

HAPPY COMPUTER
2. SCHNEIDER SONDERHEFT

HAPPY COMPUTER

Markt & Technik

DAS GROSSE HEIMCOMPUTER-MAGAZIN

Tips, Tricks und Utilities
Vom Bildschirm-Zoom bis zum Disketten-Doktor

Grundlagen: So programmiert man 3D-Grafik

Preiswert selbst gebaut:

- ★ RS232-Schnittstelle
- ★ Reset-Schalter

NEU: Hilfe beim Abtippen

Jetzt alle Listings mit Prüfsumme

Hardware richtig ausgewählt

- ★ Drucker
- ★ Floppy-Laufwerke
- ★ Speichererweiterungen



Alle Programme auch auf Diskette und Kassette erhältlich

JETZT AUF SCHNEIDER-COMPUTERN:

Turbo Lader

DIE PROGRAMM-BIBLIOTHEK FÜR **TURBO PASCAL**®



TURBO-Lader-Grundpaket

Das TURBO-Lader-Grundmodul ist eine umfangreiche Programm-Bibliothek für den TURBO-Pascal-Programmierer. Sie umfaßt zahlreiche ausführlich dokumentierte Prozeduren und Funktionen, die der Profi zur schnellen Lösung seiner Programmieraufgaben verwenden kann und dem Einsteiger das Erlernen der Pascal-Programmierung erleichtern. Das Grundpaket TURBO-Lader bietet ein breitgefächertes Spektrum von Routinen, beginnend bei Bitmanipulation über optimierte Sortierverfahren bis hin zur Anwendung von Splinefunktionen, Fouriertransformation und Regressionsanalyse. Des weiteren Disketten-Routinen zum Lesen eines Inhaltsverzeichnis oder zum Lesen und Schreiben einzelner Sektoren, Routinen zur Datenüberprüfung, ein Spooler mit Steuer-routinen, erweiterte Stringverarbeitung und vieles mehr. Alle Routinen werden im kommentierten Quellcode für den TURBO-Pascal-Compiler ausgeliefert.

Das TURBO-Lader-Grundpaket erfordert den TURBO-Pascal-Compiler. Es ist lieferbar auf 3"- und 5 1/4"-Disketten und lauffähig auf dem Schneider CPC 464, CPC 664, CPC 6128 und Joyce.

3"-Disk. Best.-Nr. MS 413
5 1/4"-Disk. Best.-Nr. MS 415 **DM 138,-***

*inkl. MwSt., unverbindliche Preisempfehlung.



TURBO-Lader Business

TURBO-Lader Business umfaßt einen komfortablen Bildschirm-Maskengenerator und eine professionelle Dateiverwaltung. Der Maskengenerator gibt dem Pascal-Programmierer ein Werkzeug zur einfachen Bearbeitung von Bildschirm-Masken in die Hand. Eine Maske kann beliebig viele Textfelder, bis zu 128 Eingabe- und 128 Ausgabefelder enthalten. Eingabefelder können auf komfortable Art editiert und auf Gültigkeit überprüft werden. Das Dateiverwaltungsmodul unterstützt die Programmierung von Datenbankanwendungen und Stammdatenverwaltungen. Es besteht aus einer komfortablen Datensatz- und Indexverwaltung mit mehreren Schlüsseln und Index-Dateien, die einen sekundenschnellen Zugriff auf beliebige Daten ermöglicht. Mit diesen beiden Modulen stehen dem Anwendungsprogrammierer zwei professionelle Werkzeuge zur zeit- und kostensparenden Erstellung kommerzieller Anwendungen zur Verfügung. Alle Routinen werden im kommentierten Quellcode für den TURBO-Pascal-Compiler ausgeliefert.

TURBO-Lader Business erfordert den TURBO-Pascal-Compiler und das TURBO-Lader-Grundpaket. Es ist lieferbar auf 3"- und 5 1/4"-Disketten und lauffähig auf dem Schneider CPC 464, CPC 664, CPC 6128 und Joyce.

3"-Disk. Best.-Nr. MS 423
5 1/4"-Disk. Best.-Nr. MS 425 **DM 148,-***



TURBO-Lader Science

TURBO-Lader Science ist eine Sammlung technisch/wissenschaftlicher Funktionen und professioneller statistischer Verfahren für die Bereiche Medizin, Betriebs- und Volkswirtschaft, Technik und Naturwissenschaften. Das Modul enthält alle arithmetischen Operationen zur Verarbeitung komplexer Variablen inklusive der Umrechnung der Darstellung und die wichtigsten komplexen Funktionen wie Potenz, Wurzel, trigonometrische, transzendente und exponentielle Funktionen. Darüber hinaus ist ein vollständiges Paket zur Verarbeitung komplexer Matrizen und Vektoren enthalten. Der Statistikteil ist ein praktisches und direkt verwendbares Werkzeug zur computerunterstützten, effektiven Datenanalyse. Er umfaßt eine Vielzahl statistischer Funktionen mit den Schwerpunkten Regression und Korrelation, deskriptive Statistik, Faktoranalyse und Testverfahren. Alle Routinen werden im kommentierten Quellcode für den TURBO-Pascal-Compiler ausgeliefert.

TURBO-Lader Science erfordert den TURBO-Pascal-Compiler und das TURBO-Lader-Grundpaket. Es ist lieferbar auf 3"- und 5 1/4"-Disketten und lauffähig auf dem Schneider CPC 464, CPC 664, CPC 6128 und Joyce.

3"-Disk. Best.-Nr. MS 433
5 1/4"-Disk. Best.-Nr. MS 435 **DM 189,-***

TURBO-Pascal® ist ein Warenzeichen der Borland Inc., USA. TURBO-Lader, TURBO-Lader Business und TURBO-Lader Science sind Warenzeichen der Fa. Lauer & Wallnitz.

Diese Markt & Technik Softwareprodukte erhalten Sie in den Computer-Abteilungen der Kaufhäuser Horten, Karstadt, Kaufhof, Quelle, bei Ihrem Computerhändler und bei unseren Depot-Buchhändlern. Wenn Sie direkt beim Verlag bestellen wollen: per Nachnahme oder gegen Vorauskassa durch Verrechnungsscheck oder mit der eingeleiteten Zahlkarte.

Markt & Technik
**Schneider CPC-
Software**

Hans-Pinsel-Straße 2, 8013 Haar bei München
Schweiz: Markt & Technik Vertriebs AG, Kollerstrasse 3, CH-6300 Zug, ☎ 042/41 56 56
Österreich: Microcomput-ique Schiller, Fasngasse 21, A-1030 Wien, ☎ 02 22/78 56 61

Bestellungen im Ausland bitte an nebenstehende Adressen.

Für Auskünfte stehen Ihnen Herr Barsa, Tel. 0 89/46 13-133, und Herr Teller, Tel. 0 89/46 13-205, gerne zur Verfügung.



Andreas Hagedorn

Der Goldene Schnitt

Seit dem Altertum spricht man vom »goldenen Schnitt« wenn zwei Linien in einem idealen harmonischen Längenverhältnis zueinander stehen. Bei allen Computern aus dem Hause Schneider kann man das Verhältnis zwischen Preis und Leistung auch als »goldenen Schnitt« bezeichnen. Damit ist es kein Geheimnis, warum sich in Zeiten, in denen andere die Segel streichen müssen, der Schneider als Newcomer auf Platz 2 der Verkaufshitparade schieben konnte. Allerdings befriedigt der Computer allein noch nicht alle Wünsche seines Besitzers.

Was man braucht, sind Informationen. Und dies ist das Motto der zweiten Schneider-Ausgabe aus der Sonderheft-Reihe von Happy-Computer, die Sie jetzt in den Händen halten. Immer mehr Anfragen zu verschiedensten Themen erreichen täglich die Redaktion. Zu den am häufigsten genannten, finden Sie hier Anregungen — sowohl für Anfänger wie auch für Profis.

Eins haben sich alle Schneider-Fans gewünscht. Tips und Tricks für jeden Zweck. Und hier sind sie. Sie finden RSX-Erweiterungen genauso wie eine optimale Anpassung von CP/M-Programmen (am Beispiel Wordstar), und jeder Besitzer einer Diskettenstation wird sich über »Help«, einen Disk-Doktor, freuen. Probleme, wie beispielsweise versehentlich gelöschte Programme, gibt es nicht mehr.

Ein weiteres Kapitel beschäftigt sich mit den Firmware-Routinen Ihres Schneiders. Firmware, das sind die Maschinencode-Programme, die im ROM fest eingebaut sind. Sie werden vom Basic-Interpreter bei seiner Arbeit benutzt, um den Computer anzuleiten. Es gibt aber auch Routinen, die normalerweise von Basic nicht benutzt werden. Wußten Sie zum Beispiel schon, daß man mit einem einfachen Befehl Zeichen auf

dem Bildschirm auch nach unten scrollen kann? Diesen und viele andere »Calls« finden Sie ausführlich erklärt. Eine RSX-Anweisung mit der Sie das alles ausprobieren können, finden Sie auf Seite 67.

Wer mit seinem Computer spielen will, für den hat unser Spiele-Freak Heinrich Lenhardt die interessantesten Programme zusammengestellt. Aber wer lieber seinen Computer mit Listings »füttern« will, der kann seine kriminalistischen Fähigkeiten beim Fall »Famit AG« beweisen.

Erstmals in einer deutschen Programm-Sammlung für Schneider-Computer finden Sie jedes Basic-Listing mit einer Prüfsumme abgedruckt. Wenn Sie das Eingabeprogramm »Explora« (Seite 13) benutzen, wird diese Zahl nach jeder Zeile auf dem Bildschirm ausgegeben. Sie können so sofort überprüfen, ob die Eingabe korrekt war. Und noch eine Erleichterung haben wir für Sie eingebaut. Die Bildschirmsteuerzeichen werden alle in Klartext ausgegeben. Somit sollte es für Sie ein leichtes sein, alle Listings abzutippen.

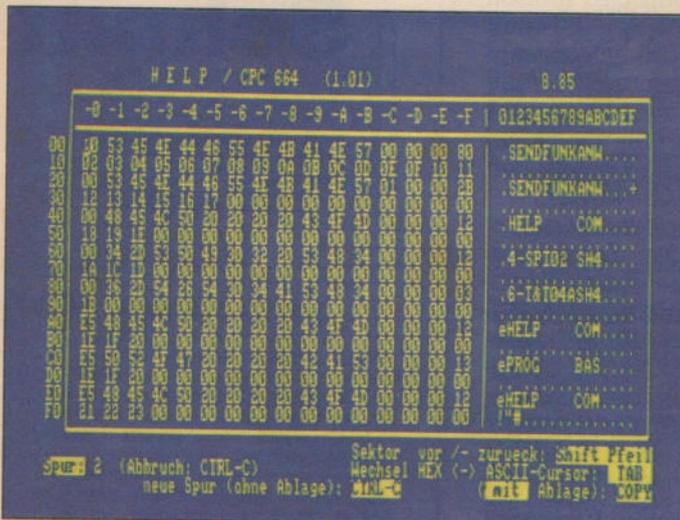
Wem das allerdings zu viel Arbeit ist, für den bieten wir natürlich auch wieder alle Programme auf 3-Zoll-Diskette (34,90 Mark, Bestell-Nummer LH 86S1D) und auf zwei Kassetten (29,90 Mark, Bestell-Nummer LH 86S1K) an. Bestellen können Sie diese ganz einfach mit der Zahlkarte in der Mitte des Hefts.

An dieser Stelle möchte ich mich bei allen Lesern bedanken, die sich aktiv an der Happy-Computer beteiligen. Nur mit Ihren Anregungen ist es uns möglich, Ihnen die Informationen zu bieten, die Sie sich wünschen. Aber auch Kritik wird von uns gern gelesen. Denn nur mit Ihren Ideen können wir die Zeitschrift — und die Sonderhefte — so gestalten, wie Sie sich diese wünschen.

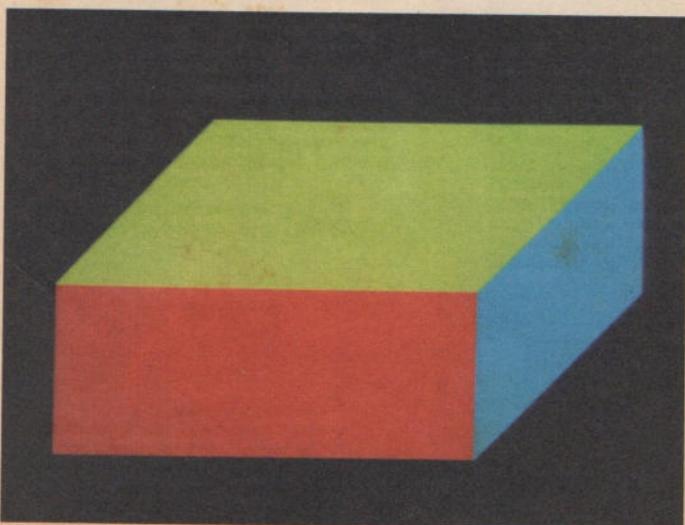
Andreas Hagedorn



Es gibt Alternativen vom Schneider-Laufwerk 6



Disketten Byte für Byte gelesen 100



Des Schneiders dritte Dimension 34

Hardware

Hardware richtig ausgewählt	
Speichern mit Luxus	6
Die Qual der Druckerwahl	8
Was Sie schon immer über Joysticks wissen wollten	10
Mehr Speicher für den CPC	12

Bastelei

Datenfernübertragung muß nicht teuer sein	
RS232-Schnittstelle	14
Reset ohne Datenverlust	
Reset-Schalter	19
Hifi am Schneider	21
Schluß mit dem Kabelärger	21

Spieletestk

Schneider verspielt	24
---------------------	----

Grundlagen

Wenn der Recorder streikt	43
Firmware-Routinen in Basic genutzt	45
Ihr Schneider mal ganz intern	48
Welcher Wert steht wo?	51

Grafik-Grundlagen

So programmiert man 3D-Grafik	
Ein Zuckerhut für den Schneider	28
3D auf dem CPC	34
Schnelle Kreise	39

Logo-Grundlagen

Dr. Logo — mehr als nur eine Schildkröte	132
--	-----

CP/M

»Wanzen-Tod« mit DDT	90
Wordstar stark verbessert	93

Anwendungs-Listings

»Supermon CPC-1002«	
Der Maschinensprache-Monitor	58
Daten im direkten Zugriff	84

Immer auf dem neuesten Stand	114
Geldüberweisen — natürlich mit Computer	120
Nebenkosten einfach abgerechnet	123

Spiele-Listing

Der Fall »Famit AG«	125
Autofahren ohne Tempo-Limit	131

CP/M-Listings

»Hilfe« in Sicht	
Disketten-Doktor	100
Drucken ohne Platzprobleme	107
Druckersteuerung leicht gemacht	109
Verschiedene Tastaturen unter CP/M	111

Tips & Tricks

Keine Eingabefehler mit »Explora«	
Eingabehilfen	13
Backup muß nicht teuer sein	65
Versteckte Bytes	67
Gläserne Firmware	67
Basic-Zeilen automatisch erzeugt	68
Vier statt zwei	69
Wo ist das Programm?	69
Löschtaste für den Schneider	72
Strings — ein sicheres Versteck	72
Groß, größer, »Dehni«	
Bildschirm-Zoom	73
Windows à la Macintosh	73
Kreise mit einem Basic-Befehl	75
Schnelles Laden von Kassette	76
Schere statt Bleistift und Radiergummi	77
Programmieren — schnell und einfach	78
Unsichtbare Programme	80
Tricks mit dem Joystick	82
Schnell gespeichert	82
Kleine Buchstaben ganz groß	83
Endlich wieder sichtbar	83

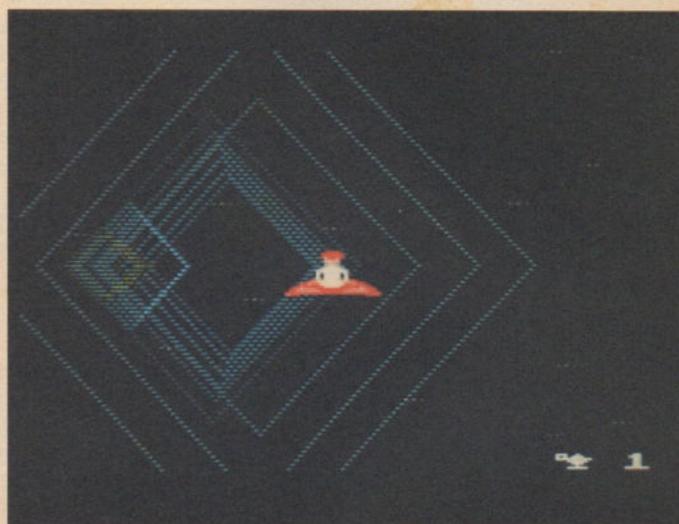
Rubriken

Editorial	3
Impressum	138

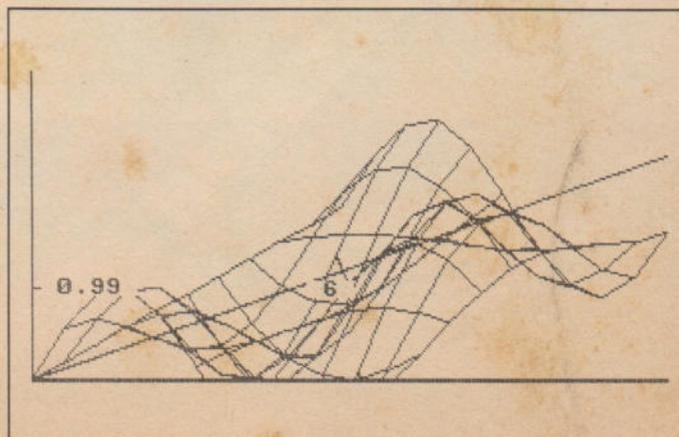


Was muß ein Drucker alles können?

8



Entführen Sie Ihren Schneider in Aladins Wunderland 24



Grafiken in drei Ebenen

28

Speichern mit Luxus



Jeder CP/M-Computer ist in seiner Leistung von einer guten Diskettenstation abhängig. Welches Laufwerk ist nun für den Schneider am besten geeignet? Zwei Laufwerke stehen zur Auswahl für den Anwender.

Viele Käufer begnügen sich zum Speichern zunächst mit dem ab Werk eingebauten Datenrecorder. Die dadurch bedingte Ladezeit erfordert aber viel Geduld. Auch eine Erhöhung der SAVE-Baudrate bringt meist nicht den gewünschten Erfolg. Spätestens jetzt geht die Überlegung in Richtung Diskettenstation. Schnelle Zugriffszeiten und hohe Datensicherheit sind nur einige Vorteile. Durch die CP/M-Fähigkeit des CPC 464 in Verbindung mit einem geeigneten Laufwerk wird Kompatibilität unterschiedlichster Geräte gewährleistet. Aus diesem Grund liegen allen Laufwerken eine CP/M-Softwareschnittstelle auf Diskette bei.

Was verbirgt sich hinter dem Zauberwort CP/M? Nun, CP/M (Control Program for Microcomputers) ist ein diskettenunterstütztes Betriebssystem der Firma Digital Research. Es wurde für Mikrocomputer mit dem Intel 8080 Mikroprozessor entwickelt, wird heute aber fast ausschließlich im Zusammenhang mit dem Z80-Mikroprozessor von Zilog verwendet. Ihn benutzen auch der CPC 464, 664 und der 6128. CP/M ist eine anwenderfreundliche Softwareschnittstelle, die eine von der Hardware unabhängige Benutzerumgebung schafft. Deshalb setzt der Umgang mit Ihrem CPC unter CP/M keinerlei Vorkenntnisse der diversen Hardwarestrukturen des Computers voraus.

Selbstverständlich ist auch das im CPC 464 eingebaute Basic durch das Diskettenbetriebssystem Amdos (Amstrad Disc Operating System) beziehungsweise Vdos (Vortex Disc Operating System) um einige Befehle erweitert worden, die die Arbeit mit einer Diskettenstation unter Basic in vollem Umfang erlauben. Mit Basic und Amdos oder Vdos einerseits und CP/M 2.2 andererseits, stehen Ihnen zwei Betriebssysteme auf dem CPC 464 zur Verfügung. Vor allem für CP/M 2.2 gibt es mehrere Tausend qualitativ hochwertige Anwenderprogramme. Dazu gehören fast alle Programmiersprachen (Basic, Pascal, Fortran, Forth, Cobol, C), Textverarbeitungsprogramme (Wordstar, Datastar etc.), Datenbanksysteme (dBase etc.) und auch eine Vielzahl an Assemblern. Viele der Programme sind im 3-Zoll-Format erhältlich, die meisten CP/M-Programme haben allerdings 5¼-Zoll-Format. Alle ungesicherten Programme, die Sie bis jetzt auf Kassette gespeichert haben, können auf Diskette kopiert werden. Auch können Sie weiterhin den Datenrecorder benutzen, um zum Beispiel Dateien von Diskette auf Kassette zu sichern. Da es für den Schneider CPC 464 zur Zeit zwei verschiedene Diskettenformate gibt, stellt sich natürlich die Frage, welche Diskettenstation für welchen speziellen Anwendungsbereich die richtige ist. Es gibt eine 5¼-Zoll-Sta-

tion von Vortex und die Schneider-3-Zoll-Laufwerke. Es sei gleich am Anfang gesagt, daß keines der beiden Laufwerke, ob nun Schneider oder Vortex, dem anderen auf irgend eine Art und Weise nachsteht. Um Ihnen die Qual der Wahl etwas zu erleichtern, erst einmal ein technischer Überblick.

Disketten-Formate:

Vortex:

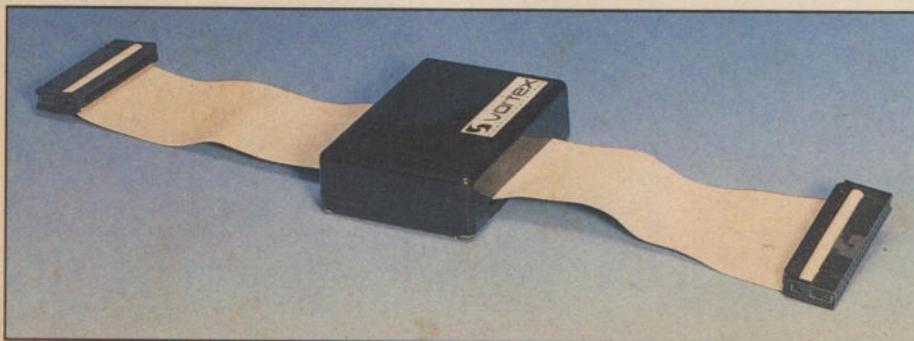
9 Sektoren/Spur
512 Bytes/Sektor
80 Spuren/Seite
2 Seiten
2 reservierte Spuren
4 KByte Blöcke
64 Directory-Einträge/Diskette
Sektornummern 1, 2, ..., 9
4 ms Steprate
720 KByte Gesamtspeicherkapazität / Diskette (unformatiert)

Schneider:

9 Sektoren/Spur
512 Bytes/Sektor
40 Spuren/Seite
1 Seite
2 reservierte Spuren
1 KByte Blöcke
64 Directory-Einträge/Diskette
Sektornummern 41, 42, ..., 49
12 ms Steprate
180 KByte Gesamtspeicherkapazität / Diskette (unformatiert)

Alle hier genannten Parameter können mit Hilfe eines speziellen Installationsprogramms geändert werden. Die Befehlsweiterungen beider Geräte sind sich sehr ähnlich. Die neuen Befehle lauten:

- CPM
- CP/M-Betriebssystem Starten
- FORMAT
- Diskette formatieren
- RESET (nur Vortex)
- Schließt eine Datei und geht ins VDOS
- S (nur Vortex)
- Wahl des Zweitlaufwerks (5¼ Zoll / 3 Zoll)
- CODE (nur Vortex)
- Programmschutz (Codeschlüssel)
- A
- Macht Laufwerk A zum Defaultlaufwerk
- B
- Macht Laufwerk B zum Defaultlaufwerk
- DIR
- Listet den Disketteninhalt
- ERA
- Löscht ein File oder eine Gruppe von Files
- REN
- Ändert Namen von Files
- SELECT (im Amdos USER)
- Wahl des User-Bereichs und des Defaultlaufwerks



Ein Adapter erlaubt den Anschluß des Schneider-Laufwerks an die Diskettenstation von Vortex

- DISC
Ein- und Ausgabekanal sind der Diskettenstation zugeordnet
- CAS (im Amsdos TAPE)
Wie Disc, allerdings für Datenrecorder
- DISC.IN
Der Eingabekanal ist der Diskettenstation zugeordnet
- CAS.IN (im Amsdos TAPE.IN)
Wie Disc.in, allerdings für Datenrecorder
- DISC.OUT
Der Ausgabekanal ist der Diskettenstation zugeordnet
- CAS.OUT (im Amsdos TAPE.OUT)
Wie Disc.out, allerdings für Datenrecorder
- ATRIBUT
Setzen von Fileattributen
- DRIVE (nur Amsdos)
Setzen des Standardlaufwerks

Alle anderen Kassettenoperationen sind weiterhin verfügbar. Es ist hierbei allerdings noch zu erwähnen, daß der Merge-Befehl bei der Schneider-Version nicht ganz einwandfrei funktioniert. Den Fehler kann man aber mit einer Software-Lösung umgehen. Vortex bietet den Vorteil, Fehlermeldungen in deutscher Sprache auszugeben.

Wie sind die zusätzlichen Diskettenoperationen nun aber eingebunden? Für Eingeweihte sind die Ausdrücke »Sprungtabelle« und »Sprungvektoren« nichts Neues. Der Kassettenbetrieb wird über eine solche Sprungtabelle mit Vektoren verwaltet. Damit alle Befehle, die die Kasette steuern auch auf Diskette wirken, werden die entsprechenden Sprungvektoren ersetzt. Das Betriebssystem beinhaltet eine Basic-Erweiterung, die mit RSX ungebunden ist.

Da diese Befehlsnamen den residenten CP/M-Befehlen sehr ähnlich sind, braucht man sich meist nur einen Befehlsnamen merken. Von Nachteil bei den RSXs ist allerdings, daß Strings nicht direkt übernommen werden dürfen, sondern in einer Variablen definiert sein müssen. Mit dem @-Pointers wird die Adresse dieser Variablen übergeben. Ein Beispiel:

Löschen des Directory-Eintrages beziehungsweise des Programms TEST.

Vorgehensweise:

\$\$ = "TEST"

ERA, @\$

Beide Laufwerke benutzen den Controller μ PD 765.

Dieser Controller ist in der Lage, hardwaremäßig bis zu vier Laufwer-

Bild 1. Klein aber fein: die 3-Zoll-Station von Schneider



Bild 2. Mehr als 700 KByte Speicherplatz bietet das 5 1/4-Zoll-Laufwerk von Vortex

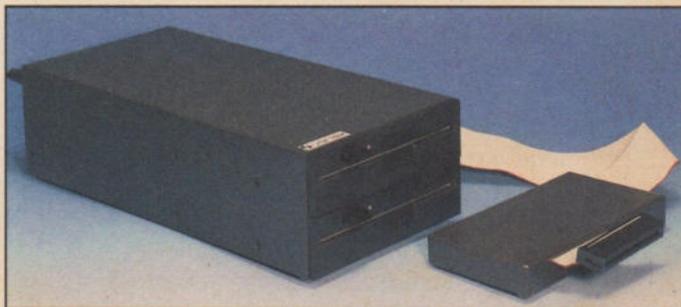


Bild 3. Das ideale Zweitlaufwerk: 3-Zoll-Gerät von Cumana

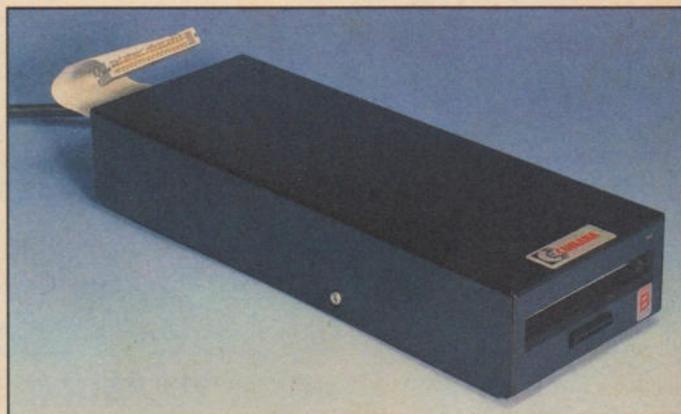
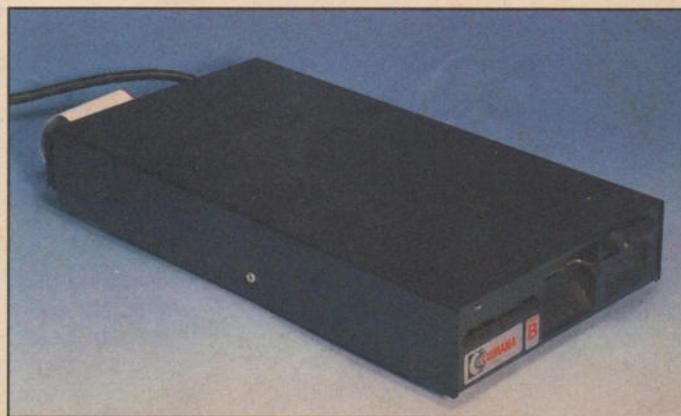


Bild 4. Wer zwei Formate haben will braucht die 5 1/4-Zoll-Station von Cumana als Zweitlaufwerk



ke (3, 3 1/2, 5 1/4 oder 8 Zoll) mit Single- oder Double-Density zu verwalten. Hierbei dürfte allerdings die 8-Zoll-Double-Density von der Übertragungsrate her (500 KBit/s) für die Z80A-CPU wohl etwas zu schnell sein. Diese Vielseitigkeit mit verschiedenen Formaten macht es bei-

den Firmen möglich, jeweils das andere Laufwerkformat kompatibel zu machen (Vortex zu 3 Zoll/Schneider zu 5 1/4 Zoll).

Weiterhin bietet der Controller die Möglichkeit, die Parameter der Diskettenformate zu verändern. Dadurch kann man Disketten von frem-

den Computern lesen beziehungsweise beschreiben. Die Firma Vortex hat bereits das Programm Para auf dem Markt, wodurch man diese Parameter der Diskettenformate menügesteuert verändern kann.

Schaut man die Laufwerke näher an, so bemerkt man zunächst einmal die unterschiedliche Speicherkapazität. Im Gegensatz zu der doppel-seitig beschreibbaren Diskette von Vortex mit einer Kapazität von 720 KByte (unformatiert), erscheint die Speicherleistung von 180 KByte (unformatiert) des nur einseitig schreibenden Schneider-Laufwerks doch recht mager. Da die Zugriffszeiten vergleichbar sind, wird in erster Linie der benötigte Speicherplatz eines Anwenders für die Wahl eines Laufwerks ausschlaggebend sein.

Beide Laufwerke machen einen eher robusten Eindruck. Unterschiede werden jedoch bei den Disketten deutlich. Im Gegensatz zu der

äußerst stabilen 3-Zoll-Diskette (Schneider), die auch eine raue Behandlung unbeschadet übersteht, verlangt die 5¼-Zoll-Diskette (Vortex) eine vorsichtigeren Handhabung.

Von beiden Firmen werden als Utilities für CP/M unter anderem ASM und DDT mitgeliefert. Das sind Programme, mit denen man Maschinenprogramme erstellen (ASM) und testen (DDT) kann, allerdings im 8080-Code.

DOS mit Monitor

Für Z80-Assembler bleibt die Möglichkeit, auf einen anderen Assembler/Monitor zurückzugreifen (zum Beispiel Mona/Gena von Schneider) oder auf ein Vortex-Laufwerk mit Betriebssystem 2.0 und ROM-residentem Monitor.

Wer VDOS 1.0 besitzt, ist nicht im

Nachteil, denn laut Vortex wird es kostenlos gegen VDOS 2.0 getauscht. Außer dem ROM-residenten Monitor hat man durch die Befehlsenerweiterung des CPC 664 noch eine schnellere Bildschirmausgabe zur Verfügung.

Beide Laufwerke entsprechen dem heutigen technischen Standard. Bezüglich der Zugriffszeit erfüllen sie gehobene Ansprüche, so daß sie für den professionellen Einsatz durchaus geeignet sind. Jedes Format hat seine Vorteile. 3-Zoll-Disketten sind zwar sehr teuer, allerdings ist 3 Zoll das Herstellerformat, die Software wird in diesem Fall also immer zu kaufen sein.

Wer sich nicht für ein Format entscheiden kann, dem sei der Zweitlaufwerk-Hersteller Cumana empfohlen. Dieser bietet preisgünstige Zweitlaufwerke (3 Zoll oder 5¼ Zoll) an. Sogar mit bis zu 1 MByte Speicherplatz. (Oliver Harms)

Die Qual der Druckerwahl



Spielen Sie mit dem Gedanken, sich einen Drucker zu kaufen? Dann stehen Sie vor der schwierigen Entscheidung, sich einen bestimmten Drucker unter hunderten auszuwählen. Es gibt aber fundamentale Eigenschaften, die ein geeignetes Gerät besitzen muß.

Jeder Drucker muß auf irgendeine Weise mit Ihrem Computer verbunden werden. Auf dem Markt gibt es zwei verschiedene Anschlußnormen:

- Drucker mit V24/RS232-Schnittstelle
- Drucker mit Centronics-Schnittstelle

Für Schneider-Computer sind nur Drucker mit Centronics-Schnittstelle geeignet. Sobald ein Gerät aber diesen Anschluß besitzt, können Sie jeden beliebigen Drucker wählen — egal ob er nun das Kürzel »CPC« im Namen führt oder nicht.

Es erleichtert die spätere Arbeit sehr, wenn Ihr Wunschdrucker den normalen »ASCII-Zeichensatz« beherrscht. Dazu gehören auch die Steuerzeichen mit den Codes 0D hex für »Wagenrücklauf« und 0Ahex für »Zeilenvorschub«. Sollte ein bestimmtes Zeichen (beispielsweise ein »A«) im Computer einen anderen Code haben als im Drucker, dann müssen Sie eine Übersetzungsroutine in das CPC-Betriebssystem einbauen. Diese Lösung ist aber wirklich nur etwas für Programmierspezialisten. Als Anfänger oder als Nur-Anwender werden Sie mit dieser

Lösung nicht glücklich werden. Es lohnt sich also, darauf zu achten, daß der Drucker den ASCII-Zeichensatz beherrscht. Wenn Sie keine allzu exotische Herstellerfirma wählen, dürfte es damit aber keine Schwierigkeiten geben.

Die Firma Schneider bietet zum Preis von etwa 50 Mark selbst ein zu Centronics-Druckern passendes Anschlußkabel an. Wenn Sie dieses aber nicht kaufen wollen, dann müssen Sie es sich selbst basteln. Das Anschlußschema finden Sie in den Handbüchern; das des Schneiders CPC 464 im Anhang 5 auf Seite 2.

Manche Drucker führen bei einem Wagenrücklauf (CR) automatisch einen Zeilenvorschub (LF) aus. Die Schneider-Computer senden jedoch zum Anfangen einer neuen Zeile ein separates Vorschubzeichen aus. Fehlt dieses, kann man zum Beispiel eine Zeile zweimal überdrucken. Ein Drucker mit automatischem Zeilenvorschub führt in Zusammenarbeit mit Ihrem Schneider den Zeilenvorschub doppelt aus. Das heißt, es wird jedesmal eine Leerzeile produziert.

Achten Sie also darauf, daß bei Ihrem Drucker dieser automatische Zeilenvorschub abschaltbar ist. Bei manchen funktioniert das durch Umschalten eines Dip-Schalters (meist im Inneren des Druckers). Bei anderen hilft das Unterbrechen der Leitung, die zum Anschluß mit der Nummer 14 des Centronics-Stek-

kers führt »AUTO FEED XT«-Leitung). Beim original Schneider-Anschlußkabel ist diese Unterbrechung schon vorhanden.

Wenn Sie mit Ihrem Computer nur Texte ausgeben wollen, dann genügt es, wenn er den Standard-ASCII-Zeichensatz beherrscht. Eventuell sollten Sie aber darauf achten, daß er auch den deutschen Zeichensatz kennt. Die Zeichen »@[\] | ~ « werden dann durch die Zeichen »\$ÄÖÜäöüß« ersetzt.

Wollen Sie aber alle Zeichen — das heißt auch die Sonderzeichen — auf dem Drucker ausgeben, dann sollte er die speziellen Schneider-Grafikzeichen ebenfalls im ROM haben. In der Regel sind das nur Drucker, bei denen der Satz »an den Schneider CPC 464/664/6128 angepaßt« explizit in der Funktionsbeschreibung auftaucht.

Manche Drucker erlauben es, einen eigenen Zeichensatz zu definieren. Lassen Sie sich mit dieser an sich positiven Eigenschaft jedoch nicht von einem Händler einwickeln, wenn sie auf die Schneider-Grafikzeichen Wert legen. Denn wenn Sie kein Profi sind, ist die eigene Definition von 127 Grafikzeichen eine rechte Fleißarbeit. Übrigens kann der »¼-Profi« eigene Zeichen auch ohne diesen frei definierbaren Zeichensatz selbst erzeugen: Man ändert einfach die Betriebssystemroutine für die Druckerausgabe so ab, daß vor einem Grafikzeichen der Drucker in den Grafikmodus umgeschaltet wird und dieses Zeichen im »Bitbildmodus« geplottet wird.

Achten sie jedoch darauf, daß all diese Funktionen mit Codes angesteuert werden, deren Wert kleiner als 127 ist. Der 7 Bit breite Druckerport des Schneider kann nämlich Werte größer als 128 nicht ausgeben.

Der Zeichensatz wird durch ein Steuerzeichen umgeschaltet

Im Schneider-Drucker NLQ401 wird dazu ein interessanter Trick benutzt: Durch einen Steuercode wird der normale Zeichensatz samt den normalen Steuerzeichen abgeschaltet und die Grafikzeichen mit Codes zwischen 128 und 255 auf die freigewordenen Codes mit kleinen Nummern gelegt. Erst durch ein weiteres Steuerzeichen wird der normale Modus wiederhergestellt. So können Sie die Verwendung des



Am Schneider-Drucker muß sich die Konkurrenz messen

achten Bits umgehen. Ihr Drucker sollte etwas Ähnliches aufweisen.

Wenn Sie später einmal vorhaben, eine Bildschirm-Hardcopy auszudrucken (auch wenn Sie jetzt noch nicht wissen, wie das funktioniert), dann muß Ihr Drucker grafikfähig sein. Und die nötige Software muß es zu kaufen geben oder das Gerät zumindest leicht zu programmieren sein.

Grafikfähige Drucker als Voraussetzung

Achten Sie aber auch hier darauf, daß die Grafikausgabe mit Codes, die einen Wert unter 127 haben, erfolgt. Bei vielen Druckern wird nämlich das achte Bit (Wert größer als 127) benutzt, das die Schneider gerade nicht angeben können.

Noch ein paar Hinweise sind wichtig: Investieren Sie ruhig 100 Mark mehr für einen schnellen Drucker. Während des Wartens auf das Ende eines Listings ist schon viel Wasser den Rhein hinuntergeflossen — besonders wenn der Drucker keinen Traktor hat und man vor dem Drucker ausharren muß, um alle halbe Minute ein neues Blatt von Hand einzuspannen.

Merken Sie sich: Je schöner die Schrift, desto langsamer ist der Drucker (bei konstantem Preis). Der Standard für Schnellschrift liegt zur Zeit bei etwa 100 Zeichen pro Sekunde — für Schönschrift bei etwa 20 Zeichen pro Sekunde.

Schätzen Sie vor dem Druckerkauf Ihren Papierbedarf und schla-

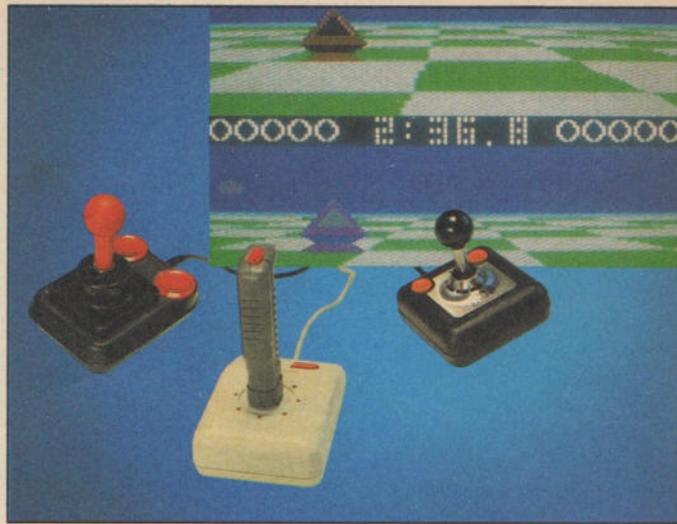
gen sie mindestens das Doppelte dazu (Sie haben noch gar keine Vorstellung davon, was so ein Drucker alles ausdrucken kann). Den Kaufpreis für den Drucker sollten Sie dann immer zusammen mit dem Preis für das eventuell notwendige Spezialpapier berechnen. Sie werden sich wundern, wie teuer dann ein ach so günstiger Thermo-Drucker wegen des Spezialpapiers wird. Übrigens: Spezialpapier ist grundsätzlich samstags, fünf Minuten nach Ladenschluß zu Ende. Es soll schon Leute gegeben haben, die Ihre Listings in so einem Fall auf Toilettenpapier ausgedruckt haben.

Wenn Sie Briefe schreiben wollen, dann sollten Sie auch darauf achten, daß Sie ungelochte Einzelblätter einspannen können. Nicht alle Drucker lassen sich mit einem Einzelblatteinzug aufrüsten. Das Einspannen einzelner Blätter wird mit wechselnder Zahl der Briefe lästig. Und alle Extras kosten Geld.

Farbe lohnt sich selten, kostet viel

Ein Farbdrucker lohnt sich nur, wenn Sie viele Grafiken ausdrucken möchten und sich gut auskennen. Ober Sie bekommen die entsprechende Grafik-Software gleich mitgeliefert. Aber auch in diesem Fall sollten Sie sich genauestens über den Preis und die Haltbarkeit des Spezialfarbbands im Vergleich zu einem schwarzen Farbband informieren. (Helmut Tischer)

Was Sie schon immer über Joysticks wissen wollten



**664
6128
464**

Eine Joystickbuchse hat Ihr Computer auch. Aber was ist, wenn Sie zu zweit spielen wollen? Der Schneider-eigene Joystick ist für Spielefans keine Lösung.

Ohne ihn sind Computerspiele nur halb so schön. Fast unentbehrlich für die Bedienung von Grafikprogrammen, ist er auch als Alternative zur Maus nicht zu unterschätzen: der Joystick.

Wer aber weiß schon genau, wie so ein »Steuerknüppel« funktioniert! Wie erfaßt der Computer die Daten? Was ist zu tun, wenn der Joystick seinen Geist aufgibt? Gerade bei den Schneider-Computern gibt es hier Probleme. Es funktionieren nämlich einige handelsübliche Joysticks nicht auf Anhieb. Auch können nicht ohne weiteres zwei Steuerknüppel gleichzeitig betrieben werden — es sei denn, man benutzt die original »Schneider-Sticks«.

Machen Sie mit uns eine Reise durch das Innenleben eines Joysticks, und sehen Sie, wie einfache Probleme gelöst werden können.

Einsteigen wollen wir mit dem Joystickport, an dem die Daten ankommen und an den Computer übergeben werden.

Ein kleines Programm, um den Port richtig testen zu können, ist hier recht nützlich: »10 PRINT JOY(0), JOY(1) : GOTO 10«.

Nachdem es mit »RUN« gestartet wurde, erscheinen auf dem Bild-

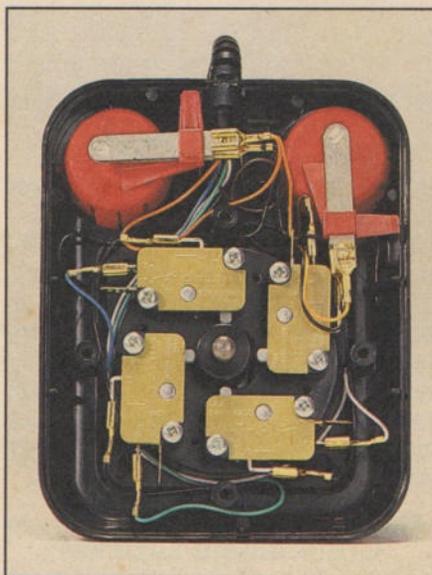


Bild 2. Ein Joystick mit Mikroschaltern

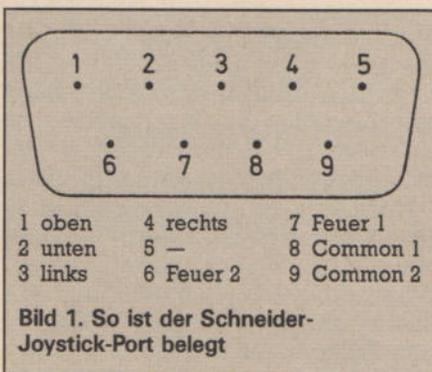


Bild 1. So ist der Schneider-Joystick-Port belegt

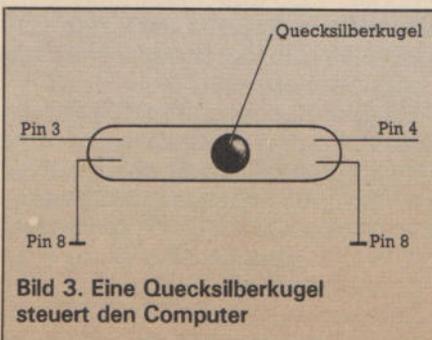


Bild 3. Eine Quicksilberkugel steuert den Computer

schirm zwei Zahlenreihen. Die linke zeigt den aktuellen Wert für den Joystick 1 an, die rechte Reihe für den Joystick 2. Zum Testen nehmen Sie nun ein kurzes Stück Kabel und verbinden die Pins des Joystickports beliebig miteinander. Dazu halten Sie einfach die Kabelenden daran — keine Angst, es kann Ihnen und dem Computer nichts passieren. Betrachten Sie nun die auf dem Bildschirm angezeigten Werte, so stellen Sie fest, daß nur bestimmte Kombinationen einen anderen Wert als 0 ergeben. Diese Werte ungleich 0 sind Eingangsdaten, die nach folgenden Voraussetzungen entstehen: Auf Bild 1 sehen Sie die Belegung des Anschlusses von Pin 1 bis Pin 9. Halten Sie Ihr Kabel nun einmal mit einem Ende an Pin 8 und mit dem anderen der Reihenfolge nach an die Pins 1 bis 7. Sie werden sehen, daß auf dem Bildschirm nun unterschiedliche Werte für Joystick 1 (linke Zahlenreihe) angezeigt werden. Diese Werte sind in der Variablen JOY(0) abgelegt. Legen Sie nun das Kabel von Pin 8 auf Pin 9 und wiederholen Sie den Vorgang.

Sie sehen, daß sich nun verschiedene Werte für den anderen Joystick (JOY(1)) ergeben haben. Tabelle 1 zeigt, welche Werte durch Verbindung der Pins 1 bis 7 mit Pin 8 und 9 auftreten. Verbindet man mehrere Pins (1 bis 7) mit den Massepins (8 oder 9), so wird der Variablen JOY(x) die Summe der Einzelwerte zugewiesen. Auf diese Weise lassen sich nun alle Zahlen zwischen 0 (keine Verbindung) bis 127 darstellen. Diese Variante, Daten über den Joystickport an den Computer zu übermitteln, eröffnet neue Wege der Dateneingabe. Wir wollen uns jedoch hier nur mit dem Anschluß von Joysticks beschäftigen. Im Inneren solch eines Peripheriegerätes ge-

schiebt nichts anderes, als daß verschiedene Kontakte geschlossen werden, die vom Computer als Richtung und »Feuer« interpretiert werden (Tabelle 2). Zwischenrichtungen (beispielsweise links oben) werden durch gleichzeitiges Verbinden der entsprechenden Kontakte (aus links = 4 und oben = 1 ergibt sich links oben = 5) erreicht.

Wie nun ein Kontakt geschlossen wird, daran unterscheiden sich die verschiedenen Joysticks. Hier ist dem Erfindungsgeist kaum eine Grenze gesetzt. Eine der stabilsten Lösungen ist die Verwendung von Mikroschaltern. Diese robusten Taster versprechen eine lange Nutzungsdauer und lassen sich bei einem Defekt problemlos ausbauen und erneuern (Bild 2). Aber Qualität hat auch ihren Preis und deshalb sind solche Joysticks (zum Beispiel der Competition Pro) auch nicht billig. Die sogenannten Metallzungenkontakte, eine häufig verwendete Bauart, sind im Bild 2 an den Feuerknöpfen zu sehen. Diese Art von Kontakten werden bei vielen Joysticks auch für die Richtungsschaltung verwendet. Die Haltbarkeit erwies sich zwar als nicht so hoch wie bei Mikroschaltern, was der niedrigere Preis aber ausgleicht.

Eine ganz andere Konstruktion sind sogenannte Quecksilberschalter. Solch ein Schalter besteht aus einem Glasröhrchen, in dessen Inneren sich ein Quecksilbertropfen befindet. Je nach Neigung fließt diese an ein Ende des Röhrchens und verbindet dort zwei Kontakte. Der Strom kann fließen (Bild 3). Diesen Schalter findet man zum Beispiel im »The Stick«, der nur aus einem Griff besteht und dadurch etwas gewöhnungsbedürftig ist.

Es gibt also verschiedenste Funktionsprinzipien. Wie gesagt, funktionieren nicht alle direkt am Schneider. Verschiedene Computer fassen nämlich die Steuersignale verschieden auf. Das sollte aber kein Grund sein vom Kauf solch eines Joysticks zurückzuschrecken. Ist das Funktionsprinzip einmal verstanden, braucht man den Joystick nur noch aufschrauben und die entsprechenden Kontakte auf Schneider-Norm umstecken beziehungsweise umlöten. Welches Kabel an welchen Kontakt führt, das stellen wir am einfachsten mit einem Widerstandsmeßgerät fest. Alternativ reicht auch eine Taschenlampenbirne mit Kabel und Batterie. Pin 1 muß mit dem Schalter »oben« verbunden werden. Dazu reicht meistens einfach das Umstecken — gelegentlich

läßt sich jedoch der Einsatz eines Lötkolbens nicht vermeiden. Die anderen Kabel werden genauso behandelt. Pin 2 ist dabei mit »unten«, Pin 3 mit »links«, Pin 4 mit »rechts«, Pin 6 mit »Feuer 2« und Pin 7 mit »Feuer 1« belegt. Pin 8 liefert Masse für den ersten Joystick, Pin 9 für den zweiten. Man kann sich also zwei verschiedene bauen. Tun Sie das aber nicht. Wir wollen den Adapter so gestalten, daß er die Auswahl (1 oder 2) macht. Wir benützen für Masse nur Pin 8. Eins sollte man noch bedenken: Beim Aufschrauben erlischt die Garantie für den Joystick.

Grundsätzliche Schwierigkeiten gibt es beim »Quickshot II«. Dieser Joystick ist mit einer »Autofeuer«-

Pin	8	9
1	1	1
2	2	2
3	4	4
4	8	8
5	64	64
6	16	16
7	32	32

Tabelle 1. Die Werte der verschiedenen Kontakte

Funktion	Joystick 1 Masse = Pin 8	Joystick 2 Masse = Pin 9
oben	1	1
unten	2	2
links	4	4
rechts	8	8
Feuer	16	32

Das Signal von Pin 5 (Wert=64) wird für die Joystickabfrage nicht benötigt.

Tabelle 2. Die Werte der Joystickabfrage

Funktion ausgestattet. Das »Autofeuer« wird von einer Triggerschaltung erzeugt, die eine Spannung von 5 V benötigt. Der Schneider gibt, im Gegensatz zum C64, diese über den Joystickport nicht aus. Somit ist dieser Joystick für den Schneider-Besitzer von vorneherein uninteressant. Höchstens man legt zwischen Hauptstromversorgung und 5-V-Eingang ein Kabel.

Was jetzt noch fehlt ist der Adapter für den Anschluß von zwei Joysticks. Zuerst besorgt man sich einen 9poligen weiblichen Stecker wie er am Joystick-Kabel zu finden ist. Ferner zwei männliche Stecker und 14 kurze Kabelstücke. An die männlichen Stecker werden die Joysticks gesteckt, der weibliche wird auf die Buchse des Computers gesteckt. An die Pins 1 bis 4 und 6 sowie 7 des weiblichen Steckers lötet man jeweils zwei Drähte, an die Pins 8 und 9 hingegen nur einen. Pin 5 bleibt unbenutzt. Die Pins 1 bis 4, 6 und 7 der männlichen Stecker werden direkt mit einem der jeweils zwei Kabel des weiblichen Steckers verbunden. Der Draht von Pin 8 gehört zum Stecker des Joystick 1 (auch Pin 8), der von Pin 9 wird mit Pin 8 des Joystick 2 verlötet. Die Belegung finden Sie im Bild 4. Zum Austesten dient wieder unser kleines Programm vom Anfang. Bedingung dafür, daß die Schaltung funktioniert, ist, daß der Joystick an Pin 1 bis 4 die Richtung, Pin 6 und 7 Feuer und Pin 8 Masse zur Verfügung stellt. Bei den Schneider-Joysticks benutzt der erste nur »Feuer 1«, der zweite hingegen »Feuer 2«.

(Alfred Otto)

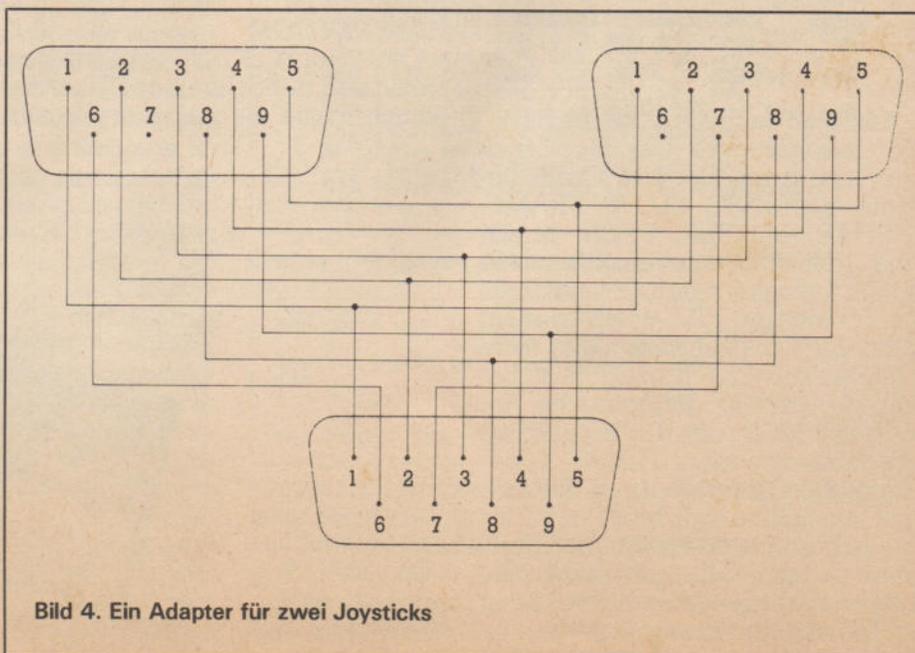


Bild 4. Ein Adapter für zwei Joysticks

Mehr Speicher für den CPC



Eine Erweiterung, die den Speicher des »CPC 464« bis auf 576 KByte aufbläst, sorgt dafür, daß größere CP/M-Programme nun auch auf dem kleinsten Schneider-Computer laufen.

Die Vortex-Speichererweiterung für den CPC 464 wird in verschiedenen Ausbaustufen geliefert. Eine Erweiterung mit 512 KByte kostet 589 Mark, 320 KByte sind für 528 Mark zu haben, bei 256 KByte reichen schon 478 Mark, bei 128 KByte 348 Mark und bei 64 KByte 275 Mark.

Neben der eigentlichen Karte, auf der auch bei den kleinen Ausführungen die Sockel für den Vollausbau von 512 KByte vorhanden sind, bekommt man ein dünnes Handbuch, eine Kassette mit der Systemsoftware, ein Kühlblech und eine Isolierfolie. Die Erweiterung wird in den Computer eingebaut, was jedoch wegen der guten Bauanleitung nicht allzu schwer ist. Leider erlöschen dadurch die Garantieansprüche an den Computer-Hersteller. An Werkzeug wird lediglich ein Kreuz- und ein Schlitzschraubendreher benötigt.

Ob alles ordnungsgemäß funktioniert, erkennt man gleich an der neuen Einschaltmeldung. Bei der hier vorgestellten 256-KByte-Version erscheint unterhalb der üblichen Einschaltmeldung ein Rechteck mit folgendem Inhalt:

vortex	RAM-Expansion	Card ID
5	160	128
Bks	Prgm	Data
		Spl

Jetzt sind auch die 37 neuen Basic-Befehle verfügbar. Darunter befinden sich beispielsweise Grafik-Befehle wie FAST, SLOW, FRAME, GCHAR, GRAPER, GPEN, MASK und UNMASK, die aus dem 464 schon beinahe einen 664 machen. Leider fehlt der FILL-Befehl, der auf dem CPC 464 softwaremäßig nicht ohne weiteres nachzubilden ist. Bemerkenswert ist dagegen die Anweisung FAST, die die Bildschirmausgabe um den Faktor 2 beschleunigt. Alle anderen Befehle (mit Ausnahme von MON) sind für die Arbeit mit dem erweiterten Speicher gedacht. Mit ihnen kann man beispielsweise eine relative Datei im Speicher eröffnen und verwalten. Erstmals lassen sich auf dem CPC

464 Unterprogramme mit lokalen und globalen Variablen schreiben. Leider ist es unmöglich, die einzelne Variable als COMMON (allgemein gültig) oder lokal zu definieren. Es werden immer ganze Bereiche behandelt, wie bei den bekannten Funktionen DEF INT, DEF STR und DEF REAL des Schneider-Basic. Der Zeitbedarf für das Umschalten zwischen den einzelnen Bänken ohne Variablenübertragung liegt bei $\frac{1}{125}$ Sekunden pro Schaltvorgang. Somit können verschiedene Programme gleichzeitig im Speicher gehalten und getrennt bearbeitet werden. Der Befehl SPOOL.ON richtet einen 32 KByte großen Druckerpuffer ein, der zwar den Hauptspeicher verkleinert und zu Lasten der Rechengeschwindigkeit geht, aber diese Einbußen lassen sich, in Anbetracht der deutlich geringeren Wartezeiten beim Ausdrucken von Dateien und Listings, leicht verschmerzen.

Doch nun zum CP/M: Zuerst muß eine neue Systemdiskette für das CP/M mit 62 KByte Speicherplatz erzeugt werden. Man erhält dann eine Diskette mit folgenden Files:

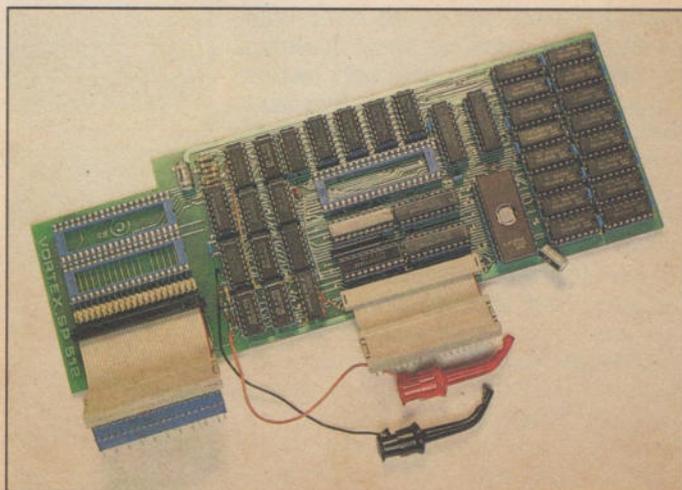
\$OSC.SYS	12 K
SPOOL.COM	1 K
RAMDISK.COM	1 K

Die Datei \$OSC.SYS wird bei den Benutzern der Schneider-Diskettenstation keine Begeisterung auslö-

sen, denn sie verkleinert den ohnehin knappen Speicherplatz noch weiter. Alle weiteren Systemdisketten werden nun als Kopie dieser Diskette bevorzugt. Das Programm »Sptest« überprüft die einzelnen Speicherplätze auf Fehler. Dabei wird jede Adresse des erweiterten Speichers mit einem Wert beschrieben und später wieder gelesen. Stimmt der ausgelesene Wert nicht mit dem eingeschriebenen überein, so ist die Speicherstelle defekt. »Sptest« meldet das mit Angabe der Speicherstelle, dem Soll- und dem Istwert. Auf diese Weise werden alle vorhandenen Bänke getestet.

Es bleibt die Frage der Kompatibilität zu den alten Programmen. Diesbezüglich von der Firma Schneider nicht unbedingt verwöhnt, sind die Erwartungen an die Vortex-Erweiterung ziemlich hoch angesetzt. Alle Basic-Programme arbeiten problemlos. Unter CP/M ist das anders, denn hier stellen sich manchmal ungewöhnliche Effekte ein. So kann man normalerweise mit der Anweisung »DISCCOPY« auch unformatierte Disketten beschreiben. Nach dem Einbau der Speichererweiterung wird die Formatier-Routine nicht mehr aufgerufen. Auch Programme, die die letzten vier Byte unter HIMEM (Adressen 42615 bis 42619) benutzen, laufen nicht mehr korrekt. Diese vier Adressen benötigen nämlich die Systemerweiterung.

Das abschließende Urteil kann nur lauten: Sehr empfehlenswert für alle, denen 64 KByte RAM einfach zu wenig sind. Das weitere Aufrüsten der kleineren Versionen der Karte ist problemlos mit dem RAM- und einem Logik-Baustein möglich. Auch für den 664 sind die Erweiterungen zum gleichen Preis erhältlich — allerdings ohne Basic-Erweiterung. Diese wird 1986 als EPROM für zirka 40 Mark nachgeliefert. (Alfred Otto)



Die ganze Platine verschwindet im Gehäuse

Keine Eingabefehler mit »Explora«



Mit einer Prüfsumme wird jede Zeile, die Sie eingeben, überwacht. Fehler im Listing sind damit fast unmöglich.

Wenn Sie das Programm »Explora« abtippen, haben Sie eine wertvolle Eingabehilfe. Eine Maschinencode-Routine überwacht Ihre Arbeit daraufhin, ob sämtliche Zeichen (auch Steuersymbole) sowie Leerstellen und Zeilennummer korrekt im Speicher stehen. Nach Beenden einer Zeile mit Enter wird direkt in die untere linke Ecke des Bildschirms die vierstellige Hexadezimalzahl angezeigt, die Sie im Listing in der eckigen Klammer neben jeder Programmzeile finden.

Voraussetzung für die Überwachungsfunktion ist allerdings, daß Sie die Programmzeile genauso eingeben, wie sie abgedruckt ist. Abkürzungen, die vom Interpreter auch verstanden werden, dürfen Sie nicht benutzen (also kein »?« für »PRINT«). Auch müssen Sie große und kleine Buchstaben wie vorgegeben eintippen. Der Interpreter würde für »PRINT« auch »print« akzeptieren — Explora hingegen nicht. Steuerzeichen und mehrere Leerzeichen, die in Strings aufeinander folgen, sind in geschweiften Klammern im Klartext angegeben. So bedeutet [5 Space], daß an dieser Stelle fünfmal die Leertaste gedrückt werden muß. [CTRL A] bedeutet, daß die Ctrl-Taste gemeinsam mit dem »A« gedrückt werden muß (siehe im Beispiellisting Zeile 430 und 440). Aber Vorsicht, daß Sie solch ein übersetztes Zeichen nicht mit dem ASCII-Sonderzeichen »{« beziehungsweise »}« verwechseln. Die Bedeutung der geschweiften Klammer erkennen Sie aber leicht, denn als ASCII-Sonderzeichen steht sie meist allein. Im anderen Fall umschließt sie Control-Zeichen oder Leerfelder. Der AUTO-Befehl darf übrigens nicht verwendet werden, da sonst die Prüfsumme falsch berechnet wird.

Da die Tastatur der Schneider-Computer sehr leicht umdefiniert werden kann, arbeitet jeder Benutzer mit einer anderen Tastenbelegung. Alle Listings in diesem Heft sind deshalb mit dem ASCII-Zeichensatz ausgedruckt. Deutsche Sonderzeichen werden dabei als Klammern oder andere amerikanische Sonderzeichen interpretiert. Benutzen Sie eine deutsche Tastatur, so dürfen Sie anstelle dieser Zeichen die deutschen benutzen. Explora merkt dies. Welche amerikanischen und deutschen Zeichen sich entsprechen finden Sie in Tabelle 1. Das Zeichen »~« (für das »ß«) wird mit Ctrl 2 aufgerufen.

Listing 1 ist die Routine für die Prüfsumme. Diese liegt ab Adresse 40960 im Speicher. Das Basic-Lader darf gelöscht werden. Eingeschaltet wird Explora mit »POKE &A01F,&F5«, ausgeschaltet mit »POKE &A01F,&C9«. Probleme kann es nur bei Listings geben, die ein Maschinencode-Programm (das sind die Basic-Lader) erzeugen. Eventuell funktioniert der MEMORY-Befehl nicht richtig. In diesem Fall darf er ersatzlos gestrichen werden. Beim Speichern der Binärfelder müssen alle Adressen aber genau beachtet werden.

Der Übersicht halber sind die Zeilen-Nummern nach links herausgezogen. Die eigentlichen Befehle beginnen immer nach einer Leerstelle hinter der Zeilennummer. Listing 2 zeigt fünf Zeilen als Beispiel. Übrigens: Alle Programme können auch wie gewohnt, ungeprüft, eingegeben werden. (hg)

Sonderzeichen	
amerikanische	deutsche
@	§
[Ä
\	Ö
]	Ü
{	ä
	ö
}	ü
~	ß

Das Symbol »~« steht für »!«

Tabelle 1. Die deutschen und die amerikanischen Sonderzeichen im Vergleich

```

10 * *****
20 *
30 *   Fuer Schneider CPC
40 *   464, 664 und 6128:
50 *
60 *   Happy-Computers
70 *
80 *   -----
90 *   Explora 1.0
100 *
110 *
120 *   (c) Martin Kotulla
130 *
140 * *****
150 MEMORY 40959
160 FOR i=40960 TO 41094:READ a:POKE i,a
:NEXT i
170 POKE &160,&CD:POKE &161,&0:POKE &162
,&B9:POKE &163,&3A
180 POKE &164,&2:POKE &165,&C0:POKE &166
,&32:POKE &167,&6D
190 POKE &168,&1:POKE &169,&C9:CALL &160
200 cpcversion=PEEK(&16D)
210 IF cpcversion=0 THEN 290
220 IF cpcversion>1 THEN 260
230 POKE &A006,&5B:POKE &A013,&5B:POKE &
A019,&5C
240 POKE &A024,&8A:POKE &A035,&8A
250 GOTO 290
260 IF cpcversion<>2 THEN PRINT "Kein CP
C-464/664/6128!":END
270 POKE &A006,&5E:POKE &A013,&5E:POKE &
A019,&5F
280 POKE &A024,&8A:POKE &A035,&8A
290 PRINT:PRINT:PRINT "Checksummer ist a
ktiviert!"
300 PRINT:PRINT "Einschalten: POKE &A01F
,&F5"
310 PRINT:PRINT "Ausschalten: POKE &A01F
,&C9":PRINT
320 CALL &A005 ' Checksummer einschalten
330 DATA &00,&00,&00,&00,&00,&21,&3A,&BD
,&11,&02,&A0,&01,&03,&00,&ED,&B0
340 DATA &3E,&C3,&32,&3A,&BD,&21,&1C,&A0
,&22,&3B,&BD,&C9,&CD,&02,&A0,&F5
350 DATA &C5,&D5,&E5,&01,&A4,&AC,&21,&00
,&00,&0A,&5F,&16,&00,&19,&03,&FE
360 DATA &00,&20,&F6,&DD,&21,&A4,&AC,&01
,&00,&00,&DD,&7E,&3E,&5F,&16,&00
370 DATA &19,&04,&F5,&A8,&47,&F1,&09,&DD
,&23,&FE,&00,&20,&ED,&3E,&0D,&CD
380 DATA &5A,&BB,&3E,&0A,&CD,&5A,&BB,&3E
,&5B,&CD,&5A,&BB,&7C,&CD,&76,&A0
390 DATA &7C,&CD,&7A,&A0,&7D,&CD,&76,&A0
,&7D,&CD,&7A,&A0,&3E,&5D,&CD,&5A
400 DATA &BB,&E1,&D1,&C1,&F1,&C9,&1F,&1F
,&1F,&1F,&E6,&0F,&C6,&30,&FE,&3A
410 DATA &3B,&02,&C6,&07,&C3,&5A,&BB
420 END

```

Listing 1. »Explora« macht Fehler fast unmöglich

```

400 DATA &BB,&E1,&D1,&C1,&F1,&C9,&1F,&1F
,&1F,&1F,&E6,&0F,&C6,&30,&FE,&3A [C4C0]
410 DATA &3B,&02,&C6,&07,&C3,&5A,&BB [633C]
420 NEW [EE40]
430 PRINT "{CTRL A}{CTRL Y}{CTRL Y}{CTRL
A}" [8DBB]
440 PRINT "{5 SPACE}WW{" [BCDE]

```

Listing 2. Im Beispiel müßten Sie die Zeile 400 wie folgt eingeben (MODE 1):

```
400 DATA &BB,&E1,&D1,&C1,&F1,&C9,&1F,&1F
,&1F,&1F,&E6,&0F,&C6,&30,&FE,&3A.
```

Zeile 430 besteht in der PRINT-Anweisung aus den vier Tastendruckern Ctrl A, Ctrl Y, Ctrl Y und Ctrl A. Zeile 440 aus dem String " WW{"

Datenfernübertragung muß nicht teuer sein



Zur Datenfernübertragung brauchen Sie einen Computer, ein Modem, eine Verbindung zwischen beiden und ein DFÜ-Programm. Den Computer haben Sie und das Modem müssen Sie kaufen. Den Rest bekommen Sie hier.

Die hier vorgestellte Schnittstelle ist speziell für den 300-Baud-Betrieb ausgelegt. Trotzdem kann sie per Software auf 1200 Baud umgestellt werden.

Der Preis für unsere Bastelei richtet sich hauptsächlich nach dem eingesetzten Spannungswandler. Mit Bausteinen der Serie 75188 und 75189 ist sie günstiger (maximal 100 Mark). Allerdings braucht man dann von außen Spannungen und zwar +12 Volt und -12 Volt. Mit dem integrierten Spannungswandler NM232C steigt zwar der finanzielle Aufwand um zirka 30 bis 40 Mark, aber man braucht auch nur eine Stromversorgung mit 5 Volt.

Kernstück der Schnittstelle ist der integrierte Seriell/parallel-Wandler 6850. Er wird über zwei Register programmiert und bedient. Von Basic aus geht das mit den Befehlen IN und OUT. Ein weiterer wichtiger Teil der Schaltung ist der Frequenzgenerator. Er erzeugt mit einer Standard-Quarz-Schaltung die Frequenz von 2,4576 MHz. Diese liefert, je

nachdem wie oft sie geteilt wird, alle nötigen Baudraten. Der 2x4-Bit-Zähler 74LS393 teilt die 2,4576 MHz auf die Frequenz von 19,2 kHz. Vom 6850 werden diese 19,2 kHz noch einmal geteilt, so daß letztendlich die Baudrate von 300 (bei Teilung durch 64) oder von 1200 (bei Teilung durch 16) zur Verfügung steht.

Für die V.24-Pegel brauchen wir einen Spannungswandler. Entweder den integrierten Spannungswandler NM232C oder Standard-V.24-Treiberbausteine, von denen einer allerdings mit +12 und -12 Volt versorgt werden muß. Das noch eingesetzte IC 74LS08 dient zur Adreßdecodierung.

Beim Schreiben in das Controlregister (Adresse FBF0hex) müssen bestimmte Parameter gesetzt werden.

Bit 0 und 1 bestimmen die Teilerrate des Chips. Hier kann also per Software festgelegt werden, wie oft die Frequenz, die an den Pins 3 und 4 anliegt, geteilt werden soll. Die Werte finden Sie in Tabelle 1.

Bit 1	Bit 0	
0	0	Die Teilerrate ist 1. Diese Teilerrate ist bei unserer Anwendung nicht zu empfehlen, da es bei der asynchronen Datenübertragung zu »Interferenzstörungen« kommt.
0	1	Teilerrate 16. Das bewirkt bei unserer Schaltung eine Baudrate von 1200 Baud (beim Senden und Empfangen).
1	0	Teilerrate 64 hat eine effektive Rate 300 Baud zur Folge.
1	1	Master Reset, wird in Bit 1 und Bit 0 eine Eins geschrieben, wird der 6850 vollständig zurückgesetzt (empfehlenswert am Beginn eines Programms).

▲ Tabelle 1. Bit 0 und 1 bestimmen die Baudrate
Tabelle 2. Bit 2, 3 und 4 steuern das Übertragungsformat ▼

Die Bits 2 bis 4 bestimmen das Übertragungsformat der Schnittstelle (siehe Tabelle 2).

Bit 4	Bit 3	Bit 2	
0	0	0	7 Bits + Even Parity + 2 Stop-Bits
0	0	1	7 Bits + Odd Parity + 2 Stop-Bits
0	1	0	7 Bits + Even Parity + 1 Stop-Bit
0	1	1	7 Bits + Odd Parity + 1 Stop-Bit
1	0	0	8 Bits + 2 Stop-Bits
1	0	1	8 Bits + 1 Stop-Bit
1	1	0	8 Bits + Even Parity + 1 Stop-Bit
1	1	1	8 Bits + Off Parity + 1 Stop-Bit

Die von Mailboxen am häufigsten verwendeten Übertragungsformate sind 100 (8 Bit und 2 Stop-Bit) und 101 (8 Bit und 1 Stop-Bit).

Die Bits 5 und 6 müssen bei normalen asynchronen Anwendungen beide 0 sein. Wird in beiden eine 1 geschrieben, dann wird ein Break-Signal geschickt, was nichts anderes bedeutet, als daß die V.24-Sendeleitung für eine bestimmte Zeit auf einen festen Pegel gelegt wird. Bit 7 ist ebenfalls immer 0.

Wenn Sie die Schnittstelle fertig aufgebaut haben, dann sollten Sie erst noch einmal alles prüfen. Eine gute Testmöglichkeit von seriellen V.24-Schnittstellen ist das Verbinden der Sende- und Empfangsleitungen (Pin 2 und 3), so daß die Schnittstelle sich selber Daten schickt. Listing 1 testet Ihre Schnittstelle.

Nach dem Anschließen der Schnittstelle und dem Starten dieses Testprogramms müssen gedrückte Tasten als Buchstaben auf dem Bildschirm erscheinen. Mit solch einem Programm können bereits über einen Akustikkoppler Mailboxen bedient werden.

Die Programmierung des 6850

Für die Schaltung gibt es zwei Anschlußmöglichkeiten. Entweder am herausgeführten Datenbus oder im Innern des Computers direkt auf dem Sockel des Z80. Der Z80 selbst wird dann in die Schaltung integriert. In beiden Fällen werden alle benötigten Signale erreicht. Beide Lösungen haben aber ihre Schwächen.

Der Anschluß am Datenbus führt leicht zu einer wackeligen Lösung. Auch die Diskettenstation ist schwierig gleichzeitig mit anzustecken. Der Haken bei der Einbaulösung ist, daß eventuell noch bestehende Garantie erlischt. Die Schaltung ist leicht in der Wire-Wrap-Technik aufzubauen, genauso einfach ist allerdings auch eine gelötete Verdrahtung auf einer Lochrasterplatine. Für den Einbau der Schaltung in den Computer ist ein 40poliger Wire-Wrap-Sockel notwendig, da die Beinchen lang genug sein müssen, um in den Sockel zu passen.

Wie bereits oben gesagt, hat der 6850 zwei Register, die beide sowohl beschrieben als auch gelesen werden können. In das eine Register, das Datenregister, werden die zu übertragenden Daten hineingeschrieben. Beim Empfang werden hieraus die Daten abgeholt. Es hat

die Adresse FBF1hex. Das andere Register dient zur Steuerung des Bausteins, es ist das Control-Register. Seine Adresse ist hexadezimal FBF0. Die Bedienung dieses Registers ist leider etwas kompliziert. Beim Lesen aus dem Register erhält man folgende Informationen (bitweise aufgeschlüsselt, Tabelle 3):

Wenn der Buffer voll ist, so werden die einzutragenden Zeichen ignoriert, das heißt sie gehen verloren. Allerdings müßten bei einer 300-Baud-Übertragung 2,2 Minuten lang ununterbrochen Daten eintreffen, um den Buffer wirklich ganz zu füllen.

Das Maschinencode-Modul muß

fehle einzugeben, ohne daß dabei der untere Teil des Bildschirms zerstört wird. In den Befehlseingabemodus gelangt man durch zweimaliges Drücken der Taste Esc. Die dann einzugebenden Anweisungen sind weitgehend am CP/M-Betriebssystem orientiert. Man kann durch die Bufferstruktur beliebig lang in diesem Fenster arbeiten und Befehle ausführen lassen, ohne daß dabei Daten verlorengehen. Voraussetzung ist allerdings, daß der Host-Rechner der Mailbox auf das XON/XOFF-Protokoll vorschriftsmäßig reagiert.

Im Fenster können folgende Befehle aufgerufen werden (Eingabe wie unter CP/M):

A: Das Laufwerk A ist ab sofort Standardlaufwerk.

B: Umschalten auf Laufwerk B.

BYE: Verlassen des Programms.

DIR: Das Inhaltsverzeichnis der Diskette wird angezeigt. Dabei sind wie beim CP/M auch einzelne Dateinamen mit den Wildcards * und ? möglich.

DL CLOSE: Die Datei wird geschlossen und auf Diskette endgültig angelegt.

DL OFF: Download-Modus aus. Mit DL ON und DL OFF kann das Übertragen der empfangenen Daten in die durch DL OPEN spezifizierte Datei gestoppt und gestartet werden.

DL ON: Der Download-Modus wird angeschaltet, vorher muß der Befehl DL OPEN ausgeführt worden sein. Alle Zeichen, die vom Computer empfangen werden, werden in die durch DL OPEN spezifizierte Datei geschrieben. Braucht der Computer Zeit, um empfangene Daten abzuspeichern oder zu bearbeiten, dann stoppt er das Senden des Hosts mit XOFF und startet es danach wieder mit XON.

DL OPEN: Die Download-Datei wird geöffnet.

ERA: Wie CP/M-Befehl ERA, mit Dateinamen und Wildcards.

ESC: Ein Escape-Zeichen wird an den Host-Rechner geschickt.

FDX: Die Übertragung von Daten erfolgt im Fullduplex-Modus, das heißt der Host-Rechner muß empfangene Zeichen zur Bestätigung wieder zurückschicken.

HDX: Halbduplex — Der Host schickt empfangene Zeichen nicht zurück, das Terminalprogramm erzeugt sie selber.

PRN OFF: Schaltet den Drucker wieder aus. Zeichen, die sich noch im Druckerbuffer befinden, werden noch ausgegeben.

PRN ON: Der Drucker wird eingeschaltet (natürlich gebuffert), das

Bit 0	ist gesetzt, wenn das chipinterne Datenregister voll ist, also im Datenregister ein empfangenes Zeichen abgeholt werden kann.
Bit 1	ist gesetzt, wenn die Übertragung eines zu sendenden Zeichens fertig ist, das bedeutet, das nächste Byte darf in das Datenregister geschrieben werden, um übertragen zu werden.
Bit 2	Hier ist der logische Pegel von Pin 23 (Data Carrier Detect) zu finden. Es ist also immer nicht gesetzt.
Bit 3	Hier liest man den Zustand der Leitung CTS (Clear to Send), in diesem Beispiel immer gesetzt.
Bit 4	ist gesetzt, wenn ein sogenannter »Framing Error« erkannt wurde. Dann stimmt irgendetwas an den Übertragungsparametern nicht.
Bit 5	ist gesetzt, wenn ein Zeichen aus dem Datenregister nicht rechtzeitig geholt wurde und dieses mit dem neuen Zeichen überschrieben wurde (Overrun Error).
Bit 6	ist gesetzt, wenn beim Datenempfang ein Parity-Error erkannt wurde. Dazu muß allerdings die entsprechende Betriebsart eingestellt sein.
Bit 7	enthält eine logische Verknüpfung aus den Signalen CTS, TDRE, DCD, Overrun und so weiter (Interrupt request).

Tabelle 3. Die Informationen des Control-Registers

Das Maschinencode-Modul »FLIRTM.BIN« ist darauf ausgelegt, ein Datenfernübertragungsprogramm in Basic zu unterstützen. Es beginnt bei Adresse 7000hex, besitzt einen interruptgesteuerten Empfangsbuffer (4 KByte) und einen interruptgesteuerten Druckerbuffer (ebenfalls 4 KByte groß). Mit diesen Buffern wird die Bedienung der Schnittstelle zeitlich gesehen problemlos. Das Programm kann mit dem abgedruckten Basic-Lader generiert werden.

Die Bedienung des Moduls erfolgt über RSX-Befehle:

INIT

Mit diesem Befehl werden die Buffer initialisiert und die Schnittstelle auf das Übertragungsformat 8 Datenbit und 1 Stop-Bit gesetzt.

SEND,asc("a")

Mit diesem Befehl wird ein Zeichen über die Schnittstelle geschickt. Es wird dabei so lange gewartet, bis die Übertragung des vorhergehenden Zeichens beendet wurde.

RECV,@a

Mit diesem Befehl wird der interruptgesteuerte Empfangsbuffer ausgelesen. In der Integer-Variable a befindet sich der ASCII-Code des empfangenen Zeichens.

PRINT,asc("a")

Ein Zeichen wird an den Druckerbuffer übergeben. Das Drucken des Zeichens übernimmt die Interrupt-Routine.

am Anfang einmal mit dem Befehl CALL &7000 initialisiert werden. Von diesem Modul werden zusätzlich die Diskettenfehler »FILE NOT FOUND« und »DISK FULL« und so weiter abgefangen, damit beim Auftreten eines solchen Fehlers das Programm nicht abgebrochen wird. Tritt solch ein Fehler auf, so wird das Basic-Programm mit der Meldung »Error 50« fortgesetzt. Diese Meldung kann dann mit dem Befehl ON ERROR GOTO abgefangen werden. Das bedeutet aber, daß das Maschinencode-Modul den Ready-Modus nicht zuläßt. Es kann daher nur von Programmen aus aufgerufen und bedient werden. In der Error-Routine wird dann der Diskettenfehler durch den Inhalt der Variablen ERR erkannt.

Der Befehlssatz ist reichhaltig

Das Übertragungsprogramm aus Listing 3 enthält als Kern das beschriebene Maschinencode-Modul »FLIRTM.BIN«. Dadurch treten keine Geschwindigkeitsprobleme auf. Das Basic kann gemütlich die gebufferten Daten verarbeiten. Dieses Programm läuft nur auf dem CPC 464 mit Diskettenlaufwerk. Nach dem Starten finden Sie im oberen Teil des Bildschirms das »Kommando-Fenster«. Es wird benutzt um Be-

heißt alle Zeichen, die empfangen werden, werden auch auf dem Drucker mitprotokolliert.

TYPE: Wie CP/M-Befehl TYPE. Die angegebene Datei wird im Fenster ausgegeben. Die anzuzeigende Datei sollte im ASCII-Format vorliegen.

UL E: Upload-Echo-Mode — Eine Datei wird an den Host-Rechner geschickt. Das Unterprogramm wartet nach dem Senden eines Zeichens auf das Echo vom Host. Diese Betriebsart ist nur bei Fullduplex praktisch.

UL X: Upload mit XON/XOFF-Protokoll. Die Datei wird an den Host übertragen, dabei kann der Host die Ausgabe der Datei mit dem XON-XOFF-Protokoll stoppen und starten.

Hauptschaltung:

- 1 6850
- 1 74LS393
- 1 74LS04
- 1 74LS08
- 1 2,4576-MHz-Quarz
- 2 k Ω Widerstände
- 1 22 pF Kondensator

Spannungswandler:

entweder 1 NM232C-V.24-Treiber mit integriertem Spannungswandler oder 1 75189- und 1 75188-V.24-Treiber.

Sonstiges:

Sockel,
Platine und so weiter,
je nach Art des Aufbaus.

Durch die einfache Erkennung erleichtert, können beliebig viele zusätzliche Befehle (wie zum Beispiel deutscher Zeichensatz oder deutsche Tastatur) hinzugefügt werden. Dazu muß einfach eine Zeile mit der

Befehlsweiterungen

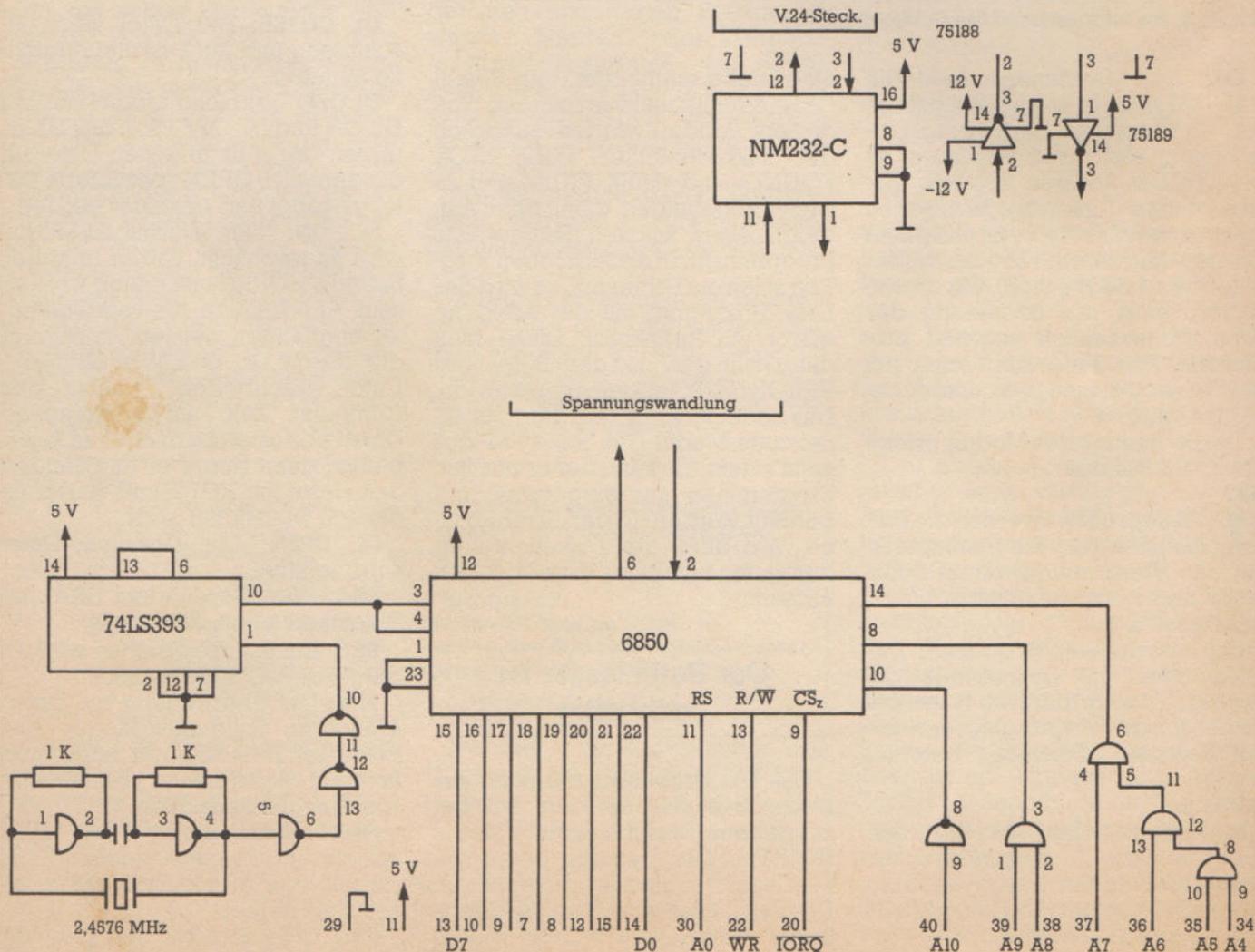
Syntax »IF Befehl\$="xyz" THEN GOSUB 2000« eingebaut werden.

Noch ein Hinweis zum Betrieb der Schnittstelle: Die Datenübertragungspins 2 und 3 des V.24-Steckers können, je nach Art des Geräts, das angeschlossen werden soll, auch vertauscht richtig sein. Deshalb sollte man auf der Platine die Möglichkeit vorsehen, die beiden Pins einfach vertauschen zu können.

(Thomas Obermair)

Tabelle 4. Die Bauteileliste

Bild 1. Der Schaltplan



```

10 DEFINT a-z [32C6]
20 d=&FBF1:REM Datenregister [1DAE]
30 c=&FBF0:REM Kontrollregister [6A7E]
40 OUT c,&X11:REM Master Reset [0DB0]
50 OUT c,&X10010:REM 300 Baud und 8 Date [75A0]
  nbits 2 Stopbits
60 a$=INKEY$:IF a$<>"" THEN OUT d,ASC(a$ [D5B8]
  )
70 a=INP(c):IF a AND 1 THEN a=INP(d):PRI [8F28]
  NT CHR$(d) [328E]
80 GOTO 60

```

Listing 1. Ein Testprogramm für RS232-Schnittstellen

```

10 PRINT [6C1C]
100 MEMORY &6FFF:zeile=1000 [CB52]
110 FOR x=&7000 TO &71D0 STEP 8:check=0 [D932]
120 FOR a=0 TO 7:READ d$:d=VAL("&"+d$):c [04C8]
  heck=check+d:POKE x+a,d:NEXT
130 READ d$:IF VAL("&"+d$)<>check THEN 2 [5556]
  00
140 zeile=zeile+10:PRINT CHR$(13);"Zeile [E9CC]
  ";zeile;"ok(6 SPACE)";:NEXT
150 PRINT:PRINT "alles ok." [E14E]
160 SAVE "flirtm.bin",b,&7000,&1D0 [1FBA]
170 PRINT "Datei flirtm.bin angelegt. ": [0F64]
  PRINT:END
200 PRINT CHR$(13)+CHR$(7)+"Fehler in Ze [2146]
  ile ";zeile:END [2CA0]
1000 DATA 01,73,70,21,8C,70,CD,D1,039F [F6C0]
1010 DATA BC,3E,C3,32,01,AC,21,90,034D [0CC6]
1020 DATA 70,22,02,AC,CD,C7,70,CD,0411 [2E40]
1030 DATA BC,70,CD,A9,71,C9,CD,9C,0545 [6C4A]
1040 DATA 70,CB,4F,CA,1E,70,DD,7E,043D [43DA]
1050 DATA 00,CD,B4,70,C9,CD,43,70,043A [0EB4]
1060 DATA DD,66,01,DD,6E,00,77,23,0329 [A712]
1070 DATA 3E,00,77,C9,DD,7E,00,CD,03A6 [B3BE]
1080 DATA 70,71,C9,CD,16,71,DA,59,0431 [E912]
1090 DATA 70,E6,7F,21,5C,70,BE,CA,044A [07C6]
1100 DATA 59,70,CB,7E,C0,23,C3,4E,0406 [D218]
1110 DATA 70,3E,00,C9,01,02,03,04,0181 [3032]
1120 DATA 05,06,0E,0F,10,11,12,13,006E [1370]
1130 DATA 14,15,16,17,18,19,1A,1C,00BD

```

```

1140 DATA 1D,1F,80,7E,70,C3,1E,70,02FB [D9E8]
1150 DATA C3,2D,70,C3,3C,70,53,45,0367 [A790]
1160 DATA 4E,C4,52,45,43,D6,50,52,0364 [B578]
1170 DATA 49,4E,D4,00,00,00,00,00,016B [3A34]
1180 DATA 1E,32,C3,94,CA,CD,90,71,043F [06F6]
1190 DATA CD,9C,71,C9,C5,01,F0,FB,0554 [B922]
1200 DATA ED,78,C1,C9,C5,01,F0,FB,05A0 [B124]
1210 DATA ED,79,C1,C9,C5,01,F1,FB,05A2 [E72E]
1220 DATA ED,78,C1,C9,C5,01,F1,FB,05A1 [592C]
1230 DATA ED,79,C1,C9,3E,03,CD,A4,04A2 [E82E]
1240 DATA 70,3E,16,CD,A4,70,C9,21,038F [34D4]
1250 DATA 00,80,22,DA,70,22,DC,70,035A [B68A]
1260 DATA 21,00,90,22,DE,70,22,E0,0323 [0548]
1270 DATA 70,C9,00,00,00,00,00,00,0139 [E3E8]
1280 DATA 00,00,C5,23,22,FA,70,01,0275 [D046]
1290 DATA FF,8F,37,3F,ED,42,20,05,0358 [CDF2]
1300 DATA 21,00,80,C1,C9,2A,FA,70,03BF [DDCA]
1310 DATA C1,C9,00,00,C5,23,22,14,02A8 [1260]
1320 DATA 71,01,FF,9F,37,3F,ED,42,03B5 [7900]
1330 DATA 20,05,21,00,90,C1,C9,2A,028A [3064]
1340 DATA 14,71,C1,C9,00,00,2A,DA,0313 [BC7C]
1350 DATA 70,ED,4B,DC,70,37,3F,ED,0457 [5C26]
1360 DATA 42,CA,31,71,2A,DA,70,7E,03A0 [ADD2]
1370 DATA CD,E2,70,22,DA,70,37,3F,0401 [D6C4]
1380 DATA C9,37,C9,2A,DE,70,ED,4B,0479 [BD3C]
1390 DATA E0,70,37,3F,ED,42,CA,4E,040D [2B14]
1400 DATA 71,2A,DE,70,7E,CD,FC,70,04A0 [B324]
1410 DATA 22,DE,70,37,3F,C9,37,C9,03AF [4206]
1420 DATA 2A,DC,70,CD,E2,70,ED,4B,04CD [8862]
1430 DATA DA,70,37,3F,ED,42,CA,6E,0427 [4418]
1440 DATA 71,2A,DC,70,77,CD,E2,70,047D [A3FC]
1450 DATA 22,DC,70,37,3F,C9,37,C9,03AD [5A06]
1460 DATA 2A,E0,70,CD,FC,70,ED,4B,04EB [1C6A]
1470 DATA DE,70,37,3F,ED,42,CA,8E,044B [E046]
1480 DATA 71,2A,E0,70,77,CD,FC,70,049B [AE04]
1490 DATA 22,E0,70,37,3F,C9,37,C9,03B1 [1FC6]
1500 DATA CD,2E,BD,08,CD,33,71,D8,04D9 [0E64]
1510 DATA CD,31,BD,C9,CD,9C,70,E6,0543 [443E]
1520 DATA 01,C8,CD,AC,70,CD,50,71,0440 [6DD8]
1530 DATA C9,F3,01,88,71,ED,43,38,0451 [3DD4]
1540 DATA B9,3E,C3,32,3A,B9,FB,C9,04A3 [513A]
1550 DATA E5,C5,D5,F5,CD,95,70,F1,0637 [BE1C]
1560 DATA D1,C1,E1,08,DA,70,B9,C3,0541 [E9E8]
1570 DATA 3D,B9,00,00,00,00,00,00,00F6 [F52A]
1580 DATA 00,00,00,00,00,00,00,00,0000 [3390]

```

Listing 2. Der Basic-Lader für unser Maschinencode-Modul

HAPPY- COMPUTER SOFTWARE- SERVICE

Hallo, Happy-Computer-Fans!

Nur für Euch gibt's jetzt T-Shirts. Bestellen könnt Ihr gegen Voreinsendung des jeweiligen Betrags mit der Zahlkarte vom Programm-Service in diesem Heft. Bestellnummer bitte nicht vergessen - und falls vorhanden, Eure Kundennummer.

Größentabelle:	S	M	L	XL
Größe	4	5	6	7
Damen	38	40	42	44
Herrn	46	48	50	52
Kinder	176			

Alle Artikel sind vom Umtausch ausgeschlossen!

**Markt & Technik
BUCHVERLAG**

Hans-Pinsel-Straße 2, 8013 Haar bei München



1. T-Shirt »Happy«

3farbiger, kleiner Aufdruck,
100% Baumwolle, weiter Schnitt,
Jersey, Farbe: weiß

Größe	Best.-Nr.	
4 = S	TS 214S	DM 19,90
5 = M	TS 215M	DM 19,90
6 = L	TS 216L	DM 19,90
7 = XL	TS 217XL	DM 19,90

2. T-Shirt »Happy«

3farbiger, großer Aufdruck,
100% Baumwolle, weiter Schnitt,
Jersey, Farbe: weiß

Größe	Best.-Nr.	
4 = S	TS 204S	DM 19,90
5 = M	TS 205M	DM 19,90
6 = L	TS 206L	DM 19,90
7 = XL	TS 207XL	DM 19,90

```

10 MODE 2
20 DEFINT a-z
30 MEMORY &6FFF
40 LOAD "flirtm.bin"
50 CALL &7000
60 ON ERROR GOTO 690
70 DIM d1buf(1000):dlptr=1
80 ein=-1:aus=0:xon=17:xoff=19
90 drucker=aus:hdx=aus:dload=aus
100 GOSUB 640
110 GOSUB 300
120 WINDOW #1,2,79,2,7
130 WINDOW #0,1,80,9,25
140 WINDOW SWAP 0,1:PRINT "FLIRT - by T.
Obermair":WINDOW SWAP 0,1
150 GOSUB 730
160 a$=INKEY$:IF a$<>" " THEN GOSUB 190
170 a=0:RECV,@a:IF a<>0 THEN GOSUB 250
180 GOTO 160
190 REM ----- tastatur -> schnittstell
e
200 IF a$=CHR$(0) THEN GOSUB 370
210 IF a$=CHR$(12) THEN CLS:RETURN
220 !SEND,ASC(a$)
230 IF hdx=ein THEN PRINT a$;
240 RETURN
250 REM ----- schnittstelle -> bildsch
.
260 PRINT CHR$(a);
270 IF drucker=ein THEN !PRINT,a
280 IF dload=1 THEN d1buf(dlptr)=a:dlptr
=d1ptr+1:IF dlptr=900 THEN GOSUB 111
0
290 RETURN
300 REM ----- fenster zeichnen -----
310 PRINT CHR$(150);STRING$(78,CHR$(154)
);CHR$(156);
320 FOR i=1 TO 6
330 PRINT CHR$(149);STRING$(78," ");CHR$(
149);
340 NEXT i
350 PRINT CHR$(147);STRING$(78,CHR$(154)
);CHR$(153);
360 RETURN
370 REM ----- ESC - Funktionstastenver
waltung -----
380 KEY DEF 79,1,127,127,127
390 WINDOW SWAP 0,1
400 PRINT ">":CALL &BB81:LINE INPUT bef
ehl$
410 befehl$=UPPER$(befehl$)
420 IF befehl$="ESC" THEN !SEND,27:PRINT
"escape geschickt":GOTO 400
430 IF befehl$="BYE" THEN !BASIC
440 IF MID$(befehl$,1,3)="DIR" THEN !SEN
D,xoff:m$=MID$(befehl$,5):
!DIR,@m$:!SEND,xon:GOTO 4
00
450 IF befehl$="A:" THEN !SEND,xoff:A:
!SEND,xon:PRINT "drive A aktiv":
GOTO 400
460 IF befehl$="B:" THEN !SEND,xoff:B:
!SEND,xon:PRINT "drive B aktiv":
GOTO 400
470 IF MID$(befehl$,1,3)="ERA" THEN !SEN
D,xoff:m$=MID$(befehl$,5):
!ERA,@m$:!SEND,xon:GOTO 4
00
480 IF MID$(befehl$,1,4)="TYPE" THEN !SE
ND,xoff:OPENIN MID$(befehl$,6):
WHILE (NOT EOF) AND INKEY
$="" :LINE INPUT #9,zeile$:PRINT zeil
e$:
WEND:CLOSEIN:
!SEN
D,xon:GOTO 400
490 IF befehl$="FDX" THEN hdx=aus:PRINT
"fullduplex":GOTO 400
500 IF befehl$="HDX" THEN hdx=ein:PRINT
"halfduplex":GOTO 400
510 IF befehl$="PRN ON" THEN drucker=ein
:PRINT "drucker eingeschaltet":GOTO
400
520 IF befehl$="PRN OFF" THEN drucker=au
s:PRINT "drucker ausgeschaltet":
GOTO 400
530 IF befehl$="UL E" THEN GOSUB 780:GOT
0 400
540 IF befehl$="UL X" THEN GOSUB 910:GOT
0 400
550 IF befehl$="DL OPEN" THEN GOSUB 1060
:GOTO 400
560 IF befehl$="DL ON" THEN GOSUB 1150:G
OTO 400
570 IF befehl$="DL OFF" THEN GOSUB 1180:
GOTO 400
580 IF befehl$="DL CLOSE" THEN GOSUB 120
0:GOTO 400
590 IF befehl$<>" " THEN PRINT " Befehlsu
bersicht: DIR,HDX,FDX,A:,B:,ERA,BYE

```

[DFF0]
[52C8]
[2F54]
[8CFA]
[625C]
[93CA]
[4ADC]
[582E]
[FCBE]
[48D6]
[87CA]
[6274]
[4AD0]
[D6EC]
[82E0]
[30E4]
[EAA6]
[BE52]
[7CEE]
[329C]
[E288]
[3966]
[75FA]
[A72C]
[1106]
[71A6]
[29AC]
[CC84]
[9D36]
[EFFC]
[2276]
[3D58]
[B77E]
[EBFE]
[3884]
[CC32]
[AB5C]
[90A4]
[9838]
[D9C6]
[750E]
[A21A]
[0B3C]
[BC98]
[DC8C]
[F894]
[5282]
[3C5E]
[7352]
[13FC]
[656E]
[ABE0]
[179A]
[11B8]
[B4A6]
[8F7E]
[F702]
[732C]

```

,PRN ON,PRN OFF,TYPE":PRINT "{20 SPA
CE}UL E,UL X,DL OPEN,DL CLOSE,DL ON,
DL OFF{3 SPACE}":GOTO 400
600 CALL &BB84
610 WINDOW SWAP 0,1
620 KEY DEF 79,1,8,8,8
625 GOSUB 730
630 RETURN
640 REM ----- initialisieren -----
650 KEY DEF 66,1,0,0,0
660 KEY DEF 79,1,8,8,8
670 KEY DEF 16,1,12,12,12
680 RETURN
690 REM ----- errors abfangen -----
700 IF ERR=50 THEN CLOSEIN:CLOSEOUT:PRIN
T:PRINT "Diskettenfehler":
PRINT "alle Dateien gesch
lossen.":RESUME 400
710 PRINT "interner fehler";ERR;"in Zeil
e";ERL
720 GOTO 720
730 CALL &BB81:RETURN
740 REM ----- input-buffer loeschen ---
-----
750 FOR i=1 TO 10
760 WHILE a<>0:RECV,@a:WEND
770 NEXT i:RETURN
780 REM ----- upload - echo mode -----
-----
790 PRINT "Upload - Echo-Mode"
800 PRINT "Dateiname: ";:LINE INPUT dn$
810 GOSUB 740:REM buffer clear
820 OPENIN dn$
830 WHILE (NOT EOF) AND INKEY$=""
840 LINE INPUT #9,z$:z$=z$+CHR$(&D)+CHR$(
&A)
850 FOR i=1 TO LEN(z$)
860 !SEND,ASC(MID$(z$,i,1))
870 PRINT MID$(z$,i,1);
880 GOSUB 1040
890 NEXT i
900 WEND:CLOSEIN:RETURN
910 REM ----- upload - xon-xoff-mode -
-----
920 PRINT "Upload - xon/xoff mode"
930 PRINT "Dateiname: ";:LINE INPUT dn$
940 GOSUB 740:REM buffer clear
950 OPENIN dn$
960 WHILE (NOT EOF) AND INKEY$=""
970 LINE INPUT #9,z$:z$=z$+CHR$(&D)+CHR$(
&A)
980 FOR i=1 TO LEN(z$)
990 !RECV,@a:IF a=xoff THEN GOSUB 1040:R
EM wait bis xon
1000 !SEND,ASC(MID$(z$,i,1))
1010 PRINT MID$(z$,i,1);
1020 NEXT i
1030 WEND:CLOSEIN:RETURN
1040 REM ----- wait bis irgendeinzeichen
oder xon
1050 a=0:WHILE a=0:RECV,@a:WEND:RETURN
1060 REM ---- dl-open ----
1070 PRINT "download - open"
1080 PRINT "filename: ";:LINE INPUT dlfn$
1090 IF dlfn$="" THEN RETURN
1100 !SEND,xoff:OPENOUT dlfn$:!SEND,xon:
dlo=1:RETURN
1110 REM ---- abspeichern
1120 !SEND,xoff
1130 FOR i=1 TO 899:PRINT #9,CHR$(d1buf(
i));:NEXT i
1140 dlptr=1:!SEND,xon:RETURN
1150 REM ----- dl ein -----
1160 IF dlo=0 THEN PRINT "erst oeffnen":
RETURN
1170 PRINT "dl ein":dload=1:RETURN
1180 REM ----- dl aus -----
1190 PRINT "dl ausgeschaltet":dload=0:RE
TURN
1200 REM ----- dload close -----
-----
1210 !SEND,xoff:FOR i=1 TO dlptr-1:PRINT
#9,CHR$(d1buf(i));:NEXT i
1215 PRINT #9,CHR$(26);:REM EOF MARKER
1220 CLOSEOUT:dlptr=1:dlo=0:dload=aus:
!S
END,xon
1230 PRINT "ok, geschlossen.":RETURN

```

Listing 3.
Das DFÜ-Programm
ist überwiegend in Basic geschrieben

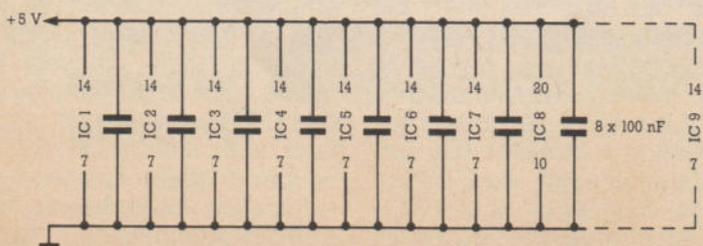
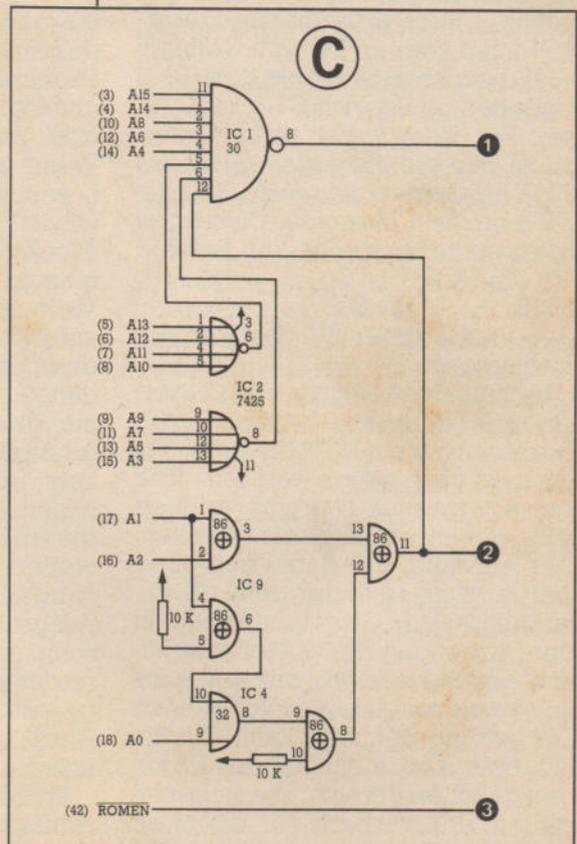
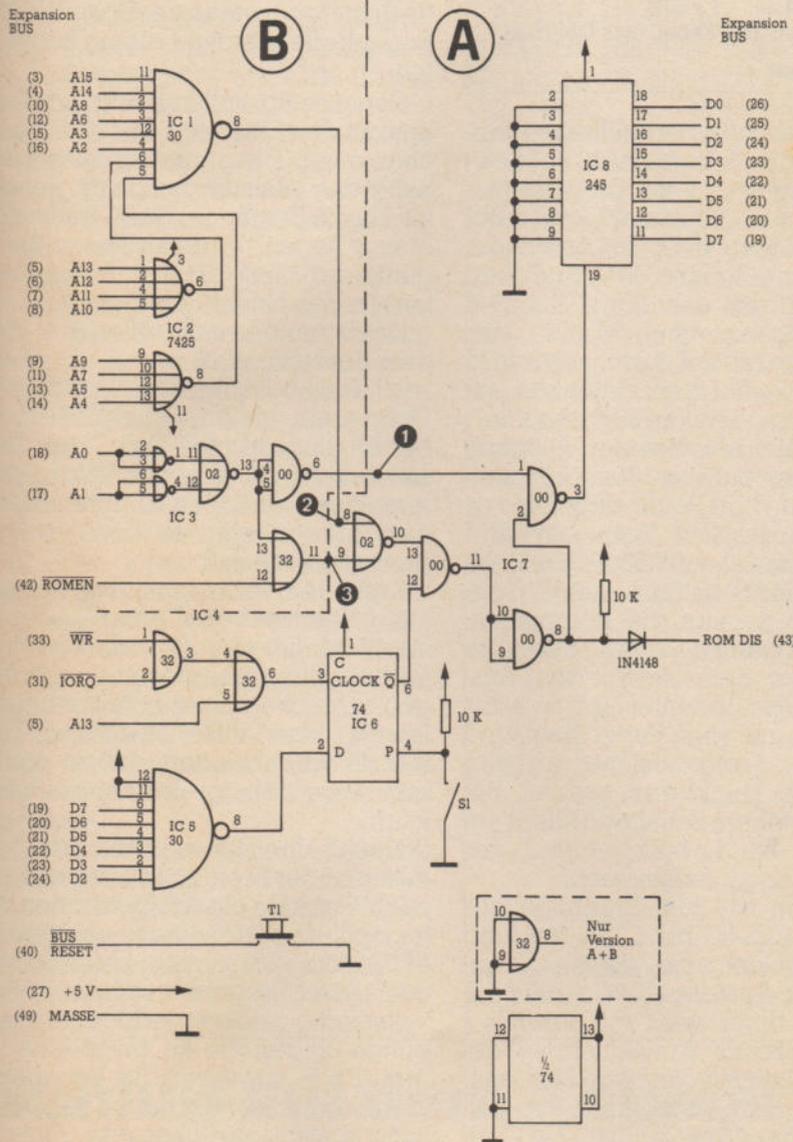
Reset ohne Datenverlust



Nach einem Zurücksetzen des Computers sind normalerweise alle Daten verloren. Besonders bei Maschinencode-Routinen kein wünschenswerter Zustand. Aber das muß nicht sein.

Der Schneider ist serienmäßig mit einem Reset-Schalter ausgestattet. Dieser arbeitet allerdings nur softwaremäßig. Wenn aber von einem Assembler-Programm der Interrupt unterbunden wird, so besteht keine Möglichkeit mehr durch Benutzung der Tasten Shift, Ctrl und ESC das laufende Programm zu unterbrechen. Hier hilft nur noch ein Aus- und wieder Einschalten des Computers, was aber mit dem Verlust aller Daten teuer bezahlt werden muß.

Ein weiteres Manko im Betriebssystem besteht darin, daß der Speicherbereich von 40 bis ABFF hex beim Kaltstart (Aufruf der Adresse



0000 hex oder Reset mit Shift, Ctrl und Esc) mit lauter Nullen beschrieben wird. Daß dieser Zustand äußerst lästig ist, werden einige »Maschinencode-Programmierer« sicherlich bestätigen können. Ein noch fehlerhaftes Assembler-Programm bleibt mit Vorliebe in einer Schleife hängen, aus der es nicht mehr zurückgeholt werden kann. Der nicht zu umgehende Reset führt zu einem vollständig leeren Speicher. Ebenso wird beim Löschen eines Basic-Programms mit dem Befehl »NEW« der Speicherbereich bis HIMEN mit lauter Nullen überschrieben. Es ist nichts mehr zu retten. Oder?

Jedem Problem kann man natürlich zu Leibe rücken. In diesem Fall

Bild 1. Der Schaltplan für zwei Platinen. Der Bereich A muß für den 664 und 6128 durch C vertauscht werden.

gibt es zwei total verschiedene Wege. Entweder man ändert die Software, das heißt man muß den Inhalt des ROMs in einer neuen Version in ein Eprom brennen oder man schafft hardwaremäßig Abhilfe.

Die zweite Lösung — die Hardware abzuändern — ist entschieden die einfachere. Sie bietet unter anderem nämlich die Möglichkeit, zwischen Originalsoftware und abgeändertem Programm hin- und herzuschalten. Außerdem kann Sie jeder benutzen, man braucht keinen Eprom-Brenner.

Um diese Änderung nun anzugehen, muß man erst einmal wissen, wo sich die Routine, welche für den Löschvorgang zuständig ist, im ROM befindet. Beim 464 steht sie im Basic-ROM und zwar im Bereich C13E bis C14D hex; beim 664 sowie beim 6128 hingegen im Bereich C145 bis C154 hex. Für unser Vorhaben sind allerdings nur die Adressen C14C bis C14E hex beziehungsweise C153 bis C155 hex von Interesse. Diese drei Bytes enthalten die beiden Befehle

LD (HL),A 77
LDIR ED B0

— wobei letzterer für das Löschen zuständig ist.

Um nun den Löschvorgang zu verhindern, muß an die genannten Adressen nur ein NOP, das heißt der Wert 00 hex gelegt werden. Wie sich so etwas hardwaremäßig lösen läßt, wollen wir uns hier ansehen. Der Aufwand, der dabei getrieben werden muß, ist leider ziemlich umfangreich. Der Grund dafür liegt in der Verwaltung der Speicherbereiche des Schneiders, der ja zusätzlich zu seinem internen ROM noch 252 externe benutzen kann. Zusätzlich belegt noch ein Teil des RAMs den gleichen Bereich. Um sicherzustellen, daß wirklich nur die oben genannten Adressen im Basic-ROM ausgeklammert werden, wird an den I/O-Port »DF« ein Wert größer als 251 gesendet. Die Werte von 0 bis einschließlich 251 stehen nämlich für die externen ROMs zur Verfügung.

Um diesen Wert zu erkennen und zwischenspeichern, dazu dienen in der Schaltung (Bild 1) die ICs 4, 5 und 6. Die zwei Oder-Gatter des IC 4 registrieren, wenn ein Schreibzyklus auf den Port »DF« gegeben wird; dabei gehen die Leitung »WR« sowie »IORQ« auf Low-Signal, das heißt auf 0 Volt. Damit wird ein Beschreiben eines Ports angekündigt. Um welchen Port es sich dabei handelt, wird durch die Adreßbits A8 bis A15 bestimmt. Wenn man sich den Wert DF00 binär (1101 1111 0000

IC 1, IC 5	SN 74 LS 30
IC 2	SN 7425
IC 3	SN 74 LS 02
IC 4	SN 74 LS 32
IC 6	SN 74 LS 74
IC 7	SN 74 LS 00
IC 8	SN 74 LS 245
IC	SN 74 LS 86
	(nur bei 664/6128)
Widerstände	2 x 10 k Ω $\frac{1}{4}$ W (2 x 10 k Ω $\frac{1}{4}$ W nur bei 664/6128)
Kondensatoren	8 x 100 nF Keramik
Diode	1 N 4148
Taster T1	1 x EIN
Schalter S1	1 x EIN
Lochrasterplatte, Kabel, evtl. Gehäuse	
Die Bauteile	

0000) anschaut, so ist leicht zu sehen, daß sich das Adreßbit A13 auf Low-Pegel befindet. Diese Tatsache machen wir uns zunutze, um einen Zugriff auf Port »DF« zu erkennen. Wenn nun alle drei Leitungen auf Low sind, liegt das Signal am IC 6 Clock-Eingang an. Wenn der Clockeingang auf Low geht, wird gleichzeitig auf den Datenbus ein Wert gelegt, der vom IC 5 decodiert wird. Dabei schaltet der Ausgang des Gatters auf Low, wenn alle Eingänge auf High-Pegel sind. Dies ist nur dann der Fall, wenn ein Wert größer oder gleich 252 (FC hex) anliegt. Ist der Schalter S1 am IC 6 geöffnet, so gelangt die Information, die am Eingang »D« des IC 6 liegt, invertiert zu dem invertierten Ausgang Q. Dies bedeutet hier, daß der Ausgang auf High-Pegel liegt und somit ein Freigabesignal an den restlichen Logikkreis sendet. Bei geschlossenem Schalter S1 liegt am Ausgang ein Low-Signal und der Rest der Logik ist gesperrt.

Kommen wir zum Erkennen der Adressen C14C bis C14E hex und zum Ausklammern dieses Bereiches. Die Adresse C14C (1100 0001 0100 1100 binär) wird von den ICs 1 und 2 (Abschnitt B im Schaltbild) decodiert. Der Ausgang von IC 1 geht nur dann auf Low, wenn an den beiden ICs die Adresse C14C bis C14F hex anliegt.

Daß nur die Adressen C14C bis C14E hex Gültigkeit haben, wird von den Adreßbits A0 bis A1 hex sowie den Gattern des IC 3 bestimmt. Den Zugriff auf ein ROM verrät uns die ROMEN-Leitung. Schaltet die Leitung auf Low, so greift die CPU auf ein ROM zu. Ebenso ist es beim 664 und 6128, sieht man von den veränderten Adressen ab. Auch hier decodieren die ICs 1 und 2 (Abschnitt C im Schaltbild) die Adressen im Bereich C150 (1100 0001 0101 0000 bin)

bis C157 hex (1100 0001 0101 0111 bin). Allerdings kommt noch ein IC hinzu. Dieses IC (IC 9) decodiert die Adressen A0 bis A2 hex, um sicherzustellen, daß wirklich nur die Adressen C153 bis C155 hex akzeptiert werden.

Sobald alle Signale in der benötigten Form vorliegen, haben wir am Ausgang 8 des Gatters von IC 7 einen High-Pegel. Dieser liegt über eine Diode an dem ROMDIS-Eingang des Erweiterungsbusses des Computers an und schaltet somit das interne ROM ab. Die Diode ist erforderlich, weil mehrere Erweiterungen auf die gleiche Leitung zugreifen.

Gleichzeitig gelangt ein Low-Signal an den Eingang 1 des IC 8, welches einen Datentransfer vom Eingang des ICs zum Ausgang gestattet. Da die Eingänge des IC 8 alle auf Masse liegen, wird zu diesem Zeitpunkt der Wert 00 hex auf den Datenbus des Computers gelegt.

Das Schaltbild unterteilt sich in die drei Bereiche A, B und C. Der Bereich A ist bei allen drei Schneider-Computern gleich. Schaltbildteil B betrifft dann den 464, Teil C den 664 sowie den 6128. Wie die Abschnitte jeweils miteinander verbunden werden, erkennt man an den Verbindungspunkten 1, 2 und 3.

Aufbauen kann man das Ganze am einfachsten auf einer Lochrasterplatte. Ob es außerhalb oder innerhalb des Gehäuses unterzubringen ist, sei dem Bastler selbst überlassen. Man sollte aber den Taster T1 so anbringen, daß er nicht aus Versehen gedrückt werden kann.

Noch eine kleine Anmerkung: Wer jetzt der Meinung ist, aus einem »NEW« wieder ein »OLD« machen zu können, der wird leider ein wenig enttäuscht. Zwar wird beim Reset das Basic-Programm nicht mehr vollständig gelöscht, jedoch werden einige Zeiger, die für Basic-Anfang und -Ende sowie für die Variablen zuständig sind, wieder in den Einschaltzustand zurückgesetzt. Außerdem werden am Anfang des Basics einige Speicherzellen mit Nullen sowie mit den Zeigern der Variablen überschrieben. Dieser Umstand macht das »rückgängig machen« des »NEW«s zwar nicht gerade einfach, aber immerhin doch möglich. So bringt diese Hardware-Bastelei — vor allem für leidgeplagte Assembler-Programmierer — einige Vorteile mit sich. Wer das in der Praxis erfahren hat, wird sie für ihre nur zirka 25 Mark auch nicht mehr missen wollen. (Michael Bauer)

Hifi am Schneider



Für die Schneider CPC 664 und 6128 kann man einen normalen Kassetten-Recorder als Backup-Medium benutzen. Allerdings ist der Ausgang nicht der DIN-Norm entsprechend belegt.

Mancher Käufer eines CPC 664 oder 6128 stellte recht schnell fest, daß die anfängliche Euphorie über die gelungene Kombination von Heimcomputer und Diskettenlaufwerk nicht immer berechtigt war. 3-Zoll-Disketten sind teuer und so absolut sicher sind die Daten auf ihnen auch nicht gespeichert. Also braucht man von jeder Diskette eine Sicherheitskopie — und das wird teuer.

Kassetten hingegen sind zwar ein langsames Speichermedium, aber billig. Also muß der (meist vorhandene) Kassetten-Recorder aus dem Hi-Fi-Turm als Backup-Speicher her. Die Buchse des Computers sieht wie ein normaler Stereoanschluß nach DIN-Norm aus — da dürfte es eigentlich keine Schwierigkeiten geben. Doch weit gefehlt. Auf den fünf Polen einer Stereo-Verbindung liegen normalerweise zwei Aufnahme- und zwei Wiedergabekanäle, sowie Masse. Der Schneider-Computer arbeitet hingegen mit nur einem Auf- und einem Wiedergabekanal. Neben der Masse gibt es dann noch zwei Anschlüsse für die Motorsteuerung des Kassetten-Recorders.

Diese verschiedenen Kontakte überlagern sich, so daß man das Übertragungskabel selbst bauen muß. Am einfachsten (und billigsten) geht es mit einer normalen, fünfpoligen Stereo-Verbindung. Die Stecker

dürfen nicht verschweißt sein, da man an einem die Anschlußbelegung ändern muß.

Als erstes (nach dem Öffnen eines Steckers) schreibt man sich die Nummern der Stifte und die zugehörige Farbe des Kabels auf. Die Ziffern befinden sich meist auf der Innenseite des Steckers. Dann wird das Kabel von Pin 1 auf Pin 5 und das Kabel von Pin 3 auf Pin 4 umgelötet. Die ursprünglichen Kabel an Pin 4 und 5 bleiben getrennt. Gegebenenfalls kann hier die Steuerung des Kassetten-Recorders angeschlossen werden. Der umgebaute Stecker gehört dann direkt an den Computer (eine Markierung ist hier sinnvoll), während die unveränderte Seite an den Kassetten-Recorder gesteckt wird.

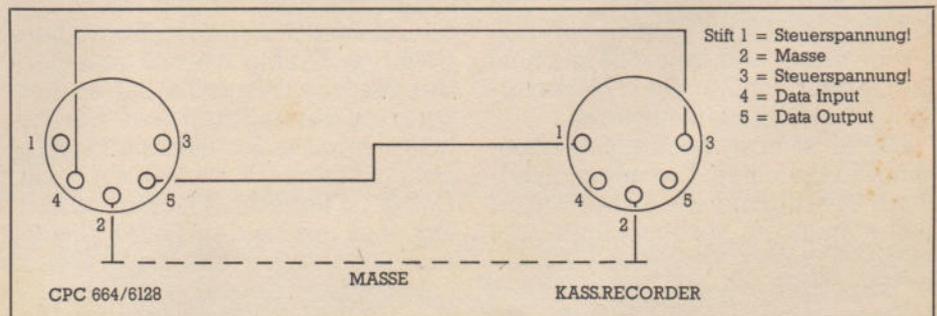
Wer einen Recorder mit Cinch-Steckern hat, der muß sein Kabel so anschließen, daß der Eingang der

Daten an Pin 4 und der Ausgang an Pin 5 (Computerseite) erfolgt. Die Belegung am Kassetten-Recorder entnimmt man am besten der Betriebsanleitung. So lange man nur den Datenein- und -ausgang vertauscht, kann man auch nichts zerstören. Schlimmer ist eine Verwechslung mit Pin 1 beziehungsweise 3. Zur Motorsteuerung liegt nämlich eine Spannung an, die die Datenleitungen nicht verkraften.

Wer sich einen speziellen »Datenrecorder« zulegen will, dem sei noch gesagt, daß ein höherer Preis nicht unbedingt ein besseres Funktionieren garantiert. Eher umgekehrt. Mono ist besser als Stereo und damit reicht ein Gerät der 70-Mark-Klasse vollkommen aus. Der Walkman kann natürlich auch als Datenspeicher mißbraucht werden. Allerdings ist bei den kleinen Steckern das Löten schon fast ein Gesellenstück für Elektroniker — wenn es ordentlich gemacht sein soll.

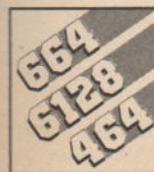
Bei wem sich nach getaner Arbeit kein Erfolg einstellt (obwohl die Töne übertragen werden), der sollte den Artikel »Wenn der Recorder streikt« in diesem Heft lesen. Dort findet man Hilfe, wenn die Daten nicht so laufen wollen, wie sie sollen.

(Helmut Jungkunz)



Was nach DIN aussieht, muß nicht DIN sein: Der Schaltplan für ein Verbindungskabel von Ihrem Schneider zum Kassetten-Recorder

Schluß mit dem Kabelärger



Haben Sie sich auch schon mal über die kurzen Verbindungskabel zwischen Computer und Monitor geärgert?

Bösartige Zeitgenossen behaupten, daß ein Kabel nur zwei Zustände kennt: entweder zu kurz oder zu lang. Verlängert man es, so ist das zusammengesetzte Kabel nach dieser Theorie allerdings immer noch zu kurz, oder — aufgrund

überlanger Kabelschleifen — zu lang. Das Dilemma wird noch schlimmer, wenn man abseits großer Städte versucht, passende Kabel zu kaufen.

Alles, was wir aber brauchen, ist ein LötKolben, etwas Lötzinn und zir-

ka zwei bis drei Meter eines Stereo-Verbindungskabels (5polig plus Abschirmung). Für die genaue Länge sollten Sie an die oben angeführte Theorie denken. Ferner brauchen Sie vier (beziehungsweise beim CPC 664 und 6128) sechs Stecker.

Bei den Steckern handelt es sich um etwas ungewöhnliche, das heißt nicht so häufig angebotene, Steckverbindungen, die jedoch beide der DIN-Norm entsprechen. Ein bißchen Suchen und Sie müßten die benötigten Verbindungen bei Ihrem Radio- (oder Elektronik-) Händler finden. Sollten alle Stricke reißen, so hilft ein Blick in Elektronikzeitschrif-

ten, um Entsprechendes in deren Anzeigenteilen zu finden.

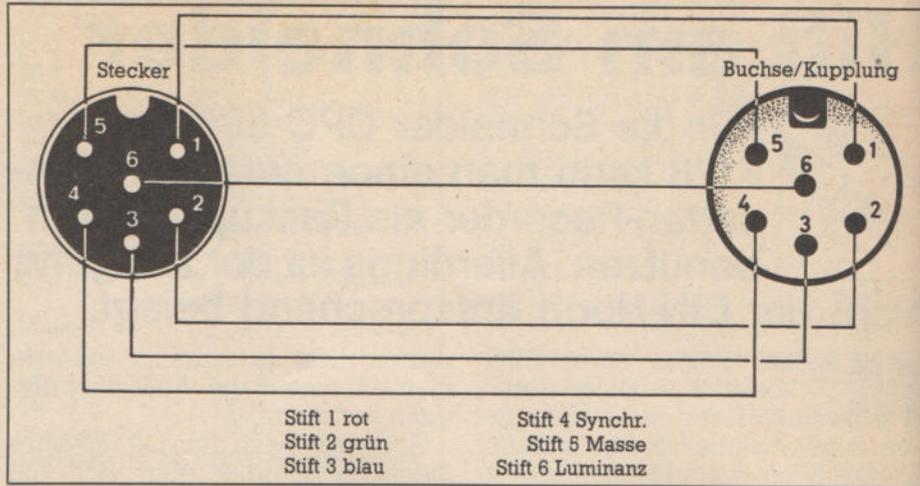
Für unsere Steckverbindung benötigen Sie jeweils einen männlichen Stecker und einen zugehörigen weiblichen Stecker pro anzufertigendes Kabel — insgesamt also zwei oder drei Steckerpaare. Die männlichen Stecker sehen dabei genauso aus, wie die Exemplare, die bereits an den Kabelenden angebracht sind. Die entsprechenden Gegenstücke sind dann die weiblichen Stecker.

