

128er

SOFTWARE
EXTRA

THE BEST OF **128er** VOLUME 1

Eine 5¼"-Diskette



Markt&Technik

Markt&Technik Verlag Aktiengesellschaft · Hans-Pinsel-Straße 2 · 8013 Haar bei München

Die Informationen in diesem Produkt werden ohne Rücksicht auf einen eventuellen Patentschutz veröffentlicht.

Warennamen werden ohne Gewährleistung der freien Verwendbarkeit benutzt.

Bei der Zusammenstellung von Texten und Abbildungen wurde mit größter Sorgfalt vorgegangen.

Trotzdem können Fehler nicht vollständig ausgeschlossen werden.

Verlag, Herausgeber und Autoren können für fehlerhafte Angaben und deren Folgen weder eine juristische Verantwortung noch irgendeine Haftung übernehmen.

Für Verbesserungsvorschläge und Hinweise auf Fehler sind Verlag und Herausgeber dankbar.

Alle Rechte vorbehalten, auch die der fotomechanischen Wiedergabe und der Speicherung in elektronischen Medien.

Die gewerbliche Nutzung der in diesem Produkt gezeigten Modelle und Arbeiten ist nicht zulässig.

15 14 13 12 11 10 9 8 7 6 5 4 3
91 90 89

© 1988 by Markt & Technik Verlag Aktiengesellschaft,
Hans-Pinsel-Straße 2, D-8013 Haar bei München/West-Germany
Alle Rechte vorbehalten
Einbandgestaltung: Grafikdesign Heinz Rauner
Druck: Bosch-Druck, Landshut
Printed in Germany

Inhaltsverzeichnis

Vorwort	5
<hr/>	
Mastertext 128	
Die Super-Textverarbeitung	7
<hr/>	
COLOR-PACK 1	
Hochauflösende Grafik im 80-Zeichen-Modus des C128	49
<hr/>	
TOP-FLOP	
Die Diskette unter der Lupe	59
<hr/>	
DOUBLE-ASS	
Assembler mit Doppelherz	77
<hr/>	
BUTLER	
Eine Hilfe für Basic- und Assemblerfans	93
<hr/>	
WINDOW-TECH	
»Echte« Windows im 80-Zeichen-Modus	113
<hr/>	
CENTRONICS-SCHNITTSTELLE	
Druckeranschluß ohne Hardware-Interface	117
<hr/>	
THIRTY-SECONDS	
Blitz-Backup für den C128	121
<hr/>	
DOUBLE-TOUCH	
Weltrekord: Disketten-Backup in 8 Sekunden	125

CP/M-FORMATTER

Mehr als 78 verschiedene Diskettenformate unter CP/M 3.0 verfügbar 129

UNIBOOT

Booten leichtgemacht 139

MICRO-HARDCOPY

Gestochen scharfe Hardcopies 153

VECTORS

Ein tolles Vergnügen 155

Vorwort

Der C128 besitzt, wohl als einziger Computer der Welt, gleich drei Betriebssysteme: Zum einen können im C64-Mode alle C64-Programme gefahren werden, im CP/M-Mode steht Ihnen eine große Zahl professioneller Anwendungen zur Verfügung. Für diese beiden Modi gibt es mehr als ausreichend Software.

Nicht so für den C128-Modus, der mit Abstand leistungsfähigste der drei Modi: Gegenüber dem C64 wurde die Taktfrequenz um 100% heraufgesetzt, der freie Arbeitsspeicher ist doppelt so groß und hinsichtlich der Grafikauflösung sind doppelt soviel Punkte ansprechbar.

Bedingt durch das geringe Software-Angebot für den C128-Modus muß man oft auf den C64-Modus ausweichen und kann dadurch die Leistungsfähigkeit seines Computers nicht voll ausreizen. Mit dem ersten 128er-Extra erweitern Sie Ihren Software-Bestand um 13 leistungsfähige Programme aus wichtigen Anwendungsgebieten:

Mit Mastertext 128 können Sie professionell Textverarbeitung betreiben. Das beliebte Programm für den C64 wurde voll an die Hardware des C128 angepaßt. Neben höherer Arbeitsgeschwindigkeit ist die gesamte Breite einer DIN-A4-Seite darstellbar, das lästige horizontale Scrollen gehört der Vergangenheit an.

Zwei Kopierprogramme stellen Weltrekorde auf: Als stolzer Besitzer von zwei Laufwerken können Sie ein komplettes Backup in 8! Sekunden ziehen. Dazu ist weder ein Parallel-Kabel noch eine Hardware-Erweiterung notwendig. Auch wenn Sie nur ein Laufwerk besitzen, ist eine Kopie in 30 Sekunden fertiggestellt, dreimal schneller als mit dem schnellsten C64-Kopierprogramm.

Die Basic-Erweiterung »Color-Pack 1« unterstützt die Grafik-Programmierung des 80-Zeichen-Bildschirms. Die Auflösung ist bei weitem höher als bei C64-Programmen: 480*240 Punkte. Auch die übrige Leistungsfähigkeit wird Sie positiv überraschen.

Für Assembler-Programmierer ist der 2-Pass-Assembler »Double-Ass« ein Muß. Er erlaubt sogar das Ansprechen des sonst im C128-Mode schlummernden Z80-Prozessors!

Wir wünschen Ihnen viel Freude beim Einsatz der Programme.

Ihre 64'er-Redaktion

MasterText 128

Sorgte MasterText für den C 64 schon für Aufsehen, so können wir Ihnen mit »MasterText 128« ein »absolutes Spitzenprogramm« vorstellen! Es ist eine Textverarbeitung, die keine Wünsche mehr offenläßt und es mit jedem professionellen Programm für den C128 aufnehmen kann.

MasterText 128 arbeitet im 80-Zeichen-Modus des C 128, verfügt über eine raffinierte Window-Technik, ist voll menügesteuert und bietet zudem einen Terminalmodus für Datenfernübertragung, der jederzeit aus dem Programm heraus erreichbar ist. Bei einem Textspeicher von 64000 Zeichen und einer enormen Vielfalt von Befehlen ist es ein Vergnügen, mit diesem Programm zu arbeiten. Ein integrierter Taschenrechner, eine Uhr mit Alarmfunktion sind nur einige der hervorragenden Eigenschaften, die dieses Programm zu bieten hat. Wer die C 64-Version von MasterText kennt, hat auch keine Probleme, sich bei diesem Programm einzuarbeiten. Auch eine Kompatibilität der Texte ist gegeben. Was mit MasterText 64 geschrieben wurde, kann problemlos mit MasterText 128 weiterverarbeitet werden. Es ist also das ideale Programm für Ein- und Aufsteiger.

Die Installation von MasterText 128

Bevor Sie MasterText 128 zum ersten Mal starten, muß die grundsätzliche Anpassung für Ihren Drucker vorgenommen werden. Laden und starten Sie das Programm »MASTERTEXT/INST« im 80-Zeichen-Modus des C 128:

```
RUN "MASTERTEXT/INST"
```

Es handelt sich dabei um das Installationsprogramm. Auf dem Bildschirm erscheint nun eine Auswahl verschiedener Druckertypen:

Die Zahlen müssen zweistellig, also »01« für Nummer 1, eingegeben werden. Nummer 1 ist die Installation für den MPS 801, VC 1525, GP 100-VC oder sonstige Kompatible. Den Punkt 2 wählt man als Besitzer eines MPS 802 oder VC 1526. Bei Betrieb eines Epson-kompatiblen Druckers am User-Port muß man sich zwischen den Punkten 3 und 4 entscheiden. Dies hängt

von der Position des DIP-Schalters »Auto Line Feed« am Drucker ab. Ist er ausgeschaltet, so wählen Sie Punkt 3, ansonsten Punkt 4. Kommen wir nun zu den Epson-kompatiblen Druckern mit angeschlossenem Hardware-Interface. Die Punkte 5 und 6 sind für Epson-kompatible Drucker mit Centronics-Schnittstelle gedacht, die per Interface (Wiesemann oder Data-Becker) am seriellen Bus betrieben werden. Wieder gibt es zwei Versionen, je nach Stellung des eben genannten DIP-Schalters. Die Nummern 7 und 8 bewirken die gleichen Installationen für das Görlitz-Interface. Die Punkte 9 und 10 sind speziell für den STAR NL-10, einmal für die ASCII-Betriebsart und einmal für die Commodore-Betriebsart. Der Drucker wird jedoch auf jeden Fall softwaremäßig in die wesentlich leistungsfähigere ASCII-Betriebsart umgeschaltet. Bleibt noch Punkt 11. Dieser ist für eine Brother-Schreibmaschine CE-60 mit RS232-Schnittstelle gedacht. Allerdings müssen Sie noch die richtige Übertragungsgeschwindigkeit im Menü »Druckerparameter« einstellen.

Alle Installationen lassen sich jederzeit verändern und speichern, um das Programm auch an Drucker-»Exoten« anpassen zu können. Wie man in diesem Fall verfährt, wird später detailliert beschrieben.

Ist die Installation abgeschlossen, so erscheint die Meldung »BOOTING MASTER-TEXT 128«, und das Programm wird in den Speicher des C 128 geladen. Danach können Sie die Systemdiskette aus dem Diskettenlaufwerk nehmen. MasterText steht komplett im Speicher und lädt keine Programmteile nach.

Wenn Sie das nächste Mal mit MasterText 128 arbeiten wollen, brauchen Sie die Installation natürlich nicht wieder vorzunehmen. Sie legen dann einfach die Diskette in das Diskettenlaufwerk und schalten den Computer an. MasterText 128 wird dann automatisch von der Diskette gebootet, sofern ein Boot-Sektor installiert wurde. Das in diesem Paket befindliche Programm »UNIBOOT« ermöglicht das Anbringen eines solchen.

Nach dem Booten von MasterText 128 befindet man sich im Editor (Bild 1) und kann sofort mit dem Schreiben beginnen. Man muß sich also nicht erst durch Hunderte von Menüs hindurchquälen, wie das bei anderen Textverarbeitungen oft nötig ist.

Mit der Taste <ESC> gelangt man ins Hauptmenü von MasterText 128. Dort kann man die einzelnen Funktionen von MasterText 128 anwählen. Mit den Cursor-Tasten <CRSR-rechts> und <CRSR-links> nimmt man zunächst eine Grobauswahl vor. Außer bei dem Menüpunkt

»EDITOR« erscheint ein Untermenü in Form eines Windows (Bild 2). In diesem wird die gewünschte Funktion genau spezifiziert. Dazu stehen die Cursor-Tasten <CRSR-aufwärts> und <CRSR-abwärts> zur Verfügung. Mit <RETURN> wird die entsprechende Funktion eingeleitet. Falls noch weitere Angaben notwendig sind, werden diese in einer weiteren Eingabemaske vorgenommen. Zwischen den einzelnen Feldern der verschiedenen Eingabemasken wählt man mit <CRSR-aufwärts> und <CRSR-abwärts>. <CRSR-rechts> und <CRSR-links> dienen zur Bewegung innerhalb eines Feldes. Mit <INST/DEL> können Sie ein ganzes Feld und einzelne Zeichen im Feld mit <CTRL INST/DEL> löschen. Mit <RETURN> wird die Eingabe abgeschlossen und der Befehl ausgeführt. Soll ein Befehl nicht mehr ausgeführt werden, so gelangt man mit <ESC> zurück ins Hauptmenü. Eventuelle Parametereintragungen werden dann nicht übernommen.

Unten auf dem Bildschirm sind drei Statuszeilen zu sehen. Hier finden Sie wichtige Angaben über momentan aktivierte Funktionen des Programms. In der ersten Statuszeile ist neben dem Hauptmenü noch eine Angabe über die Zeile und Spalte untergebracht, in der sich der Cursor augenblicklich befindet. Nach dem Start ist dies immer Zeile und Spalte 1. Das Feld dahinter, das jetzt noch frei ist, zeigt an, welcher Schreibmodus (Überschreiben oder Einfügen) aktiviert ist.

Der Editiermodus von MasterText 128

Die zweite Statuszeile ist noch frei. Hier werden alle Fehlermeldungen und Aufforderungen an den Benutzer zur Anzeige gebracht. Die Zahlen am Ende dieser Zeile geben die Uhrzeit an. Natürlich ist die Uhr anfangs nicht richtig eingestellt.

Die letzte Statuszeile gibt Auskunft über den zuletzt geladenen Text und über das aktivierte Diskettenlaufwerk. Dahinter folgen Angaben über die Geräte- und Sekundäradresse des angeschlossenen Druckers. Bei Betrieb der RS232-Schnittstelle finden sich hinter »R« Angaben über die eingestellten Parameter. Hinter »P:« steht die Position, auf die der Stackpointer des Taschenrechners gerade zeigt. In dem grünen Feld wird die von Ihnen zuletzt gegebene Antwort (j/n) zur Anzeige gebracht.

Damit Sie vor allem in der Einarbeitungsphase nicht ständig in der Anleitung blättern müssen, stellt MasterText 128 eine vorbildliche Funktion zur Verfügung. Drückt man im Editiermodus die HELP-Taste, so erscheint ein über zwei Bildschirmseiten verteilter Hilftext, der in Kurzform alle Befehle darstellt.

Der Text wird durchgehend eingetippt, es wird also über die Zeilen hinaus einfach weitergeschrieben. MasterText 128 nimmt während des Ausdrucks selbständig die Einteilung in das eingestellte Format vor. <RETURN> dient nur zum Beenden eines Absatzes. Drückt man diese Taste im Überschreibmodus, so wird der Rest der Zeile gelöscht und es erscheint ein Absatzendezeichen. Im Einfügemodus wird der Rest der Zeile in die folgende verschoben. <SHIFT RETURN> positioniert den Cursor an den Anfang der nächsten Zeile, ohne ein Absatzendezeichen zu erzeugen. Die Taste <LINE FEED> hat die gleiche Funktion.

Will man eine Zeile einrücken, so verwendet man hierfür <SHIFT SPACE>. Es erscheint ein »u«-ähnliches Symbol. Benutzt man nur <SPACE>, so rückt der Computer beim Drucken nicht ein, da er Leerzeichen am Anfang eines Absatzes überliest. Es muß also mindestens ein <SHIFT SPACE> zu Anfang eines Absatzes stehen, danach kann mit <SPACE> fortgefahren werden.

Trennungsvorschläge

Um das Trennen von langen Wörtern zu ermöglichen, kann man mit der Taste <←> Trennungsvorschläge vorgeben, die beim Drucken bei Bedarf ausgeführt werden. Die Taste ist einfach an der Stelle zu drücken, an der getrennt werden darf. Gerade beim Ausdruck im Blocksatz sollte man von dieser Möglichkeit regen Gebrauch machen, da so die Leerzeichen, die für den Randausgleich in die Zeilen eingefügt werden, möglichst gering gehalten werden.

Die Cursor-Tasten

Mit den Cursor-Tasten bewegt man sich zeichen- oder zeilenweise durch den Text. Bei Erreichen der oberen oder unteren Bildschirmzeile tritt das »Scrolling« in Funktion. Der Text wandert also in entsprechender Richtung vertikal über den Monitor. Es existieren jedoch eine Reihe von Tastenkombinationen, die das Bewegen durch den Text sehr komfortabel gestalten.

SHIFT-Kombinationen

Zusammen mit <SHIFT> erhalten die grauen Cursor-Tasten in der obersten Reihe der Tastatur eine neue Funktion. <SHIFT CRSR-aufwärts> bewegt den Cursor 10 Zeilen in Richtung Textanfang, <SHIFT CRSR-abwärts> 10 Zeilen in Richtung Textende. Dadurch wird ein schnelles Bewegen durch den gesamten Text ermöglicht. Bei 800 möglichen Zeilen ist dies auch erforderlich. <SHIFT CRSR-links> bewegt den Cursor an den Anfang des vorherigen Wortes, <SHIFT CRSR-rechts> entsprechend an den Anfang des nächsten Wortes.

CTRL-Kombinationen

Mit <CTRL CRSR-aufwärts> springt der Cursor an den Anfang des Textes, also in Zeile 1, Spalte 1. Die gleiche Funktion erreichen Sie auch mit <SHIFT CLR/HOME>. Mit <CTRL CRSR-abwärts> gelangt man an das Textende. Der Cursor steht exakt auf der nächsten Schreibposition, es sei denn, der Speicher ist voll. Mit <CTRL CRSR-rechts> und <CTRL CRSR-links> positioniert man den Cursor an das Ende beziehungsweise an den Anfang einer Bildschirmzeile.

CBM-Kombinationen

Auch mit der CBM-Taste lassen sich Funktionen erreichen. <CBM CRSR-aufwärts> setzt den Cursor an den Anfang des vorherigen Absatzes. <CBM CRSR-abwärts> bewirkt das Gegenteil: Es setzt den Cursor an den Anfang des folgenden Absatzes.

<INST/DEL>

Diese Taste löscht das Zeichen links neben dem Cursor. <CTRL INST/DEL> löscht das Zeichen unter dem Cursor. In beiden Fällen wird der Text hinter der Cursor-Position herangezogen und schließt die entstandene Lücke.

Einfüge- und Überschreibmodus

Mit <SHIFT INST/DEL> schaltet man zwischen Einfügemodus und Überschreibmodus hin und her. Nach dem Starten von MasterText 128 befindet sich das Programm im Einfügemodus. Dieser ist daran zu erkennen, daß im Feld am Ende der ersten Statuszeile die Meldung »EINF« erscheint. Drückt man nun <SHIFT INST/DEL>, so verschwindet die Meldung. Dies bedeutet, daß nun alle eingegebenen Zeichen den folgenden Text überschreiben. Die Erfahrung zeigt, daß es von Vorteil ist, ständig im Einfügemodus zu arbeiten.

Floskeltexte

Mit <CTRL Z> kann man die Tastenkombinationen <CTRL 1> bis <CTRL 8> mit oft benötigten Zeichenketten oder Floskeltexten belegen. Jede Zeichenkette kann bis zu 64 Zeichen lang sein. Will man ein Absatzende in einem Floskeltext verwenden, so geschieht dies mit <CTRL RETURN>. <RETURN> beendet die Eingabe und die Floskeltexte werden in den Speicher übernommen. Die Floskeltexte können selbstverständlich auch auf Diskette gespeichert werden. Dies erreicht man über die Funktion »Tabulatoren speichern«.

Die Blockbefehle

Die Blockbefehle dienen dazu, größere Textabschnitte im Speicher des Computers zu verarbeiten. MasterText 128 bietet die Möglichkeit, Blöcke zu löschen, zu kopieren, zu verschieben, auf Diskette zu speichern und von Diskette zu laden. Blöcke lassen sich zeilenweise definieren.

Block definieren

Um einen Block zu bearbeiten, muß er zunächst definiert werden. Dies geschieht mit zwei Tastenkombinationen. <CTRL A> definiert den Anfang eines Blockes. Entscheidend ist dabei die Position des Cursors. <CTRL B> erfüllt dieselbe Funktion für das Blockende. Um nur die Zeile, in der sich der Cursor befindet, als Block zu definieren, drückt man einfach

<CTRL A> , <CTRL B> . Die Zeile beziehungsweise der Block wird invers dargestellt. Die Markierung des Blocks kann mit <CTRL 9> ein- und mit <CTRL 0> ausgeschaltet werden.

Block löschen

Ein definierter Block kann mit <CTRL L> gelöscht werden. Mit dem nachfolgenden Text wird die entstandene Lücke wieder geschlossen.

Block kopieren

Beim »Block kopieren« ist die Position des Cursors von Bedeutung, denn an diese Stelle wird der vorher definierte Block kopiert, wobei er an seiner Ursprungsposition erhalten bleibt. Um diese Funktion auszulösen, drücken Sie <CTRL K> .

Block verschieben

<CTRL V> lautet der Befehl zum Verschieben eines Blockes. Auch bei diesem Befehl ist die Cursor-Position wichtig, denn an diese Stelle wird der Block verschoben. Der definierte Block wird an seiner alten Position gelöscht und an der neuen eingefügt.

Block speichern

Mit <CTRL W> wird ein Block auf Diskette gespeichert. In der rechten oberen Ecke des Bildschirms erscheint eine Eingabemaske, in die der Name eingegeben werden muß, den der Block auf der Diskette erhalten soll. Sind Sie Besitzer von mehreren Diskettenlaufwerken, so können Sie auch die Geräteadresse und die Laufwerksnummer frei wählen. In die entsprechenden Felder gelangt man mit <CRSR-abwärts> und <CRSR-aufwärts> .

Nach <RETURN> wird der Block auf der Diskette gespeichert. Sollte schon ein Text unter gleichem Namen auf der Diskette vorhanden sein, so meldet der Computer »FILE EXISTS«. Diese Fehlermeldung muß, wie jede andere Fehlermeldung auch, mit <RETURN> bestätigt

werden. Danach erscheint in der Kommentarzeile die Frage, ob die alte Datei durch die neue ersetzt werden soll. Bei Eingabe von <J> wird dies ausgeführt. Nach <N> befindet man sich wieder in der Eingabemaske und kann einen neuen Namen eingeben oder die Eingabemaske mit <ESC> verlassen.

Block laden

Der umgekehrte Vorgang wird mit <CTRL R> eingeleitet. Natürlich braucht für diese Funktion kein Block definiert zu werden. Jedoch ist die Cursor-Position entscheidend, denn an diese Stelle wird der Block geladen. Die Eingabemaske bei dieser Funktion und deren Bedienung ist analog zum »Block schreiben«. Der von der Diskette geladene Block ist auch gleich als solcher definiert; er kann mit den anderen Blockbefehlen weiterverarbeitet werden. Sollte durch das Einlesen eines Blockes die Kapazität des 64 000 Zeichen umfassenden Textspeichers überschritten werden, so wird der Ladevorgang abgebrochen.

Text löschen

Mit <CTRL CLR/HOME> kann der gesamte im Speicher befindliche Text gelöscht werden. Da man sich eine solch schwerwiegende Entscheidung immer noch einmal überlegen sollte, erfolgt eine Sicherheitsabfrage. Erst wenn Sie diese Frage mit »J« beantworten, wird die Funktion ausgeführt.

Suchen und Ersetzen

Die Funktionen »Suchen« und »Suchen und Ersetzen« werden mit der gleichen Tastenkombination, nämlich <CTRL S> aufgerufen. In der linken oberen Ecke des Bildschirms erscheint eine Eingabemaske. In dieser kann im ersten Feld das Suchwort eingetragen werden. Sind bestimmte Buchstaben dieses Wortes unbekannt oder nicht relevant, so kann mit einem »Joker« gearbeitet werden. Nach dem Drücken von <CTRL ?> erscheint auf dem Bildschirm ein reverses Fragezeichen, das für einen beliebigen Buchstaben steht. Soll nach einem Absatzendezeichen gesucht werden, so verwendet man hierfür <CTRL RETURN>.

Im nächsten Feld wird das Wort eingetragen, durch welches das Suchwort ersetzt werden soll. Hierbei darf der Joker nicht verwendet werden.

Wenn der Computer nicht zwischen Groß- und Kleinbuchstaben differenzieren soll, so ist in das nächste Feld »n« einzutragen. Das Suchwort sollte in diesem Fall jedoch nur aus Kleinbuchstaben bestehen, da alle Buchstaben im Text vor dem Vergleich mit dem Suchwort auch in solche umgewandelt werden.

Trennungsvorschläge:

Dieses Feld ist wichtig, wenn man nicht genau weiß, ob in dem Suchwort Trennungsvorschläge benutzt wurden. Wird »n« in diesem Feld eingetragen, beachtet der Computer die im Text vorkommenden Trennungsvorschläge nicht mehr.

Wortweise suchen (j/n):

Gibt man beispielsweise »ende« als Suchwort ein, findet MasterText 128 beim wortweisen Suchen nur dieses Wort. Bei Eingabe von »n« findet das Programm jedoch auch Wortteile wie etwa »suchende« oder »Zwölfender«.

Soll nach ganzen Wörtern gesucht werden, so geht das begreiflicher Weise schneller. Bei der Suche nach Satzzeichen sollte man jedoch immer buchstabenweise suchen, da der Computer sonst Satzzeichen am Ende eines Wortes nicht finden kann.

Im letzten Feld wird entschieden, ob automatisch oder mit Bestätigung gesucht oder ersetzt werden soll. Bei Eingabe von »j« fragt MasterText 128 sowohl, ob das Wort ersetzt als auch, ob der Suchvorgang weitergeführt werden soll. Durch die Eingabe von »j« kann man ein Wort suchen, ohne es zu ersetzen.

Nach Eingabe der Suchparameter wird der Vorgang mit <RETURN> gestartet. <RUN/STOP> unterbricht den Prozeß. Es erfolgt dann eine »j/n«-Abfrage, ob wirklich unterbrochen werden soll. Der Suchvorgang beginnt immer ab der aktuellen Cursor-Position. Soll also der ganze Text durchsucht werden, muß mit <CTRL CRSR-aufwärts> an den Textanfang gesprungen werden.

Tabulatoren

MasterText 128 kennt zwei Arten von Tabulatoren. Die Spaltentabulatoren markieren bestimmte Stellen in einer Zeile und werden an der aktuellen Cursor-Position mit <CTRL H> definiert. Diese Spalten können von links der Reihe nach mit <TAB> oder <RUN/STOP> angesprungen werden. Ist kein Tabulator-Stopp mehr definiert, so springt der Cursor in die Spalte 80 der jeweiligen Zeile.

Mit der zweiten Tabulator-Art von MasterText 128 können bestimmte Positionen im Text definiert werden. Da sich der Computer hierbei Spalte und Zeile der Cursor-Position merken muß, ist die Zahl der definierbaren Stellen auf 13 begrenzt. Gesetzt wird eine Position mit <CTRL T>. Gelöscht werden die Tabulatoren mit <CTRL C> und anschließendem <CTRL T>. Dabei muß der Cursor an der entsprechenden Tabulator-Position sein. Mit <CLR/HOME> können die Texttabulatoren der Reihe nach angesprungen werden. Die Reihenfolge der Definition ist dabei entscheidend.

Dateilänge zeigen

Oft ist es nützlich zu wissen, wie lang der eben geschriebene Text ist, wenn man sich beispielsweise nach gewissen Vorgaben zu richten hat. MasterText zeigt nach Eingabe von <CTRL D> die Textlänge sowohl in Zeichen als auch in Blocks auf der Diskette an. Die Angabe der Blöcke gilt nicht für Kompakt-Files, zu denen wir später noch kommen. Diese sind in der Regel wesentlich kürzer. Die Meldung der Textlänge braucht nicht mit <RETURN> bestätigt zu werden. Sie erlischt automatisch mit dem nächsten Tastendruck. Will man einen Text speichern, wird die Dateilänge automatisch in der mittleren Statuszeile eingeblendet.

Text formatieren

Im Formular werden Werte festgelegt, die das spätere Aussehen des Textes bestimmen. Das Formular wird durch <CTRL F> aufgerufen (Bild 3). In der linken Hälfte des Bildschirms erscheint eine Eingabemaske, in der die grundlegenden Werte für das Format des Textes festgelegt werden. Das erste Feld bestimmt den oberen Rand. Bei »000« fängt der Computer in der

ersten Zeile an zu drucken. Der linke Rand wird im nächsten Feld bestimmt. Da die Zählung bei 0 beginnt, bedeutet eine Randangabe von 9, daß ab Position 10 gedruckt wird. Oder anders ausgedrückt: Es werden 9 Leerzeichen am Anfang einer jeden Zeile gedruckt. Der rechte Rand bestimmt die Spalte, bis zu der (einschließlich) gedruckt wird. Da auch hier von 0 an gezählt wird, bedeutet ein Wert von 74, daß bis einschließlich Spalte 75 gedruckt wird. Der untere Rand bestimmt die Zeile, bis zu der gedruckt werden soll. Die Anzahl der Zeilen pro Seite beträgt bei DIN A4 normalerweise 72. Die Zahl ist gerade bei Endlospapier wichtig, wenn das Papier nicht nach jeder Seite neu eingespannt werden soll.

Schriftart

In diesem Feld wird eine von drei Schriftarten gewählt. Dies funktioniert allerdings nur, wenn Ihr Drucker diese beherrscht und die Codes für die Umschaltung in der Steuerzeichentabelle definiert wurden. Für einen MPS 801 läßt sich jedoch nichts wählen. Gleiches gilt für den Zeilenabstand, den Sie im nächsten Feld festlegen. Die Festlegung von Schriftart und Zeilenabstand geschieht durch Senden einer Steuersequenz an den Drucker. Wie diese Steuersequenz definiert wird, beschreiben wir später.

Format

Dieses Feld ist unabhängig vom verwendeten Druckertyp. Hier wird das Format des Textausdrucks bestimmt. Zur Auswahl stehen linksbündiges Drucken (=0), rechtsbündiges Drucken (=1), Zentrieren in die Mitte der Zeile (=2) und Blocksatz, der links- und rechtsbündiges Drucken veranlaßt (=3).

Einzelblatt

Bei Eingabe von »j« wird der Ausdruck nach jeder Seite zum Wechseln des Papiers unterbrochen. Bei Verwendung von Endlospapier ist hier »n« einzutragen.

Fußnoten

In den nächsten Feldern können bis zu 6 Fußnoten beziehungsweise Kopfzeilen festgelegt werden. Das sind Zeilen, die auf jeder Seite wieder erscheinen. Zu jeder Fußnote gehören noch zwei Felder, in denen die Zeile und Spalte der Fußnote, in der diese gedruckt werden sollen, eingegeben werden. Es ist darauf zu achten, daß die Zeilennummern immer außerhalb des Druckbereichs liegen müssen, also entweder kleiner als der obere Rand (Kopfzeile) oder größer als der untere Rand (Fußnote), da Fußnoten und Kopfzeilen sonst nicht gedruckt werden. Wenn 35 Zeichen für eine Fußnote nicht ausreichen, so kann diese in der nächsten Fußnote oder Kopfzeile fortgesetzt werden. Die Spalten- und Zeilenangabe für den zweiten Teil der Fußnote ist dann irrelevant. Wenn Fußnoten oder Kopfzeilen nicht verkettet werden sollen, muß allerdings das letzte Buchstabenfeld der jeweiligen Fußnote (Kopfzeile) freigelassen werden.

Mit Hilfe der letzten drei Felder lassen sich Formulare während des Druckens nachladen. Das Formular wird in der angegebenen Zeile nachgeladen. So erhalten Sie beispielsweise die Möglichkeit, in bestimmten Zeilen den Text einzurücken und Platz für Bilder zu schaffen oder andere Fußnoten zu drucken.

Mit <RETURN> werden die Werte übernommen.

Uhr mit Alarm

Die in der mittleren Statuszeile angezeigte Uhr kann mit der Tastenkombination <CTRL U> gestellt werden. In der Mitte des Bildschirms erscheint dann eine Eingabemaske (Bild 4). Im ersten Feld wird die Uhrzeit im Format »hhmmss« eingetragen. Das zweite Feld wird im gleichen Format ausgefüllt. Dadurch wird die Alarmzeit festgelegt. Es folgen noch drei Felder, die jeweils die Eingabe von »j« oder »n« erfordern. Hier wird festgelegt, ob der Alarm aktiviert ist, ob zu jeder vollen Stunde ein Gong ertönen soll und zuletzt, ob die Uhrzeit gesetzt beziehungsweise geändert werden soll. <RETURN> beendet wie immer die Eingaben. Mit <ESC> kann das Menü auch ohne Neueinstellungen verlassen werden.

Steuerzeichen

Steuerzeichen im Text übernehmen verschiedene Aufgaben. MasterText 128 kennt zwei verschiedene Arten von Steuerzeichen. Bei der einen Art wird eine vorher in der Steuerzeichentabelle festgelegte Sequenz an den Drucker gesendet. Diese Steuerzeichen funktionieren nur, wenn der angeschlossene Drucker sie beherrscht. Im Gegensatz dazu funktioniert die andere Art von Steuerzeichen auf allen Druckern. Es sind Steuerzeichen, die das Format des Textausdrucks verändern, den Rand neu setzen, das Datum oder die Seitennummer drucken. Alle Steuerzeichen der ersten Gruppe werden mit Kleinbuchstaben, alle Steuerzeichen der zweiten Gruppe werden mit Großbuchstaben ausgelöst. Um die Steuerzeichen von normalem Text zu unterscheiden, werden sie mit der ALT-Taste eingeleitet. Auf dem Bildschirm wird diese Taste durch ein inverses »S« dargestellt.

D: Datum einfügen

Um das im Textausgabe-Window definierte Datum an einer bestimmten Stelle zu drucken, verwendet man die Tastenkombination <ALT> <SHIFT D>.

F: Formatanweisung

Zu den Steuerzeichen, die auf allen Druckern funktionieren, gehört weiterhin das Steuerzeichen, das das Format des weiteren Textes festlegt. Es wird mit »F« eingeleitet. Danach folgt eine Zahl zwischen 0 und 3, die die gleiche Bedeutung hat wie im Formular. Die Formatanweisung wirkt sofort ab der Zeile, in der sie gegeben wird. (Beispiel: <ALT> F2 – Zentrierung einschalten).

