 154 <237>
0 <237>
1430 IF E$="{F1}" AND I>1 THEN I=I-2:GOTO 1 <206>
570 <206>
1440 IF E$="{F3}" THEN X=0:Y=0:GOSUB 1310:P <160>
RINT LEFT$(A$,AZ(I,2));:GOSUB 1310:GO <018>
TO 1400 <045>
1450 IF E$="{F5}" GOTO 1860 <018>
1460 IF E$="{F7}" GOTO 1600 <045>
1470 IF E$="{DOWN}" AND Y<40 AND AZ(I,2)>Y+ <247>
39 THEN Y=Y+39:GOTO 1500 <247>
1480 IF E$="{UP}" AND Y>39 THEN Y=Y-41:GOTO <208>
1500 <208>
1490 IF E$="{RVSON}" OR E$="{RVOFF}" OR E$="{ <220>
DEL}" OR E$="{UP}" OR E$="{DOWN}" OR E$ <220>
=CHR$(148) OR E$="{HOME}" OR E$="{CLR}" <220>
GOTO 1410 <220>
1500 Y=Y+1:IF X<Y THEN X=Y <227>
1510 IF E$="{LEFT}" THEN Y=Y-2:IF Y<0 THEN <020>
Y=0:GOTO 1410 <020>
1520 PRINT E$;:IF Y=AZ(I,2) GOTO 1540 <051>
1530 GOTO 1400 <008>
1540 GOSUB 1310:POKE 180,X <120>
1550 FE$(DS+I)="" :POKE 213,80:POKE 129,255 <166>
:INPUT#3,FE$(DS+I):POKE 129,32:POKE 1 <166>
,55 <166>
1560 IF RIGHT$(FE$(DS+I),1)<>CHR$(32) THEN <105>
FE$(DS+I)=FE$(DS+I)+CHR$(154) <105>
NEXT I <130>
1570 NEXT I <130>
1580 IF FL GOTO 1620 <134>
1590 DS=DS+F:SYS 932:GOTO 1360 <129>
1600 IF FL GOTO 1620 <154>
1610 D=DS:DS=0 <210>
1620 POKE 1,55:POKE 207,0:POKE 204,1:PRINT <218>
#3,:CLOSE 3:X=0:GOTO 1130 <218>
1630 REM S P E I C H E R N <111>
1640 IF D=0 THEN RETURN <037>
1650 GOSUB 2950:PRINT "{HOME,DOWN}" SPC(10) " <148>
DATEI ABSPEICHERN " <148>
1660 PRINT "{3DOWN}" W$ <038>
1670 GOSUB 810:UD=0 <067>
1680 OPEN 2,8,2,"@:"+NA$+" ,S,W" <134>
1690 PRINT#2,F:PRINT#2,A1%:PRINT#2,A2% <140>
1700 FOR I=1 TO F:FOR X=0 TO 2:PRINT#2,AZ( <135>
I,X):NEXT X,I <135>
1710 PRINT#2,DF:FOR I=0 TO DF:PRINT#2,PZ$( <137>
I):IF PZ$(I)="" GOTO 1730 <137>
NEXT <206>
1720 NEXT <206>
1730 PRINT#2,D:IF D=0 GOTO 1770 <241>
1740 FOR I=1 TO D <196>
1750 PRINT#2,FE$(I) <032>
1760 NEXT I <064>
1770 CLOSE 2:GOSUB 3710 <085>
1780 RETURN <058>
1790 REM D S L O E S C H E N <165>
1800 POKE 214,24:POKE 211,0:SYS 58640:PRIN <178>
T "{RVSON} SIND SIE SICHER? (J/N) (RVOFF <154>
):":GOSUB 3010 <178>
1810 IF E$<>"J" AND E$<>"N" GOTO 1130 <154>
1820 D=D-F:POKE 857,F*3:IF DS=D THEN DS=DS <252>
-F:GOTO 1840 <252>
1830 SYS 932:FOR I=DS+1 TO D STEP F:SYS 82 <191>
7FE$(I)FE$(I+F):NEXT <191>
1840 GOTO 1130 <114>
1850 REM B Y T E S F R E I <250>
1860 F1=PEEK(214):F2=PEEK(211):POKE 214,24 <088>
:POKE 211,0:SYS 58640:SYS 861 <088>
1870 PRINT "[EREI: {6SPACE}]{7LEFT}" FRE(0);: <021>
POKE 214,F1:POKE 211,F2:SYS 58640:GOT <021>
O 1410 <021>
1880 REM S U C H E N <253>
1890 GOSUB 2950:PRINT "{HOME,DOWN}" SPC(12) " <226>
SUCHEN " <226>
1900 E$="":INPUT "{2DOWN} FELDNUMMER";E$:L=1 <036>
1910 POKE 180,80:V=VAL(E$):IF V>F THEN PRI <111>
NT "{2UP}";:GOTO 1900 <111>
1920 S$="":Z=1:Y=1:POKE 1,53:POKE 129,1:IN <251>
PUT "{2DOWN} SUCHWORT: ";S$:POKE 1,55:P <220>
OKE 129,32 <251>
1930 IF S$="" GOTO 1130 <220>
1940 LE=LEN(S$):P=F:IF V=0 THEN P=1:V=1 <116>
1950 PRINT "{CLR,2DOWN} ANZAHL FELDER: "D:PRI <094>
NT "{DOWN,6SPACE}-STES FELD ABGESUCHT" <094>
1960 FOR X=V TO D STEP P:PRINT "{UP}" X:E=LE <054>
N(FE$(X))-1:F1=E-L:IF LE>E GOTO 1970 <054>
1970 FOR Y=0 TO E:IF Y<=F1 THEN FOR Z=1 TO <138>
LE:IF MID$(S$,Z,1)=MID$(FE$(X),Y+Z,1 <138>
) THEN NEXT:GOTO 2030 <138>
1980 NEXT Y <158>
1990 NEXT X <160>
2000 IF FL=8 GOTO 1130 <010>
2010 PRINT "{DOWN}" S$ " NICHT GEFUNDEN!!"; <111>
2020 GOSUB 3010:GOTO 1160 <118>
2030 DS=F*INT((X-1)/F):FL=8:L=0:GOTO 1160 <189>
2040 GOSUB 3010:IF E$<>"J" GOTO 1130 <182>
2050 V=X+P:GOTO 1950 <134>
2060 REM S O R T <071>
2070 IF D<F*2 OR D=0 THEN RETURN <139>
2080 GOSUB 2950:PRINT "{HOME,DOWN}" SPC(10) " <050>
SORTIER-ROUTINE " <050>
2090 PRINT "{2DOWN}-1- ALLES ALPHABETISCH S <113>
ORTIEREN" <113>
2100 PRINT "{DOWN}-2- BESTIMMTE GRUPPEN HER <071>
AUSSORTIEREN {DOWN}":GOSUB 2970:POKE 8 <079>
57,F*3 <126>
2110 GOSUB 3010:IF E>2 OR E<1 GOTO 2110 <190>
2120 UD=1:IF E=2 GOTO 2230 <190>
2130 POKE 1,54 <190>
2140 PRINT "{DOWN} GEBEN SIE DIE FELDNUMMER <005>
EIN NACH DER":INPUT "SORTIERT WERDEN S <005>
OLL: ";E$ <005>
2150 Y=VAL(E$):POKE 1,55:IF Y<1 OR Y>F THE <033>
N POKE 214,11:SYS 58640:GOTO 2130 <033>
2160 PRINT "{2DOWN}":Z=D/F:PRINT "MAX. "Z" JUR <096>
CHGAENGE {2DOWN}":S=D-F:L=0 <145>
2170 PRINT "DURCHGANG: 1" <145>
2180 FOR E=1 TO Z:L=0 <136>
2190 FOR I=Y TO S STEP F:IF FE$(I)>FE$(I+F <151>
) THEN L=1:X=I-Y+1:SYS 827FE$(X+F)FE$( <151>
X) <151>
2200 NEXT I:IF L THEN PRINT,"{UP}" E:NEXT E <217>
2210 GOTO 1120 <198>
2220 REM S O R T B E S T . K R I T . <093>
2230 INPUT "{DOWN} NACH WELCHEM FELD SOLL SO <108>
RT. WERDEN";S$:S=VAL(S$):IF S>F GOTO <108>
2230 <108>
2240 POKE 180,80:POKE 129,1:POKE 1,53:INPU <248>
T "{DOWN} SORTIERKRITERIUM UNTERE GRENZ <248>
E: ";X$ <248>
2250 INPUT "{DOWN} SORTIERKRITERIUM OBERE GR <174>
ENZE: ";Y$:POKE 1,55:POKE 129,32 <174>
2260 PRINT "{DOWN}<1> NEUE DATEI OD. AUSGABE <152>
AUF <2> DRUCKER"; <152>
2270 INPUT E$:FL=VAL(E$):IF FL<1 OR FL>2 G <057>
OTO 2270 <057>
2280 IF FL=2 THEN GOSUB 2790:GOTO 2320 <131>
2290 INPUT "{DOWN} NAME DER DATEI";N$:IF N$= <225>
"↑" THEN RETURN <225>
2300 E$="M/" +N$:GOSUB 3060:IF F1 GOTO 2380 <230>
2310 NA$=N$ <152>
2320 IF E$="" THEN RETURN <116>
2330 DS=0:E=0:FOR L=S TO D STEP F:E$=LEFT$( <228>
FE$(L),LEN(Y$)):IF E$<X$ OR E$>Y$ GOTO <228>
2350 <228>
2340 DS=L-S:ON FL GOSUB 2400,2340,2470,273 <073>
0 <073>
2350 NEXT L <172>
2360 IF FL=1 THEN D=E:DS=D <238>
2370 IF FL<>1 THEN OPEN 4,4,7:PRINT#4:CLOS <187>
E 4:UD=0 <187>
2380 RETURN <152>
2390 REM E I N S O R T I E R E N <081>
2400 SYS 827FE$(E+1)FE$(DS+1):E=E+F:RETURN <234>
2410 REM D A T E I V O L L <246>
2420 POKE 207,0:POKE 204,1 <071>
2430 PRINT "{CLR,4DOWN}":E=1:GOSUB 2970:PRI <080>
NT SPC(10) "DATEI VOLL!":PRINT T$:GOS <196>
UB 2970 <121>
2440 GOSUB 3010:E$="{F7}":RETURN <215>
2450 REM D R U C K 1 <197>
2460 Y=-1:GOSUB 2860 <197>
2470 OPEN 4,4,7 <011>
2480 X=0:Y=0:IF E<>1 GOTO 2500 <066>
2490 FOR DS=0 TO D-F STEP F <066>
2500 FOR I=0 TO DF-1 <003>

```

Listing »ProDat« (Fortsetzung)

```

2510 IF PZ$(I)="0" GOTO 2610 <137>
2520 F1=LEN(PZ$(I)):FOR Z=1 TO F1 <036>
2530 E$=MID$(PZ$(I),Z,1) <178>
2540 IF E$<>"[" GOTO 2570 <038>
2550 PRINT#4,FE$(DS+VAL(MID$(PZ$(I),Z+1,2) <034>
);) <254>
2560 Z=Z+2:GOTO 2600
2570 IF E$="J" AND X=1 THEN X=0:E$=CHR$(15) <186>
:GOTO 2590
2580 IF E$="J" THEN E$=CHR$(14):X=1 <059>
2590 PRINT#4,E$; <069>
2600 NEXT Z <024>
2610 PRINT#4 <073>
2620 NEXT I <164>
2630 IF E=1 THEN NEXT DS <023>
2640 IF FL<>3 THEN PRINT#4 <171>
2650 CLOSE 4:RETURN <005>
2660 REM H A R D C O P Y <092>
2670 Y=-1:GOSUB 2860:OPEN 4,4,7:IF L=2 THE <103>
N RETURN
2680 PRINT"(HOME)";:OPEN 5,3:L=1024:POKE 1 <209>
,53
2690 FOR I=L TO L+39:POKE 1024,PEEK(I):PRI <094>
NT"(HOME)";:GET#5,E$:PRINT#4,E$;:NEXT <020>
:PRINT#4:L=I
2700 IF I<>2024 GOTO 2690
2710 X=0:CLOSE 4:CLOSE 5:POKE 1,55:RETURN <115>
2720 REM D R U C K Z <138>
2730 OPEN 4,4,7:IF E<>2 GOTO 2750 <011>
2740 FOR DS=0 TO D-F STEP F <062>
2750 FOR I=1 TO F:PRINT#4,FE$(I+DS):NEXT I <017>
:IF E=2 THEN NEXT DS
2760 IF FL<>4 THEN PRINT#4 <164>
2770 CLOSE 4:RETURN <125>
2780 REM D R U C K E R - M E N U E <175>
2790 IF D=0 THEN RETURN <171>
2800 GOSUB 2980:PRINT"(2DOWN)-1- GESAMTE D <149>
ATENBANK MIT DEFINIERTEN(7SPACE)FELDE <009>
RN DRUCKEN
2810 PRINT"(DOWN)-2- GESAMTE DATENBANK DRU <152>
CKEN":Y=0
2820 PRINT"(DOWN)-3- DATENSAETZE AUS SORTI <162>
ERROUTINE MIT(6SPACE)DEFINIERTEN FELD <158>
ERN DRUCKEN
2830 PRINT"(DOWN)-4- GANZE DATENSAETZE AUS <105>
SORTIERROUTINE(4SPACE)DRUCKEN <027>
2840 PRINT"(DOWN)-1- ZURUECK"
2850 GOSUB 3010:IF E$="↑" THEN RETURN
2860 L=1
2870 POKE 768,61:OPEN 4,4,7:PRINT#4:IF ST= <223>
-128 THEN GOSUB 840:L=2:GOTO 2870
2880 CLOSE 4:POKE 768,139:IF Y=-1 THEN RET <061>
URN
2890 IF L=2 GOTO 2790 <069>
2900 IF E<1 OR E>4 GOTO 2850 <246>
2910 IF E>2 THEN FL=E:RETURN <184>
2920 IF FL=2 GOTO 2850 <063>
2930 ON E GOTO 2470,2730 <192>
2940 REM <208>
2950 E=3 <121>
2960 PRINT"(CLR)"; <104>
2970 FOR I=1 TO E:PRINT"(BLACK)***** <147>
***** (LIG. BLU <136>
E)";:NEXT:RETURN
2980 GOSUB 2950:PRINT"(HOME,DOWN)"SPC(10) " <240>
PRODAT ":PRINT"(18DOWN)":GO S
2990 PRINT,"(2UP,SPACE)BY WOLFGANG BAHN(SP <044>
ACE,HOME,3DOWN)":RETURN
3000 REM G E T A L L G . <087>
3010 GET E$:E=VAL(E$):IF E$="" GOTO 3010 <028>
3020 RETURN <086>
3030 REM D A T E I Z U E N D E <090>
3040 FL=9:E=1:PRINT"(CLR,3DOWN)":GOSUB 297 <235>
0:PRINT SPC(10)"DATEI ZUENDE":GOSUB 2 <138>
970:GOTO 1200
3050 REM S A V E M A S K E <057>
3060 SYS 827(E$)E$:E=PEEK(141)*256+PEEK(14 <026>
0):POKE 139,PEEK(E):POKE 140,PEEK(E+1
3070 POKE 141,PEEK(E+2):POKE 176,0:POKE 17 <208>
7,204:GOSUB 810:SYS 995:GOSUB 3710:RE
3080 REM A D D I T I O N <198>
TURN
3090 IF D=0 OR D<F*2 THEN RETURN
3100 GOSUB 2950:PRINT"(HOME,DOWN)"SPC(11) " <037>
ADDITIONSROUTINE "
3110 INPUT"(3DOWN)WELCHE FELDER SOLLEN ADD <139>
IERT WERDEN";E$:E=VAL(E$):Y=0
3120 IF E<1 OR E>F GOTO 3110 <016>
3130 INPUT"(2DOWN)DATENSATZ FUER ERGEBNIS <012>
ANLEGEN";E$
3140 FOR I=E TO D STEP F:Y=Y+VAL(FE$(I)):N <231>
EXT I:Y=INT(Y*100+.5)/100
3150 IF E$<>"J" GOTO 3180 <102>
3160 FOR I=1 TO F:FE$(I+D)=CHR$(250):IF I= <105>
E THEN FE$(I+D)="(LEFT)" +STR$(Y)
3170 NEXT I:DS=D:D=D+I-1:UD=1:GOTO 1130 <227>
3180 PRINT"(2DOWN)GESAMTSUMME FELDER-NR. "E <071>
"="Y:PRINT"(2DOWN)"T$:GOSUB 3010:RET
URN
3190 REM D I R E C T O R Y <195>
3200 GOSUB 810:Y=0 <244>
3210 GOSUB 2950:PRINT"(HOME,DOWN,3RIGHT,SP <069>
ACE)DIRECTORY "
3220 IF Y=1 THEN RETURN <037>
3230 OPEN 3,8,0,"$":GET#3,E$,E$ <129>
3240 PRINT:GET#3,E$,E$,S$,E$ <177>
3250 E=ASC(S$+CHR$(0))+ASC(E$+CHR$(0))*256 <247>
3260 PRINT MID$(STR$(E)+"(4SPACE)",2,4); <057>
3270 GET E$:IF E$<>" " THEN CLOSE 3:RETURN <043>
3280 GET#3,E$:IF ST GOTO 3360 <247>
3290 IF E$<>CHR$(34) GOTO 3280 <206>
3300 GET#3,E$:IF E$<>CHR$(34) THEN PRINT E $ <209>
;:GOTO 3300
3310 GET#3,E$:IF E$="" GOTO 3310 <224>
3320 PRINT TAB(21); <195>
3330 PRINT E$;:GET#3,E$:IF E$<>" " GOTO 3330 <208>
3340 IF PEEK(214)>20 THEN PRINT:PRINT"(DOW <148>
N)"T$;:GOSUB 3010:Y=1:GOSUB 3210
3350 GOTO 3240 <164>
3360 PRINT"BLÖCKE FREI" <002>
3370 CLOSE 3 <095>
3380 GOSUB 3710 <194>
3390 PRINT:PRINT T$:GOSUB 3010 <023>
3400 RETURN <156>
3410 REM S C R A T C H <015>
3420 E=4:GOSUB 2960:PRINT"(HOME,DOWN)"SPC( <162>
3)"FILE LOESCHEN "
3430 PRINT SPC(21)"↑D = DIRECTORY " <182>
3440 INPUT"(2DOWN)FILENAME";E$ <128>
3450 IF E$="↑D" THEN GOSUB 3200:GOTO 3420 <077>
3460 IF E$="↑" THEN RETURN <240>
3470 GOSUB 810:OPEN 1,8,15,"S:" +E$+"M/" +E <097>
$:GOSUB 3730
3480 GOTO 3420 <006>
3490 REM V A L I D A T E <187>
3500 GOSUB 2950:PRINT"(HOME,DOWN)"SPC(13) " <080>
VALIDATE(SPACE,4DOWN)"
3510 PRINT W$ <050>
3520 GOSUB 810:OPEN 1,8,15,"V":GOSUB 3730 <209>
3530 RETURN <030>
3540 REM D I S K F O R M A T . <125>
3550 GOSUB 2950:PRINT"(HOME,DOWN)"SPC(10) " <099>
DISK FORMATIEREN "
3560 INPUT"(3DOWN)DISK-NAME";E$:IF E$="↑" T <160>
HEN RETURN
3570 INPUT"(DOWN)DISK-NUMMER";Y$:IF Y$="↑" <189>
THEN RETURN
3580 PRINT"(2DOWN)"W$:GOSUB 810:OPEN 1,8,1 <143>
5,"N:" +E$+"Y$:GOSUB 3730
3590 RETURN <092>
3600 REM R E N A M E <084>
3610 GOSUB 2950:PRINT"(HOME,DOWN)"SPC(9) " <040>
FILE UMBENENNEN "
3620 INPUT"(4DOWN)ALTER NAME";Y$ <175>
3630 IF Y$="↑" THEN RETURN <060>
3640 INPUT"(2DOWN)NEUER NAME";E$ <021>
3650 IF E$="↑" THEN RETURN <176>
3660 GOSUB 810:OPEN 1,8,15,"R:" +E$+"=" +Y$ <225>
3670 PRINT#1,"R:M/" +E$+"M/" +Y$:GOSUB 3730 <057>
3680 IF F1<>0 GOTO 3610 <060>
3690 GOTO 3200 <122>
3700 REM D I S K - E R R O R <039>
3710 CLOSE 1 <165>
3720 OPEN 1,8,15:POKE 1,55 <092>
3730 INPUT#1,F1,S$,F2,F3 <132>

```

Listing »ProDat« (Fortsetzung)

```

3740 CLOSE 1 <195>
3750 IF F1=0 THEN RETURN <248>
3760 PRINT:E=1:GOSUB 2970 <167>
3770 PRINT SPC(6)F1;S$;F2;F3 <058>
3780 PRINT T$ <042>
3790 GOSUB 2970:GOSUB 3010 <146>
3800 RETURN <046>
3810 REM M A P R O ' S <204>
3820 FOR Z=820 TO 1020:READ X:POKE Z,X:S=S
+X:NEXT <146>
3830 DATA 133,095,134,096,076,191,163,032, <192>
158,173,165,100,133,140,165,101,133
3840 DATA 141,032,158,173,160,000,177,140, <173>
133,139,177,100,145,140,165,139,145
3850 DATA 100,200,192,003,208,239,096,169, <111>
000,133,139,169,004,133,140,160,000
3860 DATA 177,139,056,233,128,048,002,145, <001>
139,192,255,200,208,242,165,140,201
3870 DATA 007,240,103,230,140,208,230,169, <202>
000,133,139,133,172,169,004,133,140
3880 DATA 169,204,133,173,160,000,177,139, <214>
145,172,192,255,200,208,247,165,140
3890 DATA 201,007,240,068,230,140,230,173, <002>
208,233,169,000,133,139,133,172,169
3900 DATA 204,133,140,169,004,133,173,160, <141>
000,177,139,145,172,192,255,200,208
3910 DATA 247,165,140,201,207,240,031,230, <158>
140,230,173,208,233,160,000,162,008
3920 DATA 032,186,255,166,140,164,141,165, <141>
139,032,189,255,162,000,160,204,169
3930 DATA 000,032,213,255,096,160,001,162, <017>
008,032,186,255,166,140,164,141,165
3940 DATA 139,032,189,255,162,255,160,207, <082>
169,176,032,216,255,096
3950 IF S<>29538 THEN PRINT"FEHLER IN DATA
S!":STOP <162>
3960 RETURN <208>
3970 REM I N I T I A L I S I E R E N <070>
3980 T$="{9SPACE}JASTE DRUECKEN!":DIM F$(
1500),PZ$(60),AZ(40,2):POKE 650,128:E
=1 <033>
3990 A$="{40SPACE}":A$=A$+A$:Y=0:L=0:UD=3 <040>
4000 W$="BITTE WARTEN...":RETURN <190>

```

© 64'er

Listing »ProDat« (Schluß)

Es erscheint dann das Hauptmenü:

HYPRA-COPY

- :C: Copy Files
- :S: Scratch Files
- :D: Directory
- :O: Order Disk

Die Bedienung ergibt sich damit eigentlich schon von selbst. Drückt man die Taste »O«, erscheint auf dem Bildschirm eine eckige Klammer mit einem blinkenden Cursor dahinter.

Nun kann man den Befehl eingeben, der zur Floppy geschickt werden soll. Reagiert die Floppy mit einer Fehlermeldung, so wird diese auf dem Bildschirm ausgegeben.

Bei Betätigung der Taste »D« erscheint das Directory. Der Ausdruck kann durch die CTRL-Taste angehalten werden. Durch Drücken der »S«-Taste gelangt man in den Scratch-Modus (löschen von Files). Um Verwechslungen mit dem Copy-Modus auszuschließen, wird erst einmal mit einer dicken, reversen Balkenüberschrift darauf aufmerksam gemacht, daß man sich tatsächlich im Scratch-Modus befindet. Danach erscheint das Directory, jedoch erscheint hinter jedem Filenamem ein »(Y/N)«; relative Files sowie Files, deren Länge gleich 0 oder größer als 232 Blöcke ist, werden bei dieser Befragung übergangen, denn diese Routine wird auch beim Kopieren gebraucht und Files mit diesen Eigenschaften kann Hypra-Copy nicht kopieren.

Files, die sich Hypra-Copy merken soll, müssen mit »Y« markiert werden. Ist man versehentlich im Scratch-Modus gelandet, so kann man diesen Modus mit der STOP-Taste verlassen (ansonsten dient die STOP-Taste dazu, Hypra-Copy zu beenden; man kann es dann aber wieder mit »RUN« starten).

So werden nach und nach alle Files der Directory durchgegangen. Will man nur einige wenige Files am Anfang einer ellenlangen Directory löschen (oder kopieren), so braucht man die restlichen Files nicht mehr mit »N« zu bearbeiten, es genügt ein Druck auf die »↑«-Taste, und man gelangt zum Ende der Directory. Hier wird dann gefragt, ob man nun auch sicher ist, daß diese Files gelöscht werden sollen. Beantwortet man diese Frage mit »N«, gelangt man zurück in das Hauptmenü, andernfalls werden die markierten Files gelöscht.

Damit man weiß, wie weit das Programm mit dem Löschen ist, wird immer der Name des Files ausgegeben, das gerade gelöscht wird. Am Ende des Löschvorgangs gelangt man automatisch wieder in das Hauptmenü.

Der Copy-Modus

So, nun zum Copy-Modus, in den man mit der »C«-Taste kommt. Zuerst werden wieder, genau wie beim Löschen, die Namen der zu kopierenden Files eingelesen.

Danach wird gefragt, ob man die Files einzeln oder gesammelt kopieren will und ob beim Speichern ein »VERIFY« durchgeführt werden soll. Verzichtet man auf dieses Verify, so wird etwas schneller kopiert. Um den Bedienungskomfort zu erhöhen, werden die Antworten auf diese Fragen, wie auch auf die »SURE?«-Frage (sure = Sind Sie sicher?) im Scratch-Modus und die »SAVE BUFFER AGAIN?«-Frage (Soll das gleiche noch einmal gespeichert werden?), nicht per GET (\$ffe4), sondern per BASIN (\$ffc4) eingelesen, wobei die wohl gebräuchlichere Antwort schon vorgegeben ist; man braucht also nur noch RETURN drücken; natürlich kann diese vorgegebene Antwort aber auch überschrieben werden.

Nun werden die Programme geladen. Ist die Summe der Blöcke der einzelnen Files kleiner als 233, können sogar alle auf einmal eingelesen werden; andernfalls wird ein erneuter Ladeanlauf nötig.

Sind die Files geladen, wird man aufgefordert, die Ziel-Disk einzulegen; außerdem wird noch die Anzahl der zu speichernden Blocks angegeben, und ein Untermenü erscheint.

Schnell kopiert mit Hypra-Copy

Hypra-Copy ist ein schnelles und komfortables Filecopy-Programm für den C 64. Das Kopieren wird um das Vier- bis Fünffache beschleunigt.

Trotz der aufwendigen Lade- und Speicherroutinen ist es gelungen, das Programm insgesamt recht kurz zu halten; Hypra-Copy belegt, gespeichert auf der Diskette, nur ganze 15 Blöcke. Somit ist der verfügbare Arbeitsspeicher sehr groß: Hypra-Copy kann sich bis zu 30 Filenamen »merken«, und in einem Ladegang können maximal 232 Blöcke eingelesen werden. Hat man das Programm (siehe Listing) mit dem MSE abgetippt und gespeichert, kann es später ganz normal mit »LOAD "HYPRA-COPY",8« geladen und mit »RUN« gestartet werden.

Es besteht jetzt noch die Möglichkeit, diverse Directories anzusehen oder Befehle zur Floppy zu schicken.

Man kann also in aller Ruhe eine passende Diskette aussuchen. Ist man gewillt fortzufahren, so drückt man einfach die Funktionstaste F7, dann beginnt nämlich das Speichern. Anschließend wird man gefragt, ob man eben diese Files noch einmal auf eine andere Diskette schreiben will. Beantwortet man diese Frage mit »Y«, beginnt der komplette SAVE-Vorgang von neuem. Andernfalls wird normal fortgefahren und, wie schon erwähnt, können erneute Ladeanläufe erforderlich werden.

Wurden alle Files kopiert, so wird in das Hauptmenü zurückgekehrt.

Nun zur Fehlerbehandlung: Mit »Fehlern« sind nicht Fehler gemeint, die in der Programmstruktur von Hypra-Copy liegen, sondern Fehler, die von der Floppy signalisiert werden; sei es, daß man einen unkorrekten Befehl an die Floppy schickt, man

beim Kopieren die falsche Diskette einlegt oder Write- oder Read-Errors auftreten.

Erste Hilfe bei Fehlern

In solchen oder ähnlichen Fällen besteht immer die Möglichkeit, den Vorgang, bei dem der Fehler aufgetreten ist, zu wiederholen beziehungsweise zu übergehen. Treten Fehler beim Öffnen eines Files auf, so kann man sogar nochmals Directories ansehen und Befehle an die Floppy schicken. War es nicht möglich, ein File korrekt zu laden und wurde es übergangen, so wird dies bei der Angabe der zu speichernden Blöcke, wie auch beim Speichern selbst, berücksichtigt. Können Files aus irgendeinem Grund nicht korrekt gespeichert werden, so kann man diese auch überspringen.

(Burkhard Graves/ev)

programm : hypra-copy 0801.16b0

```

0801 : 0b 08 c1 07 9e 32 30 36 0a
0809 : 31 00 00 00 a9 0b 0d 20 a4
0811 : d0 8d 21 d0 78 a9 c1 8d 01
0819 : 18 03 58 a2 00 86 05 8e 83
0821 : 3b 09 a7 40 85 9d a7 d5 eb
0829 : a0 08 20 1e ab 20 e1 09 ef
0831 : 4c 1c 08 a5 f7 18 69 13 4e
0839 : 85 f7 90 02 e6 f8 c9 ea 51
0841 : 60 a9 b0 85 f7 85 f9 a9 39
0849 : 16 85 f8 85 fa 60 52 12 1c
0851 : 20 21 20 53 43 52 41 54 e9
0859 : 43 48 20 46 a9 4c 45 53 44
0861 : 20 21 20 11 0d 00 0d 11 63
0869 : 43 4f 50 59 20 53 45 50 e5
0871 : 41 52 41 54 45 4c 59 20 13
0879 : 3f 20 4e 9d 00 0d 11 56 69
0881 : 45 52 49 46 59 20 3f 20 de
0889 : 4e 9d 00 0d 11 53 55 52 ed
0891 : 45 20 3f 20 59 9d 00 0d 57
0899 : 11 3a 54 3a 20 5d 09 c4
08a1 : 20 41 47 41 49 4e 0d 11 b9
08a9 : 3a 46 37 3a 20 54 4f 20 3e
08b1 : 43 4f 4e 54 49 4e 55 45 a1
08b9 : 11 0d 00 0d 11 53 41 56 50
08c1 : 45 20 42 55 46 46 45 52 a2
08c9 : 20 41 47 41 49 4e 20 3f 8a
08d1 : 20 4e 9d 00 93 05 11 1d 60
08d9 : 3e 20 48 59 50 52 41 2d 5c
08e1 : 43 4f 50 59 20 3c 0d 1d 5d
08e9 : 1d c4 c4 c4 c4 c4 c4 c4 41
08f1 : c4 c4 c4 c4 c4 11 11 0d 14
08f9 : 3a 43 3a 20 43 4f 50 59 0a
0901 : 20 46 49 4c 45 53 11 0d 6e
0909 : 3a 53 3a 20 53 43 52 41 9b
0911 : 54 43 48 20 46 49 4c 45 87
0919 : 53 0d 11 3a 44 3a 20 44 9e
0921 : 49 52 45 43 54 4f 52 59 09
0929 : 11 0d 3a 4f 3a 20 4f 52 c0
0931 : 44 45 52 20 44 49 53 4b 23
0939 : 11 0d 00 91 3a 46 37 3a 2a
0941 : 20 54 4f 20 43 4f 4e 54 f4
0949 : 49 4e 55 45 11 0d 00 12 55
0951 : 28 59 2f 4e 29 92 00 14 0b
0959 : 14 14 14 14 9d 3c 2d 00 6f
0961 : 0d 11 4c 49 53 54 20 46 18
0969 : 55 4c 4c 20 21 11 0d 00 ca
0971 : 0d 49 4e 53 45 52 54 20 99
0979 : 53 4f 55 52 43 45 20 44 7b
0981 : 49 53 4b 0d 00 0d 49 4e 13
0989 : 53 45 52 54 20 54 41 52 ec
0991 : 47 45 54 20 44 49 53 4b 06
0999 : 20 c3 20 00 20 42 4c 4f 87
09a1 : 43 4b 53 20 54 4f 20 53 4a
09a9 : 41 56 45 0d 00 0d 11 11 d7
09b1 : 44 49 53 4b 20 45 52 52 f2
09b9 : 4f 52 20 00 0d 11 11 4c 70
09c1 : 4f 41 44 20 45 52 52 4f 95
09c9 : 52 20 21 00 20 42 08 a0 e9
09d1 : 00 98 91 f7 c8 d0 fb e6 51
09d9 : f8 a6 f8 e0 20 d0 f3 60 9f
09e1 : a9 00 c6 20 e4 ff f0 c8
09e9 : fb a6 05 0 03 4c c9 0a 65
09f1 : c9 43 d0 4a 20 cd 09 a2 b3
09f9 : 00 86 03 20 42 08 20 34 4f
0a01 : 0b a2 00 bd b0 16 d0 01 16
0a09 : 60 a9 67 a0 08 20 1e ab 7d
0a11 : a9 00 85 c6 85 41 20 cf 77
0a19 : ff c9 59 d0 02 85 41 a9 12
0a21 : 4c 8d a9 15 a9 7e a0 08 62
0a29 : 20 1e ab a9 00 85 c6 20 00
0a31 : cf ff c9 59 d0 05 a9 2c d2
0a39 : 8d a9 15 4c 02 0d c9 53 c0
0a41 : f0 03 4c c9 0a a9 a4 a0 c1
0a49 : 13 20 2d 0b a9 4f a0 08 c1
0a51 : 20 1e ab a9 00 85 03 20 19
0a59 : cd 09 20 42 08 20 34 0b 64
0a61 : 20 42 08 a9 00 85 c6 a9 74
0a69 : 8c a0 08 20 1e ab 20 cf ab
0a71 : ff c9 59 f0 01 60 a9 0d 9d
0a79 : 20 d2 ff 20 d2 ff a9 08 ea
0a81 : 20 b1 ff a9 6f 20 93 ff f5
0a89 : a0 00 b1 f7 f0 35 a9 53 9b
0a91 : 20 a8 ff 20 d2 ff a9 3a 51
0a99 : 20 a8 ff 20 d2 ff c8 b1 c5
0aa1 : 47 c9 2c f0 0f a8 ff da
0aa9 : 20 d2 ff 4c 9f 0a a9 0d c7
0ab1 : 20 a8 ff 20 d2 ff a9 08 0d
0ab9 : 20 ae ff 20 c9 0c 20 34 1a
0ac1 : 08 d0 bb a9 08 4c ae ff f3
0ac9 : c9 88 d0 01 60 c9 54 d0 72
0ad1 : 01 60 c9 44 00 e0 a2 19 37
0ad9 : 86 03 20 34 0b a6 cb e0 46
0ae1 : 40 f0 fa 60 c9 4f d0 31 21
0ae9 : a9 5d 20 d2 ff a9 08 20 51
0af1 : b1 ff a9 6f 20 93 ff a9 ec
0af9 : 00 85 c6 20 cf ff 20 a8 40
0b01 : ff c9 0d d0 f6 20 d2 ff fe
0b09 : a9 08 20 ae ff 20 c9 0c d4
0b11 : b0 01 60 20 69 0c 90 0d 39
0b19 : 60 c9 03 d0 03 4c bb 0c d2
0b21 : 4c e5 09 a9 0d 20 d2 ff f4
0b29 : a0 27 a9 c3 20 d2 ff 88 e9
0b31 : d0 fa 60 20 47 0c a9 60 d7
0b39 : 85 b9 a9 01 a2 d0 a0 ff 58
0b41 : 20 8d 0c 90 03 4c 60 0c 69
0b49 : a5 ba 20 b4 ff a5 b9 20 3e
0b51 : 96 ff a9 00 85 90 a0 03 b7
0b59 : 20 4f 0c 85 02 20 4f 0c 4b
0b61 : 88 d0 f5 84 05 a6 02 20 2d
0b69 : cd bd a9 20 20 d2 ff 20 5c
0b71 : 4f 0c a4 03 d0 42 c9 22 da
0b79 : d0 3e 85 05 20 d2 ff a0 44
0b81 : 13 a9 00 85 06 91 f7 88 f7
0b89 : 10 fb c8 20 4f 0c 20 d2 48
0b91 : ff c9 22 f0 05 c8 91 f7 e8
0b99 : d0 f1 a9 2c c8 91 f7 20 8b
0ba1 : 4f 0c c9 50 f0 08 c9 53 90
0ba9 : f0 04 c9 55 d0 05 c8 91 34
0bb1 : f7 d0 05 20 d2 ff d0 97 96
0bb9 : aa f0 05 20 d2 ff d0 af f0
0bc1 : a5 02 f0 73 c9 e9 b0 6f 9f
0bc9 : a5 05 f0 6b a9 1d 85 d3 dc
0bd1 : a9 50 a0 a9 09 20 1e ab a9 e1
0bd9 : 00 85 c6 85 fb 20 e4 ff 52
0be1 : f0 fb c9 03 d0 08 20 47 fe
0be9 : 0c 68 68 4c 1c 08 c9 5e b3
0bf1 : d0 05 85 06 4c 24 0c c9 10
0bf9 : 59 d0 24 a9 9d 8d 5d 09 c6
0c01 : 20 19 0c a0 00 a5 02 91 1d
0c09 : f7 20 34 08 d0 29 a9 61 de
0c11 : a0 09 20 1e ab 4c 47 0c 54
0c19 : a9 58 a0 09 4c 1e ab c9 30
0c21 : 4e d0 ba a9 00 8d 5d 09 af
0c29 : 20 19 0c a0 00 98 91 f7 e8
0c31 : a5 06 f0 03 4c 47 0c a9 f8
0c39 : 0d 20 d2 ff ad 8d 02 d0 fc
0c41 : fb a0 02 4c 59 0b a9 08 3b
0c49 : 20 c3 ff 4c e7 ff 20 a5 1e
0c51 : ff a6 90 f0 12 68 68 20 2c
0c59 : 47 0c 20 c9 0c 90 08 20 8d
0c61 : 69 0c b0 03 4c 34 0b 60 b0
0c69 : a9 98 a0 08 20 1e ab a0 6a
0c71 : 00 84 c6 20 e4 ff aa a9 b5
0c79 : 00 e0 54 d0 02 18 60 e0 3d
0c81 : 88 d0 02 38 60 e0 03 d0 b4
0c89 : ea 4c bb 0c 20 bd ff 20 3a
0c91 : c1 f5 a9 08 85 b8 85 ba 62
0c99 : a9 08 20 b1 ff a9 6f 20 d0 e3
0ca1 : 93 ff a9 55 20 a8 ff a9 e0
0ca9 : 49 20 a8 ff a9 08 20 ae e5
0cb1 : ff 20 c0 ff b0 03 4c c9 d8
0cb9 : 0c 60 20 4f 0c 20 15 fd f8
0cc1 : a9 0d 20 d2 ff 4c 9d e3 f4
0cc9 : a9 08 20 b4 ff a9 6f 20 60
0cd1 : 96 ff 20 a5 ff aa 38 e9 2e
0cd9 : 30 85 8c f0 0a 8a 48 a9 76
0ce1 : ae a0 09 20 1e ab 68 a6 54
0ce9 : 8c f0 03 20 d2 ff 20 a5 ab
0cf1 : ff c9 d0 f2 a9 08 20 0f
0cf9 : ab ff 18 a5 8c f0 01 38 23
0d01 : 60 20 42 08 a9 00 85 8b cb
0d09 : a9 00 85 ae 85 06 a8 99 48
0d11 : 53 16 c8 c0 5a d0 f8 a9 1d
0d19 : 19 85 af a0 00 b1 f7 d0 04
0d21 : 01 60 20 24 0b a5 8b f0 cd
0d29 : 10 a9 71 a0 09 20 1e ab e0
0d31 : a9 00 85 c6 20 e4 ff f0 1f
0d39 : fb a9 e8 85 40 85 fd e6 ea
0d41 : 8b a0 00 b1 f7 85 fe f0 dc
0d49 : 72 c5 40 90 06 20 bc 0d 2e
0d51 : 4c 09 0d a2 00 86 b9 a5 20
0d59 : 40 38 e5 fe 85 40 a5 fd fb
0d61 : 38 e5 fe 85 fd a5 ae 85 cf
0d69 : 69 a5 af 85 6a a5 69 85 c6
0d71 : ae a5 6a 85 af 20 5c 0e c7
0d79 : b0 11 20 cc 0e 90 17 a9 69
0d81 : bd a0 09 20 1e ab 20 69 67
0d89 : 0c 90 e2 a5 fd 18 65 fe 1f
0d91 : 85 fd a9 00 f0 02 a9 ea 7b
0d99 : 85 ff a4 06 a5 af 99 53 ed
0da1 : 16 c8 a5 ae 99 53 16 c8 79
0da9 : a5 ff 99 53 16 c8 84 06 e4
0db1 : 20 34 08 f0 06 a5 41 d0 40
0db9 : 94 f0 86 a9 00 85 c1 85 da
0dc1 : 06 a9 19 85 c2 a5 f9 85 df
0dc9 : f7 a5 fa 85 f8 20 24 0b 39
0dd1 : a9 86 a0 09 20 1e ab a9 c9
0dd9 : e8 38 e5 fd aa a9 00 20 4f
0de1 : cd bd a9 0d a0 09 20 1e ba
0de9 : ab a9 0d 8d 3b 09 85 05 7a
0df1 : a9 1b a0 09 20 1e ab 20 53
0df9 : e1 09 c9 88 d0 0e a4 06 ed
0e01 : b9 53 16 f0 38 85 af c8 07
0e09 : b9 53 16 85 ae c8 b9 53 61

```

Listing »Hypra-Copy«. Beachten Sie bitte die Eingabehinweise auf Seite 8.

```

0e11 : 16 c8 84 06 aa f0 19 a9 57
0e19 : 01 85 b9 20 5c 0e b0 10 68
0e21 : 20 8d 12 20 47 0c 20 c9 79
0e29 : 0c 90 05 20 69 0c 90 eb d4
0e31 : a5 ae 85 c1 a5 af 85 c2 3a
0e39 : 20 34 08 d0 c1 a9 bc a0 2d
0e41 : 08 20 1e ab a9 00 85 c6 95
0e49 : 20 cf ff c9 59 d0 03 4c 4b
0e51 : bc 0d a5 f7 85 f9 a5 f8 ad
0e59 : 85 fa 60 a2 05 a5 b9 2f 7e
0e61 : 01 d0 07 a2 20 a0 49 4c a5
0e69 : 6d 0e a0 51 20 2f f1 8a 88
0e71 : 20 d2 ff a6 f7 a4 f8 e8 29
0e79 : d0 01 c8 a9 12 20 8d 0c a2
0e89 : b0 0c a9 0b a0 4f 99 00 ee
0e89 : d8 88 10 fa 18 60 20 47 9c
0e91 : 0c a5 b9 85 fe a9 00 8d e7
0e99 : 3b b9 a9 0d 20 d2 ff 85 08
0ea1 : 05 a9 1b a0 09 20 1e ab b7
0ea9 : a9 91 20 d2 ff 20 d2 ff c9
0eb1 : a9 98 a0 08 20 1e ab 20 b1
0eb9 : e1 09 c9 54 d0 07 a5 fe f6
0ec1 : 85 b9 a9 0c 0e c9 88 d0 b4
0ec9 : d1 38 60 a5 ba 20 b4 ff 03
0ed1 : a5 b9 20 96 ff a0 00 20 73
0ed9 : a5 ff a2 34 86 01 91 ae c1
0ee1 : a2 37 86 01 c8 0c 02 d0 1d
0ee9 : ee 20 47 0c a5 ae 18 69 3e
0ef1 : 02 85 ae 90 09 e6 af a5 45
0ef9 : a7 c9 08 f0 17 00 a9 01 bf
0f01 : 85 a7 a9 00 85 90 a5 a7 87
0f09 : 20 0c ed a9 6f 20 b9 ed 9b
0f11 : a5 90 1e 83 e6 a7 a5 a7 10
0f19 : c9 10 d0 e6 a9 8d a2 10 ad
0f21 : 85 a7 86 a8 a9 00 a2 03 5c
0f29 : 85 a9 86 aa a9 08 20 0c ed
0f31 : ed a9 6f 20 b9 ed a9 4d 1f
0f39 : 20 dd ed a9 2d 20 dd ed 1f
0f41 : a9 57 20 dd ed a0 00 a5 89
0f49 : a9 20 dd ed a5 aa 20 dd 23
0f51 : ed a9 1e 20 dd ed b1 a7 02
0f59 : 20 dd ed c8 c0 1e 90 f6 29
0f61 : 20 fe ed 18 a5 a7 69 1e f8
0f69 : 85 a7 90 03 e6 a8 18 a5 a6
0f71 : a9 a6 aa 69 1e 85 a9 90 1b
0f79 : 02 e6 aa e0 05 90 ad c9 d4
0f81 : 00 90 a9 a9 08 20 0c ed f6
0f89 : a9 6f 20 b9 ed a9 4d 20 cb
0f91 : dd ed a9 2d 20 dd ed a9 71
0f99 : 45 20 dd ed a9 8b 20 dd 57
0fa1 : ed a9 04 20 dd ed a9 0b 72
0fa9 : 8d 11 d0 20 fb ed ea ea a8
0fb1 : 78 a9 01 85 a7 a0 ff 20 af
0fb9 : 37 10 c0 ff f0 44 a2 34 4c
0fc1 : 86 01 a2 02 a5 a7 f0 02 10
0fc9 : a2 04 ad 2e 03 d0 07 ee 4f
0fd1 : 2f 03 ad 2f 03 2c a9 00 0b
0fd9 : 85 a8 bd 2e 03 91 ae e6 2d
0fe1 : ae d0 02 e6 af e8 e4 a8 7c
0fe9 : d0 f0 a2 00 86 a7 a2 37 79
0ff1 : 86 01 ad 2e 03 d0 be a9 0b
0ff9 : 1b 8d 11 d0 a9 40 85 90 0d
1001 : 18 60 a9 1b 8d 11 d0 ea 92
1009 : a9 1d 38 60 a9 0b 8d 00 84
1011 : dd 2c 00 dd 10 fb a9 03 4e
1019 : 8d 00 dd a2 05 ca ea d0 66
1021 : fc a2 04 ad 00 dd 2a 2a 11
1029 : 66 b0 6a 66 b0 ea ca d0 7e
1031 : f2 a5 b0 49 ff 60 20 0d e9
1039 : 10 c9 ff f0 f8 a0 00 a9 34
1041 : 0b 8d 00 dd 2c 00 dd 10 29
1049 : fb a9 03 8d 00 dd a2 07 13
1051 : ca d0 fd ad 00 dd 2a 2a a4
1059 : 66 b0 6a 66 b0 ea ea ad e8
1061 : 00 dd 2a 2a 66 b0 6a 66 82
1069 : b0 ea ea ad 00 dd 2a 2a eb
1071 : 66 b0 6a 66 b0 ea ea ad 00
1079 : 00 dd 2a 2a 66 b0 6a 66 9a
1081 : b0 a5 b0 49 ff 99 2e 03 e5
1089 : c8 d0 b4 60 a5 00 29 06 fe
1091 : c9 02 f0 03 4c 9e fd ea 7f
1099 : a9 05 85 09 a2 5a 86 4b f5
10a1 : a2 00 a9 52 85 24 20 56 9f
10a9 : f5 50 fe b8 ad 01 1c c5 7c
10b1 : 24 f0 09 c6 4b d0 ef a9 b7
10b9 : 0a 4c 69 f9 50 fe b8 ad be
10c1 : 01 1c 95 25 e8 e0 07 d0 2e
10c9 : f3 20 97 f4 a5 16 45 17 9f
10d1 : 45 18 45 19 45 1a f0 07 8e
10d9 : c6 09 d0 c0 4c 1e f4 a5 45
10e1 : 18 c5 06 f0 03 4c 0b f4 24
10e9 : 85 22 a9 06 85 31 4c 3c 36
10f1 : 04 a5 12 a6 13 85 16 86 e4
10f9 : 17 a5 06 85 18 a5 07 85 eb
1101 : 19 a9 00 45 16 45 17 45 0a
1109 : 18 45 19 85 1a 20 34 f9 22
1111 : a2 5a 20 56 f5 a0 00 50 b8
1119 : fe b8 ad 01 1c d9 24 00 20
1121 : f0 06 ca d0 ed 4c 51 f5 53
1129 : c8 c0 08 d0 ea 20 56 f5 62
1131 : 50 fe b8 ad 01 1c 91 30 7c
1139 : c8 d0 f5 a0 ba 50 fe b8 96
1141 : ad 01 1c 99 00 01 c8 d0 76
1149 : f4 20 e0 f8 a5 38 c5 47 66
1151 : f0 03 4c f6 f4 20 e9 f5 99
1159 : c5 3a f0 03 4c 02 f5 a0 c6
1161 : 00 a9 55 20 52 04 b9 00 bb
1169 : 06 85 77 2c 00 18 10 fb 8e
1171 : a9 10 8d 00 18 2c 00 18 99
1179 : 30 fb a2 00 8a 66 77 2a 5e
1181 : 2a 66 77 2a 2a 8d 00 18 41
1189 : 8a 66 77 2a 2a 66 77 2a 72
1191 : 2a 8d 00 18 8a 66 77 2a 93
1199 : 2a 66 77 2a 2a 8d 00 18 59
11a1 : 8a 66 77 2a 2a 66 77 2a 8a
11a9 : 2a 8d 00 18 a2 02 ca d0 a4
11b1 : fd a9 0f 8d 00 18 c8 d0 7e
11b9 : ad ea ea ea ea ea ea 7b
11c1 : ad 00 1c 09 08 8d 00 1c bc
11c9 : ad 00 06 d0 03 4c 9e fd 1b
11d1 : c5 18 d0 f9 85 06 ad 01 57
11d9 : 06 85 07 4c 65 03 85 77 61
11e1 : 2c 00 18 10 fb a9 10 8d 7e
11e9 : 00 18 2c 00 18 30 fb a2 39
11f1 : 07 a9 00 66 77 2a 2a 66 d5
11f9 : 74 2a 2a 8d 00 18 ca d0 4f
1201 : f0 ea ea ea ea ea ea a9 83
1209 : 0f 8d 00 18 60 06 85 00 01
1211 : 58 a5 00 30 fc 78 60 78 48
1219 : ea ea ea ea ea ea 18 5d
1221 : 8d 00 06 85 06 a5 19 8d ee
1229 : 01 06 85 07 a9 04 85 78 31
1231 : a9 e2 20 82 04 c9 02 90 5b
1239 : 33 a0 00 84 78 a4 78 b9 4f
1241 : db fe f0 12 58 20 76 d6 28
1249 : 78 a9 e2 20 82 04 c9 02 c6
1251 : 90 1a e6 78 d0 e7 a9 c0 2b
1259 : 20 82 04 a9 e2 20 82 04 32
1261 : c9 02 90 08 a9 ff 20 52 10
1269 : 04 4c 22 ab ad 0c 0e f0 6e
1271 : f8 c5 18 f0 c4 ad 00 06 36
1279 : 85 06 ad 01 06 85 07 4c ce
1281 : a0 04 ea ea ea ea 00 c4
1289 : b9 19 f4 99 ad 00 dd 8d ac
1291 : 1f 01 a5 ba 20 0c ed a5 57
1299 : b9 20 b9 ed a9 00 85 90 60
12a1 : 20 dd ed 20 dd ed 20 fe fb
12a9 : ed a5 90 f0 02 18 60 a9 61
12b1 : 1e a2 14 85 ac 86 ad a9 df
12b9 : 46 a2 01 85 14 86 15 a2 50
12c1 : 05 a5 ba 20 0c ed a9 6f 01
12c9 : 20 b9 ed a0 00 b9 16 14 a4
12d1 : 20 dd ed c8 c0 03 d0 f5 c8
12d9 : a5 14 20 dd ed a5 15 20 ed
12e1 : dd ed a9 1e 20 dd ed a0 cd
12e9 : 00 b1 ac 20 dd ed c8 c0 e3
12f1 : 1e 90 f6 20 fe ed a5 ac 68
12f9 : 18 69 1e 85 ac 90 02 e6 23
1301 : ad a5 14 18 69 1e 85 14 4f
1309 : 90 02 e6 15 ca d0 b2 a5 40
1311 : ba 20 0c ed a9 6f 20 b9 a6
1319 : ed a0 00 b9 19 14 20 dd fc
1321 : ed c8 c0 05 d0 f5 a0 00 82
1329 : 20 fe ed 78 a2 00 8e 00 b7
1331 : dd ad 11 d0 29 ef 8d 11 ae
1339 : d0 bd b2 14 9d 2e 03 e8 40
1341 : d0 f7 ad 00 dd 30 fb 20 08
1349 : 02 14 a2 00 bd b2 15 9d ff
1351 : 2e 03 e8 d0 f7 20 02 14 06
1359 : 20 8e fb a2 01 20 d1 fc 66
1361 : b0 1f a0 34 84 01 a0 00 22
1369 : b1 ac a0 37 84 01 9d 2f a4
1371 : 03 e8 20 db fc e0 ff 90 64
1379 : e4 20 d1 fc b0 03 a9 ff 4b
1381 : 2c a9 00 48 8d 2e 03 8e fe
1389 : 2f 03 20 02 14 a2 01 68 ad
1391 : d0 cb 18 a9 00 48 ad 11 9d
1399 : d0 09 10 8d 11 d0 ad 1f 30
13a1 : 01 8d 00 dd 78 20 15 fd fd
13a9 : 68 60 b9 2e 03 85 95 a2 6d
13b1 : 00 2c 00 dd 50 fb 2c 00 19
13b9 : dd 10 05 e8 d0 f8 f0 d2 3b
13c1 : a9 10 8d 00 dd aa 66 95 ce
13c9 : 6a 66 95 6a 4a 8a 8d 00 46
13d1 : dd 8a 66 95 6a 66 95 6a 45
13d9 : 4a 4a 8d 00 dd 8a 66 95 a3
13e1 : 6a 66 95 6a 4a 4a 8d 00 5e
13e9 : dd 8a 66 95 6a 66 95 6a 5d
13f1 : 4a 4a 8d 00 dd 6a ee 9a ea 7a
13f9 : a9 00 8d 00 dd c8 d0 aa c2
1401 : 60 a0 00 98 59 2e 03 c8 69
1409 : d0 fa 85 14 20 ab 13 88 f7
1411 : a5 14 4c ae 13 4d 2d 57 a8
1419 : 4d 2d 45 b7 01 a0 00 84 63
1421 : 11 a9 02 8d 00 18 a9 04 a9
1429 : 2c 00 18 f0 fb a9 00 8d a2
1431 : 00 18 a2 03 ca d0 fd a2 b7
1439 : 0a ad 00 18 4a 6a 4a 66 0b
1441 : 85 0a 0a 66 85 ad 00 18 11
1449 : 4a 6a 4a 66 85 0a 6a 66 c5
1451 : 85 ad 00 18 4a 6a 4a 66 9e
1459 : 85 0a 0a 66 85 ad 00 18 29
1461 : 8e 00 18 4a 6a 4a 66 85 dc
1469 : 0a 0a 66 85 a5 85 91 30 f0 fd
1471 : 45 11 85 11 c8 d0 aa 88 91
1479 : 60 85 31 20 46 01 b1 30 80 80
1481 : 48 20 4a 01 68 91 30 a5 ab
1489 : 11 f0 ed 4c 43 e8 78 a9 c8
1491 : 0a 8d 00 18 a2 00 88 d0 53
1499 : fd ca d0 fa a9 0a 85 69 63
14a1 : a9 00 85 30 a9 03 20 a2 2a
14a9 : 01 a9 04 20 a2 01 4c 3d 62
14b1 : 04 a5 00 a2 01 86 00 29 73
14b9 : 02 f0 10 a6 98 86 32 20 d3
14c1 : 2e 04 20 af 03 20 03 04 35
14c9 : 4c 69 f9 a2 08 20 35 03 f9
14d1 : a2 0a 20 35 03 a2 08 20 cd
14d9 : f5 03 a2 0a 20 f5 03 a5 43
14e1 : 8c 30 e8 4c 69 f9 86 32 2e
14e9 : 86 98 a5 8c 10 61 b5 83 a0
14f1 : d0 5d bd 31 04 20 a2 01 d3
14f9 : c8 a6 32 a9 80 95 83 a5 e4
1501 : 80 95 00 a5 81 95 01 b1 2d
1509 : 30 f0 1c 20 21 f1 a6 82 fe
1511 : f6 b5 d0 02 f6 bb a0 00 26
1519 : a5 80 91 30 c8 a5 81 91 4c
1521 : 30 a5 80 c5 22 f0 02 84 b7
1529 : 8c a6 32 a9 80 85 30 85 c2
1531 : 33 85 2e 85 36 85 0c 85 2e
1539 : 50 a9 bb 85 34 bd 31 04 fb
1541 : 85 2f 20 e9 f5 85 3a bd 93
1549 : 32 04 20 a3 f7 a6 32 b5 e3
1551 : 83 f0 52 20 2e 04 ad 00 9f
1559 : 1c 29 10 d0 03 4c 81 f5 ac
1561 : 20 10 f5 a2 09 50 fe b8 db
1569 : ca d0 fa a9 ff 8d 03 1c 40
1571 : ad 0c 1c 29 1f 09 c0 8d a9
1579 : 0c 1c a9 ff a2 05 8d 01 88
1581 : 1c b8 50 fe b8 ca 00 fa 08
1589 : a0 bb b1 0c 50 fe b8 8d f0
1591 : 01 1c c8 d0 f5 b1 30 50 3b
1599 : fe b8 8d 01 1c c8 d0 f5 ae
15a1 : 50 fe 4c 00 fe 60 86 32 f5
15a9 : 2c 24 04 86 98 b5 83 f0 e0
15b1 : f4 20 2e 04 20 0a f5 a0 2d
15b9 : bb b1 0c 50 fe b8 4d 01 47
15c1 : 1c d0 19 c8 1f c8 b1 30 78
15c9 : 50 fe b8 4d 01 1c d0 0c bc
15d1 : c8 c0 fd d0 f1 a6 32 a9 03
15d9 : 00 95 83 60 4c c5 f6 bd db
15e1 : 31 04 85 31 bd 32 04 85 24
15e9 : 0d 00 05 01 06 04 58 20 aa
15f1 : 19 f1 a9 84 d5 a7 f0 05 66
15f9 : 95 a7 20 42 d0 a9 40 8d 29
1601 : f9 02 a9 01 85 83 20 07 89
1609 : d1 90 03 4c f8 cf 20 3e 78
1611 : de f6 b5 a9 00 85 8b 85 72
1619 : 8d a9 80 85 8c a5 80 85 4f
1621 : 06 a9 e0 85 00 a5 00 30 72
1629 : fc f0 24 c9 01 f0 ea a5 6e
1631 : 18 85 06 a5 19 85 07 a2 61
1639 : 00 a9 b0 20 7d d5 20 99 78
1641 : d5 a9 e2 20 7d d5 20 99 e2
1649 : d5 a5 8c 20 c0 ff ce 4c 56
1651 : 23 db 2a 30 31 2a 30 32 7c
1659 : 2a 30 33 2a 30 34 2a 30 5b
1661 : 35 2a 30 36 2a 30 37 2a d4
1669 : 30 38 2a 30 39 2a 31 30 50
1671 : 2a 31 31 2a 31 32 2a 31 75
1679 : 33 2a 31 34 2a 31 35 2a ea
1681 : 31 36 2a 31 37 2a 31 38 78
1689 : 2a 31 39 2a 32 30 2a 32 91
1691 : 31 2a 32 32 2a 32 33 2a 00
1699 : 32 34 2a 32 35 2a 32 36 90
16a1 : 2a 32 37 2a 32 38 2a 32 e9
16a9 : 39 2a 33 30 00 00 00 ff ca

```

Listing »Hypra-Copy« (Schluß)

Programme leichter laden

Dieses Programm vereinfacht das Laden von Programmen durch direkte Auswahl aus dem Directory.

Programme werden häufig nach dem Listen des Directorys durch Voranstellen eines »LOAD« und Ergänzung durch »8:« oder »8,1« geladen. Oftmals vergißt man dabei allerdings das Eintippen des Doppelpunktes oder man macht andere kleine Fehler, worauf sich der Computer mit einem »SYNTAX ERROR« meldet.

Mit »Directory-Lader« (Listing) kann das Directory einer Diskette nun so manipuliert werden, daß wahlweise ein »8:« oder ein »8,1« an den Filenamen angehängt wird. Das vereinfacht den Ladevorgang um einiges: Nur noch das Directory listen, mit dem Cursor in die gewünschte Zeile fahren, »LOAD« eingeben und schon geht's los.

So funktioniert der »Directory-Lader«

Jedes Programm auf einer Diskette ist in einem Directoryblock verzeichnet. Dort belegt der Name des Programmes 16 Byte. Ist der Name kürzer als 16 Byte, wird er durch ein »SHIFT Space« (CHR\$ 160) ergänzt. Dadurch wird auch das Ende eines Namens vom DOS erkannt.

Beim Listen des Directorys werden nun alle 16 Byte des Namens auf dem Bildschirm angezeigt. Das zu lösende Problem besteht also darin, den gewünschten Zusatz ebenfalls hinter dem Programm anzuzeigen.

Dazu wird zuerst der erste Block des Directorys eingelesen. Der Bufferpointer wird auf den ersten Namen des Blocks gesetzt und der Name ausgelesen. Hierbei werden alle 16 Byte des Namens verwendet. Durch Betätigen entsprechender Tasten wird nun der Name wie folgt bearbeitet:

- @ Löschen des Zusatzes
- ↑ Überspringen des Namens ohne Änderung
- M Schreiben des Zusatzes »8,1«
- B Schreiben des Zusatzes »8:«

Beim Schreiben des Zusatzes wird der Name des Programms von hinten durch den Zusatz ergänzt. Dabei werden Programmnamen, die länger als 11 Zeichen bei Maschinenprogrammen oder länger als 12 Zeichen bei Basic-Programmen sind, entsprechend verkürzt.

In dieser Weise werden sämtliche Namen eines Directoryblocks bearbeitet.

Da der Bufferpointer jedesmal weitersetzt wird, kann es vorkommen, daß auch bereits gelöschte Programme bearbeitet werden. Das hat jedoch keinen weiteren Einfluß.

Nach der Bearbeitung eines Blocks wird dieser zurückgeschrieben, und das Programm fährt mit dem nächsten Block fort.

Tabelle 1 zeigt die verwendeten Variablen, Tabelle 2 die einzelnen Routinen des Programms.

Während der Bearbeitung wird ständig der nächste zu bearbeitende Block angezeigt. Erscheinen die Werte 0 und 255, so ist der letzte Block erreicht. Man hat jetzt eine Diskette mit modifiziertem Directory und damit einfach zu ladenden Programmen vorliegen (Bild 1).

(Holger Weihe/ev)