Wenn Sie beide Kabel — beziehungsweise im Fall des 664 und des 6128 die drei — zwischen Monitor und Tastaturboard herausziehen (bei ausgeschaltetem Computer natürlich), so sind Sie mit zwei unterschiedlichen Steckertypen konfrontiert. Nehmen wir uns zunächst einmal den Monitorstecker vor. Dies ist ein 6poliger DIN-Stecker. In einem silbrigem Zylinder, der die Abschirmung gewährleistet, sehen Sie die sechs einzelnen Stifte, die die Signale für die Aufstufung der einzelnen Farben des Monitors, für die Farbinintensität (Luminanz) sowie das Synchronisationssignal an den Monitor weiterleiten.

Daneben haben wir es noch mit einem (beim CPC 664 und 6128 zwei) weiteren Stecker(n) zu tun, die die Stromversorgung des Computers gewährleisten. Vom Monitor zum Computergehäuse führt die 5-V-Stromversorgung, beim 664 und 6128 in Gegenrichtung die 12-V-Versorgung des Diskettenlaufwerks.

Dabei ist aber noch eines unbedingt zu beachten. Die unterschiedliche Befestigung der Kabel wurde unter anderem vorgenommen, um sicherzustellen, daß der 12-V-Anschluß nicht aus Versehen mit der 5-V-Eingangsbuchse verbunden wird. Dies hätte nämlich die Zerstörung Ihres Computers zur Folge. Sie sollten daher auch, nachdem die Kabel zusammengebaut sind, eine Markierung anbringen. Entweder kennzeichnen Sie die beiden Verbindungskabel mit verschiedenfarbigen Klebebändern oder Sie kleben die Kupplungsstellen mit Tesafilm fest aneinander. Und zwar so, daß immer eine Buchse mit einem Stecker an einem Ende zusammen liegt.

Kommen wir zur eigentlichen Verdrahtung. Für die (beiden) spannungsführende(n) Leitung(en), gilt derselbe »Schaltplan«. Wenn Sie den Stecker, der vom Monitor zum Computer führt, aus der mit »5V DC« bezeichneten Buchse ziehen, so blicken Sie außen auf eine silbrige



Der Schaltplan für das Kabel für die Monitorsignale

Hülse. In der Mitte dieses Zylinders — durch eine Isolierschicht getrennt — liegt ein zweiter kleiner Blechzylinder. An diesem liegt die Spannung (5 oder 12 V) an. Die Hülse ist mit Masse belegt. Die gekaufte Buchse und der Stecker müssen nun mit zwei Kabeln innen und außen verbunden werden.

Beschaffungsprobleme und Kosten (ein langes Kabel ist billiger als zwei kurze) lassen uns für alle Verbindungen das 5polige Kabel verwenden.

Nach dem Abisolieren unseres Stereokabels fassen wir drei Leitungen zusammen. Den zweiten Kontakt stellen die zwei übrigen Adern und die Abschirmung her. Zusammengehörige Drähte werden miteinander verdrillt. Wenn Sie diesen Vorgang auf der anderen Seite wiederholen, erhalten Sie ein 2-poliges Kabel mit einem höheren Querschnitt. Dabei ist unbedingt darauf zu achten, daß auf beiden Seiten dieselben Leitungen (dieselben Farben) miteinander verdrillt werden. Ansonsten bekommen Sie beim Einschalten einen Kurzschluß, der Ihren Computer ruinieren kann.

Wenn Sie einen Durchgangsprüfer oder ein Ohmmeter zur Hand haben, so sollten Sie daher prüfen, ob die beiden Leitungen wirklich gegeneinander isoliert sind. Im Zentrum der beiden Stecker finden Sie nun einen kleinen Lötspunkt, der mit dem Innenteil unseres Steckers in Verbindung steht (beziehungsweise bei der Buchse auf dem Stift geschaltet ist). An diesen Pol löten Sie die drei gerade zusammengeflochtenen Adern an. Der andere Kontakt wird mit den zwei Adern und der mit ihnen verdrillten Abschirmung belegt. Wenn Sie dies auf beiden Seiten gemacht haben (vor dem Anlöten sollten allerdings die Gehäusenhälften der Stecker über das Kabel

geschoben werden), so sind die Netzteilverlängerungen fertig.

Nun brauchen wir noch die Monitorsteckverbindung. Dazu benutzen wir unser sechsadriges Kabel (fünf Adern und Abschirmung) unverdrillt. Wir müssen also sowohl die fünf Adern jeweils einzeln mit den Lötspunkten verbinden, wie auch noch die Abschirmung als sechste Leitung. Diese sollte dabei auf den Zentralstift beziehungsweise bei der Buchse auf das zugehörige Loch gelötet werden.

Falls Sie über ein Spannungsmeßgerät verfügen, so sollten Sie, bevor Sie die Kabelverbindung einsetzen, diese durchprüfen. Nach dem Einschalten müssen Sie an einem Stecker 5 V messen, wobei Minus außen und Plus im Innern des Steckverbinders liegt. Beim 664 (6128) muß die Buchse der zweiten Verbindung (ebenfalls Hülse gegen Kern) 12 V aufweisen. Messen Sie Plus und Minus vertauscht, so haben Sie eine falsche Beschaltung vorgenommen. Messen Sie gar nichts, so haben Sie entweder einen Kurzschluß oder eine kalte Lötstelle (keine Verbindung).

Wichtig!

Eine Fehlpolung der Spannungsstecker, das heißt Plus und Minus vertauscht, kann zu einer Zerstörung des Computers führen. Diese ist selbstverständlich nicht durch die Garantie gedeckt. Im Zweifelsfall sollten Sie daher eher einen befreundeten Hobbyelektroniker zu Rate ziehen. (Carsten Strauß)

- 2 bis 3 m Stereokabel (5polig abgeschirmt)
- 1 (2) Steckerpaar(e) Niederspannungssteckverbinder nach DIN
- 1 6poliger DIN-Stecker mit Kupplung
- () für CPC 664, 6128

Die Liste der Bauteile

Bestellungen aus der Schweiz richten Sie bitte direkt an: Markt & Technik Vertriebs AG, Kollerstr. 3, CH-6300 Zug, Tel. 042/41 56 56.

Bestellungen aus anderen Ländern bitte per Auslandspostanweisung! Achtung: Nicht die eingehaftete Zahlkarte verwenden! Bestellungen aus Österreich richten Sie bitte direkt an: Bücherzentrum Meidling, Schönbrunner Str. 261, A-1120 Wien, Tel. 0222/833196, microcomput-ique, E. Schiller, Fasangasse 21, A-1030 Wien, Tel. 0222/785661.

HAPPY COMPUTER

Achtung!

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

Fehlende Hefte erhalten Sie bei Markt & Technik Vertrieb Happy Computer Hans-Pinsel-Straße 2 8013 Haar bei München

PROGRAMM-SERVICE

Programme aus früheren Ausgaben

Commodore 64 / Commodore 128

Taxi. Aus Ausgabe 1/86.
Musik und Farbe. Aus Ausgabe 12/85.
SDB-Sprite Mover. Aus Ausgabe 1/86.
ES-AE. Aus Ausgabe 1/86.
Ultraload. Aus Ausgabe 1/86.
Error 64. Aus Ausgabe 1/86.
Scroll 64. Aus Ausgabe 1/86.
Schatzsuche. Aus Ausgabe 12/85.
SLAD. Aus Ausgabe 12/84.
Alle 9 Programme auf Diskette für den Commodore 64/128
Bestell-Nr. LH 8601 CD,
DM 29,90*, sFr. 24,90*

Atari 800XL/130XE/800

Turbo-Basic. Aus Ausgabe 12/85.
AMPEL. Atari-Maschinen-Programm-Eingabe-Listing. Aus Ausgabe 12/85.
Atari-Prüfsummer. Jumper II. Magic-Painter. Aus Ausgabe 3/85.
Alle 5 Programme auf einer Diskette für den Atari 800XL/130XE/800.
Bestell-Nr. LH 8512B,
DM 29,90*, sFr. 24,90*

Schneider CPC

Programmtransfer leicht gemacht (zwei Programme, S. 72)
Tasword 464* mit DIN-Tastatur (S. 73)
Bewegte Grafik mit drei Befehlen (S. 74)
Maschinencode-Routinen in BASIC umgesetzt (S. 75). Aus Ausgabe 10/85.
Sam. Aus Ausgabe 11/85.
Deutscher Zeichensatz unter CP/M. Aus Ausgabe 12/85.
Hardcopy. Aus Ausgabe 12/85.
RSX-Befehle mit direkter Stringvariable. Aus Ausgabe 12/85.
Alle 8 Programme auf einer Kassette für den Schneider CPC.
Bestell-Nr. LH 8512 G,
DM 29,90*, sFr. 24,90*

Commodore 64

Ausgabe 10/85
Aquantor, Zykloide, Nebenkostenabrechnung, Neuer Checksummer, Plakat, Datenzeilen-Wandler, Super-Saver, More Memory, Antolitschutz, Grafik-Window-Zeichner.
Ausgabe 11/85
Flugplanung, Finanzen 64, User-Port-Anzeige, Amadeus, ZX81-Utility, Long-Screen, Chess-Screen, Colour-Screen, Auto-beep.
Alle 19 Programme auf einer doppelseitig bespielten Diskette für den Commodore 64.
Bestell-Nr. LH 8511 A, DM 29,90*, sFr. 24,90*

Spectrum

Das andere Grafikprogramm. Aus Ausgabe 7/85.
Mini-Textverarbeitung. Aus Ausgabe 8/85.
Terminal-Programm. Listing des Monats aus Ausgabe 9/85.
Alle 3 Programme auf einer Kassette für den Sinclair Spectrum.
Bestell-Nr. LH 8510 D, DM 19,90*, sFr. 16,90*

Atari

Prüfsummer
Eingabehilfe für alle in Happy-Computer veröffentlichten Basic-Programme.
Geröllheimer
Mit Screen-Editor und 20 fertigen Szenen (Spiel), aus Ausgabe 5/85.
24 Farben in Grafikstufe 0
Routine für farbige Schrift (Utility), aus Ausgabe 6/85.
Diskhelp
Für die schnelle Rettung (Utility), aus Ausgabe 8/85.
Ölsuche
Mit dem Atari auf Ölsuche (Spiel), aus Ausgabe 8/85.
Autostart
Basic-Programme automatisch starten (Utility), aus Ausgabe 9/85.

Dudu 4.0

Mehr Speicher mit der 1050 Floppy (Utility), aus Ausgabe 10/85.
Alle 7 Programme auf einer Diskette für den Atari 800 XL.
Bestell-Nr. LH 8510 B, DM 29,90*, sFr. 24,90*

Sonderheft: Spectrum

Kassette / Bestell-Nr.
LH 85S1 D, DM 19,90*, sFr. 16,90*

Sonderheft: Schneider

3" Diskette, Bestell-Nr.: LH 85S2 D, DM 34,90*
5 1/4" Diskette, Best-Nr.: LH 85S2 V, DM 34,90*
Kassette, Bestell-Nr.: LH 85S2 K, DM 29,90*

Commodore 64

Alle 12 Programme auf Diskette für den Commodore 64.
Bestell-Nr. LH 8509 A, DM 29,90*, sFr. 24,90*
Schnelle Grafik aus dem Compiler
Listing des Monats aus der Ausgabe 8/85.
Mondlandung. Aus Ausgabe 8/85.
Komfort-Routinen. Aus Ausgabe 8/85.
Grafik-Hardcopy. Aus Ausgabe 8/84.
Psycho. Aus Ausgabe 8/85.
Tab-Calc. Aus Ausgabe 8/85. Aus Ausgabe 9/85.
Woodshot, Sprite Mover, Short Save, Sprite-Dreher, Echtzeituhr, Animation

Commodore 64

Risiko. Aus Ausgabe 7/85.
Mini-Grafik. Aus Ausgabe 7/85.
Muso. Aus Ausgabe 7/85.
Maskenbildner. Aus Ausgabe 7/85.
Aller Anfang ist schwer. Aus Ausgabe 7/85.
Alle 5 Programme auf Diskette für den Commodore 64.
Bestell-Nr. LH 8507 A, DM 29,90*, sFr. 24,90*

Schneider CPC 464

Diassembler. Aus Ausgabe 4/85.
Grafik. Aus Ausgabe 4/85.
Dateiverwaltung. Aus Ausgabe 4/85.
Alle 3 Programme auf Kassette für den Schneider CPC 464.
Bestell-Nr. LH 8505 G, DM 29,90*, sFr. 24,90*
Text. Aus Ausgabe 3/85.
Gespensterjagd. Aus Ausgabe 2/85.
Alle 2 Programme auf Kassette für den Schneider CPC 464.
Bestell-Nr. LH 8503 G, DM 29,90*, sFr. 24,90*
BW-COM-Compiler. Aus Ausgabe 7/85.
CHAIN MERGE. Aus Ausgabe 6/85.
Protokollfunktion. Aus Ausgabe 8/85.
Schneider-Kurs
Alle 4 Programme auf Kassette für den Schneider CPC 464.
Bestell-Nr. LH 8508 G, DM 29,90*, sFr. 24,90*

Bitte beachten Sie, daß der Kassette/Diskette keinerlei Informationen beiliegen. Lesen Sie daher aufmerksam die Anleitung in dem jeweiligen Artikel nach. Eventuelle systematische Fehler, die sich in den Programmen noch befinden können, müssen von Ihnen selbst, nach Studium der Nachhallseite, korrigiert werden.

* Alle Preise inklusive Mehrwertsteuer, unverbindliche Preisempfehlung. Programm-Service-Produkte sind nur für Endkunden, nicht für Wiederverkäufer.

Paint Magic

Das magische Zeichenprogramm aus den USA für Ihren Commodore 64

- elf gespeicherte »Traumbilder«
- gleichzeitiges Malen auf zwei Bildschirmen
- einfache Bedienung durch übersichtliche Menütechnik
- eigenes Farbmenü (16 Farben)
- umfangreiche Diskettenbefehle (Speichern, Löschen, Laden)
- 100% Maschinensprache

Markt & Technik-Programme erhalten Sie bei Ihrem Buchhändler.

Bestellkarten bitte an Ihren Buchhändler oder an eine unserer Depotbuchhandlungen. Adressverzeichnis am Ende des Heftes. Beim Markt & Technik Verlag eingehende Bestellungen werden von den Depot-Händlern ausgeliefert.

Markt & Technik
BUCHVERLAG

Hans-Pinsel-Straße 2, 8013 Haar bei München
Schweiz: Markt & Technik Vertriebs AG, Kollerstrasse 3, CH-6300 Zug, ☎ 042/41 56 56
Österreich: Rudolf Lechner & Sohn, Heizwerkstraße 10, A-1232 Wien, ☎ 02 22/67 75 26



- ★ Deutsches Auswahlménú auf Diskette
- ★ Deutsches Anleitungshéft

DM 59,-

Inkl. MwSt.
unverbindliche
Preisempfehlung
(Str. 54, 50/65 460,20)
Bestell-Nr. MD 230 A

Werden Sie mit den »magischen Malereien« zum »elektronischen Künstler!«
Sie brauchen Ihren Commodore 64 — ein Diskettenlaufwerk — Joystick.

Schneider verspielt



Vor lauter CP/M und RSX sollten Sie die unterhaltsamen Seiten Ihres Schneider-Computers nicht vergessen. Für den CPC gibt es ein sehenswertes Angebot von Computerspielen, aus dem wir Ihnen einige Leckerbissen vorstellen.

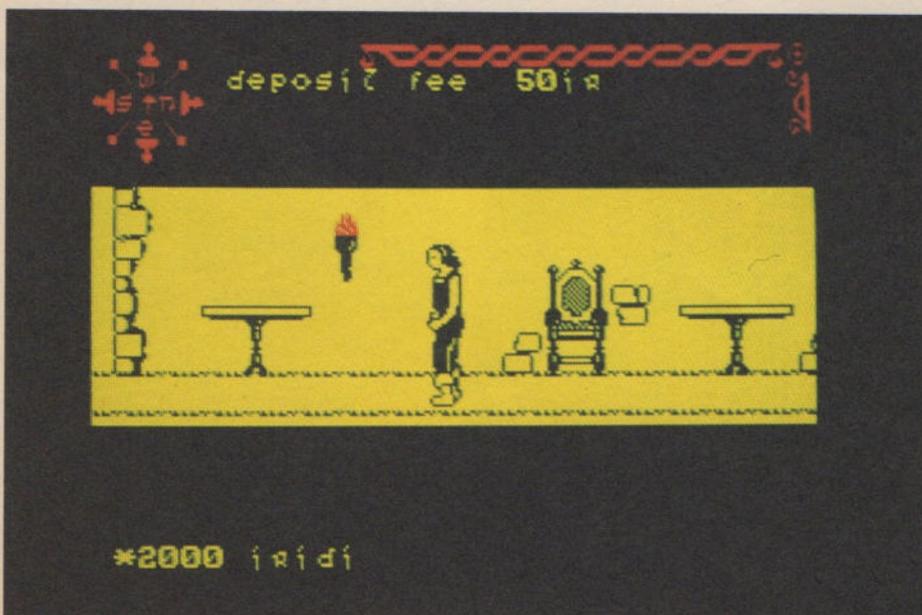
Schneiders CPC-Computer führen ein schillerndes Doppelleben: Auf der einen Seite sind sie hochkarätige Heimcomputer, die sich mühelos für professionelle Einsätze hochpäppeln lassen. In diesem Artikel wollen wir uns der anderen, schillernden Seite widmen: der spielerischen. Vor allem aus England gibt es eine anhaltende Welle von Computerspielen für den CPC. Das liegt vor allem daran, daß der mit dem Schneider identische Amstrad-Computer in Großbritannien ähnlich weit verbreitet ist wie bei uns, und in England bekanntlich viele große Softwarefirmen zu Hause sind.

Mit seiner hochauflösenden Grafik, der breiten Farben-Palette und dem schon in der Grundausstattung stattlichen Arbeitsspeicher von 64 KByte bietet sich der CPC auch förmlich als Spielcomputer an. Als reine Spielmaschine ist er allerdings nicht zu empfehlen, da hier der berühmt-berüchtigte Commodore 64 dank Sprites und leistungsstarken Audio- und Video-Chips bei den 8-Bit-Computern immer noch

die Nase vorn hat. Aber wenn man die Kiste schon mal im trauten Heim hat, um damit zu programmieren oder Briefe zu schreiben, will man sich auch mal bei einem Spielchen entspannen. Mit welchen Titeln das besonders gut geht, verraten wir Ih-



Grafisch abwechslungsreich: »The Devils Crown«



»Tir Na Nog« mit hervorragender Animation

nen in diesem Artikel. Die Bezugsquellen für die hier aufgeführten Spiele finden Sie am Ende. Die Preise für die Kassetten-Versionen bewegen sich in der Regel zwischen 25 und 39 Mark; einige Titel sind auch auf Diskette erhältlich.

Leider gibt es bis heute nur relativ wenige Spiele, die auf dem Schneider geschrieben wurden und die Fähigkeiten des Computers so richtig ausreizen. Bei den zahlreichen Umsetzungen von C 64-Spielen ist Vorsicht geboten. Die Schneider-Adaptionen erreichen fast nie die Qualität des Originals; die Handlung ist zwar meistens identisch, aber vor allem bei der Grafik hapert es mitunter gewaltig. Wer ein tolles Spiel auf dem Commodore 64 sieht, sollte mit einem spontanen Blind-

kauf der Schneider-Version etwas vorsichtig sein. Am besten schaut man sich die Umsetzung im Software-Shop vor dem Kauf an oder läßt sich über die Qualität der Umsetzung vertrauensvoll informieren.

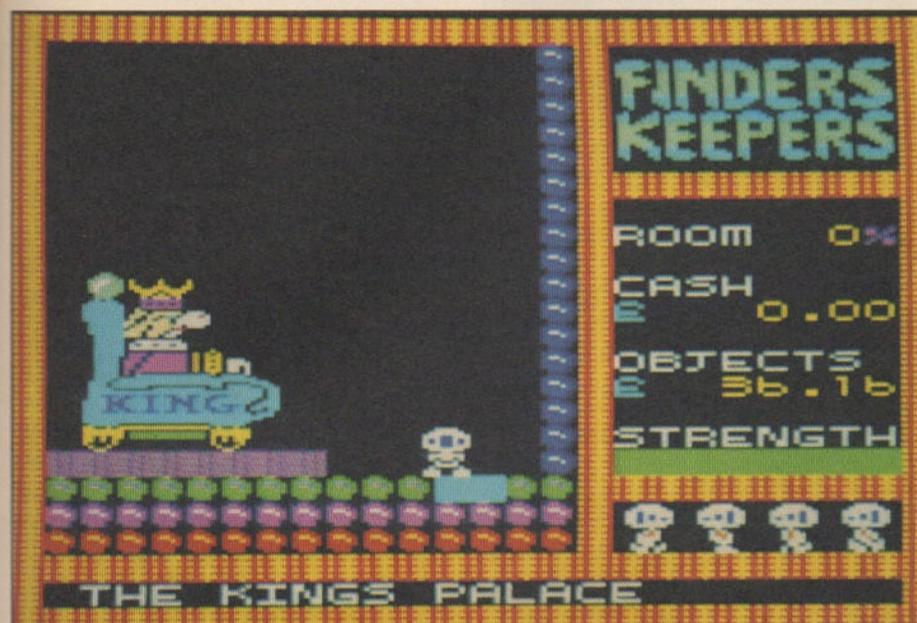
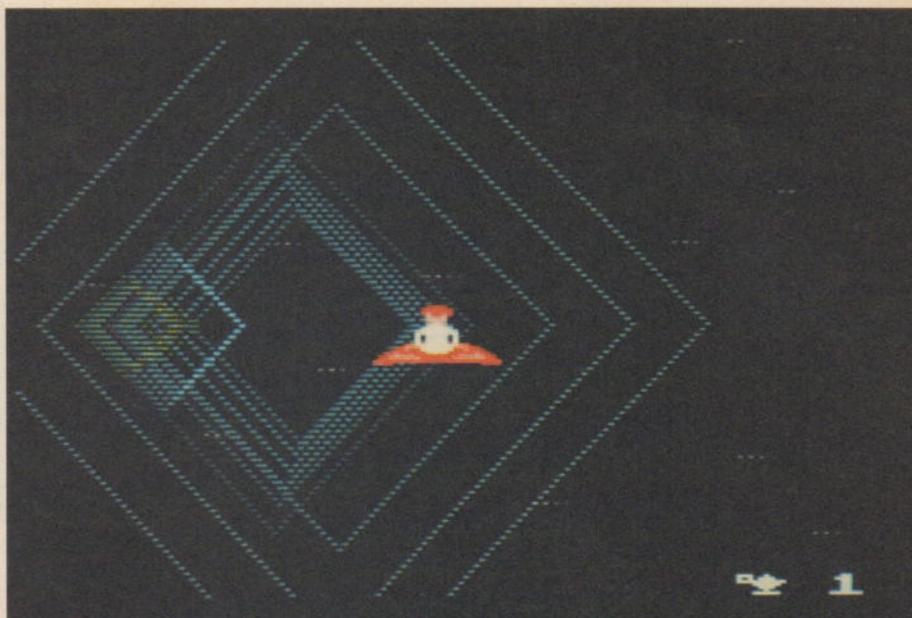
Gut umgesetzt ist halb gewonnen

Ein gelungenes Beispiel für eine gute Adaption ist »Bounty Bob strikes back«. Das Spiel hat eine simple Grundidee: Der Bergarbeiter Bounty Bob muß in 25 Bergwerksstollen alle Gerüstteile absichern. Dies geschieht dadurch, daß er über jedes Teil mindestens einmal läuft und so das Gerüst farbig ausfüllt. In das nächste Bild gelangt man erst, wenn man einen Stollen komplett abge-

räumt hat. Um bei diesem Spiel alle Screens kennenzulernen, ist man leicht ein paar Monate beschäftigt, da es nur so von tödlichen Mutanten und trickreichen Fallen wimmelt. »Bounty Bob« ist eines der besten Plattformspele überhaupt und — zumindest was meinen Geschmack angeht — wesentlich witziger und interessanter als der Bestseller »Jet Set Willy«.

Und noch eine gelungene Umsetzung: »Master of the Lamps« ist ein schnelles Geschicklichkeits-Spiel für C 64, das vor allem von den tollen Grafik- und Sound-Effekten lebt, da der Spielablauf eher simpel ist. Die Adaption der Musikstücke klingt auf dem Schneider dank des mickrigen

**Gelungene Umsetzung:
»Master of the Lamps«**



»Finders Keepers«, das preiswerte Action-Adventure

Lautsprechers äußerst flau, doch die flotte 3D-Grafik steht dem Original in nichts nach. Der Spieler düst als Prinz mit seinem Zauberteppich durch abstrakte Energie-Rauten, die in Windeseile auf ihn zuflitzen. Nach jeder heil überstandenen Flugsequenz kann man sich einen Teil der begehrten Königskrone schnappen, indem man eine Melodie richtig nachspielt. Vom Handlungsablauf her wirklich nicht besonders aufregend, aber ein tolles Spiel für Liebhaber schneller, schöner Grafik-Effekte.

Strategie mit Witz

Kevin Toms ist ein englischer Programmierer, der mit zwei sehr originellen Strategie-Spielen viel Erfolg hat: »Football Manager« und

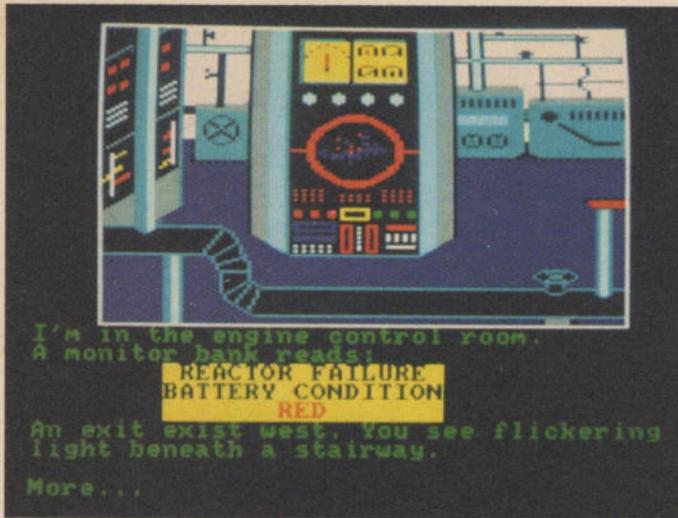
»Software Star«. Beide Programme sind nicht übermäßig kompliziert und schnell zu lernen, doch der Spielreiz ist auch längerfristig sehr hoch. Das liegt wahrscheinlich vor allem daran, daß man bei beiden Programmen einen Spielstand auf Kassette speichern und später von da aus wieder weitermachen kann — quasi ein Spiel fürs Leben. Beide Male schlüpft man in verantwortungsvolle Rollen: Einmal versucht man sich als Trainer einer Fußballmannschaft, einmal als Leiter einer Computersoftware-Firma. Bei »Football Manager« können Sie Spieler an- und verkaufen, Ihre Mannschaft aufstellen und die Kicker (hoffentlich) zu Siegen führen. Der Bundesliga-Aufstieg winkt und eines Tages vielleicht sogar die Meisterschaft. Bei »Software Star« geht es um Verkaufserfolge in den Computerspiel-

Charts. Sie bestimmen, wie lange ein Programm entwickelt wird, wieviel für Werbung investiert wird, etc. Beide Spiele ähneln sich etwas, sind inhaltlich aber sehr eigenständig. Einsteigern sei »Football Manager« empfohlen und wem dieses Spiel Spaß macht, der ist mit »Software Star« auch sehr gut bedient.

Ein neues Spielgenre erfreut sich immer wachsender Beliebtheit: die sogenannten »Action-Adventures«. Bei dieser Mischform muß man sowohl seine Spielfigur durch verschiedene Bilder steuern als auch Gegenstände aufsammeln und im richtigen Moment gebrauchen. Da man oft nur eine bestimmte Anzahl von Dingen bei sich tragen kann, sind Logik und natürlich viel Geduld gefragt. Ein ausgesprochen preiswerter Vertreter dieses Genres ist das zirka 10 Mark teure »Finders Keepers«, das etwas an diverse Plattformspele erinnert, aber auch Funktionen wie Gegenstände nehmen, verkaufen etc. erlaubt. Inhaltlich ähnlich, aber grafisch abwechslungsreicher, ist »The Devils Crown«. Hier müssen Sie durch immerhin 40 Bilder flitzen und sieben Juwelen finden. Behindert wird man dabei von liebreizenden Zeitgenossen wie den Geister-Piraten und den Killer-Fischen (nicht zu verwechseln mit Kuno, dem Killer-Karpfen).

Trickfilm-Grafik

Wer eine hervorragend animierte Grafik auf seinem Schneider erleben will, greife zu »Tir Na Nog« und »Dun Darach«. Auch diese beiden exotisch klingenden Titel kann man als Action-Adventures bezeichnen.



Disketten-Adventure »Frankie crashed on Jupiter« ...



... mit witzigen Texten und Grafiken

Hier muß man in Adventure-Manier wieder eine bestimmte Aufgabe erfüllen, doch das Besondere an den beiden Spielen ist die grafische Gestaltung. Die Figur, die der Spieler steuert, wandelt beinahe in Zeichentrick-Qualität über die Mattscheibe. Im Gegensatz zu »echten« Adventures ist man bei seinen Handlungen allerdings beschränkt: Bei »Dun Darach« kann man den Blickwinkel wählen, Gegenstände aufsammeln, hinlegen und anderen Spielfiguren anbieten. Auch hier darf man nur eine bestimmte Anzahl von Dingen bei sich tragen: drei Gegenstände plus Bargeld nach Belieben.

Doch wir wollen auch die »richtigen« Adventures nicht vergessen. Nahezu alle Schneider-Abenteuerspiele werden auf Kassette angeboten und laden nicht nach; Text und Grafiken müssen also in die 64 KByte Arbeitsspeicher gestopft werden. Daß dabei trotzdem ein gutes Spiel mit vielen Bildern und interessanten Texten herauskommen kann, beweisen die Titel der englischen Adventure-Spezialisten Level 9. Ihr jüngstes Werk »Red Moon« liegt auch in einer sehr guten CPC-Fassung vor.

Ein englischsprachiges Grafik-Abenteuer der klassischen Art mit über 200 Orten, guten bis sehr guten Bildern und einer interessanten Fantasy-Handlung mit viel Magie.

Vor allem für Einsteiger ist das Disketten-Adventure »Frankie crashed on Jupiter« interessant. Das Spiel ist nicht allzu schwierig und macht dank witziger, englischer Texte und mitunter sehr sehenswerten Grafiken Spaß.

In diesem Artikel konnten wir aus Platzgründen nur einige interessante Schneider-Spiele kurz anreißen. Für Interessierte folgt zum krönenden Abschluß deshalb etwas hemmungslose Eigenwerbung: Wer sich für Computerspiele im allgemeinen interessiert, sollte mal einen Blick auf das Sonderheft »Spiele-Tests« von Happy-Computer werfen. Neben Stories und einer Marktübersicht findet man hier über 100 kritische Tests. Viele der vorgestellten Spiele sind auch für den Schneider CPC erhältlich. Und wer sich aktuell jeden Monat über das Spiele- und sonstige Schneider-Geschehen informieren will, greife jeden Monat zur neuen Happy-Computer.

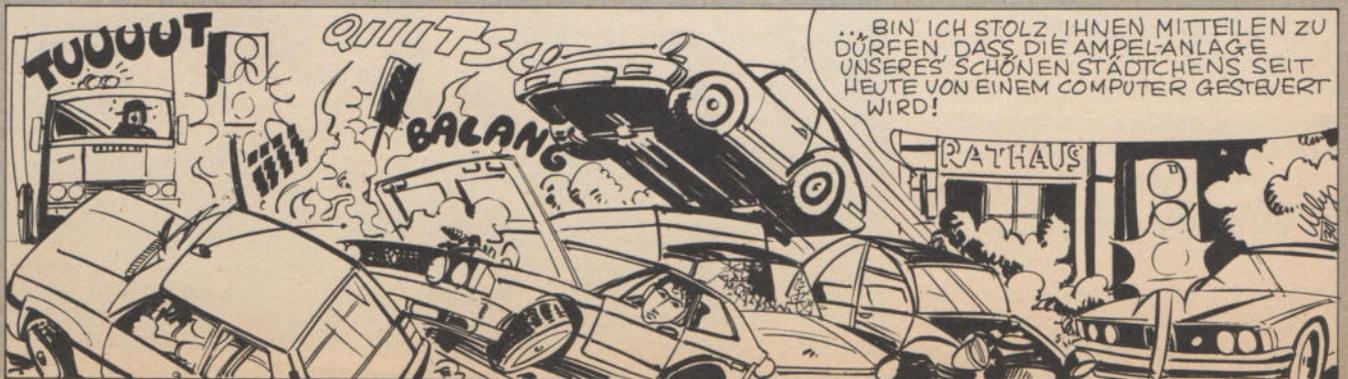
Zu guter Letzt noch ein kleiner Ausblick auf kommende Schneider-Spiele: So ist für Dezember 1985 die langerwartete Auslieferung der CPC-Adaption des Kultspiels »Elite« angekündigt. Die Grafik ist mit der erfolgreichen C 64-Version so gut wie identisch; inhaltlich wird sogar noch mehr geboten: Die Schneider-Fassung enthält nämlich zwei Spezial-Missionen mehr.

Eine neue englische Softwarefirma namens Electric Dreams beschäftigt einige erfahrene Z80-Experten in seinem Programmiererteam. Von den ersten drei Spielen dieser Firma bekamen wir bisher ein Video-Demo zu sehen. Vor allem »Winter Sports«, die Antwort auf »Winter Games« für den C 64, dürfte ein Riesenhit werden. Das Sportspiel bietet neun Disziplinen inklusive einer starken Eishockey-Simulation. Auf diese Neuerscheinungen kann man nur gespannt sein. Sie zeigen deutlich, daß der Schneider spielerisch noch lange nicht ausgereizt ist.

(Heinrich Lenhardt)

Bezugsquellen:
 »Frankie crashed on Jupiter« gibt's bei Kingsoft, Fritz Schäfer, Schnackebusch 4, 5106 Roetgen, Tel. (024 08) 83 19
 Alle anderen Spiele erhalten Sie bei Joysoft, Humboldtstr. 84, 4000 Düsseldorf 1, Tel. (02 11) 680 1403

Kosinus von GUBA & ULLY



**DER GROSSE SCHNEIDER-SONDERTEIL
 IST FÜR VIELE SCHNEIDER-ANWENDER
 GRUND GENUG, SICH JEDEN MONAT
 DIE NEUESTE AUSGABE
 VON »HAPPY-COMPUTER«
 ZU KAUFEN.**

Das Januar-Heft bringt:

- ★ Listing: Über 50 verschiedene Schriftarten auf Ihrem Schneider-Drucker
- ★ Hardware-Test: Speicher-Erweiterungskarte mit bis zu 512 KByte RAM
- ★ Listing: Schnelles Drucken mit Zwischenspeicher
- ★ Software-Test: Platinenlayout-Programm
- ★ Listing: Basic-Erweiterung ohne RSX

**Das Januar-Heft
 erhalten Sie ab 9.12.
 überall im Zeitschriftenhandel!**

Außerdem läßt »Happy-Computer« in einem Vergleichstest Amiga und Atari 520 ST gegeneinander antreten und nimmt Software-Preisbrecher unter die Lupe: Wie gut sind Spiele unter 10 Mark? Sechs Programmiersprachen stellen sich vor: Assembler, Basic, Logo, Forth, Pascal und »C«. Ein Grundlagen-Beitrag sagt Ihnen, wie die Sprachein- und ausgabe funktioniert und natürlich finden auch C64-Fans wieder ihren eigenen Sonderteil mit der Fortsetzung eines Grafik-Kurses, dem Floppy-Beschleuniger »Ultra-Load« und einem Assembler zum Abtippen.

WWW.HOMECOMPUTERWORLD.COM



Gutschein

FÜR EIN KOSTENLOSES PROBEEXEMPLAR VON HAPPY COMPUTER

JA, ich möchte »Happy-Computer« kennenlernen. Senden Sie mir bitte die aktuellste Ausgabe kostenlos als Probeexemplar. Wenn mir »Happy-Computer« gefällt und ich es regelmäßig weiterbeziehen möchte, brauche ich nichts zu tun: Ich erhalte »Happy-Computer« dann regelmäßig frei Haus per Post und bezahle pro Jahr nur DM 66,- statt DM 72,- Einzelverkaufspreis (Ausland auf Anfrage).

Vorname, Name

Straße

PLZ, Ort

Datum

1. Unterschrift

Mir ist bekannt, daß ich diese Bestellung innerhalb von 8 Tagen bei der Bestelladresse widerrufen kann und bestätige dies durch meine zweite Unterschrift. Zur Wahrung der Frist genügt die rechtzeitige Absendung des Widerrufs.

Datum

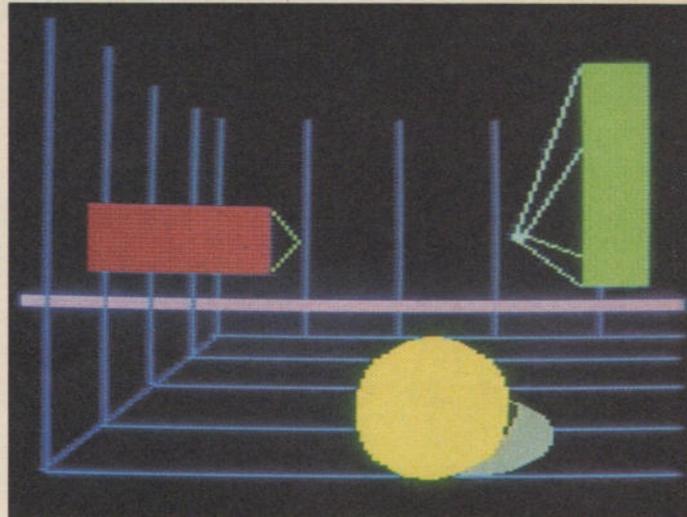
2. Unterschrift

Gutschein ausfüllen, ausschneiden, in ein Kuvert stecken und absenden an: Markt & Technik Verlag Aktiengesellschaft, Vertrieb, Postfach 1304, 8013 Haar

Ein Zuckerhut für den Schneider



Ein beliebtes Demonstrationsojekt auf dem Computer sind Funktionen, die von zwei Variablen abhängen. Die spektakulär aussehenden 3D-Grafiken sind einfacher als man denkt.



Aus den Werbeprospekten für teure Profi-Computer kennt sie jeder — die Zuckerhüte, gebogenen Ebenen und viele andere beeindruckende Grafiken. Daß es sich im Endeffekt meist nur um sehr einfache mathematische Formeln handelt, wissen nur wenige. Wie programmiert man nun eine solche Funktion, die von zwei Variablen abhängt? Folgende Überlegungen sollen uns zu einer universellen Darstellungsroutine führen, die es erlaubt, jeden Graphen in einem frei wählbaren Bereich abzubilden. Die notwendigen Unterfunktionen (wie beispielsweise Sinus (SIN) und Cosinus (COS) müssen allerdings in der Syntax des Computers vorgesehen sein.

Bevor wir uns endgültig in die Problematik vertiefen, müssen wir uns zunächst wohl oder übel mit einigen grundsätzlichen Problemen im Bereich der Funktionsdarstellung beschäftigen. Bleiben wir dabei zunächst im zweidimensionalen Bereich und beginnen mit einer ganz einfachen Funktion, die von nur ei-

ner Variablen abhängig ist: $y = \sin(x)$. Wenn Sie die Schleife »ORIGIN 0,200:FOR i=0 TO 4*PI:PLOT i,SIN(i):NEXT i« eingeben, sollte Ihr Schneider eine Sinus-Kurve auf dem Schirm ausgeben.

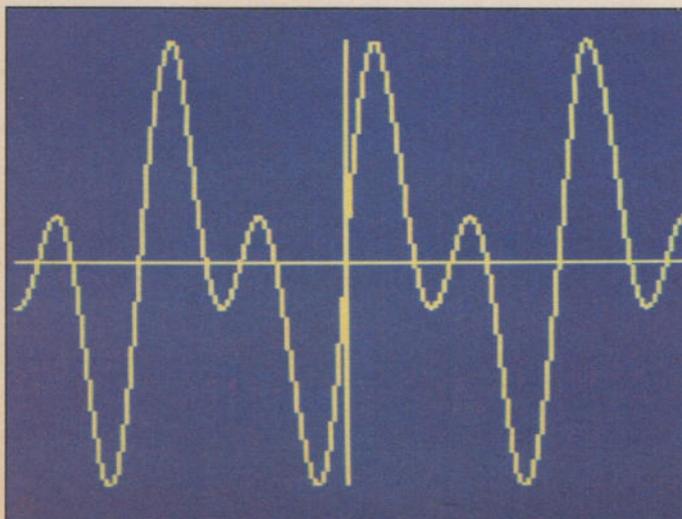
Wir erhalten jedoch nur einen kleinen geraden Strich als Resultat. Der Grund: falsche Skalierung! Die Sinus-Funktion schwankt bekanntlicherweise nur in einem relativ geringen Wertebereich zwischen -1 und 1 . Damit werden nur maximal zwei Bildschirmlinien in der Höhe adressiert. Ohne Vergrößerung können diese Werte aber nicht ausreichend gut aufgelöst gezeichnet werden. Führen wir einen Multiplikator ein, so ist dieses Problem gelöst. Da die Schwankungsbreite bekannt ist, können wir ohne weiteres errechnen, wie groß unser Multiplikator sein soll, um den Schirm in Y-Richtung möglichst optimal auszufüllen. Wir verfügen über 400 Pseudopunkte in der Höhe. Den Ursprung unseres Koordinatensystems legen wir in die Bildschirmmitte (Bildschirmzeile 200). Zur Dar-

stellung unserer Sinusfunktion können wir also 200 Punktlinien nach oben und unten nutzen. Da der Sinus zwischen -1 und 1 schwankt, wird bei einer Multiplikation mit 200 demnach der gesamte Schirm ausgefüllt.

Man muß sich allerdings bei diesen Überlegungen darüber im klaren sein, daß man nur 400 Bildschirmseudopunkte zur Verfügung hat. Effektiv kann man nur mit 200 einzelnen Bildpunkten arbeiten. Dies läßt sich mit einem kleinen Experiment belegen. Geben Sie einmal nach CLS die folgenden Kommandos »PLOT 320,200« und »PLOT 320,201« ein.

Nach dem ersten PLOT erscheint ein gelber Punkt in der Bildschirmmitte. Das zweite PLOT dagegen hat keinerlei Auswirkungen. Der Grund ist einfach. Beide PLOT-Befehle adressieren denselben Bildpunkt. In Wirklichkeit haben wir es also horizontal nur mit 200 trennbaren Bildschirmlinien zu tun. Einen ähnlichen Effekt kann man auch bei der Betrachtung der X-Angaben in den PLOT- und DRAW-Befehlen erkennen. In den Modi mit höherer Farbauflösung werden Punkte in X-Richtung zusammengefaßt. Dabei maximal vier PLOT-Kommandos im Mode 0, dieselbe Punktgruppe, nämlich »PLOT 320,200«, »PLOT 321,200«, »PLOT 322,200« und »PLOT 323,200«.

Um diese Zusammenhänge zu verdeutlichen, wurde deshalb der Begriff Pseudopunkte gebraucht. Es stellt sich die Frage, warum der Schneider eigentlich einen viel größeren Zahlenbereich für die Ansprache verwendet, als er dann effektiv benötigt. Dafür gibt es zwei Gründe: Zum einen ist es durch eine größere Wertangabe für die Y-Komponente möglich, bei schräglaufen-



Im Modus 0 sieht die Kurve sehr eckig aus

den Linien den optimal zu setzenden Bildpunkt durch Interpolation genauer zu bestimmen. Zum anderen sollte dadurch ein besseres Verhältnis von X- zu Y-Werten erreicht werden. Beim Zeichnen von Kreisen beispielsweise ergäbe sich bei gleichem Abstand ansonsten eine Ellipse. Für unsere weiteren Programme sind diese Überlegungen für die Frage der Auflösung wichtig. Es hat beispielsweise keinen Sinn, die Abstände bei einer parallel zur X-Achse verlaufenden Linienschar in Y-Richtung in $\frac{1}{400}$ -Schritten zu variieren. Dadurch würde nur zweimal derselbe Punkt gesetzt. Außer einer längeren Rechenzeit wäre das Resultat einer solchen »Verbesserung der Auflösung« gleich Null. Wenn wir auch bei unseren Berechnungen weiterhin den Rahmen von 640 x 400 Bildpunkten als Grundlage benutzen müssen, können wir aber gegebenenfalls in größeren als lers-Schritten vorgehen.

Bei der Programmentwicklung müssen wir zunächst unsere Funktion nach demselben Verfahren auch in X-Richtung strecken. Denn auch hier »kleben« die einzelnen Punkte noch viel zu dicht aufeinander. 4PI entspricht ungefähr dem Wert 12,5.

Da wir 640 Pseudopunkte in Richtung der Abszisse zur Verfügung haben, liefert uns hier $630/12,5=51,2$ den optimalen Multiplikationsfaktor. Mit »ORIGIN 0,200:FOR i=0 TO 4*PI:PLOT 51,2*i,200*SIN(i)« ist dann eine vernünftige Darstellung erreicht. Wie sieht nun die Funktion im negativen Bereich aus? Dazu verändern wir den Bereich, in dem i variiert, zu »FOR i=-2*PI TO 2*PI«. Statt des erhofften Kurvenverlaufs zwischen -2PI und 0 wird jetzt nur noch ein Teil der ersten Schwingung dargestellt, daß der Ursprung unseres Koordinatensystems ungünstig gewählt ist. Der Nullpunkt liegt nämlich ganz am linken Bildschirmrand, wodurch der Computer die Werte, die sich für negative X ergeben, nicht mehr darstellen kann. Die richtige Wahl der Skalierung und des Mittelpunktes stellt aber kein Problem dar, falls man die Maxima- und die Minima-Angaben für X beziehungsweise Y kennt. Es seien XMIN — die minimale X-Koordinate, XMAX — die maximale X-Koordinate, YMIN — die minimale Y-Koordinate und YMAX — die maximale Y-Koordinate.

Der optimale Punkt für den Koordinatenursprung berechnet sich

dann nach folgenden Formeln (siehe Bild 1):

$$X_0 = \frac{-X_{\min}}{X_{\max}-X_{\min}} \cdot 639 \text{ und}$$

$$Y_0 = \frac{-Y_{\min}}{Y_{\max}-Y_{\min}} \cdot 399$$

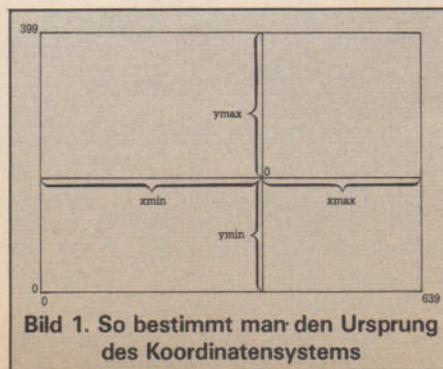
Als Nebenbedingung muß jedoch $X_{\min}, Y_{\min} < 0$ und $X_{\max}, Y_{\max} < 0$ gelten, damit dieser Punkt auch mit dem Ursprung des Koordinatensystems identisch ist. Als nächstes benötigen wir noch die Skalierungsfaktoren, die auch relativ einfach zu bestimmen sind.

$$CX = \frac{X_{\max}-X_{\min}}{639} \text{ und}$$

$$CY = \frac{Y_{\max}-Y_{\min}}{399}$$

Offen ist jetzt noch die Frage, woher wir die Angaben für Y_{\max} und Y_{\min} erhalten. Einmal können wir diese, genau wie auch den Wertebereich von X, über eine Eingaberoutine abfragen. Voraussetzung dafür ist aber, ungefähre Vorstellung von der Größe der auftretenden Ergebnisse zu haben. Speziell bei komplexen Funktionen, die von einer ganzen Reihe von Unterfunktionen abhängig sind, ist dies nicht praktikabel. Es gibt aber die Möglichkeit, eine automatische Grenzbestimmung durch ein Programm machen zu lassen. Das heißt, der Schneider bestimmt selbständig die Extremwerte und danach den Ursprung des Koordinatensystems sowie die Skalierungsfaktoren. Dazu wird unsere Zeichenschleife einfach zweimal durchlaufen. Das erste Mal allerdings in einer leicht veränderten Form. Statt des PLOT-Kommandos werden mit »MIN« und »MAX« die Extremwerte bestimmt. Diese neue Zeile lautet dann in unserem Beispiel:

```
»YMAX = -1E38:YMIN = 1E38:FOR
i = 0 TO 4*PI:YMAX = MAX(YMAX,
SIN(i)):YMIN = MIN(YMIN,SIN(i)):
NEXT i«.
```



Zunächst werden die Variablen, die die Grenzwerte aufnehmen sollen (XMAX und YMIN), auf möglichst kleine beziehungsweise große Werte gesetzt, um eine definierte Ausgangsposition zu schaffen. Als nächstes werden dann für alle i die Funktionswerte bestimmt und parallel dazu, wenn nötig, die Eckangaben in YMAX und YMIN verändert. Damit können wir nun eine einfache Funktionsdarstellungs-Routine (Listing 1) schreiben.

Am Anfang finden Sie die automatische Extremwertbestimmung, bevor dann Mittelpunkt und Streckungsfaktor festgelegt werden. Es folgt die Darstellung des Achsenkreuzes und der Funktion, die in Zeile 250 steht. Dieses Programm erhebt keinerlei Anspruch darauf, bereits eine komfortable Darstellungsroutine zu sein. Es soll nur das Prinzip illustrieren und vor dem Einstieg in die dreidimensionale Grafik eine lauffähige zweidimensionale Version präsentieren.

Mehrere Punkte lassen bei dem Beispiel noch zu wünschen übrig. Zum einen ist die Funktionseingabe mangelhaft. In einem benutzerfreundlichen Programm kann es nicht angehen, daß dieses abgebrochen und nach Eingabe der mathematischen Formel (hier in Zeile 230) gestartet werden muß. Ferner sollte der Computer die Achsen selbständig bezeichnen, damit konkrete Angaben für einzelne Punkte zu erkennen sind. Wir wollen diese Mängel im Kopf behalten, wenn wir uns nun damit beschäftigen, wie das 3D-Funktionsprogramm aussehen soll.

3D-Funktion

Grundlage jeder dreidimensionalen Abbildung sind die Projektionsgleichungen. Wenn wir eine Funktion darstellen wollen, die von zwei Variablen abhängig ist, so benötigen wir zunächst einmal ein dreidimensionales Achsenkreuz. Auf der x-Achse und der in den Raum führenden y-Achse (diese ist nicht mit der y-Achse im zweidimensionalen Koordinatensystem identisch) trägt man dabei die sich ändernden Eingabegrößen auf. Der Wert auf der senkrecht dazu stehenden z-Achse gibt das Ergebnis der Funktion wieder. Nun haben wir aber als Darstellungsfläche nur die zweidimensionale Bildschirmenebene zur Verfügung. Um dennoch eine dreidimensionale Funktion — eine Funktion, die von zwei verschiedenen Variablen abhängig ist — abbilden zu können, müssen wir daher auf eine

Projektion (perspektivische Darstellung) zurückgreifen. Dazu teilen wir die Angaben für y so auf x und z auf, daß ein räumlicher Eindruck entsteht. Als Verfahren dient uns dabei die Parallelprojektion.

Auf der Basis dieser Projektionstechnik zerlegen wir den Wert für y mit Hilfe von Winkelfunktionen in eine x - und eine z -Komponente. Wie diese Verschiebung sich auswirkt, verdeutlicht Bild 2. Bei $y=0$ haben wir es mit einer ganz normalen zweidimensionalen Darstellung zu tun. In der Zeichnung wird dies durch den Beispielpunkt $(5/0,5)$ verdeutlicht, wobei die Koordinaten in der Reihenfolge x , y und z angegeben sind. Die Verschiebung, die sich ergibt, wenn zusätzlich auch noch $y=5$ gesetzt ist, zeigt der zweite Beispielpunkt. Je kleiner d gewählt wird, desto stärker schlägt y auf die darzustellende x -Komponente durch. Umgekehrt führen große Winkel zu einer erheblich über das Maximum von z heranwachsenden yz -Komponente.

Nun sind wir aber zunächst einmal in der Lage, aus den dreidimensionalen Angaben unsere entsprechenden zweidimensionalen Darstellungswerte zu berechnen. Haben wir also mit x , y und z die Werte im dreidimensionalen Koordinatensystem und mit d den Winkel zwischen der x - und der y -Achse in der Darstellungsebene gegeben, so ergeben sich für eine Umformung die Formeln $X = x + y \cdot \cos \alpha$ und $Z = z + y \cdot \sin \alpha$.

In X und Z sind die Angaben für das zweidimensionale Achsensystem als Abszissen- und Ordinatenwert enthalten. Wenn wir nun also in einer Doppelschleife x und y variieren und die zugehörigen z -Werte bestimmen, so können wir mit Hilfe dieser Formeln und gegebenenfalls den weiter oben schon beschriebenen Umrechnungsoperationen X und Z zur Skalierung die Werte bestimmen, die in unserer zweidimen-

sionalen Darstellung gesetzt werden müssen.

Hierbei ergibt sich allerdings ein Problem. Durch die Projektion der y -Werte auf die x - und z -Achse kann es, wie wir ja schon gesehen haben, dazu kommen, daß die Extremwerte für X und Z weit über die Angaben hinausgehen, die wir für X und Z als Grenzwerte angegeben haben. Formal sind dafür in den Gleichungen die Terme mit den Winkelfunktionen verantwortlich. Durch die Projektion ergibt sich das Maximum von X nun beispielsweise nicht mehr als Maximum von x , sondern zu diesem Term ist auch noch das Maximum von y multipliziert mit dem Projektionsfaktor $\cos \alpha$ zu addieren. Dies beeinflusst sowohl die Wahl des Mittelpunktes als auch die Werte für die Skalierungsfaktoren. Bild 3 zeigt die Zusammenhänge. Obwohl die Schwankungsbreite für y deutlich kleiner ist als die Veränderungen für x und z liegt der rechte Extrempunkt p dennoch weit außerhalb unserer Schranken. Für die Bestimmung des größten und kleinsten darzustellenden Werts (sowohl in X - wie auch in Z -Richtung) müssen wir also unsere Bestimmungsgleichungen verändern. Wir bezeichnen das Minimum von X mit MIX , das Maximum mit MX . Das Minimum von Z erhält die Bezeichnung MIZ und der zugehörige Maximalwert wird mit MZ beschrieben. Des weiteren definieren wir:

$$\begin{aligned} \text{MINYPX} &= \text{MINY} \cdot \cos \alpha \\ \text{MINYPZ} &= \text{MINY} \cdot \sin \alpha \\ \text{MAXYPX} &= \text{MAXY} \cdot \cos \alpha \\ \text{MAXYPZ} &= \text{MAXY} \cdot \sin \alpha \end{aligned}$$

wobei Minimum (y) und Maximum (y) die Extremwerte für y in der dreidimensionalen Darstellung, MINYPX , MINYPZ und so weiter ihre Projektionen auf die zweidimensionalen X - beziehungsweise Z -Achse sind. Die absoluten Schranken ergeben sich dann als

$$\begin{aligned} \text{MIX} &= \text{MIN}(\text{MINYPX} + \text{MINX}, \\ &\text{MINX}, 0) \\ \text{MX} &= \text{MAX}(\text{MAXYPX} + \text{MAXX}, \\ &\text{MAXX}, 0) \\ \text{MIZ} &= \text{MIN}(\text{MINYPZ} + \text{MINZ}, \\ &\text{MINZ}, 0) \\ \text{MZ} &= \text{MAX}(\text{MAXYPZ} + \text{MAXZ}, \\ &\text{MAXZ}, 0) \end{aligned}$$

All diese Formen beruhen auf dem gleichen Prinzip. Es ist ein Extremwert aus mehreren Grenzwerten zu bilden, die sich aus den Achsenextrema mit und ohne Projektion ergeben. Die 0 wurde dabei in allen

Gleichungen eingeführt, um sicherzustellen, daß der Nullpunkt unseres Koordinatensystems auf dem Bildschirm dargestellt wird. Er ist ja eine wichtige Orientierungshilfe. Wenn die Maxima auf jeder Achse kleiner als Null gewählt wurde, beziehungsweise sich als keiner Null ergeben, so würde der Nullpunkt nicht abgebildet werden. Derselbe Fall tritt natürlich auch ein, wenn alle drei Wertebereiche deutlich positiv gewählt wurden, also alle Minima im positiven Bereich liegen. Aus Gründen der Auflösung kann es aber in manchen Fällen geraten sein, auf die Darstellung des Achsenkreuzes zu verzichten. Dies ist ratsam, wenn man in relativ weiter Entfernung vom Achsenkreuz einen nur kleinen Wertebereich analysieren will (beispielsweise bei $50 < x < 51$, $50 < y < 51$). Wenn man hier das Achsenkreuz darstellen läßt, so kleben am rechten Bildschirmrand einige Funktionswerte, am linken Rand wird das Achsenkreuz abgebildet und der Zwischenraum bleibt leer. In solch einem Fall sind die Nullvergleichswerte zur Bestimmung der Extremwerte zu streichen.

Die Koordinaten des Nullpunktes erhalten wir nun in Analogie zur zweidimensionalen Darstellung. Dabei wurde allerdings unser Bildschirmfenster in der Höhe etwas verkleinert, um oberhalb und unterhalb der Darstellung noch Platz für Beschriftungen und so weiter zu reservieren. Dafür wurden jeweils 40 Bildschirmlinien vorgesehen. Es gilt dann:

$$X_0 = \frac{-\text{MIX}}{\text{MX} - \text{MIX}} * 639$$

und

$$Y_0 = 40 + \frac{-\text{MIZ}}{\text{MZ} - \text{MIZ}} * 320$$

Nach dieser »Vorarbeit« können wir uns nun daran machen, unser 3D-Darstellungsprogramm zu schreiben (Listing 2).

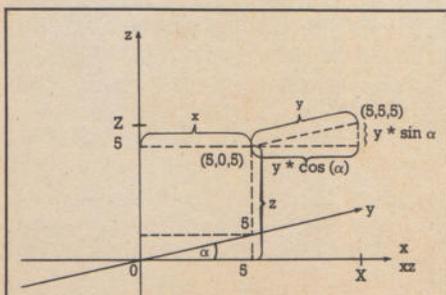


Bild 2. Die dritte Dimension muß in die Ebene projiziert werden

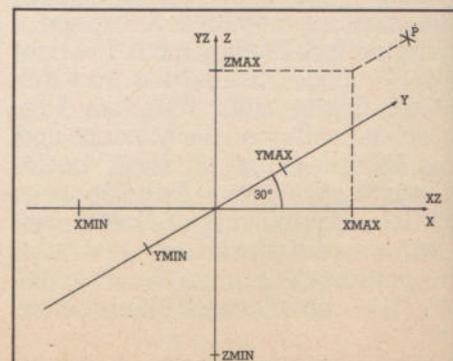


Bild 3. Die Extrempunkte der zweidimensionalen Abbildung liegen weiter auseinander

```

10 *****
20 ** 2-D-Graph **
30 *****
40 CLS
50 ymin=1E+38:ymax=-1E+38
60 INPUT"Untergrenze X";xmin
70 INPUT"Obergrenze(2 SPACE)X";xmax
80 sx=(xmax-xmin)/639
90 PRINT"Sind die Y-Extrema bekannt j/n"

100 z$=LOWER$(INKEY$):IF z$="n" THEN 140
    ELSE IF z$<>"j" THEN 100
110 INPUT"Untergrenze Y";ymin
120 INPUT"Obergrenze(2 SPACE)Y";ymax
130 GOTO 150
140 ymin=1E+38:ymax=-1E+38:FOR x=xmin TO
    xmax STEP sx:GOSUB 250:ymax=MAX(yma
    x,y):ymin=MIN(ymin,y):NEXT x
150 cx=639/(xmax-xmin)
160 cy=399/(ymax-ymin)
170 ox=-xmin/(xmax-xmin)*639
180 oy=-ymin/(ymax-ymin)*399
190 ORIGIN ox,oy
200 CLS
210 PLOT 0,0:DRAWR 0,400:DRAWR 0,-800:MO
    VE 0,0:DRAWR 640,0:DRAWR-1280,0
220 x=xmin:GOSUB 250:PLOT x*cx,y*cy
230 FOR x=xmin TO xmax STEP sx:GOSUB 250
    :DRAW x*cx,cy*y:NEXT x
240 GOTO 240
250 y=COS(3*x)*SIN(x)+SIN(2*x)
260 RETURN
    
```

Listing 1. Zweidimensionale Kurven sind leicht gezeichnet

```

10 *****
20 ** 3D-Graphikmaster **
30 **by Carsten Straush**
40 ** 9/10/1985 **
50 *****
60 *****
70 ** Initialisierung **
80 *****
90 MODE 1:INK 0,0:INK 1,24:INK 2,11:PEN
    1

100 CLS:DEG
110 co=COS(18):si=SIN(18):ta=si/co
120 RAD
130 *****
140 ** Funktionsabfrage **
150 *****
160 KEY DEF 45,1,106:KEY DEF 46,1,110
170 PRINT"Bitte geben Sie die Funktion e
    in":INPUT f$
180 PRINT"Ist diese Funktion o.k. (j/n)?
    "
190 KEY 131,"Goto 90"+CHR$(13):KEY DEF 4
    6,1,131
200 PRINT:PRINT f$:PEN 0:KEY 128,"1210 "
    +f$+CHR$(13)+"goto 250"+CHR$(13)
210 KEY DEF 45,1,128:END
220 *****
230 ** Werte einlesen **
240 *****
250 PEN 1:PRINT"Bitte geben Sie den Bere
    ich ein,(2 SPACE)in demdie X-Werte d
    iffieren sollen."
260 KEY DEF 45,1,106:KEY DEF 46,1,110
270 PRINT
280 INPUT"Minimalwert X";minx
290 INPUT"Maximalwert X";maxx
300 IF maxx<minx THEN PRINT"Falsche Eing
    abe":PRINT:GOTO 280
310 PRINT
320 PRINT"Bitte geben Sie den Bereich ei
    n,(2 SPACE)in demdie Y-Werte differi
    eren sollen."
330 PRINT
340 INPUT"Minimalwert Y";miny
350 INPUT"Maximalwert Y";maxy
360 IF maxy<miny THEN PRINT"Falsche Eing
    abe":PRINT:GOTO 340
370 *****
380 ** Darstellungsart **
390 *****
400 PRINT"Bitte geben Sie die Darstellun
    gsort ein:"
410 PRINT"laufende X-Werte(5 SPACE)(1)"
420 PRINT"laufende Y-Werte(5 SPACE)(2)"
430 PRINT"Vernetzung(11 SPACE)(3)"
440 z$=INKEY$:IF z$="" THEN 440 ELSE IF
    ASC(z$)<49 OR ASC(z$)>51 THEN 440 EL
    SE d=VAL(z$)
450 IF d=1 THEN sx=(maxx-minx)/100:sy=(m
    axy-miny)/100
460 IF d=2 THEN sx=(maxx-minx)/100:sy=(m
    axy-miny)/100
470 IF d=3 THEN sx=(maxx-minx)/100:sy=(m
    axy-miny)/100
480 CLS:PRINT"Grenzwertbestimmung"
490 PRINT"Sind die Grenzwerte fuer Z bek
    annnt j/n?"
    
```

```

500 z$=LOWER$(INKEY$):IF z$="j" THEN 610
    ELSE IF z$<>"n" THEN 500
510 *****
520 ** autom. Grenzwertbestimmung **
530 *****
540 maxx=1E+38:minz=1E+38:mix=1E+38:mx=
    -1E+38
550 FOR y=maxy TO miny STEP -sy
560 FOR x=maxx TO minx STEP -sx
570 GOSUB 1210
580 maxx=MAX(maxx,z):minz=MIN(minz,z)
590 NEXT x,y
600 GOTO 660
610 INPUT"Minimalwert Z";minz
620 INPUT"Maximalwert Z";maxz
630 *****
640 ** Koordinatensystem berechnen **
650 *****
660 minypx=co*miny:minypz=si*miny
670 maxypx=co*maxy:maxypz=si*maxy
680 mix=MIN(minypx+minx,minypx,minx,0):m
    iz=MIN(minypz+minz,minypz,minz,0)
690 mx=MAX(maxypx+maxx,maxypx,maxx,0):m
    az=MAX(maxypz+maxz,maxypz,maxz,0)
700 ORIGIN -mix/(mx-mix)*639,40+(-miz/(m
    az-miz))*320
710 CLS
720 cx=(mx-mix)/639
730 cy=(maz-miz)/320
740 *****
750 ** Funktion darstellen **
760 *****
770 ** laufendes Y **
780 *****
790 IF d=1 THEN GOSUB 930:GOTO 1030
800 IF d=2 THEN GOSUB 820:GOTO 1030
810 IF d=3 THEN GOSUB 930:GOSUB 820:GOTO
    1030
820 FOR x=maxx TO minx STEP -sx
830 y=maxy:GOSUB 1210:GOSUB 1230
840 PLOT xz,yz,2
850 FOR y=maxy TO miny STEP -sy
860 GOSUB 1210:GOSUB 1230
870 DRAW xz,yz,2
880 NEXT y,x
890 RETURN
900 *****
910 ** laufendes X **
920 *****
930 FOR y=maxy TO miny STEP -sy
940 x=maxx:GOSUB 1210:GOSUB 1230
950 PLOT xz,yz,2
960 FOR x=maxx TO minx STEP -sx
970 GOSUB 1210:GOSUB 1230
980 DRAW xz,yz,2:MOVE xz,yz-2:DRAW xz,-
    400,0
990 NEXT x,y:RETURN
1000 *****
1010 ** Eckangaben und Kreuz **
1020 *****
1030 MOVE -320,0:DRAWR 960,0,1:MOVE 0,-4
    00:DRAWR 0,800:MOVE 0,0:DRAWR 640,2
    00:DRAWR-1280,-400
1040 MOVE minx/cx,0:DRAWR 0,10,3:DRAWR 0
    ,-20
1050 PRINT CHR$(22)+CHR$(1):TAG:PRINT RO
    UND(minx,2);
1060 MOVE maxx/cx,0:DRAWR 0,10:DRAWR 0,-
    20
1070 MOVER -40,0:PRINT ROUND(maxx,2);
1080 MOVE 0,maxz/cy:DRAWR -10,0:DRAWR 20
    ,0
1090 MOVER 0,8:PRINT ROUND(maxz,2);
1100 MOVE 0,minz/cy:DRAWR -10,0:DRAWR 20
    ,0
1110 MOVER 0,8:PRINT ROUND(minz,2);
1120 MOVE maxypx/cx,ta*maxypx/cx:DRAWR -
    5,12:DRAWR 10,-24
1130 MOVER-40,0:PRINT ROUND(maxy,2);
1140 MOVE minypx/cx,ta*minypx/cx:DRAWR -
    5,12:DRAWR 10,-24
1150 PRINT ROUND(miny,2);
1160 TAGOFF:PRINT CHR$(22)+CHR$(0)
1170 IF INKEY$="" THEN 1170 ELSE RUN
1180 *****
1190 ** Formel ausrechnen **
1200 *****
1210 z=x*y
1220 yz=si*y+z:RETURN
1230 xz=(co*y+x)
1240 yz=yz/cy:xz=xz/cx
1250 RETURN
1260 *****
1270 ** Hilfsfunktionen **
1280 *****
1290 PRINT"mix,mx,miz,maz,cx,cy":PRINT m
    ix,mx,miz,maz,cx,cy
1300 END
1310 z=0.01*(y^2)+x+10
1320 IF x=0 THEN RETURN ELSE z=SIN(x)/(x
    -(x=0))
1330 z=x^2+y^2
    
```

Listing 2. 3D-Grafik mit Komfort

Die Skalierungsfaktoren berechnen sich ähnlich der zweidimensionalen Grafik. Die Berechnung finden Sie in den Zeilen 720 und 730. Das Programm enthält natürlich auch eine Routine zur automatischen Bestimmung der Extremwerte der eingegebenen Funktion (Zeile 540 und folgende). Diese können auch »von Hand« festgelegt werden. Die eigentliche Zeichenroutine liegt ab Zeile 750. Ein und dieselbe dreidimensionale Funktion kann eine Fülle verschiedener Figuren als grafisches Ergebnis haben, je nach Größe der Schrittweite in x- und y-Richtung. Das Programm beherrscht drei Darstellungsvarianten. Zum ersten ist es möglich, x in sehr kleinen Schritten zu variieren und dabei y sprungweise zu verändern. Dieses Verfahren bietet sich an, wenn kleine oder häufige (periodische) Änderungen des Funktionswertes bei Änderung von X auftreten (beispielsweise wenn in dem Graphen ein Winkelfunktionsterm [cos(x), sin(x) und so weiter] auftaucht). y läuft in diesem Fall in zehn gleichen Schritten von MINY bis MAXY. Die weite Abbildungsart ist die Umkehrung dieser Methode. Fein verändertes y bei einer groben Variation von x. In beiden Fällen ergibt sich eine Linienschar. Als dritter Fall bleibt dann noch die Kombination aus beiden Verfahren übrig. Es entsteht ein dreidimensionales Gittermodell. Beim Zeichnen der Linien wird dabei zunächst der äußerste linke, beziehungsweise bei laufendem y der oberste Punkt einer jeden Linie mit PLOT gesetzt, worauf dann mit DRAW einzelne Linien gezogen werden.

Automatische Grenzwertbestimmung

Bevor wir uns nun mit einigen weiteren Punkten beschäftigen, sollten Sie das Programm eintippen und laufen lassen. Als Funktion wählen Sie zuerst $z=x+y$ und lassen sich die Funktion im Bereich -10 bis 10 für X und Y zeichnen. Nach Wahl der automatischen Grenzwertbestimmung (bei Grenzwerte für z »n« eingeben) erhalten Sie das erste Diagramm. Auf der x- und y-Achse müssen nun die Extrema aufgetragen sein (-10 beziehungsweise 10) und die Z-Achse ist mit -20 und 20 beschriftet. Ob auch sonst alles stimmt, können Sie nach Exc mit »GOTO 1290« kontrollieren. Hier werden die wichtigsten Variablen

Funktion	XMIN	XMAX	YMIN	YMAX
$z=x*x-0.1*y*y*y$	-10	10	-10	10
$z=\text{SIN}(x)+\text{COS}(y)$	0	6	0	6
$z=\text{SIN}(x)*\text{COS}(x)$	0	6	0	6
$z=x*y$	-10	10	-10	10
$z=x*x+y*y$	-	10	-10	10
$z=(x-5)^3*y*\text{SIN}(y)$	-5	5-5	5	

Tabelle 1. Die Funktionen besitzen eine tolle Grafik

ausgegeben. Da wegen der Vielzahl von Parametern mit ähnlichen Bezeichnungen schon mal ein Eingabefehler vorkommt, kann hier die notwendigen Angaben zur Kontrolle:

MIX = -19.51
 MX = 19.51
 MIZ = -23.09
 MAZ = 23.09
 CX = 6.10659E-02
 CY = 0.144313

Nach dieser Probe können wir uns nun einige wirklich interessante Funktionen anschauen. Lassen Sie sich dabei ruhig ein wenig überraschen.

Wir wollen uns nun mit einigen Problemen befassen, die bei dieser Art der Darstellung auftreten. Nehmen wir einmal das Beispiel eines von hinten nach vorne gebogenen Sattels. Bei der Abbildung als Gittermodell durchdringen die hinteren Linien die vorderen. Bei einem einfachen Sattel ist es möglicherweise noch sinnvoll, weil die Biegung in der Linienführung die Grundstruktur der Figur noch betont. Wenn man jedoch auf der Y-Achse eine Sinus-Funktion aufträgt, ergeben sich mehrere voneinander liegende »Wellenberge«. Die klare Übersicht leidet und es ergibt sich ein ziemlicher Linienbrei. Für diese Problemfälle unter den Funktionen gibt es nun die Möglichkeit, die weiter hinten liegenden Funktionsteile auszublenden. Man erlebt die Figur dann nur noch so, als wenn sie undurchsichtig wäre. Die dazu notwendigen Voraussetzungen bietet unser Programm bereits. Wir müssen nur noch einige kleine Änderungen vornehmen. Das Prinzip ist dabei ganz einfach: Wenn wir laufende x-Werte darstellen, also y sprungweise variieren, so müssen wir die Änderung auf der Y-Achse — beginnend mit dem Maximum zum Minimum hin — vornehmen und dabei jeweils die Fläche unter der gerade dargestellten Kurve löschen. Bewegt sich die Funktion wie bei einer Sattelfunktion dann wieder nach oben, so werden die hinten liegenden Flächen

überschrieben. Um dies zu erreichen, müssen wir die Darstellungsroutine für laufende x ein wenig ändern. Zunächst muß die Schrittweite für sx verkleinert werden, damit auch jeder Punkt einzeln angesprochen wird. Der Divisor von 100 in Zeile 50 muß also herabgesetzt werden. Als nächstes müssen wir dann nur noch den ersten DRAW-Befehl in Zeile 980 durch PLOT ersetzen, durch Streichen des REM-« die nachfolgenden beiden Befehle aktivieren und schon erhalten wir diese neue Darstellung. Da durch die feinere Auflösung die Rechenzeit erheblich länger ist, benötigt diese Abbildung allerdings viel mehr Zeit.

Gittergrafiken mit laufendem x

Doch es gibt eine Reihe von Funktionen, wo diese Technik Vorteile bringt, beispielsweise bei $z=x*y$. Obwohl das Gittermuster, das man bei einer »laufenden x«-Darstellung erhält, durch die Änderungen teilweise gelöscht wird, wirkt die Figur dennoch natürlicher. Welche Darstellungsart Ihnen am meisten zusagt, ist natürlich Geschmackssache. Probieren Sie es einfach aus.

Sollten Ihnen die oben angegebene Funktionen zu »normal« und bekannt erscheinen, so probieren Sie einmal den folgenden Term aus: $z=1-\text{SIN}(Y)*\text{COS}(X)*X*Y$ jeweils im Bereich von -4 bis 4 . Möglicherweise werden Sie sich auch gefragt haben, wie denn die genaue Ansprache und Definition der Funktion vor sich geht. Das lästige Eintippen des Funktionsterms mit vorangestellter Zeilennummer entfällt ja im Gegensatz zu unserem 2D-Programm in der 3D-Routine. Die Antwort auf diese Frage finden Sie am Anfang des Programms. Hier nur zwei Tips, die weiterhelfen:

1. Was man nicht sieht, kann trotzdem passieren.
2. Eine einfache Abfrage muß nicht unbedingt eine einfache Abfrage sein.

(Carsten Straush)

Spitzen-Software für Schneider-Computer und Commodore 128 PC

BRANDNEU
Jetzt auch für den
Schneider Joyce

WordStar 3.0 mit MailMerge Der Bestseller unter den Textverarbeitungsprogrammen für PCs bietet Ihnen bildschirmorientierte Formatierung, deutschen Zeichensatz und DIN-Tastatur sowie integrierte Hilfstexte. Mit MailMerge können Sie Serienbriefe mit persönlicher Anrede an eine beliebige Anzahl von Adressen schreiben und auch die Adreßaufkleber drucken.

WordStar/MailMerge für den Schneider CPC 464*, CPC 664*

Bestell-Nr. MS 101 (3"-Diskette)

Bestell-Nr. MS 102 (5 1/4"-Diskette im VORTEX-Format)

WordStar/MailMerge für den Schneider CPC 6128

Bestell-Nr. MS 104 (3"-Diskette)

WordStar/MailMerge für den Schneider Joyce PCW 8256

Best-Nr. MS 105 (3"-Diskette)

Hardware-Anforderungen: Schneider CPC 464*, CPC 664*, CPC 6128 oder Joyce,

beliebiger Drucker mit Centronics-Schnittstelle

* Der Standard-Speicherplatz beim CPC 464/664 erlaubt ohne Speichererweiterung Blockverschiebe-Operationen nur bedingt und Simultan-Drucken gar nicht.

WordStar/MailMerge für den Commodore 128 PC

Bestell-Nr. MS 103 (5 1/4"-Diskette)

Hardware-Anforderungen: Commodore 128 PC, Diskettenlaufwerk, 80-Zeichen-Monitor,

beliebiger Commodore-Drucker oder ein Drucker mit Centronics-Schnittstelle

Und dazu die weiterführende Literatur:

WordStar für den Schneider CPC
Best-Nr. MT 779, ISBN 3-89090-180-8
WordStar für den Commodore 128 PC
Best-Nr. MT 780, ISBN 3-89090-181-6



dBASE II, Version 2.41 dBASE II, das meistverkaufte Programm unter den Datenbanksystemen, eröffnet Ihnen optimale Möglichkeiten der Daten- u. Dateihandhabung. Einfach u. schnell können Datenstrukturen definiert, benutzt und geändert werden. Der Datenzugriff erfolgt sequentiell oder nach frei wählbaren Kriterien, die integrierte Kommandosprache ermöglicht den Aufbau kompletter Anwendungen wie Finanzbuchhaltung, Lagerverwaltung, Betriebsabrechnung usw.

dBASE II für den Schneider CPC 464*, CPC 664*

Bestell-Nr. MS 301 (3"-Diskette)

Bestell-Nr. MS 302 (5 1/4"-Diskette im VORTEX-Format)

dBASE II für den Schneider CPC 6128

Bestell-Nr. MS 304 (3"-Diskette)

dBASE II für den Schneider Joyce PCW 8256

Best-Nr. MS 305 (3"-Diskette)

Hardware-Anforderungen: Schneider CPC 464*, CPC 664*, CPC 6128 oder Joyce,

beliebiger Drucker mit Centronics-Schnittstelle

* dBASE II für den Schneider CPC 464/664 ist lauffähig mit der VORTEX-Speichererweiterung auf 128 KByte. Diese erhalten Sie direkt bei der Firma VORTEX oder bei Ihrem Computerhändler.

dBASE II für den Commodore 128 PC

Bestell-Nr. MS 303 (5 1/4"-Diskette)

Hardware-Anforderungen: Commodore 128 PC, Diskettenlaufwerk, 80-Zeichen-Monitor,

beliebiger Commodore-Drucker oder ein Drucker mit Centronics-Schnittstelle



dBASE II für den Schneider CPC
Best-Nr. MT 837, ISBN 3-89090-188-3
dBASE II für den Commodore 128 PC
Best-Nr. MT 838, ISBN 3-89090-189-1



MULTIPLAN, Version 1.06 Wenn Sie die zeitraubende manuelle Verwaltung tabellarischer Aufstellungen mit Bleistift, Radiogummi und Rechenmaschine satt haben, dann ist MULTIPLAN, das System zur Bearbeitung »elektronischer Datenblätter«, genau das richtige für Sie! Das benutzerfreundliche und leistungsfähige Tabellenkalkulationsprogramm kann bei allen Analyse- und Planungsberechnungen eingesetzt werden wie z. B. Budgetplanungen, Produktkalkulationen, Personalkosten usw. Spezielle Formatierungs-, Aufbereitungs- und Druckerweisungen ermöglichen außerdem optimal aufbereitete Präsentationsunterlagen!

MULTIPLAN für den Schneider CPC 464*, CPC 664*

Bestell-Nr. MS 201 (3"-Diskette)

Bestell-Nr. MS 202 (5 1/4"-Diskette im VORTEX-Format)

MULTIPLAN für den Schneider CPC 6128

Bestell-Nr. MS 204 (3"-Diskette)

MULTIPLAN für den Schneider Joyce PCW 8256

Best-Nr. MS 205 (3"-Diskette)

Hardware-Anforderungen: Schneider CPC 464*, CPC 664*, CPC 6128 oder Joyce,

beliebiger Drucker mit Centronics-Schnittstelle

* MULTIPLAN für den Schneider CPC 464/664 ist lauffähig mit der VORTEX-Speichererweiterung auf 128 KByte.

MULTIPLAN für den Commodore 128 PC

Bestell-Nr. MS 203 (5 1/4"-Diskette)

Hardware-Anforderungen: Commodore 128 PC, Diskettenlaufwerk, 80-Zeichen-Monitor,

beliebiger Commodore-Drucker oder ein Drucker mit Centronics-Schnittstelle



MULTIPLAN für den Schneider CPC
Best-Nr. MT 835, ISBN 3-89090-186-7
MULTIPLAN für den Commodore 128 PC
Best-Nr. MT 836, ISBN 3-89090-187-5

Jedes Buch kostet DM 49,-
(sFr. 45,10/öS 382,20).
Erhältlich bei Ihrem Buchhändler.



Sie erhalten jedes **WordStar**, **dBASE II**- und **MULTIPLAN-Programm** für Ihren Schneider-Computer oder Commodore 128 PC fertig angepaßt (Bildschirmsteuerung und Druckerinstallation).

Jeweils Originalprodukte! Jedes Programmpaket enthält außerdem ein ausführliches Handbuch mit kompakter Befehlsübersicht. Die VORTEX-Speichererweiterung für den Schneider CPC 464 erhalten Sie direkt bei der Firma VORTEX oder bei Ihrem Computerhändler.

Diese **Markt & Technik-Softwareprodukte** erhalten Sie in den **Computer-Abteilungen der Kaufhäuser, bei Ihrem Computerhändler oder im Buchhandel.**

Wenn Sie direkt beim Verlag bestellen wollen: per Nachnahme oder gegen Vorauskasse durch Verrechnungsscheck oder mit der eingehafteten Zahlkarte.

Bestellungen im Ausland bitte an nebenstehende Adressen.

Für Auskünfte steht Ihnen Herr Teller, Telefon 089/46 13-205, gerne zur Verfügung.

WWW.HOMECOMPUTERWORLD.COM

Jedes Programm kostet DM 199,- (sFr. 178,-) * inkl. MwSt. Unverbindliche Preisempfehlung

Markt & Technik
**128er-/Schneider CPC
Software**

Hans-Pinsel-Straße 2, 8013 Haar bei München
Schweiz: Markt & Technik Vertriebs AG, Kollerstrasse 3, CH-6300 Zug, ☎ 042/41 56 56
Österreich: Microcomput-ique Schiller, Fasangasse 21, A-1030 Wien, ☎ 0222/78 56 61

3D auf dem CPC



Dreidimensionale Grafiken sind der Traum vieler Programmierer. Damit es nicht nur ein Traum bleibt, finden Sie hier die Grundlagen dazu.

Seit einiger Zeit geistern zwei Schlagworte durch die Computwelt: CAD und CAM. Der erste Begriff steht für computerunterstützte Entwicklung (Computer Aided Design), der zweite für computerunterstützte Produktion (Computer Aided Manufacturing). Die Entwicklung geschieht auf dem Bildschirm und eine Reihe nachgeschalteter Maschinen produziert nach den gezeichneten Angaben dann das reale Produkt. CAD und CAM sind jedoch nur zwei Anwendungsbereiche für den Einsatz von Computergrafiken. Für den Normalbenutzer viel interessanter ist beispielsweise ein anderes Gebiet: Die Darstellung von Funktionen und Relationen, speziell Funktionen, die von mehreren Variablen abhängig sind. Hierbei wird die Grafik, genauer die dreidimensionale Grafik, benutzt, um auch sehr komplexe Zusammenhänge anschaulich zu machen. Beginnen wir zuerst mit den Grundlagen der 3D-Programmierung, mit einigen Figuren im zweidimensionalen Raum. Den Schneider befähigen zwei Befehle, fast jede zweidimensionale Funktion abzubilden: PLOT und DRAW, wahlweise noch mit angehängtem R für die relative Koordinatengabe. Das erste Kommando zeichnet einen Punkt. Das zweite zieht eine Linie von der letzten Position des Grafik-Cursors zu dem neuen Zielpunkt. Einfache Figuren bestehen aus einem geschlossenen Linienzug. Um eine solche Figur zu malen, müssen wir also nur einen Anfangspunkt mit PLOT setzen und darauffolgend die Eckpunkte unserer Linien angeben. Beispielsweise liefert uns »CLS: PLOT 270,150: DRAW 370,150: DRAW 370,250: DRAW 270,250: DRAW 270,150« ein Rechteck in der Mitte des Bildschirms. Etwas komplizierter wird es, wenn wir Kreise oder Ellipsen zeichnen wollen. In jedem Fall ist hier der Kreis Ausgangsbasis. Gegebenenfalls wird er in eine Richtung verzerrt. Die Berechnung der Kreislinie geschieht dabei ent-

weder mit Hilfe von Winkelfunktionen oder nach dem Satz des Pythagoras. Die Kreislinie erscheint dann entweder als geschlossener Linienzug oder sie wird durch Einzelpunkte mit Hilfe von PLOT dargestellt. Bild 1 zeigt die grundsätzlichen Zusammenhänge verschiedener Kreispunkte. Die Hypothenuse des rechtwinkligen Dreiecks ist mit dem Radius identisch. $R \cdot \sin \alpha$ gibt uns den Y-Wert, $R \cdot \cos \alpha$ den entsprechenden X-Wert an. Der Y-Wert für gegebene X läßt sich auch mit Hilfe der Wurzel berechnen ($Y = \sqrt{R^2 - X^2}$).

Wenn Sie das eingezeichnete Dreieck an der X- und Y-Achse spiegeln, bekommen Sie vier spiegelbildliche Dreiecke. Die Zahlenwerte der Koordinaten sind immer gleich, nur die Vorzeichen ändern sich. Mit einer einzigen Berechnung lassen sich also gleichzeitig vier Punkte auf der Kreislinie bestimmen. Einzige Voraussetzung: Der Mittelpunkt des Koordinatensystems muß mit ORIGIN in das Kreissystem verlegt werden, damit auch die negativen Werte dargestellt werden können. Das Programm aus Listing 1 befähigt uns, nun einen Kreis als Linienzug mit Hilfe der Winkelfunktionen zu zeichnen.

In Abhängigkeit von s (je größer s,

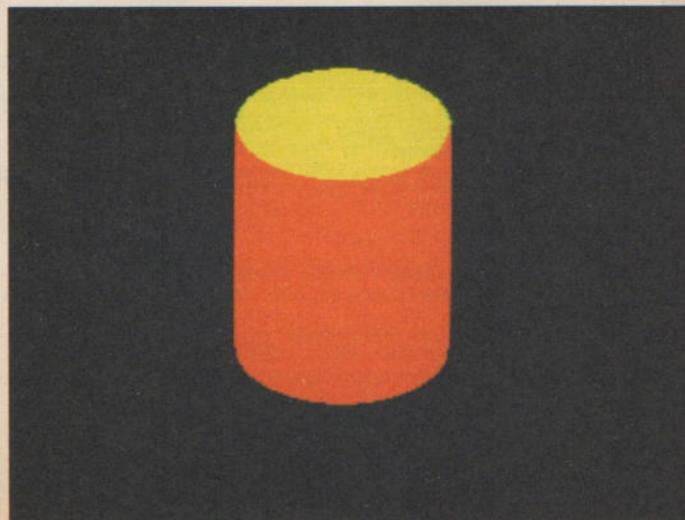
desto perfekter der Kreis) nähert sich das anfängliche Vieleck immer mehr einem Kreis an. Je nach gewünschter Auflösung vergrößert sich umgekehrt natürlich auch die Rechenzeit. Die höchste Auflösung erreicht man, wenn jeder Punkt einzeln angesprochen wird. In diesem Fall ist es dann natürlich sinnvoll, DRAW durch PLOT zu ersetzen. Mit 50 Punkten (das heißt 13 Berechnungen) wird der Kreis aber auch schon sehr genau.

Der Unterschied zwischen Kreis und Ellipse ergibt sich relativ einfach. Man multipliziert lediglich die X- beziehungsweise Y-Werte mit einem Verzerrungsfaktor. Listing 2 demonstriert diesen Effekt. Zunächst wird der Bildschirm gelöscht und der Mittelpunkt unseres Koordinatensystems in das Bildschirmzentrum verschoben.

In einer Schleife werden dann jeweils vier Punkte gleichzeitig gesetzt. Die Variable e enthält dabei den Verzerrungsparameter. Für e=1 ergibt sich ein Kreis, bei allen anderen Werten eine Ellipse. Wenn Sie mit diesem Parameter »spielen«, so resultieren daraus völlig verschiedene Ellipsenformen.

Mit der Verzerrung unseres Kreises ist nun auch schon der Schritt von der zweidimensionalen zur dreidimensionalen Grafik getan. Schauen wir uns zunächst einmal anhand von zwei Beispielen (einem Quader, Listing 3 und einem Zylinder, Listing 4) an, wie dreidimensionale Körper programmiert werden können.

Im Bereich der dreidimensionalen Grafik gibt es zwei grundsätzliche Darstellungstypen: das Gittermodell und die Vollgrafik. Bei einem Gittermodell werden nur die Begrenzungs- und Decklinien eines Körpers dargestellt; die dazwi-



3D-Grafiken auf dem Bildschirm sind nicht schwer

```

10 *****
20 ** Vielecke zeichnen **
30 *****
40 CLS
50 INPUT "Seitenzahl";s
60 ORIGIN 320,200
70 PLOT 150,0
80 FOR i=1 TO s
90 x=cos(2*PI*i/s)*150
100 y=sin(2*PI*i/s)*150
110 DRAW x,y
120 NEXT i
    
```

```

[55DC]
[345A]
[D5E0]
[D4CC]
[8982]
[A2B0]
[8578]
[6F3C]
[61BC]
[C718]
[593A]
[C9F6]
    
```

Listing 1. Aus einem Vieleck wird ein Kreis

```

10 *****
20 ** Ellipse **
30 *****
40 CLS:ORIGIN 320,200
50 INPUT "Laenge der Halbachse";a
60 INPUT "Exzentrik";e
70 CLS
80 FOR x=-a TO 0
90 y=e*SQR(a^2-x^2)
100 PLOT x,y:PLOT x,-y:PLOT -x,y:PLOT -x,-y
110 NEXT
    
```

```

[9994]
[08BE]
[3998]
[2BB6]
[366E]
[707C]
[EFD2]
[4DCA]
[9A76]
[9F06]
[55E2]
    
```

Listing 2. Aus einem Kreis wird eine Ellipse

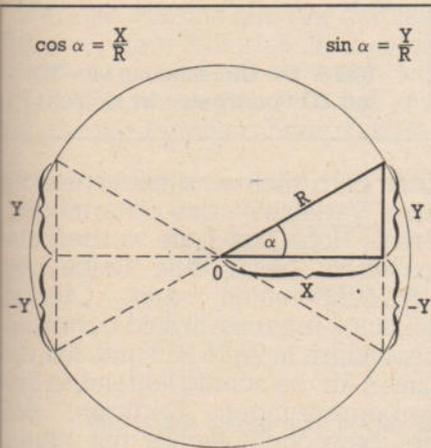


Bild 1. Vier Dreiecke sind spiegelgleich

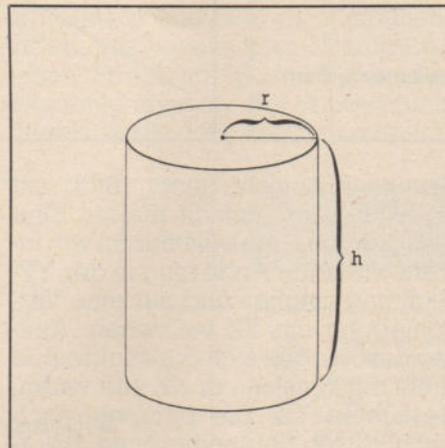


Bild 3. Der Zylinder in der Ebene

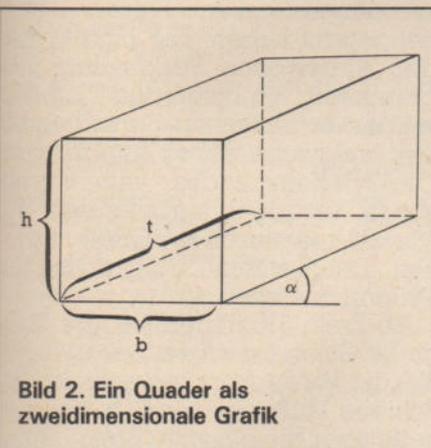


Bild 2. Ein Quader als zweidimensionale Grafik

Das Hauptproblem einer jeden dreidimensionalen Grafik ist die perspektivische Darstellung der räumlichen Zusammenhänge. Der Winkel, unter dem die nach hinten laufenden Linien erscheinen, ist hierbei der ausschlaggebende Aspekt. Ist er ungünstig gewählt, so erscheint die ganze Figur künstlich und unwirklich. Die Begründung hierfür ist einfach.

Das menschliche Auge liefert ein zweidimensionales Abbild unserer räumlichen Umwelt. Die dritte Dimension nehmen wir nur indirekt wahr, nämlich als Projektion in den zweidimensionalen Raum. Die Tiefenempfindlichkeit, also die Wahrnehmung der dritten Dimension, läuft über einen Denkvorgang als Schlußfolgerung ab. Das beste Beispiel liefert eine Eisenbahnschiene oder eine sehr lange gerade Straße. Sobald zwei parallel laufende Begrenzungslinien sich einander nähern, so interpretiert das Auge dies als eine Eisenbahnschiene oder Straße, die sich immer weiter vom Betrachter entfernt und tief in den Raum hineinreicht. Daß dieses Verfahren nicht ganz fehlerlos ist, belegen die zahlreichen sogenannten optischen Täuschungen. Uns interessiert jedoch nur die Art und Weise, wie das menschliche Auge seine Umwelt wahrnimmt.

Wollen wir einen dreidimensionalen Eindruck erreichen, so müssen wir also eine zweidimensionale Abbildung erzeugen, in die die dritte Dimension hineinprojiziert wird. Dazu eignen sich mehrere Verfahren. Wir beschäftigen uns zunächst mit der relativ einfachen Parallelprojektion. Um einen dreidimensionalen Körper als zweidimensionale Projektion abzubilden, sind zwei Schritte notwendig. Zunächst einmal muß der Körper in einem geeigneten dreidimensionalen Koordinatensystem erfaßt werden, bevor die dabei festgestellten Koordinaten in eine zweidimensionale Darstellung transformiert werden.

Basis jeder Transformation ist zunächst einmal ein dreidimensionales Koordinatensystem, wie Sie es bei der Würfeldarstellung in Bild 4 vorfinden. Der Würfel ist durch die acht Punkte A, B, C, D, E, F, G und H im dreidimensionalen System definiert. Wenn (wie in Bild 4) die linke untere Ecke unseres Körpers (A) im Schnittpunkt der drei Achsen X, Y und Z liegt, und jede Kante unseres Würfels die Länge 1 hat, lassen sich die Eckpunkte der Figur leicht angeben. D hätte zum Beispiel die Koordinaten (1,0,0); E (1,1,0) und G (1,1,1).

Bei einer Parallelprojektion legen wir nun über dieses dreidimensionale Koordinatensystem ein zweidimensionales Achsenkreuz mit den Achsenbezeichnungen xz beziehungsweise yz. Die neue Achse xz kommt dabei genau auf die dreidimensionale Koordinatenachse x und yz genau auf z zu liegen. Der Sinn dieser, auf den ersten Blick etwas verwirrenden Operation, ist relativ einfach einzusehen. Es muß nur eine Transformation des (dreidimensionalen) Y-Wertes erfolgen. Die X- und Z-Koordinaten unseres dreidimensionalen Körpers können wir einfach übernehmen.

In Bild 5 sehen Sie, wie der Y-Wert in seine X- und Z-Komponente zerlegt wird. Die Koordinaten des darzustellenden Punktes (P) sind dabei wesentlich vom Betrachtungswinkel abhängig. Das ist der Winkel (alpha), un-

schenliegenden Flächen des Modells bleiben frei. Allerdings sind auch die eigentlich durch den Körper verdeckten Eckkanten sichtbar. Im anderen Fall, bei der Vollgrafik, wird das Modell ausgemalt. Es sind nur Flächen zu sehen, die man auch bei der Betrachtung eines realen Körpers sehen würde. Die einzelnen Seiten sind dabei meist mit unterschiedlicher Farbe hervorgehoben, um den räumlichen Aspekt noch zu vertiefen. Wenden wir uns den beiden »Versuchsobjekten« Quader und Zylinder zu.

Wie die beiden Varianten als Figuren aussehen, zeigen die Bilder 2 und 3. In beiden Fällen wurde dabei das Gittermodell gewählt.

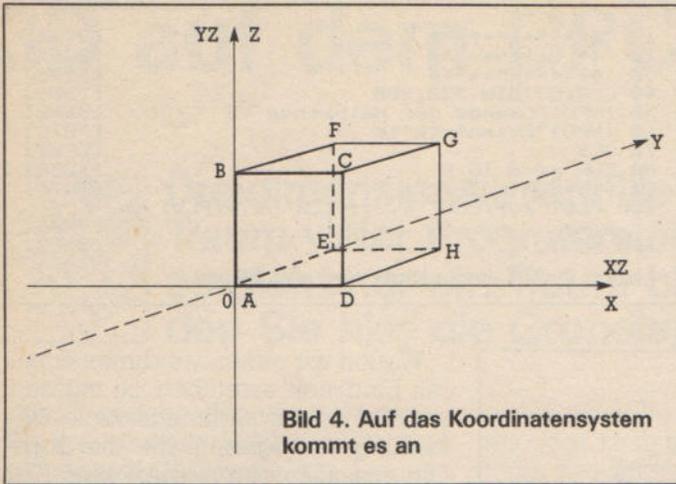


Bild 4. Auf das Koordinatensystem kommt es an

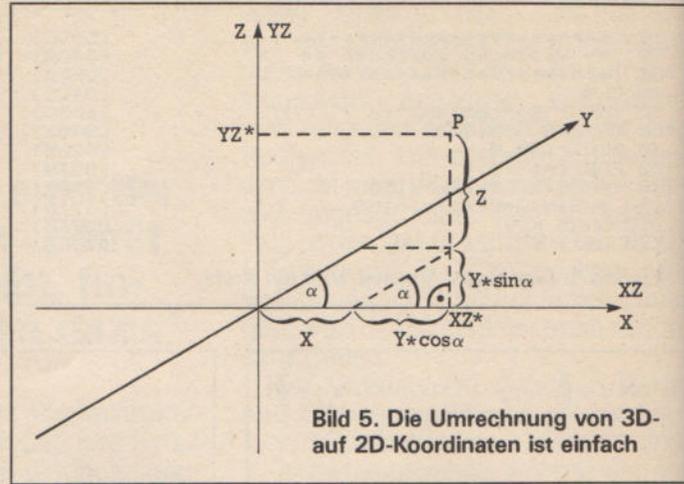


Bild 5. Die Umrechnung von 3D- auf 2D-Koordinaten ist einfach

ter dem X- und Y-Achse in der Darstellung aufeinandertreffen. Bei dem kleinen Dreieck rechts unten handelt es sich um ein rechtwinkliges Dreieck, so daß wir mit den Winkelfunktionen arbeiten dürfen. Die Hypotenuse ist als Y-Wert gegeben, d, der Winkel zwischen X- und Y-Achse, steht ebenfalls fest. Wir erhalten damit die folgenden XZ- beziehungsweise YZ-Werte für unseren Punkt: $XZ = X + Y \cdot \cos \alpha$ und $YZ = Z + Y \cdot \sin \alpha$.

Im ersten Quadranten führt ein Y-Wert größer als Null dazu, daß der darzustellende Punkt gegenüber $Y = 0$ nach rechts wandert. Zu den »normalen« zweidimensionalen Strecken X und Z werden jeweils die Winkelfunktionsterme addiert.

Ein häufig benutzter Spezialfall ergibt sich, wenn wir die Y-Achse unter einem Winkel von 45 Grad zeichnen. In diesem Fall ist $\sin \alpha = \cos \alpha = 0,5$, womit sich die Gleichungen von oben zu $XZ = X + 0,5 \cdot Y$ und $YZ = Z + 0,5 \cdot Y$ vereinfachen. Auch für andere Winkel empfiehlt es sich, die Sinus- beziehungsweise Cosinus-Werte vor der eigentlichen Koordinatenumrechnung einmalig zu bestimmen. Diese Werte werden danach in zwei Variablen (beispielsweise »si« und »co«) zwischengespeichert. Dann können Sie problemlos mit der unteren Formel weiterarbeiten, wobei statt der 0,5 in der oberen Gleichung »co« und in der unteren »si« einzusetzen ist. Das Rechnen mit einer Konstanten geht bedeutend schneller als mit den Winkelfunktionen.

Nun wollen wir unser Wissen für unseren Quader benutzen. Beginnen wir mit der einfachsten Variante, dem Gittermodell.

Zur Abbildung genügen hier simple DRAW-Kommandos unterschiedlicher Länge und Richtung. In unserem Beispiel wollen wir mit einem Winkel von 45 Grad arbeiten.

Damit läßt sich unser Bild mit DRAWR sehr einfach malen. Eine waagrechte Linie bekommen wir, indem wir keine Änderung in der YZ-Richtung machen und nur eine Verschiebung um XZ vollziehen. Analog dazu, ergibt sich die senkrechte Linie mit konstantem XZ und veränderlichem YZ. Die perspektivisch nach hinten laufende Linie (in Y-Richtung) benötigt ein gleichförmiges Ändern von XZ und YZ, um den gewünschten Effekt zu erzielen. Nun bleibt nur noch die Frage offen, mit welcher Länge die Linien gezogen werden sollen. Hier ist auf eine Unzulänglichkeit des Auges Rücksicht zu nehmen. Wir empfinden nämlich nach »hinten« laufende Linien als deutlich länger, als Begrenzungen, die parallel zur Betrachtungsfläche laufen. Während wir also die Frontlinien ohne Korrekturen zeichnen können, müssen wir bei den nach hinten laufenden Seitenlinien eine Längenkorrektur vornehmen. Ein Faktor, der die Längenangabe relativ treffend verkürzt, ist $1/\sqrt{2}$.

Für Besitzer des CPC 464 gibt es einige Probleme, wenn die nun erhaltenen Flächen ausgefüllt werden sollen. Mit einem 664 oder 6128 ist es natürlich keine Schwierigkeit. Sie wenden einfach das FILL-Kommando an. Ansonsten müssen wir uns hier mit einer ganzen Reihe nebeneinandergezogener Linien behelfen. In einer Schleife ist das kein Problem. Man muß nur über die gesamte Breite Linie neben Linie zeichnen. Gleiches gilt für die Deckfläche und die eine (rechte) Seite. Mit diesen Grundgedanken arbeitet das nebenstehende Programm »Quader« (Listing 3).

Am Anfang der Routine finden Sie die Auswahl nach dem Darstellungsmodus. Unser Quader kann wahlweise als Gittermodell oder als Vollgrafik gezeichnet werden. Ab

Zeile 160 treffen wir zunächst auf die erste Variante — das Gittermodell. Breite, Höhe und Tiefe werden eingegeben, bevor eine Reihe von DRAWR-Befehlen den Quader zeichnet. Interessant sind dabei insbesondere in Zeile 230 und 250 die Linien für die schräg laufenden Seitenbegrenzungen. Dadurch, daß die Bewegung jeweils mit identischer Verschiebung in X- und Y-Richtung erfolgt, ergibt sich der schräge Verlauf. Ansonsten werden dieselben Techniken angewandt, die wir schon bei den einfachen zweidimensionalen Figuren kennengelernt haben. Die Linien wurden dabei ohne Verkürzung programmiert, um Ihnen den Tiefeneffekt zu verdeutlichen. Sie sollten für die praktische Arbeit nun in allen DRAWR-Kommandos mit identischen Verschiebungsangaben die Variable »t« durch »t/SQR(2)« ersetzen. Dann stimmen die optischen Proportionen wieder.

Ab Zeile 310 finden wir die Vollgrafik. Zunächst wird in Zeile 370 bis 390 die Vorderfront gezeichnet. Die Schleife läuft hier von 0 bis b, das heißt über die gesamte Breite unserer Vorderfront in 1er-Schritten. Dabei wird jeweils der Fußpunkt mit MOVE gesetzt, bevor dann mit DRAWR eine Linie in Farbe 2 (rot) gezogen wird. Die Zeilen 410 bis 430 bilden die Seitenfläche ab. Die Berechnung ist hier etwas schwieriger, der Effekt ist allerdings derselbe. Den Fußpunkt der unteren Begrenzungslinie erhalten wir, indem wir zu dem Punkt (b, 0) jeweils den Vektor (i, i) addieren und dabei i über die gesamte Tiefe von 0 bis t laufen lassen. Als Korrektur ist t dann wieder durch »t/SQR(2)« zu ersetzen. Von diesem Fußpunkt werden nun wieder Linien der Höhe h mit der Farbe 3 (hellblau) nach oben gezogen. Die Deckplatte unseres Quaders wird dann mit Zeile 440 bis 460 gezeich-

net. Die Flächenlinien werden jetzt von links nach rechts gezeichnet. Dazu wird der Grafik-Cursor auf der äußeren, linken Seitenlinie gesetzt und dann die Fläche mit DRAWR in Farbe 1 ausgefüllt.

Den Abschluß unseres Programms bildet ein sogenanntes »ewiges« GOTO. Diese Abfrageschleife wurde eingesetzt, um zu verhindern, daß die erstellte Grafik durch ein »Ready« gestört wird.

Wenden wir uns nun der zweiten Figur zu, dem Zylinder. In diesem Programm (Listing 4) benutzen wir eine Reihe von geometrischen Figuren, die wir bereits kennen. Die Re-

de ist hier von der Ellipse, die zumindest in unserem Gittermodell recht schnell und einfach die obere beziehungsweise untere Grundplatte bildet. Zwei weitere mit DRAW gezeichnete Linien vervollständigen das Bild. Etwas schwieriger wird es dagegen, wenn wir eine Vollgrafik verlangen. Hier muß zwar nur noch eine Ellipse dargestellt werden, nämlich die obere Deckplatte, dafür ist diese aber vollständig auszufüllen. Dies ist nun aber nicht möglich, indem wir jeweils vom Mittelpunkt einen Strahl auf einen Punkt der Randlinie senden. Bei Anwendung dieses Verfahrens bleiben nämlich

einige kleine Punkte in der Deckplatte ohne Farbbelegung.

Wir wollen daher ein anderes Verfahren anwenden. Wenn wir davon ausgehen, daß das Zentrum unserer Deckplatte den Mittelpunkt eines Koordinatensystems bildet (vergleiche Bild 6), so können wir eine gute Flächendeckung erreichen, indem wir von jedem Punkt unseres Umfangs eine Senkrechte auf die Y-Achse ziehen. Diese »übereinandergestapelten« Linien decken dann die Fläche vollständig ab.

Am Anfang des Programms finden sich die schon gewohnten Abfragen nach der Darstellungsart.

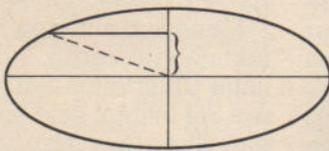


Bild 6. Die Ellipse wird in vier Teilen ausgefüllt

```

10 ***** [A034]
20 *** Quader zeichnen ** [69CE]
30 ***** [8038]
40 MODE 1 [11F4]
50 INK 0,0:INK 1,21:INK 2,6:INK 3,11 [90B8]
60 BORDER 0 [B228]
70 PAPER 0:PEN 1 [047A]
80 CLS [F8D4]
90 PEN 3:PRINT "{5 SPACE}Q U A D E R {3 SP [E6C0]
   ACE}Z E I C H N E N"
100 LOCATE 12,4:PRINT "Gittermodell {2 SPA [740A]
   CE} (1)"
110 LOCATE 12,6:PRINT "Vollgrafik {4 SPACE [C4DC]
   } (2)"
120 z$=INKEY$:IF z$<"1" AND z$<"2" THE [FE6C]
   N 120 ELSE IF z$="2" THEN 300
130 ***** [A49E]
140 *** Gittermodell ** [D260]
150 ***** [96A2]
160 CLS:ORIGIN 50,50 [E562]
170 INPUT "Breite":b [DA08]
180 INPUT "Hoehe":h [6532]
190 INPUT "Tiefe":t [FB54]
200 CLS [8528]
210 DRAWR b,0,1:DRAWR 0,h:DRAWR -b,0:DRA [B538]
   WR 0,-h [A336]
220 MOVER b,0 [1C28]
230 DRAWR t,t:DRAWR 0,h:DRAWR -b,0:DRAWR [693A]
   -t,-t [ABD6]
240 MOVER b,0 [751A]
250 DRAWR t,t [C65A]
260 MOVE 0,0:DRAWR t,t:DRAWR b,0:MOVER - [0A02]
   b,0:DRAWR 0,h [FDB6]
270 GOTO 470 [CFF4]
280 ***** [8E2C]
290 *** Vollgrafik ** [7102]
300 ***** [E52C]
310 CLS [4F4E]
320 INPUT "Breite":b [A92C]
330 INPUT "Hoehe":h [7736]
340 INPUT "Tiefe":t [C6B8]
350 ORIGIN 50,50 [EEAE]
360 CLS [68FC]
370 FOR i=0 TO b [48F8]
380 MOVE i,0 [A3D2]
390 DRAWR 0,h,2 [682A]
400 NEXT i [36FE]
410 FOR i=0 TO t [BDD8]
420 MOVE b+i,i:DRAWR 0,h,3 [D6E2]
430 NEXT i [5AF2]
440 FOR i=0 TO t [AAEE]
450 MOVE 0+i,h+i:DRAWR b,0,1
460 NEXT
470 IF INKEY$="" THEN 470 ELSE RUN

```

Listing 3. Der Einstieg in die Welt der 3D-Grafiken

```

10 ***** [55DC]
20 *** Zylinder zeichnen ** [A6AC]
30 ***** [D5E0]
40 MODE 1 [11F4]
50 INK 0,0:INK 1,21:INK 2,6:INK 3,11 [90B8]
60 BORDER 0 [B228]
70 PAPER 0:PEN 1 [047A]
80 CLS [F8D4]
90 PEN 3:PRINT "{3 SPACE}Z Y L I N D E R { [29DA]
   3 SPACE}Z E I C H N E N"
100 LOCATE 12,4:PRINT "Gittermodell {2 SPA [740A]
   CE} (1)"
110 LOCATE 12,6:PRINT "Vollgrafik {4 SPACE [C4DC]
   } (2)"
120 z$=INKEY$:IF z$<"1" AND z$<"2" THE [0470]
   N 120 ELSE IF z$="2" THEN 320 [A47E]
130 ***** [D260]
140 *** Gittermodell ** [96A2]
150 ***** [541E]
160 CLS:ORIGIN 320,300 [8720]
170 INPUT "Radius":a [6532]
180 INPUT "Hoehe":h [8638]
190 CLS [0E1E]
200 FOR x=-a TO 0 [0526]
210 y=0.5*SQR(a^2-x^2)
220 PLOT x,y,1:PLOT x,-y:PLOT -x,y:PLOT [5386]
   -x,-y:PLOT x,y-h:PLOT x,-y-h:PLOT -x [6A48]
   ,y-h:PLOT -x,-y-h [111A]
230 PLOT x+1,y,1:PLOT x+1,-y:PLOT -x+1,y [B99E]
   :PLOT -x+1,-y:PLOT x+1,y-h:PLOT x+1, [5B10]
   -y-h:PLOT -x+1,y-h:PLOT -x+1,-y-h [F6EE]
240 NEXT x [1A60]
250 MOVE -a,-h:DRAW -a,0 [715E]
260 MOVE -a+1,-h:DRAW -a+1,0 [CFF4]
270 MOVE a,-h:DRAW a,0 [DDAB]
280 MOVE a+1,-h:DRAW a+1,0 [2DF8]
290 GOTO 560 [8030]
300 ***** [C61E]
310 *** Vollgrafik ** [8D30]
320 ***** [83EA]
330 CLS [8438]
340 INPUT "Radius":a [3FEE]
350 INPUT "Hoehe":h [7912]
360 ORIGIN 320,300 [E22C]
370 CLS [5B2A]
380 x=-a:s=0.1 [E3C8]
390 x=x-s [766A]
400 x=x+s:IF x>=0 THEN 480 [EAC6]
410 y=0.5*SQR(a^2-x^2) [7F22]
420 MOVE 0,y:DRAW x,y,1 [6F4A]
430 MOVE 0,y:DRAW -x,y [9500]
440 MOVE 0,-y:DRAW x,-y [D61A]
450 MOVE 0,-y:DRAW -x,-y [453A]
460 s=(1+10/a)*s [062A]
470 GOTO 400 [CDAE]
480 ORIGIN 320,300-h [3BF2]
490 FOR x=-a TO 0 [D00C]
500 y=0.5*SQR(a^2-x^2) [6C3A]
510 MOVE x,-y [1222]
520 DRAWR 0,h,2 [D8EE]
530 MOVE -x,-y
540 DRAWR 0,h
550 NEXT x
560 IF INKEY$="" THEN 560 ELSE RUN

```

Listing 4. Ein Zylinder in zwei Modellen

Das Gittermodell ist relativ schnell beschrieben. Die Ellipsengleichung und ihre Darstellung kennen wir ja schon. Es werden nun nur zusätzlich zu den vier normalerweise in einer Ellipse gesetzten Punkten noch gleichzeitig vier weitere Punkte für die Bodenplatte gesetzt (einfach durch Subtrahieren der Höhe h). Etwas außergewöhnlich ist dabei Zeile 230. Sie wiederholt nämlich denselben Vorgang noch einmal, jedoch mit gegenüber den in Zeile 220 befindlichen um 1 erhöhten X-Werten. Dies hat einen relativ einfachen Effekt. Unsere Ellipse wird nämlich noch einmal mit einer Verschiebung um 1 in X-Richtung gezeichnet. Die Umfanglinie wird doppelt so dick. Dasselbe Verfahren wenden wir dann bei den senkrechten Seitenlinien an. Sie werden jeweils mit der notwendigen Verschiebung um den Radius A rechts und links doppelt nebeneinander gezeichnet.

Etwas interessanter wird es, wenn wir unseren Zylinder als Vollgrafik zeichnen. Am Anfang steht wieder die Eingabe der zwei benötigten Parameter für Radius und Höhe. Der Mittelpunkt unseres Koordinatensystems wird auf das Bildschirmzentrum gesetzt (Zeile 360). Dann beginnen die eigentlichen Darstellungszeilen. Wir beginnen dabei am äußersten linken Rand unserer Figur. X weist den Wert $-a$ auf — und y dementsprechend 0. Der Schrittparameter s wird in Zeile 380 auf 0,1 gesetzt. Er legt fest, in welchen X-Schritten die Berechnung erfolgt. Die eigentliche Schleife läuft von Zeile 400 bis Zeile 470. Als erstes wird x jeweils um den Schrittparameter s erhöht. Beim ersten Durchlauf wird allerdings durch das vorherige Abziehen in Zeile 390 die Erhöhung wieder korrigiert. Wir operieren zu Anfang also mit $x = -a$.

Als nächstes wird abgefragt, ob x größer oder gleich 0 ist. Dies wird als Ende-Kriterium benutzt. In solch einem Fall haben wir nämlich den Mittelpunkt unserer Figur erreicht. Da wir alle vier Viertel unserer

Deckplatte parallel zeichnen lassen, ist in diesem Fall die Figur vollendet. Beim eigentlichen Zeichnen unserer Ellipsenviertel bestimmen wir zunächst in Zeile 410 den Y-Wert mit Hilfe des Satzes des Pythagoras. Als Wert für die Exzentrizität unserer Ellipse wurde 0,5 eingesetzt. Es folgen die eigentlichen DRAW-Zeilen. Dabei wird der Grafik-Cursor immer mit $X=0$ auf den gerade errechneten Y-Wert gesetzt. Dann erfolgt das Zeichnen der Linien bis zum Punkt auf der Randlinie. Zeile 460 beinhaltet eine etwas trickreiche Veränderung des Schrittparameters s . Vielleicht ist Ihnen beim »Spielen« mit dem ersten Ellipsenprogramm (Listing 2) auch schon aufgefallen, daß der dort verwendete Algorithmus nicht optimal ist. Der Grund dafür ist einfach. Wenn man vom äußersten linken Rand der Figur ausgeht, so bewirken bereits kleine Änderungen von X eine ziemlich große Änderung des Y-Wertes. Je mehr man dagegen zum Mittelpunkt kommt, in desto größeren Schritten könnte man eigentlich X variieren, da sich der Abstand von der X-Achse, das heißt der Wert für Y , nur noch geringfügig erhöht.

Neben diesen grundsätzlich für jeden Kreis oder jede Ellipse geltenden Problem ist allerdings noch eine andere Tatsache zu beachten. Wir müssen nämlich X auch im Verhältnis zur Größe unserer Figur variieren. All diesen Gesetzmäßigkeiten trägt die schrittweise Erhöhung von S in Zeile 460 Rechnung. Mit jedem Schritt, den wir uns dem Zentrum unserer Figur nähern, erfolgt die Änderung in immer größeren Sprüngen. S wird Schritt für Schritt durch laufende Multiplikation immer weiter vergrößert. Der Faktor, mit dem S bei jedem Schritt multipliziert wird, ist jedoch nicht fix vorgegeben, sondern hängt wiederum vom Radius ab. Je größer der Radius ist, desto kleiner wird dieser Erhöhungsfaktor gewählt. Damit setzt der Übergang zu einer größeren Schrittweite erst später ein.

Der Rest zeichnet die senkrechte Vorderfront. Der Fußpunkt unseres Systems wird mit ORIGIN um die Höhe h verschoben. Der Rand der unteren Deckplatte wird wieder einmal mit der Ellipsengleichung bestimmt. Von dieser ausgehend, ziehen wir dann — gleichzeitig von rechts und links auf das Zentrum zustrebend — Linien der Höhe h nach oben. Diese treffen damit genau auf die Unterkante der oberen Deckplatte. Zur Verstärkung des räumlichen Effekts werden hierbei für Deckplatte und Zylinderumfang unterschiedliche Farben benutzt. Eine sehr schöne Zylinderausgabe erhalten Sie zum Beispiel, wenn Sie für den Radius den Wert 75 und für die Höhe 150 oder 200 eingeben.

Man kann nun den räumlichen Effekt noch steigern, indem man farbliche Unterteilungen macht. Probieren Sie einmal eine einfache Dreiteilung aus. Dazu werden die Zeilen 520 und 540 wie folgt geändert:
 520 DRAWR 0,h/3,2
 540 DRAWR 0,h/3,2
 mit den zwei neuen Zeilen
 525 DRAWR 0,2*h/3,3
 545 DRAWR 0,2*h/3,3

Das untere Drittel der Säule bleibt weiterhin rot. Darüber hat sich jedoch nun ein hellblauer Halbzylinder aufgebaut. Diese Technik kann sehr gut benutzt werden, um dreidimensionale Diagramme darzustellen. Man ist hierbei natürlich nicht auf zwei Farben oder zwei Flächen beschränkt. Es ist auch problemlos möglich, mit mehreren Unterteilungen und (speziell im MODE 0) auch mit mehreren Farben zu arbeiten.

Ein letztes wichtiges Kapitel ist die Parallelprojektion. Sie muß allerdings als ein Verfahren betrachtet werden, das zwar dem Computer entgegenkommt, aber sich der Betrachtungsweise des Auges nicht ausreichend annähert. Der Hauptgrund liegt darin, daß bei dieser Projektionstechnik gleichgroße Flächen die in der Tiefe voreinander gelagert sind auch mit identischer

Fortsetzung auf Seite 42

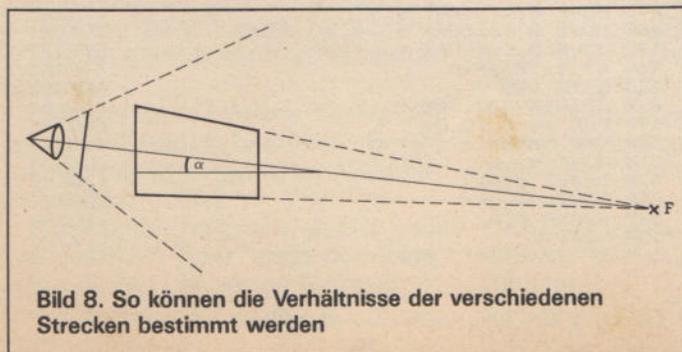


Bild 8. So können die Verhältnisse der verschiedenen Strecken bestimmt werden

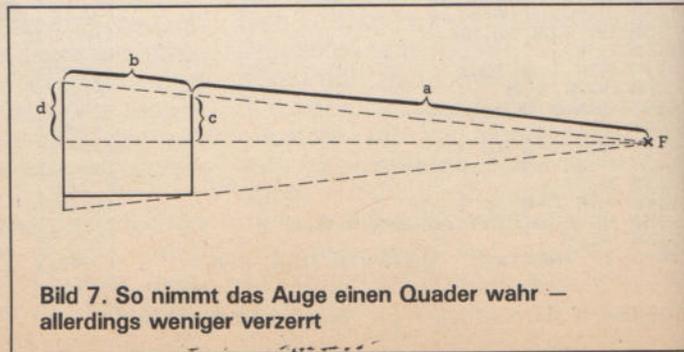
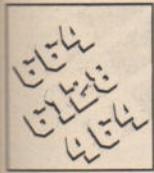


Bild 7. So nimmt das Auge einen Quader wahr — allerdings weniger verzerrt

Schnelle Kreise

Wichtiger als Programmkünste in Maschinensprache ist oft das Wissen um einen guten Algorithmus.

Auch Kreisroutinen in Basic können sehr schnell sein. Man muß nur wissen wie.



Eine sehr umständliche — und damit auch zeitaufwendige — Methode, Kreise auf den Bildschirm zu bringen, finden Sie in Listing 1. Jeder Punkt wird zuerst einzeln berechnet und dann mit PLOT gesetzt. Gerade aber das Bestimmen der Sinus- und Cosinuswerte dauert lange. Deshalb sollten Sie mal das Programm aus Listing 2 ausprobieren. Sie werden verblüfft sein, wie schnell jetzt der Kreis gezeichnet wird.

Die erste Routine benötigt auf dem Schneider volle 13 Sekunden, während es mit der zweiten in nur 1,1 Sekunden zu schaffen ist. Das wäre also 12mal schneller und sieht außerdem noch wesentlich besser aus. Bei 100 Kreisen beträgt dann die Rechenzeit statt 22 Minuten nur 2 Minuten. Eine recht eindrucksvolle Zeitersparnis.

Wer nun wissen möchte, wie man solche Routinen selbst entwirft, für den ist dieser Artikel gedacht. Leser, die keinen Schneider besitzen, können mit Hilfe der Erklärungen aus Tabelle 1 die Grafikbefehle für ihren eigenen Computer umsetzen.

Geben Sie zuerst Listing 3 ein und starten Sie das Programm, das gemäß Ihrer Angabe ein regelmäßiges n-Eck auf den Bildschirm zeichnet. Probieren Sie einmal verschiedene Werte für n aus. Sie werden bemerken, daß ab einer bestimmten Eckenanzahl (beim Radius 200 ungefähr ab n=45) die Kurve sich optisch nicht mehr von einem Kreis unterscheidet. Statt unnötig viele Kreispunkte (wie in Listing 1) zu bestimmen, braucht man also nur ein regelmäßiges 45-Eck zu zeichnen,

$$\tan(d/2) = \text{Gegenkathete/Ankathete} = \sqrt{r^2 - (r-d)^2} / (r-d)$$

Für sehr kleine Werte von d gilt (in Bogenmaß): $\tan d$ ungefähr gleich d. Also ist $d/2 = \sqrt{r^2 - (r-d)^2} / (r-d)$ oder umgerechnet

$$d/2 = \sqrt{r^2 / (r-d)^2 - 1}$$

Da d ja gegenüber r sehr klein ist ($d = 0,5$; Radius r aber bis zu 200), gilt ungefähr

$$d/2 = \sqrt{r-1}$$

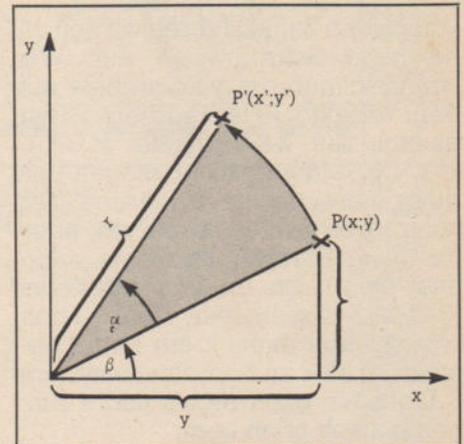


Bild 3. Konventionell berechnet muß man die Sonderfälle beachten

Da aber wie gesagt r wesentlich größer als 1 ist, fällt der Abzug der Zahl 1 vom Radius gar nicht mehr ins Gewicht.

Als endgültiges Ergebnis für den maximalen Winkel d, bei dem sich ein Vieleck nicht von einem Kreis unterscheidet, erhalten wir den Wert $d = 2 \sqrt{r}$. Die zugehörige Eckenanzahl ist dabei $n = \text{Vollkreis}/d$. Da die Rechnung wie gesagt nur im Bogenmaß gilt, ergibt sich für die kritische Eckenanzahl der Wert

$$n = 2\pi / 2\sqrt{r} = \pi / \sqrt{r}$$

oder gerundet

$$N = \text{int}(\pi * \sqrt{r} + 1)$$

Für einen Radius von 199 ist das Resultat aus dieser Rechnung genau der auch experimentell gefundene Wert 45. Allein die Anwendung dieses Tricks bewirkt eine Erhöhung der Rechengeschwindigkeit um den Faktor 7. Ähnliche Betrachtungen kann man natürlich für alle möglichen Anwendungen anstellen, um die Geschwindigkeit zu optimieren. Wem das zu kompliziert ist, der braucht sich einfach nur die Formel

$$N = \text{int}(\pi \sqrt{r} + 1) \text{ merken.}$$

Die zweite Verbesserung des Programms wurde dadurch möglich, daß zur Herleitung der Formeln

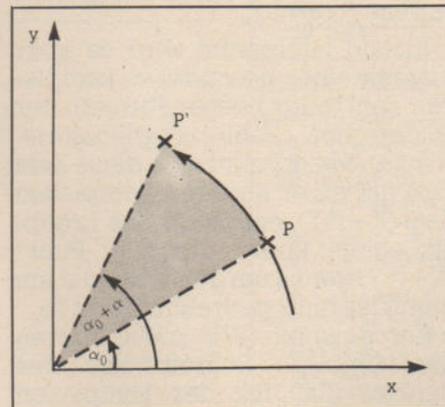


Bild 2. Mit komplexen Zahlen ist eine Drehung um α sehr einfach

um ein und denselben Effekt auf dem Bildschirm zu erzielen. Man berechnet einzig sparsame 44 Punkte anstatt der 360, verbindet diese durch Geraden und erhält seinen »Kreis«.

Wie läßt sich nun, ohne lange zu probieren, die minimale Eckenanzahl bestimmen? Betrachten wir dazu das Bild 1. Um das Vieleck nicht mehr von einer Kreislinie unterscheiden zu können, darf die Entfernung von einer Kante des Vielecks zur »idealen Kreislinie« maximal einen halben Bildschirmpunkt betragen. Für das schraffierte (rechtwinkelige) Dreieck gilt dann: Die drei Seitenlängen betragen r, r-d und (nach dem Satz des Pythagoras)

$$\sqrt{r^2 - (r-d)^2}$$

Für den Winkel d gilt:

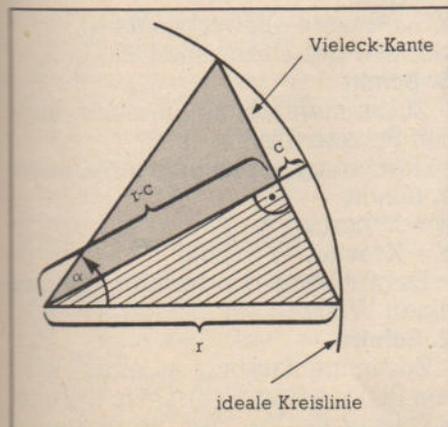


Bild 1. Die ideale Kreislinie wird durch ein Vieleck angenähert

Dabei ist es unumgänglich, alle Größenangaben für die jeweilige Bildschirmauflösung umzuschreiben. Dies ist aber durch Änderung einer einzigen Programmzeile immer sehr leicht möglich.

komplexe Zahlen zu Hilfe gezogen wurden. Dies ist nicht nur in diesem Beispiel sehr nützlich, sondern es vereinfacht auch in vielen anderen Grafikanwendungen zusätzlich eine Erhöhung der Rechengeschwindigkeit. Deshalb hier eine kleine Einführung in das Rechnen mit komplexen Zahlen.

Möchte man die Lage eines Punktes P eindeutig bestimmen, so gibt man normalerweise seine x- und seine y-Koordinate an (beispielsweise: $P=(x;y)$). Bei der Darstellung mit komplexen Zahlen führen wir nun eine neue Schreibweise ein: Wir kennzeichnen die y-Koordinate mit dem Vorfaktor »i« und addieren dazu einfach die x-Koordinate: $P:=X+1*Y$. Die Mathematiker mögen uns diese vereinfachte Erklärung der komplexen Zahlen verzeihen, aber für unser Problem reicht sie völlig aus. Steht also ab jetzt irgendwo »3+i*4«, wissen wir, daß damit schlicht und einfach ein Punkt gemeint ist, der im Koordinatensystem 3 Einheiten nach rechts und 4 Einheiten nach oben liegt.