RR+/-xx, RL +/-xx: Randeinstellung

Die Randeinstellung läßt sich mit »R« verändern. Danach folgt ein »L« oder ein »R«, je nachdem, ob man den linken oder rechten Rand versetzen will. Das nächste Zeichen ist ein »+« oder ein »-«. Dann folgt eine zweistellige Dezimalzahl, die angibt, um wie viele Zeichen

der Rand versetzt werden soll (<ALT>RL+05: versetzt den linken Rand um 5 Zeichen nach links; <ALT>RR-10: versetzt den rechten Rand um 10 Zeichen nach rechts). Die Randanweisung gilt erst für die nächste Zeile.

<↑> : Seitenvorschub

Dieses Steuerzeichen löst einen Seitenvorschub beim Drucker aus. Der folgende Text wird auf der nächsten Seite weitergedruckt.

S: Seitennummer

Die Verwendung dieses Steuerzeichens ist vor allen Dingen im Formular als Fußnote sinnvoll (<ALT>S – druckt die Fußnote im Format: – 1 –).

L: Nachladen von Texten

Die Textverkettung durch das Nachladen von Texten beim Ausdruck ist mit dem L-Befehl möglich. Nach <ALT><L> erscheint folgendes auf dem Bildschirm: [invers S] L"0:[15 SPACE]t",8. Es muß also nur der Name des Textes eingetragen werden. Wenn Sie ein Formular nachladen wollen, so müssen Sie das »t« in ein »f« ändern, für Druckerparameter entsprechend in ein »d«. Der Text wird während des Ausdrucks in den Text, der sich im Speicher befindet, eingefügt. Wenn der Text, in dem dieser Befehl gegeben wurde, jedoch auch schon von der Diskette nachgeladen wird, kann der neue Text nicht mehr eingefügt werden. In diesem Fall werden die Texte nur verkettet.

W: Warten

Sollten die Texte nicht mehr auf eine Diskette passen oder Sie benötigen Zeit zum Typenradwechsel während des Druckens der Texte, so kann der Textdruck mit dem Steuerzeichen »W« angehalten werden. Der Computer wartet dann so lange, bis eine Taste gedrückt wird, ehe er mit der Ausgabe fortfährt.

k0: Kursivschrift aus	r0: Revers-Schrift aus	z1: Zeilenabstand 1
k1: Kursivschrift ein	r1: Revers-Schrift ein	z2: Zeilenabstand 2
-0: Unterstreichung aus	f0: Fettschrift aus	z3: Zeilenabstand 3
-1: Unterstreichung ein	fl: Fettschrift ein	!: Ausrufezeichen-Funktion
u: Subscript ein	sl: Schriftart 1	Erklärung im Text
o: Superscript ein	s2: Schriftart 2	0-9: weitere frei definierbare
n: Normalschrift ein	s3: Schriftart 3	Steuerzeichen von 0 bis 9

Tabelle 1. Die druckerspezifischen Steuerzeichen und ihre Funktion.

Axx: Einfügen von Daten bei Rundschreiben

Dieser Punkt wird weiter unten ausführlich behandelt.

Druckerspezifische Steuerzeichen

Die druckerspezifischen Steuerzeichen werden mit kleinen Buchstaben abgekürzt. Da das Ergebnis dieser Steuerzeichen vom angeschlossenen Drucker abhängt, müssen diese erst in der Steuerzeichentabelle definiert sein. In der vorliegenden Version finden Sie jedoch in den Druckertreibern schon fertige Tabellen. Die Steuerzeichen und deren Funktion finden Sie in Tabelle 1.

Am Beispiel der Kursivschrift sei die Verwendung der Steuerzeichen hier veranschaulicht:

<ALT>k1: Kursivschrift einschalten

Eine Sonderstellung nimmt das Steuerzeichen »!« ein. Zunächst wird auch bei diesem Steuerzeichen die Sequenz gesendet, die Sie in der Steuerzeichentabelle für diese Funktion festgelegt haben. Danach wird jedoch noch die Hex-Zahl an den Drucker gesendet, die diesem Zeichen im Text folgt. Auf diese Weise lassen sich bei Epson-Druckern verschiedene Schriftarten einstellen. Es muß in diesem Fall der Code für »ESC!«, was \$1B21 entspricht, in der Steuer-

zeichentabelle eingetragen werden. Soll diese Funktion beim Drucken aktiviert werden, so ist im Text folgendes zu schreiben: <ALT>!xx. »xx« steht für den zu sendenden Code. Bei Epson-Druckern löst beispielsweise der Code \$21 den Druck in Fettschrift aus.

Der Taschenrechner

Für Rechnungen im Text steht ein Taschenrechner zur Verfügung, der mit <CTRL M> aufgerufen wird. Mit diesem sind die vier Grundrechenarten und Prozentrechnungen möglich. Beim Aufruf erscheint auf der linken Seite des Bildschirms der »Stack« des Taschenrechners. Dieser verfügt über 32 Register, so daß auch umfangreiche Rechnungen durchgeführt werden können. Daneben findet sich eine Übersicht über den numerischen Tastenblock des C 128 mit der im Taschenrechner verwendeten Tastenbelegung. Der Taschenrechner arbeitet mit der sogenannten UPN-Notation (umgekehrte polnische Notation), wie Taschenrechner von Hewlett Packard. Man gibt dabei erst die Zahlen und dann die Operatoren ein. Eine <=>-Taste gibt es bei UPN-Rechnern nicht. Nach der Eingabe der Operation (+,-,*,/) erscheint sofort das Ergebnis. Diese Eingabeweise ist zwar etwas gewöhnungsbedürftig, besitzt aber auch einige Vorzüge: Zwischenergebnisse werden sofort angezeigt und man kommt ohne Klammern aus. Beispiele:

Aufgabe: 3+4
Eingabe: 3 <ENTER> 4 <+>

Aufgabe: 3+4*5
Eingabe: 3 <ENTER> 4 <ENTER> 5 <*> <+>
oder 4 <ENTER> 5 <*> 3 <+>

Aufgabe: (3+4)*5
Eingabe: 3 <ENTER> 4 <+> 5 <*>

Aufgabe: ((5 + 4) * (4 + 2) + 9) * 3
Eingabe: 5 <ENTER> 4 <+> 4 <ENTER> 2 <+> <*> 9 <+> 3 <*>

Es sind übrigens weniger Tastendrucke als bei einem normalen Taschenrechner nötig. Dank des Stack sind Klammern überflüssig. Da der Taschenrechner auch zum Schreiben von Rechnungen gedacht ist, ist die Prozentrechnung integriert. Die »%«-Funktion liegt auf <F3>. Die darauffolgende Taste löst verschiedene Funktionen aus:

< * > berechnet die Höhe des Prozentsatzes einer Zahl.

Beispiel: 5 Prozent von 145.
145 <ENTER> 5 <F3> < * >; Ergebnis: 7,25

< + > addiert den Prozentsatz zur eingegebenen Zahl hinzu,

< - > zieht den errechneten Prozentsatz ab. Mit

< F7 > berechnet man, wieviel Prozent eine Zahl von einer anderen darstellt.

Des weiteren existieren Funktionen zur Manipulation des Stack. Mit <CRSR-abwärts> und <CRSR-aufwärts> markiert man die Zahlen im Stack, <INST/DEL> löscht ein Stack-Register, <SHIFT INST/DEL> verschiebt den Stack-Inhalt und schafft Platz für eine neue Zahl. Register 31 geht dabei allerdings verloren. Mit <SHIFT ENTER> wird das Ergebnis der letzten Rechnung beziehungsweise die letzte in den Stack übernommene Zahl an die aktuelle Position kopiert. Mit <CTRL F> gelangt man in das Formatfeld unten rechts am Bildschirm. Dort kann die Zahl der angezeigten Nachkommastellen, die maximale Länge der Zahlen und die Art des Trennungszeichens (»,.« oder »,«) festgelegt werden. Die Länge der Zahl ist nur bei Übernahme von Zahlen in den Text relevant. Die Zahlen werden in einem Feld von Leerzeichen rechtsbündig dargestellt.

Zahlentransfer zwischen Taschenrechner und Text-Editor

Natürlich können auch Werte im Taschenrechner importiert und exportiert werden. Beide dazu nötigen Befehle werden im Editor gegeben, da jeweils die Cursor-Position wichtig ist.

- <CTRL I> überträgt eine Zahl aus dem Editor in den Taschenrechner. Dazu muß der Cursor vorher auf den Anfang dieser Zahl bewegt werden. Die Übernahme erkennt man auch daran, daß sich der Stackpointer in der unteren Statuszeile automatisch um den Wert eins erhöht.
- <CTRL O> ist die Umkehrfunktion. Die Zahl, auf die der Stackpointer zeigt, wird an der Cursor-Position in den Text übertragen. Dieser wird je nach Moduswahl (Einfügen/Überschreiben) gelöscht oder nach rechts verschoben.

Menüpunkt »Laden« und »Speichern«

Bei Verwendung dieses Menüpunktes werden Daten von der Diskette in den Arbeitsspeicher des Computers geladen. Die Art der Daten kann in einem Untermenü gewählt werden.

Die HELP-Funktion

Ein Druck auf <HELP> in einer Eingabemaske zeigt alle Dateien auf Diskette an, die dem aktuellen Dateityp entsprechen. Aus dieser Liste kann mit den Cursor-Tasten eine Datei ausgewählt und in die Eingabemaske übernommen werden (Bild 5). Diese sehr nützliche HELP-Funktion ist, ausgenommen bei den Speicherfunktionen, bei allen Diskettenzugriff-Funktionen erreichbar.

Kompakt-Files

Falls Sie nicht gerade Texte mit MasterText 64 austauschen wollen, sollten Sie alle Blöcke und Texte als »Kompakt-Files« speichern. Diese Einstellung wird in den Laden/Speichern-Masken mit »j« oder »n« im entsprechenden Feld vorgenommen. Die Texte beanspruchen dann nicht so viel Speicherplatz auf der Diskette, da alle Leerzeichen eliminiert werden.

Laden von Texten

Wählt man den ersten Punkt im Untermenü, so gelangt man in eine Eingabemaske. Diese ist schon vom Blockladen und Blockspeichern her bekannt. Auch die Bedienung ist identisch. Der Unterschied zum Blockbefehl besteht darin, daß ein eventuell im Speicher vorhandener Text gelöscht wird. Dies ist natürlich nur dann der Fall, wenn der neue Text auch gefunden wurde. Vorher wird jedoch noch eine Sicherheitsabfrage durchgeführt, um versehentliches Löschen des Textes zu verhindern. Der Ladevorgang selbst kann mit <RUN/STOP> abgebrochen werden. Das zweite Feld in der Eingabemaske (der 16. Buchstabe des Dateinamens), erreichbar mit <CRSR-abwärts>, bestimmt den Dateityp. Dabei gilt folgende Vereinbarung:

- t Text-File
- k Kompakt-File
- f Formular
- b Tabulatoren, Stack, Zahlenformat, Floskeltexte
- d Druckerparameter
- ü DFÜ-Parameter
- r Rundschreibendatei
- p Printerfiles (Umleitung der Druckerausgabe auf Diskette)

Laden von Formularen

Hier werden Formulare von der Diskette in den Speicher des Computers geladen. Unter dem Formular versteht man die Werte, die man im Editor mit <CTRL F> ändern kann. Es sind dies Daten für Randeinstellungen, Format (Links-, Rechtsbündig, Blocksatz), Schrift etc. Ansonsten ist die Bedienung identisch mit dem Punkt »Text laden«. <HELP> zeigt hier alle Formulare, die auf der Diskette sind.

Laden von Tabulatoren, Floskeltexten und Stack

Mit dieser Funktion werden Tabulatoren von der Diskette geladen, die zuvor als Spalten- oder als Texttabulatoren definiert und gespeichert wurden. Außer den Tabulatoren werden aber auch

noch der Stack und die Formatangaben des Taschenrechners geladen. Die mit <CTRL Z> definierten Textkonstanten werden ebenfalls mit diesem Menüpunkt wieder in den Speicher übertragen.

Laden von Druckerparametern

Die Druckerparameter-Files auf Diskette umfassen eine Tabelle der Steuerzeichen für den Drucker, der Druckertabelle sowie die Angaben über Geräteadresse und Sekundäradresse des Druckers. Bei Verwendung der RS232-Schnittstelle kommen noch die Angaben für die Datenübertragung hinzu.

Laden von DFÜ-Parametern

Mit Hilfe dieses Menüpunktes ist es möglich, die Tabellen für den Datenaustausch über die RS232-Schnittstelle in den Speicher des Computers zu laden. Diese Tabellen entsprechen in etwa denen der Druckerparameter. Die genaue Erläuterung folgt beim Menüpunkt »DFÜ«.

Speichern

Die Untermenüpunkte entsprechen denen des »Laden«-Menüs. Tritt die Fehlermeldung »file exists« auf, so fragt der Computer, nach Bestätigung mit <RETURN>, ob er die alte Datei durch die neue ersetzen soll. Beantworten Sie die Frage mit »J«, so löscht der Computer die alte Datei und speichert die neue unter gleichem Namen. Bei Eingabe von »N« springt der Cursor in die Eingabemaske zurück.

Menüpunkt »Ausgabe«

Zur formatierten Anzeige des Textes auf dem Bildschirm oder zum Drucken benötigt man diesen Menüpunkt. Hierbei stehen vier Untermenüs zur Verfügung. Nach Anwahl einer dieser Punkte erscheint zunächst eine Eingabemaske (Bild 6). Der Eintrag in diese bestimmt, wie oft der Text gedruckt werden soll oder wie die erste Seitennummer lautet, falls das Steuerzeichen

für die Seitennummer in einer Fußnote im Formular verwendet wurde. Die nächsten Felder legen den Druckbereich fest, das heißt, die erste und die letzte Seite, die gedruckt werden soll. Auch ist es beim Punkt »Jede x. Seite« möglich, festzulegen, daß nur jede zweite Seite ausgedruckt werden soll. Auf diese Weise haben Sie die Möglichkeit, erst alle Vorderseiten und in einem zweiten Durchgang alle Rückseiten zu drucken. Dies ist bei Verwendung von Einzelblattpapier sinnvoll. Diese Option beansprucht das dritte Eingabefeld. Schließlich kann das Tagesdatum auf dem vierten Feld eingetragen werden, wenn im Text das Steuerzeichen zum Einfügen des Datums verwendet wurde.

Bei Belassen aller Werte auf »000« druckt MasterText 128 alle Seiten. Mit <RETURN> wird die Eingabe abgeschlossen. Der Computer beginnt dann mit der Ausgabe des Textes.

Normale Ausgabe

Dies ist die einfachste Ausgabeform. Es wird immer eine Seite des Textes auf dem Bildschirm dargestellt. Dabei wird die volle Bildschirmbreite und -länge ausgenutzt. Nach jeder Bildschirmseite ergeht an Sie die Aufforderung, die <RETURN>-Taste zu drücken. Der Computer zeigt dann die nächste Bildschirmseite. Die Ausgabe kann mit der <ESC>-Taste abgebrochen werden.

Formatierte Ausgabe

Auch bei der formatierten Ausgabe handelt es sich um die Ausgabe auf dem Bildschirm. Es erscheint jedoch im gleichen Format wie später auf dem Papier. So können Sie sich ein Bild vom späteren Aussehen des Textes machen. Da der Bildschirm jedoch nur 25 Zeilen darstellt, wird der Text, wenn der untere Bildschirmrand erreicht wird, immer um eine Zeile nach oben gescrollt. Um den Text trotzdem lesen zu können, kann die Ausgabe mit der Taste <SPACE> gestoppt werden. Die Ausgabe wird nach einem beliebigen Tastendruck fortgesetzt. Wie schon bei der normalen Ausgabe, können Sie auch die formatierte Ausgabe mit <ESC> abbrechen. Auch bei dieser Ausgabeform erscheint nach jeder Seite die Aufforderung, <RETURN> zu drücken, bevor die nächste Seite gedruckt wird.

Unterstreichung oder Kursivschrift wird hier direkt am Bildschirm angezeigt (Bild 7).

Druckerausgabe

Bei diesem Untermenü erscheinen drei weitere Felder in der Eingabemaske. Man kann die Ausgabe des Textes auch auf ein Diskettenlaufwerk umleiten. Sie müssen dazu lediglich die Geräteadresse im letzten Feld von der des Druckers auf die des Diskettenlaufwerks ändern. Das Printerfile auf Diskette ist dann eine sequentielle ASCII-Datei mit der Filenamen-Erweiterung »p«.

Die Ausgabe erfolgt nach den Vorgaben im Formular. Vor der Ausgabe erhalten Sie die Aufforderung, Papier einzulegen und <RETURN> zu drücken. Diese Aufforderung erscheint nach jeder gedruckten Seite, es sei denn, Sie haben Endlospapier im Formular gewählt. In diesem Fall druckt der Computer durchgehend.

Rundschreiben

Bei der Anwahl dieses Punktes gelangt man vor der Ausgabe des Rundschreibens noch in eine zweite Eingabemaske. In dieser muß im ersten Feld eingegeben werden, wieviele Datenfelder ein Datensatz enthält. Haben Sie in der Datei etwa Adressen mit Anrede, Name, Vorname, Straße, Wohnort gespeichert, so sind dies 5 Datenfelder. Diese Zahl muß in das erste Feld eingetragen werden. Der Name einer Rundschreibendatei muß im 16. Zeichen mit »r« gekennzeichnet sein.

Die Kennzeichnung der Datenfelder im Text ist folgende: Wie alle Steuerzeichen, werden auch die Stellen, an denen ein bestimmtes Datenfeld gedruckt werden soll, zunächst mit <ALT> eingeleitet. Danach folgt ein »A« für Adressenfeld. Die zweistellige Dezimalzahl dahinter gibt an, das wievielte Feld eines Datensatzes gedruckt werden soll. Um beispielsweise mit der eben beschriebenen Datei eine Adresse zu drucken, muß folgendes im Editor eingegeben werden:

<ALT>A00	Anrede
<ALT>A02<ALT>01	Vorname, Nachname
<ALT>A03	Straße
<ALT>A04	Wohnort

Bei den Rundschreibendateien handelt es sich um sequentielle Dateien. Die Daten müssen der Reihe nach in dieser Datei gespeichert sein. Die einzelnen Daten müssen durch ein »CR« (RETURN = CHR\$(13)) voneinander getrennt sein. Am Ende der Datei muß sich noch ein zusätzliches Zeichen als Endkennung befinden, da sonst der letzte Datensatz nicht mehr gedruckt wird.

Menüpunkt »Floppy«

Dieser Menüpunkt dient der Kommunikation mit dem Diskettenlaufwerk und beinhaltet folgende Unterpunkte:

Inhaltsverzeichnis

In der Eingabemaske kann die Laufwerksnummer verändert werden, falls Sie ein zweites Laufwerk besitzen. Interessieren nur bestimmte Teile des Directorys, so können Sie das Feld »Suchmaske« ausfüllen. Der Computer zeigt dann nur die Files an, auf die die Buchstabenkombination in diesem Feld zutrifft. »?« und »*« dienen dabei in der bekannten Weise als Joker. Nach <RETURN> erscheint in der linken Hälfte des Bildschirms das Inhaltsverzeichnis. Mit <SPACE> oder <NO SCROLL> wird die Ausgabe gestoppt; mit <ESC> abgebrochen.

Dateien umbenennen

Rechts oben auf dem Bildschirm erscheint wieder eine Eingabemaske. Das Feld »alter Name« ist schon mit dem aktuellen Dateinamen vorbesetzt.

Im Feld darunter wird der neue Dateiname angegeben. Hierbei können Sie keinen Dateityp angeben, da dieser nicht verändert werden darf. Die beiden übrigen Felder sind wieder nur für Besitzer von mehreren Laufwerken interessant. Eventuelle Fehlermeldungen erscheinen in der zweiten Statuszeile und müssen mit <RETURN> bestätigt werden. Die HELP-Funktion ist hier möglich.

Dateien löschen

Wieder erscheint eine Eingabemaske. Der Dateiname ist schon mit dem aktuellen Dateinamen vorbesetzt. Auch in dieser Eingabemaske muß der Dateityp angegeben werden. Auch hier ist der Einsatz der HELP-Funktion erlaubt.

Nach Auswahl der zu löschenden Datei erfolgt eine Sicherheits-Abfrage, die mit <J> oder <N> bestätigt werden muß. Bei Eingabe von <N> gelangt man zurück in die Eingabemaske. Nach dem Löschen zeigt der Computer an, wieviele Dateien gelöscht wurden. Diese Meldung wird mit <RETURN> bestätigt. Auch hier kann die HELP-Taste verwendet werden.

Befehl senden

Dieser Menüpunkt dient zum Senden eines beliebigen Befehls an das Diskettenlaufwerk. Es sind alle Befehle möglich, die auch im Handbuch zum Diskettenlaufwerk stehen. Die Syntax bei MasterText 128 befindet sich in Tabelle 2. Die Befehle sind in das erste Feld der Eingabemaske einzutragen. Die Pfeile unter diesem Feld kennzeichnen immer Anfang und Ende des Dateinamens, wenn die Dateinamen 16 Buchstaben lang sind. Dies ist bei allen Dateien der Fall, die mit MasterText erzeugt werden.

Im zweiten Feld der Eingabemaske wird bei Verwendung mehrerer Laufwerke die Geräteadresse gewählt. Ausgeführt wird der Befehl mit <RETURN>. Eventuelle Fehlermeldungen erscheinen in der mittleren Statuszeile. Dieser Menüpunkt dient zum Anpassen von MasterText 128 an Ihren Drucker.

n0:	Diskettenname,id	Diskette formatieren
v0:		Diskette neu ordnen
s0:	Dateiname	Datei löschen
r0:	neuer Dateiname=0:alter Dateiname	Datei umbenennen
c0:	neuer Dateiname=0:alter Dateiname	Datei kopieren

Tabelle 2. Die Syntax der Diskettenbefehle unter MasterText 128. Der Dateityp muß mit angegeben werden.

Menüpunkt Drucker

Druckerparameter

Links oben auf dem Bildschirm erscheint eine Eingabemaske. In dieser können Sie die Geräteadresse des Druckers (4) oder der RS232-Schnittstelle (2) einstellen und die Sekundäradresse des Druckers festlegen. Bei Verwendung der RS232-Schnittstelle für den Druck müssen die benötigten Parameter im dritten Feld eingetragen werden. Die Werte müssen dazu jeweils summiert und dann in eine Hexadezimalzahl umgerechnet werden. Aus jeder Tabelle darf nur eine Zahl genommen werden. Wie sich die beiden Hexadezimalzahlen berechnen, können Sie Tabelle 3 entnehmen:

dez.	hex.	Funktion
Erste Zahl:		
Übertragungsrate (Bit/Sek.)		
1	01	50
2	02	75
3	03	110
4	04	134.5
5	05	150
6	06	300
7	07	600
8	08	1200
9	09	1800
10	0A	2400

Tabelle 3. Diese Werte benötigen Sie zur Errechnung der DFÜ-Parameter (Fortsetzung nächste Seite).

dez.	hex.	Funktion
Zweite Zahl:		
Übertragungsart		
0	00	Vollduplex
16	10	Halbduplex
Paritätsprüfung		
0	00	keine Paritätsprüfung, kein 8. Datenbit
32	20	ungerade Parität
96	60	gerade Parität
160	0A	8. Datenbit immer 1, keine Paritätsprüfung
224	0E	8. Datenbit immer 0, keine Paritätsprüfung

Tabelle 3. (Fortsetzung)

Nach der Berechnung werden die beiden Hexadezimalzahlen in das dafür vorgesehene Feld eingetragen und mit <RETURN> übernommen.

Druckertabelle

Hier wird jedem Zeichen auf dem Bildschirm ein Code zugeordnet, der zum Ausdruck des Zeichens an den Drucker gesendet werden muß. Dazu drückt man die Taste, der ein bestimmter Code zugeordnet werden soll. Hinter dem Wort »Code« erscheint die Zahl, die bisher diesem Buchstaben zugeordnet wurde. Diese können Sie jetzt verändern. Nach <RETURN> wird die Zahl in die Tabelle übernommen. <RETURN> ohne Eingabe eines Zeichens führt zurück ins Hauptmenü. Diese auf den ersten Blick etwas umständliche Funktion kann bei der

dez.	hex.	Funktion
Erste Zahl:		
Anzahl der Datenbits		
0	00	8 Bit
32	20	7 Bit
64	40	6 Bit
96	60	5 Bit
Anzahl der Stoppbits		
0	00	1 Stoppbit
128	80	2 Stoppbit
Zweite Zahl:		
Handshake		
0	00	3-Draht-Handshake
1	01	x-Draht-Handshake

Tabelle 3. (Schluß)

Anpassung von exotischen Druckern sehr segensreich sein, da man hier beispielsweise den deutschen Umlauten und anderen Sonderzeichen die Codes, welche im Druckerhandbuch aufgeführt sind, eindeutig zuweisen kann. MasterText 128 erhält durch diese Funktion eine enorme Flexibilität für die Druckeranpassung.

Steuerzeichen

Zunächst erscheint eine Tabelle (Bild 8). In dieser können den druckerspezifischen Steuerzeichen die Sequenzen zugeordnet werden, mit denen die entsprechenden Funktionen im Drucker ausgelöst werden. Alle Werte werden in hexadezimaler Schreibweise eingegeben.

Als Beispiel sei hier das Einschalten der Unterstreichung bei Epson-Druckern angeführt. Der vom Drucker benötigte Code lautet:

ESC-1

Umgesetzt in hexadezimale Schreibweise bedeutet dies:

ESC = CHR\$(27) = \$1B

- = CHR\$(45) = \$2D

1 = CHR\$(49) = \$31

Es muß also im Feld »-1« »1b2D31« eingetragen werden. Entsprechend ist auch mit den übrigen Feldern zu verfahren. Eine Sonderstellung nehmen die Steuerzeichen »CR« und »Reset« ein. Diese können nicht vom Text aus gesendet werden. »Reset« wird automatisch vor dem Drucken eines Textes gesandt. Es kann dabei beispielsweise der deutsche Zeichensatz eingeschaltet, die Papiererkennung aktiviert werden etc. »CR« wird am Ende jeder Zeile gesendet. Es bewirkt den Zeilenvorschub und den Wagenrücklauf (Carriage Return + Line Feed).

Nach der Installation eines Druckers sind alle drei Tabellen schon mit Standardwerten vorbesetzt. Die Tabellen brauchen also nur verändert zu werden, wenn der Ausdruck nicht richtig funktioniert oder spezielle Steuerzeichen benötigt werden.

Menüpunkt »DFÜ«

Bei dieser Funktion kommt MasterText 128 einer Tendenz entgegen, Textverarbeitungsprogramme mit DFÜ-Eigenschaften auszustatten. Da Textverarbeitung und Terminalprogramme sehr viele ähnliche Funktionen und Programmroutinen besitzen, ist die Implementation solcher Features kein großer Aufwand und erhöht die Leistungsfähigkeit und Flexibilität eines Programms enorm.

Es ist möglich, Daten in Textform zu empfangen und zu senden. Außerdem ist ein Terminalmodus vorhanden. Zwischen diesen einzelnen Funktionen können Sie im ersten Untermenü zur DFÜ wählen. Mit <CRSR-rechts> gelangt man ins zweite DFÜ-Menü. In diesem können die Parametereinstellungen analog zur Druckeranpassung vorgenommen werden. Dadurch hat man die Möglichkeit, eine Code-Umwandlung zwischen verschiedenen Textprogrammen oder ASCII-Formaten vorzunehmen.

Text empfangen

Mit dieser Funktion können Texte über die RS232-Schnittstelle empfangen werden. Auf dem Bildschirm erscheint zunächst eine Eingabemaske. Diese muß nur dann ausgefüllt werden, wenn nicht über die RS232-Schnittstelle Texte empfangen werden sollen, sondern beispielsweise ein Text, der mit einer anderen Textverarbeitung erstellt wurde, von Diskette gelesen werden soll. Bei Laufwerk ist dann »8« oder »9« einzutragen. Bei der Übertragung in den Speicher wird dann eine Konvertierung der Zeichen gemäß der DFÜ-Empfangstabelle vorgenommen. Dies ist vor allem sinnvoll, wenn man Texte von anderen Programmen einlesen will, die eine andere Codierung der Zeichen verwenden. Nach <RETURN> ist der Computer im Empfangsmodus. Mit <RUN/STOP> kann die Übertragung von Hand abgebrochen werden. Es ist möglich, mehrere Texte hintereinander zu empfangen. MasterText 128 schreibt den Text an die Stelle in den Speicher, an der der Cursor steht. Bei dieser Empfangsart werden auch eventuell gesendete Steuerzeichen erkannt.

Text senden

Bei diesem Menüpunkt gelangt man in dieselbe Eingabemaske wie bei der Druckerausgabe. Es wird jetzt jedoch mit den Parametern der DFÜ-Tabellen gesendet. Man kann also mit anderen Steuerzeichen und Zeichencodes arbeiten als bei der Druckerausgabe. Dies ist gerade bei Commodore-Druckern wichtig. Man hat so die Möglichkeit, den Drucker mit der Commodore-eigenen Zeichenbelegung zu steuern und gleichzeitig über einen Akustikkoppler mit ASCII-Werten zu übertragen.

Terminal

Dies ist der eigentliche DFÜ-Modus. Der Bildschirm wird gelöscht und der Computer wartet auf Eingaben von Ihnen oder dem zweiten, per Telefon angeschlossenen Computer. Wurde Halbduplex als Übertragungsart gewählt (bei den Empfangsparametern), so erscheinen Ihre Eingaben sofort auf dem Bildschirm. Will man den Datentransfer protokollieren, so können Sie dies mit <CTRL P> tun. Die Schriftfarbe ändert sich dabei von grün auf hellblau. Ihre weiteren Eingaben und die des anderen Computers werden nun auch gespeichert. Der vorherige Text bleibt dabei erhalten. Die empfangenen Texte können also nachbearbeitet werden. Mit der gleichen Tastenkombination läßt sich die Protokollfunktion wieder rückgängig machen. Die bei der DFÜ üblichen Codes wie <CTRL S> etc. können verwendet werden.

Empfangsparameter

Entsprechend den Druckerparametern kann man hier die Geräteadresse und Sekundäradresse des Gerätes festlegen, von dem Daten empfangen werden. Normalerweise wird dies die Geräteadresse 2 sein. In diesem Fall müssen auch die Übertragungsparameter noch berechnet werden (siehe Tabelle 3). Mit <RETURN> werden die Eingaben beendet.

Empfangstabelle

Um dem Benutzer die Arbeit zu erleichtern, sind alle Tabellen schon mit Standardwerten definiert, die beim Start des Programms automatisch geladen werden. Dennoch soll hier auf diese Tabellen eingegangen werden, da sich bei intensiver Nutzung durchaus die Notwendigkeit einer Umdefinition ergeben kann. In der Empfangstabelle wird jeder Zahl ein Buchstabe zugewiesen. Damit nun nicht 256 Zahlen auf dem Bildschirm stehen müssen, ist die ganze Tabelle in einer Matrix dargestellt. In den Zeilen befindet sich jeweils der höherwertige Teil der Zahl, entsprechend in den Spalten der niederwertige Teil. Will man nun einer Zahl einen Wert zuordnen, so bewegt man den Cursor mit <CRSR-abwärts> in die richtige Zeile und mit <CRSR-rechts> in die richtige Spalte. An der entsprechenden Stelle drückt man einfach die Taste, die dieser Zahl zugeordnet werden soll. Die RETURN-Taste wird mit <CTRL RETURN> eingetragen, da mit <RETURN> die Eingabe abgeschlossen wird. Steuerzeichen lassen sich natürlich nicht in dieser Tabelle eintragen. Trotzdem muß der Computer wissen, durch welche Codes Steuerzeichen eingeleitet werden können. Wenn es sich um Text im ESC/P-Format handelt, so werden Steuerzeichen in der Regel durch ESC (hexadezimal \$1B) eingeleitet. Tragen Sie an dieser Stelle in der Tabelle <CTRL β> ein. Auf dem Bildschirm wird dies als reverses Fragezeichen dargestellt.

Im Terminalmodus erreicht man mit der ALT-Taste die Löschfunktion für ein Zeichen. In der Regel wird diese Funktion durch den Code \$08 ausgelöst. Tragen Sie also an der Stelle \$08 in die Tabelle »ALT« ein durch Drücken der ALT-Taste.

Sendetabelle

Wie schon beim Punkt »Text senden« erwähnt, verwendet der Computer auch hier eine andere Tabelle als bei der Druckerausgabe. Mit diesem Menüpunkt wird die Tabelle definiert. Die Bedienung ist identisch mit der der Druckertabelle.

Steuerzeichen

Beim Datenempfang werden Steuerzeichen automatisch erkannt. Damit dies möglich ist, muß diese Tabelle ausgefüllt werden. Nur so können Steuerzeichen erkannt werden. Wenn der Computer beispielsweise die Sequenz, die hinter »-1« für Einschalten der Unterstreichung steht, empfängt, so wandelt er dies automatisch in ALT-1 um. Dadurch kann der Text später ohne Probleme mit MasterText 128 weiterverarbeitet werden. Voraussetzung für eine Erkennung ist allerdings auch, daß die Zahlen, mit denen Steuerzeichen eingeleitet werden, also jeweils die erste Zahl in jedem Steuerzeichen, in der Empfangstabelle mit < CTRL β > gekennzeichnet sind. Denn nur dann versucht der Computer, die Zeichen als Steuerzeichen zu interpretieren.