```
0 " test direktory " db 2a
8 "testdirektor",8: prg
8 "1234567890123456" prg
8 "loader" ,8: prg
8 "programm" ,8: prg
8 "basic" ,8: prg
8 "maschprog" ,8,1 prg
616 blocks free.
```

Bild 1. Beispiel eines modifizierten Directorys

| | |
|----------|---|
| O | Prüfsumme für Katalog-Routine |
| K | Ausgewählte Taste bei Menü |
| T, S | Track und Sektor für ersten Block |
| P, Q | Zwischenspeicher für Track, Sektor |
| Z | Position des ersten Namens eines Blocks |
| X\$ | Warten auf Tastendruck |
| T\$, S\$ | Track, Sektor des nächsten Blocks |
| N\$ | Programmname |
| D\$ | Modifizierter Programmname |

Tabelle 1. Variablenliste zu »Directory-Lader«

| Zeile | Funktion |
|-----------|---------------------------------------|
| 100-260 | Katalog-Routine |
| 270-450 | Titelbild |
| 470-500 | Auswertung der Eingabe |
| 510-640 | Anzeigen des Direktories |
| 650-990 | Einlesen des Blocks |
| | Aufbau des Untermenüs |
| | Auswertung der Eingabe |
| 1000-1140 | Schreiben des Zusatzes |
| | »8:« bei Basic-Programm |
| 1150-1280 | Schreiben des Zusatzes |
| | »8,1« bei Maschinen-Programmen |
| 1290-1390 | Keine Änderung des Programmnamens |
| 1400-1500 | Löschen des Zusatzes |
| 1510-1610 | Rückschreiben des eingelesenen Blocks |
| 1620-1740 | Auswahl und Ende des Programmes |

Tabelle 2. Die wichtigsten Routinen

```
10 REM ***** <119>
20 REM ** DIREKTORY LOADER <104>
30 REM ** <079>
40 REM ** BY HOLGER WEIHE <031>
50 REM ** AM SILBERBERG 1 <043>
60 REM ** 3344 BOERSSUM <050>
70 REM ** <119>
80 REM ** TEL. 05334/6589 <165>
90 REM ** <139>
100 REM ***** <209>
110 REM ↑ = PFEIL NACH OBEN <228>
120 REM @ = KLAMMERAFFE <068>
130 REM ***** <239>
140 REM ** DIREKTORY ROUTINE <217>
150 REM ***** <003>
160 REM <222>
170 REM <232>
180 DATA 169,36,133,251,169,251,133,187,16
9,0,133,188,169,1,133,183,169,8 <195>
190 DATA 133,186,169,96,133,185,32,213,243
,165,186,32,180,255,165,185,32,150 <216>
200 DATA 255,169,0,133,144,160,3,132,251,3
2,165,255,133,252,164,144,208,47 <072>
210 DATA 32,165,255,164,144,208,40,164,251
,136,208,233,166,252,32,205,189 <029>
220 DATA 169,32,32,210,255,32,165,255,166,
144,208,18,170,240,6,32,210,255 <200>
230 DATA 76,76,192,169,13,32,210,255,160,2
,208,198,32,66,246,96,234 <032>
240 FOR I=49152 TO 49152+105:READ D:POKE I
,D:0=0:D:NEXT <000>
250 IF OK>15577 THEN POKE 199,1:PRINT" <2SP
ACE>FEHLER IN DATAS <2SPACE>":STOP <119>
```

```

260 REM <068>
270 REM <078>
280 REM ***** <017>
290 REM ** TITEL <056>
300 REM ***** <037>
310 REM <118>
320 REM <128>
330 POKE 53280,0:POKE 53281,0:PRINT CHR$(1 <013>
52) <010>
340 PRINT CHR$(14):PRINT CHR$(147) <010>
350 PRINT "{5SPACE}0000000000000000000000 <225>
00000" <154>
360 PRINT "{5SPACE}0{27SPACE}0" <154>
370 PRINT "{5SPACE}0{5SPACE}0DIREKTORY LOADE <072>
R{6SPACE}0" <174>
380 PRINT "{5SPACE}0{27SPACE}0" <174>
390 PRINT "{5SPACE}0000000000000000000000 <027>
00000" <248>
400 PRINT <248>
410 PRINT <019>
420 PRINT "{5SPACE}";CHR$(18);" BITTE AUSWA <060>
EHLEN ";CHR$(146):PRINT:PRINT <060>
430 PRINT "{8SPACE}";CHR$(18);"1";CHR$(146) <066>
;" {3SPACE}DIREKTORY ANZEIGEN":PRINT <066>
440 PRINT "{8SPACE}";CHR$(18);"2";CHR$(146) <172>
;" {3SPACE}DIREKTORY BEARBEITEN":PRINT <172>
450 PRINT "{8SPACE}";CHR$(18);"3";CHR$(146) <136>
;" {3SPACE}00UIT":PRINT <136>
470 GET X$:IF X$="" THEN 470 <190>
480 K=VAL(X$):IF K<1 OR K>3 THEN 470 <083>
490 ON K GOTO 510,660,1740 <030>
500 GOTO 330 <238>
510 REM <062>
520 REM <074>
530 REM ***** <160>
540 REM ** DIREKTORY ANZEIGEN <033>
550 REM ***** <180>
560 REM <114>
570 REM <124>
580 PRINT CHR$(147):PRINT:PRINT:POKE 199,1 <011>
:PRINT" BITTE DISK EINLEGEN UND JASTE <011>
DRUECKEN " <217>
590 GET X$:IF X$="" THEN 590 <217>
600 PRINT CHR$(147):SYS 49152 <039>
610 PRINT:PRINT:POKE 199,1:PRINT" JASTE DR <015>
UECKEN " <085>
620 GET X$:IF X$="" THEN 620 <114>
630 GOTO 330 <194>
640 REM <204>
650 REM <005>
660 REM ***** <205>
670 REM *** KOMMA ACHTER <025>
680 REM ***** <244>
690 REM <254>
700 REM <160>
710 T=18:S=1:Z=5 <124>
720 OPEN 1,8,15 <231>
730 OPEN 2,8,2,"#" <110>
740 PRINT#1,"U1";2;0;T;S <194>
750 GET#2,T$,S$ <094>
760 P=T:Q=S <221>
770 IF T$="" THEN T=0:S=255:GOTO 800 <003>
780 T=ASC(T$):S=ASC(S$): <029>
790 PRINT#1,"B-P";2;1 <091>
800 PRINT#1,"B-P";2;Z: <200>
810 FOR I=1 TO 16 <038>
820 GET#2,N$ <029>
830 D$=D$+N$ <162>
840 NEXT I <126>
850 PRINT#1,"B-P";2;Z <076>
860 IF D$="" THEN 1510
870 PRINT CHR$(147):PRINT:PRINT:PRINT" {3SP <204>
ACE}NEXT DIREKTORY DLOCK ":" <228>
880 PRINT:PRINT" {3SPACE}TRACK ":";T;" SEKTO <228>
R ":";S <217>
890 PRINT:PRINT:PRINT:PRINT:PRINT" {3SPACE} <090>
PROGRAMM ":";D$ <021>
900 PRINT:PRINT <004>
915 PRINT" {3SPACE}";CHR$(18);"0";CHR$(146) <021>
;"ASIC ODER ";CHR$(18);"1";CHR$(146);" <004>
ASCHINENPROGRAMM" <021>
920 PRINT <004>
930 PRINT:PRINT:PRINT:PRINT:POKE 199,1:PRI <020>
NT" {4SPACE}<↑> WEITER {3SPACE}<@> LOESC <020>
HEN{6SPACE}"

```

```

940 GET X$:IF X$="" THEN 940 <121>
950 IF X$="B"THEN 1070 <028>
960 IF X$="M"THEN 1160 <163>
970 IF X$="↑"THEN 1300 <230>
980 IF X$="@" THEN 1420 <009>
990 GOTO 940 <028>
1000 REM <044>
1010 REM <054>
1020 REM ***** <042>
1030 REM ** BASICPROGRAMM <025>
1040 REM ***** <064>
1050 REM <096>
1060 REM <106>
1070 D$=LEFT$(D$,12) <059>
1080 NE$=D$+CHR$(160)+"",8:" <021>
1100 PRINT#2,NE$ <053>
1110 NE$="":D$="":Z=Z+32 <198>
1120 IF Z>256 THEN Z=5:GOTO 1510 <074>
1130 GOTO 800 <098>
1140 REM <186>
1150 REM <196>
1160 REM ***** <184>
1170 REM ** MASCHPROGRAMM <010>
1180 REM ***** <204>
1190 REM <236>
1200 REM <246>
1210 D$=LEFT$(D$,11) <191>
1220 NE$=D$+CHR$(160)+"",8,1" <102>
1240 PRINT#2,NE$ <193>
1250 NE$="":D$="":Z=Z+32 <082>
1260 IF Z>256 THEN Z=5:GOTO 1510 <214>
1270 GOTO 800 <238>
1280 REM <072>
1290 REM <082>
1300 REM ***** <021>
1310 REM ** KEINE AENDERUNG <178>
1320 REM ***** <041>
1330 REM <122>
1340 REM <132>
1350 PRINT#2,D$:D$="" <060>
1360 Z=Z+32 <055>
1370 IF Z>256 THEN Z=5:GOTO 1510 <070>
1380 GOTO 800 <094>
1390 REM <182>
1400 REM <192>
1410 REM ***** <131>
1420 REM ** LOESCHEN DES ANHANGS <247>
1430 REM ***** <151>
1440 REM <232>
1450 REM <242>
1460 D$=LEFT$(D$,12)+CHR$(160)+CHR$(160)+C <094>
HR$(160)+CHR$(160) <180>
1470 PRINT#2,D$:D$="" <175>
1480 Z=Z+32 <190>
1490 IF Z>256 THEN Z=5:GOTO 1510 <214>
1500 GOTO 800 <046>
1510 REM <056>
1520 REM <251>
1530 REM ***** <138>
1540 REM ** RUECKSCHREIBEN <017>
1550 REM ***** <098>
1560 REM <108>
1570 REM <248>
1580 IF T=0 AND S=255 THEN PRINT#1,"U2";2; <191>
0;P;Q:GOTO 1690 <149>
1590 PRINT#1,"U2";2;0;P,Q <148>
1600 P=T:Q=S:CLOSE 2:CLOSE 1:GOTO 720 <158>
1610 REM <168>
1620 REM <107>
1630 REM <081>
1640 REM ***** <127>
1650 REM ** ENDE <208>
1660 REM ***** <218>
1670 REM <087>
1680 REM <123>
1690 PRINT:PRINT:POKE 199,1:PRINT" {11SPACE <061>
}JASTE DRUECKEN{10SPACE}" <020>
1720 GET X$:IF X$="" THEN 1720
1730 CLOSE 2:CLOSE 1:GOTO 330
1740 SYS 64738

```

© 64'er
 Listing »Directory-Lader«.
 Bitte beachten Sie die Eingabehinweise auf Seite 6.

Disk-Füller

Ab und zu passiert es, daß auf einer Diskette nur ein paar Blöcke fehlen. Mit dem Disk-Füller können in Notfällen bis zu 17 zusätzliche Blöcke auf der Diskette genutzt werden.

Das Problem ist wohl jedem bekannt, der eine gut sortierte Programmsammlung hat: alle thematisch gleichen Programme bis auf eines passen auf eine Diskettenseite. Und gerade dieses letzte Programm benötigt nur einige wenige Blöcke. Aber hier kann in Notfällen Abhilfe geschaffen werden. Auf der Spur 18, auf der sich unter anderem das Directory befindet, sind meist noch ein paar Blöcke frei, die das DOS der 1541 aber nicht zur Verfügung stellt. Mit dem Programm Disk-Füller können diese freien Blöcke ausgenutzt werden. Zu diesem Zweck werden Blöcke von schon auf der Diskette befindlichen Programmen auf die freien Blöcke der Spur 18 umkopiert und die Blockzeiger angepaßt. Die ursprünglichen Programmblöcke stehen dann zur freien Verfügung.

Das Programm ist weitgehend selbsterklärend, hier aber einige wichtige Hinweise: Nach dem Laden darf das Programm erst mit »RUN« gestartet werden, wenn die zu behandelnde Diskette im Laufwerk liegt. Außerdem dürfen nur Disketten behandelt werden, deren BAM einwandfrei in Ordnung ist. Im Zweifelsfalle sollte vorher ein Validate mit »OPEN 15,8,15,"V"« durchgeführt werden.

Hier ist die genaue Funktionsweise des Programms beschrieben:

Nach einer kurzen Erläuterung für den Benutzer werden in den Zeilen 1050 und 1060 (Listing) zwei Kanäle zum Laufwerk geöffnet. Gleichzeitig wird die eingelegte Diskette initialisiert. In Zeile 2000 werden drei Variablenfelder dimensioniert. Sie dienen zum Ablegen der Zeiger auf den jeweils nächsten Directoryblock sowie der Anzahl der Einträge in diesem Block. Mit Zeile 2010 beginnt dann eine Schleife, die alle Directory-Blöcke verfolgt, ihre Lage auf der Diskette in den dimensionierten Feldern speichert und ihre Anzahl in der Variablen B festhält. Ab Zeile 2500 wird die Anzahl der freien Einträge im letzten Directory-Block bestimmt. Dazu wird jeweils das Filetyp-Byte abgefragt. Hat es den Wert Null, ist dieser Eintrag noch frei. Sollte der Benutzer die Frage, ob der angezeigte Platz genügt, negativ beantworten, wird ein Block für weitere acht Einträge freigehalten.

Als nächstes wird überprüft, welche Blöcke der Spur 18 noch frei sind. Dazu werden die entsprechenden Bytes der BAM in den Computerspeicher geholt und analysiert (Zeile 3120 bis 3230).

In Zeile 4000 beginnt das Verschieben schon belegter Blöcke in die Spur 18. Dazu werden in einer großen Schleife (bis Zeile 4680) alle Directory-Einträge abgearbeitet. Dies geschieht folgendermaßen: Der Zeiger auf den ersten Block des ersten Eintrags wird in den Variablen AS und AT zwischengespeichert, der zugehörige Block eingelesen und der Zeiger auf den Fortsetzungsblock ebenfalls gespeichert. Dann wird ein freier Block der Spur 18 gesucht und der alte Zeiger auf diesen umgelenkt. Der Computer gibt nun eine Protokollzeile aus. Dieser Block wird nun als belegt und der alte als frei gekennzeichnet. Als letztes wird endlich der Inhalt des zu verschiebenden Blocks auf den Block der Spur 18 geschrieben. Sollte nach dem ersten Eintrag immer noch Platz auf der Spur 18 sein, fährt das Programm mit dem zweiten Eintrag fort, bis die Spur 18 komplett belegt ist.

(U. Gerlach/bs)

```

100 REM ***** <140>
110 REM * * <159>
120 REM * * <169>
130 REM * DISK FUELLER * <115>
140 REM * * <189>
150 REM * * <199>
160 REM * * <209>
170 REM * VON UWE GERLACH * <039>
180 REM * * <229>
190 REM * BRUEHLSTR. 23 * <186>
200 REM * * <249>
210 REM * * <003>
220 REM * 6440 BEBRA 1 * <164>
230 REM * * <023>
240 REM * * <033>
250 REM * * <043>
260 REM * * <055>
270 REM * IM APRIL 1985 * <238>
280 REM * * <075>
290 REM * * <085>
300 REM ***** <086>
400 : <122>
500 : <222>
1000 POKE 53280,0: POKE 53281,0: REM FARBE <068>
1010 PRINT CHR$(14);" (CLR,GREY 2,2SPACE)VC
1541(4SPACE)D I S K(3SPACE)E U E L L
E R":PRINT:PRINT:PRINT <073>
1015 PRINT"DIESES PROGRAMM WURDE GESCHRIEB
EN VON:";PRINT:PRINT <051>
1020 PRINT," (LIG.BLUE,2SPACE)UWE GERLACH":
PRINT," (2SPACE)BRUEHLSTR. 23": PRINT
: PRINT," (2SPACE)6440 BEBRA 1 (GREY 2)
" <080>
1025 PRINT:PRINT:PRINT"ES SCHAFFT AUF VOLL
EN DISKETTEN NOCH" <012>
1030 PRINT"ETWAS PLATZ, INDEM ES DIE EIGEN
TLICH" <149>
1035 PRINT"FUER DAS DIRECTORY VORGESEHENE
SPUR 18" <209>
1040 PRINT"BELEGT.":PRINT:PRINT <248>
1045 : <005>
1050 OPEN 1,8,15,"I": REM FUER BEFEHLE <004>
1060 OPEN 2,8,2,"#": REM PUFFERKANAL <091>
1090 : <050>
1100 REM ***** <096>
1110 REM DIRECTORYLAENGE ERMITTELN <193>
1120 REM ***** <116>
1130 : <090>
2000 DIM T(25): DIM S(25): DIM E(25) <001>
2010 PRINT# 1, "U1:"2;0;18;0: REM BAM <085>
2020 GET# 2,T$,S$: REM ZEIGER AUF DIR. <111>
2030 T(0)=ASC(T$+CHR$(0)): REM TRACK <218>
2040 S(0)=ASC(S$+CHR$(0)): REM SECTOR <222>
2050 T=T(0): REM TRACK DIR. ANFANG <167>
2060 S=S(0): REM SECTOR DIR. ANFANG <220>
2070 B=1: REM BLOCKZAEHLER <212>
2100 PRINT# 1, "U1:"2;0;T;S <176>
2110 GET# 2,T$,S$: REM ZEIGER <065>
2120 T=ASC(T$+CHR$(0)) <231>
2130 S=ASC(S$+CHR$(0)) <172>
2140 IF T=0 OR T>35 OR S>20 OR B=25 THEN 2
400: REM LETZTER DIRECTORY-BLOCK <043>
2150 T(B)=T: S(B)=S: REM ZEIGER MERKEN <174>
2155 E(B-1)=8: REM EINTRAEGE IM BLOCK <111>
2160 B=B+1: REM ZAEHLER ERHOEHEN <082>
2190 GOTO 2100: REM NAECHSTER BLOCK <174>
2400 M=B: REM DIR. BLOCKZAHL <228>
2500 F=0: REM ZAEHLER FREIE EINTRAEGE <071>
2510 FOR I=2 TO 226 STEP 32 <227>
2520 : PRINT# 1, "B-P:"2,I: REM ZEIGER <006>
2530 : GET# 2,W$: REM FILETYP <176>
2540 : IF W$="" THEN F=F+1: REM FREI <091>
2590 NEXT I: REM ALLE EINTR. IM BLOCK <090>
2600 E(B-1)=8-F: REM EINTRAEGE/BLOCK <232>
2610 IF F=B AND M=1 THEN 7800: REM LEER <017>
2700 PRINT "GENUEGEN";F;"FREIE DIRECTORY-B
LAETZE ? "; <031>
2750 GET A$: IF A$="" THEN 2750 <230>
2760 IF A$<>"N" THEN A$="J" <234>
2770 PRINT A$: PRINT: PRINT: PRINT <213>
2790 : <226>
3000 REM ***** <218>
3010 REM LEERE DIR. BLOECKE FESTST. <168>
3020 REM ***** <238>
3030 : <212>
3050 PRINT# 1, "U1:"2;0;18;0: REM BAM <109>
3060 PRINT# 1, "B-P:"2,73: REM SPUR 18 <008>
3100 B=0: REM BLOCKZAEHLER <210>

```

Listing »Disk Füller«. Bitte beachten Sie die Eingabehinweise auf Seite 6.

Schreibschutz per Software

Eleganter als das Hantieren mit Klebestreifen ist ein Disketten-Schreibschutz per Software. Hier ist ein Programm, das einen solchen Schreibschutz nicht nur anbringen, sondern ihn auch wieder entfernen kann.

Die Funktion dieses Programms basiert darauf, das Formatkennzeichen (»A«) auf der Diskette zu ändern (hier in »X«). Versucht man nun, Daten auf die Diskette zu schreiben, so reagiert das Floppy-Laufwerk mit der Fehlermeldung »CBM DOS V2.6 1541«. Das Lesen ist jedoch weiterhin problemlos möglich. Damit erspart man sich die zuweilen umständliche Verwendung von Schreibschutz-Aufklebern auf der Diskette. Außerdem ist dieser softwaremäßig realisierte Schreibschutz um einiges sicherer, denn er kann nicht durch Leichtsinns- oder Versehen entfernt werden.

Eine Schwierigkeit ergibt sich aber, wenn das neue Formatkennzeichen wieder in »A« geändert werden soll, um zum Beispiel noch ein Programm auf der Diskette zu speichern. Da ja jegliches Schreiben unmöglich ist, hat man selbst mit den Direktzugriffs-Befehlen keine Möglichkeit mehr, wieder ein »A« auf die Diskette zu bringen.

```

3110 L=19: REM ZAHL LEERE DIR-BLOECKE <082>
3120 DIM Z(25): REM ZUSTAND DER BLOCKS <004>
3130 FOR I=1 TO 3: REM BYTES SPUR 18 <154>
3140 : GET# 2, W$: REM BELEGUNGSCODE <176>
3150 : W=ASC(W$+CHR$(0)) <011>
3160 : FOR J=1 TO 8: REM DUALSTELLEN <207>
3170 : W=W/2 <181>
3180 : IF W=INT(W) THEN Z(B)=1 <203>
3190 : IF Z(B)=0 AND A$="N" THEN A$="": Z <092>
      (B)=1: REM EINEN BLOCK FREIHALTEN <093>
3195 : IF Z(B)=1 AND B<19 THEN L=L-1 <071>
3200 : B=B+1 <134>
3085 : W=INT(W) <076>
3220 : NEXT J <010>
3230 NEXT I <076>
3310 IF L<1 THEN 8000
3390 PRINT: PRINT: PRINT "ES WERDEN NUN";L
      ;" _BLOECKE FREIGESTELLT !": PRINT <233>
3990 : <156>
4000 REM ***** <202>
4010 REM FUELLEN DER DIR.- BLOECKE <251>
4020 REM ***** <222>
4021 : <187>
4025 B=0: REM BLOCKZAehler, SIEHE OBEN <163>
4027 FOR E=0 TO E(B)-1: REM EINTR/BLOCK <038>
4030 AT=T(B): REM ALTER DIR-TRACK <099>
4035 AS=S(B): NS=AS: REM ALTER SECTOR <166>
4040 IF AT=0 AND AS=0 THEN 7050 <004>
4050 I=0 <173>
4060 Z=E*32+3: REM STELLE DES ZEIGERS <170>
4100 R=0: REM MARKE DIR.-BLOCK AENDERN <117>
4150 PRINT# 1, "U1:"2;0;AT;AS: REM ALT <111>
4160 PRINT# 1, "B-P:"2,Z: REM ZEIGER <151>
4170 GET# 2, T$, S$: REM ZEIGER FORTS. <185>
4180 T=ASC(T$+CHR$(0)) <003>
4190 S=ASC(S$+CHR$(0)) <200>
4191 IF T=18 THEN 4670: REM BEARBEITET <245>
4192 IF T=0 OR T>35 OR S>20 THEN Z=1: GOTO <211>
      4520: REM FILEENDE
4200 FOR I=0 TO 18 STEP 10: REM SUCHE NAC <037>
      H FREIEM SECTOR IM SECTORABSTAND
4210 : IF Z(I)=0 THEN 4500: REM LEER <196>
4212 : IF I=9 THEN GOTO 4230 <056>
4215 : IF I>8 THEN I=I-9: GOTO 4210 <096>
4220 NEXT I <240>
4230 Z=-1: REM MARKE: SPUR 18 VOLL <231>
4300 GOTO 4520: REM RUECK OHNE AENDER. <214>
4500 PRINT# 1, "B-P:"2,Z: REM ZEIGER <237>
4510 PRINT# 2, CHR$(18);CHR$(I);: REM ZE <111>
      IGER VERBIEGEN
4515 IF AT=18 AND AS=NS THEN R=1: REM IN <181>
      DEN BLOCK, AUS DEM GELESEN WURDE
4520 PRINT "{6SPACE}_BLOCK";RIGHT$(" {2SPACE <252>
      }"+STR$(AT),3);", "RIGHT$(" {2SPACE}"+S
      TR$(AS),3); <252>
4525 IF R=1 THEN PRINT "{2SPACE}ZEIGER";E+1 <023>
      ;"GEAENDERT": GOTO 4570
4530 PRINT "{2SPACE}-->{2SPACE}18,";RIGHT$ <128>
      (" {2SPACE}"+STR$(NS),3)
4550 PRINT# 1, "B-A:"0,18,NS:REM IN BAM <251>
4560 PRINT# 1, "B-F:"0,AT,AS: REM ALTEN B <209>
      LOCK FREIGEBEN
4570 Z(I)=1: REM BLOCK IN LISTE BELEGEN <254>
4580 PRINT# 1, "U2:"2;0;18;NS:REM RUEC <235>
4650 AT=T: AS=S: NS=I: REM NORM. BLOCK <035>
4655 IF Z=-1 THEN GOTO 9000: REM OK <109>
4660 IF Z<>1 THEN Z=0: GOTO 4100 <234>
4670 NEXT E: REM NEUER DIR.-EINTRAG <120>
4680 B=B+1: GOTO 4027: REM WEITER MIT NAE <087>
      CHSTEM DIRECTORYBLOCK
6990 : <108>
7000 REM ***** <154>
7010 REM FEHLERMELDUNGEN <130>
7020 REM ***** <174>
7030 : <148>
7050 PRINT: PRINT "ES WAREN NICHT GENUG BE <036>
      LEGTE _BLOCKS DA."
7060 PRINT " SPUR 18 IST IMMERNOCH TEILWEI <099>
      SE FREI!"
7090 GOTO 9000 <236>
7800 PRINT:PRINT:PRINT"DIE DISKETTE IST LE <146>
      ER!"
7810 GOTO 9000 <194>
8000 PRINT "TUT MIR LEID, DA IST NICHTS ZU <162>
      MACHEN."
8010 PRINT "DIE GESAMTE SPUR 18 IST VOLL B <160>
      ELEGT !"
9000 PRINT <212>
9010 CLOSE 1: CLOSE 2: END <099>

```

Listing »Disk Füller« (Schluß)

```

1 REM ***** <051>
2 REM ** ** <002>
3 REM ** SCHREIBSCHUTZ ** <255>
4 REM ** ** <004>
5 REM ** '85 BY ** <128>
6 REM ** MANFRED LINS ** <031>
7 REM ** REITACKER 1 ** <221>
8 REM ** 6492 SINNTAL-ZUENTERSBACH ** <140>
9 REM ***** <059>
10 PRINT CHR$(147):REM CLEAR/HOME <146>
20 X=7:Y=2:GOSUB 510 <017>
30 PRINT"D I S K{2SPACE}-{2SPACE}S C H U T <035>
      Z"
40 X=2:Y=6:GOSUB 510 <217>
50 PRINT"ALTES FORMATKENNZEICHEN :"; <075>
60 OPEN 1,8,15,"I":OPEN 2,8,2,"# " <051>
70 PRINT#1,"U1 2 0 18 0 " <130>
80 PRINT#1,"B-P 2 2 " <244>
90 GET#2,AF$: REM ALTES FORMATKENNZ. <169>
100 PRINT AF$ <188>
110 X=2:Y=9:GOSUB 510 <034>
120 PRINT"WIRD GEAENDERT IN{3SPACE}"; <012>
130 IF AF$="A"THEN PRINT"X":GOTO 150 <096>
140 PRINT"A" <091>
150 X=20:Y=9:GOSUB 510 <236>
160 INPUT NF$ <016>
170 PRINT#1,"M-W"CHR$(1)CHR$(1)CHR$(1)CHR$ <032>
      (65) <088>
180 PRINT#1,"B-P 2 2 " <161>
190 PRINT#2,NF$:REM NEUES FORMATKENNZ. <133>
200 PRINT#1,"U2 2 0 18 0 " <232>
210 CLOSE 2:CLOSE 1
220 IF NF$="A"THEN D$="{RVSON,2SPACE}KEIN <001>
      SCHREIBSCHUTZ !{2SPACE}":GOTO 240
230 D$="{RVSON,2SPACE}SCHREIBSCHUTZ AKTIVI <022>
      ERT !{2SPACE}"
240 X=6:Y=14:GOSUB 510 <177>
250 PRINT D$ <196>
260 END <008>
500 REM ROUTINE ZUM CURSORSETZEN <250>
510 POKE 211,X:POKE 214,Y:SYS 58640:RETURN <011>
520 REM UNTERPROGRAMMENDE <169>

```

Listing »Schreibschutz«. Bitte beachten Sie die Eingabe- hinweise auf Seite 6.

Hier liegt nun die eigentliche Neuheit dieses Programms. Mit einem Memory-Write-Befehl (Listing) wird in die Speicherstelle 257 (\$0101) der Floppy der ASCII-Code von »A« (65) geschrieben. In dieser Speicherstelle wird bei jedem Initialisieren einer Diskette deren Formatkennzeichen im ASCII-Format abgelegt. Ist nun dieser Inhalt gleich 65, so schreibt die Floppy wieder ohne Probleme auf die Diskette, und genau dies wird in dem Programm ausgenutzt.

Nun einige Bemerkungen zur Bedienung des Programms: Nach dem Starten mit »RUN« wird das Formatkennzeichen der Diskette eingelesen und ausgedruckt. Daraufhin gibt man das Format-Zeichen ein, das die Diskette erhalten soll. Drückt man nur RETURN, so wird das vorgegebene Zeichen übernommen und es kommt immer zu einer Änderung des Formatkennzeichens. Zum Schluß wird dann der aktuelle Zustand der Diskette angezeigt.

Achtung! Unmittelbar nach dem Programmlauf ist der Schreibschutz nicht aktiv, da in der Speicherstelle 257 noch immer 65 steht. Erst nach dem Initialisieren einer geschützten Diskette ist dieser Schutz auch wirksam. Auch gegen versehentliches Formatieren, bei dem ja alle Programme gelöscht werden, schützt dieses Programm nicht, da bei diesem Befehl logischerweise kein Formatkennzeichen gelesen wird.

In allen anderen Fällen ist der Schutz jedoch voll wirksam und macht Schreibschutz-Etiketten überflüssig.

(Manfred Lins/ev)

Disk-Tester

Mit diesem Programm lassen sich Disketten schneller, komfortabler und schonender auf defekte Sektoren untersuchen, als das mit dem entsprechenden Programm auf der Test/De-modiskette der Fall ist.

Das Programm ist insbesondere bei Verwendung älterer Disketten oder bei Nutzung der Disketten-Rückseite zu empfehlen. Es überprüft in knapp 10 Minuten die gesamte Diskette auf schadhafte Stellen und markiert die eventuell gefundenen unbrauchbaren Sektoren als belegt, so daß sie von der 1541 nicht mehr zur Datenspeicherung verwendet werden. So ist es möglich, auch teilweise beschädigte Disketten noch zur Datenspeicherung heranzuziehen, ohne daß es zu WRITE- oder READ-ERRORS kommt.

Programmbeschreibung:

Nach dem Starten des Programms (siehe Listing) muß man entscheiden, ob man nur die vollen Blocks (durch Lesen) oder alle Blocks (durch Lesen und Beschreiben) testen will, oder ob man die schon bei vorherigen Tests als fehlerhaft erkannten Blocks nach einem VALIDATE wieder belegen will. Danach wird gefragt, ob der Befehl »VALIDATE« an die Floppy gesendet werden soll. Dies ist nur bei Disketten sinnvoll, auf denen die Daten öfters geändert werden und die keine relativen Dateien, ISAM-Dateien oder ähnliche, im Direktzugriff arbeitenden Speicherformate, enthalten. Bei Disketten mit den oben genannten Dateien, von denen man nicht sicher weiß, ob die beschriebenen Blöcke auch in der BAM (Block Availability Map, Sektorbelegungstabelle) als belegt gekennzeichnet sind, sollte die Option »Leere Blocks testen« auf keinen Fall angewandt werden, da sonst Daten verlorengehen können.

Es folgt nach kurzer Lese- und Decodierzeit die symbolische Belegung der Diskette auf dem Bildschirm, auf der die beleg-

ten Sektoren als »-« und die nicht belegten als »*« gekennzeichnet sind. Wie kommt es dazu? Nun, jede Spur belegt in der BAM (Spur 18, Block 0, Positionen 4-143) 4 Byte. Zuerst kommt die Anzahl der freien Blöcke auf der Spur, im Programm bezeichnet mit FR\$. Dann kommt nacheinander die Belegung der Sektoren 7-0, 15-8 und 23-16. Ein 1-Bit kennzeichnet dabei einen freien Block, ein 0-Bit einen besetzten. Dies wird nun in der Routine »Analysieren der BAM« aufgelöst und in B%(T,S) festgehalten, wobei T die Spur und S den Sektor enthält. Die darauffolgende Testroutine beschränkt sich jetzt nur darauf, die Spuren und Sektoren einzeln durchzuzählen und in Abhängigkeit von B%(T,S) und der Variable TS (0, wenn alle, 1, wenn nur volle Blocks getestet werden sollen) entweder den Block zu lesen, oder ihn mit dem einmal aufgefüllten Puffer zu beschreiben. Danach wird der Fehlerkanal auf die Fehlernummer abgefragt, da dies schneller geht, als immer die ganze Fehlermeldung zu holen. Tritt ein Fehler auf, so wird dieser angezeigt und auf dem Monitor als roter Punkt in der BAM markiert. Gleichzeitig wird die Variable FE erhöht. Tritt kein Fehler auf, so wird die Kennzeichnung für den gerade getesteten Block auf dem Monitor gelöscht. Sind alle Blocks getestet, werden die eventuell vorhandenen defekten Blocks auf der Diskette belegt und in dem Format »Spur,chr\$(13),Block,chr\$(13)« in dem USR-File ».DISKFEHLER« gespeichert. Das erste Byte entspricht in diesem Fall der Anzahl aller gefundenen Fehler.

Die Routine »Kaputte Blocks belegen« hat nun nichts weiter zu tun, als dieses File zu lesen und die Blocks zu belegen.

Änderungen für andere Computer

Für einen der »großen« CBMs muß die Zeile 135 und sämtliche POKEs in den Zeilen 150 und 470 gelöscht werden, da sie keine Farbdarstellungen besitzen. Die Zeile 140 muß so geändert werden:

```
140 BI=32768
```

Für einen VC 20 mit 40-Zeichen-Bildschirm müssen nur die farbgebenden POKEs in Zeile 150, die Strings in Zeile 135 und die Variablen in Zeile 140 angepaßt werden. Für die Besitzer eines VC 20 mit 22-Zeichen-Bildschirm sind größere Änderungen erforderlich, da die BAM nicht angezeigt werden kann und sie sich somit mit einer numerischen Darstellung des Testablaufs zufrieden geben müssen. Es werden die freien und die belegten Blocks sowie der zuletzt getestete Sektor angezeigt. Die Zeilen 140, 470 und 635 ff. müssen vollständig gelöscht werden. Folgende Zeilen müssen geändert werden:

```
355 PRINTL$T$ " BESETZTE BLOCKS: " O$683-FR-  
PRINTT$ " FREIE BLOCKS": " O$FR;D$D$  
450 INPUT #15,F:IFF < 20GOTO480  
480 PRINTG$ " T: " T;Z$ " S;Z$ " " U
```

Zehn Minuten für das Überprüfen einer Diskette mögen zwar immer noch recht lang anmuten, wenn man jedoch bedenkt, wieviel Ärger (READ-ERROR, WRITE-ERROR) ein fehlerhafter Sektor verursachen kann, dann rentiert es sich bestimmt, diese Zeit zu investieren. (Ulrich Langler/ev)

```
100 REM*DISKTESTER <068>  
105 REM*ULI LANG* <153>  
110 REM*TALSTR.10* <167>  
115 REM*8607 BISCHBERG* <015>  
120 REM*TEL.0951/67389* <141>  
125 : <101>  
130 L$=" {CLR}":D$=" {DOWN}":U$=" {UP}":Z$="{  
LEFT}":REM*LOESCHEN,CRSR DOWN,CRSR UP,  
CRSR BACK* <215>
```

Listing »Disk-Tester«. Bitte beachten Sie die Eingabehinweise auf Seite 6.

```

135 G$="{GREEN}":O$="{ORANGE}":B$="{LIG.BL
    UE}":T$="{CYAN}":R$="{LIG.RED}":REM*GR
    UEN,ORANGE,HELLBLAU,TUERKIS,ROT* <199>
140 FARBSPEICHER=55296:BILDSCHIRMSPEICHER=
    1024 <242>
145 DIM T%(100),S%(100):REM*FEHLERHAFT E BL
    OCKS* <033>
150 PRINT L$G$D$SPC(15)"DISKTESTER"D$:POKE
    53280,0:POKE 53281,0:REM*SCWARZ* <079>
155 PRINT O$D$<"B$"A"O$"> ALLE{4SPACE}BLO
    CKS TESTEN" <221>
160 PRINT D$<"B$"B"O$"> BELEGTE BLOCKS TE
    STEN" <237>
165 PRINT D$<"B$"K"O$"> KAPUTTE BLOCKS BE
    LEGEN" <027>
170 GET A$ <124>
175 IF A$="A"THEN TS=1:GOTO 210 <250>
180 IF A$="B"THEN TS=0:GOTO 210 <096>
185 IF A$="K"THEN 575 <007>
190 GOTO 170 <230>
195 REM***** <076>
200 REM*DISKETTE TESTEN* <025>
205 REM***** <086>
210 DIM B%(35,23),F$(8):K$=",";N$=CHR$(0):
    FR=0:FE=0:FT=0 <085>
215 FOR I=0 TO 8:READ F$(I):NEXT:REM*FEHLE
    RMELDUNGEN* <102>
220 GOSUB 610:REM*AUF SPACE WARTEN* <179>
225 OPEN 15,8,15,"I" <024>
230 PRINT L$"VALIDATE?"B$"J/N" <214>
235 GET A$ <189>
240 IF A$="N"THEN 255 <164>
245 IF A$="J"THEN PRINT L$T$D$"VALIDATING.
    ..":PRINT#15,"V":GOTO 255 <060>
250 GOTO 235 <138>
255 OPEN 3,8,3,"#":IF TS THEN OPEN 5,8,5,"
    #" <216>
260 PRINT L$T$"BAM WIRD GELESEN{9SPACE}"U$ <248>
265 PRINT#15,"U1:"3;0;18;0:GET#3,A$,A$,A$ <240>
270 IF A$<>CHR$(65)THEN PRINT D$"KEIN DOS
    2.6 ODER 2.5!":GOTO 550 <096>
275 GET#3,A$ <206>
280 REM***** <167>
285 REM*ANALYSIEREN DER BAM * <230>
287 REM*AUFGELOEST IN B%(T,S)* <119>
290 REM***** <177>
295 FOR T=1 TO 35 <025>
300 REM*FREIE BLOCKS,SEKTOR 7-0,SEKTOR 1
    5-8,SEKTOR 23-16* <072>
305 GET#3,FR$:FR=FR+ASC(FR$+N$):S=0 <226>
310 IF FR THEN 320:REM*NICHT ALLE BLOCKS
    BELEGT* <058>
315 FOR S=0 TO 23:B%(T,S)=0:NEXT S:GOTO
    345 <091>
320 A=1:GET#3,BA$:BA=ASC(BA$+N$) <086>
325 FOR S=S TO S+7 <102>
330 B%(T,S)=(BA AND A)/A:A=A+A:FT=FT+B
    %(T,S) <015>
335 NEXT S <245>
340 IF S<23 THEN 320 <149>
345 NEXT T <007>
350 IF FR<>FT THEN PRINT L$R$TAB(14)"!BAM-
    FEHLER!":GOTO 550 <183>
355 GOSUB 635 <037>
360 REM***** <247>
365 REM*START DER TESTROUTINE* <078>
370 REM***** <001>
375 IF TS=0 THEN FR=683-FR:REM*VOLLE BLOCK
    S* <228>
380 IF TS=0 THEN 400 <012>
385 REM*170=X10101010 , 85=X01010101* <012>
390 PRINT#15,"B-P:"5;1:FOR I=1 TO 128:PRIN
    T#5,CHR$(170)CHR$(85):NEXT <148>
395 REM*PUFFER BRAUCHT NUR EINMAL GEFUELLT
    ZU WERDEN* <074>
400 FOR T=1 TO 35:REM*SPURZAEHLER* <138>
405 IF T<18 THEN MS=20:GOTO 425 <089>
410 IF T<25 THEN MS=18:GOTO 425 <053>
415 IF T<31 THEN MS=17:GOTO 425 <073>
420 MS=16 <083>
425 FOR S=0 TO MS:REM*BLOCKZAEHLER* <161>
430 ON TS*2+B%(T,S)GOTO 485,435,440 <140>
435 PRINT#15,"U1:"3;0;T;S:GOTO 445:REM
    *BLOCK LESEN* <031>
440 PRINT#15,"B-W:"5;0;T;S:REM*BUFFER
    AUF DISK SCHREIBEN* <197>
445 AO=40*(20-S)+T+3:REM*ADRESSOFFSET* <156>
450 INPUT#15,F:IF F<20 THEN P=32:GOTO
    480 <224>
455 IF F<29 THEN F=F-(F-20):GOTO 465 <078>
460 F$="" <255>
465 PRINT R$F;F$ T:"T;Z$ S:"S;Z$" "U
    $ <224>
470 POKE FA+AO,10:P=81+6*B%(T,S):REM*R
    OTES"Z,B" <123>
475 T%(FE)=T:S%(FE)=S:FE=FE+B%(T,S) <040>
480 POKE BI+AO,P <254>
485 NEXT S <139>
490 NEXT T:CLOSE 5:CLOSE 3 <191>
495 IF FE=0 THEN PRINT{6SPACE}!KEINE ZU B
    ELEGENDE FEHLER!";GOTO 550 <007>
500 REM***** <002>
505 REM*BELEGEN DER KAPUTTEN BLOCKS* <069>
510 REM***** <012>
515 PRINT L$T$" TEST ZUENDE, BELEGEN UND K
    ENNZEICHNEN" <220>
520 PRINT TAB(8)"DER FEHLERHAFTEN BLOCKS" <170>
525 FOR I=0 TO FE-1 <062>
530 PRINT#15,"B-A:"0;T%(I);S%(I):REM*IN
    BAM BELEGEN* <216>
535 NEXT <037>
540 OPEN 3,8,3,"@:.DISKFEHLER,U,W":PRINT#3
    ,CHR$(FE-1); <099>
545 FOR I=0 TO FE-1:PRINT#15,T%(I)CHR$(13)
    S%(I):NEXT <120>
550 CLOSE 3:CLOSE 15:END <029>
555 : <023>
560 REM***** <106>
565 REM*WIEDERBELEGUNG NACH VALIDATE* <221>
570 REM***** <116>
575 GOSUB 610 <067>
580 PRINT L$T$"WIEDERBELEGEN DER SCHADHAFT
    EN BLOCKS" <143>
585 OPEN 15,8,15,"I":OPEN 5,8,5,".DISKFEHL
    ER,U,R" <190>
590 GET#3,A$:FE=ASC(A$) <151>
595 FOR I=0 TO FE:INPUT#5,T%(I),S%(I):PRIN
    T R$T:"T%(I),"S:"S%(I):NEXT:CLOSE 5 <124>
600 FOR I=0 TO FE:PRINT#15,"B-A:"0;T%(I);S
    %(I):NEXT:CLOSE 15:END <087>
605 : <073>
610 PRINT L$TAB(9)O$"TESTDISKETTE{2SPACE}E
    INLEGEN":PRINT TAB(17)"<"B$"SPACE"O$">
    " <021>
615 GET A$:IF A$<>" THEN 615 <216>
620 RETURN <170>
625 : <093>
630 REM*BAM DARSTELLEN* <210>
635 PRINT L$; <240>
640 FOR S=20 TO 0 STEP-1 <173>
645 PRINT O$S;TAB(4); <038>
650 FOR T=1 TO 35 <126>
655 IF T<18 THEN MS=20:GOTO 675 <091>
660 IF T<25 THEN MS=18:GOTO 675 <055>
665 IF T<31 THEN MS=17:GOTO 675 <075>
670 MS=16 <079>
675 IF S>MS THEN 690 <066>
680 IF B%(T,S)=0 THEN PRINT B$"-";:GOT
    O 690 <183>
685 PRINT G$"*"; <173>
690 NEXT T:PRINT <056>
695 NEXT S:PRINT TAB(4); <214>
700 FOR T=1 TO 35:PRINT RIGHT$(STR$(T),1);
    :NEXT <078>
705 PRINT:RETURN <228>
710 REM*FEHLERMELDUNGEN* <008>
715 DATA"KEIN HEADER{6SPACE}","KEIN SYNC C
    HAR{3SPACE}","KEIN BLOCK VORH. " <119>
720 DATA"PRUEFSUMME DATEN ","BITMUSTERFEHL
    ER{2SPACE}","SCHREIBFEHLER{4SPACE}" <254>
725 DATA"SCHREIBSCHUTZ{4SPACE}","PRUEFSUMM
    E HEADER","BLOCK ZU LANG{4SPACE}" <253>

```

Listing »Disk-Tester« (Schluß)

Betriebs- system selbst gemacht

Ein eigenes Betriebssystem zu schreiben, ist nicht jedermanns Sache. Aber warum soll man nicht das vorhandene für sich abändern?

Wie schön wäre es doch, sein eigenes, kleines Betriebssystem für den C 64 zu schreiben. Doch gewiß ist das nicht gerade jedermanns Sache. Daraus könnte sich schnell ein »Fulltime-Job« mit Beschäftigung bis ins hohe Rentenalter entwickeln. Es liegt viel näher, das vorhandene Betriebssystem einfach so lange abzuändern, bis ein eigenes, den Bedürfnissen angepaßtes Kernall entsteht. Beste Beispiele sind das Hypra-Kernal und Hypra-Perfekt (64'er, Ausgaben 12/84, 1/85 und 4/85). Bestimmt hat so mancher inzwischen schon eine kleine Kernall-Sammlung auf Diskette oder EPROM.

Ganz interessant wäre es dann, auf einfache Weise diese Systeme weiter abzuändern. Ihnen beispielsweise einen deutschen Basic-Interpreter zu verpassen, mit deutschen System- und Fehlermeldungen.

Gesagt, getan: Vorliegendes Programm erledigt dies alles. Es ist abgestimmt auf die Kernall-Version aus der 64'er-Ausgabe 1/85 sowie auf das Hypra-Perfekt-Kernal aus Ausgabe 4/85, arbeitet jedoch auch mit vielen anderen Betriebssystemen. Probieren geht hier über Studieren.

Im Gegensatz zu den oben genannten EPROM-Versionen, erzeugt dieses Programm ein lauf- und damit testfähiges Betriebssystem im freien RAM-Bereich unter dem Kernall-ROM ab \$E000. Nicht jeder wird wohl gleich ein EPROM program-

mieren, sondern lieber zuerst etwas experimentieren wollen. Zuerst werden Basic und Kernall ins RAM kopiert (Init-Routine ab 190). Erst dann kann geändert werden. Das Programm (siehe Listing) erlaubt auch (falls vorhanden) ein Betriebssystem von Diskette zu laden (Zeile 23, siehe Tabelle 1 und 2). Danach kann die Farbkombination beim Warm- und Kaltstart ausgetestet werden. Der SYS-Befehl in Zeile 201 löst einen Reset des Bildschirms innerhalb der geänderten Kernallroutine aus, so daß die Kombination sofort beurteilt werden kann. Erst nach Bestätigen verzweigt das Programm nach Zeile 213. Dort kann eine OLD-Routine aktiviert werden. Beginn ist ab Adresse 64608. Dies gilt es zu beachten, wenn ein Kernall abgeändert werden soll, das eventuell diesen Bereich benutzt. Es handelt sich dabei um einen Teil der normalerweise überschriebenen Kassettenroutinen des Betriebssystems. Hat man ein Betriebssystem geladen, das mit der OLD-Routine kollidiert, muß letztere verschoben werden.

Die genannten Hypra-Kernal sind diesbezüglich unkritisch. Beachtet werden muß lediglich, daß sich deren Adressen für Funktionstasten-Belegung unterscheiden. Soll OLD also auch auf Funktionstaste gelegt werden, fragt das Programm selbst nach der verwendeten Version (Zeile 218). Bei anderen Kernall-Versionen kommt es auf das berühmte Experiment an. Betriebssysteme ohne eine Belegung der Funktionstasten können selbstverständlich auch nicht den OLD-Befehl per Funktionstaste unterstützen. Es fehlen ganz einfach die nötigen Routinen (siehe Original Kernall V2.0)!

Hier bleibt ja immer noch der SYS-Befehl zum Aufruf von OLD, den man sich eben merken muß. Für ganz zerstreute Freaks ist die nachfolgende Routine am idealsten. OLD als gewöhnlicher Basic-Direktbefehl (also auch abkürzbar mit »O SHIFT L«): Zeile 224 sorgt für die entsprechende Programmierung. Da hier, wie nachher auch bei den Fehlermeldungen in deutscher Sprache, das Basic mitverändert wird, muß natürlich auch das Basic-ROM (beziehungsweise -RAM) abgespeichert werden. Wie, das wird zum Schluß erklärt.

Wurden alle Eingaben gemacht, startet das Programm die Initialisierungs-Routine, die Kernall und Basic mit deutschen System- und Fehlermeldungen versieht. Zu guter Letzt wird das neue Betriebssystem aktiviert (Zeile 133). Jetzt können ruhig mal alle Neuheiten durchgespielt werden. Gar nicht so einfach, alle Fehler-Meldungen zu »provizieren«.

Und dabei gibt es immer noch Programmierer, die sich bei Meldungen wie »Syntax Error« die Haare raufen. Auf jeden Fall: einen »Syntax Error« wird es für jene jetzt wohl kaum mehr geben.

Jetzt noch kurz zu den Änderungen gegenüber den EPROM-Versionen. Um ein EPROM zu brennen, muß das Programm in einem Bereich liegen, auf den das EPROM-Programmiergerät zugreifen kann. Daher liegen die Hypra-Kernal ab \$6 000 im Speicher. Die Adressen bleiben unkorrigiert, so daß als EPROM ein lauffähiges Betriebssystem im Originalbereich ab \$E000 entsteht. Ausgetestet werden kann aber nur, wenn das Betriebssystem wieder an seinen Ursprung verschoben wird. Das kann am komfortabelsten mit einem Monitorprogramm durchgeführt werden.

Ohne Monitor muß mit einer Basic-PEEK/POKE-Schleife der Speicherbereich \$6000 bis \$7FFF auf die Adressen ab \$E000 verschoben werden:

```
forx=24576to49151:for y=57344to65535
poke y,peek(x):next y:next x
```

Zuvor muß sichergestellt sein, daß das RAM unter dem ROM ab \$E000 aktiviert wurde (wie durch die Routine ab Zeile 193). Am besten, man speichert sich diese Basic-Routine separat als Programm, da sie beim Einlesen eines Betriebssystems immer wieder benötigt wird. Aus Zeitgründen wäre eine solche Routine in Maschinen-Code auch nicht zu verachten.

Der verschobene Speicherbereich muß nun lediglich noch auf Disk abgespeichert werden. Ohne Monitor kann dazu zum

| | |
|---------|---|
| 100-101 | Feststellen ob ROM/RAM aktiv Feststellen ob Kernall nachgeladen |
| 102 | RAM aktivieren |
| 103 | Routine zum Data-Einlesen |
| 123-127 | Anfangs- und Endadressen der POKE-Schleifen |
| 128-132 | Variablenübergabe an DATA-Einleseroutine und Prüfsummen der Datenblöcke |
| 133 | Aktivieren des neuen Kernall/Basic |
| 134-154 | Datenblock 1 Fehlermeldungen |
| 155-159 | Datenblock 2 Interpreter-Meldungen |
| 160-166 | Datenblock 3 Startkennung |
| 167-175 | Datenblock 4 System-Meldungen Kernall |
| 176-182 | Datenblock 5 Old-Routine |
| 183-189 | Prüfsummen-Statement |
| 190-197 | ROM-RAM Kopieroutine |
| 198-209 | Austesten der Farbgebung |
| 210-230 | Old-Routine anpassen |
| 231-239 | Nachladen eines Betriebssystems |

| | |
|-----|---------------------------------------|
| N\$ | = Filename des nachgeladenen Kernalls |
| P | = Zeropage-Adresse 1 auf ROM oder RAM |
| A | = Anfangsadresse der DATA-Schleifen |
| B | = Endadresse der DATA-Schleifen |
| S | = Prüfsumme |
| Z | = Zeichenfarbe |
| R | = Rahmenfarbe |
| H | = Hintergrundfarbe |
| K | = verwendete Kernallversion |

▲ **Tabelle 1. Programmbeschreibung**

◀ **Tabelle 2. Die wichtigsten Variablen**

Beispiel das kleine Programm im 64'er, Ausgabe 2/85, Seite 91 verwendet werden.

Mit einem Maschinensprache-Monitor geht's einfacher:

Verschieben: .T6000,7FFF,E000-

Save Kernal: .S »K-Name«,08,E000,FFFF

Save Basic: .S »B-Name«,08,A000,BFFF

Beim SMON hat der Verschiebe-Befehl den Buchstaben »W«, und Komma, Minuszeichen und Geräteadresse entfallen.

Übrigens kann auch das ROM ins RAM mittels Monitor kopiert werden. Die Befehle dazu lauten

Kernal: T E000, FFFF, E000-Basic: T A000,BFFF,A000- (danach POKE 1,53).

Auf der Diskette hat man jetzt ein Betriebssystem, das nur mehr vom Programm nachgeladen zu werden braucht. Es kann aber auch schon vorher ein Betriebssystem im RAM aktiv sein. Das Bearbeitungsprogramm erkennt automatisch, ob das Original oder ein neues Kernal aktiv ist (Adresse 1 gleich 55 oder 53, Zeile 100/101)! Andernfalls wird immer das Original-Betriebssystem bearbeitet.

Grundsätzlich muß vor jedem Laden einer Kernal- oder Basic-Version von ROM auf RAM umgePOKEt werden.

Arbeiten mit dem neuen Betriebssystem:

1. ROM auf RAM umPOKEn
2. Zeropage-Adresse 1 mit POKE 1,53 umschalten
3. Kernal laden mit »LOAD "Name";8,1«
4. Basic laden mit »LOAD "Name";8,1«

Zur Kontrolle, daß tatsächlich das neue System läuft, kann mit SYS 58648, SYS 58260 in den Warmstart gesprungen werden. Ein SYS 64738 schaltet immer auf das Original-ROM zurück und sollte vermieden werden. Falls er doch einmal ausgelöst wird (zum Beispiel Run-Stop/Restore), kann mit POKE 1,53 jederzeit wieder ins RAM geschaltet werden.

Da das Bearbeitungsprogramm auch das Basic-ROM ändert, muß auch immer Basic von Disk geladen werden. Andernfalls hat man englische Fehlermeldungen und OLD nur als SYS-Befehl.

Zum Abschluß muß noch erwähnt werden, daß der OLD-Befehl den Basic-Befehl END eliminiert. Bei Fremdprogrammen sollte im Hinblick auf Syntax-Fehler daran gedacht werden. Eine Verlängerung der Basic-Befehlsliste ist aus Platzgründen in dieser Programmversion nicht realisiert worden.

(Richard Diezmann/rg)

```

100 IF N$="" THEN P=PEEK(1):IF P<>53 THEN 1  <198>
    91
101 IF N$="" THEN P=PEEK(1):IF P=53 THEN 23  <136>
    2
102 POKE 1,53:GOTO 201 <212>
103 FOR I=A TO B:READ D:POKE I,D:S=S+D:NEX  <220>
    T I
104 RETURN <162>
105 REM***** <117>
106 REM* * <155>
107 REM*   BETRIEBSSYSTEM-EDITOR * <100>
108 REM* * <157>
109 REM*   COMMODORE 64 * <244>
110 REM*   KERNAL UND BASIC * <176>
111 REM* * <160>
112 REM*   1985 BY RIDI * <095>
113 REM* ----- * <209>
114 REM*   RICHARD DIEZMANN * <058>
115 REM*   LOISACHSTRASSE 5 * <182>
116 REM*   8400 REGENSBURG * <099>
117 REM* * <166>
118 REM*   TEL. (0941)49542 * <192>
119 REM* * <168>
120 REM***** <132>
121 POKE 53280,1:POKE 53281,1:PRINT CHR$(1  <190>
    47)CHR$(152)
122 PRINT"DEUTSCHES BETRIEBSSYSTEM WIRD";:  <170>
    PRINT CHR$(13)"INITIALISIERT !"
123 A1=41374:E1=41767 <221>
124 A2=41828:E2=41865 <117>
125 A3=58463:E3=58540 <017>
126 A4=61629:E4=61738 <109>
    
```

```

127 A5=64608:E5=64667 <198>
128 A=A1:B=E1:GOSUB 103:IF S<>29905 THEN 1  <167>
    83
129 A=A2:B=E2:S=0:GOSUB 103:IF S<>1919 THE  <088>
    N 183
130 A=A3:B=E3:S=0:GOSUB 103:IF S<>4464 THE  <071>
    N 183
131 A=A4:B=E4:S=0:GOSUB 103:IF S<>7664 THE  <122>
    N 183
132 A=A5:B=E5:S=0:GOSUB 103:IF S<>6755 THE  <169>
    N 183
133 POKE 41812,219:SYS 58648:SYS 58260 <154>
134 REM DATENBLOCK 1
    BASIC-FEHLERMELDUNGEN <240>
135 REM ----- <181>
136 DATA 90,85,32,86,73,69,76,69,32,70,73,  <038>
    76,69,211,79,70,70,69,78,161,32
137 DATA 32,32,78,73,67,72,84,32,79,70,70,  <190>
    69,78,161,32,78,73,67,72,84,32,71
138 DATA 69,70,85,78,68,69,206,71,69,82,65  <202>
    ,69,84,32,65,85,83,32,186,32,32
139 DATA 32,32,32,32,75,69,73,78,32,69,73,  <044>
    78,71,46,70,73,76,197,75,69,73,78
140 DATA 32,65,85,83,71,65,46,70,73,76,197  <018>
    ,70,73,76,69,78,65,77,69,32,70,69
141 DATA 72,76,84,161,32,32,73,76,76,69,71  <225>
    ,65,76,69,32,71,69,82,65,69,84,78
142 DATA 85,77,77,69,210,78,69,88,84,32,79  <143>
    ,72,78,69,32,70,79,210,32,32,32
143 DATA 83,80,82,65,67,200,82,69,84,85,82  <242>
    ,78,32,79,72,78,69,32,71,79,83,85
144 DATA 194,32,32,32,68,65,84,69,78,90,69  <057>
    ,73,76,69,206,70,65,76,83,67,72
145 DATA 69,32,90,65,72,76,186,32,32,32,90  <090>
    ,65,72,76,32,90,85,190,75,69,73
146 DATA 78,32,82,65,77,32,77,69,72,210,90  <075>
    ,69,73,76,69,32,78,73,67,72,84,32
147 DATA 68,65,32,161,32,78,73,67,72,84,32  <099>
    ,73,78,32,68,73,77,161,82,69,68
148 DATA 73,77,32,65,82,82,65,89,32,173,68  <190>
    ,73,86,73,83,73,79,78,32,68,85,82
149 DATA 67,72,32,176,78,85,82,32,80,82,71  <051>
    ,77,79,68,85,83,161,90,65,72
150 DATA 76,32,83,84,65,84,84,32,83,84,82,  <134>
    73,78,71,161,32,90,85,32,76,65,78
151 DATA 71,161,32,70,73,76,69,32,68,65,84  <112>
    ,193,83,84,82,73,78,71,32,90,85
152 DATA 32,75,79,77,80,76,69,88,161,32,75  <214>
    ,65,78,78,32,78,73,67,72,84,161
153 DATA 32,32,32,85,78,68,69,70,46,32,70,  <237>
    85,78,75,84,73,79,78,161,80,82,85
154 DATA 69,70,197,76,65,68,197 <244>
155 REM <217>
156 REM DATENBLOCK 2
    MELDUNGEN DES INTERPRETERS <175>
157 REM ----- <023>
158 DATA 13,79,75,13,0,32,70,69,72,76,69,8  <016>
    2,0,32,73,78,32,0,13,10,66,69,82
159 DATA 69,73,84,13,10,0,13,10,80,65,85,8  <128>
    3,69,0,160
160 REM <222>
161 REM DATENBLOCK 3
    STARTKENNUNG BETRIEBSSYSTEM <145>
162 REM ----- <028>
163 DATA 0,32,66,89,84,69,83,32,70,65,83,8  <221>
    4,76,79,65,68,73,78,71,0,147,13
164 DATA 32,32,32,32,32,32,42,32,67,79,77,  <243>
    77,79,68,79,82,69,32,54,52,32,75
165 DATA 69,82,78,65,76,32,86,50,46,48,32,  <088>
    42,32,13,13,40,67,41,49,57,56,53
166 DATA 32,66,89,32,82,73,68,73,46,32,0,1  <147>
    29
167 REM <229>
168 REM DATENBLOCK 4
    SYSTEMMELDUNGEN KERNAL <052>
169 REM ----- <035>
170 DATA 13,69,47,65,32,70,69,72,76,69,82,  <084>
    186,13,83,85,67,72,69,32,78,65,67
171 DATA 72,13,32,32,32,160,68,82,85,69,67  <089>
    ,75,69,32,80,76,65,89,84,65,83,84
172 DATA 69,160,68,82,85,69,67,75,69,32,82  <240>
    ,69,67,79,82,68,43,80,76,65,89,84
173 DATA 65,83,84,197,32,32,32,13,13,76,65  <209>
    ,68,69,160,32,32,13,13,83,65,86
    
```

Listing »Betriebssystem-Editor«. Bitte beachten Sie die Eingabehinweise auf Seite 6

```

174 DATA 69,160,32,13,13,80,82,85,69,70,69
,160,13,70,65,78,68,160,32,13,79 <067>
175 DATA 75,141 <229>
176 REM <238>
177 REM DATENBLOCK 5
KERNAL-OLDROUTINE <000>
178 REM ----- <044>
179 DATA 165,43,164,44,133,34,132,35,160,3
,200,177,34,208,251,200,152,24,101 <098>
180 DATA 34,160,0,145,43,165,35,105,0,200,
145,43,136,162,3,230,34,208,2,230 <102>
181 DATA 35,177,34,208,244,202,208,243,165
,34,105,2,133,45,165,35,105,0,133 <079>
182 DATA 46,96 <182>
183 PRINT"FEHLER IN DER DATEN-SUMME " <154>
184 IF A=A1 THEN PRINT"IN BLOCK 1!" <057>
185 IF A=A2 THEN PRINT"IN BLOCK 2!" <124>
186 IF A=A3 THEN PRINT"IN BLOCK 3!" <191>
187 IF A=A4 THEN PRINT"IN BLOCK 4!" <003>
188 IF A=A5 THEN PRINT"IN BLOCK 5!" <070>
189 PRINT"BITTE UEBERPRUEFEN.":END <161>
190 REM <252>
191 REM INIT-ROUTINE ROM-RAM <169>
192 REM ----- <058>
193 POKE 53280,0:POKE 53281,0:PRINT CHR$(1
47)CHR$(152) <000>
194 PRINT"RAM WIRD AKTIVIERT, BITTE 60 SEC
WARTEN!" <010>
195 FOR X=40960 TO 49152:POKE X,PEEK(X):NE
XT <017>
196 FOR X=57344 TO 65535:POKE X,PEEK(X):NE
XT <070>
197 GOTO 234 <053>
198 REM <004>
199 REM FARBZUSAMMENSTELLUNG <212>
200 REM ----- <066>
201 SYS 58648 <094>
202 PRINT"FARBEN BEIBEHALTEN ?" <142>
203 GET F$:IF F$=""THEN 203 <152>
204 IF F$="J"THEN 213 <102>
205 INPUT"ZEICHENFARBE ... ";Z <082>
206 INPUT"RAHMENFARBE .... ";R <035>
207 INPUT"HINTERGRUNDFARBE ";H <253>
208 POKE 58677,Z:POKE 60633,R:POKE 60634,H <051>
209 SYS 58648:GOTO 202 <031>
210 REM <016>
211 REM OLD-BEFEHL EINRICHTEN <003>
212 REM ----- <078>
213 PRINT CHR$(147)"SOLL DIE FUNKTIONSTAST
E F8 MIT DEM " <057>
214 PRINT"OLD-BEFEHL BELEGT WERDEN (J/N)" <016>
215 GET F$:IF F$=""THEN 215 <229>
216 IF F$="J"THEN 218 <194>
217 GOTO 224 <057>
218 INPUT"KERNAL STANDARD/HYPRA 1{17SPACE}
KERNAL HYPRA-PERFECT {2SPACE}2{2SPACE}"
;K <197>
219 ON K GOTO 220,221 <222>
220 POKE 64346,54:POKE 64347,48:GOTO 224 <170>
221 POKE 64469,83:POKE 64470,121:POKE 6447
1,54:POKE 64472,52:POKE 64473,54 <144>
222 POKE 64474,48:POKE 64475,56:POKE 64476
,13:POKE 64477,136:POKE 64778,255 <062>
223 POKE 64479,255 <045>
224 PRINT CHR$(147)"SOLL DER OLD-BEFEHL AL
S BASIC-DIREKT-" <253>
225 PRINT"BEFEHL VERWENDET WERDEN (J/N)" <060>
226 GET F$:IF F$=""THEN 226 <177>
227 IF F$="J"THEN 229 <229>
228 GOTO 121 <220>
229 POKE 40972,95:POKE 40973,252 <085>
230 POKE 41118,79:POKE 41119,76:POKE 41120
,196:GOTO 121 <079>
231 REM <037>
232 REM LADEN EINES VORHANDENEN KERNALS <140>
233 REM ----- <099>
234 POKE 1,55:PRINT CHR$(147)"SOLL EIN BES
TEHENDES BETRIEBSSYSTEM VON" <008>
235 PRINT"DISK BELADEN WERDEN ?" <245>
236 GET F$:IF F$=""THEN 236 <251>
237 IF F$="N"THEN POKE 1,53:GOTO 201 <115>
238 INPUT"FILENAME ";N$ <180>
239 LOAD N$,8,1 <203>

```

© 64'er

Listing »Betriebssystem-Editor« (Schluß)

Mehr Platz auf der Diskette durch »File-Compactor«

Die Zeit, in der man aus Platzgründen nur ein oder zwei Programme auf einer Diskette unterbringen kann, ist vorbei. Mit diesem neuartigen File-Compactor sind Sie in der Lage, nicht nur Grafiken, sondern sogar beliebige Programme um bis zu 55 Prozent zu kürzen.

Das Problem, Programme zu packen, um so Platz auf Disketten zu sparen, ist mit einem einfachen File-Compactor, der sich darauf beschränkt, Folgen gleicher Bytes zu erkennen und zu vereinfachen, noch lange nicht beseitigt. Ein solcher Compactor (zum Beispiel der File-Compactor aus dem 64'er, Ausgabe 7/85) kann mit Maschinensprache-Routinen wenig anfangen, da dort selten mehr als zweimal das gleiche Byte hintereinander vorkommt. Ein guter Programm-Compactor muß daher anders arbeiten.

Ein neuartiges Computerverfahren

Hier ist nun ein Compactor, der sich die geringe Unordnung in Programmen zunutze macht. Dadurch wird wirklich jedes Programm (mit einer Länge von mindestens 25 Blocks) erheblich gekürzt. Es gibt praktisch kein Programm, in dem die verschiedenen Zeichen gleich häufig vorkommen. In Basic-Programmen wird das besonders deutlich, weil dort Leerzeichen, Doppelpunkte, das \$00-Byte oder auch einzelne Buchstaben (wie zum Beispiel das »e«) besonders oft zu finden sind. Es gibt aber auch Zeichen, die in einem Programm nur ganz selten oder überhaupt nicht vorkommen. Diese Tatsache kann man sich zunutze machen, indem man beispielsweise für häufig vorkommende Zeichen, einen 3 Bit langen Code definiert. Jeder kann sich vorstellen, was das für eine Platzersparnis mit sich bringt. Je mehr in dem Programm einige wenige Zeichen dominieren, desto geringer ist der durchschnittliche Informationswert und desto stärker läßt es sich kürzen. Natürlich müssen dabei für die weniger oft vorkommenden Zeichen längere Codes (mit einer Länge von 8 Bit und länger) definiert werden, damit die Codierung eindeutig bleibt.

Zusätzlich wurde in diesem Compactor das Verfahren aus Ausgabe 7/85 (Zählen gleicher Bytfolgen) angewandt, da es besonders bei Spielen mit viel Grafik sehr effizient arbeitet.

Das Programm, das gekürzt werden soll, muß ab dem Basic-Start (Adresse \$0801) in den Speicher geladen werden (also laden mit »LOAD "Name",8«) und darf nicht länger als 48639 Bytes sein. Nach dem Kürzen steht eine SYS-Zeile am Anfang, die den Entpacker startet. Dieser wandelt den komprimierten Code wieder in ein lesbares Programm. Auch Basic-Programme können auf diese Weise von dem Computer bearbeitet werden. Mit »RUN« wird dann zuerst der Entpacker ge-

startet, der dann nach abgeschlossener Umwandlung das eigentliche Basic-Programm ausführt.

Beispiele zur Benutzung

Das Spiel »Fort Apocalypse«, das gerne als Maßstab für die Güte eines Compactors herangezogen wird, belegt normalerweise 153 Blocks auf der Diskette. Nach der Behandlung schrumpft der Platzbedarf auf 74 Blocks. Doch vor allem bei Basic-Programmen zeigt sich die Stärke des Packers: Sie schrumpfen im Schnitt um zirka 20 Prozent, auch wenn sie überhaupt keine Folgen gleicher Bytes enthalten.

1. Eingabe des Listings »File-Compactor« mit dem MSE.
2. Den Verdichter mit »LOAD "...,8« laden und mit »RUN« starten. Der Verschiebelader befördert die Routine automatisch in den richtigen Speicherbereich.
3. Jetzt kann man das zu kürzende Programm nachladen.

4. Den Compactor mit »SYS 51000« aufrufen. Nach getaner Arbeit zeigt er den Grad der Verdichtung (in Prozent) an.
5. Sofern man der Meinung ist, daß sich der Aufwand lohnt, kann man das Programm danach wieder mit »SAVE "...,8« abspeichern.

Modifizierte SAVE-Routine

Wenn Programme gekürzt werden sollen, die nicht in den Basic-Speicher passen und sich bis unter das ROM erstrecken, dann gibt es allerdings Probleme mit dem Abspeichern. Für diesen Fall wurde eine modifizierte SAVE-Routine implementiert. Statt »SAVE "...,8« gibt man in diesem Fall »SYS 50965 "...,8« ein; die Geräteadresse 8 wird in diesem Fall automatisch gesetzt.

(Arndt Brenschede/ev)

```

programm : compactor           0001 0e0f
-----
0001 : 0d 08 c1 07 9e 32 30 36 0c
0009 : 33 3a a2 00 00 00 a2 05 97
0011 : a0 00 b9 31 08 99 15 c7 77
0019 : c8 d0 f7 ee 15 08 ee 18 a2
0021 : 08 ca 10 ee b9 6f 0c 20 f8
0029 : d2 ff c8 c0 5e d0 f5 60 4a
0031 : a5 34 48 a9 cf 85 34 20 72
0039 : d4 e1 a9 36 85 01 a9 08 46
0041 : 85 ba a6 2d a4 2e a9 2b 2b
0049 : 20 d8 ff a9 37 85 01 68 7f
0051 : 85 34 60 a5 2e c9 09 b0 74
0059 : 03 4c 48 b2 c9 c6 b0 f9 74
0061 : a0 00 98 99 00 cd 99 00 2f
0069 : ce 99 00 cf c8 d0 f4 b9 58
0071 : 8b cb 20 d2 ff c8 c0 2e ea
0079 : d0 f5 a5 2d a6 2e 85 fd 41
0081 : 86 fe a9 00 a2 c7 85 fb 67
0089 : 86 fc a9 36 85 01 a2 00 a9
0091 : 20 11 cb 85 fa e8 f0 0a ac
0099 : 20 11 cb 90 37 c5 fa f0 b6
00a1 : f4 24 ca a4 fa 85 fa 20 f6
00a9 : 91 c7 f0 e9 c0 f3 f0 1e 43
00b1 : e0 04 b0 0a 98 48 20 2a a1
00b9 : cb 68 ca d0 f8 60 98 20 ba
00c1 : 2a cb 8a 20 2a cb a9 f3 07
00c9 : 20 2a cb a2 00 60 e0 04 d4
00d1 : b0 ec 90 ee a4 f0 20 91 bf
00d9 : c7 bd b9 cb 9d 01 08 bd e4
00e1 : b9 cc 9d 01 09 e8 d0 f1 87
00e9 : 8a 9d 69 09 e8 d0 f9 8a cf
00f1 : a8 b9 01 cf d9 00 cf 90 ae
00f9 : 36 d0 08 b9 00 ce d9 01 b0
0101 : ce b0 2c b9 00 ce 48 b9 75
0109 : 00 cf 48 b9 69 09 aa b9 37
0111 : 01 ce 99 00 ce b9 01 cf 3e
0119 : 99 00 cf b9 6a 09 99 69 05
0121 : 09 8a 99 6a 09 68 99 01 5f
0129 : cf 68 99 01 ce a2 01 c8 4a
0131 : c0 ff d0 bd ca f0 b8 a9 47
0139 : 37 85 01 a9 00 38 e5 fb fa
0141 : 85 63 a9 c7 e5 fc 85 62 fc
0149 : a2 90 38 20 49 bc a9 91 8a
0151 : 8d 00 cd a9 00 a0 cd 20 03
0159 : 0f bb a2 05 a0 cd 20 d4 32
0161 : bb 20 49 cb 86 fa a4 fa c1
0169 : b9 00 ce 85 63 b9 00 cf 2a
0171 : 85 62 a2 90 38 20 49 bc 05
0179 : a5 61 d0 03 4c ed c8 a9 0e
0181 : 05 a0 cd 20 28 ba a9 57 fb
0189 : a0 00 20 67 b8 20 ca bb 4d
0191 : a9 5c a0 00 20 50 b8 a9 4b
0199 : 91 8d 00 cd a2 0a a0 cd 43
01a1 : 20 d4 bb a9 00 85 f9 8d 7f
01a9 : 01 cd a9 90 38 e5 f9 8d c3
01b1 : 0f cd a9 0f a0 cd 20 a2 31
01b9 : bb a9 0a a0 cd 20 67 b8 c1
01c1 : 46 66 a9 00 a0 cd 20 5b 54
01c9 : bc c9 ff d0 0f a2 00 a0 cb
01d1 : cd 20 d4 bb e6 f9 a5 f9 24
01d9 : c9 10 d0 ce a4 f9 88 b9 68
01e1 : f0 cd 18 69 01 99 f0 cd 27
01e9 : a9 91 38 e5 f9 8d 0f cd 09
01f1 : a9 5c a0 00 20 a2 bb a9 4a
01f9 : 0f a0 cd 20 67 b8 20 c7 1c
0a01 : bb e6 fa f0 03 4c 4b c8 5d
0a09 : a9 91 8d 0f cd a9 0f a0 68

0a11 : cd 20 a2 bb a9 5c a0 00 0e
0a19 : 20 50 b8 a5 61 c9 81 90 d0
0a21 : 5c a9 8f 38 e5 61 a8 a5 94
0a29 : 66 30 32 b9 f0 cd d0 04 34
0a31 : 88 10 f8 60 aa ca 8a 99 6a
0a39 : f0 cd c8 b9 f0 cd aa e8 73
0a41 : 8a 99 f0 cd a2 80 8e 10 16
0a49 : cd 84 f9 a9 90 38 e5 f9 62
0a51 : 8d 0f cd a9 0f a0 cd 20 7c
0a59 : 67 b8 4c 00 c9 c8 b9 f0 db
0a61 : cd d0 05 c0 0f d0 f6 60 04
0a69 : aa ca 8a 99 f0 cd 88 b9 61
0a71 : f0 cd aa e8 8a 99 f0 cd e4
0a79 : c8 a2 00 f0 c9 a2 00 bd de
0a81 : 69 09 a8 8a 99 00 ce a9 12
0a89 : 00 9d 00 cf e8 d0 f0 20 6b
0a91 : 49 cb 8e 10 cd a9 00 85 9b
0a99 : f9 85 fa a4 fa b9 f0 cd 85
0aa1 : 85 63 a9 00 85 62 a2 01 59
0aa9 : 38 20 49 bc a9 90 38 e5 a7
0ab1 : fa 8d 0f cd a9 0f a0 cd 21
0ab9 : 20 28 ba a9 57 a0 20 80 8c
0ac1 : f0 88 20 ca bb 20 9b bc 8a
0ac9 : a5 fa 0a a8 a5 65 aa 99 e6
0ad1 : d0 cd c8 a5 64 99 d0 cd 61
0ad9 : d0 09 8a d0 06 a5 f9 d0 02
0ae1 : 0c f0 02 e6 f9 e6 fa a5 d1
0ae9 : fa c9 10 d0 00 a0 00 84 df
0af1 : f8 84 f9 a4 f8 b9 f0 cd fb
0af9 : 48 18 79 bf cd 99 c0 cd ec
0b01 : 68 a8 c0 00 f0 10 88 a6 ec
0b09 : f9 bd 69 09 aa a5 f8 9d 53
0b11 : 00 cf e6 f9 db ec e6 f8 f4
0b19 : a5 f8 c9 10 d0 d5 a9 36 7d
0b21 : 85 01 ac cf cd 88 98 18 00
0b29 : 69 6a 85 fd 8d 4b 08 a9 8f
0b31 : 08 85 ff 69 01 85 fe 8d 7c
0b39 : 47 08 a0 00 84 f7 84 f8 b8
0b41 : b1 fb e6 fb d0 02 e6 fc d5
0b49 : aa a5 fc c9 7 90 06 a5 a2
0b51 : fb c9 01 b0 5a bd 00 cf ba
0b59 : a8 bd c0 ce 38 f9 bf cd a8
0b61 : 84 fa c0 08 b0 08 85 f8 e7
0b69 : 98 69 08 a8 90 02 85 f7 ec
0b71 : c0 0f f0 07 06 f7 26 f8 81
0b79 : c8 d0 f5 a5 fa f0 10 0a 67
0b81 : a8 b9 ce cd 65 f7 85 f7 8f
0b89 : b9 cf cd 65 f8 85 f8 06 f5
0b91 : f7 26 f8 26 f9 c6 ff d0 16
0b99 : 10 a9 08 85 ff a0 00 a5 81
0ba1 : f9 91 fd e6 fd 00 02 e6 fb
0ba9 : fe c6 fa 10 e2 30 8b a6 f6
0bb1 : ff 0e 08 f0 0e a5 f9 0a 4b
0bb9 : ca d0 fc 91 fd e6 fd d0 0d
0bc1 : 02 e6 fe a9 37 85 01 a2 14
0bc9 : c8 bd be cd 9d 3a 09 bd 7a
0bd1 : cf cd 9d 4a 09 ca d0 f1 46
0bd9 : a2 10 ca bd f0 cd f0 ca 25
0be1 : 8e fb 08 a4 fe a5 fd 85 24
0be9 : 63 d0 01 88 c6 63 98 38 60
0bf1 : e9 08 85 62 a2 90 20 49 4e
0bf9 : bc 20 0c bc a4 2e 8c a0 8f
0c01 : 08 a5 2d 8d 9e 08 85 63 0e
0c09 : d0 01 88 c6 63 98 38 e9 04
0c11 : 08 85 62 a2 90 20 49 bc 71
0c19 : 20 14 bb a5 fd a6 fe 85 03
0c21 : 2d 86 2e 20 e2 ba 20 e2 6b
0c29 : ba 4c d7 bd a0 0a a5 fd 53
0c31 : d0 02 c6 fe c6 fd b1 fd b3

0c39 : a4 fe c0 08 18 d0 04 a4 ef
0c41 : fd f0 01 38 60 a4 fb d0 ba
0c49 : 02 c6 fc c6 fb a0 00 91 ae
0c51 : fb a8 b9 00 ce 18 69 01 64
0c59 : 99 00 ce b9 00 cf 69 00 01
0c61 : 99 00 cf 60 a2 0a a9 00 1b
0c69 : 95 56 ca d0 fb 60 41 55 68
0c71 : 46 52 55 46 20 3a 20 53 f9
0c79 : 59 53 35 31 30 30 30 d4 4f
0c81 : 55 45 42 45 52 4c 41 4e db
0c89 : 47 45 20 50 52 47 27 53 28
0c91 : 20 53 50 45 49 43 48 45 72
0c99 : 52 4e 20 3a 20 53 59 53 0a
0ca1 : 35 30 39 36 35 0d 12 46 94
0ca9 : 4c 45 58 49 42 4c 45 20 b3
0cb1 : 43 4f 44 45 20 43 4f 4d 4a
0cb9 : 50 41 43 54 4f 52 20 41 90
0cc1 : 2e 42 52 c6 2e c3 43 48 6b
0cc9 : 45 44 45 d0 52 45 53 54 69
0cd1 : 20 3a 20 25 0e 08 ca a8 59
0cd9 : 9e 32 30 36 35 20 46 43 57
0ce1 : 43 00 00 00 a0 00 b9 69 e8
0ce9 : 07 99 00 cd b9 69 08 99 b1
0cf1 : 00 ce b9 69 09 99 00 cf f1
0cf9 : c8 d0 eb 4c c2 cd 78 a0 6b
0d01 : ff 84 fb a9 c6 85 fc a9 56
0d09 : 36 85 01 8d 20 d0 c8 a5 eb
0d11 : 2d f0 02 c6 2e c3 4d a6 1b
0d19 : 2e e0 30 d0 04 c9 01 f0 52
0d21 : 0f b1 2d 91 fb a5 a5 f0 d0 05
0d29 : 02 c6 fc c6 fb 4c d3 cd b3
0d31 : a2 08 a9 01 86 2e 85 2d ac
0d39 : 84 ff 20 50 ce c9 f3 d0 7c
0d41 : 27 20 50 ce aa 86 fa c9 c5
0d49 : 44 b0 04 a9 f3 d0 03 20 ee
0d51 : 50 ce a0 00 91 2d c8 c6 64
0d59 : fa d0 f9 9e 18 65 2d 85 b9
0d61 : 2d 90 02 e6 2e 4c 34 ce e7
0d69 : a0 00 91 2d e6 2d f0 f3 97
0d71 : a9 01 a2 60 e4 2e d0 c2 d8
0d79 : c5 2d d0 be a9 37 85 01 4d
0d81 : a9 fe 8d 20 d0 58 20 59 14
0d89 : a6 4c ae a7 a2 ff 86 f7 2a
0d91 : 86 f8 e8 a9 01 85 fe a9 8e
0d99 : 7f 85 fd c6 ff 10 10 e6 c1
0da1 : fb d0 02 e6 fc a9 07 85 a6
0da9 : ff a0 00 b1 fb 85 f9 06 0e
0db1 : f9 b0 0a a4 fe a5 fd 39 a1
0db9 : f7 00 99 f7 00 8a 0a a8 e3
0dc1 : a5 f7 38 f9 e2 ce a5 f8 dc
0dc9 : f9 e3 ce 90 0e e0 0d f0 77
0dd1 : 0a e8 38 66 fd b0 c4 c6 30
0dd9 : fe f0 bc 8a f0 0f a5 f7 de
0de1 : 38 f9 e0 ce 85 f7 a5 f8 c8
0de9 : f9 e1 ce 85 f8 a4 fe f0 ca
0df1 : 07 a5 f8 85 f7 88 84 f8 81
0df9 : a5 fd 4a 90 07 46 f8 66 95
0e01 : f7 4c be ce bd d2 ce 65 20
0e09 : f7 a8 b9 00 cf 60 e0 01 48
    
```

Listing »File-Compactor«
(Bitte mit dem MSE eingeben. Beachten Sie die Hinweise auf Seite 8)

Bildschirm-Effekte leicht-gemacht

Mit dieser Basic-Erweiterung können Sie in beliebigen Bildschirmanschnitten Spezialeffekte wie Blinken, Invertieren oder Auffüllen mit beliebigen Zeichen einfach erzeugen.

Diese kleine Maschinensprache-Routine (»Screen«) kann in nahezu allen Programmen Verwendung finden, um zum Beispiel fehlerhafte Eingaben durch Blinken zu markieren oder um Eingabefelder zu löschen. Auch für die Gestaltung von Titeln bietet sich »Screen« an. Folgende Routinen sind vorhanden:

- (0) Invertieren des gewählten Bereichs
- (1) Reversdarstellung eines Bildschirm-Bereichs
- (2) Normaldarstellung (Gegenstück zu (1))
- (3) Auffüllen eines Bereichs mit beliebigen Zeichen

Das Programm (Listing 1 und 2) kann auf zwei verschiedene Arten aufgerufen werden. Entweder starten Sie es mit »SYS (Anfangsadresse+36)« oder Sie benutzen statt dessen das Ausrufezeichen »!«. Der erste Weg bietet sich an, wenn schon eine andere Befehlserweiterung das »!« benutzt. Die korrekte Syntax des »!«-Befehls entnehmen Sie bitte Tabelle 1.

Das Programm ist frei im Speicher verschiebbar; die beiden notwendigen Vektoren für die Befehlserweiterung werden in Zeile 195-200 automatisch angepaßt. Möchte man lieber mit dem SYS-Befehl arbeiten, dann können die Zeilen 165-225

```

155 SA= 49152 <034>
160 : <136>
165 HB=INT((SA+11)/256):REM VEKTOR HIGH <116>
170 LB=(SA+11)-HB*256:REM VEKTOR LOW <203>
175 : <151>
180 REM*** VERBIEGEN DES VEKTORS ***** <016>
185 REM*** + NEUE CHRGET ROUTINE ***** <095>
190 : <166>
195 POKE SA ,162:POKE SA+1,16 <244>
200 POKE SA+2,160:POKE SA+3,16 <063>
205 FOR I=SA+4 TO SA+35:READ X:POKE I,X:NE
XT <078>
210 DATA 142, 8, 3,140, 9, 3, 96 <066>
215 REM--- <245>
220 DATA 32,115, 0,201, 33,240, 6 <145>
225 DATA 32,121, 0, 76,231,167, 32,115,
0, 32, 36,192, 32,121, 0, 76,231,167 <062>
230 : <206>
235 REM*** EIGENTLICHES PROGRAMM ***** <023>
240 : <216>
245 S=0:FOR I=SA+36 TO SA+217 <252>
250 READ X:POKE I,X:S=S+X <037>
255 NEXT <009>
260 IF S<>23004 THEN PRINT"*** DATA FEHLER
***":STOP <081>
265 DATA 216, 32,250,174, 32,158,183,134,
89, 32,241,183,134, 90, 32,241,183,134 <109>
270 DATA 91, 32,241,183,134, 92, 32,247,17
4, 32,241,183,134,253, 32,241,183,134 <244>
275 DATA 254,165, 90, 56,229, 89, 24,105,
1,133,252,165, 92, 24,105, 1,133, 92 <040>
280 DATA 56,229, 91,133,251,169,128,133, 9
3,173,136, 2,133, 88,162, 0,134, 87 <164>
285 DATA 166, 89,240, 16,165, 87, 24,105,
40,133, 87,165, 88,105, 0,133, 88,202 <025>
290 DATA 208,240,173, 17,208, 42,144,250,1
69, 11,141, 17,208,184,166,253,164, 91 <238>
295 DATA 177, 87, 80, 26,145, 87,200,196,
92,208,245,165, 87, 24,105, 40,133, 87 <004>
300 DATA 165, 88,105, 0,133, 88,198,252,2
08,226, 80, 38,177, 87,224, 0,240, 14 <246>
305 DATA 224, 1,240, 20,224, 2,240, 20,2
24, 3,240, 8,208, 18, 36, 93,240, 6 <243>
310 DATA 208, 8,165,254, 80,200, 9,128,
80,196, 41,127, 80,192,169, 27,141, 17 <100>
315 DATA 208, 96 <195>
    
```

Listing 1. »Screen« als DATA-Lader. Bitte beachten Sie die Eingabehinweise auf Seite 6.

```

100 REM***** <150>
105 REM** DEMO-PROGRAMM ZU '!SCREEN' ** <078>
110 REM** ** <110>
115 REM** 28.6.85 STEFAN WAGNER ** <214>
120 REM** HOEHNSTR.102 ** <011>
125 REM** 5090LEVERKUSEN 3 ** <059>
130 REM** ** <130>
135 REM***** <185>
140 : <116>
145 !(0,24,0,39),3,235 <186>
150 FOR I=1 TO 24 <173>
155 : <131>
160 !(I,I,0,39-I),2,0 <111>
165 : FOR K=1 TO 200-8*I <090>
170 : NEXT K <090>
175 : <151>
180 NEXT I <008>
185 : <161>
190 PRINT CHR$(147) <219>
195 : <171>
200 FOR F= 0 TO 3 STEP 3 <002>
205 :FOR V=20 TO 0 STEP-5 <122>
210 : FOR I= 2 TO 8 <198>
215 : FOR J=10 TO 12+I <224>
220 : C=RND(0)*255 <232>
225 : !(2,I,10,J),F,C <161>
230 : FOR X=1 TO V <169>
235 : NEXT X <108>
240 : NEXT J <144>
245 : NEXT I <133>
250 :NEXT V <091>
255 NEXT F <059>
260 : <238>
265 PRINT CHR$(19)" {2SPACE}ABSCHALTEN MIT
STOP" <071>
270 : <248>
275 !(0,24,0,39),0,0 <068>
280 FOR X=1 TO 150:NEXT:GOTO 275 <181>
    
```

Listing 2. Demonstrationsprogramm zu »Screen«. Bitte beachten Sie die Eingabehinweise auf Seite 6.

Tabelle 1. Die Syntax des neuen »!«-Befehls.

Initialisieren: SYS SA
 Syntax:
 !(VZ,BZ,VS,BS),F,C
 VZ = Von Zeile
 BZ = Bis Zeile
 VS = Von Spalte
 BS = Bis Spalte
 F = Funktion (0-3)
 0 = Invers
 1 = Revers
 2 = Normal
 3 = Füllen
 C = Füllcode
 (= Bildschirmcode)

| Adresse | Inhalt | Erklärung |
|---------|--------|----------------------------|
| 87 | BL | Bildschirmadresse LowByte |
| 88 | BH | Bildschirmadresse HighByte |
| 89 | ABZE | ab Zeile |
| 90 | BISZE | bis Zeile |
| 91 | ABSP | ab Spalte |
| 92 | BISSP | bis Spalte |
| 93 | TEST | enthält #128 |
| 251 | ANZSP | Anzahl Spalten |
| 252 | ANZZE | Anzahl Zeilen |
| 253 | FKT | Funktionscode |
| 254 | CODE | Füllcode |

Tabelle 2. Die verwendeten Speicherstellen in der Zero-Page

ersatzlos entfallen, in diesem Fall findet das Programm auch im Kassettenpuffer Platz. Tabelle 2 zeigt die Belegung der Zero-Page durch das Programm.

Zu bemerken ist noch, daß das Programm aus Platzgründen nicht auf ungültige Zeichen oder Spaltenwerte testet. In Ihren Programmen müssen Sie daher darauf achten, daß der Bildschirmbereich eingehalten wird, da sich sonst Probleme ergeben können. Das Demonstrationsprogramm in Listing 2 sieht zwar unscheinbar aus, erzeugt aber überraschende Effekte.

(Stefan Wagner/ev)

Aus eins mach zwei

Mit dieser kleinen Routine »verdoppeln« Sie Ihren Computer. Sie sind damit in der Lage, zwei Maschinenprogramme praktisch gleichzeitig laufen zu lassen.

Die Routine selbst belegt etwa 190 Byte und ist im freien 4-KByte-RAM-Bereich (ab Adresse 49152) abgelegt. Außerdem benötigt sie zwei Pages (Page 193 und 194) als Zwischenspeicher. Zusätzlich wird noch der Basic-Interpreter und das Betriebssystem in das darunterliegende RAM kopiert. Schließlich »verbiegt« die Routine den Interruptvektor.

Das Programm ermöglicht das »gleichzeitige« Betreiben zweier Maschinenroutinen. Jede dieser Routinen läuft nun nur noch mit der halben Geschwindigkeit.

Die Routine wird folgendermaßen gehandhabt: Die Startadresse des ersten Maschinenprogramms, das man benutzen möchte, wird in Adresse 49158 (Lo-Byte) und 49159 (Hi-Byte) abgelegt. Analog verfährt man mit der Startadresse des zweiten Maschinenprogramms, die in Adresse 49156 und 49157 gePOKEt werden muß. Der Befehl SYS 49168 aktiviert die Routine, der Befehl POKE 1,55 (oder in Maschinensprache LDA #55; STA 1) stellt sie wieder ab. Diese Umschaltung, sei es Basic oder Assembler, wird am besten durch das erste Programm durchgeführt, da nur von dort wieder im Basic-Modus weiter gearbeitet werden kann.

Basic und Maschinensprache gleichzeitig

Grundvoraussetzung für das Koppel-Programm ist, daß die beiden zu betreibenden Routinen in Maschinensprache geschrieben sind. Auf den ersten Blick mag es daher so aussehen, als wären die Einsatzmöglichkeiten dieser Routine sehr beschränkt. Das Gegenteil ist jedoch der Fall. Da auch das Basic-ROM »Maschinensprache« ist, kann man ohne weiteres mit RTS zum Basic-Modus zurückkehren, anstatt die erste Maschinenroutine zu beginnen. Um dies zu erreichen, ersetzt man die Zahl 108 (JMP indirekt) in Speicherstelle 49187 durch eine 96 (RTS-Code). Jetzt kann man in Basic programmieren und seine Basic-Programme auch entsprechend abarbeiten lassen, während eine zweite Maschinensprache-Routine quasi »im Hintergrund« abgearbeitet wird. Denkbare Anwendungen sind das Spielen von Melodien oder die automatische Bewegung von Sprites während des Programm-Laufs.

Nachdem Sie »Multiprogramming« (Listing 1) mit RUN gestartet haben, sehen Sie ein merkwürdiges Geflimmer auf dem Bildschirm. Dies ist kein Fehler im Programm (dessen dokumentierter Source-Code in Listing 2 zu finden ist), sondern eine kleine Demonstration. Eine Maschinenroutine ändert fortwährend die Rahmen-, eine zweite die Hintergrundfarbe. Beide kleinen Programme (Adresse 49350 und 49375) laufen jetzt quasi »gleichzeitig« ab. Die Anwendungsbereiche der Basic-Routine sind sehr vielfältig. Sie reichen von der Ausgabe auf zwei Drucker über die »gleichzeitige« Abfrage von Port A und B bis hin zu komplexeren Realtime-Aufgaben. Da gerade bei einem solchen Programm die Dokumentation sehr wichtig ist, drucken wir sowohl den Source-Code (Listing 2) als auch das Flußdiagramm (siehe Bild) mit ab. (Thomas Weinbrenner/ev)

Daten für das Programm:

| | | |
|-------|-----|---------------------------------------|
| 49152 | xxx | ;y-Register |
| 49153 | xxx | ;x-Register |
| 49154 | xxx | ;Akkumulator |
| 49155 | 33 | ;Statusregister |
| 49156 | xxx | ;Startadresse 2. Programm (Lo und Hi) |
| 49157 | xxx | ;Startadresse 1. Programm (Lo und Hi) |
| 49158 | xxx | ;Startadresse 1. Programm (Lo und Hi) |
| 49159 | xxx | ;Stapelzeiger 1. Programm |
| 49160 | xxx | ;Stapelzeiger 2. Programm |
| 49161 | 249 | |

Listing 2. Source-Code des Programms
»Aus eins mach zwei«.

Aktivierungsroutine:

| | | | | |
|-------|-----|---------|--------------|-------------------------|
| 49168 | SEI | | 120 | |
| 49169 | JSR | 49191 | 32,39,192 | |
| 19172 | LDA | #53 | 169, 53 | ;RAM-Konfiguration |
| 49174 | LDY | #98 | 160, 98 | |
| 49176 | LDX | #192 | 162, 192 | |
| 49178 | STA | 1 | 133, 1 | |
| 49180 | STY | 65534 | 140, 254,255 | |
| 49183 | STX | 65535 | 142, 255,255 | |
| 49186 | CLI | | 88 | |
| 49187 | JMP | (49158) | 108, 6, 192 | ;Sprung zum 1. Programm |

Hier werden die Zeiger auf die eigene Interruptroutine gesetzt. Danach wird das erste Programm angesprochen.

| | | | | |
|-------|-----|--------|-----------|---------------------------------------|
| 49191 | LDY | #0 | 160, 0 | ;Werte setzen zum Basic-ROM kopieren |
| 49193 | STY | 38 | 132, 38 | |
| 49195 | DEY | | 136 | |
| 49196 | STY | 39 | 132, 39 | |
| 49198 | JSR | 49212 | 32,60,192 | ;Basic-ROM kopieren |
| 49201 | LDY | #191 | 160, 191 | ;Werte setzen zum Kernal-ROM kopieren |
| 49203 | STY | 39 | 132, 39 | |
| 49205 | JSR | 49212 | 32,60,192 | ;Kernal-ROM kopieren |
| 49208 | JSR | 49231 | 32,79,192 | ;Daten setzen |
| 49211 | RTS | | 96 | |
| 49212 | LDX | #32 | 162, 32 | ;32 Pages kopieren |
| 49214 | LDY | #0 | 160, 0 | |
| 49216 | DEY | | 136 | |
| 49217 | LDA | (38),Y | 177, 38 | |
| 49219 | STA | (38),Y | 145, 38 | |
| 49221 | CPY | #0 | 192, 0 | |
| 49223 | BNE | 49216 | 208, 247 | |
| 49225 | DEC | 39 | 198, 39 | |
| 49227 | DEX | | 202 | |
| 49228 | BNE | 49214 | 208, 240 | |
| 49230 | RTS | | 96 | |

Diese Routine kopiert das Betriebssystem und den Basic-Interpreter in das »darunter«-liegende RAM.

| | | | | |
|-------|-----|---------|--------------|--|
| 49231 | LDA | #0 | 169, 0 | ;Entscheidungsfaktor |
| 49233 | STA | 2 | 133, 2 | |
| 49235 | LDY | #5 | 160, 5 | ;Daten für späteren Interruptrücksprung in Page 194 setzen |
| 49237 | LDA | 49152,Y | 185, 0,192 | |
| 49240 | STA | 49914,Y | 153, 250,194 | |
| 49243 | DEY | | 136 | |
| 49244 | BPL | 49237 | 16, 247 | |
| 49246 | RTS | | 96 | |

Interruptroutine:

| | | | | |
|-------|-----|---------|------------|---|
| 49250 | PHA | | 72 | ;x-y-Register und Akku auf den Stack retten |
| 49251 | TXA | | 138 | |
| 49252 | PHA | | 72 | |
| 49253 | TYA | | 152 | |
| 49254 | PHA | | 72 | |
| 49255 | LDA | 2 | 165, 2 | |
| 49257 | BNE | 49298 | 208, 39 | |
| 49259 | LDY | #255 | 160, 255 | ;Stack in Page 193 kopieren |
| 49261 | LDA | 256,Y | 185, 0,1 | |
| 49264 | STA | 49408,Y | 153, 0,193 | |
| 49267 | DEY | | 136 | |
| 49268 | CPY | #255 | 192, 255 | |
| 49270 | BNE | 49261 | 208, 245 | |

```

49272 LDA 49664,Y 185, 0,194 ;Page 194 in den
49275 STA 256,Y 153, 0,1 Stack kopieren
49278 DEY 136
49279 CPY #255 192, 255
49281 BNE 49272 208, 245
49283 TSX 186 ;Stapelzeiger ver-
49284 STX 49160 142, 8,192 tauschen
49287 LDX 49161 174, 9,192
49290 TXS 154
49291 LDY #1 169,1 ;Entscheidungs-
49293 STA 2 133, 2 fakten ändern
49295 JMP 49334 76, 182,192

49298 LDY #255 160, 255 ;Stack in Page
49300 LDA 256,Y 185, 0,1 194 kopieren
49303 STA 49664,Y 153, 0,194
49306 DEY 136
49307 CPY #255 192, 255
49309 BNE 49300 208, 254
49311 LDA 49408,Y 185, 0,193 ;Page 193 in den
49314 STA 256,Y 153, 0,1 Stack kopieren
49317 DEY 136
49318 CPY #255 192, 255
49320 BNE 49311 208, 245
49322 TSX 186 ;Stapelzeiger ver-
49323 STX 49161 142, 9,192 tauschen
49326 LDX 49160 174, 8,192
49329 TXS 154
49330 LDA #0 169, 0 ;Entscheidungs-
49332 STA 2 133, 2 Faktor ändern

49334 PLA 104 ;x-y-Register und
49335 TAY 168 ;Akku vom Stack
49336 PLA 104 holen
49337 TAX 170
49338 PLA 104
49339 JMP 65352 76, 72,255
    
```

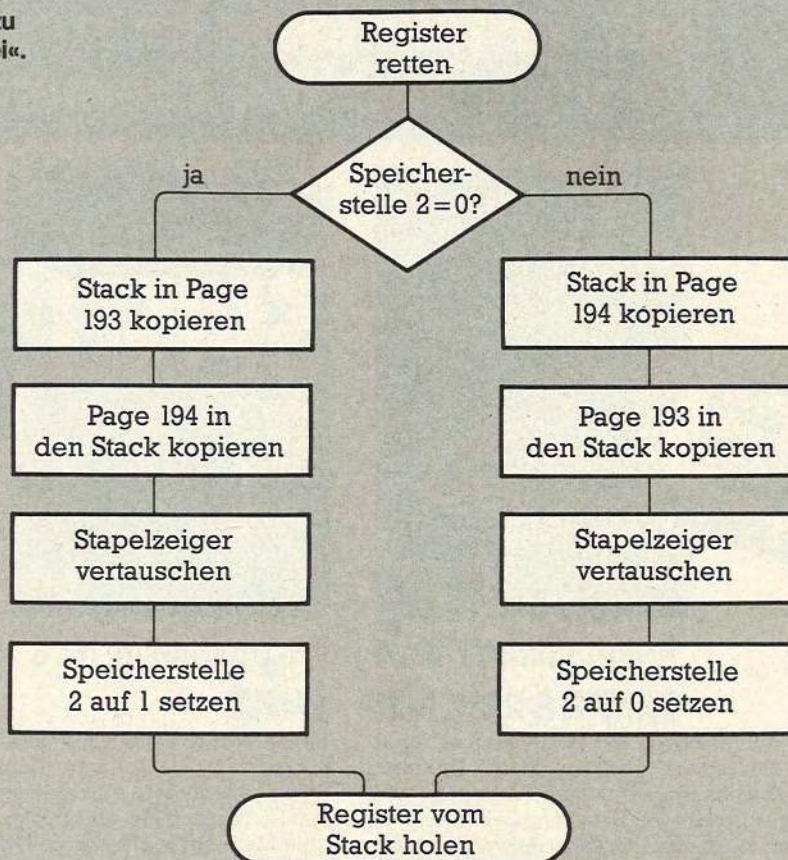
programm : mupro 0801 091d

```

0801 : 0c 08 c1 07 9e 20 32 30 77
0809 : 36 32 00 00 00 a0 00 b9 d1
0811 : 1c 08 99 00 c0 c8 d0 f7 1d
0819 : 4c 10 c0 00 00 00 21 df e2
0821 : c0 c6 c0 f0 f9 54 45 58 9a
0829 : 20 41 53 78 20 27 c0 a9 5f
0831 : 35 a0 62 a2 c0 85 01 8c f9
0839 : fe ff 8e ff ff 58 4c c6 5c
0841 : c0 ea a0 00 84 26 88 84 43
0849 : 27 20 3c c0 a0 bf 84 27 10
0851 : 20 3c c0 20 4f c0 60 a2 85
0859 : 20 a0 00 88 b1 26 91 26 b9
0861 : c0 00 d0 f7 c6 27 ca d0 c7
0869 : f0 60 a9 00 85 02 a0 05 e9
0871 : b9 00 c0 99 fa c2 88 10 96
0879 : f7 60 54 2e 57 48 8a 48 ee
0881 : 98 48 a5 02 d0 27 a0 ff b0
0889 : b9 00 01 99 00 c1 88 c0 68
0891 : ff d0 f5 b9 00 c2 99 00 29
0899 : 01 88 c0 ff d0 f5 ba 8e d3
08a1 : 08 c0 ae 09 c0 9a a9 01 60
08a9 : 85 02 4c b6 c0 a0 ff b9 9e
08b1 : 00 01 99 00 c2 88 c0 ff 0c
08b9 : d0 f5 b9 00 c1 99 00 01 dd
08c1 : 88 c0 ff d0 f5 ba 8e 09 45
08c9 : c0 ae 08 c0 9a a9 00 85 fc
08d1 : 02 68 a8 68 aa 68 4c 48 ee
08d9 : ff ff ff ff ff ff ff ff d8
08e1 : ff a9 00 8d 20 d0 a9 01 98
08e9 : 8d 20 d0 4c c6 c0 ff ff b6
08f1 : ff ff ff ff ff ff ff ff f0
08f9 : ff ff a9 00 8d 21 d0 a9 db
0901 : 01 8d 21 d0 4c df c0 ff f2
0909 : ff ff ff ff ff ff ff ff 08
0911 : ff ff ff ff ff ff ff ff 10
0919 : ff ff ff 60 5f b4 4c 64 b9
    
```

Listing 1 des Programms »Aus eins mach zwei«.
Bitte beachten Sie die Eingabehinweise auf Seite 8.

Das Flußdiagramm zu
»Aus eins mach zwei«.



Leichte Eingabe

Die Benutzerfreundlichkeit eines Programmes hängt von der Behandlung der Eingabe beziehungsweise Fehleingabe ab. Dieses Programm beseitigt viele damit verbundene Probleme.

Das aufgeführte Programm (Listing 1) 2, das eine echte Hilfe beim Erstellen von benutzerfreundlichen Programmen darstellt, ist eine Basic-Eingaberoutine, das heißt, es ist in Basic geschrieben und ersetzt den normalen INPUT-Befehl.

Es ist in der Handhabung sehr einfach gehalten und dabei noch recht kurz.

Es werden folgende Variablen benutzt:

| | | |
|------|---|--|
| IN\$ | : | Übergabevariable für den einzulesenden String |
| LT | : | maximale Länge des einzulesenden Strings |
| ZT | : | Hilfsvariable zur Simulation des Cursors |
| ZC | : | Hilfsvariable zur Simulation des Cursors |
| ZL | : | aktuelle Länge des einzulesenden Strings |
| Z\$ | : | zuletzt gelesenes Zeichen |
| Z | : | ASCII-Code des gelesenen Zeichens und Laufvariable |

Variablenliste zur Eingabe-Routine

Der Benutzer muß dabei nur IN\$ und LT vorbesetzen (siehe Variablenliste). In der Routine wird als erstes IN\$ ausgegeben. Dies hat den Vorteil, daß Standardantworten vorgegeben werden können. Danach werden die für die Darstellung des Cursors notwendigen Variablen vorbesetzt.

Dann wartet die Routine auf die Eingabe eines Zeichens. Der Cursor wird währenddessen dadurch simuliert, daß ein Leerzeichen laufend invertiert wird.

Komfortabler als »INPUT«

Nachdem ein Zeichen eingegeben wurde, wird zuerst geprüft, ob überhaupt noch Zeichen angenommen werden dürfen. Ist dies der Fall, wird untersucht, ob das Zeichen ein Steuerzeichen ist.

Falls es keins ist, wird das Zeichen ausgegeben und an den Übergabestring angehängt. Dann wird das Hochkommaflag zurückgesetzt, um weiterhin eine vernünftige Eingabe zu gewährleisten.

Falls das eingegebene Zeichen ein Steuerzeichen ist, gibt es nur vier Möglichkeiten:

1. Es ist ein »RETURN«, dann ist die Eingabe beendet.
2. Es ist ein »DELETE«, dann wird der Übergabestring um ein Zeichen vermindert.
3. Es ist ein »SHIFT RETURN«, dann wird der gesamte String gelöscht, und der Cursor an den Anfang des Eingabefeldes positioniert.
4. Es ist ein anderes Steuerzeichen, dann wird es ignoriert.

Bei den Punkten 1. bis 3. wird auch noch getestet, ob überhaupt schon ein Zeichen eingegeben wurde. Ist dies nicht der Fall, dann haben auch diese drei Steuerzeichen keine Wirkung.

(Willi Burmeister/rg)

64ER ONLINE

Weitere aktuelle 64'er-Sonderhefte mit ausführlichen Informationen zu ausgewählten Themen:



Jetzt überall im Zeitschriftenhandel

Programmheft Nr. 4 GRAFIK & DRUCKER

Randvoll mit Informationen: Rund 28 Listings der Spitzenklasse. Darunter Top-Listings zur räumlichen Darstellung von Körpern aus beliebigen Betrachtungsrichtungen. Weiters: Prüfsummenlistings, Drucker-Anwendung, Basic-Erweiterung, Hardcopy-Routinen, Zeichengenerator, Grundlagen, Tips & Tricks.



Nur noch bis zum 30.9.85 erhältlich

Programmheft Nr. 3 SPIELE

Heiße Listings für alle Spiele-Fans: Sportspiele, Schießspiele, Denkspiele, Spielegeneratoren, Abenteuerspiele, Brettspiele, Taktikspiele, Geschicklichkeitsspiele und eine Marktübersicht aller in Deutschland erhältlichen professionellen Spiele bringen alles, was das Herz der Spiele-Fans höher schlagen läßt.