Was ist eine komplexe Zahl?

Das Zeichen »i« ist dabei eine willkürlich gewählte Bezeichnung, die in der realen Zahlenwelt keine Entsprechung hat. Man kann also beispielsweise nicht etwa sagen, »i hat den Wert 27« und überall, wo in einer Formel das Zeichen i vorkommt, statt dessen 27 einsetzen. Dies funktioniert nur mit gewöhnlichen Variablen.

Wenn man hingegen die Fläche eines Quadrats mit 3 m Kantenlänge berechnen möchte, schreibt man einfach »3m*3m=9m²« und spricht das Ergebnis als »Quadratmeter« aus. Wenn dieser Vorgang nicht schon durch jahrelange Übung ins Unterbewußtsein übergegangen wäre, würde man sich dabei denken »3m*3m=3*m*3*m=3*3*m*m=9*m²«. Mit Einheiten läßt sich genauso rechnen, als ob es Platzhalter für später einzusetzende Zahlen wären — nur daß man diese Einsetzung nicht willkürlich durchführen darf.

Ebenso verhält es sich mit dem Zeichen »i«. Quadriert gibt es als Ergebnis -1 ($i^2 = -1$). Für den Anfang mag diese Art zu Rechnen noch ungewohnt erscheinen, mit der Zeit wird es aber genauso geläufig wie der Ausdruck »Meter hoch 2« oder »Quadratmeter« als Ergebnis von zwei miteinander multiplizierten Längen.

Mathematisch korrekt bezeichnet man »i« als »imaginäre Einheit«, und ein Ausdruck »A+i*B« wird als komplexe Zahl bezeichnet.

Rechnen mit komplexen Zahlen

Jetzt stellt sich natürlich die Frage, was nützt diese neue Schreibweise? Stellen Sie sich vor, Sie haben einen Punkt an der Stelle (X;Y) und wollen wissen, wo er sich befindet, wenn Sie ihn um 3 Einheiten nach rechts und um 4 Einheiten nach oben verschieben. Mit komplexen Zahlen ist die Berechnung sehr einfach: Man addiert die komplexe Schreibweise des Punktes (X;Y) zur komplexen Schreibweise des Punktes (3;4).

$$\begin{aligned} (X+i*Y)+(3+i*4) \\ = X+i*Y+3+i*4 \\ = X+3+i*Y+i*4 \\ = (X+3)+i*(Y+4) \end{aligned}$$

Der neue Punkt liegt also an der Position (X+3;Y+4).

In einem Programm erscheint dann nur noch das Endergebnis »PLOT X+3,Y+4«.

Richtig interessant wird es aber erst mit einer besonderen komplexen Zahl (oder besser Formel), der »Eulerschen Zahl«: $E(\alpha) = \cos(\alpha) + i*\sin(\alpha)$. Multipliziert man diese Zahl $E(\alpha)$ mit einer anderen komplexen Zahl $(X+i*Y)$, entspricht das Ergebnis einem Punkt, der zum Punkt $(X+i*Y)$ genau um den Winkel α um den Ursprung gedreht ist (Bild 2).

Ein Beispiel: Wir multiplizieren die komplexe Schreibweise des Punktes (2;2) mit der komplexen Zahl $E(45)$. (45 in normalen Winkelgraden, das heißt ein Vollkreis hat 360 Grad.) Dann gilt: $(2+i*2)*E(45) = (2+i*2)*(\cos(45)+i*\sin(45)) = 2*\cos(45)+i*2*\sin(45)+i*2*\cos(45)+i^2*2*\sin(45)$

Mit $i^2 = -1$ (Definition von i) kann man unsere Formel umschreiben: » $i^2*2*\sin(45)$ « wird durch » $(-1)*2*\sin(45)$ « oder » $-2*\sin(45)$ « ersetzt: $2*\cos(45)+i*2*\sin(45)+1*2*\cos(45)-2*\sin(45) = (2*\cos(45)-2*\sin(45))+ix(2*\sin(45)+2*\cos(45))$

Das Ganze ausgerechnet gibt als Ergebnis:

$$0+i*2\sqrt{2} \approx 0+i*2,82.$$

Vollziehen wir das Ganze in einer »konventionellen« Rechnung nach: Der Ausgangspunkt (2;2) liegt auf der Winkelhalbierenden des 1. Quadranten und damit in einem Winkel von 45 Grad zur x-Achse. Eine Drehung um 45 Grad versetzt ihn genau auf die y-Achse. Am Anfang war er nach dem Satz von Pythago-

ras $\sqrt{2^2+2^2} = \sqrt{8} = 2\sqrt{2}$ Einheiten vom Ursprung entfernt, was natürlich auch für die Endposition gilt. Als Koordinaten der Endposition des Punktes erhält man also genau denselben Wert, den wir auch durch komplexe Rechnung ermittelt haben (0;2 $\sqrt{2}$). Das ist kein Zufall, sondern gilt für jeden beliebigen Punkt. In diesem speziellen Beispiel bringt die komplexe Rechnung keine großen Vorteile.

Das liegt daran, daß unser Beispiel sehr einfach ist. Für beliebige Punkte erspart die komplexe Rechnung sehr viel Aufwand. Vergleichen wir einmal beide Wege, um den Punkt (X;Y) um den Winkel α zu drehen:

a) allgemeine »konventionelle« Rechnung: (Bild 3)

1. Schritt

Der Punkt $P=(X;Y)$ schließt mit der x-Achse den Winkel $\beta = \arctan(y/x)$ ein.

2. Schritt

Beachtung von Sonderfällen: — Der Arcustangens von (y/x) liefert denselben Wert wie der Arcustangens von $(-y/x)$, obwohl der Punkt -x, -y ganz woanders liegt. Ist die x-Koordinate also negativ, muß man zu β noch 180 Grad addieren.

— Ist die x-Koordinate 0, ist der Arcustangens nicht definiert. Die Fälle $\beta = 90$ Grad und $\beta = 270$ Grad müssen also gesondert betrachtet werden.

3. Schritt

Der Punkt $P=(X;Y)$ hat vom Ursprung den Abstand $r = \sqrt{x^2+y^2}$ (Satz von Pythagoras).

4. Schritt

Jetzt die eigentliche Rotation für $P'=(X';Y')$

$$X' = r*\cos(\beta + \alpha); Y' = r*\sin(\beta + \alpha),$$

wobei α der Drehwinkel ist.

5. Schritt

Zusammenfassung in einer kurzen Routine (Listing 4).

b) Rechnung mit komplexen Zahlen

1. Schritt

$$X' = X*\cos(\alpha) - Y*\sin(\alpha)$$

$$Y' = X*\sin(\alpha) + Y*\cos(\alpha)$$

Es gibt keine Sonderfälle und nur einen Winkel.

2. Schritt

Zusammenfassung in einer kurzen Routine (Listing 5). Da ja die alte x-Koordinate bei der Berechnung der y-Koordinate noch einmal benötigt wird, muß zur Zwischenspeicherung allerdings eine Hilfsvariable T verwendet werden.

Selbst wenn die Erklärungen zur komplexen Rechnung nicht interessiert haben, der sollte zumindest die eben genannten Formeln und ihre Anwendung im Kopf behalten.

```

10 'Kreisschlecht (abschreckendes Beispiel) [9ACE]
20 DEG'Winkelmodus 360 Grad pro Vollkreis [965C]
30 r=199;mx=320;my=200'Radius, Ursprung [BBEC]
40 CLS [D4CC]
50 FOR i=0 TO 360 [4BBC]
60 PLOT mx+r*SIN(i),my+r*COS(i) [F45A]
70 NEXT [B98C]
    
```

Listing 1. So sollte man Kreise nicht PLOTen

```

10 'Kreisgut (daf)r aber komplizierter) [D1D8]
20 DEG'Winkelmodus 360 Grad pro Vollkreis [965C]
30 r=199;mx=320;my=200'Radius, Ursprung [BBEC]
40 n=INT(PI*SQR(r)+1):sn=SIN(360/n):cs=C [0622]
   OS(360/n)'Konstanten [EED6]
45 CLS [EED6]
50 x=r;y=0:MOVE mx+r,my'Startwerte [F99A]
60 FOR i%=1 TO n [19B4]
70 t=cs*x-sn*y:y=sn*x+cs*y:x=t [5EB6]
80 DRAW mx+x,my+y [302B]
90 NEXT [C590]
    
```

Listing 2. Kreise schnell gezeichnet

```

10 'Eckexperiment [C6F8]
20 DEG [A7A4]
30 r=199;mx=320;my=200 [436A]
40 INPUT"Eckenzahl";e [C71A]
50 w=360/e [D5CC]
60 CLS [E2D0]
70 PLOT mx+r,my [3BF4]
80 FOR i=w TO 361 STEP w [F638]
90 DRAW mx+r*COS(i),my+r*SIN(i) [F63E]
100 NEXT [71E0]
110 GOTO 40 [95DE]
    
```

Listing 3. Ein Experiment mit Ecken

```

10 'Lageberechnung (konventionell) [A376]
20 'nicht zur Nachahmung empfohlen [7C70]
30 DEG [C2A6]
40 INPUT"Ausgangslage-x-Koordinate";x [7F7E]
50 INPUT"Ausgangslage-y-Koordinate";y [3184]
60 INPUT"Drehwinkel";w [9632]
70 IF x=0 AND y<=0 THEN b=270 [51EE]
80 IF x=0 AND y>0 THEN b=90 [1E1A]
90 IF x>0 THEN b=ATN(y/x) [2900]
100 IF x<0 THEN b=ATN(y/x)+180 [87D4]
110 r=SQR(x*x+y*y) [0E12]
120 x=r*COS(w+b):y=r*SIN(w+b) [4A12]
130 PRINT"Ergebnispunkt:"x","y [4402]
140 GOTO 40 [F8E4]
    
```

Listing 4. Die Lageberechnung dauert Zeit...

```

10 'Lageberechnung (komplex) [9166]
20 'So geht's besser [20E2]
30 DEG [C2A6]
40 INPUT"Ausgangslage-x-Koordinate";x [7F7E]
50 INPUT"Ausgangslage-y-Koordinate";y [3184]
60 INPUT"Drehwinkel";w [9632]
70 t=x*COS(w)-y*SIN(w):y=x*SIN(w)+y*COS [2956]
   (w):x=t [4402]
130 PRINT"Ergebnispunkt:"x","y [F8E4]
140 GOTO 40
    
```

Listing 5. ...oder auch nicht

```

10 'Rotation (komplex) [4392]
20 DEG:ORIGIN 320,200 [B88E]
30 'Daten der Figur einlesen [DD02]
40 n%=9 [381A]
50 DIM x(n%),y(n%) [4AC8]
60 FOR i%=1 TO n%:READ x(i%),y(i%):NEXT [8692]
70 DATA -50,-50,-50,50,0,75,50,50,50,-50 [F002]
   ,-50,-50,50,50,-50,50,50,-50 [5C64]
80 'Rotieren und Zeichnen [E14E]
90 w=15:sn=SIN(w):cs=COS(w) [7126]
100 CLS [80EC]
110 PLOT x(n%),y(n%) [B858]
120 FOR i%=1 TO n% [9CBA]
130 DRAW x(i%),y(i%) [BAD4]
140 'Rotation eines Punktes [8F6C]
150 t=x(i%)*cs-y(i%)*sn:y(i%)=x(i%)*sn+y [2450]
   (i%)*cs:x(i%)=t [6198]
160 NEXT [391E]
170 GOTO 100 [E31A]
    
```

Listing 6. Komplexe Rotation ist schnell

```

10 'Rotation(konventionell) [DFA2]
20 DEG:ORIGIN 320,200 [B88E]
30 'Daten der Figur einlesen [DD02]
40 n%=9 [381A]
50 DIM x(n%),y(n%) [4AC8]
60 FOR i%=1 TO n%:READ x(i%),y(i%):NEXT [8692]
70 DATA -50,-50,-50,50,0,75,50,50,50,-50 [F002]
   ,-50,-50,50,50,-50,50,50,-50 [5C64]
80 'Rotieren und Zeichnen [4346]
90 w=15 [7126]
100 CLS [80EC]
110 PLOT x(n%),y(n%) [B858]
120 FOR i%=1 TO n% [9CBA]
130 DRAW x(i%),y(i%) [BAD4]
140 'Rotation eines Punktes [8F6C]
150 IF x(i%)=0 AND y(i%)>=0 THEN b=90 [2450]
151 IF x(i%)=0 AND y(i%)<0 THEN b=270 [6198]
152 IF x(i%)>0 THEN b=ATN(y(i%)/x(i%)) [391E]
153 IF x(i%)<0 THEN b=ATN(y(i%)/x(i%))+1 [E31A]
   80 [A79E]
154 r=SQR(y(i%)*y(i%)+x(i%)*x(i%)) [75EC]
155 x(i%)=r*COS(b+w):y(i%)=r*SIN(b+w) [EE44]
160 NEXT
170 GOTO 100
    
```

Listing 7. Rotation: Einfach zu verstehen, aber langsam

```

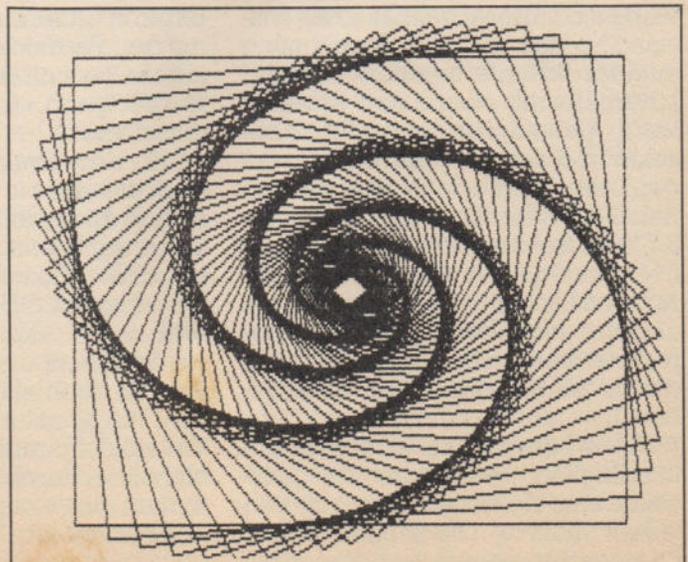
10 'Schachtel-Quadrate [8436]
20 DEG:ORIGIN 320,200 [B88E]
30 'Parameter: Radius,Winkel,Faktor,Anza [F680]
   hl [79DC]
40 r=199:w=5:f=0.92:n=37 [45DE]
50 cs=f*COS(w):sn=f*SIN(w) [521E]
60 x=r:y=r [EFD2]
70 CLS [F7B8]
80 FOR i%=1 TO n [E50A]
90 PLOT x,y [8F92]
100 DRAW -y,x [BFEE]
110 DRAW -x,-y [AC96]
120 DRAW y,-x [393E]
130 DRAW x,y [6E12]
140 t=x*cs-y*sn:y=x*sn+y*cs:x=t [CDEA]
150 NEXT
    
```

Listing 8. Mit einer Drehstreckung steigen wir ein

```

10 'Kreis-Experimente [2FB2]
20 DEG:ORIGIN 320,200 [B88E]
30 PRINT"Durch Komma getrennt Radius,Sch [04DC]
   rittweite,Faktor und Anzahl eingeben" [2DB2]
40 INPUT "Eingabe:";r,w,f,n [45DE]
50 cs=f*COS(w):sn=f*SIN(w) [E2D0]
60 CLS [E708]
70 x=r:y=0:PLOT x,y [F7B8]
80 FOR i%=1 TO n [8EBA]
90 t=x*cs-y*sn:y=x*sn+y*cs:x=t [3338]
100 DRAW x,y [55E2]
110 NEXT
120 LOCATE 1,24:PRINT"Radius"r,"Schritt [DF24]
   weite" w,"Faktor" f,"Anzahl" n [72EE]
130 PRINT"Weiter? - eine Taste druecken" [D40E]
140 q$=INKEY$:IF q$="" THEN GOTO 140 [C8E4]
150 GOTO 30
    
```

Listing 9. Experimente mit dem »komplexen« Kreis



Listing 9 erzeugt diese und andere Grafiken

Radius	Winkel	Faktor	Anzahl
1	10	1,01	600
100	144	1	5
150	144	0,95	100
100	11,25	1	32
150	90	0,97	100

Tabelle 1. Eingabevorschlage fur Listing 9

PLOT x,y:	Zeichnet den Punkt (x,y) und setzt den Grafikkursor an die Stelle (x,y)
MOVE x,y:	Setzt den Grafikkursor an die angegebene Position
DRAW x,y:	Zieht eine Linie von der Grafikkursorposition an die Position (x,y) und setzt den Cursor an diese Position.
ORIGIN x0,y0:	Setzt den Koordinatenursprung an die angegebene Position. Gibt es diesen Befehl nicht, mu bei jedem PLOT, MOVE und DRAW-Befehl x0 zur x-Koordinate und y0 zur y-Koordinate addiert werden
LOCATE x,y:	Setzt den Textcursor an die angegebene Position

Tabelle 2. Die Grafik-Befehle des Schneiders im ublick

Eine Anwendung zu den gewonnenen Erkenntnissen sieht man in Listing 2: Die neuen Formeln gelten ja fur jeden beliebigen Punkt in der Ebene. Speziell im Kreis braucht man also nicht immer den Ausgangspunkt in die Gleichung einzusetzen, sondern kann genauso gut den jeweils letzten berechneten Punkt verwenden. Durch diesen Trick erhalt man einen weiteren Vorteil: Die Dehnung pro Schritt ist ja konstant, und genau diese Schrittweite mussen wir in die Gleichung einsetzen. Die Sinus- und Cosinuswerte sind also im ganzen Kreis konstant, so da man die Berechnung aus der Schleife herausziehen und innerhalb mit Konstanten arbeiten kann. Allein durch diesen Trick beschleunigt sich die Verarbeitung gegenuber dem »abschreckenden Beispiel« in Listing 1 um den Faktor 2. Dabei werden die neuen Routinen noch gar nicht richtig ausgenutzt. Auch im Listing 1 mussen der Radius und der aktuelle Winkel nicht berechnet werden, sondern werden ja durch die Schleife schon vorgegeben.

Diese Berechnung wird aber unumganglich, wenn man beispielsweise eine beliebige Figur rotieren lassen mochte. Die Punkte liegen dann nicht mehr schon geordnet vor, sondern konnen irgendwo auf

dem Bildschirm verteilt sein. In diesem Fall mu bei konventioneller Rechnung der Radius und der Arcustangens wirklich berechnet werden. Hier zeigen die neuen Routinen richtig, was sie konnen. Obwohl sich das Listing 6 vom Listing 7 nur in den funf Zeilen mit den aus konventioneller/komplexer Rechnung erhaltenen Gleichungen unterscheidet, ist das Programm aus Listing 6 dreimal schneller als das aus Listing 7.

Multiplikation mit einer Konstanten

Was in unserer Betrachtung noch fehlt, ist die Multiplikation einer komplexen Zahl mit einer Konstanten. Der Winkel zwischen dem Punkt und der x-Achse bleibt dabei fest, der Punkt entfernt sich nur etwas vom Ursprung. Eingefugt in die Kreisroutine wurde sich also eine Spirale ergeben. Statt

$A*(X+i*Y)*(cos(\alpha)+i*sin(\alpha))$ kann man dann naturlich genauso gut $(X+i*Y)*A*(cos(\alpha)+i*sin(\alpha))$ schreiben oder gleich $(X+i*Y)*(A*cos(\alpha)+i*A*sin(\alpha))$.

Die Multiplikation wird also gleich in die Sinus- und Cosinusberechnung mit einbezogen und so aus der Schleife entfernt. Die Eulersche Formel entartet hier zu einer Drehstreckung. Von dieser Moglichkeit macht das kleine Demoprogramm in Listing 8 Gebrauch, das ineinandergeschachtelte Quadrate zeigt. Auerdem werden hier nicht alle vier Punkte des Quadrats berechnet, sondern jeweils nur einer, wobei sich die anderen durch Punkt- und Achsenspiegelung ergeben. Experimentieren Sie doch einmal mit den Parametern in der Zeile 40, indem Sie zum Beispiel »r=199:w=5:f=0,97:n=100« einsetzen.

Fur den Praktiker zum Schlu noch die »alles in einem«-Routine (Listing 9). Hier wahlen Sie die Parameter »Anzahl der Schritte«, »Schrittweite«, »Streckfaktor« und »Anfangsradius« frei aus. Obwohl es dabei nicht sehr viele Variationen zu geben scheint, kann die Routine ohne anderung doch alles Mogliche zeichnen: Von Kreisen uber regelmaige Vielecke, Spiralen, Sterne, sich aufblahende Sterne und so weiter. Versuchen Sie es einfach einmal mit den Parametern aus Tabelle 1.

(Helmut Tischer)

Fortsetzung von Seite 38

Lange gezeichnet werden. Der Mensch nimmt jedoch weiter entfernt liegende Gegenstande kleiner wahr. In weiter Ferne schrumpfen alle Korper bei dieser Betrachtungsweise auf Null. Wir haben die Grenzen der Auflosungsfahigkeit des Auges erreicht. Der Punkt, auf den sich das Auge dabei konzentriert, wird als Fluchtpunkt bezeichnet. Bild 7 zeigt, wie sich dieser Zusammenhang in »uberspitzter« Form fur unseren Quader darstellt. Denn normalerweise ist das Verhaltnis von Abstand zum Fluchtpunkt zur Korperseitenlange gunstiger, so da die Flachen weniger asymmetrisch verlaufen. Trotzdem wirkt sich dieser Zusammenhang immer aus. Wenn Sie sich den Quader etwas naher betrachten, so haben auch Sie wahrscheinlich das Gefuhl, da die Hinterflachen langer sind als die vorderen. Hier sollte man also auf Abhilfe sinnen. Um eine korrekte, das heit dem Auge entsprechende Abbildung zu erhalten, mute man allerdings auf die Zentralprojektion zuruckgreifen.

Zunachst verzichten wir auf den asymmetrischen Verlauf der Seitenlinien und beschranken uns auf eine symmetrische Korrektur (Bild 8). Wir mussen dazu nur die Winkel unserer Seitenlinien etwas mehr auf den Fluchtpunkt orientieren. Statt zweier Linien, die um 45 Grad nach oben laufen, sollte also die obere geringfugig abgesenkt werden, die untere etwas starker angehoben verlaufen. Die hintere Seitenwand wird dadurch etwas kleiner. Die Oberkante der Ruckwand unseres Quaders ist ebenso zu verkurzen und damit werden die Seitenlinien nun auch in dieser Richtung etwas auf den Zentralstrahl hingezogen. Die optisch exakten Werte konnen Sie dabei – speziell bei einfachen Korpern – am besten durch Ausprobieren bestimmen. Wem dieses Verfahren nicht liegt, der kann sich naturlich auch zu einer mathematisch exakten Form durchringen. Die Bedingungen der Strahlensatze sind erfullt. Somit gilt: $a/c=(a+b)/d$. Nun mussen Sie nur noch die Entfernung zum Fluchtpunkt festlegen (a+b) und naturlich die Korperlange b. Sie konnen dann bei gegebenen d den Wert von c problemlos bestimmen. Diese Berechnung mu naturlich in beiden Achsenrichtungen erfolgen, wobei man bei einem Wurfel (einem Spezialfall) naturlich immer mit demselben Korrekturfaktor arbeiten kann.

(Carsten Straush)

Wenn der Recorder streikt

Wie ein Ungetüm aus grauer Computersteinzeit residiert der Datarecorder im Bauch des CPC 464. Wie er funktioniert und was man tun kann, wenn er nicht einwandfrei funktioniert, das lesen Sie hier.



Der CPC 64 benutzt — wie eine Reihe anderer Heimcomputer auch — einen umgebauten Kassettenrecorder als Datenspeicher. Schließlich sind diese in Massenproduktion hergestellten Geräte sehr billig. Da sie aber ursprünglich nicht für die Datenspeicherung entwickelt wurden, ergeben sich bei der Anwendung öfters Probleme.

Ein scheinbarer Fehler äußert sich darin, daß nach der Aufforderung die PLAY-Taste zu drücken nichts passiert. Häufig liegt der Fehler bei Ihnen. Wenn Sie die PLAY-Taste schon vorab gedrückt haben, betätigen Sie nach »LOAD "Programmname"« schnell zweimal hintereinander die ENTER-Taste, da die PLAY-Taste ja gedrückt ist.

Nun ist das Speichern von Daten beim Schneider aber eine relativ komplizierte Angelegenheit. Es läuft in zwei Stufen ab. Zuerst wird an der Speicherobergrenze 4 KByte Platz reserviert. Dieser reicht, um eine Ausgabe- und eine Eingabedatei gleichzeitig offen zu halten. Die Daten werden nämlich in Blöcken zu 2 KByte auf Band geschrieben. Um diesen Bereich anzulegen, muß nach dem Ladebefehl der Speicher umorganisiert werden. Dies kann — speziell, wenn der Speicher mit größeren Stringarrays gefüllt ist — einige Zeit dauern. Während dieser Zeit macht der Computer nichts anderes, auch die Tastatur wird nicht abgefragt. Er bekommt das zweite Enter nicht mit. Mit Ungeduld kommt man hier nicht weiter. Die einzige Möglichkeit, dies zu vermeiden, ist ein vorangestelltes Ausrufezeichen »!« im Ladebefehl. Bei »LOAD "!"Programmname"« unterdrückt der Schneider nämlich alle Kommentare und damit auch die Aufforderung zur Quittierung.

Ein anderes Problem tritt meist erst nach einiger Zeit auf — klem-

mende Kassetten. Es gibt wohl kaum eine Anwendung für Kassetten, wo diese so oft hin und her gespult werden, wie bei der Computerei. Eine Datei wird geSAVEt und mit »CAT« geprüft. Dann wird eine andere Stelle auf dem Band gesucht, um ein anderes Programm zu laden. Diese »Bandschieberei« führt dazu, daß die einzelnen Windungen auf der Spule gegeneinander leicht verschoben werden, während bei kontinuierlich in einer Richtung laufenden Bändern Umdrehung für Umdrehung ordentlich eine Schicht auf die nächste gespult wird. Der Effekt ist, daß die Kassette klemmt. Oft reicht aber schon das einfache Herausziehen und Wiedereinschieben der Kassette, um das Band wieder zum Laufen zu bringen. In besonders hartnäckigen Fällen muß man die Kassette einmal kurz mit der Unterkante auf den Tisch schlagen. Dadurch werden die Bandwicklungen enger zusammengeschoben und das Band läuft wieder besser.

Schwieriger wird es, wenn wir es nicht mit solch einfachen Fehlern, sondern mit wirklichen Lese- beziehungsweise Schreibfehlern zu tun haben. Beispielsweise wenn das Band zwar läuft, aber durch Fehler beim Schreiben oder Lesen permanent »Read-Error« auftreten. Dies kann wiederum verschiedene Gründe haben. Zum einen setzen sich im Laufe der Zeit auf dem Tonkopf und auch auf dem Band kleine Staubteilchen fest. Diese stören den Kontakt zwischen Band und Tonkopf. Dadurch nimmt die Lautstärke und auch die Deutlichkeit der als Töne gespeicherten Daten ab. Man kann dies sogar hören. Einfach den Lautstärkereglervoll aufdrehen, die Datenmusik klingt ein wenig dumpf.

Im Handel erhältliche Reinigungsbänder für Hifi-Kassettenrecorder oder etwas Alkohol helfen hier. Um

den Tonkopf zu reinigen, taucht man ein Wattestäbchen in Alkohol und säubert damit die Oberfläche. Dabei darf das Wattestäbchen nicht fusseln. Denn dies würde gerade dem Zweck der Aktion zuwiderlaufen. Und noch eins muß beachtet werden. Sie dürfen den Tonkopf nie mit einem metallischen Gegenstand berühren.

Ein zweiter Grund für solche Schreib-/Lesefehler kann darin bestehen, daß der Tonkopf nicht mehr richtig justiert ist. Dies ist ziemlich häufig der Fall und rührt daher, daß der Tonkopf nur durch eine einzige Schraube auf Position gehalten wird. Diese verdreht sich im Laufe der Zeit, wodurch der Tonkopf abgesenkt oder angehoben wird. Das Fachwort dafür ist »Azimuth-Fehler«. Der Spalt des Tonkopfs steht dann nicht ganz senkrecht zum Magnetfeld (Bild 1). Man kann die Tonkopfstellung natürlich korrigieren. Dazu benötigen Sie einen Uhrmacherschraubenzieher Nummer 2 oder Nummer 3. Diesen erhalten Sie in jedem Elektronikfachgeschäft und auch mancher Werkzeughändler führt ihn. Es handelt sich dabei um einen relativ kleinen Schraubenzieher, der mit dem Schaft durch den Schlitz vor dem Kassettenfach des Datarecorders passen muß.

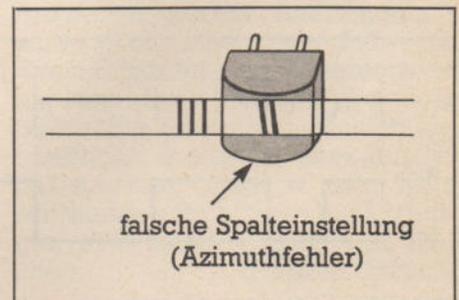


Bild 1. Der Tonkopf muß senkrecht zum Magnetfeld stehen

Entfernen Sie nun zuerst die Kassette aus dem Laufwerk und lassen Sie im folgenden den Deckel geöffnet. Wenn man die PLAY-Taste drückt, schiebt sich aus dem Bereich der Tastatur die Mechanik mit dem Tonkopf nach vorne. In der Mitte sehen Sie einen silbernen Würfel, den Tonkopf. Links daneben befindet sich ein anderer weißlicher kleiner Würfel. Das ist der Löschkopf. Auf der linken Seite des Tonkopfes (silbriger Würfel) befindet sich eine kleine Schraube. Sie können diese durch den Schlitz in der Abdeckung des Kassettenrecorders auch bei laufendem Gerät erreichen. Zunächst drehen Sie diese Schraube ganz hinein, das heißt im Uhrzeiger-

sinn bis zum Anschlag. Nun legen Sie ein fertig gekauftes bespieltes Band ein und spulen so lange vor, bis gerade die ersten Töne eines Programms ertönen. Der Lautstärkenregler an der rechten Seite des Gehäuses sollte dabei auf Maximum stehen. Während diese Töne hörbar sind, drehen Sie die Schraube solange gegen den Uhrzeigersinn wieder heraus, bis die Lautstärke ein Maximum erreicht. Normalerweise sind das zwei bis drei volle Umdrehungen. Nachdem Sie die Qualität Ihrer Einstellung mit einigen Kassetten überprüft haben (möglichst die Problemfälle testen), können Sie die Schraube mit Nagellack oder ähnlichem gegen Verdrehen sichern.

Eine andere Fehlerquelle können sogenannte »Drop outs« sein. Drop outs sind Stellen in der Magnetspur des Bandes, die sich nicht magnetisieren lassen. Der Ton wird an dieser Stelle unterbrochen. Dadurch verschwindet das hier gespeicherte Bit und der Computer meldet »Read error b«. Bei einem echten Drop out ist guter Rat teuer. Dieses Problem taucht allerdings oft auch bei verstettem Tonkopf auf. Will man die Einstellung allerdings wie oben beschrieben mit Hilfe von »CAT« und dem Anfangston korrekt einstellen, so bekommt man Ärger mit dem Be-

ren Kassetten auftreten. Manche Bandsorten scheinen die Aufspeichrfrequenz des Schneiders nicht zu mögen. Jedenfalls produzieren sie bei »SPEED WRITE 1« laufend Fehlermeldungen. Das Interessante dabei ist, daß man durch Wahl einer höheren Schreibgeschwindigkeit diese Bänder zum perfekten Funktionieren bringen kann.

Zum Ändern der Schreibgeschwindigkeit existiert eine einfach zu bedienende Routine im Betriebssystem des Computers: CAS SET SPEED. Beim Anruf dieser Routine muß das Registerpaar HL die Länge für ein halbes Null-Bit (in Mikrosekunden) enthalten. Der Akkumulator A nimmt einen weiteren Parameter auf, die Vorprüflänge. Hier muß einiges zur Datenspeicherung gesagt werden. Jedem Bit im Speicher des Computers wird beim Aufzeichnen eine Rechteckschwingung zugeordnet. Bei einem Null-Bit ist die Periode halb so lang wie bei einem Eins-Bit. Bild 2 zeigt den Zusammenhang. Am Anfang jedes Bits steht ein »High-Teil« von der in HL gespeicherten Länge. Danach wird dieselbe Zeitspanne noch einmal mit Low-Pegel geschrieben. In Bild 2 ist Bit 1=0. Es folgt als nächstes ein Eins-Bit. Das Speicherformat ist das gleiche, jedoch doppelt so lang.

Bei dieser Art der Datenaufzeich-

Dieses besteht aus vier Maschinencode-Befehlen, die am Anfang in den Speicher ab 40000 geladen werden. Danach wird die Maschinencode-Routine aufgerufen. Sie setzt mit dem ersten Kommando HL auf den Wert 59 hex (dezimal 89) und mit dem nachfolgenden 3E hex (LLA) den Akkumulator auf die Vorprüflänge 7. Zum Schluß wird CAS SET SPEED aufgerufen und die neue Geschwindigkeit gesetzt. Mit den hier gewählten Parametern verdoppelt sich die Geschwindigkeit gegenüber der gewohnten »SPEED WRITE 1«. Man erreicht knapp 4000 Baud. Um Ihnen die Möglichkeit zu geben, selbst die Geschwindigkeit noch individueller zu wählen, sind ein paar Programmzeilen angefügt. In ihnen können Sie einfach Werte für HL und A angeben, die dann automatisch gesetzt werden. Haben Sie Ihre endgültige Wahl getroffen, so brauchen Sie nur noch die DATA-Zeilen ändern und können bis auf die ersten vier Zeilen alles löschen. Statt der 59 hex setzen Sie den Wert ein, den Sie durch »PRINT HEX\$(40001)« ausgegeben bekommen.

Dieselbe Operation für die Speicherstelle 40002 durchgeführt, liefert den Wert, der für die 00 hex an Stelle des dritten DATA\$-Werts einzusetzen ist. »PRINT HEX\$(40004)« gibt schließlich noch die Angabe,

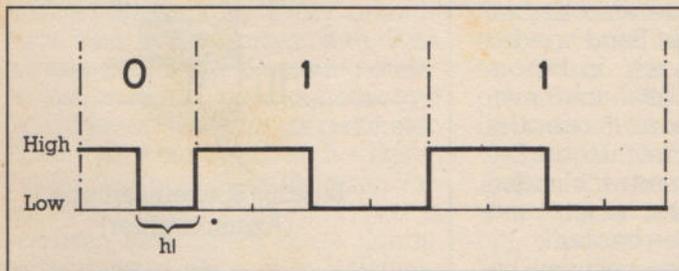


Bild 2. Regel wider der Mathematik:
Die 1 ist doppelt so groß wie die 0

```

10 MEMORY 39999 [0F22]
20 DATA 21,59,00,3E,07,CD,68,BC,C9,X [E84C]
30 FOR i=40000 TO 41000:READ a$:IF a$<>" [AA2A]
  X" THEN POKE i,VAL("&"+a$):NEXT [6D68]
40 CALL 40000 [1AE4]
50 INPUT"HL in Mikrosekunden":hl [0438]
60 INPUT"Vorprueflaenge A":a [AFF8]
70 POKE 40001,VAL("&"+MID$(HEX$(hl,4),3) [BF9C]
  ) [B7BA]
80 POKE 40002,VAL("&"+LEFT$(HEX$(hl,4),2) [D4DC]
  )
90 POKE 40004,a
100 GOTO 40
    
```

Listing 1. So wird Ihr Kassettenrecorder superschnell

triebssystem. Während beim Auftreten von »Read error a« (dieser gibt an, daß ein gelesenes Bit entweder zu kurz oder zu lang gespeichert ist und damit nicht eindeutig als 0 oder 1 identifiziert werden kann) jederzeit die Test-Kassette zurückgespult werden kann, um denselben Vorspann wieder zu benutzen, hängt sich der CPC bei einmaligem Auftreten von »Read error b« schon auf. Auch bei erfolgreicher Korrektur bringt ein einfaches Rückspulen keine Erfolgsmeldung. Nach »Read error b« muß der Ladevorgang immer mit der Escape-Taste abgebrochen und neu gestartet werden.

Interessant ist noch eine andere Art von Fehlern, die auch bei teure-

nung taucht nun aber ein Fehler auf. Einige Bausteine des Kassettenrecorders haben die Tendenz, die Flanken (Übergänge von High nach Low und umgekehrt) auszugleichen. Ein Eins-Bit wird beim Lesen dadurch weniger lang empfangen, ein Null-Bit dagegen länger. Als Korrekturfaktor dient da die Vorprüflänge. Diese Konstante wird zur Einsperiode addiert und von der Nullperiode abgezogen. Beim Lesen ergeben sich optimierte Werte. Nun wollen Sie aber sicher überprüfen, wie man mit diesem Wissen die Schreibgeschwindigkeit ändern kann. Dazu benutzt man am einfachsten ein kleines Hilfsprogramm (Listing 1).

die statt der 07 hex einzusetzen ist. Damit haben Sie Ihre kleine Schnellschreibroutine fertig und kann in jedes Programm eingebaut werden.

Nach diesem Ausflug in die Feinheiten der Datenspeicherung soll nun noch ein Lösungsweg für ganz problematische Fehler beschrieben werden. Speziell bei längeren Programmen lassen sich nämlich öfters sehr typische Fälle des »13. Murphy'schen Computergesetzes« finden: Je länger ein Programm ist, desto unwahrscheinlicher ist die Existenz einer Sicherungskopie.

Was tun, wenn man nun in dieser wenig beneidenswerten Situation mit solch einem Drop out konfrontiert ist? Zunächst ist zu sagen, daß

es keine perfekte Abhilfe gibt. Man kann aber sein wertvolles Programm meist wenigstens teilweise retten. Wie weiter oben schon gesagt, speichert der Schneider die Daten in Blöcken zu je 2 KByte. Dies ist auch die Voraussetzung dafür, daß überhaupt noch etwas zu retten ist. Da die Blöcke 1 bis X-1 (wobei X die Nummer des zerstörten Blocks darstellt) sauber gelesen werden können, können wir diese auch in den Speicher holen. Hierbei hilft uns der Befehl »CAT«. Mit diesem Kommando wird nämlich jeder Block einzeln eingelesen und ausgewertet. Die Anfangsadresse, ab der der CPC die hereinströmenden Daten ablegt, hängt von der Obergrenze des Benutzerspeichers (HIMEN) ab. Das Prinzip unserer Rettungsaktion ist nun relativ einfach. Zuerst müssen wir mit »CAT« und verschiedenen Speichergrenzen die einzelnen Blöcke hintereinander an der Speicherobergrenze ablegen. Nachdem wir das gesamte Programm bis zum fehlerhaften Block eingelesen haben, muß dieses dann noch in die normalen Adressen, an denen Basic-Programme normalerweise liegen, verlagert werden.

Ein Beispiel: In einem Programm ist der 7. Block zerstört. Es können also sechs Blöcke gerettet werden. Als erstes müssen wir einen Dummy schaffen, ein Programm, das um mindestens einen Block länger als die Anzahl der zu sichernden Blöcke ist. Dieses Programm hat eine Platzhalterfunktion und ist bei den folgenden Operationen permanent im Speicher. Überzeugen Sie sich dabei durch probeweises Abspeichern, daß das neue Programm

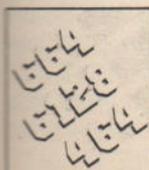
auch die richtige Blockzahl hat. An dieses Programm fügen Sie am Ende mit der höchsten Zeilennummer noch eine Zeile an: »Nummer FOR i=Anfang TO 41855:POKE i-Anfang+368,PEEK(i):NEXT:END«. »Anfang« stellt dabei die Untergrenze des abzuspeichernden Bereiches dar. Dabei muß folgendes berücksichtigt werden. Der Schneider reserviert ja bei jeder Ladeoperation 4 KByte als Zwischenspeicher. Dabei sind die oberen 2 KByte der aktuellen Eingabedatei vorbehalten, die unteren 2 KByte gehören der Ausgabe. Wenn Sie also ein Programm mit »CAT« überprüfen, wird in den Bytes HIMEN-4095 bis HIMEN-2048 ein 2 KByte langer Block eingelesen. Nun kann der Wert des Platzhalters »Anfang« leicht bestimmt werden. Jeder Block besteht aus 2048 Bytes (= 2 KByte). Wir verschieben nun die Speicherobergrenze mit MEMORY um die einzulesende Blockzahl nach unten. Der Befehl hierzu heißt »MEMORY 43903-2048*(Blockzahl-1)«. In unserem Beispiel wären dies $6 \times 2048 \text{ Bytes} = 12288 \text{ Bytes}$. Wenn Sie nun »CAT« ausführen lassen, wird in den 2 KByte unterhalb dieser Grenze der erste Block eingelesen. Nachdem Sie das »OK« für diesen ersten Block auf dem Bildschirm haben, muß das Programm mit der Escape-Taste abgebrochen werden. Mit »MEMORY HIMEN+2048« wird die Speichergrenze 2 KByte nach oben verschoben.

Noch ein »CAT« und der nächste Block ist im Speicher. In unserem Fall muß dieser Vorgang sechsmal wiederholt werden, bis das defekte Programm im Speicher ist.

Mit der letzten Zeile unseres Dum-

mys verschieben wir die geretteten Befehle. »Anfang« ist dabei durch die Untergrenze unseres Programms (allgemein 43903-2048* Blockzahl) zu ersetzen. Die Routine verschiebt die gesamten sechs Blöcke in den Basic-Programmbe- reich ab Adresse 368. Wenn Sie nun den Dummy wieder listen, finden Sie dort die überlagerten Zeilen des zerstörten Programmes wieder. Die letzte Zeile sieht allerdings meist etwas seltsam aus. Dies resultiert daraus, daß wir das Programm nur teilweise übertragen haben. Ein solches Teil-Programm bringt nun beim Speichern Probleme. Auch auf ein »DELETE diese Zeilennummer« reagiert der CPC unwirsch. Daher müssen wir als letztes auf einen Trick zurückgreifen, die Abspeicherung von Programmen als ASCII-Datei. Wir eröffnen dazu mit »OPENOUT Programmname« eine Datei zum Schreiben, an die wir dann mit »LIST (n-1),#9« unser Programm ausgeben. »n« ist die erste unsinnige Zeile (manchmal interpretiert der CPC nämlich fröhlich weiter Unsinn). Nun stellen Sie den Computer mit Ctrl+Shift+Esc zurück und laden das gerade gesicherte Programm wie gewohnt. Der nächste SAVE-Befehl funktioniert wieder ganz normal. Damit haben Sie — wenigstens teilweise — das Programm gerettet. Allerdings mit einem erheblichen (Zeit-) Aufwand. Einen Ersatz für eine Sicherungskopie kann diese Methode also nicht bieten. Sie sollten sich deshalb angewöhnen, spätestens jede Stunde, bei schneller Eingabe oder sehr schwierigen Problemen auch häufiger, eine Sicherheitskopie zu machen. (Carsten Straush)

Firmware-Routinen in Basic genutzt



Folgen Sie unserem Streifzug durch das Betriebssystem der Schneider-Computer, soweit es für eigene Basic-Programme genutzt werden kann.

Das fest im ROM installierte Betriebssystem der Schneider-Computer, die sogenannte Firmware, ist in eine Reihe wichtiger Abschnitte unterteilt, die jeweils spezielle Aufgaben wahrnehmen. In ihrer Gesamtheit steuert die Firm-

ware alle Zugriffe auf die Hardware, bewirkt also beispielsweise die Ausgabe von Zeichen auf dem Bildschirm, die Tastaturabfrage oder die Ansteuerung des Kassetten-Re-corders (beim CPC 464).

Alle Firmware-Routinen sind zum

Aufruf durch Maschinensprache-Programme konzipiert, das heißt, daß alle Parameter in der Regel in den Prozessor-Registern übergeben werden. Da der CALL-Befehl die direkte Übergabe von Zahlenwerten an die Register des Z80-Prozessors nicht unterstützt, können die weitaus meisten der etwa 250 verschiedenen Routinen von Basic-Programmen aus nicht direkt aufgerufen werden. Allerdings ist das in vielen Fällen auch nicht nötig, denn um die große Vielfalt der Basic-Funktionen zu realisieren, bedient sich auch der Interpreter des CPC 464/664/6128 der Firmware-Routinen.

Eine Reihe interessanter Effekte lassen sich aber durch geschickte Anwendung spezieller Firmware-CALLs erreichen. Im Folgenden

werden die wichtigsten und interessantesten Firmware-Routinen und Betriebssystemadressen mit ihrer Wirkung vorgestellt.

Tastatur (Key Manager --KM):

Wie bei den meisten anderen Firmware-Abschnitten gibt es auch für den Tastaturbereich eine Initialisierungs-Routine, welche die Tastatur in den Zustand direkt nach dem Einschalten des Computers versetzt. Diese Routine mit Namen KM INITIALISE kann über »CALL &BB00« aufgerufen werden. Dieser Aufruf ist zum Beispiel dann empfehlenswert, wenn man mit vielen »KEY«- und »KEY DEF«-Befehlen die Tastatur soweit »vermurkst« hat, daß ein vernünftiges Arbeiten nicht mehr möglich ist.

Eine ähnliche, wenngleich nicht ganz so weitgehende Routine ist KM RESET (BB03hex). Diese Routine löscht eine eventuell vorhandene Belegung der Funktionstasten und den Tastaturpuffer. Ein »CALL &BB03« könnte am Ende eines Programms stehen, das die Funktionstasten für eigene Zwecke umdefiniert hatte und den ursprünglichen Zustand wieder herstellen möchte.

Oft kommt es in einem Anwenderprogramm zu der Situation, daß man auf einen Tastendruck des Benutzers warten muß. In der Regel formuliert man das als »A\$="" : WHILE A\$="" : A\$=INKEY\$: WEND«.

Unter Ausnutzung der Firmware-Routine KM WAIT CHAR (BB06hex) geht es auch kürzer: »CALL &BB06« wartet solange, bis irgendeine Taste

grammlaufs. Sehen Sie schon den Nutzen dieser Routine? Probieren Sie einmal das Mini-Programm aus Listing 1 aus.

Nach »RUN« geschieht folgendes: Der Aufruf von KM BREAK EVENT bewirkt das Erscheinen des Cursors; der Programmlauf ist angehalten. Die Routine wartet auf das Drücken einer beliebigen Taste. Dieser Tastendruck wird durch »A\$=INKEY\$« festgehalten. Zeile 20 erzeugt ein Echo der Eingabe ähnlich wie bei INPUT. Bei Drücken der Enter-Taste wird das Programm beendet. Damit wird es sehr einfach, beispielsweise einen formatierten INPUT-Befehl für Bildschirmmasken zu programmieren.

Ein Problem ist allerdings die Esc-Taste. Da die Routine ja praktisch die einmalige Betätigung der Esc-Taste simuliert, ist jetzt äußerster Vorsicht geboten, um nicht versehentlich Esc zu berühren, was den Abbruch des laufenden Programms zur Folge hätte. Abhilfe schafft hier entweder das völlige Abschalten der BREAK-Funktion mit »ON BREAK GOSUB . . .« oder (einfacher) das Umdefinieren der Esc-Taste: Nach »KEY DEF 66,0,0,252,252« wirkt die Esc-Taste nur noch bei Betätigung zusammen mit Shift oder Ctrl — eine einfache Sicherung gegen unabsichtliche Betätigung.

Wem das Hantieren mit BREAK-Routinen und Esc-Taste nicht so ganz geheuer ist, für den gibt es noch zwei interessante Routinen, mit denen das Erscheinen des Cursors am

halten wird. Man muß sich bei dieser Methode wieder der bewährten WHILE-WEND-Technik bedienen.

Übrigens zeigt dieses Beispiel eine Art der Anwendung des CALL-Befehls, die zu übersichtlicheren Programmen führt: Die Aufrufadresse von Maschinenprogrammen wird an Variable zugewiesen, so daß Maschinenprogramme praktisch mit Namen aufgerufen werden. Wenn diese Namen dann noch halbwegs das ausdrücken, was die Maschinenroutine macht, dann kann man die CALL-Befehle als echte Erweiterung des Schneider-Basic auffassen. Die beiden Routinen zum Ein- und Ausschalten des Cursors sind besonders nützlich, denn jeder Programmierer sollte den späteren Benutzer des Programms immer ganz klar darauf aufmerksam machen, wenn eine Eingabe erwartet wird. Beim INPUT-Befehl ist das ja kein Problem, da erscheint der Cursor von alleine. Bei Eingaben durch INKEY\$ kann man mit den beiden eben beschriebenen Routinen arbeiten, um es ganz deutlich zu machen: Hallo Benutzer, ich erwarte eine Eingabe . . . hier ist der Cursor!

Wenn man den Cursor lieber blinkend mag (wie von anderen Heimcomputern bekannt), dann ist das auch beim Schneider kein Problem. Man muß nur das kleine Listing 3 eingeben.

Ähnlich wie beim KEY MANAGER gibt es auch für die drei Betriebssystemteile, die sich mit dem Bildschirm befassen, spezielle Ini-

```
10 CALL &BB4B:a$=INKEY$ [24FA]
20 PRINT a$;:IF a$<>CHR$(13) THEN 10 [F7BA]
```

Listing 1. Die Tastatur kann sehr luxuriös abgefragt werden

```
10 cplace=&BB8A 'Cursor ein [9E54]
20 cremove=&BB8D 'Cursor aus [7C48]
30 CALL cplace:a$="" [49CE]
40 WHILE a$="" : a$=INKEY$:WEND [FE56]
50 CALL cremove:PRINT a$; [4DB2]
60 IF a$<>CHR$(13) THEN 30 [1B78]
```

Listing 2. Eine Tastaturabfrage mit Echo

```
10 cplace=&BB8A 'Cursor ein [9E54]
20 cremove=&BB8D 'Cursor aus [7C48]
30 flag%=-1:EVERY 25,1 GOSUB 90 [435E]
40 CALL cplace:a$="" [3AD0]
50 WHILE a$="" : a$=INKEY$:WEND [D258]
60 CALL cremove:PRINT a$; [EFB4]
70 flag%=REMAIN(1):REM Timer abschalten [7C6A]
80 IF a$<>CHR$(13) THEN 30 [797C]
90 flag%=NOT flag%:IF flag% THEN CALL cremove ELSE CALL cplace [1B4C]
100 RETURN [AA22]
```

Listing 3. Der Cursor kann auch blinken

gedrückt wurde. Allerdings läßt sich anschließend nicht mehr feststellen, welche Taste das war, was aber in sehr vielen Fällen ja auch überflüssig ist.

Eine andere, noch interessantere Routine ist KM BREAK EVENT (&BB4B). Der Aufruf dieser Routine bewirkt das gleiche, wie das einmalige Drücken der Esc-Taste: Das Programm wird angehalten, und es erscheint ein Cursor. Ein einfacher Druck auf Esc bricht das Programm jetzt sofort ab. Jede andere Taste bewirkt eine Fortsetzung des Pro-

Bildschirm sehr einfach gesteuert werden kann. Beide Routinen befinden sich im TEXT-Abschnitt der Firmware: TXT PLACE CURSOR (BB8Ahex) läßt den Cursor an der aktuellen Cursorposition erscheinen, TXT REMOVE CURSOR (BB8Dhex) entfernt ihn wieder vom Bildschirm. Mit diesen beiden Routinen sieht das Beispiel zur Tastaturabfrage mit Echo wie in Listing 2 aus.

Dieses Beispiel ist länger als das erste (mit KM BREAK EVENT), weil das Programm durch das Setzen des Cursors nicht automatisch ange-

tialisierungsroutinen: TXT INITIALISE (BB4Ehex) löscht alle Windows, schaltet einen eventuellen Transparent-Ausgabemodus ab, führt PAPER 0 und PEN 1 aus und setzt den Cursor schließlich in die linke obere Bildschirmecke.

GRA INITIALISE (BBBAhex) führt die analogen Operationen für die hochauflösende Grafik aus. SCR INITIALISE (BBFFhex) schließlich initialisiert die gesamte hardwarenahe Bildschirmansteuerung. Im einzelnen werden alle Farben auf den Einschaltzustand zurückgesetzt

und ein eventuelles Blinken abgeschaltet. Weiter wird MODE 1 gewählt, dabei der Bildschirm gelöscht und die Startadresse des Bildschirms auf den Standardbereich C000 bis FFFFhex gesetzt.

Wenn man sich nun etwas mit seiner Farbwahl vertan hat und auf dem Bildschirm kaum noch etwas erkennen kann, dann reicht oft ein Aufruf von SCR RESET (BC02hex), um den Schaden wieder zu beheben. SCR RESET setzt alle Farben wieder auf den Einschaltzustand. Reicht das noch nicht aus, dann sollte man noch ein »CALL &BB4E« nachschieben (TXT INITIALISE), um auch PEN und PAPER wieder vernünftig zu setzen. Bei der Ausgabe von Grafik auf dem Bildschirm kann es zu gewissen Flimmereffekten kommen. Dies läßt sich elegant umgehen durch Aufruf von MC WAIT FLYBACK (BD19hex). Der Aufruf dieser Routine entspricht dem Basic-Befehl FRAME im Basic 1.1 des 664/6128. Die Routine wartet einfach auf den Rücklauf des Elektronenstrahls, der das Bild auf den Bildschirm zeichnet. Während der Rücklaufphase des Bildschirms wirken sich Manipulationen am Bildschirminhalt nicht auf den optischen Eindruck aus. Die Routine muß also vor jeder kritischen Bildschirmausgabe (besonders bei schneller Grafik) aufgerufen werden, um Flimmerfreiheit zu garantieren. Allerdings leidet natürlich die Ausführungsgeschwindigkeit des Programms darunter, denn der einzige Zweck der Routine MC WAIT FLYBACK ist das Warten — darauf, daß der Elektronenstrahl der Bildröhre ein Bild fertig gezeichnet hat.

Zur Grafik gibt es noch einen interessanten POKE-Befehl, der den beim CPC 464 nicht vorhandenen Basic 1.1-Befehl MASK simuliert: Probieren Sie mal »POKE &B338, PEEK(&B338) AND &AA: MOVE 100,100! DRAW 500,300« (Achtung, funktioniert nur beim CPC 464!). Die Speicherstelle B338hex enthält nämlich eine Farbmaske für den grafischen Zeichenstift. Diese Farbmaske kann man nochmals maskieren (mit AND und einem Bit-Muster), so daß unterbrochene Linien gezeichnet werden.

Der SCREEN PACK (SCR) enthält noch eine weitere Routine, die ab und zu nützlich sein kann: SCR HW ROLL (BC4Dhex) rollt den Bildschirm um eine Textzeile aufwärts oder abwärts. Dieses Rollen wird hardwaremäßig realisiert, indem einfach nur der Bildschirm-Offset-Wert in einem Register des Video-

Chips verändert wird. Daher ist die Methode sehr schnell. Allerdings kann immer nur der gesamte Bildschirm so gescrollt werden, kleinere Windows lassen sich nicht einzeln bewegen.

Ob aufwärts oder abwärts gescrollt wird, entscheidet der Inhalt des B-Registers der Z80-CPU bei Aufruf der Routine. Ist B gleich Null, wird nach unten gescrollt, in allen anderen Fällen nach oben. Beim Aufruf eines Maschinenprogramms mit dem CALL-Befehl kann das B-Register, wie die meisten anderen CPU-Register, leider nicht direkt beeinflusst werden. Es enthält bei Eintritt in die aufgerufene Routine immer die Höchstzahl 32 der möglichen Parameter des CALL-Befehls minus die Anzahl der tatsächlich übergebenen Register. Die Adresse der Maschinenroutine zählt dabei nicht als Parameter. Dadurch enthält B immer den Wert 32, falls kein weiterer Parameter angegeben wurde. Für die Routine zum Scrollen des Bildschirms bedeutet dies, daß normalerweise immer ein Scrollen nach oben realisiert wird. Um nach unten zu scrollen, müssen 32 Parameter zum CALL-Befehl angegeben werden. Die Werte dieser Parameter spielen keine Rolle, es kommt nur auf die Anzahl von 32 an, damit das B-Register auf Null kommt. Ein Aufruf der Art »CALL &BC4D, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32« zum Beispiel läßt den Bildschirm eine Textzeile nach unten scrollen. Nicht sehr praktikabel, aber immerhin ein Notnagel, wenn man ohne Maschinensprache auskommen will. Als reine Basic-Alternative bietet es sich natürlich noch an, einfach den Cursor mit LOCATE in die oberste oder unterste Zeile zu setzen, und dann mit Steuerzeichen für »Cursor up« oder »Cursor down« ebenfalls ein Scrollen des Bildschirms oder sogar einzelner Windows zu erreichen.

Sound Manager

Der Sound Manager ist derjenige Teil der Firmware, der sich mit der Ansteuerung des Sound-Chips befaßt. Hier gibt es drei mehr oder minder interessante, durch CALL-Befehle nutzbare Routinen: SOUND RESET (BCA7hex) löscht alle Sound-Warteschlangen und bringt den Sound-Chip zum Schweigen. SOUND HOLD (BCB6hex) dagegen unterbricht einen gerade gespielten Ton-Effekt, ohne ihn zu beenden. Mit SOUND CONTINUE (BCB9hex) wird die unterbrochene Tonfolge korrekt zu Ende gespielt.

Cassette Manager

Der Cassette Manager kümmert sich, wie der Name schon sagt, um alles, was mit Aufzeichnung und Lesen von Daten auf Kassette zusammenhängt. Für den Basic-Programmierer interessant ist zunächst wieder die Initialisierung: CAS INITIALISE (BC65hex) markiert alle Dateien als geschlossen und setzt die Aufzeichnungsgeschwindigkeit auf SPEED WRITE 0.

Wenig bekannt ist, daß zwei Dateien gleichzeitig geöffnet sein können, eine zum Lesen und eine zum Schreiben. Der Cassette Manager verwaltet Schreiben und Lesen von/zur Kassette/Diskette vollkommen getrennt. Bei Verwendung von Kassetten zur Datenspeicherung muß man nur darauf achten, daß man je nach Aufforderung des Betriebssystems (»Press Play then any key« oder »Press Record&Play then any key«) immer zwischen Schreib- und Lesekassette wechselt. Auf diese Art und Weise lassen sich zum Beispiel Dateien führen, die umfangreicher sind als der verfügbare Speicherplatz.

Brauchbare Routinen des Cassette Manager sind CAS START MOTOR (BC6Ehex) und CAS STOP MOTOR (BC71hex), mit denen der Motor des Kassetten-Recorders ein- und ausgeschaltet werden kann. Die Routinen sind sinnvoll, wenn man dem Benutzer von einem Programm aus die Möglichkeit geben will, mit der PLAY-Taste das Band genau auf eine bestimmte Zählwerkstelle zu positionieren. Ein anderer denkbarer Anwendungszweck ist das automatische Vorspulen zu einem bestimmten Programm auf der Kassette. Dazu muß die Vorspulzeit bei schnellem Vorlauf bekannt sein, dann kann man das Band diese Zeit vorspulen. Natürlich ist das recht umständlich im Vergleich zum Komfort einer Diskettenstation.

Das waren die Firmware-Routinen der Schneider-Computer, soweit sie sich durch CALL-Befehle von Basic aus nutzen lassen und nicht bereits direkt durch Basic-Befehle realisiert sind. Vielleicht haben Sie hieraus einige Anregungen zur effektiveren Programmierung ziehen können. Eines aber sollten Sie auf jeden Fall bedenken: Übertreiben Sie den Gebrauch von CALL, PEEK und POKE nicht, denn er führt sehr schnell zu völlig unleserlichen Programmen. Wenn Sie jemals ein Listing etwa für den Commodore 64 gesehen haben, dann wissen Sie, was gemeint ist ...

(Volker Everts)

Ihr Schneider mal ganz intern



Ein Computer kann viel mehr als sein Basic vermuten läßt. Im Betriebssystem findet man oft sehr interessante Anweisungen. Hier finden Sie die wichtigsten ROM-Routinen.

In jedem Computer sind wichtige Maschinencode-Routinen, die immer wieder benötigt werden, im ROM fest eingebaut — Programme, die beispielsweise eine Linie auf dem Bildschirm zeichnen, ein Zeichen an den Drucker ausgeben und so weiter. Wenn man die Adressen dieser Routinen kennt, kann man sich bei eigenen Programmen einen Haufen Arbeit sparen.

Bei den drei Schneider-Computern können die Betriebs-Routinen indirekt aufgerufen werden: Im RAM-Bereich zwischen B900 und BD39hex befinden sich Sprungbefehle zu verschiedenen Routinen. Möchte man solch eine Befehlsfolge aufrufen, dann wird in der Regel nur der entsprechende Sprungbefehl in dieser Tabelle aufgerufen. Erst dieser Befehl leitet die Bearbeitung der eigentlichen Routine ein. Dieser Umweg hat mindestens zwei Vorteile:

— Das eigentliche Unterprogramm kann im Speicher an einer beliebigen Stelle stehen. Dadurch ist es möglich, daß sich der Einsprung auf eine bestimmte Adresse bezieht, die eigentliche Routine aber ganz woanders steht.

Programme, die nur diese indirekten Sprungvektoren verwenden, sind damit vom Computertyp unabhängig.

— Durch die Lage der Sprungbefehle im RAM kann die Tabelle leicht abgeändert werden. Durch den Einbau einer »Umleitung« kann eine fest eingebaute Systemfunktion ergänzt oder sogar ganz durch eine andere ersetzt werden.

In den Schneider-Computern gibt es ungefähr 200 auf diese Weise standardisierte Unterprogramm-funktionen. Die große Zahl macht es unmöglich, alle Routinen einzeln aufzuzählen. In der folgenden Tabelle sind deshalb nur die wichtigsten aufgeführt. Es fehlt beispielsweise die Maschinencode-Sequenz

zum Löschen des Bildschirms (diese kann ja leicht durch Senden des Steuerzeichens mit dem Code 0Chex) ersetzt werden. Auch andere Funktionen (hauptsächlich den Bildschirm betreffend) lassen sich so leicht aufrufen.

Die Routinen stammen vom CPC 464. Sie laufen aber ohne Änderung teilweise auch auf dem 664 oder 6128. Hier hilft einfach ausprobieren. Starten Sie eine Entdeckungsreise in das Betriebssystem Ihres Computers.

(Helmut Tischer)

Die RST-Befehle erlauben beim Z80 frei programmierbare Objektcodes. In den CPC-Computern werden diese als erweiterte CALL/JP-Befehle eingesetzt. Einem normalen CALL-Befehl folgen zwei Bytes, die die Adresse eines Unterprogramms im Speicher angeben. Einem RST-Befehl folgen hierbei zwei Bytes, die eine Adresse enthalten. Damit kann eine beliebige Adresse im gesamten Speicher des Computers ausgewählt werden — sowohl im eingebauten RAM oder ROM als auch in beliebigen Erweiterungs-ROMs.

RST 08hex: Diese Anweisung wirkt ähnlich einem JP-Befehl. Die beiden nachfolgenden Bytes enthalten nur in den niederwertigen 14 Bits eine Adresse. Deshalb kann auch nur eine Routine im Adressbereich von 0000hex bis 3FFFhex aufgerufen werden. Dafür ist aber der Zustand der ROMs während der Ausführung des ausgewählten Programms frei wählbar: Hat Bit 15 den Wert 0, so kann während der Ausführung das obere ROM gelesen werden. Hat Bit 14 den Wert 0, so wird das untere ROM eingeschaltet. Nach Beenden der Routine wird der vorherige ROM/RAM-Zustand wiederhergestellt.

RST 10hex: Dieser RST-Befehl ist einem CALL-Befehl vergleichbar. Die unteren 14 Bits der nachfolgenden zwei Bytes geben wieder eine Adresse an, die jedoch diesmal zu einem Sprung ins obere ROM führt. Die Bits 14 und 15 werden vor dem Aufruf des Unterprogramms zur laufenden ROM-Nummer addiert. Steht der »RST 10hex«-Befehl in einem ROM mit der Nummer x und haben die Bits 14 und 15 den Wert 10bin(=2), so wird ein Unterprogramm im Erweiterungs-ROM Nummer x+2 aufgerufen.

RST 18hex: Das ist der effektivste CALL-Befehl bei den Schneider-Computern. Damit kann jede beliebige Adresse in jedem beliebigen ROM/RAM aufgerufen werden. Dazu sind auch hier nur zwei nachfolgende Bytes nötig. Diese zwei Bytes sind ein Zeiger auf eine Tabelle im zentralen RAM. Diese Tabelle enthält drei Bytes mit der vollständigen Adresse: Die ersten beiden Bytes geben die 16-Bit-Adresse des Unterprogramms an und das dritte Byte die Nummer des während des Aufrufs selektierten Erweiterungs-ROMs. Ausnahmen bilden die ROM-Nummern FC bis FFhex: Bei diesen bleibt das schon vorher gewählte ROM unverändert. Die Bits 0 und 1 regeln analog zum »RST 08hex«, welches ROM/RAM ein- oder ausgeschaltet ist.

RST 20hex: Diesem RST-Befehl folgt kein dazugehöriges Byte. Er wirkt wie ein »LD A,(HL)«-Befehl, liest jedoch immer aus dem RAM.

RST 28hex: Dieser Befehl wirkt ähnlich einem »JP«-Befehl. Vor dem Ansprung des Unterprogramms wird aber das untere ROM eingeschaltet. Der Zustand des oberen bleibt unverändert. Wegen der zwei nachfolgenden Bytes, die als 16-Bit-Adresse interpretiert werden, kann ein Programm im gesamten Speicherbereich angesprungen werden.

001Bhex: Diese Routine wirkt ähnlich wie der »RST 18hex«-Befehl. Die ROM-Nummer befindet sich jedoch im Register C und die Adresse des Programms im Register HL.

BCD4hex: Diese Routine sucht einen RSX-Befehl im gesamten Speicher. Beim Aufruf muß HL auf den Namen des Befehls zeigen, wobei Bit 7 im letzten Namensbyte gesetzt ist. Als Ergebnis erhält man im C-Register die Nummer des ROMs, in dem sich der Befehl befindet und im HL-Register die Adresse des Befehls.

B900hex: Schaltet das obere ROM ein.
B903hex: Schaltet das obere ROM aus.
B906hex: Schaltet das untere ROM ein.
B909hex: Schaltet das untere ROM aus.

Weitere Routinen zum Bankswitching

Adresse	Übergabeparameter	Ergebnis	Bedeutung
BB00 hex	—	—	Original-Tastenbelegung wiederherstellen
BB09 hex	—	CY: 1, wenn ok.	Ist eine Taste gedrückt?
BB0C hex	a:Code	A: Code	Nächstes Lesen von der Tastatur gibt diese Zeichen
BB0F hex	B:Stringnummer	—	Funktionsstring setzen
BB15 hex	C:Länge HL:Adresse String	CY: 1, wenn ok.	Speicher für Funktionsstrings reservieren
BB27 hex	DE:Adresse HL:Länge	—	Belegung einer Taste ändern
BB2D hex	A:Tastenummer B:neuer Code	—	Belegung Taste mit Shift ändern
BB33 hex	wie BB27 hex	—	Belegung Taste mit Ctrl ändern
BB39 hex	wie BB27 hex	—	Repeat-Status ändern
BB3F hex	A:Tastenummern B:FF = Repeat/ 00 = nicht H:Anfangsverzögerung L:Wiederholverzögerung	—	Repeatgeschwindigkeit einstellen

(CY= carry flag)

Wichtige Routinen zur Tastaturverwaltung

Adresse	Eingabe	Ergebnis	Bedeutung
BB5A hex	A:Code	—	(Steuer)zeichen ausgeben
BB60 hex	—	CY:1, wenn ok. A:Code	Zeichen an Cursorposition lesen
BB87 hex	H:Spalte L:Zeile	CY=1 => Korrigieren rollt Bildschirm bei B:FF nach oben und bei B:00 nach unten H:neue Spalte L:neue Zeile	Testen ob zu testende Stelle im Fenster ist: wenn nein, neue Position angeben
BBAE hex	—	wie BBAB hex	Definiert Matrix vorhanden?
BBB1 hex	—	HL:Adresse	Bestimmt die Adresse der Steuerzeichen Sprungtabelle. Für jedes Steuerzeichen von 00 bis 1F hex drei Byte: 1. = Anzahl der Parameter, 2. = Routineadresse

Wichtige Routinen zur Zeichenausgabe

Adresse	Eingabe	Ergebnis	Bedeutung
BBC0 hex	—	—	MOVE-Befehl
BBC3 hex	—	—	MOVER-Befehl
BBC9 hex	—	—	ORIGIN-Befehl
BBEA hex	—	—	PLOT-Befehl
BBED hex	—	—	PLOTR-Befehl
BBF0 hex	—	A:Farbe	TEST-Befehl
BBF3 hex	—	A:Farbe	TESTR-Befehl
BBF6 hex	—	—	DRAW-Befehl
BBF9 hex	—	—	DRAWR-Befehl
BBCF hex	DE:links HL:rechts	—	Breite des Grafikwindows setzen
BBD2 hex	DE:oben HL:unten	—	Höhe des Grafikwindows setzen
BBDB hex	—	—	Grafikwindow löschen
BBDE hex	A:Farbe	—	Grafik-Farbstift setzen
BBE1 hex	A:Farbe	—	Grafikhintergrundfarbe

Wichtige Grafik-Routinen

Adresse	Eingabe	Ergebnis	Bedeutung
BD19 hex	—	—	auf Bildschirm-Strahlrücklauf warten
BD2B hex	A:Code	—	Zeichen drucken
BD2E hex	—	CY:l=ok.	Drucker frei?
BD34 hex	A:Registernummer C:Daten	—	Sound-Chip mit Daten versorgen
BC68 hex	HL:Geschwindigkeit A:Kompensation	—	Aufzeichnungsgeschwindigkeit Baud-Rate = 333333/HL
BC6B hex	A:00 Meldungen da FF keine Meldungen	—	Meldungen »Press Play« und so weiter Ein-/Ausschalten
BC6E hex	—	—	Kassetten-Motor starten
BC71 hex	—	—	Kassetten-Motor stoppen
BC77 hex	B:Länge Dateiname HL:Adresse Namen DE: 2 KByte Puffer	CY:l = > o.k. Z:l = > Break 0 = > already open	OPENING-Funktion
BC7A hex	—	CY:0 = > not open	CLOSEIN-Funktion
BC7D hex	—	—	sofortiges Schließen der Eingabe (für Fehlerfall und so weiter)
BC80 hex	—	A:Byte CY:l=ok. Z:l = > Break 0 = > EOF	Zeichen von der Datei lesen
BC89 hex	—	CY=l:nicht erreicht Z:l -> break	EOF-Funktion
BC8C hex	wie BC77 hex	-	OPENOUT-Befehl
BC8F hex	wie BC7A hex	-	CLOSEOUT-Befehl
BC92 hex	wie BC7D hex	-	wie BC7D hex, ohne Zurückzuschreiben des Puffers
BC95 hex	A:Byte	wie BC80 hex	Zeichen an Kassette ausgeben
BC9B hex	—	CY:0 = > Fehler	CAT-Befehl

Wichtige Routinen zur Ein-/Ausgabe

Adresse	Eingabe	Ergebnis	Bedeutung
BC05 hex	HL:Versatz	—	Lage der linken oberen Ecke relativ zum Bildschirm
BC08 hex	A:Höherwertiges Byte	—	Bildschirmanfangesadresse setzen
BC20 hex	HL:Adresse	HL:Adresse	Bildschirmadresse berechnen, die auf der Anzeige rechts neben der angegebenen liegt
BC23 hex	HL:Adresse	HL:Adresse	wie BC20 hex, jedoch links
BC26 hex	HL:Adresse	HL:Adresse	wie BC20 hex, jedoch unten
BC29 hex	HL:Adresse	HL:Adresse	wie BC20 hex, jedoch oben
BC3E hex	H:Zeit 1. Farbe L:Zeit 2. Farbe	—	Blinkzeiten einstellen
BC4D hex	B:00 = > unten FF = > oben	—	Bildschirm hardwaremäßig rollen
BC50 hex	A:Farbe zum Löschen H:links (Byte) D:rechts L:oben (Rasterzeile) E:unten A:Farbe B:Wie BC4D hex	—	Window softwaremäßig rollen

Wichtige Routinen zur Bildschirmsteuerung (Wenn nichts anderes erwähnt, enthält DE die x-Koordinate und HL die y-Koordinate)

Welcher Wert steht wo?



In einem Programm alle benutzten Variablen zu kennen, ihre Position zu finden und bei Änderungen keine zu vergessen, das ist auch für geübte Programmierer ein risikoreiches Unterfangen. Hilfe bietet dieses Cross-Reference-Programm, das die Namen heraussucht und auf die dazugehörigen Zeilen verweist.

Cross-Reference-Routinen, die in Basic geschrieben sind, haben zwei erheblich störende Nachteile: Zum einen sind sie sehr langsam und zum zweiten muß das zu untersuchende Programm in Form einer ASCII-Datei vorliegen. Normalerweise werden Basic-Befehls-worte als »Token« verkürzt gespeichert. In einer ASCII-Datei würden die vier Werte 47, 4F, 54 und 4F (alle hexadezimal) dem GOTO entsprechen. Mit dem Schneider-Computer ist es zwar einfach aus einem Basic-Programm eine ASCII-Datei zu machen (einfach mit dem Zusatz »A« speichern). Der dazugehörige Speicher- und Ladevorgang ist aber umständlich. Aus diesem Grunde ist »Cref« vollständig in Maschinencode geschrieben. Auch 20 KByte lange Programme werden somit in Sekundenschnelle bearbeitet und ausgewertet.

Der Basic-Interpreter Ihres Schneiders stellt beim Bearbeiten eines Programms eine Tabelle der benutzten Variablen auf, um bei späteren Wertzuweisungen (oder Bildschirmausgaben) ohne langes Suchen die richtige Variable mit ihrem aktuellen Wert zu finden.

Grundidee von »Cref« ist es, diese Tabelle durch Aufruf geeigneter Interpreter-Routinen aufbauen zu lassen ohne das Programm zu starten. Dadurch werden vorzeitige Wertzuweisungen verhindert. Für jeden einzelnen Namen aus der Tabelle wird das Programm nach entsprechenden Referenzen durchsucht. Somit spart man sich den Aufbau einer speziellen sortierten Referenzli-

ste. Der Speicherbedarf reduziert sich dadurch erheblich, so daß auch große Programme ohne Probleme bearbeitet werden können.

»Cref« liegt im Speicherplatz A000 hex und ist damit auch auf den Betrieb des Diskettenlaufwerks von Schneider abgestimmt. Besitzer eines anderen Laufwerks, eines CPC 664 oder eines CPC 6128, müssen unbedingt die von »Cref« benutzten System-Routinen und -Vektoren anpassen.

Token statt Befehle

Zum besseren Verständnis der Routine sind einige Kenntnisse über die Ablage von Basic-Zeilen beim Schneider nötig. Betrachten wir hierzu Listing 1. In Bild 1 finden Sie das gleiche Programm so wie es der Computer nach dem Befehl LOAD gespeichert hat. Es fällt sofort auf, daß die Zeilen nicht im ASCII-Code vorliegen, sondern »mundgerecht« für den Interpreter aufbereitet sind. Anstelle der Basic-Befehle tauchen die sogenannten Tokens auf. Dabei handelt es sich um nichts anderes als eine (allerdings sehr trickreich) verkürzte Schreibweise für den Befehl. Da ein Token maximal zwei Byte lang ist, wird er vom Interpreter schneller erkannt und bearbeitet. Weiter sieht man sofort, daß Integer-Zahlen in eine zwei Byte lange hexadezimale Zahl umgewandelt werden und daß die Zeilennummer im 3. und 4. Byte der Zeile (hexadezimal) vorliegt. Die ersten zwei Byte ei-

ner Zeile geben die Länge der vom Interpreter umgewandelten Zeile an. Diese ist meist nicht identisch mit der Länge der ursprünglichen Zeile. Die Namen der Variablen entsprechen bis auf die letzte Stelle der normalen ASCII-Schreibweise. Zum Wert des letzten Zeichens wird immer 80 hex addiert, also das siebte Bit gesetzt. Die Schreibweise in Groß- und Kleinbuchstaben wird beibehalten, obwohl der Computer beim späteren Bearbeiten keinen Unterschied macht (»variable« ist für ihn beispielsweise das gleiche wie »VARIABLE« oder »vAriABLE«).

Von der Syntax her ist es jetzt leicht verständlich, wie der Interpreter eine Zeile bearbeitet. Nach der Zeilennummer steht ja entweder ein Befehl (also ein Token) oder aber eine Variable — von Blanks und Fehlern einmal abgesehen. Um den Variablennamen zu bekommen, werden alle Zeichen gelesen, bis hin zu dem Wert bei dem Bit 7 in einem Byte gesetzt ist. Ganz so einfach ist es dann aber doch nicht. Vor jedem Variablennamen stehen nämlich noch drei weitere Bytes. Untersuchen wir das erste. Bei »alpha« aus dem Beispiel hat es den Wert 0Dhex, bei »index%« den Wert 02hex, bei »mes\$« den Wert 03hex und Sie vermuten richtig, daß hierdurch der Typ der Variablen festgelegt ist. Also bedeutet:

- 0Dhex eine Variable ohne Typen-kennung
- 04hex eine Real-Variable
- 03hex eine String-Variable
- 02hex eine Integer-Variable

Die beiden Bytes vor dem Variablennamen haben so lange den Wert Null, bis das Programm gestartet wird. Nach »RUN« ändert sich hier nämlich etwas, wie man in Bild 2 sieht. Die Änderung hängt mit dem Aufbau der Variablen-tabelle durch den Interpreter zusammen. Für unser Beispiel ist diese in Bild 3 wiedergegeben. Die Variablen werden also in einer Tabelle mit ihren aktuellen Werten — falls es hier Zahlen sind — abgelegt. Für Integerzahlen sind dabei 2 Byte, bei Realzahlen 5 Byte reserviert. String-Variable belegen in der Tabelle 3 Byte: das erste gibt die Länge des Strings an, die beiden nächsten die Adresse, wo die Zeichenkette zu finden ist. »mes\$« beispielsweise hat die Länge 4 und die Zeichenkette steht bei 01CA hex. Zu beachten ist ferner, daß in der Variablen-tabelle auch die Typenkennung — jeweils um eins vermindert — mit abgelegt ist.

Im RAM des CPC steht bei AE85hex der Zeiger auf diese Varia-

```

23 00 0A 00 0D 00 00 61 6C 70 68 E1 EF 1F 8F C2 F5 30 82 01 0D 00 00 58 B1 EF 0D 00 00 61 6C 70 68 E1 00
# . . . . . a l p h a o . B u 0 . . . . . X l o . . . . . a l p h a .
18 00 14 00 04 00 00 67 61 6D 6D E1 EF 0D 00 00 61 6C 70 68 E1 F7 10 00
. . . . . g a m m a o . . . . . a l p h a w . .
13 00 1E 00 0D 00 00 00 F8 EF FF 07 28 0D 00 00 58 B1 29 00
. . . . . ) . x o . ( . . . . . X l ) .
12 00 28 00 03 00 00 6D 65 F3 EF 22 6F 2E 68 2E 22 00
. . ( . . . . . m e s o " o . k . " .
18 00 32 00 02 00 00 69 6E 64 65 F8 EF 02 00 00 69 6E 64 65 F8 F4 0F 00
. . 2 . . . . . i n d e x o . . . . . i n d e r t . .
20 00 3C 00 0D 00 00 61 72 72 61 F9 28 02 00 00 69 6E 64 65 F8 29 EF 0D 00 00 67 61 6D 6D E1 00
. < . . . . . a r r a y ( . . . . . i n d e r ) o . . . . . g a m m a .
18 00 46 00 A1 20 02 00 00 69 6E 64 65 F8 EF 19 0A 20 EB 20 1E 64 00 00
. . F . ! . . . . . i n d e r o . . . . . k . . d . .
18 00 50 00 0D 00 00 61 6C 70 68 E1 EF 04 00 00 67 61 6D 6D E1 F6 10 00
. . P . . . . . a l p h a o . . . . . g a m m a v . .
0A 00 5A 00 A0 20 1E 0A 00 00
. . Z . . . . .
14 00 64 00 BF 22 41 6C 6C 65 73 20 55 6E 73 69 6E 6E 22 00
. . d . ? " A l l e s U n s i n n " .
    
```

Bild 1. So sieht das Beispiel aus Listing 1 aus der Sicht des Interpreters aus

blentabelle (in unserem Fall ist dies 0258hex). Die zwei Byte vor »alpha« in der ersten Zeile von Bild 2 als hexadezimale Zahl interpretiert, ergeben 09hex. Diesen Wert addieren wir zum Tabellenanfang (0258hex) und erhalten die Zahl 0261hex. Ziehen wir 1 ab, so haben wir die Adresse, an der der aktuelle Wert der Variablen in der Variablen-tabelle steht (0261hex - 1). Damit ist das Geheimnis der zwei Byte vor einem Variablen-namen gelöst: Hier steht die Zahl (Offset), die, zum Tabellenanfang addiert, die Adresse für den aktuellen Wert einer Variablen ergibt. Falls dieser Offset Null ist, »weiß« der Interpreter, daß die Variable noch nicht in der Tabelle angelegt ist. Damit ist auch die Wirkung des »CLEAR«-Befehls beim CPC klar. Es wird nämlich einfach der Offset vor jeder Variablen innerhalb einer Interpreterzeile gelöscht. Diese Tatsache wollen wir für »Cref« ausnutzen.

Untersuchen wir den Aufbau der Variablen-tabelle, so stellen wir fest, daß sie nicht sortiert abgelegt wird. Betrachten wir also wieder die zwei Byte vor dem Variablen-namen in der Tabelle (Bild 3). Bei fast allen Variablen steht der Wert Null. OE hex (bei X) addiert zum Tabellenanfang 0258hex ergibt 0266hex und bei 0267hex steht X1. Die Vermutung, daß OEhex als Offset zu verstehen ist, ist richtig. Mit den zwei Byte vor dem Variablen-namen in der Tabelle baut der Interpreter für die Namen mit gleichem Anfangsbuchstaben eine verkettete Liste auf. In den Adressen ADD0hex bis AE03hex legt er den Offset für den ersten Eintrag eines Variablen-namens entsprechend dem Anfangsbuchstaben ab.

Taucht beim Bearbeiten eines Programms nun eine andere Variable mit dem gleichen Anfangsbuchstaben auf, so trägt er deren Namen am Ende der Variablen-tabelle ein. Dann wird anhand der Einträge in den Adressen bei ADD0hex bis AE03hex überprüft, ob es schon eine Variable mit diesem Anfangsbuchstaben gibt. Falls nicht, wird der Offset des zuletzt eingetragenen Namens entsprechend dem Anfangsbuchstaben in einer Adresse zwischen ADD0hex und AE03hex abgelegt. Falls ja, trägt er den Offset des vorher eingetragenen Namens in den zwei Byte vor dem letzten Eintrag ein und legt den Offset der zuletzt eingetragenen Variablen zwischen ADD0hex und AE03hex neu ab. In der Variablen-tabelle entsteht so eine Kette für Variable mit gemeinsamen Anfangsbuch-

```

0258 00 00 41 4C 50 48 C1 04 ..ALPHA.
0260 8F C2 F5 30 82 00 00 58 .Bu0...X
0268 91 04 8F C2 F5 30 82 00 ...Bu0..
0270 00 47 41 4D 4D C1 04 8F .GAMMA..
0278 C2 F5 30 81 0E 00 08 04 Bu0...X.
0280 3A 8C 10 7E 84 00 00 4D :.B...M
0288 45 D3 02 04 CA 01 00 00 ES..J...
0290 49 4E 44 45 D8 01 0A 00 INDEX...
0298 00 00 41 52 52 41 D9 04 ..ARRAY.
02A0 3A 00 01 0B 00 00 00 00 :.....
02A8 00 00 8F C2 F5 30 81 8F ...Bu0..
02B0 C2 F5 30 81 8F C2 F5 30 Bu0..Bu0
02B8 81 8F C2 F5 30 81 8F C2 ..Bu0..B
02C0 F5 30 81 8F C2 F5 30 81 u0..Bu0.
02C8 8F C2 F5 30 81 8F C2 F5 .Bu0..Bu
02D0 30 81 8F C2 F5 30 81 8F 0..Bu0..
02D8 C2 F5 30 81 96 00 A8 22 Bu0...("
02E0 6D 6F 6E 61 22 2C 1A 30 mona*..0
02E8 75 00 19 00 A0 00 BF 22 u... .?"
02F0 4C 6F 61 64 69 6E 67 20 Loading
    
```

Bild 3. Die Variablen-tabelle zu dem Beispielprogramm

```

10 alpha=2.765:X1=alpha
20 gamma!=alpha/2
30 x=EXP(X1)
40 mes$="o.k."
50 index%=index%+1
60 array(index%)=gamma
70 IF index%=10 THEN 100
80 alpha=gamma!*2
90 GOTO 10
100 PRINT"Alles Unsinn"
    
```

Listing 1. Wo steht welche Variable?

```

0170 23 00 0A 00 0D 09 00 61 6C 70 68 E1 EF 1F 8F C2 F5 30 82 01 0D 13 00 58 B1 EF 0D 09 00 61 6C 70 68 E1 00
# . . . . . a l p h a o . B u 0 . . . . . X l o . . . . . a l p h a .
0193 18 00 14 00 04 20 00 67 61 6D 6D E1 EF 0D 09 00 61 6C 70 68 E1 F7 10 00
. . . . . g a m m a o . . . . . a l p h a w . .
01AB 13 00 1E 00 0D 29 00 F8 EF FF 07 28 0D 13 00 58 B1 29 00
. . . . . ) . x o . ( . . . . . X l ) .
01BE 12 00 28 00 03 34 00 6D 65 F3 EF 22 6F 2E 68 2E 22 00
. . ( . . . . . m e s o " o . k . " .
01D0 18 00 32 00 02 3F 00 69 6E 64 65 F8 EF 02 3F 00 69 6E 64 65 F8 F4 0F 00
. . 2 . . . . . i n d e x o . ? . i n d e r t . .
01E8 20 00 3C 00 0D 08 00 61 72 72 61 F9 28 02 3F 00 69 6E 64 65 F8 29 EF 0D 20 00 67 61 6D 6D E1 00
. < . . . . . a r r a y ( . ? . i n d e r ) o . . . . . g a m m a .
0208 18 00 46 00 A1 20 02 3F 00 69 6E 64 65 F8 EF 19 0A 20 EB 20 1D 41 02 00
. . F . ! . . . . . i n d e r o . . . . . k . . d . .
0220 18 00 50 00 0D 09 00 61 6C 70 68 E1 EF 04 20 00 67 61 6D 6D E1 F6 10 00
. . P . . . . . a l p h a o . . . . . g a m m a v . .
0238 0A 00 5A 00 A0 20 1D 6F 01 00
. . Z . . . . .
0242 14 00 64 00 BF 22 41 6C 6C 65 73 20 55 6E 73 69 6E 6E 22 00
. . d . ? " A l l e s U n s i n n " .
    
```

Bild 2. Nach der Anweisung »RUN« verändern sich die Zeiger auf die Variablen

staben. Deren erstes Element findet man über den Offset aus der dem Anfangsbuchstaben entsprechenden Adresse zwischen ADD0hex und AE03hex. Das letzte Element dieser Kette ist durch den Offset Null gekennzeichnet.

Daß die Variablen »alpha« und »array« nicht verkettet sind (bei beiden ist der Offset in der Variablen-tabelle nach Bild 3 Null), liegt daran, daß »alpha« eine skalare Variable ist, »array« dagegen eine indizierte. Für skalare und indizierte Variablen werden nämlich zwei verschiedene Tabellen erstellt. Den Aufbau der Array-Tabelle (für indizierte Variablen) brauchen wir aber nicht näher untersuchen, da bei »Cref« skalare und indizierte Variable gleich behandelt werden.

Wo stehen die Sprungadressen?

Zu einer vernünftigen Cross-Referenz-Liste gehört auch die Ausgabe von Sprungadressen. Wie werden diese nun vom Interpreter des Schneiders abgelegt? Schauen wir uns dazu erst Bild 1 an. Bei »GOTO 10« (Zeile 90) und »THEN 100« (Zeile 70) werden die Zeilennummern hexadecimal abgelegt. Zuvor wird noch die Kennung 1Ehex eingetragen. Nach »RUN« ändert sich diese Kennung zu 1Dhex (Bild 2) und die Zeilennummer ist verschwunden. Der Interpreter wandelt nämlich die Zeilennummern in Adressen der Zeile um (ein Grund für die Geschwindigkeit des CPC-Basics). Erkennt der Interpreter die Kennung 1Ehex, so sucht er die Anfangsadresse der entsprechenden Zeile, ersetzt die Nummer durch die gefundene Adresse und ändert die Kennung in 1Dhex. Beim nächsten Mal hat er dann die Sprungadresse direkt zur Verfügung.

Wir haben nun genug »Interpreter-Rüstzeug« und können mit der Programmbeschreibung von »Cref« beginnen. Der erste Programmteil (ab A000hex) dient dem Einbinden von »Cref« als RSX-Befehls-erweiterung. Nach Laden des Binärfiles (ab A000hex) — vorher das zu untersuchende Basic-Programm mit »MEMORY &9FFF« schützen — und »CALL &A000« kann die Routine mit »CREF« aufgerufen werden.

Da wir mit »Cref« auf Routinen des Interpreters zugreifen, muß zuerst das entsprechende ROM freigegeben werden. Der Ausgabekanal wird auf den Drucker gegeben bevor in der Routine GVAR die Ausga-

be der Referenzen für die Variablen vorbereitet wird. Hierzu muß zunächst die Variablen-tabelle gelöscht werden. Von PSTART (AE81hex) wird die Adresse des Programm-anfangs in das Register HL geladen. Die Routine NXLNE liest die Zeilenlänge in das Register BC und speichert in ZNR die Zeilennummer. Ist die Länge einer Zeile Null, so ist das Programmende erreicht. Ansonsten wird die Interpreter-Routine NXTELM aufgerufen, die in das Register A ein Token zurückgibt und HL entsprechend erhöht. Je nach Inhalt von A wird dann in der Zeile weiter nach einem Token gesucht, die nächste Zeile geholt oder überprüft ob eine Variable vorliegt. Letzteres erledigen die Routinen VARTST und TSTTYP. Handelt es sich um eine Variable, so wird mit TABEXT durch Aufruf entsprechender Routinen deren Name in die Tabelle eingetragen.

Auf einen wichtigen Punkt sei in diesem Zusammenhang noch hingewiesen. Wie schon erwähnt, verwaltet der Interpreter des CPC für skalare und indizierte Variablen zwei verschiedene Tabellen. Um nicht beide hintereinander bearbeiten zu müssen und den Programmieraufwand in Grenzen zu halten, werden durch den Aufruf von TABEXT die Variablen-namen nur in die Tabelle für skalare Variablen eingetragen. Wir haben also keinen Unterschied zwischen skalaren und indizierten Variablen (vergleiche hierzu Bild 4, das die Variablen-tabelle für unser Beispiel nach Aufruf von »Cref« wiedergibt). Im Gegensatz zu Bild 3 sind hier die Variablen »array« und »alpha« verkettet. Dies wirkt sich aber nur bei Programmen aus, in denen beispielsweise »a\$«, »a%« und »a!« gleichzeitig verwendet werden.

Eine weitere Einschränkung ist bei der Bearbeitung von benutzereigenen Funktionen zu machen. Die mit DEF FN definierten Funktionen werden mit dem Token E4hex gekennzeichnet, ansonsten aber wie Variablen behandelt. Da »Cref« das

Token E4hex nicht berücksichtigt, erscheinen die Funktionen in der Referenzliste als Variablen-namen.

Nachdem in GVAR das Programmende erkannt wurde, folgt der Aufruf der Routine REFVAR. Diese besteht aus einer Schleife, in der für jeden Buchstaben des Alphabets (der entsprechende ASCII-Wert steht in Register C) folgende Routinen aufgerufen werden:

- BUFINI: setzt den Zeiger BUFPTR auf den Anfang des Buffers,
- FINDVA: überprüft, ob für den jeweiligen Buchstaben in der Variablen-tabelle ein Eintrag vorliegt,
- MRKEND: markiert das Ende des Buffers,
- VAROUT: gibt die entsprechende Referenzliste aus.

Um in der Variablen-tabelle einen Eintrag zu überprüfen, wird die Routine TABADR benutzt. Diese gibt in Register HL eine Adresse zurück, an

```

0258 00 00 41 4C 50 48 C1 04 ..ALPHA.
0260 00 00 00 00 00 00 00 58 .....X
0268 91 04 00 00 00 00 00 .....
0270 00 47 41 4D 4D C1 04 00 .GAMMA..
0278 00 00 00 00 0E 00 D8 04 .....X.
0280 00 00 00 00 00 00 00 4D .....M
0288 45 D3 02 00 00 00 00 ES.....
0290 49 4E 44 45 D8 01 00 00 INDEX...
0298 01 00 41 52 52 41 D9 04 ..ARRAY.
02A0 00 00 00 00 00 00 00 67 .....g
02A8 02 00 00 01 0B 00 00 .....
02B0 00 00 00 8F C2 F5 30 81 ....Bu0.
02B8 8F C2 F5 30 81 8F C2 F5 .Bu0..Bu
02C0 30 81 8F C2 F5 30 81 8F 0..Bu0..
02C8 C2 F5 30 81 8F C2 F5 30 Bu0..Bu0
02D0 81 8F C2 F5 30 81 8F C2 ..Bu0..B
02D8 F5 30 81 8F C2 F5 30 81 u0..Bu0.
02E0 8F C2 F5 30 81 2C 1A 30 .Bu0..0
02E8 75 00 19 00 A0 00 BF 22 u... ?"
02F0 4C 6F 61 64 69 6E 67 20 Loading
    
```

Bild 4. Die »Cref«-Routine verändert auch die Variablen-tabelle

Cross-Reference CPC 464 Vers. 2.0 Page 1

ALPHA!	10	20	80
ARRAY!	60		
GAMMA!	20	60	80
INDEX%	50	60	70
MES\$	40		
X!	30		
X1!	10	30	

Bild 5. Die Cross-Referenz-Tabelle des Beispielprogramms


```

10 *H * ICREF - Cross-Reference CPC 464 *
20 ;*****
30 ;** Befehlsweiterung ICREF fuer den CPC 464 **
40 ;** (c) 1985 **
50 ;** Dipl.-Math. Horst Udo Hanenberg **
60 ;** Am Muehlenteich 20, 5010 Bergheim 12 - Glessen **
70 ;** Tel. 0 22 38 / 4 16 85 **
80 ;*****
90 ;** MEMORY %9FFF Code schuetzen **
100 ;** CALL %A000 Einbinden in BASIC **
110 ;** ICREF Aufruf von BASIC **
120 ;*****
130
140 ; System-Routinen und -Vektoren
150
B08D 160 BUFEND: EQU #B08D ; Beginn Stringbereich
AE89 170 BUFSTA: EQU #AE89 ; Ende Variablen
C18C 180 CLEAR: EQU #C18C ; Variablen loeschen
EE79 190 INTOUT: EQU #EE79 ; Ausgabe Integer
FFB8 200 HLEQDE: EQU #FFB8 ; (HL) = (DE) ?
C3A8 210 LFEED: EQU #C3A8 ; Zeilenvorschub
BCD1 220 LOGEXT: EQU #BCD1 ; Einbinden RSX
F73E 230 MEMFUL: EQU #F73E ; "MEMORY FULL"
E943 240 NXTLNI: EQU #E943 ; naechstes TOKEN lesen
AC23 250 PPOS1: EQU #AC23 ; Druckerposition
C356 260 PRTCHR: EQU #C356 ; Zeichen an Drucker
AE81 270 PSTART: EQU #AE81 ; Programmstart
B90F 280 SELROM: EQU #B90F ; Freischalten ROM
AC21 290 STREAM: EQU #AC21 ; Ausgabe-Kanal
C341 300 STROUT: EQU #C341 ; Text ausgeben
D5DB 310 TABADR: EQU #D5DB ; Adr. Var. berechnen
B0C2 320 VARADR: EQU #B0C2 ; Variablenadresse
AE85 330 VARSTA: EQU #AE85 ; Variablenstart
B0C1 340 VARTYP: EQU #B0C1 ; Typenennung
AC24 350 WIDTH: EQU #AC24 ; Zeilenbreite Drucker
360
A000 370 ORG #A000
380
390 ; Einbinden in BASIC
400
A000 0109A0 410 LD BC,BEFTAB ; Adresse neuer Befehl
A003 2113A0 420 LD HL,KERNAL ; 4 Bytes fuer's System
A006 C3D1BC 430 JP LOGEXT ; Routine einbinden
440
A009 0EA0 450 BEFTAB: DEFV TABLE ; Befehlswort "CREP"
A00B C317A0 460 JP MAIN
470
A00E 435245 480 TABLE: DEFM "CRE"
A011 C6 490 DEFB "F"+#80
A012 00 500 DEFB 0 ; Ende Befehl
510
A013 520 KERNAL: DEFS 4 ; fuer den Kern
530
540 ; Hauptroutine
550
A017 0E00 560 MAIN: LD C,0 ; BASIC-ROM
A019 CD0FB9 570 CALL SELROM ; einblenden
A01C 3E08 580 LD A,B ; Drucker als
A01E 3221AC 590 LD (STREAM),A ; Ausgabekanal
A021 CD27A0 600 CALL $VAR
A024 C3B9A1 610 JP $LNR
620
630 ; Variablen - Referenzen
640
A027 CD9CC1 650 $VAR: CALL CLEAR ; Variablen loeschen
A02A 2A81AE 660 LD HL,(PSTART) ; Programmstart nach HL
A02D CD65A2 670 $VAR1: CALL NXTLINE ; naechste Zeile holen
A030 2812 680 JR Z,$VAR3 ; Null -> Programmende
A032 CD43E9 690 $VAR2: CALL NXTLNI ; naechstes TOKEN lesen
A035 FE0E 700 CP #0E ; interessant sind nur
A037 30F9 710 JR NC,$VAR2 ; TOKEN kleiner #0E
A039 FE01 720 CP #01 ; "1"
A03B 38F0 730 JR C,$VAR1 ; "1"
A03D 29F3 740 JR Z,$VAR2 ; Null = Zeilenende
A03F CD50A0 750 CALL VARTST ; Variable ?
A042 18EE 760 JR $VAR2 ; Variable ?
A044 210000 770 $VAR3: LD HL,#0000 ; Seitenzahl
A047 221CA3 780 LD (PAGE),HL ; auf Null
A04A CD6DA2 790 CALL FFEED1 ; 1. Zeile drucken
A04D C3B3A0 800 JP REFVAR ; Referenzen ausgeben
810
820 ; Test auf Variable
830
A050 E5 840 VARTST: PUSH HL
A051 2B 850 VRTST1: DEC HL ; bis zum
A052 BE 860 CP (HL) ; TOKEN
A053 20FC 870 JR NZ,VRTST1 ; lesen
A055 22C2B0 880 LD (VARADR),HL
A058 CD6CA0 890 CALL TSTTYP ; Variable ?
A05B E1 900 POP HL
A05C C0 910 RET NZ ; nein -> zurueck
A05D E5 920 PUSH HL
A05E 2AC2B0 930 LD HL,(VARADR) ; Variable in
A061 CD66A0 940 CALL TABEXT ; Tabelle eintragen
A064 E1 950 POP HL
A065 C9 960 RET
970
980 ; Variable in Tabelle eintragen
990
A066 CD06D9 1000 TABEXT: CALL #D906
A069 C3B8D6 1010 JP #D6B8
1020
1030 ; Variablentyp pruefen (= 0D97FH CPC-ROM)
1040
A06C FE0B 1050 TSTTYP: CP #0B ; kleiner #0B ?
A06E 3902 1060 JR C,TSTTP1 ; ja
A070 C6F7 1070 ADD A,#F7 ; #F7 + #0D = #04
A072 FE04 1080 TSTTP1: CP #04 ; Real-Variable ?
A074 2806 1090 JR Z,TSTTP2 ; ja
A076 D0 1100 RET NC ; keine Variable
A077 FE02 1110 CP #02 ; Integer- oder
A079 3003 1120 JR NC,TSTTP3 ; String-Variable ?
A07B C9 1130 RET
A07C 3E05 1140 TSTTP2: LD A,#05 ; ja
A07E 32C1B0 1150 TSTTP3: LD (VARTYP),A ; Typ merken
A081 BF 1160 CP A ; ZERO-Flag setzen
A082 C9 1170 RET
1180
1190 ; Referenzen fuer Variable ausgeben
1200
A083 01411A 1210 REFVAR: LD BC,#1A41 ; 1A=26, 41=ASCII "A"

```

Listing 3. Das Assembler-Listing für die Cross-Reference

Grundlagen

A141	F1	2430	POP AF						
A142	04	2440	INC B						
A143	BE	2450	CP (HL)		: Laenge plus 1				
A144	23	2460	INC HL		: letztes Zeichen ?				
A145	30ED	2470	JR NC,VROUT1		: naechstes Zeichen				
A147	7E	2480	LD A,(HL)		: Variablentyp				
A148	E5	2490	PUSH HL		: drucken				
A149	210BA3	2500	LD HL,TYPE-1						
A14C	85	2510	ADD A,L						
A14D	4F	2520	LD L,A						
A14E	7E	2530	LD A,(HL)						
A14F	CD56C3	2540	CALL PRTOCHR						
A152	E1	2550	POP HL						
A153	2B	2560	DEC HL						
A154	2214A3	2570	LD (VARAKT),HL		: aktuelle Variable				
A157	78	2580	LD A,B		: und				
A158	3216A3	2590	LD (VARLEN),A		: Laenge merken				
A15B	CD62A1	2600	CALL REFOUT		: Referenzen ausgeben				
A15E	DDE1	2610	POP IX		: naechste Variable				
A160	18BE	2620	JR VAROUT		: drucken				
		2630							
		2640			: Programm nach Variablenamen durchsuchen				
		2650							
A162	2A81AE	2660	REFOUT: LD HL,(PSTART)		: Programmstart nach HL				
A165	CD65A2	2670	RFOUT1: CALL NXTLINE		: naechste Zeile holen				
A168	C8	2680	RET Z		: Null -> Prg.ende				
A169	0B	2690	DEC BC		: Laenge minus 4				
A16A	0B	2700	DEC BC						
A16B	0B	2710	DEC BC						
A16C	0B	2720	DEC BC						
A16D	09	2730	ADD HL,BC		: Ende Zeile				
A14E	E5	2740	PUSH HL						
A16F	EB	2750	EX DE,HL		: nach DE				
A170	CD76A1	2760	CALL INLINE		: Zeile durchsuchen				
A173	E1	2770	POP HL						
A174	18EF	2780	JR RFOUT1						
		2790							
		2800			: Zeile nach Variablenamen durchsuchen				
		2810							
A176	2A14A3	2820	INLINE: LD HL,(VARAKT)		: Adresse akt. Var.				
A179	3A16A3	2830	LD A,(VARLEN)		: Laenge				
A17C	3217A3	2840	LD (VST),A		: merken				
A17F	1A	2850	INLN1: LD A,(DE)		: Zeichen aus Zeile				
A180	E6DF	2860	AND #DF		: Klein- in Grossbuchst.				
A182	BE	2870	CP (HL)		: Zeichen gleich ?				
A183	2810	2880	JR Z,INLN3						
A185	0B	2890	DEC BC						
A186	1B	2900	DEC DE						
A187	78	2910	LD A,B		: Zeile				
A188	B9	2920	CP C		: fertig ?				
A189	C8	2930	RET Z						
A18A	18F3	2940	JR INLN1						
A18C	2B	2950	DEC HL						
A18D	0B	2960	DEC BC						
A18E	1B	2970	DEC DE						
A18F	1A	2980	LD A,(DE)		: naechstes Zeichen				
A190	E6DF	2990	AND #DF						
A192	BE	3000	CP (HL)		: auch gleich ?				
A193	20E1	3010	JR NZ,INLINE		: nein -> neu anfangen				
A195	3A17A3	3020	INLN3: LD A,(VST)		: Variable				
A198	3D	3030	DEC A		: fertig ?				
A199	2808	3040	JR Z,INLN4						
A19B	3217A3	3050	LD (VST),A						
A19E	78	3060	LD A,B		: Zeile				
A19F	B1	3070	OR C		: fertig ?				
A1A0	C8	3080	RET Z						
A1A1	18E9	3090	JR INLN2						
A1A3	1B	3100	DEC DE		: (DE) zeigt auf				
A1A4	1B	3110	DEC DE		: den Typ				
A1A5	1B	3120	DEC DE		: der Variablen				
A1A6	1A	3130	LD A,(DE)		: ist's eine				
A1A7	CD6CA0	3140	CALL TSTTYP		: Variable ?				
A1AA	20CA	3150	JR NZ,INLINE						
A1AC	CD3A2	3160	CALL TAB		: naechster Tabulator				
A1AF	CD0DA2	3170	CALL TABOK		: Tabulator o.k.?				
A1B2	2A12A3	3180	LD HL,(ZNR)		: Zeilennummer				
A1B5	CD79EE	3190	CALL INTOUT		: drucken				
A1B8	C9	3200	RET						
		3210							
		3220			: Zeilen - Referenzen				
		3230							
A1B9	CD8CA2	3240	GLNR1: CALL BUFINI						
A1BC	2A81AE	3250	LD HL,(PSTART)		: Programmstart nach HL				
A1BF	CD65A2	3260	GLNR1: CALL NXTLINE		: naechste Prg.zelle				
A1C2	2827	3270	JR Z,GLNR4		: Null=Programmende				
A1C4	CD43E9	3280	GLNR2: CALL NXTELM		: naechstes TOKEN				
A1C7	FE01	3290	CP #01		: "1"				
A1C9	38F4	3300	JR C,GLNR1		: neue Zeile				
A1CB	28F7	3310	JR Z,GLNR2		: neues TOKEN				
A1CD	FE1E	3320	CP #1E		: Zeilennummer ?				
A1CF	2007	3330	JR NZ,GLNR3						
A1D1	E5	3340	PUSH HL						
A1D2	CD4A1	3350	CALL ZNRSTO		: eintragen				
A1D5	E1	3360	POP HL						
A1D6	18EC	3370	JR GLNR2						
A1D8	FE1D	3380	GLNR3: CP #1D		: Zeilenadresse ?				
A1DA	20E8	3390	JR NZ,GLNR2						
A1DC	E5	3400	PUSH HL						
A1DD	56	3410	LD D,(HL)		: Zeilennummer				
A1DE	2B	3420	DEC HL		: holen				
A1DF	5E	3430	LD E,(HL)						
A1E0	EB	3440	EX DE,HL						
A1E1	23	3450	INC HL						
A1E2	23	3460	INC HL						
A1E3	23	3470	INC HL						
A1E4	23	3480	INC HL						
A1E5	CD4A1	3490	CALL ZNRSTO		: eintragen				
A1E8	E1	3500	POP HL						
A1E9	18D9	3510	JR GLNR2						
A1EB	CD8A2	3520	GLNR4: CALL FFEED		: Seitenvorschub				
A1EE	CD27A2	3530	CALL REFZNR		: Referenzen ausgeben				
A1F1	C3A8C3	3540	JP LFEED						
		3550							
		3560			: Zeilennummer mit Referenz-Zeile sortiert eintragen				
		3570							
A1F4	7E	3580	ZNRSTO: LD A,(HL)		: Zeilennummer				
A1F5	3211A3	3590	LD (HZNR+1),A						
A1F8	2B	3600	DEC HL		: eintragen				
A1F9	7E	3610	LD A,(HL)						
A1FA	3210A3	3620	LD (HZNR),A						
A1FD	CD9AA2	3630	CALL NXTPTR		: noch Platz ?				
A200	E5	3640	PUSH HL						
A201	EB	3650	EX DE,HL						
A202	2110A3	3660	LD HL,HZNR		: Referenz und				
A205	010400	3670	LD BC,#0004		: Zeilennummer				
A208	EDB0	3680	LDIR		: in den Buffer				
A20A	EB	3690	EX DE,HL		: eintragen				
A20B	2219A3	3700	LD (BUFFTR),HL		: Zeiger merken				
A20E	E1	3710	POP HL						
A20F	7E	3720	ZNRST1: LD A,(HL)		: schon letzter				
A210	23	3730	INC HL		: Eintrag ?				
A211	B6	3740	OR (HL)						
A212	2B	3750	DEC HL						
A213	C8	3760	RET Z						
A214	E5	3770	PUSH HL		: DE auf letzten				
A215	D1	3780	POP DE		: Eintrag und				
A216	2B	3790	DEC HL		: HL auf				
A217	28	3800	DEC HL		: vorletzten				
A218	2B	3810	DEC HL		: Eintrag				
A219	2B	3820	DEC HL		: setzen				
A21A	CD77A2	3830	CALL NRCOMP		: vergleichen				
A21D	D0	3840	RET NC		: kann so bleiben				
A21E	E5	3850	PUSH HL						
A21F	0604	3860	LD B,#04		: Eintragungen				
A221	CD81A2	3870	CALL STLHDE		: vertauschen				
A224	E1	3880	POP HL						
A225	18E8	3890	JR ZNRST1						
		3910			: Referenzen fuer Zeilennummern ausdrucken				
		3920							
A227	CDACA2	3930	REFZNR: CALL MRKEND		: Buffer-Ende markieren				
A22A	AF	3940	XOR A						
A22B	57	3950	LD D,A		: (DE) = 0				
A22C	5F	3960	LD E,A						
A22D	2A89AE	3970	LD HL,(BUFSTA)		: HL zeigt				
A230	23	3980	INC HL		: auf den ersten				
A231	23	3990	INC HL		: Eintrag				
A232	7E	4000	RFZNR1: LD A,(HL)		: falls				
A233	23	4010	INC HL		: Zeilennummer				
A234	47	4020	LD B,A		: gleich				
A235	B6	4030	OR (HL)		: Null,				
A236	C8	4040	RET Z		: -> fertig				
A237	78	4050	LD A,B						
A238									

```

4900 ; Naechster Zeiger fuer Buffer o.k.?
4910
A29A 2A19A3 4920 NXTPTR: LD HL, (BUFFTR) ; Pointer holen
A29D E5 4930 PUSH HL
A29E 23 4940 INC HL ; plus 2
A29F 23 4950 INC HL
A2A0 ED5B8DB0 4960 LD DE, (BUFEND) ; Anfang Stringber.
A2A4 CDB6FF 4970 CALL HLEODE ; vergleichen
A2A7 E1 4980 POP HL
A2A8 D23EF7 4990 JP NC, MEMFUL ; "MEMORY FULL"
A2AB C9 5000 RET
5010
5020 ; Ende Buffer markieren
5030
A2AC CD9AA2 5040 MRKEND: CALL NXTPTR ; noch Platz ?
A2AF AF 5050 XOR A ; zweimal
A2B0 77 5060 LD (HL), A ; Null
A2B1 23 5070 INC HL ; eintragen
A2B2 77 5080 LD (HL), A
A2B3 C9 5090 RET
5100
5110 ; Zeilenvorschub
5120
A2B4 CDA9C3 5130 CRLF: CALL LFEED
A2B7 3A1BA3 5140 LD A, (LINE) ; Zeilenzaehler erhoehen
A2BA 3C 5150 INC A
A2BB 321BA3 5160 LD (LINE), A
A2BE FE39 5170 CP 56 ; mehr als 56 Zeilen ?
A2C0 3026 5180 JR NC, FFEED ; Seitenvorschub
A2C2 C9 5190 RET
5200
5210 ; naechster Tabulator
5220
A2C3 3E20 5230 TAB: LD A, " "
A2C5 CD56C3 5240 CALL FRTCHR
A2C8 3A23AC 5250 LD A, (PPOS) ; Druckerposition
A2CB C5 5260 PUSH BC ; grosser
A2CC 47 5270 LD B, A ; als
A2CD 3A24AC 5280 LD A, (WIDTH) ; Zeilenbreite
A2D0 D607 5290 SUB 7 ; minus 7 ?
A2D2 B8 5300 CP B
A2D3 78 5310 LD A, B
A2D4 C1 5320 POP BC
A2D5 DAB4A2 5330 JP C, CRLF ; ja --> Zeilenvorschub
A2D8 E607 5340 AND #07 ; TAB-Position ?
A2DA 20E7 5350 JR NZ, TAB
A2DC C9 5360 RET
5370
5380 ; Druckerposition auf 24 setzen, falls kleiner
5390
A2DD 3A23AC 5400 TABOK: LD A, (PPOS) ; Druckerposition
A2E0 FE14 5410 CP #14 ; grosser 20 ?
A2E2 D0 5420 RET NC
A2E3 CDC3A2 5430 CALL TAB
A2E6 18F5 5440 JR TABOK
5450
5460 ; Seitenvorschub
5470
A2E8 3E0C 5480 FFEED: LD A, #0C ; ASCII Seitenvorschub
A2EA CD56C3 5490 CALL FRTCHR

```

```

A2ED E5 5500 FFEED1: PUSH HL
A2EE 2A1CA3 5510 LD HL, (PAGE) ; Seitenzaehler erhoehen
A2F1 23 5520 INC HL
A2F2 221CA3 5530 LD (PAGE), HL
A2F5 211EA3 5540 LD HL, HL, LINE ; Kopfzeile drucken
A2F8 CD41C3 5550 CALL STROUT
A2FB 2A1CA3 5560 LD HL, (PAGE) ; Seitenzahl drucken
A2FE CD79EE 5570 CALL INTOUT
A301 3E01 5580 LD A, 1 ; Zeilenzaehler
A303 321BA3 5590 LD (LINE), A ; auf eins
A306 E1 5600 POP HL
A307 CDB4A2 5610 CALL CRLF
A30A 18A8 5620 JR CRLF
5630
5640 ; Datenbereich
5650
A30C 25240021 5660 TYPE: DEFB #25, #24, #00, #21 ; "X", "S", "00", "!"
A310 5670 HZNR: DEFS 2 ; Zeilenreferenz
A312 5680 ZNR: DEFS 2 ; Zeilennummer
A314 5690 VARAKT: DEFS 2 ; Adresse Variablenname
A316 5700 VARLEN: DEFS 1 ; Laenge Variablenname
A317 5710 VST: DEFS 2 ; Hilfsspeicher
A319 5720 BUFFTR: DEFS 2 ; Pointer fuer Buffer
A31B 5730 LINE: DEFS 1 ; Druckerzeile
A31C 5740 PAGE: DEFS 2 ; Druckerseite
A31E 43726F73 5750 HLINE: DEFM "Cross-Reference CPC 464 Vers. 2.0 Page "
A349 00 5760 DEFB #00 ; Uberschrift

```

```

BEFTAB A009 BUFEND B08D BUFINI A28C
BUFFTR A319 BUFSTA AE89 CLEAR C18C
COMP A103 COMP1 A105 COMP2 A111
COMP3 A119 CRLF A284 FFEED A2E8
FFFEED1 A2ED FINDVA A0A5 FNDV1 A0A8
GLNR A1B9 GLNR1 A1BF GLNR2 A1C4
GLNR3 A1D8 GLNR4 A1EB GVAR A027
GVAR1 A02D GVAR2 A032 GVAR3 A044
HLEODE FFB8 HLINE A31E HZNR A310
INLINE A176 INLN1 A17F INLN2 A18C
INLN3 A195 INLN4 A1A3 INLN5 A1AC
INTOUT EE79 KERNAL A013 LFEED C3A8
LINE A31B LOGEXT BCD1 MAIN A017
MEMFUL F73E MRKEND A2AC NRCOMP A277
NXTL E943 NXTLINE A265 NXTPTR A29A
PAGE A31C PPOS AC23 FRTCHR C356
PSTART AE81 REFOUT A162 REFVAR A083
REFZNR A227 RFOUT1 A165 RFVAR1 A086
RFZNR1 A232 RFZNR2 A240 RFZNR3 A254
SELROM B90F SORT AOCA SORT1 A0D2
STHLDE A281 STORE A0BC STREAM AC21
STROUT C341 TAB A2C3 TABADR D5DB
TABEXT A066 TABLE A00E TABOK A2DD
TSTTP1 A072 TSTTP2 A07C TSTTP3 A07E
TSTTYP A06C TYPE A30C VARADR B0C2
VARAKT A314 VARLEN A316 VAROUT A120
VARSTA AE95 VARTST A050 VARTYP B0C1
VROUT1 A134 VROUT2 A13E VRTST1 A051
VST A317 WIDTH AC24 ZNR A312
ZNRST1 A20F ZNRST0 A1F4

```

Listing 3. Assembler-Listing für die Cross-Reference (Schluß)



Das neueste deutsche Grafik-Adventure für Ihren Commodore 64

Suchen Sie die Pforte zu einer anderen Welt!
Beweisen Sie Ihren Spürsinn, denn der richtige Weg ist schwer zu finden, und überall lauern Gefahren!

- hochauflösende Grafik
- ausführliche Spielanweisungen
- riesiger Befehlsvorrat
- Eingabe von ganzen Sätzen möglich
- variabler Spielablauf

Markt & Technik
BUCHVERLAG

Hans-Pinsel-Straße 2, 8013 Haar bei München
Schweiz: Markt & Technik Vertriebs AG, Kollerstrasse 3, CH-6300 Zug, ☎ 042/41 56 56
Österreich: Rudolf Lechner & Sohn, Heizwerkstraße 10, A-1232 Wien, ☎ 0222/677526

Eine neue Dimension der Abenteuerspiele:
Kein Spiel gleicht dem anderen — Sie geraten in Situationen, in denen Sie Ihre Spieltaktik völlig ändern müssen.
Überzeugen Sie sich selbst!

Happy Software gibt's beim Buchhändler, bei Horten, Quelle und im Computershop. Bestellkarten bitte an Ihren Buchhändler oder an eine unserer Depotbuchhandlungen.
Adressenverzeichnis am Ende des Heftes!

Gordon Saga

Best.-Nr. MD 240 A
DM 39,-* (Sfr. 35,50 / öS 351,-)
WWW.HOMECOMPUTERWORLD.COM
Inkl. MwSt. Unverbindliche Preisempfehlung.

»Supermon CPC-1002« — Der Maschi- nensprache- Monitor



Jeder, der das ROM
seines Schneiders oder
Maschinencode-Routinen
untersuchen will, findet

hier ein professionelles Werkzeug
zum Abtippen. Kleine Routinen in
Z80-Assembler können natürlich
auch programmiert werden.

»Supermon CPC-1002« ist ein Monitor, der neben allgemein üblichen Funktionen (Speicherbereiche auflisten, Inhalte ändern und Maschinenprogramme starten) über einige leistungsstarke Anweisungen zum Laden, Speichern und Kopieren bestimmter Programm- und Datendateien verfügt. Viel Wert wurde auf möglichst einfache und schnelle Handhabung gelegt. Zahlreiche Befehle dienen deshalb ausschließlich der Fehlersuche in selbsterstellten Programmen. Breakpoints können gesetzt und gelöscht werden, die Inhalte der Z80-Register werden auf Tastendruck sichtbar und können gegebenenfalls vom Anwender verändert werden. Natürlich ist auch ein schneller Disassembler integriert, des weiteren lassen sich Speicherbereiche beliebig verschieben oder mit Daten belegen.

Eine Reihe von Befehlen sind an spezielle Eigenschaften der Schneider-Computer angepaßt. So kann mittels eines einzigen Kommandos zwischen verschiedenen Speicherbänken umgeschaltet werden. Auf diese Art und Weise kann man sehr einfach einen Blick ins Betriebssystem- oder Basic-ROM werfen. Auch die Analyse beliebiger Erweiterungsmodule ist möglich. Andere Befehle schalten zwischen 40- und 80-Zeichen-Darstellung um oder wählen die Aufzeichnungsgeschwindigkeit bei Kassettenbetrieb. In Erweiterungs-ROMs enthaltene externe Kommandos lassen sich ebenfalls aufrufen, was besonders bei Verwendung eines Disketten-Laufwerks von Vorteil ist. Alle DOS-Kommandos können nämlich direkt vom Monitor aus aufgerufen werden.

Der Supermon CPC-1002 ist sowohl bei reinem Kassettenbetrieb (CPC 464) als auch bei Verwendung eines Diskettenlaufwerks voll funktionsfähig. Allerdings ist der Bildaufbau speziell an den 464 angepaßt. Alle Routinen zum Laden, Speichern oder Kopieren von Dateien sowie die »CAT«-Funktion arbeiten wahlweise mit Kassetten- oder Diskettenlaufwerk zusammen.

Weitere Leistungsmerkmale des Supermon CPC-1002 sind:
— Aufruf von Maschinenprogrammen mit vorherigem Laden der Z80-Register

- Laden von Programmen in beliebige Speicherbereiche
- Anzeige aller relevanten Header-Informationen beim Laden
- Erweiterbar durch beliebig belegbaren »User Jump«
- Hexadezimale Addition und Subtraktion möglich
- Berechnung relativer Sprungdistanzen
- 7/8-Bit-ASCII-Modus wählbar
- Alle Monitor-Funktionen wahlweise auch über Drucker auszugeben.

Bevor Sie ans Ausprobieren dieses Monitors gehen, sollten Sie die folgende Einführung in das Arbeiten mit dem Supermon gelesen haben. Bitte denken Sie immer daran, daß Sie mit diesem Monitor ohne Einschränkung Ihr gesamtes Schneider-System auf einer sehr Hardware-nahen Ebene in der Hand haben. Fehlbedienungen oder Irrtümer bei der Dateneingabe können daher leicht zum »Absturz« des Systems und damit zum vollständigen Datenverlust führen.

Listing 2 ist der Basic-Lader für »Supermon CPC-1002«. Das Binärfeld wird ab Speicherzelle 8000 hex erzeugt und automatisch unter dem Namen »SMON.BIN« gespeichert. Listing 1 ist das Startprogramm und muß auf der Kasette direkt vor »SMON.BIN« stehen. Nach dem Programmstart (ohne Listing 1 mit »CALL&8000«) meldet sich der Monitor mit

** SUPERMON CPC-1002 **

Copyright © 1985 by Happy Computer

Links unter den beiden Titelzeilen erscheint das Zeichen »>«, gefolgt von einem Cursor als sogenanntes »Prompt«-Symbol, also als Zeichen, daß der Supermon bereit ist, Befehle anzunehmen. Diese Befehle bestehen durchweg aus einem Zeichen, das bei den meisten Kommandos eine leicht zu merkende Abkürzung für die volle Befehlsbezeichnung ist (zum Beispiel »L« für »Load«). Bei einigen Befehlen genügt schon die Eingabe dieses Zeichens, um eine Aktion auszulösen (die Enter-Taste ist also nicht notwendig). Die meisten Monitor-Kommandos benötigen aber zumindest eine Adresse oder weitere Parameter. Dann wird nach dem Befehlssymbol ein Leerzeichen ausgegeben, und der Cursor erscheint wieder, um anzuzeigen, daß jetzt zusätzliche Eingaben erwartet werden. Alle Zahlenangaben müssen grundsätzlich hexadezimal (ohne vorangestelltes »&«) eingegeben werden — in der Regel vierstellig, da es sich zumeist um Adressen handelt. In einigen Fällen (beim Laden und Speichern von Programmen) muß auch ein Programmname angegeben werden, der vom Monitor mit »FILENAME ?« erfragt wird. Derartige Namenseingaben müssen immer mit der Enter-Taste abgeschlossen werden, da der Computer ja nicht voraussehen kann, wie lang der Name werden wird. Im übrigen stehen bei Namenseingaben alle vom Basic her bekannten Editiermöglichkeiten zur Verfügung.

Anders verhält es sich bei den (hexadezimalen) Adressen- und Dateneingaben. Bei diesen braucht die Enter-Taste nicht extra betätigt werden, sondern der gewählte Befehl wird nach Eingabe der letzten Ziffer sofort ausgeführt. Es werden dabei nur hexadezimale Zahlen angenommen und alle anderen Eingaben ignoriert.

Falls Sie sich bei der Dateneingabe einmal vertippt haben sollten, können Sie den gewählten Befehl entweder durch Drücken von Esc oder Enter abbrechen. Der Monitor meldet in einem solchen Fall »NOT OK«, zum Zeichen, daß die Funktion nicht ordnungsgemäß durchgeführt wurde. Die gleiche Meldung erscheint, wenn ein Befehl zwar ordnungsgemäß erkannt und ausgeführt worden ist, sich aber während der Ausführung eine Fehlerbedingung ergab (beispielsweise, wenn Sie während des Ladens einer Programmdatei die Esc-Taste gedrückt haben oder wenn ein Lesefehler auftrat).

Wurde eine Funktion ordnungsgemäß zu Ende geführt, dann meldet sich der Monitor mit »OK« zurück. In beiden Fällen — also auch nach der »NOT OK«-Meldung — ist der Monitor sofort wieder bereit zur Ausführung weiterer Befehle.

Befehlsübersicht Supermon CPC-1002

A	ASCII-Dump	!	Bye (Monitor verlassen)
B	Breakpoint setzen	:	RAM/ROM-Bank wählen
C	Copy (Datei kopieren)	!	Externes Kdo. aufrufen
D	Disassembler	@	CATalog-Funktion
F	Find (Bytefolgen suchen)	#	Drucker ein
G	Go (Routine ausführen)	&	40/80 Zeichen umschalten
H	Header anzeigen	(User Jump definieren
J	Jump (Programm aufrufen))	User Jump anzeigen
K	Kill (Breakpoint löschen)	0	SPEED 0 wählen (Kassette)
L	Load (Programm laden)	1	SPEED 1 wählen (Kassette)
M	Memory Dump	.	7/8-Bit-Modus umschalten
O	On/Off (Kass.Motor ein/aus)	+	Hexadezimal addieren
R	Register anzeigen	-	Hexadezimal subtrahieren
S	Save (Programm speichern)	!	Sprungdistanz berechnen
T	Text eingeben	X	Speicherbereich verschieben
U	User Jump ausführen	Y	Yank (Progr. verschob. laden)
W	Write (Bytes schreiben)	Z	Fill (Speicherber. belegen)

Esc	Funktion abbrechen	IBASIC	Basic-Kaltstart
Enter	Monitor-Warmstart	Ctrl C	Monitor-Kaltstart

Externe DOS-Kommandos (nur bei angeschlossenem Floppy-Laufwerk)

IA	Aktuelles Laufwerk ist Laufwerk A
IB	Aktuelles Laufwerk ist Laufwerk B
IDRIVE	Aktuelles Laufwerk wählen
IDIR	Directory auflisten
IERA	Datei löschen
IREN	Datei umbenennen
IDISC	LOAD/SAVE von/auf Diskette
IDISC.IN	LOAD von Diskette
IDISC.OUT	SAVE auf Diskette
ITAPE	LOAD/SAVE von/auf Kassette
ITAPE.IN	LOAD von Kassette
ITAPE.OUT	SAVE auf Kassette
ICPM	CP/M aufrufen (danach kann mit S-MON nicht mehr gearbeitet werden)
IUSER	(Von S-MON aus nicht verwendbar)

Die Supermon-Kommandos

Befehlsgruppe 1: Allgemeine Monitor-Kommandos

Die Befehlsgruppe 1 umfaßt Kommandos zum Auflisten, Verändern und Verschieben von Speicherbereichen sowie zum Aufruf von Maschinencode-Programmen. Ferner ist die Umschaltung zwischen verschiedenen ROM/RAM-Bänken vorgesehen.

ASCII-Dump (Speicherbereich als ASCII-Zeichen listen):

A <Anfangsadresse> — Nach Eingabe einer Adresse wird der Speicherinhalt von der gewünschten Stelle an als ASCII-Zeichenfolge aufgelistet. Die Inhalte der Speicherstellen werden dabei als ASCII-Zeichen interpretiert und am Bildschirm dargestellt. In einem Maschinenprogramm enthaltene Textpassagen lassen sich damit mühelos und schnell auffinden und identifizieren.

Memory-Dump (Speicherbereich hexadezimal listen):

M <Anfangsadresse> — Nach Eingabe einer Adresse wird der Speicherinhalt von der gewünschten Stelle an in hexadezimaler Schreibweise (Hex-Dump) aufgelistet. Um Text- und Tabellen-Bereiche leichter identifizieren zu können, erscheinen am rechten Rand jeder Zeile die dargestellten Speicherzellen-Inhalte nochmals als ASCII-Zeichenfolge.

Disassembler (Speicherbereich als Z80-Mnemonics listen):

D <Anfangsadresse> — Nach Eingabe einer Adresse wird der Speicherinhalt von der gewünschten Stelle an als Maschinencode-Programm interpretiert und in Form von Z80-Mnemonics (Assemblerbefehlen) aufgelistet. Um einen mög-

lichst umfassenden Überblick zu gewährleisten, werden zusätzlich die entsprechenden Bytes in hexadezimaler Form und als ASCII-Zeichen ausgegeben.