Eventuelle Änderungen bei diesen Tabellen und Parametern können, wie oben schon erwähnt, im Speicher-Menü unter einem eigenen Namen auf Diskette abgelegt werden. Wir wünschen Ihnen viel Vergnügen mit einem Textverarbeitungsprogramm der Spitzenklasse, das Ihnen sicher noch viele gute Dienste leisten wird.

Name	Startadresse	Länge in Blöcken
MASTERTXT/INST	\$1C01	9
MASTERTXT V1.0	\$1C01	9
MASTERTXT T1	\$4000	83
MASTERTXT T2	\$2000	4
MASTERTXT T3	\$0380	1
MASTERTXT T4	\$D000	40
ZEICHENSATZ	\$7000	17
FORMULAR	\$0400	2
DRUCKERTREIBER	\$BD00	3
MPS801(T)	\$2400	1
MPS802(T)	\$2400	3
CENTRONIC(T)	\$2400	1
TABULATOREN	\$0700	3

Tabelle 4. Die einzelnen Programmteile von MasterText 128 mit Blockanzahl und Startadresse.

(Martin Pahl/sk/nj)

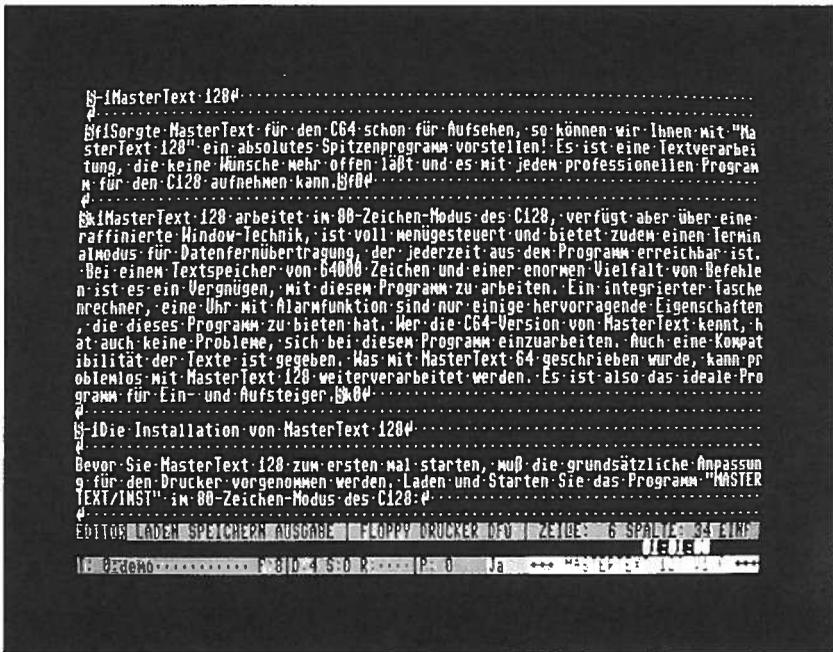


Bild 1. Der Editor von MasterText 128. Deutlich zu sehen sind die Statuszeilen.



Bild 2. Pro Menüpunkt wird ein Window eingeblendet.



Bild 4. Die Eingabemaske zum Setzen der Uhr- und Alarmzeit.

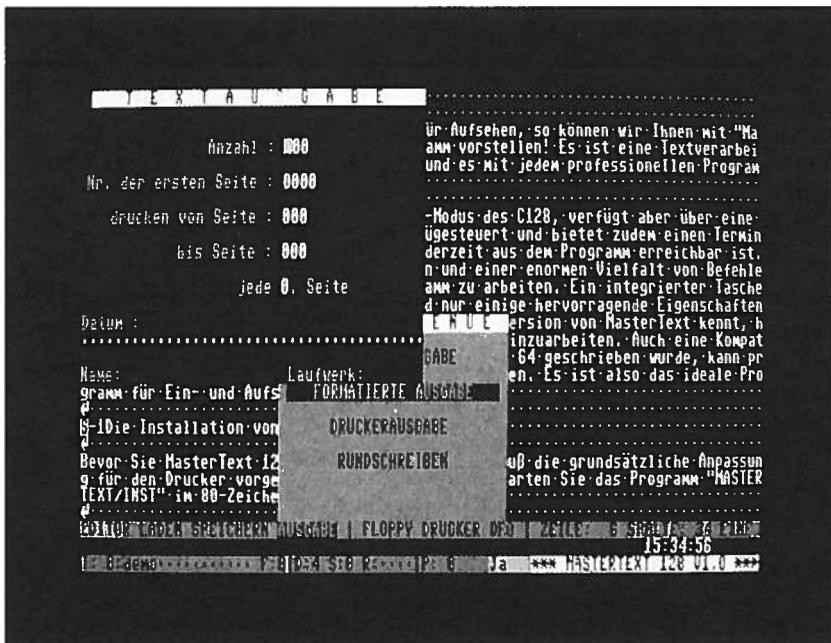


Bild 6. Die Bildschirmmaske für die Ausgabe-Funktion.

MasterText 128

Sorgte MasterText für den C64 schon für Aufsehen, so können wir Ihnen mit "MasterText 128" ein absolutes Spitzenprogramm vorstellen! Es ist eine Textverarbeitung, die keine Wünsche mehr offen läßt und es mit jedem professionellen Programm für den C128 aufnehmen kann.

MasterText 128 arbeitet in 80-Zeichen-Modus des C128, verfügt aber über eine raffinierte Window-Technik, ist voll menügesteuert und bietet zudem einen Terminalmodus für Datenfernübertragung, der jederzeit aus dem Programm erreichbar ist. Bei einem Textspeicher von 64000 Zeichen und einer enormen Vielfalt von Befehlen ist es ein Vergnügen, mit diesem Programm zu arbeiten. Ein integrierter Taschenrechner, eine Uhr mit Alarmpfunktion sind nur einige hervorragende Eigenschaften, die dieses Programm zu bieten hat. Nur die C64-Version von MasterText kennt, hat auch keine Probleme, sich bei diesem Programm einzuarbeiten. Auch eine Kompatibilität der Texte ist gegeben. Was mit MasterText 64 geschrieben wurde, kann problemlos mit MasterText 128 weiterverarbeitet werden. Es ist also das ideale Programm für Ein- und Aufsteiger.

Die Installation von MasterText 128

Bevor Sie MasterText 128 zum ersten mal starten, muß die grundsätzliche Anpassung für den Drucker vorgenommen werden. Laden und Starten Sie das



Bild 7. Formatierte Ausgabe auf dem Bildschirm. Unterstreichungen und Kursivschrift werden angezeigt.

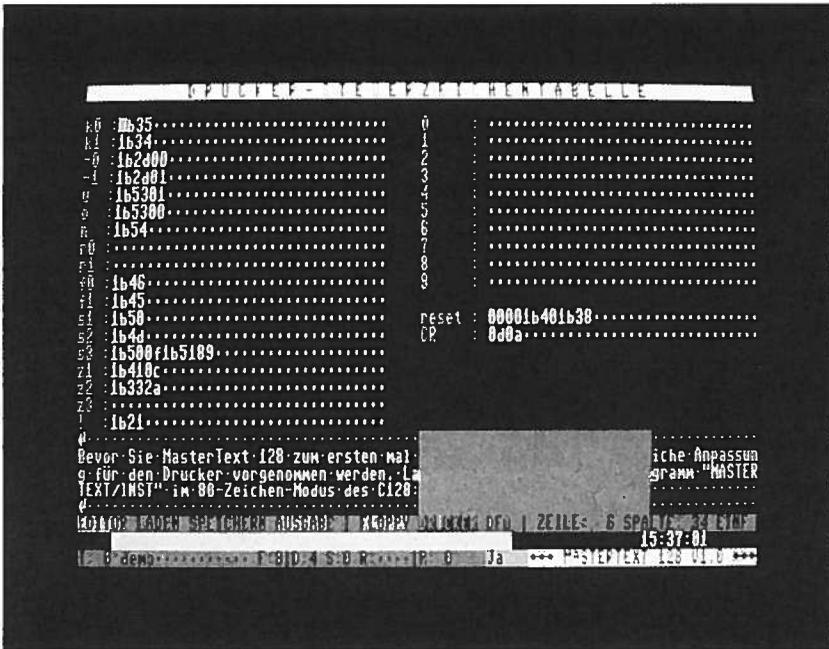


Bild 8. Die Steuerzeichen-Tabelle, eingestellt für einen Epson-Drucker.

Color-Pack 1

Jetzt können auch Basic-Programmierer den VDC-Chip vollkommen ausnutzen – eine neue Basic-Erweiterung läßt die normalen Grafikkommandos des C 128 auch im 80-Zeichen-Modus wirksam werden, wobei eine Grafikauflösung von 480 mal 240 Punkten erreicht wird.

Die komfortablen Grafikkommandos des C 128 gestatten die Erstellung von hochauflösenden Grafiken auf sehr bequeme Weise. Doch gilt dies nur für den 40-Zeichen-Bildschirm, der eine Grafikauflösung von immerhin 320 mal 200 Einzelpunkten zuläßt. Der zweite Bildschirm-Modus des C 128, die 80-Zeichen-Darstellung, wird von diesen Anweisungen allerdings nicht berührt, was um so unverständlicher ist, wenn man erfährt, daß sich hier Grafikbilder der doppelten Auflösung von 640 mal 200 entwerfen lassen.

Aus diesem Grund führen selbst einfache Grafikaktionen wie das Setzen und Löschen von Punkten in diesem Grafik-Modus zu einer mühsamen Auseinandersetzung mit dem VDC-Chip. Denn jener für die fantastische 80-Zeichen-Grafik verantwortliche Baustein ist kompliziert zu programmieren, wobei man ohne Assembler-Kenntnisse kaum auskommt.

Besitzer der Basic-Erweiterung »Color-Pack 1« kennen jedoch solche Probleme nicht, denn nach dem Booten dieses Programmes stehen alle bekannten Grafik-Befehle wie GRAPHIC, COLOR, CHAR, DRAW oder CIRCLE bereit, um solch eindrucksvolle Grafiken wie in Bild 9 auf den 80-Zeichen-Monitor zu zaubern. Doch Color-Pack 1 ist nicht nur »irgendeine« Grafik-Erweiterung. Einige Befehle wurden in ihrer Funktion modifiziert und verwöhnen den Anwender mit noch größerer Leistungsfähigkeit.

Die CHAR-Anweisung zur Darstellung von Text innerhalb einer Grafik dreht nun einen Text in jedem beliebigen Winkel, während zwei neue Parameter die Ausdehnung der Schrift in Höhe und Breite bestimmen. Schmal- oder Kleinschrift in jeder nur erdenklichen Richtung ist damit problemlos zu verwirklichen (Bild 10).

Mit dem SCALE-Befehl läßt sich endlich der Koordinatenbereich beliebig setzen, da das Color-Pack auch mit Gleitkommazahlen arbeitet. So darf die X-Achse zum Beispiel von $-2*\pi$ bis $3*\pi$ skaliert werden. Wer mathematische Funktionen auf dem Bildschirm darstellen will, wird diese Verbesserung zu schätzen wissen.

COLOR packt die Farben an

Eine besonders starke Wandlung mit angenehmen Folgen erfuhr der COLOR-Befehl des C 128. Mit seiner Hilfe lassen sich ähnlich dem Multicolor-Modus des 40-Zeichen-Bildschirms bis zu drei Farben für Vorder- und Hintergrund wählen, auf die die Grafikbefehle später schnell und einfach zugreifen können. Ein neuer COLOR-Modus gibt dem Anwender zusätzlich die Fähigkeit, Teile seiner Grafik nachträglich einzufärben, ohne dabei Gefahr zu laufen, ein bereits erstelltes Bild zu zerstören.

Zwei weitere Befehle des Basic 7.0 erhielten vollkommen neue Funktionen. GSHAPE und SSHAPE dienen nun zur Sicherung der Grafik auf Diskette oder in einen der beiden Bildbereiche, die jeweils eine 80-Zeichen-Grafik aufnehmen können. Eine modifizierte Belegung der Funktionstasten (Tabelle 5) erlaubt das Umschalten beider Grafikbereiche auf Knopfdruck.

Ungewöhnliches Format

Darüber hinaus präsentiert sich der Bildschirm mit Color-Pack 1 in einer neuen und zunächst ungewöhnlichen Form. Statt der üblichen Auflösung von 640 mal 200 Punkten erhält der Computer-Grafiker eine Arbeitsfläche von 480 mal 240 Grafikpunkten. Damit stehen anstelle von 128000 Pixel nurmehr 115200 ansprechbare Punkte zur Verfügung. Dieser scheinbare Nachteil erweist sich jedoch bald als recht vorteilhaft, da sich hierbei ein fast quadratisches Bild ergibt, das eine ausreichende Auflösung in Y-Richtung bietet. Der enorme Auflösungsunterschied von 640 zu 200 Punkten wird damit zum Großteil ausgeglichen. Daneben entspricht das neue Format in X-Richtung genau der erreichbaren Auflösung eines Epson- oder kompatiblen Druckers für DIN-A4-Formate, so daß ein Ausdruck ohne Verzerrungen erfolgen kann.

Selbstverständlich besitzt das Color-Pack für diesen Zweck eine Hardcopy-Routine, die den Ausdruck der Grafik auf einem Epson-Drucker und Kompatiblen in verschiedenen Größen gestattet. Der Druck erfolgt jedoch nur als Schwarzweiß-Bild. Unterschiedliche Farben werden dabei nicht (als Graustufen) berücksichtigt.

Starten von Color-Pack 1:

Bitte schalten Sie Ihren Monitor auf 80 Zeichen um, drücken die 40/80- und die ASCII/DIN-Taste und geben

`RUN "COLOR-PACK"`

ein.

Color-Pack 1 installiert sich nun im Speicher des C 128 und steht dann mit erstaunlichen Grafikbefehlen bereit, die wir im folgenden Abschnitt näher betrachten wollen. Wir werden hier allerdings nur jene Anweisungen erläutern, die einer Funktionsänderung unterworfen sind. Die verbleibenden Befehle werden wie bei der Arbeit mit der 40-Zeichen-Grafik verwendet. Die genaue Funktion können Sie dem Handbuch zum C 128 entnehmen.

Funktionsänderungen mit Folgen

GRAPHIC1, <Vordergrundfarbe>, <Hintergrundfarbe>

schaltet in den 80-Zeichen-Grafikmodus. Dabei dürfen die Werte (0–15) der Zeichen- und die Hintergrundfarbe des gesamten Bildschirms angegeben werden. Fallen diese Angaben weg, werden die voreingestellten Farbwerte übernommen. Die Grafik-Modi 2 bis 4 sind nicht mehr verwendbar und erzeugen einen »ILLEGAL QUANTITY ERROR«.

GRAPHIC 0 oder GRAPHIC

desaktiviert den Grafikmodus. Der C 128 kehrt in die Textdarstellung zurück.

GRAPHIC Modus, Rand

druckt eine Hardcopy der aktuellen Grafik auf einem Epson-Drucker und Kompatiblen. »Modus« bestimmt dabei die Größe des Druckbildes, während »Rand« die Wahl des Abstandes zum linken Papierrand erlaubt. Mögliche Kombinationen sind:

Befehl	Wirkung	Bereich von »Rand«
GRAPHIC 10, Rand GRAPHIC 11,0	kleines Bild doppelte Bildbreite (genaue Blattbreite DIN A4)	0 bis 68
GRAPHIC 12, Rand GRAPHIC 13,0	doppelte Bildhöhe großes Bild mit doppelter Bildbreite und -höhe	0 bis 68

Beispiel: GRAPHIC 10,34
Es wird ein kleines Bild genau in der Blattmitte gedruckt.

COLOR 1, Vordergrundfarbe, Hintergrundfarbe

COLOR 2, Vordergrundfarbe, Hintergrundfarbe

COLOR 3, Vordergrundfarbe, Hintergrundfarbe

Mit COLOR lassen sich drei Farbquellen für Vordergrund und Hintergrund unabhängig setzen, die in den Zeichenbefehlen durch Angabe der Farbquell-Nummer (1 bis 3) verwendet werden können.

Beispiele: BOX 2,10,10,100,100
Es wird ein Rechteck mit der Vordergrund- und Hintergrundfarbe aus Farbquelle 2 gezeichnet.

CIRCLE 3,120,100,50,25

Es wird ein Kreis mit den Farben aus Farbquelle 3 gezeichnet.

Die Angabe der Farbquelle darf auch weggelassen werden. In diesem Fall entspricht die Zeichen- und Hintergrundfarbe den im Farb-RAM befindlichen Farbwerten. Die Ausführung eines Befehls wird dadurch erheblich beschleunigt:

CIRCLE,120,100,50,25

COLOR 7, Vordergrundfarbe, Hintergrundfarbe, x1, y1, x2, y2

färbt das durch (x1,y1) und (x2,y2) vorgegebene Rechteck in der angegebenen Vorder- und Hintergrundfarbe, ohne die Grafik zu zerstören. Dabei stehen die Koordinaten (x1,y1) für die linke obere, (x2,y2) für die rechte untere Ecke des Rechtecks.

Die Farben können nicht pixelweise, sondern nur in Grafikblöcken zu je 8 mal 8 Punkten gesetzt werden, woraus sich für x1, x2, y1 und y2 folgende Bereichsgrenzen ergeben:

$0 \leq x1 \leq x2 \leq 59$

$0 \leq y1 \leq y2 \leq 29$

CHAR Farbquelle,x, y, Winkel, a\$, x-Breite, x-Höhe

schreibt einen beliebigen Text (a\$) mit der Farbe der gewünschten Farbquelle in den Grafikbildschirm an Position (x,y), und dreht ihn im angegebenen Winkel (0 bis 360 Grad). Auch hier führt ein Weglassen der Farbquelle zu einer deutlichen Beschleunigung der Ausführung. Da die »schrägen« Schriften aber nur schwer lesbar sind, empfiehlt es sich, nur die Winkel 0, 90, 180 und 270 Grad zu verwenden.

Die letzten beiden Parameter bestimmen die Breite und Höhe der Buchstaben. Sie dürfen Werte von 0 bis 255 annehmen und sind bei den Schreibrichtungen von 0 und 180 Grad von Bedeutung.

Bei Winkeln von 90 und 270 Grad, also bei senkrechter Schreibrichtung, sind zwei weitere Parameter wichtig:

CHAR Farbquelle, x, y, Winkel, a\$, y-Breite, y-Höhe

»y-Breite« und »y-Höhe« geben nun die Breite und Höhe der Buchstaben an. »x-Breite« und »x-Höhe« sind hier ohne Wirkung und können mit leeren Kommata übergangen werden. Ein kleines Beispiel im Programm CHAR-DEMO demonstriert die sich daraus ergebende Schriftvielfalt (Bild 10)

Beliebige Skalierung des Grafikbereichs

SCALE 3, xf, xs, yf, ys

ändert die Skalierung des Grafikbildschirms. Damit sind nun beliebige Definitionsbereiche in X- und Y-Richtung möglich. Die Parameter stellen jedoch nicht die jeweiligen Unter- und Obergrenzen für die X- und Y-Achse dar, sondern bestimmen die Skalierungsfaktoren und -summanden der Koordinatenachsen. Will man festgelegte Grenzen definieren, sind deshalb einige Berechnungen nötig:

```
xmin,xmax : Unter- und Obergrenze der X-Achse
ymin,ymax : Unter- und Obergrenze der Y-Achse
xf= (479-0)/(xmax-ymin)
xs= 0-xf*xmin
yf= (0-239)/(ymax-ymin)
ys= 239-yf*ymin
SCALE 3,xf,xs,yf,ys
```

Ein Beispiel finden Sie im File 'SCALE' das mit veränderter Skalierung eine Sinuskurve erstellt.

Die veränderte Skalierung wirkt nun auf die Befehle LOCATE und DRAW, wobei auch Fließkommawerte gestattet sind. Haben Sie beispielsweise die X- und Y-Achse jeweils von -10 bis 10 definiert, lassen sich nun mit DRAW Punkte oder Linien zeichnen, wie etwa:

```
DRAW 1,-5,-5
```

```
DRAW 2,-10,-5 TO -3.45,4.76
```

Die restlichen Grafik-Befehle beziehen sich jedoch nicht auf die veränderte Skalierung. Sie werden weiterhin im normalen Koordinatennetz gezeichnet.

LOCATE x,y

setzt den Grafik-Cursor im definierten Bereich an beliebige Koordinaten, wobei auf die Skalierung mit SCALE 3 Rücksicht genommen wird. ein Beispiel für die oben definierten Bereiche ist:

```
LOCATE -2.65,4.72
```

Von dem mit LOCATE angesteuerten Punkt aus kann dann normal gekennzeichnet werden.

LOCATE x, y, Variable

Mit dieser Variation von LOCATE kann getestet werden, ob an der Koordinate (x,y) ein Punkt gesetzt ist oder nicht. Bei einem gesetzten Punkt erhält die angegebene Variable den Wert 1, andernfalls bekommt die den Wert 0 zugewiesen.

SSHAPE Bildbereich

sichert das aktuelle Grafikbild in einem der beiden Bildbereiche (1 oder 2). So lassen sich zwei Grafikbilder gleichzeitig im Speicher des C 128 ablegen und bearbeiten.

SSHape Bildbereich, "Bildname"

speichert den angegebenen Bildbereich (1 oder 2) unter einem beliebigen Namen auf Diskette.

Grafiken sichern

GSHape Bildbereich

kopiert, als Pendant zu SSHape, die im angegebenen Bildbereich abgelegte Grafik zur weiteren Bearbeitung zurück in den aktuellen Grafikbildschirm.

GSHape Bildbereich "Bildname"

lädt eine Grafik des 80-Zeichen-Bildschirms in einen der beiden Bereiche (1 oder 2).

Taste	Belegung	Taste	Belegung
F1	löscht den Bildschirm unterhalb des Cursors	F5	Umschalten der beiden Grafikbilder
F2	DLOAD	F6	Listing auf Drucker ausgeben
F3	DIRECTORY	F7	LIST
F4	DSAVE	F8	MONITOR

Tabelle 5. Color-Pack 1 belegt die Funktionstasten neu

Damit wären alle Besonderheiten von Color-Pack 1 erläutert. Die verbleibenden Grafik-Anweisungen CIRCLE, BOX und PAINT sind wie gewohnt verwendbar, arbeiten nun aber wie alle anderen Grafik-Befehle im 80-Zeichen-Grafikmodus. Wurde das Color-Pack gestartet, ist die 40-Zeichen-Grafik nicht mehr aktivierbar. Man wird sie allerdings nicht vermissen.

(Rainer Franzen/Michael Thomas/nj)

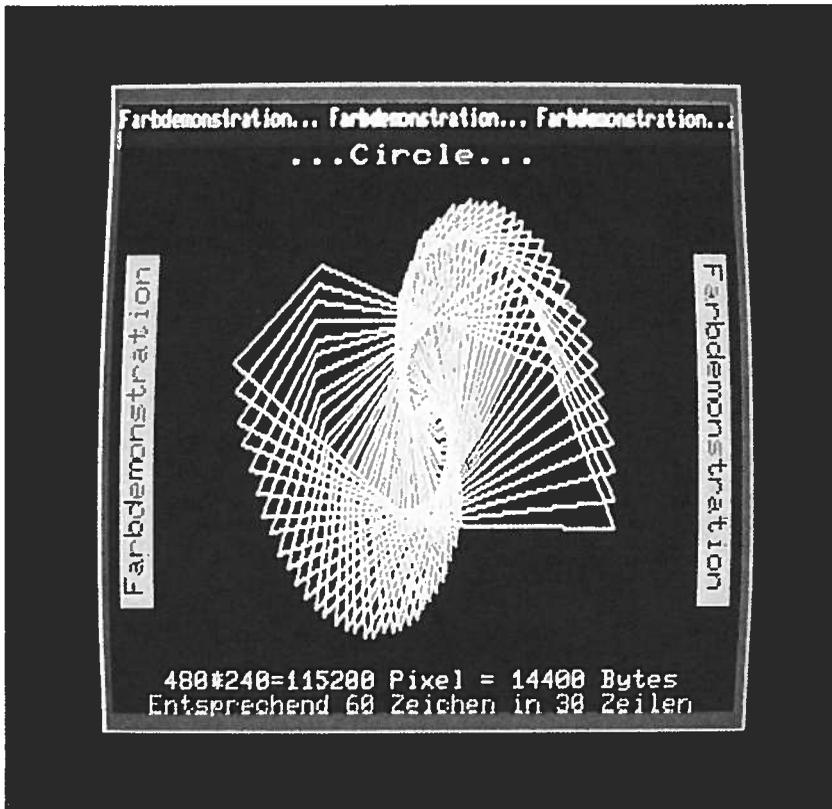


Bild 9. Eine eindrucksvolle Grafik im 80-Zeichen-Modus, die sich mit „Color-Pack 1« schnell erstellen läßt.



Bild 10. Eine nicht alltägliche Besonderheit: Schriften lassen sich in Größe und Lage beliebig verändern.

Top-Flop

Endlich gibt es einen Floppy-Monitor, der die zahlreichen speziellen Vorteile des C 128 voll ausnutzt und dabei sehr komfortabel und anwenderfreundlich zu bedienen ist.

Der größte Nachteil der vielen Floppy-Monitore für den C 64 ist wohl die begrenzte Zeichenzahl auf dem Bildschirm. Denn auf einer Matrix von 40 * 25 Zeichen läßt sich ein Sektor nicht komplett im Hex- und ASCII-Dump darstellen, und die Kompromißlösungen sind meistens unzufriedenstellend und gehen sehr auf Kosten der Benutzerfreundlichkeit. Beim C 128 ist dieses Problem jedoch nicht mehr gegeben, 80 Zeichen pro Zeile sind für diesen Zweck ideal. Bei Top-Flop gerät der 40-Zeichen-Bildschirm jedoch nicht in Vergessenheit – er wird für einen Hilfstext benutzt, der dem Anwender jederzeit durch Umschalten des Monitoradapters zur Verfügung steht.

Top-Flop ist in Basic geschrieben und wird durch mehrere Maschinenroutinen sinnvoll unterstützt. So fallen Unterprogramme, die sonst extrem zeitaufwendig sind (Floppy-Zugriffe, Darstellung des Sektors, Hardcopy etc.), kaum noch ins Gewicht. Jedoch dürfte es durch den modularen Aufbau trotzdem noch möglich sein, das Programm zu verstehen und eventuell nach eigenen Vorstellungen abzuändern.

Das Hauptprogramm lädt die Maschinenroutinen in den reservierten Grafikspeicher ab \$2200, was mehrere Vorteile hat:

Erstens nehmen sie auf diese Weise keinen anderen, vielleicht anderweitig benötigten Speicherplatz ein (das Programm benutzt ja keine Grafik), und zweitens ist so die Kompatibilität zu allen Basic-Compilern gesichert.

Starten des Programms

Das Programm wird mit

```
RUN "TOP-FLOP.BAS" <RETURN>
```

geladen und gestartet, wobei die Taste <40/80 DISPLAY > gedrückt sein sollte. Nun meldet sich Top-Flop mit der Angabe des Laufwerkstyps (»1570« auch bei angeschlossener 1571 mit

nicht zweiseitig formatierter Diskette). In der anschließenden kurzen Wartezeit wird das Programm initialisiert und das Unterprogramm »TOP-FLOP.SUB« nachgeladen. Sollte dieses auf der eingelegten Diskette nicht auffindbar sein, gibt Top-Flop eine Fehlermeldung aus. Nach dem Laden erscheint der noch leere Arbeitsbildschirm, und die Diskette im Laufwerk wird initialisiert (nur bei 1570/71 – zweimaliges Rattern im Laufwerk). Jetzt ist auch parallel auf dem 40-Zeichen-Bildschirm eine Liste aller gültigen Programmbefehle in alphabetischer Reihenfolge zu sehen (Bild 11). Der Arbeitsbildschirm teilt sich folgendermaßen auf: In der ersten Zeile stehen Programmname und -art. In der nächsten stehen die Informationen über den aktuellen Track und Sektor, den zuletzt gelesenen Track und Sektor (eine Information, die Einsteiger wie Fortgeschrittene gleichermaßen zu schätzen wissen) und der gerade aktive Anzeigemodus, über den Sie noch im Rahmen des Befehls »M« mehr erfahren werden (Bild 13). In dem folgenden abgegrenzten Feld wird nach jeder Diskettenoperation der Diskstatus im üblichen Format angezeigt. Das große Hauptfeld ist für die Darstellung des Sektors beziehungsweise für das Directory und das Floppybefehle-Untermenü reserviert. Am unteren Rand des Bildschirms ist noch der Autoren- und Copyright-Vermerk zu lesen, der aus Gründen der Übersichtlichkeit durch Invertierung dem Rahmen angepaßt wurde. Nach der Initialisierung des Programms ertönt ein »Gong«: Sie befinden sich im Befehlsmodus. Sie können nun einen beliebigen Top-Flop-Befehl eingeben. Diese Befehle bestehen aus einzelnen prägnanten Buchstaben, die das Merken des Kommandos erheblich vereinfachen. Sie können bei der Benutzung des Programms praktisch kaum etwas falsch machen, da unkorrekte Eingaben mit einem kurzen Brummtönen quittiert werden. Richtige Eingaben und Beendigungen von Routinen bestätigt Top-Flop mit dem bekannten Klingelzeichen CHR\$(7). Es ist also zu empfehlen, bei der Arbeit mit dem Programm immer den Lautsprecher angeschaltet zu haben.

<L> Lesen des eingestellten Sektors

Nach Drücken von <L> wird der eingestellte Sektor gelesen. Hierbei nützt Top-Flop den schnellen Burst-Modus der Commodore-Floppies 1570 und 1571. Ist der Schreib-Lese-Kopf des Laufwerks schon auf der richtigen Spur positioniert, so dauert die Übertragung nur noch einen Sekundenbruchteil und ist gerade durch ein kurzes Aufleuchten des roten Lämpchens

festzustellen. Aber auch Besitzer der 1541 brauchen sich nicht zu langweilen, dank Maschinensprache ist die konventionelle Übertragung immerhin auch noch recht schnell. Die nun folgenden Programmteile, die Ausgabe des Hex- und des ASCII-Dumps, sind wegen des relativ hohen Rechenaufwands ebenfalls in Maschinensprache gehalten und gehen schnell vonstatten. Alles zusammen dauert mit einer schnellen Floppy nur ungefähr 1 Sekunde. Das Ende des Lesevorgangs wird mit der Klingel bestätigt.

Beide Dumps (»dump« ist englisch und heißt soviel wie »Abfall-, Schutthaufen«) stellen in jeder der 16 Zeilen 16 Werte dar, die zur Steigerung der Übersichtlichkeit beide in der Mitte geteilt wurden. Aus dem gleichen Grund wurde auch auf die (seltsamerweise) sonst übliche Invertierung des ASCII-Dumps verzichtet. Die Darstellung im ASCII-Dump erfolgt gemäß dem eingestellten Anzeigemodus sowie nach einem weitverbreiteten Standard, es werden also keine Grafik- oder Steuerzeichen ausgegeben.

<S> Schreiben des eingestellten Sektors

Die Schreibroutine ist sehr einfach und daher kurz und schnell. Bei Eingabe des Befehls »S« werden die Daten sofort, wie sie am Bildschirm zu sehen sind, auf den eingestellten Sektor der Diskette geschrieben. Vorsicht! Dieses Kommando ist nicht mehr rückgängig zu machen. Anfängern ist also zu empfehlen, sich erst eine Sicherheitskopie der Arbeitsdiskette anzufertigen, da im schlimmsten Falle eine ganze Diskette auf diese Weise unbrauchbar gemacht werden kann. Auch ist nicht immer sicher, wie Programme auf Veränderungen mit dem Monitor reagieren; im Zweifelsfalle kann eine Hardcopy des ursprünglichen Sektors nützlich sein. Natürlich berücksichtigt die Routine aber auch den Schreibschutz auf einer Diskette, in diesem Falle wird ein »WRITE PROTECT ERROR« ausgegeben und der Schreibversuch abgebrochen.

Nach dem Klingelzeichen befindet sich das Programm wieder im Befehlsmodus.

<E> Angezeigten Dump editieren

Der Befehl »E« für Editieren ist sehr leistungsfähig. Mit ihm ist es möglich, den dargestellten Sektor beliebig zu verändern. Der Cursor erscheint in der linken oberen Ecke des Hex-Dumps

und kann mit Hilfe der Cursor-Tasten beliebig über die beiden Dumps bewegt werden. Das Verlassen dieses Bereichs ist nicht möglich. Um eine hexadezimale Zahl zu ändern, überschreiben Sie sie einfach. Dabei müssen allerdings immer beide Ziffern geändert werden, das Editieren nur einer der beiden Ziffern wird vom Programm nicht akzeptiert. Außerdem nimmt Top-Flop im Bereich des Hex-Dumps nur die Tasten <0> bis <9> beziehungsweise <A> bis <F> an, wodurch wiederum Fehleingaben ausgeschlossen sind. Wurde ein Wert neu eingegeben, so nimmt das Programm automatisch die entsprechende Korrektur im ASCII-Dump vor, wobei dieselben Regeln für die Darstellung gelten wie beim Lesen. Der Cursor springt anschließend auf die erste Ziffer des nächsten Wertes (auch über die Spalte in der Mitte) beziehungsweise nach dem letzten Byte der Zeile an den Anfang der nächsten (in der letzten Zeile an den Anfang derselben). Da eine korrekte Eingabe mit einem Klingelzeichen bestätigt wird, können Sie ohne Probleme auch längere Wertelisten eingeben, ohne ständig den Bildschirm kontrollieren zu müssen.