```

10 REM ***** <119>
12 REM*(25SPACE)* <066>
14 REM*(5SPACE)ILLI JURMEISTER(4SPACE)* <011>
16 REM*(25SPACE)* <070>
18 REM*(5SPACE)DLZKRUGWEG 30(6SPACE)* <133>
22 REM*(5SPACE)2390 ELENBURG(6SPACE)* <221>
24 REM*(25SPACE)* <078>
26 REM*(5SPACE)TEL.: 0461/92164(4SPACE)* <190>
28 REM*(25SPACE)* <082>
30 REM*(9SPACE)BASIC -(9SPACE)* <052>
32 REM*(5SPACE)EINGABEROUTINE(6SPACE)* <246>
34 REM*(25SPACE)* <088>
36 REM***** <219>
40 GOTO 1000 <236>
50 REM ***** <159>
60 REM * BEGIN DER ROUTINE * <097>
70 REM ***** <179>
100 PRINT IN$: ; ZT=TI : ZC=4 : IF (LT<1)
    THEN LT=1 <168>
110 GET Z$: IF Z$<>"" THEN 140 <108>
120 IF (ZT<=TI) THEN PRINT MID$(" (SPACE,UP
    ,DOWN,RVSON,SPACE,RVOFF)",ZC,3);"(LEFT
    )"; ; ZC=5-ZC : ZT=TI+25 <110>
130 GOTO 110 <074>
140 Z=ASC(Z$) : ZL=LEN(IN$) <150>
150 IF (ZL<LT) AND ((Z AND 127)>31) THEN I
    N$=IN$+Z$ : PRINT Z$: ; POKE 212,0 : G
    OTO 110 <109>
160 PRINT "(SPACE,LEFT)"; <002>
170 IF (Z=13) AND (ZL>0) THEN RETURN <001>
180 IF (Z=20) AND (ZL>0) THEN IN$=LEFT$(IN
    $,ZL-1) : PRINT "(LEFT)"; ; GOTO 110 <027>
190 IF (Z=141) AND (ZL>0) THEN FOR Z=1 TO
    ZL : PRINT "(LEFT,SPACE,LEFT)"; ; NEXT
    : IN$="" <089>
200 GOTO 110 <144>
210 REM ***** <063>
220 REM * ENDE DER ROUTINE * <187>
230 REM ***** <093>
1000 PRINT"(CLR)" <226>
1010 PRINT"BASIC-EINGABEROUTINE BEISPIEL" <187>
1020 PRINT : PRINT <210>
1030 IN$="VORBESETZUNG" : LT=15 <201>
1040 GOSUB 100 <224>
1050 PRINT "(2DOWN)"; : PRINT IN$ <246>
    
```

@ 64'er

Listing 1. »Eingaberoutine«. Bitte bei der Eingabe Seite 6 beachten.

- POKE649,0 — Sperre der Eingabe per Tastatur
- POKE649,10 — Auflösen der Sperre
- POKE657,128 — Sperre der Zeichensatzumschaltung mit SHIFT+C=
- POKE657,0 — auflösen der Sperre
- POKE768,143 — Reset nach Programmbeendigung
- POKE768,145 — Sperre der Fehlermeldungen im Direkt-Modus
- POKE788,81 — schnell blinkender Cursor
- POKE788,56 — Cursor aus
- POKE788,56 — RUN/STOP blockiert + Cursor aus + interne Uhr bleibt stehen
- POKE56325,x — Cursor-Geschwindigkeit verändern (1=sehr schnell...255=langsam)
- POKE56325,0 — langsames LIST

(Rainer Sprehe/tr)

Die Rechengenauigkeit des C 64

In den 64'er Ausgaben 9 und 10 des letzten Jahres wurde im Leserforum die mangelhafte Rechengenauigkeit des C 64 angesprochen.

Die folgende Zeile zeigt das deutlich:

```
5 A=A+0,01 : ?A : GOTO 5
```

Die Ungenauigkeit hat schon manches Mathematikprogramm scheitern lassen. Der C 64 ist eben nicht in der Lage, eine Zahl mehrmals um eine Kommazahl zu vergrößern. Man kann diese Ungenauigkeit aber mit folgender Rundungsformel umgehen:

```
5 A=A+0,01 : A=INT (A*100+0.5)/100 : ?A : GOTO 5
```

Die »100« in der Formel bewirkt ein Runden auf zwei Nachkommastellen. Wenn man auf drei Stellen hinter dem Komma runden will, muß man die »100« durch eine »1000« ersetzen.

(Andreas Schilling/tr)

Tips für Anfänger

List-Bremse

Dieses kleine Programm hält das Listen einer Zeile so lange an, wie eine Taste gedrückt ist. Dazu verändert es zwei Zeiger, die normalerweise auf die Ausgabe der Schlüsselwörter zeigen. Es wird durch die Adresse 203 geprüft, ob eine Taste gedrückt ist. Falls ja, so wird mit der Ausgabe weitergemacht. Wenn nein, so wird noch einmal abgefragt.

Das Programm ist mit SYS 49152 zu starten.

(Jörg Blaumeiser/tr)

```

PROGRAMM : LIST-STOP.OBJ C000 C016
C000 : A9 0B 8D 06 03 A9 C0 8D EF
C008 : 07 03 60 48 A5 CB C9 40 12
C010 : F0 FA 68 4C 1A A7 31 EA 9A
    
```

Und wieder einige POKE's für den C 64:

- POKE768,61 — Fehlermeldungen unterdrücken
- POKE768,139 — kein Unterdrücken der Fehlermeldungen mehr
- POKE19,64 — Sperre des Fragezeichens bei INPUT
- POKE19,0 — Auflösen der Sperre
- POKE157,128 — LOADING, SEARCHING usw. erscheinen auch im Programm

Buntes Directory

Bestimmt hat sich der eine oder andere schon Gedanken darüber gemacht, wie er sein Directory bunt gestalten könnte. Nichts einfacher als das:

Man muß das Programm nur folgendermaßen speichern: SAVE" (SHIFT-SPACE) (FARBTASTE) TEST",8

Nach dem Laden des Directory erscheint der Programm-Name »TEST« in der zuvor eingegebenen Farbe. Selbstverständlich kann man nach der Farb-Taste auch noch die Revers-Taste drücken.

Beim Laden des Programmes selbst ist jedoch darauf zu achten, daß man nach dem LOAD-Befehl ebenfalls die gleiche Tastenkombination eintippt. In unserem Beispiel würde die Programm-Zeile folgendermaßen aussehen:

```
LOAD" (SHIFT-SPACE) (FARBTASTE) TEST",8
```

Am nützlichsten erscheint mir diese Form des Speicherns bei Dateien. Man könnte die verschiedenen File-Typen farblich kennzeichnen, wodurch eine bessere Übersicht und somit auch eine gewisse Ordnung entsteht.

Die REL-Files könnten zum Beispiel gelb, die SEQ-Files revers und weiß gekennzeichnet werden.

Folgende Zeile müßte im Programm so aussehen:

```
OPEN 1,8,1,"(SHIFT-SPACE) (FARBTASTE) TEST,S,W": CLOSE 1
```

Gleichzeitig entsteht so ein gewisser Programm-Schutz: Die abgespeicherten Programme stehen nämlich wie folgt im Directory:

```
67 " "TEST PRG
```

Auf normalem Weg sind diese Programme nicht zu laden. Es muß eben wie schon erwähnt, das SHIFT-SPACE dazu.

(Dieter Plaum/tr)

Formeln perfekt gedruckt

Endlich ist es möglich, mit dem Epson FX-80 mathematische Sonderzeichen zu drucken.

Mit diesem Programm können mathematische Formeln in korrekter Schreibweise zu Papier gebracht werden. Es ist lauffähig auf dem Commodore 64 mit Epson-Drucker FX-80 und Görlitz-Interface.

Das Programm beinhaltet folgende Funktionen:

1. Umwandlung des Basic-Zeichens $\frac{\text{a}}{\text{b}}$ in einen echten Bruch (siehe Bild 1, Beispiel 1).
2. Umwandlung des Basic-Zeichens a^{b} in echte Potenzschreibweise (siehe Bild 1, Beispiel 2).
3. Umwandlung der Basic-Funktion $\text{EXP}(\text{X})$ in echte Exponentialschreibweise (siehe Bild 1, Beispiel 3).
4. Umwandlung der Basic-Funktion $\text{SQR}(\text{X})$ in echte Wurzel-schreibweise (siehe Bild 1, Beispiel 4).
5. Mischung von Bruch-, Wurzel-, Exponential- und Potenz-schreibweise (siehe Bild 1, Beispiel 5 bis 18).
6. Beachtung der Klammerregeln (siehe Bild 1, Beispiel 8 bis 12).
7. Ausgabe des Bruchstrichs exakt auf der Höhe der ganzrationalen Funktionsterme.
8. Einbindung in andere Programme ohne weiteres möglich, es muß nur die Funktion und die Druckposition übergeben werden.
9. Nach Ende des Druckvorgangs befindet sich der Druckkopf auf der gleichen Höhe wie zu Beginn, genau hinter der ausgedruckten Funktion. Es können also in einer Reihe ohne Probleme mehrere Funktionen ausgegeben werden.

Soweit eine kurze Übersicht über die Fähigkeiten des Programms. Kommen wir nun zu einer Beschreibung des Programmaufbaus (Tabelle 1).

In den Zeilen 120 bis 1150 befindet sich ein Beispielprogramm. Es bewirkt einen Ausdruck der Beispiele aus Bild 1.

Die Funktionen werden definiert und in G\$(1) bis G\$(18) festgehalten. In der sich anschließenden Schleife (Zeilen 1040 bis 1130) werden nun nacheinander die Funktionen einmal in Basic-Schreibweise und durch den Aufruf des Unterprogramms (Zeile 1120) in exakter Schreibweise ausgegeben. Vor dem Aufruf des Unterprogramms werden in den beiden Variablen BB und F\$ die zu übergebenden Informationen definiert (siehe Tabelle 2). Zum einen ist das der Abstand vom linken Rand (hier 55) und zum anderen die Funktion selber (hier von G\$(1) bis G\$(18)).

Das Beispielprogramm soll nur einen kurzen Überblick über die vielfältigen Möglichkeiten von »Formel-Druck« geben und demonstrieren, wie die vom Programm benötigten Informationen zu übergeben sind. Wichtig ist in diesem Fall auch Zeile 120. In F\$(..) werden im Programm später die einzelnen Terme der zu druckenden Funktion abgelegt. Ist eine Funktion sehr lang, kann es leicht zu einer größeren Anzahl an Einzeltermen als 11 kommen. Um allen Eventualitäten vorzubeugen, ist es besser, F\$ gleich genügend groß zu dimensionieren. Hier wurde der Wert 50 gewählt, aber auch eine kleinere Zahl reicht meistens aus und kann vom Anwender nach eigenem Willen eingesetzt werden.

Ab Zeile 50000 beginnt nun das eigentliche Programm, die Routine zum Umwandeln der Basic-Funktionen mit sich anschließendem Ausdruck.

In Zeile 50170 ist zunächst einmal das sehr häufig benutzte Unterprogramm zum Auslesen eines Zeichens (in T\$) aus der Funktion (FU\$) angelegt. In den Zeilen bis 50400 werden nun die nötigen Anfangsbedingungen eingestellt. Das ist erstens in Zeile 50180 die Festlegung des Zeichensatzes für den Drucker (Groß-/Kleinschrift mit Zeilenabstand von sieben Punkten), in der für das Görlitz-Interface typischen Form. Zweitens die Definition der Sonderzeichen zum Wurzelausdruck (das $\frac{\text{a}}{\text{b}}$ wird zum Strich über den Zeichen und das $\sqrt{\text{a}}$ zum Wurzelzeichen). Schließlich werden noch die benötigten Variablen festgelegt und der Fettdruck eingeschaltet.

In den Zeilen 50420 bis 50720 werden die beiden Basic-Befehls-worte EXP und SQR aus der Funktion eliminiert und durch für das Programm später verständliche Befehle ersetzt (Zeile 50550 und 50710). Durch die Verminderung von L wird die sich verkürzende Funktionslänge berücksichtigt.

Anschließend folgt der Test auf gebrochenrationale Funktion (Zeile 50780 bis 50850). Tritt in der zu druckenden Funktion ein Divisionszeichen auf, wird in Zeile 50980 verzweigt und die Funktion in ganzrationale und gebrochenrationale Terme zerlegt. Zu diesem Zweck werden zum einen die vor dem Bruchstrich (Zeile 50980 bis 51070) und zum anderen die hinter dem Bruchstrich (Zeile 51080 bis 51180) auftretenden Zeichen betrachtet. Als Abbruchkriterium gilt jeweils das erste $\frac{\text{a}}{\text{b}}$, $\sqrt{\text{a}}$ oder a^{b} -Zeichen, das nicht mehr zum Zähler oder Nenner gehört. In den Zeilen 51190 bis 51260 findet dann die Zuweisung der entsprechenden Einzelelemente in F\$(SC) statt. Hierbei wird SC bei jedem neu hinzukommenden Term um eins erhöht. Nun wird in Zeile 51260 wieder nach 50780 zurückgesprungen und der restliche Funktionsteil untersucht. Das geschieht so lange, bis alle Divisionszeichen abgearbeitet sind. Dann wird der restliche Teil in F\$(SC) abgelegt und zur Hauptschleife des Programms zurückgekehrt (Zeile 50910 bis 50920).

Diese Schleife beginnt in Zeile 51320. Mit jedem Durchlauf der Schleife wird ein in F\$(SC) abgelegter Term aufgerufen, die jeweilige Druckposition festgelegt und der Fettdruck gesetzt.

Jeder einzelne Term wird in den Zeilen 51450 bis 51530 auf gebrochenrationale Funktion geprüft. Liegt eine vor, muß diese in den Zeilen 51650 bis 51720 in Zähler und Nenner zerlegt werden. Während dieser Prozedur werden auch gleich die überflüssigen Klammern entfernt (Zeile 51680 und Zeile 51720). Handelt es sich um einen ganzrationalen Funktionsteil, wird dieser in Zeile 51590 im Zähler abgelegt und der Nenner für spätere Kontrollen mit einem Leerstring belegt.

Um hinterher beim Ausdruck ein exaktes Druckbild zu erhalten (Bruchstrich mit der richtigen Länge und Zähler und Nenner zentriert übereinander) muß die Verkürzung des Druckbildes durch Potenzausdruck und Wegfall von einigen Zeichen durch die Eliminierung von SQR berücksichtigt werden. Das Ausmaß der Verkürzung wird in den Zeilen 51780 bis 52020 berechnet. Bei jeder auftretenden Potenz verkürzt sich die Länge des Ausdrucks um 0,5 und durch Wegfall des Zeichens $\sqrt{\text{a}}$ um eine Druckposition. Dies wird jeweils für Zähler und Nenner getrennt in Zeile 51970 beziehungsweise 51980 festgehalten.

Die Verkürzung durch den Wegfall der Klammern beim Wurzelausdruck wird jeweils in Zeile 52010 beziehungsweise 52020 berücksichtigt.

Die erhaltene Gesamtverkürzung ist in WO beziehungsweise WU enthalten. Diese wird beim Umwandeln von Zähler und Nenner auf die gleiche Länge (Zeile 52030 bis 52050) mit einberechnet. Dadurch wird eine ordentliche Bruchschreibweise gewährleistet.

Ab Zeile 52110 beginnt die Druckroutine. Als erstes wird der Zähler jedes einzelnen Terms ausgegeben. Hierbei wird ein ganzrationaler Funktionsteil als Zähler mit dem Nenner als Leerstring betrachtet. Handelt es sich um einen gebrochenra-

| | |
|---------------|---|
| 120 - 1150 | Beispielprogramm |
| 50180 - 50400 | Einstellung der Grundbedingungen; Definition der Sonderzeichen für den Wurzelausdruck |
| 50460 - 50560 | Umwandlung der Basic-Funktion EXP in die später zum Ausdruck benötigte Form |
| 50620 - 50720 | Umwandlung der Basic-Funktion SQR in die später zum Ausdruck benötigte Form |
| 50780 - 50920 | Test auf gebrochenrationale Funktion zur Faktorzersetzung |
| 50980 - 51260 | Zerlegung in ganzrationale und gebrochenrationale Funktionsterme |
| 51320 - 51390 | Schleife zum Ausdruck der einzelnen Terme |
| 51450 - 51530 | Test auf gebrochenrationale Funktion im Term |
| 51590 | Einstellen der Bedingungen für ganzrationalen Term |
| 51650 - 51720 | Zerlegung der gebrochenrationalen Funktion in Zähler und Nenner |
| 51780 - 52050 | Berechnung der Verkürzung von Zähler und Nenner durch Potenz- und Wurzelausdruck |
| 52110 - 52230 | Zähler drucken |
| 52280 - 52360 | Bruchstrich drucken |
| 52420 - 52520 | Nenner drucken |
| 52580 - 52850 | Potenzausdruck |
| 52910 - 53090 | Wurzelausdruck |
| 53150 - 53170 | Ende |

Tabelle 1. Programmbeschreibung »Formeldruck«

| | |
|---------|--|
| F\$ | Funktion, muß vom Hauptprogramm übergeben werden und liegt hinterher unverändert vor. |
| BB | Abstand vom linken Rand, muß vom Hauptprogramm übergeben werden und enthält hinterher die Position des Funktionsendes. |
| T\$ | Einzelne Zeichen der Funktion |
| FU\$ | Funktion während der Bearbeitung |
| F\$(SC) | Funktionsterme nach der Zerlegung |
| SC | Anzahl der Funktionsterme |
| L | Länge der Funktion |
| E\$ | Vergleichsvariable für EXP und SQR |
| WR | Zeichenzahl im Zähler (bei der Zerlegung) |
| WV | Zeichenzahl im Nenner (bei der Zerlegung) |
| KL,EK | Anzahl der offenen Klammerpaare |
| LO | Länge des Zählers |
| LU | Länge des Nenners |
| OB\$ | Zähler |
| UB\$ | Nenner |
| Z | Markierung für Zähler (Z = 1) oder Nenner (Z = 2) |
| WO | Stellenverkürzung für den Zähler durch Potenz- und Wurzelausdruck |
| WU | Stellenverkürzung für den Nenner durch Potenz- und Wurzelausdruck |
| AA | Bruchstrichlänge |
| M | Stellenverkürzung für ganzrationale Funktion durch Potenz- und Wurzelausdruck |
| WW | Markierung für Wurzelausdruck |
| WM | Stellenverkürzung bei Wurzelausdruck |

E;G;I;J;O;Q;R;T;U;Y Zählvariablen

Tabelle 2. Variablenliste »Formeldruck«

Ansonsten kommt es zur Ausgabe des Bruchstrichs (Zeile 52280 bis 52360). Die Länge des Bruchstrichs wird in AA festgehalten und aus den oben genannten Verkürzungswerten WO und WU und der Länge des Zählers und Nenners berechnet (Zeile 52300 bis 52310).

Als letztes wird dann der Nenner ausgedruckt. Dies erfolgt auf die gleiche Weise wie beim Zähler. Auch hier wird am Ende ein Rücktransport des Papiers durchgeführt. Das Ende eines gebrochenrationalen Funktionsteils ergibt sich aus der Länge des Bruchstrichs und wird in BB gespeichert.

In den Zeilen 52580 bis 53090 stehen die beiden Unterprogramme zum Potenz- beziehungsweise Wurzelausdruck. Die Potenzausdruckroutine beginnt in Zeile 52580. Es wird die Superscript- und die Schmalschrift eingestellt. Dann wird jedes dem Potenzsymbol folgende Zeichen überprüft, ob es noch im Exponenten steht oder nicht. Als Abbruchkriterium gelten die gängigen mathematischen Kriterien (Zeilen 52650 bis 52705). In den Zeilen 52720 bis 52750 wird überprüft, ob die Potenz unter der Wurzel steht, in diesem Fall ist WW = 1, und der entsprechende Wurzelstrich wird ausgegeben. Nach Abarbeitung aller in Frage kommenden Zeichen wird wieder die Normalschrift und der Fettdruck eingeschaltet und zurück ins Hauptprogramm gesprungen (Zeile 52810 bis 52850). Die Variable M (Zeile 52760) dient zur Berechnung der Verkürzung des Terms durch den Potenzausdruck und wird bei ganzrationalen Funktionen berücksichtigt.

In der Wurzelausdruckroutine läuft im Prinzip das gleiche ab. Sie beginnt in Zeile 52910 und druckt zuerst das Wurzelsymbol und über alle folgenden Zeichen den Wurzelstrich. Zur Ausgabe dieses Strichs wird der Druckkopf jeweils um ein Zeichen zurück und einen Matrixpunkt nach oben bewegt und anschließend das Sonderzeichen ausgegeben (Zeile 53020 bis 53060).

Den Abschluß des Programms bilden die Zeilen 53150 bis 53170. Hier wird der Drucker normiert, das heißt in den Einschaltzustand zurückversetzt, alle offenen Kanäle geschlossen und der Epson-Modus mit automatischem Zeilenvorschub eingestellt.

(Patrick Bödeker/rg)

| | | |
|---------------|------------------------------------|---|
| Beispiel 1 : | 1/X | $\frac{1}{x}$ |
| Beispiel 2 : | X^2 | x^2 |
| Beispiel 3 : | EXP(X) | $e^{(x)}$ |
| Beispiel 4 : | SQR(X) | \sqrt{x} |
| Beispiel 5 : | SQR(EXP(X)) | $\sqrt{e^{(x)}}$ |
| Beispiel 6 : | EXP(SQR(X)) | $e^{(\sqrt{x})}$ |
| Beispiel 7 : | SQR(X^(3*X+4)) | $\sqrt{x^{(3*x+4)}}$ |
| Beispiel 8 : | 1/(7*X^6+3*X^2) | $\frac{1}{7*x^6+3*x^2}$ |
| Beispiel 9 : | (3*X^2-2*X+4)/(5*X^3+2*X-12) | $\frac{3*x^2-2*x+4}{5*x^3+2*x-12}$ |
| Beispiel 10 : | 3*X^2-2*X+4/(5*X^3+2*X-12) | $\frac{3*x^2-2*x+4}{5*x^3+2*x-12}$ |
| Beispiel 11 : | (3*X^2-2*X+4)/5*X^3+2*X-12 | $\frac{3*x^2-2*x+4}{5} * x^3+2*x-12$ |
| Beispiel 12 : | 3*X^2-2*X+4/5*X^3+2*X-12 | $3*x^2-2*x+\frac{4}{5} * x^3+2*x-12$ |
| Beispiel 13 : | (3*X^(7*X+4))/(2*X^2-7*X+2) | $\frac{3*x^{(7*x+4)}}{2*x^2-7*x+2}$ |
| Beispiel 14 : | EXP(X)+SQR(X^4+X^3+X^2)/7*X^3+14 | $e^{(x)} + \frac{\sqrt{x^4+x^3+x^2}}{7} * x^3+14$ |
| Beispiel 15 : | (X^3*SQR(3*X+4)+SQR(X^4+3))/7 | $\frac{x^3*\sqrt{3*x+4}+\sqrt{x^4+3}}{7}$ |
| Beispiel 16 : | EXP(X)/SQR(X^3+4*X)+SQR(3*X)-1/X^2 | $\frac{e^{(x)}}{\sqrt{x^3+4*x}} + \sqrt{3*x} - \frac{1}{x^2}$ |
| Beispiel 17 : | 3/4*(7*X^2-3*X)+(7*X^2)/(14*X^3) | $\frac{3}{4} * (7*x^2-3*x) + \frac{7*x^2}{14*x^3}$ |
| Beispiel 18 : | SQR(X^23-4)+1/(3*X)-SQR(2*X^3)+1/X | $\sqrt{x^23-4} + \frac{1}{3*x} - \sqrt{2*x^3} + \frac{1}{x}$ |

Bild 1. Druckbeispiele »Formeldruck« (verkleinert)

tionalen Funktionsteil, wird in Zeile 52130 das Papier um eine Position rückwärts transportiert, um später den Bruchstrich auf Höhe der ganzrationalen Funktion ausgeben zu können. Zum Ausdruck wird jedes Zeichen des Zählers untersucht, ob es sich um ein Potenz- oder Wurzelsymbol handelt. Ist dies der Fall, werden die entsprechenden Routinen zum Potenz- beziehungsweise Wurzelausdruck ausgeführt. Alle anderen Zeichen werden sofort gedruckt.

Handelt es sich beim ausgegebenen Term um eine ganzrationale Funktion, wird ein Rückwärtstransport des Papiers von einer Zeile durchgeführt und das Ende des Terms in BB festgehalten, damit die nächsten Funktionsteile direkt im Anschluß gedruckt werden können.

Tips & Tricks

```

10 REM ***** <060>
20 REM * <069>
30 REM * FORMEL-DRUCK * <085>
40 REM * * <089>
50 REM * 1985 BY PATRICK BOEDEKER * <160>
60 REM * NEUSTR. 11 * <230>
70 REM * 4350 RECKLINGHAUSEN * <169>
80 REM * * <129>
90 REM ***** <140>
100 : <076>
110 : <086>
120 DIM F$(50) <153>
130 DIM G$(18) <174>
140 G$(1)="1/X" <057>
150 G$(2)="X^2" <061>
160 G$(3)="EXP(X)" <224>
170 G$(4)="SQR(X)" <026>
180 G$(5)="SQR(EXP(X))" <171>
190 G$(6)="EXP(SQR(X))" <198>
200 G$(7)="SQR(X^(3*X+4))" <236>
210 G$(8)="1/(7*X^6+3*X^2)" <245>
220 G$(9)="(3*X^2-2*X+4)/(5*X^3+2*X-12)" <224>
230 G$(10)="3*X^2-2*X+4/(5*X^3+2*X-12)" <089>
240 G$(11)="(3*X^2-2*X+4)/5*X^3+2*X-12" <148>
250 G$(12)="3*X^2-2*X+4/5*X^3+2*X-12" <185>
260 G$(13)="(3*X^(7*X+4))/(2*X^2-7*X+2)" <085>
270 G$(14)="EXP(X)+SQR(X^4+X^3+X^2)/7*X^3+
14" <222>
280 G$(15)="(X^SQR(3*X+4)+SQR(X^4+3))/7" <222>
290 G$(16)="EXP(X)/SQR(X^3+4*X)+SQR(3*X)-1
/X^2" <220>
300 G$(17)="3/4*(7*X^2-3*X)+(7*X^2)/(14*X^
3)" <110>
310 G$(18)="SQR(X^23-4)+1/(3+X)-SQR(2*X^3)
+1/X" <010>
1000 OPEN 1,4,8 <100>
1010 PRINT#1,"{30SPACE}BASIC{20SPACE}"; <207>
1020 PRINT#1,"{4SPACE}FORMEL-DRUCK" <135>
1030 PRINT#1:CLOSE 1 <252>
1040 FOR A=1 TO 18 <114>
1050 OPEN 1,4 <113>
1060 PRINT#1:PRINT#1 <025>
1070 PRINT#1,CHR$(27)" T" <211>
1080 PRINT#1,"{2SPACE}BEISPIEL";A;" : {3SPA
CE}"; <152>
1090 PRINT#1,CHR$(27)" G"; <230>
1100 PRINT#1,G$(A); <133>
1110 CLOSE 1 <105>
1120 BB=55:F=G$(A):GOSUB 50000 <058>
1130 NEXT A <134>
1140 OPEN 1,4,4:PRINT#1,CHR$(12):CLOSE 1 <014>
1150 END <136>
1160 : <120>
1170 : <130>
50000 REM ***** <227>
50010 REM * <018>
50020 REM * AUSGABE VON MATHEMATI- * <254>
50030 REM * SCHEN FORMELN IN BRUCH-, * <153>
50040 REM * WURZEL- UND POTENZ- * <208>
50050 REM * SCHREIBWEISE AUF DEN * <202>
50060 REM * DRUCKER FX-80 MIT * <071>
50070 REM * GOERLITZ INTERFACE. * <056>
50080 REM * * <088>
50090 REM * DIE FUNKTION MUSS IN F$, * <158>
50100 REM * DER ABSTAND VOM LINKEN * <085>
50110 REM * RAND IN BB UEBERGEHEN * <179>
50120 REM * WERDEN * <058>
50130 REM ***** <101>
50140 : <075>
50150 : <085>
50160 GOTO 50180 <149>
50170 T$=MID$(FU$,T,1):RETURN <112>
50180 OPEN 1,4:OPEN 2,4,10,"7" <164>
50230 : <167>
50240 REM ***** <216>
50250 REM * ZEICHENDEFINITION * <029>
50260 REM ***** <236>
50270 : <207>
50280 PRINT#1,CHR$(27);": ";CHR$(0);CHR$(0)
;CHR$(0); <225>
50290 PRINT#1,CHR$(27);"&";CHR$(0);"%";CH
R$(139); <249>
50300 PRINT#1,CHR$(0);CHR$(4);CHR$(4);CHR$(
3);CHR$(2);CHR$(4); <061>
50310 PRINT#1,CHR$(8);CHR$(16);CHR$(32);CH
R$(64);CHR$(128);CHR$(0); <047>
50320 PRINT#1,CHR$(27);"%";CHR$(0);"&&";CH
R$(139); <119>
50330 PRINT#1,CHR$(128);CHR$(128);CHR$(128
);CHR$(128);CHR$(128);CHR$(128); <083>
50340 PRINT#1,CHR$(128);CHR$(128);CHR$(128
);CHR$(128);CHR$(128);CHR$(0); <078>
50350 PRINT#1,CHR$(27);"%";CHR$(1);CHR$(0)
; <047>
50360 FU$=F$ <063>
50370 SC=0:Y=0 <047>
50390 PRINT#1,CHR$(27);"E"; <019>
50400 L=LEN(FU$) <033>
50410 : <091>
50420 REM ***** <010>
50430 REM * EXPONENTIALFUNKTION * <186>
50440 REM ***** <032>
50450 : <133>
50460 I=1 <116>
50470 T=I:GOSUB 50170 <094>
50480 IF T$="E"THEN 50520 <184>
50490 I=I+1 <180>
50500 IF I<=L THEN 50470 <223>
50510 GOTO 50620 <021>
50520 FOR R=I TO I+2 <170>
50530 T=R:GOSUB 50170 <043>
50540 E$=E$+T$:NEXT R <027>
50550 IF E$="EXP"THEN FU$=LEFT$(FU$,I-1)+
E$+RIGHT$(FU$,L-I-2):L=L-1 <146>
50560 E$="":GOTO 50490 <066>
50570 : <253>
50580 REM ***** <074>
50590 REM * WURZELFUNKTION * <209>
50600 REM ***** <094>
50610 : <037>
50620 I=1 <020>
50630 T=I:GOSUB 50170 <254>
50640 IF T$="S"THEN 50680 <048>
50650 I=I+1 <084>
50660 IF I<=L THEN 50630 <103>
50670 GOTO 50780 <087>
50680 FOR R=I TO I+2 <074>
50690 T=R:GOSUB 50170 <205>
50700 E$=E$+T$:NEXT R <189>
50710 IF E$="SQR"THEN FU$=LEFT$(FU$,I-1)+
Z$+RIGHT$(FU$,L-I-2):L=L-2 <212>
50720 E$="":GOTO 50650 <070>
50730 : <159>
50740 REM ***** <205>
50750 REM *GEBROCHENRATIONALE FUNKTION* <142>
50760 REM ***** <225>
50770 : <199>
50780 I=1:KL=0 <007>
50790 T=I:GOSUB 50170 <160>
50800 IF T$="( "THEN KL=KL+1 <046>
50810 IF T$=")"THEN KL=KL-1 <056>
50820 IF T$="/"AND KL>0 THEN 50840 <061>
50830 IF T$="/"THEN 50980 <014>
50840 I=I+1 <020>
50850 IF I<=L THEN 50790 <091>
50860 : <033>
50870 REM ***** <107>
50880 REM * GANZRATIONALE FUNKTION * <075>
50890 REM ***** <127>
50900 : <073>
50910 SC=SC+1:F$(SC)=FU$ <111>
50920 GOTO 51320 <097>
50930 : <103>
50940 REM ***** <249>
50950 REM * FAKTORZERLEGUNG * <051>
50960 REM ***** <015>
50970 : <145>
50980 WR=I:KL=0 <154>
50990 WR=WR-1:T=WR:GOSUB 50170 <221>
51000 IF T$=")"THEN KL=KL+1 <120>
51010 IF T$="( "THEN KL=KL-1 <130>
51020 IF T$="+"AND KL=0 THEN 51070 <018>
51030 IF T$="-"AND KL=0 THEN 51070 <029>
51040 IF T$="*"AND KL=0 THEN 51070 <166>

```

Listing »Formeldruck«. Bitte beachten Sie die Eingabe-
hinweise auf Seite 6.

| | | | |
|---|-------|---|-------|
| 51050 IF WR=1 THEN WR=0:GOTO 51070 | <216> | 51810 IF Z=1 THEN T%=MID\$(OB\$,I,1) | <089> |
| 51060 GOTO 50990 | <098> | 51820 IF Z=2 THEN T%=MID\$(UB\$,I,1) | <228> |
| 51070 WR=I-WR-1 | <082> | 51830 IF T%="↑"THEN 51870 | <109> |
| 51080 WV=I | <122> | 51831 IF Z=1 AND T%="%"THEN WO=WO+2 | <083> |
| 51090 WV=WV+1:T=WV:GOSUB 50170 | <098> | 51832 IF Z=2 AND T%="%"THEN WU=WU+2 | <143> |
| 51100 IF T%="("THEN KL=KL+1 | <092> | 51840 IF Z=1 AND I<LO THEN 51800 | <137> |
| 51110 IF T%=")"THEN KL=KL-1 | <102> | 51850 IF Z=2 AND I<LU THEN 51800 | <203> |
| 51120 IF T%="+"AND KL=0 THEN WV=WV-1:GOTO 51170 | <172> | 51860 GOTO 52000 | <069> |
| 51130 IF T%="-"AND KL=0 THEN WV=WV-1:GOTO 51170 | <183> | 51870 IF Z=1 THEN T%=MID\$(OB\$,I,1) | <149> |
| 51140 IF T%="*"AND KL=0 THEN WV=WV-1:GOTO 51170 | <064> | 51880 IF Z=2 THEN T%=MID\$(UB\$,I,1) | <032> |
| 51150 IF WV=L THEN 51170 | <253> | 51890 IF T%="("THEN KL=KL+1 | <120> |
| 51160 GOTO 51090 | <181> | 51900 IF T%=")"AND KL=0 THEN 51800 | <123> |
| 51170 WV=WV-I | <072> | 51910 IF T%=")"AND KL>0 THEN KL=KL-1 | <190> |
| 51180 IF KL<0 THEN WV=WV-1 | <201> | 51920 IF T%="+"AND KL=0 THEN 51800 | <144> |
| 51190 SC=SC+1 | <215> | 51930 IF T%="-"AND KL=0 THEN 51800 | <155> |
| 51200 F\$(SC)=LEFT\$(FU\$,I-1-WR) | <119> | 51940 IF T%="*"AND KL=0 THEN 51800 | <036> |
| 51210 FU%=RIGHT\$(FU\$,L-I+WR+1) | <253> | 51945 IF T%="/"AND KL=0 THEN 51800 | <171> |
| 51220 SC=SC+1 | <247> | 51950 I=I+1 | <114> |
| 51230 F\$(SC)=LEFT\$(FU\$,WR+WV+1) | <173> | 51960 IF T%=")"THEN 52000 | <241> |
| 51240 F\$(SC+1)=RIGHT\$(FU\$,L-I-WV) | <217> | 51970 IF Z=1 THEN WO=WO+.5:IF T%="↑"THEN W | <105> |
| 51250 FU%=RIGHT\$(FU\$,L-I-WV):L=LEN(FU\$) | <068> | 0=WO+.5 | |
| 51260 GOTO 50780 | <171> | 51980 IF Z=2 THEN WU=WU+.5:IF T%="↑"THEN W | <054> |
| 51270 : | <191> | U=WU+.5 | |
| 51280 REM ***** | <222> | 51990 GOTO 51870 | <123> |
| 51290 REM * SCHLEIFE * | <011> | 52000 IF Z=1 THEN 51790 | <053> |
| 51300 REM ***** | <242> | 52030 IF LO-WO+1.5<LU-WU THEN OB\$=" "+OB\$+ | |
| 51310 : | <231> | " ":LO=LO+2:D=1:GOTO 52030 | <210> |
| 51320 Y=Y+1 | <057> | 52040 IF D=1 THEN D=0:GOTO 52060 | <193> |
| 51330 IF Y>SC THEN 51350 | <062> | 52050 IF LU-WU+1.5<LO-WO THEN UB\$=" "+UB\$+ | |
| 51340 IF F\$(Y)="("THEN 51320 | <247> | " ":LU=LU+2:GOTO 52050 | <159> |
| 51350 FU%=F\$(Y) | <163> | 52060 : | <219> |
| 51360 CLOSE 4:OPEN 4,4,4 | <105> | 52070 REM ***** | <109> |
| 51370 PRINT#4,CHR\$(27);CHR\$(108);CHR\$(BB); | <240> | 52080 REM * ZAEHLER DRUCKEN * | <230> |
| 51380 PRINT#4,CHR\$(27);"E"; | <017> | 52090 REM ***** | <129> |
| 51390 CLOSE 2:OPEN 2,4,10,"7" | <079> | 52100 : | <003> |
| 51400 : | <065> | 52110 M=0 | <242> |
| 51410 REM ***** | <214> | 52120 PRINT#1,CHR\$(27);"E"; | <227> |
| 51420 REM * BRUCH ? * | <206> | 52130 IF UB\$<>"("THEN PRINT#4,CHR\$(27);CHR\$(| |
| 51430 REM ***** | <234> | 106);CHR\$(22);:CLOSE 3:OPEN 3,4,10, | |
| 51440 : | <105> | "7" | <165> |
| 51450 L=LEN(FU\$) | <067> | 52140 O=1:Z=1 | <143> |
| 51460 I=1:KL=0 | <181> | 52150 T=O:T%=MID\$(OB\$,T,1) | <005> |
| 51470 T=I:GOSUB 50170 | <078> | 52160 IF T%="↑"THEN E=0:GOSUB 52580:O=E:M= | |
| 51480 IF T%="("THEN KL=KL+1 | <220> | M+1 | <120> |
| 51490 IF T%=")"THEN KL=KL-1 | <230> | 52170 IF T%="%"THEN Q=0:GOSUB 52910:O=Q | <047> |
| 51500 IF T%="/"AND KL>0 THEN 51520 | <222> | 52180 IF S=1 THEN S=0:GOTO 52200 | <201> |
| 51510 IF T%="/"THEN 51650 | <051> | 52190 PRINT#1,T\$; | <082> |
| 51520 I=I+1 | <194> | 52200 O=O+1 | <230> |
| 51530 IF I<=L THEN 51470 | <239> | 52210 IF O<=LO THEN 52150 | <055> |
| 51540 : | <207> | 52220 IF UB\$=")"THEN PRINT#1:PRINT#4,CHR\$(2 | |
| 51550 REM ***** | <045> | 7);CHR\$(106);CHR\$(21):CLOSE 3:OPEN 3 | |
| 51560 REM * KEIN BRUCH * | <125> | ,4,10,"7" | <031> |
| 51570 REM ***** | <065> | 52225 IF UB\$=")"THEN BB=BB-INT(M-.5):GOTO 5 | |
| 51580 : | <247> | 1320 | |
| 51590 OB\$=FU\$:UB\$="":LO=L:BB=BB+L:GOTO 521 | <063> | 52230 : | <135> |
| 10 | <011> | 52240 REM ***** | <057> |
| 51600 : | <063> | 52250 REM * BRUCHSTRICH * | <154> |
| 51610 REM ***** | <032> | 52260 REM ***** | <077> |
| 51620 REM * ZERLEGUNG * | <083> | 52270 : | <175> |
| 51630 REM ***** | <051> | 52280 CLOSE 3:OPEN 3,4,9,"7" | <197> |
| 51640 : | <213> | 52290 PRINT#3 | <214> |
| 51650 OB\$="" | <073> | 52300 IF LO-WO<=LU-WU THEN AA=LU-WU+.5 | <043> |
| 51660 FOR J=1 TO I-1:T=J:GOSUB 50170 | <217> | 52310 IF LO-WO>=LU-WU THEN AA=LO-WO+.5 | <207> |
| 51670 OB\$=OB\$+T\$:NEXT J:LO=LEN(OB\$) | <217> | 52320 FOR I=1 TO INT(AA) | <149> |
| 51680 IF LEFT\$(OB\$,1)="("AND RIGHT\$(OB\$,1) | | 52330 PRINT#3,"\$";:NEXT I | <072> |
| =")"THEN OB%=MID\$(OB\$,2,LEN(OB\$)-2): | | 52340 PRINT#3 | <008> |
| LO=LO-2 | <162> | 52350 CLOSE 3:OPEN 3,4,10,"7" | <095> |
| 51690 UB\$="" | <021> | 52360 BB=BB+INT(AA+1) | <194> |
| 51700 FOR J=I+1 TO L:T=J:GOSUB 50170 | <024> | 52370 : | <019> |
| 51710 UB\$=UB\$+T\$:NEXT J:LU=LEN(UB\$) | <104> | 52380 REM ***** | <096> |
| 51720 IF LEFT\$(UB\$,1)="("AND RIGHT\$(UB\$,1) | | 52390 REM * NENNER DRUCKEN * | <115> |
| =")"THEN UB%=MID\$(UB\$,2,LEN(UB\$)-2): | | 52400 REM ***** | <116> |
| LU=LU-2 | <042> | 52410 : | <059> |
| 51730 : | <143> | 52420 U=1:Z=2 | <194> |
| 51740 REM ***** | <233> | 52430 T=U:T%=MID\$(UB\$,T,1) | <176> |
| 51750 REM * POTENZANZAHL * | <151> | 52440 IF T%="↑"THEN E=U:GOSUB 52580:U=E | <220> |
| 51760 REM ***** | <253> | 52450 IF T%="%"THEN Q=U:GOSUB 52910:U=Q | <107> |
| 51770 : | <183> | 52460 IF S=1 THEN S=0:GOTO 52480 | <104> |
| 51780 Z=0:WO=0:WU=0 | <008> | 52470 PRINT#1,T\$; | <108> |
| 51790 I=0:KL=0:Z=Z+1 | <099> | 52480 U=U+1 | <123> |
| 51800 I=I+1 | <220> | 52490 IF U<=LU THEN 52430 | <126> |

Listing »Formeldruck (Fortsetzung)

```

52500 PRINT#1 <154>
52510 PRINT#4,CHR$(27);CHR$(106);CHR$(40); <074>
52520 GOTO 51320 <175>
52530 : <181>
52540 REM ***** <002>
52550 REM * POTENZAUSDRUCK * <066>
52560 REM ***** <022>
52570 : <221>
52580 CLOSE 3:OPEN 3,4,9,"7" <243>
52590 PRINT#1,CHR$(27);"F"; <205>
52600 PRINT#1,CHR$(27);"S";CHR$(0);CHR$(15 <150>
); <188>
52610 E=E+1 <097>
52620 T=E
52630 IF Z=1 THEN T$=MID$(OB$,T,1):AO=AO+. <226>
5 <251>
52640 IF Z=2 THEN T$=MID$(UB$,T,1):AU=AU+. <132>
5 <084>
52645 IF T$="%" THEN Q=E:PP=0:GOSUB 52910:E <094>
=Q:M=M+PP <120>
52650 IF T$="(" THEN EK=EK+1 <115>
52660 IF T$=")" AND EK=0 THEN 52810 <126>
52670 IF T$=")" AND EK>0 THEN EK=EK-1 <007>
52680 IF T$="+" AND EK=0 THEN 52810 <142>
52690 IF T$="-" AND EK=0 THEN 52810 <094>
52700 IF T$="*" AND EK=0 THEN 52810 <081>
52705 IF T$="/" AND EK=0 THEN 52810 <205>
52710 PRINT#1,T$; <234>
52720 IF WW=1 THEN PRINT#4,CHR$(27);CHR$(1 <044>
06);CHR$(3);:CLOSE 3:OPEN 3,4,10,"7" <114>
52730 IF WW=1 THEN PRINT#1,CHR$(8); <057>
52740 IF WW=1 THEN PRINT#1,"&"; <013>
52750 IF WW=1 THEN PRINT#4,CHR$(27);CHR$(7 <145>
4);CHR$(3);:CLOSE 3:OPEN 3,4,9,"7" <175>
52760 M=M+.5 <077>
52770 IF Z=1 AND E=LO THEN S=1:GOTO 52810 <073>
52780 IF Z=2 AND E=LU THEN S=1:GOTO 52810 <001>
52790 E=E+1 <078>
52800 GOTO 52620 <164>
52810 PRINT#4,CHR$(27);CHR$(84);CHR$(18); <098>
52820 EK=0 <041>
52830 PRINT#1,CHR$(27);"E"; <058>
52840 CLOSE 3:OPEN 3,4,10,"7" <229>
52850 RETURN <090>
52860 : <214>
52870 REM ***** <097>
52880 REM * WURZELAUSTRUCK * <058>
52890 REM ***** <229>
52900 : <090>
52910 PRINT#1,"%";:KL=1:Q=Q+1:WW=1:WM=0 <214>
52920 Q=Q+1 <097>
52930 T=Q
52940 IF Z=1 THEN T$=MID$(OB$,T,1)
52950 IF Z=2 THEN T$=MID$(UB$,T,1)
52960 IF T$="+" THEN E=Q:GOSUB 52580:Q=E:WM <058>
=WM+1 <168>
52970 IF S=1 THEN S=0:GOTO 52920
52980 IF T$="(" THEN KL=KL+1 <194>
52990 IF T$=")" THEN KL=KL-1 <204>
53000 IF T$=")" AND KL=0 THEN S=1:GOTO 5307 <214>
0 <142>
53010 PRINT#1,T$; <175>
53015 PP=PP+.5
53020 PRINT#4,CHR$(27);CHR$(106);CHR$(3);: <045>
CLOSE 3:OPEN 3,4,10,"7" <047>
53030 PRINT#1,CHR$(8); <149>
53040 PRINT#1,"&";
53050 PRINT#4,CHR$(27);CHR$(74);CHR$(3);:C <227>
LOSE 3:OPEN 3,4,10,"7" <160>
53060 GOTO 52920 <053>
53070 WW=0 <115>
53080 M=M+WM+2 <059>
53090 RETURN <243>
53100 : <080>
53110 REM ***** <168>
53120 REM * ENDE * <100>
53130 REM ***** <027>
53140 :
53150 PRINT#4,CHR$(27);"@";:CLOSE 1:CLOSE <059>
2:CLOSE 3:CLOSE 4 <188>
53160 OPEN 3,4,8,"12":CLOSE 3 <139>
53170 RETURN

```

© 64'er

Listing »Formeldruck« (Schluß)

Schreiben Sie Ihre eigenen Fehlermeldungen

Wer die öden Meldungen des Betriebssystems wie »READY« oder »SYNTAX ERROR« satt hat, kann mit dem »Error-Changer« alle Fehlerausgaben ändern und auf Diskette speichern.

Denken Sie einmal an klassisches Basic-Programmieren: Drei Zeilen eintippen, Probelauf, Fehlermeldung, Korrektur, Probelauf ... und so weiter. Und bei jedem Fehler erscheinen die sattsam bekannten, äußerst langweiligen Einheitsfehlermeldungen.

Mit dem hier vorgestellten Programm »Error-Changer« ist es endlich möglich, den Wortlaut der Fehlermeldungen frei zu wählen (Bild 1 und 2). Nun, das ist immer noch nichts so Besonderes. Besonderes ist, daß der neue Text jeder Fehlermeldung bis zu 80 (achtzig) Zeichen lang sein kann.

Ob Sie den »UNDEFD STATEMENT ERROR« mit »QUO VADIS« oder mit »WOHIN, MEIN FREUND??« übersetzen, bleibt Ihrer Phantasie überlassen. Es schaut auch besser aus, wenn statt dem faden »READY«, zum Beispiel das warmstartsichere »ANDYS SOFTWARE IST IMMER EIN GEWINN« erscheint. Individualität ist Trumpf, Einheits-ROM ist out.

| | |
|--|--|
| <pre> LOAD"FEHLER.OBJ",8,1 SEARCHING FOR FEHLER.OBJ READING READY. SYS49152 JETZT BIN ICH FERTIG. PRINT 1/0 ?NICHT DURCH NULL TEILEN!!! JETZT BIN ICH FERTIG. CONT ?DAS GEHT JETZT NICHT MEHR JETZT BIN ICH FERTIG. DIMA\$(10000) ?ICH HABE KEINEN PLATZ MEHR JETZT BIN ICH FERTIG. </pre> | <pre> LOAD"FEHLER.OBJ",8,1 SEARCHING FOR FEHLER.OBJ READING READY. SYS49152 READY. PRINT 1/0 ?DIVISION BY ZERO ERROR READY. CONT ?CAN'T CONTINUE ERROR READY. DIMA\$(10000) ?OUT OF MEMORY ERROR READY. </pre> |
|--|--|

Bild 1. So könnten zum Beispiel geänderte Fehlermeldungen aussehen

Bild 2. Im Vergleich zu Bild 1: So sehen die Originalmeldungen aus

Das Programm ist kinderleicht zu bedienen: Error-Changer (siehe Listing) laden und starten.

Es erscheint der Text der alten, englischen Fehlermeldung. Sollte sie nicht geändert werden, so wird sie normal mit der RETURN-Taste übernommen. Ansonsten wird der neue Text eingetippt. Die Länge des neuen Textes kann frei gewählt werden und ist nur durch die 80 Zeichen des INPUT-Befehls begrenzt. Sollte der neue Text kürzer sein als der alte, so ist der Rest mit Leerzeichen zu überschreiben. Es kann auch die Abbruchmeldung BREAK und die READY-Meldung geändert werden. Jedoch erscheint, bedingt durch das Betriebssystem, nicht immer der geänderte Text.

Nach Beendigung schaltet das Programm automatisch auf die neuen Fehlermeldungen um. Diese können auf Kassette oder auf Disk gespeichert werden, um die kreativen Ergüsse für die Nachwelt zu erhalten.

Die neuen Fehlermeldungen werden mit LOAD "FEHLER.OBJ",8,1 (für Datasette ,1,1) geladen, mit SYS 49152 ein- und mit SYS 49226 ausgeschaltet. Nach dem Laden sollten Sie unbedingt NEW eingeben.

Nun zum Programmtechnischen.

Es wird der »Vektor auf Fehlermeldungen« \$0300-\$0301 (dezimal 768-769) auf das neue Programm »umgebogen«. Tritt ein Fehler auf (Fehlernummer größer 0), so stellt das Programm normale Ein- und Ausgabeverhältnisse her (Bildschirm, Tastatur). Die Fehlernummer bildet einen Vektor auf die Adresse des ersten Buchstabens der entsprechenden Fehlermel-

dung. Die Texte der Fehlermeldungen werden hintereinander gespeichert. Als Trennzeichen fungiert das siebte Bit des letzten Buchstabens. Ist es gesetzt, so weiß das Betriebssystem, daß dieser Buchstabe der letzte ist. Somit ist der erste Buchstabe (Zeiger im Steuerprogramm) und der letzte definiert (siebtes Bit gesetzt). Jetzt muß die Meldung nur noch geschrieben werden. (Das Ganze kann mit dem K-Befehl des SMON angeschaut werden. Nicht vergessen: Auf Kleinschriftmodus umschalten!)

Um Speicherplatz zu sparen, wurden im Programm die Fehlermeldungen ohne das Wort »ERROR« gespeichert, da dieses bei allen Meldungen gleich ist. Bei den neuen Meldungen muß der ganze Wortlaut angegeben werden, da bei manchen Scherzfehlermeldungen das Wort »ERROR« oder »FEHLER« stören würde.

Beschreibung der wichtigsten Variablen:

T\$(I)=Text der alten und nach dem Überschreiben der neuen Fehlermeldung.

LV = Lesevektor. Liest alte Meldung aus dem ROM.

SA = Schreibvektor. Schreibt neuen Text in das RAM.

VA = Schreibvektor auf Lo/Hi-Adressen des ersten Buchstabens der neuen Meldung.

SA und EA = Start- und Endadresse als Parameter für den SAVE-Befehl

FZ = Fehlerzähler (identisch mit der betriebssysteminternen Fehlernummer!)

(Andreas Knipp/tr)