Da Z80-Befehle einschließlich Operand bis zu 4 Byte lang sein können, kann es beim D-Befehl passieren, daß die von Ihnen eingegebene Anfangsadresse nicht mit dem ersten Byte eines Z80-Befehls übereinstimmt. In diesem Falle werden erfahrungsgemäß die ersten zwei bis drei Befehle falsch disassembliert; danach hat sich der Disassembler eingependelt und übersetzt wieder korrekt. Wenn Sie also nicht genau sagen können, ob die von Ihnen angegebene Adresse genau der Anfang eines Z80-Befehls war, dann sollten Sie die ersten paar ausgegebenen Zeilen mit Vorsicht genießen. Oft empfiehlt es sich in solchen Fällen, mit der Disassemblierung mehrere Bytes vor der eigentlich interessierenden Stelle zu beginnen, um ein verlässliches Ergebnis zu erhalten.

Besondere Aufmerksamkeit sollten Sie Datenfeldern, Tabellen und Texten innerhalb von Maschinenprogrammen widmen, da der Disassembler in derartigen Programmteilen natürlich keine brauchbaren Ergebnisse liefern kann. Der vom Disassembler erzeugte Z80-Assembler-Code ist weitgehend kompatibel zu den meisten Z80-Assemblern. Zwei Unterschiede sind jedoch anzumerken: Alle Zahlenausgaben des Disassemblers sind hexadezimal, jedoch nicht weiter gekennzeichnet. Die meisten Assembler verlangen ein nachgestelltes »H« als Kennzeichnung für hexadezimale Zahlenwerte, vereinzelt auch ein vorangestelltes »\$«, »&« oder »#«. Aus Gründen der verbesserten Übersicht werden bei relativen Sprüngen nicht die Sprungdistanzen, sondern direkt das Sprungziel ausgegeben.

Write Bytes (Speicherinhalte ändern):

W <Anfangsadresse> — Mit diesem Kommando können Speicherinhalte geändert werden. Nach Eingabe einer Adresse können beliebig viele Bytes in hexadezimaler Form eingegeben werden, die von der angegebenen Speicherstelle an abgelegt werden. Jedes Byte wird als ein Paar von Hexadezimalziffern eingegeben. Nach einem eingegebenen Byte wird automatisch ein Leerzeichen am Bildschirm ausgegeben, um die Übersichtlichkeit zu wahren.

Nach Eingabe der zweiten Ziffer eines Bytes wird dieses sofort an der entsprechenden Adresse im Speicher abgelegt und überschreibt dabei den vorherigen Speicherinhalt. Bei Abbruch der Eingabe nach dem Tippen der ersten Ziffer bleibt der Speicher unverändert. Der W-Befehl wird in erster Linie zur Änderung einzelner Bytes im Speicher oder zur Eingabe kompletter Maschinenprogramme benutzt.

Text (Text in Speicher schreiben):

T <Anfangsadresse> — Mit diesem Kommando können Sie beliebige Texte direkt in den Speicher schreiben. Auch Control-Zeichen (mit Ausnahme von Ctrl-M (= CHR\$(13) = Enter) sind erlaubt. Nach Eingabe einer Adresse können Sie den Text sofort direkt eintippen.

Beachten Sie aber bitte unbedingt, daß Sie dabei direkt in den Speicher schreiben, das heißt jedes eingetippte Zeichen wird sofort im Speicher abgelegt und überschreibt die vorher dort stehenden Informationen. Es bestehen daher auch keine Editier-Möglichkeiten außer Neuschreiben fehlerhafter Passagen. Dieses Kommando ist als Hilfe zum einfachen Einfügen und Ändern von Texten in Maschinenprogrammen gedacht.

Find (Bytefolgen suchen):

F <Anfang> <Ende> <Byte1> <Byte 2> <Byte 3> <Byte 4> — Dieses Kommando sucht in einem anzugebenden Speicherbereich nach Bytefolgen mit einer Länge von maximal vier Bytes. Dazu müssen Anfang und Ende des Suchbereichs angegeben werden sowie die bis zu vier zu suchenden Bytes. Das erste der vier zu suchenden Bytes muß naturge-

mäß in jedem Falle angegeben werden, für jedes einzelne der drei folgenden Bytes kann auf eine Spezifizierung des Wertes verzichtet werden, indem die Enter-Taste betätigt wird. Statt eines Wertes erscheint am Bildschirm für das betreffende Byte nur »XX«. Ein so gekennzeichnetes Byte spielt während des Suchvorganges die Rolle eines Jokers, das heißt es kann jeden beliebigen Ein-Byte-Wert repräsentieren.

Ein Beispiel soll das verdeutlichen: Nehmen wir an, Sie wollten wissen, an welchen Stellen ein Maschinenprogramm, das im Speicher von 2000 bis 4000 hex steht, andere Routinen im Bereich A000 bis A0FF hex aufruft. Das heißt also, Sie suchen zwischen 2000 und 4000 hex alle CALL-Befehle (Z80-Code »CD«), bei denen das High-Byte der Adresse A0 hex ist. Diese Befehle findet der Supermon für Sie in Sekundenbruchteilen durch Eingabe von »F 2000 4000 CD XX A0 XX«, wobei Sie natürlich statt »XX« jeweils die Enter-Taste drücken. Diese Eingabe findet alle Folgen aus 4 Byte zwischen 2000 und 4000 hex, die mit »CD« beginnen und deren drittes Byte »A0« ist. Nach Eingabe des vierten zu suchenden Bytes wird der angegebene Speicherbereich nach passenden Bytefolgen durchsucht. Wird eine Übereinstimmung gefunden, so wird die gefundene Bytefolge samt Adresse ausgegeben und die Suche fortgesetzt.

Jump (Maschinenprogramm aufrufen):

J < Startadresse > — Dieses Kommando dient zum Aufruf von Maschinencode-Programmen. Nach Eingabe einer Startadresse wird die an dieser Adresse stehende Routine wie ein Unterprogramm gestartet. Wenn das Programm keine Stack-Manipulation vornimmt, bewirkt ein »RET«-Befehl die Rückkehr zum Supermon CPC.

Da der J-Befehl die Kontrolle vollständig an das aufgerufene Maschinenprogramm übergibt, ist bei der Anwendung dieses Befehls besondere Sorgfalt geboten. Durch fehlerhafte oder unüberlegte Anwendung kann es leicht zum vollständigen »Absturz« des Computersystems kommen.

Im Gegensatz zum G-Befehl (GO, siehe Befehlsgruppe 2) wird der Monitor-Registerbereich durch das J-Kommando nicht verändert.

Transfer (Speicherbereich verschieben):

X < Blockanfang > < Blockende > < Zieladresse > — Dieses Kommando dient zum Verschieben von Speicherblöcken. Dazu müssen drei Adressen angegeben werden. Die ersten beiden Adressen bezeichnen dabei Anfang und Ende des zu verschiebenden Speicherbereichs, die dritte Adresse bezeichnet das Verschiebeziel, also die Adresse, zu der der Speicherbereich verschoben werden soll.

Beispiel 1: X 3000 3100 2000

Dieser Befehl kopiert den Speicherbereich von 3000 bis 3100 hex (einschließlich) in den Bereich 2000 bis 2100 hex.

Beispiel 2: X 3000 3FFF 3001

Dieser Befehl verschiebt den Speicherbereich von 3000 bis 3FFF hex (einschließlich) um ein Byte nach oben. Das X-Kommando arbeitet »intelligent«, das heißt abhängig von der Wahl der Zieladresse wird die Verschiebung stets so vorgenommen, daß der gewählte Speicherblock unverändert übertragen wird (vergleiche Beispiel 2).

Fill (Speicherbereich mit Konstante füllen):

Z < Anfang > < Ende > < Byte > — Der Speicherbereich von < Anfang > bis < Ende > wird mit einem konstanten < Byte > aufgefüllt. Besondere Vorsicht ist bei Anwendung dieses Kommandos geboten, um nicht versehentlich wichtige Speicherbereiche zu überschreiben.

Beispiel: Z C000 FFFF 00

Dieser Befehl löscht den Bildschirm, indem der gesamte Video-Speicher mit Null-Bytes gefüllt wird.

Switch Bank (Speicherbank schalten):

: < Bank-Auswahl-Adresse > — Mit diesem Kommando gibt es keine unzugänglichen Ecken mehr in Ihrem Schneider-Computer. Sie können jedes eingesetzte ROM- oder RAM-Modul anwählen, insgesamt bis zu 252 Speicherbänke. Weiterhin können Sie mit diesem Kommando bestimmen, ob das Betriebssystem-ROM (Adreßbereich 0000 bis 3FFF hex) ein- oder ausgeblendet ist, desgleichen für das obere ROM (Adreßbereich C000 bis FFFF hex). Im oberen ROM-Bereich wird je nach gewählter ROM-Auswahl-Adresse das Basic-ROM oder ein externes ROM oder auch ein RAM-Bereich ein- oder ausgeblendet.

Die ROM-Auswahl-Adresse ist ein Ein-Byte-Wert, der für Werte von 0 bis 252 (00 bis FB hex) eine ROM-Bank auswählt. Das eingebaute Basic hat die ROM-Auswahl-Adresse 0, das Amsdos- und CP/M-ROM im Floppy-Controller die ROM-Auswahl-Adresse 7. Da die Adreßleitungen in der Normalkonfiguration nicht vollständig decodiert sind, werden beide ROMs auch unter anderen Adressen ein- oder ausgeblendet. Die ROM-Auswahl-Adresse kann vier Werte mit speziellen Bedeutungen annehmen, die aber die eingestellte ROM-Auswahl nicht verändern:

FC hex	Oberes ROM ein, unteres ROM ein
FD hex	Oberes ROM ein, unteres ROM aus
FE hex	Oberes ROM aus, unteres ROM ein
FF hex	Oberes ROM aus, unteres ROM aus

Mit ROM-Auswahl-Adresse »FF« stellen Sie den normalen Zustand Ihres Schneiders wieder her.

Alle Routinen können mit Enter, Esc oder der Leertaste unterbrochen werden. Ein nachfolgender Druck auf Esc bricht das Kommando ab, jede andere Taste läßt das Programm fortfahren.

Befehlsgruppe 2: Debug-Kommandos

Die Befehle der zweiten Gruppe dienen in erster Linie der Fehlersuche und dem Austesten von Maschinencode-Routinen. Zum besseren Verständnis dieser Befehle sollen zuvor zwei mit dem »Debugging« (Fehlersuche) zusammenhängende Begriffe geklärt werden:

Breakpoint:

Ein Breakpoint (Unterbrechungspunkt) kann auf jeden Z80-Maschinenbefehl gesetzt werden. Erreicht das Maschinenprogramm während des Probelaufs diesen Punkt, dann wird es unterbrochen und es erscheint eine »Break«-Meldung, gefolgt von der Adresse, an der die Unterbrechung erfolgte. Außerdem werden die Inhalte aller Z80-Register angezeigt, so daß der Programmierer sofort einen Überblick über den internen Zustand des Prozessors bei Erreichen des Breakpoints hat. Der Breakpoint ist beim Supermon durch den vom Betriebssystem nicht verwendeten »RST 6«-Befehl realisiert. Nach Eingabe der gewünschten Breakpoint-Adresse mit dem B-Kommando wird der ursprüngliche Inhalt dieser Adresse geteilt und anschließend der Code für den »RST 6«-Befehl eingeschrieben. Der Z80-Prozessor verzweigt bei Ausführung des »RST 6« zur Adresse 0030 hex, wo bei der Initialisierung des Monitors ein Sprung zur Breakpoint-Routine abgelegt wurde.

Verwenden Sie daher den »RST 6«-Befehl nicht in eigenen Programmen, wenn diese mit dem Supermon CPC ausgetestet werden sollen.

Registerspeicher:

Die Breakpoint-Routine des Supermon rettet alle Primärregister des Z80 sowie die beiden Indexregister in einem speziell-

len Speicherbereich innerhalb des Monitors, den Registerspeicher. Dieser belegt die Adressen 7110 bis 711D hex. Der Registerspeicher wird bei Auftreten eines Breakpoints mit den CPU-Registern geladen, desgleichen bei der Rückkehr von einem G-Befehl. Bei Aufruf eines Maschinencode-Programms mittels G-Befehl werden vor dem eigentlichen Programmstart die Z80-Register aus dem Registerbereich geladen. Der Registerspeicher kann mittels R-Kommando aufgelistet werden oder per M-Kommando (zum Beispiel »M 8110«). Um den Registerspeicher mit bestimmten Startwerten zu füllen, benutzen Sie bitte das W-Kommando (zum Beispiel »W 8110«). Die Register sind 16-Bit-weise in der Reihenfolge AF, BC, DE, HL, IX, IY im Registerspeicher abgelegt.

Breakpoint setzen:

B < Adresse > — Mit diesem Befehl kann in ein Maschinenprogramm ein Unterbrechungspunkt (Breakpoint) gesetzt werden. Der Breakpoint muß immer auf das erste Byte eines Z80-Befehls gesetzt werden, da er sonst nicht die gewünschte Wirkung erzielt, sondern im Gegenteil zu unvorhersehbaren Ergebnissen bei der Programmausführung führen kann. Es kann jeweils nur ein Breakpoint gesetzt sein. Beim Versuch, weitere Breakpoints mittels B-Kommando zu setzen, erscheint die Meldung »NOT OK«.

Kill Breakpoint (Breakpoint löschen):

K — Das K-Kommando dient zum Löschen eines gesetzten Breakpoints.

Go (startet Maschinenprogramm):

G < Startadresse > — Mit dem G-Kommando wird ein Maschinenprogramm an der gewählten Adresse gestartet, wobei zuvor die Prozessor-Register aus dem Registerspeicher (siehe oben) geladen werden. Damit ist es möglich, Routinen zu starten, die bestimmte Registerinhalte erwarten. Der Befehl ist ebenfalls nützlich zum »wieder-starten« eines Programms nach Unterbrechung durch einen Breakpoint. Es muß dazu nur der Breakpoint mit dem K-Befehl gelöscht werden; ein anschließender G-Befehl auf die Break-Adresse setzt das Maschinenprogramm dann ab der Unterbrechungsstelle mit der richtigen Registerbelegung wieder fort. Die gestartete Maschinencode-Routine kehrt nach einem »RET«-Befehl wieder zum Monitor zurück, wobei die Z80-Register im Registerspeicher abgelegt werden. Eine eventuelle Meldung »BREAK«, gefolgt von einer Adresse und einer Auflistung der Z80-Registerinhalte bedeutet, daß das gestartete Maschinenprogramm auf einen Breakpoint gestoßen ist.

Register anzeigen:

R — dieser Befehl listet die Inhalte der im Registerspeicher festgehaltenen Z80-Register auf, wie sie vom G-Befehl verwendet werden. Eine Änderung der Registerinhalte ist über den W-Befehl möglich (siehe auch unter »Registerspeicher«).

User-Jump ausführen:

U — diese Funktion führt ein vom Anwender per (-Kommando definiertes Unterprogramm aus. Das U-Kommando ist insbesondere nützlich, um den Befehlssatz des Supermon zu erweitern, oder um häufig gebrauchte, selbstgeschriebene Routinen schnell und sicher aufrufen zu können. Die User-Routine wird sofort ausgeführt.

User-Jump definieren:

(< Startadresse > — dieses Kommando legt die Startadresse einer Anwender-Routine zur Erweiterung des Supermon-Befehlssatzes fest. Nach diesem Kommando wird beim U-Befehl stets die hier angegebene Startadresse angesprungen. Die User-Routine muß mit »RET« abgeschlossen sein und darf den Stack nicht manipulieren. Ist beim Verlassen der User-Routine das Carry-Flag gesetzt, dann wird nach dem U-

Befehl die Meldung »Ok« ausgegeben, andernfalls die Meldung »NOT OK«. Um den User-Jump abzuschalten, geben Sie die Startadresse 0000 ein; es wird dann kein User-Jump mehr ausgeführt.

User-Jump anzeigen:

) — der augenblicklich definierte User-Jump kann mit diesem Kommando angezeigt werden. 0000 bedeutet, daß kein User-Jump definiert ist.

Befehlsgruppe 3: Laden und Speichern

Die Befehle dieser Gruppe dienen dem Datenverkehr zwischen Computer und Disketten-Kassettenlaufwerk. Bei Kassettenbetrieb muß beachtet werden, daß die Namenskonvention für Kassetten- und Diskettendateien unterschiedlich ist: Kassettdateien dürfen beliebige Namen mit einer Länge von maximal 16 Zeichen haben, Namen für Diskettendateien bestehen dagegen aus maximal 8 Zeichen plus 3 Zeichen Namensweiterung (Extension). Dateiname und Extension müssen durch einen Punkt ».« getrennt sein. Gültige Disketten-Dateinamen sind also zum Beispiel »TEST.BAS« oder »PROGRAMM.BIN«. Die Konvention für Diskettendateinamen ist mit der Namenskonvention unter CP/M weitgehend identisch.

Um Probleme bei der Übertragung speziell von Kassette zu Diskette zu vermeiden, werden alle Namen von Kassettenprogrammen vom Supermon CPC automatisch an das Namensformat für Diskettendateien angepaßt. Aus »TESTPROGRAMM« wird damit »TESTPROG.RAM«.

Dem Kassettenbenutzer sei im Hinblick auf die spätere Anschaffung eines Diskettenlaufwerks empfohlen, von vorneherein nur Namen entsprechend der Disketten-Konvention zu verwenden.

Folgende Dateitypen werden von den Schneider-Computern unterstützt:

00	Basic-Programm
01	Basic-Programm (protected)
02	Maschinensprache-Programm
03	Maschinensprache-Programm (protected)
04	Bildschirmdatei
05	Bildschirmdatei (protected)
16	Sequentielle Datei (mit OPENOUT erzeugt, kann nicht als Programm geladen werden)

Load (Programm laden):

L < Filename > — dieses Kommando dient zum Laden eines Programmes von Kassette oder Diskette. Es wird automatisch in den gleichen Speicherplatz geladen, aus dem es abgespeichert wurde. Während des Ladevorganges werden alle relevanten Informationen aus dem File-Header (Vorspann) angezeigt. Wenn Sie nicht sicher sind, ob das zu ladende Programm eventuell wichtige Daten (oder sogar den Supermon) überschreibt, sollten Sie das Y-Kommando verwenden.

Yank (Programm verschoben laden):

Y < Filename > — das Y-Kommando funktioniert weitgehend analog zum L-Befehl. Allerdings wird hier das Programm nicht unbedingt an den Original-Speicherplatz geladen, sondern kann an eine beliebige Adresse geladen werden. Dazu werden wiederum alle Header-Informationen ausgegeben. Anschließend erfragt das System mit »LOAD:« die Speicheradresse, an die das Programm geladen werden soll. Das Y-Kommando ist besonders nützlich, wenn das zu ladende Programm sich normalerweise in einem ungünstigen Speicherbereich befindet, sich also beispielsweise mit dem Supermon

Maschinensprache-Monitor

CPC speichermäßig überschneiden würde. Mit dem Y-Befehl kann das Programm in einem solchen Falle in einen »ungefährlichen« Speicherbereich geladen und dort analysiert werden. In der Regel ist es dort aber nicht lauffähig.

Header-Informationen ausgeben:

H — dieses Kommando listet die wichtigen Header-Informationen zu dem zuletzt geladenen Programm auf.

Copy (Programm kopieren):

C — das C-Kommando kopiert das zuletzt geladene File mit den Parametern, wie sie im H-Kommando aufgelistet werden.

Save (Programm speichern):

S <Filename> <Filetyp> <Anfangsadresse> <Endadresse> <Startadresse> — dieses Kommando dient zum Speichern von Programmen oder beliebigen Speicherbereichen auf Kassette oder Diskette. Angegeben werden müssen hierzu der gewünschte Filename, der Filetyp (vergleiche L-Befehl), der Speicherbereich, der aufgezeichnet werden soll (<Anfangsadresse> bis <Endadresse>) und schließlich noch die Startadresse. Die Startadresse ist die Adresse, ab der ein Maschinenprogramm gestartet werden muß (falls die zu speichernde Datei kein Maschinensprache-Programm darstellt, ist die Startadresse 0000).

Catalog (Inhalt einer Kassette oder Diskette anzeigen):

@ — das @-Kommando entspricht der CAT-Funktion in Basic. Der Inhalt der eingelegten Kassette/Diskette wird am Bildschirm aufgelistet.

Speed 0 (Aufzeichnungsgeschwindigkeit 0 wählen):

0 — das Kommando wählt eine Aufzeichnungsgeschwindigkeit von 1000 Baud für Kassetten-Operationen (entspricht »SPEED WRITE 0« in Basic).

Speed 1 (Aufzeichnungsgeschwindigkeit 1 wählen):

1 — das Kommando wählt eine Aufzeichnungsgeschwindigkeit von 2000 Baud für Kassettenoperationen (entspricht »SPEED WRITE 1« in Basic).

On/Off (Kassettenmotor ein/aus):

0 — dieses Kommando schaltet den Kassettenmotor wechselweise ein/aus. Es dient zur Feinjustierung der Kassette auf eine bestimmte Bandstelle.

Externe Kommandos (DOS-Kommandos)

I <Kommandoname>, <Parameter 1>, <Parameter 2>
Mit »I« können (wie von Basic aus) externe Kommandos, in erster Linie natürlich die DOS-Kommandos, aufgerufen werden.

Ohne Diskettenlaufwerk (CPC 464) wird nur das Kommando »IBASIC« unterstützt, das die Kontrolle wieder an das eingebaute Schneider-Basic abgibt. Nach diesem Kommando kann nicht mehr mit dem Monitor gearbeitet werden. Strings können übrigens direkt übergeben werden — nicht umständlich mittels Variablen, wie normalerweise bei Erweiterungsbefehlen.

Beispiele:

»I ERA, NAME.BIN« löscht das File »NAME BIN« von der Diskette.

»IREN, NEUNAME, ALTNAME« benennt das File »ALTNAME« in »NEUNAME« um.

Befehlsgruppe 4: Verschiedenes

Bye (Verlassen des Monitors):

! — der !-Befehl bewirkt ein »weiches« Verlassen des Supermon CPC durch einen Sprung zum Basic-Warmstart (Ready-Routine). Alle Programme und Speicherinhalte bleiben dabei erhalten.

Drucker einschalten:

— Das #-Kommando bewirkt, daß der Drucker während der Ausführung des nächsten Monitor-Befehls zugeschaltet wird. Hierzu muß der Drucker angeschlossen und eingeschaltet sein. Das gewünschte, mit dem Drucker zu protokollierende Monitor-Kommando kann direkt im Anschluß an den #-Befehl gegeben werden. Falls die Meldung »NOT OK« erscheint, ist der Drucker nicht richtig angeschlossen.

40/80-Zeichen-Umschaltung:

& — der &-Befehl schaltet zwischen 40- und 80-Zeichen-Modus hin und her.

7/8-Bit-ASCII-Umschaltung:

. — der .-Befehl bewirkt eine Umschaltung der Anzeige von 7- auf 8-Bit-ASCII-Modus und umgekehrt. Im 8-Bit-Modus werden alle in Aufzählung vorkommenden Zeichen vom Monitor als vollständige 8-Bit-Zeichen gelistet. Im 7-Bit-Modus wird Bit 7 jedes Zeichens auf Null gesetzt; es ergibt sich ein 7-Bit-ASCII-Code.

Der 7-Bit-Modus ist insbesondere zur Suche von Befehlslisten in Maschinenprogrammen zu verwenden, bei denen Bit 7 einzelner Zeichen gesetzt ist, wodurch die Zeichenfolgen im 8-Bit-Modus unleserlich werden.

Bei Druckerausgabe wird ausschließlich der 7-Bit-Modus verwendet.

Hex Addition:

+ <Wert 1> <Wert 2> — dieses Kommando ermöglicht die Addition zweier vierstelliger Hexadezimal-Zahlen. Es ist bei Adressenberechnungen häufig nützlich.

Hex Subtraktion:

- <Wert 1> <Wert 2> — dieses Kommando ermöglicht die Subtraktion zweier vierstelliger Hexadezimal-Zahlen.

Relative Sprungdistanz berechnen:

↑ <Sprungadresse> <Sprungziel> — Dieses Kommando berechnet das »Distanz-Byte« bei relativen Sprunganweisungen aus der Adresse, an der der Sprungbefehl steht und der Adresse, zu der gesprungen werden soll. Das Kommando ist bei der Eingabe kleiner Maschinenprogramme eine häufig benutzte Arbeitserleichterung. Falls die Meldung »NOT OK« auftritt, liegt das Sprungziel außerhalb des beim Z80 mit relativen Sprüngen zu erreichenden Bereichs.

Monitor-Warmstart:

Enter — die Enter-Taste ohne Befehlseingabe bewirkt einen Monitor-Warmstart.

Monitor-Kaltstart

Ctrl-C — die Tastenkombination Ctrl und C bewirkt einen Monitor-Kaltstart, also eine neue Initialisierung des Monitors. Ein solcher Kaltstart hilft in vielen Fällen, wo die Bildschirmanzeige durch nicht einwandfrei laufende Maschinenprogramme durcheinander geraten ist.

(Volker Everts)

```
10 * ***** [D164]
20 * (c) Volker Everts * [747A]
30 * Drachensestr. 12 * [4BB4]
40 * 8000 Muenchen 70 * [8B42]
50 * ***** [CD6C]
60 MEMORY &7FFF:LOAD"smon.bin",&8000:CAL
L &8000 [DDDC]
```

Listing 1. Kurz aber effektiv: Das Startprogramm zu »Supermonitor CPC-1002«

```

10 DATA 31,0,C0,CD,0,BB,CD,4E,BB,CD [C4C2B]
20 DATA FF,BB,CD,65,BC,CD,47,BC,E,0 [AA90]
30 DATA CD,15,B9,7C,32,2B,81,3A,1D,81 [C297E]
40 DATA 3D,E6,1,3C,32,FD,80,CD,E,BC [E116]
50 DATA AF,32,FC,80,3A,1E,81,47,AF,AF [E6CE]
60 DATA CD,32,BC,3A,1F,81,47,4F,3E,1 [E33E]
70 DATA CD,32,BC,3A,20,81,47,4F,CD,38 [E7A8]
80 DATA BC,11,2E,81,CD,2D,82,31,0,C0 [C106]
90 DATA CD,E0,81,AF,32,FC,80,3E,3E,CD [DF10]
100 DATA FE,81,CD,44,82,CD,74,82,21,8A [D7F8]
110 DATA 80,4F,23,23,7E,B7,28,F3,23,B9 [C7B8]
120 DATA 20,F6,CD,FE,81,11,7C,80,D5,7E [0616]
130 DATA 23,66,6F,E9,11,8F,81,30,3,11 [770C]
140 DATA 94,81,CD,E0,81,CD,2D,82,18,C1 [A4F4]
150 DATA 21,20,83,3A,5C,83,41,F0,83,4D [0882]
160 DATA E,84,57,39,84,48,5A,84,4C,9A [E2A6]
170 DATA 84,7C,80,83,40,36,85,53,F0,84 [E676]
180 DATA 43,3E,85,59,DC,84,23,49,83,47 [09A8]
190 DATA 9D,85,26,39,83,54,D4,85,44,FD [45D2]
200 DATA 87,58,83,85,42,F8,85,52,65,86 [5B8E]
210 DATA 4B,A9,86,46,8B,86,2B,70,87,2D [04E6]
220 DATA 89,87,5E,93,87,55,85,87,4A,80 [CECC]
230 DATA 87,28,C0,87,29,C9,87,30,D4,87 [F0AC]
240 DATA 31,DB,87,5A,E5,87,2,4D,80,3 [EE16]
250 DATA 0,80,4F,74,BC,2E,2F,83,3F,80 [EF74]
260 DATA 88,0,0,0,0,0,0,7,88,0,0 [B2DE]
270 DATA 0,0,0,0,0,0,0,0,0,0 [DDBE]
280 DATA 0,0,0,0,0,0,0,0,0,0 [E5C0]
290 DATA 0,0,0,0,0,1,1,18,1,0 [163A]
300 DATA 0,0,0,0,0,0,0,0,0,0 [DCB2]
310 DATA 8,AD,20,20,20,20,20,20,20,2A [CBB8]
320 DATA 2A,20,53,55,50,45,52,4D,4F,4E [3184]
330 DATA 20,43,50,43,2D,31,30,3C,32,20 [49F0]
340 DATA 2A,2A,A,A,D,20,20,43,6F,70 [586E]
350 DATA 79,72,69,67,68,74,20,28,63,29 [9158]
360 DATA 20,31,39,38,35,20,62,79,20,48 [0B18]
370 DATA 61,70,70,79,20,43,6F,6D,70,75 [F06E]
380 DATA 74,65,72,A,A,D,0,25,25,25 [DDD6]
390 DATA 20,56,6F,6C,6B,65,72,20,45,76 [C190]
400 DATA 65,72,74,73,20,25,25,25,0,4E [CDDC6]
410 DATA 4F,54,20,7,4F,4B,0,4C,6F,61 [65E2]
420 DATA 64,3A,0,53,5A,2D,48,2D,50,4E [AD3C]
430 DATA 43,20,20,41,20,20,42,43,20,20 [4DC4]
440 DATA 20,44,45,20,20,20,48,4C,20,20 [27FC]
450 DATA 20,49,58,20,20,20,49,59,0,42 [1BA4]
460 DATA 52,45,41,4B,20,7,0,20,46,69 [075E]
470 DATA 6C,65,6E,61,6D,65,20,3F,20,0 [E832]
480 DATA 3A,2B,81,87,CA,3A,BD,C3,5E,BD [9880]
490 DATA F5,3A,86,82,87,20,3,F1,C9,F5 [3AA0]
500 DATA 3E,A,CD,FE,81,3E,D,CD,FE,81 [90CC]
510 DATA F1,C9,E6,FF,FE,20,30,2,3E,2E [CA9C]
520 DATA F5,3A,FC,80,87,28,23,F1,F5,FE [C144]
530 DATA D,28,C,FE,A,28,8,CB,BF,FE [EFCF]
540 DATA 20,30,2,3E,2E,C5,6,20,4F,79 [6CBE]
550 DATA CD,2B,BD,38,5,10,F8,AF,32,FC [C5EA]
560 DATA 80,C1,F1,CD,5A,BB,C9,F5,D5,1A [366E]
570 DATA B7,28,6,CD,FE,81,13,18,F6,D1 [F9A8]
580 DATA F1,C9,F5,3E,20,8B,FE,81,F1,C9 [C572]
590 DATA F5,3E,7,CD,5A,BD,F1,C9,7C,CD [4444]
600 DATA 51,82,7D,F5,F,F,F,CD,5A [B4AE]
610 DATA 82,F1,E6,F,FE,A,38,2,C6,7 [9770]
620 DATA C6,30,CD,FE,81,C9,CD,8A,BB,CD [98B8]
630 DATA 6,BB,F5,CD,8D,BB,F1,C9,CD,68 [A846]
640 DATA 82,FE,61,D8,FE,7B,D0,D6,20,C9 [2650]
650 DATA CD,74,82,FE,FC,C8,FE,D,C8,D6 [5E60]
660 DATA 30,38,F3,FE,A,38,B,D6,7,FE [63FC]
670 DATA 10,30,E9,FE,A,3F,30,E4,F5,CD [4DD0]
680 DATA 5A,82,F1,C9,CD,3C,82,CD,80,82 [3D2E]
690 DATA D0,C5,7,7,7,4F,CD,80,82 [5640]
700 DATA 30,2,B1,37,C1,C9,CD,8D,82,D0 [9594]
710 DATA EB,CD,42,82,0,67,CD,A5,82,6F [A35A]
720 DATA C9,21,A4,AC,3A,2B,81,B7,28,3 [3C7C]
730 DATA 21,8A,AC,36,0,CD,D6,81,D0,E5 [A9A2]
740 DATA AF,6,FF,4,BE,23,20,FB,E1,37 [BC68]
750 DATA C9,3A,FC,80,6,10,B7,C0,3A,FD [8FC8]
760 DATA 80,F,D0,6,8,C9,CD,9,BB,3F [9898]
770 DATA D8,FE,D,28,8,FE,20,28,4,FE [7C0E]
780 DATA FC,20,EF,CD,68,82,FE,FC,C8,37 [87A6]
790 DATA C9,CD,E0,81,CD,4C,82,CD,3C,82 [1B6C]
800 DATA CD,3C,82,C9,7C,BA,C0,7D,BB,C9 [F198]
810 DATA 3E,FD,CD,60,83,AF,CD,60,83,CD [6D7E]
820 DATA E0,81,C3,64,C0,3A,F7,81,EE,80 [BBFA]
830 DATA 32,F7,81,37,C9,3A,FD,80,3D,80 [7BE2]
840 DATA 2,3E,2,32,FD,80,CD,E,BC,37 [F6E0]
850 DATA C9,3E,1,32,FC,80,CD,2E,BD,3F [73F6]
860 DATA D0,3E,D,CD,2B,8D,E1,C3,5F,80 [5D08]
870 DATA CD,A2,82,D0,FE,FC,30,6,4F,CD [2A16]
880 DATA F,B9,37,C9,F,F5,DC,9,B9,F1 [4840]
890 DATA F5,D4,6,B9,F1,F,F5,DC,3,B9 [E130]
900 DATA F1,D4,0,B9,37,C9,CD,3C,82,3E [CABC]
910 DATA FF,32,EB,84,CD,C7,82,D0,7B,B7 [F556]
920 DATA C8,E5,48,6,0,3E,2C,ED,B1,ED [4750]
930 DATA 9B,83,2B,2B,CB,FE,1E,0,D0,21 [DAE2]
940 DATA A,81,23,BE,20,15,23,FD,21,2 [63CA]
950 DATA 81,CD,CA,83,30,B,DD,2B,D0,2B [B300]
960 DATA FD,21,5,81,CD,CA,83,E1,D5,CD [9DF8]
970 DATA D4,BC,D1,D0,7B,CD,1B,0,37,C9 [CCEA]
980 DATA FD,36,0,FF,FD,75,1,FD,74,2 [CF08]
990 DATA 1C,FD,E5,E3,DD,75,0,DD,74,1 [C480]
1000 DATA E1,2B,23,FD,34,0,7E,B7,C8,FE [CB24]
1010 DATA 2C,20,F5,36,0,23,37,C9,CD,BD [A9D6]

```

```

1020 DATA 82,D0,CD,E5,82,CB,20,CB,20,C5 [9C70]
1030 DATA CD,D,83,7E,CD,F6,81,23,10,F9 [BC1A]
1040 DATA C1,CD,F4,82,38,EF,3F,C9,CD,BD [DF04]
1050 DATA 82,D0,CD,E5,82,C5,CD,D,83,E5 [143A]
1060 DATA C5,7E,CD,51,82,CD,3C,82,23,10 [BF4A]
1070 DATA F6,C1,E1,CD,3C,82,7E,CD,F6,81 [06C6]
1080 DATA 23,10,F9,C1,CD,F4,82,38,DE,3F [3180]
1090 DATA C9,CD,BD,82,D0,CD,E0,81,CD,E5 [4EFA]
1100 DATA 82,C5,CD,D,83,3E,8,CD,FE,81 [E7D6]
1110 DATA CD,A2,82,30,4,77,23,10,F7,C1 [4F92]
1120 DATA 38,EB,3F,C9,2A,0,81,CD,E0,81 [E604]
1130 DATA E5,DD,E1,CD,67,85,CD,2D,82,DD [91E6]
1140 DATA 7E,12,CD,E0,81,CD,51,82,CD,3C [9088]
1150 DATA 82,DD,6E,15,DD,66,16,CD,4C,82 [F9B6]
1160 DATA DD,5E,18,DD,56,19,2B,CD,3C [6FA2]
1170 DATA 82,CD,4C,82,DD,6E,1A,DD,66,18 [558A]
1180 DATA CD,3C,82,CD,4C,82,37,C9,CD,A0 [2FBA]
1190 DATA 84,D0,18,23,11,C9,81,CD,2D,82 [0F1A]
1200 DATA CD,C7,82,D0,AF,CD,6B,BC,11,0 [E64A]
1210 DATA 8,CD,77,BC,30,21,22,0,81,CD [184C]
1220 DATA 5A,84,DD,6E,15,DD,66,16,C9,CD [73AB]
1230 DATA E0,81,E5,CD,83,BC,E1,30,A,DD [5A3A]
1240 DATA 75,15,DD,74,16,CD,7A,BC,D8,CD [F4C0]
1250 DATA 7D,BC,B7,C9,CD,A0,84,D0,CD,E0 [8CEE]
1260 DATA 81,11,97,81,CD,2D,82,CD,BD,82 [AC62]
1270 DATA 30,E9,18,D3,11,C9,81,CD,2D,82 [4B3E]
1280 DATA CD,C7,82,D0,11,0,8,CD,8C,BC [7CC2]
1290 DATA 30,2F,AF,CD,6B,BC,CD,E0,81,CD [C108]
1300 DATA A5,82,30,23,ED,47,CD,8B,82,30 [DB28]
1310 DATA 1C,87,ED,52,3F,30,16,23,EB,E5 [0562]
1320 DATA CD,8D,82,44,4D,E1,30,B,ED,57 [2526]
1330 DATA CD,9B,BC,30,4,CD,8F,BC,D8,CD [0D9C]
1340 DATA 92,BC,B7,C9,11,0,8,CD,9B,BC [D5C8]
1350 DATA 37,C9,CD,67,85,EB,11,0,8,CD [E99C]
1360 DATA 8C,BC,30,E7,DD,2A,0,81,DD,66 [632E]
1370 DATA 16,DD,6E,15,DD,56,19,DD,5E,18 [4F92]
1380 DATA DD,46,18,DD,4E,1A,DD,7E,12,CB [A1EC]
1390 DATA 87,18,C1,C5,E5,2A,0,81,7E,FE [8508]
1400 DATA 20,30,1,23,11,A4,AC,D5,6,8 [6A90]
1410 DATA CD,8F,85,30,9,3E,2E,12,13,6 [D44C]
1420 DATA 3,CD,8F,85,AF,12,7B,D1,93,E1 [C218]
1430 DATA C1,47,C9,7E,B7,C8,FE,20,28,2 [4604]
1440 DATA 12,13,23,10,F4,37,C9,CD,BD,82 [1E26]
1450 DATA D0,CD,E0,81,11,AD,85,D5,E5,CD [F2AE]
1460 DATA 1C,86,C9,F5,CD,35,86,F1,C9,CD [DDC2]
1470 DATA B8,82,D0,B7,ED,52,3F,D0,44,4D [5D80]
1480 DATA CD,BD,82,D0,CD,1A,83,38,8,9 [EFBC]
1490 DATA EB,9,3,ED,B8,37,C9,EB,3,ED [0FA6]
1500 DATA B0,C9,CD,8D,82,D0,CD,E5,82,CB [E6EA]
1510 DATA 20,CB,20,48,CD,D,83,CD,68,82 [0AEE]
1520 DATA FE,FC,3F,C8,FE,D,3F,C8,77,CD [88DC]
1530 DATA F6,81,23,10,EE,41,18,EB,CD,BD [F178]
1540 DATA 82,D0,7C,B5,C8,ED,5B,D,81,7A [013E]
1550 DATA B3,C0,E7,32,C,81,36,F7,22,D [2668]
1560 DATA 81,3E,C3,21,4E,86,32,30,0,22 [765E]
1570 DATA 31,0,37,C9,ED,4E,11,81,ED,58 [8EEE]
1580 DATA 13,81,2A,F,81,E5,F1,2A,15,81 [1AC]
1590 DATA DD,2A,17,81,FD,2A,19,81,C9,ED [DDAC]
1600 DATA 43,11,81,ED,53,13,81,22,15,81 [69A0]
1610 DATA F5,E1,22,F,81,DD,22,17,81,FD [B8F2]
1620 DATA 22,19,81,C9,CD,35,86,E1,2B,CD [3660]
1630 DATA E9,81,11,C1,81,CD,2D,82,CD,4C [5176]
1640 DATA 82,CD,65,86,C3,7C,8D,CD,E0,81 [0770]
1650 DATA CD,E9,81,11,9D,81,CD,2D,82,CD [80AC]
1660 DATA E0,81,3A,F,81,6,8,4F,3E,30 [9EEE]
1670 DATA CB,21,CE,0,CD,FE,81,10,F5,CD [ED64]
1680 DATA 3C,82,3A,10,81,CD,51,82,21,11 [CFDE]
1690 DATA 81,6,5,5E,23,56,23,CD,3C,82 [C240]
1700 DATA EB,CD,4C,82,EB,10,F2,CD,E0,81 [4FC4]
1710 DATA CD,E9,81,37,C9,2A,D,81,7C,B5 [D628]
1720 DATA C8,3A,C,81,77,21,0,0,22,D [6148]
1730 DATA 81,37,C9,CD,88,82,D0,B7,ED,52 [9690]
1740 DATA 3F,D0,23,44,4D,21,21,81,C5,CD [D324]
1750 DATA 37,87,C1,D0,DD,21,21,81,EB,CD [4572]
1760 DATA E0,81,DD,7E,0,B7,C8,DD,7E,1 [2BE]
1770 DATA ED,B1,37,E0,E5,FD,E1,CD,F4,82 [4CEA]
1780 DATA 3F,DD,DD,7E,2,B7,28,8,DD,7E [9B00]
1790 DATA 3,FD,BE,0,20,E3,DD,7E,4,87 [246C]
1800 DATA 28,8,DD,7E,5,FD,BE,1,20,D5 [424E]
1810 DATA DD,7E,6,87,28,8,DD,7E,7,FD [F594]
1820 DATA BE,2,20,C7,2B,CD,4C,82,CD,3C [573E]
1830 DATA 82,7E,CD,51,82,23,7E,CD,51,82 [B14E]
1840 DATA FD,7E,1,CD,51,82,FD,7E,2,CD [0B00]
1850 DATA 51,82,CD,E0,81,18,A6,6,4,36 [2F38]
1860 DATA FF,23,CD,3C,82,CD,80,82,30,1B [C317]
1870 DATA 7,7,7,7,E6,F0,77,CD,80,82 [D094]
1880 DATA 30,7,86,77,23,10,E4,37,C9,FE [A0CE]
1890 DATA FC,B8,3E,8,CD,FE,81,FE,FC,C8 [F0EC]
1900 DATA 3E,58,CD,FE,81,CD,FE,81,2B,36 [FB00]
1910 DATA 0,23,18,E2,CD,88,82,D0,19,CD [DBEC]
1920 DATA 7D,87,CD,4C,82,37,C9,CD,3C,82 [E9A2]
1930 DATA 3E,3D,CD,FE,81,CD,3C,82,C9,CD [C70E]
1940 DATA B8,82,EB,D0,B7,ED,52,18,E2,CD [B5C6]
1950 DATA B8,82,D0,B7,ED,52,2B,2B,7C,FE [CDC6]
1960 DATA FF,28,2,B7,C0,AD,E6,80,C0,CD [C558]
1970 DATA 7D,87,7D,CD,51,82,37,C9,CD,BD [91CC]
1980 DATA 82,D0,E9,CD,E0,81,2A,29,81,7C [D674]

```

Listing 2. »Supermonitor CPC-1002« erlaubt Einblicke in jede Maschinencode-Routine

```

1990 DATA B5,C8,37,E9,CD,8D,82,D0,22,29 [E59AJ]
2000 DATA 81,37,C9,CD,3C,82,2A,29,81,CD [C258J]
2010 DATA 4C,82,37,C9,3E,19,21,4D,1,18 [E68EJ]
2020 DATA 5,3E,32,21,A7,0,CD,68,BC,37 [C456J]
2030 DATA C9,CD,88,82,D0,B7,ED,52,3F,D0 [Q0BB8J]
2040 DATA CD,A2,82,EB,13,4F,71,23,1B,7A [L6F5AJ]
2050 DATA B3,20,F9,37,C9,CD,8D,82,D0,CD [C6B6J]
2060 DATA F4,82,3F,D8,CD,D,83,11,8,AD [L28B8J]
2070 DATA ED,53,2C,81,E,0,CD,3,89,FD [L7550J]
2080 DATA 21,A4,AC,FE,DD,20,4,CB,C1,18 [E72EJ]
2090 DATA F1,FE,FD,20,4,CB,C9,18,E9,FE [L68C8J]
2100 DATA ED,28,68,FE,CB,CA,BA,88,CB,7F [L8F26J]
2110 DATA 20,30,CB,77,20,8,11,E6,8A,CD [D0E82J]
2120 DATA 1B,89,18,BF,11,A2,8A,47,CD,62 [L4E50J]
2130 DATA 8A,FD,36,0,20,FD,23,EB,78,F [L17AAJ]
2140 DATA F,F,CD,78,8A,FD,36,0,20,FD [L8394J]
2150 DATA 23,78,CD,78,8A,CD,AC,89,18,9B [L6A2J]
2160 DATA CB,77,20,20,11,AD,8A,47,F,F [L57AJ]
2170 DATA F,E6,7,CD,8F,8A,EB,78,CD,78 [L2A24J]
2180 DATA 8A,CD,AC,89,18,81,11,79,81,CD [L5274J]
2190 DATA E0,81,CD,20,82,C9,11,5C,8C,E6 [L4286J]
2200 DATA 3F,CD,1B,89,C3,1,88,CD,3,89 [LDA0AJ]
2210 DATA D6,40,38,18,11,C3,8D,FE,3C,38 [L373AJ]
2220 DATA B,D6,60,38,D,11,E,8F,FE,1C [L8D42J]
2230 DATA 30,6,CD,1B,89,C3,1,88,CD,57 [L2C76J]
2240 DATA 8A,C3,1,88,CD,3,89,47,E6,C0 [L8584J]
2250 DATA 20,17,78,11,C5,8A,F,F,F,E6 [L6510J]
2260 DATA 7,CD,8F,8A,78,C3,1,88,8A,EB,CD [L2A2AJ]
2270 DATA AC,89,C3,1,88,11,DD,8A,17,17 [L8CE0J]
2280 DATA 17,E6,3,3D,CD,8F,8A,78,F,F [LAD70J]
2290 DATA F,E6,7,C6,30,FD,77,0,FD,23 [L823CJ]
2300 DATA FD,36,0,20,FD,23,78,CD,78,8A [Q9220J]
2310 DATA EB,CD,AC,89,C3,1,88,7E,23,F5 [L4744J]
2320 DATA E5,2A,2C,81,FE,20,36,2E,38,1 [L8DC2J]
2330 DATA 77,23,22,2C,81,E1,CD,51,82,F1 [L9B0EJ]
2340 DATA C9,EB,47,B7,28,8,23,7E,FE,60 [L714J]
2350 DATA 38,FA,10,F8,7E,FE,7B,20,5,CD [L493EJ]
2360 DATA 57,8A,18,7A,D6,20,6,5,FD,77 [LAC6CJ]
2370 DATA 0,FD,23,23,5,7E,FE,60,30,6A [LDB64J]
2380 DATA FE,20,20,F0,FD,77,0,FD,23,10 [L2AE0J]
2390 DATA F9,FD,2B,FD,77,0,FD,23,23,7E [L5052J]
2400 DATA FE,60,30,52,FE,5E,28,44,FE,21 [Q9F58J]
2410 DATA 28,45,FE,2A,28,12,FE,28,28,1A [L6C4CJ]
2420 DATA FE,40,20,E1,EB,FD,77,0,FD,23 [L526J]
2430 DATA CD,3,89,EB,FD,77,0,FD,23,EB [L5AFEJ]
2440 DATA CD,3,89,EB,18,CD,FD,36,0,40 [LACAEJ]
2450 DATA FD,23,EB,CD,3,89,EB,E5,6F,26 [L76EJ]
2460 DATA 0,7,30,2,26,FF,19,FD,75,0 [L9A64J]
2470 DATA FD,23,7C,E1,18,AD,CD,11,8A,18 [L096J]
2480 DATA AB,CD,2C,8A,18,A3,FD,36,0,0 [L5AA4J]
2490 DATA FD,21,A4,AC,CD,3C,82,3A,86,B2 [L8DB0J]
2500 DATA FE,10,38,F6,FD,7E,0,FD,23,87 [L753CJ]
2510 DATA 28,2C,FE,40,28,F,FE,2A,28,1A [Q0A0CJ]
2520 DATA FE,20,30,2,3E,25,CD,FE,81,18 [L958EJ]
2530 DATA E5,FD,6E,0,FD,66,1,CD,4C,82 [L50DCJ]
2540 DATA FD,23,FD,23,18,D6,FD,7E,0,CD [L275EJ]
2550 DATA 51,82,FD,23,18,CC,E5,2A,2C,81 [L5556J]
2560 DATA 1,8,AD,B7,ED,42,EB,CD,3C,82 [L21EEJ]
2570 DATA 3A,86,B2,FE,23,38,F6,A,CD,F6 [L454AJ]
2580 DATA 81,3,1D,20,FB,D1,C9,E5,26,58 [L2ECCJ]
2590 DATA 2E,49,CB,41,20,A,26,59,CB,49 [L4DEJ]
2600 DATA 20,4,26,4C,2E,48,FD,75,0,FD [L262CJ]
2610 DATA 23,7C,E1,C9,FD,36,0,28,FD,23 [L83F8J]
2620 DATA CD,11,8A,FD,77,0,FD,23,FE,4C [L5856J]
2630 DATA 28,16,FD,36,0,28,FD,23,EB,CD [L842CJ]
2640 DATA 3,89,EB,FD,36,0,2A,FD,23,FD [L5FD8J]
2650 DATA 77,0,FD,23,3E,29,C9,6,4,FD [L4026J]
2660 DATA 36,0,3F,FD,23,10,FB,C9,F5,C5 [L6066J]
2670 DATA 6,3,1A,FD,77,0,13,FD,23,10 [L28D8J]
2680 DATA F7,FD,36,0,20,FD,23,C1,F1,C9 [L5510J]
2690 DATA E6,7,DD,21,A5,8A,32,83,8A,DD [L6C2AJ]
2700 DATA 7E,0,FE,21,CC,2C,8A,FD,77,0 [L80C8J]
2710 DATA FD,23,C9,B7,28,6,13,13,3D [L93AEJ]
2720 DATA 20,FA,CD,62,8A,FD,36,0,20,FD [LAD22J]
2730 DATA 23,C9,4C,44,20,42,43,44,45,48 [Q0BB2CJ]
2740 DATA 4C,21,41,41,44,44,41,44,43,53 [L4386J]
2750 DATA 55,42,53,42,43,41,4E,44,58,4F [LCECCJ]
2760 DATA 52,4F,52,20,43,50,20,52,4C,43 [L76A4J]
2770 DATA 52,52,43,52,4C,20,52,52,20,53 [L8086J]
2780 DATA 4C,41,53,52,41,53,4C,4C,53,52 [L82D8J]
2790 DATA 4C,42,49,54,52,45,53,53,45,54 [L8DBAJ]
2800 DATA 6E,4F,50,6C,44,20,42,43,2C,40 [LDFE6J]
2810 DATA 6C,44,20,28,42,43,29,2C,41,69 [L3CBEJ]
2820 DATA 4E,43,20,42,43,69,4E,43,20,42 [L6B00J]
2830 DATA 64,45,43,20,42,6C,44,20,42,2C [LDA66J]
2840 DATA 2A,72,4C,43,41,65,58,20,41,46 [LDC88J]
2850 DATA 2C,41,46,27,61,44,44,20,5E,2C [LECD4J]
2860 DATA 42,43,6C,44,20,41,2C,28,42,43 [L3AEJ]
2870 DATA 29,64,45,43,20,42,43,69,4E,43 [L21B4J]
2880 DATA 20,43,64,45,43,20,43,6C,44,20 [L878EJ]
2890 DATA 43,2C,2A,72,52,43,41,64,4A,4E [L8CF8J]
2900 DATA 5A,20,2B,6C,44,20,44,45,2C,40 [L28D4J]
2910 DATA 6C,44,20,28,44,45,29,2C,41,69 [L53C8J]
2920 DATA 4E,43,20,44,45,69,4E,43,20,44 [L6B8EJ]
2930 DATA 64,45,43,20,44,6C,44,20,44,2C [LAFB0J]
2940 DATA 2A,72,4C,41,6A,52,20,2B,61,44 [L4600J]
2950 DATA 44,20,5E,2C,44,45,6C,44,20,41 [L4AD0J]
2960 DATA 2C,28,44,45,29,64,45,43,20,44 [L93ACJ]
2970 DATA 45,69,4E,43,20,45,64,45,43,20 [L9A9EJ]
2980 DATA 45,6C,44,20,45,2C,2A,72,52,41 [L76D6J]
2990 DATA 6A,52,20,4E,5A,2C,2B,6C,44,20 [L0E2AJ]
3000 DATA 5E,2C,40,6C,44,20,28,40,29,2C [L8BDAJ]
3010 DATA 5E,69,4E,43,20,5E,69,4E,43,20 [L8900J]
3020 DATA 48,64,45,43,20,48,6C,44,20,48 [L8B9CJ]
3030 DATA 2C,2A,64,41,41,6A,52,20,5A,2C [L86E6J]
3040 DATA 2B,61,44,44,20,5E,2C,5E,6C,44 [L6D08J]
3050 DATA 20,5E,2C,28,40,29,64,45,43,20 [L76A8J]
3060 DATA 5E,69,4E,43,20,4C,64,45,43,20 [L2EDAJ]
3070 DATA 4C,6C,44,20,4C,2C,2A,63,50,4C [L891CJ]
3080 DATA 6A,52,20,4E,43,2C,2B,6C,44,20 [L70FAJ]
3090 DATA 53,50,2C,40,6C,44,20,28,40,29 [L59A8J]
3100 DATA 2C,41,69,4E,43,20,53,50,69,4E [L52D4J]
3110 DATA 43,20,21,64,45,43,20,21,6C,44 [LCA6EJ]
3120 DATA 20,21,2C,2A,73,43,46,6A,52,20 [L6A88J]
3130 DATA 43,2C,2B,61,44,44,20,5E,2C,53 [L80DAJ]
3140 DATA 50,6C,44,20,5E,2C,28,40,29,64 [L8FD0J]
3150 DATA 45,43,20,53,50,69,4E,43,20,41 [L3D88J]
3160 DATA 64,45,43,20,41,6C,44,20,41,2C [L859CJ]
3170 DATA 2A,63,43,46,72,45,5A,20,4E,5A [L82D4J]
3180 DATA 70,4F,50,20,42,43,6A,50,20,4E [L16BCJ]
3190 DATA 5A,2C,40,6A,50,20,40,63,41,4C [L38D2J]
3200 DATA 4C,20,4E,5A,2C,40,70,55,53,48 [L6E2EJ]
3210 DATA 20,42,43,61,44,44,20,2A,72,53 [L3F70J]
3220 DATA 5A,20,30,72,45,5A,20,5A,72,45 [L5180J]
3230 DATA 54,6A,50,20,5A,2C,40,78,63,41 [L8EC2J]
3240 DATA 4C,4C,20,5A,2C,40,63,41,4C,4C [L9A14J]
3250 DATA 20,40,61,44,43,20,2A,72,53,54 [L8E76J]
3260 DATA 20,31,72,45,5A,20,4E,43,70,4C [L87AAJ]
3270 DATA 50,20,44,45,6A,50,20,4E,43,2C [L68BAJ]
3280 DATA 40,6F,55,5A,20,2A,2C,41,63,41 [L85C6J]
3290 DATA 4C,4C,20,4E,43,2C,40,70,55,53 [L86ECJ]
3300 DATA 48,20,44,45,73,55,42,20,2A,72 [L9686J]
3310 DATA 53,54,20,32,72,45,5A,20,43,65 [L6E68J]
3320 DATA 58,58,6A,50,20,43,2C,40,69,4E [L81DCJ]
3330 DATA 20,41,2C,2A,63,41,4C,4C,20,43 [L6FC6J]
3340 DATA 2C,40,78,73,42,43,20,2A,72,53 [L87AAJ]
3350 DATA 5A,20,33,72,45,5A,20,50,4F,70 [L808AJ]
3360 DATA 4F,50,20,5E,6A,50,20,50,4F,2C [L57FEJ]
3370 DATA 40,65,58,20,28,53,50,29,2C,5E [L1FC4J]
3380 DATA 63,41,4C,4C,20,50,4F,2C,40,70 [L72E2J]
3390 DATA 55,53,48,20,5E,61,4E,44,20,2A [L74D8J]
3400 DATA 72,53,54,20,34,72,45,5A,20,50 [LCA64J]
3410 DATA 45,6A,50,20,28,5E,29,6A,50,20 [L84C4J]
3420 DATA 50,45,2C,40,65,58,20,4A,45,2C [L53ACJ]
3430 DATA 48,4C,63,41,4C,4C,20,50,45,2C [L3BE8J]
3440 DATA 40,78,78,4F,52,20,2A,72,53,54 [LCA66J]
3450 DATA 20,35,72,45,5A,20,50,70,4F,50 [L7E88J]
3460 DATA 20,41,46,6A,50,20,50,2C,40,64 [L8C94J]
3470 DATA 49,63,41,4C,4C,20,50,2C,40,70 [L95C8J]
3480 DATA 55,53,48,20,41,46,6F,52,20,2A [L17B8J]
3490 DATA 72,53,54,20,36,72,45,5A,20,40 [L8DA0J]
3500 DATA 6C,44,20,53,50,2C,5E,6A,50,20 [L87D6J]
3510 DATA 40,2C,40,65,49,63,41,4C,4C,20 [L8FDECJ]
3520 DATA 40,2C,40,78,63,50,20,2A,72,53 [L86C8J]
3530 DATA 5A,20,37,69,4E,20,42,2C,28,43 [L3C88J]
3540 DATA 29,6F,55,5A,20,28,43,29,2C,42 [L87CAJ]
3550 DATA 73,42,43,20,48,4C,2C,42,43,6C [L96D4J]
3560 DATA 44,20,28,40,29,2C,42,43,6E,45 [L80BAJ]
3570 DATA 47,72,45,5A,4E,69,40,20,30,6C [L83F2J]
3580 DATA 44,20,49,2C,41,69,4E,20,43,2C [L80DAJ]
3590 DATA 28,43,29,6F,55,5A,20,28,43,29 [L2CC0J]
3600 DATA 2C,43,61,44,43,20,42,43,6C,44 [L6DA2J]
3610 DATA 20,42,43,2C,28,40,29,78,72,45 [L82B0J]
3620 DATA 54,49,78,6C,44,20,52,2C,41,69 [L94E6J]
3630 DATA 4E,20,44,2C,28,43,29,6F,55,54 [L8FEAJ]
3640 DATA 20,28,43,29,2C,44,73,42,43,20 [L878EJ]
3650 DATA 48,4C,2C,44,45,6C,44,20,28,40 [L8DDCJ]
3660 DATA 29,2C,44,45,78,78,69,40,20,31 [L560AJ]
3670 DATA 6C,44,20,41,2C,49,69,4E,20,45 [L8EE6J]
3680 DATA 2C,28,43,29,6F,55,5A,20,28,43 [L8CDAJ]
3690 DATA 29,2C,45,61,44,43,20,48,4C,2C [L46E2J]
3700 DATA 44,45,6C,44,20,44,45,2C,28,40 [L8E80J]
3710 DATA 29,78,78,69,40,20,32,6C,44,20 [L26FEJ]
3720 DATA 41,2C,52,69,4E,20,48,2C,28,43 [L80DEJ]
3730 DATA 29,6F,55,5A,20,28,43,29,2C,48 [L8DD8J]
3740 DATA 73,42,43,20,48,4C,2C,48,4C,6C [L2702J]
3750 DATA 44,20,28,40,29,2C,48,4C,7B,78 [L1F04J]
3760 DATA 7B,72,52,44,69,4E,20,4C,2C,28 [L5C0EJ]
3770 DATA 43,29,6F,55,5A,20,28,43,29,2C [L8BD6J]
3780 DATA 4C,61,44,43,20,48,4C,2C,48,4C [L8DF0J]
3790 DATA 6C,44,20,48,4C,2C,28,40,29,78 [L970CJ]
3800 DATA 7B,78,72,4C,44,78,78,73,42,43 [L1328J]
3810 DATA 20,48,4C,2C,53,50,6C,44,20,28 [L51CCJ]
3820 DATA 40,29,2C,53,50,78,78,7B,69 [L562EJ]
3830 DATA 4E,20,41,2C,28,43,29,6F,55,54 [L84E8J]
3840 DATA 20,28,43,29,2C,41,61,44,43,20 [L8D8AJ]
3850 DATA 48,4C,2C,53,50,6C,44,20,53,50 [L87D6J]
3860 DATA 2C,28,40,29,6C,44,49,63,50,49 [L54DCJ]
3870 DATA 69,4E,49,6F,55,5A,49,78,78,78 [L7368J]
3880 DATA 7B,6C,44,44,63,50,44,69,4E,44 [L9A02J]
3890 DATA 6F,55,54,44,78,78,78,78,6C,44 [L866CJ]
3900 DATA 49,52,63,50,49,52,69,4E,49,52 [L8FC8J]
3910 DATA 6F,54,49,52,78,78,78,78,6C,44 [L1F62J]
3920 DATA 44,52,63,50,44,52,69,4E,44,52 [L18AEJ]
3930 DATA 6F,54,44,52,78,78,78,78,78,0 [L8EECJ]
3940 MEMORY &7999 [L7E2J]
3950 FOR i=32768 TO 36697:READ a#:POKE [L946J]
i,VAL("&"+a#):NEXT i [L866EJ]
3960 SAVE"smon.bin",b,&8000,&F59

```

Listing 2. »Supermonitor CPC-1002« (Schluß)

Backup muß nicht teuer sein



Eine Sicherheitskopie auf Diskette ist teuer. Speichern Sie Ihr Backup doch einfach auf Kassette!

Mit den Befehlen »TAPE« und »DISC« ist es einfach, Daten zwischen Recorder und Diskettenstation auszutauschen. Umständlich ist es allerdings, eine ganze Diskettenseite auf Band zu speichern. Mit dem Befehl »SAVE« (unter CP/M) kann man

zwar jedes Programm auf Kassette überspielen, aber um eine Diskette mit 20 (oder mehr) Files zu überspielen, muß man sehr viel »Tipparbeit« machen. Hier finden Sie eine Routine, die eine ganze Diskette direkt auf Kassette überspielt.

3-Zoll-Disketten kosten immer noch weit über zehn Mark. Sicherheitskopien auf Kassette sind deshalb bedeutend preisgünstiger als Backups auf Diskette. Das hier vorgestellte Programm erlaubt Speichergeschwindigkeiten zwischen 1200 und 4000 Baud. Je besser die Qualität der Kassetten ist, desto höher können Sie die Aufzeichnungsgeschwindigkeit wählen. Voreingestellt sind 3600 Baud. Um damit lesesicher Kopien zu erstellen, muß die Bandqualität allerdings sehr gut sein. Über eins sollte man sich allerdings im klaren sein. Der Zeitbedarf für ein Backup auf Kassette ist bedeutend höher als der für eine Sicherheitskopie auf Diskette.

So dauert das Sichern einer ganzen Diskettenseite auch mit 3600 Baud mehr als 10 Minuten. Doch da der Computer allein agiert und die Kostenersparnis sehr groß ist, wird das Programm viele Anhänger finden.

(Thomas Arnold/Andreas Lohrum)

```

10 ***** [9FCC]
20 *(c) Thomas Arnold/Andreas Lohrum * [43C2]
30 * Am Graben 25, 7700 Singen * [0780]
40 ***** [E9D2]
50 MODE 2 [F7F8]
60 LOCATE 35,10 [9BA6]
70 PRINT "BACKUP-DISC" [04FC]
80 LOCATE 13,12 [B5A6]
90 PRINT CHR$(164)+" August 1985 by Andr [0E0A]
   eas Lohrum und Thomas Arnold" [DEB0]
100 [034C]
110 REM INITIALISIERUNG [727C]
120 MEMORY 16999
130 OPENOUT "dum":MEMORY HIMEM-1:CLOSEOU [6862]
   T [F710]
140 DIM dir$(63),sdir$(63) [55FE]
150 DEFSTR a [C32A]
160 DEFINIT i-n [F724]
170 br=3600 [06C0]
180 [E4D8]
190 REM Maschinenroutinen laden [088E]
200 REM Sektor lesen/schreiben [0EBE]
210 FOR i=1 TO 29 [0AA2]
220 READ a [5BF2]
230 POKE 40959+i,VAL("&"+a) [2EFC]
240 NEXT i [7DA6]
250 REM Speichertransfer [66B8]
260 FOR i=1 TO 12 [8AAC]
270 READ a [93FE]
280 POKE 40699+i,VAL("&"+a) [A506]
290 NEXT i [4DCA]
300 REM Satz von/auf Band lesen/schreibe [65B0]
   n [89A4]
310 FOR i=1 TO 12 [30F2]
320 READ a [EBFE]
330 POKE 40499+i,VAL("&"+a) [816A]
340 NEXT i [0B66]
350 REM Schreibgeschwindigkeit [D1AE]
360 FOR i=1 TO 9 [BAFE]
370 READ a [3608]
380 POKE 40599+i,VAL("&"+a) [16D4]
390 NEXT i [09B8]
400 CALL 40600 [E968]
410 [7A32]
420 REM Menue [BA04]
430 CLS [E088]
440 LOCATE 30,2:PRINT "BACKUP-DISC" [923A]
450 LOCATE 1,4:PRINT "Bitte waehlen Sie: [88A2]
   " [A5D6]
460 LOCATE 20,6:PRINT "Diskette auf Band [4BB0]
   sichern(11SPACE ) (1)" [0ECE]
470 LOCATE 20,8:PRINT "Band auf Diskette [71F0]
   zurueckschreiben(2SPACE ) (2)"
480 LOCATE 20,12:PRINT "Programm beenden [
   (20SPACE ) (4)"
490 LOCATE 20,10:PRINT "Baudrate aendern [
   (20SPACE ) (3)"
500 LOCATE 20,14:PRINT "Eingestellte Bau [
   drate ";br" Baud"
510 a=INKEY$

520 IF a="4" THEN CLS:;DISC:END [BC6E]
530 IF a<>"1" AND a<>"2" AND a<>"3" THEN [C80A]
   510 [CF672]
540 IF a="3" THEN 2290
550 LOCATE 11,18:PRINT "Bitte legen Sie [74F6]
   Band und Diskette ein." [DD18]
560 IF a="1" THEN 590 ELSE 810 [E3C6]
570 [403C]
580 REM auf Band sichern [B150]
590 ;TAPE.OUT:;DISC.IN
600 LOCATE 11,20:PRINT"Druecken Sie PLAY [3AB4]
   und RECORD, dann [ENTER]" [AE18]
610 WHILE INKEY$<>CHR$(13):WEND [9B34]
620 CLS [244E]
630 GOSUB 1850 [2260]
640 REM Adressen poken [60AA]
650 POKE 40988,&B4 [1F7A]
660 POKE 40703,&11 [6484]
670 POKE 40706,&21 [1BBE]
680 POKE 40509,&9E [476C]
690 FOR i=0 TO 7 [E658]
700 FOR j=0 TO 4 [0B68]
710 FOR k=1 TO 9
720 GOSUB 1660:REM Sektor le
   sen
730 GOSUB 1770:REM Sektor ko
   pieren
740 NEXT k [A5B8]
750 NEXT j [8A0A]
760 CALL 40500 :REM Satz auf [DF0A]
   Band schreiben [602C]
770 NEXT i [320C]
780 GOTO 430 [9D5E]
790 [71CE]
800 REM Band auf Diskette zurueckschreib [7B8C]
   en [0146]
810 ;TAPE.IN:;DISC.OUT
820 LOCATE 11,20:PRINT"Druecken Sie PLAY [DF30]
   , dann [ENTER]" [B020]
830 WHILE INKEY$<>CHR$(13):WEND [42B4]
840 GOTO 2120 [5266]
850 REM Adressen poken [2CB2]
860 POKE 40988,&85 [1482]
870 POKE 40703,&21 [8188]
880 POKE 40706,&11 [77AC]
890 POKE 40509,&A1 [3860]
900 FOR i=0 TO 7
910 CALL 40500 :REM Satz vom
   Band lesen [4E1A]
920 FOR j=0 TO 4 [A860]
930 FOR k=1 TO 9 [0D70]

```

Listing. Preiswerte Sicherheitskopien mit »BACKDISC«

Tips & Tricks

```

940          GOSUB 1770:REM Sektor ko
pieren      [2BBE]
950          GOSUB 1660:REM Sektor sc
hreiben     [946B]
960          NEXT k      [AC12]
970          NEXT j      [F112]
980 NEXT i    [4A12]
990 GOTO 430  [0364]
1000         [9110]
1010 REM Data fuer Sektor lesen [CB02]
1020 DATA 21,1c,a0 [96FA]
1030 DATA cd,d4,bc [B234]
1040 DATA 22,1d,a0 [6602]
1050 DATA 79        [5920]
1060 DATA 32,1f,a0 [510C]
1070 DATA 21,20,a0 [C0A0]
1080 DATA 5e        [DE7A]
1090 DATA 23        [F412]
1100 DATA 56        [320E]
1110 DATA 23        [4604]
1120 DATA 4e        [B96A]
1130 DATA 21,30,a0 [929C]
1140 DATA df        [B1D4]
1150 DATA 1d,a0,c0,84 [3094]
1160 REM ld hl,&a01c (Adresse des Ko
mmandos)      [750B]
1170 REM call &bcd4 (Vektor &bcd4) [6DB6]
1180 REM ld (&a01d),hl (Adresse sicher
n)           [C0FE]
1190 REM ld a,c (Romselect nach
a)          [4892]
1200 REM ld (&a01f),a [A006]
1210 REM ld hl,&a020 (Adresse ab der
Sektornummern usw. gepoked werden) [77A4]
1220 REM ld e,(hl) (Laufwerknummer
nach e)      [6FA8]
1230 REM inc hl    [8F30]
1240 REM ld d,(hl) (Spurnummer nac
h d)        [607A]
1250 REM inc hl   [4B34]
1260 REM ld c,(hl) (Sektornummer m
it Offset nach c) [70F8]
1270 REM ld hl,&a030 (Bufferanfang) [A75C]
1280 REM rst &18 (ruft Routine a
uf)         [C116]
1290 REM 4 DATEN-BYTES [D2C8]
1300         [9316]
1310 REM Data fuer Ldir [D3C0]
1320 DATA 01,00,02 [F036]
1330 DATA 11,68,42 [445E]
1340 DATA 21,30,a0 [9E54]
1350 DATA ed,b0    [0C80]
1360 DATA c9
1370 REM ld bc,512 (Bufferlaenge) [BCFE]
1380 REM ld de,17000 (Zieladresse w
ird jeweils neu gepoked) [5752]
1390 REM ld hl,Bufferanfang [E14C]
1400 REM ldir      [782B]
1410 REM ret      [DF6A]
1420         [971C]
1430 REM Data fuer Satz Schreiben [425A]
1440 DATA 3e,16   [B09C]
1450 DATA 11,00,5a [5FAB]
1460 DATA 21,68,42 [D868]
1470 DATA cd,9e,bc [9750]
1480 DATA c9      [2486]
1490 REM ld a,&16 (Satzkennzeichen) [C742]
1500 REM ld de,&5a00 (Laenge) [DE90]
1510 REM ld hl,17000 (Anfang) [C55E]
1520 REM call &bc9e (Satz schreiben) [84AE]
1530 REM ret      [1370]
1540         [9322]
1550 REM Data fuer Schreibgeschwindigkei
t           [12A8]
1560 DATA 21,5d,00 [ECB4]
1570 DATA 3e,0a    [58F8]
1580 DATA cd,68,bc [A4F4]
1590 DATA c9      [098A]
1600 REM ld hl,005d (Laenge des halben
Nullbits)   [7B48]
1610 REM ld a,&a (Vorprueflaenge) [FD8C]
1620 REM call &bc68 (Set Speed CAS) [15CE]
1630 REM ret      [FD72]
1640         [C124]
1650 REM Sektor lesen/schreiben [E702]
1660 POKE 40992,0 :REM Drive 0 [1D90]
1670 POKE 40993,i*5+j :REM Track [3244]
1680 POKE 40994,k+64 :REM Sektor [5DD4]
1690 CALL 40960 :REM Sektor le
sen/schreiben [AEBC]
1700 LOCATE 11,24 [0D68]
1710 PRINT "Track "; [3354]
1720 PRINT USING "##";i*5+j; [4534]
1730 PRINT "{2SPACE }wird kopiert !" [2B62]
1740 RETURN      [8C98]
1750         [9628]

1760 REM Sektor kopieren [90CE]
1770 zs=17000+j*4608+(k-1)*512 [5F52]
1780 zs#=HEX$(zs) [400A]
1790 POKE 40704,VAL("&"+MID$(zs$,3,2)) [F2A2]
1800 POKE 40705,VAL("&"+MID$(zs$,1,2)) [AB90]
1810 CALL 40700 [BB42]
1820 RETURN      [7D96]
1830         [C026]
1840 REM Directory auslesen und als Date
i auf Band schreiben [8070]
1850 m=PEEK(&BB5A):POKE &BB5A,&C9:CAT:PO
KE &BB5A,m [56F2]
1860 num=PEEK(43282) [7FEA]
1870 b=14953 [AB2A]
1880 FOR i=0 TO 63 [8634]
1890 dir$(i)=STRING$(11," ") [7714]
1900 NEXT i [BA64]
1910 FOR i=0 TO num [0EF6]
1920 POKE @dir$(i)+1,b-(INT(b/256)*2
56) [8862]
1930 POKE @dir$(i)+2,INT(b/256) [CC18]
1940 b=b+14 [7BFE]
1950 NEXT i [846E]
1960 REM Directory anzeigen [E434]
1970 FOR i=0 TO num [4302]
1980 IF ASC(LEFT$(dir$(i),1))=0 THEN
2030 dir$(i)=LEFT$(dir$(i),8)+". "+RI
GHT$(dir$(i),3) [E64A]
2000 PRINT dir$(i), [6AB0]
2010 sdir$(i)=dir$(i) [AB9A]
2020 NEXT [FD46]
2030 OPENOUT "!dir" [14A6]
2040 FOR j=0 TO i-1 [93DC]
2050 PRINT#9,sdir$(j) [B51A]
2060 NEXT j [A362]
2070 CLOSEOUT [4FAE]
2080 RETURN [C194]
2090 END [0584]
2100         [BC14]
2110 REM Directory einlesen und auf Bild
schirm ausgeben [A18C]
2120 CLS [1ABE]
2130 OPENIN"!dir" [8BA6]
2140 z=0 [2D9C]
2150 WHILE NOT EOF [15DB]
2160 z=z+1 [FDEC]
2170 INPUT#9,d$ [FB14]
2180 PRINT #0,USING"\{12SPACE }";d$
: [BE2E]
2190 IF z=5 THEN z=0:PRINT [6048]
2200 WEND [3E24]
2210 CLOSEIN [77E4]
2220 PRINT:PRINT:PRINT"Wirklich zuruecks
chreiben auf Diskette J/N ?" [A6BA]
2230 a=INKEY$:IF a="" THEN 2230 [6C54]
2240 a=UPPER$(a) [41D0]
2250 IF a<>"J" AND a<>"N" THEN 2230 [FB6E]
2260 IF a="J" THEN GOTO 860 ELSE GOTO 43
0 [7208]
2270         [9724]
2280 REM Baudrate aendern [362A]
2290 CLS [919E]
2300 PRINT "Folgende Baudraten sind moeg
lich:" [2840]
2310 PRINT [C7E6]
2320 FOR i=1 TO 8 [19C0]
2330 PRINT "{12SPACE }";i".", (i+2)*4
00;" Baud" [4216]
2340 NEXT i [8662]
2350 PRINT:PRINT"Bitte druecken Sie die
Ziffer, die vor der gewünschten Ba
udrate steht." [0884]
2360 a=INKEY$:IF a="" THEN 2360 [EC64]
2370 n=ASC(a) [1D40]
2380 IF n<49 OR n>56 THEN 2360 [EECA]
2390 nb12=INT(1000000/((n-46)*1200)+0.5)
[FD4E]
2400 hb=INT(nb12/256) [AAC6]
2410 lb=nb12-hb*256 [203C]
2420 POKE 40601,lb:POKE 40602,hb [BD4E]
2430 CALL 40600 [753E]
2440 br=(n-46)*400 [F51E]
2450 GOTO 430 [5EB6]
2460 END [4286]

```

Listing. Preiswerte Sicherheitskopien mit »BACKDISC« (Schluß)

Versteckte Bytes



Ein Speicherbereich, der auch bei Reset nicht gelöscht wird, ist ideal für Hilfsroutinen. Im Parameterbereich des Basic-Interpreters findet sich dafür ein freier Platz.

Beim Schneider CPC 464 sind 512 Byte ab der Adresse AE8B hex für Parameter des Basic-Interpreters reserviert. Diese Parameter (für GOSUB-Aufrufe, FOR...NEXT... - und WHILE...WEND-Schleifen, Adressen von AFTER- und EVERY-Befehlen und so weiter) liegen auf einem Stapel, der mit einem Zeiger (an den Adressen B08B und B08C hex) verwaltet wird.

Dieser Bereich zählt zum Betriebssystem und wird deshalb auch bei einem Reset nicht gelöscht. Probieren Sie einmal folgende Befehlsfolge aus: Mit »PRINT PEEK(&AF10)« (direkt nach dem Einschalten) bekommen Sie auf dem Bildschirm den Wert »0« ausgegeben. Mit »POKE &AF10,10« steht an dieser Stelle eine »10« (testen mit »PRINT PEEK(&AF10)«). Setzen Sie jetzt den Computer durch gleichzeitiges Drücken der drei Tasten Ctrl, Shift und Esc zurück. Mit »PRINT PEEK(&AF10)« bekommen Sie wieder die »10« auf den Bildschirm und nicht eine »0«, wie nach dem Einschalten.

Normalerweise sollte man in diesen Bereich des Betriebssystems nichts schreiben, da das zum System-Absturz führen kann. Der Basic-Interpreter des Schneider CPC 464 belegt aber nur 119 der 512 reservierten Bytes tatsächlich und so kann man ab der Adresse AFOO hex (bis B089 hex) eigene Maschinencode-Programme ablegen. 393 Bytes sind also frei für Hilfsroutinen, die erst mit Ausschalten des Computers gelöscht werden. Da der Interpreter seine Parameter von unten her (das heißt ab Adresse AE8B hex) ablegt, ist es sinnvoll, die Programme möglichst weit oben in den geschützten Bereich zu stellen. Die letzte Adresse sollte zwischen B080 und B089 hex liegen.

(Oliver Harms)

Gläserne Firmware



Ein RSX-Befehl — und schon kann jede Routine im Speicher des Schneiders problemlos untersucht werden.

Wer kennt das Problem nicht? Man will eine Firmware-Routine kurz ausprobieren, muß aber verschiedene Register mit bestimmten Werten laden. Dies geht nur mit einem kleinen Maschinencode-Programm — oder mit dem RSX-Befehl »USR«.

Als Parameter muß die Start-Adresse übergeben werden, damit die betreffende Routine ausgeführt wird. Dies kann man zwar auch mit dem Basic-Befehl »CALL Adresse« erreichen, aber dieser erlaubt es nicht, die Register des Z80 zu laden. Die meisten Firmware-Programme arbeiten aber in Abhängigkeit dieser Register, so daß es zum Ausprobieren (und Kennenlernen) der Routinen wichtig ist, die Register zu manipulieren.

Den »USR«-Befehl gibt es in zwei verschiedenen Formaten. Mit »USR,Adresse« wird eine Routine ohne vorheriges Setzen der Register abgearbeitet. Befindet sich die Adresse zwischen 4000 und BFFF hex, so liegt das Maschinenprogramm im RAM, in allen anderen Fällen (0000 bis 3FFF hex und C000 bis FFFF hex) im ROM. Nach Ende der Routine befindet sich der Computer wieder im normalen Eingabemodus.

Mit »USR,Adresse,A,B,DE,HL,x« werden die Register A, B, DE und HL zuerst mit den übergebenen Werten geladen, bevor die Startadresse aufgerufen wird. In Abhängigkeit von der Variablen x werden die Registerwerte nach dem Bearbeiten der Routine ausgegeben (x=1) oder nicht (x=0). Ein Beispiel: »USR,&BB18« wartet auf einen Tastendruck. »USR,&BB18,0,0,0,0,1« wartet auf einen Tastendruck und gibt im Register A den ASCII-Wert der gedrückten Taste aus. Was für Werte in den Registern sinnvoll sind und welche Informationen zurückgegeben werden, das entnehmen Sie dem Firmware-Buch für den Schneider. Das Programm wird mit dem Basic-Lader installiert und beginnt bei Adresse A4CA hex.

(Erik Pfeiffer)

```

10 '***** [9FCC]
20 '* (c) Erik Pfeiffer * [B846]
30 '* Spreenweg 5, 2000 Noderstedt 1 * [1BFC]
40 '***** [E9D2]
50 FOR i=1 TO 434:READ a:b=b+a:NEXT:IF b
   <>49749 THEN PRINT "Fehler!":END [526C]
60 PRINT"OK":RESTORE [679E]
70 b=&A4CA:MEMORY b-1:FOR i=b TO &A67B:R
   EAD a:POKE i,a:NEXT:CALL b:NEW [7B8A]
80 DATA 1,211,164,33,119,166,195,209,188
   ,215,164,24,4,85,83,210,0,254,6,40,59
   ,254,1,32,16,221,110,0,221,102,1,34,
   ,117,166,62,0,50,123,166,24,72,33,0,16
   ,6,26,126,205,90,187,35,16,249,201,8
   ,7,114,111,110,103,32,110,117,109,98,1
   01 [373E]
90 DATA 114,32,111,102,32,112,97,114,97,
   ,109,101,116,101,114,115,221,126,0,50,
   ,123,166,221,110,10,221,102,11,34,117,
   ,166,221,110,2,221,102,3,221,94,4,221,
   ,86,5,221,70,6,221,126,8,223,117,166,5
   ,0,117,166,58,123,166,254,0,200,58,117
   ,166,229,213 [B6E0]
100 DATA 197,245,62,4,50,105,166,62,13,2
   ,05,90,187,62,10,205,90,187,62,65,205
   ,90,187,62,32,205,90,187,62,61,205,9
   ,0,187,241,111,38,0,245,205,112,166,2
   ,41,111,38,0,205,47,166,62,10,205,90,
   ,187,62,13,205,90,187,62,66,205,90,18
   ,7,62,32,205 [5F70]
110 DATA 90,187,62,61,205,90,187,193,104
   ,38,0,197,205,112,166,193,104,38,0,2
   ,05,47,166,62,2,50,105,166,62,10,205,
   ,90,187,62,13,205,90,187,62,68,205,90
   ,187,62,69,205,90,187,62,61,205,90,1
   ,87,209,213,235,205,57,166,225,205,23
   ,8,165,225,62,10 [443E]
120 DATA 205,90,187,62,13,205,90,187,62,
   ,72,205,90,187,62,76,205,90,187,62,61
   ,205,90,187,229,205,57,166,225,205,
   ,238,165,201,205,102,166,62,38,205,90
   ,187,17,0,16,205,15,166,17,0,1,205,
   ,15,166,17,16,0,205,15,166,17,1,0,205
   ,15,166,201 [7D18]
130 DATA 205,193,189,77,6,0,33,31,166,9,
   ,126,205,90,187,235,201,48,49,50,51,5
   ,2,53,54,55,56,57,65,66,67,68,69,70,2
   ,05,102,166,62,38,205,90,187,24,201,2
   ,05,102,166,17,16,39,205,91,166,17,23
   ,2,3,205,91,166,17,100,0,205,91,166,1
   ,7,10 [CAEC]
140 DATA 0,205,91,166,17,1,0,205,91,166,
   ,201,205,193,189,125,198,48,205,90,18
   ,7,235,201,62,32,6,2,205,90,187,16,25
   ,1,201,205,102,166,24,211,16,187,252,
   ,166,211,164,1 [8952]

```

Listing. Der Basic-Lader für »USR«

Basic-Zeilen automatisch erzeugt



Eine Routine, die neue Basic-Zeilen automatisch in das laufende Programm einfügt, ist für

viele Zwecke praktisch und wichtig.

```

10 ***** [9FCC]
20 * (c) Bernhard Straub * [C490]
30 *An der Tagmess 13, 7600 Offenburg* [2CAE]
40 ***** [E9D2]
50 MODE 2 [F7F8]
60 PRINT STRING$(80,""); [CD78]
70 PRINT STRING$(32,"");"{2SPACE }P U T
  L I N E{2SPACE }";STRING$(31,""); [FE3E]
80 PRINT STRING$(80,""); [B17C]
90 PRINT STRING$(28,"");"{2SPACE }(c) 1
  985 by BERNISOFT{2SPACE }";STRING$(27
  ,""); [C678]
100 PRINT STRING$(80,"") [E258]
110 PRINT"RSX Basic-Befehl PUTLINE zur" [1C6E]
120 PRINT"automatischen Programmzeilen" [35CA]
130 PRINT"Generierung." [1E5C]
140 PRINT [F184]
150 PRINT"Basicloader erzeugt Maschinenc
  ode." [600A]
160 PRINT"Startadresse: &A400{5SPACE }En
  dresse: &A490" [EF9C]
170 PRINT [428A]
180 PRINT"Befehlsformat:" [BE0C]
190 PRINT"!PUTLINE,<Zeilennummer>,<Stri
  ngvariable>" [06D8]
200 PRINT:PRINT [A40C]
210 PRINT "Bitte nicht stoeren, Maschine
  ncode wird gepokt!!!" [D0A4]
220 FOR adresse=&A400 TO &A48D [8DAB]
230 READ mc$ [D0CA]
240 mc=VAL("&"+mc$) [9D00]
250 pruef=pruef+mc [ED66]
260 POKE adresse,mc [43D4]
270 NEXT [60F0]
280 IF pruef<>16136 THEN PRINT "Sie habe
  n leider einen Fehler beim Abtippen
  gemacht, bitte ueberpruefen!":END [024C]
290 PRINT "Alles okay!":PRINT [27B2]
300 INPUT "Wollen Sie den Maschinencode
  absAVEN ? (j/n) ",frage$ [0A9A]
310 IF UPPER$(frage$)<>"J" THEN END [95B6]
320 INPUT "Welche Geschwindigkeit ? (0/1
  ) ",geschw [99CE]
330 IF geschw<>0 AND geschw<>1 THEN GOTO
  320 [BC58]
340 SPEED WRITE geschw [72E8]
350 SAVE "PUTLINE",b,&A400,&90 [B588]
360 PRINT "Fertig !!!" [DD1C]
370 END [A822]
380 DATA 01,09,A4,21,16,A4,C3,D1,BC,0E,A
  4,C3,1A,A4,50,55 [E71A]
390 DATA 54,4C,49,4E,C5,00,00,00,00,00,D
  F,1D,A4,20,A4,FD [47FA]
400 DATA FE,02,28,05,1E,02,C3,94,CA,2A,B
  B,B0,22,8E,A4,DD [1974]
410 DATA 66,03,DD,6E,02,CD,0D,FF,CD,82,E
  E,E5,0E,00,7E,B7 [4BFA]
420 DATA 28,04,23,0C,18,F8,E1,06,00,11,A
  4,AC,ED,B0,3E,20 [4AFA]
430 DATA 12,13,DD,66,01,DD,6E,00,E5,DD,E
  1,DD,4E,00,DD,66 [FCA2]
440 DATA 02,DD,6E,01,ED,B0,AF,12,21,3A,B
  D,36,F1,23,36,C9 [9970]
450 DATA 21,A4,AC,CD,B8,C0,21,3A,BD,36,C
  F,23,36,98,2A,BE [6CAA]
460 DATA A4,22,8B,B0,ED,7B,77,AE,2A,75,A
  E,C3,8B,DD [CA90]
  
```

Listing 1. Ein Schritt in Richtung »Künstliche Intelligenz«:
»PUTLINE«

Der RSX-Befehl »PUTLINE« erzeugt aus einem String eine Programmzeile und fügt diese automatisch in die laufende Routine ein. Damit kann sich ein Programm selbst verändern. Die RSX-Anweisung braucht als Parameter nur die Zeilennummer des neuen Befehls und in einem String, die zu kreierende Anweisung. Die Syntax ist damit denkbar einfach: »IPUTLINE,Zeilennummer,String«. Zuvor muß die Routine für PUTLINE allerdings mit »CALL &A400« eingebunden werden.

Der Basic-Lader aus Listing 1 installiert die RSX-Anweisung. Listing 2 zeigt ein Beispiel (anhand eines DATA-Generators), wie man den PUTLINE-Befehl sinnvoll nutzen kann. Für alle, die diese Routine verschieben wollen: Die Maschinencode-Befehle liegen ab der Adresse A400 hex.

Mit dem Befehl »PUTLINE« kann man Programme schreiben, die sich selbst verändern. Beispielsweise könnte ein Grafik-Programm für allgemeine mathematische Funktionen den zu bearbeitenden Graphen als »DEF FN«-Befehl in das Programm einfügen. Die Zeilen dazu müßten so aussehen:

```

100 INPUT "Funktion f(x)=",f$
110 Zeile$="DEF FN f(x)=" +f$
120 IPUTLINE,200,Zeile$
  
```

Nach »RUN« steht die Funktion in Zeile 200.

(Bernhard Straub)

```

10 'DATA maker [D0E4]
20 '(C) 9/85 by Wolfgang A. Jaeger [0FFE]
30 ' Postfach 1731 [5A9C]
40 ' D-7600 Offenburg 1 [48F0]
50 ' Tel. 0781/78268 [CA5A]
60 MEMORY &A400-1:MODE 2:LOCATE 1,8 [8F60]
70 LOAD "!putline":CALL &A400 [25FC]
80 PRINT:INPUT"Anfangsadresse{10SPACE }="
  ",von [70C8]
90 von=von-65536*(von<0) [D1B8]
100 PRINT:INPUT"Endadresse{14SPACE }=" ",
  bis [5CA2]
110 bis=bis-65536*(bis<0) [E28C]
120 IF bis<von THEN LOCATE 1,12:GOTO 100 [D1B4]
130 PRINT:INPUT"1. DATA-Zeilennummer {4SP
  ACE }=" ",nr [0AF2]
140 IF nr<=410 THEN LOCATE 1,12:PRINT;:G
  OTO 130 [E804]
150 PRINT:INPUT"Abstand der DATA-Zeilen
  =" ",ab [2D46]
160 laenge=bis-von+1:zeilenzahl=INT(laen
  ge/10)-(laenge/10<>INT(laenge/10)) [4116]
170 DIM z$(laenge-1),x$(zeilenzahl) [B974]
180 CLS:LOCATE 32,12:PRINT"-- bitte warte
  n --" [331E]
190 FOR i=0 TO laenge-1 [B588]
200 z$(i)="" +HEX$(PEEK(von+i),2) [BBE2]
210 NEXT [6CE4]
220 FOR k=1 TO zeilenzahl:x$(k)="DATA " :
  NEXT [8CE6]
230 k=0 [0F1A]
240 FOR i=0 TO laenge-1 [D080]
250 IF i MOD 10=0 THEN k=k+1 [6436]
260 IF i MOD 10=9 OR i=laenge-1 THEN
  x$(k)=x$(k)+z$(i):GOTO 280 [FD48]
270 x$(k)=x$(k)+z$(i)+", " [3888]
280 NEXT i [4F04]
290 k=1 [E328]
300 !PUTLINE,ab*(k-1)+nr,@x$(k) [61A4]
310 IF k<zeilenzahl THEN k=k+1:GOTO 300 [0E88]
320 PEN 1:CLS:CLEAR:END [B36E]
  
```

Listing 2. Mit »PUTLINE« kann der Datengenerator
»DATAGEN« sehr einfach realisiert werden



Vier statt zwei



Um vier Farben in ein Zeichen zu bekommen muß man den Aufbau des Bildspeichers kennen. Die Routine »Farbenpoke« macht es einfach, den Bildschirm farbig zu gestalten.

Nach dem Befehl »MODE 1« kann man vier verschiedene Farben gleichzeitig auf den Bildschirm bringen, mit den Basic-Befehlen PEN und PAPER aber nur zwei pro Zeichenmatrix (8 x 8 Punkte). Der Befehl POKE erlaubt das direkte Beschreiben des Bildspeichers und das auch mit vier Farben in einem Zeichen.

Den Zusammenhang zwischen dem Wert eines Bytes im Bildspeicher und der Farbe auf dem Monitor erkennen Sie am leichtesten mit dem Programm aus Listing 1. Wie Sie sehen, liegen Adressen untereinander liegender Pixelreihen genau 2048 (800 hex) Bytes auseinander. Im Modus 1 werden zwei Adressen für eine Pixelreihe benötigt.

Ändern Sie nun die Farbe (beispielsweise mit »PEN 3« oder »PAPER 2«) und lassen die zwei Zeilen noch einmal bearbeiten, so ändern sich zwar die Werte, aber das »a« bleibt immer erkennbar. Am einfachsten erkennt man den Zusammenhang in Bild 1. Die niederwertigen Bits (0 bis 3) werden unter die höherwertigen (4 bis 7) geschrieben. Eine »1« zeigt einen gesetzten Wert an, eine »0« einen ungesetzten. Die Wertigkeit eines Bits berechnet sich aus der Bitnummer zur Basis 2 potenziert (zum Beispiel hat Bit 6 die Wertigkeit $2^6 = 64$). Bildet man nun die Spaltensummen (immer wenn ein Bit gesetzt ist, dessen Wert addiert) und zählt diese zusammen, so hat man den Wert, den der Computer zur Farbinterpretation braucht. In unserem Beispiel bekommt das Byte den Wert 83. Mit dem Befehl »MODE 1:POKE &C000,83:LOCATE 1,2« sehen Sie in der oberen linken Ecke des Bildschirms vier Punkte in den Farben 0, 1, 2 und 3. In der Spalte a ist weder oben, noch unten ein Bit gesetzt (ergibt Farbe 0), in den Spalten b und c ist jeweils ein Bit (oben oder unten) gleich 1 (Farbe 1 oder 2), während in der Spalte d beide Bits gesetzt sind (Farbe 3).

Mathematisch kann man das ganze so ausdrücken:

Farbe 0	erscheint nur bei der Spaltensumme 0
Farbe 1	erscheint, wenn sich die Spaltensumme als ein Vielfaches von 16 darstellen läßt
Farbe 2	erscheint, wenn die Spaltensumme 1, 2, 4 oder 8 beträgt
Farbe 3	erscheint, wenn die Spaltensumme ein Vielfaches von 17 beträgt

Mit diesem Wissen können Sie sehr einfach vierfarbige Zeichen darstellen. Damit Sie aber nicht sämtliche Adressen und Werte umständlich berechnen müssen, nimmt Ihnen Listing 2 diese Arbeit ab. (Thomas Dröge)

Spalte	a	b	c	d
Bitnummer	7	6	5	4
Wert	0	1	0	1
Bitnummer	3	2	1	0
Wert	0	0	1	1

Bild 1. Die rechte und die linke Hälfte eines Bytes ergeben die Farbe für vier Bildpunkte

```

10 MODE 1:LOCATE 1,1:PRINT "a":PRINT [94D4]
20 FOR t=&C000 TO &FB00 STEP &800:PRINT
   BIN$(PEEK(t),8),BIN$(PEEK(t+1),8):NEX
   T t [8910]

```

Listing 1. Im Bildspeicher sind die Farben codiert abgelegt

```

10 ***** [9FCC]
20 *(c) Thomas Droege * [CB66]
30 * Hohenzollernstr. 18 * [068C]
40 * 4700 Hamm 1 * [33FC]
50 ***** [87D4]
60 DIM w(2,8):MODE 1:FOR i=1 TO 8:LOCATE
   1,1:PRINT i". Pixelreihe des Zeichen
   s":FOR t=1 TO 4:LOCATE 1,5 [CBCA]
70 PRINT"Farbe fuer "t". Pixel "":INPUT
   farbe:IF farbe<0 OR farbe>3 THEN 70 E
   LSE ON farbe+1 GOTO 110,80,90,100 [2ECE]
80 s=2^(8-t):GOTO 110 [4A2E]
90 s=2^(4-t):GOTO 110 [9E2B]
100 s=2^(4-t)+2^(8-t) [20FB]
110 w(1,i)=w(1,i)+s:s=0:NEXT:s=0:FOR t=5
   TO 8:LOCATE 1,5 [F47B]
120 PRINT"Farbe fuer "t". Pixel "":INPUT
   farbe:IF farbe<0 OR farbe>3 THEN 12
   0 ELSE ON farbe+1 GOTO 160,130,140,1
   50 [6742]
130 s=2^(12-t):GOTO 160 [0EE6]
140 s=2^(8-t):GOTO 160 [8592]
150 s=2^(8-t)+2^(12-t) [A060]
160 w(2,i)=w(2,i)+s:s=0:NEXT:s=0:NEXT [8DAC]
170 LOCATE 1,7:PRINT"welche Koordinaten
   soll Ihr Zeichen haben?":INPUT"x-Ko
   ordinate "":x:IF x<1 OR x>40 THEN 170 [F856]
180 INPUT"y-Koordinate "":y:IF y<1 OR y>2
   5 THEN 180 [F4DA]
190 MODE 1:PRINT"das Zeichen "":sp=4915
   1+(y-1)*80+x*2-1:FOR i=1 TO 8:POKE &
   C028+(i-1)*800,w(1,i):POKE &C029+(i
   -1)*800,w(2,i):NEXT [9406]
200 LOCATE 1,3:PRINT"die Werte "":FOR i=1
   TO 8:LOCATE 1,4+:PRINT"&X"BIN$(w(1
   ,i),8):PRINT"{2 SPACE}"w(1,i):LOCAT
   E 20,4+i:PRINT"&X"BIN$(w(2,i),8)" {2
   SPACE}"w(2,i):NEXT [BC7A]
210 PRINT:PRINT"Die Speicherstellen "":
   PRINT:FOR i=1 TO 8:PRINT sp+(i-1)*20
   48"{2 SPACE}"HEX$(sp+(i-1)*2048)" {2
   SPACE}"sp+(i-1)*2048+1"{2 SPACE}"HEX
   $(sp+(i-1)*2048+1):NEXT [FFD6]

```

Listing 2. Ohne viel zu berechnen, kann man mit »Farbenpoke« vierfarbige Zeichen auf den Bildschirm bringen

Wo ist das Programm?



Auf irgendeiner Diskette ist das Programm, das man sucht. Aber auf welcher? Mit einer sortierten Liste sämtlicher Dateien gehört diese Suche der Vergangenheit an.

Der CAT-Befehl des Schneider wird bei diesem Listing für neue Aufgaben genutzt. Mit ihm lassen sich Inhaltsverzeichnisse von verschiedenen Disketten lesen und sortiert ausdrucken. Die Programmnamen werden alphabetisch geordnet, können gespeichert und die Liste jederzeit aktualisiert werden. Im Ausdruck (auf dem Drucker oder dem Bildschirm) findet man zu jedem Namen die Diskette, auf der das Programm steht. (Werner Renziehausen)

Tips & Tricks

```

10 ' Sortierte CAT-Liste aller Disketten
    auf dem Drucker [9898]
20 ' [6052]
30 ' von Werner Renziehausen
    (Juni 1985) [D07E]
40 ' Hauptstr.32, 3360 Osterode 1 [010E]
50 ' [6158]
60 CLEAR:OPENOUT"Dummy":MEMORY (HIMEM-1)
:CLOSEOUT [CE36]
70 MODE 1 [FDFA]
80 CALL &BC02: PEN 1: PAPER 0: BORDER 1 [B0D4]
90 PRINT"{2 SPACE}*****" [D76C]
***** [F62B]
100 PRINT"{2 SPACE}*{33 SPACE}*" [496E]
110 PRINT"{2 SPACE}*{3 SPACE}Prgm: Sorti
    erte Liste aller{3 SPACE}*" [B82B]
120 PRINT"{2 SPACE}*{9 SPACE}Diskettenfi
    les{10 SPACE}*" [0D2E]
130 PRINT"{2 SPACE}*{33 SPACE}*" [26C4]
140 PRINT"{2 SPACE}*****" [C9A4]
***** [E08B]
150 PRINT"{CTRL J}{CTRL J}{CTRL J}{2 SPA
    CE}Auswahl: [2D64]
160 PRINT"{CTRL J}{CTRL J}{2 SPACE}1{2 S
    PACE}Neueingabe der Disketten [C5C2]
170 PRINT"{CTRL J}{2 SPACE}2{2 SPACE}Akt
    ualisierung der Diskettenliste [A2CA]
180 PRINT"{CTRL J}{2 SPACE}3{2 SPACE}Abs
    peicherung der Liste [CCBE]
190 PRINT"{CTRL J}{2 SPACE}4{2 SPACE}Aus
    druck der Liste [835C]
200 PRINT"{CTRL J}{2 SPACE}5{2 SPACE}Anz
    eige der Liste [D9BC]
210 PRINT"{CTRL J}{2 SPACE}6{2 SPACE}Lis
    te der Diskettenbezeichnungen [C796]
220 z$=INKEY$: IF z$="" THEN 220 [5A3C]
230 z=ASC(z$)-48
240 IF z<1 OR z>6 THEN 220
250 ON z GOSUB 300,2050,1170,950,940,232
    0 [AC5E]
260 GOTO 70 [B5F0]
270 ' [83C0]
280 ' Neueingabe der Disketten [6974]
290 ' [ADC4]
300 CLS: CLEAR:OPENOUT"Dummy":MEMORY (HI
    MEM-1):CLOSEOUT [920B]
310 PRINT" ** NEUEINGABE DER DISKETTEN *
    *" [D40B]
320 PRINT"{CTRL J}{2 SPACE}Anzahl der zu
    erfassenden [684B]
330 INPUT "{2 SPACE}Diskettenseiten: ";di
    scseit [45D4]
340 discseit=INT(discseit): IF discseit<
    1 THEN 320 [8364]
350 INPUT "{CTRL J}{2 SPACE}Datum: ";datu
    m$ [C64A]
360 DIM file$(64,discseit,2): ',0-Name
    ,1-Zusatz ,2-Laenge [2522]
370 DIM zeiger(64*discseit,2) [F9AA]
380 zz=0 [973B]
390 WHILE disc<discseit [731B]
400 z=0: disc=disc+1 [52EB]
410 PRINT"{CTRL J}Bezeichnung der einge
    legten": INPUT"Diskettenseite: ";file
    $(0,disc,0) [71F8]
420 GOSUB 430: GOTO 600 [9FAC]
430 PEN 3: CAT: PEN 1: bloecke=0 [02D2]
440 FOR j=&9E7C TO &A67A STEP 14 [50FA]
450 IF PEEK(j+1)<32 THEN 560 [2E1E]
460 z=z+1 [2F0E]
470 FOR k=1 TO 8: i=j+k [1EEC]
480 file$(z,disc,0)=file$(z,disc,0)+C
    HR$(PEEK(i)) [AB52]
490 NEXT k [02BE]
500 FOR k=9 TO 11: i=j+k [8444]
510 file$(z,disc,1)=file$(z,disc,1)+C
    HR$(PEEK(i)) [CA4A]
520 NEXT k [61B2]
530 file$(z,disc,2)=STR$(PEEK(j+12)) [E59A]
540 bloecke=bloecke+PEEK(j+12) [C6F2]
550 NEXT j [1046]
560 maxblocks=PEEK(&A895+(PEEK(&A700)*6
    4))-1 [CA02]
570 file$(0,disc,1)=STR$(maxblocks-bloe
    cke) [C17C]
580 zz=zz+z: file$(0,disc,2)=STR$(z) [2430]
590 RETURN [3A7C]
600 WEND [49CB]
610 GOSUB 640: GOSUB 720: GOTO 70 [073C]
620 ' [0EBE]
630 ' Zeiger-Grundsetzung fuer die Adres
    sierung der files [61C2]
640 p=0: d=1 [DB86]
650 FOR i=1 TO zz [A9E0]
660 p=p+1: IF p>VAL(file$(0,d,2)) THEN
    p=1: d=d+1 [C0F6]
670 FOR k=0 TO 2 [08A2]
680 zeiger(i,k)=d+p/100 [A362]
690 NEXT: NEXT [3F2E]
700 RETURN [AC2E]
710 ' [09BE]
720 ' Unterprogramm - Sortieren [1F36]
730 ' [E7C2]
740 CLS: PEN 2: PRINT"{CTRL J}";zz;" Fil
    es sind gespeichert" [7946]
750 PRINT"{CTRL J}{CTRL J} CPC sortiert
    Programme nach:" [A1E2]
760 PRINT"{CTRL J}{2 SPACE}- dem Namen [5090]
770 FOR i=0 TO 1 [285E]
780 IF i=1 THEN PRINT"{CTRL J}{2 SPACE}
    - dem Zusatz" [0A46]
790 z=1:schleianz=0 [0C60]
800 WHILE z<>0 [362A]
810 z=0: schleianz=schleianz+1: LOCATE
    20,7+i*2: PRINT(";schleianz;") [1666]
820 FOR j=1 TO zz-1 [F71C]
830 d1=INT(zeiger(j,i)): d2=INT(zeige
    r(j+1,i)) [B334]
840 p1=(zeiger(j,i)-d1)*100: p2=(zeig
    er(j+1,i)-d2)*100 [77B0]
850 IF file$(p1,d1,i)>file$(p2,d2,i)
    THEN z=zeiger(j,i):
    zeiger(j,i)=zeiger(j+1,i)
    : zeiger(j+1,i)=z [9DC4]
860 NEXT j [5C8E]
870 WEND [A11A]
880 IF i=0 THEN FOR j=1 TO zz: zeiger(j
    ,1)=zeiger(j,0): zeiger(j,2)=zeiger(
    j,0) : NEXT [BDD0]
890 NEXT i [4312]
900 RETURN [4232]
910 ' [0DC2]
920 ' Ausgaberroutinen der sortierten Fi
    les - Bildschirm und Drucker [B9E0]
930 ' [0FC6]
940 aus=0:CLS:GOSUB 1580:IF zz=0 THEN GO
    SUB 1390:GOTO 1000 ELSE GOTO 1000 [9B68]
950 aus=8:MODE 1:IF zz=0 THEN GOSUB 1390
    ELSE PRINT"{CTRL J}{CTRL J}{2 SPACE}
    Datum:{2 SPACE}";:INPUT datm$ [8F88]
960 IF datm$<>"" THEN datum$=datm$ [F956]
970 GOSUB 1580 [E05C]
980 PRINT"{CTRL J}{CTRL J} Drucker einsc
    halten - Taste druecken":CALL &BB06 [D852]
990 PRINT#aus,CHR$(27);CHR$(64);CHR$(27)
    ;"E";SPACE$(16)"Programmliste vom "
    ; datum$;CHR$(27);"F" [5AC4]
1000 IF aus=0 THEN MODE 2:PEN 1:PAPER 0:
    BORDER 1:INK 1,0:INK 0,13 [1554]
1010 IF wahl=2 THEN 2410 [63C4]
1020 PRINT#aus,CHR$(27);"N";CHR$(10) [8AFC]
1030 lbrnt=PEEK(&B08B): hbrnt=PEEK(&B0
    8C) [52D2]
1040 FOR j=1 TO zz [C136]
1050 d1=INT(zeiger(j,wahl)):p1=INT((zei
    ger(j,wahl)-d1)*100+0.5) [9F4C]
1060 PRINT#aus,SPACE$(10);USING"####";j
    ; [ECBE]
1070 PRINT#aus,"{2 SPACE}";file$(p1,d1,
    0);". ";file$(p1,d1,1);"{2 SPACE}";
    [ECEA]
1080 PRINT#aus,USING"##";VAL(file$(p1,d
    1,2)); [597A]
1090 PRINT#aus,"K";SPACE$(8);USING"##";
    p1; [F0EB]
1100 PRINT#aus,"{2 SPACE}";file$(0,d1,0)
    ; [4090]
1110 IF aus=0 AND j/25-INT(j/25)=0 THEN
    CALL &BB06 [7D18]
1120 NEXT:IF aus=8 THEN RETURN [A932]
1130 PRINT"{CTRL J}Ende der Liste - Weit
    er, Taste druecken":CALL &BB06:RETU
    RN [33E6]
1140 ' [B31A]
1150 ' Programmliste auf Diskette speich
    ern [605C]
1160 ' [B51E]
1170 MODE 1:LOCATE 1,10:PRINT"{7 SPACE}D
    aten-Diskette einlegen - [9A04]
1180 PRINT"{CTRL J}{CTRL J}{12 SPACE}Tas
    te druecken [003E]
1190 CALL &BB06 [F86E]
1200 OPENOUT"disclist.dat" [872E]
1210 WRITE#9,discseit,z,datum$ [C7B4]
1220 FOR k=0 TO discseit [21FE]
1230 FOR i=0 TO VAL(file$(0,k,2)) [B968]
1240 FOR l=0 TO 2 [9C38]
1250 WRITE#9,file$(i,k,l) [5510]
1260 NEXT:NEXT:NEXT [1B34]
1270 FOR i=0 TO zz [B23C]
1280 FOR k=0 TO 2 [6BF6]
1290 WRITE#9,zeiger(i,k) [152C]
1300 NEXT:NEXT [C338]
1310 CLOSEOUT [4AA6]
1320 LOCATE 5,20:a$="n": INPUT" 'BAK'-Fil
    e loeschen (j/n): ";a$ [8652]
1330 IF a$="j" THEN b$="disclist.bak": I
    ERA,@b$: GOTO 1350 [E2A6]
1340 IF a$<>"n" THEN 1320 [6EFA]

```

Listing. Mit dieser Routine findet man jedes Programm

```

1350 RETURN [AA92]
1360 ' [B522]
1370 ' Programmliste von Diskette einles
en [ACB6]
1380 ' [B826]
1390 neudisc=0 [6B8A]
1400 MODE 1:LOCATE 1,10:PRINT"(7 SPACE)D
aten-Diskette einlegen -: PRINT"(C
TRL J){CTRL J}{12 SPACE}Taste druec
ken": CALL &BB06 [D3EE]
1410 OPENOUT"Dummy":MEMORY (HIMEM-1):CLO
SEOUT [01B4]
1420 OPENIN"disclist.dat": INPUT#9,discs
eit,zz,datum$ [771E]
1430 DIM file$(64,discseit+neudisc,2),ze
iger(64*(discseit+neudisc),2) [D51A]
1440 FOR k=0 TO discseit [6806]
1450 INPUT#9,file$(0,k,0),file$(0,k,1),
file$(0,k,2) [324A]
1460 FOR i=1 TO VAL(file$(0,k,2)) [F374]
1470 FOR l=0 TO 2 [6042]
1480 INPUT#9,file$(i,k,l) [1624]
1490 NEXT:NEXT:NEXT [153E]
1500 FOR i=0 TO zz [9934]
1510 FOR k=0 TO 2 [54F6]
1520 INPUT#9,zeiger(i,k) [F12E]
1530 NEXT:NEXT [BE42]
1540 CLOSEIN: RETURN [EC62]
1550 ' [9224]
1560 ' Sortierkriterium bestimmen (Name,
Zusatz oder nach Disketten) [24F2]
1570 ' [9428]
1580 PRINT"(CTRL J){CTRL J} Ausgabe der
Liste: [D87E]
1590 PRINT"(CTRL J){CTRL J}{3 SPACE}1{2
SPACE}Nach dem File-Namen sortiert [7C56]
1600 PRINT"(CTRL J){3 SPACE}2{2 SPACE}Na
ch dem File-Zusatz sortiert [6C78]
1610 PRINT"(CTRL J){3 SPACE}3{2 SPACE}Un
sortiert wie eingegeben [5C82]
1620 PRINT"(7 SPACE)(diskettenseitenweis
e) [8972]
1630 a$=INKEY$: IF a$="" THEN 1630 [CF30]
1640 wahl=INT(VAL(a$))-1: IF wahl<0 OR w
ahl>2 THEN 1630 [3DFC]
1650 RETURN [C198]
1660 ' [E328]
1670 ' Aktualisieren einzelner Disketten
- Hauptteil [5FB4]
1680 ' [952C]
1690 IF zz=0 THEN GOSUB 1390 [C9B8]
1700 MODE 1 [44BC]
1710 PRINT: PRINT [AFBA]
1720 INPUT "Diskettenseite: ";disc$ [D8FE]
1730 IF disc$="" THEN RETURN [6462]
1740 FOR i=1 TO discseit [3E0A]
1750 IF disc$=file$(0,i,0) THEN GOTO 21
00: 'Diskette gefunden [3356]
1760 NEXT [F85A]
1770 PRINT"(CTRL J){CTRL J}Diskette in d
er Liste nicht enthalten [BCF6]
1780 PRINT"(CTRL J){CTRL J}Wahl: [FEF2]
1790 PRINT"(CTRL J){2 SPACE}1{2 SPACE}Di
sketteninhalt aendern [3206]
1800 PRINT"(CTRL J){2 SPACE}2{2 SPACE}Di
skettenliste erweitern [782A]
1810 PRINT"(CTRL J){2 SPACE}3{2 SPACE}An
zeige der Diskettenbezeichnungen [5D48]
1820 PRINT"(CTRL J){2 SPACE}4{2 SPACE}Li
ste neu sortieren [4478]
1830 PRINT"(CTRL J){2 SPACE}5{2 SPACE}Ru
ecksprung ins Hauptmenue [1932]
1840 a$=INKEY$: IF a$="" THEN 1840 [A53C]
1850 a=VAL(a$): IF a<1 OR a>5 THEN 1840 [371A]
1860 ON a GOTO 1690,2190,1910,1880,1870 [967C]
1870 RETURN [C3A0]
1880 IF zz=0 THEN CLS: PEN 3: PRINT"(CTR
L J){CTRL J} Liste nicht geladen":
PEN 1: GOTO 1780 [62F2]
1890 GOSUB 720: RETURN [EECA]
1900 ' [BF22]
1910 ' Diskettenliste anzeigen [D4DE]
1920 ' [C126]
1930 aus=0: menue=2 [870C]
1940 IF zz=0 THEN GOSUB 1390 [8EB4]
1950 CLS: PRINT#aus,"{CTRL J){CTRL J}
1960 IF aus=8 THEN PRINT#8,SPACE$(10); [FA0C]
1970 PRINT#aus,"{3 SPACE}** Diskettenlis
te **": PRINT#aus [14BA]
1980 PRINT#aus: IF aus=8 THEN PRINT#8,SP
ACE$(10); [D882]
1990 PRINT#aus,"{3 SPACE}Nr {2 SPACE}frei
{3 SPACE}Diskettenseite":PRINT#aus [A964]
2000 FOR i=1 TO discseit [B674]
2010 IF aus=8 THEN PRINT#8,SPACE$(10); [14F6]
2020 PRINT#aus,USING"####";i:VAL(file$
(0,i,1));:PRINT#aus,"K{3 SPACE}";fi
le$(0,i,0) [ADE0]
2030 NEXT: IF aus=0 THEN PRINT"(CTRL J){
CTRL J} Weiter - Taste druecken": C
ALL &BB06 [6C98]
2040 IF menue=1 THEN RETURN [0AD8]
2050 CLS: PRINT"(CTRL J){CTRL J)** Aktua
lisierung der Diskettenliste ** [AEBA]
2060 GOTO 1780 [9B22]
2070 ' [9720]
2080 ' Diskettenseite gefunden [D6C4]
2090 ' [C124]
2100 PRINT"(CTRL J){CTRL J} Diskette ein
legen - Taste druecken":CALL &BB06 [91E2]
2110 z=VAL(file$(0,i,2)) [610E]
2120 FOR j=1 TO z [B342]
2130 FOR k=0 TO 1 [29F2]
2140 file$(j,i,k)=" [ADA6]
2150 NEXT:NEXT [C940]
2160 zz=zz-z: z=0: disc=i [BF32]
2170 GOSUB 430: GOSUB 640: GOTO 2050 [9E5E]
2180 ' [E424]
2190 ' Erweiterung der Diskettenliste [5E16]
2200 ' [BE16]
2210 CLS: PRINT"(CTRL J){2 SPACE}Anzahl
der neu zu erfassenden [39F0]
2220 INPUT "{2 SPACE}Diskettenseiten: ";n
eudisc [575A]
2230 neudisc=INT(neudisc): IF neudisc<1
THEN 2210 ELSE GOSUB 1400 [8D32]
2240 INPUT "(CTRL J){2 SPACE}Datum: ";dat
um$ [CBAA]
2250 disc=discseit: discseit=discseit+ne
udisc [A3FC]
2260 WHILE disc<discseit [2D74]
2270 z=0: disc=disc+1 [2156]
2280 PRINT"(CTRL J)Bezeichnung der eing
elegten": INPUT"Diskettenseite: ";
file$(0,disc,0) [1966]
2290 GOSUB 430 [E188]
2300 WEND [4D26]
2310 GOSUB 640: RETURN [BBB4]
2320 ' [C21C]
2330 ' Auswahl Bildschirm oder Drucker [A2BE]
2340 ' [C420]
2350 CLS: PRINT"(CTRL J){CTRL J){CTRL J}
{CTRL J){5 SPACE}Ausgabe auf Bildsc
hirm/Drucker [99A2]
2360 INPUT"(CTRL J){12 SPACE}0/1{3 SPACE
}";aus [E47A]
2370 aus=INT(aus): IF aus<0 OR aus>1 THE
N 2350 [290E]
2380 aus=aus*8: menue=1 [9A00]
2390 GOSUB 1940: RETURN [DF2C]
2400 ' [921A]
2410 ' Ausgabe unsortiert nach Disketten
[07D4]
2420 ' [941E]
2430 zeile=1 [26E0]
2440 FOR i=1 TO discseit [1106]
2450 z=VAL(file$(0,i,2)) [925C]
2460 zhalbe=z-z\2 [5BDC]
2470 zeile=zeile+zhalbe+6 [7DFC]
2480 IF aus=8 AND zeile>65 THEN zeile=z
halbe+6:PRINT#8,CHR$(12);:GOTO 2510 [020A]
2490 PRINT#aus:PRINT#aus [9176]
2500 IF aus=0 THEN PRINT#aus,SPACE$(15)
;"Diskettenseite: ";file$(0,i,0):
PRINT#aus [BF52]
2510 IF aus=8 THEN PRINT#aus,SPACE$(15)
;"Diskettenseite: ";CHR$(27);"E";
file$(0,i,0);CHR$(27);"F
":PRINT#aus [2D4E]
2520 FOR k=1 TO zhalbe [0184]
2530 FOR x=k TO k+zhalbe STEP zhalbe [6F64]
2540 IF x>z THEN 2570 [27CE]
2550 PRINT#aus,SPACE$(10);USING"####";
x: [C41E]
2560 PRINT#aus,"{2 SPACE}";file$(x,i,
0);".";file$(x,i,1);USING"####";
VAL(file$(x,i,2
));:PRINT#aus,"K"; [D2B4]
2570 NEXT [25DA]
2580 PRINT#aus [CB50]
2590 NEXT [5D9E]
2600 PRINT#aus:PRINT#aus,SPACE$(15);fil
e$(0,i,1);" Bloecke frei" [B266]
2610 IF aus=0 THEN CALL &BB06 [3B52]
2620 NEXT [F752]
2630 RETURN [8996]

```

Listing. Mit dieser Routine findet man jedes Programm (Schluß)

Löschtaste für den Schneider



Ein Tastendruck reicht aus, um den Bildschirm zu löschen und den Cursor in die linke obere Ecke zu setzen.

Normalerweise wird Tab beim Schneider-Computer nicht benutzt. Die einzige Funktion, die serienmäßig auf dieser Taste liegt, wurde von Schneider in seinen Handbüchern »vergessen«, mit »Ctrl-Tab« wird der Einfügemodus des Bildschirm-Editors aufgehoben. Ein Text kann ja normalerweise nur überschrieben werden, wenn die zu streichenden Buchstaben zuvor mit »Del« oder »CLR« gelöscht werden. Nach »Ctrl-Tab« werden hingegen alle Buchstaben direkt gelöscht. Unser kleines Programm belegt die Tab-Taste in der Normalebene mit Cursor-Home und, falls gemeinsam mit »Shift« gedrückt mit, Clearscreen.

(Martin Kotulla)

```

10 ***** [F3BB]
20 * (c) Martin Kotulla * [1830]
30 * Grabbestr. 9. 8500 Nuernberg 90* [7896]
40 ***** [E5BE]
50 KEY 140,CHR$(13)+"PRINT CHR$(8)+CHR$(
17):=LOCATE 1,1"+CHR$(13) [A698]
60 KEY 141,CHR$(13)+"CLS"+CHR$(13) [7B72]
70 KEY DEF 68,0,140,141 [5998]
80 NEW [D9E4]
    
```

Listing. Cursor-Home und Clearscreen mit einem Tastendruck

Strings — ein sicheres Versteck



Maschinencode-Programme in einer Zeichenkette abulegen und mit dem Variablen-Descrptor (@) aufzufinden, erlaubt eine sehr komfortable Programmierung.

Viele unter Ihnen kennen Strings als Speicherplatz für Maschinencode-Programme. Ist doch dort ein Überschreiben unmöglich. Im Listing 1 finden Sie als Beispiel eine Routine, die einen Pfeifton ausgibt. In Assembler sieht sie so aus:

Viele unter Ihnen kennen Strings als Speicherplatz für Maschinencode-Programme. Ist doch dort ein Überschreiben unmöglich. Im Listing 1 finden Sie als Beispiel eine Routine, die einen Pfeifton ausgibt. In Assembler sieht sie so aus:

LD A,7

JP &BB5A (&BB5A ist die Tonausgabe-Routine)

Da sich beim häufigen Benutzen des CALL-Aufrufs der Speicherplatzbedarf für die Formel schnell erhöht, versuchen manche Programmierer, dies zu umgehen, indem sie einmal die Adresse des Strings berechnen und im übrigen als Variable angeben. Wie das geht finden Sie im Listing 2.

Diese Methode ist aber sehr gefährlich, da eine meist unvermutet auftretende Garbage Collection die Strings neu ordnet und damit die Startadresse verschiebt. Der CALL-Aufruf geht dann ins Leere und führt zum »Aussteigen« des Computers.

Abhilfe schafft da ein geradezu simpler Trick. Verstehen kann ihn jeder, der den Aufbau des Variablen-Descriptors kennt. Bei »PRINT @a\$« wird nämlich bekannterweise nicht die Adresse des Strings angegeben, sondern die Adresse des zugehörigen Descriptors. Und zwar in folgendem Format:

1. Byte: Stringlänge
2. Byte: Lowbyte der Stringadresse
3. Byte: Highbyte der Stringadresse

Mit »CALL @a\$« würde die String-Descriptor-Adresse aufgerufen und als Maschinencode interpretiert werden.

Mit einem Trick kann man das umgehen. Die Stringlänge wird auf 195 Bytes festgelegt, was gleichzeitig den Z80-Opcode für den JP-Sprungbefehl darstellt. Wird der Descriptor aufgerufen, interpretiert die CPU den Stringdescriptor als »JP String«. Da in dem String der Maschinencode abgelegt ist, kann das Programm nun ordnungsgemäß abgearbeitet werden. Als Programm sieht man das in Listing 3.

Eine Einschränkung muß man aber dabei beachten: Da der Basic-Interpreter den String nach Belieben im Speicher hin- und herschieben kann, darf die Maschinencode-Routine keine absoluten Adressen benutzen. So sind folgende Befehle beispielsweise verboten:

LD HL,(START+2), wobei START für die Programmstart-Adresse steht
 CALL \$+20, \$ symbolisiert den Stand des Programmzählers

Absolute Sprünge und Lesebefehle aus dem Programm sind hingegen erlaubt:

CALL &BB5A
 LD HL,(&C000)
 JP &0000

Der Vorteil der Belegung von Strings mit Maschinencode-Programmen liegt einerseits in der schnelleren Verfügbarkeit der Programme (ein INPUT #9-Befehl genügt schon), andererseits darin, daß keine Gefahr besteht, daß mehrere Maschinenprogramme im gleichen Adreßbereich zusammenstoßen.

(Martin Kotulla)

```

10 a$=CHR$(&3E)+CHR$(&7)+CHR$(&C3)+CHR$(&5A)+CHR$(&BB) [F3CC]
20 CALL PEEK(@a$+1)+PEEK(@a$+2)*256 [BDDE]
    
```

Listing 1. Ein Pieps aus dem String

```

10 a$=CHR$(&3E)+CHR$(&7)+CHR$(&C3)+CHR$(&5A)+CHR$(&BB) [F3CC]
20 adr=PEEK(@a$+1)+PEEK(@a$+2)*256 [D42E]
30 CALL adr [C0EC]
    
```

Listing 2. Der Platzbedarf wird geringer

```

10 ***** [F3BB]
20 * (c) Martin Kotulla * [14D4]
30 * Grabbestr. 9. 8500 Nuernberg 90* [7896]
40 ***** [E5BE]
50 a$=STRING$(&C3,0) [4916]
60 FOR i=1 TO 5:READ a: MID$(a$,i)=CHR$(a) [A0D0]
):NEXT i [BBFC]
70 DATA &3E,&07,&C3,&5A,&BB [BBFC]
80 FOR i=1 TO 20:CALL @a$:FOR k=1 TO 200 [461C]
):NEXT k,i
    
```

Listing 3. »195« heißt der geheimnisvolle Code

Tips & Tricks

```

190 ' *****
      *****
200 ' ***** Initialisierung des Pro [B2EA]
gramms ***** [09A2]
210 ' ***** [B21C]
      *****
220 ' *****
      ***** [1BB2]
230 ' ***** [07B8]
240 MEMORY 24999
      :REM RAMTOP setzen [EA54]
250 a=15:b=70:c=7:d=18
      :REM Windowgroesse
festlegen [2D76]
260 mo=8:mo1=2
      :REM Hilfsvariablen
zum Window einrahmen [35E4]
270 MODE 2:INK 0,0:PEN 0:INK 1,13:PAPER
13:BORDER 13:CLS :REM Bildschirm ini
tialisieren [B010]
280 WINDOW#1,a,b,c,d:PAPER #1,13
      :REM Window deklari
eren [12A4]
290 ' [ADC4]
300 ' *****
      ***** [6BB0]
310 ' *****
      ***** [5A1E]
320 ' ***** Bildschieberoutine
      ***** [2148]
330 ' *****
      ***** [A822]
340 ' *****
      ***** [CFB8]
350 ' ***** [05BE]
360 FOR i=25000 TO 25023
      :REM In die Adresse
n 25000 bis 25023 die Werte aus Zeil
e 300 und 310 [3FA4]
370 READ z
      :REM auslesen und e
inschreiben [BCC8]
      POKE i,z [81B2]
380 NEXT i [A388]
390 ' [08B6]
400 '
410 DATA 1,255,63,17,12,98,33,0,192,237,
176,201 [AF88]
420 DATA 1,255,63,17,0,192,33,12,98,237,
176,201 [028A]
430 ' [07BC]
440 ' *****
      ***** [89BA]
450 ' *****
      ***** [9228]
460 ' ***** Bildschirm fuellen (Hie
r steht ***** [BDF2]
470 ' ***** normal das Anwenderprog
ramm) ***** [78B2]
480 ' *****
      ***** [392E]
490 ' *****
      ***** [CAC4]
500 ' ***** [06B8]
510 LOCATE 1,3:PRINT"Es gibt Programme,
bei denen man nicht umhin kommt, dem
Benutzer die" [A3EA]
520 LOCATE 1,4:PRINT"Moeglichkeit des Ei
nblendens von Beschreibungen oder Hi
lfstexten in" [6112]
530 LOCATE 1,5:PRINT"die laufende Bearbe
itung eines Programms zu ermoeglich
e" [1770]
540 LOCATE 1,8:PRINT"Der momentane Bilda
ufbau soll aber weder geloescht noch
so ueberschrieben" [BE30]
550 LOCATE 1,9:PRINT"werden, dass, nachd
em der Hilfstext gelesen wurde nicht
mehr mit dem gleichen" [43D8]
560 LOCATE 1,10:PRINT"Bild und an der gl
eichen Stelle weitergearbeitet werde
n kann." [2DAE]
570 LOCATE 1,13:PRINT"Dies bedeutet, wir
muessen den momentanen Bildaufbau
RETTEN". [0332]
580 LOCATE 1,16:PRINT"Die Ihnen vorliege
nde Routine, die in BASIC geschriebe
n ist und eine" [74E0]
590 LOCATE 1,17:PRINT"Einrahmung sowie d
ie Beschriftung der Windows zuloesst
, uebernimmt diese Arbeit." [3CCA]
600 LOCATE 1,20:PRINT"Die Routine kann p
roblemlos in jedes Programm eingebun
den werden." [882A]
610 LOCATE 1,24:PRINT"E = ENDE -- ? = HE
LP" [2F74]
620 a$=INKEY$:IF a$="" THEN 620
      :REM Warten auf Ein
gabe [4F54]
630 a$=UPPER$(a$)
      :REM Umwandlung in
Grossbuchstaben [B44A]

640 IF a$="E" THEN CLS:END:ELSE IF a$<>"
?" THEN 620 ELSE CALL 25000 [B1DA]
650 GOSUB 870:CALL 25012:GOTO 620
      :REM Wenn 'E' gedru
eckt wird dann Programmende [1B28]
660 '
      :REM Wenn '?' gedru
eckt wird dann wird der momentane
au gerettet (CALL 25000) [D470]
680 '
      :REM Danach Sprung
zum Textblock in Zeile 620 [E01C]
690 '
      :REM Nach der Rueck
kehr aus dem Textblock wird die
wieder zurueckgeschoben. [62DA]
700 '
      :REM alte Bildmaske
wieder zurueckgeschoben. [0D18]
710 '
      :REM Danach Sprung
nach Zeile 470 [6404]
720 ' [08C0]
730 ' *****
      ***** [28BE]
740 ' *****
      ***** [932C]
750 ' ***** Routine zum Einrahmen u
nd ***** [F4EE]
760 ' ***** Beschriften der Windows
      ***** [F216]
770 ' *****
      ***** [AC32]
780 ' *****
      ***** [CBC8]
790 ' ***** [71CE]
800 aa=a-1:bb=b:cc=(c-1)*mo1:dd=d*mo1 [BCE6]
810 aa=aa*mo:bb=bb*mo:cc=cc*mo:dd=dd*mo [0C36]
820 PLOT aa-2,399-cc+2,0:DRAW bb,399-cc+
2,0:DRAW bb,399-dd,0:DRAW aa-2,399-d
d,0:DRAW aa-2,399-cc+2,0 [B2BE]
830 PLOT aa-3,399-cc+3,0:DRAW bb+1,399-c
c+3,0:DRAW bb+1,399-dd-1,0:DRAW aa-3
,399-dd-1,0:DRAW aa-3,399-cc+3,0 [CEB4]
840 PLOT aa-4,399-cc+4,0:DRAW bb+2,399-c
c+4,0:DRAW bb+2,399-dd-2,0:DRAW aa-4
,399-dd-2,0:DRAW aa-4,399-cc+4,0 [5FCA]
850 LOCATE (aa+30)/mo,(dd+mo)/16
      :REM Beschriftungsp
unkt des Windows festlegen [469C]
860 RETURN
      :REM Ruecksprung zu
Zeile 570 [0372]
870 ' [B3CC]
880 ' *****
      ***** [8DCA]
890 ' *****
      ***** [9638]
900 ' ***** Textblock
      ***** [B048]
910 ' *****
      ***** [962A]
920 ' *****
      ***** [09C0]
930 ' ***** [0FC6]
940 GOSUB 690
      :REM Sprung zur Ein
rahmroutine [9948]
950 WINDOW SWAP 1,0
      :REM Umschalten von
Fenster 0 auf Fenster 1 [B41C]
960 CLS
      :REM Bildschirm loe
schen [5822]
970 LOCATE 17,2:PRINT CHR$(24);" H E L P
- W I N D O W ";CHR$(24) [C23A]
980 LOCATE 5,4:PRINT"Hier koennen Sie si
ch Bemerkungen oder" [B868]
990 LOCATE 5,5:PRINT"Texte u.s.w. ablege
n, die dann bei Bedarf" [275E]
1000 LOCATE 5,6:PRINT"einfach in die lau
fende Bildschirmmaske" [EA10]
1010 LOCATE 5,7:PRINT"eingebendet werde
n koennen." [5848]
1020 LOCATE 5,8:PRINT"Man kann auch mehr
ere Windows miteinander" [AE8E]
1030 LOCATE 5,9:PRINT"verschachteln." [BEF2]
1040 LOCATE 14,11:PRINT CHR$(24);" Weite
r mit irgendeiner Taste ";CHR$(24) [FC50]
1050 a$=INKEY$:IF a$="" THEN 1050
      :REM Warten auf
Eingabe [570C]
1060 WINDOW SWAP 0,1
      :REM Zurueckschal
ten von Fenster 1 auf Fenster 0 [79BC]
1070 RETURN
      :REM Ruecksprung
zu Zeile 410 [47B8]

```

Listing »Bildschieberoutine« (Schluß)

Kreise mit einem Basic-Befehl



Die Grafik der Schneider-Computer erlaubt in Verbindung mit den RSX-Befehls-erweiterungen den Aufbau spezieller Grafikbefehle, wie bei kaum einem anderen Heimcomputer. Ein Kreis gehört natürlich dazu.