Das Editieren des ASCII-Dumps läuft ganz entsprechend ab: Das Byte unter dem Cursor wird durch Überschreiben mit einem der zulässigen Zeichen (dieselben, die auch dargestellt werden können) direkt verändert und der entsprechende hexadezimale Wert sofort angeglichen. Bei einer unzulässigen Eingabe bleibt das Byte so lange auf dem ursprünglichen Wert, bis eine korrekte Taste gedrückt wird. Auch hier ist kontinuierliches Schreiben ohne weiteres möglich, da der Cursor automatisch die nächste logische Position einnimmt. Vergessen Sie aber bei allem Komfort nicht, daß die Taste <INST/DEL> keine Funktion mehr hat. Trotzdem dürfte es Ihnen nicht schwerfallen, auch längere Texte ohne große Umstände zu schreiben. Der Editiermodus wird mit <RETURN> beendet, worauf natürlich die Klingel ertönt, der Cursor verschwindet und der nächste Befehl eingegeben werden kann.

Übrigens werden alle Veränderungen im Editiermodus nur am Bildschirm und im Speicher durchgeführt, jedoch nicht auf der Diskette. Sie kann also mit <L> noch rückgängig gemacht werden, erst nach <S> ist die Veränderung endgültig.

<C> Eingabe der neuen Sektornummer

Mit Hilfe dieses Kommandos können Sie die aktuelle Sektornummer ändern. Nach der Eingabe von <C> (englische Schreibweise: Sector) wird die bisherige Nummer gelöscht und weiterhin in der Infozeile bei »LAST TR/SE« angezeigt. Nun sind Sie aufgefordert, die gewünschte Nummer einzugeben. Dies erfolgt übrigens in zweistellig dezimaler Form, das heißt, es werden nur Zifferntasten angenommen, und einstellige Zahlen müssen mit einer Null erweitert werden (zum Beispiel 05 statt 5). Dadurch sparen Sie sich die Bestätigung mit <RETURN> und somit eventuell einen dritten Tastendruck. Auch hier löscht Top-Flop unkorrekte Eingaben und quittiert mit einem Warnton. Bitte beachten Sie auch die unterschiedliche Anzahl von Sektoren auf den verschiedenen Tracks, die übrigens ebenfalls vom Programm berücksichtigt wird.

<T> Eingabe der neuen Track- und Sektornummer

Der Befehl <T> entspricht <C> mit dem Unterschied, daß hiermit auch der aktuelle Track eingegeben werden kann. Das Format ist wieder dezimal zweistellig. Sofort nach Aufruf der Routine kopiert das Programm die bisherige aktuelle Adresse an die Stelle »LAST TR/SE«. Sie können nun also jede beliebige Adresse auf der Diskette ansprechen. Im 1541- und 1570-Modus ist die höchste zulässige Tracknummer 35, bei Verwendung beider Seiten im 1571-Modus beträgt sie 70.

Auf den eingestellten aktuellen Track und Sektor beziehen sich übrigens nur die Routinen »Lesen« und »Schreiben«, »Hardcopy« und »Editieren« greifen verständlicherweise auf den gerade dargestellten Sektor zurück.

<W> Diskette im Laufwerk initialisieren

Die Aufforderung für das Programm, die Diskette im Laufwerk zu initialisieren, ist eigentlich nur für den 1570- und 1571-Modus wichtig. Denn bevor auf eine Diskette mit Hilfe der Burst-Befehle zugegriffen werden kann, muß ihre ID eingelesen werden, ansonsten ist der Befehl nicht ausführbar und führt zu einem »DISK ID MISMATCH«-Fehler. Die Initialisierung wird

am Anfang vom Programm automatisch durchgeführt, Sie müssen <W> also nur noch nach jedem Diskettenwechsel eingeben. Solange Sie dann nur mit einer Diskette arbeiten, ist <W> nicht mehr nötig.

<N> Nächsten Sektor der Datei laden

<N> ist wahrscheinlich das am häufigsten benötigte Kommando. Die Routine liest den nächsten Sektor der Datei, dessen Position auf der Diskette ja in den ersten beiden Bytes des letzten Sektors steht, und liest diesen ein. Dabei wird die Adresse des alten Sektors wieder in der Infozeile angezeigt. Top-Flop erkennt auch hier Sektorverkettings, die das Ende von Dateien anzeigen (1. Byte: 00, zweites Byte: Zahl der noch benutzten Byte), und lädt in einem solchen Fall Sektor 1 auf Track 18, also den ersten Block des Inhaltsverzeichnisses nach. Die Ausgabe entspricht dann der von <L>. Mit <N> ist es theoretisch möglich, eine Datei von 100 Blöcken Länge mit einer 1570/71 innerhalb von zirka 1 Minute und 40 Sekunden komplett durchzublätern. Diese Geschwindigkeit ist ein Beispiel dafür, wie gut eingesetzte Maschinenprogramme auch innerhalb von Basicprogrammen für sehr hohes Tempo sorgen können.

<M> Aktuellen Anzeigemodus ändern

Die Möglichkeit, zwischen vier verschiedenen Anzeigemodi zu unterscheiden, ist wohl eine der komfortabelsten und auch leistungsfähigsten Optionen. Hinzu kommt, daß diese doch sehr praktische Differenzierung in keinem mir bekannten Programm dieser Art angeboten wird. Über den Befehl <M> können Sie grundsätzlich festlegen, welche Bytewerte als welche Zeichen interpretiert und dargestellt werden. Dabei sind folgende Möglichkeiten angeboten:

1. »GA« »GA« steht für Grafikzeichensatz im ASCII-Format, das heißt, die Werte von 32 bis 95 werden mit Großbuchstaben dargestellt. Da dies den üblichen Standard bildet, wurde es als Voreinstellung gewählt.

2. »TA« Nun kommen für den Textzeichensatz die Werte 97 bis 122 als Großbuchstaben hinzu, 65 bis 90 werden klein angezeigt. Das ASCII-Format wird vor allem zur Speicherung von Text in Basicprogrammen für die Ausgabe mit PRINT und zur Angabe der Dateinamen im Directory verwendet.
3. »GB« Im Bildschirm-Code-Modus legt der Computer die Texte im Bildschirmspeicher ab. Im Bildschirm-Code belegen die einfachen Buchstaben die Werte 1 bis 26, die Sonderzeichen sind dem ASCII-Code identisch.
4. »TB« Zusätzlich werden hier noch die Werte 65 bis 90 als Großbuchstaben entsprechend TA dargestellt.
Der Bildschirmcode wird oft von Maschinenprogrammen benützt, die die Texte direkt in den Bildschirmspeicher übertragen.
Zwischen diesen vier Möglichkeiten können Sie nach Eingabe von <M> mit Hilfe der Cursortasten wählen, wobei <CURSOR abwärts> und <CURSOR rechts> beziehungsweise <CURSOR aufwärts> und <CURSOR links> die gleiche Wirkung haben. Bestätigen Sie die gewünschte Einstellung mit <RETURN>. Die Einstellung ist sowohl für das Lesen als auch für das Editieren eines Blocks wirksam.

<H> Hardcopy des angezeigten Dumps

Das Erstellen einer Hardcopy des kompletten Dumps dürfte für Druckerbesitzer eine sehr nützliche Option sein, denn oft erleichtert es die Arbeit ungemein, wenn mehrere Sektoren direkt verglichen werden können. Achten Sie darauf, daß bei der Eingabe von <H> der Drucker eingeschaltet und die richtige Diskette im Laufwerk ist. Andernfalls wird die Routine abgebrochen. Das Programm lädt nämlich den Namen der Diskette nach und gibt ihn zusammen mit der passenden Sektoradresse, der letzten Sektoradresse, dem aktiven Anzeigemodus sowie last not least mit dem kompletten Dump des Sektors aus (Bild 12). Um die Ausgabe zu möglichst allen Druckern kompatibel zu halten, gibt das Programm auch Buch-

staben, die am Bildschirm klein geschrieben sind, als Großbuchstaben aus. Der ganze Ausdruck wird mit zwei gestrichelten Linien oben und unten begrenzt. Das Drucken selber benötigt auf einem MPS 803 nicht mehr als 35 Sekunden, wiederum dank Maschinensprache. Die Standardeinstellung des Programms schickt die Hardcopy an die Primäradresse 4, erfahrenen Programmierern dürfte es aber keine Schwierigkeiten bereiten, die Adresse im Basic- und im Maschinenprogramm ab \$233A entsprechend abzuändern.

<I> Suchen nach GEOS-Icons oder Sprites

Nach dem Aufruf dieser Funktion durch <I> erscheint am unteren Maskenrand die Aufforderung, auf den 40-Zeichen-Bildschirm umzuschalten. Dort werden die aktuelle Sektoradresse, das Basis-Byte, ab dem die Sprite-Information aus dem Sektor übernommen wird sowie das Sprite selber in doppelter Größe ausgegeben. Nun ist es möglich, das Basis-Byte mit <+> und <-> jeweils um 1 zu erhöhen beziehungsweise zu erniedrigen. Um die Suche innerhalb eines Sektors zu beschleunigen, verändern die jeweiligen Tasten zusammen mit <SHIFT> die Basis in Zehnerschritten. Nach <RETURN> stellt Top-Flop wieder den Hilfstext zur Verfügung, löscht die Aufforderung zum Umschalten und befindet sich im Normalmodus. Mit Hilfe dieser Funktion können Sie sehr leicht Spriteinformationen aus fremden Programmen herausuchen und mit Hilfe von <Y> (siehe weiter unten) für eigene Verwendungen umkopieren.

<D> Ausgabe des Directory

Mit <D> wird im Arbeitsfeld das Directory der im Laufwerk liegenden Diskette ausgegeben. Dazu wird der Hexdump – allerdings nur vorübergehend – gelöscht, so daß Sie zum Beispiel nach der Anzeige eines Directory-Blockes direkt mit dem Directory vergleichen können (Bild 13). Der Einfachheit halber wurde im Programm der Basic-7.0-Befehl DIRECTORY beziehungsweise CATALOG verwendet, somit dürfte das Format keine weiteren Erklärungen nötig machen. Sollte die gesamte Ausgabe mehr als 16 Zeilen lang sein, so scrollt die Liste innerhalb des Fensters. Sie können die Anzeige aber jederzeit mit <NO SCROLL> anhalten. Wenn Sie nach der Ausgabe der Anzahl der freien Blöcke eine beliebige Taste drücken, so wird intern die Leseroutine aufgerufen und der gesamte Dump nochmals dargestellt.

<F> Das Floppy-Menü

Über <F> gelangt man in ein kleines Menü mit Floppy-Befehlen, das wie das Directory auch am Platz des nun gelöschten Hex-Dump ausgegeben wird. Sie finden dort die folgenden Menüpunkte:

<S> Scratchen (Löschen) einer Datei

Nach der Eingabe des Dateinamens erfolgt eine Sicherheitsabfrage, die Sie mit <RETURN> (Voreinstellung <J> = Ausführung des Befehls) oder mit jeder anderen Taste und <RETURN> (= Rückkehr zum Menü) beantworten können.

<H> Header Disk (Diskette neu formatieren)

Zusätzlich zum Disknamen können Sie auch eingeben, ob die Diskette mit ID (Löschen aller Sektoren) oder ohne ID (nur Löschen des Inhaltsverzeichnisses) formatiert werden soll. Die ID können Sie aufgrund eines Fehlers im Basic 7.0 beim HEADER-Befehl nur mit einer Änderung direkt im Programm bestimmen. Auch hier werden Sie durch eine Sicherheitsabfrage wie bei <S> vor einer vorschnellen Fehlentscheidung bewahrt.

<V> Validate Disk (BAM neu ordnen)

Dieses Kommando wird direkt ausgeführt. Da die Floppy währenddessen nicht nutzbar ist, gibt das Programm die Meldung »BITTE WARTEN...« aus und hält den Programmablauf an, bis die Floppy wieder normal angesprochen werden kann. Nach der Ausführung jedes der drei Befehle wird der Diskstatus unter dem Text (also nicht wie normal in der speziellen Zeile) ausgegeben, was gleichzeitig das Ende der Routine und die Aufforderung zum Tastendruck signalisiert. Danach erscheint wieder das Menü.

<Z> Fenster löschen

Das Programm löscht das aktuelle Fenster und lädt den Sektor wieder nach, anschließend befindet es sich im Normalmodus.

<X> Reset oder Neustart des Programms

<X> dient zum Verlassen oder Neuinitialisieren des Programms. Bestätigen Sie die Sicherheitsabfrage nach dem Löschen des Bildschirms mit <RETURN>, so führt der Computer einen Reset aus. Beachten Sie, daß dabei eventuell von der Diskette gebootet wird, falls der Boot-Sektor beschrieben ist. Bei einer anderen Taste als <J> und <RETURN> wird das Programm über RUN neu gestartet, wonach Sie wieder auf alle Voreinstellungen zurückgreifen können.

<Y> Aufruf des TEDMON oder Neustart des Programms

Der Befehl »Y« ähnelt dem »X«-Kommando, jedoch springt das Programm bei Bestätigung der Sicherheitsabfrage in den TEDMON und zeigt automatisch mit dem TEDMON-Befehl »M« den Speicherbereich von \$2000 bis \$20FF, in dem Top-Flop den aktuellen Sektor abgelegt hat. Sie können nun beispielsweise den Bereich disassemblieren, um Maschinenprogrammen auf die Spur zu kommen oder nach einer bestimmten Bytefolge automatisch suchen zu lassen. Der editierte Sektor kann dann folgendermaßen gespeichert werden:

```
<X> <RETURN>  
LET TR=gewünschte Tracknummer <RETURN>  
LET SE=gewünschte Sektornummer <RETURN>  
GOTO 1430 <RETURN>  
<Y> <RETURN>
```

Nun befinden Sie sich wieder wie vorhin im Monitor.

Problemlose Erweiterung des Programms

Sie können das Programm natürlich dank des modularen Aufbaus ohne weiteres um eigene Routinen erweitern, denn natürlich hat jeder Anwender seine persönlichen Bedürfnisse. So könnte etwa der Einbau eines schnellen Kopierprogramms wie Tornado-Copy eine durchaus sinnvolle Erweiterung für Top-Flop sein.

(Nikolaus Huber/sk)

Der Aufbau von Top-Flop im einzelnen:

10-80	Startbildschirm
90-150	Erkennung des Floppy- und Diskettentyps
160-240	Maschinenroutinen laden
250-260	Hilfsbildschirm aufbauen
270-380	Variablen initialisieren
420-650	Bildschirmmaske aufbauen
660	bei 1570/71 Disk initialisieren
700-890	Hauptmenü
980-1110	Parameter für Burst-Leseroutine
1120	Burst-Leseroutine aufrufen
1150-1170	1541-Leseroutine
1220	Aufruf Hexdump
1270-1290	Dump in ASCII-Code
1310-1330	Dump in Bildschirmcode
1490-1600	Parameter für Burst-Schreibroutine
1610	Aufruf Burst-Schreibroutine
1640-1660	1541-Schreibroutine
1740-1770	Ausgabe Diskstatus
1820-1850	aktuelle Adresse nach letzter Adresse
1860-1930	Adresse aus Sektor holen und setzen
1940	Verzweigung zur Leseroutine
2010-2090	Diskname lesen
2110-2190	Infokopf für Hardcopy schreiben
2210	Aufruf Hardcopy-Routine
2220-2270	untere gestrichelte Linie schreiben
2350-2410	Cursor auf linke obere Position setzen
2440-2470	Beenden des Editierens
2490-2500	Auswahl Hex-/ASCII-Dump
2520-2560	neue Cursor-Position
2570	Cursor ausgeben
2580	bei ungültiger Eingabeposition zurück
2590	zur Eingabe zweier Hex-Ziffern

Der Aufbau von Top-Flop im einzelnen

2600–2620	Umwandlung in Hexzahl
2630	Änderung im abgespeicherten Sektor
2650	neue Ausgabe des ASCII-Dumps
2660–2790	Eingabe zweier Hex-Ziffern und Ausgabe
2800	zurück, wenn Cursor auf Spalte
2840–2880	bei ungültiger Eingabe Wiederholung
2890–2920	bei GA ohne Großbuchstaben
2930–2970	bei ASCII direkt im Dump ausgeben
2980–3050	bei BC Umwandlung und Ausgabe
3060–3140	neue Cursor-Position im Hexdump
3150–3190	neue Cursor-Position im ASCII-Dump
3250–3280	aktuelle Adresse nach letzter Adresse
3290–3350	Eingabe zweistelliger Dezimalzahl
3360–3370	auf zulässige Tracknummer prüfen
3430–3470	aktuelle Adresse nach alter Adresse
3480–3540	Eingabe zweistellige Dezimalzahl
3550–3580	auf zulässige Sektornummer prüfen
3680	Anzeigemodus invertieren
3690–3760	mit Cursor-Tasten ändern
3770	Anzeigemodus normal darstellen
3860	Hexdump löschen
3870	Window für Directory definieren
3880	Directory ausgeben
3930–3940	Directory löschen und Windows aufheben
3950	Sektor neu lesen und darstellen
3990–4210	Hilfstext auf Bildschirm 0 ausgeben
4250–4310	Reset oder Neustart
4370	Hexdump löschen
4380–4420	Hauptmenü ausgeben
4430–4480	Sprungtabelle
4490–4570	Datei löschen
4580–4670	Diskette formatieren

Der Aufbau von Top-Flop im einzelnen	
4680-4730	BAM neu ordnen
4740-4760	Bereich löschen und Windows aufheben
4770	Sektor neu lesen und darstellen
4880-4920	TEDMON-Befehl für Dump in Tastaturpuffer
4940-4950	Tastaturpuffer setzen
4960	Aufruf Monitor
5010-5030	Vermerk auf Bildschirm 5
5040-5140	Ausgabe Maske auf gelöschten Bildschirm 0
5150-5160	Sprite im Kasten anzeigen
5170	Spritedaten aus Sektor umkopieren
5180-5320	Auswertung der Eingabe
5200-5260	bei Ende Standardmasken wiederherstellen
5330-5340	Ausgabe Basisbyte
5360-5400	Editier-Unterroutine VDC-Position setzen
5460-5510	Parameter für Burst-Disk-Initialisierung
5520	Routine Burst-Befehl senden
5530-5600	Parameter für Burst-Lesen setzen
5610	Burst-Lesen aufrufen

Tabelle 6. Struktur des Programms »Top-Flop«

Interessante Programmiertricks in Top-Flop

- Nutzung des durch GRAPHIC 1,1 reservierten Grafikspeichers zur unproblematischen Speicherung von Daten und Maschinenprogrammen
- Ausgabe eines Hilfstextes auf dem 40-Zeichen-Bildschirm bei einem Programm mit 80-Zeichen-Darstellung
- Erkennung des Floppytyps durch Abfrage der Speicherstelle \$8000 im Floppy-speicher (Zeile 90-140)
- Ausgabe einer Zahl im Hexadezimal-Format mit folgendem Leerzeichen an der aktuellen Cursor-Position durch `SYS DEC("B8C2")`, Zahl (Zeile 630)
- Löschen der Zeile ab der aktuellen Cursorposition durch `SYS DEC("C01E")`, 81 (Zeile 1750)
- Aufheben des definierten Windows durch `SYS DEC("CA24")` (Zeile 3940, 4760)
- Übergabe von Befehlen über das Programmende hinaus im Tastaturpuffer (Zeile 4890-4950)

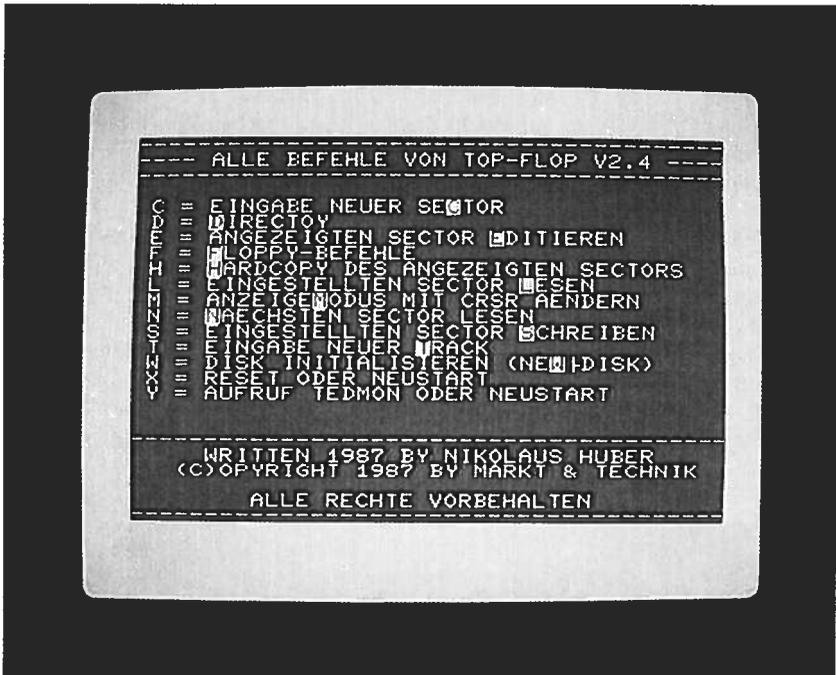


Bild 11. Übersichtlicher Hilfstext auf dem 40-Zeichen-Bildschirm (alle gültigen Programmbefehle).

TOP - FLOP 1571 FLOPPY-MONITOR																	
TRACK 18		SECT R 4		LETZTER T/S 18 1		ANZEIGEModus GA											
IISF STATUS 00 OK 00 00																	
00	12	07	02	00	01	4D	50	53	38	30	31	20	20	20	20IPS 001	
10	20	20	20	20	44	00	00	00	00	00	00	00	00	00	03	D.....	
20	00	00	02	00	03	41	50	53	38	30	32	20	20	20	20NPS 002	
30	26	28	26	20	44	00	00	00	00	00	00	00	00	00	03CEI TPONIC M	
40	00	00	02	00	07	43	45	4E	54	52	4F	4E	49	43	2F	4D	LF D.....
50	4C	46	20	20	44	00	00	00	00	00	00	00	00	00	03	00	LF D.....
60	00	00	02	00	0C	43	45	4E	54	52	4F	4E	49	43	2F	4FCEN TRC TO 0
70	4C	46	20	20	44	00	00	00	00	00	00	00	00	00	03	00	LF D.....
80	00	00	02	00	13	45	50	53	4F	4E	31	2F	41	4C	46	20EPS 013/MLF
90	20	20	20	20	44	00	00	00	00	00	00	00	00	00	03	00	D.....
00	00	00	02	00	00	45	50	53	4F	4E	31	2F	4F	4C	46	20EPS 013/MLF
10	20	20	20	20	44	00	00	00	00	00	00	00	00	00	03	00	I.....
20	00	00	02	00	01	44	52	55	43	40	45	52	20	20	20	20DRU CKER
30	20	20	20	20	44	00	00	00	00	00	00	00	00	00	03	00	D.....
40	00	00	02	00	03	45	50	53	4F	4E	34	2F	4F	4C	46	20EPS 013/MLF
50	20	20	20	20	44	00	00	00	00	00	00	00	00	00	03	00	D.....

WRITTEN 1987 BY NIKOLAUS HUBER / (C) 1987 BY MARKI & TECHNIK

Bild 12. Der Standardbildschirm von Top-Flop.

```

TOP - FLOP      1571 FLOPPY-MONITOR
TRACK 18 SECTOR: 4 LETZTER 1/S: 18/1 ANZEIGEMODUS: GA
DISK STATUS: 00, OK,00,00
00      0  "-----" DEL      ....MPS 801
10      16 "CP/M FORMATTER" PRG      D.....
20      5  "MFM-SCAN" PRG      ....MPS 802
30      0  "-----" DEL      D.....
40      27 "UNIBOOT" PRG      ....CEN TRONIC/M
50      0  "-----" DEL      LF D.....
60      1  "MHC128" PRG      ....CEN TRONIC/20
70      0  "-----" DEL      LF D.....
80      23 "VECTORS.BOOT" PRG     ....EPS 0N1/MLF
90      1  "VECTORS.OBJ" PRG     D.....
00      0  "-----" DEL      ....EPS 0N1/OLF
10      1  DIRECTORY PRG      D.....
20      1  TYP SE      ....DRU CAEP
30      6  DEMO PPG     D.....
40      31 BLOCPS FREE. ....EPS 0N4 OLF
50      0  "-----" DEL      D.....
60
70
80
90

```

WRITTEN 1987 BY NIKOLAUS HUBER / (C) 1987 BY MARKT & TECHNIK

Bild 13. Ausgabe des Directorys.

Double-Ass

Darauf haben viele gewartet: Ein Assembler für beide Prozessoren des C 128 mit jeder Menge nützlicher Funktionen, der bis zu 58 Kbyte Quellcode im Speicher verarbeitet.

Mit »Double-Ass« erhalten Sie einen leistungsfähigen 2-Pass-Assembler, mit dem Sie Maschinencode für den Z80- und den 8502-Prozessor (CPU) generieren können. In Tabelle 7 finden Sie einen Überblick über die wichtigsten Eigenschaften von Double-Ass.

Der Assembler ist vollständig in Maschinsprache geschrieben und belegt zirka 9 Kbyte in Bank 1 des Commodore 128 und nimmt somit keinen Platz des Basic-Programmspeichers ein. Allerdings werden etwa 48 Kbyte des Variablenspeichers vom Assembler für Labels reserviert. Dieser Speicherbereich wird aber beim Arbeiten mit dem Assembler ohnehin nicht anderweitig benötigt.

Bitte starten sie Double-Ass folgendermaßen:

```
DLOAD "DOUBLE-ASS"  
RUN
```

Mit einem Boot-Sektor-Programm kann der Boot-Sektor der Diskette so bearbeitet werden, daß der Assembler nach Einschalten des Rechners automatisch gestartet wird.

Nach einem Reset wird der Assembler durch Eingabe von

```
BANK 1: SYS 5376 neu initialisiert.
```

Der Assembliervorgang selbst wird – wie in Basic – mit »RUN« gestartet, wenn in der ersten Zeile der Befehl »* =« (siehe unten) zusammen mit der Adresse, ab der das Programm generiert werden soll, steht.

Double-Ass fest im Speicher integrieren

Es besteht die Möglichkeit, den Assembler in ein 16K-EPROM (27128) zu brennen und in den freien Steckplatz (U36) des C 128 zu stecken. Dies empfiehlt sich besonders bei häufigem Arbeiten mit dem Assembler, aber auch für gelegentliches Benutzen, da so alle anderen

Programme völlig unbeeinflusst laufen. Die Kompatibilität bleibt voll erhalten. Um den Assembler auf den Steckplatz U36 zu bannen, benötigt man lediglich ein 16K-EPROM und einen EPROM-Brenner. Das Programm »Double-Ass« kann ohne Änderungen auf das EPROM gebrannt werden.

Schalten Sie bitte anschließend Ihren Computer aus, schrauben ihn auf und stecken Sie das EPROM in den freien Steckplatz U36. Beachten Sie dabei unbedingt die Einbaurichtung (die Kerbe am EPROM muß der Kerbe am Sockel entsprechen). Nach dem Einstecken in U36 des C 128 kann der Assembler durch Eingabe von

BANK 4: SYS 58449

initialisiert werden.

So bedient man den Assembler

In das Betriebssystem wurden zwei zusätzliche Editorfunktionen eingefügt: die Routinen »FIND« und »MERGE«. »FIND« ermöglicht dem Benutzer, Zeichensequenzen innerhalb eines Quelltextes zu finden. Der Befehl hat folgendes Format:

FIND (Zeichenkette) Nach dem Befehlswort folgt ohne Leer- oder Anführungszeichen die gesuchte Zeichenfolge.

Mit Hilfe des MERGE-Befehls können verschiedene Quellprogramme aneinandergesetzt werden. Er hat folgende Syntax:

MERGE "Dateiname",GN, wobei »GN« die Geräteadresse ist.

Außerdem wurde ein zusätzlicher Basic-Befehl eingebaut, der es ermöglicht, den Assemblierungsvorgang von einem Programm aus zu starten. Er lautet

»*=« und wird, gefolgt von der Adresse, ab der das Maschinenprogramm generiert werden soll, eingegeben.

Wie gibt man Source-Code ein ?

Ein Maschinenprogramm wird – wie ein Basic-Programm – mit Zeilennummern eingegeben, wobei auch sämtliche Editierfunktionen des Basic-Editors zur Verfügung stehen. Es ist auch möglich, mehrere Befehle durch Doppelpunkt getrennt in eine Zeile zu schreiben, wodurch sich die Übersichtlichkeit der Programme wesentlich erhöhen läßt.

Der Assembler erkennt außerdem, ob der 40- oder 80-Zeichen-Modus aktiviert ist. Die Ausgabe von Symboltabellen etc. wird entsprechend angepaßt.

Unser Assembler besitzt eine Vielzahl von Pseudo-Opcodes. Das sind Befehle, die dem Assembler zusätzliche Anweisungen für den Assemblierungsablauf, für die Ausgabe von Listings, das Nachladen von Symboltabellen usw. geben.

Zunächst sollen einige Begriffe, die im Zusammenhang mit den Pseudo-Opcodes Verwendung finden, erklärt werden.

LABEL:

Ein Label ist eine Zeichenkette, die für eine Adresse oder einen Wert steht. Das erste Zeichen des Labels muß ein Buchstabe sein. Die folgenden Zeichen können Buchstaben oder Zahlen sein. Beispiel:

```
VIC001 = $D000
```

Bei einem Label sind die ersten 40 Zeichen relevant (in Basic sind es nur die ersten beiden). Wichtig: Ein Label darf nicht mit einem reservierten Befehlswort übereinstimmen. Reservierte Befehle sind Pseudo-Opcodes (siehe unten) und Maschinenbefehle des Z80 und 8502.

OPCODE:

Die Opcodes sind Befehlswörter für einen Prozessor (z.B. LDA).

PARAMETER:

Parameter sind Daten, die der Befehl für seine Ausführung benötigt. Ein Parameter kann auch aus mehreren verknüpften Elementen bestehen. Für die Zusammensetzung eines Parameters lassen sich folgende Operatoren verwenden:

- »+« addiert Werte
- »-« subtrahiert Werte
- »*« multipliziert Werte
- »!« logisches OR
- »&« logisches AND
- »↑« logisches XOR

Die Ausdrücke werden in der Reihenfolge ihres Auftretens abgearbeitet. Falls dies nicht gewünscht wird, so müssen Klammern gesetzt werden.

Die Operanden können folgende Formate haben:

\$1000	(Hexadezimal)
4096	(Dezimal)
%10000000	(Binär)
"Z"	(ASCII)
(\$1000+345*2)	zusammengesetzter Ausdruck

Ist das erste Zeichen des Parameters ein »>« oder ein »<«, so wird entsprechend das High- oder Low-Byte des gesamten nachfolgenden Ausdrucks verarbeitet.

Zu den folgenden Pseudo-Opcodes eine Anmerkung vorab: Die Parameter in eckigen Klammern sind optional, müssen also nicht unbedingt eingegeben werden. Aber nun die Opcodes:

=

Format: Label = Parameter

Dieser Pseudo-Opcode weist einem Label einen festen Wert zu. Beispiel:

```
VIC = $D000
```

weist dem Label »VIC« den Wert \$D000 zu.

==

Format: Label == Parameter

Dieser Opcode weist einem bereits existierenden Label einen neuen Wert zu. Beispiel:

```
LAUF = $0001
```

```
LAUF == LAUF + 1
```

*=

Format: *= Parameter

Weist dem Program-Counter »PC« einen neuen Wert zu. Achtung! Nach diesem Befehl werden sämtliche durch »OBJ« geöffnete Dateien geschlossen. Beispiel:

```
*= $8000
```

gibt dem »PC« die Startadresse, ab der das Programm generiert werden soll, an.

.LIS

Für diesen Befehl gibt es die folgenden drei Formate:

.LIS SCREEN oder **.LIS S**

Veranlaßt den Assembler, ein formatiertes Listing des Quelltextes auf dem Bildschirm auszugeben.

.LIS NOTHING oder **.LIS N**

Veranlaßt den Assembler, ohne Ausgabe eines Listings zu assemblieren.

.LIS GA [,SA] [,FN]

Dieser Opcode veranlaßt den Assembler, ein formatiertes Listing auf einem Peripheriegerät (Drucker, Floppy) auszugeben.

GA = Geräteadresse

SA = Sekundäradresse

FN = Dateiname

Dateiname und Sekundäradresse können weggelassen werden. Diese Kürzel gelten auch für weitere Opcodes. Beispiel:

.LIS 4,7

gibt ein formatiertes Listing des Quellcodes in Kleinschrift (diese nur auf MPS-Druckern und Kompatiblen) auf dem Drucker aus.

.OBJ

Auch für den Befehl »OBJ« gibt es mehrere Formate:

.OBJ MEMORY [,bank] oder **.OBJ M [,bank]** (bank = 0 bis 15)

Ist der Wert »bank« nicht angegeben, wird Bank 0 angenommen. »OBJ« schreibt den generierten Maschinen-Code im Speicher in die durch »bank« festgelegte RAM-Bank.