```

110 REM *****
130 REM *
140 REM * E R R O R - C H A N G E R *
150 REM *
151 REM * C 64
152 REM *
160 REM * BY ANDREAS KNIPP
170 REM *
180 REM *****
190 REM
200 POKE 53280,4:POKE 53281,6:GOTO 250
250 PRINT"(CLR,WHITE)WAEHLBARE FEHLERMELDU
NGEN BY A.KNIPP KPS":DIM T$(40):FZ=1
260 T$(30)="BREAK ERROR"
270 T$(31)="READY.":HI=256
280 FOR I=49152 TO 49232
300 READ X:POKE I,X:NEXT
310 POKE 768,139:POKE 769,227:REM FEHLERME
LDUNGEN AUF ROM STELLEN.
320 LV=41372:SV=49293:VA=49233
330 FOR LV=41373 TO 41767:REM EINLESEN DER
ALTEN FEHLERMELDUNGEN
340 T$(FZ)=T$(FZ)+CHR$(PEEK(LV)AND 127):IF
PEEK(LV)<127 THEN 370
350 T$(FZ)=T$(FZ)+" ERROR":PRINT T$(FZ)
360 FZ=FZ+1
370 NEXT LV:FZ=FZ+1:REM PLATZ FUER READYME
LDUNG
380 FOR I=1 TO FZ:REM MOEGlichkeit DER AEN
DERUNG
390 PRINT"{2RIGHT}"T$(I)CHR$(13)"{UP}";:IN
PUT T$(I)
400 IF LEN(T$(I))<2 THEN 390
410 NEXT I
420 INPUT"ALLES RICHTIG J/N";AR$:IF AR$<"
J"THEN 380
430 FOR J=1 TO FZ-1:REM NEUE FM IN DEN SPE
ICHER SCHREIBEN
440 FOR I=1 TO LEN(T$(J))-1
450 POKE SV+I-1,ASC(MID$(T$(J),I,1))
460 NEXT I
470 POKE SV+I-1,ASC(MID$(T$(J),I,1))+128
480 A=SV:GOSUB 690:POKE VA+1,AH$:POKE VA,A
LZ
490 VA=VA+2:SV=SV+I:NEXT J
500 POKE SV,13
510 FOR J=1 TO LEN(T$(31)):POKE SV+J,ASC(M
ID$(T$(31),J,1)):NEXT:REM READY

```

```

<122>
<179>
<229>
<199>
<066>
<201>
<105>
<219>
<192>
<252>
<124>
<197>
<104>
<150>
<139>
<225>
<202>
<210>
<043>
<236>
<052>
<100>
<061>
<126>
<152>
<146>
<240>
<054>
<109>
<031>
<061>
<034>
<126>
<187>
<038>
<189>
<167>
520 POKE SV+J,13:A=SV:GOSUB 690:POKE 49220
,AL$:POKE 49222,AH$:POKE SV+J+1,0 <085>
530 SYS 49152 <080>
540 INPUT" SPEICHERUNG J/N";S$:IF S$="N"THE
N END <008>
550 IF S$<"J"THEN 540 <069>
560 SA=49152:AE=SV+J+2 <207>
570 PRINT"GERAETENUMMER"; <238>
580 PRINT"{3SPACE}FLOPPY = Z.B.{2SPACE}8{2
4SPACE}DATASETTE ={4SPACE}1"; <196>
590 INPUT DN <110>
600 A$="@:FEHLER.OBJ":AZ=LEN(A$) <056>
610 A=681:GOSUB 690:POKE 183,AZ:POKE 187,A
LZ:POKE 188,AHZ <122>
620 FOR I=1 TO AZ:POKE 680+I,ASC(MID$(A$,I
)):NEXT:REM FILENAME <204>
630 A=SA:GOSUB 690:POKE 251,AL$:POKE 252,A
H$:REM STARTADRESSE <252>
640 A=AE:GOSUB 690:POKE 781,AL$:POKE 782,A
H$:REM ENDADRESSE <109>
650 POKE 186,DN:POKE 780,251:SYS 65496:REM
SAVE:DN = DEVICENUMMER <027>
660 PRINT"{WHITE}SPEICHERUNG ERFOLGTE UNTE
R "CHR$(34)"FEHLER.OBJ"CHR$(34) <133>
670 PRINT"ABSOLUT LADEN UND MIT SYS 49152
STARTEN" <254>
680 END <174>
690 AHZ=A/HI:ALZ=A-AHZ*HI <231>
700 RETURN <250>
710 DATA 169,11,160,192,141,0,3,140,1,3,96
,138 <111>
720 DATA 16,3,76,67,192,10,170,189,79,192,
133,34 <186>
730 DATA 189,80,192,133,35,32,204,255,169,
0,133,19 <008>
740 DATA 32,215,170,32,69,171,160,0,177,34
,72,41 <019>
750 DATA 127,32,71,171,200,104,16,244,32,1
22,166,164 <132>
760 DATA 58,200,240,3,32,194,189,169,187,1
60,194,76 <174>
770 DATA 120,164 <009>
780 DATA 169,139,160,227,76,4,192 <192>

```

© 64'er

Listing »Error Changer«. Beachten Sie die Eingabehinweise auf Seite 6.

Kopierschutz ohne Read Error

Read Errors sind für die meisten Kopierprogramme kein Problem, deshalb wird hier ein anderer Weg besprochen. Basic- und Maschinen-Programme sind nur dann lauffähig, wenn vor dem Start ein Paßwort eingegeben wird.

Das Programm ist in Maschinensprache geschrieben und belegt den Speicherbereich von \$CF00 bis \$CF46. Der Bereich von \$CF47 bis \$CFFF ist für ein Paßwort reserviert, das eingegeben werden muß, nachdem der zuvor abgetippte Lader (siehe Listing) mit RUN gestartet wurde. Von nun an werden alle Programme, die auf Diskette gespeichert werden, beim Speichern codiert. Solange der Computer eingeschaltet bleibt, können Programme, die mit dem eingegebenen Paßwort gespeichert wurden, ganz normal geladen werden. Programme, die unter einem anderen Paßwort gespeichert wurden lassen sich zwar auch noch Laden, sind aber nicht mehr lauffähig. Sollen solche Programme gestartet werden, muß dem Computer vor dem Laden das neue Paßwort übermittelt werden. Dies geschieht mit einem kleinen Basic-Programm:

```
10 INPUT C$: C=LEN(C$)
20 FOR I=1 TO C
30 POKE 53072+I,ASC(MID$(C$,I,1))
40 NEXT
50 POKE 13*1613-1,C
60 END
```

```
10 REM ***** <063>
15 REM * <064>
20 REM * >> CODE S/L LADER << * <115>
25 REM * * <074>
30 REM * MICHAEL DIETZ * <100>
35 REM * 3303 VECHELDE * <217>
40 REM * BERLINER STR. 23 * <115>
45 REM * TEL.:05302/5991 * <072>
50 REM * * <099>
55 REM ***** <108>
60 : <036>
65 : <041>
100 POKE 53280,6:POKE 53281,6:POKE 646,14 <158>
110 PRINT"CLR" <098>
120 FOR I=0 TO 41:READ A:POKE 828+I,A:NEXT <211>
130 SYS 828 <231>
140 FOR I=52992 TO 53061:READ A <089>
150 POKE I,A:NEXT <247>
160 INPUT"2DOWN,2SPACE"CODE ";C$:C=LEN(C$) <225>
170 FOR I=1 TO C <021>
180 POKE 53072+I,ASC(MID$(C$,I,1)):NEXT <165>
190 POKE 62954,32:POKE 62955,00:POKE 62956,207 <002>
200 POKE 62626,32:POKE 62627,00:POKE 62628,207 <105>
210 POKE 62731,76:POKE 62732,12:POKE 62733,207 <229>
220 POKE 63019,76:POKE 63020,30:POKE 63021,207 <042>
```

```
230 POKE 13*1613-1,C:POKE 1,53:NEW <183>
240 : <216>
250 : <226>
260 DATA 169,0,133,95,133,90,133,88,169 <059>
270 DATA 160,133,96,169,192,133,91,133,89,32 <094>
280 DATA 191,163,169,0,133,95,133,90,133 <103>
290 DATA 88,169,224,133,96,169,0,133,91,133,89 <119>
300 DATA 76,191,163 <246>
310 : <032>
320 : <042>
330 DATA 142,253,207,162,1,142,254 <222>
340 DATA 207,174,253,207,96,32,46,207 <229>
350 DATA 141,252,207,56,173,253,207 <028>
360 DATA 237,252,207,164,147,76,14 <069>
370 DATA 245,141,252,207,32,46,207 <243>
380 DATA 24,109,252,207,32,221,237 <196>
390 DATA 76,46,246,206,254,207,208 <100>
400 DATA 6,173,255,207,141,254,207 <186>
410 DATA 142,253,207,174,254,207,189 <166>
420 DATA 80,207,174,253,207,96 <208>
```

Listing. Lader zum Programm »Kopierschutz ohne Read Error«. Bitte beachten Sie die Eingabehinweise auf Seite 6.

Soll ein normales, also uncodiertes, Programm geladen oder gespeichert werden, so muß man den Codierer mit POKE 1,55 ausschalten. Mit POKE 1,53 läßt er sich wieder einschalten. (Michael Dietz/ah)



Tape Save — Maschinen- programme einfach gespeichert

Der Save-Befehl des C 64 erlaubt es leider nicht, bestimmte Speicherbereiche und damit Maschinenprogramme abzuspeichern. Mit »Tape Save« können Sie dies sogar im Dialogverfahren und sowohl auf Diskette als auch auf Datasette.

Wer kennt das Problem nicht: Man hat einen DATA-Lader für ein Maschinenprogramm und kann das Programm wegen eines fehlenden Monitors nicht direkt abspeichern! Hier setzt nun unser Programm »Tape Save« ein, das es auch Anfängern ermöglicht, im Dialogverfahren Maschinenprogramme auf Kassette oder Diskette zu speichern.

»Tape Save« (siehe Listing) tippen Sie bitte mit dem MSE ab. Es kann dann mit »LOAD "TAPE SAVE" ,1,1« beziehungsweise

»...8,1« geladen werden. Gestartet wird »Tape Save« mit SYS 49152. Vor dem Start müssen Sie noch NEW eingeben, um keinen »OUT OF MEMORY ERROR« zu erhalten.

Bedienungshinweise

Als erstes wird nach dem Namen des zu speichernden Programms gefragt. Der Computer akzeptiert höchstens 16 Zeichen, verlangt aber mindestens ein Zeichen, sonst wird die Eingabe abgelehnt. Danach gibt man Start- und Endadresse in hexadezimaler Form ein. Dabei erscheint allerdings kein Cursor, die eingetippten Werte werden aber angezeigt. Nach Eingabe der beiden vierstelligen Zahlen fährt das Programm mit der Frage nach der Nummer des Peripheriegerätes fort. Auch hier wird nur eine gültige Eingabe akzeptiert, wobei »1« für Datasette und »8« für das Diskettenlaufwerk einzugeben ist. Danach startet »Tape Save« den Speichervorgang. Nach dem Speichern ermöglicht es dann dem Benutzer, mittels der Funktionstasten den weiteren Programmablauf zu gestalten. Man kann den Speichervorgang wiederholen, das Programm abbrechen oder aber nochmals beginnen, um ein weiteres Programm zu speichern.

Hier noch einige wichtige Bedienungshinweise:

Diskettenbenutzer müssen aufpassen, wenn sie mit der Funktionstaste »F3« nochmals ohne Parameteränderung speichern wollen. Wurde die Diskette nicht gewechselt, erhält man einen »FILE EXISTS ERROR«, weil auch der Filename nicht geändert wird.

Der wichtigste Nachteil von »Tape Save« ist seine Lage im Speicher: Es liegt von \$c000 bis \$c369. Das bedeutet, daß Programme, die im \$c-Bereich liegen, auch nicht von »Tape Save« gespeichert werden können. Ebenso können keine Maschinensprache-Programme, die sich im Kassettenpuffer befinden, auf Kassette gespeichert werden. Hier ist nur die Speicherung auf Diskette möglich.

(Johannes Rest/bs)

programm : tape save c000 c369

```
c000 : 4c c9 c0 54 41 50 45 20 d7
c008 : 53 41 56 45 20 56 31 2e 10
c010 : 30 28 43 29 20 42 59 20 04
c018 : 4a 4f 48 41 4e 4e 45 53 57
c020 : 20 52 45 53 54 46 49 4c 5a
c028 : 45 4e 41 4d 45 3a 53 54 aa
c030 : 41 52 54 41 44 52 45 53 6a
c038 : 53 45 3a 45 4e 44 41 44 fa
c040 : 52 45 53 53 45 3a 55 45 7a
c048 : 42 45 52 54 52 41 47 55 43
c050 : 4e 47 20 42 45 45 4e 44 d2
c058 : 45 54 2e 47 45 52 41 45 b2
c060 : 54 45 41 44 52 45 53 53 73
c068 : 45 3a 46 31 20 2d 20 45 f9
c070 : 4e 44 45 46 33 20 2d 20 24
c078 : 43 4f 50 59 20 53 41 4d de
c080 : 45 20 46 49 4c 45 49 4e 41
c088 : 53 45 52 54 20 4e 45 57 d5
c090 : 20 44 49 53 4b 20 49 46 f7
c098 : 20 4e 45 43 45 53 53 41 58
c0a0 : 52 59 2e 46 35 20 2d 20 3d
c0a8 : 43 4f 50 59 20 41 4e 4f b6
c0b0 : 54 48 45 52 20 46 49 4c b6
c0b8 : 45 12 3e 3e 3e 20 d2 45 18
c0c0 : 54 55 52 4e 20 3c 3c 3c 6b
c0c8 : 92 20 44 e5 a9 0e 20 d2 69
c0d0 : ff a2 00 bd 03 c0 9d 00 85
c0d8 : 04 e0 0d f0 04 e8 4c d3 0e
c0e0 : c0 a2 00 bd 11 c0 9d 28 87
c0e8 : 04 e0 13 f0 04 e8 4c e3 c0
c0f0 : c0 a2 00 bd 25 c0 9d a0 c9
c0f8 : 04 e0 08 f0 04 e8 4c f3 2d
c100 : c0 a9 0a 85 d3 a9 04 85 6e
c108 : d6 20 10 e5 a9 00 85 fb 58
c110 : 20 cf ff a6 fb c9 0d f0 11
c118 : 0d 9d d0 cf e0 0f f0 0d 86
c120 : e8 86 fb 4c 10 c1 e0 01 68
c128 : b0 03 4c 19 c1 86 b7 a2 05
```

```
c130 : 00 bd 2e c0 9d f0 04 e0 e6
c138 : 0c f0 04 e8 4c 31 c1 a9 83
c140 : 06 85 d6 a9 0e 85 d3 20 90
c148 : 10 e5 20 68 c1 a2 00 bd 0d
c150 : 77 02 c9 00 f0 f9 c9 0d 5b
c158 : f0 f0 9d fe 04 9d e4 cf 68
c160 : e8 e0 04 f0 6b 4c 4f c1 b1
c168 : a2 00 a9 00 9d 77 02 e8 e4
c170 : e0 05 d0 f8 85 c6 60 a9 89
c178 : 20 a2 00 9d fe 04 e8 e0 13
c180 : 05 d0 f8 60 c0 01 f0 05 19
c188 : a0 00 a2 01 60 a0 00 a2 41
c190 : 02 60 4c 67 c2 a2 00 bd 7f
c198 : e4 cf c9 41 b0 0e c9 30 01
c1a0 : e0 03 4c 84 c1 c9 3a 90 ea
c1a8 : 0a 4c a2 c1 c9 47 90 10 f2
c1b0 : 4c a2 c1 38 e9 30 95 fb 33
c1b8 : e0 03 f0 10 e8 4c 97 c1 2b
c1c0 : c9 00 f0 ec 38 e9 37 95 3e
c1c8 : fb 4c b8 c1 60 4c 4a c1 65
c1d0 : 20 68 c1 20 95 c1 e0 03 8a
c1d8 : b0 04 e0 01 f0 ef a5 fb 00
c1e0 : 20 30 c2 18 65 fc 8d fb 32
c1e8 : cf a5 fd 20 30 c2 18 65 51
c1f0 : fe 8d f9 cf 4c f7 c1 a9 0c
c1f8 : 06 85 d3 a9 07 85 d6 20 23
c200 : 10 e5 a2 00 bd 3b c0 9d 9f
c208 : 18 05 e0 0a f0 04 e8 4c 88
c210 : 04 c2 20 68 c1 a2 00 bd 37
c218 : 77 02 c9 00 f0 f9 c9 0d 23
c220 : f0 f0 9d 24 05 9d e4 cf e5
c228 : e8 e0 04 f0 08 4c 17 c2 64
c230 : 0a 0a 0a 0a 60 20 68 c1 2f
c238 : a0 01 20 95 c1 e0 02 f0 21
c240 : b6 a5 fb 20 30 c2 18 65 10
c248 : fc 8d fa cf a5 fd 20 30 ee
c250 : c2 18 65 fe 8d fb cf a2 95
c258 : 00 bd 5b c0 9d 68 05 e0 19
c260 : 0e f0 04 e8 4c 59 c2 a2 e4
c268 : 00 86 fb a9 09 85 d6 a9 4b
c270 : 10 85 d3 20 10 e5 20 cf 8c
```

```
c278 : ff 85 fc a6 fb e0 01 f0 fa
c280 : 05 e6 fb 4c 76 c2 ad 78 a6
c288 : 05 c9 31 f0 07 c9 38 f0 5e
c290 : 03 4c 92 c1 38 e9 30 8d 45
c298 : fc cf a9 0e 85 d6 a9 00 5e
c2a0 : 85 d3 20 10 e5 ae fc cf 80
c2a8 : 20 ba ff a2 d0 a0 cf a5 16
c2b0 : b7 20 bd ff ae f9 cf ac 3a
c2b8 : f8 cf 86 fb 84 fc a9 fb 88
c2c0 : ae fb cf ac fa cf 20 8d 56
c2c8 : ff 20 44 e5 a2 00 bd 46 53
c2d0 : c0 9d 00 04 e8 e0 15 f0 ab
c2d8 : 03 4c ce c2 a2 00 bd 6a 03
c2e0 : c0 9d 50 04 e8 e0 09 f0 9f
c2e8 : 03 4c de c2 a2 00 bd 73 29
c2f0 : c0 9d 78 04 e8 e0 13 f0 e1
c2f8 : 03 4c ee c2 a2 00 bd a3 9e
c300 : c0 9d a0 04 e8 e0 16 f0 07
c308 : 03 4c fe c2 20 87 ea a5 7e
c310 : cb c9 04 f0 0b c9 05 f0 d4
c318 : 0b c9 06 f0 0d 4c 0c c3 92
c320 : 20 44 e5 60 ad fc cf 4c 82
c328 : 2d c3 4c c9 c0 a2 00 c9 38
c330 : 01 f0 31 bd 86 c0 9d 18 c2
c338 : 05 e8 e0 1d f0 03 4c 33 4c
c340 : c3 a9 0a 85 d6 20 10 e5 86
c348 : a2 00 bd b9 c0 20 d2 ff e9
c350 : e8 e0 10 f0 03 4c 4a c3 0e
c358 : 20 87 ea a5 cb c9 01 f0 9c
c360 : 03 4c 58 c3 a2 00 4c 9a a8
c368 : c2 ff 00 ff 00 ff 00 ff 2a
```

Das Listing »Tape Save« muß mit dem MSE eingegeben werden. Beachten Sie die Eingabehinweise auf Seite 6.

Linker 64 — Schluß mit dem Nachladen

Wenn Sie öfters mehrere Maschinenprogramme gleichzeitig im Speicher haben wollen und es leid sind, jedes Programm einzeln nachzuladen, dann ist der Linker das ideale Arbeitswerkzeug für Sie.

Seine Anwendungen sind vielfältig: vom einfachen Basic-Start-Generator für Maschinenprogramme bis zum Verketter (= Linker) von mehrteiligen Spielen. Das Endprodukt, also das generierte Programm, kann dann ganz einfach auch von der Datasette geladen werden, was bisher an den Diskettenladerroutinen der Programme scheiterte. Diese Laderoutinen müssen natürlich vorher entfernt werden.

Die Bedienung ist denkbar einfach: Diskette mit den einzelnen Programmen nehmen, Linker 64 (siehe Listing) starten und bedienen. Es wird eine lauffähige Version generiert, die man mit LOAD und RUN starten kann.

Die einzige Denkarbeit besteht in der Überlegung, in welcher Reihenfolge die Programme wieder an ihre ursprünglichen Adressen verschoben werden sollen. Das ist wichtig, da sonst während des Verschiebevorgangs bereits verschobene Programmteile, die noch nicht verschoben überschreiben könnten. Im allgemeinen wird man zuerst die Maschinenroutinen verschieben, die an die höchste Adresse geschrieben werden sollen.

Die Arbeitsweise des Linkers

Der Linker generiert drei Programmteile, die am Ende verkettert werden: Teil »S« besteht aus der Basic-Zeilenummer, dem SYS-Befehl und dem anschließenden Text, gleich einer REM-Zeile. Teil »V« besteht aus dem generierten Verschiebeprogramm. Für Interessierte sei gesagt, daß bei einer Verschiebung nach oben die betriebssysteminterne Blockverschieberoutine (\$a3bf) angesprungen wird. Bei Transfer nach unten kommt eine »handgestrickte« Routine zur Anwendung. Teil »P« enthält schließlich die aneinandergehängten Maschinenprogramme.

Der Linker kennt zwei Betriebsarten: entweder Verschiebeteil vor oder nach den gelinkten Programmen. Im Modus »0« ist die Reihenfolge der drei Teile S-V-P, im Modus »1« ist sie S-P-V. Der Modus »0« (LoMem-Modus) wird fast immer benutzt, da nur sehr selten Programmteile im Basic-Startbereich (2049 = \$0801) laufen. Sollte dies aber doch der Fall sein, oder sollen Basic-Programme mit Maschinenprogrammen zusammengeknüpft werden, so muß Modus »1« (HiMem-Modus) verwendet werden. Die Verschieberoutine würde sich sonst selbst überschreiben.

Anwendung des Linkers

Beispiel: Wir wollen aus dem SMON \$c000, einer RENEW-Routine mit der Startadresse 36000 und einem Assembler ASS \$9000 ein einziges Programm machen. Dieses sollte

nach dem Laden die einzelnen Teile gleich an die richtigen Adressen im Speicher versetzen und mit »RUN« zu starten sein. Weiterhin soll danach gleich der SMON starten:

(Alle drei Programme müssen sich auf der gleichen Diskette befinden.) Zuerst lädt man den Linker und startet ihn. Dann gibt man die Zeilenummer, in der später der SYS-Befehl stehen soll, einen erklärenden REM-Text und den Namen ein, den das fertige Programm erhalten soll. Als Betriebsart wählen wir »0«, da der Basic-Startbereich von unseren Programmen nicht berührt wird. Die Frage nach der Anzahl der Programme beantworten wir mir »3«. Nun werden die Namen der einzelnen Programme eingegeben, woraufhin der Linker deren Startadressen berechnet und ausgibt. Diese könnten jetzt noch geändert werden, was aber nicht sehr ratsam ist. Maschinenprogramme haben im allgemeinen nämlich die Eigenschaft, nur in dem Speicherbereich zu laufen, für den sie geschrieben wurden. In unserem Fall übernehmen wir also die Startadressen mit der RETURN-Taste.

Jetzt fragt der Linker nach der Reihenfolge, in der später die Programme verschoben werden sollen (ist in unserem Fall egal, da sich unsere Beispielprogramme nicht im geringsten gegenseitig stören). Der Linker meldet sich mit:

```
1 = SMON $c000
2 = ASS $9000
3 = RENEW
4 = ENDE
```

Wir tippen zum Beispiel »2« für die Verschiebung des Assemblers. Nun wird für ihn die Einsprungsart verlangt: EINSPRUNG: 0=KEINER, 1=JSR, 2=JMP

Liste der wichtigsten Variablen:

```
U$ = Überschrift
T$ = REM-Text, der hinter dem SYS-Befehl im generierten
    Programm steht.
NA(U) = Neue Anfangsadresse (Adresse, an der das Einzel-
    programm nach dem Verschieben steht)
NE(U) = Neue Endadresse
S(U) = Alte Anfangsadresse (dort steht der Programmteil im
    fertigen Programm)
E(U) = Alte Endadresse
K = Anzahl der zu linkenden Programme (maximal acht)
BS$ = Floppy-Befehlsstring
LH = LoMem- oder HiMem-Modus
NA$ = Name des fertigen Programms
```

Die Variablenliste des Linkers

| Eingangsvariable | Ausgangsvariable | Beschreibung | Einsprung |
|------------------|--|---|-----------|
| A=2 byte-Zahl | AL% (LO-Anteil) AH% (HI-Anteil) | | 940 |
| X\$=filename | C1 0=vorhanden, 62=nicht vorhanden | Testet auf Vorhandensein von Programmen | 950 |
| - | - | Fehlerbehandlung: Return nur bei ok oder File not found, sonst Abbruch. | 960 |
| F\$=filename | SA=Startadresse, EA=Endadresse V=Verbrauch | Berechnet SA, EA, V und schreibt aktuellen Track und Sektor | 1 120 |

Tabelle. Beschreibung der wichtigsten Unterprogramme

Da nach der Verschiebung des Assemblers dieser nicht gleich gestartet werden soll, geben wir eine »0« ein. Es erscheint am Bildschirm:

```
1 = SMON $c000
2 =
3 = RENEW
4 = ENDE
```

Sie sehen, der Assembler ist »verschwunden«. Der Linker wartet auf das nächste Kommando. Wir geben »3« für die RENEW-Routine ein. Da diese ebenfalls nicht gleich gestartet werden soll, beantworten wir die nachfolgende Frage mit »0«. Dann verschieben wir mit »1« den SMON. Wir haben vorher gesagt, daß das fertige Programm diesen gleich starten sollte. Also Einsprungsart »2«. Die Einsprungsadresse, die der Linker daraufhin verlangt, ist 49152 (SMON).

Es wird noch einmal nach der Einsprungsart gefragt, da bei der Eingabe von »1« (JSR) noch weitere Einsprünge erfolgen

könnten. Wir quittieren das Ganze mit »0« und das Abenteuer ist für uns erledigt.

Jetzt wirft der Linker die DOS-interne Copy-Routine an, die die vorher generierten Teile »V« und »S« auf der Diskette direkt aneinanderhängt. Nach einiger Zeit meldet sich der Computer mit READY, und der Link-Vorgang ist beendet. Auf der Diskette befindet sich jetzt das fertige Programm, das mit LOAD'name',8 geladen und mit »RUN« gestartet werden kann. Es werden zuerst die gelinkten Programme in der festgelegten Reihenfolge an ihren Platz im Speicher verschoben und die durch JMP oder JSR definierten Adressen angesprungen. Achtung: Vor einem erneuten Start durch »RUN« sollte das Programm erst noch einmal geladen werden, ebenso sollte es nach dem Verschieben nicht mehr gespeichert werden.

Die wichtigsten Unterprogramme des Linkers sind in der Tabelle beschrieben. Die Routine »Test auf Vorhandensein von Programmen auf Disk« eignet sich gut zum Einbau in eigene Programme. (Andreas Knipp/tr)

| | | | |
|---|-------|---|-------|
| 10 REM ***** | <025> | 510 PRINT#3,CHR\$(AH)CHR\$(133)CHR\$(96); | <215> |
| 20 REM * LINKER 64 * | <204> | 520 A=E(U)+1:GOSUB 940 | <109> |
| 30 REM * (C) ANDREAS KNIPP * | <188> | 530 PRINT#3,CHR\$(169)CHR\$(AL)CHR\$(133)CHR\$(90)CHR\$(169); | <004> |
| 40 REM ***** | <055> | 540 PRINT#3,CHR\$(AH)CHR\$(133)CHR\$(91); | <207> |
| 50 POKE 53280,4:POKE 53281,5 | <206> | 550 A=NE(U)+1:GOSUB 940 | <059> |
| 60 U\$="{CLR,ORANGE,RVSDN,40SPACE,RVOFF}" + CHR\$(13) | <104> | 560 PRINT#3,CHR\$(169)CHR\$(AL)CHR\$(133)CHR\$(88)CHR\$(169); | <049> |
| 70 U\$=U\$+"{WHITE,11SPACE}L I N K E R{3SPACE}6 4"+CHR\$(13) | <103> | 570 PRINT#3,CHR\$(AH)CHR\$(133)CHR\$(89); | <041> |
| 80 U\$=U\$+"{6SPACE,DOWN}PROGRAMMED BY ANDREAS KNIPP"+CHR\$(13) | <060> | 580 PRINT#3,CHR\$(32)CHR\$(191)CHR\$(163);:Z=Z+27:GOTO 740 | <164> |
| 90 PRINT U\$ | <172> | 590 A=S(U):GOSUB 940:A1=Z+4 | <204> |
| 100 OPEN 15,8,15,"I":GOSUB 960 | <084> | 600 PRINT#3,CHR\$(160)CHR\$(0)CHR\$(185)CHR\$(AL)CHR\$(AH)CHR\$(153); | <095> |
| 110 Z=0:A\$="":LL\$="{37SPACE}" + CHR\$(13) + "U P" | <177> | 610 Z=Z+6:A=NA(U):GOSUB 940:A2=Z+1 | <247> |
| 120 INPUT "{HOME,10DOWN}ZEILENNUMMER ";ZN | <253> | 620 PRINT#3,CHR\$(AL)CHR\$(AH)CHR\$(200) + CHR\$(192);:A=NE(U)+1-AL | <042> |
| 130 INPUT "{DOWN}TEXT{9SPACE}";T\$ | <041> | 630 GOSUB 940:Z=Z+4 | <046> |
| 140 PRINT LL\$ZN;"SYSRRRR "T\$ | <126> | 640 A3=AH | <135> |
| 150 INPUT "{DOWN}NAME DES GENERIERTEN PRG";NA\$:X\$=NA\$:GOSUB 950 | <043> | 650 PRINT#3,CHR\$(AL)CHR\$(240)CHR\$(12)CHR\$(152)CHR\$(208); | <214> |
| 160 IF C1=0 THEN PRINT "{UP}"NA\$"{2SPACE}BE REITS VORHANDEN{8SPACE}":GOTO 120 | <086> | 660 PRINT#3,CHR\$(242)CHR\$(238); | <057> |
| 170 INPUT "{DOWN}WIEVIELE PRG WERDEN VERKET TET ";K | <215> | 670 Z=Z+7:A=A1:GOSUB 940 | <164> |
| 180 PRINT "{DOWN}VERSCHIEBEPRG VOR (=0)" | <138> | 680 PRINT#3,CHR\$(AL)CHR\$(AH)CHR\$(238); | <101> |
| 190 INPUT "ODER NACH (=1) PROGRAMMBLOCK";LH | <025> | 690 Z=Z+3:A=A2:GOSUB 940 | <191> |
| 200 INPUT "{DOWN}ALLES RICHTIG J/N";AR\$ | <004> | 700 PRINT#3,CHR\$(AL)CHR\$(AH)CHR\$(24)CHR\$(144)CHR\$(233); | <253> |
| 210 IF AR\$<"J" THEN 120 | <087> | 710 PRINT#3,CHR\$(173)CHR\$(AL)CHR\$(AH); | <072> |
| 220 DIM NA(20),NE(20),S(20),E(20),N\$(20),N1\$(20) | <122> | 720 Z=Z+8 | <218> |
| 230 Z=2064+LEN(T\$) | <142> | 730 PRINT#3,CHR\$(201)CHR\$(A3)CHR\$(144)CHR\$(237);:Z=Z+4 | <066> |
| 240 IF LH=0 THEN SE=Z:Z=Z+K*40:PA=Z | <112> | 740 INPUT "EINSPRUNG: 0=KEINER,1=JSR,2=JMP";E:IF E<0 OR E>2 THEN 740 | <166> |
| 250 IF LH THEN Z=Z+2 | <104> | 750 IF E=1 OR E=2 THEN INPUT "EINSPRUNG";A:GOSUB 940:IF INT(AH/256)GOTO 750 | <075> |
| 260 PRINT U\$ | <088> | 760 IF E=2 THEN PRINT#3,CHR\$(76)CHR\$(AL)CHR\$(AH);:Z=Z+3 | <118> |
| 270 FOR U=1 TO K | <223> | 770 IF E=1 THEN PRINT#3,CHR\$(32)CHR\$(AL)CHR\$(AH);:Z=Z+3 | <162> |
| 280 PRINT "NAME DES"U"{LEFT}.PRG":INPUT N\$(U):N1\$(U)=N\$(U) | <112> | 780 IF E<>0 THEN 740 | <025> |
| 290 X\$=N\$(U):GOSUB 950 | <247> | 790 GOTO 420 | <012> |
| 300 IF C1 THEN PRINT "DISCS PRG GIBT ES NICHT":GOTO 280 | <017> | 800 IF LH=0 THEN FOR U=(Z+1)TO(PA-2):PRINT #3,CHR\$(RND(U)*255);:NEXT | <052> |
| 310 NEXT U | <236> | 810 CLOSE 3 | <075> |
| 320 FOR U=1 TO K | <017> | 820 SA\$=MID\$(STR\$(SE)+"{5SPACE}",2,5) | <117> |
| 330 F\$=N\$(U):GOSUB 1120 | <111> | 830 OPEN 2,8,2,"@:+S,P,W":GOSUB 960 | <168> |
| 340 PRINT "{2DOWN}STARTADRESSE VON{2SPACE}"N\$(U):PRINT "{RIGHT}"SA:INPUT "{UP}";SA | <246> | 840 A=ZN:GOSUB 940 | <011> |
| 350 EA=SA+V | <015> | 850 PRINT#2,CHR\$(1)CHR\$(8)CHR\$(13+LEN(T\$))CHR\$(8); | <094> |
| 360 NA(U)=SA:NE(U)=EA:S(U)=Z | <136> | 860 PRINT#2,CHR\$(AL); | <067> |
| 370 E(U)=S(U)+V:Z=E(U)+2 | <034> | 870 PRINT#2,CHR\$(AH); | <073> |
| 380 NEXT U | <050> | 880 PRINT#2,CHR\$(158); | <231> |
| 390 IF LH=0 THEN Z=SE | <007> | 890 PRINT#2,SA\$+" "+T\$; | <099> |
| 400 IF LH=1 THEN Z=Z-2:SE=Z | <230> | 900 FOR I=1 TO 4 | <110> |
| 410 OPEN 3,8,3,"@:+V,P,W" | <115> | 910 PRINT#2,CHR\$(0);:NEXT:CLOSE 2 | <065> |
| 420 PRINT U\$"{5DOWN}" | <144> | 920 IF LH THEN PRINT#15,"R:G=+S":GOSUB 960:GOTO 1000 | <104> |
| 430 FOR I=1 TO K:PRINT I,N\$(I):NEXT:PRINT K+1,"ENDE":N\$(K+1)="↑" | <024> | 930 PRINT#15,"C:G=+S,+V":GOSUB 960:GOTO 1000 | <143> |
| 440 INPUT "NAECHSTE PRG-NUMMER";U:IF U>K+1 OR N\$(U)="" THEN 440 | <248> | 940 AH=INT(A/256):AL=A-AH*256:RETURN | <120> |
| 450 IF U=K+1 THEN 800 | <202> | | |
| 460 N\$(U)="" | <093> | | |
| 470 IF E(U)=NE(U)GOTO 740 | <251> | | |
| 480 IF E(U)>NE(U)GOTO 590 | <134> | | |
| 490 A=S(U):GOSUB 940 | <134> | | |
| 500 PRINT#3,CHR\$(169)CHR\$(AL)CHR\$(133)CHR\$(95)CHR\$(169); | <238> | | |

Listing »Linker 64«. Beachten Sie bitte die Eingabehinweise auf Seite 6

```

950 OPEN 6,8,6,X$:CLOSE 6 <231>
960 INPUT#15,C1,C$,C2,C3:IF C1=0 OR C1=62 <116>
    THEN RETURN <000>
970 PRINT C1,C$,C2,C3 <063>
980 PRINT"PROGRAMMABBRUCH!!!!" <050>
990 CLOSE 15:SYS 65511:OPEN 1,8,15,"S:+?": <035>
    CLOSE 1:END <133>
1000 FOR I=1 TO K:N$(I)=N1$(I) <093>
1010 N1$(I)="+"+RIGHT$(STR$(I),1):PRINT#15 <215>
    ,"R:"+N1$(I)+"="+N$(I):NEXT <125>
1020 BS$="C:H=G," <244>
1030 FOR I=1 TO K <013>
1040 BS$=BS$+N1$(I) <036>
1050 IF I=K THEN 1070 <064>
1060 BS$=BS$+", " <065>
1070 NEXT <125>
1080 IF LH THEN BS$=BS$+",+V"
1090 PRINT#15,BS$:GOSUB 960:PRINT#15,"S:G"
1100 FOR I=1 TO K:PRINT#15,"R:"+N$(I)+"="+ <164>
    N1$(I):NEXT <041>
1110 PRINT#15,"R:"+NA$+"="M":GOTO 990 <113>
1120 OPEN 2,8,2,"#" <189>
1130 T=18:S=1 <245>
1140 PRINT#15,"U1";2;0;T;S <182>
1150 PRINT#15,"B-P";2;0 <255>
1160 GOSUB 1440:T=A <008>
1170 GOSUB 1440:S=A <097>
1180 FOR X=0 TO 7 <024>
1190 PRINT#15,"B-P";2;X*32+3 <243>
1200 GOSUB 1440:T1=A <252>
1210 GOSUB 1440:S1=A <072>
1220 FF$=""
1230 FOR Y=1 TO 16 <238>
1240 GOSUB 1440 <116>
1250 IF FF$=F$THEN 1310 <234>
1260 FF$=FF$+A$ <042>
1270 IF A$<>MID$(F$,Y,1)THEN Y=16 <109>
1280 NEXT Y <222>
1290 NEXT X <224>
1300 GOTO 1140 <114>
1310 PRINT U$"{3DOWN}IN VERARBEITUNG:{RVSD <138>
    N}"+F$+"{RVOFF}" <214>
1320 T=T1:S=S1
1330 PRINT#15,"U1";2;0;T;S:PRINT LL$;T;S:P <179>
    RINT" {UP}"; <118>
1340 PRINT#15,"B-P";2;0 <191>
1350 GOSUB 1440:T=A <200>
1360 GOSUB 1440:S=A
1370 IF Q=0 THEN Q=1:GOSUB 1440:SL=A:GOSUB <183>
    1440:SH=A:SA=SL+SH*256
1380 IF T=0 THEN EA=SA+PZ*254+S-3;V=EA-SA: <149>
    GOTO 1400 <164>
1390 PZ=PZ+1:GOTO 1330 <131>
1400 PRINT "{DOWN}STARTADRESSE:"SA <171>
1410 PRINT"ENDADRESSE(2SPACE):"EA <144>
1420 PRINT"VERBRAUCH(3SPACE):"V
1430 CLOSE 2:Q=0:PZ=0:RETURN <009>
1440 GET#2,A$:IF A$=""THEN A$=CHR$(0) <247>
1450 A$=ASC(A$):RETURN <248>

```

64'er
Listing »Linker 64«. (Schluß)

Variablen-Dump

64ER ONLINE

Mit diesem kleinen Programm und einem SYS-Befehl haben Sie alle Ihre Basic-Variablen voll im Griff. Eine, nicht nur zum Austesten von Programmen, nützliche Erweiterung.

»Dump« bedeutet übersetzt Speicherauszug oder Speicher Ausdruck eines Computers. Ein Variablen-Dump ist eine Übersicht aller Variablen, die in einem (Basic-)Programm verwendet werden. Es gibt verschiedene Möglichkeiten, die Erweiterung »Dump« zu benutzen:

- SYS 49152,A: Listet alle im Programm verwendeten indizierten Variablen mit dem Namen A auf; also A(1), A(2) und so weiter.
- SYS 49152 ohne Zusatz: Listet alle nichtindizierten Variablen.
- SYS 49152,\$: Listet nur alle nichtindizierten Stringvariablen.
- SYS 49152,%: Listet nur alle nichtindizierten numerischen Variablen (das Gegenstück zu SYS 49152,\$).

(Herbert Kunz/tr)

```

programm : dump c000 c000 c2e2
c000 : a0 00 b1 7a f0 26 20 fd 19
c008 : ae 85 f7 20 13 b1 90 03 82
c010 : 4c 54 c1 20 73 00 a5 f7 b8
c018 : c9 24 d0 06 a9 01 85 02 a5
c020 : d0 0e c9 25 d0 06 a9 02 f6
c028 : 85 02 d0 04 a9 00 85 02 18
c030 : a5 2d a4 2e 85 14 84 15 90
c038 : c4 30 d0 02 c5 2f 90 01 a3
c040 : 60 69 02 90 01 c8 85 22 98
c048 : 84 23 a9 0d 20 d2 ff 20 43
c050 : 94 c0 20 b3 c0 20 ec c0 05
c058 : a5 d3 c9 03 f0 08 a9 20 f0
c060 : 20 d2 ff 4c 58 c0 a9 3d 1f
c068 : 20 d2 ff a5 09 c9 01 d0 2a
c070 : 06 20 30 c1 4c 84 c0 c9 4a
c078 : 02 d0 06 20 17 c1 4c 84 22
c080 : c0 20 27 c1 a5 14 a4 15 0a
c088 : 18 69 07 90 a7 c8 4c 34 83
c090 : c0 4c 08 af 20 87 ea a5 a4
c098 : cb c9 39 f0 04 20 2c a8 f8
c0a0 : 60 20 87 ea a5 cb c9 40 b0
c0a8 : f0 f7 c9 39 f0 f3 a9 00 83
c0b0 : 85 c6 60 a0 00 84 09 b1 70
c0b8 : 14 10 0a a9 02 85 09 c5 88
c0c0 : 02 f0 1e d0 12 c8 b1 14 32
c0c8 : 10 d0 a9 01 85 09 c5 02 a5
c0d0 : f0 0f a5 02 d0 0c 60 a5 2c
c0d8 : 02 c9 00 f0 04 c9 02 00 15
c0e0 : 01 60 a9 91 20 d2 ff 68 17
c0e8 : 68 4c 84 c0 a0 00 b1 14 a8
c0f0 : 29 7f 20 d2 ff c8 b1 14 70
c0f8 : f0 05 29 7f 20 d2 ff a5 89
c100 : 09 c9 01 d0 08 a9 24 20 e7
c108 : d2 ff 4c 16 c1 c9 02 d0 c4
c110 : 05 a9 25 20 d2 ff 60 a0 27
c118 : 00 b1 22 aa c8 b1 22 a8 c3
c120 : 8a 20 95 b3 4c 2a c1 20 f3
c128 : a6 bb 20 dd bd 4c 1e ab 7e
c130 : 20 4f c1 a0 02 b1 22 85 bd
c138 : 25 88 b1 22 85 24 88 b1 51
c140 : 22 85 26 f0 0a b1 24 20 cb
c148 : d2 ff c8 c4 26 d0 f6 a9 fd
c150 : 22 4c d2 ff a9 00 85 f7 ed
c158 : 85 f8 20 79 00 85 f7 20 dd
c160 : 73 00 f0 0a 90 08 20 79 0d
c168 : 00 20 13 b1 90 09 85 f8 cd
c170 : f0 36 20 73 00 f0 31 85 49
c178 : f9 20 73 00 a5 f9 c9 24 fa
c180 : d0 0d a9 01 85 09 a9 80 aa
c188 : 05 f8 85 f8 4c ac c1 c9 4e
c190 : 25 f0 03 4c 08 af a9 02 20
c198 : 85 09 a9 80 05 f7 85 f7 32
c1a0 : a9 80 05 f8 85 f8 d0 04 55
c1a8 : a9 00 85 09 a5 2f a4 30 9b
c1b0 : 85 14 84 15 c4 32 d0 07 32
c1b8 : a5 14 c5 31 d0 01 60 a0 d7
c1c0 : 00 98 aa b1 14 c5 f7 d0 de
c1c8 : 07 c8 b1 14 c5 f8 00 28 5a
c1d0 : a0 02 b1 14 18 65 14 85 68
c1d8 : f9 90 01 e8 c8 b1 14 18 11
c1e0 : 65 15 85 15 8a f0 02 e6 da
c1e8 : 15 a5 f9 85 14 c5 31 d0 d5
c1f0 : ce a5 15 c5 32 d0 c8 60 1c
c1f8 : a0 04 b1 14 85 f9 aa c8 ee
c200 : b1 14 99 37 03 a9 00 99 b9
c208 : 7f 03 c8 b1 14 99 37 03 62
c210 : a9 00 99 7f 03 ca d0 e7 a9
c218 : c8 98 18 65 14 85 fb 90 5e
c220 : 01 e8 a5 15 85 fc 8a f0 ee
c228 : 02 e6 fc a9 8d 20 d2 ff 37
c230 : 20 94 c0 20 ec c0 a9 28 9a
c238 : 20 d2 ff a5 f9 85 fa a5 79
c240 : fa 38 e9 01 0a a8 b9 85 c9
c248 : 03 aa b9 84 03 20 cd fd 83
c250 : c6 fa a5 fa f0 08 a9 2c aa
c258 : 20 d2 ff 4c 3f c2 a9 29 6e
c260 : 20 d2 ff a9 3d 20 d2 ff 3e
c268 : a5 fb 85 22 a5 fc 85 23 4f
c270 : a5 09 c9 01 d0 07 20 30 52
c278 : c1 a9 03 d0 10 c9 02 d0 e2
c280 : 07 20 17 c1 a9 02 d0 05 8d
c288 : 20 27 c1 a9 05 18 65 fb 80
c290 : 85 fb 90 02 e6 fc a6 f9 5c
c298 : 8a 0a 38 e9 03 20 cd ff 0a
c2a0 : 03 b9 85 03 18 69 01 99 4c
c2a8 : 85 03 b9 85 03 d0 09 b9 1c
c2b0 : 84 03 18 69 01 99 84 03 de
c2b8 : b9 84 03 d9 3c 03 d0 08 df
c2c0 : b9 85 03 d9 3d 03 f0 03 ee
c2c8 : 4c 2b c2 ca f0 12 a9 00 fa
c2d0 : 99 84 03 99 85 03 88 88 43
c2d8 : a9 8d 20 d2 ff 4c a1 c2 19
c2e0 : 60 00 00 00 00 00 00 41

```

Listing »Variablen-Dump«. Bitte beachten Sie die Eingabehinweise auf Seite 8.

Escape-Taste für Gänsefüßchen-Modus

Beim Editieren geschieht es oft, daß man anstelle einer Cursorbewegung ein reverses Feld erhält, auch wenn man das gar nicht wünscht. Der einzige Weg zurück zur normalen Cursorkontrolle führte bis jetzt über das Drücken der RETURN-Taste. Mit dieser Routine kommen Sie durch einfachen Druck auf die »F7«-Taste aus dem »Gänsefüßchen-Modus« heraus.

Vielleicht haben Sie sich auch schon über die etwas exotisch klingenden Ausdrücke im Titel gewundert, aber sie beziehen sich auf eine typische Eigenschaft des Commodore Basic-Editors. Wenn man Anführungszeichen oder die INSERT-Taste getippt hat, wird jedes Drücken einer Cursortaste nicht durch eine direkte Bewegung des Cursors, sondern durch ein reverses Steuerzeichen quittiert. Nebenbei: dasselbe geschieht beim Drücken der Funktions-, der CLR/HOME- oder der CTRL-Taste zusammen mit den Farbentasten. Dieser Modus, in dem sich der Editor dann befindet, wird Quote-Modus, zu deutsch »Gänsefüßchen-Modus« oder programmierter Cursor genannt.

Manchmal ist das sehr erwünscht, wenn man zum Beispiel die Cursor- oder Farbsteuersequenzen ins Programm einbinden möchte. Oft ist es aber auch unerwünscht, und man überschreibt durch Unachtsamkeit eine halbe Programmzeile mit reversen »Q« oder sonstigen Steuerzeichen. Aus dem »Gänsefüßchen-Modus« heraus gelangt man nur, indem man noch einmal die Taste für Anführungsstriche oder RETURN drückt. Wie gut wäre es, wenn man mit einem einfachen Tastendruck aus dem störenden Quote-Modus herausgelangen könnte, ohne irgendein Zeichen überschreiben zu müssen.

Die Escape-Routine realisiert diese Möglichkeit vollständig. Durch simplen Druck auf die Funktionstaste »F7« werden alle Bits gelöscht, die verhindern, daß sich der Cursor normal steuern läßt. »F7« ermöglicht also das Entkommen (englisch: escape) aus diesem Modus.

Damit Sie auch immer genau wissen, wann welcher Modus aktiv ist, stehen in der rechten oberen Bildschirmcke die entsprechenden Buchstaben in weißer Farbe. Haben Sie Anführungsstriche geschrieben, dann erscheint ein »Q« als Zeichen dafür, daß der Quote-Modus eingeschaltet ist. Das »Q« steht dort so lange, bis der Modus ausgeschaltet wird, das heißt, bis entweder RETURN, Schlußzeichen oder »F7« gedrückt wurde. Wollen Sie etwas in reverser Schrift erscheinen lassen, dann müssen Sie CTRL/RVS ON drücken und schon erscheint ein weißes »R« als Abkürzung für Revers-Modus in der Bildschirmcke. Als drittes Zeichen kann noch das »I« vorkommen, nämlich dann, wenn die INSERT-Taste benutzt wurde. Im Insert-Modus kann man nicht einmal ein falsch eingetipptes Zeichen mit der DEL-Taste löschen, da nur reverse »T« erzeugt

| programm : | escape-t.cod | c000 | c08f |
|------------|-------------------------|------|------|
| c000 : | 78 a9 11 a2 c0 8d 14 03 | b4 | |
| c008 : | 8e 15 03 58 a9 02 85 fb | a5 | |
| c010 : | 60 c6 fb f0 03 4c 31 ea | 1d | |
| c018 : | a9 02 85 fb a9 00 85 fc | 4e | |
| c020 : | a5 d4 f0 11 a9 ff 85 fc | 38 | |
| c028 : | a9 11 8d 27 04 a9 01 8d | 4f | |
| c030 : | 27 d8 4c 3a c0 a9 20 8d | 13 | |
| c038 : | 27 04 a5 c7 f0 0d a9 12 | 06 | |
| c040 : | 8d 26 04 a9 01 8d 26 d8 | dd | |
| c048 : | 4c 50 c0 a9 20 8d 26 04 | 31 | |
| c050 : | a5 d8 f0 11 a9 ff 85 fc | 6a | |
| c058 : | a9 09 8d 25 04 a9 01 8d | 3b | |
| c060 : | 25 d8 4c 6a c0 a9 20 8d | 47 | |
| c068 : | 25 04 a5 d7 c9 88 d0 1b | 4e | |
| c070 : | a2 00 86 d4 86 c7 86 d8 | c1 | |
| c078 : | a5 fc f0 0f e8 86 c6 a4 | e0 | |
| c080 : | d3 88 a5 ce 91 d1 a9 9d | 64 | |
| c088 : | 8d 77 02 4c 31 ea 00 a0 | 87 | |

Listing 2. Das MSE-Ladeprogramm zu »Escape-Taste«. Bitte beachten Sie die Eingabehinweise auf Seite 6

Eingabehinweise

Wer im Besitz eines Assembler ist, kann das Assemblerprogramm (Listing 1) abtippen, alle anderen verwenden bitte den Weg über den MSE (Listing 2).

Die Startadresse des Programmes liegt bei \$C000 gleich 49152 dezimal, die Endadresse bei \$C08F oder 49295.

Das Programm wird natürlich nach dem Abtippen und Speichern wieder mit der Sekundäradresse 1, das heißt absolut geladen. Nach dem Laden muß »NEW« eingegeben werden, damit die Basic-Zeiger, die in den Bereich \$C000-\$CFFF gestellt wurden, wieder »zurechtgebogen« werden.