Unsere Befehls-erweiterung soll die Form »CIRCLE, x-Koord., y-Koord., Radius« haben. Damit die Routine hinreichend schnell wird, dürfen wir die Darstellung des Kreises in Polarkoordinaten $((x,y) = r(\cos(\varphi), \sin(\varphi)))$ nicht benutzen. Die Polynomapproximationen für cos und sin des Interpreters sind zu zeitaufwendig. Aber auch mit der Formel $y = \sqrt{r^2 - x^2}$ geht es nicht schneller. Der CPC hat leider nur eine sehr langsame Wurzelfunktion über den Umweg:

$$\sqrt{x} = x^{1/2} = \exp\left(\frac{1}{2} \ln(x)\right).$$

Außerdem müßte noch — da die Koordinaten eines Bildpunktes ganzzahlig sind — eine zeitraubende Umwandlung von Fließkomma- in Integerformat vorgenommen werden. Eine eigene Routine muß also her. Wir nehmen dazu die Newton-Approximation für \sqrt{x} :

$$x_{n+1} = \frac{1}{2} \left(x_n + \frac{x}{x_n} \right),$$

mit $n = 0, 1, 2, \dots$ und $x_0 = x$.

Gerechnet wird nur mit ganzzahligen Werten und maximal zehn Iterationsschritten. Die entsprechende Routine ist etwa um den Faktor zehn schneller als die Routine für Gleitkommazahlen des Basic-Interpreters.

Noch drei Faktoren sind zu berücksichtigen, wenn man eine schnelle Routine erhalten will. Die Größe eines Grafikpunktes hängt vom jeweils gewählten Modus ab. So ist es natürlich falsch, im Mode 1 (1 Pixel = 2 x-Koordinaten) den Kreis wie im Mode 2 (1 Pixel = 1 x-Koordinate) zu zeichnen. Aufgrund der Symmetrie des Kreises brauchen wir des weiteren nur die Koordinate (x,y) eines Punktes zu berechnen und erhalten durch Vorzeichenwechsel die entsprechenden Symmetriepunkte $(-x,y)$, $(x,-y)$ und $(-x,-y)$. Letztlich muß man noch entscheiden, ob der Kreis durch PLOT oder DRAW entstehen soll. Mit PLOT gibt es bei großen Kreisen Schwierigkeiten, da hier die y-Abschnitte bei zwei benachbarten x-Koordinaten mehr als ein Pixel auseinanderliegen. Es wären also zusätzliche Punkte zu plotten, um einen geschlossenen Kreis zu bekommen — natürlich auf Kosten der Geschwindigkeit. Bleibt also nur noch DRAW.

Der erste Programmteil bindet die Befehls-erweiterung »Circle« ein. Zu Beginn der Routine prüft das Programm die Anzahl der übergebenen Parameter. Anhand des Bildschirm-Modus wird ein Zähler für die Berechnung der x-Abschnitte bestimmt und in COUNT abgelegt. Nach Sichern der aktuellen Koordina-

ten setzt das Programm den Ursprung auf den übergebenen Kreismittelpunkt. Der Kreis wird beginnend bei der Koordinate (0,R) gezeichnet.

Der Rest des Programms ist lediglich die leider immer mühsame Umsetzung der Formel $y = \sqrt{r^2 - x^2}$. Eine große Hilfe sind hierbei natürlich die vom Betriebssystem zur Verfügung gestellten Routinen für die Integer-Arithmetik.

(Horst Udo Hanenberg)

```

1000 ' BASIC - Lader f)r CIRCLE [674C]
1010 '(c) 1985 Dipl.-Math. Horst Udo [06E4]
      Hanenberg [C1B0]
1020 ' Am M}hlenteich 20
1030 ' 5010 Bergheim 12 - Gle [EEAA]
      ssen
1040 ' Tel. 0 22 38 / 4 16 85 [04EA]
      [B61A]
1050 ' [C422]
1060 MEMORY &9FFF [4156]
1070 DZEILE=1260 [CAB8]
1080 ANF=&A000+2^16;ENDE=&A130+2^16 [535C]
1090 FOR I=ANF TO ENDE STEP 16 [BF88]
1100 SUM=0 [334C]
1110 DZEILE=DZEILE+10 [87F0]
1120 FOR J=I TO I+15 [2CE4]
1130 IF J>ENDE THEN 1190 [4D3A]
1140 READ W$ [0154]
1150 WERT=VAL("&"+W$) [98DE]
1160 POKE J,WERT [77FA]
1170 SUM=SUM+WERT [EC26]
1180 NEXT J [0E44]
1190 READ W$ [044C]
1200 WERT=VAL("&"+W$)
1210 IF WERT <> SUM THEN PRINT"FEHLER IN [9A34]
      DATA-ZEILE";DZEILE:END [671A]
1220 NEXT I [EA32]
1230 CALL &A000 [18A2]
1240 NEW
1250 ' POKE - Werte [D1FC]
      Pr}fsumme [9320]

1260 '
1270 DATA 01,09,A0,21,15,A0,C3,D1,BC,0E, [2E50]
      A0,C3,19,A0,43,49,0686
1280 DATA 52,43,4C,C5,00,00,00,00,FE, [4A06]
      03,C0,CD,11,BC,06,0507
1290 DATA 04,38,06,06,02,28,02,06,01,78, [630E]
      32,20,A1,CD,CC,BB,043A
1300 DATA E5,D5,DD,66,01,DD,6E,00,22,21, [FB58]
      A1,22,25,A1,22,27,065E
1310 DATA A1,DD,56,05,DD,5E,04,DD,66,03, [CF84]
      DD,6E,02,CD,C9,BB,07FC
1320 DATA 2A,21,A1,ED,5B,21,A1,CD,BE,BD, [047C]
      22,23,A1,21,00,00,0645
1330 DATA 22,2B,A1,2A,25,A1,3A,20,A1,47, [CE66]
      2B,10,FD,7C,3C,28,0538
1340 DATA 1A,22,25,A1,E5,D1,CD,BE,BD,EB, [4C7E]
      2A,23,A1,B7,ED,52,08CF
1350 DATA CD,90,A0,22,29,A1,CD,B6,A0,18, [013E]
      DB,D1,E1,C3,C9,BB,09F5
1360 DATA 7C,B5,C8,22,2D,A1,22,2F,A1,06, [D6E2]
      0A,C5,2A,2D,A1,ED,0695
1370 DATA 5B,2F,A1,D5,CD,C1,BD,D1,19,11, [6C0C]
      02,00,CD,C1,BD,22,07B5
1380 DATA 2F,A1,C1,10,E6,C9,ED,5B,27,A1, [7574]
      2A,2B,A1,E5,D5,CD,08DD
1390 DATA C0,BB,ED,5B,25,A1,2A,29,A1,E5, [79E8]
      D5,CD,F6,BB,ED,5B,09FD
1400 DATA 27,A1,2A,2B,A1,CD,C7,BD,CD,C0, [9FE2]
      BB,D1,E1,CD,C7,BD,0A5A
1410 DATA CD,F6,BB,E1,CD,C7,BD,EB,E1,CD, [7D1A]
      C0,BB,2A,25,A1,CD,0B81
1420 DATA C7,BD,E5,EB,2A,29,A1,E5,CD,F6, [6AAE]
      BB,2A,27,A1,CD,C7,0A31
1430 DATA BD,EB,2A,2B,A1,CD,C7,BD,CD,C0, [474A]
      BB,E1,CD,C7,BD,D1,0B3A
1440 DATA CD,F6,BB,2A,25,A1,22,27,A1,2A, [09AA]
      29,A1,22,2B,A1,C9,0703
1450 DATA 00,00,00,00,00,00,00,00,00, [0F48]
      00,00,00,00,00,00,0000 [0DE2]
1460 DATA 00,0000

```

Listing 1. »CIRCLE« heißt der neue Befehl

```

10 FOR j=0 TO 2 [F0E8]
20 MODE j [5062]
30 PRINT "Mode ";j [96F2]
40 LOCATE 1,1 [EDD4]
50 PRINT "Mode ";j [7EF6]
60 FOR i=10 TO 200 STEP 10 [D4CC]
70 !CIRCLE,320,200,i [EE92]
80 NEXT [218E]
90 FOR k=1 TO 1000 [2D1A]
100 NEXT [71E0]
110 NEXT [55E2]
120 GOTO 10 [C9DA]

```

Listing 2. »CIRCLE« funktioniert in allen drei Modi

Schnelles Laden von Kassette



Auch mit dem Kassetten-Recorder soll das Laden eines Programms nicht zum Frust werden.

Man muß nur das Aufzeichnungsformat ändern und die Daten in einem Block speichern.

Ein längeres Programm von Kassette zu laden ist zeitaufwendig. Um beispielsweise 20000 Byte mit 2000 Baud zu laden, braucht der Schneider-Computer ungefähr 140 Sekunden. Es gibt viele Tricks, die Baudrate zu ändern (bis zirka 4000 Baud ist dies mit ausreichender Datensicherheit zu realisieren), aber das eigentliche Problem stellt das Aufzeichnungsformat dar. Die Programme werden nämlich in 2 KByte lange Blöcke zerlegt. Jeder Block hat einen eigenen Kopf (Header), in dem Ladeadresse, Länge, Name des Programms und so weiter vermerkt sind. Greift man nun direkt auf die Firmware-Routine CAS WRITE zurück, so kann man die Daten ohne Header in einem Block speichern. Die Zeit halbiert sich dann in unserem Beispiel auf 72 Sekunden (bei 2000 Baud).

Zum Laden der Programme benutzt der Schneider eine Firmware-Routine mit dem Namen CAS READ. Auf diese kann man auch direkt zugreifen, so daß die in einem Block gespeicherten Daten wieder korrekt gelesen werden können.

Die beiden Routinen müssen nun an unsere Wünsche angepaßt werden. Listing 1 ist der Basic-Lader für die Routine CAS READ. Sie muß immer vor dem zu ladenden Programm stehen — sinnvollerweise mit dessen Namen. Die Anweisung hierzu lautet: »SPEED WRITE 1:SAVE "Name"«

In Listing 2 finden Sie den Basic-Lader, der die Speicherroutine erzeugt. Das Maschinencode-Programm berechnet die tatsächliche Länge des im Speicher stehenden Basic-Programms, lädt die Register des Z80 mit den erforderlichen Werten und ruft die eigentliche Routine CAS WRITE auf. Nachdem Listing 2 einmal gelaufen ist, kann man es löschen und das Programm laden, das man speichern will. Mit »CALL HIMEM+1« wird die Speicherroutine gestartet und das Programm auf Kassette geschrieben.

Beim Lesen von der Kassette wird zuerst die Laderoutine mit »RUN« geladen und dann automatisch das eigentliche Programm in den Speicher geholt. Dieses muß übrigens von Hand gestartet werden, da dies automatisch nicht möglich ist.

Mit der hier vorgestellten Betriebssystem-Erweiterung kann man übrigens auch jedes Maschinencode-Programm speichern und laden. Man muß nur die Startadresse und die Länge der Routine kennen (mit einem Headreader-Programm leicht zu bestimmen). Zum Laden muß ein freier Speicherplatz vorhanden sein, der mindestens 12 Byte aufnehmen kann. Das Listing 1 muß um die zwei Zeilen aus Listing 3 erweitert werden. Die Variable »begin« steht für den Anfangsplatz des freien Bereichs minus eins. Die Variablen l, h, e und d sind Platzhalter für folgende Werte:

- l Lowbyte der Startadresse des Maschinencode-Programms
- h Highbyte der Startadresse des Maschinencode-Programms
- e Lowbyte der Länge des Maschinencode-Programms
- d Highbyte der Länge des Maschinencode-Programms

Wenn in einem Programm solch eine Maschinencode-Routine nachgeladen werden soll, muß der normale Ladebefehl durch »CALL begin+1« ersetzt werden — und die Daten im entsprechenden Format vorliegen.

Wenn man das ursprüngliche Maschinencode-Programm im Speicher hat, benutzt man die zwei Zeilen aus Listing 4, um das Programm in der neuen Form zu speichern. Die Variablen sind die gleichen wie im Listing 3.

(Thomas Dröge)

```

10 ***** [9FCC]
20 * (c) Thomas Droege * [CB66]
30 * Hohenzollernstr. 18 * [068C]
40 * 4700 Hamm 1 * [33FC]
50 ***** [87D4]
60 MEMORY &A9FF:FOR t=HIMEM+1 TO HIMEM+3
0:READ a:POKE t,a:NEXT [6248]
70 CALL HIMEM+1 [971E]
80 DATA &21,&34,&AE,&11,&5B,&2,&3E,&10,&
CD,&A1,&BC,&2A,&B3,&AE,&ED,&4B,&B1,&A
E,&ED,&42,&EB,&2A,&B1,&AE,&3E,&10,&CD
&A1,&BC,&C9 [5FAE]
    
```

Listing 1. Schnelles Laden aus einem Block

```

10 ***** [9FCC]
20 * (c) Thomas Droege * [CB66]
30 * Hohenzollernstr. 18 * [068C]
40 * 4700 Hamm 1 * [33FC]
50 ***** [87D4]
60 MEMORY &A9FF:FOR t=&AA00 TO &AA28:REA
D a:POKE t,a:NEXT [01EB]
70 DATA &21,&5D,&0,&3E,&A,&CD,&68,&BC,&2
1,&34,&AE,&11,&5B,&2,&3E,&10,&CD,&9E,
&BC,&2A,&B3,&AE,&2A,&B3,&AE,&ED,&4B,&
B1,&AE,&ED,&42,&EB,&2A,&B1,&AE,&3E,&1
0,&CD,&9E,&BC,&C9 [5B2E]
    
```

Listing 2. Viel Zeit spart man, wenn die Daten in einem Block gespeichert werden

```

10 FOR t=1 TO 12:READ a:POKE begin+t,a:N
EXT t [C79C]
20 DATA &21,1,h,&11,e,d,&3e,&10,&cd,&a1,
&bc,&c9 [29CA]
    
```

Listing 3. Mit dieser Ergänzung können auch Maschinencode-Routinen in der neuen Form geladen werden

```

10 FOR t=1 TO 12:READ a:POKE begin+t,a:N
EXT t:CALL begin+1 [3E4A]
20 DATA &21,1,h,&11,e,d,&3e,&10,&cd,&9e,
&bc,&c9 [49E2]
    
```

Listing 4. Mit zwei Zeilen bekommen auch Maschinencode-Programme das richtige Format

Schere statt Bleistift und Radiergummi



Folgende Situation: Man sucht verzweifelt nach einem Programm und findet es einfach nicht. Warum?

Weil die Beschriftung auf den zu den Disketten gehörigen Karten nicht mehr aktuell ist. Mit »Autolabel« bekommen Sie fast automatisch ein Inhaltsverzeichnis.

Wieder einmal ist man guter Vorsätze. In Zukunft wird jede Änderung auf der Diskette mit Radiergummi und Bleistift auf dem entsprechenden Zettel vermerkt. Doch schon die nächste große Disketten-Aufräumaktion läßt alle Wünsche in Vergessenheit geraten. Das Chaos beginnt von neuem.

Tippen Sie unser Programm ab, und schon gehören diese Probleme der Vergangenheit an. Das Inhaltsverzeichnis wird mit Randmarkierung säuberlich ausgedruckt, einfach ausschneiden, und schon paßt der Ausdruck in die Diskettenhülle.

Nach Programmstart wird zuerst das Anschalten des Druckers gefordert, bevor Sie die zu beschriftende Diskette einlegen müssen. Der Titel (wird zuerst angefordert) darf insgesamt 37 Buchstaben beinhalten. Die Namen der verschiedenen Programme werden nun auf den Drucker ausgegeben. Die Handhabung von Drucker und Diskettenstation wird von dem Programm gesteuert.

Das Programm wurde ursprünglich für den Star-Drucker geschrieben. In den Zeilen 460 bis 560 kann man aber auch jeden anderen Drucker anpassen. Allerdings muß das Gerät über Zeilenvorschub in 1/2-Zoll-Schritten verfügen. Die passenden SteuerCodes finden Sie in Ihrem Handbuch.

(Thomas Arnold)

```

10 ***** [9FCC]
20 *(c) Thomas Arnold/Andreas Lohrum * [43C2]
30 * Am Graben 25, 7700 Singen * [0780]
40 ***** [E9D2]
50 MEMORY 42619 [EE08]
60 OPENOUT "dummy":MEMORY HIMEM-1:CLOSED [42D2]
   UT [FBFC]
70 MODE 2 [A3DE]
80 CLEAR [83D0]
90 DEFINIT i,p
100 LOCATE 31,3:PRINT CHR$(164)+" by Tho [BD2C]
   mas Arnold"
110 LOCATE 10,6:PRINT "Drucker bitte ON- [888A]
   LINE schalten !" [9AD2]
120 x=40571 [78FC]
130 DIM disc$(64):n=63
140 LOCATE 10,8:PRINT "Bitte Diskette ei [29F2]
   nlegen, dann Taste druecken !" [3160]
150 a$=INKEY$: IF a$="" THEN 150
160 PRINT:PRINT:INPUT "Titel der Diskett [FABC]
   e (maximal 37 Zeichen) ";:titel$ [18B4]
170 l=LEN(titel$):IF l>37 THEN 160 [9C60]
180 MODE 2 [9924]
190 CAT [532B]
200 free=PEEK(49088)
210 IF tf=0 THEN freea=free [FCF4]

```

```

220 IF tf=1 THEN freeb=free [CBFA]
230 PRINT:PRINT "{33SPACE }Bitte warten [33EE]
   !" [BFC8]
240 FOR i=0 TO n
250 IF PEEK(x+1+14*i)=0 THEN GOSUB 7 [2B2E]
80:GOTO 420 [DE16]
260 m1=14*i [5A50]
270 merke=x+2+m1 [75C4]
280 FOR i1=0 TO 7
290 disc$(i)=disc$(i)+CHR$(PEEK( [6E8C]
   merke+i1)) [8C58]
300 NEXT i1 [1220]
310 disc$(i)=disc$(i)+"." [ACA6]
320 merke=x+10+m1 [5FB4]
330 FOR i2=0 TO 2
340 disc$(i)=disc$(i)+CHR$(PEEK( [3CAC]
   x+10+14*i+i2)) [2364]
350 NEXT i2 [FD4E]
360 disc$(i)=disc$(i)+"{2SPACE }"
370 IF PEEK(x+13+m1)<10 THEN disc$(i) [3C12]
   )=disc$(i)+" "
380 disc$(i)=disc$(i)+STR$(PEEK(x+13 [8AB2]
   +m1))
390 IF tf=0 THEN disc$(i)=disc$(i)+" [78E8]
   k{6SPACE }:"
400 IF tf=1 THEN disc$(i)=disc$(i)+" [CDA6]
   k" [EEFA]
410 NEXT i [5C6E]
420 IF tf=1 THEN 470
430 PRINT:PRINT"Diskette umdrehen, dann [C588]
   Taste druecken !" [2028]
440 a$=INKEY$:IF a$="" THEN 440 [4202]
450 tf=1 [0758]
460 GOTO 180 [38B6]
470 p=200/druck:IF p<4 THEN p=4
480 PRINT #8,CHR$(15) [291A]
   :REM Schmalschrift ein [D288]
490 PRINT #8,STRING$(55,"-")
500 PRINT #8,CHR$(27)+"G"; [9118]
   :REM Doppeldruck ein [4AD6]
510 PRINT #8,SPACE$(27-INT(1/2));:titel$
520 PRINT #8,CHR$(27)+"H"; [5C38]
   :REM Doppeldruck aus [605C]
530 IF druck<18 THEN 550
540 PRINT #8,CHR$(27)+"A"+CHR$(p); [2742]
   :REM Setze LF auf p/72 inch [AFB2]
550 PRINT #8,STRING$(55,"-")
560 PRINT #8,"{7SPACE }Seite A{13SPACE [3080]
   }:{6SPACE }Seite B{12SPACE }." [8F86]
570 PRINT #8,STRING$(55,"-") [823C]
580 PRINT #8,"{3SPACE }FREI : ";
590 PRINT #8,CHR$(27)+"G"; [7A2A]
   :REM Doppeldruck ein [F65C]
600 PRINT #8,USING "####";freea;:PRINT #8 [AD38]
   , " k"; [1CD6]
610 PRINT #8,CHR$(27)+"H"; [0A20]
   :REM Doppeldruck aus [980C]
620 PRINT #8,"{15SPACE }FREI : "; [A0B6]
630 PRINT #8,CHR$(27)+"G"; [122A]
   :REM Doppeldruck ein [C958]
640 PRINT #8,USING "####";freeb;:PRINT #8 [330C]
   , " k{11SPACE }." [548C]
650 PRINT #8
660 FOR i=0 TO druck
670 PRINT #8,"{2SPACE }";:disc$(i);" [B954]
   {5SPACE }." [695C]
680 NEXT i [1636]
690 PRINT #8,STRING$(55,"-") [A120]
700 PRINT:PRINT "Noch ein Label ? (j/n)" [E416]
   [AFF4]
710 n$=INKEY$:IF n$="" THEN 710 [9F28]
720 n$=UPPER$(n$) [4548]
730 IF n$="N" THEN END [21BC]
740 IF n$<>"J" THEN 710 [4210]
750 CLEAR:MODE 2:GOTO 120 [757E]
760 END [898A]
770 REM Automatischer Laengenabgleich [9D34]
780 IF tf=0 THEN aleng=i-1:RETURN: ELSE [1F86]
   IF tf=1 THEN bleng=i-1 [C86E]
790 druck=MAX(aleng,bleng)
800 IF bleng>aleng THEN 840
810 IF aleng>bleng THEN 890
820 RETURN
830 REM Mehr Files auf B-Seite
840 FOR i=aleng+1 TO bleng [86E]
850 disc$(i)="{8SPACE }.{7SPACE }0 k [A59C]
   {6SPACE }:"+"{disc$(i) [550C]
860 NEXT i [A33E]
870 RETURN [8BBE]
880 REM Mehr Files auf A-Seite
890 FOR i=bleng+1 TO aleng [2B78]
900 disc$(i)=disc$(i)+"{8SPACE }.{7S [CC60]
   PACE }0 k" [7DF2]
910 NEXT [DE36]
920 RETURN [C426]
930 END

```

Listing. Mit »Autolabel« bekommen Sie ein Inhaltsverzeichnis Ihrer Disketten

Programmieren — schnell und einfach



Mit »Tastatur« können Sie mit nur einem Tastendruck die wichtigsten Basic-Befehle aufrufen.

Basic-Befehlsbegriffe können sehr lang sein. Deshalb gibt es im Locomotive-Basic zwei Abkürzungen für besonders häufig gebrauchte Routinen — »?« für »PRINT« und »<« für »REM«. Alle

anderen Anweisungen müssen aber Buchstabe für Buchstabe eingetippt werden. Außer man benutzt »Tastatur«, eine Programmierhilfe.

Drückt man einen Buchstaben gemeinsam mit der Shift-Taste, dann wird nicht der Großbuchstabe, sondern eine Basic-Anweisung aktiviert. Die Belegung der Tastatur finden Sie in der Tabelle. Die normalerweise in dieser Ebene liegenden Großbuchstaben können nur noch mit der Ctrl-Taste oder mittels »Caps Lock« aufgerufen werden. In das Programm sind auch noch die deutschen Sonderzeichen integriert. Da das »ö« die gleiche Belegung hat wie der senkrechte Strich (»|«) der RSX-Befehle, darf dieser Buchstabe nicht in REM- oder DATA-Zeilen benutzt werden.

Das Programm besteht aus zwei Teilen, wobei der erste alle Erklärungen enthält. Der zweite ist für die eigentliche Umdefinition zuständig und kann auch alleine benutzt werden. Wer andere als die hier vorgestellten Befehle mit einem Tastendruck aufrufen will, der kann diese im zweiten Teil einbauen.

(Karlheinz Battermann)

```
*****
***          TASTATUR          ***
***          eine              ***
***          Programmierhilfe  ***
*****
```

* Taste	allein mit SHIFT	mit CTRL	** Taste	allein mit SHIFT	mit CTRL	** Taste	allein mit SHIFT	mit CTRL	*		
	oder	**		oder	**		oder	**			
		CAPS LOCK			CAPS LOCK			CAPS LOCK			
* die definierten Basic-Befehle sind											
* A	a	AND	A	** H	h	GOSUB	H	** R	r	REM ***	R
* B	b	BORDER	B	** I	i	INPUT	I	** S	s	SYMBOL	S
* C	c	CLS	C	** L	l	LOCATE	L	** T	t	THEN	T
* D	d	DATA	D	** M	m	MODE	M	** U	u	USING	U
* E	e	ELSE	E	** N	n	NEXT	N	** V	v	VAL	V
* F	f	FOR	F	** O	o	OR	O	** W	w	WINDOW	W
* G	g	GOTO	G	** P	p	PRINT	P	** Z	z	ZONE	Z
* J	j	J	q	** K	k	K	p	**			
* Q	q	Q	** X	x	X	X	K	** Y	y	Y	
* t	t	LIST	*	** ;	;	LIST#	+	**			
* ,	,	INK	<	** .	.	PEN	>	** /	/	PAPER	?
* Anmerkung: Die Zeichen q p und K sind Steuerzeichen in PRINT-Zeilen, z.B. PRINT "q zur nächsten Zeile" usw.											
* die deutschen Zeichen sind											
* X	ä	Ä	.	** Ü	ö	Ö	.	** ö	ü	Ü	
.	.	.	.	** S	ß	.	.	**	.	.	.
* Anmerkung: Das kleine > ö < darf nicht in REM- und DATA-Zeilen benutzt werden !!!											

Tabelle. Die Belegung der Tasten mit »Tastatur«

```

10 REM ***** [B2E2]
20 REM *** TASTATUR *** [3CF4]
30 REM *** eine *** [8448]
40 REM *** Programmierhilfe *** [B922]
50 REM *** [910A]
60 REM ***(c) Karlheinz Battermann*** [875C]
70 REM *** Weissdornweg 20 *** [3C54]
80 REM *** 3422 Bad Lauterberg *** [748E]
90 REM ***** [12F2]
100 INK 0,0:INK 1,11:INK 2,1:INK 3,9 [71B2]
110 MODE 1:PEN 3:PAPER 0:BORDER 3:CLS [D102]
120 LOCATE 6,4:PRINT STRING$(30,"*") [E3A2]
130 ZONE 5:PRINT,"***{7 SPACE}TASTATUR{9 SPACE}***" [AAE0]
140 PRINT,"***{9 SPACE}eine{11 SPACE}***" [099E]
150 PRINT,"***{4 SPACE}Programmierhilfe{4 SPACE}***" [D278]
160 PRINT,STRING$(30,"*") [8C72]
170 LOCATE 7,25:PRINT CHR$(164);"1985 by kabasoft-computing" [ACDA]
180 WINDOW 6,35,15,19:PEN 1:PAPER 2:CLS [DE62]
190 PRINT:PRINT" Wuenschen Sie Instruktionen?" [677E]
200 ZONE 12:PRINT:PRINT,"> j/n <" [E660]
210 a$=INKEY$:IF a$="" THEN 210 [D014]
220 IF a$="n" THEN CLS:GOTO 350 [1AA6]
230 IF a$="j" THEN 240 ELSE 210 [61AA]
240 WINDOW 6,35,11,22:PEN 1:PAPER 2:CLS [A748]
250 PRINT"Dies Programm soll die Arbeit beim Abtippen von Listings undbeim eigenen Programmieren erleichtern." [F518]
260 PRINT"Deshalb wurden die wichtigsten Basic-Befehle auf die Tastaturgelegt, sodass nur eine Taste zusammen mit SHIFT gedreuckt{2 SPACE}werden muss um z.B. den Basic-Befehl > WINDOW < zu geben." [BB9A]
270 ZONE 6:PRINT:PRINT,"weiter{2 SPACE}->{2 SPACE}Taste" [F4E4]
280 CALL &BB18:CLS [5C4A]
290 PRINT"Alle Buchstabentasten, ausser Q Y J K und X, sind mit Basic-Befehlen belegt." [4C44]
300 PRINT:PRINT"Die Grossbuchstaben der mit{3 SPACE}Basic-Befehlen belegten Tastensind daher nur noch zusammen{2 SPACE}mit CAPS LOCK oder CTRL er-{3 SPACE}reichbar!" [4252]
310 ZONE 6:PRINT:PRINT:PRINT,"weiter{2 SPACE}->{2 SPACE}Taste" [DA68]

```

Listing 1. »Tastatur — eine Programmierhilfe« macht Programmieren einfach

```

320 CALL &BB18:CLS
330 PRINT"Auch der deutsche Zeichensatz
wurde definiert. Hierbei darf jedoch
das kleine >oe< nicht{2 SPACE}in RE
M- oder DATA-Zeilen be-{2 SPACE}nutz
t werden!"
340 PRINT
350 PRINT"Wuenschen Sie die Ausgabe des
deutschen Zeichensatzes und{3 SPACE}
der Basic-Befehle mit den zu- gehoer
igen Tasten ueber den{3 SPACE}Drucke
r?{3 SPACE}> j/n <"
360 a$=INKEY$:IF a$="" THEN 360
370 IF a$<>"j" THEN 760
380 PAPER 0:CLS
390 WINDOW 6,35,14,19:PEN 1:PAPER 2:CLS
400 PRINT:PRINT"Der deutsche Zeichensatz
und{2 SPACE}die Basic-Befehle mit d
en zu- gehoerigen Tasten werden ausg
edruckt. Bitte warten ....."
410 PRINT#8,CHR$(27);CHR$(87);CHR$(1);CH
R$(27);CHR$(69);CHR$(27);CHR$(71);ZO
NE 6
420 PRINT#8,,STRING$(30,"*")
430 PRINT#8,,***{7 SPACE}TASTATUR{9 SPA
CE}***
440 PRINT#8,,***{9 SPACE}eine{11 SPACE}
***
450 PRINT#8,,***{4 SPACE}Programmierhil
fe{4 SPACE}***
460 PRINT#8,,STRING$(30,"*")
470 PRINT#8,CHR$(27);CHR$(87);CHR$(0);CH
R$(27);CHR$(70);CHR$(15)
480 ZONE 12:WIDTH 138
490 PRINT#8,,STRING$(120,"-")
500 PRINT#8,,*";TAB(15)"Taste";TAB(22)"
allein";TAB(29)"mit SHIFT";TAB(40)"m
it CTRL";TAB(51)"* *";TAB(55)"Taste"
;TAB(62)"alleih";TAB(69)"mit SHIFT"
;TAB(80)"mit CTRL";
510 PRINT#8,TAB(91)"* *";TAB(95)"Taste"
;TAB(102)"allein";TAB(109)"mit SHIFT"
;TAB(120)"mit CTRL";TAB(132)"*";
520 PRINT#8,,*";TAB(43)"oder";TAB(51)"*
*";TAB(83)"oder";TAB(91)"* *";TAB(1
23)"oder";TAB(132)"*";
530 PRINT#8,,*";TAB(40)"CAPS LOCK";TAB(
51)"* *";TAB(80)"CAPS LOCK";TAB(91)"
* *";TAB(120)"CAPS LOCK";TAB(132)"*";
540 PRINT#8,,*";STRING$(116,"-");" *";
550 PRINT#8,,*";TAB(17)"die definierten
Basic-Befehle sind ....." ;TAB(
132)"*";
560 PRINT#8,,*";STRING$(116,"-");" *";
570 FOR d=1 TO 7:READ t1$,k1$,b1$,g1$,t2
$,k2$,b2$,g2$,t3$,k3$,b3$,g3$
580 PRINT#8,,*";TAB(17)t1$;TAB(25)k1$;T
AB(31)b1$;TAB(44)g1$;TAB(51)"* *";TA
B(57)t2$;TAB(65)k2$;TAB(71)b2$;TAB(8
4)g2$;TAB(91)"* *";TAB(97)t3$;TAB(10
5)k3$;TAB(111)b3$;TAB(124)g3$;TAB(13
2)"*";
590 NEXT
600 DATA A,a,AND,A,H,h,GOSUB,H,R,r,REM *
*,R,B,b,BORDER,B,I,i,INPUT,I,S,s,SY
MBOL,S,C,c,CLS,C,L,l,LOCATE,L,T,t,TH
EN,T,D,d,DATA,D,M,m,MODE,M,U,u,USING
,U,E,e,ELSE,E,N,n,NEXT,N,V,v,VAL,V,F
,f,FOR,F,O,o,OR,O,W,w,WINDOW,W,G,g,G
OTO,G,P,p,PRINT,P,Z,z,ZONE,Z
610 dg$=CHR$(27)+"=";dn$=CHR$(0);dr$=CHR
$(27)+"6";dm$=CHR$(27)+"7"

```

```

[1540]
[A188]
[FD88]
[C644]
[CA2C]
[B744]
[793E]
[2666]
[1C18]
[B4CE]
[7E7E]
[075E]
[38B2]
[C88C]
[9E86]
[B340]
[E1E2]
[B3F2]
[4026]
[FC84]
[695A]
[B8EE]
[CD18]
[04A2]
[5B1C]
[0512]
[2BE4]
[71FA]
[6AC0]
[DCAE]
620 PRINT#8,,*";TAB(17)"J";TAB(25)"j";T
AB(31)"J";TAB(44) dg$;CHR$(241);dn$;
TAB(52)"* *";TAB(58)"K";TAB(66)"k";T
AB(72)"K";TAB(85) dg$;CHR$(240);dn$;
TAB(93)"* *";TAB(134)"*";
630 PRINT#8,,*";TAB(17)"Q";TAB(25)"q";T
AB(31)"Q";TAB(51)"* *";TAB(57)"X";T
AB(65)"x";TAB(71)"X";TAB(84);dg$;CHR$
(203);dn$;TAB(92)"* *";TAB(98)"Y";T
AB(106)"y";TAB(112)"Y";TAB(133)"*";
640 PRINT#8,,*";TAB(17)";TAB(25)";";T
AB(31)"LIST";TAB(44)"*";TAB(51)"* *";
TAB(57)";";TAB(65)";";TAB(71)"LIST#
8";TAB(84)"+";TAB(91)"* *";TAB(132)"
*";
650 PRINT#8,,*";TAB(17)";TAB(25)";";T
AB(31)"INK";TAB(44)"<";TAB(51)"* *";
TAB(57)";";TAB(65)";";TAB(71)"PEN";T
AB(84)">";TAB(91)"* *";TAB(97)"T";T
AB(105)"/";TAB(111)"PAPER";TAB(124)"?";
TAB(132)"*";
660 PRINT#8,,*";TAB(17)"Anmerkung: Die
Zeichen ";dg$;CHR$(241);dn$;" ";dg$;
CHR$(240);dn$;" und ";dg$;CHR$(203);
dn$;" sind Steuerzeichen in PRINT-Ze
ilen, z.B. PRINT ";dg$;CHR$(34);CHR$
(241);dn$;
670 PRINT#8," zur n(chsten Zeile";dg$CHR
$(34);dn$;" usw. ";TAB(137)"*";
680 PRINT#8,,*";STRING$(116,"-");" *";
690 PRINT#8,,*";TAB(17)"die deutschen Z
eichen sind ....." ;TAB(132)"*";
700 PRINT#8,,*";STRING$(116,"-");" *";
710 PRINT#8,,*";dm$;TAB(18)"[";dr$;TAB(
27)"(";TAB(33)"[";TAB(46)";";dm$;TAB
(54)"* *";TAB(60)"J";dr$;TAB(69)";";
TAB(75)"\";TAB(88)";";dm$;TAB(96)"*
*";TAB(102)"\";dr$;TAB(111)";";TAB(1
17)"J";TAB(130)";";TAB(138)"*";
720 PRINT#8,,*";dm$;TAB(18)";";dr$;TAB(
27)";";TAB(33)";";TAB(46)";";dm$;TAB
(54)"* *";TAB(60)"e";dr$;TAB(69)";";
TAB(75)";";TAB(88)";";dm$;TAB(96)"*
*";TAB(102)";";dr$;TAB(111)";";TAB(1
17)";";TAB(130)";";TAB(138)"*";
730 PRINT#8,,*";TAB(17)"Anmerkung: Das
kleine > ! < darf nicht in REM- und
DATA-Zeilen benutzt werden !!!";TAB(
132)"*";
740 PRINT#8,,STRING$(120,"-")
750 PRINT#8,TAB(17) dg$;CHR$(164);dn$;"
1985 by kabasoft-computing"
760 PAPER 0:CLS
770 WINDOW 6,35,14,19:PEN 1:PAPER 2:CLS
780 LOCATE 6,3:PRINT"weiter bitte Taste"
;LOCATE 12,5:PRINT"> j/n <"
790 a$=INKEY$:IF a$="" THEN 790
800 IF a$="j" THEN 870
810 CLS:LOCATE 8,3:PRINT"* *{2 SPACE}E N
D E{2 SPACE}* *":CALL &BB18:END
820 REM ***
830 REM *** Programm >TASTATUR 2< ***
840 REM *** Dieses loescht den Pro-***
850 REM *** gramnteil >TASTATUR< ***
860 REM *** ***
870 CLS:PRINT:PRINT" Das Programm > TAST
ATUR 2 <{3 SPACE}wird geladen.":PRIN
T" Bitte warten ....."
880 RUN"!Tastatur 2"

```

Listing 1. »Tastatur — eine Programmierhilfe« macht Programmieren einfach (Schluß)

```

10 REM *****
20 REM *** TASTATUR ***
30 REM *** eine ***
40 REM *** Programmierhilfe ***
50 REM *****
60 REM *** Tastatur zuruecksetzen ***
70 REM ***
80 CALL &BB00
90 REM ***
100 REM *****
110 REM *** Basic-Befehle definieren **
120 REM *** 1. Buchstabe=Basic-Befehl **
130 REM *** **
140 KEY 128,"AND"
150 KEY 129,"BORDER"
160 KEY 130,"CLS"
170 KEY 131,"DATA"
180 KEY 132,"ELSE"
190 KEY 133,"FOR"
200 KEY 134,"GOTO"
210 KEY 135,"INPUT"
220 KEY 136,"LOCATE"
230 KEY 137,"MODE"
240 KEY 138,"NEXT"
250 KEY 139,"OR"

```

```

[AE76]
[AC74]
[99C8]
[60A2]
[C67E]
[6740]
[968E]
[08DC]
[4E92]
[8E96]
[3176]
[EE96]
[7394]
[0E38]
[A412]
[2E4C]
[AAC0]
[3CE2]
[6162]
[F2F8]
[E8AA]
[71FE]
[E9DC]
[3C14]
[EFDC]
[CB86]
[9494]
[8040]
[E9F6]
[E696]
[2054]
[9E42]
[E90E]
[000C]
[6A60]
[4166]
[8782]
[0296]
[5112]
[ACBA]
[75AA]
[7A98]
[17AE]
[F052]
[002A]
260 KEY 140,"PRINT"
270 KEY 141,"REM ***"
280 KEY 142,"SYMBOL"
290 KEY 143,"THEN"
300 KEY 144,"USING"
310 KEY 145,"VAL"
320 KEY 146,"WINDOW"
330 KEY 147,"ZONE"
340 REM *** weitere Basic-Befehle ***
350 KEY 148,"INK"
360 KEY 149,"PEN"
370 KEY 150,"PAPER"
380 KEY 151,"GOSUB"
390 KEY 152,"LIST"
400 KEY 153,"LIST#8"
410 REM ***
420 REM *** deutsche Zeichen definieren
430 REM ***
440 SYMBOL AFTER 90
450 SYMBOL 91,198,16,56,108,198,254,198,
0

```

Listing 2. Im zweiten Teil wird die Tastatur angepaßt

```

460 SYMBOL 92,102,56,108,198,198,108,60,
0 [6C0A]
470 SYMBOL 93,102,0,102,102,102,102,60,0 [8E44]
480 SYMBOL 123,102,0,120,12,124,204,118,
0 [4AB0]
490 SYMBOL 124,102,0,60,102,102,102,60,0 [1F44]
500 SYMBOL 125,102,0,102,102,102,102,59,
0 [DEA0]
510 SYMBOL 126,120,198,198,252,198,198,2
48,192 [0926]
520 REM *** Tastatur neu definieren *** [7E00]
530 REM *** [AFB0]
540 REM *** Basic-Befehle auf *** [B69E]
550 REM *** Tasten + SHIFT legen *** [88BA]
560 REM *** [90B6]
570 REM *** urspruengliche Funktion *** [AEB4]
580 REM *** Tasten + SHIFT jetzt *** [C90C]
590 REM *** Tasten + CTRL !!!!! *** [21A4]
600 REM *** [E8AC]
610 FOR t1=1 TO 26:READ a,b,c,d:KEY DEF
a,1,b,c,d:NEXT [F96C]
620 DATA 69,97,128,65,54,98,129,66,62,99
,130,67,61,100,131,68,58,101,132,69,
53,102,133,70,52,103,134,71,35,105,1
35,73,36,108,136,76,38,109,137,77,46
,110,138,78,34,111,139,79,27,112,140
,80,50,114,141,82,60,115,142,83,51,1
16,143,84,42,117,144,85 [7758]
630 DATA 55,118,145,86,59,119,146,87,71,
122,147,90,39,44,148,60,31,46,149,62
,30,47,150,63,44,104,151,72,29,58,15
2,42,28,59,153,43 [5476]
640 REM *** [BCB4]
650 REM *** Deutsche Zeichen auf *** [58A4]
660 REM *** Tasten mit und ohne *** [500C]
670 REM *** SHIFT legen. *** [1A68]
680 REM *** [50BC]
690 FOR t2=1 TO 4:READ a,b,c:KEY DEF a,0
,b,c:NEXT [9DD4]
700 DATA 17,123,91,19,124,92,22,125,93,2
6,126,166 [005E]
710 REM *** [B1B0]
720 REM *** Zahlenblock auf Zahlen *** [43C8]
730 REM *** umstellen *** [DD26]
740 REM *** [92B6]
750 FOR t3=1 TO 12:READ a,b,c,d:KEY DEF
a,0,b,c,d:NEXT [4C6E]
760 DATA 10,55,55,55,11,56,56,56,3,57,57
,57,20,52,52,52,12,53,53,53,4,54,54,
54,13,49,49,49,14,50,50,50,5,51,51,5
1,15,48,48,48,7,46,44,46,6,13,13,13 [9414]
770 REM *** [4FBC]
780 REM *** Auch nach dem Loeschen *** [3402]
790 REM *** des Listings bleiben *** [23F4]
800 REM *** die Funktionen voll er- *** [EBD8]
810 REM *** halten. *** [A046]
820 REM *** Nach einem Symbol after *** [FED4]
830 REM *** werden jedoch die *** [473E]
840 REM *** deutschen Zeichen ge- *** [7DFC]
850 REM *** loescht!!!! [74E8]
860 REM *** [EBEC]
870 CLS:LOCATE 2,2:PRINT" In welchem MOD
E >1 oder 2<{4 SPACE}michten Sie ein
tippen?{2 SPACE}";:LINE INPUT;a$:a=V
AL(a$) [9136]
880 ON a GOTO 900,1020 [F2BE]
890 REM *** [B682]
900 REM *** MODE 1 *** [91DE]
910 REM *** [4974]
920 MODE 1:INK 1,11:BORDER 3 [5FEF]
930 WINDOW 1,40,1,1:PEN 3:PAPER 0:CLS [B97A]
940 ZONE 14:PRINT,"* TASTATUR *" [8B82]
950 WINDOW 1,40,25,25:PEN 3:PAPER 0:CLS [A656]
960 ZONE 6:PRINT,CHR$(164);" 1985 by ka
basoft-computing" [E260]
970 WINDOW 1,40,2,24:PEN 1:PAPER 2:CLS [EDEE]
980 LOCATE 9,13:PRINT"Programm lischen?
>j/n<" [6156]
990 a$=INKEY$:IF a$="" THEN 990 [8250]
1000 IF a$="j" THEN NEW [7B82]
1010 CLS:END [3CAA]
1020 REM *** MODE 2 *** [2974]
1030 REM *** [02C8]
1040 MODE 2:INK 1,15:BORDER 1 [6348]
1050 WINDOW 1,80,1,1:PEN 0:PAPER 1:CLS [6BD2]
1060 ZONE 34:PRINT,"* TASTATUR *" [78DA]
1070 WINDOW 1,80,25,25:PEN 0:PAPER 1:CLS [D4AE]
1080 ZONE 26:PRINT,CHR$(164);" 1985 by
kabssoft-computing" [AC18]
1090 WINDOW 1,80,2,24:PEN 1:PAPER 0:CLS [BB46]
1100 GOTO 980 [70B8]

```

Listing 2. Im zweiten Teil wird die Tastatur angepaßt (Schluß)

Unsichtbare Programme



Fünf Kommas und ein Doppelpunkt reichen aus, um eine Programmzeile zu verstecken. Mit »Superlistschutz« schützen Sie das ganze Programm vor dem LIST-Befehl.

Es gibt unzählige Tricks, Programmzeilen aus einem Listing scheinbar verschwinden zu lassen. Man muß verhindern, daß bestimmte, für das Programm wichtige Anweisungen gelistet werden. Der Interpreter soll diese Befehle aber trotzdem finden und bearbeiten können. Manipulationen an Adressen, in denen Zeilennummer oder -länge stehen, sind am einfachsten zu realisieren.

Hier finden Sie nun eine neue Methode, verschiedenste Anweisungen (oder auch das ganze Programm) optisch verschwinden zu lassen. Gibt man hinter der Zeilennummer einen Doppelpunkt und daran anschließend fünf Kommas ein, so genügt ein POKE-Befehl, um die ganze Zeile zu verstecken. Falls

Sie es nicht glauben wollen, geben Sie einfach folgende Zeile ein:

```
10 :,,,,PRINT "Dieser Text ist unLISTbar" und direkt »POKE 368+5,0« ein.
```

Mit »LIST« wird auf dem Bildschirm nur »10:« ausgegeben. Nach »RUN« wird die Zeile 10 aber korrekt bearbeitet und »Dieser Text ist unLISTbar« ist zu lesen. So kann man beispielsweise einen Copyright-Vermerk in ein Programm einbauen, der nur sehr schwer wieder zu löschen ist.

Falls man jede einzelne Zeile so manipuliert hat, kann das ganze Programm zwar abgearbeitet, aber nicht gelistet werden. Der Aufwand ist jedoch sehr groß. Mit dem Programm »Superlistschutz« werden die zu schützenden Zeilen automatisch umgewandelt. Nach dem Eingeben der Routine (ab Zeile 50000) muß das zu bearbeitende Programm mit »MERGE "Name"« geladen werden. Danach nur noch die Funktionstaste »f9« (Taste »9« auf dem Zahlenblock) drücken und »Superlistschutz« startet.

- Drei Punkte muß man allerdings beachten:
1. Da das Programm aus Platz- und Geschwindigkeitsgründen mit Integerzahlen arbeitet, dürfen die Zeilennummern nicht größer als 32767 sein.
 2. Die Zeilennummer 0 ist nicht zulässig.
 3. Da jede Zeile um sechs Zeichen länger wird, darf die ursprüngliche Folge nicht mehr als 249 Zeichen haben.

Die wichtigste Anweisung finden Sie mit dem EDIT-Befehl in der letzten Programmzeile. Sämtliche Zeilen des zu schützenden Programms werden automatisch um einen Doppelpunkt und fünf Kommas ergänzt und durch den POKE-Befehl in Zeile 50880 »versteckt«.

(Jörg Braun)

```

40000 REM ***** [26B4]
40010 REM * (c) Joerg Braun * [D0A0]
40020 REM * Waldenburger Str. 5 * [635B]
40030 REM * 3330 Helmsstedt * [7844]
40040 REM ***** [22BC]
50000 MEMORY HIMEM -41:
REM Bis Zeile 50250 befindet sich
die [C1B6]
50010 DEF FNmsb(a)=&FF AND INT(a)/256:
REM Mergeroutine fuer Floppybesitz
er. [B2DC]
50020 DEF FNlsb(a)=&FF AND UNT(a):
REM User ohne Floppy bitte nicht a
b- [6D6E]
50030 FOR i=HIMEM +1 TO HIMEM +38:
REM tippen. [0DEC]
50040 READ byte [CB12]
50050 POKE i,byte [7C64]
50060 NEXT i [81C6]
50070 POKE HIMEM +3, FNlsb(HIMEM +39) [13A4]
50080 POKE HIMEM +4, FNmsb(HIMEM +39) [73AA]
50090 POKE HIMEM +9, FNlsb(HIMEM +41) [C7A6]
50100 POKE HIMEM +10, FNmsb(HIMEM +41) [E6E8]
50110 POKE HIMEM +18, FNlsb(HIMEM +1) [D090]
50120 POKE HIMEM +19, FNmsb(HIMEM +1) [AE96]
50130 REM cas_in_char [500E]
50140 POKE HIMEM +39, PEEK(&BC80+0) [5640]
50150 POKE HIMEM +40, PEEK(&BC80+1) [3234]
50160 POKE HIMEM +41, PEEK(&BC80+2) [013A]
50170 POKE &BC80+0, &C3 [F244]
50180 POKE &BC80+1, FNlsb(HIMEM +1) [8434]
50190 POKE &BC80+2, FNmsb(HIMEM +1) [503A]
50200 DATA &e5, &2a, &00, &00, &22, &80, &bc [9BC2]
50210 DATA &3a, &00, &00, &32, &82, &bc [64F4]
50220 DATA &cd, &80, &bc, &21, &00, &00 [BA54]
50230 DATA &22, &81, &bc, &21, &80, &BC [D090]
50240 DATA &36, &c3, &e1, &d8, &c8, &fe, &1a [5120]
50250 DATA &37, &3f, &c0, &b7, &37, &c9 [5096]
50260 '*** Hier Ende der Floppyroutine *
** [AB3E]
' zn:Zeilennummer [6C04]
' ad:Zeilenadresse [4066]
' a:Schleifenvariable [8AF2]
' b$:Schleifenstring (fuer Datas) [3008]
50300 : [0CA4]
50310 MODE 2:WINDOW #2,1,80,4,22:CLG 1:C
LS#2 [6594]
50315 WINDOW#3,1,80,10,15 [45B8]
50320 POKE &B1C8,0:POKE &B1CF,&F0:POKE &
B1D0,&F: REM Bildschirmmode umsch
alten [67E4]
50330 LOCATE 2,2:PRINT "Listschutz-Spezi
al" [6EE0]
50340 LOCATE 2,24:PRINT " "CHR$(164)"198
5 by J.Braun " [4D6C]
50350 POKE &B1C8,2:
REM Wieder nach Mode 2 [331A]
50360 WINDOW SWAP 3 [5548]
50370 PRINT"Merge Programmname" [C7CC]
50380 PRINT"Starten mit Funktionstaste 9
" [4C2E]
50390 KEY 9,"run 50790"+CHR$(13):END [807E]
50400 '*** *** *** *** *** [5E6C]
50410 : [15A8]
50420 REM Hier beginnt die Tastenpufferm
anipulation [6B50]
: [03AC]
50440 RESTORE 50530:FOR a=1 TO 16 [1670]
50450 READ b$:b$="&"+b$ [52F0]
50460 POKE &B514+a-1,VAL(b$) [2080]
50470 NEXT [09BE]
50480 POKE &B53C,13 [8126]
50490 POKE &B53D,8 [52D2]
50500 POKE &B53E,9 [CDC6]
50510 POKE &B53F,0 [0FB8]
50520 RETURN [04F8]
50530 DATA 00,02,00,02,00,02,00,02,01,40
,00,20,02,10,02,04 [0006]
'*** *** *** *** [E576]
50540 : [236E]
50550 RESTORE 50640:FOR a=1 TO 20 [17F4]
50560 READ b$:b$="&"+b$ [A184]
50570 POKE &B514+a-1,VAL(b$) [AAC2]
50580 NEXT [4C26]
50590 POKE &B53C,11 [0E16]
50600 POKE &B53D,10 [401C]
50610 POKE &B53E,11 [08BC]
50620 POKE &B53F,0 [F1FC]
50630 RETURN [3F72]
50640 DATA 00,02,00,02,00,02,00,02,00,02
,00,02,01,40,00,20,02,10,02,04 [6A7A]
'*** *** *** *** [5372]
50660 RESTORE 50700:FOR a=1 TO 14 [D1C8]
50670 READ b$ [56B2]
50680 POKE &B514+a-1,VAL("&"+b$) [AFC6]
50690 NEXT [94D0]
50700 DATA 00,02,00,02,00,02,01,40,00,20
,02,10,02,04 [6E20]
50710 POKE &B53C,14 [6E20]
50720 POKE &B53D,7 [F7C8]
50730 POKE &B53E,8 [C9CE]
50740 POKE &B53F,0 [19C2]
50750 RETURN [E102]
50760 '*** *** *** *** [687E]
50770 REM Ende Tastenpuffermanipulation [2B96]
50780 : [1CBC]
50790 REM Hauptprogramm [1020]
50800 KEY 2,CHR$(0):
REM Es wird das Druecken dieser [37AE]
50810 KEY 3,"",,,,,"+CHR$(13)+CHR$(13):
REM 3 Tasten sowie mehrer Cursor [6F6E]
-
50820 KEY 4,"goto 50880"+CHR$(13):
REM tasten simuliert. [A608]
50830 : [2BB4]
50840 zn=PEEK(370)+256*PEEK(371):ad=368 [1C6C]
50850 GOTO 50900 [77F2]
50860 '***** Ende Initialisierung ****
*** [5744]
50870 : [17BC]
50880 POKE ad+5,0: REM
Zeilenadresse+5 auf 0 poken [9654]
50890 ad=ad+PEEK(ad): REM
Auf naechste Zeile erhoehen [5B3E]
50900 zn=PEEK(ad+2)+256*PEEK(ad+3): REM
Neue Zeilennummer auslesen [5D18]
50910 : [4DB2]
50920 IF PEEK(ad)+256*PEEK(ad+1)>249 THE
N MODE 1:
PRINT"Zeile "zn" ist z
u lang":CALL &BB03:STOP:
REM Puffer
entleeren,noch erlaubte Zeilenlae
nge ? [27D6]
50925 : [3EBE]
50930 IF zn>=50000 THEN CLS:CALL &BB03:
PRINT"Geschuetztes
Programm kann gespeichert werden":
DELETE
50000- [4DB2]
: [54B8]
50940 :
50950 IF zn>99 AND zn <1000 THEN GOSUB 5
0440: 'Tastenpuffermanipulation auf
rufen [06E6]
50960 IF zn< 100 THEN GOSUB 50660 [3070]
50970 IF zn>999 THEN GOSUB 50550 [CF66]
50980 : [48C0]
50990 IF zn>32767 THEN MODE 1:PRINT"{CTR
L G}Zeilennummer "zn" zu gross":
CALL &BB03:S
TOP:
REM Noch Integer ? [576C]
51000 : [08A0]
51010 lbzn=zn MOD 256: REM Lowbyte Zeile
nummer= zn Mod 256 [33A0]
51020 hbzn=zn \ 256 : REM Highbyte Zeil
ennummer= zn \ 256 [ED26]
51030 : [06A6]
51040 POKE PEEK(&AE83)+256*PEEK(&AE84)-5
,lbzn: REM Zeilennummer bei Ed
it [9D12]
51050 POKE PEEK(&AE83)+256*PEEK(&AE84)-4
,hbzn: REM veraendern [C87C]
51060 : [1886]
51070 REM Zeiger auf Programmende wenige
r 5 Bytes = Zeilennummer bei edit [54FE]
51080 EDIT 51040 [FBBC]

```

Listing. Machen Sie mit »Superlistschutz«
Ihre Programme unsichtbar

Tricks mit dem Joystick



Lesen Sie, was Sie mit dem Joystick alles machen können.

Der CPC 464 hat mit anderen Computern, beispielsweise dem 64, gemeinsam, daß die Joysticks auch mit der Tastatur abgefragt werden können. Wer Programme mit Joystick-Bedienung schreibt, sollte deshalb »INKEY(x)« oder »INKEY\$« statt »JOY(x)« verwenden, damit sich Besitzer ohne Joystick das Programm mit einigen KEY DEF-Befehlen anpassen können.

Wie aus dem Anhang II (Seite 16) des Benutzerhandbuchs zu ersehen, besitzt die Tastatur für diese KEY DEF-Befehle eine besondere Numerierung. Die Werte gehen von 0 bis 79, wobei die Werte 72 bis 77 für den Joystick 0 reserviert sind. Die Werte 48 bis 53 für den Joystick 1 liegen parallel zu den Tasten 5, 6, F, G, R und T. Die Numerierung erlaubt, daß die Zuordnung der Joystick-Bewegung und Feuertasten leicht verändert werden kann.

Der Joystick als Cursor:

Mit »RUN 50« wird der Joystick so definiert, daß er anstelle der Cursorstasten verwendet werden kann. Die vier Richtungen entsprechen den Cursorrichtungen, die beiden Feuertasten der Copy-Taste. Drücken Sie gleichzeitig Shift und den Joystick in eine Richtung, so erscheint der Copy-Cursor. Mit Ctrl und Joystick nach links wird der Cursor an den Anfang der Bildschirmzeile, bei Joystick nach rechts gemeinsam mit Ctrl an das Ende der Zeile gesetzt. Mit Ctrl und Joystick nach oben gelangt man an den Anfang der Eingabezeile, bei Joystick nach unten an das Ende der Eingabezeile. Nach kurzer Gewöhnungszeit läßt sich der Joystick angenehm benutzen.

Dauerfeuer:

Haben Sie keinen Joystick mit Dauerfeuer-Knopf, so sind die Zeilen 130 bis 210 für Sie besonders interessant: Nach »RUN 130« verfügen beide Joysticks über Dauerfeuer — aber nur, wenn die Joysticks mit »INKEY(x)« oder »INKEY\$« beziehungsweise den Maschinensprache-Äquivalenten abgefragt werden. Bei »JOY(x)« hängt es von der Programmierung ab, ob ein Dauerfeuer möglich ist.

Joystick-Zuordnung verändert

Für Programme, die nur Joystick 1 abfragen und die Joysticks mit »INKEY(x)« oder »INKEY\$« abfragen, können Sie mit »RUN 220« die Zuordnung des Joysticks 0 auf 1 ändern.

Caps-Lock und Shift-Lock:

Oft ist es nützlich zu wissen, ob die Caps-Lock oder Shift-Lock-Tasten gedrückt worden sind. Dies kann man durch Abfrage der Speicherstellen B4E7hex und B4E8hex erfahren. Nach »RUN 300« wird der aktuelle Zustand ausgegeben.

Mit Ctrl und Caps-Lock wird der Modus Shift-Lock eingeschaltet und auch wieder abgeschaltet. In diesem Modus werden auch die Tasten der Zahlenreihe in der Shift-Ebene abgefragt. Zahlen können dann nur noch über den Zehnerblock eingegeben werden.

Mit POKE-Befehlen an dieser Adresse kann Shift-Lock oder Caps-Lock eingeschaltet werden. Starten Sie mehrmals »RUN 340«. Jedesmal wird zwischen Groß- und Kleinschreibung hin und her geschaltet.

(Martin Kotulla)

```

10 (C) Martin Kotulla [B7C6]
20 Grabbestr. 9 [C3D8]
30 8500 Nuernberg [F9BE]
40 [8256]
50 Joy-0 als Cursor ***** [9D32]
60 KEY DEF 72,1,&F0,&F4,&F8 [2450]
70 KEY DEF 74,1,&F2,&F6,&FA [0C70]
80 KEY DEF 73,1,&F1,&F5,&F9 [E65C]
90 KEY DEF 75,1,&F3,&F7,&FB [BE7C]
100 KEY DEF 76,1,&E0,&E0,&E0 [3D90]
110 KEY DEF 77,1,&E0,&E0,&E0 [F394]
120 END [AB14]
130 Dauerfeuer ***** [057E]
140 - Joystick 0 [DCD2]
150 CALL &BB00 'Tastatur initialisieren [2168]

160 KEY DEF 76,1,&58,&58,&58 [C16C]
170 KEY DEF 77,1,&5A,&5A,&5A [07A6]
180 - Joystick 1 [8ADC]
190 KEY DEF 52,1,&67,&47,&7 [1EF6]
200 KEY DEF 53,1,&66,&46,&6 [16E2]
210 END [A014]
220 Joy-0 als Joy-1 ***** [6708]
230 KEY DEF 72,1,&36,&26 [18D0]
240 KEY DEF 74,1,&72,&52,&12 [763E]
250 KEY DEF 73,1,&35,&25 [DAD2]
260 KEY DEF 75,1,&74,&54,&14 [9850]
270 KEY DEF 76,1,&67,&47,&7 [8D00]
280 KEY DEF 77,1,&66,&46,&6 [92FE]
290 END [0824]
300 Abfrage von Caps/Shift Lock ***** [8E08]
310 IF PEEK(&B4E8)=255 THEN PRINT "Caps [11F8]
Lock!" [F2E6]
320 IF PEEK(&B4E7)=255 THEN PRINT "Shift [A41A]
Lock!" [B606]
330 END [2A36]
340 Setzen von Caps/Shift Lock ***** [9F20]
350 POKE &B4E8,255 AND NOT PEEK(&B4E8)
360 END
    
```

Listing. Der Joystick hat viele Vorteile

Schnell gespeichert



Zwei POKEs an der richtigen Stelle reichen aus, um die Speicher-Geschwindigkeit zu erhöhen.

Wie man die Geschwindigkeit des eingebauten Kassettenrecorders verändert, das können Sie in dem Artikel »Was tun, wenn's Fehler hagelt?« lesen. Mit zwei POKE-Anweisungen lassen sich aber auch direkt die Werte für die CAS SET SPEED-Routine beeinflussen.

Der notwendige Basic-Befehl lautet »POKE &B8D1,x:POKE &B8D2,y«, wobei Sie die Werte für x und y aus der Tabelle übernehmen können.

(Martin Kotulla)

	&B8D1	&B8D2
1000 Baud (SPEED WRITE 0)	6	83
2000 Baud (SPEED WRITE 1)	12	41
3600 Baud (Maximalwert)	2	23

Dieser Wert muß gePOKEt werden

Kleine Buchstaben ganz groß



Machen Sie Ihr Programm mit einer kleinen Hilfsroutine interessanter. Mit ihr können Sie Buchstaben und andere Zeichen in vierfacher Größe auf den Bildschirm zaubern.

Der Trick der Routine »Big Letters« besteht darin, daß der gesamte Zeichensatz mit dem Befehl »SYMBOL AFTER 0« ins RAM kopiert wird. Dort kann er — im Gegensatz zum ROM, in dem er normalerweise steht — mit PEEK ausgelesen werden. Das Programm wandelt nun jede Zahl, die einer Rasterzeile des Buchstabens entspricht, in einen Binärstring um. Der Befehl in Zeile 200 prüft, welche Bits in dem Zeichen gesetzt sind. Die PLOT-Anweisungen in den Zeilen 210 bis 240 setzen dann diesen Punkt in vierfacher Größe.

Sie sind übrigens nicht auf eine bestimmte Buchstabengröße festgelegt. Fügen Sie einige PLOT-Befehle dazu oder lassen Sie einige weg, so verändert sich die Größe des ausgegebenen Zeichens.

(Martin Kotulla)

```

10  BIG LETTERS                [B9BA]
20  [6052]
30  *****                  [7B30]
40  [8256]
50  (C) Martin Kotulla        [4FCE]
60  Grabbestr. 9              [EBE0]
70  8500 Nuernberg           [39C6]
80  [865E]
90  CLS:SYMBOL AFTER 0:MODE 1 [051A]
100 INPUT "Welcher Text? ",ascii$ [BED8]
110 ORIGIN 1,300              [9414]
120 FOR k=1 TO LEN(ascii$)    [6BA6]
130  ascii=ASC(MID$(ascii$,k,1)) [4924]
140  charadr=HIMEM+ascii*8+8   [D760]
150  FOR i=1 TO 8              [2F5E]
160  a=PEEK(charadr)          [9940]
170  binaer$=BIN$(a,8)        [603A]
180  FOR j=1 TO 8              [4866]
190  segment$=MID$(binaer$,j,1) [EDCA]
200  IF segment$="0" THEN 250  [3A5E]
210  PLOT x+j*4,i*4           [EAEO]
220  PLOT x+2+j*4,i*4         [DF9C]
230  PLOT x+j*4,i*4+2        [DB9E]
240  PLOT x+2+j*4,i*4+2      [705A]
250  NEXT j:charadr=charadr-1:NEXT i [A702]
260  x=x+32                   [B5EA]
270  NEXT k                    [4106]
280  LOCATE 1,23:END          [8CCA]
    
```

Listing.

Viermal so große Buchstaben erzeugt »Big Letters«

Welcher Text? Big Letters

Big Letters

»Big Letters« — Buchstaben ganz groß

Endlich wieder sichtbar



Der Befehl »SAVE "Name",p« ist zwar praktisch, wenn man nicht jedem sein Programm zeigen oder es schützen will. Aber was macht man, falls noch etwas geändert werden muß?

Wer hat sich nicht schon über den eigenen Leichtsinn geärgert, ein Programm geschrieben und es dann dummerweise geschützt (protected) gespeichert zu haben? Besonders wenn keine andere Kopie von dem Meisterwerk vorhanden ist! Jetzt kann man zwar das eigene Programm laden und ablaufen lassen, aber Änderungen sind nicht mehr möglich. Um Ihnen aus diesem Dilemma herauszuhelfen, soll die kleine Routine dienen.

Ein Basic-Programm, das nur aus zwei Zeilen besteht, macht aus einem »protected« ein »unprotected«. Die Routine steht im Speicher an einem Platz, der von der Betriebssystem-Routine »CAS IN DIREKT« direkt aufgerufen wird. Sie schreibt den Wert Null in die Speicherzelle AE2Chex, die das Protected-Flag darstellt. Der Wert Null bedeutet in diesem Fall, daß dieses Programm nicht geschützt ist.

Wer nun aber glaubt, auch professionelle Programme damit knacken zu können, der wird enttäuscht sein. Die Software-Häuser verlassen sich schon lange nicht mehr auf den eingebauten Schutz. Sie benutzen eigene Routinen.

(Michael Bauer)

```

10  *****                  [3E10]
20  * (c) Michael Bauer      * [A9A6]
30  * Schwanthalerstr. 180  * [9204]
40  * 8000 Muenchen 2      * [19DB]
50  *****                  [3E18]
60  FOR p=&BE7F TO &BE84:READ d:POKE p,d: [919B]
    NEXT p                    [EC4C]
70  DATA &3E,0,&32,&2C,&AE,&C
    
```

Listing. Mit zwei Zeilen können Sie Ihr versehentlich gesichert abgespeichertes Programm wieder lesen



HOSE?BAR

Daten im direkten Zugriff



Relative Dateiverwaltung ist mit dem DOS der Schneider-Diskettenstation nicht ohne weiteres möglich. Im ROM sind die notwendigen

Routinen aber schon vorgesehen. Mit einer Befehlsweiterung kann man sie leicht benutzen. Adressen komfortabel zu speichern ist dann beispielsweise kein Problem mehr.

Dem komfortablen Basic des CPC fehlt eine wichtige Routine, die relative Dateiverwaltung. Leider sieht Amsdos diese Möglichkeit nicht vor, selbst nicht über die Sprungvektoren.

Es gibt aber dennoch einen brauchbaren Weg relative Dateien zu erzeugen. Beim Studium der ROM-Routine der Diskettenstation fällt auf, daß Amsdos-Dateien in sogenannten Records (das sind Aufzeichnungseinheiten zu je 128 Bytes) auf Diskette geschrieben sind. Dies geschieht unabhängig von der Größe der physikalischen Sektoren, die 512 Bytes umfassen.

Glücklicherweise gibt es Amsdos-Routinen, die — ähnlich wie unter CP/M — einen bestimmten Record einer Datei lesen und beschreiben können, ohne die davorliegenden erst lesen zu müssen. Dies ist Voraussetzung für den Direktzugriff.

Eine relative Datei wird zunächst einmal als sequentielle Datei angelegt. Dies geschieht am einfachsten, indem wir mit SPACE\$ Leerstrings erzeugen, die exakt der gewünschten Datensatzlänge entsprechen. Um die relative Datei später wahlweise auch sequentiell lesen zu können, fügen wir an jeden Leerstring noch ein Carriage Return (CR)-Zeichen (CHR\$(13)) und ein Line Feed (LF)-Zeichen (CHR\$(10)) an. Allerdings ist das für die relative Datei nicht unbedingt erforderlich. Immerhin werden dadurch für jeden Datensatz zwei Bytes mehr benötigt.

Wir schreiben nun die Anzahl von Leerstrings (einschließlich CR und LF) in die sequentielle Datei, die die relative Datei an Datensätzen erhalten soll. Dann können wir mit der Befehlsweiterung »RECWRITE« je

den einzelnen Datensatz beschreiben und mit »RECREAD« wieder lesen.

Übrigens muß die Datensatzlänge keinesfalls mit der Recordlänge von 128 Bytes übereinstimmen. Sie kann frei zwischen 1 und 255 Zeichen gewählt werden; selbst wenn ein Datensatz über zwei oder drei Records verteilt ist. Die Befehlsweiterung ist deshalb so aufgebaut, daß ein String mit gegebener Länge ab dem n-ten Zeichen der Datei geschrieben beziehungsweise gelesen wird. Das erste Zeichen in der Datei hat dabei den Wert Null. Das Maschinencode-Programm errechnet hieraus automatisch den zugehörigen Record und die Position in diesem, ab der gelesen oder geschrieben werden soll.

Bei jedem Schreib- und Lesevorgang werden drei hintereinanderliegende Records in einen Puffer gelesen beziehungsweise von dort aus wieder auf die Diskette geschrieben. Die Anzahl der Bytes ist in der Stringlänge gegeben.

Hier ein Beispiel, das diesen Vorgang verdeutlichen soll. Angenommen, wir haben eine Datensatzlänge von 200 Bytes, einschließlich CR und LF, und möchten den 38. Datensatz lesen und ihn in der Stringvariablen a\$ ablegen. Zunächst muß mit »a\$ = SPACE\$(198)« ein Leerstring mit 198 Zeichen (200 abzüglich CR und LF) erzeugt werden. Als nächstes werden drei aufeinanderfolgende Records in den Puffer gelesen, beginnend mit dem, der das 7600.(=38*200) Zeichen der Datei enthält. Nun ist aber 7600/128 = 59 Rest 48 weshalb wir die Records 59, 60 und 61 einlesen müssen. Da die Bytezahl der Records nicht bei 1, sondern bei Byte 0 beginnt, greifen wir

auf Record 59, Byte 47 zu. Der 38. Datensatz ist somit in Record 59, Byte 47 bis 127 (80 Bytes) und Record 60, Byte 0 bis 117 (118 Bytes) (ohne CR, LF) gespeichert. Es erübrigt sich, in diesem Fall auf den dritten Record zuzugreifen. Er wird aber immer mitgelesen und -geschrieben. Zum Schluß wird der Datensatz aus dem Recordpuffer in den vorher angelegten Leerstring zur weiteren Bearbeitung übertragen.

Beim Schreiben werden zuerst die drei betroffenen Records von Diskette gelesen. Dann wird der Inhalt des Datensatzes in den Puffer an der Stelle abgelegt, von wo sie wieder auf Diskette geschrieben werden. Die Größe der relativen Datei ist lediglich durch die Aufzeichnungskapazität der Diskette beschränkt, die beim Schneider 180 KByte (Datenformat) beträgt.

Beachten Sie aber, daß sowohl vor Schreib- als auch Lesezugriffen in relativen Dateien immer die OPENIN-Anweisung nötig ist. Niemals darf aber OPENOUT benutzt werden, da dies eine neue Datei anlegen würde.

Ebenfalls unter allen Umständen zu vermeiden ist ein Schreibzugriff auf eine höhere Datensatznummer, als für die die Datei ausgelegt ist. Deshalb sollten Sie eine Sicherheitsabfrage in Ihre Programme einbauen, die so etwas verhindert. Wenn darüber hinaus beim Anlegen der Datei immer zwei zusätzliche Records reserviert werden, sind Sie vor unangenehmen Überraschungen sicher, wenn Sie auf einen der letzten Sätze zugreifen.

Bevor nun die Arbeit mit den neuen Befehlen an einem Beispiel beginnt, müssen Sie den Basic-Lader (Listing 1) für das Maschinenprogramm eingeben. Die Routine liegt dann ab A000 hex. Möchten Sie die Erweiterung in Ihre eigenen Programme einbinden, gibt es drei verschiedene Wege. Entweder Sie benutzen den Basic-Lader und laden später Ihr eigenes Programm. Ein anderer Weg: Sie hängen den Lader mit Hilfe von »CHAINMERGE« an Ihre Programme an und rufen ihn dann als Unterprogramm auf. Dazu müssen Sie die Programmzeilen der Routine mit »RENUM« neu ordnen, so daß sie höhere Zeilennummern benutzt als Ihr BASIC-Programm. Achten Sie aber darauf, daß Sie in Zeile 240 NEW durch RETURN ersetzen.

Die letzte und bequemste Möglichkeit besteht darin, den Lader abzuarbeiten und als Binärdatei auf Diskette zu schreiben. Dies ge-

```

100 REM BASIC-Lader fuer Maschinenprogra mm [2B32]
110 REM Relative Dateiverwaltung auf [AE48]
130 REM CPC 464, 664 und 6128 [3D20]
140 REM ===== [FAA6]
== [DBE0]
150 : [CAE2]
160 : [A10C]
170 MEMORY &A000-1 [4ABA]
180 RESTORE [F5AC]
190 FOR i=&A000 TO &A17F [C39E]
200 READ a [11F0]
210 POKE i,a [6CE6]
220 NEXT [C9D0]
230 CALL &A000 [8840]
240 NEW [D7E2]
250 : [DCE4]
260 :
1000 DATA &01,&0A,&A0,&21,&22,&A0,&CD,&D [D9EC]
1 DATA 1 [D9EC]
1010 DATA &BC,&C9,&12,&A0,&C3,&CF,&A0,&C [7876]
3 [7876]
1020 DATA &0D,&A1,&52,&45,&43,&52,&45,&4 [3DA6]
1 [3DA6]
1030 DATA &C4,&52,&45,&43,&57,&52,&49,&5 [F1B2]
4 [F1B2]
1040 DATA &C5,&00,&00,&00,&00,&00,&00,&0 [F336]
0 [F336]
1050 DATA &00,&00,&00,&00,&00,&00,&00,&0 [9E08]
0 [9E08]
1060 DATA &00,&C3,&CF,&A0,&C3,&0D,&A1,&F [0E78]
E [0E78]
1070 DATA &02,&C2,&B4,&A0,&DD,&6E,&02,&D [F762]
D [F762]
1080 DATA &66,&03,&2B,&7E,&FE,&04,&C2,&B [A540]
4 [A540]
1090 DATA &A0,&23,&11,&26,&A0,&01,&05,&0 [0B7E]
0 [0B7E]
1100 DATA &ED,&B0,&DD,&6E,&00,&DD,&66,&0 [D766]
1 [D766]
1110 DATA &2B,&7E,&FE,&02,&C2,&B4,&A0,&2 [A13E]
3 [A13E]
1120 DATA &11,&2E,&A0,&01,&03,&00,&ED,&B [3CD6]
0 [3CD6]
1130 DATA &3A,&2A,&A0,&FE,&B1,&30,&0D,&2 [B312]
1 [B312]
1140 DATA &00,&00,&22,&2B,&A0,&3E,&00,&3 [5C94]
2 [5C94]
1150 DATA &2D,&A0,&37,&C9,&FE,&98,&D2,&B [9274]
4 [9274]
1160 DATA &A0,&21,&29,&A0,&CB,&FE,&3A,&2 [D25A]
A [D25A]
1170 DATA &A0,&FE,&97,&28,&0F,&21,&2A,&A [E534]
0 [E534]
1180 DATA &34,&2B,&CB,&3E,&2B,&CB,&1E,&2 [7386]
B [7386]
1190 DATA &CB,&1E,&18,&EA,&2A,&2B,&A0,&2 [074A]
2 [074A]
1200 DATA &2B,&A0,&3A,&27,&A0,&CB,&3F,&3 [852E]
2 [852E]

```

```

1210 DATA &2D,&A0,&3A,&2E,&A0,&FE,&00,&C [3568]
A [3568]
1220 DATA &B4,&A0,&37,&C9,&37,&3F,&C9,&2 [6344]
A [6344]
1230 DATA &2B,&A0,&23,&22,&2B,&A0,&11,&2 [33C8]
9 [33C8]
1240 DATA &00,&2A,&7D,&BE,&19,&3A,&2B,&A [9F3A]
0 [9F3A]
1250 DATA &77,&23,&3A,&2C,&A0,&77,&C9,&C [5D48]
D [5D48]
1260 DATA &37,&A0,&D0,&CD,&DC,&A0,&CD,&F [4EB4]
8 [4EB4]
1270 DATA &A0,&ED,&B0,&C9,&CD,&BE,&A0,&2 [F6A4]
1 [F6A4]
1280 DATA &80,&A1,&CD,&5A,&A1,&CD,&B7,&A [7986]
0 [7986]
1290 DATA &21,&00,&A2,&CD,&5A,&A1,&CD,&B [685E]
7 [685E]
1300 DATA &A0,&21,&80,&A2,&CD,&5A,&A1,&C [0E38]
9 [0E38]
1310 DATA &3A,&2D,&A0,&5F,&16,&00,&21,&8 [53D6]
0 [53D6]
1320 DATA &A1,&19,&ED,&5B,&2F,&A0,&3A,&2 [5868]
E [5868]
1330 DATA &A0,&4F,&06,&00,&C9,&CD,&37,&A [A328]
0 [A328]
1340 DATA &D0,&CD,&BE,&A0,&21,&80,&A1,&C [3E7A]
D [3E7A]
1350 DATA &5A,&A1,&CD,&F8,&A0,&EB,&ED,&B [ECCE]
0 [ECCE]
1360 DATA &CD,&BE,&A0,&21,&80,&A1,&CD,&6 [D264]
1 [D264]
1370 DATA &A1,&CD,&B7,&A0,&21,&00,&A2,&C [9952]
D [9952]
1380 DATA &5A,&A1,&CD,&F8,&A0,&EB,&ED,&B [10D4]
0 [10D4]
1390 DATA &CD,&BE,&A0,&21,&00,&A2,&CD,&6 [365C]
1 [365C]
1400 DATA &A1,&CD,&B7,&A0,&21,&80,&A2,&C [8456]
D [8456]
1410 DATA &5A,&A1,&CD,&F8,&A0,&EB,&ED,&B [ABC8]
0 [ABC8]
1420 DATA &CD,&BE,&A0,&21,&80,&A2,&CD,&6 [E560]
1 [E560]
1430 DATA &A1,&C9,&DF,&5E,&A1,&C9,&92,&D [4590]
3 [4590]
1440 DATA &07,&DF,&65,&A1,&C9,&6B,&A1,&0 [4430]
7 [4430]
1450 DATA &E5,&D5,&C5,&E5,&11,&00,&00,&C [D03C]
D [D03C]
1460 DATA &98,&CA,&CD,&10,&D4,&D2,&A9,&D [808A]
3 [808A]
1470 DATA &EB,&E3,&CD,&F3,&D9,&C3,&A6,&D [AAD4]
3 [AAD4]

```

Listing 1. Mit dem Basic-Lader erzeugen Sie das Binärfeld. Für andere Programme müssen Sie dieses mit »SAVE 'ERWBIN',b,&A000,&18180« speichern

```

10 REM Anlegen einer relativen Datei [C9C6]
20 OPENOUT "dummy":MEMORY HIMEM-1:CLOSED [7ECA]
UT [0EB6]
30 OPENOUT "reldatei" [2CB0]
40 FOR i=1 TO 300 [7CFA]
50 PRINT#9, RIGHT$(" {2 SPACE}" +STR$(i),3 [CS8A]
)+". Datensatz" [4C9E]
60 NEXT [13EC]
70 PRINT#9,SPACE$(128);SPACE$(128)
80 CLOSEOUT

```

Listing 2. Die relative Datei wird mit einer sequenziellen erzeugt

```

10 REM Relative Datei lesen [22FE]
20 OPENIN "reldatei" [7EF2]
30 FOR i= 300 TO 1 STEP -1 [16A2]
40 a$=SPACE$(14) [4318]
50 z=(i-1)*16 [9DCA]
60 !RECREAD,@z,@a$ [889E]
70 PRINT a$ [8372]
80 NEXT i [81A0]
90 CLOSEIN [F02C]

```

Listing 3. Lesen einer relativen Datei

schieht mit »SAVE »ERWBIN«,B&A 000,&17F«. Sie können dann in Ihr Programm eine Ladeanweisung für das des Maschinencode-Programms einsetzen. Zuvor muß jedoch die Speicherobergrenze auf 9FFFhex herabgesetzt werden.

In Listing 2 finden Sie ein Beispiel, das eine relative Datei als sequentielle Datei anlegt. Zeile 20 reserviert einen Ein-/Ausgabepuffer, worauf Zeile 30 die sequentielle Datei »reldatei« öffnet. Anschließend werden in einer FOR...NEXT-Schleife

300 Datensätze mit folgendem Inhalt auf Diskette geschrieben:

1. Datensatz
2. Datensatz

·
·
·

300. Datensatz

Jeder Datensatz ist 14 Zeichen lang. Beim sequentiellen Schreiben wird automatisch ein CR und LF angefügt, so daß wir beim relativen Lesen und Schreiben von einer Datensatzlänge mit 16 Zeichen ausgehen

müssen. Zeile 70 schreibt schließlich sicherheitshalber noch zwei Leerstrings mit je 128 Bytes auf Diskette.

Nun wollen wir die Datei per Direktzugriff lesen. Wir beginnen mit dem 300. Datensatz und hören mit dem 1. auf (Listing 3). Zeile 20 öffnet wieder die Datei »reldatei«, diesmal aber zum relativen Lesen. Zeile 40 definiert den String a\$, in dem die Datensätze abgelegt werden sollen. Die absolute Position z des Datensatzes in der Datei errechnet sich in

Grundlagen

Zeile 50. Zeile 60 liest den Datensatz aufgrund der Angaben in z und a\$.

Als letztes Beispiel noch ein Programm, das es Ihnen erlaubt, einen bestimmten Datensatz neu zu schreiben. Achten Sie darauf, die relative Datei auch hier mit OPENIN zum Schreiben zu öffnen (Listing 4).

Das Programm fragt, welcher Datensatz neu geschrieben werden soll. Es folgt eine Abfrage auf Gültigkeit der Datensatznummer. Dann wird der betreffende Satz zunächst gelesen, um den Inhalt zu kennen. Geben Sie ihn daraufhin neu ein. Zeile 100 format ihn in die richtige Länge von 14 Zeichen um, wonach er in Zeile 110 auf Diskette abgelegt wird.

Als ausführliches Beispiel für eine relative Dateiverwaltung soll das komplette Adressenverwaltungsprogramm (Listing 5) dienen. Es ist modular aufgebaut und menügesteuert. In leicht abgeänderter Form kann es auch für jede andere Art von Dateiverwaltung eingesetzt werden.

Das Programm benutzt eine indexsequentielle Dateiverwaltung. Indexsequentiell bedeutet, daß die Nummer und ein spezieller Index von jedem Datensatz zusätzlich in einer sequentiellen Datei abgelegt werden. Diese werden dann zusammen mit der relativen Datei auf Diskette geschrieben. Dieser Index dient als Suchkriterium zum Auffinden eines Datensatzes.

Für unser Adressenverwaltungsprogramm wurde der Nachname als Index gewählt. Bevor nun Datensätze angelegt oder bearbeitet werden können, lädt man die Indexdatei in den Arbeitsspeicher des Computers. Sie bleibt dort so lange, wie wir mit der relativen Datei arbeiten und wird im Anschluß daran wieder auf Diskette zurück geschrieben.

Wenn wir nun einen Datensatz anlegen, »weiß« die Indexdatei bereits,

welche Einträge belegt und welche noch frei sind. Sie sucht den nächstverfügbaren Eintrag heraus und legt dort den Datensatz ab. Gleichzeitig erhält auch die Indexdatei einen neuen Eintrag, der den Nachnamen als Index und die dazugehörige Datensatznummer enthält.

Suchen wir nun umgekehrt einen Datensatz (beispielsweise zum Namen Müller), so wird zunächst die Indexdatei durchsucht, um festzustellen, ob der Satz überhaupt in der Datei vorhanden ist. Ist dies der Fall, so steht auch gleichzeitig die Nummer des fortlaufenden Eintrags zur Verfügung. Dieser kann dann leicht aufgerufen werden.

Werfen wir einen Blick auf das Menü, das Sie nach dem Starten des Programms auf dem Bildschirm finden (Bild 1).

Es ist bereits bekannt, daß wir, um Eintragungen vornehmen zu können, eine relative Datei anlegen müssen. Genau dies geschieht durch den Menüpunkt 5. Zusätzlich wird hier noch die Indexdatei erzeugt und auf Diskette geschrieben. Wenn Sie sich nun das Disketten-Inhaltsverzeichnis ansehen, enthält es eine relative und eine sequentielle Datei. Beide sind neu angelegt und haben noch keine Einträge. Damit es keine Verwechslungen mit den Dateinamen gibt, erhält die relative Datei einen frei wählbaren Namen und die sequentielle Indexdatei den gleichen Namen mit der Extension »IND«. Heißt eine Datei »KUNDEN«, so erhält die relative Datei den Namen »KUNDEN.«, die Indexdatei die Bezeichnung »KUNDEN.IND«.

Bevor Sie nun Datensätze anlegen oder bearbeiten können, müssen Sie auf jeden Fall Menüpunkt 4 auswählen, der die relative Datei öffnet. Öffnen ist hier nicht nur unbedingt im Sinne der OPENIN- oder OPENOUT-Anweisung zu verstehen, sondern bedeutet vielmehr eine Initiali-

sierung der Adressendatei. Dabei wird die Indexdatei in den Speicher geladen und gleichzeitig festgestellt, welche Einträge belegt und welche noch frei sind.

Nach dem Öffnen können Sie Menüpunkt 1 anwählen, worauf Sie aufgefordert werden, verschiedene Daten einzugeben (Bild 2).

Direkt unter der Überschrift befindet sich eine Zahl, die angibt, der wievielte Datensatz gerade bearbeitet wird. In diesem Fall ist es der fünfzehnte. Bevor Sie nun die eigentlichen Daten eingeben, erscheint unter der fortlaufenden Nummer zunächst die Aufforderung, den Code für die Anrede einzugeben, während der Rest des Bildschirms noch leer bleibt. Bei der Anrede können Sie unter sechs Möglichkeiten auswählen:

Code	Anrede
0	keine
1	Herrn
2	Frau
3	Fräulein
4	Herrn und Frau
5	Firma

Diese Codes sollten Sie sich vorher merken oder notieren, da sie einfacher einzugeben sind, als immer wieder die gesamte Anrede. Nach dem Drücken der Enter-Taste erscheint als nächstes die Aufforderung zur Eingabe des (Nach-)Namens. Sie geben ihn ein und drücken wiederum die Enter-Taste. Das gleiche wiederholt sich dann für die Straße und den Ort mit Postleitzahl. Ist ein Datensatz komplett, erscheint unten im Bildschirm der Hinweis, entweder die Enter- oder die Leertaste zu drücken. Letztere führt Sie wieder zurück ins Menü, die Enter-Taste gleich zum nächsten Datensatz. Der Vorgang wiederholt sich dann von vorne, allerdings mit einer um eins erhöhten Datensatznummer.

```

10 REM Datensatz schreiben          [CC0C]
20 OPENIN "reldatei"                [7EF2]
30 INPUT "Datensatz-Nr.":i          [79C8]
40 IF i<1 OR i>300 THEN 30         [61EC]
50 a$=SPACE$(14)                    [401A]
60 z=(i-1)*16                       [A1CC]
70 !RECREAD,@z,@a$                 [7FA0]
80 PRINT a$                          [9F74]
90 INPUT "Neuer Datensatz":a$       [BA14]
100 a$=LEFT$(a$+SPACE$(14),14)      [8434]
110 !REWRITE,@z,@a$                [69D4]
120 CLOSEIN                          [2180]

```

Listing 4. Schreiben einer relativen Datei

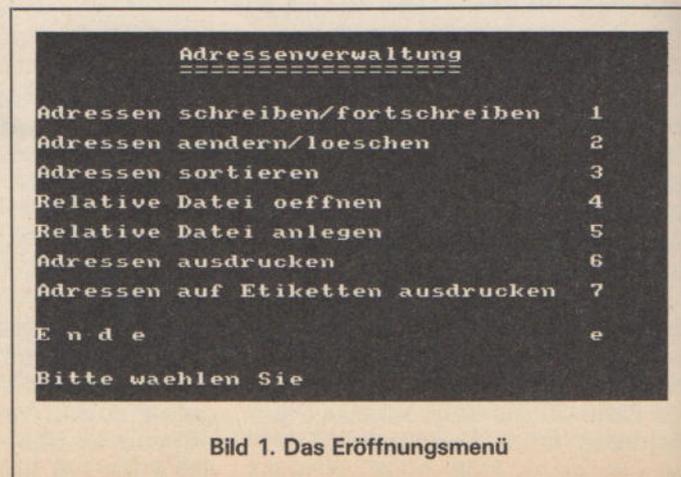


Bild 1. Das Eröffnungsmenü

Achten Sie aber darauf, daß die einzelnen Eingabewerte eine bestimmte Anzahl von Zeichen nicht überschreiten dürfen:

Name	15 Zeichen
Vorname	15 Zeichen
Straße	20 Zeichen
PLZ/Ort	21 Zeichen

Kommen wir zum zweiten Menüpunkt, der zum Ändern und Löschen von Adressen dient. Sie werden zunächst aufgefordert, den betreffenden Namen als Suchindex einzugeben. Daraufhin schaut das Programm nach, ob dieser Name in der Datei vorhanden ist. Wenn nein, erscheint ein Hinweis »Nicht gefunden« und Sie können den nächsten Namen suchen oder ins Menü zurückkehren.

Wurde der Name dagegen gefunden, wird der komplette Datensatz auf dem Bildschirm ausgegeben (ähnlich wie unter Menüpunkt 1, wobei der Cursor eine Zeile unter dem Anredecode steht). Wünschen Sie keine Änderung vorzunehmen, drücken Sie einfach die Enter-Taste. Anderenfalls schreiben Sie den neuen Datenwert unmittelbar unter den alten und drücken ebenfalls die Enter-Taste. Nach Beendigung des Korrekturvorgangs können Sie wie in Punkt 1 entweder fortfahren (En-

ter-Taste) oder ins Menü zurückkehren.

Soll ein Datensatz gelöscht werden, wählen Sie ebenfalls Menüpunkt 2 und überschreiben den Anredecode mit dem Buchstaben l. Dann drücken Sie die Enter-Taste, worauf der betreffende Datensatz aus der Indexdatei entfernt wird. In der relativen Datei bleibt er jedoch zunächst erhalten; er ist aber für einen neuen Eintrag freigegeben und kann mit einer neuen Adresse überschrieben werden.

Menüpunkt 3 sortiert die Datensätze in alphabetischer Reihenfolge nach dem Shell-Sortierverfahren. Sortiert wird lediglich die Indexdatei. In der relativen Datei ändert sich nichts.

Die letzten beiden Menüpunkte dienen zum Ausdrucken der Datei. Punkt 6 liefert einen normalen Listenausdruck, während Sie mit Punkt 7 Etiketten beschriften können. Dabei wird davon ausgegangen, daß die Etiketten genau 9 Druckzeilen auseinanderliegen. Jedoch kann das Programm auch für andere Etikettengrößen leicht angepaßt werden.

Bei Abschluß der Bearbeitung ist unbedingt die E-Taste zu drücken, damit die Indexdatei wieder auf Diskette geschrieben wird. Wird dies unterlassen, können beim Wieder-

einlesen schlimme Folgen auftreten; besonders dann, wenn Datensätze gelöscht oder geändert wurden.

Hier noch ein paar Anmerkungen zum Listing. Zunächst wird das Maschinencode-Programm »ERW BIN« geladen und initialisiert (siehe oben), falls es sich noch nicht im Speicher befindet. Anschließend wird die maximale Anzahl der Datensätze auf 200 festgelegt (sie kann jedoch gegebenenfalls geändert werden). Es folgt die Definition der einzelnen Stringlängen.

Das Anlegen der relativen Datei erfolgt wiederum sequentiell. Gleichzeitig wird auch die Indexdatei erzeugt und auf Diskette geschrieben. Abgelegt im Feld n\$(i), in das sie auch später zur Bearbeitung der Datei eingelassen wird, verbleibt sie während der gesamten Bearbeitungsdauer im Speicher. Da die Indexdatei noch leer ist, wird als erster Wert die Zahl Null hineingeschrieben, gefolgt von 200 Elementen, die aus 15 Sternchen und der fortlaufenden Datensatznummer bestehen.

```
0
***** 1
***** 2
***** 3
und so weiter.
```

Die Sternchen kennzeichnen einen leeren oder als gelöscht gekennzeichneten Eintrag. Wird der Eintrag später belegt, erscheint oben statt der Null die Anzahl der belegten Einträge und statt der Sternchen der betreffende Name als Suchindex.

Beim »Öffnen« der relativen Datei wird lediglich die Indexdatei eingelese. Vorausgesetzt wird, daß, wenn sich eine Indexdatei auf der Diskette befindet, auch die zugehörige relative Datei vorhanden sein muß. Wird die Indexdatei nicht gefunden, erscheint die Amsdos-Fehlermeldung »(Filename) not found«.

(Jürgen Hückstädt)

Adressen schreiben/fortschreiben

```
1
Anrede (0-5)? 1
Name      ? Huber
Vorname   ? Karl
Strasse   ? Talstr. 12
PLZ, Ort  ? 6000 Frankfurt 40

Naechste Eingabe - Enter-Taste druecken
Zurueck ins Menue - Leertaste druecken
```

Bild 2.
Die Eingabemaske

```
1000 REM Adressenverwaltungsprogramm (re
      lativ) [0B52]
1010 REM===== [33AB]
1020 REM fuer CPC 464, 664 und 6128 [DA22]
1030 REM ----- [BCF4]
1040 REM (c) Juergen Hueckstaedt [2844]
1050 REM Richard-Wagner-Str. 29 [94AE]
1060 REM 7310 Plochingen [46BA]
1070 IF PEEK(&A000)=1 AND PEEK(&A001)=10
      THEN 1120 [E788]
1080 MEMORY &9FFF [7426]
1090 LOAD "erwbin",&A000 [A130]
1100 CALL &A000 [9F2A]
1110 OPENDOUT "dummy":MEMORY HIMEM-1:CL0
      SEOUT [7BCC]
1120 ma=200:REM max.Anzahl der Datensae
      tze [8508]
1130 MODE 1 [06B6]
1140 k=0:REM Datensaeetze im Speicher [007A]
1150 ln=15:REM Zeichen Laenge Namen [A0C6]
1160 lv=15:REM Zeichen Laenge Vornamen [CC86]
1170 ls=20:REM Zeichen Laenge Strasse [DAB8]
```

```
1180 lo=21:REM Zeichen Laenge PLZ/Ort [339E]
1190 la=1:REM Zeichen Laenge Anredecode [D940]
1200 fl=0:REM Flag ob Datei im Speicher [A736]
1210 r1=74:REM Recordlaenge+2 (cr/lf) [C52E]
1220 cl$=CHR$(8):REM Cursor nach links [A2AC]
1230 cr$=CHR$(9):REM Cursor nach rechts [7F8C]
1240 cd$=CHR$(10):REM Cursor nach unten [DE04]
1250 cu$=CHR$(11):REM Cursor nach oben [2E1E]
1260 l=ln+lv+ls+lo+la+1 [68BA]
1270 g$="*****" [0174]
1280 n$="(36 SPACE)" [C7FC]
1290 DIM n$(ma) [752E]
1300 DIM ar$(5) [18B6]
1310 ar$(0)=" [F4BC]
1320 ar$(1)="Herrn" [99BE]
1330 ar$(2)="Frau" [44E0]
1340 ar$(3)="Fraeulein" [C5FE]
1350 ar$(4)="Herrn und Frau" [A1F4]
1360 ar$(5)="Firma" [CCEA]
1370 : [CE4A]
1380 : [D14C]
1390 : Listing 5. [D44E]
      Eine Adreßverwaltung mit direktem Datenzugriff
```

Grundlagen

1400	REM Menue	[83C6]	2220	REM Adressen aendern/loeschen	[8738]
1410	CLS	[1890]	2230	CLS	[1892]
1420	PRINT TAB(10);"Adressenverwaltung"	[A140]	2240	PRINT	[BBEA]
1430	PRINT TAB(10);"=====	[13CE]	2250	PRINT	[9FEC]
1440	PRINT	[9BEC]	2260	PRINT "{7 SPACE}Adressen aendern/lo	
1450	PRINT	[B4EE]		eschen"	[71DA]
1460	PRINT "Adressen schreiben/fortschre		2270	PRINT	[A5F0]
	iben{3 SPACE}1"	[E924]	2280	PRINT	[0CF2]
1470	PRINT	[AAF2]	2290	IF dn\$="" THEN PRINT "Bitte Datei o	
1480	PRINT "Adressen aendern/loeschen{10			effnen":GOTO 2940	[7CAA]
	SPACE}2"	[DB04]	2300	PRINT "Name{8 SPACE}";:INPUT na\$	[76DE]
1490	PRINT	[08F6]	2310	na\$=LEFT\$(na\$+n\$,ln)	[16DB]
1500	PRINT "Adressen sortieren{17 SPACE}		2320	i=1	[FB7C]
	3"	[18B4]	2330	WHILE i<=k	[119C]
1510	PRINT	[BCE8]	2340	IF LEFT\$(n\$(i),ln)=na\$ THEN 2400	[6CDE]
1520	PRINT "Relative Datei oeffnen{13 SP		2350	i=i+1	[D9AA]
	ACE}4"	[BAE2]	2360	WEND	[0532]
1530	PRINT	[96EC]	2370	PRINT	[0DF2]
1540	PRINT "Relative Datei anlegen{13 SP		2380	PRINT "nicht gefunden"	[2FC0]
	ACE}5"	[92DA]	2390	GOTO 2910	[6E26]
1550	PRINT	[A0F0]	2400	d=VAL(MID\$(n\$(i),ln+1,10))	[588A]
1560	PRINT "Adressen ausdrucken{16 SPACE		2410	r\$=SPACE\$(r1-2)	[00B0]
	>6"	[B83A]	2420	po=r1*d	[3AE0]
1570	PRINT	[9AF4]	2430	OPENIN dn\$	[A890]
1580	PRINT "Adressen auf Etiketten ausdr		2440	!RECREAD,@po,@r\$	[CC52]
	ucken{2 SPACE}7"	[5612]	2450	CLOSEIN	[4FF0]
1590	PRINT	[74F8]	2460	na\$=LEFT\$(r\$,ln)	[DB80]
1600	PRINT	[DBE8]	2470	nv\$=MID\$(r\$,ln+1,lv)	[F80E]
1610	PRINT "E n d e{28 SPACE}e"	[2834]	2480	ns\$=MID\$(r\$,ln+lv+1,ls)	[A71E]
1620	PRINT	[D5EC]	2490	no\$=MID\$(r\$,ln+lv+ls+1,lo)	[D524]
1630	PRINT	[CEEE]	2500	a\$=RIGHT\$(r\$,1)	[821E]
1640	PRINT "Bitte waehlen Sie"	[A832]	2510	PRINT i	[DDFC]
1650	m\$=INKEY\$:IF m\$="" THEN 1650	[7C28]	2520	PRINT	[A0EC]
1660	IF m\$="e" THEN 4850	[A6A6]	2530	PRINT "Anrede (0-5){2 SPACE}";a\$	[725A]
1670	m=VAL(m\$)	[28BA]	2540	PRINT	[96F0]
1680	IF m<1 OR m>7 THEN 1650	[8FEC]	2550	PRINT "Name{10 SPACE}";na\$	[EF98]
1690	ON m GOSUB 1740,2220,3040,3340,363		2560	PRINT	[BCF4]
	0,3990,4360	[D426]	2570	PRINT "Vorname{7 SPACE}";nv\$	[C6B4]
1700	GOTO 1400	[9A0C]	2580	PRINT	[D2F8]
1710	:	[C446]	2590	PRINT "Strasse{7 SPACE}";ns\$	[C1CC]
1720	:	[BF48]	2600	PRINT	[B4EA]
1730	:	[D24A]	2610	PRINT "PLZ, Ort{6 SPACE}";no\$	[169A]
1740	REM Adressen schreiben/fortschreibe		2620	FOR q=1 TO 8:PRINT cu\$;:NEXT q	[1426]
	n	[C52A]	2630	FOR q=1 TO 12:PRINT cr\$;:NEXT q:INP	
1750	CLS	[229E]		UT ww\$	[0670]
1760	PRINT	[ABF6]	2640	IF ww\$<>"" THEN a\$=ww\$	[D31C]
1770	PRINT	[A2F8]	2650	a=VAL(a\$)	[8D88]
1780	PRINT "{4 SPACE}Adressen schreiben/		2660	IF a\$="1" THEN GOSUB 4700:GOTO 2910	
	fortschreiben{4 SPACE}"	[8A0C]			[8290]
1790	PRINT	[D0FC]	2670	IF a<0 OR a>5 OR LEN(a\$)<>1 THEN PR	
1800	PRINT	[D7EC]		INT cu\$;:GOTO 2630	[FE24]
1810	IF dn\$="" THEN PRINT "Bitte Datei o		2680	PRINT	[CCFA]
	effnen":GOTO 2120	[BE90]	2690	FOR q=1 TO 12:PRINT cr\$;:NEXT q:INP	
1820	k=k+1	[CFB4]		UT ww\$	[EC7C]
1830	PRINT k	[F908]	2700	IF ww\$<>"" THEN na\$=ww\$	[B432]
1840	PRINT	[D3F4]	2710	PRINT	[C7EE]
1850	IF k>ma THEN PRINT "Datei voll":GOT		2720	na\$=LEFT\$(na\$+n\$,ln)	[A6E2]
	O 2090	[FA2C]	2730	FOR q=1 TO 12:PRINT cr\$;:NEXT q:INP	
1860	PRINT "Anrede (0-5)";:INPUT a\$	[45B8]		UT ww\$	[9472]
1870	a=VAL(a\$)	[B08E]	2740	IF ww\$<>"" THEN nv\$=ww\$	[6F64]
1880	IF a<0 OR a>5 OR LEN(a\$)<>1 THEN PR		2750	PRINT	[ABF6]
	INT cu\$;:GOTO 1860	[9830]	2760	nv\$=LEFT\$(nv\$+n\$,lv)	[9D4E]
1890	PRINT	[B8FE]	2770	FOR q=1 TO 12:PRINT cr\$;:NEXT q:INP	
1900	PRINT "Name{8 SPACE}";:INPUT na\$	[1DE8]		UT ww\$	[507A]
1910	PRINT	[DCF0]	2780	IF ww\$<>"" THEN ns\$=ww\$	[EE66]
1920	na\$=LEFT\$(na\$+n\$,ln)	[D0E4]	2790	PRINT	[BFEE]
1930	PRINT "Vorname{5 SPACE}";:INPUT nv\$		2800	ns\$=LEFT\$(ns\$+n\$,ls)	[1532]
		[B506]	2810	FOR q=1 TO 12:PRINT cr\$;:NEXT q:INP	
1940	PRINT	[C1F6]		UT ww\$	[CC70]
1950	nv\$=LEFT\$(nv\$+n\$,lv)	[EB4E]	2820	IF ww\$<>"" THEN no\$=ww\$	[DE54]
1960	PRINT "Strasse{5 SPACE}";:INPUT ns\$		2830	no\$=LEFT\$(no\$+n\$,lo)	[8320]
		[AC20]	2840	r\$=na\$+nv\$+ns\$+no\$+RIGHT\$(STR\$(a),1	
1970	PRINT	[D2FC]) +CHR\$(13)+CHR\$(10)	[F498]
1980	ns\$=LEFT\$(ns\$+n\$,ls)	[F642]	2850	d=VAL(MID\$(n\$(i),ln+1,10))	[B79C]
1990	PRINT "PLZ, Ort{4 SPACE}";:INPUT no		2860	OPENIN dn\$	[CC9E]
	\$	[6902]	2870	po=r1*d	[BDF2]
2000	no\$=LEFT\$(no\$+n\$,lo)	[A90A]	2880	!REWRITE,@po,@r\$	[5340]
2010	r\$=na\$+nv\$+ns\$+no\$+RIGHT\$(STR\$(a),1		2890	CLOSEIN	[5C00]
) +CHR\$(13)+CHR\$(10)	[4E82]	2900	n\$(i)=na\$+STR\$(d)	[46C8]
2020	d=VAL(MID\$(n\$(k),ln+1,10))	[CDBA]	2910	PRINT	[13F2]
2030	po=r1*d	[C1DA]	2920	PRINT	[80F4]
2040	OPENIN dn\$	[A88A]	2930	PRINT "Naechste Eingabe - Enter-Tas	
2050	!REWRITE,@po,@r\$	[302A]		te druecken"	[47BE]
2060	CLOSEIN	[58EA]	2940	PRINT	[DEF8]
2070	f1=1	[5C52]	2950	PRINT "Zurueck ins Menue - Leertast	
2080	n\$(k)=na\$+STR\$(d)	[62CA]		e druecken	[2566]
2090	PRINT	[77F0]	2960	m\$=INKEY\$:IF m\$="" THEN 2960	[D33C]
2100	PRINT	[9AE0]	2970	IF m\$=CHR\$(13) AND dn\$<>"" THEN 222	
2110	PRINT "Naechste Eingabe - Enter-Tas			0	[BF42]
	te druecken"	[D9AA]	2980	IF m\$="" THEN 3000	[4ECC]
2120	PRINT	[C0E4]	2990	GOTO 2960	[933C]
2130	PRINT "Zurueck ins Menue - Leertast		3000	RETURN	[8986]
	e druecken	[1352]	3010	:	[BE3C]
2140	m\$=INKEY\$:IF m\$="" THEN 2140	[F114]	3020	:	[C33E]
2150	IF m\$=CHR\$(13) AND dn\$<>"" THEN 174		3030	:	[C040]
	0	[BA3A]	3040	REM Adressen sortieren	[8336]
2160	IF m\$="" THEN CLOSEIN:GOTO 2180	[B748]	3050	CLS	[0694]
2170	GOTO 2140	[5B14]	3060	PRINT	[9BEC]
2180	RETURN	[C096]	3070	PRINT	[B4EE]
2190	:	[0B4C]	3080	PRINT "{10 SPACE}Adressen sortieren	
2200	:	[C43C]		{7 SPACE}"	[9258]
2210	:	[C73E]	3090	PRINT	[BEF2]

3100 PRINT	[B3E2]	4020 PRINT	[CCE6]
3110 g=INT(k/2)	[1122]	4030 PRINT "{7 SPACE}Adressen ausdrucken	[A444]
3120 WHILE g>0	[5FA8]	PRINT "{11 SPACE}"	[BAEA]
3130 FOR i=g+1 TO k	[6D4A]	4040 PRINT	[A1EC]
3140 j=i-g	[6618]	4050 PRINT	[AC92]
3150 WHILE j>0	[8CB4]	4060 OPENIN dn\$	[952E]
3160 IF n\$(j)<n\$(j+g) THEN j=0:GOTO 321		4070 FOR i=1 TO k	[4796]
0	[64BA]	d=VAL(MID\$(n\$(i),ln+1,10))	[5DEA]
3170 a\$=n\$(j+g)	[7118]	4090 po=r1*d	[B1AC]
3180 n\$(j+g)=n\$(j)	[81AA]	4100 r\$=SPACE\$(r1-2)	[E54A]
3190 n\$(j)=a\$	[29F8]	4110 !RECREAD,@po,@r\$	[13E4]
3200 j=j-g	[1614]	4120 PRINT#8,LEFT\$(r\$,ln);	[3836]
3210 WEND	[1828]	4130 PRINT#8," ";	[D11A]
3220 NEXT	[D94C]	4140 PRINT#8,MID\$(r\$,ln+1,lv);	[403A]
3230 g=INT(g/2)	[0C20]	4150 PRINT#8," ";	[7232]
3240 WEND	[2E2E]	4160 PRINT#8,MID\$(r\$,ln+lv+1,ls);	[303E]
3250 PRINT	[CEEE]	4170 PRINT#8," ";	[C442]
3260 PRINT	[BBF0]	4180 PRINT#8,MID\$(r\$,ln+lv+ls+1,lo);	[1842]
3270 PRINT "{10 SPACE}Zurueck ins Menue"	[2B14]	4190 PRINT#8," ";	[DDC0]
		4200 PRINT#8,RIGHT\$(r\$,1)	[45C4]
3280 PRINT "{7 SPACE}beliebige Taste dru	[2DD0]	4210 IF i-INT(i/64)*64<>0 THEN 4250	[6DC4]
ecken"	[38F0]	4220 FOR j=1 TO 8	[7CA2]
3290 m\$=INKEY\$:IF m\$="" THEN 3290	[8C8C]	4230 PRINT#8	[A566]
3300 RETURN	[BA42]	4240 NEXT j	[9566]
3310 :	[C944]	4250 NEXT i	[5AF2]
3320 :	[CC46]	4260 CLOSEIN	[9FF4]
3330 :	[A464]	4270 PRINT	[C2F6]
3340 REM Relative Datei oeffnen	[149A]	4280 PRINT	
3350 CLS	[CDF2]	4290 PRINT "{10 SPACE}Zurueck ins Menue"	[D31A]
3360 PRINT	[94F4]		
3370 PRINT		4300 PRINT "{7 SPACE}beliebige Taste dru	[78C4]
3380 PRINT "{6 SPACE}Relative Datei oeff	[C386]	ecken"	[3A18]
nen{7 SPACE}"	[D6F8]	4310 m\$=INKEY\$:IF m\$="" THEN 4310	[8E92]
3390 PRINT	[DDE8]	4320 RETURN	[C348]
3400 PRINT	[E62A]	4330 :	[D44A]
3410 INPUT "Dateiname";dn\$	[D7EC]	4340 :	[D14C]
3420 PRINT	[F990]	4350 :	[36B6]
3430 i=0	[4544]	4360 REM Adressen auf Etiketten drucken	[0BA0]
3440 OPENIN dn\$+".ind"	[AE56]	4370 CLS	[D4F8]
3450 f1=0	[F1E0]	4380 PRINT	[0DFA]
3460 INPUT#9,k	[B30C]	4390 PRINT	
3470 PRINT k	[DD4C]	4400 PRINT " Adressen auf Etiketten ausd	[9558]
3480 IF k>0 THEN f1=1		rucken"	[9DEC]
3490 IF i=ma THEN PRINT "Datei voll":GOT	[052A]	4410 PRINT	[ACEE]
0 3530	[95A6]	4420 PRINT	[8894]
3500 i=i+1	[4F9A]	4430 OPENIN dn\$	[9530]
3510 INPUT#9,n\$(i)	[7C3E]	4440 FOR i=1 TO k	[3F98]
3520 IF EOF=0 THEN 3490	[54F0]	4450 d=VAL(MID\$(n\$(i),ln+1,10))	[51B6]
3530 CLOSEIN	[B7F2]	4460 r\$=SPACE\$(r1-2)	[97EE]
3540 PRINT	[9EF4]	4470 po=r1*d	[105E]
3550 PRINT		4480 !RECREAD,@po,@r\$	[7EB2]
3560 PRINT "{10 SPACE}Zurueck ins Menue"	[B918]	4490 PRINT#8	[61A2]
		4500 PRINT#8	[6F44]
3570 PRINT "{7 SPACE}beliebige Taste dru	[B0D4]	4510 a=VAL(RIGHT\$(r\$,1))	[C950]
ecken"	[5638]	4520 PRINT#8,ar\$(a)	[FD92]
3580 m\$=INKEY\$:IF m\$="" THEN 3580	[C0A2]	4530 PRINT#8,LEFT\$(r\$,ln+lv)	[52AC]
3590 RETURN	[C546]	4540 PRINT#8,MID\$(r\$,ln+lv+1,ls)	[E2D0]
3600 :	[C248]	4550 PRINT#8	[6CB0]
3610 :	[D34A]	4560 PRINT#8,MID\$(r\$,ln+lv+ls+1,lo)	[C9B2]
3620 :	[555A]	4570 PRINT#8	[0A74]
3630 REM Relative Datei anlegen	[239E]	4580 PRINT#8	[64EE]
3640 CLS	[B6F6]	4590 NEXT i	[A1F0]
3650 PRINT	[A3F8]	4600 CLOSEIN	[B4F2]
3660 PRINT		4610 PRINT	
3670 PRINT "{7 SPACE}Relative Datei anle	[2C7C]	4620 PRINT	
gen{10 SPACE}"	[D1FC]	4630 PRINT "{10 SPACE}Zurueck ins Menue"	[6516]
3680 PRINT	[0AFE]		
3690 PRINT	[8F2E]	4640 PRINT "{7 SPACE}beliebige Taste dru	[10D2]
3700 INPUT "Dateiname";dn\$	[BEF0]	ecken"	[AAF4]
3710 PRINT	[E138]	4650 m\$=INKEY\$:IF m\$="" THEN 4650	[CBA0]
3720 r\$=SPACE\$(r1-2)+CHR\$(13)+CHR\$(10)	[3C5A]	4660 RETURN	[C356]
3730 OPENOUT dn\$	[83F8]	4670 :	[D258]
3740 FOR i=0 TO ma	[5178]	4680 :	[C55A]
3750 PRINT i;cu\$	[E2AC]	4690 :	[A41C]
3760 PRINT#9,r\$;	[8A72]	4700 REM Datensatz loeschen	
3770 NEXT i	[C2EA]	4710 z\$=LEFT\$(g\$,ln)+MID\$(n\$(i),ln+1,10)	[4CSA]
3780 PRINT#9,SPACE\$(128);SPACE\$(128);	[66C2]		[345C]
3790 CLOSEOUT	[2806]	4720 FOR j=i+1 TO k	[DB46]
3800 OPENOUT dn\$+".ind"	[0D62]	4730 n\$(j-1)=n\$(j)	[B070]
3810 PRINT#9,0	[0BF8]	4740 NEXT j	[2732]
3820 FOR i=1 TO ma	[0576]	4750 n\$(k)=z\$	[B2C4]
3830 PRINT i;cu\$	[1756]	4760 k=k-1	[2354]
3840 n\$(i)=LEFT\$(g\$,ln)+STR\$(i)	[92A2]	4770 FOR q=1 TO 8:PRINT cd\$;: NEXT q	[D844]
3850 PRINT#9,n\$(i)	[E172]	4780 PRINT "geloescht"	[741E]
3860 NEXT i	[69C0]	4790 PRINT cu\$;cu\$;	[A242]
3870 CLOSEOUT	[6666]	4800 IF k=0 THEN f1=0	[CB9A]
3880 f1=1	[F398]	4810 RETURN	[D450]
3890 k=0	[D3F2]	4820 :	[E752]
3900 PRINT	[DAF4]	4830 :	[E254]
3910 PRINT		4840 :	[0A56]
3920 PRINT "{10 SPACE}Zurueck ins Menue"	[9518]	4850 REM Programm beenden	[343E]
		4860 IF f1=0 THEN 4930	[4E16]
3930 PRINT "{7 SPACE}beliebige Taste dru	[6CD4]	4870 OPENOUT dn\$+".ind"	[03E8]
ecken"	[4E38]	4880 PRINT#9,k	[AA08]
3940 m\$=INKEY\$:IF m\$="" THEN 3940	[C4A2]	4890 FOR i=1 TO ma	[349C]
3950 RETURN	[CD58]	4900 PRINT#9,n\$(i)	[465A]
3960 :	[E05A]	4910 NEXT	[5CBA]
3970 :	[D35C]	4920 CLOSEOUT	[5F8E]
3980 :	[6614]	4930 END	
3990 REM Adressdatei ausdrucken	[1E8C]		
4000 CLS	[9DE4]		
4010 PRINT			
		4020 PRINT	
		4030 PRINT "{7 SPACE}Adressen ausdrucken	
		PRINT "{11 SPACE}"	
		4040 PRINT	
		4050 PRINT	
		4060 OPENIN dn\$	
		4070 FOR i=1 TO k	
		d=VAL(MID\$(n\$(i),ln+1,10))	
		4090 po=r1*d	
		4100 r\$=SPACE\$(r1-2)	
		4110 !RECREAD,@po,@r\$	
		4120 PRINT#8,LEFT\$(r\$,ln);	
		4130 PRINT#8," ";	
		4140 PRINT#8,MID\$(r\$,ln+1,lv);	
		4150 PRINT#8," ";	
		4160 PRINT#8,MID\$(r\$,ln+lv+1,ls);	
		4170 PRINT#8," ";	
		4180 PRINT#8,MID\$(r\$,ln+lv+ls+1,lo);	
		4190 PRINT#8," ";	
		4200 PRINT#8,RIGHT\$(r\$,1)	
		4210 IF i-INT(i/64)*64<>0 THEN 4250	
		4220 FOR j=1 TO 8	
		4230 PRINT#8	
		4240 NEXT j	
		4250 NEXT i	
		4260 CLOSEIN	
		4270 PRINT	
		4280 PRINT	
		4290 PRINT "{10 SPACE}Zurueck ins Menue"	
		4300 PRINT "{7 SPACE}beliebige Taste dru	
		ecken"	
		4310 m\$=INKEY\$:IF m\$="" THEN 4310	
		4320 RETURN	
		4330 :	
		4340 :	
		4350 :	
		4360 REM Adressen auf Etiketten drucken	
		4370 CLS	
		4380 PRINT	
		4390 PRINT	
		4400 PRINT " Adressen auf Etiketten ausd	
		rucken"	
		4410 PRINT	
		4420 PRINT	
		4430 OPENIN dn\$	
		4440 FOR i=1 TO k	
		4450 d=VAL(MID\$(n\$(i),ln+1,10))	
		4460 r\$=SPACE\$(r1-2)	
		4470 po=r1*d	
		4480 !RECREAD,@po,@r\$	
		4490 PRINT#8	
		4500 PRINT#8	
		4510 a=VAL(RIGHT\$(r\$,1))	
		4520 PRINT#8,ar\$(a)	
		4530 PRINT#8,LEFT\$(r\$,ln+lv)	
		4540 PRINT#8,MID\$(r\$,ln+lv+1,ls)	
		4550 PRINT#8	
		4560 PRINT#8,MID\$(r\$,ln+lv+ls+1,lo)	
		4570 PRINT#8	
		4580 PRINT#8	
		4590 NEXT i	
		4600 CLOSEIN	
		4610 PRINT	
		4620 PRINT	
		4630 PRINT "{10 SPACE}Zurueck ins Menue"	
		4640 PRINT "{7 SPACE}beliebige Taste dru	
		ecken"	
		4650 m\$=INKEY\$:IF m\$="" THEN 4650	
		4660 RETURN	
		4670 :	
		4680 :	
		4690 :	
		4700 REM Datensatz loeschen	
		4710 z\$=LEFT\$(g\$,ln)+MID\$(n\$(i),ln+1,10)	
		4720 FOR j=i+1 TO k	
		4730 n\$(j-1)=n\$(j)	
		4740 NEXT j	
		4750 n\$(k)=z\$	
		4760 k=k-1	
		4770 FOR q=1 TO 8:PRINT cd\$;: NEXT q	
		4780 PRINT "geloescht"	
		4790 PRINT cu\$;cu\$;	
		4800 IF k=0 THEN f1=0	
		4810 RETURN	
		4820 :	
		4830 :	
		4840 :	
		4850 REM Programm beenden	
		4860 IF f1=0 THEN 4930	
		4870 OPENOUT dn\$+".ind"	
		4880 PRINT#9,k	
		4890 FOR i=1 TO ma	
		4900 PRINT#9,n\$(i)	
		4910 NEXT	
		4920 CLOSEOUT	
		4930 END	

Listing 5.

Eine Adreßverwaltung mit direktem Datenzugriff (Schluß)

»Wanzen-Tod« mit DDT



Auf der CP/M-Diskette von Schneider finden Sie unter anderem einen leistungsfähigen Maschinensprache-Monitor. Was fehlt, ist eine Beschreibung im Handbuch.

Keine Sorge, wir empfehlen hier keineswegs den Einsatz des Insektenvertilgungsmittels DDT. Wir haben etwas besseres. Zu jeder Schneider-Diskettenstation mit dem CP/M-Betriebssystem gehören diverse Hilfsprogramme. Unter anderem auch das Dynamic Debugging Tool, kurz DDTCOM genannt. Mit diesem leistungsfähigen Maschinensprache-Monitor können Sie in Ihren Programmen auf die Suche nach Bugs (Wanzen beziehungsweise Programmierfehlern) gehen.

Leider konnte sich Schneider nicht dazu entschließen, die Dienstprogramme auf der CP/M-Diskette eingehend zu beschreiben. In der Betriebsanleitung wird kurz und bündig auf die »weiterführende Literatur« verwiesen.

Einer für (fast) alle

Die Aufgabe eines Standard-Betriebssystems läßt sich kurz etwa so beschreiben: Es soll für Programme die gleiche Umgebung auf einer Vielzahl verschiedener Computer schaffen. Dazu müssen gewisse Voraussetzungen in der Hardware gegeben sein. Am wichtigsten ist dabei die Verwendung gleichartiger Mikroprozessoren (CPUs).

Das Betriebssystem CP/M 2.2 (manchmal auch CP/M-80 genannt), läuft auf drei verschiedenen Prozessoren: dem Intel 8080, dem Intel 8085 und dem Zilog Z80, von dem der Typ Z80A in den Schneider-Computern eingebaut ist. Einsichtig mag noch sein, daß der 8085 kompatibel zum 8080 ist und damit für CP/M verwendet werden kann. Der Z80 hingegen besitzt eine Ausnahmestellung: er ist, obwohl von einer anderen Firma produziert, eine Weiterentwicklung der 8080/8085-Reihe. So verfügt er zusätzlich über relative Sprünge, die beiden Index-Register IX und IY, einen zweiten Re-

gistersatz, Makrobefehle wie LDI, LDIR, LDD und LDDR zur Blockverschiebung und andere Programmierer-freundliche Erweiterungen. Damit aber Programme auf allen CP/M-geeigneten 8-Bit-Prozessoren laufen, sollte sich der Anwender — auch wenn es schwer fällt — auf die auf allen CPUs vorhandenen Befehle beschränken. Wer dagegen nur Schneider-Programme schreiben will, kann den Befehlssatz des Z80 voll ausschöpfen.

Alle drei Mikroprozessoren codieren ihre Maschinenbefehle in gleicher Weise. So bedeutet der Code 60 hex (dezimal 96) auf allen drei CPUs, daß das H-Register mit dem Inhalt des B-Registers zu laden ist. Aus Urheberrechtsgründen konnte Zilog, der Z80-Hersteller, nicht die Standard-Bezeichnungen von Intel verwenden. So hat man sich völlig neue Mnemonics ausgedacht, die inzwischen bei den meisten Programmierern beliebter sind als die 8080-Originale. Das ist durchaus verständlich, da die Z80-Befehls-worte logischer aufgebaut und vor allem leichter zu erlernen sind. So entsprechen beispielsweise dem Z80-Allzweck-Ladebefehl LD in 8080-Notation eine Vielzahl von Mnemonics: LXI, STAX, MVI, LDAX, SHLD, LHLD, STA, LDA, MOV und SPHL.

Einige Beispiele, die die gegensätzliche »System-Philosophie« der Hardware-Designer beider Firmen zeigen:

8080-Assembler:	Z80-Assembler:
MOV BC	LD BC
MVI E,3	LD E,3
SHLD 2000	LD (2000),HL
SPHL	LD SP,HL

Welchen Schneider-Besitzer soll das interessieren, werden Sie fragen, der CPC hat einen Z80 und keinen 8080 oder 8085! Ganz einfach: alle CP/M-Dienstprogramme, die

auf der Systemdiskette mitgeliefert werden, bedienen sich der 8080-Mnemonics. Wollen Sie also unter CP/M programmieren oder ganz einfach das CP/M-Betriebssystem untersuchen, dann kommen Sie um die ungewohnten 8080-Befehle nicht herum!

In Tabelle 1 haben wir die 8080-Befehle und ihre Z80-Äquivalente in alphabetischer Reihenfolge zusammengestellt.

Der DDT-Debugger ist ein komfortabler Maschinensprache-Monitor, der zum Austesten und Verbessern von eigenen und fremden Maschinenprogrammen unter CP/M gedacht ist. So ist ein einfacher Assembler integriert, Maschinencodieroutinen können aufgerufen werden, wahlweise mit Breakpoints oder im Einzelschritt-Modus. Ferner kann man sich Speicherinhalte ausgeben lassen und ändern.

DDT: Monitor mit Komfort

Sobald Sie das CP/M-System durch Eingabe von »CPM« gestartet haben, erscheint die Bereitschaftsmeldung A>. Suchen Sie sich nun eine Diskette heraus, auf der sich das Programm DDTCOM befindet.

Der DDT-Monitor kann auf zwei Arten aufgerufen werden:

mit »A>DDT«
oder »A>DDT filename.ext«

Im ersten Fall wird einfach DDT geladen und gestartet. Bei Angabe eines Dateinamens lädt der Debugger dieses Programm dazu. Es kann dann am normalen TPA-Beginn &0100 gefunden werden. (TPA ist die Transient Program Area, der für CP/M-Programme frei verfügbare Arbeitsspeicher.) Der DDT verschiebt sich in diesem Fall automatisch an das obere Speicherende.