.OBJ NOTHING oder **.OBJ N**

Veranlaßt den Assembler, einen Quelltext lediglich auf Fehler zu überprüfen, ohne Maschinen-code zu generieren.

.OBJ GA [,SA] [,FN]

Mit diesem Opcode wird der generierte Code auf einem Peripheriegerät ausgegeben. Beispiel:

.OBJ 8,1, "TEST"

schreibt den generierten Maschinen-Code unter dem Dateinamen »Test« auf Diskette.

.SYM

Folgende Formate sind möglich:

.SYM SCREEN oder **.SYM S**

Dieser Opcode veranlaßt den Assembler, eine formatierte Symboltabelle auf dem Bildschirm auszugeben.

.SYM GA [,SA] [,FN]

Damit wird eine formatierte Symboltabelle auf ein Peripheriegerät ausgegeben. Beispiel:

.SYM 4, 7

gibt eine Symboltabelle auf dem Drucker aus.

.LST (Load Symbol Table)

Format: **.LST GA, SA, FN**

Wobei »SA« normalerweise »0« ist (für »PRG«-Dateien).

Dieser Opcode veranlaßt den Assembler, eine bereits gespeicherte Symboltabelle in den Speicher zu laden und mit zu verarbeiten. Beispiel:

.LST 8,0, "TABELLE"

lädt die Symboltabelle mit dem Namen »TABELLE«.

.SST (Save Symbol Table)

Format: .SST GA, SA, FN

»SA« ist normalerweise »1« (für »PRG«-Dateien).

Dieser Befehl bewirkt, daß die im Speicher befindliche Symboltabelle unter dem Dateinamen »FN« auf Diskette gespeichert wird. Beispiel:

```
.SST 8,1, "TABELLE"
```

schreibt die Symboltabelle unter dem Namen »TABELLE« auf Diskette.

.MOD

Format: .MOD 1 beziehungsweise .MOD 2

Dieser Opcode schaltet zwischen den beiden Assemblierungsmodi des Assemblers um. ».MOD 1« schaltet auf den Z80-Modus um. Das heißt, nach diesem Opcode darf nur Z80-Code folgen, entsprechend nach ».MOD 2« nur 8502-Code. Die Pseudo-Opcodes bleiben in beiden Modi gleich erhalten.

.FIL

Format: .FIL GA, FN

Dieser Befehl lädt einen weiteren Quelltext nach und assembliert diesen (siehe auch ».END«).

Assemblierung mehrerer Programmteile

.LF

Format: .LF GA, FN

Den Pseudo-Opcode ».LF« leitet »PASS 2« ein (bei mehreren Programmteilen). Besteht ein Quellprogramm aus mehreren Teilen (weil nicht alle gleichzeitig in den Speicher passen), so muß am Ende des ersten Teils der Opcode ».FIL« mit dem Namen des zweiten Teils folgen,

welcher dann im ersten Pass assembliert wird. Dies passiert so oft, wie Teile vorhanden sind. Im letzten Teil muß dann am Ende des Quellprogramms der Opcode »LF GA, FN« stehen, wobei für »FN« der Dateiname des ersten Teils steht, der sodann nachgeladen und im 2. Pass assembliert wird.

.END

Format: .END (optional)

Nach diesem Befehl beginnt der Assembler mit »PASS 2«. Der Befehl wird nur benötigt, wenn kein Assemblerbefehl als letzter Befehl im Listing steht. Es kann also beispielsweise ein Basic-Programm einfach an den Source-Code angehängt werden.

.FAS

Format: .FAS

Dieser Pseudo-Opcode schaltet den Prozessor in den 2-MHz-Modus (nur für die Assemblierung). Hierbei wird der Bildschirm ausgeblendet. Am Ende des Assemblierungsprozesses wird automatisch in den 1-MHz-Modus zurückgeschaltet.

.SLO

Format: .SLO

Schaltet den Prozessor in den 1-MHz-Modus.

.ERR

Format: .ERR

Ist der Prozessor im 2-MHz-Modus, der Bildschirm also ausgeblendet, so können Fehler nicht gesehen werden. Durch diesen Pseudo-Opcode wird, sobald ein Fehler auftritt, der Bildschirm eingeblendet. Nach einem beliebigen Tastendruck wird wieder fortgesetzt.

.BYT

Format: .BYT byte, byte, ...

Nach ».BYT« folgen, durch Kommas getrennt, beliebig viele Parameter bis zum Ende der Zeile. Hier können also alle Operatoren verwendet werden. »byte« kann Werte von dezimal 0 bis 255 annehmen.

.WOR

Format: .WOR wort, wort, ...

Dieser Befehl ist weitgehend identisch mit »BYT«, nur »wort« kann Werte von dezimal 0–65535 annehmen.

.ASC

Format: .ASC "text"

»text« ist eine Zeichenkette, deren Länge lediglich durch die Länge der Eingabezeile begrenzt ist (160 Zeichen). »text« kann aus allen ASCII-Zeichen bestehen (groß und klein). Die Werte der letzten drei durch die Befehle ».BYT«, ».WOR« und ».ASC« festgelegten Parameter werden direkt im Speicher abgelegt.

.IF <

Format: .IF < Wert 1, Wert 2, Zeilennummer

Dieser Befehl dient zur bedingten Assemblierung. »Wert 1« wird mit dem »Wert 2« verglichen. Ist »Wert 1« kleiner als »Wert 2«, so wird der Assemblierungsvorgang bei der angegebenen Zeilennummer fortgesetzt. Andernfalls wird mit dem nach ».IF« folgenden Befehl fortgesetzt.

.IF=

Format: .IF= Wert 1, Wert 2, Zeilennummer

Wie ».IF <«, nur wird hier bei Gleichheit von »Wert 1« und »Wert 2« verzweigt. Mit Hilfe der Befehle ».IF <«, ».IF=« und »==« können Schleifen aufgebaut werden.

Tips für die Befehlseingabe

Nun noch einige Hinweise zum Eingabeformat: Zwischen Labels und Opcodes (bzw. Pseudo-Opcodes) muß ein Leerzeichen stehen. Ebenso zwischen Opcodes (bzw. Pseudo-Opcodes) und Parametern. Beispiele:

falsch : *= \$8000

richtig : *= \$8000

falsch : LDA #1

richtig : LDA #1

Dies dient zusätzlich der Übersichtlichkeit.

In Tabelle 8 finden Sie alle Z80- und 8502-Opcodes sowie die Pseudo-Opcodes des Assemblers im Überblick.

Hinweis zur Zeropage-Adressierung:

Der Assembler erkennt selbständig (im 8502-Modus), ob eine Zeropage-Adressierung möglich ist. Sollte dies der Fall sein, benutzt er diese auch. Falls dies nicht gewünscht wird, muß dem gesamten Ausdruck ein Ausrufezeichen »!« vorangehen.

Beispiel:

```
LDA !$80
```

Im Speicher steht dann:

```
AD 80 00
```

(»AD« entspricht »LDA« absolut)

So werden beide Prozessor-Codes gemischt

Nach dem Starten des Assemblers befindet sich dieser im 8502-Modus. Um den Assembler auf die Maschinensprache des Z80 bzw. 8502 umzuschalten, existieren die Pseudo-Opcodes ».MOD 1« und ».MOD 2«. Zusätzlich muß dem einen Prozessor aber auch noch mitgeteilt werden, daß er die Kontrolle nun dem anderen Prozessor übergeben muß. Dieses wird nicht durch den Pseudo-Opcode ».MOD« erledigt, da dieser sich nur auf die Steuerung des Assemblers bezieht. Die Umschaltung muß vom Source-Programm selbst erfolgen. Nach der Umschaltung mit ».MOD 1« auf Z80-Code dürfen nur noch Z80-Maschinenbefehle folgen. Umgekehrt nach ».MOD 2« nur noch 8502-Befehle.

Die Vorbereitung zur Übergabe der Kontrolle muß also vorher in 8502-Maschinencode erfolgt sein. Zu dieser Kontrollübergabe nun ein Beispielprogramm:

```
10 *= $8000 ; PROGRAMM NACH $8000
20 .LIS S ; LISTING AUF BILDSCHIRM
30 .OBJ MEMORY ; OBJEKTCODE IN SPEICHER
10000 SEI ; INTERRUPT VERHINDERN
10010 LDA FFO0:PHA ; KONFIG.REG. RETTEN
10020 LDA # $C3 :STA $FFEE ; Z80-BEFEHL JP (JUMP)
10030 LDA # <Z80:STA $FFEF ; LOW-BYTE
10040 LDA # >Z80:STA $FFFO ; HIGH-BYTE
10050 LDA # $3E :STA $FF00 ; SETZEN D. KONFIG.REG.
10060 LDA $D505:PHA ; MCR RETTEN
10070 LDA # $B0 :STA $D505 ; MMU-REGISTER: Z80 EIN
10080 NOP ; WARTEN (WICHTIG !!!)
10090 PLA :STA $D505 ; MCR KORRIGIEREN
10100 PLA :STA $FF00 ; KONFIG.REG. KORRIGIEREN
10110 CLI :RTS ; INTERRUPT ERLAUBEN, RETURN
```

```

11000 Z80 .MOD 1          ; ES FOLGT Z80-CODE
11010 LD A,$3F           ; SETZEN DES KONFIGURATIONS-
11020 LD ($FF00),A       ; REGISTERS
11030 LD A,$51           ; Z80 -      --
11040 LD ($6000),A       ;      PROGRAMM --
11050 JP $FFE0           ; BETRIEBSSYSTEMROUTINE:
11060                    ; 8502 AKTIVIEREN

```

An diesem Beispiel wollen wir uns einmal genau ansehen, wie das Umschalten gemacht wird. Zunächst ist der 8502-Prozessor aktiv. Nach »STA \$D505« in Zeile 10070 wird der 8502 angehalten und der Z80 übernimmt die Kontrolle. Das folgende »NOP« (Zeile 10080) ist notwendig, um gewisse Timing-Probleme der MMU (Memory Managing Unit; dieser Baustein kümmert sich um die Speicherverwaltung) zu überbrücken. Nachdem in Zeile 10100 auch das Konfigurationsregister entsprechend gesetzt wurde, bearbeitet nun die Z80-CPU die Befehle ab Adresse \$FFEE (hier hatte der Z80 nach dem Boot-Versuch von CP/M aufgehört zu arbeiten). Diese Tatsache wird genutzt, um die Übergabe noch im 8502-Modus vorzubereiten. So wurde in der Zeile 10020 ein Z80-Sprungbefehl (\$C3 = JP) in die Adresse \$FFEE eingegeben. Es folgen das Low-Byte (Zeile 10030) und High-Byte (Zeile 10040) des eigentlichen Z80-Programms. Der Z80-Prozessor springt dann quasi in die Zeile 11010 und arbeitet ab dort weiter.

Um wieder den 8502-Prozessor zu aktivieren, genügt ein Sprung in die Betriebssystem-Routine »\$FFE0« (Zeile 11050). Danach fährt der 8502 mit dem »CLI« in Zeile 10110 fort. Auf diese Weise können Sie die teilweise höhere Arbeitsgeschwindigkeit des Z80 nutzen oder auf Befehle zurückgreifen, die der 8502-Prozessor nicht hat. Beispielsweise kann der Z80 mit einem Befehl einen kompletten Speicherblock verschieben.

Welche Möglichkeiten sich insgesamt durch das Mischen der beiden Prozessor-Codes dem Programmierer eröffnen, können wir hier nur errahnen. Die Anwendung bleibt Ihrer eigenen Kreativität überlassen.

Übrigens, auch das Spiel »Vectors« wurde mit diesem Assembler geschrieben.

(Harald Beine/Arne Jansen/M. Jobst)

- Mischen von Z80- und 8502-(6510-)Maschinencode
- Editierung und Assemblierung von bis zu 58 Kbyte Quellcode im Speicher
- File-Verkettung möglich (endloses Assemblieren!)
- automatische Anpassung an 40/80-Zeichen-Darstellung
- formatfreie Eingabe über Basic-Editor
- gleichzeitige Verwaltung von bis zu 7000 Labels
- komfortable Ausgabesteuerung durch umfangreiche Pseudocodes
- erweiterter Editor
- auf Wunsch immer resident im Speicher vorhanden

Tabelle 7. Die Leistungsmerkmale unseres Assemblers

Z80-Opcodes:							
ADC	ADD	AND	BIT	CALL	CCF	CP	
CPD	CPDR	CPI	CPIR	CPL	DAA	DEC	
DI	DJNZ	EI	EX	EXX	HALT	IM	
IN	INC	IND	INDR	INI	JP	JR	
LD	LDD	LDDR	LDI	LDIR	NEG	NOP	
OR	OTDR	OTIR	OUT	OUTD	OUTI	POP	
PUSH	RES	RET	RETN	RL	RLA	RLCA	
RLC	RLD	RR	RRA	RRC	RRCA	RRD	
RST	SBC	SCF	SET	SLA	SRA	SRL	
SUB	XOR						
8502-Opcodes:							
ADC	AND	ASL	BIT	BPL	BMI	BVC	BVS
BCC	BCS	BNE	BEQ	CMP	CPX	CPY	DEC
EOR	INC	JMP	JSR	LDA	LDX	LDY	LSR
NOP	ORA	PHA	PHP	PLA	PLP	RTI	RTS
ROL	ROR	SBC	SEC	SED	SEI	TAX	TAY
TSX	TXA	TXS	TYA	STA	STX	STY	
Pseudo-Opcodes:							
=	==	*=	.ASC	.BYT	.WOR	.MOD	
.OBJ	.LIS	.SYM	.LST	.SST	.END	.FIL	
.LF	.IF=	.IF<	.SLO	.FAS	.ERR		

Tabelle 8. Liste aller reservierten Befehle (Z80- und 8502-Opcodes sowie die Pseudo-Opcodes des Assemblers).

Butler

Machen Sie mehr aus Ihrem C 128. »Butler« gestattet Ihnen ein komfortableres Arbeiten mit dem Computer. Der Editor wurde verbessert, und neben einer Reihe von nützlichen Befehlen lassen sich von nun an auch Maschinenroutinen des Z80-Prozessors vom Basic aus aufrufen.

Bei dem vorliegenden Programm BUTLER handelt es sich um eine Betriebssystemerweiterung, die zu Beginn aktiviert werden muß, dann aber stets verfügbar bleibt. Das Programm setzt sich aus zwei Teilen zusammen, von denen der eine im freien Speicherbereich von \$1300 bis \$1BFF und der andere am oberen Ende des Textspeichers zwischen \$F000 und \$FEFF liegt. Durch Änderung des Zeigers, der das Ende des verfügbaren Textspeichers definiert, schützt es sich vor Überschreiben durch einen Basic-Text.

Verwendungsmöglichkeiten

Das Programm ermöglicht ein komfortableres Arbeiten mit dem C 128. Es ist nur im C 128-Modus verwendbar, da es intensiv Betriebssystemroutinen nutzt und speziell auf die Besonderheiten und zahlreichen Möglichkeiten des C 128 zugeschnitten ist.

Das Betriebssystem wurde im wesentlichen an zwei Stellen erweitert: Die Verwendung der Tastatur wurde erleichtert und um einige Funktionen erweitert. Das ohnehin umfangreiche Basic 7.0 wurde mit weiteren nützlichen Befehlen und Funktionen ausgestattet.

Aber nicht nur für Basic-Programmierer dürfte dieses Programm interessant sein, sondern auch Freunden der Maschinensprache bietet es viele Unterroutinen, auf die zurückgegriffen werden kann. Detailliertere Information ist dem betreffenden Abschnitt dieser Beschreibung zu entnehmen.

Aktivierung

Um das Programm zu starten, benutzt man am einfachsten den RUN-Befehl:

```
RUN " BUTLER "
```

Das kleine Basic-Programm lädt die Dateien »/BUTLER.1« und »/BUTLER.2« mit dem BOOT-Befehl in den Speicher, wobei sie auch gleich initialisiert werden. Die Erweiterung bleibt aktiv, bis der RESET-Taster betätigt oder mittels dem Basic-Aufruf

```
SYS DEC("1303")
```

die Abschaltung veranlaßt wird. Eine erneute Aktivierung ist mit

```
SYS DEC("1300")
```

möglich.

Tastaturerweiterungen

Vereinfachte Eingabe von Escape-Sequenzen

Wer fand es nicht auch schon lästig, beim Einfügen oder Löschen von Bildschirmzeilen ständig wechselnd die <ESC>-Taste und den jeweiligen Buchstaben drücken zu müssen? All jenen ist geholfen! Nun kann man die <ALT>-Taste gedrückt halten und dann die betreffende Kommandotaste betätigen. Dies funktioniert bei allen Escape-Sequenzen und natürlich mit automatischer Wiederholung, falls die Kommandotaste ebenfalls gedrückt gehalten wird.

Cursorfunktionen

Nichts wesentlich Neues, aber zur Verwendung für den einen oder anderen vielleicht vorteilhafter, sind folgende neue Escape-Sequenzen:

<ESC> + <HOME> oder <ALT + HOME>: Cursor an den Anfang der untersten Fensterzeile

<ESC> + <Ü> oder <ALT+Ü> : Fenster ab der aktuellen Cursor-Position löschen (für deutsche Tastatur)

<ESC> + <INS/DEL> oder <ALT + INS/DEL> : Fensterzeile löschen und restlichen Teil des Fensters nach oben schieben (=ESC+D)

<ESC> + <SHIFT+INS/DEL> oder <ALT + SHIFT + INS/DEL> : Fensterzeile einfügen; dazu alle Zeilen ab der aktuellen nach unten schieben <ESC> + <I>

LIST-Funktionen

Neben den einfacheren Funktionen des letzten Abschnitts bietet der BUTLER aber auch komplexere Kommandos, um das gespeicherte Basic-Programm zu LISTen oder über den Bildschirm zu scrollen. Die Kommandos können durch Escape-Sequenzen oder als ALT-Tastenkombinationen eingegeben werden und stellen keine Basic-Befehle dar. Dabei ist die aktuelle Position des Cursors stets gleichgültig.

Um die Kommandos erklären zu können, ist etwas zu deren Arbeitsweise vorzuschicken. Wird ein Kommando ausgelöst, dann wird das Bildschirmfenster nach einer Zeilennummer abgesucht, und zwar je nach Kommando entweder von oben nach unten, um die oberste oder von unten nach oben, um die unterste Zeilennummer zu finden. Damit die Suchzeit möglichst kurz bleibt, wird eine Zeilennummer nur erkannt, wenn sie in der ersten Spalte beginnt. Doch dies ist ja normalerweise immer der Fall. Wird keine Nummer gefunden, behandelt das Programm dies als sinnvollen Sonderfall.

Die Kommandos

<ESC> + <CRSR-unten> beziehungsweise <ALT + CRSR-unten> : weiterblättern unterste Zeilennummer suchen; falls keine existiert, ab Programmanfang, sonst ab der Basic-Zeile mit nächstgrößerer Nummer so viele Zeilen listen, wie im Fenster Platz haben. Falls nicht genügend nachfolgende Basic-Zeilen existieren, wird stets die letzte Seite des Programms gelistet.

<ESC> + <CRSR-hoch> beziehungsweise <ALT + CRSR-hoch>: zurückblättern oberste Zeilennummer suchen; falls keine existiert, bis Programmende, sonst bis zur Basic-Zeile mit nächstkleinerer Nummer so viele Zeilen listen, wie im Fenster Platz haben. Falls nicht genügend vorausgehende Basic-Zeilen existieren, wird die erste Seite des Programms gelistet.

<ESC> + <CRSR-rechts> beziehungsweise <ALT + CRSR-rechts>: Zeile weiter unterste Zeilennummer suchen; falls keine existiert, die erste Seite des Programms listen, sonst die Basic-Zeile mit nächstgrößerer Nummer unter der Zeile mit der gefundenen Zeilennummer ausgeben.

<ESC> + <CRSR-links> beziehungsweise <ALT + CRSR-links>: Zeile zurück oberste Zeilennummer suchen; falls keine existiert, die letzte Seite des Programms listen, sonst die Basic-Zeile mit nächstkleinerer Nummer in der Fensterzeile vor der Zeile mit der oberen Zeilennummer ausgeben.

<ESC> + <LF> beziehungsweise <ALT + LF>: Seite noch einmal oberste Zeilennummer suchen; falls keine existiert, ab Programmanfang, sonst ab der zugehörigen Basic-Zeile so viele Zeilen ausgeben, wie in das Fenster passen. Falls nicht genügend Basic-Zeilen existieren, um das Fenster zu füllen, wird die letzte Seite des Programms gelistet.

<ALT + HELP>: Bereich um Fehlerzeile listen wie der Basic-Befehl HELP, nur werden hier außer der fehlerhaften Zeile zusätzlich einige Zeilen davor und danach ausgegeben.

Wichtige Hinweise

Anmerkung: Beim Listen der letzten Programmseite oder beim Zurückblättern führen Basic-Zeilen, die sich über mehr als eine Fensterzeile erstrecken, dazu, daß einige Zeilen unnötig ausgegeben werden.

Bemerkungen zur Verwendung

Da die Kommandos sich stets am aktuellen Fensterinhalt orientieren, ist es gleichgültig, wie bereits gelistete Zeilen auf den Bildschirm gekommen sind. Man kann sich also jederzeit mit dem LIST-Befehl einen Zeilenbereich ausgeben lassen und dann zeilenweise weiterscrollen oder seitenweise weiterblättern.

Möchte man sich das Basic-Programm von vorn seitenweise ansehen, löscht man einfach das Bildschirmfenster und blättert mit <ESC> + <CRSR-unten> oder <ALT + CRSR-unten>. Will man von hinten anfangen, löscht man ebenfalls das Fenster und blättert entsprechend mit <ESC> + <CRSR-oben> oder <ALT + CRSR-oben>.

Ist man sich über die Arbeitsweise der Kommandos im klaren, dann lassen sich diese noch vorteilhafter einsetzen, indem man eine Zeilennummer geeignet im Fenster positioniert und dann eines der Kommandos auslöst.

Basic-Erweiterungen

Speicherbefehle

Wer in Basic mit Bytes hantiert, mußte bisher mit PEEK und POKE vorliebnehmen. Doch gerade bei so einfachen, maschinennahen Operationen macht sich die relativ langsame Arbeitsgeschwindigkeit schnell bemerkbar, weil man in der Regel nicht nur einzelne Bytes, sondern ganze Speicherbereiche zu bewältigen hat. Will man trotzdem nicht gleich auf Assembler ausweichen, sind die neuen Operationen dieses Abschnitts bestimmt nützlich.

Wie bei PEEK und POKE muß vor Ausführung eines Befehls die zu benutzende Speicherbank angewählt werden. Dies kann, wie gewohnt, mit dem BANK-Befehl erfolgen. Da es jedoch zum Beispiel beim Kopierbefehl auch möglich sein soll, von einer Speicherbank auf eine andere umzukopieren, sind manchmal zwei vorzuzählende Speicherbanken erforderlich. Diese können mit folgendem Befehl eingestellt werden:

```
BANKS  BankNr1, BankNr2
```

für bestimmte Operationen Speicherbank BankNr1 und BankNr2 vorwählen; der erste Operand überschreibt die mit dem BANK-Befehl ausgewählte Speicherbank; der zweite Operand wird gesondert gespeichert.

Im folgenden gilt für die Befehlsoperanden normalerweise stets die erste Banknummer. Nur bei Befehlen mit zwei Speicherbereichen wird für den zweiten Bereich die zweite Banknummer verwendet. Der Befehl BANKS wird in den nächsten Abschnitten noch mehrfach auftauchen. Der Erklärung der Speicherbefehle ist jeweils ein Beispiel angefügt. Die Beispiele operieren im 40-Zeichen-Bildschirmspeicher, so daß die Wirkungsweise der Befehle direkt mitverfolgt werden kann.

Die Speicherbefehle im einzelnen

MEMFILL Adr1,Adr2,Wert

Speicherbereich von Adr1 bis einschließlich Adr2 mit Wert füllen. Beispiel:

```
BANK 0
FOR I=0 TO 127:MEMFILL 1024,2023,I:SLEEP 1:NEXT
MEMCOPY Adr1a,Adr1b,Adr2a
```

Speicherbereich von Adr1a bis einschließlich Adr1b in den Bereich ab Adr2a kopieren. Beispiel:

```
BANKS 0,1:MEMCOPY 1024,2023,30000
SCNCLR 0
BANKS 1,0:MEMCOPY 30000,30999,1024
MEMSWAP Adr1a,Adr1b,Adr2a
```

Speicherbereich von Adr1a bis einschließlich Adr1b mit dem Bereich ab Adr2a austauschen. Beispiel:

```
BANKS 0,1
FOR I=1 TO 10:MEMSWAP 1024,2023,30000:NEXT
MEMINV Adr1,Adr2
```

Speicherbereich von Adr1 bis einschließlich Adr2 invertieren, also die Reihenfolge aller Bytes im angegebenen Bereich umdrehen. Beispiel:

```
BANK 0
FOR I=1 TO 10:MEMINV 1024,2023:SLEEP 1:NEXT
MEMROT Adr1,Adr2,Anz
```

Speicherbereich von Adr1 bis einschließlich Adr2 gemäß Anz vorwärts rotieren lassen. Dies bewirkt, daß der Bereich von Adr1 bis Adr1+Anz ans Ende, der restliche Bereich zum Anfang kopiert wird. Beispiel:

```
BANK 0:MEMROT 1024,2023,40
```

Anmerkung: Der Befehl eignet sich nicht dazu, den 40-Zeichen-Bildschirm rotieren zu lassen, da die Daten zwischenzeitlich scheinbar wild durcheinander geraten. Der Befehl ist vielmehr bei der Vertauschung von aneinandergrenzenden Datenbereichen unterschiedlicher Größe nützlich, wenn kein Zwischenspeicher zur Verfügung steht.

```
MEMLOC (Adr1,Adr2,String)
```

Speicherbereich von Adr1 bis einschließlich Adr2 nach dem ersten Auftreten der Zeichenkette String absuchen. Eine Zeichenkette wird nur erkannt, wenn sie mit voller Länge im angegebenen Bereich liegt. Falls die Suche erfolgreich war, wird die Anfangsadresse, sonst stets Null zurückgegeben. Beispiel:

```
BANK 0:PRINT MEMLOC (1024,2023,"123")
```

Anmerkung: Möchte man den Bildschirm nach einer anderen Zeichenkette absuchen, ist der spezielle Bildschirmcode zu beachten. Bei Ziffern deckt er sich mit dem normalen Code.

Adreßoperationen

Möchte man auf zwei aufeinanderfolgende Bytes zugreifen, weil diese beispielsweise einen Zeiger repräsentieren, so ist dies mit PEEK und POKE etwas umständlich. Ferner ist es im Basic oft gar nicht möglich, den Wert eines Sprungvektors zu verändern, der in der Unterbre-

chungsroutine verwendet wird, ohne daß zwischenzeitlich ein inkonsistenter Sprungvektor entsteht, der dann zum Absturz des Computers führt.
Erleichterung schaffen folgende neuen Operationen:

CHG Adr,Wert

Speicherstellen Adr und Adr+1 mit dem niederwertigen und dem höherwertigen Byte des Operanden Wert beschreiben. Unterbrechungen werden während des Änderns nicht zugelassen.

VEC (Adr)

Werte der Speicherstellen Adr und Adr+1 auslesen und in eine Adresse umrechnen. Bei Adr muß das niederwertige, bei Adr+1 das höherwertige Byte stehen. Beispiel:

BANK 0	
PRINT VEC(4624)-VEC(45)	Länge des Basic-Textes
CHG 4633,DEC("8000")	USR-Vektor ändern
PRINT USR(1)	und verwenden
CHG 808,DEC("F4A2")	<RUN/STOP>-
	Taste inaktivieren
CHG 808,DEC("F66E")	und wieder aktivieren
10 INPUT "NR";B	Nummer einlesen,
20 A=45:BANK 0	Anfangsadresse
30 DO:A=VEC(A):NR=VEC(A+2)	der zugehörigen
40 LOOP UNTIL A=0 OR NR=B	Basic-Zeile suchen
50 PRINT "ADR: ";A	und ausgeben

Mit den beiden letzten Operationen läßt sich also das Zerlegen eines Wertes in niederwertiges und höherwertiges Byte vermeiden. Sollten diese Bytes aber dennoch von Interesse sein, lassen sie sich mit folgenden zwei Funktionen leicht berechnen:

LOW (Wert)	niederwertiges Byte von Wert
HIGH (Wert)	höherwertiges Byte von Wert

Beispiel:

```
PRINT LOW(64000)
PRINT HIGH(64000)
```

Keiner langen Vorrede bedarf der Befehl

OLD

zum Wiederherstellen eines mit NEW gelöschten Basic-Programms. Häufiger wird diese Anweisung jedoch dann angewendet, wenn nach Drücken des RESET-Knopfs das Programm wiederhergestellt werden soll. Da bei der Neuinitialisierung nach Drücken des RESET-Knopfs auch die Basic-Erweiterung inaktiviert wird, ist es in diesem Fall zunächst erforderlich, sie mit SYS DEC("1300") erneut zu aktivieren.

Vielseitiger als der OLD-Befehl ist der Befehl

PIX

HIDE Nummer oder HIDE,

der es ermöglicht, den Anfang des Basic-Textes vorübergehend zur angegebenen Zeile oder ganz ans Programmende zu legen, so daß der vordere Teil des Programms abgespalten bleibt, bis er mit dem Befehl

UNITE

wieder mit dem restlichen Programmtext verbunden wird.

Manipulieren und Editieren des Programmtextes

Die beiden letzten Befehle eröffnen interessante Anwendungen.

– schnelleres Editieren großer Programme:

Mit HIDE den vorderen Teil abtrennen. Dann läßt sich der Rest bequemer editieren, weil das Eingeben von Zeilen schneller erfolgt. Zum Schluß ist mit UNITE wieder alles zu verbinden.

- getrenntes Speichern eines hinteren Programmteils:
Mit HIDE an der gewünschten Stelle das Programm trennen, hinteren Teil wie gewohnt speichern, und dann entweder mit UNITE alles wieder verbinden oder aber mit NEW den gespeicherten Teil löschen und erst dann UNITE verwenden.
- Verbinden von Programmteilen:
Ersten Programmteil laden, mit HIDE verstecken, zweiten Programmteil normal laden, umnummerieren und dann mit UNITE anhängen.
- Programmteile während des Programmlaufs austauschen oder nachladen: Beispiel:

```

1 HIDE 10
2 BLOAD "ROUTINE 1",ON BO,P(VEC(45))
3 UNITE:GOSUB 9
4 HIDE 10
5 BLOAD "ROUTINE 2",ON BO,P(VEC(45))
6 UNITE:GOSUB 9
7 HIDE 10:BANK 0:CHG VEC(45),0:UNITE
8 END
9 REM *** Aufrufadresse ***

```

Dieses Beispiel setzt voraus, daß auf der Diskette die zwei Unterprogramme ROUTINE 1 und ROUTINE 2 stehen, die mit einer Zeilennummer beginnen. Ferner müssen die Routinen mit RETURN enden, da sie mit GOSUB aufgerufen werden.

Wie diesem Beispiel zu entnehmen ist, kann das Programm im versteckten Teil weiterlaufen. In diesem Teil werden jedoch keine Zeilennummern mehr gefunden, da der Basic-Interpreter mit der Suche erst beim restlichen Teil beginnt.

Nach so vielen Erläuterungen nun noch zu einem anderen Befehl, der weniger mit Programmzeigern, sondern viel mehr mit Programmzeilen hantiert:

```
TRANS Nummer1a,Nummer1b,Nummer2
```

Alle Zeilen des Zeilenbereichs von Nummer1a bis Nummer1b vor der Zeile mit Nummer2 einfügen.

Aufruf des Z80-Prozessors von Basic aus

Achtung: Die Nummern werden beim Kopieren nicht verändert. Daher sollten die Zeilen anschließend mit dem RENUMBER-Befehl neu durchnummeriert werden. Da das Neunummern nur dann durchgeführt wird, wenn alle Zeilenreferenzen aufgelöst werden können, sollte dies schon vorher sichergestellt sein, da die Korrektur des Programms nach dem Kopieren wegen der dann nicht streng steigenden Zeilennummerierung oft nicht mehr möglich ist.

Der letzte Befehl ist also zwar sehr nützlich, aber auch äußerst tückisch, wenn man nicht genügend aufpaßt. Vorsicht ist stets angebracht und nur geübte Benutzer sollten diesen Befehl bei ungesicherten Programmen anwenden.

In der Regel führt der Z80-Prozessor im C 128 ein stilles Dasein. Wer mit der Programmierung dieses Mikroprozessors vertraut ist, kann ihn jetzt vom Basic aus mit dem Befehl

Z8OCALL Adresse

aufrufen. Die Unterroutine muß stets in der Speicherbank 0 stehen.

Die Verwendung dieses Prozessors kann mitunter sinnvoll sein, da er zum Beispiel einen mächtigen Kopierbefehl kennt.

Beispiele:

im Monitor einzugeben:

```
>0A000 01 00 20 11 00 80 21 00
>0A008 20 ED B0 C9
$01 $00 $20      LD BC,2000H   (Anzahl)
$11 $00 $80      LD DE,8000H   (Ziel)
$21 $00 $20      LD HL,2000H   (Quelle)
$ED $B0          LDIR
$C9              RET
```

Aufruf vom Basic aus:

```
Z8OCALL DEC("A000")
```

Diese kurze Routine kopiert 8 Kbyte von Adresse \$2000 bis \$3FFF nach Adresse \$8000 bis \$9FFF, und dies einiges schneller als der 8502-Prozessor.