```

10 SYS9*4096
15 .OPT P,00
20 ;
25 ;*****
30 ;===== ESCAPE-TASTE FUER QUOTE-MODE =====
35 ;*                AUF COMMODORE 64                *
40 ;*****
45 ;*
50 ;*                CHRISTIAN SPOERRI                *
60 ;*                LANGWEID 9                        *
70 ;*                CH-8045, ZUERICH/SCHWEIZ          *
75 ;*
80 ;*                TEL.00411/463 65 94              *
85 ;*
90 ;*****
95 ;
100 ;
105 ;
110 COUNT    = $FB    ; VERZOEGERUNGSZAEHLER
115 DELAY    = $02    ; ANZAHL DURCHLAEFUE BIS
120 :        ; ZUR ABFRAGE
125 LASTKEY  = $D7    ; ASCII-CODE DER LETZTEN
130 :        ; GEDRUECKTEN TASTE
135 QUOTE    = $D4    ; FLAGGE FUER QUOTE-MODE
140 :        ; (GAENSEFUESSCHEN-MODUS)
145 INVERS   = $C7    ; RVS FLAG FUER BILDSCHIRM
150 NINSERT  = $DB    ; ANZAHL FAELLIGER INSERTS
155 FLAG     = $FC    ; FLAGGE, OB ZEICHEN GE-
160 :        ; LDESCHT WERDEN MUSS
165 BSPALTE  = $D3    ; ZEIGER AUF BILDSCHIRM-
170 :        ; SPALTE
175 ZEILE    = $D1    ; VEKTOR, ZEIGT AUF ZEILENBE-
180 :        ; GINN IM BILDSCHIRMSPEICHER
185 CHCURSOR = $CE    ; ZEICHEN UNTER CURSOR
190 NOKEYS   = $C6    ; ANZAHL ZEICHEN IM TASTATUR-
195 :        ; PUFFER
200 KBBUFFER = $277   ; BEGINN DES TASTATURPUFFERS
205 F7       = $B8    ; ASCII-CODE DER FUNKTIONS-
210 :        ; TASTE 'F7'
215 QSCODE   = $11    ; BILDSCHIRMCODE VON "Q"

```

Listing 1. Das Assemblerlisting zum Programm »Escape-Taste«

```

220 RSCODE   =#12   ;BILDSCHIRMCODE VON "R"
225 ISCODE  =#09   ;BILDSCHIRMCODE VON "I"
230 NORMIRQ =#EA31 ;EINSPRUNGADRESSE FUER NOR-
235 :        ;MALE INTERRUPTROUTINE
240 IRQVCT  =#314  ;ADRESSE DES INTERRUPTVECTOR
245 COLOR   =#01   ;FARBE DER INDIKATORBUCH-
250 :        ;STABEN (WEISS)
255 SCREEN  =#427  ;SPEICHERPLATZ IM BILD-
260 :        ;SCHIRMSPEICHER
265 CLRTBL  =#D827 ;SPEICHERPLATZ IN DER
270 :        ;COLORTABELLE
275 :
280 :
285 *= $C000
290 :
295 :        SEI          ;INTERRUPT VERHINDERN
300 :        LDA #<START ;STARTADRESSE DER NEU-
305 :        LDX #>START ;EN ROUTINE LADEN
310 :        STA IRQVCT  ;IRQ-VEKTOR AUF NEUE
315 :        STX IRQVCT+1 ;ROUTINE STELLEN
320 :        CLI          ;INTERRUPT ERLAUBEN
325 :        LDA #DELAY  ;VERZOEGERUNGS-
330 :        STA COUNT  ;ZAEHLER SETZEN
335 :        RTS         ;ZURUECK INS BASIC
340 :
345 START   DEC COUNT  ;VERZ. ZAEHLER-1
350 :        BEQ CHANGE  ;GENUEGEND UEBERSPR.
355 :        JMP NORMIRQ ;NEIN, NORMALER IRQ
360 CHANGE  LDA #DELAY  ;JA,NEUE ROUTINE
365 :        STA COUNT  ;ZAEHLER HOCHSETZEN
370 :        LDA #$00
375 :        STA FLAG   ;FLAGGE LOESCHEN
380 :        LDA QUOTE  ;EDITOR IM QUOTEMODE
385 :        BEQ NOTQUOTE ;NEIN, UEBERSPRINGEN
390 :        LDA #$FF
395 :        STA FLAG   ;FLAGGE SETZEN
400 :        LDA #QSCODE ;LADE "Q"
405 :        STA SCREEN ;BRINGE AUF BILDSCHIRM
410 :        LDA #COLOR  ;SETZE FARBE INS
415 :        STA CLRTBL ;FARBAM
420 :        JMP REVERS  ;NAECHSTE ABFRAGE
425 NOTQUOTE LDA # " " ;LEERZEICHEN AN STELLE
430 :        STA SCREEN ;DES "Q" SCHREIBEN
435 REVERS  LDA INVERS  ;EDITOR IN REVERSMODE
440 :        BEQ NOREVERS ;NEIN, UEBERSPRINGE
445 :        LDA #RSCODE ;LADE "R", PRINTE EINE
450 :        STA SCREEN-1 ;STELLE LINKS VON "Q"
455 :        LDA #COLOR  ;FARBE
460 :        STA CLRTBL-1 ;SETZEN
465 :        JMP INSERT  ;NAECHSTE ABFRAGE
470 NOREVERS LDA # " " ;LEERZEICHEN AN STELLE
475 :        STA SCREEN-1 ;DES "R" SCHREIBEN
480 INSERT  LDA NINSERT ;WURDE 'INSERT'
485 :        ;GEDRUECKT
490 :        BEQ NOINSERT ;NEIN, UEBERSPRINGE
495 :        LDA #$FF
500 :        STA FLAG   ;FLAGGE SETZEN
505 :        LDA #ISCODE ;"I" LINKS VON "R"
510 :        STA SCREEN-2 ;HINSCHREIBEN
515 :        LDA #COLOR  ;FARBE
520 :        STA CLRTBL-2 ;SETZEN
525 :        JMP KEYTEST ;ZUR ABFRAGE VON 'F7'
530 NOINSERT LDA # " " ;BLANK AN STELLE
535 :        STA SCREEN-2 ;DES "I" SCHREIBEN
540 KEYTEST  LDA LASTKEY ;LETZTE GEDRUECKTE
545 :        CMP #F7    ;TASTE GLEICH 'F7'
550 :        BNE END     ;NEIN, ZUM ENDE
555 :        LDX #$00   ;JA
560 :        STX QUOTE  ;QUOTE-MODE LOESCHEN
565 :        STX INVERS ;REVERS-FLAGGE AUS
570 :        STX NINSERT ;INSERT-MODUS AUS
575 :        LDA FLAG  ;WENN FLAGGE GLEICH
580 :        BEQ END   ;NULL, DANN ZEICHEN
585 :        ;NICHT LOESCHEN
590 :        INX       ;JETZT 1 IM X-REG
595 :        STX NOKEYS ;TASTATURBUFFER ENT-
600 :        ;HAELT EIN ZEICHEN
605 :        LDY BSPALTE ;CURSOR EINE STELLE
610 :        DEY       ;ZURUECKBEWEGEN
615 :        LDA CHCURSOR ;CHAR. UNTER CURSOR
620 :        STA (ZEILE),Y ;AUF BILDSCHIRM
625 :        LDA #*9D   ;ASCII-CODE FUER
630 :        ;CURSOR ZURUECK
635 :        STA KBBUFFER ;IN TASTATURPUFFER
640 END      JMP NORMIRQ ;ENDE, GEHE ZUR NOR-
645 :        ;MALEN IRQ-ROUTINE
READY.

```

Listing 1. Das Assemblerlisting zum Programm »Escape-Taste« (Schluß)

werden, aber ein Druck auf »F7« genügt, und das Problem ist behoben.

Das Programm wird dann mit SYS49152 aufgerufen, und ab sofort ist die Routine »Escape-Taste« in den Systeminterrupt eingebunden. Das allerdings nur solange bis RUN/STOP-RESTORE gedrückt wird, dann wird der Interruptvektor wieder direkt auf die normale Interruptroutine gelenkt und die neue Routine wird so lange übergangen, bis das Programm neu gestartet wird.

Da diese Routine für den Einsatz beim Programmieren gedacht ist, sollten dabei keine Probleme entstehen.

Als einzige Einschränkung muß gesagt werden, daß die Benutzer von Kassettengeräten bei einem Zugriff auf die Datensette die Routine mit RUN/STOP-RESTORE ausschalten sollten, da Kassettenoperationen den Interruptvektor auf spezielle Kassettenroutinen umschreiben und sich ein schon verstellter IRQ-Vektor negativ auswirkt. (Christian Spörri/aw)

Daten komprimieren — Sparen Sie Speicherplatz und Ladezeit!

Schreiben Sie öfter Programme, bei denen Sie größere Datenmengen verwalten müssen? Mit dem »Daten-Packer« wird der Aufwand an Ladezeit und Speicherplatz auf Diskette um 25 Prozent reduziert.

Dieses Programm kürzt Strings auf $\frac{3}{4}$ der normalen Länge. Wenn man zum Beispiel einen String mit 12 Zeichen Inhalt hat, und den »Daten-Packer« darauf losläßt, ist der String nur noch neun Zeichen lang, bei gleichem Inhalt!

Um ihn sich ansehen zu können, muß er natürlich erst wieder in eine lesbare Form gebracht, also »entpackt« werden.

Bei selbstgeschriebenen Adressenverwaltungen läßt sich das gut anwenden:

Bevor man die Adressen, die ja meist in Strings enthalten sind, auf der Diskette abspeichert, »packt« man sie mit dem Daten-Packer, und speichert sie dann. So spart man eine Menge wertvollen Diskettenplatz und Ladezeit.

Nach dem Einladen müssen die Adressen wieder »entpackt« werden, bevor man mit ihnen weiterarbeiten kann.

Das Programm (siehe Listing) wird mit dem MSE eingegeben und ist dann sofort einsatzbereit.

Umgang mit dem Daten-Packer

Packen: SYS 49152, A\$, B\$

In A\$ muß der zu packende String stehen. Danach enthält B\$ die gepackte Version.

Entpacken: SYS 49339, C\$, D\$

In C\$ muß der gepackte String stehen. In D\$ ist dann der entpackte String zur Weiterverarbeitung bereit.

Der C 64 lernt sprechen

Dieses Programm ist wohl die billigste Sprachein-/ausgabe, die es gibt. Man kann sie vielseitig einsetzen, zum Beispiel bei Spielen, sprechenden Menüs oder einfach als Spielerei.

Außer dem C 64 ist für dieses Programm eine Datasette erforderlich. Nachdem das Listing abgetippt und das Programm mit SYS 49231 gestartet wurde, ändern sich die Bildschirmfarben, und eine Einschaltmeldung erscheint. Jetzt stehen Ihnen 2045 Byte für Basic-Programme zur Verfügung. Der Grund dafür ist, daß die Aufzeichnung von Sprache äußerst speicherintensiv ist. Selbst der Bereich unter dem Basic-ROM (\$ A000 bis \$BFFF) wird ausgenutzt. Die Tonwiedergabe hat zwar keine Hi-Fi-Qualität, wenn man aber laut und deutlich spricht, kann man sich gut wiedererkennen. Während der Tonein-/ausgabe wird der Bildschirm abgeschaltet, damit der Prozessor nicht ständig vom Video-Controller unterbrochen wird.

Das Arbeiten mit diesem Programm ist denkbar einfach. Es stehen fünf neue Basic-Befehle zur Verfügung. Der Speicher des C 64 wird in verschiedene Abschnitte aufgeteilt (Bild 1), in denen Wörter gespeichert werden. Hinter allen Befehlen (außer **—G**) ist anzugeben, in welchen Bereich gespeichert werden soll.

x = 1 bis 16: Diese Abschnitte sind für ein- bis zweisilbige Wörter reserviert. Hier können zum Beispiel Zahlen oder kurze Wörter abgelegt werden.

x = 17: Dies ist ein Abschnitt für 4 Sekunden Sprechzeit.

Hier kann ein kleiner Satz abgelegt werden.

x = 255: Hier wird der gesamte Speicher beschrieben, also auch die Abschnitte 1 bis 17. Es stehen etwa 13 Sekunden Sprechzeit zur Verfügung.

Alle anderen Zahlen werden als 17 interpretiert.

Befehle

Doch nun zu den Befehlen:

—Lx: »learn« — Nach Eingabe dieses Befehls wird der Bildschirm abgeschaltet. Die Tonsignale werden digitalisiert und im Speicher abgelegt. **x** entscheidet, in welchen Bereich die Sprache gespeichert wird.

—Sx: »say« — Bei diesem Befehl wird ebenfalls der Bildschirm gelöscht. Aus dem Lautsprecher des Fernsehgerätes ertönt nun Ihre eigene Stimme. Im Gegensatz zu dem Befehl »learn« benötigt dieser Befehl keinerlei Hardware. Er kann also ohne irgendwelchen Umbau benutzt werden. **x** entscheidet, aus welchem Bereich die Sprache entnommen wird.

—P''Filename''x: »put« — Mit diesem Befehl kann die Sprache, die sich im Speicher befindet, auf Diskette gespeichert werden. Da das Speichern von Sprache sehr aufwendig ist, belegen 13 Sekunden Sprechzeit 178 Blöcke. **x** gibt an, in welchem Bereich die Sprache gespeichert wird.

—G''Filename'': »get« — Dieser Befehl wird zum Laden der Sprache benutzt.

—R: »rate« — Mit diesem Befehl läßt sich die Sprechgeschwindigkeit einstellen. Nach dem Befehl muß ein numerischer Aus-

Natürlich kann an Stelle von A\$, B\$, und D\$ jede andere Stringvariable stehen.

Das geht doch nicht mit rechten Dingen zu, wird der eine oder andere Leser jetzt vielleicht denken.

Nun, dieses Programm kürzt 4 Byte auf 3 Byte. Ein Byte hat bekannterweise 8 Bit, gebraucht werden aber nur 6 Bit, um die ASCII-Werte darzustellen, aus denen Datenstrings ja bestehen.

Mit diesen 6 Bit kann man alle alphanumerischen Zeichen und einige Sonderzeichen darstellen. Der Algorithmus des Programms besteht nun darin, vom ASCII-Wert des zu packenden Byte 32 abzuziehen, um eine Darstellung mit 6 Bit zu ermöglichen. Dann wird jedes vierte Byte auf die vorhergehenden 3 Byte verteilt, bei denen die letzten 2 Bit nach unserer Subtraktion nun nicht mehr gebraucht werden. »Überhang-Bytes«, also die letzten Zeichen eines Strings, die keine vollen 4 Byte mehr ergeben, werden nicht gepackt, sondern nur an den gepackten String angehängt.

Beim Entpacken läuft der Vorgang in umgekehrter Reihenfolge ab. (Joachim Matzner/gk)

PROGRAMM : PACKROUTOBJ C000 C1EB

```

C000 : 20 9C C0 20 6D C1 A0 00 0A
C008 : A2 00 B1 22 38 E9 20 0A C2
C010 : 0A 9D EE C1 E8 C8 E0 03 3B
C018 : D0 F0 B1 22 38 E9 20 8D 7F
C020 : F1 C1 AC F3 C1 18 6E F1 16
C028 : C1 6E EE C1 6E F1 C1 6E 6E
C030 : EE C1 6E F1 C1 6E EF C1 AB
C038 : 6E F1 C1 6E EF C1 6E F1 88
C040 : C1 6E F0 C1 6E F1 C1 6E 07
C048 : F0 C1 AD EE C1 99 F5 C1 A6
C050 : C8 AD EF C1 99 F5 C1 C8 05
C058 : AD F0 C1 99 F5 C1 C8 8C CA
C060 : F3 C1 AD F4 C1 18 69 04 CB
C068 : 8D F4 C1 CD F2 C1 F0 28 EB
C070 : 18 69 04 CD F2 C1 F0 02 FD
C078 : B0 06 AC F4 C1 4C 08 C0 15
C080 : AE F3 C1 AC F4 C1 B1 22 96
C088 : 38 E9 20 9D F5 C1 C8 E8 D3
C090 : CC F2 C1 D0 F1 8E F3 C1 47
C098 : 20 BB C1 60 20 FD AE 20 FF
C0A0 : 8B B0 85 64 84 65 20 A3 AC
C0A8 : B6 A0 00 B1 47 8D F2 C1 15
C0B0 : C8 B1 47 85 22 C8 B1 47 91
C0B8 : 85 23 60 20 9C C0 20 6D 16
C0C0 : C1 A8 A2 00 B1 22 9D EE FE
C0C8 : C1 C8 E8 E0 03 D0 F5 20 12
C0D0 : 76 C1 AC F3 C1 A2 00 8D 7D
C0D8 : EE C1 99 F5 C1 E8 C8 E0 14
C0E0 : 04 D0 F4 8C F3 C1 AD F4 09
C0E8 : C1 18 69 03 8D F4 C1 CD 93
C0F0 : F2 C1 F0 52 69 03 CD F2 15
C0F8 : C1 F0 23 B0 06 AC F4 C1 2D
C100 : 4C C2 C0 AC F4 C1 AE F3 73
C108 : C1 B1 22 18 69 20 9D F5 27
C110 : C1 C8 E8 CC F2 C1 D0 F1 6D
C118 : 8E F3 C1 4C 69 C1 A2 00 C9
C120 : AC F4 C1 B1 22 9D EE C1 3B
C128 : C8 E8 E0 03 D0 F5 2C EE 48
C130 : C1 30 19 70 17 2C EF C1 74
C138 : 30 12 70 10 2C F0 C1 30 41
C140 : 0B 70 09 4C 03 C1 AD F3 2C
C148 : C1 4C 69 C1 20 76 C1 AC D8
C150 : F3 C1 A2 00 8D EE C1 99 5A
C158 : F5 C1 E8 C8 E0 04 D0 F4 DC
C160 : AD F3 C1 18 69 04 8D F3 4F
C168 : C1 20 BB C1 60 A9 00 8D CF
C170 : F4 C1 8D F3 C1 60 A9 00 EC
C178 : 8D F1 C1 AA 18 2E F0 C1 FE
C180 : 2E F1 C1 2E F0 C1 2E F1 97
C188 : C1 2E EF C1 2E F1 C1 2E 6A
C190 : EF C1 2E F1 C1 2E EE C1 F6
C198 : 2E F1 C1 2E EE C1 2E F1 8F
C1A0 : C1 8D EE C1 4A 4A 18 69 5E
C1A8 : 20 9D EE C1 E8 E0 03 D0 CE
C1B0 : F0 AD F1 C1 18 69 20 8D 94
C1B8 : F1 C1 60 20 FD AE 20 8B 93
C1C0 : B0 85 49 84 4A 85 64 84 81
C1C8 : 65 20 A3 B6 AD F3 C1 20 BF
C1D0 : 75 B4 A0 00 B9 61 00 91 91
C1D8 : 49 C8 C0 03 D0 F6 A0 00 5D
C1E0 : B9 F5 C1 91 62 C8 C4 61 79
C1EB : D0 F6 60 A5 B6 D0 25 33 ED
    
```

Das Listing zum »Datenpacker«. Eingabe mit dem MSE. Beachten Sie bitte die Eingabehinweise auf Seite 8

```

programm : sprache      c000 c258
-----
c000 : 93 90 20 20 20 20 20 20 20 ab
c008 : 2a 2a 2a 2a 20 56 4f 49 43 62
c010 : 45 20 36 34 20 53 59 53 22
c018 : 54 45 4d 20 56 32 2e 30 76
c020 : 20 2a 2a 2a 0d 0d 31 39 96
c028 : 38 35 20 42 59 20 4d 2e 73
c030 : 4b 4c 49 4e 47 45 52 20 e6
c038 : 20 32 30 34 35 20 42 41 e4
c040 : 53 49 43 20 42 59 54 45 d8
c048 : 53 20 46 52 45 45 00 a9 59
c050 : 00 a0 c0 20 1e ab a9 0f d8
c058 : 8d 21 d0 a9 0b 8d 20 d0 1e
c060 : 4c 42 c2 00 00 00 20 73 e5
c068 : 00 08 c9 5f 0f 04 28 4c 33
c070 : e7 a7 28 20 73 00 08 c9 24
c078 : 4c f0 24 c9 53 f0 2b ea be
c080 : 4c 3a c1 ea 20 9b b7 08 85
c088 : e0 11 90 04 ea 4c ad c1 e1
c090 : ea a9 06 69 09 ca d0 fb 20
c098 : 85 fb 69 09 85 45 00 28 a8
c0a0 : 20 9b b7 08 20 88 c0 4c 5f
c0a8 : 5d c1 28 20 9b b7 08 20 cc
c0b0 : 88 c0 4c dd c1 28 20 13 6b
    
```

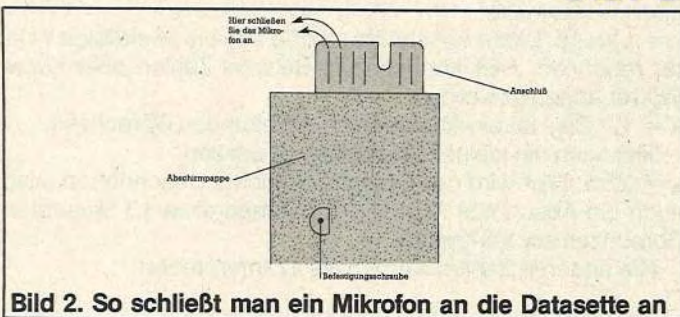
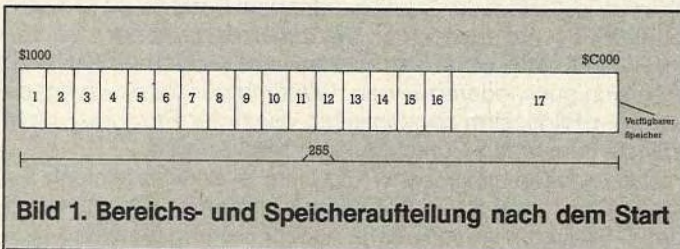
```

c0b8 : c1 ea ea ea ea ea ea ea 8e
c0c0 : a2 08 a0 01 20 ba ff a2 cc
c0c8 : 3c a0 03 a5 ff 20 bd ff c2
c0d0 : a9 00 a2 00 a0 00 20 d5 58
c0d8 : ff ea 4c e4 a7 20 13 c1 47
c0e0 : 28 20 9b b7 08 20 88 c0 1b
c0e8 : a9 36 85 01 a2 08 20 ba 8e
c0f0 : ff a2 3c a0 03 a5 ff 20 01
c0f8 : bd ff a2 00 a4 fb 86 fd 9e
c100 : 84 fe a9 fd a2 01 a4 fc ec
c108 : 20 d8 ff a9 37 85 01 28 bd
c110 : 4c e7 a7 20 73 00 c9 22 e0
c118 : f0 03 4c 08 af ea a0 00 73
c120 : 20 73 00 c9 22 f0 0b 99 3c
c128 : 3c 03 c8 c0 10 f0 03 4c 5d
c130 : 20 c1 ea 84 ff 60 00 00 7f
c138 : 00 00 c9 47 f0 0c c9 50 cb
c140 : f0 9b c9 52 f0 07 28 4c 3b
c148 : 08 af 4c b5 c0 28 20 9b f7
c150 : b7 08 e0 14 90 03 4c 48 a9
c158 : 20 c1 e3 c1 ea ad 11 d0 5b
c160 : 29 ef 8d 11 d0 a9 7f 8d 7a
c168 : 0d dc 78 a9 00 85 fa a5 9a
c170 : fb 85 fb a0 00 a2 08 ad d1
c178 : 0d dc 4a 4a 4a 4a 85 fe da
c180 : 06 ff a5 ff 05 fe 85 ff 4d
    
```

```

c188 : ca d0 ec 91 fa 86 ff 88 1c
c190 : d0 e3 e6 fb a5 fb c5 fc d6
c198 : d0 d9 ad 11 d0 09 10 8d 93
c1a0 : 11 d0 58 a9 81 8d 0d dc d7
c1a8 : 28 4c e7 a7 ea e0 ff f0 7d
c1b0 : 09 a9 9f 85 fb a9 c0 85 41
c1b8 : fc 60 a9 10 85 fb a9 c0 b1
c1c0 : 85 fc 60 8a 69 0f 8d 0d 8c
c1c8 : c2 28 4c e7 a7 00 00 00 29
c1d0 : 00 00 00 00 00 00 00 00 d1
c1d8 : 00 00 00 00 00 a9 36 85 0a
c1e0 : 01 a9 7f 8d 0d dc 78 ad 3c
c1e8 : 11 d0 29 ef 8d 11 d0 ea 24
c1f0 : a9 00 85 bb a5 fb 85 bc 3c
c1f8 : a0 00 a2 08 b1 bb 0a 48 f4
c200 : 90 0a a9 0f 8d 18 d4 a9 22
c208 : 00 8d 18 d4 4c 1c c2 ea f6
c210 : ea ea ea ea ea ea ea ea 0f
c218 : ea ea ea ea ea ea ea ea 17
c220 : ea ea 68 ca d0 d8 88 d0 8a
c228 : d1 e6 bc a5 bc c5 fc d0 e0
c230 : c7 ad 11 d0 09 10 8d 11 96
c238 : d0 58 a9 37 85 01 ea 4c 2a
c240 : a3 c1 a9 66 8d 08 03 a9 73
c248 : c0 8d 09 03 a9 10 85 34 0b
c250 : 85 38 20 44 a6 4c 74 a4 6a
    
```

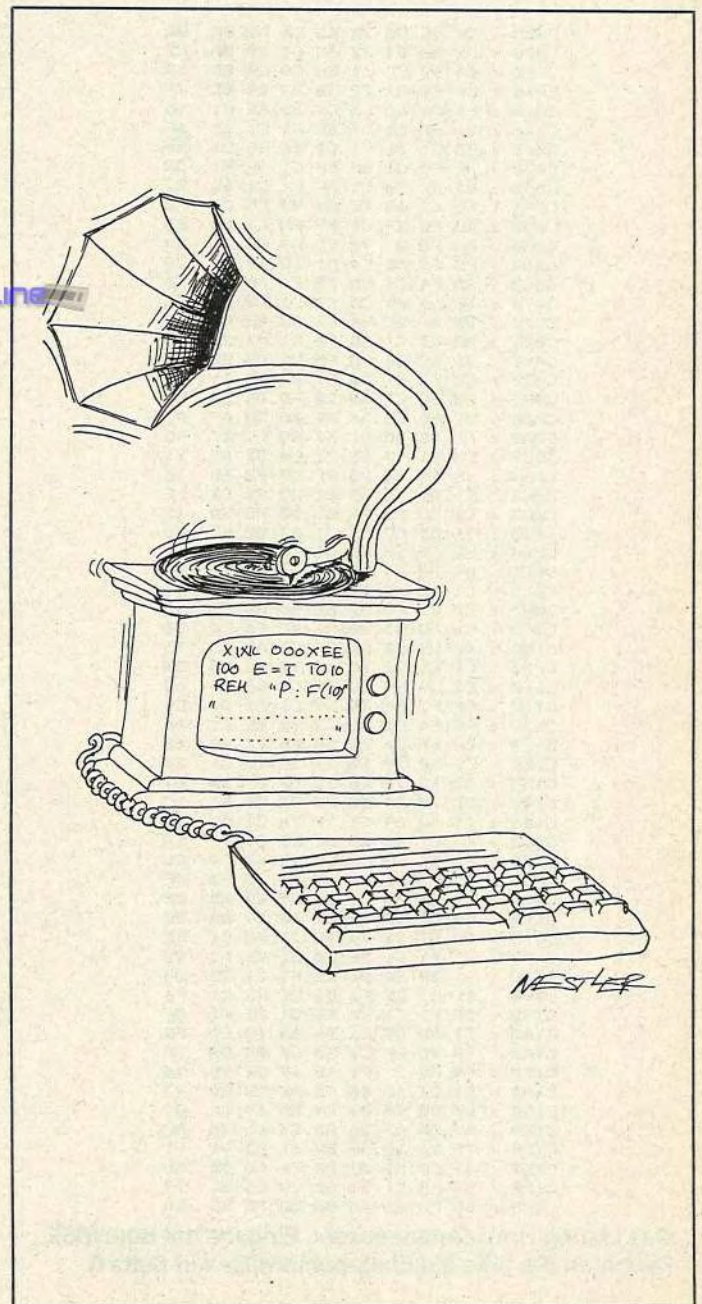
Listing zum Programm »Der C 64 lernt sprechen«. Bei der Eingabe Seite 8 beachten.



druck folgen, der die Geschwindigkeit angibt. Erlaubt sind Zahlen von 0 bis 19. Dabei entspricht 19 der schnellsten Sprechgeschwindigkeit und 0 der langsamsten. Alle anderen Zahlen werden mit einer Fehlermeldung quittiert. Die Grundeinstellung ist 13. Das ist etwa die normale Sprechgeschwindigkeit.

Jetzt werden Sie sich fragen, wie man in den Computer hineinspricht. Dazu ist natürlich ein Mikrofon erforderlich und somit auch ein wenig Umbau. Aber keine Angst, der Umbau hält sich in Grenzen. Es ist jedoch erforderlich, die Datasette zu öffnen. Lösen Sie dazu die vier Schrauben auf der Rückseite. Nachdem der Deckel abgenommen wurde, läßt sich auf einer Seite eine Platine erkennen, die mit einer Aluminiumfolie abgeschirmt ist. Auf der Seite, wo sich die Drucktasten befinden, ragt eine Anschlußleiste unter der Folie hervor (Bild 2). Löten Sie nun an die in Bild 2 gekennzeichneten Punkte eine Mikrofonbuchse an. Das war schon der ganze Umbau. Wollen Sie die Datasette wieder als Massenspeicher benutzen, darf das Mikrofon nicht mit der Buchse verbunden sein, da Geräusche im Raum den Datentransfer zwischen Datasette und Computer stören.

(Martin Klinger/ah)



ON ERROR GOTO

Routinen, die bei einem auftretenden Programmfehler zu einer bestimmten Zeile verzweigen, sind inzwischen Standardausrüstung jeder guten Basic-Erweiterung. Das vorliegende Maschinenprogramm ist nur 88 Byte lang.

Um einen Fehler während des Programmlaufs abzufangen, kennen viele Computer den Befehl ON ERROR GOTO. Mit diesem kurzen Maschinenprogramm kann auch Ihr C 64 auf Fehler im Programm reagieren.

Nachdem das Programm (Listing 1) im Speicher steht, kann es mit SYS 50000, ZL aktiviert werden. Dabei bedeutet ZL die Zeilennummer, zu der bei einem Fehler verzweigt werden soll. Die Fehlernummer wird dann in Speicherstelle 2 gespeichert und kann mit PEEK(2) abgefragt werden. Tabelle 1 zeigt, welche Nummer welcher Fehlermeldung entspricht. Die Zeile, in der der Fehler auftrat, wird in Speicherstelle 249 und 250 gespeichert. Damit Sie zu der fehlerhaften Zeile zurückspringen können, wurde noch die Routine »GOTO X« eingebaut. Damit läßt sich zu einer berechneten Zeile springen. Aufgerufen wird diese Routine mit SYS 50076, ZL. Schauen wir uns einmal das Beispiel ab Zeile 200 im Basic-Lader an. In Zeile 210 wird mit SYS 50000, 1000 die Fehlerbehandlungsroutine auf Zeile 1000 gelegt. In Zeile 220 wird der Benutzer aufgefordert, eine Zahl einzugeben. Solange diese Zahl positiv ist, wird in Zeile 230 die Wurzel aus dieser Zahl errechnet und ausgegeben. Bei einer negativen Zahl wird zur Zeile 1000 verzweigt. Hier wird die Fehlernummer ausgegeben. In diesem Falle 14. In Zeile 1010 wird nun die Zeile errechnet, in der der Fehler auftrat und ausgegeben. Zeile 1020 schließlich ruft die »GOTO X«-Routine auf und springt in die Zeile ZL -10.

Listing 2 zeigt den dokumentierten Assemblerteil der Routine. (Kunz/tr)

| Fehlernummer | Fehlermeldung |
|--------------|-----------------------|
| 1 | TOO MANY FILES |
| 2 | FILE OPEN |
| 3 | FILE NOT OPEN |
| 4 | FILE NOT FOUND |
| 5 | DEVICE NOT PRESENT |
| 6 | NOT INPUT FILE |
| 7 | NOT OUTPUT FILE |
| 8 | MISSING FILENAME |
| 9 | ILLEGAL DEVICE NUMBER |
| 10 | NEXT WITHOUT FOR |
| 11 | SYNTAX |
| 12 | RETURN WITHOUT GOSUB |
| 13 | OUT OF DATA |
| 14 | ILLEGAL QUANTITY |
| 15 | OVERFLOW |
| 16 | OUT OF MEMORY |
| 17 | UNDEF'D STATEMENT |
| 18 | BAD SUBSCRIPT |
| 19 | REDIM'D ARRAY |
| 20 | DIVISION BY ZERO |
| 22 | TYPE MISMATCH |
| 23 | STRING TOO LONG |
| 24 | FILE DATA |
| 25 | FORMULA TOO COMPLEX |
| 27 | UNDEF'D FUNCTION |
| 28 | VERIFY |
| 29 | LOAD |

Tabelle 1. Die Fehlermeldungen des Betriebssystems, jeweils mit dem Zusatz »ERROR«.

```

100 REM *** ON ERROR UND GOTO X ***          <029>
101 DATA 32,253,174,32,107,169,165,20      <202>
102 DATA 164,21,133,247,132,248,169,116    <191>
103 DATA 160,195,141,0,3,140,1,3,96,169    <188>
104 DATA 139,160,227,141,0,3,140,1,3      <006>
105 DATA 96,138,48,30,165,58,201,255      <148>
106 DATA 240,27,134,2,165,57,164,58,133   <145>
107 DATA 249,132,250,165,247,164,248     <193>
108 DATA 133,20,132,21,32,163,168,76     <254>
109 DATA 174,167,76,116,164,138,76,58    <188>
110 DATA 164,32,253,174,32,138,173,32    <156>
111 DATA 247,183,76,143,195              <154>
112 S=0:FOR I= 50000 TO 50087 :READ D      <150>
113 POKE I,D:S=S+D:NEXT                    <177>
114 PRINT"DIE DATAZEILEN SIND"           <206>
115 IF S<>10886 THEN PRINT"FEHLERHAFT":STO <163>
    P                                       <113>
116 PRINT"IN ORDNUNG":END                  <096>
120 :                                       <106>
130 :                                       <028>
200 REM BEISPIEL FUER ON ERROR UND GOTO X <244>
210 SYS 50000,1000                          <237>
220 INPUT"EINE ZAHL ";A                     <130>
230 WU=SQR(A):PRINT"DIE WURZEL AUS"A"="WU  <242>
240 END                                       <214>
1000 PRINT"FEHLER NUMMER"PEEK(2)           <239>
1010 ZL=PEEK(249)+PEEK(250)*256-10         <039>
1020 SYS 50076,ZL
    
```

Listing 1. Der Basic-Lader zu »On-Error«. Bitte verwenden Sie zur Eingabe den neuen Checksummer (Seite 6.)

```

.ba=50000
jsr $aeFd ; prüft auf Komma
jsr $a96b ; holt Zeilennummer nach $14/$15
lda $14 ;
ldy $15 ; Zeilennummer in
sta $f7 ;
sty $f8 ; $f7/$f8 speichern
lda #$74
ldy #$c3 ; Vektor für Fehlermeldung
sta $0300 ;
sty $0301 ; auf neue Routine
rts
    
```

```

lda #$8b ;
ldy #$e3 ; Vektor für Fehlermeldung
sta $0300 ; zurückstellen
sty $0301 ;
rts
    
```

```

txa
bmi noerr ; größer 127 dann kein Fehler
lda $3a ;
cmp #$ff ; Direktmodus?
beq dirmod ; ja
stx $02 ; Fehlernummer speichern
lda $39 ;
ldy $3a ; Momentane Zeilennummer
sta $f9 ; nach $f9/$fa
sty $fa ;
lda $f7 ;
ldy $f8 ; Zeilennummer für on error
sta $14 ; nach $14/$15
sty $15 ;
go jsr $a8a3 ; goto-Befehl
jmp $a7ae ; Zur Interpreterschleife
noerr jmp $a474 ; ready ausgeben
dirmod txa
jmp $a43a ; Fehlermeldung ausgeben
routine goto x
jsr $aeFd ; prüft auf Komma
jsr $ad8a ; numerischen Wert holen
jsr $b7f7 ; nach $15/$14 wandeln
jmp go
    
```

Listing 2. Der dokumentierte Assemblerteil.

Übersichtliches Listing

Diese kleine Erweiterung spaltet beim Listen eine Programmzeile in mehrere Zeilen auf. Dadurch wird die Analyse auch komplizierter Programmteile zum Kinderspiel.

Die Idee zu dieser Erweiterung kam durch den Einzeiler-Wettbewerb im 64'er-Magazin, bei dem die abgedruckten Programme durch ihre gedrängte Darstellung schlecht lesbar waren. Damit ist nun Schluß. Wenn in einer Programmzeile ein Doppelpunkt vorkommt, so wird der darauf folgende Befehl in die nächste Zeile geschrieben.

Beispiel:

Aus

```
10 A=53280:POKE A,I:POKE A+1,I+1:=PEEK(53248+18):GOTO 10
```

wird

```
10 A= 53280
   POKE A,I
   POKE A+1,I+1
   I=PEEK(53248+18)
   GOTO 10
```

Die Erweiterung liegt in dem Bereich von 49152 bis 49239, nimmt also keinen Basic-Speicherplatz in Anspruch. Sie läßt sich aber beliebig verschieben, damit sie auch bei Basic-Erweiterungen funktioniert, die den Bereich ab 49152 benutzen (man muß nur den Wert der Variablen ADRESSE in Zeile 0 ändern). Die 87 Byte lange Erweiterung läßt sich mit POKE 2,0 einschalten und mit POKE 2,1 wieder ausschalten.

So arbeitet die Routine

Das Maschinenprogramm liegt als Basic-Lader (Listing 1) und als Source-Code (Listing 2) vor. Zuerst wird der LIST-Vektor auf die neue Routine gePOKEt, dann erfolgt ein Rücksprung zum Basic. In der neuen LIST-Routine wird abgefragt, ob es sich bei dem zu listenden Zeichen um einen Befehl handelt. Ist dies der Fall, so wird zu der Routine gesprungen, die den Befehl ausgibt.

Anschließend erfolgt eine Überprüfung auf einen Doppelpunkt. Handelt es sich nicht um dieses Zeichen, so wird normal weitergelistet. Andernfalls findet eine Überprüfung der Speicherstelle 2 statt. Ist sie nicht 0, so wird der Doppelpunkt ausgegeben und mit dem normalen Listing weitergemacht. Steht in der Speicherstelle 2 eine 0, so wird ein CHR\$(13) ausgegeben.

Im darauffolgenden Teil werden die untereinanderstehenden Teilprogrammzeilen linksbündig gemacht und ausgegeben. Danach wird wieder zum »normalen« Listing gesprungen.

Die Routine »Übersichtliches Listing« ist in erster Linie natürlich für das Arbeiten mit einem Drucker gedacht, denn eine Bildschirmausgabe mit einem Basic-Befehl pro Zeile würde den Vorteil der erhöhten Übersicht wohl bald zunichte machen, jedenfalls bei längeren Programmen. Der Einsatzschwerpunkt dieser Routine liegt damit bei der Analyse fremder, unübersichtlicher Programme mittels eines Druckers.

(Frank Barcikowski/ev)

```
0 ADRESSE = 49152:REM BELIEBIG <087>
1 REM ***** <113>
2 REM * WRITTEN BY * <207>
3 REM *FRANK BARCIKOWSKI* <009>
4 REM * MORAENENWEG 33 * <250>
5 REM *3320 SALZGITTER-1* <077>
6 REM ***** <118>
10 FOR I=ADRESSE TO ADRESSE+87 <213>
20 READ A:S=S+A:POKE I,A <215>
30 NEXT <040>
31 IF S<>10407 THEN PRINT"DATA ERROR!":END <062>
35 REM DEM JEWEILIGEN BERREICH ANPASSEN <103>
36 HI=INT((ADRESSE+11)/256) <157>
37 LO=ADRESSE+11-INT((ADRESSE+11)/256)*256 <179>
38 POKE ADRESSE+6,HI <107>
39 POKE ADRESSE+1,LO <069>
40 SYS ADRESSE:POKE 2,0:END <006>
50 REM ----- <100>
60 REM EINSCHALTEN DER ERWEITERUNG : <008>
70 REM POKE 2,0 <089>
80 REM ----- <130>
90 REM AUSSCHALTEN DER ERWEITERUNG : <104>
99 REM POKE 2,<>0 <007>
100 REM ----- <150>
101 DATA 169,11,141,6,3,169,48,141,7,3,96, <093>
    16,3,76,28,167,201,58,240,3,76,243
102 DATA 166,166,2,240,3,76,243,166,169,13 <003>
    ,32,210,255,152,72,160,2,177,95
103 DATA 170,200,177,95,133,98,134,99,162, <209>
    144,56,32,73,188,32,223,189,162
104 DATA 0,189,0,1,240,3,232,208,248,169,3 <038>
    2,32,210,255,202,208,250,104,168
105 DATA 169,32,76,243,166,48,141,7,7,96 <180>
```

© 64'er

Listing 1. »Übersichtliches Listing« als Basic-Lader. Beachten Sie die Eingabehinweise auf Seite 6.

```
20: 3000          .opt pl
25: 3000          list = $0306
26: 3000          altlist = $a71c
30: 3000          *= $3000
40: 3000 a9 0b    lda #<nlist
50: 3002 0d 06 03 sta list ;listvektor verbiegen
60: 3005 a9 30    lda #>nlist
70: 3007 0d 07 03 sta list +1
80: 300a 60      rts ;wieder zu basic
;-----
; neue listroutine -----
90: 300b 10 03   nlist bpl nint ;pruefung auf inter-
100: 300d 4c 1c a7 jmp altlist ;pretercode (>$80)
110: 3010 c9 3a   nint cmp #$3a ;code fuer
110: 3012
120: 3012 f0 03   beq doppelp ;code gefunden
130: 3014 4c f3 a6 jmp $a6f3 ;weiterlisten
;-----
; doppelpunkt gefunden -----
137: 3017 a6 02   doppelp ldx 2 ;2 laden
138: 3019 f0 03   beq ok ;wenn 0 dann newlist
139: 301b 4c f3 a6 jmp $a6f3 ;sonst altlist
140: 301e a9 0d   ok lda #13 ;return
150: 3020 20 d2 ff jsr $ffd2 ;printen
160: 3023 98     tya ;y zwischenspeichern
170: 3024 48     pha ;fuer spaeter
;-----
; berechnung der zu -----
; printenden spaces -----
210: 3025 a0 02   ldy #2
220: 3027 b1 5f   lda ($f),y ;zeilennr. lo
230: 3029 aa     tax ;merken
240: 302a c8     iny ;zeiger erhoehen
250: 302b b1 5f   lda ($f),y ;zeilennr. hi
260: 302d 05 62   sta $62
270: 302f 06 63   stx $63
280: 3031 a2 90   ldx #$90
290: 3033 38     sec
300: 3034 20 49 bc jsr $bc49 ;zeilennr. in fac
310: 3037 20 df bd jsr $bddf ;fac nach ascii und $0100
; nach $0100
330: 303a a2 00   ldx #0
340: 303c bd 00 01 loop lda $100,x
350: 303f f0 03   beq endloop ;0 = stringende
360: 3041 e8     inx ;zaehler fur space
370: 3042 d0 f8   bne loop
;-----
; space x mal -----
380: 3044 a9 20   endloop lda #32 ;space laden
400: 3046 20 d2 ff prin jsr $ffd2 ;printen
410: 3049 ca     dex
420: 304a d0 fa   bne prin ;schon fertig
430: 304c 68     pla ;y wiederholen
435: 304d a8     tay
440: 304e a9 20   lda #32 ;space in accu
450: 3050 4c f3 a6 jmp $a6f3 ;accu printen
; und weiterlisten
```

Listing 2. Assembler-Listing zu »Übersichtliches Listing«. Es dient nur zur Dokumentation, braucht also nicht eingegeben zu werden.

Programmieren Sie Ihre Tastatur!

Mit diesem Programm werden nicht nur die Funktionstasten, sondern die gesamte Tastatur frei programmierbar.

Wenn Sie das Programm KEY-PROGGER (Listing 1) eingegeben haben, können Sie es mit LOAD "KEY-PROGGER",1,1 oder, wenn Sie ein Diskettenlaufwerk besitzen, mit LOAD "KEY-PROGGER",8,1 laden. Nach dem Laden geben Sie am besten den Basic-Befehl NEW ein, da Sie sonst später eine OUT OF MEMORY-Meldung erhalten. Nun sind Sie soweit, daß Sie das Programm mit SYS 49152 starten können. Nach dem Starten ist jedoch keine Veränderung zu bemerken. Durch den SYS-Befehl wird nur die Tastaturabfrage über eine neue Routine geleitet, welche die Tastenfunktionen aus einer Tabelle liest, die im RAM steht, und daher verändert werden kann. Wie aber ändert man nun die Tastenbelegung? Dazu ist der leistungsfähige #-Befehl implementiert. Damit KEY-PROGGER mit möglichst vielen Erweiterungen zusammenarbeitet, ist es möglich, diesen Befehl ein- und auszuschalten. Um also die Tastenbelegung zu ändern, muß dieser Befehl zuerst eingeschaltet werden. Dies geschieht mit SYS 49152+3. Ausgeschaltet wird er mit SYS 49152+6. Die Anwendung des #-Befehls ist sehr einfach. Sie geschieht in der Form:

```
#nr,string
nr entspricht in den allermeisten Fällen dem ASCII-Wert eines Zeichens. Alle möglichen Tastaturordnungen sind in Bild 1 aufgeführt. Wo dies nicht so ist, kann die Nummer mit einem kleinen Programm festgestellt werden:
10 FOR I=0 TO 255
20 #I,STR$(I)
30 NEXT I
```

Nachdem dieses kleine Programm mit RUN gestartet wird, erscheint beim Druck auf jede Taste eine Zahl auf dem Bildschirm. Diese Zahl ist die Nummer der Taste. Es sind jedoch nicht alle 256 Werte von der Tastatur aus erreichbar. Insgesamt erhält man jedoch 187 programmierbare Tastenfunktionen. Diese Zahl reduziert sich jedoch noch einmal, da es nicht sinnvoll ist, manche Tasten, wie zum Beispiel die RETURN-Taste umzudefinieren, da diese wichtige Funktionen erfüllen.

Mit dem #-Befehl können jeder Taste bis zu 10 Zeichen zugeordnet werden. Versuchen Sie es doch einmal, zum Beispiel mit:

```
#12,"LIST"+CHR$(13)
```

Wenn Sie nun gleichzeitig CTRL und L drücken, erscheint der Befehl LIST auf dem Bildschirm und wird gleich ausgeführt. Haben Sie nach einer Zeit des Herumprobierens den Wunsch,

| Tastencodes: | |
|--|--------|
| - 1 2 3 4 5 6 7 8 9 0 + - £ HOME DEL | F1 133 |
| 95 49 50 51 52 53 54 55 56 57 48 43 45 92 19 20 | |
| CTRL Q W E R T Y U I O P @ * ! RESTORE | F3 134 |
| 81 87 69 82 84 89 85 73 79 80 64 42 94 | |
| STOP SHIFT A S D F G H J K L : ; = RETURN | F6 135 |
| 3 LOCK 65 83 68 70 71 72 74 75 76 58 59 61 13 | |
| C= SHIFT Z X C V B N M , . / SHIFT ! - | F7 136 |
| 90 88 67 86 66 78 77 44 46 47 17 29 | |
| SPACE 32 | |
| Zusammen mit SHIFT: | |
| - ! " # \$ % & / () 0 + - £ CLR INST | F2 137 |
| 33 34 35 36 37 38 39 40 41 219 221 169 147 148 | |
| CTRL Q W E R T Y U I O P @ * π RESTORE | F4 138 |
| 209 215 197 210 212 217 213 201 207 208 186 192 222 | |
| RUN SHIFT A S D F G H J K L [] = RETURN | F6 139 |
| 131 LOCK 193 211 196 198 199 200 202 203 204 91 93 141 | |
| C= SHIFT Z X C V B N M < > ? SHIFT ! - | F8 140 |
| 218 216 195 214 194 206 205 60 62 63 145 157 | |
| SPACE 160 | |
| Zusammen mit C=: | |
| - 1 2 3 4 5 6 7 8 9 0 + - £ CLR DEL | F2 141 |
| 129 149 150 151 152 153 154 155 166 220 168 | |
| CTRL Q W E R T Y U I O P @ * π RESTORE | F4 142 |
| 171 179 177 178 163 183 184 162 185 175 164 223 | |
| RUN SHIFT A S D F G H J K L [] = RETURN | F6 143 |
| LOCK 174 172 187 165 180 181 161 182 | |
| C= SHIFT Z X C V B N M < > ? SHIFT ! - | F8 144 |
| 173 189 188 190 191 170 167 | |
| SPACE | |
| Zusammen mit CTRL: | |
| - 1 2 3 4 5 6 7 8 9 0 + - £ HOME DEL | F1 145 |
| 6* 144 5* 28* 159 156 30* 31* 158 18* 146 28* | |
| CTRL Q W E R T Y U I O P @ * ! RESTORE | F3 146 |
| 17* 23 5* 18* 20* 25 21 9 15 16 0 30* | |
| STOP SHIFT A S D F G H J K L : ; = RETURN | F5 147 |
| LOCK 1 19* 4 6* 7 8 10 11 12 27 29* 31* | |
| C= SHIFT Z X C V B N M , . / SHIFT ! - | F7 148 |
| 26 29 3* 22 2 141 3* | |
| SPACE | |
| *Die Tastenfunktion kann auch auf andere Weise erzeugt werden, zum Beispiel CTRL-2 = CTRL-E oder CTRL-M = RETURN | |

Bild 1. Tastaturzuordnung auf einen Blick

```
programm : key-progger      c000 c1b4
c000 : 4c 72 c1 4c 82 c0 4c 67 ad
c008 : c1 4c 4b c0 4c 7f c1 4c 3b
c010 : 97 c1 4c af c1 a9 00 a0 3b
c018 : a0 85 f7 84 f8 a2 20 a0 70
c020 : 00 b1 f7 91 f7 c8 d0 f9 26
c028 : e6 f8 ca d0 f4 60 a9 00 50
c030 : a0 e0 85 f7 84 f8 a2 20 7b
c038 : a0 00 b1 f7 91 f7 c8 d0 e1
c040 : f9 e6 f8 ca d0 f4 a9 35 09
c048 : 85 01 60 a9 b6 a0 c1 85 1e
c050 : f7 84 f8 a2 00 a0 00 8a 36
c058 : 91 f7 c8 a9 00 91 f7 18 e9
c060 : a5 f7 69 0a 85 f7 a5 f8 3d
c068 : 69 00 85 f8 e8 d0 e6 60 c3
c070 : a7 8d a0 c0 a2 4c 78 8e ab
c078 : 34 eb 8d 35 eb 8c 36 eb 00
c080 : 58 60 a9 e3 a0 c0 8d 08 45
c088 : 03 8c 09 03 60 a9 b6 a0 e4
c090 : c1 85 f7 84 f8 86 f9 a2 93
c098 : 00 86 fa a5 f9 0a 85 fb 4d
c0a0 : a5 fa 2a 85 fc 06 f9 26 32
c0a8 : fa 06 f9 26 fa 06 f9 26 fd
c0b0 : fa 18 a5 f9 65 fb 85 f9 9f
c0b8 : a5 fa 65 fc 85 fa 18 a5 af
c0c0 : f7 65 f9 85 f7 a5 f8 65 f4
c0c8 : fa 85 f8 a0 00 84 c6 b1 7a
c0d0 : f7 99 f7 02 f0 0a e6 c6 3a
c0d8 : c8 c0 0a f0 03 4c cf f4
c0e0 : 4c 42 eb 20 73 00 c9 23 f1
c0e8 : f7 65 f9 85 f7 a5 f8 65 f4
c0f0 : 20 9b b7 86 f9 a2 00 a9 a4
c0f8 : b6 a0 c1 85 f7 84 f8 86 b4
c100 : fa a5 f9 0a 85 fb a5 fa 51
c108 : 2a 85 fc 06 f9 26 fa 06 bd
c110 : f9 26 fa 06 f9 26 fa 18 89
c118 : a5 f9 65 fb 85 f9 a5 fa 47
c120 : 65 fc 85 fa 18 a5 f7 65 1d
c128 : f9 85 f7 a5 f8 65 fa 85 48
c130 : f8 20 fd ae 20 9e ad 24 83
c138 : 0d 30 05 a2 16 4c 3a a4 e9
c140 : 20 a6 b6 c9 0b 90 05 a2 29
c148 : 17 4c 3a a4 aa a0 00 e8 2a
c150 : ca f0 09 b1 22 91 f7 c8 2b
c158 : c9 0d d0 f4 c0 0b f0 04 ab
c160 : a9 00 91 f7 4c ae a7 a9 99
c168 : e4 a0 a7 8d 08 03 8c 09 15
c170 : 03 60 20 15 c0 20 2e c0 95
c178 : 20 4b c0 20 70 c0 60 20 41
c180 : fd ae 20 d4 e1 a9 b6 a0 fe
c188 : c1 85 f7 84 f8 a9 f7 a2 9c
c190 : b7 a0 cb 20 d8 ff 60 20 dd
c198 : fd ae a9 00 85 0a 20 d4 2a
c1a0 : e1 a9 01 85 b9 a5 0a a2 7d
c1a8 : b6 a0 c1 20 d5 ff 60 20 89
c1b0 : 4b c0 20 70 c0 60 ff ff 80
```

Listing 1. Hauptprogramm »Frei programmierbare Tastatur«. Bitte beachten Sie die Eingabehinweise auf Seite 8.

die alte Tastaturbelegung wieder herzustellen, so ist dies ganz einfach mit dem Befehl SYS 49152+9 möglich.

Außerdem kann die Belegung der Tastatur gespeichert und wieder geladen werden. Die Befehle dazu sind:

Speichern: SYS 49152+12, "name",ga

Laden: SYS 49152+15, "name",ga

»ga« ist dabei die Geräteadresse, also 1 für Kassettenrecorder und 8 für Diskettenlaufwerk. So kann jeder seine »ideale« Lösung auf einfachste Weise festhalten.

Tastatur-Generator

Die Anwendungsgebiete der programmierten Tastaturen sind nahezu unbegrenzt. So läßt sich der C 64 wie der SX 64 an ein Diskettenlaufwerk anpassen. Mit:

```
#131,"LOAD"+CHR$(34)+"*" +CHR$(34)+",8"
```

wird die RUN/STOP-Taste so umdefiniert, daß nicht mehr ein Kassettenprogramm, sondern das erste Programm von der Diskette geladen wird.

Da KEY-PROGGER auch ohne Probleme mit dem DOS 5.1 zusammenarbeitet, ist jetzt auch ein Directory auf Knopfdruck möglich.

Will man gleichzeitig mit KEY-PROGGER und Hypra-Load arbeiten, so muß man erst Hypra-Load und dann KEY-

PROGGER mit SYS 49152+18 statt mit SYS 49152 starten.

Am einfachsten ist die Belegung der Tastatur mit dem Tastatur-Generator. Nach Abtippen von Listing 2 speichern Sie den Tastatur-Generator zuerst ab. Nun laden Sie KEY-PROGGER wie oben beschrieben und geben anschließend NEW ein. Danach können Sie den Tastatur-Generator laden und mit RUN starten. Der Tastatur-Generator startet KEY-PROGGER automatisch. Nach zirka fünf Sekunden meldet sich der Tastatur-Generator mit der Frage, welche Taste umdefiniert werden soll. Sie brauchen nur diese Taste zu drücken. Der Computer zeigt dann, wie die Taste bisher belegt war. Danach können Sie Ihre neue Belegung in ein 10 Zeichen breites Feld eingeben. Die Taste DEL löscht die Eingabe. Mit RETURN wird die Eingabe abgeschlossen. Haben Sie einen Befehl eingegeben, der sofort nach Drücken der Taste ausgeführt werden soll, zum Beispiel RUN, so schließen Sie die Eingabe mit SHIFT RETURN ab. Drücken Sie RETURN, ohne eine neue Belegung eingegeben zu haben, so bleibt die alte Belegung erhalten.

Nach der Eingabe erscheint ein Menü, mit dem Sie die Definitionen fortsetzen können (1), die letzte Eingabe rückgängig machen (2), die neue Tastaturbelegung speichern (3), die ursprüngliche Belegung wieder herstellen (4) oder das Programm beenden können (5). Somit kann sich jeder auf einfachste Weise eine individuelle Tastaturbelegung gestalten.

(Markus Stecher/ah)