Der Debugger meldet sich schlicht mit der Meldung:

DDT VERS 2.2

NEXT PC

xxxx yyyy

Diese beiden Werte (xxxx und yyyy) sollte man sich aufschreiben, da sie beim späteren Speichern der Datei noch gebraucht werden. NEXT gibt an, ab welcher Adresse hinter dem geladenen Programm der Speicher wieder frei ist, PC zeigt den aktuellen Stand des Programmzählers des Debug-Programms an.

Der DDT verwendet ein eigenes Bereitschaftszeichen, den Bindestrich. Sobald dieser erscheint, können einbuchstabige Befehle einge-

tippt werden. Bei einem Teil der Befehle kann man auch Argumente angeben: als Hexadezimalziffern, voneinander abgetrennt durch Leerzeichen oder Kommas. Eine fehlerhafte Eingabe wird durch ein Fragezeichen beantwortet. Alle Eingaben können wahlweise als Klein- oder Großbuchstaben gegeben werden. Die Befehle im einzelnen:

Der Assemble-Befehl

»A« steht für »Assemble«. Wie der Name schon sagt, wird dabei ein einfacher 8080-Assembler aufgerufen. Der Assembler versteht nur die normalen Mnemonics und Hexadezimalzahlen, nicht hingegen Symbole, arithmetische Ausdrücke, Labels, Pseudo-Befehle wie DS (Define Space) oder DB (Define Byte) oder gar Kommandos zur bedingten Assemblierung. Aufgrund der eingeschränkten Möglichkeiten ist er hauptsächlich für kleinere Änderungen an bestehenden Programmen gedacht. Z80-Programmierer seien nochmal auf die Vergleichstabelle hingewiesen. Abgebrochen wird die Eingabe der Assembler-Mnemonics durch einen Punkt.

Der Dump-Befehl

»D« steht für »Dump«. Dieser Befehl gibt einen Hexadezimal- und ASCII-Dump eines Speicherbereichs auf dem Bildschirm aus. Nicht darstellbare Symbole des Zeichensatzes werden durch Punkte ersetzt. Der Dump-Befehl hat drei verschiedene Formate:

- D Dump von 12 Zeilen ab dem aktuellen PC-Stand
- D0200 Dump von 12 Zeilen ab der angegebenen Adresse
- D04,0A Dump zwischen zwei Adressen

Die Bildschirmausgabe kann dabei mit »Ctrl-S« eingefroren und durch Drücken einer beliebigen Taste fortgesetzt werden. Abgebrochen wird der Dump-Befehl durch Drücken einer beliebigen Taste außer »Ctrl-S«.

Der Fill-Befehl

»F« steht für »Fill«, einen Befehl zum Füllen eines Speicherbereichs mit einem bestimmten Wert. Das Format sieht so aus:

-F2F00,3F00,60

Dieser Befehl füllt den Speicher von 2F00 bis 3F00 hex mit dem Hexwert 60 hex.

Der Go-Befehl

»G« steht für »Go« und startet ein Maschinenprogramm an der angegebenen Adresse. So löst beispielsweise »G0000« einen CP/M-Warmstart aus. Zusätzlich können noch bis zu zwei Werte angegeben werden, bei denen der Programmablauf automa-

tisch unterbrochen wird und ein Rücksprung zu DDT erfolgt.

Der Hexcalc-Befehl

»H« steht für »Hexadecimal Calculation« und erlaubt die Subtraktion und Addition zweier Hexadezimalzahlen. Beispiel:

-H0100,0001

Dies ergibt 0101 hex als Additionsergebnis und 00FF hex für die Subtraktion.

Der Input-Befehl

»I« steht für »Input«. Das I-Kommando bereitet das Einlesen einer Diskettendatei vor. Direkt nach dem »I« muß dazu der Dateiname eingegeben werden, beispielsweise

-IWSCOM

Mit diesem Befehl würde das Einlesen von Wordstar (ws) vorbereitet. Wirklich eingelesen wird die Datei dann mit dem R-Befehl (siehe weiter unten).

Der List-Befehl

»L« steht für »List«, das den integrierten Disassembler aktiviert. Die Parameter entsprechen denen des Dump-Befehls. Die Mnemonics werden in der oben erklärten Intel-Schreibweise angegeben; Sie können dazu die Vergleichstabelle zu Rate ziehen. Nicht definierte Maschinencodes werden mit »3FAC ?? = 08« angezeigt.

Problematisch wird es, wenn Sie mit dem L-Befehl das Floppy-ROM disassemblieren wollen. Dort ist der Hardware-abhängige Teil von CP/M, das BIOS (Basic Input/Output System) gespeichert, welches die Schnittstelle zwischen CP/M und den speziellen Eigenschaften des Schneiders darstellt. Die Programmierer bei Amstrad (dem englischen Entwickler) haben selbstverständlich in Z80-Maschinencode programmiert, was zur Folge hat, daß auch Z80-Befehle ohne zugehörige 8080/8085-Pendants verwendet wurden.

Keine Schwierigkeiten haben Sie hingegen, wenn Sie das BDOS, den CCP oder andere CP/M-Programme auflisten lassen. BDOS und CCP wurden nämlich von Digital Research, dem »Erfinder« von CP/M, in 8080-Assembler geschrieben.

Der Move-Befehl

»M« steht für »Move«, einen Befehl, mit dem man Speicherbereiche verschieben kann:

-M0000, 0100,4500 kopiert den Inhalt des Speichers von 0000 bis 0100 hex nach 4500 bis 4600 hex (4500 + 0100 hex)

Der Read-Befehl

»R« steht für »Read«, dieser Befehl liest eine mit dem I-Kommando vorbereitete Datei in den RAM-Spei-

cher. Dazu kann ein Offset angegeben werden, der die Ladeadresse abhängig vom Beginn der TPA bestimmt. R oder R0 lädt die Datei an den Anfang der TPA bei 0100, R0200 an die Adresse TPA + 0200 = 0100 + 0200 = 0300 (alles hex).

Der Store-Befehl

»S« steht für »Store«. Dieser einfache, aber dennoch leistungsfähige, Befehl gestattet die byteweise Änderung von Speicherbereichen. Ein S mit Enter übernimmt die letzte verwendete Adresse, es kann aber auch ausdrücklich eine neue Adresse angegeben werden. Der DDT gibt die aktuelle Speicheradresse und deren Inhalt hexadezimal aus; daraufhin haben Sie die Möglichkeit, eine neue Ziffer (hexadezimal) einzutippen, die dann an der Adresse gespeichert wird. Wollen Sie eine Adresse unverändert belassen, drücken Sie ganz einfach die Enter-Taste.

-S0600

- 0600 FF
- 0601 F3
- 0602 A0
- 0603 .

Der Trace-Befehl

»T« steht für »Trace«. Damit lassen sich Maschinenprogramme in Einzelschritten abarbeiten. Vergleichbar ist der Befehl in etwa mit dem TRON-Kommando im Schneider-Basic.

Die Abarbeitung beginnt an der letzten verwendeten Adresse; angegeben werden muß die Zahl der abzuarbeitenden Programmschritte. Sollen zum Beispiel die nächsten 15 Maschinenbefehle ausgeführt werden, lautet Ihre Eingabe:

-T0F

Bei jedem Programmschritt erfolgt eine detaillierte Ausgabe der Prozessor-Register:

C,Z,M,E,I,A=.. B=.... D=.... H=.... S=.... P=.... Kommando

Die Abkürzungen haben folgende Bedeutung:

- C = Carry-Flag
- Z = Zero-Flag
- M = Minus-Flag
- E = Parity-Flag (E=even, gerade)
- I = Halfcarry-Flag (8080-Name: Interdigit Flag)
- A = Akkumulator
- B = Doppelregister BC
- D = Doppelregister DE
- H = Doppelregister HL
- S = Stackpointer SP
- P = Program Counter PC

Kommando zeigt den mnemonischen Code des ausgeführten Befehls an.

Der Untrace-Befehl

»U« steht für »Untrace«, eine Variante des Trace-Befehls. Die Eingabe entspricht der des T-Kommandos. Der Unterschied liegt darin, daß nicht jeder ausgeführte Befehl angezeigt wird. Lediglich beim letzten Befehl erscheint die Zeile mit den Prozessor-Daten.

Der Examine-Befehl

»X« steht für »Examine«. Dieser »Untersuchungs«-Befehl zeigt den gegenwärtigen Inhalt der Prozessor-Register, genauso wie beim »T«- oder »U«-Kommando. Eine Variante des X-Befehls erlaubt die direkte Veränderung der CPU-Register. Dazu wird der Registername unmittelbar hinter dem »X« angegeben. Daraufhin erscheint der Registername und -inhalt; dieser kann dann nach Belieben verändert werden:

—XB

B = 0000 3FAC

Mit diesem Befehl wird das BC-Register mit dem Wert 3FAC hex geladen.

Nachdem Sie ein Programm mit all diesen Befehlen analysiert und eventuell verändert haben, wollen Sie es sicher auch abspeichern. Dazu drücken Sie »Ctrl-C« oder geben »G0000« ein. Dies löst einen CP/M-Warmstart aus, und der CCP-Bedienungsprozessor wird aufgerufen. Zum Abspeichern als Binärdatei dient der residente CP/M-Befehl SAVE:

»A>SAVE xx filename.ext«

Jetzt benötigen Sie wieder den Wert, der beim Start von DDT als NEXT angegeben wurde. Damit SAVE weiß, wieviel Byte Sie abspeichern wollen, müssen Sie die Zahl der belegten »Pages« (Speicherseiten) errechnen. Eine Speicherseite ist 256 Byte lang. Sie nehmen deshalb von der NEXT-Zahl die ersten beiden Ziffern (das Highbyte). Bei NEXT=6400 hex wäre das 64 hex. Diesen Wert müssen Sie ins Dezimalsystem umrechnen: 64 hex = 100. Das Ergebnis kann dann bei SAVE angegeben werden — bis auf eine Ausnahme: Wenn das Lowbyte von NEXT Null ist — was gerade beim Beispiel 6400 hex zutrifft —, muß die Zahl der Pages um eins verringert werden. Der Befehl lautet dann so:

»A>SAVE 99 FILE.COM«

Selbstverständlich kann der DDT — wie viele andere CP/M-Programme auch — die Bildschirmausgaben auf dem Drucker mitprotokollieren. Dazu drücken Sie einfach »Ctrl-P«. Abgeschaltet wird die Protokollfunktion wiederum mit »Ctrl-P«.

(Martin Kotulla)

Die Befehle des DDT

A = Assemble	Mnemonics eingeben und assemblieren
D = Dump	Speicherbereich hexadezimal und ASCII ausgeben
F = Fill	Speicherbereich füllen
G = Go	Maschinenprogramme mit Breakpoints aufrufen
H = Hexcalc	Hexadezimale Addition und Subtraktion
I = Input	Dateinamen zum Datei-Einlesen vorgeben
L = List	Disassembler-Funktion
M = Move	Speicherbereiche verschieben
R = Read	Datei in den Speicher lesen
S = Store	Speicherbereiche hexadezimal abändern
T = Trace	Maschinenprogramme im Einzelschritt-Modus
U = Untrace	Wie Trace, jedoch ohne Ausgabe der CPU-Register
X = Examine	CPU-Register anzeigen und abändern

Code	8080-Kommando	Z80-Kommando
CE	ACI by	ADC A,by
8F	ADC A	ADC A,A
88	ADC B	ADC A,B
89	ADC C	ADC A,C
8A	ADC D	ADC A,D
8B	ADC E	ADC A,E
8C	ADC H	ADC A,H
8D	ADC L	ADC A,L
8E	ADC M	ADC A,(HL)
87	ADD A	ADD A,A
80	ADD B	ADD A,B
81	ADD C	ADD A,C
82	ADD D	ADD A,D
83	ADD E	ADD A,E
84	ADD H	ADD A,H
85	ADD L	ADD A,L
86	ADD M	ADD A,(HL)
C6	ADI by	ADD A,by
A7	ANA A	AND A
A0	ANA B	AND B
A1	ANA C	AND C
A2	ANA D	AND D
A3	ANA E	AND E
A4	ANA H	AND H
A5	ANA L	AND L
A6	ANA M	AND (HL)
E6	ANI by	AND by
CD	CALL wo	CALL wo
DC	CC wo	CALL C,wo
FC	CM wo	CALL M,wo
2F	CMA	CPL
3F	CMC	CCF
BF	CMP A	CP A
B8	CMP B	CP B
B9	CMP C	CP C
BA	CMP D	CP D
BB	CMP E	CP E
BC	CMP H	CP H
BD	CMP L	CP L
BE	CMP M	CP (HL)
D4	CNC wo	CALL NC,wo
C4	CNZ wo	CALL NZ,wo
F4	CP wo	CALL P,wo
EC	CPE wo	CALL PE,wo
FE	CPI by	CP by
E4	CPO wo	CALL PO,wo
CC	CZ wo	CALL Z,wo
27	DAA	DAA
09	DAD B	ADD HL,BC
19	DAD D	ADD HL,DE
29	DAD H	ADD HL,HL
39	DAD SP	ADD HL,SP
3D	DCR A	DEC A
05	DCR B	DEC B
0D	DCR C	DEC C
15	DCR D	DEC D
1D	DCR E	DEC E
25	DCR H	DEC H
2D	DCR L	DEC L
35	DCR M	DEC (HL)
0B	DCX B	DEC BC
1B	DCX D	DEC DE
2B	DCX H	DEC HL
3B	DCX SP	DEC SP
F3	DI	DI
FB	EI	EI
76	HLT	HALT
DB	IN by	IN A,by
3C	INR A	INC A
04	INR B	INC B

Code	8080-Kommando	Z80-Kommando
0C	INR C	INC C
14	INR D	INC D
1C	INR E	INC E
24	INR H	INC H
2C	INR L	INC L
34	INR M	INC (HL)
03	INX B	INC BC
13	INX D	INC DE
23	INX H	INC HL
33	INX SP	INC SP
DA	JC wo	JP C,wo
FA	JM wo	JP M,wo
C3	JMP wo	JP wo
D2	JNC wo	JP NC,wo
C2	JNZ wo	JP NZ,wo
F2	JP wo	JP P,wo
EA	JPE wo	JP PE,wo
E2	JPO wo	JP PO,wo
CA	JZ wo	JP Z,wo
3A	LDA wo	LD A,(wo)
0A	LDAX B	LD A,(BC)
1A	LDAX D	LD A,(DE)
2A	LHLD wo	LD HL,(wo)
01	LXI B,wo	LD BC,wo
11	LXI D,wo	LD DE,wo
21	LXI H,wo	LD HL,wo
31	LXI SP,wo	LD SP,wo
7F	MOV A,A	LD A,A
78	MOV A,B	LD A,B
79	MOV A,C	LD A,C
7A	MOV A,D	LD A,D
7B	MOV A,E	LD A,E
7C	MOV A,H	LD A,H
7D	MOV A,L	LD A,L
7E	MOV A,M	LD A,(HL)
47	MOV B,A	LD B,A
40	MOV B,B	LD B,B
41	MOV B,C	LD B,C
42	MOV B,D	LD B,D
43	MOV B,E	LD B,E
44	MOV B,H	LD B,H
45	MOV B,L	LD B,L
46	MOV B,M	LD B,(HL)
4F	MOV C,A	LD C,A
48	MOV C,B	LD C,B
49	MOV C,C	LD C,C
4A	MOV C,D	LD C,D
4B	MOV C,E	LD C,E
4C	MOV C,H	LD C,H
4D	MOV C,L	LD C,L
4E	MOV C,M	LD C,(HL)
57	MOV D,A	LD D,A
50	MOV D,B	LD D,B
51	MOV D,C	LD D,C
52	MOV D,D	LD D,D
53	MOV D,E	LD D,E
54	MOV D,H	LD D,H
55	MOV D,L	LD D,L
56	MOV D,M	LD D,(HL)
5F	MOV E,A	LD E,A
58	MOV E,B	LD E,B
59	MOV E,C	LD E,C
5A	MOV E,D	LD E,D
5B	MOV E,E	LD E,E
5C	MOV E,H	LD E,H
5D	MOV E,L	LD E,L
5E	MOV E,M	LD E,(HL)
67	MOV H,A	LD H,A
60	MOV H,B	LD H,B
61	MOV H,C	LD H,C

Code	8080-Kommando	Z80-Kommando		
62	MOV	H,D	LD	H,D
63	MOV	H,E	LD	H,E
64	MOV	H,H	LD	H,H
65	MOV	H,L	LD	H,L
66	MOV	H,M	LD	H,(HL)
6F	MOV	L,A	LD	L,A
68	MOV	L,B	LD	L,B
69	MOV	L,C	LD	L,C
6A	MOV	L,D	LD	L,D
6B	MOV	L,E	LD	L,E
6C	MOV	L,H	LD	L,H
6D	MOV	L,L	LD	L,L
6E	MOV	L,M	LD	L,(HL)
77	MOV	M,A	LD	(HL),A
70	MOV	M,B	LD	(HL),B
71	MOV	M,C	LD	(HL),C
72	MOV	M,D	LD	(HL),D
73	MOV	M,E	LD	(HL),E
74	MOV	M,H	LD	(HL),H
75	MOV	M,L	LD	(HL),L
3E	MVI	A,by	LD	A,by
06	MVI	B,by	LD	B,by
0E	MVI	C,by	LD	C,by
16	MVI	D,by	LD	D,by
1E	MVI	E,by	LD	E,by
26	MVI	H,by	LD	H,by
2E	MVI	L,by	LD	L,by
36	MVI	M,by	LD	(HL),by
00	NOP		NOP	
B7	ORA	A	OR	A
B0	ORA	B	OR	B
B1	ORA	C	OR	C
B2	ORA	D	OR	D
B3	ORA	E	OR	E
B4	ORA	H	OR	H
B5	ORA	L	OR	L
B6	ORA	M	OR	(HL)
F6	ORI	by	OR	by

Code	8080-Kommando	Z80-Kommando		
D3	OUT	by	OUT	by,A
E9	PCHL		JP	(HL)
C1	POP	B	POP	BC
D1	POP	D	POP	DE
E1	POP	H	POP	HL
F1	POP	PSW	POP	AF
C5	PUSH	B	PUSH	BC
D5	PUSH	D	PUSH	DE
E5	PUSH	H	PUSH	HL
F5	PUSH	PSW	PUSH	AF
17	RAL		RLA	
1F	RAR		RRA	
D8	RC		RET	C
C9	RET		RET	
07	RLC		RLCA	
F8	RM		RET	M
D0	RNC		RET	NC
C0	RNZ		RET	NZ
F0	RP		RET	P
E8	RPE		RET	PE
E0	RPO		RET	PO
0F	RRC		RRCA	
C7	RST	0	RST0	#0000
CF	RST	1	RST1	#0008
D7	RST	2	RST2	#0010
DF	RST	3	RST3	#0018
E7	RST	4	RST4	#0020
EF	RST	5	RST5	#0028
F7	RST	6	RST6	#0030
FF	RST	7	RST7	#0038
C8	RZ		RET	Z
9F	SBB	A	SBC	A,A
98	SBB	B	SBC	A,B
99	SBB	C	SBC	A,C
9A	SBB	D	SBC	A,D
9B	SBB	E	SBC	A,E
9C	SBB	H	SBC	A,H
9D	SBB	L	SBC	A,L

Code	8080-Kommando	Z80-Kommando		
9E	SBB	M	SBC	A,(HL)
DE	SBI	by	SBC	A,by
22	SHLD	wo	LD	(wo),HL
F9	SPHL		LD	SP,HL
32	STA	wo	LD	(wo),A
02	STAX	B	LD	(BC),A
12	STAX	D	LD	(DE),A
37	STC		SCF	
97	SUB	A	SUB	A
90	SUB	B	SUB	B
91	SUB	C	SUB	C
92	SUB	D	SUB	D
93	SUB	E	SUB	E
94	SUB	H	SUB	H
95	SUB	L	SUB	L
96	SUB	M	SUB	(HL)
D6	SUI	by	SUB	by
EB	XCHG		EX	DE,HL
AF	XRA	A	XOR	A
A8	XRA	B	XOR	B
A9	XRA	C	XOR	C
AA	XRA	D	XOR	D
AB	XRA	E	XOR	E
AC	XRA	H	XOR	H
AD	XRA	L	XOR	L
AE	XRA	M	XOR	(HL)
EE	XRI	by	XOR	by
E3	XTHL		EX	(SP),HL

by — Byte (8 Bit)
wo — Wort (16 Bit)
di — Distanz (8 Bit)

8080 und Z80 im Vergleich

Wordstar stark verbessert



Auch wenn Ihr Wordstar-Programm an den Schneider angepaßt ist, so gibt es immer noch viele Wünsche, die offenbleiben. Dabei kennt diese Textverarbeitung noch viele Routinen, die das Arbeiten weitaus angenehmer machen.

Wenn Sie eine Wordstar-Version haben, die auf Ihrem Computer prinzipiell funktioniert, dann dürfen Sie diesen ersten Abschnitt ohne Bedenken überspringen.

Wenn aber nach dem Aufruf des Programms nur Unsinn auf dem Bildschirm erscheint, dann müssen Sie zunächst das Ihrem Wordstar beiliegende Install-Programm starten.

Spielen Sie dabei ruhig etwas mit den Menü-Routinen und geben Sie beliebige Änderungen ein. Nach-

dem Sie dann das Install-Programm beendet haben, testen Sie die Wirkungen der Veränderungen. Zerstören können Sie dabei übrigens nichts — denn mit der Install-Routine läßt sich das Programm immer wieder umbauen. Machen Sie sich aber trotzdem zuvor eine Sicherheitskopie, denn mit der Originaldiskette sollten Sie nie arbeiten.

Sobald Ihnen die Bedienung der Installierungsroutinen geläufig ist, müssen Sie sich bei der Erstanwendung besonders auf folgende Punkte konzentrieren:

- Die Bildschirmgröße muß auf 25 x 80 Zeichen eingestellt werden.
- Die CP/M-Entsprechung des Locate-Befehls muß unbedingt installiert werden. Im entsprechenden Menüpaket werden dazu von Ihnen verschiedene Angaben verlangt. Diese sind auf dem CPC 464 und 664:
- Die Steuerzeichenfolge, um einen Cursor zu positionieren, besteht beim CPC 464 und 664 aus dem Zeichen 1F hex (31 dez).
- Zwischen der Angabe von Zeile und Spalte sowie nach beiden, braucht keine Steuerzeichenfolge gesendet zu werden.
- Die Angabe der Zeile und Spalte erfolgt binär, wobei die Spalte vor der Zeile ausgegeben werden muß.
- Der »Offset« (Startwert), der vor der Ausgabe auf den Bildschirm zu den Koordinaten addiert wird, hat sowohl für die Zeile wie auch für die Spalte den Wert 1.
- Das Steuerzeichen für Ein- und Ausschalten der Hervorhebung (durch inverse Zeichen) ist beim CPC der Code 18 hex (24 dez).
- Der Steuercode zum Löschen einer Zeile von Cursorposition bis zum Ende ist 12 hex (18 dez). Wenn Sie im Install-Programm den entsprechenden Menüpunkt aufrufen und den Wert eingeben, dann löscht Word-

star später eine Zeile nicht mehr Zeichen für Zeichen. Das führt dann zu einer spürbaren Beschleunigung.

— Manche Terminals verlangen nach jeder Bildschirmausgabe eine »Ruhepause«. Ihr Schneider benötigt diese Ruhepause nicht. Deshalb ist es ratsam, alle Wartezeichen auf Null zu setzen. Wenn Sie mit dem Programm »DDTCOM« umgehen können, lassen sich diese Änderungen auch nachträglich (ohne das Install-Programm) durchführen, indem Sie die beiden Bytes an den Adressen 02AE und 02AF hex auf Null setzen.

— Die rechte untere Ecke des Bildschirms darf unter Wordstar nicht beschrieben werden. Der CPC scrollt sonst den Bildschirm vorzeitig, ohne daß dies vom Programm registriert wird. In der Folge würde dann irgendwann nur noch »Müll« auf dem Bildschirm stehen. Bei fremden Terminals ist das Beschreiben der Ecke aber oft erlaubt. Im Install-Programm sollte deshalb das entsprechende Flag gesetzt werden.

Wenn Sie sich mit Hilfe des Install-Programms versichert haben, daß diese Werte auch in Ihrem Wordstar korrekt eingetragen sind, muß es ordnungsgemäß laufen.

Auch wenn Ihr Wordstar ordnungsgemäß funktioniert, sollten Sie sich mit einer Minimalanpassung nicht zufrieden geben. Es gibt vier grundsätzliche Schwachstellen, die man mit wenig Aufwand beseitigen kann.

Langsamer Bildschirmaufbau:

Besonders als Anfänger werden Sie Wordstar in der Hilfsstufe 3 betreiben wollen. Dann haben Sie aber allen Grund zum Ärgern: Jedesmal, wenn der Bildschirm scrollt (und das kommt nach jeder Zeile vor), dauert es »ewig«, bis er wieder richtig aufgebaut ist. Selbst wenn Sie Wordstar in Hilfsstufe 2 benutzen, bleiben Sie davon nur verschont, solange sich der Cursor ganz unten auf dem Bildschirm befindet.

Der Zeitbedarf kommt daher, daß beim Scrollen des Bildschirms jedes Zeichen einzeln an die neuen Positionen ausgegeben wird. Außerdem prüft das Programm jedes-

mal, ob eine Taste gedrückt wurde, die die Ausgabe überflüssig macht. Drückt man keine Taste, so kostet das unnötig Zeit.

Aus dem Wordstar-Handbuch geht hervor, daß Wordstar die Bildschirmsteuersequenzen »In der Cursorzeile Leerzeile einfügen und Bildschirmbereich unterhalb abwärts scrollen« sowie »In der Cursorzeile löschen und Bildschirmbereich unterhalb nach oben scrollen« unterstützt, wenn der Computer sie versteht. Das kann der Schneider aber nicht. Es ist aber möglich, diese Funktion wenigstens zu simulieren.

Optimieren eines funktionierenden Wordstars

Sie werden es kaum glauben, aber mit dieser Änderung wird der Bildschirmaufbau um den Faktor 100 beschleunigt (von einigen Sekunden auf einen Sekundenbruchteil)! Sie brauchen nur die Steuerzeichenfolgen aus den Tabellen 1 und 2 eingeben. Doch wo ist im Speicher Platz dafür?

In den Bereich ab 0274 beziehungsweise 027B hex könnte im »WSCOM«-Programm eine entsprechende Folge eingefügt werden. Leider darf diese aber nur sechs Zeichen lang sein und keine 14 beziehungsweise 15, wie es nötig ist.

Es gibt aber trotzdem eine Lösung: Man darf nämlich nicht nur Steuerzeichen nach Wunsch neu einsetzen, sondern auch eine vollständig neue Bildschirmroutine. In der eingebauten Routine geht Wordstar sehr systematisch vor. Zunächst ruft die oberste Programmhierarchie ein Unterprogramm »Füge Zeile auf dem Bildschirm ein« auf, ohne sich darum zu kümmern, wie das Unterprogramm das bewerkstelligt. Erst dieses Unterprogramm sucht sich die geeigneten Steuercodes aus einer Tabelle und übergibt sie der endgültigen Konsolen-Ausgaberroutine — wieder ohne sich darum zu kümmern, wie diese das macht. Es ist also möglich, an dieser Stelle eine weitere Routine aufzurufen, beispielsweise eine Umsetzung

des ASCII-Codes — den Wordstar benützt — in einen anderen, den Ihr Terminal versteht (wenn Sie ein sehr exotisches Terminal besitzen).

Ein Weg, die gewünschten Strings zu installieren, ist hier, an den Positionen 0274 beziehungsweise 027B hex zwei »Phantomsteuercodes« einzutragen, die normalerweise nicht verwendet werden. Beim Schneider bieten sich dazu die Steuercodes 19 hex (Symbol-Befehl: hier nicht benötigt) und 1B hex (immer unbelegt) an. Der eigentlichen Konsolenausgabe wird mit Hilfe einer »Umleitung« eine weitere Routine vorgeschaltet, die in der Lage ist, diese »Phantomsteuercodes« zu erkennen. Soll ein beliebiges anderes Zeichen ausgegeben werden, so wird die normale Routine aufgerufen. Das geht am einfachsten mit einem direkten BIOS-Aufruf, um jegliche Veränderung durch das Betriebssystem auszuschließen. Tritt aber das Steuerzeichen 19 hex oder 1B hex auf, so soll die neue Konsolen-Ausgaberroutine nicht nur ein Zeichen, sondern den ganzen String an den Bildschirm senden. Selbstverständlich muß die neue Routine prüfen, daß nicht fälschlicherweise Parameter anderer Befehle (beispielsweise von Locate) mit dem Wert 19 oder 1B hex ausgeführt werden. Der Aufruf der neuen Konsolen-Ausgaberroutine muß in die Adresse 02C0 hex eingetragen werden.

Unsichtbarer Cursor:

Wenn sich der Cursor auf oder neben einem hervorgehobenen (Vordergrund und Hintergrundfarbe vertauscht) Zeichen befindet, ist er unsichtbar. Um dies zu ändern, schreibt man den Wert FF hex in die Speicherzelle 02B5 hex. Das Programm läßt den Cursor dann blinken. Die Werte in den Speicherzellen 02CF hex und 02D0 hex bestimmen dabei die Blinkgeschwindigkeit.

Die Drucker-Statusabfrage:

Wordstar läuft unter allen Versionen des Betriebssystems. Ältere als CP/M 2.2 haben dabei aber keine Routine, die ermittelt, ob der Drucker zum Datenempfang bereit ist oder nicht. Die Druckerroutine gibt die Kontrolle jedoch erst wie-

Steuercodes	»Basic-ähnliche Schreibweise«	Kommentar
1A 00 4F y 18	WINDOW 1,80,y,25	aktuelles Window umfaßt nur zu scrollenden Bildschirmbereich
1F 01 1A	LOCATE 1,26	Cursor außerhalb Window positionieren
20	PRINT " ";	Zeichen ausgeben -> löst Scrolling aus
1E	HOME	Cursor an Anfang der laufenden Zeile
1A 00 4F 00 18	WINDOW 1,80,1,25	aktuelles Window umfaßt wieder gesamten Bildschirm

(y = Zeile, auf der sich der Cursor befindet)

Tabelle 1. Die Steuerzeichenfolge für »Cursorzeile einfügen«

Steuercodes	»Basic-ähnliche« Schreibweise	Kommentar
1A 00 4F y 18	WINDOW 1,80,y,25	unterer Bildschirmbereich = aktuelles Window
1E	HOME	
0B	CURSOR UP	Cursor außerhalb Window positionieren
20	PRINT " ";	Zeichen ausgeben -> löst Scrolling aus
1E	HOME	Anfang der laufenden Zeile
1A 00 4F 00 18	WINDOW 1,80,1,25	aktuelles Window umfaßt wieder gesamten Bildschirm

(y = Zeile, auf der sich der Cursor befindet)

Tabelle 2. Die Steuerzeichenfolge für »Cursorzeile löschen«

der an das Hauptprogramm zurück, wenn das Zeichen wirklich beim Drucker angekommen ist.

Damit ist man zwar von der Hardware unabhängig — muß aber damit rechnen, daß der Computer »abstürzen« kann, wenn der Drucker nicht bereit ist.

Bei CP/M kann man aber mit einem direkten BIOS-Aufruf den Druckerstatus abfragen. Der Computer gerät dabei nie in eine Endlosschleife. Diese Fähigkeit kann man von Wordstar aus nutzen, wenn man eine entsprechende Routine einfügt. In der Adresse 071A hex von »WSCOM« sind 3 Byte für einen Aufruf dieser Drucker-Statusroutine reserviert.

Wenn Sie diese Änderung richtig gemacht haben, erkennen Sie das sofort beim Aufruf der Druckerroutine. Bei einem Fehler (beispielsweise Papiermangel) erscheint das Hauptmenü mit der Meldung »P=Druck halt«. Sie können den mißglückten Versuch ohne Tricks korrekt beenden.

Deutscher Zeichensatz:

Wenn Sie das für Schneider-Computer speziell angepaßte Wordstar besitzen, das vom Markt & Technik Verlag vertrieben wird, dann steht Ihnen auch ein deutscher Zeichensatz zur Verfügung. Das »Design« der Sonderzeichen ist jedoch nicht besonders ansprechend. Die Matrizen der Zeichen »Š,Ā,Ů,Ů,t,ä,ö,ü und ß« liegen in dieser Reihenfolge direkt hintereinander im Speicher des Computers. Sie belegen die Adressen von 3FCD bis 4014 hex in »WSCOM« und können dort leicht verändert werden.

Die praktische Durchführung der Wordstar-Anpassung:

Nachdem es bis jetzt sehr theoretisch zugegangen ist, finden Sie hier die Listings, mit denen Sie die Wordstar-Anpassung konkret durchführen können.

Beherrschen Sie nicht den 8080-Assembler, müssen Sie die Anpassung in vier Schritten durchführen: Als erstes brauchen Sie eine Arbeitsdiskette im CP/M-Format. Auf dieser Diskette müssen die Routinen »WSCOM«, »WSOVLY1.OVR« und »WSMSGSCOM« stehen. Die Dateien dürfen nicht schreibgeschützt sein. Nachdem Sie die Datei kopiert haben, müssen noch mindestens 20 KByte frei sein. Machen Sie sich jetzt von den anderen Dateien, die sich auf der Diskette befinden, Sicherheitskopien. Sollten Sie bei dieser Art von Experimenten einen Fehler machen, kann es Ihnen leicht pas-

sieren, daß Sie nachher keine einzige Datei mehr lesen können.

Als weiteres tippen Sie das Basic-Programm in Listing 1 ab und speichern es unter dem Namen »WSPATCH.BAS«. Starten Sie nun dieses Basic-Programm. Ist es fehlerfrei, erscheint nach einiger Zeit die Meldung »Alles ok.«. Dieses Programm erzeugte auf der Diskette eine Datei mit dem Namen »WSPATCH.COM«.

Erscheint aber die Meldung »Prüfsummenfehler in Zeile«, dann befindet sich in dieser Zeile ein Fehler. Den müssen Sie korrigieren und das Programm nochmals starten. Es befindet sich nun aber schon eine unvollständige Datei mit dem Namen »WSPATCH.COM« auf der Diskette. Diese muß gelöscht werden. Beachten Sie bitte, daß nur Tippfehler ent-

deckt werden, die sich innerhalb einer Zeile befinden. Haben Sie eine Zeile ganz vergessen oder zwei Zeilen vertauscht, ist die Datei trotz Meldung »Alles ok.« fehlerhaft. Sie dürfen das Basic-Programm also erst löschen, wenn alle Fehler ausgemerzt sind.

Die Datei »WSPATCH.COM« ist nun ein ganz normales CP/M-Programm. Rufen Sie CP/M auf und starten Sie »WSPATCH.COM« durch Eingabe des Namens (aber ohne »COM«). Jetzt muß ein Text auf dem Bildschirm erscheinen und das Laufwerk arbeiten. Erscheint aber die Fehlermeldung »FILE NOT FOUND«, dann haben Sie vergessen, die Datei »WSCOM« auf Ihre Arbeitsdiskette zu kopieren.

Wenn das »A« wieder auf dem Bildschirm erscheint, sind alle oben besprochenen Anpassungen korrekt installiert. Testen Sie Wordstar und beachten Sie folgende Punkte:

- Versuchen Sie eine Datei auszu-
- Tippen Sie einen Text ein und probieren Sie tN und tY irgendwo in der Mitte des Textes aus. Das Scrollen sollte jetzt wirklich blitzartig, ohne Neuaufbau des Bildschirms vor sich gehen.

- Blinkt der Cursor, wenn er sich neben einem inversen Feld befindet?

- Wie gefallen Ihnen die deutschen Umlaute?

Wenn irgendetwas nicht funktioniert, dann befindet sich noch ein unentdeckter Fehler im Programm »WSPATCH.BAS«. Vergessen Sie in diesem Fall nicht, die »falsche« Wordstar-Anpassung zu löschen.

Wenn Ihr Programm ordnungsgemäß funktioniert, brauchen Sie die »WSPATCH«-Programme nicht mehr. Trotzdem sollten Sie sie aufheben, vielleicht möchten Sie später einmal ein nochmals modifiziertes Wordstar installieren.

Wenn Sie sich mit der Bedienung des »DDTCOM«-Programms auskennen, dann können Sie das Patchen des »WSCOM«-Files auch in Handarbeit durchführen. In diesem Fall sparen Sie sich das Eintippen des Basic-Laders. Dafür ist aber die Gefahr, daß Sie einen unentdeckten Fehler eingeben, um ein Vielfaches höher. Entscheiden Sie sich trotzdem dafür, dann sollten Sie so vorgehen:

- Starten Sie den Debugger mit »DDT WSCOM«
- Geben Sie von Hand alle in Listing 2 dargestellten Änderungen ein. Beachten Sie dabei den Bereich von 02E0 bis 034B hex. Dieser steht für

```

10 '***** [F9FC]
20 ' (C) HELMUT FISCHER * [2224]
30 '***** [7703]
40 '***** [E9D2]
50 'Basic-Lader fuer die Datei 'WSPATCH.
    CDM [F6C6]
60 'PENDUT'WSPATCH.COM':zeile=10000 [F138]
70 pruef=0 [192C]
80 FOR j=0 TO 7 [1700]
90 READ d$ [349A]
100 IF d$="Ende" THEN CLOSEOUT:PRINT"Al
    es O.K.":END [B2CA]
110 d=VAL("%"+d$):PRINT#9,CHR$(d):pruef
    =pruef+d [EB5C]
120 NEXT [C534]
130 READ p [F64C]
140 IF pruef<p THEN CLOSEOUT:PRINT"Prue
    fsummenfehler in Zeile":zeile=zeile+10
150 '***** [EC3E]
160 GOTO 70 [DD62]
170 DATA C3,85,01,00,08,08,08,45, 470 [E9E2]
180 DATA 72,77,65,69,74,65,72,74, 886 [A3E8]
190 DATA 65,20,41,66,70,61,73,73, 747 [2FCA]
200 DATA 75,66,67,20,76,6F,6E,20, 733 [0C0A]
210 DATA 57,6F,72,64,73,74,61,72, 854 [42DC]
220 DATA 20,56,65,72,73,2E,20,33, 577 [C186]
230 DATA 2E,30,30,20,2D,20,32,2E, 347 [C1C0]
240 DATA 74,74,20,61,20,43,50,43, 533 [3D8E]
250 DATA 78,36,34,3A,0D,0A,09,2D, 361 [5B04]
260 DATA 20,43,75,72,73,6F,72,20, 702 [B1B4]
270 DATA 62,6C,69,6E,6B,74,20,61, 773 [6116]
280 DATA 78,62,20,65,20,6E,20,6E, 747 [5FF4]
290 DATA 73,65,6D,20,46,65,6C,64, 736 [64FC]
300 DATA 0D,0A,09,2D,20,57,53,20, 311 [99BC]
310 DATA 75,66,74,65,72,73,74,70, 914 [5B04]
320 DATA 74,74,20,61,20,61,65,64, 770 [7703]
330 DATA 6F,77,73,0D,0A,09,2D,20, 454 [D90E]
340 DATA 56,65,72,62,65,73,73,65, 831 [F4BC]
350 DATA 72,74,65,73,20,53,69,6D, 775 [10E2]
360 DATA 75,66,74,61,6E,20,46,72, 725 [8F1E]
370 DATA 75,63,68,65,6E,2F,45,64, 750 [A11C]
380 DATA 69,74,69,65,72,65,6E,0D, 765 [651C]
390 DATA 0A,24,08,20,1A,21,0D,40, 209 [2696]
400 DATA F9,11,07,01,0E,09,CD,05, 508 [A46A]
410 DATA 00,11,5D,00,21,87,02,01, 281 [E674]
420 DATA 0B,0D,ED,80,0E,0F,11,5C, 562 [EF3E]
430 DATA 00,CD,4D,03,3E,02,CD,76, 672 [0740]
440 DATA 02,21,01,19,22,74,00,0E, 467 [5077]
450 DATA 1B,22,FB,00,3E,02,CD,7B, 704 [9E4E]
460 DATA 02,3E,03,CD,76,02,3E,FF, 709 [DD56]
470 DATA 32,85,00,3E,C3,21,EO,02, 747 [24F6]
480 DATA 02,00,7E,02,01,00,00,21, 520 [43B4]
490 DATA 05,22,CF,00,11,EO,00,21, 520 [43B4]
500 DATA 92,02,01,20,0D,ED,80,3E, 656 [AEE6]
510 DATA 03,CD,7B,02,3E,04,CD,76, 722 [2C40]
520 DATA 02,11,80,00,21,87,02,01, 281 [E674]
530 DATA 53,00,ED,80,3E,04,CD,7B, 890 [4464]
540 DATA 02,3E,03,CD,76,02,3E,FF, 718 [D674]
550 DATA 32,98,00,3E,C3,21,4A,03, 569 [13EE]
560 DATA 32,9A,00,2E,9B,00,3E,0C, 467 [5077]
570 DATA CD,7B,02,3E,7D,CD,76,02, 842 [F97E]
580 DATA 21,05,03,11,CD,00,01,33, 315 [E096]
590 DATA 00,ED,80,3E,7D,CD,7B,02, 930 [958B]
600 DATA 3E,7E,CD,7B,02,21,3B,03, 605 [600C]
610 DATA 11,80,00,01,15,0D,ED,80, 580 [42B4]
620 DATA 3E,7E,CD,7B,02,0E,10,11, 565 [7E40]
630 DATA 5C,00,CD,05,00,C7,0E,21, 548 [3C14]
640 DATA 3C,7D,02,0E,22,32,7D,00, 545 [A5FC]
650 DATA F1,5C,00,0D,05,0D,59,57, 607 [A970]
660 DATA 53,20,20,20,20,20,43, 342 [E354]
670 DATA 4F,4D,45,4F,21,2B,03,35, 433 [34FC]
680 DATA C2,F3,02,34,79,FE,1F,C2, 1091 [A822]
690 DATA 2A,01,00,19,E9,FE,1B,CA, 784 [A864]
700 DATA 11,03,FE,19,C2,F3,02,7B, 858 [3920]
710 DATA 32,2C,03,21,29,03,06,0F, 195 [9A8B]
720 DATA C3,1A,03,7B,32,3F,03,21, 493 [2DE4]
730 DATA 3C,03,06,0E,4E,23,C5,ES, 622 [E826]
740 DATA CD,F3,02,E1,C1,05,C2,1A, 1093 [F076]
750 DATA 03,C9,01,1A,04,4F,02,1B, 336 [94D4]
760 DATA 1F,01,1A,20,1E,1A,00,4F, 225 [E5F2]
770 DATA 00,1B,1F,50,1B,20,1A,00, 217 [5FAA]
780 DATA 4F,02,1B,1E,0B,20,1E,1A, 234 [1814]
790 DATA 00,0F,00,1B,11,2A,0D,367 [0E22]
800 DATA F8,02,2F,1F,C9,3C,36,3C, 747 [3A5B]
810 DATA 66,66,3C,06,3C,C6,3B,6C, 692 [C93B]
820 DATA C6,FE,C6,C6,00,C6,3B,6C, 1210 [DE9B]
830 DATA C6,C6,6C,3B,00,C6,00, 956 [8444]
840 DATA C6,C6,70,00,1F,3B,66, 904 [C250]
850 DATA C0,00,00,00,6C,00,7B, 426 [09A4]
860 DATA 0C,7C,CC,76,00,6C,00, 626 [234C]
870 DATA 66,66,66,3C,00,6C,00, 474 [58DB]
880 DATA C6,C6,C6,7E,00,C,34,66, 904 [E250]
890 DATA 66,66,66,CE,00,CD,05, 726 [3F2A]
900 DATA 3C,0D,0E,09,11,5B,03,CD, 591 [1D3A]
910 DATA 05,00,C7,46,49,4C,45,20, 524 [D7D0]
920 DATA 4E,4E,54,20,46,4E,4E,4E, 585 [243A]
930 DATA 4E,0D,0A,24,00,00,00, 127 [A98B]
940 DATA Ende [79EB]

```

Listing 1. Der Basic-Lader für eine neue Bildschirmgestaltung

Ergänzungen des Benutzers zur freien Verfügung. Dabei benötigen Sie folgende Befehle:

— »D<Anfang>,<Ende>«: gibt den bezeichneten Speicherinhalt im Hexadezimalformat aus. Wenn Sie die Endadresse weglassen, werden nur 16 Zeilen ausgegeben.

— »L<Anfang>,<Ende>«: gibt den Speicherinhalt als Disassemblerlisting aus. Ohne Endadresse stoppt die Ausgabe wieder nach 16 Zeilen.

— »IS« hält eine laufende Ausgabe an, bis eine weitere beliebige Taste gedrückt wird.

— beliebige Taste während einer laufenden Ausgabe: bricht die laufende Ausgabe ab.

— »S<Anfang>«: startet eine hexadezimale Eingabe. Durch eine ungültige Eingabe wird sie abgebrochen. Eine leere Eingabe läßt den entsprechenden Speicherinhalt unverändert.

— »A<Anfang>«: der Mini-Assembler des DDT-Programmes. Alle Befehle werden in 8080-Mnemonics gemacht. Durch eine leere Eingabe wird dieser Modus abgebrochen.

Selbstverständlich gibt es noch eine Menge weiterer Befehle. Wenn Sie sich für diese interessieren, sollten Sie Spezialliteratur zu Rate ziehen. Für unser Problem reichen aber die hier gezeigten Befehle vollständig aus.

— Unterbrechen Sie nach dem Ändern der Datei das DDT-Programm durch Eingabe von »!C«. Das neue Wordstar-Programm befindet sich nun im Speicher. Mit »SAVE 64 WSCOM« wird es auf Diskette gespeichert.

— Jetzt sollten Sie das veränderte Wordstar (wie oben beschrieben) testen. Falls Sie eine Änderung vergessen haben, können Sie diese durch erneutes Starten des »DDT« nachholen. Wenn das Programm dann immer noch nicht richtig funktioniert, haben Sie wahrscheinlich irgendeine Eingabe durch einen Tippfehler an die falsche Adresse geschickt. In diesem Fall ist das Wordstar-Programm völlig verdorben, und Sie sollten mit einer neuen Kopie noch mal beginnen.

Ein weiteres Problem ist die Druckeranpassung. Die Standard-Installation des Wordstar-Programms unterstützt nur Eigenschaften, die allen Druckern gemeinsam sind — also fast keine. Ausgenommen sind nur Doppeldruck, Fettdruck und Unterstreichen. Diese Effekte werden aber nicht mit Hilfe der Hardware des Druckers er-

zeugt, sondern dadurch, daß eine Zeile zweimal hintereinander ohne Zeilenvorschub ausgegeben oder daß eine Leerzeile angegeben wird.

Analog der Installierung der Bildschirmsteuerzeichen gibt es im Programm auch reservierte Bereiche zum Eintragen von Druckersteuerzeichen. Diese Bereiche sind im Anhang F des Wordstar-Handbuchs beschrieben. Einfache Änderungen kann man jedoch auch mit Hilfe des Install-Programms durchführen.

Übrigens sind — mit Ausnahme der Schönschrift und der Grafikzeichen — alle Steuerzeichen eines Epson-kompatiblen Druckers und des NLQ401 gleich. Die folgende Anpassung ist also für Sie auch dann nützlich, wenn Sie keinen NLQ401, sondern »nur« einen Epson-Drucker besitzen.

Die beiden Anpassungen (Bildschirm und Drucker) sind völlig unabhängig voneinander. Sie können also mit Ihrem Wordstar entweder die eine, nur die andere oder alle beide benutzen. Die erweiterte Druckersteuerung ist damit sogar auf Programmen lauffähig, die nicht an den CPC angepaßt sind.

Starten Sie Ihr Wordstar-Programm und tippen Sie während der Bearbeitung eines Textes einmal »!P« ein. Es wird das Drucker Menü ausgegeben. Man sollte jetzt versuchen, möglichst viele Fähigkeiten des NLQ-Druckers auf die noch freien Steuerzeichen zu legen. Ein Lösungsvorschlag dafür ist:

— »!PB«, »!PD«, »!PS«, »!PX« und »!PENTER«: Schattendruck/Doppeldruck/Unterstreichen/Durchstreichen und Überdrucken. Diese Funktionen brauchen nicht verändert zu werden, da sie hardware-unabhängig installiert sind.

— »!PF«, »!PG« und »!PV«: Zeichen löschen/Phantomzeichen. Diese Funktionen beherrscht der NLQ401 nicht, und müssen deshalb weglassen werden.

— »!PT« und »!PV«: Hoch- und Tiefstellen (ein/aus). Diese Funktionen können ruhig auf ihren Buchstaben belassen werden — nur müssen sie richtig initialisiert werden.

— »!PY«: Farbband wechseln. Beim NLQ401 kann das Farbband nicht automatisch gewechselt werden. Eine Möglichkeit, dieses virtuelle Steuerzeichen doch zu benutzen ist, es mit den Steuersequenzen für Schrift in doppelter Größe zu belegen.

ermöglichkeit wird am besten mit den Drucker-codes für komprimierte Schrift belegt.

— »!PQ« und »!PW«: Diese Funktionen können vom Anwender frei belegt werden. Hier kann man Near-Letter-Quality oder Schnellschrift wählen.

— »!PR«: Auch diese Funktion kann frei belegt werden. Sinnvoll ist, damit den amerikanischen oder den deutschen Zeichensatz alternativ auszuwählen. Es können dann auch eckige Klammern und so weiter gedruckt werden.

— »!PE«: Auch diese Funktion ist frei definierbar. Damit könnte man die Grafikzeichen des Druckers auswählen. Es sind dann in einem Text beispielsweise auch griechische Sonderzeichen erlaubt.

Einige Steuerzeichenfolgen lassen sich ganz einfach in die dafür reservierten Bereiche des Wordstar-Programms eintragen. Bei anderen treten jedoch Probleme auf.

Wordstar unterstützt — selbst nach dem Patchen — nur die Funktionen »Wagen eine halbe Zeile hochrollen« und »Wagen eine halbe Zeile runterrollen« zum Hoch- und Tiefstellen von Zeichen. Die Anweisung Tiefstellen eines Zeichens wird also mit demselben internen Steuerzeichen durchgeführt wie das Hochstellen des Zeichens.

Der NLQ401 (und der Epson) simulieren nun aber das Hoch- und Tiefstellen mit Hilfe einer speziellen Schriftart. Deshalb wird die Hochstellung mit der Steuerzeichenfolge 1B54 hex gelöscht, die Tiefstellung aber mit der Steuerzeichenfolge 1B5301 hex eingeschaltet. Nur durch Eintragen der normalen Steuerzeichenfolge können Sie das Problem also nicht lösen.

Abhilfe schaffen hier wieder — wie schon bei der Installation der Bildschirmsteuerung — »Phantomsteuerzeichen«, die erst unmittelbar vor der Ausgabe in eine wirkliche Steuerzeichenfolge übersetzt werden. Die dazu nötige neue Routine muß auch mit einem »Gedächtnis« versehen werden, damit sie »weiß«, welcher Befehl nun effektiv ausgeführt werden soll. Auch zur Auswahl des Zeichensatzes steht nur ein einziges Steuerzeichen zur Verfügung. Es muß also ebenfalls ein »Phantomsteuerzeichen mit Gedächtnis« eingesetzt werden.

Grafikzeichen besitzen beim NLQ401 einen Code, bei dem das achte Bit gesetzt ist. Dies kann nun aber der CPC nicht übertragen. Deshalb gibt es ein spezielles Umschaltsteuerzeichen, das den nor-

malen Zeichensatz abschaltet und den Grafikzeichensatz auf die Codes der normalen ASCII-Zeichen legt. Ein »A«, das den Code 41 hex hat, wird also im Grafik-Modus als das Grafikzeichen mit dem Code C1 hex (41+80) ausgedruckt. Die Ausgabe im Grafik-Modus erfolgt dann so lange, bis ein »Rückschaltzeichen« ausgegeben wird. Im Grafik-Modus versteht der NLQ401 aber keinerlei Steuerzeichen. Der Code 0D hex zum Beispiel veranlaßt keinen Wagenrücklauf, sondern steuert nur die Ausgabe des Grafikzeichens mit dem Code 8D hex (0D+80).

Normalerweise kommen Grafikzeichen sehr selten vor. Es ist also sinnvoll, wenn sich der Grafik-Modus gleich nach der Ausgabe eines einzigen Grafikzeichens wieder von selbst abschaltet. Dadurch ist ein drittes »Phantomsteuerzeichen mit Gedächtnis« nötig: Beim Auftreten des Zeichens »tPE« im Text wird der Grafik-Modus aufgerufen und nach Ausgabe des nächsten Zeichens automatisch wieder abgeschaltet. Die Eingabe »tPEX« gibt damit das Grafikzeichen mit dem Code »Code(x)+80hex« auf den Drucker aus. »x« darf dabei ein beliebiges ASCII-Zeichen sein. Auf diese Weise ist also die Ausgabe der Grafikzeichen mit Codes zwischen A0 und FE hex möglich.

Für die Grafikzeichen mit Codes zwischen 81 und 9F hex und das mit dem Code FF hex steht kein normales ASCII-Zeichen mehr zur Verfügung. Der vorhandene ASCII-Satz muß also für diese Grafikzeichen

speziell interpretiert werden. Der Ausgaberroutine wird dies dadurch signalisiert, daß nach dem »tPE«-Zeichen ein weiteres »tPE«-Zeichen folgt. Dieses zweite Zeichen signalisiert, daß der als drittes Zeichen folgende ASCII-Wert anders übersetzt werden muß. Das Zeichen bekommt den neuen Code »Code(x)+40hex«. Die Steuerzeichenfolge »tPEtPEA« gibt somit das Grafikzeichen mit dem Code 81 hex aus.

Die Steuerzeichenfolge »tPEtPE @« würde nach dieser Regel dem Grafikzeichen mit dem Code 80 hex entsprechen. Beim NLQ401 gibt es dieses aber nicht, daher kann es mit dem bisher noch nicht druckbaren Grafikzeichen FF hex belegt werden.

Die neue Druckerroutine muß irgendwo im Wordstar-Programm abgelegt werden. Der für den Anwender reservierte Speicherbereich ist aber schon durch die erweiterte Bildschirmsteuerung belegt. Der einzige Platz ist jetzt der Anfang des Text-Speichers. Dieser beginnt normalerweise an der Adresse 7849 hex. Die Speicherplätze 035C und 035D hex enthalten diese Anfangsadresse. Durch Ändern auf einen anderen Wert wird der Beginn des Textspeichers geändert. Der Platz zwischen alter und neuer Startadresse kann vom Anwender beliebig benutzt werden. So auch für die Druckroutine.

Normalerweise wird vom gesamten Wordstar-Programm nur der Bereich bis zur Adresse 4014 hex auf Diskette gespeichert. Wird an der Adresse 7849 hex aber eine Ergänzung eingefügt, so muß der ganze dazwischenliegende Bereich mitgespeichert werden. Um das zu vermeiden, gibt es einen Trick: Die Routine wird zunächst einfach an die Adresse 4015 hex gelegt und Wordstar durch Einbauen einer »Umleitung« vor Programmstart veranlaßt, eine Kopieroutine aufzurufen, die die Druckroutine an ihre endgültige Position überträgt. Erst später wird das eigentliche Wordstar-Programm gestartet. Diese Kopieroutine wird ebenfalls hinter der Adresse 4015 hex abgelegt.

Für »Assembler-Unkundige« ist wieder ein Basic-Lader (Listing 3) vorgesehen, das die Routine »NLQ 401.COM« erzeugt. Gehen Sie genauso vor, wie bei den anderen Änderungen.

Das abgedruckte Programm funktioniert aber nur, wenn Sie einen von Markt & Technik für den CPC 464 angepaßten Wordstar mit deutschem Zeichensatz verwenden.

```
-D0274,0275
0274 01 19 ..
-D027B,027C
027B 01 1B ..
-D02B5,02B5
02B5 FF .
-L02C0,02C2
    02C0 JMP 02E0
    02C3
-D02CF,02D0
02CF 05 .
02D0 05 .
-L02E0,0327
02E0 MOV B,L
02E1 MOV C,A
02E2 LXI H,0328
02E5 DCR M
02E6 JNZ 02F3
02E9 INR M
02EA MOV A,C
02EB CPI 1F
02ED JNZ 02FB
02F0 MVI A,03
02F2 MOV M,A
02F3 LXI D,0009
02F6 LHLD 0001
02F9 DAD D
02FA PCHL
02FB CPI 1B
02FD JZ 0311
0300 CPI 19
0302 JNZ 02F3
0305 MOV A,B
0306 STA 032C
0309 LXI H,0329
030C MVI B,0F
030E JMP 031A
0311 MOV A,B
0312 STA 033F
0315 LXI H,033C
0318 MVI B,0E
031A MOV C,M
031B INX H
031C PUSH B
031D PUSH H
031E CALL 02F3
0321 POP H
0322 POP B
0323 DCR B
0324 JNZ 031A
0327 RET
0328
-D0328,0349
```

```
0328 01 1A 00 4F 02 18 1F 01 ...O....
0330 1A 20 1E 1A 00 4F 00 18 1F 50 18 20 1A 00 4F 02 ...O...P. ...O.
0340 18 1E 0B 20 1E 1A 00 4F 00 18 ... ..O...
-L034A,0352
```

```
034A LXI D,002A
034D CALL 02F6
0350 CMA
0351 RAR
0352 RET
0353
-D0718,0718
0718 FF
-L071A,071C
    071A JMP 034A
    071D
-D3PCD,4014
```

```
3PCD 3C 60 3C <<<
3FD0 66 66 3C 06 3C C6 38 6C C6 FE C6 C6 00 C6 38 6C ff<<.81.....81
3FE0 C6 C6 6C 38 00 C6 00 C6 C6 C6 C6 7C 00 10 38 6C ..18.....8.81
3FF0 C6 00 00 00 00 6C 00 78 0C 7C CC 76 00 6C 00 3C .....1.x.8.v.1.<
4000 66 66 66 3C 00 6C 00 00 C6 C6 C6 7E 00 1C 36 66 fff<.1.....B..6f
4010 6C 66 66 CC 00 1ff..
-G0
```

Listing 2. Patches in 8080-Assembler zur Eingabe mit dem DDT


```

-L0100,0102
    0100 JMP 4015
    0103
-D035C,035D
035C E6 78 .x
-D06B5,06B6
06B5 01 0F ..
-D06BA,06BB
06BA 01 12 ..
-D06BF,06C0
06BF 01 .
06C0 1C .
-D06C4,06C5
06C4 01 1D ..
-D06C9,06CC
06C9 03 1B 78 01 ...x
-D06CE,06D1
06CE 03 1B ..
06D0 78 00 x.
-D06D3,06D4
06D3 01 1E ..
-D06D8,06D9
06D8 01 1F ..
-D06DD,06E0
06DD 03 1B 57 ..W
06E0 01 .
-D06E2,06E5
06E2 03 1B 57 00 ..W
-L4015,401D

```

```

4015 LXI B,009D
4018 LXI D,7849
401B LXI H,4023
401E
-D401E,401F
401E ED B0 ..
-L4020,40A8
4020 JMP 3F03
4023 LXI D,78E3
4026 CPI 1C
4028 JNZ 785E
402B LDAX D
402C ANA A
402D JNZ 786B
4030 LXI H,78CF
4033 DCR A
4034 STAX D
4035 JMP 789D
4038 CPI 1D
403A JNZ 7873
403D LDAX D
403E ANA A
403F LXI H,78D3
4042 JZ 7859
4045 XRA A
4046 STAX D
4047 LXI H,78D7
404A JMP 789D
404D INX D
404E CPI 1E
4050 JNZ 788A
4053 LDAX D
4054 ANA A
4055 JNZ 7886
4058 DCR A
4059 STAX D
405A LXI H,78DA
405D JMP 789D
4060 MVI A,C0
4062 STAX D
4063 RET
4064 INX D
4065 CPI 1F
4067 JNZ 78AC
406A LDAX D
406B CMA
406C STAX D

```

```

406D LXI H,78E0
4070 ANA A
4071 JZ 789D
4074 LXI H,78DD
4077 MOV B,M
4078 INX H
4079 MOV A,M
407A PUSH H
407B PUSH B
407C CALL 78C6
407F POP B
4080 POP H
4081 DCR B
4082 JNZ 789E
4085 RET
4086 DCX D
4087 MOV C,A
4088 LDAX D
4089 ANA A
408A JZ 78C7
408D CPI C0
408F MOV A,C
4090 JNZ 78BF
4093 ANI 1F
4095 JNZ 78BF
4098 DCR A
4099 CALL 78C6
409C XRA A
409D STA 78E4
40A0 MOV C,A
40A1 LHL D,0001
40A4 LXI D,000C
40A7 DAD D
40A8 PCHL
40A9
-D40A9,40BF

```

Listing 5. Die Druckeroutine im Assembler-Code

```

40A9 03 1B 53 00 03 1B 53 ..S...S
40B0 01 02 1B 54 02 1B 3D 02 1B 37 02 1B 36 00 00 00 ...T...=.7..6...
-G0

```

```

10100,0102
d035c,035d
d06b5,06b6
d06ba,06bb
d06bf,06c0
d06c4,06c5
d06c9,06cc
d06ce,06d1
d06d3,06d4
d06d8,06d9
d06dd,06e0
d06e2,06e5
14015,401d
d401e,401f
14020,40a8
d40a9,40bf
g0

```

Listing 6. Diese Bytes müssen in »WS.COM« geändert werden

ten Sie noch einmal Ihr Install-Programm zur Hand nehmen und die Standardvoreinstellungen Ihren Wünschen anpassen.

Beispielsweise kann man im Kapitel »Terminal-Features« einstellen, daß sich Wordstar schon nach dem Einschalten in der Hilfsstufe 2 befindet. Man kann auch die Standardränder und Seitenhöhe selbst bestimmen. Wenn Sie sich die Mühe machen, diese Voreinstellungen vernünftig zu wählen, ersparen Sie sich am Anfang jedes Textes einen Haufen unnötiger Arbeit.

Wenn Sie wollen, dann können Sie auch die Einschaltmeldung (Druckername + Computernamen) frei wählen, so daß Sie gleich wissen, welche Version Sie vor sich haben. Wie wäre es mit einem Wordstar, das Sie mit Ihrem Namen begrüßt?

Beachten Sie bitte, daß manche Install-Versionen (nicht die des Markt & Technik-Verlages) die Ergänzungen, die Sie ab der Adresse 3F00 hex angehängt haben, löschen. Sollte also nach der Verwendung des Install-Programms Ihr Drucker nicht mehr funktionieren, dann müssen Sie eben das Programm »NLQ 401.COM« noch mal starten.

Noch ein paar Tips, wie man durch geschickte Bedienung das Programm beschleunigt. Wenn Sie

noch ein Anfänger sind, dann haben Sie es vielleicht noch gar nicht bemerkt. Der langwierige Neuaufbau des Bildschirms erfolgt nur, solange Wordstar nichts anderes zu tun hat. Geben Sie aber vor Ende des Bildaufbaus einen neuen Befehl, so wird dieser sofort beachtet. Oft kommt es vor, daß diese neuen Eingaben die Vervollständigung des Bildschirmaufbaus überflüssig machen. In den meisten Fällen ist Wordstar so schlau, das zu bemerken.

Wenn Sie beispielsweise schnell zehn Bildschirmseiten zurückblättern möchten, dann brauchen Sie nicht nach dem ersten »!R« Kommando 10 Sekunden zu warten, bis der Bildschirm wieder reorganisiert ist, und dann erst die nächste Taste drücken. Sie geben einfach zehnmal ganz schnell hintereinander »!R« ein. Die jetzt angezeigte Seite ist gleich die richtige. Die Zeiterparnis ist enorm.

Wenn Sie beim Schreiben so richtig in Fahrt sind, macht es gar nichts, wenn kurzzeitig in der oberen Bildschirmhälfte nur Unsinn steht. Solange der Cursor erscheint, können Sie ohne Sorge weitertippen. Sie werden sehen, daß der Text vervollständigt wird, sobald Sie eine Denkpause machen.

Dasselbe gilt übrigens für die vielen Hilfs- und sonstigen Menüs: Wenn Sie schon wissen, welchen Buchstaben die gewünschte Funktion hat, dann tippen Sie ihn sofort ein — vielleicht wird dann das zugehörige Menü gar nicht erst aufgebaut.

Generell können Sie sich merken: Sobald Sie wissen, was Sie als nächstes eingeben wollen, geben Sie sofort ein — gleichgültig, was gerade auf dem Bildschirm steht. Solange kein Zugriff auf die Disketten erfolgt, können Sie sicher sein, daß Ihre Eingaben schon irgendwo gespeichert werden, bis sie zur Ausführung kommen.

Wenn Sie einen längeren Text eingegeben haben und schnell ganz an den Anfang des Textes springen wollen, sollten Sie nicht »!QR« aufrufen, sondern statt dessen »!KS« — also den Text sichern. Als Folge der verwendeten Datenstrukturen erfolgt das Zurückblättern so wesentlich schneller und braucht außerdem weniger Diskettenspeicherplatz. Gleichzeitig ist Ihr Text gesichert.

Viele langdauernden Operationen können Sie mit »!U« abbrechen.

-D0DB5,0F5A

```

0DB5 59 3D 56 65 72 67 72 6F 65 73 73 Y=Vergroess
0DC0 65 72 74 20 20 20 65 69 6E 2F 61 75 73 20 20 20 ert ein/aus
0DD0 0E 53 3D 55 6E 74 65 72 73 74 72 65 69 63 68 2E .S=Unterstreich.
0DE0 20 65 69 6E 2F 61 75 73 20 20 42 3D 53 63 68 61 ein/aus B=Scha
0DF0 74 74 65 6E 73 63 68 72 2E 20 65 69 6E 2F 61 75 ttanschr. ein/au
0E00 73 20 20 20 20 44 3D 44 6F 70 70 61 6E 73 63 68 s D=Doppansch
0E10 6C 61 67 20 20 20 65 69 6E 2F 61 75 73 20 20 20 lag ein/aus
0E20 0E 41 3D 53 63 68 6D 61 6C 73 63 68 72 69 66 74 .A=Schmalschrift
0E30 20 65 69 6E 20 20 20 20 20 20 4E 3D 53 63 68 6D ein N=Schm
0E40 61 6C 73 63 68 72 69 74 20 20 61 75 73 20 20 20 alschrit aus
0E50 20 20 20 20 20 58 3D 44 75 72 63 68 73 74 72 65 X=Durchstre
0E60 69 63 68 65 6E 20 65 69 6E 2F 61 75 73 20 20 20 icken ein/aus
0E70 0E 4F 3D 66 65 73 74 65 72 20 4C 65 65 72 73 63 .O=fester Leersc
0E80 68 72 69 74 74 20 20 20 20 20 20 45 3D 47 72 61 70 hritt E=Grap
0E90 68 69 6B 7A 65 69 63 68 65 6E 20 66 6F 6C 67 74 hikzeichen folgt
0EA0 20 20 20 20 20 52 3D 44 65 75 74 2E 20 55 6D 6C R=Deut. Uml
0EB0 61 75 74 65 20 20 61 75 73 2F 65 69 6E 20 20 20 aute aus/ein
0EC0 0E 43 3D 44 72 75 63 6B 70 61 75 73 65 20 20 20 .C=Druckpause
0ED0 20 20 20 20 20 20 20 20 20 20 45 5E 50 45 3D 61 E*PE=a
0EE0 6E 64 65 72 65 20 47 72 61 70 68 69 6B 7A 65 69 ndere Graphikzei
0EF0 63 68 65 6E 20 45 4E 54 45 52 3D 5A 65 69 6C 65 chen ENTER=Zeile
0F00 20 75 65 62 65 72 64 72 75 63 6B 65 6E 20 20 20 ueberdrucken
0F10 0E 51 3D 53 63 68 6F 65 6E 73 63 68 72 69 66 74 .Q=Schoenschrift
0F20 20 65 69 6E 20 20 20 20 20 20 57 3D 53 63 68 6F ein W=Scho
0F30 65 6E 73 63 68 72 69 66 74 20 61 75 73 20 20 20 enschrift aus
0F40 20 20 20 20 20 4C 65 65 72 73 63 68 72 69 74 74 Leerschrift
0F50 3D 41 62 62 72 65 63 68 65 6E 20 =Abbrechen
-g0
    
```

Listing 7. Auch die Einschaltmeldung kann geändert werden

Im Zweifelsfall: einfach ausprobieren.

Stört es Sie, daß Sie vor dem Ausdrucken eines Textes erst einige Fragen beantworten müssen, obwohl Sie sowieso die Standardeinstellungen wählen? Besonders, wenn auch noch ein langes Inhaltsverzeichnis angezeigt wird, dauert das »ewig«.

Sie können aber auch sofort, nachdem Sie den Programmnamen eingegeben haben, »Esc« drücken. In diesem Fall beginnt der Drucker sofort mit seiner Arbeit. Wenn Ihre Tastatur nicht richtig belegt ist, erscheint beim Drücken der Taste »Esc« allerdings nur ein kleines »ö«. In diesem Fall können Sie auch »!Ä ([] « drücken. (Helmut Tischer)

»Hilfe« in Sicht



Wenn Sie eine Datei versehentlich gelöscht haben, dann ist guter Rat teuer. Mit einem Diskettenmonitor ist das Problem schnell behoben.

Kennen Sie das Problem auch? Das Kommando »ERA Name.Zusatz« ist schnell eingegeben. Doch das Programm wieder zu retten, ist schwierig. Auf der CP/M-Diskette von Schneider suchen Sie vergeblich nach einem Programm zur Rekonstruktion der versehentlich gelöschten Quelldateien. Die BAK-Dateien sind wegen der begrenzten Speicherkapazität sicher schon gelöscht.

»Help« gestattet den direkten Zugriff auf jede Spur der Diskette. Die Anzeige erfolgt gleichzeitig für 256 Bytes im Hex- und ASCII-Dump. Änderungen können sowohl im Hex- als auch im ASCII-Teil vorgenommen werden (Wechsel durch TAB-Taste).

Eine Spur wird jeweils am Stück eingelesen und im RAM zwischengespeichert. Dadurch erfolgt der Zugriff auf jedes Byte einer Spur sehr schnell und übersichtlich. Jede Datei auf der Diskette ist damit leicht zu ändern, auch das Betriebssystem (zum Beispiel durch Übersetzen der Fehlermeldungen). Die Anwendung von Help ist aber nicht ohne Risiko. Benutzen Sie deshalb nur Duplikate, um nach eventuellen Fehlern auf die ursprünglichen Daten zurückgreifen zu können.

Unsere zirka 3 KByte lange Routine wird im CP/M-Modus mit »Help« aufgerufen. Die zu untersuchende Spur wird direkt eingegeben. Mit den Cursorstasten kann man innerhalb des Anzeigefeldes beliebig hin- und herspringen. Die weitere Tastenbelegung befindet sich in der Fußleiste der Maske. Im einzelnen sind das:

- Shift mit Cursor nach oben oder unten — Wechsel des Anzeigebereiches innerhalb der Spur
- Tab — Wechsel des Cursors in Hex- oder ASCII-Feld (wechselnd)
- Copy — Verlassen der Spur mit Speichern der Änderungen
- Ctrl-C — Verlassen der Spur ohne Speichern der Änderungen
- Ctrl-C (im Feld Spur) — Programmende

Arbeiten mit Help

Um eine gelöschte Datei wieder aufzurufen, geht man am einfachsten wie folgt vor:

1. Help aufrufen
2. Spur 2 betrachten:

Im Anzeigefeld erscheint jetzt die Hex- und ASCII-Darstellung des 1. Teiles des Directory. Es werden je Anzeigebereich Einträge dargestellt. Gelöschte Files werden im Byte, das direkt vor dem Dateinamen steht, mit dem Wert E5 hex markiert. Ersetzt man diesen durch 00 hex, so ist der Löscheintrag rückgängig gemacht.

3. Jetzt ist noch das Sichern der geänderten Spur mit »Copy« erforderlich und das Programm ist gerettet.

Mit einer Einschränkung: Die Rückbesserung des Inhaltsverzeichnisses der Diskette ist nur erfolgreich, wenn nicht schon andere Zugriffe auf die Diskette erfolgt sind.

(Heinz-Dieter Lange)

Die Quelltexte sind mit dem Assembler M80 aufgenommen worden. Gegenüber ASM.COM weicht das Listing deshalb in folgenden Punkten ab:

— Label müssen beim M80 mit einem Doppelpunkt enden.

— Die getrennten Listings 2 und 3 sind zu einer gemeinsamen ASM-Datei zusammenzufassen.

— Verwendete Macros sind gegebenenfalls aufzulösen, das heißt die Zeilen zwischen »Macro« und »Endm« sind dort einzufügen, wo jeweils der Name des Macros als OP-Code aufgerufen wurde.

— Der Pseudo-Op-Code »IF2« wird nur im M80 verwendet. Dieser Teil muß beim ASM ausgelassen werden.

— Die Umschaltung auf Z80-Mnemonics muß durch entsprechende DB-Anweisungen (zum Beispiel für LDIR = DB OEDH DB OBOH) ersetzt werden.

— Comment-Zeilen können ausgelassen werden.

»HELP.COM« wurde in zwei Dateien erstellt und mit dem L80 gebunden. Für ASM muß diese Quelldatei zusammengefaßt werden. Dadurch entfallen die Hinweise Extn/Public. Das gilt ebenso für die Sonderform der Labeldarstellung mit zwei Doppelpunkten, die für den M80 die Public-Anweisung ersetzt, und für den Aufruf »Name # #«, der die Sonderform für den Aufruf eines extern definierten Label bedeutet.

Wer den Basic-Lader abtippen will, dem sei gesagt, daß das Programm auf Diskette die benötigte »HELP.COM«-Datei erzeugt (beziehungsweise »LPRINT.COM« der folgenden Routine).

```

10 ***** [9FCC]
20 * (c) Heinz-Dieter Lange * [FD80]
30 * Kamper Weg 90, 2210 Itzehoe * [B5E6]
40 ***** [E9D2]
50 'Basic-Lader fuer die Datei 'HELP.COM [92E4]
60 OPENOUT"HELP.COM":zeile=10000 [2B56]
70 pruef=0 [192C]
80 FOR j=i TO i+7:READ d$:IF d$="Ende" T
HEN GOTO 110 ELSE d=VAL("&"+d$):PRINT
#9,CHR$(d):pruef=pruef+d:NEXT [04E2]
90 READ p:IF pruef<>p THEN GOTO 140 [620E]
100 zeile=zeile+10:GOTO 70 [7E4C]
110 PRINT"Alles O.K." [9D70]
120 CLOSEOUT [0C42]
130 END [B416]
140 PRINT"Pruefsummenfehler in Zeile"zei
le [9A78]
150 CLOSEOUT [1048]
160 END [EF1C]
170 DATA 0E,0C,CD,21,09,21,02,06, 314 [51E0]
180 DATA CD,75,BB,0E,96,CD,21,09, 920 [7D78]
190 DATA 06,44,CD,B6,06,0E,9C,CD, 842 [8A7A]
200 DATA 21,09,21,03,06,06,12,CD, 313 [0D96]
210 DATA C0,06,21,03,48,06,12,CD, 538 [68E4]
220 DATA C0,06,21,05,38,06,11,CD, 520 [15C0]
230 DATA C0,06,21,15,06,CD,75,BB, 767 [CA2A]
240 DATA 0E,93,CD,21,09,06,44,CD, 687 [253C]
250 DATA B6,06,0E,99,CD,21,09,21, 635 [9214]
260 DATA 04,07,CD,75,BB,06,44,CD, 799 [3C64]
270 DATA B6,06,11,D0,06,21,01,10, 469 [40AA]
280 DATA CD,F9,08,11,04,07,21,03, 526 [C2E4]
290 DATA 08,CD,F9,08,11,47,07,21, 598 [F510]
300 DATA 05,04,CD,F9,08,11,83,07, 674 [5306]
310 DATA 21,16,29,CD,F9,08,AF,CD, 938 [EDB6]
320 DATA 8D,08,22,47,09,5E,23,56, 526 [3F8B]
330 DATA EB,22,4A,09,11,0A,00,2A, 421 [4DEC]
340 DATA 47,09,19,5E,23,56,EB,7E, 681 [DD36]
350 DATA 32,49,09,A7,1F,32,4C,09, 465 [5C00]
360 DATA 21,17,0A,CD,75,BB,06,03, 584 [2320]
370 DATA CD,02,05,FE,03,CA,AE,04, 849 [2F86]
380 DATA CD,10,06,FE,1C,CA,A0,06, 877 [C086]
390 DATA FE,28,D2,A0,06,32,43,09, 796 [4D26]
400 DATA CD,D4,05,AF,32,45,09,CD, 930 [EE66]
410 DATA 56,05,AF,CD,83,06,2A,3D, 711 [ED42]
420 DATA 09,22,3F,09,7E,CD,4D,06, 529 [CD40]
430 DATA 21,53,05,72,23,73,2B,CD, 633 [F1E0]
440 DATA 0A,09,FE,F0,CA,80,02,FE,1147 [EEBA]
450 DATA F1,CA,F1,02,FE,F4,CA,68,1490 [67D8]
460 DATA 04,FE,F5,CA,79,04,FE,03,1087 [03A8]
470 DATA CA,A0,06,FE,E0,CA,C1,04,1245 [15B0]
480 DATA FE,09,CA,39,03,FE,F2,CA,1223 [2ED8]
490 DATA 7B,02,FE,F3,CA,76,02,FE,1198 [21CE]
500 DATA 30,DA,D7,01,FE,3A,DA,1B,1039 [90AE]
510 DATA 02,FE,41,DA,D7,01,FE,47,1080 [9AB4]

```

```

520 DATA D2,D7,01,77,CD,F6,04,23,1035 [984C]
530 DATA CA,D7,01,CD,31,06,2A,3F, 783 [EC62]
540 DATA 09,77,23,22,3F,09,57,CD, 561 [E608]
550 DATA 78,BB,E5,2A,3B,09,E5,CD,1080 [C6B8]
560 DATA 75,BB,7A,E6,7F,FE,20,D2,1279 [60D6]
570 DATA 44,02,3E,2E,CD,12,09,E1, 635 [5C2C]
580 DATA 24,22,3B,09,E1,CD,75,BB, 872 [995A]
590 DATA 3E,09,CD,12,09,3A,46,09, 440 [6C16]
600 DATA FE,0F,C2,6C,02,3A,42,09, 706 [BA42]
610 DATA FE,0F,CA,33,03,3C,CD,83, 921 [CE86]
620 DATA 06,C3,70,02,3C,32,46,09, 504 [3DBC]
630 DATA 2A,3F,09,C3,CC,01,3E,09, 585 [F050]
640 DATA C3,1C,02,3A,41,09,B7,C2, 734 [551C]
650 DATA 8A,02,3E,08,CD,F6,04,C3, 860 [7856]
660 DATA D6,01,3A,46,09,B7,CA,73, 852 [763C]
670 DATA 04,3D,32,46,09,2A,3F,09, 308 [44F4]
680 DATA 7E,CD,69,06,11,BB,04,CD, 855 [2F86]
690 DATA FC,08,2A,3B,09,25,22,3B, 500 [DE20]
700 DATA 09,2A,3F,09,2B,C3,C9,01, 563 [3428]
710 DATA 3A,42,09,B7,CA,73,04,3D, 698 [203C]
720 DATA 32,42,09,3A,41,09,B7,C2, 634 [6BE8]
730 DATA C7,02,3E,08,CD,F6,04,2A, 768 [B55E]
740 DATA 3F,09,7E,11,F0,FF,19,22, 769 [1356]
750 DATA 3F,09,CD,69,06,11,BE,04, 599 [6550]
760 DATA CD,FC,08,3E,0B,CD,12,09, 770 [CA94]
770 DATA 3A,39,09,3D,32,39,09,3A, 359 [E30E]
780 DATA 3B,09,3D,32,3B,09,C3,70, 554 [5318]
790 DATA 02,3A,42,09,FE,0F,CA,73, 721 [1550]
800 DATA 04,3C,32,42,09,3A,41,09, 321 [53B0]
810 DATA B7,C2,09,03,3E,08,CD,F6, 910 [F354]
820 DATA 04,2A,3F,09,7E,11,10,00, 277 [82D8]
830 DATA 19,22,3F,09,CD,69,06,11, 464 [8B02]
840 DATA BE,04,CD,FC,08,3E,0A,CD, 936 [76D6]
850 DATA 12,09,3A,39,09,3C,32,39, 318 [10DE]
860 DATA 09,3A,3B,09,3C,32,3B,09, 313 [C306]
870 DATA C3,70,02,CD,B6,04,C3,C2,1089 [7070]
880 DATA 01,CD,78,BB,22,39,09,3A, 671 [F240]
890 DATA 41,09,B7,C2,52,03,3E,08, 606 [6CFA]
900 DATA CD,F6,04,2A,39,09,25,22, 634 [FB14]
910 DATA 39,09,2A,3B,09,CD,75,BB, 685 [C068]
920 DATA CD,0A,09,FE,F0,CA,17,04, 947 [9796]
930 DATA FE,F1,CA,3F,04,FE,F4,CA,1464 [7122]
940 DATA B7,04,FE,F5,CA,A0,04,FE,1258 [65C8]
950 DATA 03,CA,A0,06,FE,E0,CA,C1,1244 [8AB2]
960 DATA 04,FE,09,CA,D5,03,FE,F2,1181 [62BE]
970 DATA CA,F0,03,FE,F3,CA,DE,03,1369 [FEFC]
980 DATA FE,20,DA,58,03,FE,80,D2,1187 [54AE]
990 DATA 58,03,CD,12,09,2A,3F,09, 437 [CB24]
1000 DATA 77,23,22,3F,09,57,2A,39, 446 [7E3A]
1010 DATA 09,E5,CD,75,BB,7A,CD,69,1179 [EB2C]
1020 DATA 06,E1,3A,46,09,FE,0F,C2, 831 [80AA]
1030 DATA C1,03,3A,42,09,FE,0F,CA, 800 [48AA]
1040 DATA 33,03,3C,CD,83,06,C3,52, 733 [8760]
1050 DATA 03,3C,32,46,09,24,24,24, 300 [8EEC]
1060 DATA 22,39,09,2A,3B,09,24,22, 280 [2C1E]
1070 DATA 3B,09,C3,55,03,2A,39,09, 459 [AF5A]
1080 DATA CD,75,BB,C3,70,02,3E,09, 889 [F2C6]
1090 DATA CD,12,09,2A,3F,09,23,22, 415 [7A5E]
1100 DATA 3F,09,2A,39,09,C3,AA,03, 548 [E284]
1110 DATA 3A,46,09,B7,CA,9A,04,3D, 741 [A7A4]
1120 DATA 32,46,09,3E,08,CD,12,09, 431 [6342]
1130 DATA 2A,3F,09,2B,22,3F,09,2A, 305 [9272]
1140 DATA 39,09,25,25,25,22,39,09, 277 [E206]
1150 DATA 2A,3B,09,25,C3,CF,03,3A, 610 [B58C]
1160 DATA 42,09,B7,CA,9A,04,3D,32, 729 [7094]
1170 DATA 42,09,2A,3F,09,11,F0,FF, 701 [B786]
1180 DATA 19,22,3F,09,3E,0B,CD,12, 427 [2F8A]
1190 DATA 09,3A,39,09,3D,32,39,09, 310 [083B]
1200 DATA 2A,3B,09,2D,C3,CF,03,3A, 618 [D928]
1210 DATA 42,09,FE,0F,CA,9A,04,3C, 764 [AFCE]
1220 DATA 32,42,09,2A,3F,09,11,10, 272 [D00C]
1230 DATA 00,19,22,3F,09,3E,0A,CD, 408 [2978]
1240 DATA 12,09,3A,39,09,3C,32,39, 318 [8832]
1250 DATA 09,2A,3B,09,2C,C3,CF,03, 568 [89AC]
1260 DATA 3A,45,09,B7,CA,73,04,3D, 701 [1B86]
1270 DATA C3,BC,01,CD,B6,04,C3,D7,1185 [6106]
1280 DATA 01,3A,45,09,3C,21,4C,09, 315 [303C]
1290 DATA BE,CA,73,04,C3,BC,01,3A, 953 [DDE0]
1300 DATA 45,09,B7,CA,9A,04,3D,FS, 927 [0DBE]
1310 DATA 2A,39,09,CD,75,BB,F1,C3,1053 [28E8]
1320 DATA BC,01,CD,B6,04,C3,58,03, 866 [D1A6]
1330 DATA 3A,45,09,3C,21,4C,09,BE, 504 [9880]
1340 DATA CA,9A,04,C3,8F,04,0E,0D, 729 [33CE]
1350 DATA CD,05,00,C3,00,00,3E,07, 474 [EB46]
1360 DATA C3,12,09,08,08,08,08,08, 262 [C710]
1370 DATA 24,3A,43,09,CD,C6,08,AF, 756 [CFC0]
1380 DATA 32,44,09,21,4D,09,22,3F, 343 [572E]
1390 DATA 09,CD,A3,08,CD,EF,08,87,1020 [DC12]
1400 DATA C2,EB,05,3A,44,09,3C,21, 662 [8578]
1410 DATA 49,09,BE,CA,A0,06,32,44, 758 [5092]
1420 DATA 09,2A,3F,09,11,80,00,19, 293 [5822]
1430 DATA 22,3F,09,C3,D1,04,CD,12, 737 [4E84]
1440 DATA 09,3A,41,09,2F,32,41,09, 312 [601E]
1450 DATA B7,C9,21,53,05,48,CD,0A, 792 [2E9E]
1460 DATA 09,FE,03,C8,FE,08,CA,3A, 988 [9C16]
1470 DATA 05,FE,0D,CA,37,05,FE,30, 836 [A2D8]
1480 DATA DA,06,05,FE,7F,CA,3A,05, 875 [AF00]
1490 DATA FE,3A,D2,06,05,05,C2,2F, 779 [5DC2]
1500 DATA 05,04,3E,07,C3,31,05,77, 446 [3D26]

```

Listing 1. Der Basic-Lader für »Help«

```

1510 DATA 23,CD,12,09,C3,06,05,36, 527 [3842]
1520 DATA 1C,C9,79,B8,CA,06,05,04, 751 [A198]
1530 DATA 2B,3E,08,CD,12,09,3E,20, 439 [5A88]
1540 DATA CD,12,09,3E,08,CD,12,09, 534 [4B8C]
1550 DATA C3,06,05,00,00,00,21,05, 244 [D4D0]
1560 DATA 08,22,39,09,CD,75,BB,AF, 792 [0DCE]
1570 DATA 32,46,09,21,4D,09,3A,45, 375 [273E]
1580 DATA 09,47,7C,80,67,22,3D,09, 539 [6B60]
1590 DATA 22,3F,09,AF,4F,21,27,09, 441 [5B7C]
1600 DATA 22,37,09,2A,3F,09,7E,47, 409 [B662]
1610 DATA E5,2A,37,09,70,23,22,37, 571 [1932]
1620 DATA 09,E1,78,CD,69,06,23,3E, 767 [C99C]
1630 DATA 20,CD,12,09,0C,79,FE,10, 667 [0896]
1640 DATA C2,7E,05,22,3F,09,3E,09, 502 [7C72]
1650 DATA CD,12,09,3E,09,CD,12,09, 535 [3294]
1660 DATA 21,27,09,7E,E6,7F,FE,20, 850 [82B4]
1670 DATA D2,85,05,3E,2E,CD,12,09, 736 [5E88]
1680 DATA 23,0D,C2,AB,05,3A,46,09, 555 [6A8A]
1690 DATA FE,0F,C8,3C,32,46,09,2A, 700 [33BC]
1700 DATA 39,09,2C,22,39,09,CD,75, 532 [0F64]
1710 DATA BB,C3,73,05,CD,C6,08,AF, 1088 [DF18]
1720 DATA 32,44,09,21,4D,09,22,3F, 343 [112A]
1730 DATA 09,CD,A3,08,CD,E7,08,B7, 1012 [01F2]
1740 DATA CA,F7,05,11,7D,08,21,18, 661 [4074]
1750 DATA 05,CD,F9,08,C3,00,00,3A, 720 [837A]
1760 DATA 44,09,3C,21,49,09,BE,C8, 642 [5C8C]
1770 DATA 32,44,09,2A,3F,09,11,80, 386 [3E3E]
1780 DATA 00,19,22,3F,09,C3,E1,05, 556 [5156]
1790 DATA 21,53,05,7E,FE,1C,C8,23, 764 [C6AE]
1800 DATA 7E,FE,1C,C2,23,06,2B,7E, 812 [B1D2]
1810 DATA E6,0F,C9,E6,0F,57,2B,7E, 947 [2A04]
1820 DATA E6,0F,87,47,87,87,80,82, 979 [9A92]
1830 DATA C9,11,54,05,1A,CD,45,06, 613 [ED68]
1840 DATA 47,1B,1A,CD,45,06,A7,17, 594 [A69E]
1850 DATA 17,17,17,8D,C9,D6,30,FE, 962 [55AE]
1860 DATA 0A,D8,D6,07,C9,F5,0F,0F, 923 [05F0]
1870 DATA 0F,0F,CD,5C,06,57,F1,CD, 866 [DB12]
1880 DATA 5C,06,5F,C9,E6,0F,FE,0A, 903 [7012]
1890 DATA D2,66,06,C6,30,C9,C6,37, 1018 [11B2]
1900 DATA C9,F5,0F,0F,0F,CD,72, 825 [7C04]
1910 DATA 06,F1,E6,0F,FE,0A,D2,7E, 1092 [201A]
1920 DATA 06,C6,30,C3,12,09,C6,37, 727 [5E5A]
1930 DATA C3,12,09,32,42,09,C6,05, 550 [E62A]
1940 DATA 6F,5F,16,08,26,3A,AF,32, 557 [6EAE]
1950 DATA 46,09,2F,32,41,09,22,38, 343 [F134]
1960 DATA 09,EB,22,39,09,C3,75,BB, 843 [4DB0]
1970 DATA 21,17,0A,CD,49,06,C3,98, 793 [6FB0]
1980 DATA 01,CD,75,BB,0E,20,CD,21, 794 [34D2]
1990 DATA 09,0E,20,C3,21,09,3E,9A, 508 [747C]
2000 DATA CD,12,09,05,C2,86,06,C9, 820 [AB7A]
2010 DATA E5,CD,75,BB,E1,3E,95,CD, 1379 [864A]
2020 DATA 12,09,2C,05,C2,C0,06,C9, 669 [E266]
2030 DATA 48,20,45,20,4C,20,50,20, 425 [88E6]
2040 DATA 20,2F,20,43,50,43,20,36, 411 [88F0]
2050 DATA 36,34,20,20,20,28,31,2E, 337 [07F2]
2060 DATA 30,31,29,20,20,20,20,298 [0FC2]
2070 DATA 20,20,20,20,20,20,20,256 [66A0]
2080 DATA 20,20,20,20,20,20,20,38280 [76AE]
2090 DATA 2E,38,35,24,20,30,20,2D, 361 [9740]
2100 DATA 31,20,2D,32,20,2D,33,20, 336 [5FF4]
2110 DATA 2D,34,20,2D,35,20,2D,36, 358 [D638]
2120 DATA 20,2D,37,20,2D,38,20,2D, 342 [822A]
2130 DATA 39,20,2D,41,20,2D,42,20, 374 [A90E]
2140 DATA 2D,43,20,2D,44,20,2D,45, 403 [212C]
2150 DATA 20,2D,46,20,95,20,30,31, 457 [6FFA]
2160 DATA 32,33,34,35,36,37,38,39, 428 [E50A]
2170 DATA 41,42,43,44,45,46,24,30, 489 [29FE]
2180 DATA 30,0D,0A,20,20,20,31,30, 264 [54EC]
2190 DATA 0D,0A,20,20,20,32,30,0D, 230 [7D04]
2200 DATA 0A,20,20,20,33,30,0D,0A, 228 [5CFE]
2210 DATA 20,20,20,34,30,0D,0A,20, 251 [F3DC]
2220 DATA 20,20,35,30,0D,0A,20,20, 252 [2BE2]
2230 DATA 20,36,30,0D,0A,20,20,20, 253 [31E8]
2240 DATA 37,30,0D,0A,20,20,20,38, 278 [5E0C]
2250 DATA 30,0D,0A,20,20,20,39,30, 272 [7AF6]
2260 DATA 0D,0A,20,20,20,41,30,0D, 245 [160C]
2270 DATA 0A,20,20,20,42,30,0D,0A, 243 [2D06]
2280 DATA 20,20,20,43,30,0D,0A,20, 266 [24F6]
2290 DATA 20,20,44,30,0D,0A,20,20, 267 [6CFC]
2300 DATA 20,45,30,0D,0A,20,20,20, 268 [CEF0]
2310 DATA 46,30,24,53,65,6B,74,6F, 672 [9B48]
2320 DATA 72,20,20,76,6F,72,20,2F, 600 [2020]
2330 DATA 2D,20,7A,75,72,75,65,63, 747 [5152]
2340 DATA 6B,3A,20,18,53,68,69,66, 615 [44AE]
2350 DATA 74,20,50,66,65,69,6C,18, 668 [2144]
2360 DATA 0D,0A,20,20,20,18,53,70, 338 [300C]
2370 DATA 75,72,3A,18,20,20,20,20, 441 [B1F2]
2380 DATA 28,41,62,62,72,75,63,68, 735 [6122]
2390 DATA 3A,20,43,54,52,4C,20,43, 511 [AF40]
2400 DATA 29,20,20,20,20,20,20,265 [82AC]
2410 DATA 20,20,20,20,57,65,63,68, 519 [56EC]
2420 DATA 73,65,6C,20,48,45,58,20, 617 [5228]
2430 DATA 3C,2D,3E,20,41,53,43,49, 487 [AA5A]
2440 DATA 49,2D,43,75,72,73,6F,72, 756 [4B68]
2450 DATA 3A,20,18,20,54,41,42,20, 393 [94F6]
2460 DATA 18,0D,0A,20,20,20,20,20, 207 [53EE]
2470 DATA 20,20,20,20,20,20,20,256 [3EAB]
2480 DATA 20,20,6E,65,75,65,20,53, 608 [9922]
2490 DATA 70,75,72,20,28,6F,68,6E, 740 [FC62]
2500 DATA 65,20,41,62,6C,61,67,65, 705 [771E]
2510 DATA 29,3A,20,18,43,54,52,4C, 464 [A130]
2520 DATA 2D,43,18,20,20,20,20,296 [87E8]
2530 DATA 20,20,20,20,28,18,20,6D, 333 [62E8]
2540 DATA 69,74,20,18,20,41,62,6C, 580 [431E]
2550 DATA 61,67,65,29,3A,20,18,43, 523 [8520]
2560 DATA 4F,50,59,18,24,07,2A,2A, 399 [056A]
2570 DATA 2A,20,44,69,73,68,2E,20, 547 [3E5E]
2580 DATA 45,69,6E,67,61,62,65,2D, 728 [7174]
2590 DATA 2F,41,75,73,67,61,62,65, 743 [B144]
2600 DATA 66,65,68,6C,65,72,20,2A, 704 [3448]
2610 DATA 2A,2A,2A,3F,09,E5,C1, 656 [5DA6]
2620 DATA CD,DF,08,3A,44,09,4F,06, 656 [39C6]
2630 DATA 00,2A,4A,09,EB,CD,D1,08, 782 [85C4]
2640 DATA 44,4D,C3,D7,08,4F,2A,01, 685 [BBA0]
2650 DATA 00,11,18,00,19,E9,4F,06, 384 [FA32]
2660 DATA 00,2A,01,00,11,18,00,19, 112 [7EDE]
2670 DATA E9,2A,01,00,2E,30,E9,2A, 645 [7A8A]
2680 DATA 01,00,11,1E,00,19,E9,2A, 348 [463A]
2690 DATA 01,00,11,21,00,19,E9,2A, 351 [100A]
2700 DATA 01,00,11,24,00,19,E9,0E, 326 [7508]
2710 DATA 02,2A,01,00,11,27,00,19, 126 [C5D0]
2720 DATA E9,CD,75,BB,1A,FE,24,C8, 1258 [4850]
2730 DATA 4F,D5,CD,21,09,D1,13,C3, 962 [1288]
2740 DATA FC,08,C5,E5,CD,1B,09,E1, 1152 [7418]
2750 DATA C1,C9,C5,E5,4F,CD,21,09, 1146 [4F06]
2760 DATA E1,C1,C9,2A,01,00,2E,09, 717 [8784]
2770 DATA E9,2A,01,00,2E,0C,E9,00, 567 [648C]
2780 DATA 00,00,00,00,00,00,00,00, 0 [2436]
2790 DATA 00,00,00,00,00,00,00,00, 0 [3538]
2800 DATA 00,00,00,00,00,00,00,00, 0 [3E28]
2810 DATA 00,00,00,00,00,00,00,00, 0 [2D2A]
2820 DATA 00,00,00,00,00,00,00,19, 25 [036E]
2830 DATA E9,2A,01,00,2E,30,E9,2A, 645 [EA86]
2840 DATA 01,00,11,1E,00,19,E9,2A, 348 [5E36]
2850 DATA 01,00,11,21,00,19,E9,2A, 351 [4B06]
2860 DATA 01,00,11,24,00,19,E9,0E, 326 [9516]
2870 DATA 02,2A,01,00,11,27,00,19, 126 [DBDE]
2880 DATA E9,CD,75,BB,1A,FE,24,C8, 1258 [405E]
2890 DATA 01,01,01,01,01,01,01,01, 8 [A95A]
2900 DATA Ende [2442]
    
```

Listing 1. Der Basic-Lader für «Help» (Schluß)

```

1: TITLE Utility-Programm 'H E L P' zur Korrektur von CP/M-Disketten
2: SUBTTL --- HELP.MAC --- 18.09.1985
3:
4:
5: | (c) by Heinz-Dieter LANGE, Itzehoe |
6: | in August 1985 |
7:
8:
9: --- Adressen aus/für HKOUT.REL ---
10: PUBLIC SETCRS
11: EXTRN HELPRUE
12: EXTRN BSRRS
13: EXTRN EINGB
14: EXTRN EG_PUF
15: EXTRN PRITSEC
16: EXTRN RD_IRQ
17: EXTRN FHLR
18: EXTRN HEX
19: EXTRN PRITHEX
20: EXTRN SETZL
21: EXTRN SPRRET
22: EXTRN HURSTR
23: EXTRN VRTSTR
24: EXTRN TXT1
25: EXTRN TXT2
26: EXTRN TXT3
27: EXTRN TXT4
28: EXTRN SELDRV
29: EXTRN PRITSTR
30: EXTRN CONIN
31: EXTRN CONOUT
32: EXTRN DCONIN
33: EXTRN DCONUT
34: EXTRN HSCPUF
35: EXTRN APUPAD
36: EXTRN CRSHXC
37: EXTRN CRSHSC
38: EXTRN PUFADS
39: EXTRN PUFADR
    
```

```

40: EXTRN OUTFLG
41: EXTRN SPUR
42: EXTRN RECURD
43: EXTRN SEKTOR
44: EXTRN ZONE
45: EXTRN ZEILE
46: EXTRN DSKDPH
47: EXTRN RECP1
48: EXTRN TRKSTB
49: EXTRN SECP1
50: EXTRN PUFFER
51:
52: TAB EQU 9
53: EEZ EQU 1CH ;MbschluBmarke in EG_PUF
54: PF_UP EQU 0F0H ;Freil nach oben
55: PF_DWN EQU 0F1H ;Freil nach unten
56: SHF_UP EQU 0F4H ;Shift Freil nach oben
57: SHF_DWN EQU 0F5H ;Shift Freil nach unten
58: COPY EQU 0E0H
59:
60: LKSUNT EQU 93H ;Winkel links unten
61: LKS0BN EQU 96H ;Winkel links oben
62: REUNT EQU 99H ;Winkel rechts unten
63: RE0BN EQU 9CH ;Winkel rechts oben
64:
65: NKSPUR EQU 39
66:
67: --- Betriebsdaten-Vektoren ---
68: SETCRS EQU 08B75H
69: GETCRS EQU 08B78H
70:
71: RSEG
72: ORG 100H
    
```

Listing 2. Freunde des 8080-Assembler-Codes geben das Programm direkt ein (Teil 1)

```

73:
74: ;----> Bildschirm löschen
75: MVI C,0CH
76: CALL DCONUT
77:
78: ;----> Fenster für Sektoreinblendung setzen
79: LXI H,682H
80: CALL SETCRS ;oben links
81: MVI C,LKS08H
82: CALL DCONUT ;Ecke links oben setzen
83:
84: MVI B,68
85: CALL HORSTR ;68H horizontalen Strich ausgeben
86:
87: MVI C,REUBEH
88: CALL DCONUT ;Ecke rechts oben setzen
89:
90: LXI H,683H
91: MVI B,18
92: CALL VRTSTR ;18 Zeilen senkrechten Strich ausgeben (links)
93:
94: LXI H,48B3H
95: MVI B,18
96: CALL VRTSTR ;wie vor (rechts)
97:
98: LXI H,3805H
99: MVI B,17
100: CALL VRTSTR ;Trennlinie zwischen HEX und ASCII
101:
102: LXI H,615H
103: CALL SETCRS ;Ecke unten links
104: MVI C,LKSUNT
105: CALL DCONUT
106:
107: MVI B,68
108: CALL HORSTR ;untere Linie ziehen
109:
110: MVI C,REUNT
111: CALL DCONUT ;Ecke unten rechts
112:
113: ;----> Unterteilungsstrich
114: LXI H,704H
115: CALL SETCRS
116: MVI B,68
117: CALL HORSTR
118:
119: ;----> Kopfzeile (Überschrift)
120: LXI D,TXT1
121: LXI H,1001H
122: CALL PR1STR
123:
124: ;----> obere Leiste für Sektordarstellung ausgeben
125: LXI D,TXT2
126: LXI H,893H
127: CALL PR1STR
128:
129: ;----> seitliche Leiste
130: LXI D,TXT3
131: LXI H,405H
132: CALL PR1STR
133:
134: ;----> Textzeile unten
135: LXI D,TXT4
136: LXI H,2916H
137: CALL PR1STR
138:
139: ;----> Systemwerte zum Laufwerk holen
140: XRA A
141: CALL SELDRV ;Laufwerk auswählen
142: SHLD DSKDPH
143: MOV E,M
144: INX H
145: MOV D,M
146: XCHG
147: SHLD TRSTRB ;Adr. Übersetzungstabelle ablegen
148:
149: LXI D,10
150: LHL DSKDPH ;10. Pos. in DPH
151: DRD D
152: MOV E,H
153: INX H
154: MOV D,M
155: XCHG ;HL = Anfang Parameterliste
156: MOV A,M
157: STA RECP1 ;Zahl der Records je Spur (je 128 Bytes)
158: RAR A
159: RAR ;1/2
160: STA SECP1 ;256 Byte-Einheiten
161:
162: ;----> Spur eingeben
163: EINSPR: LXI H,0A17H
164: CALL SETCRS
165: MVI B,3 ;2 Stellen
166: CALL EINGB
167: CPI 3
168: JZ RESET
169:
170: CALL HEX
171: CPI EEZ ;keine Eingabe?
172: JZ SPRRET
173: CPI RKSPPUR1
174: JNC SPRRET ;Eingabe zu groß
175: STA SPUR
176:
177: CALL RD,TRC
178: XRA A
179: STA SEKTOR
180: CALL PR1SEC
181:
182: ;-----
183: ;--- Editor ---
184: STEING: XRA A ;Zeilenwert
185: CALL SETZL ;Basis-Werte für Zeilenanfang setzen
186: LHL PUFHOS ;Startadr. im Sektorbereich nach PUFADR
187:
188: FLGCHR: SHLD PUFADR
189: ;--- Eingabepuffer mit den akt. Werten (Hex+2) in BS versehen ---
190: FLGCH2: MOV A,M ;Binärwert aus Textpuffer
191: CALL CHVHEX## ;umwandeln in hexadez. ASCII-Zeichen
192: LXI H,EG,PUF
193: MOV M,D ;1. Byte der BS-Anzeige in den Eing.Puffer
194: INX H
195: MOV M,E ;2. Byte in EG,PUF
196: DCX H ;HL steht auf EG,PUF-Anfang
197:
198: TRASTE: CALL CONIN
199: CPI FF,UP
200: JZ UPZL ;1 Zeile aufwärts
201: CPI FF,DNN
202: JZ DWNZL ;1 Zeile abwärts
203: CPI SFF,UP
204: JZ SET,CR ;1 Sektor zurück
205: CPI SFF,DNN
206: JZ SECVOR ;1 Sektor vor
207: CPI 3 ;CTRL-C?
208: JZ SPKRET ;neue Spur (ohne Ablage)
209: CPI COPY
210: CALL ABLAGE
211: CPI TAB
212: JZ ASC,ED ;Sprung zum ASCII-Editor
213: CPI WZ2H
214: JZ SPLI ;1 Pos. nach links
215: CPI BF3H
216: JZ SPRE ;1 Pos. nach rechts
217: ;--- Zeichenannahme (nur 0-9,A-F) ---
218: CPI 30H
219: JC TRASTE ;Fehler

```

```

220: CPI 3AH
221: JC TRASTE ;Fehler
222: CPI 'R'
223: JC TRASTE ;Fehler
224: CPI 'C'
225: JNC TRASTE ;Fehler
226:
227: ;AUSG: NOV M,A ;Ablage im Puffer
228: ;AUSG2: CALL BSAUS ;EG,PUF-Adr. +1
229: INX H
230: JZ TRASTE
231: ;--- die Eingabe für 1 Zeichen im Textpuffer ist erledigt ---
232: CALL CHVBIN##
233: LHL PUFADR
234: MOV M,M ;bindes Zeichen in Textpuffer ablegen
235: INX H ;Pufferadresse erhöhen
236: SHLD PUFADR
237: MOV D,0 ;Zeichen retten
238: ;--- Ausgabe auf der ASCII-Seite ---
239: CALL GETCRS
240: PUSH H ;akt. Cursor-Pos. retten
241: LHL CRSHSC
242: PUSH H
243: CALL SETCRS ;Cursor auf ASCII-Seite setzen
244: MOV A,D ;Zeichen holen
245: ANI 7FH ;Bit 7 ausblenden
246: CPI ' ' ;20H?
247: JNC AUSG3 ;nein
248: MVI A,'.' ;Punkt ausgeben
249: ;AUSG3: CALL CONOUT
250: POP H
251: INR H
252: SHLD CRSHSC ;auf nächstes Zeichen im ASCII-Bereich
253: H
254: CALL SETCRS ;Cursor wieder im HEX-Bereich
255: MVI A,9
256: CALL CONOUT ;Leerraum überspringen
257: LDA ZRE
258: CPI 15 ;Zeilenende?
259: JNZ AUSG4 ;nein
260: ;--- Folgezeile ---
261: LDA ZEILE
262: CPI 15 ;letzte Zeile?
263: JZ STEIN2 ;Ausgangszeile Editor
264: INR A
265: SETZL ;neue Zeile einrichten
266: JMP AUSG5
267: ;AUSG4: INR A
268: STA ZRE ;Zeichenzähler+1
269: ;AUSG5: LHL PUFADR
270: JMP FLGCH2 ;nächste Position
271:
272: ;--- 1 Position nach rechts ---
273: SPRE: MVI A,9
274: JMP AUSG2
275:
276: ;--- 1 logische (1) Position nach links ---
277: SPLI: LDA OUTFLG
278: ORA A
279: JNZ SPLI2 ;1. Pos. im Hex-Doppelfeld
280: ;--- 1 Pos. zurück ---
281: MVI A,8
282: CALL BSAUS
283: JMP TRASTE-1 ;incl. Eingabepufferadr. -1
284:
285: SPLI2: LDA ZRE ;Zeichenzähler
286: ORA A ;#0?
287: JZ T,BELL ;Zeilenanfang
288: DCR A
289: STA ZRE ;Zeichenzähler -1
290: ;--- 1 Gruppe im Hexfeld zurück ---
291: LHL PUFADR
292: MOV A,H ;letztes Zeichen noch einmal holen
293: CALL PR1HEX ;und auf dem BS ausgeben
294:
295: LXI D,BSF5
296: CALL PR1STR+3 ;3B nach links
297:
298: LHL CRSHSC
299: MOV H ;Zeilenanfang
300: SHLD CRSHSC
301: LHL PUFADR
302: DCX H ;1 Stelle zurück
303: JMP FLGCHR
304:
305: ;--- 1 Zeile aufwärts ---
306: UPZL: LDA ZEILE
307: ORA A
308: JZ T,BELL ;oberste Zeile
309:
310: DCR A
311: STA ZEILE ;-1
312: LDA OUTFLG
313: ORA A ;vord. Pos. in Hex-Feld?
314: JNZ UPZL2 ;ja
315: MVI A,8
316: CALL BSAUS ;Korrektur der Position
317:
318: UPZL2: LHL PUFADR
319: MOV A,M
320: LXI D,-16 ;Pufferadr. -16 Zeichen
321: DRD D
322: SHLD PUFADR
323: CALL PR1HEX ;altes Hex.Zeichen noch einmal ausgeben
324: LXI D,BSF5+3
325: CALL PR1STR+3 ;2 Ausgabepos. wieder zurück
326: ;--- beide Cursorpositionen 1 Zeile höher setzen ---
327: MVI A,11
328: CALL CONOUT ;Cursor 1 Zeile höher
329: LDA CRSHEX
330: DCR A
331: STA CRSHEX ;Zeile für Hex-Cursor -1
332: LDA CRSHSC
333: DCR A
334: STA CRSHSC ;Zeile für ASCII-Cursor -1
335: JMP AUSG5
336:
337: ;--- 1 Zeile abwärts ---
338: DWNZL: LDA ZEILE
339: CPI 15
340: JZ T,BELL ;letzte Zeile
341:
342: INR A
343: STA ZEILE ;+1
344: LDA OUTFLG
345: ORA A ;vord. Pos. in Hex-Feld?
346: JNZ DWNZL2 ;ja
347: MVI A,8
348: CALL BSAUS ;Korrektur der Position
349:
350: DWNZL2: LHL PUFADR
351: MOV A,M
352: LXI D,-16 ;Pufferadr. +16 Zeichen
353: DRD D
354: SHLD PUFADR
355: CALL PR1HEX ;altes Hex.Zeichen noch einmal ausgeben
356: LXI D,BSF5+3
357: CALL PR1STR+3 ;2 Ausgabepos. wieder zurück
358: ;--- beide Cursorpositionen 1 Zeile tiefer setzen ---
359: MVI A,10
360: CALL CONOUT ;Cursor 1 Zeile tiefer
361: LDA CRSHEX
362: INR A
363: STA CRSHEX ;Zeile für Hex-Cursor +1

```

Listing 2. Freunde des 8080-Assembler-Codes geben das Programm direkt ein (Teil 1)

```

364 LDR CRSASC
365 R INR
366 STR CRSASC ;Zeile für ASCII-Cursor +1
367 JMP AUSGS
-----
369 ;
370 ; Rucksprung in Home-Position des Hex-Feldes
371 STEIN2: CALL BELL
372 JMP STEING
373 ;
374 ; ----- ASCII - Editor -----
375 ;
376 RSC.ED: CALL GETCRS ;akt. Cursor-Pos. ermitteln
377 SHLD CRSHEX
378 LDR OUTFLG
379 ORA A ;Pos. im Hex-Feld linksbündig?
380 JNZ A.ED1 ;ja
381 ; ----- Position 1 Stelle zurücksetzen ---
382 MVI A,8
383 CALL BSARS ;1 Stelle nach links
384 LHL CRSHEX
385 DCR H
386 SHLD CRSHEX
387 ;
388 A.ED1: LHL CRSASC ;Cursor im ASCII-Feld setzen
389 CALL SETCRS
390 ATASTE: CALL CONIN
391 CPI PF,UP ;1 Zeile aufwärts
392 JZ RZLUP
393 CPI PF,DNH ;1 Zeile abwärts
394 JZ RZLDNH
395 CPI PF,UP ;1 Sektor zurück
396 JZ ASC.ZR
397 JZ SPFDWN ;1 Sektor vor
398 JZ ASCVOR
399 CPI 3 ;CTRL-C?
400 JZ SPRRET ;neue Spur (ohne Ablage)
401 CPI COPY
402 JZ ABLAGE
403 CPI TAB
404 JZ RETHEX
405 CPI OF2H ;1 Pos. nach links
406 JZ ASPLI
407 CPI OF3H ;1 Pos. nach rechts
408 JZ ASPRE
409 ; ----- Zeichenannahme (20H - 7FH) ---
410 CPI 20H
411 JC ATASTE
412 CPI 80H
413 JNC ATASTE
414 ;
415 CALL CONOUT
416 LHL PUFADR
417 MOV M,A ;Zeichen in Sektor schreiben
418 INX H ;Zeiger +1
419 SHLD PUFADR
420 MOV D,A ;Zeichen retten für PRTHEX
421 ; ----- Ausgabe des Hexwertes ---
422 LHL CRSHEX
423 PUSH H
424 CALL SETCRS
425 MOV A,D ;Zeichen
426 CALL PRTHEX
427 POP H
428 ; ----- Zeilenwechsel? ---
429 A.ED3: LDR ZRAE
430 CPI 15
431 JNZ A.ED2 ;nein
432 ; ----- neue Zeile (BS-Ende?) ---
433 LDR ZEILE
434 CPI 15
435 JZ STEIN2 ;BS-Ende = Rucksprung in Home-Pos. Hex-Feld
436 INR A
437 CALL SETZL ;neue Grundwerte für Folgezeile und
438 JMP A.ED1 ;Cursor wieder auf ASCII-Seite setzen
439 A.ED2: INR A
440 STR ZRAE ;Zeichenzähler +1
441 INR H
442 INR H
443 INR H ;FolgePos. Hex-Feld
444 SHLD CRSHEX
445 LHL CRSASC
446 INR H
447 A.ED4: SHLD CRSASC
448 JMP ATASTE-3 ;Schleife
449 ;
450 ; ----- Wechsel ins Hex-Feld (Eingabe TAB) ---
451 RETHEX: LHL CRSHEX
452 CALL SETCRS
453 JMP AUSGS
454 ;
455 ; ----- 1 Zeichen nach rechts ---
456 HSPRE: MVI A,9
457 CALL CONOUT
458 LHL PUFADR
459 INX H
460 SHLD PUFADR
461 LHL CRSHEX
462 JMP A.ED3
463 ;
464 ; ----- 1 Zeichen nach links ---
465 ASPLI: LDR ZRAE
466 ORA A ;Zeilenanfang?
467 JZ ATBELL ;ja
468 DCR A
469 STR ZRAE ;Zeilenzähler -1
470 MVI A,8
471 CALL CONOUT ;CP/M-Pos. -1
472 LHL PUFADR
473 DCX H
474 SHLD PUFADR ;Pufferadr. -1
475 LHL CRSHEX
476 DCR H
477 DCR H
478 DCR H ;Hex-Pos. 1 Feld zurück
479 SHLD CRSASC
480 LHL CRSASC
481 DCR H ;ASCII-Cursor -1
482 JMP A.ED4
483 ;
484 ; ----- 1 Zeile aufwärts ---
485 RZLUP: LDR ZEILE
486 ORA A
487 JZ ATBELL ;oberste Zeile
488 DCR H
489 STR ZEILE
490 ;
491 LHL PUFADR
492 LXI D,-16
493 DAD D
494 SHLD PUFADR
495 ;
496 MVI A,11
497 CALL CONOUT ;Cursor 1 Zeile hoch
498 LDR CRSHEX
499 DCR H
500 STR CRSHEX ;Zeile für Hex-Pos. -1
501 LHL CRSASC
502 DCR L ;Zeile -1
503 JMP A.ED4
504 ;
505 ; ----- 1 Zeile abwärts ---
506 RZLDNH: LDR ZEILE
507 CPI 15
508 JZ ATBELL ;unterste Zeile
509 INR A
510 STR ZEILE
511 ;
512 LHL PUFADR
513 LXI D,16
514 DAD D
515 SHLD PUFADR
516 ;
517 MVI A,10
518 CALL CONOUT ;Cursor 1 Zeile tiefer
519 LDR CRSHEX
520 INR H
521 STR CRSHEX ;Zeile für Hex-Pos. +1
522 LHL CRSASC
523 INR L ;Zeile +1
524 JMP A.ED4
525 ;
526 ; ----- Wechsel des Sektors -----
527 ; ----- 1 Sektor innerhalb der Spur zurück ---
528 SEC.ZR: LDR SEKTOR
529 ORA A ;1. Sektor?
530 JZ T.BELL ;ja
531 DCR A
532 JMP P.SEC
533 T.BELL: CALL BELL
534 JMP TRASTE
535 ;
536 ; ----- 1 Sektor innerhalb der Spur vor ---
537 SEC.VR: LDR SEKTOR
538 INR A ;+1
539 LXI H,SECTP
540 CMP M ;letzter Sektor?
541 JZ T.BELL
542 JMP P.SEC
543 ;
544 ; ----- Sektorwechsel aus dem ASCII-Feld heraus -----
545 ASC.ZR: LDR SEKTOR ;1 Sektor zurück
546 ORA A ;1. Sektor?
547 JZ ATBELL ;ja
548 DCR A
549 ASCRT: PUSH PSH
550 LHL CRSHEX
551 CALL SETCRS ;Cursor ins Hex-Feld
552 POP PSH
553 JMP P.SEC
554 ATBELL: CALL BELL
555 JMP ATASTE
556 ;
557 ASCVOR: LDR SEKTOR ;1 Sektor vor
558 INR A ;+1
559 LXI H,SECTP
560 CMP M ;letzter Sektor?
561 JZ ATBELL
562 JMP ASCRT
563 ;
564 ;
565 RESET: MVI C,0DH ;Reset Disk.
566 CALL 5 ;BDOS
567 JMP 0
568 ;
569 BELL: MVI A,7
570 JMP CONOUT
571 ;
572 ;
573 BSP5: DB 8,8,8,8,8,'$' ; 5 AusgabePositionen zurück
574 ;
575 END

```

Listing 2. Freunde des 8080-Assembler-Codes geben das Programm direkt ein (Teil 1) (Schluß)

```

1: TITLE Routinen für das Utility-Programm HELP.COM
2: SUBTTL HKROUT.MHC --- 11.08.1985 ---
3:
4: EXTRN SETCRS
5: EXTRN EINSPPR
6:
7: CR EQU 0DH
8: LF EQU 0AH
9: EEZ EQU 1CH ;Endemarke im Puffer
10: STRVRT EQU 95H ;senkrechter Strich
11: STRHOR EQU 9AH ;horizontaler Strich
12:
13: ;
14: ; 1 Spur ablesen (mit evtl. Änderungen)
15:
16: ABLAGE: LDR SPUR
17: CALL SELTRK
18: ORA A
19: STR RECORD
20: LXI H,PUFFER
21: SHLD PUFADR ;Pufferadr.
22:
23: WR.LP: CALL SELREC
24: CALL WRITE
25: ORA A
26: JNZ FHLR
27: ;
28: LDR RECORD
29: INR A
30: LXI H,RECPT
31: CMP H
32: JZ SPRRET ;evtl. neue Spur eingeben
33: ;
34: STR RECORD
35: LHL PUFADR
36: LXI D,128
37: DAD D
38: SHLD PUFADR
39: JMP WR.LP ;Schleife
40:
41: ;
42: ; Ausgabe Bildschirm im Hexafeld (mit OUTFLG)
43:
44: BSARS: CALL CONOUT
45: LDR OUTFLG
46: ORA A
47: STR OUTFLG
48: ORA A
49: RET ;Abfrage im Hauptprogramm
50:
51: ;
52: ; numerisches Einabefeld
53:
54: EINGB: LXI H,EG.PUF
55: MOV C,B ;C zum Vergleich mit Max.-Wert füllen
56: EINLP: CALL CONIN
57: CPI 3
58: RZ
59: CPI 8 ;CLR?
60: JZ BSPC
61: CPI 0DH ;ENTER?

```

Listing 3. Die Unterprogramme von »Help« (Teil 2)

LOGO

Jeder kann programmieren
Computersprache für Eltern und Kinder
DANIEL WATT

LOGO... Ergebnis der Erforschung menschlicher Intelligenz

Entwickelt von Seymour Papert, Pädagoge und Mathematikprofessor. Erste Computersprache, die bewußt Strategien menschlichen Denkens dient – und in ihrer Logik der Realität gerecht wird. LOGO ersetzt BASIC, sagen Pädagogen und Mathematiker. LOGO kommt dem übergreifenden, assoziativen Denken entgegen. BASIC dagegen ist ein Setzkasten von Logik-Buchstaben.

DANIEL WATT... hat im Team von Seymour Papert gearbeitet und ein Buch geschrieben, das voller Bilder seine Erlebnisse mit Kindern am Computer wiedergibt. Ein hochwertiges Textbuch für LOGO-Kurse. Ein Buch für Lehrer, die nach einem bereits von Schulbehörden empfohlenen LOGO-Kursbuch suchen.

Ein Buch für APPLE II, C-64, IBM PC, ATARI bis 520 ST., TI-99 und Schneider CPCs.
384 Seiten, A4, DM 59,-



„Buch des Jahres 1983“
in den USA

te-wi

te-wi Verlag GmbH
Theo-Prosel-Weg 1
8000 München 40

COMPUTER FÜR KINDER



Ein Buch für Kinder und ihre Lehrer – ein kindgemäßes Buch für die erste Begegnung mit Computern, ihren Eigenwilligkeiten, und ihren unerschöpflichen Möglichkeiten. Ein Buch zu unserer Gegenwart und zur Zukunft unserer Kinder. „Computer für Kinder“ richtet sich an Kinder im Alter von 8 bis 13 Jahren, für deren Interesse an Computern keines der unzähligen Computer-Bücher geschrieben wurde.



„Computer für Kinder“ ist ganz auf Kinder eingestellt und beschäftigt sich unterhaltsam und leicht verständlich mit folgenden Themen:

- Wie arbeiten Computer
- Wie funktioniert mein Computer
- Wie programmiert man mit einfachen Flußdiagrammen
- Wie kann ich BASIC leicht verstehen
- Programme aufbauen mit Befehlen
- Farbige Graphiken entwerfen
- Erklärung von Computer-Begriffen



Sally Greenwood Larson war Kindergärtnerin, ehe sie selbst Computern begegnete und zwischen den Welten von Kindern und Computern zu vermitteln begann.

Computer für Kinder, A4 quer, Fadenheftung,
über 100 Seiten, je Ausgabe DM 29,80
vorliegend für: VC 20, C 64, Apple II, Atari



DM 29,80



DM 29,80



DM 59,-



DM 199,-



DM 239,-



DM 49,-



DM 36,-

```

62 JZ EINHRET
63 CPI 30H
64 JZ EINHLP ;0
65 CPI 7FH
66 JZ BSFC ;DEL
67 CPI 30H
68 JNC EINHLP ;>
69 DCR B ;Zähler -1
70 JNZ EINGB1
71 ; Zähler 0 = Feldlänge überschritten ---
72 INR B
73 MVI A,7 ;BELL
74 JMP EINGB2
75
76 EINGB1 MOV M,A ;Ablage in Puffer
77 INX H
78 EINGB2 CALL CONOUT
79 JMP EINHLP ;Schleife
80
81 EINHRET MVI M,EEZ
82 RET
83
84 ; --- Backspace ---
85 BSFC MOV A,C
86 OR B ;B=C? wenn ja = Feldanfang
87 JZ EINHLP
88 INR B
89 DCX H
90 MVI A,8
91 CALL CONOUT
92 MVI A,7
93 CALL CONOUT
94 MVI A,8
95 CALL CONOUT
96 JMP EINHLP
97
98 EG.PUF DS 3 ;Eingabefeldpuffer
99
100 ;
101 ; 1 Sektor (256 Bytes) auf dem Bildschirm anzeigen
102
103 PRTEC LXI H,BUSH
104 SHLD CRSHFX
105 CALL SETCRS ;Home-Pos. in Anzeigebereich
106 XRA A
107 STR ZARE ;Zeilenzähler
108 LXI H,PUFPER
109 LDR SEKTOR
110 MOV B,A
111 MOV A,H
112 ADD B
113 MOV H,A ;HL = 1. Pos. des ausgewählten Sektors im Puffer
114 SHLD PUFADS
115 SHLD PUFADR ;akt. Zeiger vorbelegen
116
117 PSCLP1 XRA A
118 MOV C,A ;Zähler für Zeilenausgabe
119 LXI H,ASCPUF
120 SHLD APUFAD ;Anfangsadr. des ASCII-Puffers
121 LHLD PUFADR
122
123 PSECLP MOV A,H
124 MOV B,A ;Zeichen in B
125 PUSH H
126 LHLD APUFAD
127 MOV M,B
128 INX H
129 SHLD APUFAD ;Zeichen im ASCII-Puffer ablesen
130 POP H
131
132 MOV A,B
133 CALL PRTHFX
134 INX H
135 MVI A,7
136 CALL CONOUT
137 INR C ;Zeichenzähler +1
138 MOV A,C
139 CPI 16
140 JNZ PSECLP ;Schleife
141
142 ; -- ASCII-Zeichen ausgeben --
143 SHLD PUFADR
144 MVI A,9
145 CALL CONOUT
146 MVI A,9
147 CALL CONOUT
148 ; (C enthält noch 16 als Zeichenzähler)
149 LXI H,ASCPUF
150 PRSC1 MOV A,H
151 ANI 7FH ;Bit 7 ausblenden
152 CPI 7
153 JNC PRSC2
154 MVI A,7 ;Ersatzzeichen
155 PRSC2 CALL CONOUT
156 INR H
157 DCR C
158 JNZ PRSC1
159 ; -- alle 16 Zeilen erledigt? --
160 LDR ZARE
161 CPI 15
162 RZ ;ja
163 INR A
164 STR ZARE
165 LHLD CRSHFX
166 INR L ;Folgezeile Cursor
167 SHLD CRSHFX
168 CALL SETCRS
169 JMP PSCLP1
170
171 ;
172 ; 1 Spur lesen
173
174 RD.TRC CALL SELTRK ;akt. Spur auswählen
175 XRA A
176 STR RECORD
177 LXI H,PUFPER
178 SHLD PUFADR
179
180 RD.LP CALL SELREC
181 CALL READ ;Sektor lesen
182 ORR A
183 JZ RD.LP2
184 ; --- Fehler ---
185 FHLR LXI D,FEHLR1
186 LXI H,510H ;Zeile 24
187 CALL PRTSTR
188 JMP
189
190 RD.LP2 LDR RECORD
191 INR A
192 LXI H,RECP1
193 CMP M
194 RZ ;Spur gelesen
195 STR RECORD
196 LHLD PUFADR
197 LXI D,128
198 DAD D
199 SHLD PUFADR
200 JMP RD.LP ;Schleife
201
202 ;
203 ; Umrechnung der numerischen Eingaben
204
205 HEX LXI H,EG.PUF
206 MOV A,H

```

```

207 CPI EEZ ;keine Eingabe?
208 RZ
209 INR H
210 MOV A,M
211 CPI EEZ
212 JNZ HEX2
213 ; 1-stellige Eingabe
214 DCX H
215 MOV A,M
216 ANI 0FH
217 RET
218 ; 2-stellige Eingabe
219 HEK2 ANI 0FH
220 MOV D,A
221 DCX H
222 MOV A,M
223 ANI 0FH
224 ;
225 ADD A ;*2
226 MOV B,A
227 ADD A
228 ADD A ;*8
229 ADD B ;*10
230 ADD D ;+ 'Einer'
231 RET
232
233 ;
234 ; hexadez. Zeichen im EG.PUF in Binärzeichen wandeln
235
236 CHVBIN LXI D,EG.PUF+1
237 LDAX D
238 CALL CNVBN ;2. Zeichen
239 MOV B,A ;retten
240 DCX D
241 LDAX D
242 CALL CNVBN ;1. Zeichen
243 ANA A ;C-Bit löschen
244 RAL
245 RAL
246 RAL
247 ORR B ;Inhalt in 1.Nibble schieben
248 ORR B ;Inhalt von B verknüpfen
249 RET ;A enthält jetzt Binärzahl
250
251 CHVN LXI SUI 30H
252 CPI 10
253 RC
254 SUI 7
255 RET ;Ziffer
256
257 ;
258 ; Binärzeichen in hexadezimaler Form in DE bereitstellen
259
260 CNVHEX PUSH PSM
261 RRC
262 RRC
263 RRC
264 RRC
265 CALL CNVNIB
266 MOV D,A ;1. Hex.Zahl -> D
267 POP PSM
268 CALL CNVNIB
269 MOV E,A ;2. Hex.Zahl -> E
270 RET
271
272 CNVNIB ANI 0FH
273 CPI 10
274 JNC CNV10
275 RDI '0'
276 RET
277 CNV10 RDI 'A'-10
278 RET
279
280 ;
281 ; Binärzeichen in hexadezimaler Form ausgeben
282
283 PRTHFX PUSH PSM
284 RRC
285 RRC
286 RRC
287 RRC
288 CALL PNIB
289 POP PSM
290
291 PNIB ANI 0FH
292 CPI 10
293 JNC P10
294 RDI '0'
295 JPF CONOUT
296 P10 RDI 'A'-10
297 JMP CONOUT
298
299 ;
300 ; Anfangswerte einer Zeile setzen
301
302 SETZL STR ZELE
303 RDI 5 ;auf Feldzeile
304 MOV L,A ;für ASCII-Cursor
305 MOV E,A ;für HEX-Cursor
306 D,B ;Spalte 9 (Hex-Feld) - DE: Hex-Cursor-Position
307 MVI H,30H ;1. Spalte im ASCII-Feld - HL wie vor (ASCII)
308 XRA A
309 STR ZARE ;Spaltenzähler auf 0
310 ORR A
311 STR OUTFLG ;PrintPos. links im Hex-Doppelfeld
312 SHLD CRSRSC
313 XCHG
314 SHLD CRSHFX
315 JMP SETCRS
316
317 ;
318 ; Löschen des 2stelligen Eingabefeldes
319
320 SPRRET LXI H,VR17H
321 CALL LSCFP
322 JMP EINSRP
323
324 LSCFP CALL SETCRS
325 MVI C,7
326 CALL CONOUT
327 MVI C,7
328 JMP CONOUT
329
330 ;
331 ; Maskenerstellung
332 ; horizontalen Strich ausgeben
333 HORSTR MVI A,STRHOR
334 CALL CONOUT
335 DCR B
336 JNZ HORSTR
337 RET
338
339 ;
340 ; vertikalen Strich ausgeben
341 VRTSTR PUSH H
342 CALL SETCRS
343 POP H
344 MVI A,STRVRT
345 CALL CONOUT
346 L INR L ;nächste Zeile
347 DCR B
348 JNZ VRTSTR
349 RET
350

```

```

351:
352: TXT1: DB 'H E L P / LPC 664 (1,01) 8,850'
353: TXT2: DB '-0 -1 -2 -3 -4 -5 -6 -7 -8 -9 -A -B -C -D -E -F /,95H'
354: DB '0123456789ABCDEF'
355: TXT3: DB '00',CR,LF,' 10',CR,LF,' 20',CR,LF,' 30',CR,LF,' 40',CR
,LF
356: DB ' 50',CR,LF,' 60',CR,LF,' 70',CR,LF,' 80',CR,LF,' 90'
,CR,LF
357: DB ' A0',CR,LF,' B0',CR,LF,' C0',CR,LF,' D0',CR,LF,' E0'
,CR,LF
358: DB ' F0'
359: TXT4: DB 'Sektor von /- zurueck',18H,Shift Pfeil',18H,CR,LF
360: DB ',18H,Spur',18H,' (Abbruch: CTRL-C)'
361: DB 'Wechsel HEX (<-) ASCII-Cursor',18H,' TAB ',18H,CR,LF
362: DB 'neue Spur (ohne Ablase)',18H,' CTRL-C',18H
363: DB '(<',18H,' mit ',18H,' Ablase)',18H,' COPY',18H,' #'
364: FEHLR1: DB 7,'### Disk. Eingabe-/Ausgabefehler ###'
365:
366: --- CP/M - Schnittstellen ---
367: Sektor auswählen
368: SELREC: LHL D,PUFADR
369: PUSH H
370: POP B ;nach BC
371: CALL SETDMA
372:
373: LDA RECORD
374: MOV C,A
375: RVI B,0
376: LHL TRSTAB
377: XCHG ;Übersetzungstabelle
378: CALL SCTRAN
379: -- HL = Phys. Sektornummer --
380: MOV B,H
381: MOV C,L
382: JMP SETSEC
383:
384: --- Diskettenroutinen ---
385: SELDRV: MOV C,A ;Laufwerksnummer in C
386: LHL I
387: LXI D,18H
388: DAD D
389: PCHL
390:
391: SELTRK: MOV C,A
392: RVI B,0 ;BC = Spurnummer
393: LHL I
394: LXI D,18H ;SETTRC
395: DAD D
396: PCHL
397:
398: SCTRAN: LHL I
399: RVI L,30H
400: PCHL
401:
402: SETSEC: LHL I
403: LXI D,1EH
404: DAD D
405: PCHL
406:
407: SETDMA: LHL I
408: LXI D,21H
409: DAD D
410: PCHL
411:
412: READ: LHL I
413: LXI D,24H
414: DAD D
415: PCHL
416:

```

```

417: WRITE: MVI C,2 ;BIOS-Flag für Sektor-Blockung
418: LHL I
419: LXI D,27H
420: DAD D
421: PCHL
422:
423:
424: PRSTR: CALL SETCRS
425: PRSTR: LDRX D
426: CPI ' '
427: RZ
428: MOV C,A
429: PUSH D
430: CALL DCNOUT
431: POP D
432: INX D
433: JMP PRSTR
434:
435: CONIN: PUSH B
436: PUSH H
437: CALL DCNIN
438: POP H
439: POP B
440: RET
441:
442: CONOUT: PUSH B
443: PUSH H
444: MOV C,A
445: CALL DCNOUT
446: POP H
447: POP B
448: RET
449:
450: DCNIN: LHL I
451: RVI L,9
452: PCHL
453:
454: DCNOUT: LHL I
455: RVI L,0CH
456: PCHL
457:
458: --- Datenfelder ---
459: ASCPUF: DS 16 ;ASCII-Puffer
460: PUFADR: DS 2 ;Zeiger in ASCII-Puffer
461: CRSHX: DS 2 ;HL-Wert des CursorPosition (Hex-Feld)
462: CRSHSC: DS 2 ;- wie vor (ASCII-Feld)
463: PUFADS: DS 2 ;Anfangswert im Textpuffer für Jeweil. Sektor
464: PUFADR: DS 2 ;akt. Zeiger im Textpuffer
465: OUTFLG: DS 1 ;PrintPos. im Hex-Doppelfeld
466: ZEILE: DS 1
467:
468: SPUR: DS 1
469: RECORD: DS 1 ;128 Bytes
470: SEKTOR: DS 1 ;256 Bytes
471: ZARE: DS 1 ;Zeilenzähler (max. 16)
472:
473: DSKDPH: DS 2 ;Anfangsadr. des DPH
474: RECDT: DS 1 ;Anzahl der Records je Spur
475: TRSTAB: DS 2 ;Übersetzungstabelle
476: SECDT: DS 1 ;Anzahl der Sektoren je Spur
477:
478: PUFFER: DS 1 ;Startadr. für Textpuffer
479:
480: END

```

Listing 3. Die Unterprogramme von »Help« (Schluß)

Drucken ohne Platzprobleme



Speicherplatz ist rar beim Schneider-Diskettenlaufwerk. Verkürzen Sie doch einfach die Systemroutinen.