Folgendes ist wieder im Monitor einzugeben:

```
>0B000 01 FF 1F 11 01 20 21 00
>0B008 20 36 01 ED B0 C9
$01 $FF $1F          LD BC,1FFFH  (Anzahl)
$11 $01 $20          LD DE,2001H  (Ziel)
$21 $00 $20          LD HL,2000H  (Quelle)
$36 $01              LD (HL),1    (Wert)
$ED $B0              LDIR
$C9                  RET
```

Aufruf vom Basic aus:

```
Z80CALL DEC("B000")
```

Bei dieser Routine wird ebenfalls der Blockkopierbefehl LDIR verwendet, hier jedoch dient er dazu, den Speicherbereich von \$2000 bis \$3FFF mit 1 zu füllen. Dies ist möglich, weil der Blockkopierbefehl von vorn nach hinten kopiert.

Anmerkung: Im Adreßbereich von \$0000 bis \$0FFF ist für den Z80 ein spezielles ROM eingebildet, so daß er auf das RAM in diesem Adreßbereich leider nicht zugreifen kann.

Flexible Erweiterungen

Die bisher vorgestellten Erweiterungen sind alle möglichst allgemein gehalten und bilden in der Regel nur eine Basis dessen, was man sich vielleicht für seine persönlichen Problemstellungen wünschen würde. Um weitergehenden Wünschen entgegenzukommen, bietet BUTLER die Möglichkeit, die Tastatur- und vor allem die Basic-Erweiterungen einfach zu ergänzen oder umzudefinieren:

```
NEWESC Wert,Adresse
```

Neue Escape-Sequenz für das Zeichen mit ASCII-Code Wert definieren. Bei der Eingabe der zugehörigen Escape-Sequenz beziehungsweise ALT-Tastenkombination wird die Routine unter der angegebenen Adresse aufgerufen. Als Speicherbank wird die zuvor eingestellte verwendet. Beispiel:

```
BANK 15
NEWESC ASC("1"),DEC("CABC")
NEWESC ASC("2"),DEC("CACA")
```

<ESC> + <1> beziehungsweise <ALT+1> scrollt danach den Bildschirm hoch.
<ESC> + <2> beziehungsweise <ALT+2> scrollt analog dazu abwärts.

NEWCMD Name, Adresse

Neuen Basic-Befehl definieren. Die Routine ab der angegebenen Adresse auf der eingestellten Bank muß eventuelle Parameter selbst einlesen und auswerten. Beispiel:

```
BANK 15
NEWCMD "BELL",DEC("C98E")
BELL
```

NEWFN Name, Adresse

Neue Basic-Funktion vereinbaren. Ein Funktionsparameter wird automatisch ausgewertet, die restlichen müssen selbst eingelesen werden. Beispiel:

```
BANK 1
POKE 30000,96:REM CODE FUER RTS
NEWFN "DUMMY",30000
PRINT DUMMY(99)
PRINT DUMMY("ABC")
```

Anmerkung: Wie an diesem Beispiel erkennbar ist, muß die Typprüfung von der Funktionsroutine vorgenommen werden.

So schön die Möglichkeiten der drei letzten Befehle auch sind, so setzen sie doch einige Erfahrung mit der Programmierung geeigneter Routinen voraus. Die notwendige Erfahrung kann man durch Analyse entsprechender Betriebssystemroutinen erlangen.

Codierung der neuen Befehle

Sicherlich von Interesse dürfte noch sein, wie die neuen Befehle verschlüsselt werden. Das Betriebssystem ermöglicht es, Erweiterungen durch zwei Byte (\$CE \$xx bei Funktionen, \$FE \$xx bei Anweisungen) zu kodieren. Das Betriebssystem selbst verwendet bereits die Kodierungen bis \$CE \$0A, beziehungsweise \$FE \$26. Die neuen Funktionen und Befehle von BUTLER verwenden Kodierungen bis \$CE \$0E, beziehungsweise \$FE \$37. Für künftige Erweiterungen sind auch die Kodierungen bis \$CE \$1A, beziehungsweise \$FE \$56 reserviert. Somit werden die selbstdefinierten Erweiterungen ab \$CE \$1B oder \$FE \$57 durchnummeriert. Dies geschieht in der Reihenfolge der Vereinbarungen und kann nicht beeinflusst werden. Mit dem Befehl

CLRNEWS

kann lediglich sichergestellt werden, daß die Codezuweisung stets ab den angegebenen Werten beginnt. Dabei werden alle zuvor definierten Erweiterungen inaktiviert. Programme, die sich Erweiterungen selbst definieren, sollten diesen Befehl verwenden, um eine definierte Startumgebung vorzufinden.

Die Wirkung des letzten Befehls, aber viel mehr noch eine gesamte Neuinitialisierung von BUTLER und einiger wichtiger Betriebssystemdaten werden durch den Befehl

RESET

erreicht. Dieser Befehl ist besonders dann sinnvoll, wenn Sprungvektoren zwischenzeitlich verändert wurden und nun wieder ihre alten Werte erhalten sollen.

Schnittstelle zur Maschinenprogrammenebene

Bankübergreifende Programmierung

Beim C128 verfügt man zwar über verschiedene Speicherbänke, die vom Basic aus relativ einfach zu handhaben sind, doch sobald Maschinenprogramme zu schreiben sind, die auf eine andere Bank zugreifen sollen, erweist sich das Speicherbankkonzept als umständlich. Schon bei der Verwendung von Betriebssystemroutinen zeigen sich die ersten Probleme. Um eine Routine direkt aufrufen zu können, muß das betreffende ROM bereits beim Aufruf eingeblendet sein. Dies schränkt die Lage und die Größe des Maschinenprogramms ein. Ferner ist nicht stets gewährleistet, daß beim Rücksprung aus der Betriebssystemroutine das aufrufende Programmstück noch (oder wieder) sichtbar ist. Um letzterem Problem aus dem Weg zu gehen, aber auch um Routinen »unter« das Betriebssystem legen zu können, verfügt BUTLER über eine spezielle Aufrufroutine, mit der Unterprogramme im ROM bequem aufgerufen werden können. Ebenso existiert eine Routine, die es gestattet, ein Unterprogramm in einer beliebigen Speicherbank aufzurufen.

Die Verwendung sei an Beispielen erläutert:

```
JSR $1396 ;ROM-Aufruf veranlassen
.WORD Adr ;ROM-Adresse low/high
JSR $13C7 ;Bank-Aufruf veranlassen
.BYTE Bank ;Bank-Nr (0 bis 15)
.WORD Adr ;Adresse der Routine (low/high)
```

Ungewöhnlich mag erscheinen, daß direkt hinter dem Aufrufbefehl Daten stehen. Diese Daten werden von den speziellen Aufruf Routinen gelesen und der Programmzähler korrigiert, so daß beim Rücksprung das Programm hinter den Daten fortgesetzt wird. Diese Übergabetechnik erlaubt es, alle Register normal verwenden zu können, setzt aber auch voraus, daß stets mit JSR aufgerufen wird, selbst wenn der nächste Befehl »nur« ein RTS ist.

Sprünge ins ROM oder auf eine andere Bank sind mit der letzten Routine nicht möglich. Ist dies doch einmal notwendig, dann kann man sich eventuell wie folgt helfen:

```

LDA #Adr-1    ;höherwertiges Byte
PHA           ;merken
LDA #Adr-1    ;niederwertiges Byte
PHA           ;merken
LDA #Konfig   ;gewünschte Konfiguration
JMP $02DF     ;umkonfigurieren und springen

```

Bei diesem Beispiel darf die anzugebende Konfiguration nicht mit der Banknummer verwechselt werden. Welche Konfiguration für welche Speicherbank zu wählen ist, kann einer Tabelle entnommen werden, wobei die Banknummer als Index dient. Das Original dieser Tabelle steht im ROM ab \$F7F0, eine Kopie aber auch in BANK 0 ab \$1310, wenn BUTLER verwendet wird. Wer sich bereits mit der bankübergreifenden Programmierung beschäftigt hat, wird sicherlich die vorhandenen Routinen ab \$02CD und \$02E3 kennen, doch deren Verwendung ist bisweilen recht umständlich und einschränkend, weil beim Aufruf Teile des ROMs aktiviert sein müssen.

Einbinden von Erweiterungen

Nicht nur direkt vom Basic aus, sondern auch auf Maschinenspracheebene lassen sich Erweiterungen definieren. Wie dies möglich ist, soll an kommentierten Beispielen erläutert werden.

Tastaturerweiterungen

Die notwendigen Daten für die Erweiterungen müssen in einer Liste abgelegt sein. Jeder Listeneintrag muß nachfolgende Struktur aufweisen:

Konfiguration, Aufrufadresse-1, ASCII-Code

Listeneinträge sind lückenlos aneinanderzuhängen, und das Ende der Liste ist nicht zu kennzeichnen. Statt dessen ist beim Aufruf die Anzahl der Listenelemente (im Akkumulator) zu übergeben. Da diese Liste nicht notwendigerweise in der Speicherbank 0 stehen muß, ist auch die Banknummer (im X-Register) anzugeben. Schließlich muß noch die Anfangsadresse der Liste in der Zeropage bei \$24/\$25 bereitgestellt werden. Beispiel:

```

        LDA #Liste      ;Zeiger auf
        LDY #Liste      ;Liste
        STA $24         ;in der Zeropage
        STY $25         ;ablegen
        LDX #0          ;BankNr der Liste
        LDA #2          ;Elementanzahl
        JMP $F15D       ;Tastatur erweitern

Liste  .BYTE $3F        ;Konfiguration
        .WORD Beleg1-1 ;Adr-1 (low,high)
        .BYTE '1'       ;ESC 1

        .BYTE $3F        ;Konfiguration
        .WORD Beleg2-1 ;Adr-1 (low,high)
        .BYTE '2'       ;ESC 2

Beleg1 LDA #Tab1       ;Zeiger auf
        LDX #Tab1       ;erste Fkts-Tastenbelegung
        LDY #TabLg1     ;und deren Länge
        BNE DefBel

Beleg2 LDA #Tab2       ;Zeiger auf
        LDX #Tab2       ;zweite Fkts-Tastenbelegung
        LDY #TabLg2     ;und deren Länge

DefBel STA $24         ;Zeiger in
        STX $25         ;Zeropage ablegen

KopBel DEY            ;
        LDA ($24),Y     ;neue Funktions-
        STA $1000,Y     ;tastenbelegung
        TYA             ;umkopieren
        BNE KopBel     ;
        RTS

```

```

Tab1 .BLOCK 10,6,10,5,6,4,5,8,4,5      ;Stringlängen
      .BLOCK 'PRINT DS$',13,'DLOAD"'    ;F1,F2
      .BLOCK 'DIRECTORY',13,'BOOT"'     ;F3,F4
      .BLOCK 'DSAVE"',      'RUN"'      ;F5,F6
      .BLOCK 'LIST',13,      'MONITOR',13;F7,F8
      .BLOCK 'RUN',13,      'HELP',13   ;RUN,HELP

```

```

TabLg1 .EQU *-Tab1

```

```

Tab2 .BLOCK 7,5,6,7,4,5,3,6,4,5      ;Stringlängen
      .BLOCK 'GRAPHIC',      'COLOR'    ;F1,F2
      .BLOCK 'LOCATE',      'SCNCLR',13 ;F3,F4
      .BLOCK 'DRAW',        'PAINT'     ;F5,F6
      .BLOCK 'BOX',         'CIRCLE'    ;F7,F8
      .BLOCK 'RUN',13,      'HELP',13   ;RUN,HELP

```

```

TabLg2 .EQU *-Tab2

```

Basic-Erweiterungen

Ganz analog verläuft die Definition von Basic-Erweiterungen, nur die Struktur der Listenelemente ist etwas anders:

Typ, BankNr, Aufrufadresse, Name, Null

Als Typen sind die Werte 0 (bei Funktionen) und 1 (bei Befehlen) zulässig. Der Name der Funktion beziehungsweise des Befehls ist direkt in die Liste einzutragen und muß durch eine Null abgeschlossen werden. Die Anfangsadresse der Liste muß in \$5A/\$5B bereitstehen. Beispiel:

```

LDA #Liste      ;Zeiger auf
LDY #Liste      ;Liste
STA $5A         ;in der Zeropage
STY $5B         ;ablegen
LDX #0          ;BankNr der Liste
LDA #2          ;Elementanzahl
JMP $F23E      ;Basic erweitern

```

```

Liste  .BYTE 0          ;Typ=Funktion
       .BYTE 0          ;BankNr
       .WORD LnAdr      ;Adr (low,high)
       .BLOCK 'LNADR'  ;Name ('line adress')
       .BYTE 0          ;Ende des Namens

       .BYTE 1          ;Typ=Befehl
       .BYTE 0          ;BankNr
       .WORD ListLn    ;Adr (low,high)
       .BLOCK 'LISTLN';Name ('list line')
       .BYTE 0          ;Ende des Namens

LnAdr  JSR $FOCF        ;Real nach Integer wandeln
       JSR $16C6        ;zugehörige Zeile suchen
       LDY $61          ;Adresse ins Y-Reg. und
       LDA $62          ;Akku. (low/high Byte)
       BCS AdrOk        ;falls Zeile gefunden
       LDA #0           ;sonst Null
       TAY              ;als Adresse
AdrOk  JMP $F105        ;Y/A nach Real wandeln

ListLn JSR $F0DB        ;Integer-Wert einlesen
       JSR $16C6        ;zugehörige Zeile suchen
       BCC Ende        ;falls nicht gefunden: Ende
       LDX $16          ;sonst Nummer
       LDA $17          ;laden und
       JSR $16D0        ;Zeile auflisten
Ende   RTS

```

Das angeführte Beispiel enthält eine Funktion zur Bestimmung der Anfangsadresse einer Basic-Zeile und einen Befehl zum Auflisten einer einzelnen Basic-Zeile. Hierbei werden mehrfach BUTLER-Routinen verwendet, die ihrerseits Routinen im ROM aufrufen. Diese ROM-Routinen könnte man auch direkt aufrufen, wenn man die speziellen Aufrufrouinen verwendet, die am Anfang des Kapitels vorgestellt wurden.

Tips zur Speicherbelegung

Die mit dem BANK-Befehl vorgewählte Speicherbank wird bekanntlich an der Adresse \$03D5 abgelegt. Die mit dem neuen BANKS-Befehl vorwählbare zweite Speicherbank ist an der Adresse \$03FD zu finden. Die Speicherstellen \$03FE und \$03FF sind für andere Anwendungen reserviert. Außerdem ist darauf zu achten, daß der Bereich von \$03E4 bis \$03EF intensiv verwendet wird und nicht anderweitig verwendet werden kann.

BUTLER selbst besteht aus zwei Teilen, die in BANK 0 in den Bereichen von \$1300 bis \$19FF und von \$F000 bis \$FCFF liegen. Ab \$1A00 bis maximal \$1BFF ist die Tabelle der Basic-Erweiterungen zu finden. Die zugehörige Sprungtabelle beginnt bei \$FD00, gefolgt von der Tabelle für die neuen Escape-Sequenzen ab \$FE00.

Abschließend bleibt noch anzuführen, daß temporär die RS232-Ein- und -Ausgabepuffer ab \$0C00 beziehungsweise \$0D00 verwendet werden. (J.Blechinger/ah/nj)

Window-Tech

Damit Sie Ihre Programme professioneller und übersichtlicher gestalten können, stellen wir hier eine Erweiterung vor, die 10 »echte« Windows bearbeiten kann.

Das Programm »Window-Tech 80« ist eine Betriebssystem-Erweiterung für den 80-Zeichen-Modus des C 128, die nicht nur 10 (!) Windows gleichzeitig bearbeitet, sondern es können auch noch die Bildschirminhalte gerettet werden. »Ganz nebenbei« wird der 80-Zeichen-Bildschirm um zwei weitere Bildschirmzeilen erweitert, von denen die zweite am unteren Bildschirmrand die aktuelle Uhrzeit (inklusive Alarm) und den aktuellen Programmzählerinhalt der CPU 8502 laufend anzeigt; die erste Zusatz-Zeile bleibt leer und dient nur zur Verschönerung des Designs, damit die Statuszeile nicht an der untersten regulären Bildschirmzeile »klebt«.

Das Programm wird mit `BLOAD "WINDOW-TECH 80",ON B0` geladen und mit `SYS DEC("IBD6")` gestartet. Danach ist diese Erweiterung aktiviert, was sich an der Einblendung der Statuszeile bemerkbar macht.

Die Anzeige des PC (Programmzähler des Prozessors) ist für Maschinenprogrammierer beim Debugging (Fehlersuche) nützlich. Bleibt der PC stehen, so durchläuft das Maschinenprogramm entweder eine Schleife, die immer wieder ihre eigene Adresse anspringt (`$5000 jmp $5000`), oder es liegt ein Prozessorabsturz vor. Beim Hängenbleiben eines Programms in einer größeren Endlosschleife kann man die kritische Adresse zumindest erahnen.

Rechts neben dem PC befindet sich eine Echtzeituhr und noch einmal rechts davon die Alarmzeit, die durch Einstellen aktiviert wird.

Nach dem Start des Programms befindet man sich im Fenster mit der Nummer » #1«. Alle zehn Fenster sind in ihrer Größe frei wählbar, sind aber nach dem Start schon voreingestellt. Neu einstellen lassen Sie sich auf die herkömmliche Art und Weise mittels ESC-Funktionen oder mit eigens dafür eingerichteten ESC-Sequenzen. Beim Verlassen eines Fensters wird die Cursor-Position zwischengespeichert; nach Einschalten dieses Fensters befindet sich der Cursor also wieder an der alten Position.

Zusätzliche Basic-Befehle für den Direktmodus

Um mit der Echtzeituhr möglichst bequem arbeiten zu können, wurden zwei zusätzliche Basic-Befehle – SET und ALARM – geschaffen, die nur im Direktmodus funktionieren.

SET

dient dem Einstellen der Uhrzeit und wird von einer sechsstelligen Angabe (ähnlich der Basic-Variablen TI\$) gefolgt. SET083410 setzt die Uhr beispielsweise auf 8 Uhr 34 und 10 Sekunden. Um eine Zeit zwischen 12 und 24 Uhr einzustellen, muß zur ersten Ziffer 8 addiert werden (Beispiel: SET914507 setzt die Uhr auf 23 Uhr 45 und 7 Sekunden, nicht jedoch SET234507!).

SET-

schaltet die Anzeige der Statuszeile ab, die Uhr läuft aber intern weiter.

SET+

dient zur Rückgängigmachung von SET-.

ALARM

stellt die Alarmzeit gemäß einer vierstelligen Angabe (ALARM0834 für Alarm um 8 Uhr 34, ALARM2345 für Alarm um 23 Uhr 45). Der Alarm dauert 1 Minute und wird mit <ESC H> abgestellt.

Zusätzliche ESC-Sequenzen

Außer den herkömmlichen ESC-Funktionen stellt das Programm »Window-Tech 80« auch noch einige weitere bereit, die zur Ausreizung seiner besonderen Möglichkeiten dienen.
<ESC ?>

fragt nach einer Fensterauswahl; durch Drücken einer Zifferntaste (0 bis 9) wird dadurch das gewünschte Fenster aktiviert. Eine andere Taste verläßt den Auswahlmodus und behält das alte Fenster bei.

< ESC >

und nachfolgendes Drücken einer Zifferntaste (0 bis 9) wählt ein Fenster direkt an.

Wird < ESC > vom Kleiner-als-Symbol (<) gefolgt, so definiert man damit die linke obere Ecke des Fensters, löscht dieses und stellt es umrandet dar.

Dasselbe bewirkt das Auslösen von < ESC > und dem Größer-als-Symbol (>) für die rechte untere Ecke.

Natürlich läßt sich die Fenstertechnik von »Window-Tech 80« auch in Basic verwenden. Auf folgende Weise wird ein beliebiges Fenster aufgerufen:

```
10 INPUT FENSTER:GOSUB 1000: REM FENSTER = 0 BIS 9
20 END
1000 PRINT CHR$(27);CHR$(48+FENSTER);:RETURN:
    REM FENSTER AUFRUFEN
```

Ebenso ist es für Maschinenprogrammierer möglich, die ESC-Sequenzen für die Fensterbehandlung mittels BSOUT oder PRIMM zu verwenden.

Eine weitere Möglichkeit besteht darin, die ESC-Sequenzen < ESC 0 > bis < ESC 9 > neu zu definieren, wobei < ESC ? > weiterhin funktioniert (!). Die Adrestabelle für die ESC-Routinen befindet sich im Speicher ab \$16C4 (\$16C4 bis \$16D3: Low-Bytes; \$16D4 bis \$16E3: High-Bytes).

Fensterparameter und -inhalte lassen sich speichern

In Basic erlaubt folgender Befehl das Speichern aller Fensterdaten einschließlich der Cursor-Koordinaten:

```
BSAVE "NAME" ON B0,P6011 TO P6071
```

Mit BLOAD "NAME" ON B0 werden diese wieder eingelesen.

Ein anderer Befehl speichert die Fensterinhalte:

```
BSAVE "NAME" ON B1, P44800 TO P65279
```

BLOAD "NAME" , ON B1 lädt diese Datei wieder.

Die Lade- und Speicherbefehle sind für Maschinenprogrammierer leicht in die Kernel-Aufrufe von SETBNK, LOAD und SAVE zu übertragen.

Abschließend wünschen wir Ihnen viel Spaß mit dieser wirklich sinnvollen Erweiterung des C 128.
(Heino Cunze/F. Müller/nj)

Centronics-Schnittstelle

Wir machen kurzen Prozeß mit den Anschlußproblemen Ihres Druckers! Mit einem leicht herstellbaren Kabel und unserem Programm können Sie mit nahezu allen Druckern problemlos am C128 arbeiten.

Der C128 besitzt mit dem Basic 7.0 einen recht umfangreichen Befehlswortschatz, der auch die hochauflösende Grafikprogrammierung unterstützt. Doch was nützt die schönste Grafik auf dem Bildschirm, wenn man sie nicht auch auf einem Drucker ausgeben kann? Wenn Sie einen Drucker mit Centronics-Schnittstelle besitzen, gehören Sie zu den Menschen, die dieses Problem nicht länger betrifft. Der abgedruckte Druckertreiber enthält eine Routine zur Umwandlung von CBM-ASCII nach Standard-ASCII, so daß im Textmodus nun weiterhin auch die deutschen Umlaute wie jedes andere Zeichen verwendet und ausgedruckt werden können. CBM-Sonder- und Grafikzeichen können jedoch nicht auf dem Drucker realisiert werden. Es ist deshalb ratsam, derartige Zeichen in eigenen Listings als CHR\$(Zeichen) darzustellen.

Der Centronics-Druckertreiber für den C128 liegt im freien RAM-Bereich von \$1300 bis \$17E3 und wird mit SYS 4864 aktiviert. Hierbei wird die Betriebssystemroutine zur Ausgabe von Zeichen auf den Treiber umgelenkt. Nach <RUN/STOP + RESTORE> oder einem Reset muß das Programm erneut aktiviert werden.

Das Treiberprogramm wird über die Sekundäradresse gesteuert. Bei Sekundäradresse Null wird der Text ohne ASCII-Wandlung ausgegeben; bei Sekundäradresse 1 bis 14 erfolgt eine Zeichensatzanpassung über eine 256 Byte lange Tabelle (\$142A bis \$1529), die auch nach eigenen Bedürfnissen angepaßt werden kann. Die Sekundäradresse 15 ist reserviert für die Ausgabe einer hochauflösenden Grafik. Durch Übertragung der Ziffern 1 oder 2 kann die Größe der Hardcopy bestimmt werden. Die größtmögliche Hardcopy (Ziffer 2) entspricht ungefähr DIN A5. Durch mehrstellige Ziffern kann die Hardcopy beliebig oft wiederholt werden. Vor jeder Hardcopy wird zur Druckjustage eine Leerzeile gedruckt.

Anpassungen

Abhängig vom Drucker oder der individuellen Anforderung des Benutzers, können einige Voreinstellungen des Treibers durch Modifikation, zum Beispiel mit dem eingebauten Maschinensprachemonitor, geändert werden. Die entsprechenden Speicherstellen und die Bedeutung der voreingestellten Inhalte sind nachfolgend beschrieben.

Speicherstelle \$1303 (dezimal 4867):

In dieser Speicherstelle ist die Gerätenummer abgelegt. Der Ursprungswert ist #4. Soll ein Drucker mit der Gerätenummer #5 angesprochen werden, so muß der Inhalt dieser Adresse entsprechend geändert werden.

Speicherstelle \$1304 (dezimal 4868)

Durch den Inhalt dieser Speicherstelle wird festgelegt, ob am Zeilenende Carriage Return (CR = Wagenrücklauf) gesendet wird (Inhalt = 0) oder Carriage Return mit Line Feed (LF = Zeilenvorschub) (Inhalt = 1). Voreingestellt ist CR und LF.

Speicherstellen \$142A bis \$1529 (dezimal 5162 bis 5417)

In diesem Speicherbereich befindet sich die Umwandlungstabelle für die Anpassung von CBM-ASCII nach Standard-ASCII. Die hier aufgeführten Werte werden bei Sekundäradresse ungleich Null anstelle der üblichen Codes an den Drucker gesendet. Zur Darstellung der Umlaute auf einem Epson-Drucker müssen die Codes über diese Tabelle gewandelt werden. Die richtigen Codes sind hierin bereits enthalten.

Speicherstellen \$1540 bis \$1544 (dezimal 5440 bis 5444)

Hier befindet sich die für Initialisierung der kleineren Standard-Hardcopy wichtige Escape-Sequenz:

1B 2A 04 40 01 (27, '*', 4, <320, >320)

Speicherstellen \$1546 bis \$154A (dezimal 5446 bis 5450)

In diesem Bereich ist die entsprechende Escape-Sequenz für die vergrößerte Hardcopy-Routine enthalten:

IB 2A 06 80 02 (27, '*', 6, <640, >640)

Speicherstelle \$156F (dezimal 5487)

In diesem Register ist der Randabstand für die Standard-Hardcopy enthalten (Voreinstellung # 20).

Speicherstelle \$1570 (dezimal 5488)

Der Randabstand für die vergrößerte Hardcopy ist in dieser Speicherstelle mit #7 voreingestellt.

Wollen Sie einen Drucker anschließen, der keine Grafik-Hardcopy beherrscht, können Sie die Hardcopy-Routinen entfernen. Dazu ist in Adresse \$1401 der Wert \$0f (dezimal 15) in \$10 (dezimal 16) zu ändern, und der Speicherbereich von \$1300 (dezimal 4864) bis einschließlich \$1529 (dezimal 5417) auf Diskette zu speichern. Auf diese Weise erhält man den reinen Druckertreiber ohne Hardcopy-Funktion.

Zum Anschluß eines Druckers mit Centronics-Schnittstelle wird ein Kabel mit Userport-Stecker und ein 36poliger Amphenol-Stecker benötigt. Als Kabel kann ein 11adriges Computer- oder ein 10adriges Fernmeldekabel mit Abschirmung verwendet werden. Die Länge des Kabels sollte 3 Meter nicht überschreiten. Die Kabel- beziehungsweise Pinbelegung können Sie Bild 14 entnehmen.

(G.M. Ritter/nj)

A	GND	16
B	Flag-Busy	11 oder
	Flag-Acknowl.	10
C	D0	2
D	D1	3
E	D2	4
F	D3	5
H	D4	6
J	D5	7
K	D6	8
L	D7	9
M	PA2-Strobe	1

Bild 14. Anschlußbelegung für ein Parallelkabel zwischen dem User-Port des C128 und einem Centronics-Drucker.

Thirty-Seconds

In nur 30 Sekunden lassen sich einseitige Disketten im 1541- und 1571-Format kopieren, und das ohne Netz und doppelten Boden, nur mit dem seriellen Kabel. Auch doppelseitige Disketten im 1571- und CP/M-Format sind für »Thirty-Seconds« kein Problem.

Für das Gespann C 64/1541 gibt es schon eine Menge schneller Kopierprogramme, die eine Diskette in annehmbarer Zeit kopieren. Dies, obwohl die Hardwarevoraussetzungen alles andere als ideal sind: Die Floppy ist nur über ein serielles Kabel mit dem Computer verbunden, so daß jedes Byte bei der Übertragung in einzelne Bits zerlegt werden muß. Auch das kodierte Aufzeichnungsverfahren des Commodore-Laufwerks ist alles andere als geeignet, Geschwindigkeitsrekorde zu brechen. Da der C 64 außerdem »nur« 64-Kbyte-RAM aufweist, ist mindestens ein dreimaliger Austausch von Quell- und Zieldiskette nötig, um eine Diskette mit 173-Kbyte-Kapazität zu kopieren.

Bei der Entwicklung des C 128 und der 1571 hat Commodore sich schließlich dazu durchgerungen, die vorhandenen Engpässe begehbarer zu machen. So befindet sich jetzt in der Floppy 1571 ebenfalls ein CIA 6526. Dieser Baustein enthält ein serielles Schieberegister, das zusammen mit einer der CIAs im C 128 den Datentransfer im seriellen Format automatisch abwickelt – ein enormer Vorteil gegenüber der 1541. Zusätzlich ermöglicht die 1571, die Taktfrequenz des eingebauten Prozessors von 1 auf 2 MHz umzuschalten, was bei der Kodierung der Daten erhebliche Vorteile bietet. Gleiches gilt auch für das 1570-Laufwerk, das zwar eine einfachere Mechanik enthält, elektronisch aber mit der 1571 übereinstimmt. Da der C 128 über 128-Kbyte-RAM verfügt, ist beim Kopieren von einseitigen Disketten nur noch ein zweimaliger Diskettenwechsel erforderlich.

Soweit zur Theorie. In der Praxis gestattet das vorliegende Programm in Verbindung mit dem C 128 und einer Floppy 1570/71, eine einseitige Diskette in 30 Sekunden vollständig zu kopieren. Mit dem 1571-Laufwerk lassen sich auch doppelseitige Disketten kopieren. In dieser Zeit sind natürlich kein Diskettenwechsel, keine Formatierung der Zieldiskette und keine Veri-

fizierung der geschriebenen Daten enthalten. Wird dies gewünscht, so werden dafür jeweils rund acht Sekunden zusätzlich benötigt, so daß 45 Sekunden je Diskettenseite bei Formatierung und Verifizierung benötigt werden.

Das Programm wird im C 128-Modus mit RUN "THIRTY-SECONDS" geladen und gestartet. Alle weiteren Diskettenoperationen beziehen sich nun auf das Laufwerk mit der Geräteadresse 8. Andere IEC-Geräte dürfen nach Herzenslust angeschlossen und eingeschaltet sein! Verwendet wird der 80-Zeichen-Bildschirm, da dieser während des Kopiervorgangs nicht abgeschaltet werden muß. Zunächst erscheint ein Menü, das drei Punkte umfaßt. Der einfachste und uninteressanteste Menüpunkt soll zuerst behandelt werden: Durch Drücken der Taste <Q> für Quit kann das Programm verlassen werden. Es wird ein Reset ausgelöst und das Programm damit beendet. Aber das Programm kann noch mehr! Mit der Directory-Funktion läßt sich das Inhaltsverzeichnis einer Diskette seitenweise anzeigen. Ein kleiner Unterschied zum DIRECTORY-Befehl des C 128 ist aber beachtenswert: Das Directory wird im 1541-Modus eingelesen. Dies hat den Vorteil, daß das Initialisieren einer einseitigen Diskette nur Bruchteile von Sekunden dauert und nicht, wie im 1571-Modus, bis zu einer halben Minute. Einziger Nachteil dieses kleinen Tricks: Bei doppelseitigen Disketten ist die Anzahl der angezeigten freien Blöcke falsch.

Der dritte und wichtigste Menüpunkt ist schließlich das Backup einer Diskette. Nach Wählen dieses Punktes werden zunächst vier Fragen gestellt, deren Bedeutung kurz besprochen werden soll. Frage Nummer eins »Single- oder Double-Sided?« betrifft das Format der Quelldiskette. Für einseitige (1541- oder 1570-)Disketten wählen Sie <S> für »Single« und bei doppelseitigen <D> für »Double-Sided«. Bei Verwendung eines 1570-Laufwerks, das ja nur einseitige Disketten kennt, ist immer <S> zu drücken. Falls Sie von vornherein nicht wissen, ob Sie ein ein- oder doppelseitiges Exemplar vor sich haben, wählen Sie <D>.

Verneinen Sie die zweite Frage »Zieldiskette formatieren?«, muß die Zieldiskette bereits formatiert sein. Wenn Sie die Frage mit »Ja« beantworten, wird die Diskette nebenbei formatiert, wofür aber acht Sekunden zusätzlich je Diskettenseite benötigt werden. Die nächste Frage »Zieldiskette verifizieren?« sollten Sie eigentlich immer mit »Ja« beantworten. In diesem Fall wird nämlich kontrolliert, ob die Daten auch richtig geschrieben werden. Risikofreudige Anwender können »Nein« eingeben und sparen dadurch acht Sekunden. Die letzte Frage

»Mehrere Leseversuche?« betrifft die Anzahl der anzustellenden Leseversuche. Geben Sie »Ja« ein, so ist »Thirty-Seconds« so hartnäckig wie das Laufwerk im Normalbetrieb. Wollen Sie verhindern, daß beim Lesen defekter Sektoren viel Zeit verlorenght, so tippen Sie »Nein«.