```

1000 REM      TASTATUR-GENERATOR          <198>
1010 REM      <054>
1020 REM 1985 VON MARKUS STECHER        <123>
1030 REM      <076>
1040 POKE 53280,0                       <074>
1050 POKE 53281,0                       <212>
1060 SYS 49152                           <102>
1070 SYS 49152+3                         <128>
1080 DIM B$(255)                        <096>
1090 GOSUB 1700:GOSUB 1740              <026>
1100 PRINT" (CLR,RVSON,11SPACE)TASTATUR-GEN
ERATOR (11SPACE)";CHR$(8)              <236>
1110 POKE 1,53                          <121>
1120 PRINT"WELCHE TASTE ? ";            <036>
1130 Z$=""                                <243>
1140 GET A$:IF A$="" THEN 1140           <111>
1150 IF A$=CHR$(13) THEN 1180           <050>
1160 Z$=Z$+A$                            <119>
1170 GOTO 1140                           <238>
1180 Z=VAL(Z$)                            <212>
1190 PRINT CHR$(34);CHR$(20);CHR$(Z)    <047>
1200 B$=B$(Z):CR=0                       <167>
1210 L=LEN(B$(Z))                        <052>
1220 IF RIGHT$(B$,1)=CHR$(13) THEN B$=LEFT$
(B$,L-1):CR=1                          <040>
1230 PRINT" (DOWN)ALTE BELEGUNG: ";CHR$(34)
;B$;CHR$(34);                            <199>
1240 IF CR THEN PRINT"+CHR$(13)";       <161>
1250 PRINT:PRINT" (DOWN)NEUE BELEGUNG: "
SPACE(12LEFT)";CHR$(34);CHR$(20);      <166>
1260 POKE 1,55                          <144>
1270 F$=""                                <047>
1280 GET A$:IF A$="" THEN 1280           <063>
1290 IF A$=CHR$(20) THEN PRINT:PRINT" (4UP)"
:GOTO 1250                                <248>
1300 IF A$=CHR$(13) THEN 1370           <201>
1310 IF A$=CHR$(141) THEN 1360         <090>
1320 IF LEN(F$)=10 THEN 1280           <245>
1330 PRINT A$;                            <085>
1340 F$=F$+A$                            <090>
1350 GOTO 1280                           <053>
1360 F$=F$+CHR$(13)                     <235>
1370 IF F$="" THEN F$=B$(Z)            <085>
1380 AL$=B$(Z)                            <143>
1390 B$(Z)=F$                             <160>
1400 PRINT                                <232>
1410 PRINT" (2DOWN)BITTE WAEHLEN: "     <003>
1420 PRINT" (DOWN)NAECHSTE EINGABE.....
....1"                                    <105>
1430 PRINT" (DOWN)LETZTE EINGABE LOESCHEN..
....2"                                    <045>
1440 PRINT" (DOWN)TASTATURBELEGUNG SPEICHER
N...3"                                    <094>
1450 PRINT" (DOWN)ALTE BELEGUNG HERSTELLEN.
....4"                                    <142>
1460 PRINT" (DOWN)ENDE.....
....5 (2DOWN)"                            <233>
1470 GET A$:IF A$<"1"OR A$>"5" THEN 1470 <235>
1480 ON VAL(A$)GOTO 1490,1500,1520,1610,16
60                                         <165>
1490 GOTO 1100                            <176>
1500 B$(Z)=AL$                            <233>
1510 GOTO 1100                            <196>
1520 INPUT" (CLR,2DOWN)NAME ";N$        <109>
1530 PRINT" (2DOWN)<D>ISK ODER <T>APE ? "; <250>
1540 GET G$:IF G$<"D"AND G$>"T" THEN 1540 <031>
1550 GA=1                                  <146>
1560 IF G$="D" THEN GA=8                 <152>
1570 PRINT" (2DOWN)BITTE WARTEN"        <126>
1580 GOSUB 1780                           <125>
1590 SYS 49152+12,N$,GA                 <008>
1600 GOTO 1100                            <032>
1610 PRINT" (CLR,2DOWN)SOLL WIRKLICH DIE AL
TE BELEGUNG"                              <221>
1620 PRINT" (DOWN)HERGESTELLT WERDEN (J/N)
?"                                          <103>
1630 GET A$:IF A$<"J"AND A$>"N" THEN 163
0                                          <199>
1640 IF A$="J" THEN SYS 49152+9         <184>
1650 GOTO 1100                            <082>
1660 PRINT" (CLR,2DOWN,SPACE)AUF WIEDERSEHE
N !"                                       <212>
1670 GOSUB 1780                           <215>
1680 POKE 1,53                            <183>
1690 END                                   <168>
1700 FOR I=0 TO 255                       <018>
1710 #I,STR$(I)+CHR$(13)                <057>
1720 NEXT I                               <024>
1730 RETURN                                <008>
1740 FOR I=0 TO 255                       <058>
1750 B$(I)=CHR$(I)                      <223>
1760 NEXT I                               <064>
1770 RETURN                                <048>
1780 FOR I=0 TO 255                       <098>
1790 #I,B$(I)                            <162>
1800 NEXT I                               <106>
1810 RETURN                                <090>

```

© 64'er

Listing 2. »Tastaturgenerator«. Bitte beachten Sie die Eingabehinweise auf Seite 6.

Komfortable Menü-steuerung

Mit diesem Basic-Unterprogramm läßt sich durch einfache Parameterübergabe eine Menü-Zeile frei definieren, mit der Sie auf Tastendruck bis zu 19 weitere Routinen anwählen können.

Das zirka 800 Byte lange Unterprogramm (ohne REM) erspart bei der Erstellung eines menügesteuerten Programms die immer wiederkehrenden Abfragen nach gedrückten oder nichtgedrückten Tasten.

Man ruft im laufenden Programm nur das Unterprogramm (Listing) auf und die gewünschte Menüzeile erscheint zentriert in der vorletzten Zeile des Bildschirms.

Mit den Cursor-Tasten (links/rechts) verschiebt man ein revers erleuchtetes Feld, das auf das Menü-Wort positioniert werden kann. Mit RETURN übernimmt man das Menü-Wort und verläßt das Unterprogramm. Um mehrere Menüs zu erstellen, sind die Angaben aller Menüs am Anfang des Hauptprogramms den Variablen d\$(1), d\$(2) und so weiter zuzuordnen.

Im laufenden Programm wird nun das Unterprogramm folgendermaßen aufgerufen:

AN=3:az=4:GOSUB 1000

AN=Anfang des Menü-Wortes (hier drittes Menü-Wort)

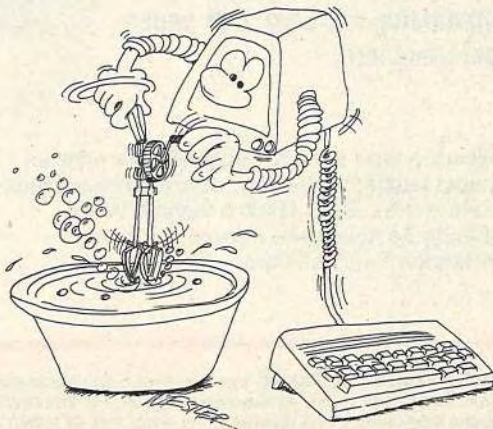
AZ=Anzahl der Menü-Worte (hier vier Worte)

Es werden also die Strings d\$(3) bis d\$(4) ausgegeben. Das Unterprogramm übergibt den Parameter kk in Abhängigkeit des gewählten Menü-Wortes und der Position der Menü-Zeile. Hat man also zum Beispiel das dritte Menü-Wort von links ausgewählt, so wird kk=3 übergeben.

Nach dem GOSUB ist eine Weiterbehandlung mit ON kk GOSUB 11,22,33,44,... am sinnvollsten.

Man erkennt sofort, welche Vorteile das Programm hat. Änderungen können sehr einfach durchgeführt werden und im laufenden Programm sind nur zwei Zeilen nötig, um eine Verzweigung zu bewirken.

Anmerkung: Die Menü-Zeile sollte nicht mehr als 38 Zeichen haben, sonst werden die Menü-Worte jeweils um ein Zeichen gekürzt dargestellt. Bei mehr als 38 Zeichen plus Anzahl der Menü-Worte erscheint eine Fehlermeldung. (M. Schacht/ah)



```

2 REM ***** MENUE-DEMO ***** <233>
4 POKE 53280,0:POKE 53281,0:PRINT CHR$(147 <206>
) <229>
5 REM * * * MENUE-WORTE EINLESEN * * * <103>
6 AM=12:DIM D$(AM+1):DIM AA(AM+1):REM AM=M <212>
AX. ANZAHL DER WORTE
7 I=1
8 READ D$(I):D$(I)=CHR$(32)+D$(I)+CHR$(32) <083>
:IF D$(I)<>" " THEN I=I+1:GOTO 8
9 DATA START,HELP,GOTO,RENUM,LIST <037>
10 DATA OLD,FIND,CTRL,CLR,NEW,RUN,ENDE, <167>
13 REM * * * * * <069>
14 POKE 214,15:PRINT:PRINT TAB(10)"<= CRSR <023>
=>(2SPACE)UND RETURN"
15 REM BEISPIEL <220>
16 AN=1:AZ=5:GOSUB 1000:REM 1.MENUE <217>
17 ON KK GOSUB 27,27,27,27,27 <071>
18 : <250>
19 AN=6:AZ=7:GOSUB 1000:REM 2.MENUE <140>
20 ON KK GOSUB 27,27,27,27,27,27,30 <200>
22 GOTO 16 <040>
25 : <001>
26 REM ANZEIGE <087>
27 POKE 214,15:PRINT:PRINT TAB(18)"(15SPAC <144>
E)"
28 POKE 214,15:PRINT:PRINT TAB(10)"GEWAEHL <253>
T: ";D$(AN+KK-1);" ";KK;"":RETURN
29 : <005>
30 PRINT CHR$(147):END <177>
32 : <116>
1000 REM ***** <036>
1010 REM **** MENUE UNTERPROGRAMM **** <120>
1020 REM ***** <056>
1030 REM **** MARIO SCHACHT **** <109>
1040 REM **** (030)4968872 **** <014>
1050 REM ***** <088>
1060 REM **** AUFRUF (BEISPIEL) **** <023>
1070 REM **** AN=1:AZ=4:GOSUB1000 **** <011>
1080 REM **** ON KK GOSUB11,22,... **** <065>
1090 REM **** - - - - - **** <067>
1100 REM **** AN: ANFANG MIT AN-TEN**** <166>
1110 REM **** STRING **** <024>
1120 REM **** AZ: ANZAHL DER MENUE-**** <003>
1130 REM **** PUNKTE **** <188>
1140 REM **** KK: 1 BIS AZ WIRD **** <011>
1150 REM **** UEBERGEHEN **** <043>
1160 REM ***** <198>
1170 REM **** VERWENDETE VARIABLEN **** <215>
1180 REM **** D$(AM);AA(AM);A$ **** <217>
1190 REM **** AM;AZ;AN;AL;DI;AY;KK **** <145>
1200 REM : **** <243>
1210 REM ***** <248>
1220 AL=0 <100>
1230 FOR I=AN TO (AN+AZ-1):AL=AL+LEN(D$(I)) <065>
:NEXT:REM AL=GESAMTLAENGE
1240 IF AL>38+AZ THEN POKE 214,22:PRINT:PR <177>
INT" MENUE-WORTE KUERZEN !":RETURN
1250 IF AL<39 THEN 1290 <097>
1260 REM WORTE KUERZEN <093>
1270 DI=AL-39:FOR I=AN TO (AN+DI):D$(I)=LEF <069>
T$(D$(I),LEN(D$(I))-1):NEXT:AL=38
1280 REM MENUE DARSTELLEN <118>
1290 POKE 214,22:PRINT:PRINT TAB((40-AL)/2 <035>
)
1300 FOR I=AN TO (AN+AZ-1) <232>
1310 PRINT CHR$(18)CHR$(30);D$(I);:NEXT <192>
1330 REM ANZEIGE <121>
1340 AA(AN)=(40-AL)/2:REM ERSTE STELLE <023>
1350 FOR I=AN TO (AN+AZ-2) <155>
1360 AA(I+1)=AA(I)+LEN(D$(I)):NEXT <074>
1380 REM STARTPOSITION <094>
1390 KK=INT(AZ/2):FL=0:FR=158:GOSUB 1550 <075>
1400 REM ABFRAGE <187>
1410 GET A$:IF A$="" THEN 1410 <190>
1420 IF A$<>CHR$(29)AND A$<>CHR$(157)AND A <149>
$<>CHR$(13) THEN 1410
1430 IF A$=CHR$(29) THEN KK=KK+1:FL=-1:FR=3 <023>
0:GOSUB 1550:IF KK=AZ THEN KK=0
1440 IF A$=CHR$(157) THEN KK=KK-1:FL=1:FR=3 <016>
0:GOSUB 1550:IF KK<0 THEN KK=AZ-1
1450 FL=0:FR=158:GOSUB 1550 <089>
1460 IF A$<>CHR$(13) THEN 1410 <017>
1470 POKE 214,22:PRINT:PRINT"(39SPACE)":RE <145>
M MENUE LOESCHEN
1500 KK=KK+1:RETURN:REM RUECKSPRUNG MIT PA <117>
RAMETER KK
1540 REM ANZEIGE GELB <160>
1550 POKE 214,22:PRINT:PRINT TAB(AA(KK+AN+ <022>
FL)CHR$(18)CHR$(FR);D$(KK+AN+FL):RET
URN

```

Listing »Komfortable Menüsteuerung«. Bitte die Eingabehinweise auf Seite 6 beachten.

Anhalter — Programmstop auf Tastendruck

Mit dieser kleinen Maschinensprach-Routine können Sie jedes Programm anhalten. Der Computer stoppt so lange, bis Sie eine Taste drücken

Dieses kleine Maschinenprogramm kann Ihnen in vielerlei Hinsicht nützlich sein. Man kann sich beispielsweise folgende Anwendungsgebiete vorstellen: In einem selbstprogrammierten Spiel wird oft eine Pause-Taste benötigt, ähnlich, wie es sie auch in manchen kommerziellen Spielen gibt. Auch das Auflisten von Programmen kann jederzeit angehalten werden und sogar der Datenfluß zur Floppy ist gegen eine Unterbrechung nicht gefeit. Mit dem Programm »Anhalter« steht es Ihnen vollkommen frei, zu entscheiden, an welchen Stellen ein Programm angehalten werden soll.

Nach dem Starten des Programms (siehe Listing 1) werden Sie zunächst gefragt, wohin das Programm verschoben werden soll (bitte Dezimalzahlen eingeben). Da das Programm vollkommen frei verschiebbar ist, brauchen Sie keine weiteren Angaben zu machen. Danach brauchen Sie nur noch die Funktionstaste F1 zu drücken um Ihr Programm zu stoppen. Soll das Programm weiter laufen, so genügt es, eine beliebige Taste zu betätigen.

Falls Sie nicht die F1-Taste, sondern eine andere mit der Anhalte-Funktion belegen möchten, so ist dies möglich. Dazu brauchen Sie nur in die Speicherstelle (Startadresse + 16)

```

0 REM *****
1 REM **   +++ ANHALTER   +++   **   <012>
2 REM **   -----**   <230>
3 REM ** WRITTEN BY GEORG KRAMER **   <107>
4 REM **   AUF DEM MOORE 15   **   <149>
5 REM **   4783 ANROECHTE   **   <233>
6 REM **   TEL.: 02947/3511   **   <140>
7 REM **   (C) 07.06.1985   **   <121>
8 REM *****   <035>
9 :   <020>
10 PRINT "{CLR,DOWN,3SPACE}++ ANHALTER ++":   <241>
    PRINT   <126>
15 INPUT "{SPACE,RVSON}STARTADRESSE : {RVDOFF   <195>
    ,SPACE}49152{7LEFT}":S \   <195>
20 IF S<0 OR S>65535 THEN 10   <022>
30 FOR I=0 TO 33:READ A:X=X+A   <226>
40 POKE S+I,A:NEXT   <221>
50 IF X<>4231 THEN PRINT"DATA-FEHLER !":EN   <146>
    D   <139>
60 POKE S+7,(S+13)/256   <098>
70 POKE S+2,(S+13)-PEEK(S+7)*256   <101>
80 SYS(S)   <039>
90 PRINT:PRINT" PROGRAMM ANHALTEN => F1"   <097>
95 END   <154>
100 REM *** MASCHINENPROGRAMM ***
101 DATA 120,169,13,141,20,3,169,192,141,2   <081>
    1,3,88,96,165,203,201,4,240,3,76
102 DATA 49,234,32,135,234,165,203,201,64,   <040>
    240,247,76,49,234
    
```

Listing 1. »Anhalter«.

Bitte beachten Sie die Eingabehinweise auf Seite 6

```

,C000 7B SEI ;Interrupt abschalten
,C001 A9 0D LDA # $0D ;Interrupt auf C00D
; setzen
,C006 A9 C0 LDA # $C0 ;
,C008 8D 15 03 STA $0315 ;
,C00B 58 CLI ;Interrupt zulassen
,C00C 60 RTS ;Rückkehr zum Basic
,C00D A5 CB LDA $CB ;hole Wert der letzten
;gedrückten Taste
,C00F C9 04 CMP # $04 ;vergleiche mit Wert
;für F1-Taste
,C011 F0 03 BEQ $C016 ;verzweige:ja
;=eig. Routine,
;nein=norm. IRQ
,C013 4C 31 EA JMP $EA31 ;verlasse
;Routine zum normalen
;IRQ
,C016 20 87 EA JSR $EA87 ;springe zur normalen
;Tastaturabfrage
,C019 A5 CB LDA $CB ;hole Wert der letzten
;gedrückten Taste
,C01B C9 40 CMP # $40 ;vergleiche mit $40
;(#64) = keine Taste
,C01D F0 F7 BEQ $C016 ;wenn keine Taste,
;dann zur Tastaturabf.
,C01F 4C 31 EA JMP $EA31 ;wenn Taste gedrückt,
;dann norm. IRQ
    
```

Listing 2. Schnell eingegeben und leicht zu verstehen — der Source-Code von »Anhalter«

den Wert der neuen Taste hineinschreiben. Diesen Wert erhalten Sie, indem Sie folgende Zeile im Direktmodus eingeben, die gewünschte Taste drücken und sich die angezeigte Zahl merken.

```
FOR I=1 TO 10000 : PRINT PEEK(203) : NEXT
```

Jetzt können Sie die Abfrage ändern:
POKE Startadresse + 16,X (X=neuer Wert)

Listing 2 erläutert die genaue Funktionsweise des Programmes.

(Georg Kramer/aw)

Variable Funktionen

Normalerweise werden Formeln in Programmen fest definiert. Mit dieser kleinen Basic-Erweiterung können Sie Funktionen auch während des Programmablaufs eingeben und ändern.

Um in einem Basic-Programm Formeln zu definieren hatte man bislang nur eine Möglichkeit: die Formel mit dem DEF FN-Befehl in einer Basic-Zeile zu definieren. Vom laufenden Programm aus war das nicht möglich. Dabei wird der Funktionswert zum Beispiel mit A=FNA(X) oder mit PRINT FNB(Y) aufgerufen.

Es geht aber auch anders, nämlich mit diesem kurzen Maschinenprogramm (Listing 1), das mit dem MSE einzugeben ist. Den Quell-Code dazu finden Sie in Listing 2. Das Programm wird mit LOAD"VARFUNKT",8,1 und nachfolgendem NEW geladen, sofern es nicht von einem anderen Programm nachgeladen wird. Das Einbinden in die Interpreter-Routine (=Initialisierung) wird mit SYS(49351) vorgenommen.

ASSEMBLERLISTING (ASSEMBLER:PROFI-ASS)

NUMERISCHER WERT EINES STRINGS

```

2
100: C010          ,OPT P2,00
105: C010          #= $C010
110: C010          CHRGET = $73
115: C010          H1      = $79
120: C010          FRNEVL = $AD9E
130: C010          H2      = 139
140: C010          H3      = $C002
150: C010          H4      = $C003
160: C010          H5      = $C004
170: C010          H6      = $C005
180: C010

;
; NUMERISCHER WERT EINES STRINGS
212: C010 A5 9D   WERT   LDA 157
214: C012 F0 05   BEQ RUN ;KEIN DIREKT-MODUS
216: C014 F2 15   LDX #21 ;ILLEGAL DIRECT ERROR
218: C016 45 37   JMP #A437
220: C019 A5 7A   RUN     LDA #7A ;PROGRAMMZEIGER SPEICHERN
230: C01B 85 8B   STA H1
240: C01D A5 7B   LDA #7B
250: C01F 85 8C   STA H2
260: C021 A0 00   LDY #0
270: C023 B1 7A   X       LDA (#7A),Y ;1. ZEICHEN NACH DEM BEFEHL EINLADEN
280: C025 F0 0D   BEQ SYNERR ;KEIN ZEICHEN DA
290: C027 C9 B2   CMP #B2
300: C029 F0 0E   BEQ PROG  ;ZEICHEN EIN '='
310: C02B E6 7A   INC #7A ;NEIN, DANN NACHSTES ZEICHEN HOLEN
320: C02D D0 F4   BNE X
330: C02F E6 7B   INC #7B ;ZURUECK ZUM 'CHECK'
340: C031 4C 23   JMP X ;"SYNTAX-FEHLER"
360: C034 A2 0B   SYNERR LDX #0B
370: C036 4C 37   JMP #A437

;
390: C039 20 73   00 PROG  JSR CHRGET ;NACHSTES ZEICHEN IN AKKU
400: C03C 20 9E   AD     JSR FRNEVL ;AUSWERTEN DES TERMES HINTER '='
410: C03F A5 7A   LDA #7A ;PROGRAMM-U.ZEILENZEIGER SPEICHERN
420: C041 8D 02   C0     STA H3
430: C044 A5 7B   LDA #7B
440: C046 8D 03   C0     STA H4
450: C049 A5 39   LDA #39
460: C04B 8D 04   C0     STA H5
470: C04E A5 3A   LDA #3A
480: C050 8D 05   C0     STA H6
490: C053 A0 00   LDY #0
500: C055 B1 64   LDA (#64),Y ;#64/#65 SIND ZEIGER AUF DEN STRINGO
ESCRIPTOR
510: C057 85 61   STA #61 ;DESCRIPTOR NACH #61-#63 HOLEN
512: C059 D0 05   BNE R1
514: C05B A2 0E   JMP #14
516: C05D 45 37   A4     JMP #A437 ;FEHLER, WEIL STRING LEER
520: C060 C8      RI     INV
530: C061 B1 64   LDA (#64),Y
540: C063 85 62   STA #62
550: C065 C8      INV
560: C068 B1 64   LDA (#64),Y
570: C06A 85 63   STA #63
580: C06C A0 00   LDY #0
590: C06E B1 8B   Z     LDA (#1),Y ;1. TEIL DES TERMS IN EINGABEPUFFER
600: C070 99 00   02     STA #0200,Y
610: C072 C8      INV
620: C074 C9 B2   CMP #B2 ;('BIS ZU '=' )
630: C076 D0 F6   BNE Z
635: C078 A5 3D   LDA #3D
637: C07A 99 FF   01     STA #03FF,Y ;'=' IN STRINGCODE WANDELN
640: C07C A2 00   LDX #0
650: C07E A1 62   LDA (#62),X
660: C080 7F 99   00     LDX #0200,Y ;2. TEIL DES TERMS UEBERTRAGEN
670: C082 C8      INV
680: C084 E6 62   INC #62
690: C086 D0 02   BNE LZ
700: C088 E6 63   INC #63
710: C08A C5 61   DEC #61
720: C08C D0 F0   L2     BNE LY
730: C08E 8A     TXA
740: C090 99 00   02     STA #0200,Y ;LETZTES ZEICHEN = 0
750: C092 A9 FF   STA #FF ;DIREKTMODUS SIMULIEREN
760: C094 A5 7A   LDA #7A
770: C096 A9 01   STA #01
780: C098 A5 7B   LDA #7B
790: C09A A9 80   STA #80
800: C09C 20 18   FE     JSR #FE18
810: C09E 20 73   00     JSR CHRGET ;1. ZEICHEN AUS EINGABEPUFFER IN AKKU
820: C0A0 F0 FF   STA #FF
830: C0A2 86 3A   STX #3A
840: C0A4 20 79   A5     JSR #A579 ;TERM IN INTERPRETERCODE WANDELN
850: C0A6 20 73   00     JSR CHRGET
860: C0A8 20 ED   A7     JSR #A7ED ;BEFEHL AUSFUEHREN
870: C0AA AD 02   C0     LDA H3 ;ZUR INTERPRETERSCHLEIFE
880: C0AC B1 7A   STA #7A
890: C0AE AD 03   C0     LDA H4
900: C0B0 85 7B   STA #7B
910: C0B2 AD 04   C0     STA H5
920: C0B4 85 39   STA #39
930: C0B6 AD 05   C0     LDA H6
940: C0B8 85 3A   STA #3A
950: C0BA A0 00   LDY #0 ;RUN-MODUS WIEDERHERSTELLEN
960: C0BC A5 9D   STA 157
970: C0BE C6 60   RTS ;ZUR INTERPRETERSCHLEIFE
1000: C0C7 A9 D2   LDA #KAUS ;VEKTOR NEU BESTIMMEN
1010: C0C9 20 80   03     JSR #A980
1020: C0CB 20 80   C0     LDA #KAUS
1030: C0CD 8D 09   03     STA #0309
1040: C0CF 60     RTS
1050: C0D1 20 73   00   AUS   JSR CHRGET ;BEFEHL "? "
1060: C0D3 C9 A6   CMP #A6
1070: C0D5 F0 FA   BEQ #FA
1072: C0D7 C9 8B   CMP #8B
1074: C0D9 F0 9C   BEQ #9C ;IF-STRUKTUR
1080: C0DB 20 79   00   ST     JSR CHRGET
1090: C0DD 4C E7   A7     JMP #A7E7 ;NEIN
1100: C0DF 20 10   C0   S     JSR WERT ;NEUEN BEFEHL AUSFUEHREN
1110: C0E1 4C E7   A7     JMP #A7AE ;ZUR INTERPRETERSCHLEIFE
1120: C0E3 20 F2   C0   T     JSR IF ;IF-BEFEHL AUSFUEHREN
1130: C0E5 4C E7   A7     JMP #A7AE ;ZUR INTERPRETERSCHLEIFE
1140: C0E7 4C FB   A8   R     JMP #A8FB
1200: C0F2 20 73   00   IF     JSR CHRGET
1200: C0F5 20 9E   AD     JSR #AD9E
1210: C0F8 20 79   00   JSR CHRGET
1220: C0FA C9 89   CMP #89
1230: C0FC F0 05   BEQ #U
1240: C0FE A9 A7   LDA #A7
1250: C101 20 FF   AE     JSR #AEFF
1260: C104 A5 61   U     LDA #61
1270: C106 D0 05   BNE V
1280: C108 20 89   A9     JSR #A909
1290: C10A F0 E2   BEQ #R
1300: C10C 20 79   00   V     JSR CHRGET
1310: C10E C9 83   CMP #83
1320: C110 4C A0   A8     JMP #A8A0
1330: C112 C9 AE   W     CMP #AE
1340: C114 F0 06   BEQ #IE
1350: C116 C9 8B   CMP #8B
1360: C118 F0 05   BEQ #IF2
1370: C11A D0 BE   BNE #E
1380: C11C 4C 10   C0   ME   JMP WERT ;JMP, WEIL NOCH INNERHALB DER IF-STRUKTUR
TUR
1390: C122 4C F2   C0   IF2  JMP IF
    
```

Listing 2.
Assembler-Quell-Listing
zu »Variable Funktionen«.

programm : varfunkt c010 c125

```

c010 : a5 9d f0 05 a2 15 4c 37 d3
c018 : a4 a5 7a 85 8b a5 7b 85 bd
c020 : 8c a0 00 b1 7a f0 0d c9 2a
c028 : b2 f0 0e e6 7a d0 f4 e6 82
c030 : 7b 4c 23 c0 a2 0b 4c 37 d4
c038 : a4 20 73 00 20 9e ad a5 c2
c040 : 7a 8d 02 c0 a5 7b 8d 03 8c
c048 : c0 a5 39 8d 04 c0 a5 3a 2c
c050 : 8d 05 c0 a0 00 b1 64 85 ce
c058 : 61 d0 05 a2 0e 4c 37 a4 20
c060 : c8 b1 64 85 62 c8 b1 64 c7
c068 : 85 63 a0 00 b1 8b 99 00 a5
c070 : 02 c8 c9 b2 d0 f6 a9 3d 85
c078 : 99 ff 01 a2 00 a1 62 99 6f
c080 : 00 02 c8 e6 62 d0 02 e6 13
c088 : 63 c6 61 d0 f0 8a 99 00 8a
c090 : 02 a9 ff 85 7a a9 01 85 1b
c098 : 7b a9 80 20 18 fe 20 73 ed
c0a0 : 00 a2 ff 86 3a 20 79 a5 98
c0a8 : 20 73 00 20 ed a7 ad 02 5d
c0b0 : c0 85 7a ad 03 c0 85 7b ca
c0b8 : ad 04 c0 85 39 ad 05 c0 df
c0c0 : 85 3a a9 00 85 9d 60 a9 e7
c0c8 : d2 8d 08 03 a9 c0 8d 09 ac
c0d0 : 03 60 20 73 00 c9 ae f0 65
c0d8 : 0a c9 8b f0 0c 20 79 00 6f
c0e0 : 4c e7 a7 20 10 c0 4c ae a3
c0e8 : a7 20 f2 c0 4c ae a7 4c e5
c0f0 : fb a8 20 73 00 20 9e ad 8d
c0f8 : 20 79 00 c9 89 f0 05 a9 96
c100 : a7 20 ff ae a5 61 d0 05 40
c108 : 20 09 a9 f0 e2 20 79 00 4a
c110 : b0 03 4c a0 a8 c9 ae f0 de
c118 : 06 c9 8b f0 05 d0 be 4c 6e
c120 : 10 c0 4c f2 c0 ff 00 ff 0e
    
```

Listing 1. »Variable Funktionen«.
Beachten Sie die Eingabeinweise auf Seite 8.

Aufruf aus dem Programm

Die Übergabe der Variablen erfolgt in der denkbar einfachsten Art: die Formel steht in einer String-Variablen wie zum Beispiel X\$. Da X\$ ein normaler String ist, kann man die Formel natürlich auch über den INPUT-Befehl oder eine GET-Schleife eingeben, wobei man bei INPUT nicht über 80 Zeichen kommt und GET eine programmbedingte Längenbegrenzung von 127 Zeichen hat. Die Syntax ist dabei die des normalen Basic mit sämtlichen Basic-Funktionen. Das heißt, daß man zum Beispiel die Formel »5 SIN X« als »5 * SIN(X)« eingibt. Dabei bedeutet X wiederum eine Variable. Natürlich kann die Formel beliebig viele Variablen, beziehungsweise beliebig verschachtelte Basic-Funktionen enthalten.

Die Syntax des neuen Befehls ist
↑ Y=B\$

Hier wird der Variablen Y der Wert des berechneten Strings X\$ zugewiesen.

Hierzu noch ein Beispielprogramm, das das Programm »Varfunkt« auf Diskette erwartet:

```

10 IF A=0 THEN A=1 : PRINT "{home}" : LOAD "VARFUNKT",8,1
20 SYS 49351
30 INPUT "[down] FUNKTION UEBER X ";X$
40 INPUT " WERT FUER X ";X
50 ↑ Y=X$
60 PRINT " FUNKTIONSWERT = ";Y
70 GOTO 30
    
```

In Zeile 10 wird beim Start die Maschinenroutine »Varfunkt« nachgeladen. Die IF-Abfrage ist nötig, da der Interpreter nach einem LOAD-Befehl das Programm erneut startet.

Der SYS-Aufruf in 20 bindet das nachgeladene Programm in das Betriebssystem ein.

In den Zeilen 30 und 40 werden Funktion und Wert für die Funktion eingegeben.

Zeile 50 schließlich übergibt den berechneten Wert aus X\$ an die Variable Y, die in 60 ausgegeben wird.

Zeile 70 schließt das Beispielprogramm ab, indem wieder in Zeile 30 zur Funktionseingabe gesprungen wird.

(Jens Weber/og)

Cross-Ref 64 — Basic-Programme unter der Lupe

Wenn man zu einem Basic-Programm eine Liste aller Sprungadressen, Schleifen und Variablen hat, ist das eine sehr gute Hilfe. Nicht nur während des Programmierens, sondern auch zur nachträglichen Dokumentation. Eine solche Liste erstellt dieses Maschinen-Programm.

Cross-Ref 64 durchsucht in zwei Durchgängen (Passes) ein Basic-Programm nach Sprungbefehlen und Variablen. Sämtliche Sprungziele und Variablennamen werden dann auf einem Drucker in Form einer Tabelle ausgegeben. Eine solche Tabelle nennt man Cross-Reference-List.

Das Basic-Programm kann dazu übrigens irgendwo im Basic-Speicher liegen. Es muß nicht bei Adresse \$0801 beginnen. Das ist wichtig, denn häufig wird der untere Basic-Speicherteil für Sprites reserviert.

1. Durchgang

Im ersten Durchlauf (Pass) werden Sprunganweisungen und FOR-NEXT-Schleifen gesucht und alle Zeilennummern der Zeilen ausgegeben, die eine Sprunganweisung enthalten oder angesprungen werden. Dabei werden alle Sprünge berücksichtigt, die das C 64-Basic 2.0 bietet. FOR-NEXT-Schleifen werden durch die Laufvariable gekennzeichnet. Eventuell vorhandene, noch nicht zu Ende geschriebene FOR-NEXT-Schleifen werden mit einem »*« markiert.

2. Durchgang

Hier werden alle Variablen eines Basic-Programms gesucht und ausgegeben.

Die Ausgabe der Cross-Reference-Liste kann in beiden Durchgängen entweder auf Bildschirm oder Drucker erfolgen. Bei Bildschirmausgabe allerdings auch im Druckerformat mit 80 Zeichen/Zeile. Das bedeutet, daß zwei Bildschirmzeilen immer eine Druckerzeile darstellen. Doch wie sinnvoll ist überhaupt eine Dokumentation auf dem Bildschirm? Das Programm ist auf einen Epson-Drucker mit Görlitz-Interface angepaßt. Es kann aber leicht so verändert werden, daß es auch für andere

Drucker geeignet ist. Es müssen nur vier Speicherzellen mit POKE-Befehlen verändert werden.

Einstellung der Geräteadresse:
POKE 50517, GA : POKE 50519, GA
Einstellung der Sekundäradresse:
POKE 50521, SA+96 : POKE 50529, SA+96

Mit einem Monitor wie dem SMON oder dem Einzeiler »Maschinenprogramme speichern« (Ausgabe 11/84) können Sie die auf Ihren Drucker angepaßte Programmversion speichern:

SYS57812a\$,x:POKE193,1s:POKE194,hs:POKE174,1e:POKE175,he:SYS62957

Dadurch ersparen Sie sich die lästigen POKE-Befehle nach jedem Laden von Cross-Ref 64. Das Programm belegt den Speicherbereich von \$C000 (49152) bis \$CAB6 (51894). Um Ihnen eventuelle Berechnungen zu ersparen: Das Highbyte (hs) von 49152 ist 192 das Lowbyte (ls) 0. Das Highbyte von 51894 (he) ist 202, das Lowbyte (le) 182.

Übersichtliches Ausgabeformat

Wie schon erwähnt, werden alle Zeilen, in denen ein Sprung erfolgt, ausgegeben. Die gezeigte Tabelle (Tabelle 1) bezieht sich auf das Beispielprogramm (Listing 1). Man kann sich auch eine Liste aller Sprungziele drucken lassen (Tabelle 2). Die Ausgabe erfolgt mit 80 Zeichen pro Zeile, dem üblichen Druckerformat.

Sollten offene FOR-NEXT-Schleifen (ohne NEXT) vorhanden sein, wird nach einem »*« die Anzahl der fehlenden NEXT-Befehle angezeigt. Sollte die Anzahl 255 betragen, heißt das, daß ein NEXT zuviel gefunden wurde.

Sollten in einer Basic-Zeile mehrere Sprünge auftreten, wird die mehrmalige Ausgabe der Zeilennummer unterdrückt.

Bei Sprungbefehlen der Form ON X GOSUB wird das Argument nach ON durch »-« ersetzt, da auch ganze Basic-Funktionen als Argument erlaubt sind. Um diese Funktionen zu berücksichtigen, wäre ein zu großer Programmieraufwand nötig. Treten nach einem THEN mehrere Sprünge auf, wird jedem Sprungziel ein THEN vorangestellt. Eine solche Befehlsfolge ist beispielsweise

```
IF X=1 THEN GOSUB 100:GOSUB 200:GOSUB 300.
```

Bei FOR-NEXT-Schleifen wird der ganze Variablenname angegeben.

Die Variablennamen werden in einer eigenen Tabelle ausgegeben (Tabelle 3). Hier werden die Namen allerdings auf zwei Zeichen gekürzt. In der Tabelle steht nach dem Namen der Variable der Variablentyp (Real, Integer oder String). Zwei Klammern »()« zeigen an, daß es sich um eine indizierte Variable handelt. Da eine Variable in einem Programmbereich sehr häufig benutzt werden kann, werden bei Mehrfachnennung einer Variablen nur noch die Zeilennummern angegeben; so lange, bis eine andere Variable aufgerufen wird. Wird einer Variablen ein anderer Wert zugeordnet (A=A+1), wird die Zeilennummer mit einem Stern »*« gekennzeichnet. Bei der sortierten

| SPRUNGBEFEHLE | | | | | |
|---------------|-----------------|--------------|----------|--------------|--------|
| IN ZEILE | SPRUNG DURCH | NACH | IN ZEILE | SPRUNG DURCH | NACH |
| 20 | GOSUB | 100 | | GOTO | 200 |
| 40 | RUN | 50 | 60 | THEN GOSUB | 100 |
| | THEN GOTO | 50 | 80 | ON - GOTO | 10,20, |
| | | 30,40 | | | |
| 100 | THEN ON - GOSUB | 100,90,70,80 | | | |

Tabelle 1. Liste der Zeilen, die Sprungbefehle enthalten

Tips & Tricks

Variablenausgabe (Tabelle 4), werden nach dem Variablennamen alle Zeilen aufgeschlüsselt, in denen die Variable vorkommt. Das Sortieren geschieht nach den folgenden Kriterien:

1. Typ: Real, Integer, Strings, Felder
2. Name alphabetisch
3. Zeilennummern in aufsteigender Reihenfolge.

Die Ausgabe kann jederzeit durch Tastendruck gestoppt werden. Eine begonnene Druckzeile wird zu Ende gedruckt. Bei einem weiteren Tastendruck fährt das Programm fort. Mit RUN/STOP kann die Ausgabe abgebrochen werden.

Tips zu Eingabe und Benutzung

Das Maschinen-Programm (Listing 2) sollten Sie mit dem MSE eingeben. Es wird durch »SYS 49152, Parameterliste, Bereich« gestartet. Die »Parameterliste« muß aus genau fünf Zeichen bestehen (Leerzeichen nicht mitgerechnet). Folgende Zeichen sind erlaubt:

1. Stelle: P — Ausgabe auf Drucker
2. Stelle: S — Ausgabe der Zeilen mit Sprungbefehlen (Tabelle 1)
3. Stelle: J — Ausgabe der angesprungenen Zeilen (nur zusammen mit »S« an zweiter Stelle möglich (Tabelle 2))
4. Stelle: V — Ausgabe der benutzten Variablen in der Reihen-

folge des Auftretens (Tabelle 3)

5. Stelle: J — Ausgabe der benutzten Variablen in sortierter Reihenfolge (Tabelle 4)

Werden andere Zeichen an der entsprechenden Stelle angegeben, dann wird der Programmpunkt nicht ausgeführt.

Die Bereichsangabe ist optional und hat die gleiche Syntax wie beim LIST-Befehl. Falls ein Zeilenbereich angegeben wird, analysiert das Programm auch nur den entsprechenden Abschnitt. Ein Beispiel:

Es sollen alle Sprungbefehle und alle Variablen im Zeilenbereich 100-200 gesucht werden. Die Ausgabe soll auf dem Drucker erfolgen. Der entsprechende SYS-Aufruf lautet dann: SYS 49152, PS.,J,100-200

Anstelle der Punkte kann dabei jedes beliebige andere Zeichen mit Ausnahme des Kommas angegeben werden.

Zum Sortieren der Variablen und für die Ausgabe der angesprungenen Zeilen wird das »Bubblesort«-Verfahren verwendet, da es besonders einfach in Assembler (Listing 3, Quellcode) programmiert werden kann. Die Variablen- und Sprungtabelle wird im Bereich \$A000 bis \$BFFF, also »unter« dem Basic-ROM, abgelegt. Falls also eine Basic-Erweiterung verwendet wird, darf dieser Speicherbereich dadurch nicht verändert werden. (Peter Rüschoff-Nadermann/hm)

| SPRUNGBEFEHLE | | | | | | | | | |
|---------------|----------------|------|-----------|----------------|--------|--|--|--|--|
| * ZIEL * | SPRUNG DURCH | * IN | ** ZIEL * | SPRUNG DURCH | * IN * | | | | |
| 10 | ON - GOTO | 80 | 20 | ON - GOTO | 80 | | | | |
| 30 | ON - GOTO | 80 | 40 | ON - GOTO | 80 | | | | |
| 50 | RUN | 40 | 50 | THEN GOTO | 60 | | | | |
| 70 | THEN ON - GSUB | 100 | 100 | THEN ON - GSUB | 100 | | | | |
| 90 | THEN ON - GSUB | 100 | 100 | GSUB | 20 | | | | |
| 100 | THEN GSUB | 60 | 100 | THEN ON - GSUB | 100 | | | | |
| 200 | GOTO | 20 | | | | | | | |

Tabelle 2. Liste der angesprungenen Zeilen

| VARIABLEN | | | | | |
|-----------|--------|----------|--------|----------|--------|
| VARIABLE | ZEILEN | VARIABLE | ZEILEN | VARIABLE | ZEILEN |
| A | 60 | 80 | 100 | 100 | |
| B | 120* | 120 | 120* | 120 | |
| D | 140 | | | G | 140 |
| E | 140 | | | K | 140 |
| E | 160* | | | G | 160 |
| D | 160 | | | K | 160 |
| D | 160 | | | K | 160 |

Tabelle 3. Alle im Beispiel-Programm verwendeten Variablen

| VARIABLEN | | | | | | | |
|-----------|--------|----------|--------|----------|--------|----------|--------|
| VARIABLE | ZEILEN | VARIABLE | ZEILEN | VARIABLE | ZEILEN | VARIABLE | ZEILEN |
| A | 60 | 80 | 100 | 100 | | | |
| B | 120 | 120 | 120* | 120* | | | |
| D | 140 | 160 | 160 | E | 140 | 160* | |
| G | 140 | 160 | | K | 140 | 160 | 160 |

Tabelle 4. Alle Variablen des Beispiel-Programms in sortierter Reihenfolge.

| programm | cross-ref | c000 | cab7 | | | | | | |
|----------|------------------------------|------|------|--|--|--|--|--|--|
| c000 | : 4c 03 c9 20 55 c0 20 e3 e8 | | | | | | | | |
| c008 | : c8 20 fe c5 4c ff c4 ad 8c | | | | | | | | |
| c010 | : b0 ca 4a 4a b0 03 20 55 50 | | | | | | | | |
| c018 | : c0 20 7e c7 20 8b c2 20 2b | | | | | | | | |
| c020 | : 36 c4 b0 01 60 20 9c c7 0e | | | | | | | | |
| c028 | : 20 8d c7 20 e3 c8 20 fe 08 | | | | | | | | |
| c030 | : c5 4c ff c4 20 c9 c0 20 47 | | | | | | | | |
| c038 | : ff c4 4c 7e c7 20 8b c2 ad | | | | | | | | |
| c040 | : 20 36 c4 b0 01 60 20 9c 8f | | | | | | | | |
| c048 | : c7 20 8d c7 20 ad c8 20 4e | | | | | | | | |
| c050 | : 16 c8 4c ff c4 a9 06 85 9a | | | | | | | | |
| c058 | : b6 20 b1 c3 20 ba c3 4c 83 | | | | | | | | |
| c060 | : 65 c0 20 b8 c2 a9 80 8d db | | | | | | | | |
| c068 | : ad ca a5 bd c9 ff d0 01 7d | | | | | | | | |
| c070 | : 60 20 e5 c3 c9 20 f0 f9 27 | | | | | | | | |
| c078 | : c9 00 f0 e6 c9 22 f0 32 30 | | | | | | | | |
| c080 | : c9 8f f0 de c9 a7 f0 22 0b | | | | | | | | |
| c088 | : c9 88 f0 1e c9 3a f0 1a fc | | | | | | | | |
| c090 | : c9 81 f0 16 20 10 c4 90 cf | | | | | | | | |
| c098 | : 09 20 96 c6 20 0f c7 20 0a | | | | | | | | |
| c0a0 | : 45 c4 a2 00 8e ad ca 4c 0a | | | | | | | | |
| c0a8 | : 71 c0 a2 80 8e ad ca 4c 4c | | | | | | | | |
| c0b0 | : 71 c0 20 e5 c3 c9 22 f0 3b | | | | | | | | |
| c0b8 | : 07 c9 00 f0 06 4c b2 c0 d1 | | | | | | | | |
| c0c0 | : 4c 71 c0 20 b8 c2 4c 71 af | | | | | | | | |
| c0c8 | : c0 a9 05 85 b6 8d ae ca 77 | | | | | | | | |
| c0d0 | : 20 b1 c3 20 ba c3 a5 bd 9a | | | | | | | | |
| c0d8 | : c9 ff f0 15 20 c3 c8 a9 16 | | | | | | | | |
| c0e0 | : 00 8d a5 ca 4c ea c0 20 c9 | | | | | | | | |
| c0e8 | : b8 c2 a5 bd c9 ff d0 01 04 | | | | | | | | |
| c0f0 | : 60 a9 01 8d a8 ca 20 e5 44 | | | | | | | | |
| c0f8 | : c3 c9 00 f0 ea c9 22 f0 25 | | | | | | | | |
| c100 | : 3b c9 8f f0 e2 c9 89 f0 a6 | | | | | | | | |
| c108 | : 1b c9 8d f0 1a c9 a7 f0 fa | | | | | | | | |
| c110 | : 19 c9 8a f0 18 c9 91 f0 c6 | | | | | | | | |
| c118 | : 17 c9 81 f0 16 c9 82 f0 2e | | | | | | | | |
| c120 | : 15 4c f6 c0 4c 53 c1 4c 30 | | | | | | | | |
| c128 | : 5b c1 4c 52 c2 4c 63 c1 61 | | | | | | | | |
| c130 | : 4c d6 c1 4c 7c c1 4c 9d 23 | | | | | | | | |
| c138 | : c1 4c f6 c0 20 e5 c3 c9 c9 | | | | | | | | |
| c140 | : 22 f0 07 c9 00 f0 06 4c 0e | | | | | | | | |
| c148 | : 3c c1 4c f6 c0 20 b8 c2 cc | | | | | | | | |
| c150 | : 4c f6 c0 20 5d c4 a9 02 f2 | | | | | | | | |
| c158 | : 4c 76 c2 20 5d c4 a9 04 3f | | | | | | | | |
| c160 | : 4c 76 c2 20 25 c4 20 19 c7 | | | | | | | | |
| c168 | : c4 90 0e 20 5d c4 ce a4 7c | | | | | | | | |
| c170 | : ca ac a4 ca a9 10 4c 76 4c | | | | | | | | |
| c178 | : c2 4c f6 c0 20 5d c4 ee 14 | | | | | | | | |
| c180 | : a6 ca a9 40 8d a3 ca 20 5f | | | | | | | | |
| c188 | : d4 c5 20 25 c4 c9 b2 f0 33 | | | | | | | | |
| c190 | : 06 20 f7 c3 4c 8a c1 20 7d | | | | | | | | |
| c198 | : d2 c4 4c f6 c0 ad a4 ca 60 | | | | | | | | |
| c1a0 | : 48 ce a6 ca 20 e5 c3 20 d3 | | | | | | | | |
| c1a8 | : 2d c4 f0 07 c9 2c f0 f1 fa | | | | | | | | |
| c1b0 | : 4c a4 c1 68 8d a4 ca 20 35 | | | | | | | | |
| c1b8 | : 5d c4 a9 20 8d a3 ca 20 47 | | | | | | | | |
| c1c0 | : d4 c5 20 25 c4 20 2d c4 af | | | | | | | | |
| c1c8 | : f0 06 20 f7 c3 4c c2 c1 ef | | | | | | | | |
| c1d0 | : 20 d2 c4 4c f6 c0 20 5d c5 | | | | | | | | |
| c1d8 | : c4 20 25 c4 c9 89 f0 07 49 | | | | | | | | |
| c1e0 | : c9 8d f0 08 4c d9 c1 a9 9b | | | | | | | | |
| c1e8 | : 0a 4c ee c1 a9 0c 20 0b 9e | | | | | | | | |
| c1f0 | : c8 ad a3 ca 29 fe 85 bd ed | | | | | | | | |
| c1f8 | : 8d a3 ca a2 00 20 d4 c5 3e | | | | | | | | |
| c200 | : 20 3c c5 20 25 c4 ce a4 b1 | | | | | | | | |
| c208 | : ca 20 d4 c7 a5 bd 20 3f 58 | | | | | | | | |
| c210 | : c2 20 ea c7 20 79 00 f0 46 | | | | | | | | |
| c218 | : 1d c9 2c d0 ef 20 f7 c3 a6 | | | | | | | | |
| c220 | : ad a5 ca c9 4b 90 e5 20 9d | | | | | | | | |
| c228 | : f7 c4 20 3c c5 20 3c c5 eb | | | | | | | | |
| c230 | : 20 3c c5 4c 0c c2 20 2e 1d | | | | | | | | |
| c238 | : c5 20 f7 c4 4c f6 c0 a0 64 | | | | | | | | |
| c240 | : 02 91 fb 20 a2 c4 a0 03 e7 | | | | | | | | |
| c248 | : a5 62 91 fb c8 a5 63 91 6c | | | | | | | | |
| c250 | : fb 60 20 25 c4 20 19 c4 63 | | | | | | | | |
| c258 | : 90 11 20 5d c4 a9 80 20 00 | | | | | | | | |
| c260 | : 0b c8 ce a4 ca ac a4 ca 52 | | | | | | | | |
| c268 | : 4c 76 c2 ce a4 ca a9 80 c2 | | | | | | | | |
| c270 | : 8d a8 ca 4c f6 c0 8d a3 80 | | | | | | | | |
| c278 | : ca 20 0b c8 20 d4 c5 20 2e | | | | | | | | |
| c280 | : 3c c5 20 e7 c7 20 d2 c4 f6 | | | | | | | | |
| c288 | : 4c f6 c0 20 00 c4 a9 00 50 | | | | | | | | |
| c290 | : 8d a7 ca 8d a5 ca 8d a6 89 | | | | | | | | |
| c298 | : ca 8d a3 ca 85 bd 8d ad 43 | | | | | | | | |
| c2a0 | : ca 85 57 a9 a0 85 58 a9 23 | | | | | | | | |
| c2a8 | : f9 8d a9 ca 85 fb a9 bf 8a | | | | | | | | |
| c2b0 | : 8d aa ca 85 fc 4c d2 c3 fb | | | | | | | | |
| c2b8 | : a0 00 b1 f9 d0 05 c8 b1 c0 | | | | | | | | |
| c2c0 | : f9 f0 0f a5 f9 85 f7 a5 a1 | | | | | | | | |
| c2c8 | : fa 85 f8 20 a5 c3 f0 11 25 | | | | | | | | |
| c2d0 | : 90 0f a5 fb 8d ab ca a5 7d | | | | | | | | |
| c2d8 | : fc 8d ac ca a9 ff 85 bd 4b | | | | | | | | |
| c2e0 | : 60 ad a7 ca 29 0e 8d a7 e3 | | | | | | | | |
| c2e8 | : ca a0 03 8c a4 ca a0 00 78 | | | | | | | | |
| c2f0 | : b1 f7 85 f9 c8 b1 f7 85 42 | | | | | | | | |
| c2f8 | : fa 60 a9 00 8d b0 ca 8d 31 | | | | | | | | |
| c300 | : b1 ca 8d b2 ca a9 ff 8d e5 | | | | | | | | |
| c308 | : b3 ca 8d b4 ca 20 79 00 ae | | | | | | | | |
| c310 | : c9 2c f0 05 a2 0b 4c 37 ee | | | | | | | | |
| c318 | : a4 20 73 00 c9 50 d0 03 12 | | | | | | | | |
| c320 | : 20 56 c5 20 73 00 c9 53 e6 | | | | | | | | |
| c328 | : d0 05 a9 08 20 92 c3 20 cc | | | | | | | | |
| c330 | : 73 00 c9 4a d0 05 a9 04 43 | | | | | | | | |
| c338 | : 20 92 c3 20 73 00 c9 56 a1 | | | | | | | | |
| c340 | : d0 05 a9 02 20 92 c3 20 23 | | | | | | | | |
| c348 | : 73 00 c9 4a d0 05 a9 01 55 | | | | | | | | |
| c350 | : 20 92 c3 20 73 00 c9 4b a3 | | | | | | | | |
| c358 | : d0 08 a9 10 20 92 c3 20 7f | | | | | | | | |

Listing 2. »Cross-Ref 64«. Das Programm sollte mit dem MSE eingegeben werden. Beachten Sie dazu bitte die Eingabehinweise auf Seite 8.

| | | | | | | | | | | | | | | | | | | | | | |
|------|---|----|----|----|----|----|----|----|----|----|------|---|----|----|----|----|----|----|----|----|----|
| c360 | : | 73 | 00 | c9 | 2c | d0 | 2b | c9 | ab | b0 | c538 | : | 20 | f7 | c4 | 60 | ad | a5 | ca | 38 | 35 |
| c368 | : | f0 | 17 | 20 | 73 | 00 | 20 | 6b | a9 | 5c | c540 | : | e9 | 0a | b0 | fc | 49 | ff | 69 | 01 | 36 |
| c370 | : | a5 | 14 | 8d | b1 | ca | a5 | 15 | 8d | 02 | c548 | : | aa | e8 | ca | f0 | 08 | a9 | 20 | 20 | c6 |
| c378 | : | b2 | ca | 20 | 79 | 00 | c9 | ab | d0 | 65 | c550 | : | f7 | c3 | 4c | 4a | c5 | 60 | a9 | 04 | 93 |
| c380 | : | 10 | 20 | 73 | 00 | 20 | 6b | a9 | a5 | cd | c558 | : | a2 | 04 | a0 | 68 | 20 | ba | ff | 20 | 4a |
| c388 | : | 14 | 8d | b3 | ca | a5 | 15 | 8d | b4 | 4c | c560 | : | b1 | ff | a9 | 68 | 20 | 93 | ff | a9 | 7a |
| c390 | : | ca | 60 | 0d | b0 | ca | 8d | b0 | ca | 55 | c568 | : | 04 | 85 | 9a | 60 | ad | b5 | ca | c9 | 29 |
| c398 | : | 60 | 20 | a2 | c4 | ec | b2 | ca | d0 | 7b | c570 | : | 35 | d0 | 19 | a9 | 37 | 85 | 01 | a9 | 80 |
| c3a0 | : | 03 | cd | b1 | ca | 60 | 20 | a2 | c4 | 6b | c578 | : | a0 | 85 | 04 | a0 | 00 | b1 | 03 | 91 | ad |
| c3a8 | : | ec | b4 | ca | d0 | 03 | cd | b3 | ca | be | c580 | : | 03 | c8 | d0 | f9 | e6 | 04 | a5 | 04 | 88 |
| c3b0 | : | 60 | a9 | a0 | 85 | 58 | a9 | 00 | 85 | 9b | c588 | : | c9 | c0 | d0 | f1 | ad | b6 | ca | 85 | ea |
| c3b8 | : | 57 | 60 | 20 | 8b | c2 | 20 | b8 | c2 | 4e | c590 | : | 7a | ad | b7 | ca | 85 | 7b | a9 | 04 | 0b |
| c3c0 | : | a5 | bd | c9 | ff | d0 | 06 | 20 | 6c | 4d | c598 | : | 20 | ae | ff | 20 | c3 | ff | a9 | 03 | fc |
| c3c8 | : | c5 | 4c | ae | a7 | 20 | 99 | c3 | 90 | 53 | c5a0 | : | 85 | 9a | ad | b5 | ca | 85 | 01 | 60 | 32 |
| c3d0 | : | ec | 60 | a5 | 2b | 85 | f9 | a5 | 2c | d2 | c5a8 | : | 20 | 09 | c4 | 20 | 87 | b4 | 20 | a6 | 6e |
| c3d8 | : | 85 | fa | 60 | ce | a4 | ca | 20 | e8 | bf | c5b0 | : | b6 | 48 | a6 | bd | d0 | 03 | 20 | c1 | 15 |
| c3e0 | : | c3 | ee | a4 | ca | 60 | ee | a4 | ca | 42 | c5b8 | : | c4 | 68 | aa | a0 | 00 | e8 | ca | f0 | c3 |
| c3e8 | : | ac | a4 | ca | b1 | f7 | 60 | 20 | 09 | e4 | c5c0 | : | 10 | b1 | 22 | 20 | f7 | c3 | c8 | c9 | 8a |
| c3f0 | : | c4 | 20 | 3e | f1 | 4c | 00 | c4 | ee | 48 | c5c8 | : | 0d | d0 | f3 | 20 | e5 | aa | 4c | be | a1 |
| c3f8 | : | a5 | ca | 20 | 09 | c4 | 20 | d2 | ff | c4 | c5d0 | : | c5 | 4c | 00 | c4 | a2 | 00 | ad | a8 | 86 |
| c400 | : | 48 | a5 | 01 | 29 | fe | 85 | 01 | 68 | 71 | c5d8 | : | ca | 0d | a3 | ca | 8d | a3 | ca | 0e | a8 |
| c408 | : | 60 | 48 | a9 | 37 | 85 | 01 | 68 | 60 | a0 | c5e0 | : | a3 | ca | b0 | 08 | e8 | e8 | e8 | e8 | 61 |
| c410 | : | 20 | 09 | c4 | 20 | 13 | b1 | 4c | 00 | da | c5e8 | : | e8 | 4c | df | c5 | a0 | 05 | bd | 7a | c5 |
| c418 | : | c4 | c9 | 3a | 10 | 06 | c9 | 30 | 30 | 21 | c5f0 | : | ca | 20 | f7 | c3 | e8 | 88 | d0 | f6 | 45 |
| c420 | : | 02 | 38 | 60 | 18 | 60 | 20 | e5 | c3 | 80 | c5f8 | : | ad | a3 | ca | d0 | e2 | 60 | ad | a9 | 7f |
| c428 | : | c9 | 20 | f0 | f9 | 60 | c9 | 00 | f0 | b3 | c600 | : | ca | 85 | fb | ad | aa | ca | 85 | fc | 52 |
| c430 | : | 04 | c9 | 3a | f0 | 00 | 60 | ad | aa | d5 | c608 | : | ad | ab | ca | 85 | 57 | ad | ac | ca | 19 |
| c438 | : | ca | cd | ac | ca | d0 | 06 | ad | a9 | b4 | c610 | : | 85 | 58 | a9 | 05 | 8d | ae | ca | 20 | 86 |
| c440 | : | ca | cd | ab | ca | 60 | 20 | 67 | c7 | 69 | c618 | : | ff | c4 | 20 | 67 | c7 | f0 | 6e | 20 | 6c |
| c448 | : | b0 | 12 | 20 | 6c | c5 | a9 | 5a | a2 | ef | c620 | : | 90 | c6 | a0 | 01 | b1 | fb | 20 | f7 | c7 |
| c450 | : | ca | 20 | 84 | c8 | ad | b5 | ca | 85 | 23 | c628 | : | c3 | c8 | b1 | fb | c9 | 20 | f0 | 03 | a2 |
| c458 | : | 01 | 4c | ae | a7 | 60 | 20 | a2 | c4 | 3b | c630 | : | 20 | f7 | c3 | 20 | ea | c6 | 20 | 3c | 1f |
| c460 | : | a0 | 04 | 91 | fb | 88 | 8a | 91 | fb | 01 | c638 | : | c5 | a0 | 03 | b1 | fb | 85 | 62 | c8 | 4b |
| c468 | : | a9 | 00 | 85 | bd | ad | a7 | ca | 4a | 02 | c640 | : | b1 | fb | 85 | 63 | a9 | 00 | 85 | bd | e9 |
| c470 | : | b0 | 12 | ad | a7 | ca | 09 | 01 | 8d | 9e | c648 | : | 20 | af | c4 | a0 | 05 | b1 | fb | 0a | 67 |
| c478 | : | a7 | ca | 20 | af | c4 | a2 | 27 | a0 | c2 | c650 | : | 90 | 05 | a9 | 2a | 20 | f7 | c3 | 20 | 24 |
| c480 | : | 02 | 4c | 88 | c4 | a2 | 22 | a0 | 07 | 2f | c658 | : | 3c | c5 | ee | af | ca | 20 | 41 | c7 | 6b |
| c488 | : | ad | a6 | ca | 85 | 63 | f0 | 0c | 85 | e5 | c660 | : | 08 | 20 | 0f | c7 | 20 | 67 | c7 | f0 | 73 |
| c490 | : | bd | 20 | ee | c5 | a9 | 00 | 85 | 62 | 47 | c668 | : | 23 | 28 | f0 | cd | ad | af | ca | c9 | ac |
| c498 | : | 20 | af | c4 | a9 | 00 | 85 | bd | 4c | b2 | c670 | : | 08 | 90 | 0c | 20 | f7 | c4 | 20 | 2e | 4a |
| c4a0 | : | 3c | c5 | a0 | 03 | b1 | f7 | aa | 88 | de | c678 | : | c5 | 20 | 90 | c6 | 4c | 86 | c6 | c9 | f2 |
| c4a8 | : | b1 | f7 | 86 | 62 | 85 | 63 | 60 | 20 | 78 | c680 | : | 04 | 90 | 03 | 20 | ee | c4 | 20 | d2 | cc |
| c4b0 | : | 09 | c4 | a2 | 90 | 38 | 20 | 49 | bc | f9 | c688 | : | c4 | 4c | 1f | c6 | 28 | 4c | ff | c4 | 81 |
| c4b8 | : | 20 | df | bd | 20 | a8 | c5 | 4c | 00 | 25 | c690 | : | a9 | 00 | 8d | af | ca | 60 | a0 | 01 | c7 |
| c4c0 | : | c4 | 49 | ff | 69 | 06 | a8 | f0 | 09 | d1 | c698 | : | 91 | fb | 20 | e5 | c3 | a0 | 02 | 20 | 75 |
| c4c8 | : | a9 | 20 | 20 | f7 | c3 | 88 | 4c | c6 | c8 | c6a0 | : | 10 | c4 | b0 | 0d | 20 | 19 | c4 | b0 | 1f |
| c4d0 | : | c4 | 60 | ad | a7 | ca | 4a | 4a | 90 | 6e | c6a8 | : | 08 | aa | a9 | 20 | 91 | fb | 4c | b7 | 0d |
| c4d8 | : | 06 | 20 | 2e | c5 | 4c | f7 | c4 | ae | 28 | c6b0 | : | c6 | 91 | fb | 20 | e5 | c3 | aa | a9 | bc |
| c4e0 | : | a5 | ca | e0 | 28 | b0 | 08 | a9 | 20 | 5a | c6b8 | : | 40 | e0 | 24 | f0 | 0a | e0 | 25 | f0 | ae |
| c4e8 | : | 20 | f7 | c3 | 4c | df | c4 | ad | a7 | a8 | c6c0 | : | 05 | 4a | 4a | 4c | cd | c6 | 4a | 48 | d3 |
| c4f0 | : | ca | 09 | 02 | 8d | a7 | ca | 60 | ad | 1f | c6c8 | : | 20 | e5 | c3 | aa | 68 | e0 | 28 | d0 | f1 |
| c4f8 | : | a7 | ca | 29 | 0d | 8d | a7 | ca | a9 | 85 | c6d0 | : | 02 | 09 | 80 | a0 | 00 | 91 | fb | 20 | 48 |
| c500 | : | 0d | 20 | f7 | c3 | a9 | 00 | 8d | a5 | b0 | c6d8 | : | a2 | c4 | a0 | 04 | 91 | fb | 88 | 8a | b5 |
| c508 | : | ca | 20 | ee | c3 | f0 | 1f | c9 | 03 | 4c | c6e0 | : | 91 | fb | ad | ad | ca | a0 | 05 | 91 | 79 |
| c510 | : | d0 | 12 | 20 | 6c | c5 | a9 | 6d | a2 | 24 | | | | | | | | | | | |
| c518 | : | ca | 20 | 84 | c8 | ad | b5 | ca | 85 | eb | | | | | | | | | | | |
| c520 | : | 01 | 4c | ae | a7 | 20 | ee | c3 | f0 | 52 | | | | | | | | | | | |
| c528 | : | fb | c9 | 03 | f0 | e5 | 60 | ce | ae | e1 | | | | | | | | | | | |
| c530 | : | ca | d0 | 08 | a9 | 05 | 8d | ae | ca | a7 | | | | | | | | | | | |

Listing 2. »Cross-Ref 64«. (Fortsetzung)