Besitzer des Schneider-Diskettenlaufwerks sind ständig auf der Suche nach ein paar zusätzlichen KByte Kapazität. Schließlich sind 180 KByte schnell verbraucht. Dateien, die nicht unbedingt gebraucht werden, sind deshalb schnell gelöscht. Zu diesen Dateien gehört das Dienstprogramm »PIP«, dessen Nutzen sich bei den meisten auf das Kommando »PIP PRN:=Dateiname.Ext« beschränkt.

Mit LPRINT Dateiname.Ext« haben Sie unter CP/M 2.2 ein eigenständiges Programm, mit dem Quelldateien entsprechend den besonderen Vereinbarungen, die für »PIP PRN« gelten, ausgedruckt werden können. Der Platzbedarf auf der Diskette beträgt aber nur 1 KByte. Angaben über den Assembler finden Sie bei dem Programm »Help« (Seite 101). (H.-D. Lange)

```

10 ***** [F9FC]
20 * (c) Heinz-Dieter Lange [FD80]
30 * Kamper Weg 90, 2210 Itzehoe * [B5E6]
40 ***** [E9D2]
50 'Basic-Lader fuer die Datei 'LPRINT.COM' [E944]
OM [E2B6]
60 OPENOUT"LPRINT.COM":zeile=10000 [192C]
70 pruef=0
80 FOR j=i TO i+7:READ d$:IF d$="Ende" T [04E2]
HEN GOTO 110 ELSE d=VAL("&"+d$):PRINT [620E]
#9,CHR$(d);:pruef=pruef+d:NEXT [7E4C]
90 READ p:IF pruef<>p THEN GOTO 140 [D7D0]
100 zeile=zeile+10:GOTO 70 [0C42]
110 PRINT"Alles O.K." [B416]
120 CLOSEOUT
130 END
140 PRINT"Pruefsummenfehler in Zeile"zei [9A78]
le [1048]
150 CLOSEOUT [EF1C]
160 END [9162]
170 DATA 21,00,00,39,22,57,02,31, 262 [0C08]
180 DATA 81,02,AF,32,7C,00,11,5C, 589 [F68A]
190 DATA 00,0E,0F,CD,05,00,FE,FF, 748 [6BC4]
200 DATA C2,26,01,11,29,02,CD,24, 534 [B1F0]
210 DATA 02,2A,57,02,F9,C9,11,81, 729 [BEF8]
220 DATA 02,D5,0E,1A,CD,05,00,11, 482 [B2FE]
230 DATA 5C,00,0E,14,CD,05,00,E1, 561 [1CD8]
240 DATA B7,C2,4E,01,01,80,00,09, 594 [0E9C]
250 DATA EB,3E,9F,3D,BA,D2,29,01, 955 [09EE]
260 DATA 11,44,02,C3,1E,01,AF,32, 538

```

```

270 DATA 5A,02,32,59,02,21,80,02, 396 [979E]
280 DATA AF,32,60,02,23,7E,FE,1A, 764 [CB4E]
290 DATA CA,21,01,2B,E5,2A,59,02, 641 [5212]
300 DATA 23,22,59,02,CD,CD,01,21, 604 [92E4]
310 DATA 5B,02,06,05,7E,CD,14,02, 457 [EBF8]
320 DATA 23,05,C2,74,01,3E,3A,CD, 676 [9F20]
330 DATA 14,02,3E,20,CD,14,02,E1, 568 [6CF0]
340 DATA 0E,01,23,7E,FE,09,CA,B3, 820 [A15E]
350 DATA 01,CD,14,02,FE,0A,C2,8A, 824 [4D52]
360 DATA 01,3A,60,02,3C,32,60,02, 365 [A9A2]
370 DATA FE,40,C2,5C,01,06,08,3E, 681 [D02C]
380 DATA 0A,CD,14,02,05,C2,A9,01, 606 [90A0]
390 DATA C3,58,01,3E,08,B9,D2,BE, 939 [266E]
400 DATA 01,C6,08,C3,B5,01,3C,F5, 889 [AA30]
410 DATA 3E,20,CD,14,02,F1,B9,C2, 941 [6E38]
420 DATA BF,01,C3,8A,01,2A,59,02, 659 [D020]
430 DATA 11,5B,02,01,F0,D8,CD,00, 772 [BB0C]
440 DATA 02,01,18,FC,CD,00,02,01, 487 [EEEE8]
450 DATA 9C,FF,CD,00,02,01,F6,FF, 1120 [B5A2]
460 DATA CD,00,02,7D,EE,30,12,21, 669 [5E20]
470 DATA 5B,02,3E,30,06,04,BE,C0, 595 [C620]
480 DATA 36,20,23,05,C2,F6,01,C9, 768 [95FC]
490 DATA 3E,30,09,3C,DA,02,02,F5, 646 [022A]
500 DATA A7,7D,91,6F,7C,98,67,F1, 1168 [318E]
510 DATA 3D,12,13,C9,F5,C5,D5,E5, 1183 [FB7A]
520 DATA 5F,0E,05,CD,05,00,E1,D1, 758 [F342]
530 DATA C1,F1,0C,C9,0E,09,C3,05, 870 [E448]
540 DATA 00,0D,0A,07,44,61,74,65, 412 [D2B6]

```

Listing 1. Der Basic-Lader für »LPRINT«

```

550 DATA 69,20,6E,69,63,68,74,20, 703 [A0DA]
560 DATA 76,6F,72,68,61,6E,64,65, 855 [D91C]
570 DATA 6E,0D,0A,24,0D,0A,07,44, 267 [441E]
580 DATA 61,74,65,69,20,7A,25,20, 722 [FECC]
590 DATA 6C,61,6E,67,0D,0A,24,0D, 477 [3F1A]
600 DATA 00,00,00,00,00,00,00,00, 0 [90C0]
610 DATA 00,00,00,00,00,00,00,00, 0 [81C2]
620 DATA 00,00,00,00,00,00,00,00, 0 [3AC4]
630 DATA 00,00,00,00,00,00,00,00, 0 [3BC6]
640 DATA 00,00,00,00,00,00,00,00, 0 [34C8]
650 DATA 00,00,09,3C,DA,02,02,F5, 536 [29EC]
660 DATA A7,7D,91,6F,7C,98,67,F1, 1168 [789C]
670 DATA 3D,12,13,C9,F5,C5,D5,E5, 1183 [518B]
680 DATA 5F,0E,05,CD,05,00,E1,D1, 758 [E550]
690 DATA C1,F1,0C,C9,0E,09,C3,05, 870 [2C56]
700 DATA 00,0D,0A,07,44,61,74,65, 412 [C2B2]
710 DATA 69,20,6E,69,63,68,74,20, 703 [C8D6]
720 DATA 76,6F,72,68,61,6E,64,65, 855 [7118]
730 DATA 6E,0D,0A,24,0D,0A,07,44, 267 [481A]
740 DATA 61,74,65,69,20,7A,25,20, 722 [E2C0]
750 DATA 6C,61,6E,67,0D,0A,24,0D, 477 [99E6]
760 DATA 00,00,00,00,00,00,00,00, 0 [98CE]
770 DATA 00,00,00,00,00,00,00,00, 0 [67D0]
780 DATA 00,00,00,00,00,00,00,00, 0 [BED2]
790 DATA 00,00,00,00,00,00,00,00, 0 [3DD4]
800 DATA 00,00,00,00,00,00,00,00, 0 [3CC4]
810 DATA 01,01,01,01,01,01,01,01, 8 [99E6]
820 DATA 01,01,01,01,01,01,01,01, 8 [76E8]
830 DATA Ende [D9E2]
    
```

Listing 1. Der Basic-Lader für »LPRINT« (Schluß)

```

100 INX H ;Adr. LFDNR +1
101 DCR B
102 JNZ FRTHR ;5 Stellen ausgeben
103
104 MVI A,0
105 CALL LPRT ;ausgeben
106 MVI A,0
107 CALL LPRT
108 --- Textzeile ausgeben ---
109 POP H ;Adr. im Textpuffer zurück
110 MVI C,1 ;Zähler für DruckPosition
111 PRINT3: INX H ;PufferPos. +1
112 MOV R,M
113 CPI 9 ;TAB?
114 JZ LP,TAB ;ja
115 CALL LPRT ;----> Ausgabe Drucker incl. 000R
116 CPI 00H ;Zeilenende?
117 JNZ PRINT3 ;Schleife
118
119 --- neue Zeile ---
120 LDR ZCNT
121 INR A ;Zeilenzähler +1
122 STA ZCNT
123 CPI ZWCHSL ;Seitenwechsel?
124 JNZ PRINT2 ;nein
125
126 ;COMMENT#
127 ; Die folgenden Programmzeilen sind zu ändern, wenn Drucker verwendet
128 ; werden, die die Ausgabe eines Seitenvorschubes als ESC-Sequenz er-
129 ; möglichen. In der vorliegenden Form wird die notwendige Zahl von Leer-
130 ; zeilen als Ersatz für den Seitenvorschub eingefügt. #
131
132 --- neue Seite ---
133 MVI B,ZVRSCH
134 MVI A,00H ;Leerzeile
135 PRT,FF: CALL LPRT
136 DCR B
137 JNZ PRT,FF
138 JMP PRINT1 ;Start neue Seite
139
140 --- Übertragung der TAB-Pos. auf den Drucker ---
141 LP,TAB: MVI A,8
142 CMP C ;akt. Pos.
143 JNC LPTAB
144 ADI 8
145 JMP LP,TAB+2 ;Folge-TAB
146 LPTAB: INR A ;Wert auf TAB-Pos. setzen
147 PUSH PSM
148 MVI A,0
149 CALL LPRT ;1 Blank ausgeben
150 POP PSM
151 CMP C ;C=0?
152 JNZ LPTAB+1 ;nein
153 JMP PRINT3
154
155 ;
156 ; Umwandlung der hexadez. Zahl in Dezimalzahl
157
158 CHVDEZ: LHL D,L0MVT ;Ergebnisfeld
159 LXI D,LFDNR
160
161 LXI B,-10000
162 CALL UP,DEZ
163 LXI B,-1000
164 CALL UP,DEZ
165 LXI B,-100
166 CALL UP,DEZ
167 LXI B,-10
168 CALL UP,DEZ
169 MOV A,L ;Einer
170 XRI 30H ;= 1.Nibble auf 3 setzen
171 STAX D ;Abgabe in Ergebnisfeld
172 ;
173 ; Ersetzen der führenden Blanks in Ergebnisfeld
174 LXI H,LFDNR
175 MVI A,30H
176 CALL BLK ;max. 4 führende Blks. setzen
177 BLK: CMP M
178 RNZ ;Ende, wenn Byte ungleich 30H
179 MVI M,20H ;Blank setzen
180 INX H
181 DCR B
182 JNZ BLK ;Schleife
183 RET
184
185 UP,DEZ: MVI A,30H ;Zähler
186 DEZ,LP: DAD B ;= minus BC
187 INR A
188 JC DEZ,LP
189 ;
190 --- Underflow ---
191 PUSH PSM ;Zähler retten
192 ANA A ;C-Bit löschen
193 MOV A,L ;L minus C
194 MOV L,A ;Ergebnis -> L
195 MOV R,H
196 SBB B ;H minus B (incl. Carry)
197 MOV H,R ;Ergebnis in HL
198 POP PSM ;Zähler zurück
199 DCR A
200 STAX D ;Ziffer in Ergebnisfeld
201 INX D ;Adr. in LFDNR +1
202 RET
203
204 ;
205 ; Ausgabe eines Zeichens (in A) auf dem Drucker
206
207 LPRT: PUSH PSM
208 PUSH B
209 PUSH D
210 PUSH H
211 MOV E,A ;Zeichen nach E
212 MVI C,5
213 CALL BDOS
214 POP H
215 POP D
216 POP B
217 POP PSM
218 INR C ;Pos.Zähler auf dem Drucker +1
219 RET
220
221 ;
222 ; 1 Zeichenkette ausgeben (Fehlermeldung)
223
224 TXT: MVI C,9
225 JMP BDOS
226
227 ;
228 FEHLR1: DB CR,LF,7,'Datei nicht vorhanden',CR,LF,'#'
229 FEHLR2: DB CR,LF,7,'Datei zu lang',CR,LF,'#'
230
231 OLDSP: DS 2
232 LOMBYT: DS 1
233 HIBYTE: DS 1 ;Zähler für lfd. Nr.
234 LFDNR: DS 5 ;dezimaler Wert der lfd. Nr.
235 ZCNT: DS 1 ;Zeilenzähler
236 ;
237 DS 32
238 STKTOP:
239 ;
240 TXTPUP: DS 1 ;Startadresse des Textpuffers
241 END START
242 ;
    
```

Listing 2. Der Assembler-Code für die speicherplatzschonende Drucker-Routine

```

1: TITLE Ausgabe einer ASCII-Datei auf dem Drucker
2: SUBTTL LPRINT.MAC --- 05.08.1985 ---
3:
4:
5: ; (C) by Heinz-Dieter Lange, Itzehoe
6: ; im August 1985
7:
8:
9: BDOS EQU 5
10: EOT EQU 100H
11: FCB EQU 5CH
12: TXENDH EQU 9FH ;Highbyte für Textspeicherung
13:
14: CR EQU 00H
15: LF EQU 000H
16: OPENF EQU 15
17: READF EQU 20
18: SETDMR EQU 26
19:
20: ;COMMENT#
21: ; Das Programm LPRINT ist auf Drucker abgestimmt, für die keine besondere
22: ; ESC-Sequenz für den Seitenvorschub vorhanden ist (z.B. Seikosa GP-100R),
23: ; Durch Festlegung von ZWCHSL + ZVRSCH kann die Zahl der je Seite zu drucke
24: ; den Zeilen (die Summe muß für DIN-A4-Hochformat 72 Zeilen sein) variiert
25: ; werden, s.auch Label 'PRT,FF'! #
26:
27: ZWCHSL EQU 64 ;Grundwert Zeilenzähler
28: ZVRSCH EQU 8 ;Leerzeilen als Zeilenvorschub
29:
30:
31: RSEG ORG 100H
32:
33: START: LXI H,0
34: DAD SP
35: SHLD OLDSP ;Stackpointer retten
36: LXI SP,STKTOP ;neu setzen
37:
38: XRA A
39: STA FCB+32
40: LXI D,FCB
41: MVI C,OPENF ;Datei öffnen
42: CALL BDOS
43: CPI -1
44: JNZ LD,DAT
45: --- Datei nicht vorhanden ---
46: LXI D,FEHLR1
47: CALL TXT
48: ENDE: LHL OLDSP
49: SPHL
50: RET
51:
52: --- Datei lesen ---
53: LD,DAT: LXI D,TXTPUF
54: LD1: PUSH D
55: MVI C,SETDMR
56: CALL BDOS
57:
58: LXI D,FCB
59: MVI C,READF
60: CALL BDOS
61: POP H ;DMA-Adresse zurück (nach HL für Addition)
62: ORA A ;A=0, wenn Laden erfolgreich
63: JNZ DRUCK
64:
65: LXI B,00H
66: DAD B
67: XCHG ;Pufferadr. um 128 Bytes (1 CP/M-Satz) erhöhen
68: MVI A,TXENDH
69: DCR A ;erobert minus 256 Bytes
70: CMP D ;hat DMA-Adr., dessen Wert erreicht?
71: JNC LD1 ;nein, Schleife
72: --- Datei zu lang ---
73: LXI D,FEHLR2
74: JMP ENDE-3
75:
76: --- Text drucken ---
77: DRUCK: XRA A
78: STA HIBYTE
79: STA LOMBYT ;hex. Wert lfd.Nr. auf 0
80: LXI H,TXTPUF-1
81:
82: PRINT1: XRA A
83: STA ZCNT ;Zeilenzähler auf 0
84: PRINT2: INX H
85: MOV R,M ;Folgezeichen im Textpuffer
86: CPI EOT
87: JZ ENDE
88: DCR H
89: PUSH H ;Textpufferadr. retten
90: --- lfd.Nr. errechnen ---
91: LHL LOMBYT
92: INX H ;hexad. Wert der lfd.Nr. +1
93: SHLD LOMBYT
94: CALL CHVDEZ ;dez. Wert der lfd.Nr. errechnen
95: --- lfd.Nr. ausgeben ---
96: LXI H,LFDNR
97: MVI B,5
98: PRTNR: MOV R,H
99: CALL LPRT ;Zeichen zum Drucker
    
```

Druckersteuerung leicht gemacht



Auch unter CP/M kann man den Drucker sehr einfach ansprechen. Eine kleine Routine hilft dabei.

Steuerzeichen an den Drucker zu senden ist unter CP/M nicht so einfach wie mit dem normalen Schneider-Betriebssystem. Wenn Sie zum Beispiel von Schön- auf Schnellschrift umschalten wollen, dann finden Sie keinen direkten CP/M-Befehl dafür. Sie können normalerweise nur CP/M verlassen, unter Basic »PRINT #8,CHR\$(27);...« eingeben und mit »ICPM« wieder in den Ausgangsmodus zurückkehren. Wie Sie sehen, ist das aber ein sehr umständlicher Weg. Einfacher geht es da mit dem hier vorgestellten Hilfsprogramm.

Wenn Sie sich mit dem CP/M-Assembler auskennen, dann erstellen Sie das Programm »LETTER.COM« anhand des 8080-Assemblerprogrammes (Listing 1). Wenn nicht, dann sollten Sie zunächst das Basic-Programm (Listing 2) eingeben.

Während der Ausführung des Basic-Laders wird eine Prüfsumme berechnet. Erscheint die Meldung »alles o.k.«, ist das Programm wahrscheinlich fehlerfrei (absolut sicher ist das jedoch leider nie). Andernfalls müssen Sie die dann angegebene fehlerhafte Zeile korrigieren und einen neuen Versuch starten.

Hat dann alles geklappt, haben Sie auf Ihrer Diskette eine neue Datei mit dem Namen »LETTER.COM«. Das ist das eigentliche CP/M-Programm. Sobald dieses fehlerfrei läuft, brauchen Sie das Basic-Programm nicht mehr und können es löschen.

Sowohl am Ende des Basic-Listings als auch am Ende des Assembler-Listings stehen die Steuerzeichenfolgen für die verschiedenen Druckerfunktionen. Jeder Datensatz besteht dabei aus einem ersten Byte, das die Länge der zugehörigen Steuerzeichenfolge angibt und einer anschließenden Folge der eigentlichen Steuercodes.

Die Reihenfolge der Sätze ist dabei folgende:

0.	»Lückenfüller«; nur im Assembler-Listing sichtbar	10.	Doppeldruck
1.	Unterstreichen abschalten	11.	Hochgestellt
2.	Unterstreichen einschalten	12.	Tiefgestellt
3.	Breitschrift ausschalten	13.	Standardzeilenhöhe wählen
4.	Breitschrift einschalten	14.	Zeilenhöhe 1/2
5.	Schnellschrift einschalten	15.	Zeilenhöhe 1
6.	komprimierte Zeichen	16.	Zeilenhöhe 1 1/2
7.	Fettdruck	17.	Zeilenhöhe 2
8.	Schönschrift (NLQ)	18.	Zeilenhöhe 2 1/2
9.	Zeichen normal ausgeben	19.	Zeilenhöhe 3

Tabelle 1. Die verschiedenen Parameter rufen auf dem NLQ 401 diese Schriftarten auf

Wenn Ihr Drucker irgendeine Funktion nicht bietet, dann sollten Sie die Zeichenfolge durch einen »unschädlichen String« ersetzen. Also etwa durch ein Byte 00hex in der Länge der Steuerzeichenfolge. Keinesfalls dürfen Sie den Datensatz aber ersatzlos wegfassen lassen.

Die hier vorliegende Version ist für den NLQ 401 von Schneider angepaßt. Die meisten Parameter werden aber auch von jedem Epson-kompatiblen Gerät verstanden. Der Aufruf der Routine erfolgt mit »LETTER abcd e«. Wie die einzelnen Variablen dabei zu wählen sind, das sehen Sie in Tabelle 2. Diesen Hilfszettel können Sie jederzeit mit »LETTER ?« aufrufen. Bei falschen Parametern wird er automatisch ausgegeben.

(Helmut Tischer)

Aufruf: 1) »LETTER ?«	:	Helptexte
2) »LETTER abcd	1« :	
a: Hervorhebung 1	0 →	nicht unterstreichen
	1 →	unterstreichen
b: Hervorhebung 2	0 →	keine Breitschrift
	1 →	Breitschrift
c: Schriftart	0 →	Schnellschrift
	1 →	Schmallschrift
	2 →	Fettdruck
	3 →	Near Letter Quality
d: Schriftvarianten:	0 →	normal
	1 →	Doppeldruck
	2 →	Hochgestellt
	3 →	Tiefgestellt
e: Zeilenabstand	1 →	0,5 Zeilen (1/12 Zoll)
	2 →	1 Zeile (1/6 Zoll)
	3 →	1,5 Zeilen (1/4 Zoll)
	...	
	6 →	3 Zeilen (1/2 Zoll)

Tabelle 2. Standardeinstellung für fehlende Parameter: »0« beziehungsweise 1/6-Zoll-Zeilenabstand. In der Schriftart »NLQ« werden die Schriftvarianten nicht ausgewertet.

```

; Verschiedene Schriftarten auf dem NLQ401
;
; (c) 26.09.1985 by
; Helmut Tischer
; Astenstraße 40, D-8052 Moosburg
;
;***** Konstanten
bdos equ 0005h
word1 equ 005dh
word2 equ 006dh
write equ 09h
print equ 05h
stack equ 8000h
org 0100h
;
;***** Programmstart
jmp start
db 'Verschiedene Schriftarten auf dem "NLQ401"',13,10
db '(c) 26.09.1985 by',13,10
db 'Helmut Tischer',13,10
db 'Astenstraße 40, D-8052 Moosburg',13,10,26
;
;***** Lokalen Stapel anlegen
start: lxi h,0000h
dad sp ;alter Stackpointer
shld stack-2 ;Zwischenspeichern
lxi h,stack-2 ;neuer Stack
sphi
;
;***** Hauptprogramm
repeat: lda word1 ;1.Parameter
cpi '?'
jz help ;Hilftext
lxi b,0201h ;2 Einträge ab 1. Tabellensatz
call switch
lda word1+1 ;2. Parameter
lxi b,0203h ;2 Einträge ab 3. Tabellensatz
call switch
lda word1+2 ;3. Parameter
lxi b,0405h ;4 Einträge ab 5. Tabellensatz
call switch
lda word1+3 ;4. Parameter
lxi b,0409h ;4 Einträge ab 9. Tabellensatz
call switch
lda word2 ;5. Parameter
lxi b,070dh ;7 Einträge ab 13. Tabellensatz
call switch
jmp finis ;Programmende
;
;***** Zuständigen Tabellensatz suchen und ausgeben
switch: cpi ' ' ;Leerzeichen gilt als 0
jnz char
mvi a,30h
char: sui '0' ;Direkte Codierung berechnen
jc error ;wenn kleiner, dann Fehler
cmp b ;Maximaler Wert
jnc error
add c ;Versatz
lxi h,tabell;Tabellenanfang
mvi d,00h
such: mov e,e ;Satzlänge
inc h
dad d ;überspringen
dcr a
jnz such ;Solange, bis gefunden
mov b,e ;Länge des gültigen Satzes
inc h
mvi c,print ;Steuercode Drucker Ausgabe

```

Listing 1. Das 8080-Assembler-Listing

```

druck: mov     e, a      ;Auszugebendes Zeichen
        inc     h
        push   h
        push   b
        call  bdos      ;Zeichen an Drucker
        pop    b
        pop    h
        dcr   b
        jnz   druck     ;noch ein Zeichen?
        ret

;
;***** Fehlermeldung
error:  pop     h
        lxi   d, fehler; Meldetext 'Falscher Parameter'
        mov  c, write; Text ausgeben
        call bdos
        lxi  h, word1 ;Eingabe löschen
        mov  a, ' '
        mov  m, a
        inc  h
        mov  m, a
        inc  h
        mov  m, a
        lxi  h, word2
        mov  m, a
        jmp  repeat ;Programm mit Standarteinstellung

;
;***** Hilfstexte ausgeben
help:   lxi   d, hilfen
        mov  c, write
        call bdos

;
;***** Programmende
finis:  lhld  stack-2 ;alter Stackpointer
        sphl ;zurück
        ret        ;Sprung ins System

;
;***** Meldetexte
fehler: db 24, 'Falscher Parameter!', 24, 13, 10
hilfen: db 13, 10, 24, 'Schriftartenwahl auf dem 'NL0401'', 24
        db 13, 10, 'Aufruf: 1) 'LETTER?', 13, 10
        db 2) 'LETTER abcd 1', 13, 10
        db 'a: Hervorhebung 1, 9, 0 -> nicht unterstreichen', 13, 10
        db 9, 9, 9, '1 -> unterstreichen', 13, 10
        db 'b: Hervorhebung 2, 9, 0 -> keine Breitschrift', 13, 10
        db 9, 9, 9, '1 -> Breitschrift', 13, 10
        db 'c: Schriftart', 9, 9, 0 -> Schnellschrift', 13, 10
        db 9, 9, 9, '1 -> Schmalschrift', 13, 10
        db 9, 9, 9, '2 -> Fettdruck', 13, 10
        db 9, 9, 9, '3 -> Near Letter Quality', 13, 10
        db 'd: Schriftvarianten: 9, 0 -> normal', 13, 10
        db 9, 9, 9, '1 -> Doppeldruck', 13, 10
        db 9, 9, 9, '2 -> Hochgestellt', 13, 10
        db 9, 9, 9, '3 -> Tiefgestellt', 13, 10
        db 'e: Zeilenabstand', 9, 1 -> 0.5 Zeilen (1/2 ' '), 13, 10
        db 9, 9, 9, '2 -> 1 Zeile (1/6 ' '), 13, 10
        db 9, 9, 9, '3 -> 1.5 Zeilen (1/4 ' '), 13, 10
        db 9, 9, 9, '...', 13, 10
        db 9, 9, 9, '6 -> 3 Zeilen (1/2 ' '), 13, 10
        db 'Standarteinstellung für fehlende Parameter: '0' bzw.
        db '1/6 ' Zeilenabstand', 13, 10
        db 'In der Schriftart 'NL0' werden die Schriftvarianten
        db 'nicht ausgewertet.',

;
;***** Steuerzeichenfolgen (jeweils Länge + Parameter)
tabell: db 00 ;Dummy, '0-ter' Tabellensatz
        db 3, 1bh, 2dh, 00h ;nicht unterstreichen
        db 3, 1bh, 2dh, 01h ;unterstreichen
        db 3, 1bh, 57h, 00h ;keine Breitschrift
        db 3, 1bh, 57h, 01h ;Breitschrift
        db 6, 12h, 1bh, 46h, 1bh, 78h, 00h ;Schnellschrift
        db 6, 0fh, 1bh, 46h, 1bh, 78h, 00h ;Schmalschrift
        db 5, 1bh, 45h, 1bh, 78h, 00h ;Fettdruck
        db 3, 1bh, 78h, 01h ;NL0
        db 4, 1bh, 48h, 1bh, 54h ;keine Variante
        db 4, 1bh, 47h, 1bh, 54h ;Doppeldruck
        db 3, 1bh, 53h, 00h ;hochgestellt
        db 3, 1bh, 53h, 01h ;tiefgestellt
        db 5, 1bh, 41h, 0ch, 1bh, 32h ;Zeilenabstand 1/6 "
        db 5, 1bh, 41h, 06h, 1bh, 32h ;1/12 "
        db 5, 1bh, 41h, 0ch, 1bh, 32h ;1/6 "
        db 5, 1bh, 41h, 12h, 1bh, 32h ;1/4 "
        db 5, 1bh, 41h, 18h, 1bh, 32h ;1/3 "
        db 5, 1bh, 41h, 1eh, 1bh, 32h ;5/12 "
        db 5, 1bh, 41h, 24h, 1bh, 32h ;1/2 "
    
```

Listing 1. Das 8080-Assembler-Listing (Schluß)

```

230 PRINT#9, p$; [2B32]
240 READ d$:WEND [8AC4]
250 [E1BC]
260 CLOSEOUT:PRINT"Alles O.K.":END [3792]
270 [E3C0]
280 [0CC2]
290 'Hauptprogramm [4A92]
300 DATA C3,77,01,56,65,72,73,63, 830 [F8BE]
310 DATA 68,69,65,64,65,6E,65,20, 754 [0DE6]
320 DATA 53,63,68,72,69,66,74,61, 820 [89BC]
330 DATA 72,74,65,6E,20,61,75,66, 789 [E6E6]
340 DATA 20,64,65,6D,20,27,4E,4C, 567 [D9FE]
350 DATA 51,34,30,31,27,0D,0A,28, 332 [04AC]
360 DATA 63,29,20,32,36,2E,30,39, 427 [52B4]
370 DATA 2E,31,39,38,35,20,62,79, 512 [69C0]
380 DATA 0D,0A,09,48,65,6C,6D,75, 539 [3A34]
390 DATA 74,20,54,69,73,63,68,65, 756 [C3D2]
400 DATA 72,0D,0A,09,41,73,74,65, 543 [F8CC]
410 DATA 72,6E,73,74,72,61,7E,65, 893 [F106]
420 DATA 20,34,30,2C,20,44,2D,38, 377 [40BA]
430 DATA 30,35,32,20,4D,6F,6F,73, 597 [4204]
440 DATA 62,75,72,67,0D,0A,1A,21, 514 [22E0]
450 DATA 00,00,39,22,FE,7F,21,FE, 759 [0C3C]
460 DATA 7F,F9,3A,5D,00,FE,3F,CA, 1046 [5BE0]
470 DATA 02,02,01,01,02,CD,B7,01, 397 [85C2]
480 DATA 3A,5E,00,01,03,02,CD,B7, 546 [8312]
490 DATA 01,3A,5F,00,01,05,04,CD, 369 [BBF4]
500 DATA B7,01,3A,60,00,01,09,04, 352 [C796]
510 DATA CD,B7,01,3A,6D,00,01,0D, 570 [DB20]
520 DATA 07,CD,B7,01,C3,0A,02,FE, 857 [F962]
530 DATA 20,C2,BE,01,3E,30,D6,30, 789 [0F22]
540 DATA DA,E6,01,88,D2,E6,01,81, 1203 [185E]
550 DATA 21,07,05,16,00,5E,23,19, 221 [1D90]
560 DATA 3D,C2,CD,01,46,23,0E,05, 585 [5728]
570 DATA 5E,23,ED,C5,CD,05,00,C1, 958 [8B5A]
580 DATA E1,05,C2,DB,01,C9,E1,11, 1084 [334A]
590 DATA 0F,02,0E,09,CD,05,00,21, 283 [D2F4]
600 DATA 5D,00,3E,20,77,23,77,23, 495 [80D6]
610 DATA 77,23,77,21,6D,00,77,C3, 729 [4AEA]
620 DATA 82,01,11,29,02,0E,09,CD, 419 [56DE]
630 DATA 05,00,2A,FE,7F,F9,C9,18, 902 [2764]
640 DATA 20,46,61,6C,73,63,68,65, 726 [D3D6]
650 DATA 72,20,50,61,72,61,6D,65, 744 [1EC0]
660 DATA 74,65,72,20,21,20,18,0D, 465 [F380]
670 DATA 0A,0D,0A,18,2D,53,63,68, 375 [DEEC]
680 DATA 72,69,66,74,61,72,74,65, 865 [71E0]
690 DATA 6E,77,61,68,6C,20,61,75, 784 [C510]
700 DATA 66,20,64,65,6D,20,27,4E, 593 [2AE6]
710 DATA 4C,51,34,30,31,27,20,18, 401 [1A8C]
720 DATA 0D,0A,41,75,66,72,75,66, 640 [10E0]
730 DATA 3A,20,31,29,20,27,4C,45, 396 [60CC]
740 DATA 54,54,45,52,20,3F,27,20, 485 [34BC]
750 DATA 20,20,20,20,20,3A,20,48, 322 [F172]
760 DATA 65,6C,70,74,65,78,74,65, 875 [9D02]
770 DATA 0D,0A,20,20,20,20,20,21, 215 [3E82]
780 DATA 20,20,32,29,20,27,4C,45, 371 [4AA6]
790 DATA 54,54,45,52,20,61,62,63, 645 [68AA]
800 DATA 64,20,6C,27,20,3A,0D,0A, 392 [47F2]
810 DATA 61,3A,20,48,65,72,76,6F, 703 [BAE6]
820 DATA 72,68,65,62,75,6E,67,20, 779 [15F6]
830 DATA 31,09,30,20,2D,3E,20,6E, 387 [D7E8]
840 DATA 69,63,68,74,20,75,6E,74, 799 [1508]
850 DATA 65,72,73,74,72,65,69,63, 865 [CADE]
860 DATA 68,65,6E,0D,0A,09,09,09, 365 [1514]
870 DATA 31,20,2D,3E,20,75,6E,74, 563 [47FE]
880 DATA 65,72,73,74,72,65,69,63, 865 [9E4E]
890 DATA 68,65,6E,0D,0A,62,3A,20, 526 [5B1E]
900 DATA 48,65,72,76,6F,72,68,65, 835 [C1FC]
910 DATA 62,75,6E,67,20,32,09,30, 567 [10CA]
920 DATA 20,2D,3E,20,68,65,69,6E, 594 [042C]
930 DATA 65,20,42,72,65,69,74,73, 750 [2EBA]
940 DATA 63,68,72,69,66,74,0D,0A, 663 [7102]
950 DATA 09,09,09,31,20,2D,3E,20, 247 [33CC]
960 DATA 42,72,65,69,74,73,63,68, 820 [40D0]
970 DATA 72,69,66,74,0D,0A,63,3A, 617 [D612]
980 DATA 20,53,63,68,72,69,66,74, 755 [A3D8]
990 DATA 61,72,74,09,09,30,20,2D, 470 [0BBE]
1000 DATA 3E,20,53,63,68,6E,65,6C, 699 [D076]
1010 DATA 6C,73,63,68,72,69,66,74, 863 [A04C]
1020 DATA 0D,0A,09,09,09,31,20,2D, 176 [C42E]
1030 DATA 3E,20,53,63,68,6D,61,6C, 694 [EC68]
1040 DATA 73,63,68,72,69,66,74,0D, 768 [E150]
1050 DATA 0A,09,09,09,32,20,2D,3E, 226 [5F36]
1060 DATA 20,64,65,74,74,64,72,75, 766 [8D20]
1070 DATA 63,6B,0D,0A,09,09,09,33, 307 [5C46]
1080 DATA 20,2D,3E,20,4E,65,61,72, 561 [504A]
1090 DATA 20,4C,65,74,74,65,72,20, 688 [0934]
1100 DATA 51,75,61,6C,69,74,79,0D, 758 [9760]
1110 DATA 0A,64,3A,20,53,63,68,72, 600 [0418]
1120 DATA 69,66,74,76,61,72,69,61, 854 [E730]
1130 DATA 6E,74,65,6E,3A,09,30,2D, 584 [A55C]
1140 DATA 2D,3E,20,6E,6F,72,6D,61, 680 [999E]
1150 DATA 6C,0D,0A,09,09,09,31,2D, 239 [A53C]
1160 DATA 2D,3E,20,44,6F,70,70,65, 643 [7558]
1170 DATA 6C,64,72,75,63,6B,0D,0A, 668 [D28A]
1180 DATA 09,09,09,32,2D,3E,20, 248 [4B28]
1190 DATA 48,6F,63,68,67,65,73,74, 821 [C958]
1200 DATA 65,6C,6C,74,0D,0A,09,09, 474 [4276]
1210 DATA 09,33,2D,2D,3E,20,54,69, 420 [E51A]
1220 DATA 65,66,67,65,73,74,65,6C, 847 [1F54]
1230 DATA 6C,74,0D,0A,65,3A,20,5A, 528 [567E]
    
```

Listing 2. Der Basic-Lader für »LETTER.COM«

```

10 '***** [9FCC]
20 '* (C) HELMUT TISCHER * [2224]
30 '* ASTERNWEG 40, 8052 MOOSBURG * [7D36]
40 '***** [E9D2]
50 'Basic-Lader fuer die Datei 'LETTER.C
   OM' [D532]
60 [845A]
70 OPENOUT"LETTER.COM" [C8A4]
80 [865E]
90 '1. Programmteil: mit Pruefsumme [6A1A]
100 FOR zeile=10010 TO 11290 STEP 10 [777A]
110 pruef=0 [EB82]
120 FOR i=0 TO 7 [2B54]
130 READ d$:d=VAL("&"+d$) [7D48]
140 PRINT#9,CHR$(d);:pruef=pruef+d [AF0A]
150 NEXT [CDEA]
160 READ p [16C6]
170 IF pruef<>p THEN PRINT"Pruefsummenfe
   hler in Zeile"zeile:CLOSEOUT:END [C344]
180 NEXT [51F0]
190 ' [05C2]
200 '2. Programmteil: ohne Pruefsumme [4432]
210 READ d$:WHILE d$<>"Ende" [A018]
220 IF LEFT$(d$,1)="" THEN p$=RIGHT$(d$,
   1) ELSE p$=CHR$(VAL("&"+d$)) [3AFE]
    
```

```

1240 DATA 65,69,6C,65,6E,61,62,73, 835 [F566]
1250 DATA 74,61,6E,64,09,31,20,2D, 558 [BD40]
1260 DATA 3E,20,30,2E,35,20,5A,65, 464 [5240]
1270 DATA 69,6C,65,6E,20,2B,31,2F, 592 [7574]
1280 DATA 31,32,20,22,29,0D,0A,09, 238 [3714]
1290 DATA 09,09,32,20,2D,3E,20,31, 288 [492A]
1300 DATA 20,20,20,5A,65,69,6C,65, 601 [031E]
1310 DATA 20,20,28,31,2F,36,20,20, 318 [6AE4]
1320 DATA 22,29,0D,0A,09,09,09,33, 176 [8326]
1330 DATA 20,2D,3E,20,31,2E,35,20, 351 [A422]
1340 DATA 5A,65,69,6C,65,6E,20,28, 687 [2284]
1350 DATA 31,2F,34,20,20,22,29,0D, 300 [ED00]
1360 DATA 0A,09,09,09,2E,2E,2E,0D, 188 [AF94]
1370 DATA 0A,09,09,09,36,20,2D,3E, 230 [963E]
1380 DATA 20,33,20,20,20,5A,65,69, 475 [6A08]
1390 DATA 6C,65,6E,20,2B,31,2F,32, 537 [B264]
1400 DATA 20,20,22,29,0D,0A,53,74, 361 [1108]
1410 DATA 61,6E,64,61,72,74,65,69, 840 [FF3A]
1420 DATA 6E,73,74,65,6C,6C,75,6E, 885 [9EB4]
1430 DATA 67,20,66,7D,72,20,66,65, 711 [4026]
1440 DATA 68,6C,65,6E,64,65,20,50, 736 [9658]
1450 DATA 61,72,61,6D,61,74,65,72, 845 [2938]
1460 DATA 3A,20,27,30,27,20,62,7A, 468 [AC28]
1470 DATA 77,2E,20,31,2F,36,20,22, 413 [9920]
1480 DATA 20,5A,65,69,6C,65,6E,61, 744 [A07C]
1490 DATA 62,73,74,61,6E,64,0D,0A, 659 [BC74]
1500 DATA 49,6E,20,64,65,72,20,53, 645 [0A22]
1510 DATA 63,68,72,69,66,74,61,72, 851 [B62A]
1520 DATA 74,20,27,4E,4C,51,27,20, 493 [2832]
1530 DATA 77,65,72,64,65,6E,20,64, 777 [A44C]
1540 DATA 69,65,20,53,63,68,72,69, 743 [042A]
1550 DATA 66,74,76,61,72,69,61,6E, 859 [9560]
1560 DATA 74,65,6E,20,6E,69,63,68, 777 [287E]

1570 DATA 74,20,61,75,73,67,65,77, 800 [EE1E]
1580 DATA 65,72,74,65,74,2E,24,00, 630 [7326]
1590 ' [BE2C]
1600 'Hier beginnen die Druckerstrings [5390]
1610 'Unterstreichen aus/ein [FD6C]
1620 DATA 03,1B,2D,00 [1BA6]
1630 DATA 03,1B,2D,01 [F1AA]
1640 'Gro~schrift aus/ein [2CFE]
1650 DATA 03,1B,57,00 [6698]
1660 DATA 03,1B,57,01 [829C]
1670 'Schnell/Schmal/Fett/Schoen [546C]
1680 DATA 06,12,1B,46,1B,7B,00 [DD32]
1690 DATA 06,0F,1B,46,1B,7B,00 [0C5A]
1700 DATA 05,1B,45,1B,7B,00 [D002]
1710 DATA 03,1B,7B,01 [329A]
1720 'normal/Doppeldruck/Hoch/Tief [B75C]
1730 DATA 04,1B,48,1B,54 [EBE8]
1740 DATA 04,1B,47,1B,54 [30E8]
1750 DATA 03,1B,53,00 [F392]
1760 DATA 03,1B,53,01 [0796]
1770 'Zeilenhoehe: Standarteinstellung [E396]
1780 DATA 05,1B,41,0C,1B,32 [5A1C]
1790 'Zeilenhoehen 1/2, 1,1 1/2, ...,3 [DAD2]
1800 DATA 05,1B,41,06,1B,32 [74F4]
1810 DATA 05,1B,41,0C,1B,32 [7D10]
1820 DATA 05,1B,41,12,1B,32 [4BF2]
1830 DATA 05,1B,41,18,1B,32 [B000]
1840 DATA 05,1B,41,1E,1B,32 [E61C]
1850 DATA 05,1B,41,24,1B,32 [BCFE]
1860 DATA Ende [6B4A]

```

Listing 2. Der Basic-Lader für »LETTER.COM« (Schluß)

Verschiedene Tastaturen unter CP/M



Was man in Basic einfach mit den Befehlen »KEY DEF« erreicht, ist unter CP/M manchmal ein schwieriges Unterfangen. Die Tastatur läßt sich zum Beispiel nur mit ein paar Tricks umdefinieren.

Normalerweise kann man unter CP/M die Tastenbelegung nur ändern, wenn man das Programm »SETUP« auf der Schneider-Systemdiskette, beziehungsweise »Install« auf der Systemdiskette von Vortex, aufruft. Oder man benutzt eine schon vorbereitete Diskette mit der gewünschten Tastenbelegung. Angenehmer ist es, wenn man eine beliebige Tastatur »mit Tastendruck« aufrufen kann. Also beispielsweise einfach »WSTASTEN« eingeben und schon steht eine für »Wordstar« besonders geeignete Belegung zur Verfügung. Das ist vor allem dann wichtig, wenn man verschiedene Programme besitzt, die unterschiedliche Codes zur Cursorsteuerung benutzen. Je nach Anwendungszweck passende Funktionstasten sollten sich leicht aufrufen lassen. Mit der hier vorgestellten Routine geht das.

Alle »...TASTEN.COM«-Programme haben einen gemeinsamen Kernteil und einen individuellen Anhang, der die spezielle Tastenbelegung enthält. Beim Aufruf durch den Namen setzt der Kernteil zunächst die Tastatur in den Ausgangszustand zurück und belegt sie dann mit den im Anhang angegebenen Codes. Dabei kann die Nummer der Funktionstaste als auch der String, der auf diese gelegt werden soll, frei gewählt werden. Zum Schluß wird noch eine vom Benutzer bestimmte Meldung ausgegeben, die beispielsweise den Namen der Tasten-

belegung oder eine Farb-/Modeeinstellung des Bildschirms enthalten kann.

Für alle, die Ihre Programme nur in Basic eingeben wollen, ist Listing 2 gedacht. Tippen Sie es zunächst bis zur Zeile 10 400 ein. Das wird später der Kernteil des COM-Programmes. Am besten speichern Sie die Zeilen dann, um später die Grundfunktionen immer wieder zur Verfügung zu haben.

In den Zeilen 100 und 120 können Sie den Namen eintragen, den später Ihre Tastenbelegung haben soll. Achten Sie aber darauf, daß der Namensteil nach dem Punkt unbedingt »COM« heißen muß. Da unser Beispiel das Programm »Wordstar« verbessern soll, heißt die Routine aus Listing 2 »WSTASTEN«.

In der Zeile 10 410 beginnen die Definitionen der maximal 32 Funktionsstrings mit Nummern zwischen 0 und 31. Sinnvollerweise schreibt man in eine Zeile genau einen Funktionsstring, so daß an erster Stelle immer die Nummer des Strings steht. An zweiter Stelle steht dann die Länge und an den folgenden Stellen die Zeichen, die den String bilden.

Dabei haben Sie die Wahl zwischen zwei verschiedenen Wegen:

— Wenn es sich um ein Zeichen mit ASCII-Code handelt (beispielsweise »A«), dann schreiben Sie ein Ausrufezeichen, gefolgt von dem jeweiligen Zeichen. Diese Methode kann jedoch nicht bei Symbolen benutzt werden, die im Basic eine besondere Bedeutung haben. Diese müssen mit Hilfe des zweiten Weges eingegeben werden.

— Wenn es sich um ein Zeichen handelt, das nicht darstellbar ist (zum Beispiel Wagenrücklauf), dann müssen Sie es hexadezimal eingeben. Soll im String also ein Anführungszeichen vorkommen, so muß es mit Code 22 eingegeben werden (ohne vorausgehendes »&«).

Die Strings dürfen dabei übrigens auch durcheinander im Programm stehen und — auch wichtig — es müssen nicht alle Strings definiert werden. Das Ende der Tabelle wird mit einem Byte FF hex gekennzeichnet.

Unmittelbar nach der Stringtabelle (in unserem Beispiel ab Zeile 10 760) folgt die Belegung der Tastatur. Der Aufbau einer Zeile der Tabelle ist mit Ausnahme des Repeat-Flags (Wiederholungsfunktion) analog zu einem »KEY DEF«-Befehl. In jeder Zeile steht zunächst die Tastennummer, danach der Code der Taste im Normalmodus. Danach der Code der Taste, wenn Sie zusammen mit Shift gedrückt ist und zum Schluß der Code der Taste im Ctrl-Modus. Die Codes 80 bis 9Fhex repräsentieren jeweils einen Funktionsstring mit einer Nummer zwischen 00 und 1Fhex. Sie können also die 32 oben definierten Funktio-

nenstrings nicht nur auf den Zahlenblock der Tastatur legen, sondern auf wählbare Tasten.

Die Eingabe der Codes erfolgt genau wie bei den Funktionenstrings entweder als Hexzahl oder als » ! «-Zeichen-Kombination. Die Länge der Tabelle ist hier ebenfalls beliebig und wird durch ein Byte mit dem Wert FF hex gekennzeichnet.

Als letztes folgt die Meldung, die beim Aufrufen der Routine ausgegeben werden soll. Der Meldetext muß hier allerdings durch ein Zeichen »\$« mit dem Code 24 hex beendet werden. Sinnvoll ist, in diesem Text auch gleich die Definition der Bildschirmfarben und des Bildschirmmode einzufügen.

Das Ende des Datenbereichs ist im Programm durch den Text »Ende« gekennzeichnet.

Geben Sie jetzt das Programm vollständig ein und starten Sie es. Wenn sich ein Fehler im Programm befindet, wird Ihnen die Nummer der fehlerhaften Zeile mitgeteilt. Wenn das Programm korrekt ist (also die Prüfsumme auch stimmt), dann erscheint die Meldung »Alles o.k.«. Auf Ihrer Diskette befindet sich nun eine neue Datei mit dem Namen, der in der Zeile 120 hinter »OPENOUT« angegeben ist. Das Basic-Programm brauchen Sie jetzt nicht mehr — außer Sie möchten die Tastenbelegung noch einmal ändern.

Schalten Sie nun in den CP/M-Modus um und tippen Sie den Namen der neuen Datei (ohne »COM«). Sofort steht die neue Tastenbelegung zur Verfügung. Wenn Sie im Basic-Programm den Namen in Zeile 120 ändern, können Sie auf derselben Diskette mehrere Tastaturbelegungen mit verschiedenen Namen ablegen.

Falls das Programm nicht funktioniert, dann befindet sich ein vom Prüfsummenzähler nicht entdeckter Fehler im Basic-Programm. Das kann beispielsweise eine vollständig vergessene DATA-Zeile sein. Dann hilft nur genaues Überprüfen der einzelnen Zeilen.

Listing 1 enthält den Kernteil aller Tastenbelegungen in 8080-Assemblercode. Dieses können Sie analog zu der oben angegebenen Beschreibung mit Ihrer persönlichen Tastenbelegung ergänzen und mit Hilfe des CP/M-Assemblers »ASM« und dem »LOAD«-Programm übersetzen. Der Aufruf des entstandenen COM-Programms belegt dann die Tastatur entsprechend Ihren Wünschen.

Das abgedruckte Beispiel-Programm enthält die Funktions-tastenbelegung für ein »Wordstar«-Programm. Diese Tastenbelegung wurde so ausgewählt, daß wirklich jede einzelne freie Taste sinnvoll belegt ist. Erst jetzt ist das Arbeiten mit »Wordstar« optimal möglich. In Tabelle 1 finden Sie die Tastenbelegung dieses Beispiels.

Natürlich können Sie eigene Änderungen einbauen — beispielsweise bei den Sonderzeichen, die wegen der deutschen Umlaute umpositioniert werden mußten. Beachten Sie aber, daß bei einer Standard-Tastatur die Lage der Control-Codes durch die übrigen Tasten vorgegeben ist. Wenn Sie sich nicht daran halten, kommen Sie bei komplexen Programmen leicht in Schwierigkeiten.

Alle Tasten, die mit Codes zwischen 40 und 55 hex belegt sind, müssen in der Ctrl-Ebene mit dem um 40 hex verminderten Code belegt sein. Die Taste »A« (mit dem Code 41 hex) liefert also in der Ctrl-Ebene den Wert 41 — 40 = 01 hex. Wenn Sie die »Y«- und die »Z«-Taste vertauschen oder die Umlaute auf der Tastatur positionieren wollen, müssen Sie das beachten. Eine unbelegte Taste müssen Sie mit FF hex kennzeichnen und nicht mit 00 hex. (Helmut Tischer)

ESC:	Hilfe (1J) Befehl abbrechen (1U) Escape (1I)	CLR:	nach rechts bis Ende löschen (1QY) rechtes Wort löschen (1T) rechts Zeichen löschen (1G)
TAB:	Tabulator löschen (1ON) Tabulator setzen (1OI) Tabulator ausführen (1I)	DEL:	links bis Anfang löschen (1QDEL) ganze Zeile löschen (1Y) linkes Zeichen löschen (DEL)

CAPS LOCK:	Randlöser (10X) Großbuchstaben feststellen Großbuchstaben feststellen	ENTER:	Einfügemodus ein/aus (1V) Leerzeile einfügen (1N) Absatz (CR)
SPACE:	Suchen wiederholen (1L) allgemein wiederholen (1QQ) Leertaste	CUR UP:	Zum Textanfang (1QR) Seite zurück (1R) Zeile nach oben (1E)
CUR LEFT:	zum linken Rand (1QS) Wort nach links (1A) Zeichen links (1S)	COPY:	Hilfe (1J) Absatz formatieren (1B) Cursor Home (1QS1QE)
CUR DOWN:	Zum Textende (1QC) Seite vorblättern (1C) Zeile runter (1X)	CUR RIG:	Zum rechten Rand (1QD) Wort rechts (1F) Zeichen rechts (1D)
F1:	abwärts rollen (1Z) Blocksatz ein/aus (1OJ)	F7:	Block speichern (1KW) Blockanfang merken (1KB)
F2:	Directory ein/aus (1KF) Absatz einrücken (1OG)	F8:	Block/Datei einfügen (1KR) Blockende merken (1KK)
F3:	aufwärts rollen (1W) Zeile zentrieren (1OC)	F9:	Block löschen (1KY) Block invertieren (1KH)
F4:	Text sichern (1KS) Laufwerk wechseln (1KL)	F0:	Rand links zum Cur. (1OL1) Linken Rand stellen (1OL)
F5:	Text beenden (1KD) Datei umbenennen (1KE)	F:	Cursor an letzte Position (1QP) Rechten Rand stellen (1R)
F6:	(1M1KRBRIEFVOR1M) Datei löschen (1KJ)	FENTER:	Merker setzen (1K) + Ziffer Merker suchen (1Q) + Ziffer neue Zeile, ohne Absatz (1QS1X)

1. Zeile: Taste + Ctrl
2. Zeile: Taste + Shift
3. Zeile: Taste allein

Hinweis zur Taste F6: Nach Eingabe des Dateinamens beim Start von WS statt Enter die Taste Ctrl+F6=> alle wichtigen Voreinstellungen für Brief.
Hinweis zur Taste Fenter: Unmittelbar nach Merker setzen/löschen muß eine Zahl eingegeben werden, um Merker genauer zu bestimmen.
Die mit F gekennzeichneten Zahlen beziehen sich auf den Ziffernblock.

Tabelle 1. Die Tastaturbelegung

```

:           Tastenbelegung für Schneider CPC464 frei wählen
:           unter CP/M 2.2
:           (c) 11.09.1985 by Helmut Tischer
:
:***** Kernprogramm *****
:
org         0100h
oldso      equ     1000h      ;Zwei unbelegte Speicherzellen
stktop    equ     5000h      ;Lokaler Stapel im zentralen RAM
:
:           Lokalen Stapel initialisieren
entrvt:    lxi     h,0        ;alten Stapel holen
           dad     sp
           shld   oldso      ;retten
           lxi     sp,stktop;neuer Stapel(im zentralen RAM!)
           call   0bb00h     ;Keyboard initialisieren
:
:           Expansionsstrings setzen
morex:    lxi     h,expan     ;Adresse der Stringtabelle
           mov     a,m        ;Stringnummer
           inx     h
           inr     a          ;255=Ende der Tabelle
           jz     noexp
           dcr     a
           mov     c,m        ;Länge eines Satzes
           inx     h
           push   h          ;(Stacktop) auf Stringanfang
           mvi    b,0
           dad     b         ;h1 zeigt auf Stringende
           xthl
           mov     b,a        ;b=Nummer, c=Länge
           call   0bb0fh     ;KM SET EXPAND
           pop     h          ;Adresse nächster Satz
           jmp    morex
:
:           Tastenbelegungen setzen
noexp:    mov     a,m        ;Nummer der Taste
           inx     h
           inr     a          ;255=Ende der Tabelle
           dcr     a
           mov     b,m        ;Code Translate
           inx     h
           push   psw        ;a ebthaelst Tastennummer
           push   h
           call   0bb27h     ;KM SET TRANSLATE
           pop     h
           pop     psw
           mov     b,m        ;Code Shift
           inx     h
           push   psw        ;a enthaelt Tastennummer
           push   h
           call   0bb2dh     ;KM SET SHIFT
           pop     h
           pop     psw
           mov     b,m        ;Code CTRL
           inx     h
           push   h
           call   0bb33h     ;KM SET CTRL
           pop     h
           jmp    noexp      ;naechste Taste
:

```

Listing 1. Das 8080-Assembler-Listing

Immer auf dem neuesten Stand



Samstag abend, 18 Uhr 30. Jetzt kommt die Tabelle der Fußball-Bundesliga in der Sportschau. »Alte Kammelle,« können Sie sagen, wenn Sie »Ligaverwaltung« benutzen.

Interessieren Sie sich auch für einen Sportverein, dessen Ergebnisse nie in der Zeitung stehen? Dann kennen Sie die Probleme, den aktuellen Tabellenstand des nächsten Gegners zu erfahren. Hier — aber auch bei Vereinen, die regelmäßig in den Medien auftauchen — hilft »Ligaverwaltung«. Der Weg »zu Fuß« (Kopfrechnen) wird vom Computer übernommen. Sie geben nur noch die neuesten Ergebnisse ein und Ihr Schneider sagt Ihnen, wo Ihr Favorit gerade steht.

Und das für jeden Verein in jeder Sportart (die in einer Liga, bei der es um Tore und Punkte geht, gespielt wird). Das Programm »Ligaverwaltung« ist menügesteuert und damit sehr leicht zu bedienen. Das Hauptaugenmerk wurde auf Komfort, Übersicht und Benutzerfreundlichkeit gelegt. Fünf der 15 Punkte im Hauptmenü behandeln die Ergebnisse von drei selbstdefinierten Gruppen. Die Daten (Spielpaarungen, Vereinsnamen und so weiter) für diese fünf werden an das Programm mit DATA-Zeilen übergeben. Die ersten neun Routinen dienen der allgemeinen Ligaverwaltung, so daß (theoretisch) jeder Verein verwaltet werden kann. Im folgenden werden die einzelnen Menüpunkte erklärt.

Menü mit vielen Punkten

1. Tabelle aufbauen

Die Überschrift erscheint, wie bei jedem Menüpunkt, in einem blauen Kasten. Es folgen die Fragen nach dem Namen der Tabelle und der Anzahl der Mannschaften. Nach der Eingabe dieser Daten kehrt das Programm zum Hauptmenü zurück. Dieser Menüpunkt dient zum Erstellen des Grundgerüsts einer Tabelle. Er wird nur bei der ersten Eingabe gebraucht. Später kann die Tabelle leicht mit Menüpunkt 3 (Aktualisieren) geändert werden.

2. Tabelle ansehen

Die Tabelle wird entweder in der Reihenfolge der Eingabe, oder (wenn sie schon geordnet ist) mit dem Besten startend auf den Bildschirm gegeben. Mit der Enter-Taste kommt man wieder zum Menü zurück.

3. Tabelle aktualisieren

Dieser Punkt dient der Erweiterung der Tabelle um die neuesten Spielergebnisse. Ist die Eingabe korrekt, so muß eine beliebige Taste gedrückt werden. Es empfiehlt sich aber, wie am Bildschirm gefordert, die »]«-Taste zu benutzen. Bei falscher Eingabe muß »[« gedrückt werden.

4. Tabelle speichern und

5. Tabelle laden

Mit diesen beiden Menüpunkten kann die Tabelle als ASCII-Datei gespeichert beziehungsweise geladen werden. Das Pro-

gramm arbeitet mit dem Kassettenrecorder. Benutzen Sie die Diskettenstation, so müssen Sie einen Namen mit übergeben (bei OPENIN und so weiter).

6. Tabelle löschen

Man kann zwischen »ganze Tabelle löschen« und »bestimmte Mannschaften löschen« wählen. Auf diese Weise kann eine neue Tabellenrechnung vorgenommen werden.

7. Tabelle ordnen

Die Tabelle wird nach folgenden vier Kriterien (in gleicher Reihenfolge) geordnet:

1. Pluspunkte
2. Punktedifferenz
3. Tordifferenz
4. geschossene Tore

8. Tabelle verbessern

Es kann gewählt werden zwischen — neue Mannschaft einfügen — einen Fehler verbessern

Die Daten werden wie bei »Tabelle aufbauen« für die neue Mannschaft — beziehungsweise fehlerhaft eingegebene — angegeben. So können Sie Fehler korrigieren.

9. Tabelle ausdrucken

Analog Punkt 2 erfolgt die Ausgabe auf den Drucker.

10. Ergebnisse eingeben

Wird dieser Menüpunkt gewählt, so befindet man sich im ersten Punkt, bei dem es um die Verwaltung von Ergebnissen geht. Zuerst hat man die Wahl, welche der drei angegebenen Gruppen man bearbeiten will. Wer jetzt denkt, daß dieser Menüpunkt eigentlich dasselbe ist wie Punkt 3, hat sich geirrt. Die Spielpaarungen dieser drei Gruppen sind nämlich im Programm ab Zeile 10000 fest vergeben. So muß der Anwender nur noch Gruppe und Spieltag eingeben und die Begegnungen des gewählten Spieletages erscheinen. Die Ergebnisse können nun sehr schnell und sicher eingegeben werden. Die Resultate sind, im Gegensatz zu Menüpunkt 3, fest durch Variablen im Speicher abgelegt.

Natürlich kann jeder die zur Demonstration dienenden drei Gruppen (Handball, Fußball, B-Jugend) ändern zugunsten von Mannschaften, auf deren Ergebnisse und Tabelle er besonders oft und schnell zurückgreifen möchte. Wie das am schnellsten geht, zeigt das »Schema zur Änderung der Gruppen«.

Nach der Eingabe erscheint die Frage, ob die Ergebnisse auf die Tabelle übertragen werden sollen. Will man dies, so werden die neuen Ergebnisse entweder zu einer schon definierten Tabelle dazugezählt — oder wenn noch keine Tabelle im Speicher ist — wird aus den Ergebnissen eine neue Tabelle berechnet. Für die drei Gruppen müssen jetzt nur noch die Ergebnisse abgespeichert werden, da sich die Tabelle aus den Ergebnissen errechnen läßt. Sämtliche Menüpunkte stehen für die Tabellen dieser Gruppen zur Verfügung. Will man eine zweite Eingabe machen, so muß nur noch der neue Spieltag angegeben werden. Die Gruppe wurde ja schon vorher gewählt.

11. Ergebnisse ausgeben

Dieser Menüpunkt ist das Gegenstück zu »Ergebnisse eingeben«. Es wird zuerst gewählt, ob die Ergebnisse eines bestimmten Spieletages oder einer bestimmten Mannschaft ausgegeben werden sollen. Wurde nicht schon vorher eine der drei möglichen Gruppen festgelegt, so kann das jetzt nachgeholt werden.

Will man die Resultate einer bestimmten Mannschaft sehen, so muß die Mannschaft und der höchste auszugebende Spieltag aufgerufen werden. Der Computer gibt dann alle Begegnungen und Ergebnisse der gewünschten Mannschaft bis zu diesem Spieltag aus.

12. Ergebnisse speichern

Es können die Ergebnisse einer gewünschten Gruppe bis zu einem bestimmten Spieltag unter einem einzugebenden

Namen gespeichert werden.

13. Ergebnisse laden

Auf Kassette gespeicherte Ergebnisse können geladen werden, wobei man nach dem Ladevorgang erinnert wird, wieviele und von welcher Gruppe die Ergebnisse geladen wurden. Es besteht danach wie bei »Ergebnisse eingeben« die Möglichkeit, sämtliche Resultate auf die Tabelle zu übertragen.

14. Ergebnisse löschen

Ergebnisse werden gelöscht, die Tabelle bleibt jedoch unberührt. Wenn man die Ergebnisse einer anderen Gruppe behandeln will, so ist zuvor dieser Menüpunkt zu wählen.

15. Programm beenden

Beenden des Programms

(Thomas Sommer)

Schema zur Änderung der Gruppen:

1. In den Zeilen 5610 bis 5630 müssen die Namen der drei Gruppen geändert werden.

2. In den Zeilen 6080 bis 6110, 6130 bis 6160 und 6180 bis 6210 müssen für jede Gruppe folgende Daten geändert werden:

— die Variable az muß auf die Anzahl der in der Gruppe beteiligten Mannschaften gesetzt werden.

— »gruppe« muß den Namen der Gruppe enthalten.

— in der DATA-Zeile müssen die in der Gruppe beteiligten Mannschaften angegeben werden.

3. In den Zeilen 10000, 20000 und 15000 müssen die einzelnen Begegnungen als Code in Form von DATA-Zeilen angegeben werden. Der Code funktioniert folgendermaßen: Es werden für die einzelnen Mannschaften Zahlen gesetzt, wobei für die erste Mannschaft jeweils eine »1« gesetzt wird, für die zweite eine »2« und so weiter. Immer zwei Zahlen nebeneinander bilden eine Paarung. Gibt es eine ungerade Anzahl von Mannschaften — also ein spielfreies Team — so muß dieses an den Schluß der DATA-Zeile gesetzt werden. Ein Spieltag wird immer in eine DATA-Zeile gesetzt. Der nächste Spieltag in die nächste und so weiter. Wer mit diesem Programm die Ergebnisse von mehr als drei Gruppen verwalten will, kann die DATA-Zeilen separat abspeichern und sie mit »merge« ins Programm zurückholen.

***** Bundesliga *****				
Mannschaft	Spiele	Tordif.	Tore	Punkte
1. Werder Bremen	12	17	34 : 17	19 : 5
2. Bor. Mönchengladbach	12	13	27 : 14	17 : 7
3. Bayern München	12	10	23 : 13	16 : 8
4. Bayer Leverkusen	12	6	22 : 16	14 : 10
5. 1.FC Kaiserslautern	12	5	21 : 16	14 : 10
6. SVW Mannheim	12	3	21 : 18	14 : 10
7. Hamburger SV	12	7	20 : 13	13 : 11
8. VfB Stuttgart	12	6	25 : 19	13 : 11
9. VfL Bochum	12	5	27 : 22	12 : 12
10. 1.FC Köln	12	-2	17 : 19	12 : 12
11. Bayer Uerdingen	12	-11	18 : 29	11 : 13
12. FC Schalke 04	12	-4	16 : 20	10 : 14
13. Eintracht Frankfurt	12	-8	10 : 18	10 : 14
14. SV Hannover 96	12	-13	20 : 33	10 : 14
15. 1.FC Nürnberg	12	-5	18 : 23	8 : 16
16. 1.FC Saarbrücken	12	-7	14 : 21	8 : 16
17. Borussia Dortmund	12	-11	18 : 29	8 : 16
18. Fortuna Düsseldorf	12	-11	19 : 30	7 : 17

Schneller als die Sportschau sind Sie mit »Ligaverwaltung«

```

10 REM ***** [0936]
20 REM *** Sporttabellenberechnung *** [5D84]
30 REM *** CPC 464 *** [C0AE]
40 REM *** [B248]
50 REM *** (c) by Thomas Sommer *** [B3E6]
60 REM *** 7901 Dorndorf *** [7E6A]
70 REM *** Ringstr. 29 *** [53D2]
80 REM ***** [D644]
90 DIM name$(25),spiele(25),pu(25),mipu( [3E3C]
    25),tor(25),dif(25)
100 DIM kator(25),w(25,25),mas$(25),er$( [6EF6]
    25,25),er(25,25)
    
```

```

110 MODE 1 [8A50]
120 ON ERROR GOTO 6210 [D678]
130 INK 1,18:INK 2,26:INK 3,6:INK 0,0:BO [20A2]
    RDER 0:PAPER 0:PEN 3 [4A12]
140 WINDOW#1,1,40,1,2 [AEC2]
150 PAPER#1,3 [51DA]
160 CLS#1 [9328]
170 TAG [6B88]
180 MOVE 48,390
190 PRINT" T A B E L L E N B E R E C H N [811C]
    U N G"; [21CC]
200 PEN 1 [180E]
210 MOVE 1,358
220 PRINT"Author:THOMAS SOMMER{3 SPACE}>D [23D8]
    esign:ARNDT JANZ"; [83D8]
230 TAGOFF [D540]
240 LOCATE 1,5 [0970]
250 SPEED INK 10,10 [39DA]
260 PEN 2 [A53C]
270 PRINT"{14 SPACE}M E N U E : " [65E0]
280 PEN 3 [8F96]
290 PRINT;TAB(3);STRING$(36,204); [41CE]
300 PEN 1
310 PRINT"{5 SPACE}Tabelle erstellen.... [7FEE]
    ....{CTRL X}-1-{CTRL X}"
320 PRINT"{5 SPACE}Tabelle ansehen..... [A2D2]
    ....{CTRL X}-2-{CTRL X}"
330 PRINT"{5 SPACE}Tabelle aktualisieren [23CC]
    ....{CTRL X}-3-{CTRL X}"
340 PRINT"{5 SPACE}Tabelle abspeichern.. [B9AE]
    ....{CTRL X}-4-{CTRL X}"
350 PRINT"{5 SPACE}Tabelle laden..... [56DA]
    ....{CTRL X}-5-{CTRL X}"
360 PRINT"{5 SPACE}Tabelle loeschen..... [BD64]
    ....{CTRL X}-6-{CTRL X}"
370 PRINT"{5 SPACE}Tabelle ordnen..... [4F8A]
    ....{CTRL X}-7-{CTRL X}"
380 PRINT"{5 SPACE}Tabelle verbessern... [8B90]
    ....{CTRL X}-8-{CTRL X}"
390 PRINT"{5 SPACE}Tabelle ausdrucken... [6680]
    ....{CTRL X}-9-{CTRL X}"
400 PEN 3 [29D4]
410 PRINT;TAB(3);STRING$(36,205); [58BC]
420 PEN 1 [23D4]
430 PRINT"{5 SPACE}Ergebnisse eingeben.. [1C5A]
    ...-10-"
440 PRINT"{5 SPACE}Ergebnisse ausgeben.. [5F78]
    ...-11-"
450 PRINT"{5 SPACE}Ergebnisse abspeicher [6FDC]
    n...-12-"
460 PRINT"{5 SPACE}Ergebnisse laden.... [BA08]
    ...-13-"
470 PRINT"{5 SPACE}Ergebnisse loeschen.. [1492]
    ...-14-"
480 PEN 3 [31E4]
490 PRINT;TAB(3);STRING$(36,204); [5F9A]
500 PEN 1 [2DD2]
510 PRINT"{5 SPACE}Programm beenden.... [FF7A]
    ...{CTRL X}-15-{CTRL X}"
520 PEN 2 [2DD8]
530 PRINT [468A]
540 INPUT" Ihre Wahl bitte{2 SPACE}(1-15) [85EC]
    {2 SPACE}>enter< ";auswahl [FE82]
550 SPEED INK 40,40
560 ON auswahl GOSUB 580,920,1150,1510,1 [6D48]
    740,1960,2460,2700,3260,3440,4160,47 [CF4E]
    90,5110,5410,5580 [3E42]
570 GOTO 110 [1B4A]
580 REM [AB34]
590 REM *** Tabelle erstellen *** [1D2E]
600 REM
610 r=1
620 u$="T A B E L L E{3 SPACE}erstellen" [68CE]
    :GOSUB 5690 [C11A]
630 PRINT:PRINT [69E0]
640 PEN 3 [2A66]
650 INPUT"Wie heisst Ihre Tabelle ";na$ [2FE0]
660 PEN 1
670 PRINT:PRINT"Wieviele Mannschaften si [4B8C]
    nd an der Tabel-le";
680 INPUT " beteiligt (2-20) ";anz:IF an [749A]
    z<2 OR anz>20 THEN CLS:GOTO 590
690 PEN 3:PRINT:PRINT"Geben sie erst Nam [4928]
    e, dann Spiele, Plus-"
700 PRINT"punkte,Minuspunkte,geschossene [2F72]
    Tore, und";
710 PRINT"zuletzt kassierte Tore an. " [5B1A]
720 PRINT:PRINT [AA1A]
730 WINDOW #2,1,40,16,24 [5932]
740 WINDOW SWAP 0,2 [C538]
750 FOR n= r TO anz [0958]
760 PEN 2:n$=STR$(n):n$=MID$(n$,2):PRINT [69D2]
    n$;". Mannschaft: ";PRINT
770 PEN 1:INPUT"{CTRL X}Name{7 SPACE}: (C [9014]
    TRL X)";name$(n)
780 PEN 3:INPUT"{CTRL X}Spiele{5 SPACE}: [C7D6]
    {CTRL X}";spiele(n)
790 PEN 1:INPUT"{CTRL X}Punkte{5 SPACE}: [1CC4]
    {CTRL X}";pu(n)
    
```

Listing. »Ligaverwaltung«

Anwendungs-Listing

800 PEN 3:INPUT"⟨CTRL X⟩Minuspunkte:⟨CTR L X⟩";mipu(n)	[5D7C]	1550 PEN 3	[E342]
810 PEN 1:INPUT"⟨CTRL X⟩Tore(7 SPACE):⟨C TRL X⟩";tor(n)	[505C]	1560 LOCATE 1,6:PRINT"Ihre Tabelle wird abgespeichert, bitte(2 SPACE)warten sie."	[08A6] [5382] [E644] [A428] [F61E] [7DCC] [C130] [E146] [94C2] [088A] [5238] [206E] [E108] [9B7A] [4AAC] [217A] [E63E] [85C6] [32A0] [0714] [34A4]
820 PEN 3:INPUT"⟨CTRL X⟩Gegentore(2 SPAC E):⟨CTRL X⟩";kator(n)	[E8C6] [223A] [B2F6] [603C] [B294] [BBE8] [D1D6] [0176] [1DC4] [9C34] [7C3E] [FB6E] [4242] [F06A] [3BE6] [B1B6]	1570 PRINT:PRINT	
830 CLS	[223A]	1580 PEN 1	[E644]
840 NEXT	[B2F6]	1590 OPENOUT "Tab"	[A428]
850 WINDOW SWAP 2,0	[603C]	1600 PRINT #9,na\$	[F61E]
860 PEN 2:LOCATE 1,17	[B294]	1610 PRINT #9,anz	[7DCC]
870 PRINT STRING\$(40,"-");	[BBE8]	1620 FOR n= 1 TO anz	[C130]
880 PRINT"⟨8 SPACE⟩Das war's schon !!!"	[D1D6]	1630 PRINT #9,name\$(n)	[E146]
890 PRINT STRING\$(40,"-");	[0176]	1640 PRINT #9,spiele(n)	[94C2]
900 FOR w= 1 TO 1000:NEXT	[1DC4]	1650 PRINT #9,pu(n)	[088A]
910 RETURN	[9C34]	1660 PRINT #9,mipu(n)	[5238]
920 REM	[7C3E]	1670 PRINT #9,tor(n)	[206E]
930 REM *** Tabelle ansehen ***	[FB6E]	1680 PRINT #9,kator(n)	[E108]
940 REM	[4242]	1690 NEXT n	[9B7A]
950 MODE 2	[F06A]	1700 CLOSEOUT	[4AAC]
960 PEN 1	[3BE6]	1710 PRINT:PRINT	[217A]
970 LOCATE 15,1	[B1B6]	1720 PEN 2	[E63E]
980 PRINT"*****⟨2 SPACE⟩";na\$;" ⟨2 SPACE⟩*****"	[5EAE] [D49E]	1730 GOTO 870	[85C6] [32A0]
990 PRINT	[D49E]	1740 REM	[0714] [34A4]
1000 PRINT"Mannschaft";TAB(33);"Spiele"; TAB(42);"Tordif.";	[52F0]	1750 REM *** Tabelle laden ***	
1010 PRINT,TAB(54);"Tore";TAB(69);"Punkt e"	[6A1A] [6414] [1226] [E894] [9A26] [4ED2] [1070]	1760 REM	
1020 PRINT STRING\$(80,154)	[6A1A]	1770 u\$="T A B E L L E⟨3 SPACE⟩laden":60 SUB 5690	[5DA8] [E64C]
1030 FOR n= 1 TO anz	[6414]	1780 PEN 3	
1040 PRINT CHR\$(24)	[1226]	1790 LOCATE 1,6:PRINT"Ihre Tabelle wird geladen, bitte(8 SPACE)warten sie."	[C024] [2156] [39EA] [C82C] [89DA] [AF38] [1654] [26D0] [A898] [8646] [D97C] [D204] [AA70] [91F2] [9782] [F046] [87CE] [90A8] [29B6] [36AC]
1050 LOCATE 1,n+4:PRINT STRING\$(80," ")	[E894]	1800 PEN 1:PRINT:PRINT	
1060 LOCATE 1,n+4	[9A26]	1810 OPENIN "	[2156] [39EA]
1070 dif(n)=tor(n)-kator(n)	[4ED2]	1820 INPUT #9,na\$	[C82C]
1080 PRINT n;" ";name\$(n);TAB(32);CHR\$(149);spiele(n);TAB(40);CHR\$(149);d if(n);TAB(50);CHR\$(149);tor(n);TAB(57);";";kator(n);TAB(65);CHR\$(149); pu(n);TAB(72);";";mipu(n)	[1070] [4F6A] [FF52] [5A30]	1830 INPUT #9,anz	[89DA]
1090 NEXT	[4F6A]	1840 FOR n= 1 TO anz	[AF38]
1100 PEN 1:PAPER 0	[FF52]	1850 INPUT #9,name\$(n)	[1654]
1110 LOCATE 20,25:PRINT"Zum Menue-Progra mm geht's mit >enter<"	[5A30]	1860 INPUT #9,spiele(n)	[26D0]
1120 a\$=INKEY\$	[A80E] [C494] [8EB6] [B406] [2D96] [EFE4] [3B9A]	1870 INPUT #9,pu(n)	[A898]
1130 IF a\$=CHR\$(13) THEN RETURN	[A80E]	1880 INPUT #9,mipu(n)	[8646]
1140 GOTO 1120	[C494]	1890 INPUT #9,tor(n)	[D97C]
1150 REM	[8EB6]	1900 INPUT #9,kator(n)	[D204]
1160 REM *** Tabelle aktualisieren ***	[B406]	1910 NEXT n	[AA70]
1170 REM	[2D96]	1920 CLOSEIN	[91F2]
1180 u\$="T A B E L L E⟨3 SPACE⟩aktualisi eren":GOSUB 5690	[EFE4] [3B9A]	1930 PRINT:PRINT	[9782]
1190 PRINT:PRINT:PEN 3	[3B9A]	1940 PEN 2	[F046]
1200 PRINT"Ich nenne ihnen nun die einze lnen Mann-"	[7078] [C25E]	1950 GOTO 870	[87CE]
1210 PRINT"schaffen, und sie geben dazu ein,wieviel";	[7078]	1960 REM	[90A8]
1220 PRINT"Tore sie geschossen und kassi ert hat."	[C25E]	1970 REM *** Tabelle loeschen ***	[29B6] [36AC]
1230 PRINT"Wenn sie bei der angezeigten Mannschaft"	[C25E]	1980 REM	
1240 PRINT"nichts aktualisieren wollen, dann geben"	[7078]	1990 u\$="T A B E L L E⟨3 SPACE⟩loeschen" :GOSUB 5690	[2E4A] [D04C]
1250 PRINT"sie bei 'geschossen' die Zahl 999 ein.";	[C25E]	2000 PRINT:PRINT:PEN 3	
1260 PRINT:PRINT:PEN 1	[C660]	2010 PRINT"⟨CTRL X⟩Wollen sie⟨CTRL X⟩":P RINT	[EEFA] [E330]
1270 WINDOW #3,1,40,14,14:WINDOW #4,1,40 ,18,25	[5A44] [F2F6]	2020 PEN 1	
1280 FOR n= 1 TO anz	[5A44]	2030 PRINT"die ganze Tabelle.....⟨C TRL X⟩-1-⟨CTRL X⟩":PRINT	[E888] [E910]
1290 PEN 2:LOCATE 1,14:PRINT "⟨CTRL X⟩"; name\$(n);"⟨CTRL X⟩"	[F2F6]	2040 PRINT"o d e r":PRINT	
1300 PRINT	[185E]	2050 PRINT"eine bestimmte Mannschaft..⟨C TRL X⟩-2-⟨CTRL X⟩":PRINT	[EB8A] [F33C]
1310 PRINT STRING\$(40,"-")	[185E]	2060 PEN 3	
1320 PRINT	[DB22]	2070 PRINT"⟨CTRL X⟩loeschen ?⟨CTRL X⟩":P RINT:PRINT	[3750] [EF3E]
1330 PEN 1:LOCATE 1,18:INPUT"⟨CTRL X⟩ges chossen :⟨CTRL X⟩";gesch	[DB22]	2080 PEN 2	
1340 IF gesch=999 THEN GOTO 1460	[90F6] [A356]	2090 INPUT"⟨CTRL X⟩Ihre Wahl (1-2) ⟨CTRL X⟩";wahl	[A956] [236E] [53E8] [404E] [BB1A] [BF92] [991E] [CD46] [F8AA] [E542]
1350 PEN 3:LOCATE 1,19:INPUT"⟨CTRL X⟩kas siert⟨3 SPACE⟩:⟨CTRL X⟩";kass	[90F6] [A356]	2100 PRINT:PRINT	
1360 PEN 2:LOCATE 1,21:PRINT"Eingabe kor rekt: 'J'⟨2 SPACE⟩falsch: '['"	[20CA] [A334]	2110 WINDOW#5,1,40,4,25	[53E8]
1370 x\$=INKEY\$:IF x\$=" " THEN 1370	[801C] [B1E2] [A9BE] [A7E6]	2120 ON wahl GOTO 2380,2140	[404E]
1380 IF x\$="J" THEN 1400	[801C]	2130	[BB1A]
1390 IF x\$="[" THEN CLS#4:GOTO 1330	[B1E2]	2140 REM bestimmte Mannschaft loeschen	[BF92]
1400 IF gesch>kass THEN pu(n)=pu(n)+2	[A9BE]	2150	[991E]
1410 IF gesch=kass THEN pu(n)=pu(n)+1:m ipu(n)=mipu(n)+1	[A7E6]	2160 CLS#5	[CD46]
1420 IF gesch<kass THEN mipu(n)=mipu(n)+ 2	[3604] [4238]	2170 LOCATE 1,6	[F8AA] [E542]
1430 tor(n)=tor(n)+gesch	[3604]	2180 PEN 3	
1440 kator(n)=kator(n)+kass	[4238]	2190 INPUT"Welche Mannschaft soll aus de r Tabelle(2 SPACE)geloescht werden ";lo\$	[F45E] [F04C]
1450 spiele(n)=spiele(n)+1	[6546]	2200 PRINT:PRINT:PEN 1	
1460 CLS#3:CLS#4	[A004] [BE50] [9792] [C82C] [42BE]	2210 INPUT"Wollen Sie diese Mannschaft w irklich(4 SPACE)loeschen (j/n)";q\$	[029C]
1470 NEXT	[A004]	2220 IF q\$="j" OR q\$="J" THEN 2230 ELSE GOTO 2310	[FD28] [8BE8] [C02E] [5CAA]
1480 PEN 2	[BE50]	2230 n2=0	
1490 LOCATE 1,14	[9792]	2240 FOR n= 1 TO anz	
1500 GOTO 870	[C82C]	2250 IF name\$(n)=lo\$ THEN n2=1	
1510 REM	[42BE]	2260 IF n2=1 THEN name\$(n)=name\$(n+1):sp iele(n)=spiele(n+1):pu(n)=pu(n+1):m ipu(n)=mipu(n+1):tor(n)=tor(n+1):ka tor(n)=kator(n+1)	[B0F8] [DF54]
1520 REM *** Tabelle abspeichern ***	[F84A] [D416] [5104] [B886] [B60A] [6B2C] [F756] [E044] [CB10] [6FBC] [3196] [3D0A] [3F9A]	2270 NEXT	
1530 REM	[F84A]	2280 IF n2=0 THEN PRINT:PRINT"nicht vorh anden !":FOR n=1 TO 1500:NEXT:GOTO 2310	[DB06] [A134] [C042] [0438]
1540 u\$="T A B E L L E⟨3 SPACE⟩abspeiche rn":GOSUB 5690		2290 anz=anz-1	
		2300 PEN 2:PRINT:PRINT:GOSUB 870	
		2310 PEN 3	

Listing. »Ligaverwaltung« (Fortsetzung)

2320	LOCATE 1,24	[0804]	anschafft"	[60AE]
2330	INPUT"Wollen sie noch eine Mannschaft loeschen(j/n)";fr\$	[C3B4]	2980 INPUT"einfuegen (j/n)";fr\$	[2058]
2340	IF fr\$="J" OR fr\$="j" THEN GOTO 2140	[4A6C]	2990 IF fr\$="j" OR fr\$="J" THEN GOTO 2870	[9396]
2350	IF fr\$="N" OR fr\$="n" THEN RETURN	[07FE]	3000 IF fr\$="n" OR fr\$="N" THEN RETURN	[BDF0]
2360	GOTO 2320	[6016]	3010 GOTO 2970	[181E]
2370		[2726]	3020	[B918]
2380	REM ganze Tabelle loeschen	[4EA0]	3030 REM Fehler verbessern	[B33E]
2390		[992A]	3040	[931C]
2400	CLS#5	[A340]	3050 CLS#5	[A544]
2410	PEN 1	[E636]	3060 PEN 1:aush=anz	[C91C]
2420	LOCATE 1,6	[E3A6]	3070 LOCATE 1,5	[E3A8]
2430	INPUT"Wollen Sie die Tabelle wirklich loeschen(j/n)";fr\$	[6D2E]	3080 INPUT"Bei welcher Mannschaft wollen Sie eine(2 SPACE)Verbesserung vornehmen";ma\$	[7D52]
2440	IF fr\$="j" OR fr\$="J" THEN ERASE name\$,spiele,tor,kator,pu,mipu:anz=0: DIM name\$(25),spiele(25),pu(25),mipu(25),tor(25),kator(25):na\$="" :PEN 2:PRINT:GOSUB 870 ELSE RETURN	[8C34]	3090 FOR n= 1 TO anz	[FA36]
2450	GOTO 110	[6AAC]	3100 IF ma\$=name\$(n) THEN r=n:anz=n:op=1	[3DD8]
2460	REM	[33A0]	3110 NEXT	[E648]
2470	REM *** Tabelle ordnen ***	[0318]	3120 IF op<>123 THEN GOTO 3180	[2A30]
2480	REM	[49A4]	3130 PEN 3:LOCATE 1,8	[C388]
2490	u\$="T A B E L L E(3 SPACE)ordnen":GOSUB 5690	[67AC]	3140 PRINT"Geben sie erst Name, dann Spiele, Plus-"	[560C]
2500	PRINT:PRINT	[5376]	3150 PRINT"punkte,Minuspunkte,geschossene Tore, und";	[1BD6]
2510	PEN 3:INPUT"Goll Ihre Tabelle wirklich geordnet werden(j/n)";w\$	[71B8]	3160 PRINT"zuletzt kassierte Tore an. "	[317E]
2520	IF w\$="j" OR w\$="J" THEN PEN 1:GOTO 2530 ELSE RETURN	[F35C]	3170 GOSUB 730:GOTO 3190	[180A]
2530	PRINT:PRINT:PRINT"Bitte warten sie":PRINT:PRINT	[F050]	3180 PRINT:PRINT:PRINT"nicht vorhanden !"	[1D0E]
2540	FOR i= 2 TO anz+1:fl=0	[19D6]	3190 r=1:anz=aush:op=0	[E288]
2550	FOR j= anz TO i STEP -1	[2E52]	3200 PEN 3:LOCATE 1,24	[99E0]
2560	IF pu(j-1)<pu(j) THEN GOSUB 2620	[069E]	3210 PRINT"Wollen sie nochmal eine Verbesserung"	[7646]
2570	IF pu(j-1)=pu(j) THEN IF pu(j-1)-mipu(j-1)<pu(j)-mipu(j) THEN GOSUB 2620	[8DDA]	3220 INPUT"vornehmen (j/n)";fr\$	[BCB8]
2580	IF pu(j)=pu(j-1) AND pu(j-1)-mipu(j-1)=pu(j)-mipu(j) THEN IF tor(j-1)-kator(j-1)<tor(j)-kator(j) THEN GOSUB 2620	[3F58]	3230 IF fr\$="j" OR fr\$="J" THEN GOTO 3030	[4168]
2590	IF pu(j)=pu(j-1) AND pu(j-1)-mipu(j-1)=pu(j)-mipu(j) AND tor(j)-kator(j)=tor(j-1)-kator(j-1) THEN IF tor(j-1)<tor(j) THEN GOSUB 2620	[DEF6]	3240 IF fr\$="n" OR fr\$="N" THEN RETURN	[0DFC]
2600	NEXT j: IF fl=0 THEN PEN 2:GOTO 870	[C09E]	3250 GOTO 3200	[9510]
2610	NEXT i:PEN 2:GOTO 870	[39A4]	3260 REM	[309E]
2620	h=pu(j):pu(j)=pu(j-1):pu(j-1)=h	[1E42]	3270 REM *** Tabelle ausdrucken ***	[2274]
2630	h=mipu(j):mipu(j)=mipu(j-1):mipu(j-1)=h	[33F4]	3280 REM	[02A2]
2640	h=kator(j):kator(j)=kator(j-1):kator(j-1)=h	[7226]	3290 u\$="T A B E L L E(3 SPACE)ausdrucken":GOSUB 5690	[8508]
2650	h=tor(j):tor(j)=tor(j-1):tor(j-1)=h	[42C8]	3300 PEN 1	[7034]
2660	h=spiele(j):spiele(j)=spiele(j-1):spiele(j-1)=h	[AB32]	3310 PRINT:PRINT:INPUT"Ist der Drucker bereit(j/n)";fr\$	[AA5C]
2670	h=name\$(j):name\$(j)=name\$(j-1):name\$(j-1)=h\$	[08DC]	3320 IF fr\$="n" OR fr\$="N" THEN RETURN	[15FA]
2680	fl=1	[9060]	3330 PEN 3	[EB3E]
2690	RETURN	[8DA2]	3340 LOCATE 1,9:PRINT"Tabelle wird ausgedruckt - etwas Geduld"	[3174]
2700	REM	[339A]	3350 PRINT#8,TAB(18):"*****{2 SPACE}";na\$;"{2 SPACE}*****"	[9AA8]
2710	REM *** Tabelle verbessern ***	[EF84]	3360 PRINT#8	[2AAB]
2720	REM	[419E]	3370 PRINT#8,TAB(2);"Mannschaft";TAB(3);"Spiele";TAB(42);"Tordif.";TAB(54);"Tore";TAB(69);"Punkte"	[D824]
2730	u\$="T A B E L L E(3 SPACE)verbessern":GOSUB 5690	[EE18]	3380 PRINT#8	[94AC]
2740	PRINT:PRINT	[5B82]	3390 FOR n= 1 TO anz	[963C]
2750	PEN 1	[7444]	3400 dif(n)=tor(n)-kator(n)	[366E]
2760	PRINT"Eine neue Mannschaft einfuegen {CTRL X}-1-{CTRL X}";	[0F3A]	3410 PRINT#8,n;" ";name\$(n);TAB(35);spiele(n);TAB(44);dif(n);TAB(52);tor(n);TAB(58);";";kator(n);TAB(67);pu(n);TAB(72);";";mipu(n)	[95B6]
2770	PRINT STRING\$(30," ");PRINT:PRINT	[8A52]	3420 NEXT n	[EB6C]
2780	PRINT"o d e r ";PRINT:PRINT	[16B4]	3430 GOTO 860	[87C2]
2790	PRINT"Einen Fehler verbessern{8 SPACE}{CTRL X}-2-{CTRL X}";	[52FC]	3440 REM	[4E9E]
2800	PRINT STRING\$(23,"_");PRINT:PRINT:PRINT	[D2D8]	3450 REM *** Ergebnisse eingeben ***	[9720]
2810	PEN 3:INPUT"{CTRL X}Ihre Wahl bitte >enter<{CTRL X}";wahl	[2BD4]	3460 REM	[1CA2]
2820	PRINT	[12F2]	3470 u\$="E R G E B N I S S E(3 SPACE)eingeben":GOSUB 5690	[7CB4]
2830	WINDOW #5,1,40,4,25	[393A]	3480 z\$="" :spt=0	[B118]
2840	ON wahl GOSUB 2870,3030	[8DB4]	3490 PRINT:PRINT	[5188]
2850	RETURN	[839E]	3500 PEN 3	[E03C]
2860		[C82E]	3510 IF wa<1 OR wa>3 THEN 3520 ELSE PRINT"{CTRL X}";gruppe\$;"{CTRL X}":GOTO 3600	[8162]
2870	REM neue Mannschaft einfuegen	[4836]	3520 PRINT"Sie koennen zwischen folgende n Gruppen(2 SPACE)waehlen:"	[134C]
2880		[C232]	3530 PRINT:PRINT	[597E]
2890	r=anz+1:anz=anz+1	[4058]	3540 PEN 1	[FB40]
2900	IF anz >20 THEN anz=20:LOCATE 1,21: PEN 2:PRINT"Tabelle ist ueberfuellt !";FOR n= 1 TO 2000:NEXT :RETURN	[1B08]	3550 PRINT"Fussball-Bundesliga.....{CTRL X}-1-{CTRL X}":PRINT	[E276]
2910	CLS#5	[224C]	3560 PRINT"Handball-Bundesliga.....{CTRL X}-2-{CTRL X}":PRINT	[FD2E]
2920	LOCATE 1,4:PRINT:PRINT"Geben sie erst Name, dann Spiele, Plus-"	[E1EA]	3570 PRINT"B-Jugend (Kreisstaffel)..{CTRL X}-3-{CTRL X}":PRINT	[FCF2]
2930	PRINT"punkte,Minuspunkte,geschossene Tore, und";	[35E0]	3580 PEN 3:PRINT:PRINT:INPUT"{CTRL X}Ihre Wahl bitte (1-3){CTRL X}";wa	[5EE6]
2940	PRINT"zuletzt kassierte Tore an. "	[3588]	3590 IF z\$="*" THEN RETURN	[7F70]
2950	GOSUB 730	[C854]	3600 PEN 1:PRINT:INPUT"{CTRL X}Welcher Spieltag bitte{3 SPACE}{CTRL X}";st	[F6FE]
2960	PEN 3:LOCATE 1,24	[8BF8]	3610 PEN 3	[DF40]
2970	PRINT"Wollen sie noch einmal eine M		3620 IF z\$="*" OR z\$="**" THEN 3670	[3694]
			3630 PRINT:PRINT:PRINT"Sollte eines der angegebenen Spiele"	[97BC]
			3640 PRINT"nicht stattgefunden haben,so geben sie"	[F23C]
			3650 PRINT"zwei mal die Zahl '999' ein."	[08F8]

Listing. »Ligaverwaltung« (Fortsetzung)

Anwendungs-Listing

3660 FOR n= 1 TO 3000:NEXT	[6C22]	4340 REM bestimmter Spieltag	[8A08]
3670 ON wa GOSUB 4020,4070,4120	[A9EA]	4350 LOCATE 1,6	[EA0E]
3680 IF az/2<>INT(az/2) THEN az2=az-1 EL		4360 z\$="**"	[43C0]
SE az2=az	[749C]	4370 IF wa<1 OR wa>3 THEN GOSUB 3500:GOT	[718C]
3690 FOR k= 1 TO az	[5E60]	O 4390	
3700 READ mas\$(k)	[058E]	4380 PEN 3:PRINT"{CTRL X}";gruppe\$;" : {CT	[C25C]
3710 NEXT k	[DE6A]	RL X}":GOSUB 3600	[C320]
3720 IF wa=1 THEN RESTORE 6280	[3CC8]	4390 FOR m= 1 TO az2 STEP 2	[C328]
3730 IF wa=2 THEN RESTORE 6340	[E3C6]	4400 READ w(m,n):READ w(m+1,n)	[CC28]
3740 IF wa=3 THEN RESTORE 6400	[D6C4]	4410 PRINT mas\$(w(m,n));" - ";mas\$(w(m+1	
3750 IF z\$="*" THEN RETURN	[476C]	,n)	[4F46]
3760 FOR n=1 TO st-1	[C2F6]	4420 IF er(m,n)=999 THEN PRINT CHR\$(11);	
3770 FOR m=1 TO az	[C322]	STRING\$(51,9);"ausgefallen !":GOTO	[01F2]
3780 READ w(m,n)	[10FA]	4450	
3790 NEXT m,n	[D4B2]	4430 PRINT CHR\$(11);STRING\$(50,9);er(m,n	[180E]
3800 n=st	[6CFA])	
3810 MODE 2:PEN 1	[1AA2]	4440 PRINT CHR\$(11);STRING\$(53,9);" : ";	[0BC0]
3820 PRINT gruppe\$	[1EA2]	er(m+1,n)	[F158]
3830 PRINT"{CTRL X}";st;". Spieltag:{CTRL		4450 NEXT	
L X}":PRINT:PRINT	[59BE]	4460 IF az2<>az THEN READ spf:PRINT:PRIN	[0AF8]
3840 IF z\$="*" THEN RETURN	[DB00]	T"Spiefrei :";mas\$(spf)	[FD16]
3850 FOR m=1 TO az2 STEP 2	[62E0]	4470 LOCATE 1,25	
3860 READ w(m,n):READ w(m+1,n)	[C73A]	4480 INPUT"Nach eine Ausgabe (j/n) ";fr\$	[E468]
3870 PRINT mas\$(w(m,n));" - ";mas\$(w(m+1			
,n)	[A658]	4490 IF fr\$="J" OR fr\$="j" THEN MODE 1:G	[72E6]
3880 PRINT CHR\$(11);STRING\$(50,9);:LINE		OTO 4170	[20FC]
INPUT er\$(m,n)	[5FCA]	4500 IF fr\$="N" OR fr\$="n" THEN RETURN	[3A24]
3890 PRINT CHR\$(11);STRING\$(53,9);" : ";		4510 GOTO 4470	
:LINE INPUT er\$(m+1,n)	[927C]	4520 REM bestimmte Mannschaft	[3EB8]
3900 er(m,n)=VAL(er\$(m,n)):er(m+1,n)=VAL		4530 z\$="**"	[306A]
(er\$(m+1,n))	[0888]	4540 LOCATE 1,5	[BAAE]
3910 NEXT m	[EB72]	4550 IF wa<1 OR wa>3 THEN GOSUB 3500:GOS	
3920 IF az2<>az THEN READ w(m+1,n):PRINT		UB 3670:CLS#5:LOCATE 1,5 ELSE PEN 3	
:PRINT"Spiefrei :";mas\$(w(m+1,n))	[5680]	:PRINT:PRINT"{CTRL X}";gruppe\$;" : {	[7604]
3930 WINDOW#7,1,80,24,25:LOCATE 1,24	[7F18]	CTRL X}":GOSUB 3670:PRINT	[ED46]
3940 GOSUB 5810	[C98C]	4560 PEN 1	
3950 spt=0	[8A6A]	4570 INPUT"Von welcher Mannschaft wollen	
3960 LOCATE 1,25	[451C]	sie die{3 SPACE}Ergebnisse sehen "	
3970 INPUT"Nach eine Eingabe (j/n) ";wah		;man\$	[8FBA]
l\$	[0DFC]	4580 PEN 3:PRINT	[7DDC]
3980 IF wahl\$="j" OR wahl\$="J" THEN z\$="		4590 INPUT"Bis zu welchem Spieltag ";spt	[C706]
***":LOCATE 1,24:GOTO 3600	[D72E]		[BA9E]
3990 IF wahl\$="n" OR wahl\$="N" THEN GOTO		4600 MODE 2:PEN 1	[FC9E]
110	[B37A]	4610 PRINT gruppe\$	
4000 GOTO 3950	[801C]	4620 PRINT"{CTRL X}";man\$;"{CTRL X}":PRI	[3B2A]
4010	[BF18]	NT:PRINT	[D88A]
4020 REM Fussball-Bundesliga	[25A2]	4630 k=0	[0556]
4030 RESTORE 4050:az=18	[305E]	4640 FOR n= 1 TO spt	[E41E]
4040 gruppe\$="Fussball-Bundesliga":RETUR		4650 FOR m= 1 TO az2 STEP 2	[CE38]
N	[A342]	4660 READ w(m,n):READ w(m+1,n)	
4050 DATA Bremen,Moenchengladb,Nuernberg		4670 IF man\$=mas\$(w(m,n)) OR man\$=mas\$(w	
,Mannheim,Frankfurt,Muenchen,Leverk		(m+1,n)) THEN GOSUB 4730	[3AFC]
usen,Koeln,Bochum,Kaiserslautern,Ue		4680 NEXT m	[AF7C]
rdingen,Hamburg,Stuttgart,Duesseldo		4690 IF az<>az2 THEN READ spf:IF mas\$(sp	
rf,Saarbruecken,Dortmund,Schalke,Ha		f)=man\$ THEN k=n:spt=spt-1	[DE0A]
nnover	[4CD4]	4700 NEXT n	[9870]
4060	[9A22]	4710 IF k<>0 THEN PRINT:PRINT man\$;" hat	
4070 REM Handball-Bundesliga	[4160]	te am";k;". Spieltag Spiefrei !"	[428C]
4080 RESTORE 4100:az=14	[2958]	4720 GOTO 4470	[C12A]
4090 gruppe\$="Handball-Bundesliga":RETUR		4730 IF mas\$(w(m,n))=man\$ THEN PRINT"{CT	
N	[8A00]	RL X}";mas\$(w(m,n));"{CTRL X}";" -	[29EC]
4100 DATA TV Grosswallstadt,FA Goepinge		" ;mas\$(w(m+1,n))	
n,VfL Guenzburg,TuSEM Essen,TBV Lem		4740 IF mas\$(w(m+1,n))=man\$ THEN PRINT m	
go,TuRu Duesseldorf,OSC Dortmund,MT		as\$(w(m,n));" - {CTRL X}";mas\$(w(m+	[DDE8]
SV Schwabing,Weiche-Handewitt,THW K		1,n));"{CTRL X}";	
iel,VfL Gummersbach,R. Fuechse Berl		4750 IF er(m,n)=999 THEN PRINT CHR\$(11);	
in,GW Dankersen,TuS Hofweier	[F5D4]	STRING\$(51,9);"ausgefallen !":GOTO	[2E0A]
4110	[C31A]	4780	
4120 REM B-Jugend (Kreisstaffel)	[9C16]	4760 PRINT CHR\$(11);STRING\$(50,9);er(m,n	[FF1A]
4130 RESTORE 4150:az=11	[4954])	
4140 gruppe\$="B-Jugend (Kreisstaffel)":R		4770 PRINT CHR\$(11);STRING\$(53,9);" : ";	[9ECC]
ETURN	[26B6]	er(m+1,n)	[8EA6]
4150 DATA SG Tiefenbach/Au,TSV Kellmuenz		4780 RETURN	[4AB0]
,SF Illerrieden,SV Ingstetten,SV Be		4790 REM	[E1AE]
uren,SV Grafertshofen,SC Voehringen		4800 REM *** Ergebnisse abspeichern ***	[26A2]
II,SSV Illerberg,FV Bellenberg,SV		4810 REM	
Jedesheim,TSV Buch	[2EC8]	4820 u\$="E R G E B N I S S E{3 SPACE}abs	[4A42]
4160 REM	[1F9E]	peichern":GOSUB 5690	[EC86]
4170 REM *** Ergebnisse ausgeben ***	[3B3A]	4830 PRINT:PRINT	
4180 REM	[25A2]	4840 IF wa<1 OR wa>3 THEN 4850 ELSE PEN	
4190 MODE 1:PEN 2	[B9A6]	3:PRINT"{CTRL X}";gruppe\$;" : {CTRL X	
4200 u\$="E R G E B N I S S E{3 SPACE}aus		3}":PRINT:GOTO 4930	[E0FA]
geben":GOSUB 5690	[B9BE]	4850 PEN 3	[044E]
4210 PEN 3	[D43A]	4860 PRINT"Von welcher Gruppe wollen sie	
4220 PRINT:PRINT	[4778]	die Er--	[6EF6]
4230 PRINT"{CTRL X}Sie haben die Wahl, E		4870 PRINT"gebnisse abspeichern ?"	[702E]
rgebnisse von:{CTRL X}"	[CA4E]	4880 PRINT:PRINT	[5190]
4240 PEN 1	[773C]	4890 z\$="**"	[777C]
4250 PRINT:PRINT	[5E7E]	4900 GOSUB 3540	[32B2]
4260 PRINT"einem bestimmten Spieltag....		4910 WINDOW#5,1,40,4,25:CLS#5	[9AE4]
{CTRL X}-1-{CTRL X}"	[61DE]	4920 LOCATE 1,6	[44B4]
4270 PRINT:PRINT"o d e r"	[A41E]	4930 ON wa GOSUB 4020,4070,4120	[EA0E]
4280 PRINT:PRINT"einer bestimmten Mannsc		4940 PEN 1	[D34A]
haft..{CTRL X}-2-{CTRL X}"	[6758]	4950 INPUT"Bis zu welchem Spieltag wolle	
4290 PEN 3	[5C4A]	n sie die{2 SPACE}Ergebnisse abspei	
4300 PRINT:PRINT:PRINT"{CTRL X}anzusehen		chern ";spt	[F2A4]
{CTRL X}"	[738E]	4960 PRINT:PRINT 3	[8BE0]
4310 PEN 2:PRINT:PRINT:PRINT:INPUT"Ihre		4970 INPUT"Unter welchem Namen ";ab\$	[0C58]
Wahl bitte (1-2)";wahl	[13A8]		
4320 WINDOW#5,1,40,4,25:CLS#5	[9CDA]		
4330 ON wahl GOTO 4340,4520	[0C1C]		

Listing. »Ligaverwaltung« (Fortsetzung)

```

4980 PRINT [F904]
4990 PEN 1 [FA54]
5000 OPENOUT ab$ [182C]
5010 PRINT#9,wa [56A6]
5020 PRINT#9,spt [51A6]
5030 PRINT#9,az [3AB0]
5040 PRINT#9,gruppe$ [AD6A]
5050 FOR n=1 TO spt [270E]
5060 FOR m=1 TO az [A416]
5070 PRINT#9,er(m,n) [3B60]
5080 NEXT: NEXT [8B4A]
5090 CLOSEOUT [FDB8]
5100 GOTO 860 [76BA]
5110 REM [3596]
5120 REM *** Ergebnisse laden *** [69A6]
5130 REM [439A]
5140 u$="E R G E B N I S S E {3 SPACE}lad [6E3A]
en":GOSUB 5690 [5D7E]
5150 PRINT:PRINT [F444]
5160 PEN 3 [715C]
5170 PRINT"Ihre Ergebnisse werden gelade [6EEE]
n," [E186]
5180 PRINT"bitte warten Sie." [F636]
5190 PRINT:PRINT [2FE6]
5200 PEN 1 [42B2]
5210 OPENIN " [69B2]
5220 INPUT#9,wa [8AB6]
5230 INPUT#9,spt [ED76]
5240 INPUT#9,az [5514]
5250 INPUT#9,gruppe$ [191C]
5260 FOR n=1 TO spt [E56C]
5270 FOR m=1 TO az [D9AC]
5280 INPUT#9,er(m,n) [7FEA]
5290 NEXT m,n [9F3E]
5300 CLOSEIN [6EA0]
5310 PEN 3 [0448]
5320 PRINT:PRINT:PRINT"Es wurden die Spi [2904]
eltage 1 -";spt;"der " [6C24]
5330 PRINT:PRINT"(CTRL X)";gruppe$;"(CTR [663A]
L X)" [CB56]
5340 PRINT:PRINT"geladen." [EC4A]
5350 FOR n= 1 TO 1500:NEXT [4014]
5360 WINDOW#7,1,40,24,25:LOCATE 1,24:PEN [8CC2]
1:GOSUB 5810 [319C]
5370 CLS#7 [4646]
5380 PEN 2 [33A0]
5390 LOCATE 1,22 [56DA]
5400 GOTO 870 [AEBE]
5410 REM [48FC]
5420 REM *** Ergebnisse loeschen *** [91CE]
5430 REM [E3DE]
5440 u$="E R G E B N I S S E {3 SPACE}loe [181A]
schen":GOSUB 5690 [881C]
5450 LOCATE 1,6:PEN 1 [8930]
5460 PRINT"Wollen sie ihre Ergebnisse wi [EE18]
rklich" [8712]
5470 INPUT"loeschen (j/n) ";lo$ [D2AB]
5480 IF lo$="J" OR lo$="j" THEN 5510 [69D4]
5490 IF lo$="N" OR lo$="n" THEN RETURN [32DB]
5500 GOTO 5420 [73D2]
5510 FOR n=1 TO st [4AAC]
5520 FOR m=1 TO az [1FD2]
5530 er(m,n)=0 [349E]
5540 NEXT m,n [7A54]
5550 st=0:az=0:wa=0:az2=0:gruppe$="" [9842]
5560 PRINT:PEN 2 [4584]
5570 GOTO 870 [0080]
5580 REM [4D82]
5590 REM *** Programm beenden *** [C75C]
5600 REM [CC1E]
5610 u$="P R O G R A M M {3 SPACE}beenden [4CAE]
":GOSUB 5690 [99AB]
5620 PEN 1 [32A0]
5630 PRINT:PRINT [A3A4]
5640 PRINT"Wollen Sie das Programm wirkl [DF90]
ich be-" [8826]
5650 INPUT"enden (j/n)";ende$ [BBAA]
5660 IF ende$="j" OR ende$="J" THEN MODE [2688]
2:END [F270]
5670 IF ende$="n" OR ende$="N" THEN RETU [1CB6]
RN [03B2]
5680 REM [8EAA]
5690 REM *** Routine fuer Ueberschrift [24A2]
5700 REM [7372]
5710 MODE 1:PEN 2 [56A6]
5720 WINDOW#6,1,40,1,3 [435C]
5730 PAPER#6,2:CLS#6
5740 PRINT CHR$(24)
5750 LOCATE 5,2
5760 PRINT u$
5770 LOCATE 1,3
5780 PRINT CHR$(24)
5790 RETURN
5800 REM
5810 REM *** Routine zur Uebertragung de [7372]
r Ergebnisse auf Tabelle [56A6]
5820 REM
5830 PRINT"Sollen die Ergebnisse auf die [435C]
Tabelle"
5840 INPUT"uebertragen werden (j/n) ";ue [7310]
ber$ [7310]
5850 IF ueber$="n" OR ueber$="N" THEN CL [D1E6]
S#7:RETURN
5860 IF ueber$="j" OR ueber$="J" THEN 58 [FDD6]
80 [20D0]
5870 CLS#7:LOCATE 1,24:GOTO 5830
5880 CLS#7:LOCATE 1,25:PRINT"(CTRL X)Wir [F924]
d gemacht ! {CTRL X}" [31D2]
5890 z$="*":GOSUB 3670
5900 IF spt=0 THEN spt=st:anf=st ELSE an [22D4]
f=1
5910 anf2=anf-1:IF anf2<1 THEN anf2=1:GO [D11C]
TO 5960 [2234]
5920 FOR k=1 TO anf2
5930 FOR l=1 TO az [DD20]
5940 READ h [972C]
5950 NEXT l,k [F0AA]
5960 FOR n= anf TO spt [806A]
5970 FOR m= 1 TO az2 STEP 2 [F72A]
5980 READ w(m,n):READ w(m+1,n) [F344]
5990 kan1=0:kan2=0 [47C4]
6000 FOR k=1 TO anz [DDE4]
6010 IF mas$(w(m,n))=name$(k) THEN kan1= [28B4]
1:zu=0:GOSUB 6120
6020 IF mas$(w(m+1,n))=name$(k) THEN kan [4372]
2=1:zu=1:GOSUB 6120 [E766]
6030 NEXT k
6040 IF kan1<>1 THEN anz=anz+1:k=anz:nam [FD54]
e$(k)=mas$(w(m,n)):zu=0:GOSUB 6120
6050 IF kan2<>1 THEN anz=anz+1:k=anz:nam [0612]
e$(k)=mas$(w(m+1,n)):zu=1:GOSUB 612 [B170]
0
6060 NEXT m
6070 IF az2<>az THEN READ zahl:IF n=1 TH [DBE4]
EN anz=anz+1:name$(anz)=mas$(zahl) [A376]
NEXT n [EEAC]
6080 na$=gruppe$ [99AC]
6100 IF anz>20 THEN anz=20 [C690]
6110 RETURN
6120 IF anz>20 THEN anz=20:CLS#7:LOCATE [E54E]
1,25:PRINT"Tabelle ueberfuellt !":F [FE32]
OR n= 1 TO 2000:NEXT:GOTO 3950 [2AEE]
6130 IF er(m+zu,n)=999 THEN RETURN
6140 IF zu=0 THEN g=1 ELSE g=0
6150 spiele(k)=spiele(k)+1:tor(k)=tor(k) [2156]
+er(m+zu,n):kator(k)=kator(k)+er(m+ [0C5E]
g,n)
6160 IF er(m+zu,n)>er(m+g,n) THEN pu(k)= [08DE]
pu(k)+2
6170 IF er(m+zu,n)=er(m+g,n) THEN pu(k)= [63B6]
pu(k)+1:mipu(k)=mipu(k)+1 [96A0]
6180 IF er(m+zu,n)<er(m+g,n) THEN mipu(k) [5598]
)=mipu(k)+2
6190 RETURN [712C]
6200 REM [579C]
6210 REM *** Routine bei Auftreten eines [FC3E]
'Data exhausted'
6220 REM
6230 PEN 1
6240 PRINT:PRINT"Data's sind ausgegangen [2898]
!" [2A1C]
6250 FOR n= 1 TO 2000:NEXT [3FE2]
6260 RESUME 110 [56A6]
6270 REM
6280 REM Data's fuer Fussball-Bundesliga [D7C2]
[38AA]
6290 REM
6300 DATA 17,1,14,4,3,9,18,7,11,6,13,2,1 [D9F2]
2,10,8,5,15,16
6310 DATA 1,18,16,12,4,11,6,13,7,15,10,8 [A1F4]
5,3,9,14,2,17
6320 DATA 14,5,4,9,15,1,3,10,8,16,18,2,1 [6CF6]
7,6,11,13,12,7 [9AA0]
6330 REM
6340 REM Data's fuer Handball-Bundesliga [3670]
[58A4]
6350 REM
6360 DATA 7,6,5,10,3,12,2,13,1,14,11,4,8 [6162]
,9
6370 DATA 13,7,12,1,14,11,6,9,8,5,4,2,10 [4064]
,3
6380 DATA 11,12,3,5,1,10,9,13,2,14,8,4,6 [D766]
,7 [44AC]
6390 REM
6400 REM Data's fuer B - Jugend [D900]
[589E]
6410 REM
6420 DATA 1,2,3,4,5,6,7,8,9,10,11 [64FC]
[64FE]
6430 DATA 8,9,6,7,4,5,2,3,11,1,10 [0700]
6440 DATA 3,11,5,2,7,4,9,6,8,10,1

```

Listing. »Ligaverwaltung« (Schluß)


```

340 READ a(c) [5A10]
350 NEXT c [39F4]
360 SYMBOL a,a(1),a(2),a(3),a(4),a(5),a(
6),a(7),a(8) [8848]
370 NEXT b [88F6]
380 KEY DEF 71,1,121,89 [8BAA]
390 KEY DEF 43,1,122,90 [409C]
400 KEY DEF 28,1,123,91 [5196]
410 KEY DEF 26,1,125,93 [AC9C]
420 KEY DEF 29,1,124,92 [EDA0]
430 KEY DEF 24,1,126 [AD6E]
440 KEY DEF 17,1,58,42 [7E40]
450 KEY DEF 19,1,59,43 [AD4A]
460 RETURN [A934]
470 REM der Buchstabe 'sz' [AA0C]
480 DATA 126 [241E]
490 DATA &x 01111000 [9372]
500 DATA &x 11001100 [BE62]
510 DATA &x 11001100 [8264]
520 DATA &x 11111000 [9F68]
530 DATA &x 11001100 [3A68]
540 DATA &x 11001100 [6E6A]
550 DATA &x 11111000 [696E]
560 DATA &x 11000000 [AD6A]
570 REM der Buchstabe 'ae' [09C0]
580 DATA 123 [751A]
590 DATA &x 01101100 [D274]
600 DATA &x 00000000 [1F5C]
610 DATA &x 01111000 [6D66]
620 DATA &x 00001100 [CE64]
630 DATA &x 01111000 [916C]
640 DATA &x 11001100 [856C]
650 DATA &x 01110110 [2970]
660 DATA &x 00000000 [4768]
670 REM der Buchstabe 'oe' [84DE]
680 DATA 124 [B61E]
690 DATA &x 01101100 [7176]
700 DATA &x 00000000 [BE5E]
710 DATA &x 00111100 [4068]
720 DATA &x 01100110 [806A]
730 DATA &x 01100110 [946C]
740 DATA &x 01100110 [A86E]
750 DATA &x 00111100 [3070]
760 DATA &x 00000000 [766A]
770 REM der Buchstabe 'ue' [1FEC]
780 DATA 125 [9722]
790 DATA &x 01101100 [4C78]
800 DATA &x 00000000 [D960]
810 DATA &x 01100110 [986A]
820 DATA &x 01100110 [176C]
830 DATA &x 01100110 [736E]
840 DATA &x 01100110 [5F70]
850 DATA &x 00111111 [4C76]
860 DATA &x 00000000 [D16C]
870 REM der Buchstabe 'AE' [3A46]
880 DATA 91 [C1C8]
890 DATA &x 11011011 [8C7E]
900 DATA &x 00111100 [9E6A]
910 DATA &x 01100110 [166C]
920 DATA &x 01100110 [9A6E]
930 DATA &x 01111110 [9974]
940 DATA &x 01100110 [4272]
950 DATA &x 01100110 [A674]
960 DATA &x 00000000 [706E]
970 REM der Buchstabe 'OE' [E564]
980 DATA 92 [ADCC]
990 DATA &x 10111010 [22BE]
1000 DATA &x 01101100 [E7BA]
1010 DATA &x 11000110 [63BC]
1020 DATA &x 11000110 [9FBE]
1030 DATA &x 11000110 [78C0]
1040 DATA &x 01101100 [F7C2]
1050 DATA &x 00111000 [68C2]
1060 DATA &x 00000000 [F9BE]
1070 REM der Buchstabe 'UE' [5AC0]
1080 DATA 93 [561E]
1090 DATA &x 10100101 [09CC]
1100 DATA &x 01100110 [9ABC]
1110 DATA &x 01100110 [BEBE]
1120 DATA &x 01100110 [42C0]
1130 DATA &x 01100110 [F6C2]
1140 DATA &x 01100110 [4AC4]
1150 DATA &x 00111100 [22C6]
1160 DATA &x 00000000 [58C0]
1170 END [E880]
1180 REM-----

1190 REM ** E I N G A B E N ** [0D92]
1200 REM----- [7B60]

1210 CLS#2 [5AB4]
1220 INPUT#1,"{CTRL J}{5 SPACE}Wieviele [AD36]
Iberweisungen wollen Sie schreiben
?(23 SPACE)( max.10){5 SPACE}",i [B94E]
1230 GOSUB 2050 [5B9A]
1240 FOR a=1 TO i [7912]
1250 GOSUB 1890 [ECB4]
1260 PRINT #1,"{CTRL J}{CTRL J}{19 SPACE
}E I N G A B E(2 SPACE)";a;"{CTRL J
}"
1270 LOCATE 14,8:INPUT" ",e$(1,a) [52BC]
1280 LOCATE 14,9:INPUT" ",e$(2,a) [AABA]
1290 LOCATE 56,9:INPUT" ",e$(3,a) [B0C0]
1300 LOCATE 14,11:INPUT" ",e$(4,a) [B6D0]
1310 LOCATE 30,11:INPUT" ",e$(5,a) [F308]
1320 LOCATE 14,14:INPUT" ",e$(6,a) [4008]
1330 LOCATE 14,15:INPUT" ",e$(7,a) [E716]
1340 LOCATE 14,16:INPUT" ",e$(8,a) [CE1C]
1350 LOCATE 54,15:INPUT" ",e$(9,a) [2922]
1360 LOCATE 14,18:INPUT" ",e$(10,a) [202C]
1370 LOCATE 30,18:INPUT" ",e$(11,a) [617C]
1380 LOCATE 32,20:INPUT" ",e$(12,a) [A77C]
1390 PRINT #1,"{CTRL J}{CTRL J}{17 SPACE
}E I G A B E(3 SPACE)D K(2 SPACE)?(
3 SPACE)J/N{CTRL J}" [2D76]

1400 z$=INKEY$
1410 IF z$="j" OR z$="J" THEN GOTO 1450 [6A32]
1420 IF z$="n" OR z$="N" THEN GOTO 1250 [E2C8]
1430 IF z$("<"j" AND z$("<"J" AND z$("<"n" [9AF4]
AND z$("<"N" THEN GOTO 1400 [6F02]
1440 IF z$="" THEN GOTO 1400 [E57A]
1450 NEXT a [1388]
1460 REM----- [8F54]

1470 REM ** DRUCKER ANWEISUNG ** [3494]
1480 REM----- [6EB2]

1490 PRINT #1,"{CTRL J}{CTRL J}{5 SPACE}
J B E R W E I S U N G E N(3 SPACE)D
R U C K E N(3 SPACE)?(2 SPACE)J/N(
CTRL J)" [D498]
1500 z$=INKEY$ [88B4]
1510 IF z$="j" OR z$="J" THEN GOTO 1550 [DECA]
1520 IF z$="n" OR z$="N" THEN END [73F8]
1530 IF z$("<"j" AND z$("<"J" AND z$("<"n" [2470]
AND z$("<"N" THEN GOTO 1500 [CF7E]
1540 IF z$="" THEN GOTO 1500 [A98C]
1550 PRINT #1,"{CTRL J}{CTRL J}{8 SPACE}
J B E R W E I S U N G E N(3 SPACE)W
E R D E N(29 SPACE)G E D R U C K T
" [1E32]
1560 CLS#2 [C846]
1570 FOR a=1 TO i [BA1E]
1580 PRINT #8,"{CTRL J}{CTRL J}" [BBF4]
1590 PRINT #8,TAB (3);e$(1,a) [BBE2]
1600 PRINT #8,TAB (3);e$(2,a); [234A]
1610 PRINT #8,TAB (4);e$(3,a); [7046]
1620 PRINT #8 [B4E2]
1630 PRINT #8,TAB (3);e$(4,a); [9454]
1640 PRINT #8,TAB (19);e$(5,a); [9850]
1650 PRINT #8,"{CTRL J}" [75DC]
1660 PRINT #8,TAB (3);e$(6,a) [55E8]
1670 PRINT #8,TAB (3);e$(7,a); [5862]
1680 PRINT #8,TAB (4);e$(9,a) [895C]
1690 PRINT #8,TAB (3);e$(8,a) [93F2]
1700 PRINT #8 [EBE0]
1710 PRINT #8,TAB (3);e$(10,a); [65AC]
1720 PRINT #8,TAB (21);e$(11,a) [819A]
1730 PRINT #8,"{CTRL J}" [A4DA]
1740 PRINT #8,TAB (19);e$(12,a) [03AE]
1750 PRINT #8,"{CTRL J}{CTRL J}{CTRL J}{
CTRL J}{CTRL J}{CTRL J}{CTRL J}" [B256]
1760 NEXT a [B85C]
1770 PRINT #1,"{CTRL J}{CTRL J}{11 SPACE}
}W E I T E R E(3 SPACE)E I G A B E
N(3 SPACE)?(2 SPACE)J/N(CTRL J)" [EEC8]
1780 z$=INKEY$ [F6DE]
1790 IF z$="j" OR z$="J" THEN RUN [81AE]
1800 IF z$="n" OR z$="N" THEN GOTO 1830 [FD0E]
1810 IF z$("<"j" AND z$("<"J" AND z$("<"n" [AF94]
AND z$("<"N" THEN GOTO 1780 [19A2]
1820 IF z$="" THEN GOTO 1780
1830 REM-----

1840 REM ** E N D E ** [1696]
1850 REM----- [F5EA]

1860 CLS#1:CLS#2: [EE9A]
1870 PRINT#2,"{CTRL J}{CTRL J}{CTRL J}{C
TRL J}{CTRL J}{CTRL J}{CTRL J}{CTRL
J}{12 SPACE}DAS PROGRAMM KANN MIT
GOTO 1460 WIEDER GESTARTET WERDEN(C
TRL J){36 SPACE}OHNE DASS DATEN(2 S
PACE)VERLOREN GEHEN !!" [BDA0]

1880 END [617C]
1890 REM----- [0D90]

1900 REM ** ZEILEN L\SCHEN ** [56A2]
[DSAC]

```

Listing. »Druckprogramm für Banküberweisungen«

1910 REM-----

```

1920 LOCATE 14,8:PRINT SPACE$(40)
1930 LOCATE 14,9:PRINT SPACE$(40)
1940 LOCATE 56,9:PRINT SPACE$(12)
1950 LOCATE 14,11:PRINT SPACE$(15)
1960 LOCATE 30,11:PRINT SPACE$(38)
1970 LOCATE 14,14:PRINT SPACE$(38)
1980 LOCATE 14,15:PRINT SPACE$(38)
1990 LOCATE 14,16:PRINT SPACE$(38)
2000 LOCATE 54,15:PRINT SPACE$(14)
2010 LOCATE 14,18:PRINT SPACE$(13)
2020 LOCATE 30,18:PRINT SPACE$(38)
2030 LOCATE 32,20:PRINT SPACE$(10)
2040 RETURN
2050 REM-----
    
```

2060 REM ** M A S K E **
2070 REM-----

```

2080 MOVE 96,10:DRAW 543,10
2090 MOVE 543,10:DRAW 543,342
2100 MOVE 543,342:DRAW 96,342
2110 MOVE 96,342:DRAW 96,10
2120 MOVE 96,40:DRAW 543,40
2130 MOVE 96,39:DRAW 543,39
2140 MOVE 96,105:DRAW 543,105
2150 MOVE 96,134:DRAW 543,134
2160 MOVE 96,133:DRAW 210,133
2170 MOVE 97,133:DRAW 97,104
2180 MOVE 97,106:DRAW 210,106
2190 MOVE 96,220:DRAW 543,220
2200 MOVE 543,219:DRAW 417,219
2210 MOVE 417,219:DRAW 417,136
    
```

[E794]
[E4B2]
[D3B6]
[6CC2]
[77D0]
[2DD8]
[D9E4]
[8DE8]
[41EC]
[80C4]
[2FC2]
[73CE]
[A2B2]
[918C]

[368C]
[65CA]

[6076]
[61DC]
[53A8]
[E1AE]
[DB26]
[56DE]
[AF00]
[SCAA]
[5CB4]
[SCA0]
[E45A]
[23A6]
[C4AC]
[9716]
[F114]

```

2220 MOVE 415,220:DRAW 415,134
2230 MOVE 417,136:DRAW 543,136
2240 MOVE 96,247:DRAW 543,247
2250 MOVE 227,247:DRAW 227,220
2260 MOVE 428,247:DRAW 428,295
2270 MOVE 543,295:DRAW 96,295
2280 MOVE 246,75:DRAW 331,75
2290 MOVE 348,75:DRAW 529,75
2300 REM-----
    
```

[E988]
[43AC]

2310 REM ** B E S C H R I F T U N G **
2320 REM-----

```

2330 LOCATE 27,24:PRINT "Bitte nicht bes
chriften"
2340 LOCATE 35,22:PRINT "Datum"
2350 LOCATE 49,22:PRINT "Unterschrift"
2360 LOCATE 14,17:PRINT "Konto Nr"
2370 LOCATE 35,17:PRINT "Auftraggeber"
2380 LOCATE 14,13:PRINT "Verwendungszwe
k (nur f)r Empf(nger)"
2390 LOCATE 59,12:PRINT "D M"
2400 LOCATE 14,10:PRINT "Konto Empf(nger
"
2410 LOCATE 31,10:PRINT "bei ?"
2420 LOCATE 40,10:PRINT "o.e.a. Konto"
2430 LOCATE 14,7:PRINT "Empf(nger (genaue
Anschrift)"
2440 LOCATE 56,7:PRINT "Bankleitzahl"
2450 LOCATE 30,5:PRINT "]BERWEISUNGSAUFT
RAG"
    
```

[D18C]
[5982]
[5BBE]
[ASD6]
[54A4]
[0D74]
[2A56]
[893E]
[FB6A]
[48D4]
[9B8C]
[DB92]
[A128]
[82CE]

Listing. »Druckprogramm für Banküberweisungen«
(Schluß)



Dr. P. Albrecht
**MULTIPLAN für den
Schneider CPC**
September 1985, 226 Seiten

MULTIPLAN wurde ursprünglich für das 16-Bit-Betriebssystem MS-DOS entwickelt. Inzwischen ist aber auch die in diesem Buch beschriebene CP/M-Version für den Schneider CPC auf dem Markt, die den vollen Leistungsumfang der 16-Bit-Version enthält. Das vorliegende Buch soll eine praktische Einführung in den Umgang mit MULTIPLAN auf dem Schneider CPC geben. Anhand von praxisnahen Beispielen werden alle Befehle und Funktionen in der Reihenfolge beschrieben, die der Arbeit in der Praxis entspricht. Bereits nach Abschluß des ersten Kapitels werden Sie in der Lage sein, eigene kleine MULTIPLAN-Anwendungen zu realisieren. Ein Merkmal von MULTIPLAN ist, daß Kalkulationen schnell und einfach erstellt werden können.