Bei allen Fragen sind bereits Standardwerte vorgegeben. Diese sind: doppelseitige Diskette, Formatierung, Verifizierung und mehrere Leseversuche. Sind Sie mit diesen Vorgaben einverstanden, so brauchen Sie nur viermal <RETURN>, andernfalls an entsprechender Stelle die geeignete Taste zu drücken.

Nun werden Sie durch eine Bildschirmmeldung und ein verschämtes Flackern der Floppy-Lampe aufgefordert, die Quelldiskette einzulegen. Auf Tastendruck beginnt der Kopiervorgang. Das Einlesen einer halben Diskettenseite dauert bei einer normalen, fehlerfreien Diskette knapp acht Sekunden. Treten allerdings Leseprobleme auf, kann es auch wesentlich länger dauern. Der Schreibvorgang nimmt im Normalfall acht bis fünfzehn Sekunden in Anspruch, abhängig von den eingestellten Parametern. Falls bei der Verifizierung Fehler festgestellt werden, wird der ganze Track nochmals formatiert und geschrieben. Dieser Vorgang wird bis zu fünfmal ausgeführt. Treten mehr als fünf Schreibfehler auf, so wurde entweder keine oder eine schreibgeschützte Diskette eingelegt.

Nun noch zu einer Funktion, die »Thirty-Seconds« von vielen anderen Kopierprogrammen unterscheidet: Durch Drücken der <RESTORE>-Taste kann man jederzeit, auch während des Kopiervorgangs, ins Hauptmenü gelangen. Das Laufwerk stellt ebenfalls alle Operationen ein. Wenn vor Abbruch mit <RESTORE> Leseprobleme aufgetreten sind, kann dies auch etwas länger dauern. In diesem Fall können Sie den Vorgang durch Aus- und Einschalten der Floppy oder durch Drücken des Reset-Tasters abkürzen.

Diese nützliche Funktion läßt sich auch verwenden, wenn Sie bei einer einseitigen Diskette »doppelseitig« angewählt haben. Nach dem dritten Einlegen der Quelldiskette werden dann Lesefehler angezeigt. Unterbrechen Sie in diesem Fall den Kopiervorgang mit <RESTORE>, da man sich das Kopieren der zweiten Seite schenken kann und die lauffähige Kopie bereits in Händen hält. (Reiner Koukal/ah/nj)

Double-Touch

Da werden sogar die Besitzer von Personalcomputern neidisch. »Double-Touch« kopiert mit zwei Laufwerken eine Diskette in sage und schreibe 8 Sekunden.

Die vielfältigen Möglichkeiten bei der Programmierung der Floppy 1570/71 fordern regelrecht dazu heraus, schnelle Kopierprogramme für diese Laufwerkstypen zu schreiben. Ganz besonders interessant wird es, wenn neben dem C 128 gleich zwei Laufwerke zur Verfügung stehen. Dann kann man sich nicht nur das lästige und zeitraubende Diskettenwechseln ersparen, sondern erzielt auch eine enorme Geschwindigkeit. »Double-Touch« kopiert eine Diskette nämlich in rund 8 (!) Sekunden und formatiert sie auch noch gleichzeitig. Wünschen Sie die Verifizierung der Kopie, so werden weitere 8 Sekunden benötigt, die jedoch jeder Anwender aus Gründen der Datensicherheit opfern sollte.

Verwendet man zwei 1571-Laufwerke, so können auch doppelseitig bespielte Disketten (aber keine doppelseitigen 1541-Formate!) bei entsprechend verdoppeltem Zeitaufwand in einem Durchgang kopiert werden.

Komfortable Menüsteuerung

»Gemischter« Betrieb mit einer 1570- und einer 1571-Floppy ist natürlich auch möglich; man ist dann, wie bei Verwendung zweier 1570-Laufwerke, auf einseitige Disketten beschränkt. Mit der Floppy 1541 arbeitet »Double-Touch« allerdings nicht zusammen.

Starten des Programms: Double-Touch wird mit

```
RUN "DOUBLE-TOUCH"
```

geladen und gestartet. Zum Betrieb von Double-Touch ist ein 80-Zeichen-Monitor erforderlich.

Der erste der vier Menüpunkte, »DIRECTORY ANZEIGEN«, gibt das Inhaltsverzeichnis aus, wahlweise von Laufwerk 8 oder 9. Das Directory wird, um die Initialisierung bei einseitigen Disketten abzukürzen, im 1541-Modus eingelesen. Dadurch wird bei doppelseitigen Disketten allerdings ein falscher Wert als Anzahl der freien Blöcke angezeigt.

Mit »BACKUP DISK« kopiert man eine Diskette. Bei den drei Eingabeaufforderungen, die nun folgen, sind bereits Standardwerte vorgegeben, die mit <RETURN> übernommen werden können.

Kopierparameter einstellen

»SINGLE- ODER DOUBLE-SIDED« wählt das Diskettenformat, wobei beim Arbeiten mit einer 1570-Floppy nur »SINGLE-SIDED« in Frage kommt.

Die zweite Frage, »ZIELDISKETTE VERIFIZIEREN?«, sollte üblicherweise mit »J« beantwortet werden, um eine fehlerfreie Kopie zu gewährleisten. Die Frage »GERÄTEADRESSE ÄNDERN« muß nur dann bejaht werden, wenn die beiden verwendeten Laufwerke auf Geräteadresse 8 eingestellt sind. Dann wird vor dem Kopiervorgang die Adresse eines Laufwerks softwaremäßig von 8 auf 9 geändert. Dies muß allerdings vor jedem Kopiervorgang geschehen. Eine hardwaremäßige Einstellung der Geräteadressen ist daher zu empfehlen.

Nun beginnt der eigentliche Kopiervorgang. Es erfolgt nacheinander die Aufforderung, Quell- und Zieldiskette einzulegen. Die LEDs (Leuchtdioden) des Laufwerks zeigen jeweils an, wo die Disketten einzulegen sind, eine Verwechslung dürfte daher schwerfallen.

Während des Kopierens können auch Fehlermeldungen auftreten. »WRITE PROTECT ON« bedeutet, daß sich ein Schreibschutz auf der eingelegten Diskette befindet oder daß noch keine Schreibkerbe angebracht wurde und führt zum sofortigen Abbruch des Kopiervorgangs. »x SECTORS MISSING ON TRACK y« zeigt an, daß die Anzahl der eingelesenen Sektoren einer Spur auf Diskette nicht stimmt; Ursache hierfür ist ein Read-Error 20 oder 21. Andere Lesefehler (Read-Error 22, 23, 24, 27 und 29) werden wie die beiden ersten mitkopiert, ohne jedoch erkannt zu werden. Ein Kopieren von kopiergeschützten Disketten ist trotzdem nicht empfehlenswert, da bestimmte Fehler der Quelldiskette, die normalerweise als »READ ERROR 21« angegeben werden, zur Zerstörung weiterer Sektoren führen können.

So werden Fehler behandelt

Weiteren Aufschluß über Fehler erhält man nur bei angewähltem Verifizieren der Zieldiskette. Hier wird versucht, die kopierte Spur einzulesen. Sektoren, bei denen dies nicht einwandfrei gelingt, werden mit »VERIFY ERROR« angezeigt. Tritt ein solcher Fehler auf, so kann dies auch an einer fehlerhaften oder kopiergeschützten Queldiskette liegen. Da »Double-Touch« nur einen Leseversuch vornimmt, empfiehlt es sich, in einem solchen Fall ein anderes – langsames – Kopierprogramm, das mehrere Leseversuche durchführt, oder einen Nibbler zu verwenden. Nun ja, Schnelligkeit hat eben seinen Preis!

Ein seltener Fehler ist »BUFFER OVERFLOW ON TRACK x«. Manchmal ist ein Read-Error 21 die Ursache, aber auch bei fehlerfreien Disketten kann dieser Fehler auftreten. Das Ergebnis ist jedoch – wie oben – immer dasselbe: Die betroffene Spur wurde nicht richtig kopiert und einige oder alle Sektoren des Tracks sind daher fehlerhaft.

Als Abhilfe kann man bei Problemdisketten zunächst versuchen, die Diskette nochmals zu kopieren. Tritt der Fehler wieder auf, hilft möglicherweise (beispielsweise, wenn die Köpfe beider Laufwerke leicht unterschiedlich justiert sind) der Hauptmenü-Punkt »QUELLE UND ZIEL VERTAUSCHEN«, womit man das Laufwerk mit der Geräteadresse 9 als Quell-Laufwerk definiert (bitte auch Disketten austauschen): Entsteht der Fehler auch bei erneutem Versuch, so muß man sich wohl oder übel damit abfinden, daß sich die betroffene Diskette mit »Double-Touch« nicht kopieren läßt. In der Regel ist die Anzahl solch hartnäckiger Disketten aber so verschwindend gering, daß dies keinen Nachteil darstellt.

Das Hauptmenü hält schließlich noch einen weiteren Menüpunkt bereit: Nach »QUIT« findet man sich kurzerhand im C 128-Modus wieder; sollte das Diskettenlaufwerk anschließend nicht ansprechbar sein, hilft ein Hardware-Reset (Reset-Schalter betätigen) oder kurzzeitiges Abschalten des Diskettenlaufwerks weiter.

Hinweis: Double-Touch läuft auf den neuen C 128D-Modellen aufgrund des geänderten Betriebssystems nicht! Zum Schluß sei noch erwähnt, daß man durch Drücken von <RESTORE> jederzeit ins Hauptmenü gelangt. Geschieht dies mitten in einem Kopiervorgang, so beenden die Laufwerke unverzüglich ihre Tätigkeit.

(Reiner Koukal/Florian Müller/nj)

.

CP/M-Formatter

Was haben zwei Programme im C128-Modus mit neuen Diskettenformaten unter CP/M 3.0 zu tun? Diese Frage könnten Sie sich beim Anblick der Listings stellen. Lesen Sie, welche ungeahnten Möglichkeiten diese beiden Programme eröffnen – oder noch besser: Probieren Sie sie doch gleich aus!

Wollten Sie schon einmal CP/M-Disketten beispielsweise im Osborne- oder Kaypro-Format herstellen, um über etwa 60 Kbyte mehr freien Speicherplatz je Diskette zu verfügen? Oder aber mit der Commodore-1571-Floppy tatsächlich nur eine Diskettenseite formatieren? Würden Sie gerne einen Blick in das Format der eingelegten Diskette werfen, etwa welche Größe die Disk-Sektoren in einem beliebigen Track besitzen und wie viele davon vorhanden sind, ob auch die Rückseite der Diskette formatiert ist und vieles mehr? Wenn ja, dann kommen Sie in diesem Artikel voll auf Ihre Kosten.

Formatieren von Fremdformaten...

Das erste Programm »CP/M-FORMATTER« formatiert alle GCR- (Commodore, 35 und 40 Tracks) und einige MFM-Formate, die der C128 unter CP/M 3.0 unterstützt. Die Auswahl des zu formatierenden Formats geschieht über ein kleines Menü mit den Cursor-Steuertasten und der <RETURN>-Taste. Formatiert wird nur auf den Commodore-Floppy-Laufwerken 1570 und 1571, zweiseitige Formate können nur mit dem Laufwerk 1571 hergestellt werden. Zur Auswahl momentan schon im Programm aufgenommener und unter CP/M 3.0 les- und schreibbarer Formate stehen Kaypro IV (beidseitig), Osborne (beidseitig), IBM-8 (einseitig), Epson QX-10 (beidseitig) sowie C64 (einseitig), C128 einseitig und schließlich C128 beidseitig. In Tabelle 9 sind jedoch schon die Angaben enthalten, um weitere 78 neue Fremdformate herstellen zu können. Wie diese Angaben umgesetzt werden, wird später ausführlich erläutert werden.

Wenn Sie nun denken, daß das Programm »Format« der CP/M-Systemdiskette zumindest die drei Commodore-(GCR-)Formate bewältigt, so ist dies nur zum Teil korrekt. Formatiert man nämlich auf einer 1571-Floppy das Format C64 oder C128 single sided, so wird die Diskette

zwar nur für einseitigen Betrieb vorbereitet, jedoch auf beiden Seiten formatiert. Informationen auf der Rückseite gehen also unnötigerweise verloren, was teilweise sehr ärgerliche Folgen hat. Der »CP/M-Formatter« formatiert hingegen nur die Seiten, die auch später benutzt werden. Beim Formatieren der Diskette im C 64- oder C 128-Single-Sided-Modus bleibt also die Rückseite »verschont«. Hierfür wird die 1571-Floppy in die 1541-Betriebsart geschaltet, was sich leider bei der Formatiergeschwindigkeit bemerkbar macht.

Unter CP/M kann dann die in einem dieser Fremdformate »vorbereitete« Diskette beschrieben (zum Beispiel mit dem Kopierprogramm »pip«) und danach beispielsweise von einem Kaypro eingelesen werden. Damit ist zum einen der Datenaustausch zwischen beiden Computer-Typen gewährleistet, zum anderen kann sich der Anwender die Vorteile von Fremdformaten zunutze machen. Manche Formate lassen sich sehr schnell formatieren, andere bieten um einiges mehr Speicherplatz auf Diskette als die Commodore-Formate (etwa Kaypro IV mit 394 Kbyte anstelle von 340 Kbyte). In Tabelle 10 ist die Speicherkapazität der einzelnen mit dem CPM-Formatter herstellbaren Formate aufgeführt. Diese Fremdformate können problemlos von den Laufwerken 1570 (alle einseitigen Formate) und 1571 unter dem Betriebssystem CP/M verarbeitet werden. Sofern sich doch ein »BDOS-Error« einschleicht, muß man lediglich durch <CTRL C> die Laufwerksparameter zurücksetzen. Das entsprechende Format wird unter CP/M auf dem Bildschirm unten links eingeblendet und muß mit <RETURN> bestätigt werden. Sofern mehrere Formate zur Auswahl stehen, kann mit den Pfeil-Tasten nach links und rechts in der obersten Tastaturreihe des C 128 gewechselt werden. Dies ist beim Epson- und Kaypro-Format der Fall. Einem Software-Austausch zwischen allen oben genannten Systemen steht also nichts mehr entgegen.

...oder aber Informationen hierzu

Das zweite Programm heißt »MFM-SCAN«. Es läuft ebenfalls auf dem C 128 im C 128-Modus wahlweise mit der Floppy 1570 oder 1571.

Als einzige Eingaben sind die Seite (1 für vorne, 2 für hinten) und die physikalische Spur (0 bis 39) nötig. Das Programm meldet dann die »Zusammensetzung« der Diskette:

Wird eine C 64- oder C 128-Diskette (auch CP/M) eingelegt, erscheint die Meldung, daß es sich um GCR-Format handelt. Noch nicht formatierte Disketten können ebenfalls erkannt werden. Bei MFM-Formaten (dafür ist das Programm ja auch gedacht), werden

- erste logische Spurnummer
- letzte logische Spurnummer
- Sektoranzahl
- Sektorgröße (in Byte) und die
- Sektorfolge

der gewählten Diskettenseite und Spur angezeigt.

Zunächst jedoch ein paar grundsätzliche Informationen zum physikalischen Aufbau einer MFM-Diskette.

Was ist MFM?

Der Begriff MFM-formatiert ist eigentlich nicht ganz korrekt, da MFM (Modified Frequency Modulation) sich nur auf die Art und Weise bezieht, in der die einzelnen Bits physikalisch auf der Diskette aufgezeichnet werden. Wie nun die Bits zu Sektoren und diese wiederum zu Spuren zusammengefaßt werden, ist nicht durch das MFM-Aufzeichnungsformat, sondern durch den IBM-34-Standard festgelegt. Eine Diskette wird in eine gewisse Anzahl von Spuren unterteilt. Die Spurenanzahl hängt vom verwendeten Laufwerk ab. Bei den Floppylaufwerken 1570 und 1571 beträgt sie 40, manche Laufwerke anderer Computer können sogar 80 Spuren (einseitig) verwalten. Bei Laufwerken mit zwei Schreib-Lese-Köpfen (wie zum Beispiel der 1571) unterscheidet man noch zwischen der Vorder- und Rückseite, die jede für sich 40 Spuren aufnehmen kann. Man spricht in diesem Zusammenhang von »Double-Sided« im Gegensatz zu Laufwerken mit nur einem Schreib-Lese-Kopf (zum Beispiel 1570), die auch als »Single-Sided« bezeichnet werden.

Auf der Spur

Jede Spur wird in eine feste Anzahl von gleich großen Sektoren unterteilt, wobei jedem Sektor ein Sektor-Header vorausgeht. In diesem »Vorspann« sind die (logische) Sektor-Nummer und die Sektor-Größe des nachfolgenden Sektors sowie die (logische) Spur- und die (physikalische) Seitennummer vermerkt.

Der Unterschied zwischen einer logischen und einer physikalischen »Nummer« besteht darin, daß sich die physikalische »Nummer« an Hardware-Vorgaben orientiert, die logische jedoch nicht. Ein Beispiel: Die Spuren, die die Hardware der 1570/1571 mit dem Schreib-Lese-Kopf anfahren kann, sind von »0« (außen) bis »39« (innen) durchnummeriert. Die Zählweise der Spuren in dem oben besprochenen Sektorheader kann jedoch zum Beispiel mit »1« beginnen und bei »40« enden. Die Spur, die auf den Disketten ganz außen liegt, hätte also die physikalische Nummer »0« (Vorgabe durch die Hardware der 1570/1571) und die logische (sprich aufgezeichnete) Nummer »1«.

Ähnlich verhält es sich mit den Sektoren: Innerhalb einer Spur kann der Sektor mit der niedrigsten Nummer (der physikalische Sektor »0«) die logische Nummer »10« haben (zum Beispiel auf der Rückseite einer Kaypro-IV-Diskette).

Ein anderes Kennzeichen einer IBM-34-Diskette ist der Sektorversatz (auch Sector-Skewfactor, Interleavingfactor oder Sector-Shift-Factor genannt). Der Sektorversatz berechnet sich aus der Reihenfolge, in der die einzelnen Sektoren innerhalb einer Spur aufeinanderfolgen. Untersucht man mit MFM-Scan zum Beispiel eine Kaypro-IV-Diskette, so wird man auf der Vorderseite folgende Sektorfolge feststellen:

00 05 01 06 02 07 03 08 04 09

Die Sektorenfolge kann natürlich auch mit einem anderen Sektor beginnen, da MFM-Scan immer den Sektor, den es zuerst findet, auch zuerst auswertet. Der Sektorversatz ist nun der Abstand zwischen zwei Sektoren, deren logische Nummern aufeinander folgen, der letzte logische Sektor jedoch ausgenommen. Sehen wir uns nun die Sektorenfolge genauer an. Auf »00« folgt »05«, dann »01«, »06«, »02« etc. Zwischen zwei aufeinanderfolgenden Zahlen steht immer ein Wert. Bei Kaypro IV ist der Sektorversatz also eins. Bei »Osborne DD« (Sektorfolge: 01 02 03 04 05) wäre der Sektorversatz null.

Was bringt der Sektorversatz?

Doch was soll dieser Sektorversatz überhaupt?

In den Commodore-Floppies 1570/1571 befindet sich ein eigener Mikroprozessor, der die Controller für das normale CBM-Format sowie für MFM bedient und die Daten dann über den seriellen Bus an den C 128 überträgt (deshalb spricht man auch von einem intelligenten Peripheriegerät). Andere Hersteller lassen die Floppycontroller direkt vom Hauptprozessor bedienen. Die dabei erreichten Datenübertragungsgeschwindigkeiten liegen im selben Bereich, die die neuesten Floppyspinner mit der guten alten, aber nicht sonderlich schnellen 1541 erreichen. Würde nun ein Sektor eingelesen, so muß dieser von CP/M erst noch verarbeitet (zum Beispiel umkopiert) werden. In dieser Zeit dreht sich die Diskette allerdings weiter. Würde man auf der Diskette auf Sektor eins gleich Sektor zwei folgen lassen, so müßte der Controller eine ganze Umdrehung warten, bis er den gewünschten Sektor einlesen kann. Liegt aber zwischen Sektor eins und zwei noch ein anderer, so kann der eingelesene Sektor verarbeitet werden, während sich der »Zwischensektor« unbeachtet unter dem Schreib-Lese-Kopf des Laufwerks »vorbeidreht«. Anschließend kann der Diskcontroller Sektor zwei einlesen, ohne eine Diskettenumdrehung warten zu müssen. Zum einwandfreien Arbeiten muß allerdings der Sektorversatz auf die Verarbeitungszeit eines Sektors abgestimmt sein! Dauert zum Beispiel das Umkopieren eines Sektors etwas länger, so müssen sich zwischen den beiden logisch aufeinanderfolgenden Sektoren mehrere »Zwischensektoren« befinden, das heißt, der Sektorversatz wird größer. Bei der 1570/1571 bringt der Sektorversatz leider keinen Geschwindigkeitsvorteil, weil wesentlich mehr Zeit zum Übertragen eines Sektors zum Computer vergeht, als durch den Sektorversatz zur Verfügung steht. Trotzdem muß man seine MFM-Disketten auf der 1570/1571 mit einem Sektorversatz formatieren, weil viele CP/M-Implementationen den Sektorversatz dazu benutzen, ein IBM-34-Diskettenformat zu erkennen (so auch das CP/M 3.0 des C 128). Soviel zur Bedeutung der Ausgabedaten von MFM-Scan.

Die Erweiterung des CP/M-Formatter

Um nun den CP/M-Formatter zu erweitern, muß zunächst das Format in einer neuen DATA-Zeile ab 1171 definiert werden.

Ein DATA-Eintrag ist wie folgt aufgebaut:

- **Formatname:** Name des Formats, erscheint in der Menüauswahl;
- **Seitenzahl:** Anzahl der zu formatierenden Seiten (eine oder zwei);
- **1. Spurnummer vorne:** logische Nummer der physikalischen Spur »0« auf der Vorderseite;
- **1. Spurnummer hinten:** logische Nummer der physikalischen Spur »0« auf der Rückseite;
- **Spuranzahl:** Anzahl der zu formatierenden Spuren, grundsätzlich »40«;
- **1. Sektor-Nummer vorne:** erste logische Sektor-Nummer auf der Vorderseite;
- **1. Sektor-Nummer hinten:** erste logische Sektor-Nummer auf der Rückseite;
- **Sektoren-Anzahl:** Anzahl der Sektoren (muß auf Vorder- und Rückseite gleich sein);
- **Sektor-Größe:** Kennung für Sektor-Größe: »0« bei 128, »1« bei 256, »2« bei 512 und »3« bei 1024 Byte je Sektor;
- **Skew-Factor:** Der aus der Sektortabelle – wie oben beschrieben – berechnete Sektorversatz plus eins.

Erweiterung in DATA-Zeilen

Außerdem muß der Startinitialisierungswert der Variablen »FZ« in Zeile 80 für jeden neuen DATA-Eintrag um eins erhöht werden (FZ bedeutet Formatzahl). Die Werte für die Einträge in die DATA-Zeilen erhält man aus der Untersuchung einer Diskette, die im gewünschten Format formatiert wurde. Dazu untersucht man mit MFM-Scan jeweils Spur »0« der Vorder- und der Rückseite. Außer der Seitenanzahl lassen sich alle Parameter direkt mit MFM-Scan ermitteln. Ist die Seitenzahl nicht bekannt, so kann man sich wie folgt behelfen: Disketten, die

auf Vorder- und Rückseite unterschiedliche Sektoren-Anzahl und/oder Sektor-Größe haben, oder deren Rückseite un- oder GCR-formatiert ist, sind nur »einseitig«. Die Ergebnisse der Vorderseite sind dann in die DATA-Zeile wie folgt einzutragen: Die erste Spurnummer hinten und erste Sektor-Nummer hinten wird auf null gesetzt, die Seitenanzahl entsprechend auf eins.

Wer sicher gehen will, untersucht die Diskette unter CP/M 3.0: CP/M durch Reset mit eingelegter Systemdiskette hochfahren, die Rückseite der Systemdiskette beziehungsweise Ihre Arbeitsdiskette (mit dem Dienstprogramm »Show«) einlegen, und das BDOS durch <CTRL C> zurücksetzen.

Wenn Sie mit einem Laufwerk arbeiten, lautet die Anweisung »SHOW E: [DRIVE]« (E: ist das virtuelle Laufwerk). Nachdem die zu untersuchende Diskette in das Laufwerk gelegt und <RETURN> gedrückt wurde, gibt Show die Diskettenparameter aus. Für die Bestimmung der Seitenanzahl ist die Angabe »Kilobyte Drive Capacity« (Gesamtkapazität der Diskette) von Bedeutung. Ist der angegebene Wert deutlich größer als 200, so handelt es sich um eine zweiseitige, ansonsten nur um eine einseitige Diskette. Um Ihnen Mühe und Arbeit zu ersparen, haben wir in Tabelle 1 von 78 Fremdformaten, die mit dem Laufwerk 1571 les-, schreib- und formatierbar sind, die notwendigen Formatierungsdaten zusammengestellt (durch MFM-Scan). Aus den meisten Formatnamen geht schon hervor, ob sie ein- oder beidseitig arbeiten (»ss« und »ds«). Grundsätzlich sind sie jedoch einseitig. Sie sind damit in der Lage, alle diese Formate herzustellen und den CP/M-Formatter nach eigenen Wünschen zu erweitern. Der nächste Schritt besteht nun darin, unter CP/M 3.0 die noch nicht definierten Formate einzufügen und sie so ins System zu integrieren. Am Ende sei noch kurz darauf hingewiesen, daß man auch den MFM-Scan noch erweitern kann, beispielsweise daß er zusätzlich das Format in Klartext ausgibt (etwa Epson QX-10 oder IBM-8). Dies läßt sich durch Vergleich mehrerer Konstanten eines Formats mit Tabellen aller möglichen Formate erreichen. Lassen Sie Ihrer Fantasie freien Lauf und nutzen bislang schlummernde Fähigkeiten der Laufwerke 1570 und 1571.

(Steffen Stempel/bj)

Formatname	logische Spaltennummer	erste logische Sektornummer	letzte logische Sektornummer	Sektoranzahl	Sektorgröße (Byte)	Sektorfolge
ACCESS MATRIX SS	0	1	9	9	512	04 05 06 07 08 09 01 02 03
ALPHATRONIC PC	0	1	16	16	256	06 07 08 09 0A 0B 0C 0D 0E 0F 10 01 02 03 04 05
AMPRO 48 TPI SS	0	1	10	10	512	04 05 06 07 08 09 0A 01 02 03
AMPRO 48 TPI DS	0	17	26	10	512	14 15 16 17 18 19 1A 11 12 13
ASSOCIATE	0	1	10	10	512	07 03 08 04 09 05 0A 01 06 02
ATR 8000	0	1	5	5	1024	04 05 01 02 03
AVATAR TC 10 48 TPI	0	1	10	10	512	04 05 06 07 08 09 0A 01 02 03
BEEHIEVE TOPPER	0	1	10	10	512	04 05 06 07 08 09 0A 01 02 03
CALIFORNIA COMP. SYSTEMS	0	1	5	5	1024	04 05 01 02 03
COLUMBIA MPC CP/M 80	0	1	8	8	512	04 05 06 07 08 01 02 03
COLUMBIA M964	0	1	10	10	512	04 05 06 07 08 09 0A 01 02 03
CROMENCO CDOS SS	0	1	10	10	512	04 05 06 07 08 09 0A 01 02 03
CROMENCO CDOS DS	0	1	10	10	512	04 05 06 07 08 09 0A 01 02 03
DATAVUE DV 80	0	129	138	10	512	84 85 86 87 88 89 8A 81 82 83
DEC VT 18X	0	1	9	9	512	04 05 06 07 08 09 01 02 03
EPSON QX10 CP/M EARLY	0	1	16	16	256	06 07 08 09 0A 0B 0C 0D 0E 0F 10 01 02 03 04 05
EPSON QX10 VALDOC	0	1	10	10	512	02 06 09 03 07 0A 04 01 05 08
EPSON MULTIFONT HX20	0	1	16	16	256	06 07 08 09 0A 0B 0C 0D 0E 0F 10 01 02 03 04 05
GROUPIL 3 CP/M	0	1	4	4	1024	04 01 02 03
HPI25 CP/M	0	0	15	16	256	05 06 07 08 09 0A 0B 0C 0D 0E 0F 00 01 02 03 04
IBM-PC CP/M 86 SS	0	1	8	8	512	04 05 06 07 08 01 02 03
IBM-PC CP/M 86 DS	0	1	8	8	512	04 05 06 07 08 01 02 03
IDEA BITELEX	0	1	16	16	256	06 07 08 09 0A 0B 0C 0D 0E 0F 10 01 02 03 04 05
I.E.S.I.	0	1	16	16	256	06 07 08 09 0A 0B 0C 0D 0E 0F 10 01 02 03 04 05
I.M.S. 5000 SS	0	1	16	16	256	06 07 08 09 0A 0B 0C 0D 0E 0F 10 01 02 03 04 05
I.M.S. 5000 DS	0	1	16	16	256	06 07 08 09 0A 0B 0C 0D 0E 0F 10 01 02 03 04 05
INTERSIL 35 TK DS	0	1	18	18	256	06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 01 02 03 04 05
ISB-80C	0	1	8	8	512	04 05 06 07 08 01 02 03
ISB 80/85	0	1	18	18	256	06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 01 02 03 04 05
I.S.M. CP/M	0	1	10	10	512	04 05 06 07 08 09 0A 01 02 03
ITT 3030	0	1	16	16	256	06 07 08 09 0A 0B 0C 0D 0E 0F 10 01 02 03 04 05
KAYPRO II	0	0	9	10	512	06 01 09 04 07 02 05 00 08 03
LNW CP/M 40TK SS	0	1	18	18	256	06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 01 02 03 04 05
LOBO CP/M	0	0	17	18	256	0D 02 08 0E 03 09 0F 04 0A 10 05 0B 11 00 06 0C 01 07
LOBO CP/M DS	0	0	17	18	256	05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 00 01 02 03 04
LOBO CP/M 512	0	0	9	10	512	03 04 05 06 07 08 09 00 01 02
LOBO CP/M 512 DS	0	0	9	10	512	03 04 05 06 07 08 09 00 01 02
MAGIC	0	1	10	10	512	04 05 06 07 08 09 0A 01 02 03
MICRAL 9050 CP/M 80	0	1	16	16	256	06 07 08 09 0A 0B 0C 0D 0E 0F 10 01 02 03 04 05
MORROW DESIGNS SS	0	1	5	5	1024	04 05 01 02 03

Tabelle 9. Parameterliste wichtiger Fremdformate

Formatname	logische Spaltennummer	erste logische Sektornummer	letzte logische Sektornummer	Sektoranzahl	Sektorgröße (Byte)	Sektorfolge
MORROW DESIGNS SS	0	1	5	5	1024	04 05 01 02 03
MORROW DESIGNS DS 128 ENTRY	0	1	5	5	1024	04 05 01 02 03
NEC PC8001A SSDD CP/M	0	1	16	16	256	06 07 08 09 0A 0B 0C 0D 0E 0F 10 01 02 03 04 05
NEC PC8801A DSDD CP/M	0	1	16	16	256	08 03 0E 09 04 0F 0A 05 10 0B 06 01 0C 07 02 0D
NOVELL	0	1	16	16	256	07 08 09 0A 0B 0C 0D 0E 0F 10 01 02 03 04 05 06
OLIVETTI M20 CP/M	0	1	16	16	256	06 07 08 09 0A 0B 0C 0D 0E 0F 10 01 02 03 04 05
OLIVETTI ETV 300 CP/M	0	1	18	18	256	06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 01 02 03 04 05
OLYMPIA ETX II	0	1	9	9	512	04 05 06 07 08 09 01 02 03
OLYMPIA ETX 100	0	1	9	9	512	04 05 06 07 08 09 01 02 03
OSBORNE DD	0	1	5	5	1024	04 05 01 02 03
OSMOSIS DD	0	1	10	10	512	04 05 06 07 08 09 0A 01 02 03
OTTRONA ATTACH	0	1	10	10	512	07 03 08 04 09 05 0A 01 06 02
PEGASUS	0	1	9	9	512	04 05 06 07 08 09 01 02 03
PMC MICROMATE DSDD 40 TRACK	0	1	5	5	1024	04 05 01 02 03
REYNOLDS & REYNOLDS TC 1000	0	0	4	5	1024	01 04 02 00 03
SANYO MBC 1000	0	1	16	16	256	06 07 08 09 0A 0B 0C 0D 0E 0F 10 01 02 03 04 05
SHARP	0	1	16	16	256	06 07 08 09 0A 0B 0C 0D 0E 0F 10 01 02 03 04 05
SIEMENS 1610 CP/M	0	1	9	9	512	04 05 06 07 08 09 01 02 03
SYSTEL II SS	0	1	9	9	512	04 05 06 07 08 09 01 02 03
SYSTEL III DS	0	1	9	9	512	04 05 06 07 08 09 01 02 03
TELETEK	0	1	18	18	256	06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 01 02 03 04 05
TELEVIDEO TS 802/803/806	0	1	18	18	256	10 02 05 08 0B 0E 11 03 06 09 0C 0F 12 01 04 07 0A 0D
TI CP/M 80	0	1	8	8	512	04 05 06 07 08 01 02 03
TOSHIBA T100	0	1	16	16	256	06 07 08 09 0A 0B 0C 0D 0E 0F 10 01 02 03 04 05
TRS80 MOD 3 FEC CP/M	0	1	18	18	256	06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 01 02 03 04 05
TRS80 MOD 3 FEC T80S	0	1	18	18	256	06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 01 02 03 04 05
TRS80 MOD 3 HOLMES	0	0	9	10	512	06 01 09 04 07 02 05 00 08 03
TRS80 MOD 3 HURRICANE LABS	0	1	5	5	1024	04 05 01 02 03
TRS80 MOD 4	0	1	18	18	256	06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 01 02 03 04 05
MONTEZUMA MICRO 1.44	0	1	18	18	256	06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 01 02 03 04 05
TRS80 MOD 4 MONTECUMA MICRO	0	1	18	18	256	06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 01 02 03 04 05
TURBODOS	0	1	5	5	1024	04 05 01 02 03
WANG MAWS CP/M	0	1	16	16	256	06 07 08 09 0A 0B 0C 0D 0E 0F 10 01 02 03 04 05
XEROX 820 II SS	0	1	17	17	256	0E 03 09 0F 04 0A 10 05 0B 11 06 0C 01 07 0D 02 08
ZENITH Z90 W/Z37 SS 8RBL	0	1	16	16	256	08 03 0E 09 04 0F 0A 05 10 0B 06 01 0C 07 02 0D
ZENITH Z90 W/CDR CP/M	0	1	10	10	512	04 05 06 07 08 09 0A 01 02 03
ZENITH Z100 WMAGNOLIA CP/M	0	1	9	9	512	04 02 09 07 05 03 01 08 06
ZENITH Z100 CP/M SS	0	1	8	8	512	04 05 06 07 08 01 02 03
ZENITH Z100 CP/M DS	0	1	8	8	512	04 05 06 07 08 01 02 03

Tabelle 9. Parameterliste wichtiger Fremdformate (Schluß)

Format	Kapazität in Kbyte
C 64 Single Sided	136
C 128 Single Sided	170
C 128 Double Sided	340
Kaypro II	195
Kaypro IV	394
Osborne Single Sided	185
IBM-8 Single Sided	156
Epson QX-10 Double Sided	304

Tabelle 10. Speicherkapazitäten der gängigsten CP/M-Formate auf dem C128.