```

c6e8 : fb 60 a0 00 b1 fb 4a 4a f4
c6f0 : 4a 4a 4a b0 0a a2 25 4a e7
c6f8 : b0 01 ca 8a 20 f7 c3 b1 61
c700 : fb 0a 90 0a a9 28 20 f7 b2
c708 : c3 a9 29 20 f7 c3 60 38 7e
c710 : a5 fb e5 b6 85 fb b0 02 02
c718 : c6 fc 60 a5 fc 85 fe 38 91
c720 : a5 fb e5 b6 85 fd b0 02 22
c728 : c6 fe 60 a0 00 18 b1 fb 19
c730 : d1 fd f0 04 b0 0a 90 08 6a
c738 : c8 c4 b6 f0 03 4c 2d c7 05
c740 : 60 20 1b c7 a9 03 85 b6 a6
c748 : 20 2b c7 08 a9 06 85 b6 3f
c750 : 28 60 a0 00 c4 b6 f0 0e b2
c758 : b1 fb 48 b1 fd 91 fb 68 7c
c760 : 91 fd c8 4c 54 c7 60 a5 fc
c768 : fc c5 58 d0 04 a5 fb c5 60
c770 : 57 60 18 a5 57 65 b6 85 39
c778 : 57 90 02 e6 58 60 18 ad b9
c780 : ab ca 65 b6 8d ab ca 90 43
c788 : 03 ee ac ca 60 38 ad ab 5d
c790 : ca e5 b6 8d ab ca b0 03 86
c798 : ce ac ca 60 ad ab ca 85 e9
c7a0 : 57 ad ac ca 85 58 ad a9 77
c7a8 : ca 85 fb c5 57 08 ad aa ae
c7b0 : ca 85 fc c5 58 d0 03 28 9d
c7b8 : f0 19 20 1b c7 20 2b c7 5a
c7c0 : 90 03 20 52 c7 20 0f c7 6d
c7c8 : 20 67 c7 d0 ed 20 72 c7 e1
c7d0 : 4c a6 c7 60 ad a4 ca 18 c9
c7d8 : 65 f7 85 7a 08 a5 f8 85 86
c7e0 : 7b 28 90 02 e6 7b 60 20 e0
c7e8 : d4 c7 20 09 c4 20 73 00 e4
c7f0 : 20 6b a9 a0 00 a5 15 91 e9
c7f8 : fb 85 62 c8 a5 14 91 fb a1
c800 : 85 63 20 0f c7 20 af c4 e7
c808 : 4c 45 c4 a0 02 0d a8 ca fd
c810 : 8d a3 ca 91 fb 60 ad ab 24
c818 : ca 85 57 ad ac ca 85 58 18
c820 : ad a9 ca 85 fb ad aa ca 72
c828 : 85 fc a9 00 8d a7 ca 8d f2
c830 : a5 ca a9 05 8d ae ca a9 12
c838 : f6 8d a6 ca 20 67 c7 f0 36
c840 : 42 a0 00 84 bd b1 fb 85 c7
c848 : 62 c8 b1 fb 85 63 20 af 4d
c850 : c4 20 3c c5 a0 02 b1 fb c5
c858 : 2c a6 ca d0 02 29 fe 8d 24
c860 : a3 ca a2 00 86 bd 20 df a8
c868 : c5 a0 03 b1 fb 85 62 c8 7b
c870 : b1 fb 85 63 20 3c c5 20 28
c878 : af c4 20 d2 c4 20 0f c7 05
c880 : 4c 3c c8 60 8d 90 c8 8e c6
c888 : 91 c8 a2 00 8e a5 ca ad c3
c890 : ff ff f0 12 c9 ff f0 0f 8c
c898 : 20 f7 c3 ee 90 c8 d0 03 1b
c8a0 : ee 91 c8 4c 8f c8 60 20 13
c8a8 : 3c c5 4c 9b c8 a9 45 a2 82
c8b0 : c9 20 84 c8 a0 02 a9 ec 5e
c8b8 : a2 c9 20 84 c8 88 d0 f6 d9
c8c0 : 4c d6 c8 a9 45 a2 c9 20 af
c8c8 : 84 c8 a0 02 a9 6f a2 c9 4d
c8d0 : 20 84 c8 88 d0 f6 a0 02 c1
c8d8 : a9 91 a2 c9 20 84 c8 88 86
c8e0 : d0 f6 60 a9 15 a2 ca 20 4a
c8e8 : 84 c8 a0 02 a9 39 a2 ca bd
c8f0 : 20 84 c8 88 d0 f6 a0 08 ed
c8f8 : a9 4f a2 ca 20 84 c8 88 a5
c900 : d0 f6 60 20 fa c2 a5 7a b9
c908 : 8d b6 ca a5 7b 8d b7 ca f0
c910 : a5 01 8d b5 ca 20 00 c4 87
c918 : ad b0 ca 4a 4a 4a 4a 90 5a
c920 : 0e 20 34 c0 ad b0 ca 4a 84
c928 : 4a 4a 90 03 20 3d c0 ad 66
c930 : b0 ca 4a 4a 90 03 20 03 c9
c938 : c0 ad b0 ca 4a 90 03 20 ca
c940 : 0f c0 4c 6c c5 0d ff ff 15
c948 : ff 20 20 20 20 53 50 52 e6
c950 : 55 4e 47 42 45 46 45 48 13
c958 : 4c 45 0d ff ff ff ff 20 ca
c960 : 3d 3d 3d 3d 3d 3d 3d 3d 60
c968 : 3d 3d 3d 3d 3d 0d 00 20 b7
c970 : 49 4e 20 5a 45 49 4c 45 8e
c978 : 20 20 20 20 53 50 52 55 60
c980 : 4e 47 20 44 55 52 43 48 88
c988 : ff 20 20 4e 41 43 48 ff b9
c990 : 00 2a 3d 3d 3d 3d 3d 3d ca
c998 : 3d 3d 2a 3d 3d 3d 3d 3d d3
c9a0 : 3d 3d 3d 3d 3d 3d 3d 3d a0
c9a8 : 3d 3d 3d 3d 3d 3d 2a 3d 5c
c9b0 : 3d 3d 3d 3d 3d 3d 3d 3d b0
c9b8 : 2a 00 ff ff ff 41 4e 47 b4
c9c0 : 45 53 50 52 55 4e 47 45 7d
c9c8 : 4e 45 20 5a 45 49 4c 45 67
c9d0 : 4e 0d ff ff ff 20 3d 3d 15
c9d8 : 3d 3d 3d 3d 3d 3d 3d 3d d8
c9e0 : 3d 3d 3d 3d 3d 3d 3d 3d e0
c9e8 : 3d 3d 0d 00 2a 20 20 5a e0
c9f0 : 49 45 4c 20 20 2a 20 20 07
c9f8 : 20 53 50 52 55 4e 47 20 45
ca00 : 44 55 52 43 48 20 20 20 32
ca08 : 20 2a 20 20 20 49 4e 20 0f
ca10 : 20 20 20 2a 00 0d ff ff f6
ca18 : ff 20 20 20 20 20 20 56 64
ca20 : 41 52 49 41 42 4c 45 4e 3d
ca28 : 0d ff ff ff ff 20 3d 3d a5
ca30 : 3d 3d 3d 3d 3d 3d 3d 0d d0
ca38 : 00 20 56 41 52 49 41 42 ff
ca40 : 4c 45 20 20 20 5a 45 49 b7
ca48 : 4c 45 4e ff ff ff 00 2a 1f
ca50 : 3d 3d 3d 3d 3d 3d 3d 3d 50
ca58 : 2a 00 0d 53 50 45 49 43 0b

```

Listing 2. »Cross-Ref 64«. (Fortsetzung)

Tips & Tricks

| | |
|-----------------------------------|-----------------------------------|
| ca60 : 48 45 52 20 56 4f 4c 4c 8d | ca98 : 47 4f 54 4f 20 20 20 20 4a |
| ca68 : 20 21 21 0d 00 0d 41 42 f5 | caa0 : 20 20 2a 00 00 00 00 00 5b |
| ca70 : 42 52 55 43 48 20 21 21 e5 | caa8 : 00 f9 bf 00 00 00 05 00 a9 |
| ca78 : 0d 00 54 48 45 4e 20 46 77 | cab0 : 00 00 00 00 00 00 00 00 b1 |
| ca80 : 4f 52 20 20 4e 45 58 54 1e | |
| ca88 : 20 52 55 4e 20 20 4f 4e cd | |
| ca90 : 20 2d 20 47 53 55 42 20 61 | |

Listing 2. »Cross-Ref 64«. (Schluß)

```

* = $C000
AKOPADR = $F7
NKOPADR = $F9
CHRGET = $0073
CHRGOT = $0079
GETZEIL = $A96B
TABENDL = $57
TABENDH = $58
ZEI1L = $FB
ZEI1H = $FC
ZEI2L = $FD
ZEI2H = $FE
LAENGE = $B6
TEMP = $BD
;*****
JMP SCONTROL
VAR JSR VSTART ; SUCHT VARIABLEN
JSR DRVAR ; AUSGABE V.KOPF
JSR VAROUT ; " DER VAR.
JMP CARET2 ; ZEILENVORSCHUB
VAR2 LDA SYSTAT ; VAR. IN REIHENF.
LSR A ;
LSR A ;
BCS VAR3 ;
JSR VSTART ; NEIN: SUCHEN
VAR3 JSR INTEND ; TAB. UM 1 KUERZ.
JSR INIT ; INITIALSIEREN
JSR TESTTAB ; TABELLE LEER?
BCS LVAR2 ;
RTS ; JA: RTS
LVAR2 JSR SORT ; TAB. SORTIEREN
JSR DECTEND
JSR DRVAR
JSR VAROUT
JMP CARET2
SPR JSR SPRUNG ; SPRUENGE SUCHEN
JSR CARET2 ; UND AUSGEBEN
JMP INTEND
AN JSR INIT ; ANGESPRUNGENE
JSR TESTTAB ; ZEILEN AUSGEBEN
BCS LAN1 ;
RTS
LAN1 JSR SORT
JSR DECTEND
JSR DRAN
JSR ZIELE
JMP CARET2
;
VSTART LDA #6 ; = ZNR+CODE+ZNR
STA LAENGE
JSR SETTAB
JSR SETBER
JMP VZEILE2
VZEILE JSR NEXTLINE
VZEILE2 LDA #10000000
STA DBSTAT
LDA TEMP ; KODE FUER ENDE?
CMP #FF ; NEIN: WEITER
BNE VARLOOP ; JA: ENDE
RTS
VARLOOP JSR NXTCHR
CMP #20
BEQ VARLOOP
CMP #0 ; ZEILENENDE ?
BEQ VZEILE
CMP #22 ; =" ODER = REM?
BEQ VARSUCH ; DANN BIS " 0.
CMP #8F ; NEUE ZEILE
BEQ VZEILE ; WEITER
CMP #A7 ; THEN
BEQ SETDB
CMP #88 ; LET
BEQ SETDB
CMP #' ; STATEMENT
BEQ SETDB
CMP #81 ; FOR
BEQ SETDB
JSR CHKALPH
BCC LTEST1
JSR VARGET
JSR INZEIL
JSR OVFLOW
LTEST1 LDX #20000000
STX DBSTAT
JMP VARLOOP
SETDB LDX #10000000
STX DBSTAT
JMP VARLOOP
VARSUCH JSR NXTCHR
CMP #22
BEQ LWS1
CMP #0
BEQ LWS2
JMP VARSUCH
LWS1 JMP VARLOOP
LWS2 JSR NEXTLINE
JMP VARLOOP
;
SPRUNG LDA #5 ; = ZNR+CODE+ZNR
STA LAENGE
STA LINES ; LF NACH 5 ZEILEN
JSR SETTAB
JSR SETBER
LDA TEMP
CMP #FF
BEQ SZEILE2
JSR DRSPR
LDA #0
STA ZLANG
JMP SZEILE1
SZEILE JSR NEXTLINE
SZEILE1 LDA TEMP ; KODE FUER ENDE?
CMP #FF
BNE SZEILE2
RTS
SZEILE2 LDA #20000001 ; STATUS FUER
STA THSTAT ; 5 LEERZEICHEN
SPRLOOP JSR NXTCHR
CMP #0 ; ZEILENENDE ?
BEQ SZEILE
CMP #22 ; =" ODER = REM?
BEQ SPRSUCH ; DANN BIS " 0.
CMP #8F ; NEUE ZEILE
BEQ SZEILE ; WEITER
CMP #89 ; GOTD ?
BEQ CGOTO
CMP #8D ; GOSUB ?
BEQ CGOSUB
CMP #A7 ; THEN ?
BEQ CTHEN
CMP #8A ; RUN ?
BEQ CRUN
CMP #91 ; ON ?
BEQ CON
CMP #81 ; FOR ?
BEQ CFOR
CMP #82 ; NEXT ?
BEQ CNEXT
JMP SPRLOOP
CGOTO JMP GOTD
CGOSUB JMP GOSUB
CTHEN JMP THEN
CRUN JMP RUN
CON JMP ON
CFOR JMP FOR
CNEXT JMP NEXT
SPRSUCH JSR NXTCHR
CMP #22
BEQ LSS1
CMP #0
BEQ LSS2
JMP SPRSUCH
LSS1 JMP SPRLOOP
LSS2 JSR NEXTLINE
JMP SPRLOOP
;
GOTO JSR ZEILOUT
LDA #20000010
JMP AUSGABE
;
GOSUB JSR ZEILOUT
LDA #20000100
JMP AUSGABE
;
RUN JSR CHKSPC ; NAE. ZCHN NUM.?
JSR CHKNUM ; JA: AUSGABE
BCC LRUN1 ; NEIN: WEITER
JSR ZEILOUT
DEC POINTER ; DA CHKSPC AUF N.
LDY POINTER ; ZCHN STELLT
LDA #200010000
JMP AUSGABE
LRUN1 JMP SPRLOOP
;
FOR JSR ZEILOUT
INC FANZ ; OFFENE FOR+1
LDA #201000000
STA TSTAT
JSR TEXTAUS
LFOR1 JSR CHKSPC ; TEXT BIS '=' AUS-
CMP #82 ; GEBEN ('='#$2)
BEQ LFOR2 ;
JSR CHR0UT ; GEBEN
JMP LFOR1
LFOR2 JSR LINEFEED
JMP SPRLOOP
;
NEXT LDA POINTER
PHA
NEXT1 DEC FANZ
NEXT2 JSR NXTCHR
JSR TRZCHN
BEQ NEXT3
CMP #' ;
BEQ NEXT1
JMP NEXT2
NEXT3 PLA
STA POINTER
JSR ZEILOUT
LDA #200100000
STA TSTAT
JSR TEXTAUS
LNEXT1 JSR CHKSPC
JSR TRZCHN ; SOLANGE ZCHN
BEQ LNEXT3 ; AUSG. BIS Z.ENDE
JSR CHR0UT
JMP LNEXT1
LNEXT3 JSR LINEFEED
JMP SPRLOOP

```

Listing 3. Der Source-Code von »Cross-Ref 64«.

```

ON      ;
LON1   JSR ZEILOUT
        JSR CHKSPC
        CMP #89 ; GOTO ?
        BEQ ONTO
        CMP #8D ; GOSUB?
        BEQ ONSUB
        JMP LON1
ONTO   LDA #Z00001010
        JMP LON2
ONSUB  LDA #Z00001100
LON2   JSR SORCOD
        LDA TSTAT
        AND #Z11111110
        STA TEMP
        STA TSTAT
        LDX #0
        JSR TEXTAUS
        JSR TABS
        JSR CHKSPC
        DEC POINTER
        JSR SETGET
LON3   LDA TEMP
        JSR TSPRUNG ; SPR.ART, ZNR U.
        JSR GETSPR2 ; ZIEL IN TABELLE
        JSR CHRROT ; ENDE STATEMENT?
        BEQ LON4 ; JA: LINEFEED
        CMP #, ; WEITERE ZEILENNR.
        BNE LON3 ; NEIN: LINEFEED
        JSR CHROUT
        LDA ZLANG
        CMP #75
        BCC LON3
        JSR CARET ; LINEFEED
        JSR TABS
        JSR TABS
        JSR TABS
        JMP LON3
LON4   JSR FUNFLIN
        JSR CARET
        JMP SPRLOOP
        ;
TSPRUNG LDY #2
        STA (ZEI1L),Y
        JSR ZNR ; ZNR AUS #14/#15
        LDY #3 ; IN TABELLE
        LDA #62
        STA (ZEI1L),Y
        INY
        LDA #63
        STA (ZEI1L),Y
        RTS
        ;
THEN   JSR CHKSPC
        JSR CHKNUM ; ZIFFER?
        BCC THENDE ;
        JSR ZEILOUT ; THEN MIT ZEILE
        LDA #Z10000000
        JSR SORCOD
        DEC POINTER ; EIN ZCHN ZURUECK,
        LDY POINTER ; DA CHKNUM ZU WEIT
        JMP AUSGABE
THENDE DEC POINTER ; TH-BIT SETZEN
        LDA #Z10000000
        STA THSTAT
        JMP SPRLOOP
        ;
AUSGABE STA TSTAT
        JSR SORCOD
        JSR TEXTAUS
        JSR TABS
        JSR GETSPR ; ZIEL IN TABELLE
        JMP LINEFEED ; UND AUSGEBEN
        JMP SPRLOOP
        ;
;*****
;
; UNTERROUTINEN
;*****
INIT   JSR ROMOFF
        LDA #0
        STA ZSTAT
        STA ZLANG
        STA FANZ
        STA TSTAT
        STA TEMP
        STA DBSTAT
        STA TABENDL
        LDA #A0
        STA TABENDH
        LDA #F9
        STA TABSTRTL
        STA ZE11L ; ANFANG DER TAB-
        LDA #BF ; ELLE AB $BFF9
        STA TABSTRH
        STA ZE11H
        JMP BASTRT
        ;
NEXTLINE LDY #0
        LDA (NKOPADR),Y
        BNE LNL1
        INY
        LDA (NKOPADR),Y
        BEQ NLENDE
LNL1   LDA NKOPADR
        STA AKOPADR
        LDA NKOPADR +1
        STA AKOPADR +1
        JSR CKBEN ; ENDE BEREICH?
        BEQ LNL2 ; NEIN
        BCC LNL2 ;
        ;
NLENDE LDA-ZEI1L ; SETZT TABELLEN-
        STA TENDL ; ENDE AUF ZE11
        LDA ZE11H
        STA TENDH
        LDA #FF ; KODE FUER
        STA TEMP ; ENDE TESTLOOP
        RTS
        ;
LNL2   LDA ZSTAT
        AND #Z1110 ; ZNR FLAG LOE.
        STA ZSTAT
        LDY #3 ; ZEIGT AUF 0.
        STY POINTER ; BASIC-ZECHEN
        LDY #0
        LDA (AKOPADR),Y ; ZEIGER AUF
        STA NKOPADR ; NAECHSTE B.
        INY ; ZEILE HOLEN
        LDA (AKOPADR),Y
        STA NKOPADR +1
        RTS
        ;
;-----
; HOLT SYSTEM-PARAMETER (SYSSTAT)
; SETZT DRUCKER
; HOLT BEREICH (IN BERST/EN)
;-----
SYSGET LDA #0
        STA SYSTAT
        STA BERSTL
        STA BERSTH
        LDA #FF
        STA BERENL
        STA BERENH
        JSR CHRROT
        CMP #,
        BEQ LSYS0
        LDX #0B
        JMP #A437 ; FEHLER: SYNTAX
LSYS0  JSR CHRGET ; DRUCKER
        CMP #'P
        BNE LSYS1
        JSR SETPRT
LSYS1  JSR CHRGET
        CMP #'S
        BNE LSYS2
        LDA #Z01000
        JSR SETSYS
LSYS2  JSR CHRGET ; ANGESSPR. ZEILEN?
        CMP #'J
        BNE LSYS3
        LDA #Z00100
        JSR SETSYS
LSYS3  JSR CHRGET ; VARIABLEN ?
        CMP #'V
        BNE LSYS4
        LDA #Z00010
        JSR SETSYS
LSYS4  JSR CHRGET ; VARIABLEN SORT.?
        CMP #'J
        BNE LSYS5
        LDA #Z00001
        JSR SETSYS
LSYS5  JSR CHRGET ; KOMMENTAR ?
        CMP #'K
        BNE LSYS6
        LDA #Z10000
        JSR SETSYS
        JSR CHRGET
LSYS6  CMP #, ; FOLGEN ZNRS?
        BNE LSYS8 ; NEIN: RTS
        CMP #AB ; "-"
        BEQ LSYS7
        JSR CHRGET
        JSR GETZEIL
        LDA #14
        STA BERSTL
        LDA #15
        STA BERSTH
        JSR CHRROT
        CMP #AB ; "-"
        BNE LSYS8 ; NEIN: ENDE
LSYS7  JSR CHRGET
        JSR GETZEIL
        LDA #14
        STA BERENL
        LDA #15
        STA BERENH
        RTS
LSYS8  RTS
SETSYS ; ORA SYSTAT ; <AC> MIT SYSTAT
        STA SYSTAT ; VERKNUEPFEN
        RTS
;*****
CKBST  JSR ZNR ; TEST, OB AKT. ZNR
        CPX BERSTH ; GROESSER ODER
        BNE LBST1 ; GLEICH BER. ANF
        CMP BERSTL
        RTS
LBST1  RTS
        ;
CKBEN  JSR ZNR ; TEST, OB AKT. ZNR
        CPX BERENH ; GROESSER ODER
        BNE LBEN1 ; GLEICH BER. ENDE
        CMP BERENL
        RTS
LBEN1  RTS
        ;
SETTAB ; LDA #A0
        STA TABENDH
        LDA #00
        STA TABENDL
        RTS
        ;
SETBER JSR INIT
LBER1  JSR NEXTLINE
        LDA TEMP
        CMP #FF
        BNE LBER2
        JSR SYSOFF
        JMP #A7AE
LBER2  JSR CKBST
        BCC LBER1
        RTS
        ;
BASTRT LDA #2B ; ADRESSE DER
        STA NKOPADR ; ERSTEN BASIC
        LDA #2C ; ZEILE HOLEN
        STA NKOPADR +1
        RTS
        ;
PREVCHR DEC POINTER ; HOLT VORIGES
        JSR AKTCHR ; ZCHN, OHNE
        INC POINTER ; ZEIGER ZU
        RTS ; VERSTELLEN
        ;
NXTCHR INC POINTER ; HOLT NAECHSTES
        ; ; ZEICHEN
AKTCHR LDY POINTER
        LDA (AKOPADR),Y
        RTS
        ;
GET     JSR ROMON
        JSR #F13E
        JMP ROMOFF
        ;
CHRROT INC ZLANG
        JSR ROMON
        JSR #FFD2
        ;
ROMOFF ; PHA
        LDA 1
        AND #Z11111110
        STA 1
        PLA
        RTS
        ;
ROMON  ; PHA
        LDA #37
        STA 1
        PLA
        RTS
        ;
CHKALPH JSR ROMON ; TEST, OB AKKU ALPHA
        JSR #B113 ; C=1, WENN BUCHST.
        JMP ROMOFF ; C=0, WENN ANDERS
        ;
CHKNUM CMP #3A ; GROESSER ALS '9'?
        BPL LCN1
        CMP #30 ; KLEINER ALS '0'?
        BMI LCN1 ; JA: C=0
        SEC ; NEIN: C=1
        RTS
LCN1   CLC

```

Listing 3. Assembler-Source-Code zu »Cross-Ref 64« (Fortsetzung)

```

RTS
;
CHKSPC JSR NXTCHR
CMP #20
BEQ CHKSPC
RTS
;
TRZCHN CMP #0 ; TESTET AUF NULL
BEQ LTZ1 ; UND ' '
CMP #' ; JA: Z=1
BEQ LTZ1 ; NEIN: Z=0
LTZ1 RTS
;
TESTTAB LDA TABSTRTH
CMP TENDH
BNE LTETA
LDA TABSTRTL
CMP TENDL
LTETA RTS
;
OVFLOW JSR TESTEND
BCS LOVF
JSR SYSOFF
LDA #<TOVER
LDX #>TOVER
JSR MASKE
JMP #A7AE
LOVF RTS
;
ZEILOUT JSR ZNR ; ZNR NACH #62/#63
LDY #4 ; ZNR IN TAB
STA (ZEI1L),Y
DEY
TXA
STA (ZEI1L),Y
LDA #0
STA TEMP
LDA ZSTAT ; BEI 1 KEINE ZNR
LSR A
BCS LZ01
LDA ZSTAT ; ZNR-FLAG =1
ORA #Z0001
STA ZSTAT
JSR ZAUSGABE
LDX #39
LDY #2
JMP LZ02
LZ01 LDX #34
LDY #7
LZ02 LDA FANZ ; FOR OFFEN?
STA #63
BEQ LZ03 ; NEIN: RTS
STA TEMP
JSR LTA2
LDA #0
STA #62
JSR ZAUSGABE
LZ03 LDA #0
STA TEMP
JMP TABS
;
ZNR LDY #3 ; HOLT NR. DER
LDA (AKOPADR),Y ; AKT. ZEILE
TAX ; NACH #62/#63
DEY ; UND A,X
LDA (AKOPADR),Y
STX #62
STA #63
RTS
;
ZAUSGABE JSR ROMON
LDX #90 ;#62/#63 WIRD ALS
SEC ;INTEGERZAHL IM
JSR #BC49 ;16-BIT ADDRESS-
JSR #BDDF ;FORMAT AUSGEBEN
JSR STROUT ; #AB1E
JMP ROMOFF
;
RECBUN EOR #FF
ADC #6
TAY
LREC1 BEQ LREC2
LDA #20
JSR CHROUT
DEY
JMP LREC1
LREC2 RTS
;
LINEFEED LDA ZSTAT
LSR A ; BEI BIT 1=0
LSR A ; KEIN LF
BCC LLF1
JSR FUNFLIN
JMP CARET
LLF1 LDX ZLANG ; KEIN LF, DANN

```

```

CPX #40 ; BIS POS. 40
BCS LLF2 ; AUFFUELLEN
LDA #20
JSR CHROUT
JMP LLF1
LLF2 LDA ZSTAT
ORA #Z0010 ; LF-FLAG SETZEN
STA ZSTAT
RTS
;
CARET LDA ZSTAT ; LF-FLAG LOESCHEN
AND #Z1101
STA ZSTAT
CARET2 LDA #00
JSR CHROUT
LDA #0 ; ZEILENLAENGE:=0
STA ZLANG
CARET3 JSR GET ; TASTE BEDRUECKT?
BEQ LCAR4 ; NEIN: RTS
LCAR1 CMP #3 ; RUN/STOP?
BNE LCAR3
LCAR2 JSR SYSOFF ; JA
LDA #<TBREAK
LDX #>TBREAK
JSR MASKE
JMP #A7AE
LCAR3 JSR GET ; 2. TASTE
BEQ LCAR3 ; FUER CONT.
CMP #3
BEQ LCAR2
LCAR4 RTS
;
FUNFLIN DEC LINES
BNE LFUNF1
LDA #5
STA LINES
JSR CARET
LFUNF1 RTS
;
TABS LDA ZLANG ; ZEILENLAENGE HOL.
SEC
LTAB1 SBC #10 ; SOLANGE POSITIV
BCS LTAB1 ; 10 ADDIEREN
EOR #FF ; ERGEBNIS INVERT.
ADC #1
TAX
INX
LTAB2 DEY
BEQ LTAB3
LDA #20
JSR CHROUT
JMP LTAB2
LTAB3 RTS
;
SETPRT LDA #4
LDX #4
LDY #68 ; SEK 0 + #60
JSR #FFBA ; FILPAR
JSR #FFB1 ; LISTEN
LDA #68 ; SEKADR
JSR #FF93 ; SEKLST
LDA #4
STA #9A
RTS
;
SYSOFF LDA PROPORT
CMP #35
BNE GETRES
LDA #37 ; ROM AN,
STA 1 ; #A000 KOPIERT WIRD
LDA #A0
STA 4
LDY #0
LROMK LDA (3),Y
STA (3),Y
INY
BNE LROMK
INC 4
LDA 4
CMP #C0
BNE LROMK
;
GETRES LDA GETSP
STA #7A
LDA GETSP +1
STA #7B
;
PRTOFF LDA #4
JSR #FFAE ; UNLIST
JSR #FFC3 ; SCHLIESSEN
LDA #3
STA #9A
LDA PROPORT
STA 1
RTS

```

```

;
STROUT JSR ROMON
JSR #B4B7 ; STRINGPARAM. HOL.
JSR #B6A6 ; FRESTRING
PHA
LDX TEMP
BNE LSTR0
JSR RECBUN
LSTR0 PLA
TAX ; STR.LAENGE
LDY #0
INX
LSTR1 DEX
BEQ LSTR2
LDA (#22),Y
JSR CHROUT ; DRUCKEN
INY
CMP #00
BNE LSTR1
JSR #AAE5
JMP LSTR1
LSTR2 JMP ROMOFF
;
TEXTAUS LDX #0
LDA THSTAT
ORA TSTAT
STA TSTAT
LTA1 ASL TSTAT
BCS FUENFAUS
INX
INX ; SETZE X AUF DEN
INX ; NAECHSTEN TEXTAB-
INX ; SCHNITT
INX
JMP LTA1
FUENFAUS LDY #5
LTA2 LDA TEXT,X
JSR CHROUT
INX
DEY
BNE LTA2
LDA TSTAT
BNE LTA1
RTS
;
;=====
; AUSGABE ALLER VARIABLEN IN DER TAB
;=====
VAROUT LDA TABSTRTL ; ZEIGER AUF
STA ZEI1L ; ANFANG DER
LDA TABSTRTH ; TABELLE SETZEN
STA ZEI1H
LDA TENDL
STA TABENDL
LDA TENDH
STA TABENDH
LDA #5 ; STARTWERT 3
STA LINES ; FUER ABSAETZ
JSR CARET2 ; ZLANG:=0 U.
LVAR01 JSR TESTEND ; TABELLE ENDE?
BEQ LVAR09 ; JA: RTS
LVAR02 JSR SETVANZ
LDY #1 ; 1. ZEICHEN
LDA (ZEI1L),Y ; AUSGEBEN
JSR CHROUT
INY ; 2. ZEICHEN
LDA (ZEI1L),Y ; AUSGEBEN,
CMP #20 ; WENN <>LEER
BEQ LVAR03
JSR CHROUT
LVAR03 JSR STATOUT
JSR TABS
LVAR04 LDY #3 ; ZNR HOLEN
LDA (ZEI1L),Y ; UND NACH
STA #62 ; #62/#63
INY
LDA (ZEI1L),Y
STA #63
LDA #0
STA TEMP
JSR ZAUSGABE
LDY #5 ; "*" HINTER ZNR
LDA (ZEI1L),Y ; WENN DEF
ASL A ; SONST WEITER
BCC LVAR05
LDA #'*
JSR CHROUT
LVAR05 JSR TABS
INC VANZ
JSR VARVERG ; ALTVAR=NEUVAR?
PHP ; JA: NUR ZNR

```

Listing 3. Der Source-Code zu »Cross-Ref 64« (Fortsetzung)

```

JSR INZEI1 ; NEIN: UND NAM
JSR TESTEND ; ENDE DER TAB?
BEQ LVAR08 ; JA: ENDE
PLP ; AUSGEBEN
BEQ LVAR04 ; (NUR ZNR)
LDA VANZ ; MEHR ALS 7
CMP #8 ; VAR AUSGEG.?
BCC LVAR06 ; JA: LF,
JSR CARET ; VANZ:=0
JSR FUNFLIN
JSR SETVANZ
JMP LVAR07
LVAR06 CMP #4 ; 4 VAR?: LF
BCC LVAR07 ; WENIGER: WEITER
JSR LLF2
LVAR07 JSR LINEFEED
JMP LVAR02
LVAR08 PLP
LVAR09 JMP CARET2
;
LDA #0
STA VANZ
RTS
;
; =====
; HOLEN EINER VARIABLEN AUS BASIC TEXT
; =====
VARGET LDY #1 ; ZEICHEN IN TAB.
STA (ZEI1L),Y
JSR NXTCHR ; NAECHSTES ZCHN
LDY #2
JSR CHKALPH ; BUCHSTABE?
BCS LARGET1 ; JA: IN TAB.
JSR CHKNUM ; ZIFFER?
BCS LARGET1 ; JA: IN TAB.
TAX ; NEIN:AC SICHERN
LDA #*20 ; LEER IN TAB.
STA (ZEI1L),Y
JMP LARGET2
LARGET1 STA (ZEI1L),Y
JSR NXTCHR
TAX
LARGET2 LDA #%01000000
CPX #'$ ; STRING?
BEQ LARGET4
CPX #'% ; INTEGER?
BEQ LARGET3
LSR A ; REAL
LSR A
JMP LARGET5
LARGET3 LSR A ; INTEGER
LARGET4 PHA ; AC SICHERN
JSR NXTCHR
TAX
PLA
LARGET5 CPX #'( ; DIMENSIONIERT?
BNE LARGET6 ; NEIN
ORA #%10000000 ; JA!
LARGET6 LDY #0 ; STATUS AN 5.
STA (ZEI1L),Y ; STELLE IN TAB.
JSR ZNR ; ZNR IN TAB.
LDY #4 ; AN 4./5. STELLE
STA (ZEI1L),Y
DEY
TXA
STA (ZEI1L),Y
LDA DBSTAT ; DEF/BEN AN 5.
LDY #5 ; STELLE
STA (ZEI1L),Y
RTS
;
; =====
; PRUEFT STATUSBYTE DER VARIABLEN
; UND GIBT TYP - $, %, () - AUS
; =====
STATOUT LDY #0
LDA (ZEI1L),Y
LSR A
LSR A ; BITS 0-3
LSR A ; UEBERLESEN
LSR A
LSR A ; REAL?
BCS LSTAT2
LDX #'% ; INTEGER?
LSR A
BCS LSTAT1
DEX ; STRING!
LSTAT1 TXA
JSR CHROUT
LSTAT2 LDA (ZEI1L),Y ; FELD?
ASL A
BCC LSTAT3 ; NEIN: RTS
LDA #'(
JSR CHROUT
LDA #' )
JSR CHROUT
LSTAT3 RTS
;

```

```

; =====
; SORTIEREN VON FELDERN NACH
; "BUBBLESORT"
; UEBERGABE:
; "LAENGE" DES FELDES
; "TABSTR" ANFANG DES FELDES
; "TABEND" ENDE
; =====
; ERHOEHE ZEIGER1 UM LAENGE
; =====
INZEI1 SEC
LDA ZEI1L
SBC LAENGE
STA ZEI1L
BCS LINZ1
DEC ZEI1H
LINZ1 RTS
;
; =====
; ROUTINE BESTIMMT ZEIGER AUF I+1-TES
; ELEMENT; PARAMETER: LAENGE, ZE11
; =====
IZEIG2 LDA ZEI1H
STA ZEI2H
SEC
LDA ZEI1L
SBC LAENGE
STA ZEI2L
BCS LADD1
DEC ZEI2H
LADD1 RTS
;
; =====
; VERGLEICH DER DURCH ZE11, ZE12 UND
; LAENGE BESTIMMTEN FELDER
; C=1, WENN A(I)>A(I+1)
; Z=1, WENN A(I)=A(I+1) (Y=LAENGE)
; =====
VERGL LDY #0
LV1 CLC
LDA (ZEI1L),Y
CMP (ZEI2L),Y
BEQ LV2 ; GLEICH
BCS LV3 ; GROESSER
BCC LV3 ; KLEINER
LV2 INY ON; NEIN: NAECHSTES
CPY LAENGE ; X=LAENGE?
BEQ LV3 ; JA: ENDE
JMP LV1 ; ELEMENT TESTEN
LV3 RTS
;
; =====
; VERGLEICH ZWEIER VARIABLEN
; Z=1, WENN NAME UND STATUS GLEICH
; =====
VARVERG JSR IZEIG2
LDA #3 ; NUR 3 BYTE
STA LAENGE ; (STAT, NAME)
JSR VERGL ; VERGLEICHEN
PHP ; LAENGE WIEDER
LDA #6 ; 6
STA LAENGE
PLP
RTS
;
; =====
; TAUSCHE ZWEI FELDER, DIE DURCH ZE11
; ZE12 UND LAENGE BESTIMMT SIND
; =====
EXCHANGE LDY #0
LEXCH1 CPY LAENGE ; SOLANGE Y<LAE.
BEQ LEXCH2
LDA (ZEI1L),Y
PHA ; VERTAUSCHE
LDA (ZEI2L),Y ; ZWEI BYTES
STA (ZEI1L),Y
PLA
STA (ZEI2L),Y
INY
JMP LEXCH1
LEXCH2 RTS
;
; =====
; ENDE DER TABENDLLE ERREICHT?
; (ZE11 = TABEND? ==> Z=1)
; =====
TESTEND LDA ZEI1H
CMP TABENDH
BNE LTEND1
LDA ZEI1L
CMP TABENDL
LTEND1 RTS
;
; =====
; TABEND UM EIN ELEMENT ERNIED
; =====

```

```

DECTAB CLC
LDA TABENDL
ADC LAENGE
STA TABENDL
BCC LDECTAB
INC TABENDH
LDECTAB RTS
;
; =====
; TEND UM EIN ELEMENT VERRINGERN
; =====
INTEND CLC
LDA TENDL
ADC LAENGE
STA TENDL
BCC LINTEN1
INC TENDH
LINTEN1 RTS
;
; =====
; TEND UM EIN ELEMENT ERHOEHEN
; =====
DECTEND SEC
LDA TENDL
SBC LAENGE
STA TENDL
BCS LDETEN1
DEC TENDH
LDETEN1 RTS
;
; =====
; SORTIEREN
; =====
SORT LDA TENDL ; TABEND SETZEN,
STA TABENDL ; GEHT BEI SORT
LDA TENDH ; VERLOREN
STA TABENDH
LSORT1 LDA TABSTRTL ; ZEIGER1 AUF
STA ZE11L ; TABSTR
CMP TABENDL
PHP ; UND
LDA TABSTRTH ; TEST, OB
STA ZE11H ; ENDE DER
CMP TABENDH ; TABELLE
BNE LSORTY ; ERREICHT
PLP ; JA: ENDE
BEQ LSORTW
LSORTY JSR IZEIG2
JSR VERGL ; VERGLEICHEN
BCC LSORT3 ; GROESSER, DANN
JSR EXCHANGE ; TAUSCHEN
LSORT3 JSR INZEI1 ; NAECHSTES
JSR TESTEND ; ENDE ERREICHT?
BNE LSORTY ; NEIN: WEITER
JSR DECTAB ; JA: NAECHSTER
JMP LSORT1
LSORTW RTS
;
; =====
; SETZT CHRGET - ZEIGER AUF AKT. ZEICH.
; =====
SETGET LDA POINTER
CLC
ADC AKOPADR
STA $7A
PHP
LDA AKOPADR + 1
STA $7B
PLP
BCC LSET1
INC $7B
LSET1 RTS
;
; =====
; HOLT SPRUNGZIEL NACH TABELLENENDE
; =====
GETSPR JSR SETGET
GETSPR2 JSR ROMON
JSR CHRGET
JSR GETZEIL
LDY #0
LDA $15
STA (ZEI1L),Y
STA $62
INY
LDA $14
STA (ZEI1L),Y
STA $63
JSR INZEI1
JSR ZAUSGABE
JMP OVFLOW
;

```

Listing 3. Der Source-Code zu »Cross-Ref 64« (Fortsetzung)

```

;=====
;   SETZT BIT FUER SPRUNG NACH ON, THEN
;=====
SORCOD  LDY #2
ORA THSTAT
STA TSTAT
STA (ZEI1L),Y
RTS
;
;=====
;   AUSGABE DER ANGESPRUNGENEN ZEILEN
;   IN DER TABELLE: H,L DES SPRUNGZIELS,
;   CODE DER SPRUNGART,
;   L,H DER AUFRUF-ZEILE
;=====
ZIELE  LDA TENDL ; ZEIGER AUF ENDE
STA TABENDL ; TEND WURDE NICHT
LDA TENDH ; GEAEENDERT
STA TABENDH
LDA TABSTRTL ; ZEIGER AUF
STA ZE11L ; ANFANG
LDA TABSTRTH
STA ZE11H
LDA #0
STA ZSTAT
STA ZLANG
LDA #5
STA LINES
LDA #Z11110110
STA FANZ
LZIEL1 JSR TESTEND ; TABELLE LEER?
BEQ STPZIEL ; JA: BEENDEN
LDY #0 ; HIGH, LOW-BYTE
STY TEMP ; F. RECBUN
LDA (ZE11L),Y ; DES ZIELS
STA #62 ; HOLEN
INY
LDA (ZE11L),Y
STA #63
JSR ZAUSGABE
JSR TABS
LDY #2 ; SPRUNGART HOLEN
LDA (ZE11L),Y
BIT FANZ
BNE LZIEL2
AND #Z11111110
LZIEL2 STA TSTAT
LDX #0
STX TEMP ; F. RECBUN
JSR LTA1 ; TEXTAUSGABE
LDY #3 ; LOW,HIGH-BYTE
LDA (ZE11L),Y ; DER AUFRUFENDEN
STA #62 ; ZEILE
INY
LDA (ZE11L),Y
STA #63
JSR TABS
JSR ZAUSGABE
JSR LINEFEED
JSR INZE11
JMP LZIEL1
STPZIEL RTS
;
;=====
;   AUSGEBEN VON TEXT
;=====
MASKE  STA ADRE+1
STX ADRE+2
LDX #0
STX ZLANG
ADRE  LDA #FFFF
BEQ MASKE3
CMP #255
BEQ MASKE4
JSR CHROUT
MASKE1 INC ADRE+1
BNE MASKE2
INC ADRE+2
MASKE2 JMP ADRE
MASKE3 RTS

```

```

MASKE4 JSR TABS
JMP MASKE1
;
;=====
;   AUSGABE DER "MASKEN" FUER SPR U. VAR
;=====
DRAN  LDA #<TSPRK
LDX #>TSPRK
JSR MASKE
LDY #2
LDRAN1 LDA #<TANR
LDX #>TANR
JSR MASKE
DEY
BNE LDRAN1
JMP DRSPR1
DRSPR  LDA #<TSPRK
LDX #>TSPRK
JSR MASKE
LDY #2
LDSP1  LDA #<TSPRR
LDX #>TSPRR
JSR MASKE
DEY
BNE LDSP1
DRSPR1 LDY #2
LDSP2  LDA #<TSPRU
LDX #>TSPRU
JSR MASKE
DEY
BNE LDSP2
RTS
;
DRVAR  LDA #<TVARK
LDX #>TVARK
JSR MASKE
LDY #2
DRVAR1 LDA #<TVARR
LDX #>TVARR
JSR MASKE
DEY
BNE DRVAR1
LDY #8
DRVAR2 LDA #<TVARU
LDX #>TVARU
JSR MASKE
DEY
BNE DRVAR2
RTS
;
;=====
;   STEUERROUTINE
;   RUFT DIE ROUTINEN ENTSPRECHEND
;   DER PARAMETER AUF
;=====
SCONTROL JSR SYSGET ; PARAMETER HOLEN
LDA #7A ; GET-ZEIGER RETTEN
STA GETSP ;
LDA #7B ;
STA GETSP +1
LDA 1 ; PROZESSORPORT
STA PROPORT ; RETTEN
JSR ROMOFF ; RDM AUSSCHALTEN
LDA SYSTAT ; ABFRAGE DER
LSR A ; PARAMETER
LSR A
LSR A
LSR A
BCC LCON1
JSR SPR
LDA SYSTAT
LSR A
LSR A
LSR A
LSR A
BCC LCON1
JSR AN
LCON1  LDA SYSTAT

```

```

LSR A
LSR A
BCC LCON2
JSR VAR
LCON2  LDA SYSTAT
LSR A
BCC LCON3
JSR VAR2
LCON3  JMP SYSOFF ; STARTZUSTAND
; WIEDERHERSTELLEN
;
;=====
;   TEXTE
;=====
TSPRK  .BYTE 13,255,255,255
.TEXT " SPRUNGBEFEHLE"
.BYTE 13,255,255,255,255
.TEXT " "
TSPRR  .BYTE 13,0
.TEXT " IN ZEILE SPRUNG DURCH"
.BYTE 255
.TEXT " NACH"
.BYTE 255,0
TSPRU  .TEXT "*****"
.TEXT "*****"
.BYTE 0
TANK  .BYTE 255,255,255
.TEXT "ANGESPRUNGENE ZEILEN"
.BYTE 13,255,255,255
.TEXT "*****"
.BYTE 13,0
TANR  .TEXT "* ZIEL * SPRUNG DURCH *"
.TEXT " IN *"
.BYTE 0
TVARK  .BYTE 13,255,255,255
.TEXT " VARIABLEN"
.BYTE 13,255,255,255,255
.TEXT " "
.BYTE 13,0
TVARR  .TEXT " VARIABLE ZEILEN"
.BYTE 255,255,255,0
TVARU  .TEXT "*****"
.BYTE 0
TOVER  .BYTE 13
.TEXT "SPEICHER VOLL !!"
.BYTE 13,0
TBREAK .BYTE 13
.TEXT "ABBRUCH !!"
.BYTE 13,0
TEXT  .TEXT "THEN FOR NEXT "
.TEXT "RUN ON - GSUB "
.TEXT "GOTO *"
TSTAT .BYTE 0
POINTER .BYTE 0
ZLANG .BYTE 0
FANZ .BYTE 0
ZSTAT .BYTE 0
THSTAT .BYTE 0
TABSTRTL .BYTE #F9
TABSTRTH .BYTE #BF
TENDL .BYTE 0
TENDH .BYTE 0
DBSTAT .BYTE 0
LINES .BYTE 5
VANZ .BYTE 0
SYSTAT .BYTE 0
BERSTL .BYTE 0
BERSTH .BYTE 0
BERENL .BYTE 0
BERENH .BYTE 0
PROPORT .BYTE 0
GETSP .BYTE 0,0
.END

```

Listing 3. Der Source-Code zu »Cross-Ref 64« (Schluß)

```

10 rem
20 gosub 100 : goto 200
30 rem
40 run 50
50 rem
60 if a=0 then gosub 100 : goto 50
70 rem
80 on a goto 10,20,30,40
90 rem
100 if a<>0 then on a gosub 100,90,70,80

```

```

110 rem
120 b=0 : if b=1 then b=b+2
130 rem
140 print d;g;e;k
150 rem
160 e=g+d+k:d=k

```

Listing 1. Dieses, für sich unsinnige, Basic-Programm soll die Leistungsfähigkeit der Dokumentationshilfe »Cross-Reference-Liste« veranschaulichen