Uniboot

Die Bootroutine des C 128 kann mehr als die meisten Besitzer dieses Computers vermuten. »Uniboot« macht ihre verborgenen Fähigkeiten jedem Anwender auf einfache Weise zugänglich. Zusätzlich wird der Aufbau des Bootsektors verständlich erklärt.

Nach einem Reset beziehungsweise nach dem Einschalten des C 128 liest die Bootroutine automatisch den Sektor 0 auf Spur 1 der eingelegten Diskette. Soll nun tatsächlich ein Bootvorgang stattfinden, muß eine spezielle Codierung (»CBM« in den ersten drei Bytes des Sektors) vorhanden sein. Ist dies der Fall, werden vom Betriebssystem weitere Blocks oder gegebenenfalls ein Programmfile geladen. Anschließend startet der C 128 ein Maschinenprogramm, das den Rest des Bootsektors belegen kann. Uniboot hilft dem Anwender beim Erstellen solcher Bootsektoren. Dabei wurde großer Wert auf eine hohe Flexibilität gelegt. Ein professioneller Softwareschutz läßt sich ebenso einfach realisieren wie individueller Programmierkomfort. Im ersten Fall schaltet Uniboot die Unterbrechungsfunktionen (Run/Stop, Restore) sowie den Reset ab und startet ein nachgeladenes Programm-File. Im zweiten Fall holt Uniboot eine individuelle Funktionstastenbelegung sowie neue Farbwerte von Disk und aktiviert sie. In Tabelle 11 finden Sie die vielen Möglichkeiten von Uniboot noch einmal übersichtlich zusammengefaßt.

Das »Geheimnis« dieser Vielseitigkeit liegt in der Verwendung einzelner Module, aus denen sich das vollständige Boot-File zusammensetzt. Ausgehend von einem Grundprogramm, das alle verfügbaren Routinen beinhaltet, können Sie nach Belieben einzelne Programmteile ein- bzw. ausschalten oder auch durch eigene Entwicklungen ergänzen und erhalten so die optimale Lösung für Ihre individuelle Problemstellung.

Bedienung des Programms

Zu Beginn kümmert sich Uniboot um die Belegung der Funktionstasten. Da diese erfahrungsgemäß selten geändert werden (man will sich ja nicht ständig an eine neue Belegung

gewöhnen), wurde hier auf ein Menü zur Eingabe verzichtet. Setzen Sie daher vor dem Start des Programms in den Zeilen 1230-1320 die gewünschte Belegung direkt in das Programm ein. Hierbei gelten folgende Konventionen:

F\$(1)-F\$(8)	enthält die Belegung der Funktionstasten,
F\$(9)	beinhaltet den Text für <SHIFT>+<RUN/STOP> ,
F\$(10)	den Text für die HELP-Taste.

Der gesamte Text darf dabei nicht länger als 254 Zeichen sein. Die Belegung von <SHIFT>+<RUN/STOP> im Programm ermöglicht übrigens das Laden eines Programms direkt aus dem Directory mit anschließendem Autostart.

Nun sollten Sie »Ihre« Uniboot-Version mit DSAVE "DATEINAME" speichern. Beim anschließenden Start mit RUN zeigt Uniboot die Funktionstastenbelegung an, wie sie auf Wunsch im Bootsektor festgehalten wird.

Funktionstasten belegen

Im folgenden Programmteil stellen Sie sich Ihr eigenes Boot-File zusammen (Bild 15). Wählen Sie mit »J« oder »N« die Aufgaben, die Ihr Boot-File haben soll. Das Programm sortiert anschließend überflüssige Module automatisch aus.

Ein Beispiel soll Ihnen den Einstieg in »Uniboot« etwas erleichtern. In Tabelle 12 finden Sie eine Schritt-für-Schritt-Anleitung, wie der Assembler »Double-Ass« (Seite 77) bootfähig gemacht wird.

Insgesamt stehen folgende Möglichkeiten zur Verfügung (die Reihenfolge entspricht der Programmstruktur):

Tastenbelegung booten

Hier entscheiden Sie, ob die vorher gezeigten Tastenbelegungen beim Booten geladen werden sollen oder nicht. Wenn ja, werden sie auf dem ersten Sektor der ersten Spur der Diskette gespeichert und stehen nach dem Bootvorgang sofort zur Verfügung.

Datenblock booten

Dies ist nur möglich, wenn keine Tastenbelegungen gebootet werden. Ein ganzer Datenblock (255 Byte lang) kann mit dieser Funktion an eine beliebige Adresse geladen werden. Geben Sie dazu bei der Abfrage (Bild 15) die Startadresse des Blocks im Speicher (beim Erzeugen des Boot-Files) sowie die Zieladresse beim Booten an. Die Eingabe kann entweder dezimal oder hexadezimal erfolgen, wobei Hex-Adressen durch ein vorangestelltes »\$«-Zeichen zu markieren sind.

Der Datenblock kann je nach Inhalt auf unterschiedlichste Weise eingesetzt werden. Man kann damit zum Beispiel Texte direkt in den Bildschirmspeicher laden, ganze Zeropage-Blöcke ändern oder ein kurzes Programm-File bzw. ein Makro nachladen. Hier sind Ihrer Fantasie kaum Grenzen gesetzt.

Wenn Sie beispielsweise den Block ab \$1000 speichern und beim Booten wieder an die gleiche Adresse laden, speichert Uniboot die Funktionstastenbelegung des Computers ab. So können Sie auch ohne Änderung des Programms die Belegung der Funktionstasten sichern. Diese ist in den meisten Fällen nicht mit der Belegung von Uniboot identisch.

Programmfile nachladen

Dieser Punkt ermöglicht das automatische Laden eines Programmfiles. Geben Sie dazu den Filenamen an (Länge maximal 16 Zeichen). Das File wird absolut (also an die Adresse im Programmheader) geladen, daher lassen sich auch Maschinenprogramme problemlos handhaben.

Benutzer-Routine anfügen

Ist im reichhaltigen Angebot der Boot-File-Aufgaben eine von Ihnen benötigte spezielle Problemlösung noch nicht enthalten? Mit dieser Funktion können sie eine eigene kleine Maschinensprache-Routine an das Grundprogramm des Boot-Files anhängen. Sie muß im Speicher ab \$0BB3 laufen und sich während des Ablaufs von »Uniboot« auch an dieser Stelle befinden. Die Gesamtlänge kann 77 Byte betragen. Diese Größe reicht für kleinere Anwen-

dungen völlig aus. Wenn Sie mehr Platz benötigen, geben Sie bei »Datenblock booten« zweimal die Startadresse \$0C00 an – damit steht Ihnen der Bereich von \$0BB3 bis \$0CFF zur Verfügung. Die Routine muß mit RTS enden, da sie als Unterprogramm aufgerufen wird. Durch Aktivieren der Benutzer-Routine verringert sich der Platz für den Boot-Text (folgender Punkt) um 3 Byte.

Eigene Boot-Texte

Text nach Booting

Beim Booten gibt der Computer die Meldung »Booting« aus, der ein beliebiger Text folgen kann. Diesen können Sie hier eingeben. Für den Text sind 16 Zeichen reserviert. Zusätzlich wird der Platz genutzt, den der Filename freiläßt. Maximal sind also 32 Zeichen möglich.

C 64-Modus einschalten

Ein Bootfile mit dieser Funktion sollte (außer bei kopiergeschützten Originalprogrammen) auf jeder Ihrer C 64-Disketten vorhanden sein. Der Computer springt dann nach dem Booten automatisch in den C 64-Modus. Vorher wird allerdings das Programmfile (und der Datenblock) nachgeladen, die dann auch im C 64-Modus verfügbar sind. Einzige Bedingung dabei: Die Startadresse des Programm-Files muß größer als \$1300 (dezimal 4864) sein. Diese Boot-Variante ist also nicht für C 64-Basic-Programme geeignet. Für Assembler oder verschiedene Tools, die sich meistens im RAM-Bereich ab \$8000 bzw. \$C000 befinden, ist diese Option aber ideal.

Falls der C 64-Modus aktiviert wurde, werden die nun folgenden Wahlmöglichkeiten übersprungen, da sie nur im C 128-Modus funktionieren.

Default-Farbwerte ändern

Am Ende dieses Menüs können Sie sich Ihre Lieblings-Farbzusammenstellung aussuchen. Dieses Modul sorgt nun dafür, daß nach jedem <RUN/STOP RESTORE> statt den üblichen Farben wieder diese Wunschfarben eingesetzt werden. Außerdem wird im 80-Zeichen-Modus nach <RUN/STOP RESTORE> automatisch der Fast-Modus aktiviert. Sollten Sie die Farbwerte geändert haben, sind natürlich die nächsten beiden Punkte unsinnig und werden daher übersprungen.

RESTORE-Taste sperren

Die Kombination <RUN/STOP RESTORE> wird hiermit unwirksam gemacht. Die <RUN/STOP>-Taste funktioniert aber wie gewohnt.

STOP-Taste sperren

Falls Sie diese Funktion in Ihr Boot-File einbauen, ist eine Unterbrechung durch <RUN/STOP> nach dem Booten nicht mehr möglich. Die Kombination <RUN/STOP RESTORE> funktioniert allerdings nach wie vor. Außerdem verliert die Sperre nach dem ersten Betätigen der <RESTORE>-Taste Ihre Wirkung. Deshalb ist diese Option nur dann sinnvoll, wenn ein versehentliches Unterbrechen des Programms verhindert, ein gewollter Ausstieg aber weiterhin machbar sein soll.

Reset-Taste inaktivieren

Dies ist für einen guten Programmschutz interessant. Wenn Sie dazu noch die RESTORE-Taste sperren, ist Ihr Programm vor fremden Zugriffen geschützt. Sie können wählen, ob der Reset im C128- oder im C64-Modus gesperrt ist. Nach dem Reset wird ein Farbstreifenmuster erzeugt, ansonsten passiert nichts. Der Rechner ist nur noch mehr durch Aus- und Einschalten wiederzubeleben, wobei natürlich alle Programme gelöscht werden.

Programme schützen

Natürlich können Sie die letzten drei Punkte auch beliebig miteinander kombinieren, um so Ihre Programme optimal zu schützen.

POKE-Befehl ausführen

Eine unkomplizierte, universelle Funktion für alle, die nicht gleich ein komplettes Programm schreiben wollen. Durch die POKES können Sie einige Sonderwünsche, wie etwa einen List- oder Save-Schutz, problemlos ausführen. Geben Sie die POKE-Adresse und den POKE-Wert an. Es wird dabei immer die Bank 15 beschrieben, alle Zeropage-Adressen sind von dort erreichbar. Mindestens ein POKE-Befehl steht zur Verfügung. Haben Sie die <STOP>- beziehungsweise <RESTORE>-Taste nicht gesperrt, erhöht sich die maximale Anzahl um jeweils eins; Sie können also im Höchstfall drei POKES einsetzen. Hier einige Beispiele für nützliche POKES:

Listschutz:	POKE 775,139
Tastatur abschalten:	POKE 2592,0
Tasten-Repeat aus:	POKE 2594,0
Save-Schutz:	POKE 818,50

Autostart durchführen

Dieses Modul ist nur wählbar, wenn auch ein Programm-File nachgeladen wird. Es schreibt die Zeichenfolge RUN <RETURN> (CHR\$(13)) in den Tastaturpuffer und führt so nach dem Ladevorgang einen Autostart aus.

Farbe nach Wunsch

Nach dem Zusammenstellen der Modulteile legen Sie im Farbmenü die gewünschte Farbkombination fest. Nur wenn »Default-Farbwerte ändern« eingestellt wurde, bleiben sie auch nach <RESTORE> erhalten. Die Farben wählen Sie durch Betätigen der Cursor-Tasten. <RETURN> bestätigt die Eingabe.

Wenn Sie mit Ihrer individuellen Kombination fertig sind, werden die Daten auf Diskette gespeichert. Legen Sie dazu die Diskette in Ihr Laufwerk ein, die das Boot-File enthalten soll. Im Zweifelsfall können Sie sich vorher das Directory zeigen lassen.

Das Programm führt vor dem Speichern einen Block-Test durch und teilt Ihnen mit, ob die benötigten Blöcke frei oder von einem alten Boot-File, von einem Programm-File oder von einem fremden Boot-File (zum Beispiel bei CP/M) belegt sind. In den beiden ersten Fällen ist das Speichern problemlos möglich, bei einer Belegung durch ein Programm-File oder einen fremden Boot-Sektor riskieren Sie Datenverlust, wenn Sie trotzdem speichern!

Sobald Sie »grünes Licht« gegeben haben, wird gespeichert. Die Blöcke werden in der BAM als belegt gekennzeichnet, um ein versehentliches Überschreiben zu verhindern. Die Anzahl der freien Blocks beträgt daher nur noch 662.

Zum Ausprobieren Ihres erzeugten Boot-Files drücken Sie einfach die Reset-Taste. Wie gewohnt erscheint zuerst die Einschaltmeldung. Dann allerdings wird die Meldung »Booting«, gefolgt von Ihrem Text, ausgegeben.

Nach dem Laden des Datenblocks läuft die Floppy gegebenenfalls erneut an, um das Programm-File zu laden, danach wird gegebenenfalls die Benutzer-Routine (siehe oben) gestartet. Erst dann springt der Computer in den Direktmodus.

So arbeitet »Uniboot«

Soviel zu den Möglichkeiten, die Sie mit Uniboot haben. Doch wie wurde all dies softwaremäßig verwirklicht?

Die Antwort auf diese Frage soll Ihnen Tabelle 13 in Verbindung mit den Tabellen 14 und 15 geben. Hier finden Sie den Aufbau des Bootsektors von Uniboot, eine Gliederung des Programms und die Liste der wichtigsten Variablen.

Zum Schluß noch ein Tip: Wollen Sie einen optimalen Software-Schutz realisieren, wird ein kurzes Ladeprogramm notwendig, welches das zu schützende Programm nachlädt. Die Unterbrechungstasten sowie der Reset werden nämlich erst nach dem Laden der Programm-Files gesperrt.
(Christian Paul/Axel Pretzsch)

- Boot-Sektor automatisch schreiben
- Funktionstastenbelegung speichern und beim Booten aktivieren
- Zusätzlichen Datenblock beim Booten laden
- Benutzer-Routine in Bootfile einbauen
- Nach »BOOTING...« einen beliebigen Text ausgeben
- C 64-Modus einschalten
- Beliebige Farben nach <RUN/STOP RESTORE> einstellen
- Nach <RUN/STOP RESTORE> im 80-Zeichen-Modus Fast-Modus einschalten
- <RUN/STOP>, <RUN/STOP RESTORE> und/oder Reset sperren
- Beim Booten bis zu drei POKE-Befehle ausführen

Tabelle II. Die Leistungen von Uniboot auf einen Blick.

Ein individueller Boot-File für »Double-Ass«:

1. Kopieren Sie »Double-Ass« auf eine neu formatierte Diskette.
2. Belegen Sie mit KEY x, "..." im Direktmodus des C 128 die Funktionstasten nach Ihren Wünschen.
3. Laden Sie »Uniboot« mit DLOAD "UNIBOOT"
4. Nach der Anzeige der Funktionstastenbelegung (die sich höchstwahrscheinlich von Ihrer Belegung unterscheidet) müssen die Abfragen des Programmes so beantwortet werden:

Tastenbelegungen booten	NEIN
Datenblock booten	JA
Adresse im Speicher	\$1000
Nachladen an Adresse	\$1000
Programmfile nachladen?	JA
Filename	DOUBLE-ASS
Benutzer-Routine anfügen?	NEIN
Text nach Booting:	DOUBLE-ASS
C64-Modus einschalten?	NEIN
Default-Farbwerte ändern?	JA
Reset-Taste inaktivieren?	NEIN
Poke-Befehl ausführen?	NEIN
Autostart durchführen?	JA
Farbwahl: Wählen Sie Ihre »persönlichen Farben«	
80-Zeichen-Werte einstellen?	JA
Folgen Sie den Anweisungen auf dem Bildschirm	
Directory zeigen?	NEIN
Wollen Sie abspeichern?	JA
5. Diskette mit »Double-Ass« einlegen und Boot-File speichern.
Sobald der Bootblock auf die Diskette geschrieben wurde, können Sie ihn durch Drücken der Reset-Taste testen.

Table 12. So wird »Double-Ass« mit einem eigenen Bootsektor versehen.

AUFGABEN DES BOOTFILES:

TASTENBELEGUNGEN BOOTEN ? **NEIN**
DATENBLOCK BOOTEN ? **JA**
ADRESSE IM SPEICHER: \$1000
NACHLADEN AN ADRESSE: \$1000
PROGRAMMFILE NACHLADEN ? **NEIN**
BEWUTZER-ROUTINE ANFUEGEN ? **NEIN**
TEXT NACH BOOTING (MAX. 32 ZEICHEN):
>DAS IST EIN BEISPIELTEXT
C-64-MODUS EINSCHALTEN ?

Bild 15. Der erste Teil der Abfragen, um ein individuelles Boot-Programm auf der Diskette zu generieren.

Byte	Inhalt und Funktion
000	CBM-Kennung für Boot-Block
003	Adresse für den Datenblock
005	Nummer der Bank, in die geladen wird
006	Anzahl der nachzuladenden Blöcke
007	Bootmeldung
	\$00 am Ende
bis	Filename
	\$00 am Ende
041	Bereich, der wahlweise die Daten für drei POKEs, einen POKE und den Zeiger auf die Defaultfarben-Routine einen POKE und die Sperrung von <RUN/STOP RESTORE> zwei POKEs und <RUN/STOP> sperren zwei POKEs und <RESTORE> sperren enthält, je nachdem welche Konfiguration bei »Defaultfarben«, »RUN/STOP sperren« und »RESTORE sperren« gewählt wurde
056	Autostart-Modul
073	Defaultfarben-Routine ändert den Warmstart-Vektor und ruft folgendes Unterprogramm auf:
097	Farben-Unterprogramm wird in jedem Fall nach dem Booten aufgerufen, gegebenenfalls auch nach <RESTORE>. Ändert Vorder-/Hintergrund- und Schriftfarbe im 40-Zeichen-Modus bzw. Schriftfarbe und Fast-Modus bei 80-Zeichen-Darstellung.
107	C 64-Reset schreibt in Bank 0 den Text »CBM80« an Adresse \$8004 und richtet den Reset-Vektor auf die Killer-Routine
118	C 128-Reset schreibt in Bank 1 an Adresse \$FFF8 den Sprungbefehl zur Killer-Routine
152	Killer-Routine wird bei Reset aufgerufen, erzeugt Farbstreifenmuster in einer Endlosschleife
166	Daten der Routinen (CMB80, RUN <RETURN>)
178	frei für Benutzer-Routine
bis	
255	

Tabelle 13. So belegt »Uniboot« den Bootsektor.

Gliederung Zeile	Funktion
1100-1160	Initialisieren
1200-1290	Funktionstastenbelegung
1330-1360	Anzeige der Funktions-Tastenbelegung geschieht einschließlich der Grafikzeichen, was durch Simulation von <INSERT> möglich wird
1350	Aussortieren von <RETURN> und <SHIFT> <RETURN>
1380-1390	Zusammenbauen der Texte zu Block 2. Format: Länge der 10 Texte, dann die Texte hintereinander ohne Trennzeichen, wird direkt nach \$1000 geladen
1400	Decodieren der Hex-Datas für Block 1
1420-2110	Auswahl der Boot-File-Funktionen
1470 + 1500	Bearbeiten der Eingabe
1520	Einlesen des Datenblocks aus dem Speicher
1530	Sprungbefehl zur Benutzer-Routine einfügen
1570	Benutzer-Routine aus Speicher einlesen
1620-1630	Boot-Text zerlegen und in Block 1 schreiben
1640-1650	Filename zerlegen und in Block 1 schreiben
1670	Sprung in den C 64-Modus einfügen
1710	<RESTORE> ausschalten
1730	<RUN/STOP> ausschalten
1770	Reset sperren
1800-1860	POKE-Befehl durch LDA #xx : STA yyyy erzeugen
1920	Autostart-Modul einschalten
1930-2040	Farbenmenü
2050-2110	Unterprogramm zum Farbenmenü - Tasten abfragen - Werte setzen
2150-2510	Daten speichern
2190	Directory zeigen
2200	Datenkanäle zur Floppy öffnen
2220-2230	Die Datenblöcke probeweise belegen. Tritt ein Fehler auf, sind sie schon belegt.

Gliederung Zeile	Funktion
2240–2270	Sind die ersten 3 Zeichen in Block 1 »CBM«? Nein: Programm-File Ja: Boot-File
2280–2300	Steht ab Byte 175 der Text »RUN«? Ja: Uniboot-File Nein: Fremdes Boot-File
2310–2370	Gegebenenfalls fragen, ob trotz Programm/fremdem Boot-File gespeichert werden soll
2410–2500	Blöcke endgültig belegen
2510	Datenkanäle schließen, Programmende
2520–2540	TRAP-Routine: Bei Fehler Warnton + Close + Help
2550–2570	Ja/Nein-Abfrage
2580–2610	Eingabe der Daten
2650–2760	Daten für Block 1

Table 14. »Uniboot« nach Programmzeilen gegliedert.

Liste der wichtigen Variablen	
Felder	
F\$(10)	Funktionstastenbelegung
B1(255)	Codes für den Boot-Block
B2(255)	Codes für den Datenblock
Strings	
A\$	Enthält die Eingaben
N\$	Filename
TX\$	Text nach »BOOTING«
Numerische Variablen	
I,J	Schleifenzähler
L	maximale Länge – der Eingabe – der Funktionstastenstrings – des Boot-Textes
AD	POKE-Adresse bzw. Adresse, an die der Datenblock geladen wird
W	POKE-Wert bzw. Adresse des Datenblocks im Speicher
C,P	Block-Zähler, speichern den Platzbedarf einzelner Module
Z	Cursor-Zeile
S	Cursor-Spalte
Flags	
FT = 1	wenn keine Tastenbelegungen und kein Datenblock gespeichert werden sollen
PF = 1	wenn ein Programm-File geladen werden soll
RS = 1	wenn im C128-Modus der Reset gesperrt wurde
BR = 1	wenn <RUN/STOP> gesperrt sein soll

Tabelle 15. Die wichtigsten Variablen von »Uniboot« finden Sie hier übersichtlich aufgeteilt.

Micro-Hardcopy

Nur ein Viertel der Normalgröße braucht eine Micro-Hardcopy und verliert dabei nicht einen Punkt des Bildes. Diese gestochen scharfen Hardcopies können vom C 128 auf Epson-Druckern ausgegeben werden. Wir sagen Ihnen, wie!

Wer würde nicht gerne seine Disketten-Etiketten mit kleinen Bildern bedrucken, die über die Art des Programms auf der Diskette Auskunft geben; speziell bei Programmen mit einem hervorragenden, hochaufgelösten Titelbild bietet es sich an, dieses gleich als Disketten-Label zu übernehmen. Dem stand bisher immer die Größe eines solchen Bildes entgegen. Nun kann man bei Druckern, die Index- und Potenzschrift besitzen, auch den Punktabstand auf einen $\frac{1}{3}$ -Punkt-Abstand verringern und durch die geschickte Ausnutzung dieser Fähigkeit ein Grafikbild auf ein Viertel seiner ursprünglichen – normalen – Fläche verkleinern, wobei die Einzelheiten der Grafik erkennbar bleiben. Benötigt wird für dieses Programm nur ein Drucker, der Epson-kompatibel ist und einen $\frac{1}{216}$ -Zoll-Abstand erlaubt (mit ESC "J", möglich ab Epson MX Typ III).

Zu Laden ist das Programm mit BLOAD "MHC128". Die Befehlssyntax lautet:
SYS 6700,FN (Grafikseite 1 automatisch).

Dazu muß vorher der Druckkanal mit OPEN FN,4[,SEK] geöffnet werden. Der Vorteil dieser Methode mit der logischen Filenummer ist, daß man die Hardcopy auch in einem Floppy-File speichern oder auf einen Drucker mit anderer Geräteadresse oder spezieller Sekundäradresse ausgeben kann. Hat Ihr Interface verschiedene Modi, so wählen Sie die Sekundäradresse für den Direktkanal aus.

Beim C 128 wird immer auf die Seite 1 zugegriffen, da sie vom Basic 7.0 durch Grafikbefehle unterstützt wird. Übrigens, das Speichern von Grafiken im C 128 geschieht mit »BSAVE-NAME",P7168 TO P16192« inklusive Farbe beziehungsweise »BSAVE "NAME",P8192 TO P16192« für Grafiken ohne Farbe. Geladen wird dann mit BLOAD "NAME". Bei Fremdgrafiken muß man probieren: Entweder funktioniert BLOAD "NAME",P7168 oder BLOAD "NAME",P8192 (Vorsicht: Bei letzterem Befehl ist ein Programmverlust möglich, zum Beispiel bei Hi-Eddi-Grafiken!).

(D. Temme/og)

Vectors

Ein Super-Action-Spiel für den C 128, das den Joystick heißlaufen läßt und Sie sicher für viele Stunden an den Bildschirm fesselt.

Dieses sehr schnelle Geschicklichkeitsspiel kann einen fast süchtig machen. Es greift die Idee zweier sich bekämpfender Schlangen auf (ähnlich der Motorrad-Sequenz im Film »Tron«) und setzt sie auf dem C 128 im 80-Zeichen-Modus auf faszinierende Weise neu um. Bei »Vectors« kommt es darauf an, den Gegner durch geschicktes Manövrieren einzukreisen, ihn gegen eine Wand rennen zu lassen und damit auszuschalten. Durch die hohe Grafikauflösung (640 * 200 Punkte) ist der Kampf der beiden Linien wesentlich spannender als bei bisherigen Versionen. »Vectors« wurde mit einigen Zusatzfunktionen ausgestattet. So kann man als weitere Schwierigkeiten zufällige Hindernisse in das Spielfeld einschalten, die auch vom Benutzer editierbar sind. Auch diagonale Linien sind möglich, wodurch die Schwierigkeit für Vektor-Profis gesteigert werden kann.

Es besteht aber auch die Möglichkeit, gegen den Computer anzutreten und verschiedene Schwierigkeiten anzuwählen. Diese können durch Ab- und Hinzuschalten verschiedener Computerstrategien zusätzlich variiert werden.

Hat man den Menüpunkt »Beschleunigen« gewählt, so läßt sich durch Drücken des Feuerknopfes der Vektor (Linie) beschleunigen.

Computerstrategien

Die Spielstärke des Computers wird durch Hinzuschalten verschiedener Strategien eingestellt. Im Programm sind die zehn sinnvollsten Kombinationen der folgenden Einzelstrategien integriert:

- Spirale
- Gasse
- Folgen
- Besoffen
- Zufall

Der Computer-Vektor ist prinzipiell so gesteuert, daß er zwei Pixel vor einem Hindernis (Gegner-Vektor, Hindernis oder Spielfeldrand) ausweicht. Durch dieses Freilassen eines Pixels sichert er sich einen Ausweg, falls er in eine Falle läuft.

Beim Ausweichen wird die neue Richtung zufällig ausgewählt (siehe auch »Folgen«).

Die Strategien im einzelnen

Spirale:

Diese Strategie überprüft, ob sich der Computer-Vektor in einer Spirale verfangen hat, und sichert sein Entkommen.

Gasse

Steuert der Computer auf einen Spalt zu, der nur einen Punkt breit ist (mögliche Sackgasse), so weicht er aus.

Folgen

Bei einem Ausweichmanöver orientiert sich der Computer an der aktuellen Position des Gegner-Vektors. Man wird von ihm verfolgt!

Besoffen

Der Computer weicht nach einer bestimmten Anzahl zurückgelegter Pixel aus (siehe auch »Folgen«), egal, ob er auf ein Hindernis stößt oder nicht. Damit wird er unberechenbar!

Zufall

Diese Strategie arbeitet mit »Besoffen« zusammen. Sie bestimmt die Anzahl der Pixel, nach der der Computer die Richtung wechselt. Falls die Strategie aktiviert ist, addiert er zu einem Startwert einen Zufallswert, ansonsten übernimmt er den Startwert.

Die Spielstärke ergibt sich nicht unmittelbar aus der Reihenfolge der Strategiekombinationen. Unserer Meinung nach ist die Strategie 7 die spielstärkste. Sehenswert ist aber auch Strategie 4.

Ändern der Hindernisse

Nach dem Laden des Programms befinden sich im Speicher zwei Tabellen. Die erste Tabelle liegt ab der Speicherstelle \$07374 in BANK 0. Es folgen 16 Byte, die jeweils den Wert 0 oder 1 haben. Der Wert 1 steht für »Hindernis gesetzt«, 0 bedeutet »Hindernis aus«.

Dabei entspricht Byte 0=Hindernis 0, Byte 2=Hindernis 2 und so weiter. Hieraus ergibt sich eine maximale Anzahl von 16 Hindernissen.

In Tabelle 2 (ab Adresse \$07384, BANK 0) sind für jedes Hindernis 2 Byte reserviert, die den Anfang eines Hindernisses in Low-Byte und High-Byte angeben:

Byte 0 und 1: Low-Byte und High-Byte der Anfangsadresse des Hindernisses Nummer 0;

Byte 2 und 3: Low-Byte und High-Byte der Anfangsadresse des Hindernisses Nummer 1 und so weiter...

Die Hindernisse können ab der Adresse \$07643 in BANK 0 abgelegt werden.

Aufbau eines Hindernisses

Ein Hindernis baut sich aus vertikalen, horizontalen und diagonalen Strichen auf. Ein Strich setzt sich aus 5 Byte zusammen:

Byte 0:

Low-Byte der x-Koordinate der Startkoordinaten des Striches

Byte 1:

High-Byte der x-Koordinate der Startkoordinaten des Striches

Byte 2:

y-Koordinate der Startkoordinaten des Striches

Byte 3:

Länge des Striches (maximal 255)

Byte 4:

Richtung des Striches

0 – Rechts

1 – Unten

2 – Links/Unten

3 – Rechts/Unten

Ein Strich mit den Koordinaten $x=317$, $y=97$ und der Länge 46 in Richtung Rechts/Unten sähe so aus:

Byte: 0 1 2 3 4

Wert: 061 001 097 046 003

Zum Austesten empfiehlt es sich, den eingebauten Monitor zu benutzen.

Start des Programms

Bitte schalten Sie Ihren Monitor auf 80 Zeichen um und drücken Sie die <40/80>-Taste auf Ihrem Computer.

Geben Sie jetzt bitte

```
RUN " VECTORS.BOOT "
```

ein. Das Programm wird geladen und gestartet.

Die Bewegung der Vektors und die Auswahl der Menüpunkte erfolgt mit dem Joystick.

Wenn die Hindernisse und Strategien trotzdem noch zu leicht sind, der sollte ruhig das Spiel erweitern. So zum Beispiel einen dritten, unabhängigen Vektor, der jeden der beiden anderen behindert. Oder knobeln Sie sich selbst noch weitere ausgekochte Hindernisse aus.

(H. Beine/A. Jansen/kn/nj)