Best.-Nr. MT 835
ISBN 3-89090-186-7

DM 49,-



Dr. P. Albrecht
dBASE II für den Schneider CPC
September 1985, 280 Seiten

Das vorliegende Buch gibt nach einer kurzen Einführung in den Komplex »Datenbanken« eine Anleitung für den praktischen Umgang mit dBASE II. Schon nach Beherrschung weniger Befehle ist der Anwender in der Lage, Dateien zu erstellen, mit Informationen zu laden und auszuwerten. Dabei hilft ihm ein integrierter Reportgenerator, der im Dialog mit dem Benutzer Berichte gestaltet und in Tabellenform ausdrückt.

Best.-Nr. MT 837
ISBN 3-89090-188-3

DM 49,-



C. Strauß
**Schneider CPC
Grafik-Programmierung**
Dezember 1985, ca. 300 Seiten

Dieses Buch wendet sich an die Schneider CPC-Besitzer, die alles über die Grafikfähigkeiten ihres Computers wissen wollen. Es bietet einen umfassenden Überblick über die verschiedenen Anwendungsbereiche der Grafikprogrammierung: zwei- und dreidimensionale Diagrammdarstellungen, Definition und Bewegung von Sprites, Entwurf von Titelgrafiken oder den Einsatz der Grafik bei der Unterstützung anderer Programme.

Best.-Nr. MT 782
ISBN 3-89090-182-4

DM 46,-



J. Hückstädt
Der Schneider CPC 6128
September 1985, 273 Seiten

Dieses Buch ist für jeden CPC 6128-Besitzer eine wertvolle Hilfe, die vielfachen Möglichkeiten dieses bisher einmaligen Computers kennenzulernen und anzuwenden. Der Computereuling wird Schritt für Schritt in den Umgang mit dem Computer und die BASIC-Programmierung eingeführt, bis er alle notwendigen Kenntnisse besitzt, die mancher Profi bereits mitbringt. Aber an dieser Stelle wird das Programmieren mit dem CPC 6128 erst interessant, nämlich dann, wenn es darum geht, eine eigene Dateiverwaltung aufzubauen oder Grafik und Sound zu programmieren.

Weiterhin erfahren Sie alles über CP/M Plus auf dem CPC 6128, dem universellen Betriebssystem für die größte Softwarebibliothek der Welt. Dadurch stehen Ihnen zusätzliche ungeahnte Möglichkeiten zur Verfügung.

• Ein unentbehrliches Lehrbuch und Nachschlagewerk für jeden, der mit dem CPC 6128 arbeitet.
Best.-Nr. MT 849
ISBN 3-89090-192-1

DM 46,-

Markt & Technik-Fachbücher erhalten Sie bei Ihrem Buchhändler.

Markt & Technik
BUCHVERLAG

Hans-Pinsel-Straße 2, 8013 Haar bei München

Nebenkosten einfach abgerechnet



Auch Hausbesitzern kann der Schneider-Computer eine Hilfe bei unangenehmen Arbeiten sein. Beispielsweise bei der jährlichen Nebenkostenabrechnung.

Mit diesem Basic-Listing können Sie die Nebenkostenabrechnung für Ihre Mieter sehr einfach und schnell erstellen. Alle benötigten Daten werden vom Programm angefordert und automatisch auf die einzelnen Parteien umgelegt. Die Rechnung kann auf jedem Drucker ausgegeben werden.

Da die Schneider-Computer vom Betriebssystem her keine Abfrage machen, ob ein Drucker angeschlossen ist, darf das Programm ohne Ausgabegerät nicht laufen. Es würde sich »aufhängen«.

(Denis Wetzel)

```

*****
JAHRESABRECHNUNG DER NEBENKOSTEN
*****
Hamburg 20, den 02.02.1986

Manfred Hausmeyer
Hauptstr. 12
2000 Hamburg 20

HERRN/FRAU
Werner Frank
Wiener Str. 1
2000 Hamburg 12

NEBENKOSTENABRECHNUNG FUER DAS JAHR 1985
K O S T E N A U F S T E L L U N G

ABWASSER, BELEUCHTUNG, WASSER INSGESAM           : 714.41 DM
DAVON FUER IHRE WOHNUNG ANTEILIG                 : 292.91 DM

EINE EVENTUELL KUERZERE MIETPERIODE WURDE
BERUECKSICHTIGT.

ZUZUEGLICH PAUSCHAL FUER 2 PERSONEN             : 240 DM
MUELLABFUHR                                       : 120 DM
ABZUEGLICH VORAUSZAHLUNGEN                       : -600 DM
ZUZUEGLICH SONSTIGES                             : 121.12 DM

DIE VON IHNEN SOMIT ZU LEISTENDE NACHZAHLUNG
BETRAEGT                                         : 174.03 DM
*****

BITTE ZAHLEN SIE DIESEN BETRAG BIS ZUM 15.03.1986
AUF UNSER UNTENSTEHENDES KONTO EIN.
BIS ZU DIESEM TAG KOEMMEN AUCH DIE BERECHNUNGS-
UNTERLAGEN EINGESEHEN WERDEN.

BANK: Hansabank Hamburg
KTO.-NR.: 123-1234567
BLZ: 111 111 11

MIT FREUNDLICHEN GRUESSEN

Manfred Hausmeyer

So kann im nächsten Jahr
Ihre Nebenkostenabrechnung auch aussehen
    
```

```

10 MODE 2:INK 0,0:BORDER 0:INK 1,18 [2C14]
20 FOR I=1 TO 80:PRINT CHR$(154);:NEXT I [08F6]
30 LOCATE 3,1:PRINT" Hausverwaltung(9 SP [564E]
   ACE){CTRL X} H A U P T M E N U E {CTR
   L X}{9 SPACE}(C)1985 SOFTWARE PRODUCT
   S ";
40 FOR I=1 TO 80:PRINT CHR$(154);:NEXT I [8EFA]
50 LOCATE 26,7:PRINT"(CTRL X) 1 (CTRL X) [609A]
   Informationen"
60 LOCATE 26,9:PRINT"(CTRL X) 2 (CTRL X) [D46E]
   Maske allgemein"
70 LOCATE 26,11:PRINT"(CTRL X) 3 (CTRL X [F734]
   ) Maske Mieter"
80 LOCATE 26,13:PRINT"(CTRL X) 4 (CTRL X [A498]
   ) Nebenkostenabrechnung drucken"
90 LOCATE 26,15:PRINT"(CTRL X) E (CTRL X [3B52]
   ) Programm-Ende"
100 ee$=INKEY$:IF ee$=""THEN 100 [FF70]
110 ee=INSTR(1,"1234eE",ee$):IF ee=0 TH [A812]
   EN 100
120 ON ee GOTO 1530,140,670,930,1670 [B612]
130 ON ERROR GOTO 140
140 CLS [752E]
150 REM [5D34]
160 REM hausverwaltung 1.1 [36D6]
170 REM IDEE FUER C 64: Dr. [6F62]
   GUENTER MERSSON
180 REM ----- [6F5A]
190 REM VERSION 1.1 fuer Schneider CPC 4 [20B0]
   64/664/6128:
200 REM by SOFTWARE PRODUCTS Ltd.- SOFTW [70E0]
   AREPROBLEMLUESUNGEN FUER CPC 464/664
   /6128 [33CC]
210 REM c/o Denis Wetzel [C03E]
220 REM Am Holzplatz 1 [F5AE]
230 REM 6730 Neustadt/Weinstrasse [4C34]
240 REM [66CE]
250 FOR i=1 TO 80:PRINT CHR$(154);:NEXT
260 PRINT" Hausverwaltung(4 SPACE){CTRL [359E]
   X} M A S K E - A L L G E M E I N {CT
   RL X}{3 SPACE}(C)1985 SOFTWARE PRODU
   CTS" [38D2]
270 FOR i=1 TO 80:PRINT CHR$(154);:NEXT [E38E]
280 PRINT
290 LOCATE 10,5:PRINT"--> BITTE ALLE EIN
   GABEN MIT DER [ENTER]-TASTE ABSCHLIE
   SSEN <--" [77F4]
300 PRINT [6180]
310 LOCATE 1,9:INPUT"Abrechnung fuer das [5D2C]
   Jahr : ";A$ [FE98]
320 IF a$=""THEN 300
330 LOCATE 1,12:INPUT"Name/Firma/Vermiet [8818]
   er : ";V$ [B78C]
340 IF V$=""THEN 330
350 LOCATE 1,12:INPUT"Strasse, Hausnumme [FAF2]
   r(2 SPACE): ";L$ [8980]
360 IF L$=""THEN 350
370 LOCATE 1,12:INPUT"Plz(18 SPACE): ";U [06C8]
   $ [599A]
380 IF U$=""THEN 370
390 LOCATE 1,12:INPUT"Ort(18 SPACE): ";O [B50A]
   RT$ [0CD0]
400 IF ORT$=""THEN 390
410 LOCATE 1,12:INPUT"Bank(17 SPACE): "; [2F12]
   Y$ [428E]
420 IF Y$=""THEN 410
430 LOCATE 1,12:INPUT"Kontonummer(10 SPA [439A]
   CE): ";X$ [D694]
440 IF X$=""THEN 430
450 LOCATE 1,12:INPUT"Bankleihzahl(9 SPA [6892]
   CE): ";D$ [4074]
460 IF D$=""THEN 450 [DA90]
470 PRINT
480 PRINT"NEBENKOSTEN FUER:" [16AA]
490 PRINT [8B94]
500 LOCATE 1,16:INPUT"W A S S E R : ";B [D096]
510 PRINT [EC86]
520 PRINT [FF88]
530 LOCATE 1,16:INPUT"A B W A S S E R : [5526]
   ";D
540 LOCATE 1,16:INPUT"B E L E U C H T U [49A2]
   N G : ";C [64BA]
550 G=B+D+C [6390]
560 PRINT
570 PRINT"DIE GESAMTKOSTEN BETRAGEN";G;: [4A06]
   PRINT"DM."
580 IF I=0 THEN PRINT CHR$(24):PRINT"-->
   SIE HABEN KEINE ANGABEN GEMACHT <--
   ";PRINT CHR$(24):FOR n=1 TO 3000:NEX
   T:GOTO 10 [849C]
590 PRINT"Verteilerschluessel nach Punkt
   en," [CBCE]
600 LOCATE 1,20:INPUT"Gesamtpunkte fuer
   das Haus : ";H [9F56]
    
```

Listing. »Nebenkostenabrechnung«

Anwendungs-Listing

```

610 IF H=0 OR H<0 THEN 600 [3C6A]
620 PRINT [058A]
630 INPUT"Heutiges Datum : ";F$ [440A]
640 PRINT [E78E]
650 INPUT"Eventuelle Nachzahlung des Mie [0024]
ters bis zum : ";0$ [7FEC]
660 GOTO 10 [723E]
670 CLS [3DDC]
680 FOR i=1 TO 80:PRINT CHR$(154);:NEXT
690 PRINT" Hausverwaltung(7 SPACE){CTRL
X} M A S K E - M I E T E R {CTRL X}{
6 SPACE}(C)1985 SOFTWARE PRODUCTS" [27DC]
700 FOR i=1 TO 80:PRINT CHR$(154);:NEXT [4FCE]
710 LOCATE 1,7:INPUT"Name des Mieters :
";G$:PRINT [70DE]
720 IF G$="" THEN 710 [5F76]
730 LOCATE 1,9:INPUT"Strasse, Hausnummer
":S$ [A470]
740 IF S$="" THEN 730 [CD96]
750 LOCATE 1,11:INPUT"Plz, Ort : ";T$ [9D8A]
760 IF T$="" THEN 750 [35A0]
770 LOCATE 1,13:INPUT"Auf den Mieter ent
fallende Punkte : ";J [C02C]
780 IF J=0 THEN GOTO 770 [F916]
790 LOCATE 1,15:PRINT"Berechnungsgrundla
ge: Mietzeit 12 Monate" [A07E]
800 LOCATE 1,17:INPUT"Kuerzere Mietzeit
(in Monaten) : ";P [81B4]
810 IF P=0 THEN P=12 [9108]
820 LOCATE 1,19:INPUT"Pauschalbetrag fue
r 2 Personen : ";R [3444]
830 LOCATE 1,21:INPUT"sonstiges(3 SPACE)
":E [6BF4]
840 LOCATE 1,21:INPUT"Muellabfuhr : ";L [93B4]
850 LOCATE 1,21:INPUT"geleistete Vorausz
ahlung : ";M [5152]
860 M=-M [C284]
870 LOCATE 1,23:INPUT"einbehaltene Siche
rheit : ";N [F2EC]
880 N=-N [CD8C]
890 K=G/H*J*P/12 [610E]
900 K=INT(K*100+0.555)/100 [66D6]
910 Q=K+L+E+R+M+N [3BD0]
920 GOSUB 10 [7278]
930 LOCATE 26,23:PRINT"{CTRL X} DRUCKER
BEREIT J/N {CTRL X}" [73B4]
940 tt$=INKEY$:IF tt$="" THEN 940 [5A18]
950 IF UPPER$(tt$)="J" THEN 970 [0FAB]
960 IF UPPER$(tt$)="N" THEN 10 [7B34]
970 PRINT CHR$(24):LOCATE 26,23:PRINT" A
BRECHNUNG WIRD GEDRUCKT - BITTE WART
EN ";PRINT CHR$(24) [F838]
980 PRINT #8, CHR$(0) [422E]
990 PRINT #8, "*****
*****
*****:PRINT #8,"(79 SP
ACE)":PRINT #8, "JAHRESABRECHNUNG DE
R NEBENKOSTEN" [FA4A]
1000 PRINT #8, "(79 SPACE)" [32B2]
1010 PRINT #8, "*****
*****
*****" [4BE0]
1020 PRINT #8, "{37 SPACE}"ORT$", den "F
$:PRINT #8, V$ [6A1A]
1030 PRINT #8, L$:PRINT #8, U$ "ORT$:PR
INT #8 [95DC]
1040 PRINT #8 [96DA]
1050 PRINT #8, "HERRN/FRAU/FIRMA" [82F0]
1060 PRINT #8, G$ [B94C]
1070 PRINT #8, S$ [4366]
1080 PRINT #8, T$ [766A]
1090 PRINT #8:PRINT #8:PRINT #8:PRINT #8
, "NEBENKOSTENABRECHNUNG FUER DAS J
AHR "A$ [F6EA]
1100 PRINT #8:PRINT #8, "K O S T E N A U
F S T E L L U N G" [88B4]
1110 PRINT #8, "-----
" [0190]
1120 PRINT #8 [ADD8]
1130 PRINT #8, "ABWASSER, BELEUCHTUNG, W
ASSER INSGESAMT(10 SPACE):(2 SPACE)
"G"DM" [5632]
1140 PRINT #8 [A9DC]
1150 PRINT #8, "DAVON FUER IHRE WOHNUNG
ANTEILIG(17 SPACE):(2 SPACE)"K"DM" [B8FC]
1160 PRINT #8 [E5E0]
1170 PRINT #8, "EINE EVENTUELL KUERZERE
MIETPERIODE WURDE(39 SPACE)BERUECKS
ICHTIGT." [EFDE]
1180 PRINT #8 [B1E4]
1190 IF R=0 THEN 1210 [4B18]
1200 PRINT #8, "ZUZUEGLICH PAUSCHAL FUER
2 PERSONEN(14 SPACE):(2 SPACE)"R"D
M":PRINT #8 [3D8E]
1210 IF L=0 THEN 1230 [4C02]
1220 PRINT #8, "MUELLABFUHR(38 SPACE):(2
SPACE)"L"DM":PRINT #8 [72A2]
1230 IF M=0 THEN 1250 [120C]
1240 PRINT #8, "ABZUEGLICH VORAUSZAHLUNG
EN(23 SPACE):(2 SPACE)"M"DM":PRINT
#8 [E48E]
1250 IF N=0 THEN 1270 [D416]

1260 PRINT #8, "GUTHABEN AUS EINBEHALTEN
ER SICHERHEIT BETRAEGT(3 SPACE):(3
SPACE)"N"DM":PRINT #8 [CE06]
1270 IF E=0 THEN 1300 [14FC]
1280 PRINT #8, "ZUZUEGLICH SONSTIGES(29
SPACE):(3 SPACE)"E"DM" [1380]
1290 PRINT #8:PRINT #8 [766C]
1300 IF Q>0 THEN 1320 [510E]
1310 IF Q<0 THEN 1430 [2410]
1320 PRINT #8, "DIE VON IHNEN SOMIT ZU L
EISTENDE NACHZAHLUNG" [8800]
1330 PRINT #8, "BETRAEGT(41 SPACE):(2 SP
ACE)"Q"DM" [421A]
1340 PRINT #8, "{53 SPACE}=====" [8596]
1350 PRINT #8:PRINT #8, "BITTE ZAHLN SI
E DIESEN BETRAG BIS ZUM " 0$ " [50B4]
1360 PRINT #8, "AUF UNSER UNTENSTEHENDES
KONTO EIN." [A21A]
1370 PRINT #8, "BIS ZU DIESEM TAG KOENNE
N AUCH DIE BERECHNUNGS-(33 SPACE)UN
TERLAGEN EINGESEHEN WERDEN." [E710]
1380 PRINT #8:PRINT #8 [EF6C]
1390 PRINT #8, "BANK: "Y$ [6DF0]
1400 PRINT #8, "KTO.-NR.: "X$ [45D4]
1410 PRINT #8, "BLZ: "D$ [B150]
1420 GOTO 1490 [831C]
1430 Q=-Q [2EE8]
1440 PRINT #8, "IHR GUTHABEN BETREAGT SO
MIT(22 SPACE):(2 SPACE)"Q"DM" [5498]
1450 PRINT #8, "{53 SPACE}=====" [AB9A]
1460 PRINT #8, "DER BETRAG WIRD IHNEN IN
DEN NAECHSTEN TAGEN AUF(32 SPACE)I
HR KONTO UEBERWIESEN." [E64E]
1470 PRINT #8, "BIS ZUM " 0$ " KOENNEN D
IE BERECHNUNGS- [946C]
1480 PRINT #8, "UNTERLAGEN EINGESEHEN WE
RDEN." [4810]
1490 PRINT #8 [9AEC]
1500 PRINT #8:PRINT #8, "MIT FREUNDLICHE
N GRUESSEN" [C49A]
1510 PRINT #8:PRINT #8, V$ [FBEE]
1520 GOTO 10 [5A44]
1530 MODE 2:INK 0,0:BORDER 0:INK 1,18 [08E4]
1540 FOR i=1 TO 80:PRINT CHR$(154);:NEXT
[3834]

1550 PRINT" Hausverwaltung(6 SPACE){CTRL
X} I N F O R M A T I O N E N {CTRL
X}(5 SPACE)(C)1985 SOFTWARE PRODUC
TS" [9DDE]
1560 FOR i=1 TO 80:PRINT CHR$(154);:NEXT
[4A38]

1570 LOCATE 5,6:PRINT"{CTRL X} a {CTRL X
} ALLGEMEINES":LOCATE 5,7:PRINT"Das
Programm Hausverwaltung ist in er
ster Linie fuer Vermieter geschriebe
n. Sie koennen so bequem die Nebenk
ostenabrechnungen fuer ihre Mieter
erstellen." [95B2]
1580 LOCATE 5,9:PRINT"Es bietet jedoch a
uch fuer Mieter eine Kontrolle zum
Ueberpruefen ihrer Neben - kostenab
rechnung." [F426]
1590 LOCATE 5,14:PRINT"{CTRL X} b {CTRL
X} EINGABE VON WERTEN":LOCATE 5,15:
PRINT"Sie muessen zuerst Menuepunkt
2 waehlen, um allgemeingueeltige We
rte (Werte, die fuer jeden Mieter G
ueeltigkeit haben) einzugeben. Jede
Eingabe muss mit [ENTER](2 SPACE)be
staetigt werden." [16FC]
1600 LOCATE 20,18:PRINT"Nach der Eingabe
kommen Sie wieder zum Hauptmenue." [209E]
1610 LOCATE 5,18:PRINT"Danach waehlen Si
e Menuepunkt 3 an, um nun Werte fue
r einen Mieter einzugeben. (2 SPACE)
Haben Sie alle Werte eingegeben, ko
mmen Sie wieder zum Hauptmenue, wo
Sie mit" [1548]
1620 LOCATE 5,20:PRINT"Menuepunkt 4 eine
n Ausdruck der Abrechnung des gerad
e eingegebenen Mieters er-(2 SPACE)
halten. Danach erscheint wieder das
Hauptmenue und Sie koennen die Dat
en fuer(3 SPACE)den naechsten Miete
r eingeben (Punkt 3) und Sie drucke
n (Punkt 4)." [B57E]
1630 LOCATE 1,25:PRINT"{CTRL X} [ENTER]-
Taste druecken {CTRL X}" [82AC]
1640 aa$=INKEY$:IF aa$="" THEN 1640 [4638]
1650 IF UPPER$(aa$)="M" THEN 10 [1000]
1660 GOTO 10 [304E]
1670 LOCATE 26,23:PRINT"{CTRL X} ENDE J/
N {CTRL X}" [697A]
1680 fe$=INKEY$:IF fe$="" THEN 1680 [F86C]
1690 IF UPPER$(fe$)="J" THEN 1720 [31E6]
1700 IF UPPER$(fe$)="N" THEN 10 [BB0C]
1710 END [EE80]
1720 MODE 1:INK 0,1:BORDER 1:INK 1,24: C
ALL &BB00:CALL &BB0 [DFC2]

```

Listing. »Nebenkostenabrechnung« (Schluß)

Der Fall »Famit AG«



Sind Sie einem großen Wirtschaftsverbrechen auf der Spur oder handelt es sich nur um persönliche Rachegefühle wie bei Dallas. Diese und andere Fragen stellen sich in dem menügesteuerten Adventure »Famit AG«.

Auf dem Gelände der »Famit AG« haben sich seltsame Dinge ereignet. Sie müssen Ihren freien Sonntag opfern, um den Verdächtigen vor der Flucht noch zu stellen. Sind die 2,3 Millionen Mark wirklich ein Lottogewinn? Wer ist der Schatz, dem Angela auf dem von Ihnen gefundenen Zettel ihre Liebe beschwört? Fragen über Fragen, die alle scheinbar zusammenhanglos im Raum stehen. Um den Fall zu klären, müssen Sie — wie Kommissar Schimantzki — zu unorthodoxen Mitteln greifen.

Eingefleischten Adventure-Freaks wird die Menüsteuerung des Programms zuerst ungewohnt sein. Aber seien Sie versichert, Sie haben so das Geschehen viel besser im Griff. Die Spielanleitung und die Aufgabe dieses Adventures wird während des Ablaufs erklärt. Viel Spaß mit dem Fall »Famit AG«.

(Jobst-Hinrich Jacke)

```

10 REM----- [2AB2]
20 REM- [5640]
30 REM- Die Akte "Famit AG" [B0EA]
40 REM- [DE44]
50 REM- 1985 Copyright by [8B5C]
60 REM- Jobst-Hinrich Jacke [4E6C]
70 REM- Hinrich Wragestr.16a [099E]
80 REM- 2427 Malente [88B6]
90 REM- [524E]
100 REM----- [CAD2]
110 DEFINT a-z [2328]
120 MODE 2:INK 0,0:INK 1,0:PAPER 0:PEN 1 [B51E]
: BORDER 0:CLS [1548]
130 GOSUB 2860 [1548]
140 raum=8:vmax=16:rmax=94:omax=123:bdma [B98E]
x=14:bmmax=6:imax=4:zmax=360:bueronu
mmer$="4623519":safekomb$="23111985"
:ok$="Okay !":foto=12:cmax=380:zug=1
150 DIM verb$(vmax),raum$(rmax),objekt$( [B98E]
omax),objekt(omax),bedingung(bdmax),
bemerkung$(bmmax),richtung(rmax,6),r
ichtung$(6),oy(16),iy(6),code(cmax)
160 bemerkung$(1)="Sie sehen nichts beso [B05C]
nderes." [0826]
170 bemerkung$(2)="So stark sind sie nic [549A]
ht !" [DD28]
180 bemerkung$(3)="Wie stellen Sie sich [DD28]
das vor ?" [E1B6]
190 bemerkung$(4)="Ihre Gedankeng(nge{2 [E1B6]
SPACE}sind leiderf}r diesen{2 SPACE}
Computer zu kompli-ziert !" [0610]
200 bemerkung$(5)="Es sind zuviele !" [0610]
210 bemerkung$(6)="Das ist nicht notwend [3DF4]
ig." [E880]
220 FOR x=1 TO vmax [E880]
230 READ verb$(x) [EC1A]
240 NEXT [58EA]
250 FOR x=1 TO omax [5A78]
260 READ objekt$(x),objekt(x) [FCAB]
270 NEXT [60F0]
280 FOR x=1 TO rmax:READ raum$(x) [CDB4]
290 FOR y=1 TO 6:READ richtung(x,y) [6514]

```

```

300 NEXT:NEXT [D2D6]
310 LOCATE 22,22:PRINT"(CTRL X) Bitte ir [B6C4]
gendeine <Taste> dr}cken (CTRL X)":C
ALL &BB06
320 MODE 2:PAPER 0:INK 0,0:BORDER 0:PEN [9E92]
1:INK 1,26:CLS
330 LOCATE 30,3:PRINT"Die AKTE 'Famit AG [A672]
"
340 PRINT:PRINT STRING$(80,210):PRINT:PR [D702]
INT"{4 SPACE}Sie sind Beamter der Kr
iminalpolizei{2 SPACE}und werden im
Laufe des Spiels mit{8 SPACE}einem w
ichtigen Fall betraut . Es ist aller
dings Eile geboten , denn der"
350 PRINT"{4 SPACE}Verd{chtigte versucht [7018]
zu fliehen . Sie m}ssen{2 SPACE}also
Ihren{2 SPACE}Sonntag opfern{8 SPAC
E}und sich auf den Weg machen , ihn
zu verhaften .{2 SPACE}Nat}rlich bra
uchen Sie{8 SPACE}daf}r Beweise ,{2
SPACE}die{2 SPACE}Sie selber finden
m}ssen .{2 SPACE}Au}erdem haben Sie
nur"
360 PRINT"{4 SPACE}von 10 - 16 Uhr Zeit [66CA]
.{2 SPACE}Wenn Sie{2 SPACE}um 16 Uhr
noch nichts erreicht haben ,{8 SPAC
E}ist dem Verd{chtigem die Flucht no
ch gegl}ckt .{2 SPACE}Sie werden{2 S
PACE}w}hrend der{8 SPACE}Ermittlung
n nicht{2 SPACE}auf andere Personen
sto}en, da es , wie oben gesagt,"
370 PRINT"{4 SPACE}Sonntag ist , und som [FEBE]
it liegt{2 SPACE}das Gel}nde der 'Fa
mit AG'{2 SPACE}verlassen da,{8 SPAC
E}bis auf eine Ausnahme !"
380 PRINT:PRINT STRING$(80,210):PRINT:PR [7F5A]
INT TAB(23):"(CTRL X) Bitte irgendei
ne <Taste> dr}cken (CTRL X)" [1904]
390 FOR x=1 TO 6:READ richtung$(x):NEXT [F200]
400 FOR x=1 TO cmax:READ code(x):NEXT [A102]
410 CALL &BB06
420 CLS:LOCATE 30,3:PRINT"Die AKTE 'Fami [77AA]
t AG "
430 PRINT:PRINT STRING$(80,210):PRINT"{4 [E11A]
SPACE}Funktion der Bildschirmfenste
r :{2 SPACE}oben links{6 SPACE}:{4 S
PACE}Ort":PRINT TAB(39):"halb oben l
inks :{4 SPACE}Ausg}nge":PRINT TAB(3
9):"unten links{5 SPACE}:{4 SPACE}Ko
mmentar":PRINT TAB(39):"mitte{11 SPA
CE}:{4 SPACE}Verben":
440 PRINT TAB(39):"oben rechts{5 SPACE}: [85E8]
{4 SPACE}Gegenst}nde/Ausg}nge":PRINT
TAB(39):"unten rechts{4 SPACE}:{4 S
PACE}Inventur"
450 PRINT:PRINT"{4 SPACE}Steuerung{22 SP [F42A]
ACE}:{4 SPACE}";CHR$(240)"{13 SPACE}
:{4 SPACE}Cursor nach oben":PRINT TA
B(39):"{5 SPACE}";CHR$(241)"{10 SPAC
E}:{4 SPACE}Cursor nach unten":
PRINT TAB(39):"<COPY>{10 SP
ACE}:{4 SPACE}Ausl'sen der Funktion" [856A]
460 PRINT"{7 SPACE}Um bei{2 SPACE}einige [F42A]
n Verben (z.B.'untersuche'}) vom{2 SP
ACE}Fenster 'Gegenst}nde'{13 SPACE}zu
m{2 SPACE}Fenster 'Inventur' und umge
kehrt umzuschalten , m}ssen Sie die{
14 SPACE}Funktionen {CTRL X}{4 SPAC
E}";CHR$(240):"{4 SPACE}{CTRL X} bzw.
{CTRL X}{4 SPACE}";CHR$(241):"{4 SP
ACE}{CTRL X} ausl'sen."
470 PRINT STRING$(80,210):PRINT:PRINT TA [71D8]
B(23):"(CTRL X) Bitte irgendeine <Ta [BD3C]
ste> dr}cken (CTRL X)":CALL &BB06
480 CLS
490 WINDOW #1,2,49,2,4:WINDOW #2,51,79,2 [A626]
,17:WINDOW #3,2,49,6,7:WINDOW #4,34,
49,9,24:WINDOW #5,51,79,19,24:WINDOW
2,32,9,24:PEN #2,1:PAPER #2,0
500 PLOT 4,8:DRAW 4,392:PLOT 3,8:DRAW 3, [C31E]
392:DRAW 636,392:DRAW 636,8:DRAW 637,
392:DRAW 637,8:DRAW 4,8:PLOT 396,39
2:DRAW 396,8:PLOT 395,392:DRAW 395,8
:PLOT 4,280:DRAW 395,280:PLOT 4,328:
DRAW 396,328:PLOT 395,120:DRAW 636,1 [7404]
20
510 PLOT 260,280:DRAW 260,8:PLOT 259,280 [48D2]
:DRAW 259,8
520 FOR n=1 TO 16:PRINT #4, " ";CHR$(144) [5396]
: " ";verb$(n):NEXT:GOTO 540 [76C4]
530 FOR n=1 TO 7000:NEXT:PRINT:PRINT:zug [F342]
=zug+1 [E8EC]
540 CLS:CLS#1:CLS#2:CLS#3:CLS#5 [98B8]
550 LOCATE #1,1,2 [7DA7]
560 PRINT #1,"Sie sind "; [3272]
570 PRINT #1,raum$(raum);:PRINT #1, "." [1FB6]
580 yi=1
590 FOR n=1 TO omax
600 IF objekt(n)<>raum THEN 640

```

Listing. Der Fall »Famit AG«

Spiele-Listing

610 PRINT #2, " ";CHR\$(144);" ";objekt\$(n)	[3556]	1200	[6F20]
620 oy(y1)=n	[D68C]	1180 IF y<=1 THEN y=1	[6024]
630 y1=y1+1	[4D4C]	1190 GOTO 1130	[9612]
640 NEXT	[74F2]	1200 IF INKEY(9)<>0 THEN 1140 ELSE PRINT	
650 PRINT#2, " ";CHR\$(144);" {CTRL X}{4 S		CHR\$(7);:o=oy(y):IF (vy=8 OR vy=11	
FACE)";CHR\$(241);" {4 SPACE}{CTRL X}"	[1E8A])AND oy(y)=0 THEN 1130 ELSE IF oy(y	
	[7CAC])=0 THEN LOCATE #2,2,y:PRINT #2,CHR	[D4DA]
660 y2=2		\$(144):GOTO 1210 ELSE LOCATE 1300	[B9CA]
670 PRINT#5, " ";CHR\$(144);" {CTRL X}{4 S	[4092]	1210 y=1:yy=1	
FACE)";CHR\$(240);" {4 SPACE}{CTRL X}"	[F172]	1220 LOCATE #5,2,yy:PRINT #5,CHR\$(144):L	[F852]
	[5E18]	OCATE #5,2,y:PRINT #5,CHR\$(143)	
680 FOR n=1 TO omx	[415C]	1230 FOR n=1 TO 200:NEXT:IF INKEY(2)=0 T	[963C]
690 IF objekt(n)<>-1 THEN 720	[92D4]	HEN yy=y:y=y+1 ELSE 1260	[8382]
700 PRINT #5, " ";CHR\$(144);" ";objekt\$(n	[6FF0]	1240 IF y>=y2-1 THEN y=y2-1	[870C]
)	[76FA]	1260 IF INKEY(0)=0 THEN yy=y:y=y-1 ELSE	
710 iy(y2)=n:y2=y2+1	[3882]	1290	[4B32]
720 NEXT	[C3A4]	1270 IF y<=1 THEN y=1	[4324]
730 PRINT #3,"Sie k'nnen nach ";gedruck	[AC0E]	1280 GOTO 1220	[AC12]
t=0	[9004]	1290 IF INKEY(9)<>0 THEN 1230 ELSE PRINT	
740 FOR x=1 TO 6	[5E0C]	CHR\$(7);:o=iy(y):IF (vy=6 AND y=1 T	
750 IF richtung(raum,x)=0 THEN 790 ELSE	[AFFE]	HEN 1220 ELSE IF y=1 THEN LOCATE #5	
gedruckt=-1	[B52E]	,2,y:PRINT #5,CHR\$(144):GOTO 1120 E	[D3C8]
760 IF POS(#3)=17 THEN PRINT #3.richtung	[FA42]	LSE 1300	
\$(x);:GOTO 790	[60C6]	1300 ON vy GOSUB 900,1500,1590,1630,1680	
770 IF POS(#3)+LEN(richtung\$(x))<48 THEN	[51F6]	,1710,1740,1830,1910,2080,2720,2770	[3B06]
PRINT #3, " ";richtung\$(x);:GOTO 79	[7382]	,2810	[57AC]
0	[4AF0]	1310 GOTO 530	
780 PRINT #3, " ";PRINT #3,richtung\$(x);:	[2E2E]	1320 PRINT"Bitte REC & PLAY dr}cken":PRI	
GOTO 790	[9FEC]	NT"Unter welchem Namen speichern":I	[D480]
790 NEXT	[ED42]	NPUR eingabe\$	
800 IF gedruckt=0 THEN PRINT #3,"nirgend	[2E34]	1330 IF LEN(eingabe\$)>16 THEN PRINT"Bitt	
wo";	[B674]	e etwas k}rzer !":GOTO 1320 ELSE ei	
810 PRINT #3, " gehen ."	[6A90]	ngabe\$="!"+eingabe\$:OPENOUT eingabe	[2F84]
820 IF raum=17 AND objekt(27)=17 AND bed	[BDA4]	\$	
ingung(12)=0 THEN PRINT"Ihr Chef{2 S	[6360]	1340 PRINT#9,raum:PRINT#9,zug:PRINT#9,fo	[EDBC]
PACE}gibt{2 SPACE}Ihnen{2 SPACE}den	[7386]	to	[52DC]
Auf-trag , sich n}her{2 SPACE}mit de	[67CE]	1350 FOR x=1 TO omx	[858E]
m FallFamit AG{2 SPACE}zu{2 SPACE}be	[1366]	1360 PRINT#9,objekt(x)	[1054]
fassen{2 SPACE}und ihndanach zu info	[B500]	1370 NEXT	
rmieren.";bedingung(12)=-1:objekt(11	[7386]	1380 FOR x=1 TO rmax:FOR y=1 TO 6:PRINT#	[A990]
7)=-1:FOR t=1 TO 7000:NEXT	[67CE]	9,richtung(x,y):NEXT:NEXT	
830 IF zug=4 AND raum=8 THEN PRINT"Das T	[1366]	1390 FOR x=1 TO bdmax:PRINT#9,bedingung([D61A]
elefon klingelt ."	[1366]	x):NEXT	[18C6]
840 IF zug<>62 THEN objekt(120)=0	[ED42]	1400 CLOSEOUT:PRINT ok\$:GOTO 530	
850 IF zug>zmax THEN MODE 2:PRINT:PRINT	[2E34]	1410 PRINT"Bitte Cassette zur}ckspulen u	
Sie k'nnen die Suche einstellen.Es i	[B674]	ndPLAY dr}cken":PRINT"Welches Spiel	
st zu sp{t !":LOCATE 22,10:PRINT"Meh	[6A90]	laden !":INPUT eingabe\$	[725E]
r Erfolg beim n}chsten Mal !":END	[BDA4]	1420 IF LEN(eingabe\$)>16 THEN PRINT"Das	
860 IF raum=42 OR raum=39 THEN PRINT"Vor	[6360]	kann nicht sein !":GOTO 1410 ELSE e	
und hinter{2 SPACE}Ihnen liegt je-w	[7386]	ingabe\$="!"+eingabe\$:OPENIN eingabe	[797E]
eils eine Fabrikhalle."	[67CE]	\$	
870 IF raum=43 THEN PRINT"Vor Ihnen lieg	[1366]	1430 INPUT#9,raum:INPUT#9,zug:INPUT#9,fo	[27CE]
t ein Geb}ude."	[1366]	to	
880 IF raum=47 THEN PRINT"Die T}r f}llt	[B500]	1440 FOR x=1 TO omx:INPUT#9,objekt(x):N	[2002]
zu.Sie h'ren, da"ein Schl}ssel im Sc	[ED42]	EXT	
hlo" gedrehtwird.":GOTO 530	[2E34]	1450 FOR x=1 TO rmax:FOR y=1 TO 6:INPUT#	[7592]
890 PRINT:PRINT"Was m'chten Sie tun ?"	[B674]	9,richtung(x,y):NEXT:NEXT	
900 y=1:yy=1	[6A90]	1460 FOR x=1 TO bdmax:INPUT#9,bedingung([E21C]
910 LOCATE #4,2,yy:PRINT #4,CHR\$(144):LO	[BDA4]	x):NEXT	[9912]
CATE #4,2,y:PRINT #4,CHR\$(143)	[6360]	1470 CLOSEIN:PRINT ok\$:GOTO 530	
920 FOR n=1 TO 200:NEXT:IF INKEY(2)=0 TH	[7386]	1480 MODE 2:LOCATE 25,5:PRINT"Mehr Erfol	[9456]
EN yy=y:y=y+1 ELSE 950	[67CE]	g beim n}chsten Mal !":END	[5B5A]
930 IF y>=16 THEN y=16	[1366]	1490 NEXT	
940 GOTO 910	[1366]	1500 IF o=9 AND zug=4 THEN PRINT"Ihr{2 S	
950 IF INKEY(0)=0 THEN yy=y:y=y-1 ELSE 9	[B500]	PACE}Chef ist{2 SPACE}am Telefon .	
80	[7386]	Siesollen sofort zu ihm kommen.";ob	[EC6C]
960 IF y<=1 THEN y=1	[67CE]	jekt(27)=17:GOTO 530	
970 GOTO 910	[1366]	1510 IF o=57 AND bedingung(2)=0 THEN PRI	
980 IF INKEY(9)=0 THEN PRINT CHR\$(7);:vy	[B500]	NT"Eine rote Lampe leuchtet auf .":	[61D8]
=y:LOCATE #4,2,y:PRINT#4,CHR\$(144):G	[7386]	bedingung(2)=-1:GOTO 530	
OTO 990 ELSE 920	[67CE]	1520 IF o=57 AND bedingung(2)=-1 THEN PR	
990 FOR n=1 TO 300:NEXT:ON vy GOTO 1000,	[1366]	INT"Die Lampe geht aus .":bedingun	[7EF6]
1120,1120,1120,1680,1210,1120,1120,1	[93D0]	(2)=0:GOTO 530	
120,1120,1120,2770,1120,1410,1320,14	[348A]	1530 IF o=61 AND bedingung(5) AND beding	[895A]
80	[08EA]	ung(2) THEN 1540 ELSE 1550	
1000 CLS#2:CLS#5:nn=0:FOR n=1 TO 6	[D744]	1540 PRINT"Der Computer arbeitet.":CALL	
1010 IF richtung(raum,n)<>0 THEN PRINT#2	[9FCA]	&BB06:PRINT"Auf dem Bildschirm ersc	
, " ";CHR\$(144);" ";richtung\$(n):nn=	[B53A]	heint.":PRINT"Ausgaben 1984{2 SPA	
nn+1:ri(nn)=richtung(raum,n)	[A23C]	}: 15.245.765 DM":PRINT"Einnahmen 1	
1020 NEXT	[2ECE]	984 : 19.639.407 DM":PRINT"gesamt{9	
1030 y=1:yy=1	[3F0C]	SPACE}: +2.136.012 DM":FOR n=1 TO	
1040 LOCATE #2,2,yy:PRINT #2,CHR\$(144):L	[AB20]	4000:NEXT:bedingung(8)=-1:GOTO 530	[A8DE]
OCATE #2,2,y:PRINT #2,CHR\$(143)	[3124]	1550 IF o=61 AND bedingung(5)=0 AND bedi	
1050 FOR n=1 TO 200:NEXT:IF INKEY(2)=0 T	[5200]	ngung(2)=-1 THEN PRINT"Es passiert	[AEDE]
HEN vv=v:y=y+1 ELSE 1080	[E826]	nichts .":GOTO 530	
1060 IF y>=nn THEN y=nn	[58DE]	1560 IF o=66 AND objekt(o)=55 THEN PRINT	
1070 GOTO 1040	[443A]	"Das Band beginnt zu laufen .":CALL	
1080 IF INKEY(0)=0 THEN yy=y:y=y-1 ELSE	[E43C]	&BB06:PRINT"Hallo Max , kommst Du	
1110	[3106]	morgen zuuns ? Benachrichtige{2 SPA	
1090 IF y<=1 THEN y=1	[A50C]	CE}uns{2 SPACE}dochbitte{2 SPACE}un	
1100 GOTO 1040		d bring{2 SPACE}das{2 SPACE}Geschen	
1110 IF INKEY(9)=0 THEN PRINT CHR\$(7);:r		kmit! Bei uns ist alles beim AI-tem	
aum=ri(y):PRINT ok\$:GOTO 530:ELSE 1		.":FOR n=1 TO 4000:NEXT:GOTO 530	[08E2]
050		1570 IF o=9 OR o=88 OR o=110 OR o=113 OR	
1120 y=1:yy=1:oy(y1)=0		o=56 THEN PRINT bemerkung\$(6):GOTO	[D4D2]
1130 LOCATE #2,2,yy:PRINT #2,CHR\$(144):L		530	[855E]
OCATE #2,2,y:PRINT #2,CHR\$(143)		1580 PRINT bemerkung\$(4):GOTO 530	
1140 FOR n=1 TO 200:NEXT:IF INKEY(2)=0 T		1590 IF o=1 AND objekt(1)=18 AND objekt(
HEN yy=y:y=y+1 ELSE 1170		13)=-1 AND bedingung(4)=-1 THEN rau	
1150 IF y>=y1 THEN y=y1			
1160 GOTO 1130			
1170 IF INKEY(0)=0 THEN yy=y:y=y-1 ELSE			

Listing. Der Fall »Famit AG« (Fortsetzung)

m=21:objekt(13)=0:PRINT ok\$:GOTO 530	[0258]	TO 530	[085E]
1600 IF o=41 OR o=12 THEN PRINT bemerkung\$(3):GOTO 530	[AB8C]	1930 IF o=7 OR o=22 THEN PRINT"Die T}ren sind schon offen.":GOTO 530	[7512]
1610 IF o=46 AND raum=34 AND objekt(122)=34 THEN PRINT"Sie(2 SPACE)haben das(2 SPACE)Schlo" (2 SPACE)der T}r abgeschraubt und k}nnen die T}rs o}ffnen. Vor Ihnen liegt ein Garten.":richtung(34,1)=35:richtung(35,2)=34:GOTO 530	[5948]	1940 IF o=58 AND richtung(46,3)=0 THEN PRINT"Sie haben die T}r ge}ffnet.":richtung(46,3)=33:GOTO 530:ELSE IF o=58 AND richtung(46,3)=33 THEN PRINT"Die T}r ist schon offen.":GOTO 530	[6342]
1620 PRINT bemerkung\$(4):GOTO 530	[BE54]	1950 IF o=59 THEN PRINT"Die T}r(2 SPACE)ist verschlossen. Sie haben keinen(2 SPACE)passenden Schl}ssel. Sie l} (2 SPACE)sich allerdings von innen }ffnen.":GOTO 530	[3368]
1630 IF foto<=0 THEN PRINT"Der Film ist voll !":GOTO 530	[8B6C]	1960 IF o=76 AND objekt(107)=-1 AND bedingung(9)=0 THEN PRINT"Sie haben den Koffer ge}ffnet.":bedingung(9)=-1:GOTO 530:ELSE IF o=76 AND objekt(107)=-1 AND bedingung(9)=-1 THEN PRINT"Der Koffer ist schon offen.":GOTO 530	[187C]
1640 IF objekt(21)<>-1 THEN PRINT bemerkung\$(3):GOTO 530	[15A8]	1970 IF o=24 OR o=102 THEN PRINT"Die Schublade ist schon offen.":GOTO 530	[E942]
1650 IF o=82 AND objekt(21)=-1 THEN bedingung(6)=-1:foto=foto-1:PRINT ok\$:GOTO 530	[2C56]	1980 IF o=106 AND bedingung(3)=0 AND objekt(103)=-1 THEN bedingung(3)=-1:PRINT ok\$:GOTO 530	[CAFC]
1660 IF o=93 AND objekt(21)=-1 THEN bedingung(7)=-1:foto=foto-1:PRINT ok\$:GOTO 530	[015E]	1990 IF o=106 AND bedingung(3)=-1 THEN PRINT"Die Schublade ist schon offen.":GOTO 530	[6032]
1670 foto=foto-1:PRINT ok\$:GOTO 530	[BA12]	2000 IF o=35 AND raum=43 AND richtung(43,1)=0 THEN richtung(43,1)=74:PRINT ok\$:GOTO 530	[5554]
1680 IF (raum=36 OR raum=61 OR raum=64 OR raum=63 OR raum=65 OR raum=66 OR raum=67) AND ((zug>200 AND zug<220) OR (zug>250 AND zug<270) OR (zug>300 AND zug<320)) THEN PRINT"Sie h}ren da} jemand(2 SPACE)im Haus ist.":GOTO 530	[9230]	2010 IF o=35 AND raum=43 AND richtung(43,1)=74 THEN PRINT"Die T}r ist schon offen.":GOTO 530	[369A]
1690 IF raum=47 THEN PRINT"Sie h}ren ein h}misches Lachen.":GOTO 530	[2C32]	2020 IF o=119 THEN PRINT"Sie(2 SPACE)k}nnen(2 SPACE)die Schublade(2 SPACE)zurzeit nicht }ffnen.":GOTO 530	[1C16]
1700 PRINT"Sie h}ren nichts besonderes.":GOTO 530	[75A6]	2030 IF o=82 THEN INPUT"Geben Sie die Zahlenkombination ein.":eingabekomb\$:IF eingabekomb\$=safekomb\$ THEN PRINT"Es(2 SPACE)war die(2 SPACE)richtige Kombination. Der Safe ist jetzt offen.":bedingung(13)=-1:GOTO 530:ELSE PRINT"Die Kombination ist falsch.":GOTO 530	[8D20]
1710 IF o=77 AND objekt(77)=-1 AND raum=53 THEN objekt(77)=53:bedingung(5)=-1:PRINT ok\$:GOTO 530	[D6F6]	2040 IF o=118 AND richtung(60,3)=0 THEN PRINT"Sie haben das Fenster ge}ffnet.":richtung(60,3)=57:richtung(57,4)=60:GOTO 530	[492C]
1720 IF objekt(o)=-1 THEN PRINT ok\$:objekt(o)=raum:GOTO 530	[7B46]	2050 IF o=118 AND richtung(60,3)=57 THEN PRINT"Das Fenster ist ge}ffnet.":GOTO 530	[1052]
1730 PRINT bemerkung\$(4):GOTO 530	[D958]	2060 IF o=116 AND objekt(35)=-1 AND richtung(43,1)=0 THEN PRINT"Sie haben die T}r ge}ffnet.":richtung(43,1)=74:GOTO 530:ELSE IF o=116 AND richtung(43,1)=74 THEN PRINT"Die T}r ist schon offen.":GOTO 530	[B9A4]
1740 IF o=5 THEN PRINT"Auf dem Zettel steht.":bueronummer\$:GOTO 530	[D746]	2070 PRINT bemerkung\$(3):GOTO 530	[7852]
1750 IF o=18 OR o=74 OR o=83 THEN PRINT bemerkung\$(1):GOTO 530	[929C]	2080 IF o=1 THEN PRINT"Es ist meiner.":GOTO 530	[7BAE]
1760 IF o=33 THEN PRINT"Famit AG Elektromaschinen":PRINT" (4 SPACE)46334":GOTO 530	[DADC]	2090 IF o=2 OR o=49 THEN PRINT"Sie f}hrt nach oben.":GOTO 530	[C5F6]
1770 IF o=36 THEN PRINT"Die deutsche Wirtschaft(5): Il-legale Gesch}fte in Elektronik-firma ?":PRINT"Dallas wird noch besser ! Jetzt tauch mit Sinatra ?":PRINT"Gluck }r 83-J(hrige : 2.3 Mil-lion DM Lottogewinn !":FOR n=1 TO 3000:NEXT:GOTO 530	[BD54]	2100 IF o=4 AND objekt(5)=0 THEN PRINT"Ich sehe einen Zettel.":objekt(5)=4:GOTO 530	[D0C0]
1780 IF o=25 THEN PRINT"Die(2 SPACE)Famit AG, Industriestra}e 8-12, hat(2 SPACE)einige(2 SPACE)merkw}rdige Gesch}fte abgeschlossen. Die bisherigen Ermittlungen haben noch zu keinem Ergebnis gef}hrt.":FOR n=1 TO 4000:NEXT:GOTO 530	[0E4A]	2110 IF o=5 THEN PRINT"Darauf steht eine Nummer.":GOTO 530	[F8FA]
1790 IF o=68 THEN PRINT"Lieber Schatz !":PRINT"Ich(2 SPACE)hoffe(2 SPACE)Du riskierst(2 SPACE)nicht zuviel. Hoffentlich(2 SPACE)kommst Du bald nach h}r.":PRINT"ich liebe Dich !":PRINT"(2 SPACE)Deine Angela":GOTO 530	[8FE6]	2120 IF o=8 AND objekt(10)=0 AND objekt(11)=0 THEN PRINT"Auf dem(2 SPACE)Schreibtisch liegt ein Dienstausweis(2 SPACE)und eine Dienstpistole.":objekt(10)=8:objekt(11)=8:GOTO 530	[EA6E]
1800 IF o=64 THEN PRINT"Das Gesch}fte bei r 542,300 DM abgeschlossen am 23.8.85 ist durchgef}hrt worden. Die strikte Geheimhaltung ist gew}hrleistet.":GOTO 530	[7C58]	2130 IF o=10 THEN PRINT"Es ist ihr Dienstausweis.":GOTO 530	[839C]
1810 IF o=95 THEN PRINT"Es sind(2 SPACE)die Frachtpapiere(2 SPACE)in 15 Kisten R}hrgen(2 SPACE)nach Belgien(2 SPACE)und(2 SPACE)eine kleinere(2 SPACE)Kistenach Mayaguana.":GOTO 530	[F190]	2140 IF o=11 THEN PRINT"Es ist ihre Dienstpistole.":GOTO 530	[DF68]
1820 PRINT bemerkung\$(4):GOTO 530	[4C58]	2150 IF o=12 AND objekt(13)=0 THEN PRINT"Auf dem Schreibtisch(2 SPACE)liegt ein Schl}ssel.":objekt(13)=9:GOTO 530	[E5A6]
1830 inv=0	[B24C]	2160 IF o=13 THEN PRINT"Sie(2 SPACE)kennen(2 SPACE)die(2 SPACE)Funktion(2 SPACE)des Schl}ssels nicht.":GOTO 530	[0A08]
1840 FOR x=1 TO omx:IF objekt(x)=-1 THEN inv=inv+1	[30C0]	2170 IF o=14 THEN PRINT"Es ist (zug\60+10);". (zug MOD 60);"Uhr":GOTO 530	[33A0]
1850 NEXT	[5F5A]	2180 IF o=15 OR o=60 THEN PRINT"Sie f}hrt nach unten.":GOTO 530	[B65C]
1860 IF inv=4 THEN PRINT"ich kann nicht mehr tragen.":GOTO 530	[96A4]	2190 IF o=17 AND objekt(18)=0 THEN PRINT"Auf dem Schreibtisch liegen Akten.":objekt(18)=11:GOTO 530	[34A8]
1870 IF o=77 AND objekt(o)<>-1 THEN objekt(77)=-1:PRINT ok\$:GOTO 530	[0D76]	2200 IF o=20 AND objekt(21)=0 THEN PRINT"Auf dem Schreibtisch(2 SPACE)liegt ein Fotoapparat.":objekt(21)=13:GOTO 530	[0C34]
1880 IF (o=10 OR o=11 OR o=13 OR o=18 OR o=21 OR o=25 OR o=34 OR o=35 OR o=36 OR o=43 OR o=44 OR o=45 OR o=46 OR o=52 OR o=51 OR o=55 OR o=64 OR o=68 OR o=74 OR o=76 OR o=81) AND objekt(o)<>-1 AND objekt(o)<>0 THEN objekt(o)=-1:PRINT ok\$:GOTO 530	[F348]	2210 IF o=21 THEN PRINT"Es sind noch":foto="Bilder drauf.":GOTO 530	[3300]
1890 IF (o=5 OR o=84 OR o=95 OR o=103 OR o=107 OR o=117) AND objekt(o)<>-1 AND objekt(o)<>0 THEN objekt(o)=-1:PRINT ok\$:GOTO 530	[E064]	2220 IF o=23 AND objekt(24)=0 THEN PRINT	
1900 PRINT bemerkung\$(2):GOTO 530	[1952]		
1910 IF o=1 AND objekt(13)=-1 THEN bedingung(4)=-1:PRINT ok\$:GOTO 530	[1556]		
1920 IF o=3 OR o=6 OR o=16 OR o=19 THEN PRINT"Die T}r ist schon offen.":GOTO 530			

Listing. Der Fall »Famit AG« (Fortsetzung)

2230	"Sie sehen eine Schublade.":objekt(24)=16:GOTO 530	[EA92]	2530	Geld.":objekt(93)=72:GOTO 530	[83E6]
	IF o=24 AND objekt(25)=0 THEN PRINT			IF o=93 THEN PRINT"Es(2 SPACE)sind(2 SPACE)ungef(2 SPACE)10(2 SPACE)MillionMark.Das die einer einfach	
	"Sie sehen eine Akte.":objekt(25)=16:GOTO 530	[F68E]		ste-hen l("t ?!":GOTO 530	[1B64]
2240	IF o=25 THEN PRINT"Sie geh'rt zum F		2540	IF o=94 THEN PRINT"Sie sehen Fracht	
	all 'Famit AG'.":GOTO 530	[6340]		papiere.":objekt(95)=73:GOTO 530	[8D3A]
2250	IF o=27 THEN PRINT"Sie(2 SPACE)spie		2550	IF o=97 THEN PRINT"Der Tote scheint	
	len(2 SPACE)mit ihrer Entlas-sung.":	[5F98]		der Gesch(fts-f)hrer der(2 SPACE)F	
	GOTO 530			irma sein . Er isterw)rgt worden ,	
2260	IF o=31 THEN PRINT"Die Telefonzelle			vermutlich mit einem Handtuch.Keine	
	(2 SPACE)ist nicht be-sch)digt !!!"	[9300]		sch'ne Artzu sterben ! Er(2 SPACE)i	
	objekt(32)=25:GOTO 530		2560	PRINT"warm ,also ist er erst vor ku	[C60A]
2270	IF o=34 AND (objekt(35)=0 OR objekt			r-zer Zeit get'tet worden. ObwohlSi	
	(35)=28) THEN PRINT"Daran h)ngt ein	[4528]		e(2 SPACE)nicht(2 SPACE)zur(2 SPACE)	
	Schl)ssel.":objekt(35)=28:GOTO 530			Mordkommissiongeh'ren , k'nnen(2 S	
	GOTO 530	[80B8]		PAGE)Sie versuchenauch den M'rder z	
2280	IF o=34 AND objekt(35)<>28 THEN PRI			u finden, den(2 SPACE)ist unter	
	NT"Daran hing ein Schl)ssel.":GOTO	[0C10]		(2 SPACE)Umst)nden(2 SPACE)hiernoch	[0A52]
	530		2570	IF o=69 THEN PRINT"{(2 SPACE)W}rg !	[7DDE]
2290	IF o=36 THEN PRINT"Es ist die Ausga			!":GOTO 530	
	be vom Samstag,dem 23. November 198	[0C10]	2580	IF o=101 THEN PRINT"Sie sehen eine	
	5.":GOTO 530			Schublade.":objekt(102)=80:GOTO 530	[C1EE]
2300	IF o=41 AND objekt(42)=0 THEN PRINT	[ACFC]			
	"Unter einem LKW liegt Werkzeug.":o		2590	IF o=102 AND objekt(103)=0 THEN PRI	
	bjekt(42)=37:GOTO 530	[3D6E]		NT"Sie(2 SPACE)sehen(2 SPACE)einen(2	[744E]
2310	IF o=42 THEN objekt(42)=0:objekt(43			SPACE)Sicherheits-schl)ssel.":obj	
)=37:objekt(44)=37:objekt(45)=37:ob	[6B70]	2600	IF o=104 AND objekt(119)=0 THEN PRI	[D9C2]
	jekt(46)=37:PRINT bemerkung\$(1): G	[5E3C]		NT"Im Schreibtisch ist eine Schub-	
	OTO 530			lade.":objekt(119)=81:GOTO 530	[8CC8]
2320	IF o=49 THEN PRINT"Sie f)hrt nach o	[F8CA]	2610	IF o=119 THEN PRINT"Die Schublade i	
	ben.":GOTO 530			st verschlossen.":GOTO 530	[E6F8]
2330	IF o=51 THEN PRINT"Es sind alte Akt	[2B76]	2620	IF o=105 THEN PRINT"Sie sehen eine	
	en und die Per-sonalliste.":objekt([20BC]		Schublade.":objekt(106)=82:GOTO 530	
	51)=0:objekt(52)=48:objekt(53)=48:G	[3590]	2630	IF o=106 AND objekt(107)=0 AND bedi	[A964]
	OTO 530			ngung(3)=-1 THEN PRINT"Sie(2 SPACE)	[ED34]
2340	IF o=56 AND objekt(57)=0 THEN PRINT	[8538]		sehen einen Spezialschl)ssel.":obj	
	"Ich sehe einen Drehknopf.":objekt([B964]	2640	IF o=106 AND bedingung(3)=0 THEN PR	[DBD6]
	57)=50:GOTO 530			INT"Die Schublade ist verschlossen.	[B9DC]
2350	IF o=57 AND bedingung(2)=0 THEN PRI	[7D00]		":GOTO 530	
	NT"Der Drehknopf steht auf Null.":G	[3B56]	2650	IF o=110 OR o=113 THEN PRINT"Alle M	
	OTO 530			aschinen(2 SPACE)sind abgeschal-tet	
2360	IF o=57 AND bedingung(2)=-1 THEN PR	[C2C2]	2660	IF o=114 AND objekt(115)=0 THEN PRI	[4844]
	INT"Der Drehknopf steht auf Eins.":	[8538]		NT"Sie sehen ein Loch im Zaun.":obj	
	GOTO 530			ekt(115)=93:richtung(93,1)=44:richt	
2370	IF o=61 AND objekt(62)=0 THEN PRINT	[C2C2]	2670	IF o=120 THEN PRINT"Der Mann(2 SPAC	
	"Neben dem(2 SPACE)Computer stehen	[8538]		E)hat Sie k.o. geschla-gen.Sie sind	
	Dis-kettenlaufwerke.":objekt(62)=53	[B964]		erst nach 20 Minu-ten wieder zu Be	
	:GOTO 530			wu'tsein gekom-men. Der Kinnhaken k	
2380	IF o=63 AND objekt(64)=0 THEN PRINT	[22E8]	2680	IF o=121 THEN PRINT"Im Holzraum ist	
	"Auf dem Schreibtisch(2 SPACE)liegt	[9EA0]		eine versteckteT)r eingebaut.":obj	
	einGesch(ftsbrief.":objekt(64)=54:	[8D2E]		ekt(122)=34:GOTO 530	[4328]
	GOTO 530		2690	IF o=121 AND bedingung(11)=-1 THEN	
2390	IF o=65 AND objekt(66)=0 THEN PRINT	[5A3C]		PRINT"Die T)r(2 SPACE)ist offen . V	
	"Auf dem(2 SPACE)Schreibtisch steht	[7D7A]		or(2 SPACE)Innenliegt ein Garten.":	
	einAnrufbeantworter.":objekt(66)=5	[78D8]		richtung(34,1)=35:GOTO 530:ELSE IF	[053A]
	5:GOTO 530			o=121 THEN PRINT"Die(2 SPACE)verste	
2400	IF o=67 AND objekt(68)=0 THEN PRINT	[C86E]	2700	IF o=123 THEN PRINT"Die(2 SPACE)T)r	
	"Auf dem(2 SPACE)Schreibtisch liegt	[A662]		(2 SPACE)ist leider(2 SPACE)von au	
	einBrief.":objekt(68)=56:GOTO 530			enverschlossen worden.":GOTO 530	[DBAA]
2410	IF o=73 AND objekt(74)=0 THEN PRINT		2710	PRINT bemerkung\$(1):GOTO 530	[CD50]
	"Im Schrank stehen B)cher.":objekt(2720	IF o=120 AND objekt(10)=-1 AND obje	
	74)=61:GOTO 530			kt(64)=-1 AND bedingung(6)=-1 AND b	
2420	IF o=75 AND objekt(76)=0 THEN PRINT			edingung(7)=-1 AND bedingung(8)=-1	
	"Unter dem(2 SPACE)Bett liegt ein H			AND objekt(77)=-1 THEN RESTORE 5250	
	art-schalenkoffer.":objekt(76)=62:G		2730	:FOR m=1 TO cmax:READ n:PRINT CHR\$([460E]
	OTO 530			n):NEXT:bedingung(10)=-1:FOR n=1 T	
2430	IF o=75 AND bedingung(10)=0 AND obj			O 15000:NEXT:CALL &BB06:GOTO 530	
	ekt(76)<>0 AND ((zug>200 AND zug<22		2740	IF o=27 AND objekt(10)=-1 THEN PRIN	
	0)OR(zug>250 AND zug<270)OR(zug>300			T"Sie sind entlassen.":FOR n=1 TO 5	[A0EE]
	AND zug<320)) THEN PRINT"Neben dem			000:NEXT:GOTO 1480	
	Bett steht ein Mann,der(2 SPACE)o		2750	IF o=97 AND objekt(10)<>-1 THEN PRI	
	ffensichtlich(2 SPACE)durch ihreGeg			NT"Wie(2 SPACE)sind Sie(2 SPACE)blo	
	enwart sehr erschrocken ist.":objek			~ zur Polizeigekommen!Die Leiche(!)	[4EEA]
	t(120)=62:GOTO 530			ist tot!":GOTO 530	
2440	IF o=76 AND bedingung(9)=-1 AND obj		2760	PRINT bemerkung\$(4):GOTO 530	[1E2C]
	ekt(77)=0 THEN PRINT"Im Koffer befi		2770	IF o<>32 THEN PRINT"Sie k'nnen(2 SP	[FD60]
	nden sich Disket-ten.":objekt(77)=-			ACE)jetzt(2 SPACE)nicht(2 SPACE)tel	
	-1:GOTO 530			e-fonieren.":GOTO 530:ELSE IF o=32	
2450	IF o=76 AND bedingung(9)=0 THEN PRI			AND objekt(117)<>-1 THEN PRINT"Sie	[018E]
	NT"Der Koffer ist verschlossen.":G		2780	INPUT"Welche Telefonnummer ":nummer	
	OTO 530				
2460	IF o=77 AND objekt(77)<>0 THEN PRIN				
	T"Auf den Disketten steht ":PRINT"				
	'Buchhaltung' .":GOTO 530				
2470	IF o=81 AND objekt(81)=-1 AND raum=				
	64 THEN PRINT"Da wo(2 SPACE)der Spi				
	egel hing , sehensie einen Safe.":o				
	bjekt(82)=64:GOTO 530				
2480	IF o=82 AND bedingung(13)=0 THEN PR				
	INT"Der Safe ist verschlossen.":G				
	OTO 530				
2490	IF o=82 AND bedingung(13)=-1 AND (o				
	bjekt(83)=0 OR objekt(83)=64) THEN				
	PRINT"Im Safe liegen Unterlagen.":o				
	bjekt(83)=64:GOTO 530				
2500	IF o=83 AND objekt(83)<>0 THEN PRIN				
	T"Es sind Vertr(ge(2 SPACE))ber ill				
	egaleGesch(fte.":GOTO 530				
2510	IF o=89 OR o=90 OR o=91 THEN PRINT"				
	Es sind 5 Kisten. In ihnen bef-inde				
	n sich Elektroger(te.":GOTO 530				
2520	IF o=92 THEN PRINT"Darin liegt viel				

Listing. Der Fall »Famit AG« (Fortsetzung)

<pre> \$:IF nummer\$=bueronummer\$ THEN 2790 ELSE PRINT"Beim n(chsten Ton ist e s";10+(zug\60);":":(zug MOD 60);"Uh r .:sound 1,478,100,7:print" Piiii epppp!!!":GOTO 430 2790 IF bedingung(10)=-1 THEN MODE 2:PRI NT:PRINT"{7 SPACE}Herzlichen Gl}ckw unsch !!!":PRINT:PRINT" Sie haben i hre Aufgabe erf}llt und ihren Chef benachrichtigt.":PRINT:END 2800 IF bedingung(10)=0 THEN PRINT"Ihr C hef{2 SPACE}sagt Ihnen durchs Te-le fon,er wolle Resultate sehen.Wenn n icht bald etwas geschehe,werde er d af}r sorgen , da~ Ih-nen etwas gesc hehe.":FOR n=1 TO 3000:NEXT:GOTO 53 0 2810 IF o=27 THEN PRINT"Sie sind von zwe i ihrem Chef zuHilfe kommanden{2 SP ACE}Polizisten er-schossen worden." :FOR n=1 TO 4000:NEXT:GOTO 1480 2820 IF o=40 AND objekt(45)=-1 THEN PRIN T"Sie haben das Fenster zerschla-ge n, und Sie k'nnen jetzt durchdas Fe nster klettern.":richtung(36,4)=61: richtung(61,3)=36:GOTO 530 2830 IF o=58 OR o=59 OR o=118 THEN PRINT "Sie haben{2 SPACE}die Alarmanlage aus-gel'st.Damit k'nnen Sie die Su- che einstellen.":FOR n=1 TO 5000:NE XT:GOTO 1480 2840 IF objekt(45)=-1 THEN PRINT"Gewalt wird Ihnen nicht helfen.":GOTO 530 2850 IF objekt(45)<>-1 THEN PRINT bemerk ung\$(2):GOTO 530 2860 SYMBOL AFTER 32 2870 SYMBOL 123,198,0,120,12,124,204,118 2880 SYMBOL 96,102,0,60,102,102,102,60 2890 SYMBOL 92,102,60,102,102,102,102,60 2900 SYMBOL 91,102,60,102,102,126,102,10 2 2910 SYMBOL 125,102,0,102,102,102,102,62 2920 SYMBOL 93,36,102,102,102,102,102,60 2930 SYMBOL 126,28,35,99,108,99,99,99,10 2 2940 SYMBOL 144,255,195,195,195,195,195, 195,255 2950 READ p1,p2,p3,p4:IF p1=-999 THEN 30 50 2960 PLOT p1,p2:DRAW p3,p4:GOTO 2950 2970 DATA 170,270,180,270,215,270,225,27 0,175,270,195,340,205,340,220,270,1 76,270,196,340,206,340,221,270,195, 340,205,340,184,295,214,295,240,270 ,250,270,240,340,250,340,280,270 2980 DATA 290,270,280,340,290,340,245,27 0,245,340,246,270,246,340,245,305,2 85,270,245,305,285,340,310,340,365, 340,310,340,310,335,365,340,365,335 ,337,340,337,270 2990 DATA 338,340,338,270,332,270,342,27 0,311,340,311,335,366,340,366,335,3 85,270,440,270,440,270,440,280,441, 270,441,280,390,270,390,340,391,340 ,391,270,385,340,440,340 3000 DATA 440,340,440,330,441,340,441,33 0,390,305,420,305,420,299,420,310,4 21,310,421,299,100,160,110,160,105, 160,105,210,106,160,106,210,100,210 ,140,210,140,210,140,205,141,210 3010 DATA 141,205,105,185,125,185,125,18 0,125,189,126,180,126,189,160,160,1 70,160,190,160,200,160,165,160,175, 210,185,210,195,160,166,160,176,210 ,186,210,196,160,176,210,186,210 3020 DATA 170,178,190,178,220,160,230,16 0,260,160,270,160,225,160,225,210,2 26,160,226,210,265,160,265,210,266, 160,266,210,245,160,265,210,245,160 ,225,210,226,210,246,160,266,210,24 6,160 3030 DATA 290,160,300,160,290,210,300,21 0,295,160,295,210,296,160,296,210,3 20,210,360,210,320,210,320,205,321, 210,321,205,360,210,360,205,361,210 ,361,205,340,210,340,160,341,160 3040 DATA 341,210,335,160,345,160,400,16 0,410,160,430,160,440,160,405,160,4 15,210,425,210,435,160,406,160,416, 210,426,210,436,160,408,178,431,178 ,416,210,426,210,497,210,490,210,-9 99,0,0,0 3050 FOR a=90 TO 180:DEG:PLOT 490+20*COS (a),190+20*SIN(a):NEXT:DRAWR 0,-10: DRAWR 1,0:DRAWR 0,10:FOR a=180 TO 3 60:DEG:PLOT 490+20*COS(a),180+20*SI N(a):NEXT:DRAW 510,185:PLOT 511,180 :DRAW 511,185:DRAW 500,185:INK 1,26 :RETURN </pre>	<pre> 3060 DATA gehe nach,bediene,benutze,foto grafiere,h're,leg,lies,nimm,'ffne,u ntersuche,verhaftete,telefoniere,zers chlage,lade,speicher,ende 3070 DATA einen Pkw,18 3080 DATA eine Treppe,2 3090 DATA eine T}r,3 3100 DATA einen Schreibtisch,4 3110 DATA einen Zettel,0 3120 DATA eine T}r,5 3130 DATA T}ren,7 3140 DATA einen Schreibtisch,8 3150 DATA ein Telefon,8 3160 DATA einen Dienstausweis,0 3170 DATA eine Pistole,0 3180 DATA einen Schreibtisch,9 3190 DATA einen Schl}ssel,0 3200 DATA eine Uhr,6 3210 DATA eine Treppe,10 3220 DATA eine T}r,10 3230 DATA einen Schreibtisch,11 3240 DATA Akten,0 3250 DATA eine T}r,12 3260 DATA einen Schreibtisch,13 3270 DATA einen Fotoapparat,0 3280 DATA T}ren,15 3290 DATA einen Schreibtisch,16 3300 DATA eine Schublade,0 3310 DATA eine Akte,0 3320 DATA einen Schreibtisch,17 3330 DATA ihren Chef,0 3340 DATA Autos,19 3350 DATA Geschirr,20 3360 DATA eine Stra~enlampe,23 3370 DATA eine Telefonzelle,25 3380 DATA ein Telefon,0 3390 DATA ein Firmenschild,27 3400 DATA eine Schl}sselleiste,28 3410 DATA einen T}rschl}ssel,0 3420 DATA eine Zeitung,28 3430 DATA einen Zaun,29 3440 DATA einen Zaun,31 3450 DATA Blumen,35 3460 DATA ein Fenster,36 3470 DATA Lastwagen,37 3480 DATA Werkzeug,0 3490 DATA Schraubenschl}ssel,0 3500 DATA Zangen,0 3510 DATA einen Hammer,0 3520 DATA Schraubenzieher,0 3530 DATA Maschinen,40 3540 DATA Maschinen,41 3550 DATA eine Treppe,46 3560 DATA einen Schreibtisch,47 3570 DATA Akten,48 3580 DATA die Personalliste,0 3590 DATA alte Akten,0 3600 DATA einen Schreibtisch,49 3610 DATA einen Stuhl,49 3620 DATA eine Schalltafel,50 3630 DATA einen Drehknopf,0 3640 DATA eine Glast}r,46 3650 DATA eine Glast}r,33 3660 DATA eine Treppe,52 3670 DATA einen Computer,53 3680 DATA Diskettenlaufwerke,0 3690 DATA einen Schreibtisch,54 3700 DATA einen Gesch}ftsbrief,0 3710 DATA einen Schreibtisch,55 3720 DATA einen Anrufbeantworter,0 3730 DATA einen Schreibtisch,56 3740 DATA einen Brief,0 3750 DATA eine Toilette,57 3760 DATA ein Geb}sch,58 3770 DATA Sessel,61 3780 DATA einen Tisch,61 3790 DATA einen Schrank,61 3800 DATA B}cher,0 3810 DATA ein Bett,62 3820 DATA einen Koffer,0 3830 DATA B Zoll-Disketten,0 3840 DATA eine Dusche,63 3850 DATA ein Waschbecken,63 3860 DATA die Garderobe,64 3870 DATA einen Spiegel,64 3880 DATA einen Safe,0 3890 DATA Unterlagen,0 3900 DATA die Haust}r,65 3910 DATA K}chenger}te,65 3920 DATA einen Tisch,67 3930 DATA St}hle,67 3940 DATA einen Gabelstapler,68 3950 DATA Kisten,69 3960 DATA Kisten,70 3970 DATA Kisten,71 3980 DATA eine Kiste,72 3990 DATA viel Geld,0 4000 DATA einen Schreibtisch,73 4010 DATA Frachtpapiere,0 4020 DATA Waschbecken,77 </pre>	<pre> [25BC] [E734] [3A68] [1812] [CD10] [3344] [4F0A] [B334] [2120] [DB60] [DD2E] [2144] [9F2A] [A6E0] [41E2] [CD8C] [6964] [3F74] [05E4] [B36E] [7D7E] [5964] [D79E] [318A] [7990] [A086] [7280] [8756] [F18C] [02D4] [36E4] [69C8] [345A] [34E4] [9878] [EF9C] [7CC0] [0610] [8504] [D23A] [18DE] [3EC8] [5FB2] [9AFE] [BFC2] [DE10] [3344] [3696] [AB9A] [68DC] [3D92] [D068] [B07A] [ED80] [908C] [85F8] [D2C4] [00A4] [B9D0] [72CA] [1EDA] [3078] [2DF0] [FD96] [AFD2] [78BA] [DAB4] [7E90] [0736] [4598] [B1D0] [D15A] [01D2] [3B72] [CEC4] [524A] [2F1E] [D1AA] [CB86] [DAF2] [253C] [3974] [F36E] [C03E] [C006] [D76C] [1FD6] [FE7E] [0292] [E468] [A55A] [115E] [E408] [FFAE] [757C] [4D72] [D030] </pre>
<pre> [CF1EC] [6700] [631C] [68A] [5300] [8F00] [A83C] [5FBA] [15BA] [6980] [D6A6] [FA66] [67BA] [924A] [462] [4E46] [79A0] [DE20] [DE3A] [35AA] [42A8] [48CE] [FCB2] [0496] [0666] [0FE2] [2882] [DB5E] </pre>	<pre> [25BC] [E734] [3A68] [1812] [CD10] [3344] [4F0A] [B334] [2120] [DB60] [DD2E] [2144] [9F2A] [A6E0] [41E2] [CD8C] [6964] [3F74] [05E4] [B36E] [7D7E] [5964] [D79E] [318A] [7990] [A086] [7280] [8756] [F18C] [02D4] [36E4] [69C8] [345A] [34E4] [9878] [EF9C] [7CC0] [0610] [8504] [D23A] [18DE] [3EC8] [5FB2] [9AFE] [BFC2] [DE10] [3344] [3696] [AB9A] [68DC] [3D92] [D068] [B07A] [ED80] [908C] [85F8] [D2C4] [00A4] [B9D0] [72CA] [1EDA] [3078] [2DF0] [FD96] [AFD2] [78BA] [DAB4] [7E90] [0736] [4598] [B1D0] [D15A] [01D2] [3B72] [CEC4] [524A] [2F1E] [D1AA] [CB86] [DAF2] [253C] [3974] [F36E] [C03E] [C006] [D76C] [1FD6] [FE7E] [0292] [E468] [A55A] [115E] [E408] [FFAE] [757C] [4D72] [D030] </pre>	<p>Listing. Der Fall »Famit AG« (Fortsetzung)</p>

Spiele-Listing

4030 DATA eine Leiche,77	[8E8C]	0	[55B8]
4040 DATA Kocht'pfe,78	[F6A2]	4900 DATA im Wohnzimmer,0,0,36,64,0,0	[F1D0]
4050 DATA Tische,79	[E43E]	4910 DATA im Schlafzimmer,0,64,0,63,0,0	[3F3C]
4060 DATA Stuhle,79	[367A]	4920 DATA im Bad,0,0,62,0,0,0	[500C]
4070 DATA einen Schreibtisch,80	[2686]	4930 DATA in einem Flur,62,65,61,66,0,0	[42EC]
4080 DATA eine Schublade,0	[D09C]	4940 DATA in einem Flur,64,0,0,67,0,0	[7010]
4090 DATA einen Schloß,0	[15E0]	4950 DATA in der Küche,0,67,64,0,0,0	[A82A]
4100 DATA einen Schreibtisch,81	[B97C]	4960 DATA im E'zimmer,66,0,65,0,0,0	[9532]
4110 DATA einen Schreibtisch,82	[E780]	4970 DATA an der Rampe,70,69,71,45,0,0	[37F8]
4120 DATA eine Schublade,0	[DE92]	4980 DATA in einer Lagerhalle,68,0,0,0,0,0	[1F60]
4130 DATA einen Spezialschloß,0	[A0C6]	0	[7862]
4140 DATA Maschinen,83	[C8A0]	4990 DATA in einer Lagerhalle,0,68,0,0,0,0	[AB26]
4150 DATA Maschinen,84	[EFA4]	0	[7436]
4160 DATA eine Schalttafel,85	[D0D4]	5000 DATA in einer Lagerhalle,73,72,0,68	[DA8C]
4170 DATA Maschinen,86	[29AC]	,0,0	[4D4A]
4180 DATA Maschinen,87	[6CB0]	5010 DATA in einer Lagerhalle,71,0,0,0,0,0	[A6F2]
4190 DATA eine Schalttafel,88	[BCE0]	0	[4D08]
4200 DATA einen Zaun,93	[A90A]	5020 DATA in einem Büro,0,71,0,0,0,0,0	[5EB4]
4210 DATA ein Loch im Zaun,0	[B426]	5030 DATA im Eingang,75,43,0,81,0,0,0	[C960]
4220 DATA eine Tür,43	[C872]	5040 DATA in einem Flur,79,74,76,80,0,0,0	[67CC]
4230 DATA etwas Kleingeld,0	[09A4]	5050 DATA in einem Flur,0,77,0,75,0,0,0	[C716]
4240 DATA ein Fenster,60	[71D2]	5060 DATA in einem Waschraum,76,0,0,0,0,0,0	[DC90]
4250 DATA eine Schublade,0	[619A]	0	[C88C]
4260 DATA einen Mann,0	[3076]	5070 DATA in einer Küche,0,0,0,79,0,0,0	[25F0]
4270 DATA einen Holzzaun,34	[FB88]	5080 DATA in der Kantine,0,75,78,0,0,0,0	[DF82]
4280 DATA eine versteckte Tür,0	[FBD0]	5090 DATA in einem Büro,82,0,75,0,0,0,0	[4780]
4290 DATA eine Tür,47	[B488]	5100 DATA in einem Büro,0,0,74,0,0,0,0	[AD0E]
4300 DATA vor dem Polizeigebäude,0,2,18,	[859C]	5110 DATA in einem Büro,0,80,0,0,0,0,0	[1394]
19,0,0	[1C66]	5120 DATA in einer Fabrikhalle,84,42,0,8	[628C]
4310 DATA im Eingang,1,3,0,5,10,0	[541E]	5,0,0	[3ACC]
4320 DATA in einem Flur,2,0,0,4,0,0,0	[1F32]	5130 DATA in einer Fabrikhalle,84,83,0,0	[24E2]
4330 DATA in einem Büro,5,0,3,0,0,0,0	[402E]	0,0,0	[B3D8]
4340 DATA in einem Flur,0,4,2,6,0,0,0	[7230]	5140 DATA in einem Schaltraum,0,0,83,0,0	[FFE2]
4350 DATA in einem Flur,0,0,5,7,0,0,0	[E3AC]	0,0,0	[F3DE]
4360 DATA in einem Flur,8,9,6,20,0,0,0	[0938]	5150 DATA in einer Fabrikhalle,87,39,0,8	[E3E6]
4370 DATA in einem Büro,0,7,0,0,0,0,0	[793A]	8,0,0	[D188]
4380 DATA in einem Büro,7,0,0,0,0,0,0	[F7EE]	5160 DATA in einer Fabrikhalle,87,86,0,0	[5602]
4390 DATA in einem Flur,0,11,0,12,0,2	[04E8]	0,0,0	[9BF0]
4400 DATA in einem Büro,10,0,0,13,0,0,0	[6746]	5170 DATA in einem Schaltraum,0,0,86,0,0	[960A]
4410 DATA in einem Flur,0,13,10,14,0,0,0	[A6EC]	0,0,0	[F2C2]
4420 DATA in einem Büro,12,0,11,0,0,0,0	[43E8]	5180 DATA auf der Straße,92,90,91,22,0,0	[8004]
4430 DATA in einem Flur,0,0,12,15,0,0,0	[4B60]	0,0,0	[CB4E]
4440 DATA in einem Flur,16,17,14,0,0,0,0	[8C94]	5190 DATA auf der Straße,89,90,90,90,0,0	[E7F2]
4450 DATA in einem Büro,0,15,0,0,0,0,0	[8A26]	5200 DATA auf der Straße,91,91,91,89,0,0	
4460 DATA im Büro des Polizeirates,15,0,	[4462]	5210 DATA auf der Straße,93,89,92,92,0,0	
0,0,0,0	[0E6A]	5220 DATA auf der Straße,93,92,94,93,0,0	
4470 DATA auf der Straße,18,18,18,1,0,0,0	[49EC]	5230 DATA auf der Straße,94,94,94,93,0,0	
4480 DATA auf der Straße,19,19,1,19,0,0,0	[1EAA]	5240 DATA vorn,hinten,links,rechts,oben,	
4490 DATA in der Kantine,0,0,7,0,0,0,0	[7588]	unten	
4500 DATA vor der Fabrik,0,0,22,23,0,0,0	[51AE]	5250 DATA 83,105,101,32,104,97,98,101,11	
4510 DATA auf der Straße,22,22,89,21,0,0,0	[C3B2]	0,32,32,100,101,110,32,77,97,110,11	
	[3EC0]	0,32,118,101,114,104,97,102,116,101	
4520 DATA auf der Straße,23,24,21,26,0,0,0	[2FC0]	,116,32,44,97,117,115,32,32,100,114	
	[28DA]	,105,110,103,101,110,100,101,109,32	
4530 DATA auf der Straße,23,24,24,24,0,0,0	[388A]	,32,86,101,114,100,97,99,104,116,32	
	[9740]	,100,117,114,99	
4540 DATA auf der Straße,26,25,25,25,0,0,0	[70A4]	5260 DATA 104,32,105,108,108,101,103,97,	
	[8BD2]	108	
4550 DATA auf der Straße,26,25,23,26,0,0,0	[11AC]	5270 DATA 101,32,71,101,115,99,104,123,1	
	[D5DC]	02,116,101,32,49,48,32,77,105,108,1	
4560 DATA am Eingang,29,0,0,28,0,0,0	[6ED6]	08,105,111,110,101,110,32,77,97,114	
4570 DATA in der Pförtnerloge,0,0,27,0,0,0	[958C]	,107,32,117,110,114,101,99,104,116,	
	[4EC2]	109,123,126,105,103,32,101,114,119,	
4580 DATA auf einem Weg,31,27,30,0,0,0,0	[56EA]	111,114,98,101,110,32,122,117,32,10	
4590 DATA auf einem Parkplatz,0,0,0,29,0,0,0	[FDE2]	4,97,98,101,110,46,32,68,101,114,32	
	[54C8]	,77	
4600 DATA auf einem Weg,32,29,0,0,0,0,0	[7C20]	5280 DATA 97,110,110,32,98,101,103,105,1	
4610 DATA auf einem Weg,34,31,38,33,0,0,0	[CA04]	10,110,116,32,122,117,32,115,112,11	
4620 DATA vor einem Verwaltungsgebäude,0,	[81BC]	4,101,99,104,101,110,32,58,32,34,73	
58,32,0,0,0	[A4D4]	,99,104,32,103,101,115,116,101,104,	
4630 DATA auf einem Weg,0,32,37,0,0,0,0	[8DDA]	101,46,32,65,110,103,101,102,97,110	
4640 DATA in einem Garten,0,34,0,36,0,0,0	[2510]	,103,101,110,32,104,97,116,32,97,10	
4650 DATA auf einer Terrasse,0,0,35,0,0,0,	[42CA]	8,108,101,115,32,109,105,116,32,83,	
0	[6D30]	116	
4660 DATA in einer Fahrzeughalle,0,0,0,3	[28C2]	5290 DATA 101,117,101,114,104,105,110,11	
4,0,0	[23A0]	6,101,114,122,105,101,104,117,110,1	
4670 DATA auf einem Weg,0,0,39,32,0,0,0	[092C]	03,44,32,115,112,123,116,101,114,32	
4680 DATA auf einem Weg,86,40,42,38,0,0,0	[F1E8]	,107,97,109,101,110,32,100,97,110,1	
4690 DATA in einer Fabrikhalle,39,0,0,0,0,	[A1E4]	10,32,105,108,108,101,103,97,108,10	
0,0	[2F98]	1,32,71,101,115,99,104,123,102,116,	
4700 DATA in einer Fabrikhalle,42,0,0,0,0,	[769A]	101,32,104,105,110,122,117,46	
0,0	[4F12]	5300 DATA 123,102,116,101,32,104,105,110	
4710 DATA auf einem Weg,83,41,43,39,0,0,0	[CF62]	,122,117,46,32,68,101,114,32,71,101	
4720 DATA auf einem Weg,0,44,0,42,0,0,0	[1A38]	,115,99,104,123,102,116,115,102,125	
4730 DATA auf einem Weg,43,0,45,0,0,0,0	[EE9C]	,104,114,101,114,32,109,101,114,107	
4740 DATA vor einem Lagerhaus,0,0,68,44,	[4A9C]	,116,101,32,101,116,119,97,115,44,3	
0,0		2,105,99,104,32,119,111,108,108,116	
4750 DATA im Eingang,48,47,0,51,52,0		,101,32,109,105,99,104,32,97,98,115	
4760 DATA in einem Büro,0,0,0,0,0,0,0		,101	
4770 DATA im Archiv,0,46,0,0,0,0,0		5310 DATA 116,122,101,110,44,32,97,98,10	
4780 DATA in einem Büro,0,51,0,0,0,0,0		1,114,32,101,115,32,119,97,114,32,1	
4790 DATA im Schaltraum,51,0,0,0,0,0,0		22,117,32,115,112,123,116,46,32,73,	
4800 DATA in einem Flur,49,50,46,57,0,0,0		99,104,32,109,117,126,116,101,32,10	
4810 DATA in einem Flur,53,54,0,55,0,46		5,104,110,32,116,96,116,101,110,46,	
4820 DATA in einem Büro,0,52,0,0,0,0,0		34	
4830 DATA in einem Büro,52,0,0,0,0,0,0			
4840 DATA in einem Büro,56,0,52,0,0,0,0			
4850 DATA im Chefbüro,0,55,0,0,0,0,0			
4860 DATA in einem Toilettenraum,0,0,51,			
0,0,0			
4870 DATA auf einem Pfad,33,0,0,59,0,0,0			
4880 DATA auf einem Pfad,60,0,58,0,0,0,0			
4890 DATA hinter dem Gebäude,0,59,0,0,0,			

Listing. Der Fall »Famit AG« (Schluß)

Autofahren ohne Tempo-Limit



Auf Ihrem Computer können Sie sich die Wünsche noch erfüllen, die im wirklichen Leben nicht mehr realisierbar sind: Fahren ohne Tempo-Limit mit dem heißen Flitzer in »Mini-Racer«.

»Mini-Racer« ist ein kleines Autorennspiel, das mit wenig Programmieraufwand eine Menge Spaß bietet. Interessant ist dieses Programm wohl deshalb, weil es nicht sehr lang — und deshalb leicht überschaubar — ist.

Sie sind der Fahrer eines hochgezüchteten Formel-1-Rennwagens und müssen einen langen gefährlichen Parcours möglichst heil überstehen. Leicht wird Ihnen die Aufgabe nicht gemacht, denn immer wieder liegen Hindernisse auf der Straße. Oder gefährliche Ölspurten und scharfe Kurven machen Ihnen das Leben schwer.

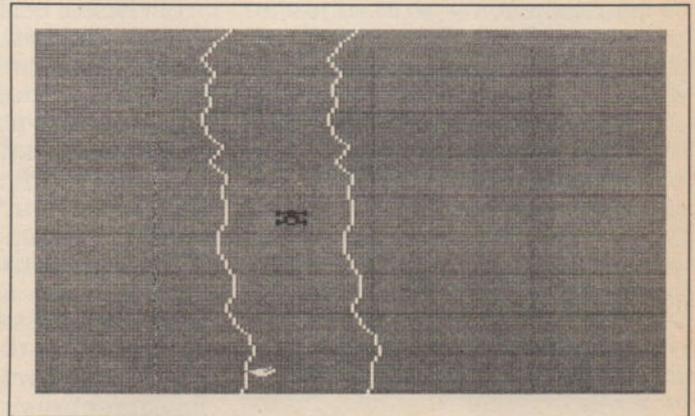
Steuern können Sie Ihren heißen Flitzer mit den beiden Cursortasten »rechts« und »links«.

Das ganze Spiel geht auf Zeit. Das heißt, Sie durchfahren die Rennstrecke. Danach wird die Zeit gestoppt, unabhängig davon wieviele Crashes Sie bauen. Nach einem Zusammenstoß jedoch verlangsamt sich Ihr Renner und erreicht erst nach einigen Sekunden wieder die Höchstgeschwindigkeit. Am Schluß des ganzen Spektakels werden Ihnen die Punkte sowie die Anzahl der »Unfälle« angezeigt.

Zum Programm selber: In den Zeilen 100 bis 140 werden die Hüllkurven, Zeichen, Farben sowie das Grundbild definiert. Von 180 bis 300 finden Sie die Hauptroutine zum Erzeugen des Parcours und der Hindernisse, sowie die zum Steuern des Wagens. Die Zeilen 340 bis 370 beenden das Programm mit einer Schlußmelodie, der Punktebewertung und der Rückkehr zum Menü.

Viel Spaß und Hot Wheels!

(Michael Bauer)



Mit Basic kann man auch schnelle Spiele schreiben

```

10 REM ***** [82E2]
20 REM * [94B4]
30 REM * MINI-RACER * [ABC4]
40 REM * * [68BB]
50 REM * MICHAEL BAUER * [DE7E]
60 REM * SCHWANTHALERSTR.180 * [B484]
70 REM * 8 MUENCHEN 2 * [77B8]
80 REM * TEL. 089/5022642 * [2690]
90 REM ***** [12F2]
100 INK 1,10:CLS:BORDER 0:INK 0,0:INK 2, [98FA]
    24:INK 3,26:INK 4,21:PRINT CHR$(23) "
    1":DEFINT a-z:DEFREAL t:we=128
110 ENV 1,15,-1,20:ENV 2,15,-1,20:ENT 1, [391E]
    20,-2,1,1,30,1,20,-2,1,50,1,1:ENT 2,
    25,4,1,200,-1,1
120 SYMBOL 240,&99,&FF,&99,&3C,&24,&A5,& [22C6]
    E7,&99:SYMBOL 241,0,&10,&2A,&4D,&55,
    &A2,&91,8:SYMBOL 242,0,&7F,&41,&41,&
    7F,8,8,8:SYMBOL 243,&38,&2C,&5A,&AA,
    &6C,&10,&10,&10:SYMBOL 244,0,2,&1D,&
    FB,&3E,&1C
130 MODE 0:ver=150:x=308:x1=x:xpl=250:xp [DB7E]
    r=350:PLOT xpl,0,1:DRAWR 0,399:PLOT
    xpr,0:DRAWR 0,399:FOR y=0 TO 360 STE
    P 20:PLOT 300,y,3:DRAWR 0,14:NEXT:PL
    OT 250,204,2:DRAWR 96,0:MOVE x,200:T
    AG:PRINT CHR$(240);:MOVE 358,208:PRI
    NT CHR$(242);
140 FOR v=0 TO 2000:NEXT:SOUND 1,60,10,1 [CS84]
    5:MOVE x,200:PRINT CHR$(240);:t1=TIM
    E+90000 [E1BA]
150 ' [04BC]
160 ' [03BE]
170 ' [CB9A]
180 PLOT 0,400,2:MOVE x,200:TAG:PRINT CH [7C36]
    R$(240);
190 IF (INKEY(1)=0 AND x1<607) THEN x1=x [8764]
    1+4 [B6BC]
200 IF (INKEY(8)=0 AND x1>1) THEN x1=x1- [6AB8]
    4 [2BC4]
210 FOR v=0 TO ver:NEXT [487C]
220 SOUND 129,300+ver*5,20,13:SOUND 130,
    340+ver*5,20,13
230 TAG:MOVE x,200:PRINT CHR$(240);
240 TAGOFF:LOCATE 1,1:PRINT CHR$(11):TAG
250 s=INT(RND*16):IF INT(RND*2)=1 THEN P [FCC8]
    LOT xpl,383,1:IF xpl>96 THEN xpl=xpl
    -s ELSE xpl=xpl :ELSE PLOT xpl,383,1
    :IF xpl<480 THEN xpl=xpl+s [3486]
260 DRAW xpl,399:PLOT xpr,383:xpr=xpl+we
    :DRAW xpr,399:IF xpr<580 AND xpl>50
    THEN IF INT(RND*30)=1 THEN PLOT 0,40
    0,RND*2+2:MOVE xpl-230+(INT((RND*2)+
    1)*186),399:PRINT CHR$(241+RND*2);
270 IF INT(RND*30)=2 THEN PLOT 0,400,1:M [ED3A]
    OVE xpl+16*INT(RND*4),399:PRINT CHR$(
    244);
280 x=x1:p=p+1:ver=ver-2:IF ver<1 THEN v [1910]
    er=1
290 IF (TEST(x,202)OR TEST (x+16,202) OR [19B6]
    TEST(x+28,202))=1 THEN SOUND 132,50
    0,200,15,1,,31:BORDER 26:INK 0,26:CA
    LL &BD19:INK 0,0:BORDER 0:ver=150:x1
    =xpl+48:x=x1:cr=cr+1 [C7F4]
300 IF TIME<t1 THEN 180 [E1B6]
310 ' [E0BB]
320 ' [DFBA]
330 '
340 IF p>hp THEN hp=p:SOUND 1,70,70,15,2 [3012]
    ,1:SOUND 2,72,70,15,2,1 ELSE SOUND 1
    ,200,225,15,,2:SOUND 2,202,225,15,,2
350 MODE 1:PEN 3:LOCATE 5,10:PRINT"SCORE [91CA]
    =" ;p;"{3 SPACE}HIGH SCORE =" ;hp:LOC
    ATE 5,12:PRINT"CRASH'S =" ;cr:LOCATE
    15,17:PRINT"nochmal j/n"
360 i$=INKEY$:IF i$="j"THEN p=0:cr=0:GOT [14AE]
    O 130 [8F54]
370 IF i$<>"n" THEN 360

```

Listing. Schaffen Sie es, ohne Unfall ins Ziel zu kommen?

Dr. Logo — mehr als nur eine Schildkröte



Für viele Hobby-Programmierer ist Logo, speziell das Dr. Logo des Schneider, Neuland. Was kann diese Sprache?

Wenn Basic der Golf unter den Programmiersprachen ist — weitverbreitet, handlich, einfach zu bedienen —, so ist Logo eher mit einem Exoten zu vergleichen: Anspruchsvoll in der Bedienung, aber komfortabel und sehr leistungsstark. Also lieber Logo als Basic? Die Antwort auf diese Frage hängt von der gewünschten Anwendung ab. Was nützt schließlich ein Sportwagen, wenn man Möbel transportieren will? Schauen wir uns zunächst einmal Logo etwas näher an.

Logo — was ist das?

Logo ist wie Basic, eine problemorientierte Sprache, das heißt die Struktur ist an der Aufgabenstellung orientiert und nicht am Computer (Gegensatz: maschinenorientierte Sprache, zum Beispiel Assembler).

Logo wurde 1967 am Massachusetts Institute für Technology (MIT) entwickelt. Wie Basic ist auch Logo eine Dialogsprache. Die einzelnen Programmteile, die in Logo Prozeduren heißen, werden am Bildschirm entwickelt und können unabhängig vom übrigen Programm bearbeitet und getestet werden. Eine jedesmal neue Gesamtübersetzung, wie zum Beispiel in Pascal, entfällt. Es ist daher sogar möglich, Logo-Prozeduren während des Programmablaufs zu ändern, ohne größere Wartezeiten zu verursachen.

Logo ist grafikfähig. Es besitzt die sogenannte Turtle-Grafik (ist aber nicht etwa mit ihr identisch), eine mit Kommandos steuerbare Schildkröte, die Linien ziehen, Schleifen malen und Kästchen zeichnen kann. Dieses »Tier«, das bei der Logo-Version des CPC zu einem einfachen Dreieck degeneriert ist, hat sich leider zum Symbol der ganzen Sprache aufgeschwungen und so dazu geführt, daß Logo zu einer Art »Baby«-Sprache abqualifiziert wurde. So verwendet man die Vorteile von Logo meist nur bei der Anwendung im

Schulunterricht, um kleinere Kinder mit der neuen Technik vertraut zu machen. Aber Logo kann viel mehr.

Mit Pascal hat Logo das modulare Prinzip gemeinsam. Jedes Programm besteht aus einer Summe von Prozeduren (Unterprogrammen), die sich untereinander aufrufen. Jede Prozedur ist ein unabhängiges Modul, das von außen mit Daten versorgt wird. Durch sogenannte lokale Variable (sind nur in einer Prozedur gültig) ist es möglich, Variablen gleichen Namens in mehreren Modulen unterschiedlich zu verwenden, ohne daß die Inhalte überschrieben werden. Durch dieses Baukastenprinzip kann man eine intensive Verflechtung von verschiedenen Programmteilen erreichen und somit eine starke Strukturierung erzwingen. Dies hat mehrere Vorteile: Zum einen kann Logo auf eine sich ständig vergrößernde Zahl von Prozeduren (fast jede Programmierung schafft ja neue Prozeduren) zurückgreifen, die dann sofort auch für andere Programme zur Verfügung stehen. Daneben braucht man sich nicht mit den Namen der Variablen, wie bei Basic, herumzuschlagen. Bei komplexen Programmen ist Basic hiermit klar im Nachteil: Alle Variablen-Namen müssen vorher bekannt sein, sonst gibt es schnell ein großes Durcheinander.

Logo ist eine lernende Sprache. Es gibt eine bestimmte Anzahl von vordefinierten Befehlen, den sogenannten Primitives, aus denen man seine (ersten) Prozeduren zusammensetzt. Ist aber eine Prozedur erst einmal definiert, kann sie einfach durch Nennung ihres Namens wie ein Logo-Befehl benutzt werden. Durch die Definition ist die Prozedur zu einem Bestandteil des Befehlssatzes geworden, der sich so beliebig erweitern läßt.

Doch das ist noch nicht alles. Die eigentliche Stärke von Logo beruht, wenn man einmal vom Prozedurkonzept (Baukastenprinzip) absieht, auf zwei Hauptpfeilern: der listenorien-

tierten Datenstruktur und der Rekursion. Das sind zwei Eigenschaften, die Logo als ernsthafte Programmiersprache höchst interessant machen. Wir wollen an dieser Stelle zunächst die theoretischen Betrachtungen unterbrechen und anschauen, wie ein einfaches Beispiel in Logo aussieht.

Jedes Logo-Programm wird als Prozedur formuliert. Jeder Prozedur muß man einen Namen geben, unter dem diese dem Interpreter bekannt gemacht wird. Der Name darf kein Primitiv sein, keine Leerstelle und auch keine Rechenzeichen enthalten. Er darf aber mit einer Ziffer beginnen. Die Definition einer Prozedur wird immer mit dem Kommando »to name (eventuell mit Variablen)« eröffnet. Hinter dem Namen der Prozedur können eine oder mehrere Variablen angegeben werden. Werden Variablen mit übergeben, so ist die Prozedur eine Funktion der angegebenen Variable n. Die Definition jeder Prozedur wird mit »END« beendet.

Ein Beispiel: Wir schreiben eine einfache Prozedur, die ein Quadrat mit der vorgegebenen Kantenlänge k zeichnet. Sie müssen als erstes Logo von der Systemdiskette laden. Sobald das Fragezeichen erscheint, ist Logo eingearbeitet und wartet auf Ihre Befehle. Nun geben Sie die nachfolgenden Zeilen, jeweils mit ENTER, beziehungsweise beim CPC 6128 RETURN, abgeschlossen ein. Dabei müssen Sie unbedingt darauf achten, daß die Leerstellen genauso auf dem Schirm erscheinen, wie auch in dem Listing 1.

Das Schlüsselwort »to« zeigt an, daß nun eine Prozedur definiert wird. Es folgt deren Name und eine Variable (:k).

Logo-Programme sind einfach

Bis hierhin sind nun auch schon eine Reihe von Punkten zu beachten. Der Doppelpunkt wird bei Logo als Kennzeichen für eine Variable, ganz gleich ob »Zahl«, »Wort«, »Satz« oder »Liste« verwandt. Er dient als Unterscheidungsmerkmal zwischen Variablen und Prozeduren, beziehungsweise einfachen Befehlen (Primitives). Dabei ist ein »Wort« eine Aneinanderreihung von ASCII-Zeichen. »Satz« steht in Logo für eine Reihe von Worten. Eine »Liste« beinhaltet eine Reihe von »Worten« oder »Zahlen«, kann also sowohl Zahlen wie Zeichen beinhalten. Ihre Aufgabe ist zunächst einmal die Daten-

speicherung, einem Array oder einem Data-Feld in Basic vergleichbar. Ihre Anwendung reicht jedoch weit über diese einfachen Basic-Strukturen hinaus. Es ist möglich, Listen im Prinzip wie Strings in Basic zu behandeln, allerdings mit dem Unterschied, daß eine Logo-Liste eine Aneinanderreihung selbständiger Elemente darstellt. Wir kommen auf diese Fähigkeiten später noch einmal zurück. Es existiert in Logo übrigens keine Unterscheidung zwischen den Variablenarten, die sich anhand eines Kürzels schon im Namen ablesen ließe (wie beispielsweise %, \$ und so weiter in Basic). :t kann also sowohl eine Liste wie auch eine Zahl repräsentieren.

Delimiter: Was ist das?

Als zweiten Punkt wollen wir etwas zu den Trennmarkierungen, den sogenannten Delimitern, sagen. Vom Schneider-Computer ist man ja bereits an Klarheit und Vorsicht bei der Verwendung von Leerzeichen gewöhnt. Noch stärkere Bedeutung erhalten Leerstellen allerdings in Logo. Sie fungieren hier nämlich als Trennzeichen (etwa so wie in Basic der Doppelpunkt), grenzen also Befehle beziehungsweise Prozeduren voneinander und von Variablen ab. Da es keine Unterteilung eines Logo-Programms durch Zeilennummern gibt, stellen die Leerzeichen die einzigen Mittel dar, ein Programm zu unterteilen. Sie müssen also immer exakt so eingegeben werden, wie in den einzelnen Programmen beschrieben. 90 Prozent aller Fehler bei der Eingabe sind auf falsche Leerzeichen zurückzuführen.

Als ersten Befehl in der zweiten Reihe finden Sie das Wort »Repeat«. Es gehört zu den Primitives, also zum Grundbefehlssatz des Dr. Logo. Die nachstehende Zahl gibt an, wie oft die folgende Anweisung wiederholt werden soll. Was soll nun aber eigentlich getan werden? Die Antwort darauf liefert der Ausdruck in der eckigen Klammer. Diese Klammern spielen ebenfalls eine große Rolle in der Logo-Syntax. Sie grenzen den Inhalt von Listen ab.

Die Umklammerung definiert, was zusammengehört. In unserem Beispiel sind dies die zwei Befehle »fd« und »rt«, jeweils mit zugehöriger Parameterangabe. Beide Kommandos sind Primitives und gehören zu den Zeichenfunktionen, die die Schildkröte bewegen. »fd« weist diese Schildkröte an, vorwärts zu ge-

hen. Vorwärts, das heißt bei Logo immer in Richtung der Pfeilspitze. Wenn man also im Ausgangszustand in eine andere Richtung als nach oben spazieren will, so muß der Turtle erst »der Kopf verdreht werden«. Dazu gibt es zwei Befehle. »lt« dreht die Schildkröte um die angegebenen Winkelgrade nach links, »rt« bewegt sie in der Gegenrichtung. Was passiert nun also?

Mit »fd« geht die Turtle zunächst nach oben und zwar um die in :k gespeicherte Anzahl von Schritten. Es gilt dabei derselbe Rahmen wie in Basic, also 639 Schritte in der Horizontalen und 399 in der Vertikalen. Danach kommt ein »rt 90«. Die Nase der Turtle dreht sich um 90 Grad nach rechts. Normalerweise würde nun zur nächsten Anweisung weitergegangen. Durch die Umklammerung und das Repeat-Kommando wiederholt sich dieser Vorgang aber viermal. Die Schildkröte geht also wieder :k-Schritte vorwärts. Da die Spitze nun aber nach rechts weist, wird der Strich nach rechts gezogen. Noch ein Dreh und ein gleichlanger Strich nach unten folgt. Nach dem letzten »rt« und :k-Schritten ist die Schildkröte wieder am Ausgangspunkt angelangt.

»End« beschließt die Definition der Prozedur. Dieses Schlüsselwort muß immer gebraucht werden, um dem Interpreter das Ende der Prozedur mitzuteilen. Damit ist der neue Befehl definiert. Für Logo gibt es keinen wesentlichen Unterschied mehr zwischen unserem Befehl »Quadrat« und einem seiner Primitives. Der Befehl »Quadrat« kann jetzt ohne Schwierigkeiten in anderen Prozeduren aufgerufen werden. Dazu geben Sie einfach das Befehlswort »quadrat 60« ein, und ein Quadrat mit der Kantenlänge 60 wird auf den Schirm gezeichnet. Bevor Sie weiterlesen, sollten Sie unbedingt diese erste Prozedur ausprobieren, um sich mit Logo vertraut zu machen. Dazu können Sie ruhig auch einmal im Programmtext editieren. Versuchen Sie beispielsweise einmal, ein gleichseitiges Dreieck zu erhalten. Dazu muß die Zahl nach dem »repeat« auf 3 herabgesetzt und die Spitze der Turtle jedesmal um 12 (statt 90) Grad gedreht werden. Wie dabei editiert wird, gibt Ihnen Tabelle 1 wieder.

Wie Sie sehen, ist es wirklich sehr einfach mit Logo schnell brauchbare Grafiken auf den Bildschirm zu zaubern. Allerdings bleibt zu sagen, daß das Schneider-Logo relativ langsam im Vergleich mit anderen Versionen ist. Auch wenn man die

Turtle abschaltet, ist Dr. Logo noch langsamer als Basic. Dennoch eignet sich die Sprache recht gut dazu, schnell und nachvollziehbar einfache Grafiken zu erstellen. Unter diesem Gesichtspunkt ist Logo den entsprechenden Basic-Befehlen deutlich überlegen. Zum einen sind die Definitionen kürzer und ohne viel Rechnen zu programmieren. Zum anderen wurden einige »Macken« der Schneider-Befehle verbessert. So gibt es bei Logo verschiedene Bildschirmmodi (insgesamt 3), die festlegen, was passiert, wenn die Turtle den Bildschirm verlassen würde. Bei »Window« reagiert sie wie gewohnt. Was außerhalb des Bildschirms liegt, wird nicht gezeichnet. Bei »Fence« gibt es in solch einem Fall eine Fehlermeldung. Zeichnet die Schildkröte im »Wrap«-Modus, so wird der Bildschirm zu einem Kugelschirm. Läuft die Turtle an einer Seite aus dem Bild, so taucht sie gleichzeitig auf der anderen Seite wieder auf.

Diese schönen Eigenschaften bei der Grafikprogrammierung sind jedoch alleine kein Grund, auf Logo umzusteigen. Interessanter wird es aber, wenn wir die Grafik ganz aus dem Spiel lassen und Logo als ernsthafte Programmiersprache untersuchen. Und hier lohnt es sich aufzuhorchen. Logo arbeitet mit Zahlen bis 14 Stellen Genauigkeit. Dabei sind dreistellige Exponenten erlaubt. Während selbst gute Taschenrechner bei der Fakultätsberechnung von Zahlen über 69 meist einen Fehler melden, wird von Dr. Logo auch die Fakultät von 100 noch korrekt berechnet.

Prozeduren rufen sich selbst auf

Und damit kommen wir zu einer weiteren Eigenheit von Logo — der rekursiven Programmierung. Worum handelt es sich dabei? Eine rekursive Prozedur benötigt zu ihrer Definition sich selbst, gibt Daten an sich selbst weiter und erhält diese wieder von sich selbst zurück. Dies klingt auf den ersten Blick etwas kompliziert und vielleicht fragen Sie sich jetzt, wofür das eigentlich gut ist. Nun, zum Beispiel um den Effektivzins eines Ratenkredites zu berechnen und festzustellen, wieviel man denn noch wirklich dafür bezahlt. Der Zusammenhang ist hierbei wie folgt: Bei einem Ratenkredit zahlt man eine gleichbleibende monatliche Rate. Diese setzt sich aus einer gleichbleibenden Tilgung und

einem gleichbleibenden Zinsanteil zusammen, wobei die Zinsen, und dies ist das Schwierige, zu dem anfänglichen Auszahlungsbetrag hinzugezählt werden. Will man nun exakt berechnen, was der Kredit genau kostet, so kommt man nur durch Probieren zu einem Ergebnis (so geht es in Basic). Weitere Wege führen über eine Näherungsformel (so machen es teilweise die Banken) oder man verwendet Logo und programmiert das Ganze rekursiv.

Neben mathematischen Formeln, die zu ihrer Lösung rekursives Programmieren benötigen, gibt es jedoch auch noch eine ganze Reihe anderer Anwendungen, wo die Rekursion buchstäblich Berge versetzt. In unserem nun folgenden zweiten Beispiel werden zwar nur Befehlsberge versetzt, aber Logo wird durchschaubar.

Sie kennen sicher das Spiel »Türme von Hanoi«. Wenn nicht, hier eine kurze Einführung: Bei den Türmen von Hanoi handelt es sich um ein Strategiespiel. Vorgegeben sind drei Plätze. Auf einem Platz ist dabei ein Turm aus Scheiben aufgebaut, wobei die größten Scheiben zuunterst liegen. Die Spiel-Aufgabe besteht nun darin, den gesamten Turm auf einen der beiden anderen Plätze zu verlagern. Es sind allerdings zwei Nebenbedingungen zu beachten:

— Es darf immer nur eine Scheibe bewegt werden (dies entspricht einem Zug).

— Es darf nur eine kleinere auf einer größeren Scheibe liegen, nie umgekehrt.

Dieses Spiel ist im Listing 2 als Logo-Simulation ausgeführt. Das eigentliche Programm besteht aus einer Anzahl selbständiger Prozeduren, die durch eine Oberprozedur zusammengebaut werden. Sie müssen also zuerst alle Prozeduren einzeln eingeben. Das Programm wird dann durch die »Main«-Prozedur »TvH« aufgerufen. Es besteht inklusive dieser aus 13 Prozeduren. Diese müssen dabei wie im Listing 2 eingegeben werden. Besonders auf die Leerzeichen ist zu achten. Wie die Prozeduren zusammenhängen, zeigt Tabelle 3. Nun zunächst ein paar Informationen zu den einzelnen Prozeduren:

Die Hauptprozedur ist TvH. Sie erlaubt in einem Menü die Auswahl zwischen zwei Spielalternativen: Unter Punkt 1 simuliert der Computer die optimale Lösung (bei x Steinen braucht man mindestens $2^x - 1$ Züge). Bei Auswahl von Punkt 2 kann sich der Spieler selbst versuchen.

Interessant ist das Starten der Simulation. Hier wird eine weitere Besonderheit des Dr. Logo benutzt, die Listenverarbeitung. Wie die Programmiersprache Lisp ist Logo eine listenorientierte Sprache. Logo macht damit keinen Unterschied zwischen »Zahl« oder »Zeichen« als Inhalt einer Variablen — ganz im Gegensatz zu Basic. Entsprechend unterscheidet es nicht zwischen Strings, Real- und Integervariablen. Eine Variable in Logo kann alles sein. Ungewohnt ist das Fehlen von indizierten Variablen (= Feldern). Daten-Felder werden statt dessen in Listen gespeichert. Dieses Konzept löst sich von der Speicherplatzorientierten Ablage von Daten und vermeidet überdies lästige Dimensionsanweisungen. Eine Liste ist eine Ansammlung von Daten (Zahlen, Worte, Sätze oder Listen), deren Anordnung einzig durch die Reihenfolge vorgegeben ist. Wie bei der Stringverarbeitung in Basic ist es nun möglich, zu solch einer Liste Glieder vorne oder hinten zu addieren, Listen in Teil-Listen zu zerlegen und sie sogar als Programm zu interpretieren. Das Dr. Logo bietet dazu eine Vielzahl von Befehlen.

Programme ändern sich selbst

Mit RUN wird beispielsweise der Inhalt einer Liste als Befehlsliste interpretiert und ausgeführt. Mit ITEM wird die gewünschte Liste ausgewählt. Logo ist also in der Lage, ohne komplizierte Maschinen-POKES und ähnliche Tricks, höchst komfortabel Programme selbst umzubauen. Die Prozedur, bei der zwischen verschiedenen Programmvarianten zu unterscheiden ist, führt eine Liste aus, die dann jeweils entsprechend den aktuellen Bedürfnissen des Benutzers geändert werden kann. Schauen wir uns nun die anderen Prozeduren an.

simul — präsentiert die Eingabemaske für die Computer-Simulation

turm — diese Zeilen enthalten die ganze Logik des Spiels. Ein entsprechendes Basic-Programm würde zwischen 20 und 30 Zeilen umfassen. Hier spielt Logo seine ganze Stärke aus: die Rekursion. Wenn die Variable :m größer als 1 ist, so ruft die Prozedur sich selbst, allerdings mit vermindertem :m auf.

ausgabe — Ausgabe-Maske der Simulation. Hier gibt Logo das Ergebnis seiner Berechnungen aus.

selbst — Eingabe-Maske für den Selbstversuch.

turmliste — Um diese Prozedur zu verstehen, muß etwas zum grundsätzlichen Ablauf des Programms gesagt werden. Beim Selbstversuch werden drei Listen angelegt: eine enthält am Anfang alle Elemente, die beiden anderen sind vorerst leer. Mit Hilfe von »turmliste« werden nun in die erste Liste die Elemente durch die Rekursion hintereinander eingeschrieben. In »selbst« wird dann diese Liste mit zwei »Leer-Listen« zu einer Gesamtliste zusammengebaut, die dann die Elemente aller drei Plätze enthält. Jede Scheibe ist dabei durch eine Zahl repräsentiert, wobei kleinere Zahlen auch kleinere Scheiben bedeuten. Das Programm arbeitet maximal mit einem Stapel von neun Scheiben. »turmliste« stellt ein atypisches Beispiel für die rekursive Programmierung dar, ist aber auch relativ einfach zu erklären. Beim Aufruf der Prozedur muß eine Variable übergeben werden. Diese wird in :t eingelesen. Wenn :t Null ist erfolgt der Rücksprung, ansonsten wird die Liste t1 um das Element :t erweitert. :t wird dabei an die erste Stelle der Liste gesetzt.

Alle anderen Elemente der Liste werden weiter nach hinten geschoben. Als nächstes — und dies ist nun die eigentliche Rekursion — wird »turmliste« mit dem um 1 verminderten :t wieder aufgerufen. Wir erhalten eine Liste, die nacheinander die Zahlen von 1 bis :t enthält, wobei die 1 an erster Stelle steht, da jedes neue Element vorn angefügt wird.

ziehen — Eingabemaske der Züge
zugtest — Hier wird der Zug auf seine Durchführbarkeit getestet:

1. Ist die Zugliste leer?
2. Ist der oberste Stein identisch mit dem, der gezogen werden soll?
3. Ist der gezogene Stein auch kleiner als der oberste auf dem Zielstapel? Wenn diese Bedingungen nicht erfüllt sind, wird »err« aufgerufen.

err — stellt eine fehlerhafte Eingabe fest und gibt wieder an »ziehen« ab.

listzug und **cut** — Hier wird der Zug ausgeführt. Das erste Element der Ausgangsliste wird vor das erste Element der Zielliste gesetzt.

ende — Test, ob zwei oder drei Unterlisten leer sind. Wenn ja, ist das Spiel zu Ende.

fertig — bringt den Siegertext.

space — Da in Logo das Leerzeichen als Befehlsdelimiter dient, kann es nicht einfach ausgedruckt werden. Eine Liste etwa [], die ein Leerzeichen und sonst nichts enthält, gilt als leer. Deshalb muß der Umweg über den ASCII-Code des

Leerzeichens (32) gemacht werden.

Noch ein paar Bemerkungen zur Variablendefinition. In den Prozeduren »zugtest« und »listzug« werden teilweise dieselben Variablennamen benutzt. Hier kann man gut sehen, wie einfach die Variablenbenutzung unter Logo ist. Es handelt sich nämlich bei beiden trotz gleichen Namens um unterschiedliche Werte. Die doppelten Variablen sind als »local« definiert. Sie gelten also nur innerhalb derselben Prozedur. Namenskollisionen sind dadurch ausgeschlossen.

Aufgerufen wird das Spiel mit »TvH« und Enter. Der Rest erfolgt mit Benutzerführung. Es wird die Spielvariante abgefragt und danach die gewünschte Anzahl von Steinen. Bei der maximalen Zahl (neun) der Scheiben auf dem Stapel sind dabei 511 Züge nötig.

Der Streit um die »richtige« Programmiersprache ist so alt wie der Computer selbst. Eingefleischte Basic-Freaks werden beim Wort Logo nur mit den Schultern zucken, Assembler-Helden sowieso nur lächeln und souverän auf die überlegene Rechengeschwindigkeit hinweisen. Genauso werden überzeugte Logo-Fans bei einem Vergleich mit Basic reagieren, sind sie doch überzeugt, daß Logo immer das bessere und elegantere Konzept bieten wird.

Was läßt sich daraus folgern? Warum soll man Logo überhaupt einsetzen? Die Verwendung von Logo ist dann gegenüber Basic sinnvoll, wenn

— es sich um Programme handelt, die mit Listen arbeiten sollen oder eine listenorientierte Verarbeitung von Vorteil ist

— man die Rekursion in ihrer ganzen Mächtigkeit ausschöpfen kann

— die Prozedurendefinition von Vorteil ist. Das ist dann der Fall, wenn eine Reihe ähnlicher Probleme die Unterprogramme aus einem Pool verwenden sollen. Bei vernünftiger Programmierung kann man die Prozeduren ohne weitere Anpassung sofort in neue Programme übernehmen.

Logo wirft Probleme auf, wenn

- es sich um zeitaufwendige oder
- kritische Anwendungen handelt.
- der vorhandene Speicherplatz für eine Rekursion nicht ausreicht.

Eine einfache Antwort gibt es also nicht. Wenn Sie sich aber an das eine oder andere Programmierproblem aus der Vergangenheit erinnern, werden Ihnen sicher einige Aufgaben einfallen, die in Logo einfacher zu lösen gewesen wären.

So schreibt und ändert man Logo-Programme

Neuschreiben: Wenn das Fragezeichen am linken Bildschirmrand steht, dann kann man Befehle direkt eingeben. Durch »to« und den nachgestellten Prozedurrenamen wird die Definition einer Prozedur begonnen.

Ändern: Beim »?« wird mit »ed "Prozedurrename" « eine bereits definierte Prozedur zur Änderung aufgerufen. Da zur Editierung wenig im Handbuch steht, hier die wichtigsten Tasten:

Mit den Cursortasten kann man im Programmtext jede Stelle aufsuchen und beliebig ändern. Man muß generell eine geänderte Zeile bei Einfügungen neu schreiben (Ausnahme siehe Enter). Gelöscht werden kann dagegen ohne Probleme. Reichen die einfachen schrittweisen Bewegungen mit dem Cursor nicht aus, so kann man mit gleichzeitigem Drücken von Ctrl an den Prozeduranfang, beziehungsweise das -ende oder in der aktuellen

Zeile an Anfang (Ctrl+Clr links) oder Ende (Ctrl+Clr rechts) springen.

Clr und Del wirken wie in Basic. Mit CtrlK kann der Rest einer Programmzeile (ab dem Cursor nach rechts) gelöscht werden.

Enter teilt eine Zeile auf. Der nach dem Cursor stehende Rest der Zeile wird in die nächste Zeile übernommen, so daß man dann auf dem Rest der aktuellen Zeile neue Befehle einfügen kann.

Copy beendet die Editierung und gibt die geänderte Prozedur an den Interpreter-Speicher weiter. Hat man größere Änderungen vorgenommen und stellt vor dem Druck auf Copy fest, daß die alte Prozedur vielleicht doch die bessere war, so kann man mit Esc den Edit-Modus ohne Übernahme der geänderten Prozedur verlassen. In diesem Fall wird die geänderte Version allerdings gelöscht.

Logo in Stichworten

Ausgabemöglichkeiten:

pr = Basic-Befehl PRINT
type = dasselbe ohne Wagenrücklauf (wie PRINT +";")
show = Ausdruck einer Liste

Eingabemöglichkeiten:

rl = Read List, Eingabe von Zahlen und Zeichen (ist mit Enter abzuschließen $\hat{=}$ INPUT in Basic)
rc = Read Character 1,
Zeichen ohne Enter einlesen $\hat{=}$ INKEY in Basic
mit make " vom Variablennamen ohne Doppelpunkt gefolgt, oder bei einem Prozeduraufruf an die nachgestellten Variablen. Im Beispiel »quadrat« wurde so k auf den Eingabewert gesetzt.
Format if <Bedingung> [then-Teil in eckigen Klammern] [else-Teil in eckigen Klammern]

Wertzuweisungen:

Listenverarbeitung:

Befehle, die Listen unterteilen, Daten in Listen einlesen, Elemente aus einer Liste lösen und so weiter.

Arithmetik:

+, -, *, / Dr. Logo kennt weder Wurzel noch Potenzen als Primitives. Diese müssen also vom Benutzer bei Bedarf erst noch geschrieben werden.

Turtle-Grafik und Sound bieten im Sound- beziehungsweise Grafikbereich dieselben Möglichkeiten wie Basic. Daneben existieren noch Kommandos zum Zugriff auf die Peripherie und den Speicher (Äquivalente zu PEEK und POKE).

Schachtelung

TvH	simul selbst	turm turmliste ziehen	ausgabe	
	space		zugtest listzug ende	err cut fertig

So wirken die Programme aufeinander

```
to quadrat :k
repeat 4 [fd :k rt 90]
end
```

Listing 1. Mit dieser Prozedur malen Sie ein Quadrat

```
to TvH
ts ct space 5
pr [** Tuerme von Hanoi **]
repeat 4[pr "]
pr [(1) - Computer-Simulation]
pr [(2) - Selbstversuch]
repeat 3[pr "]
type [Waelhlen Sie bitte] space 2
make "wahl rc if or (:wahl<0) (:wahl>2) [TvH]
make "i 0
run item wahl [[simul] [selbst]]
end

to simul
ct
space 5 pr [** Tuerme von Hanoi **]
space 3 pr [** Computer - Simulation **]
pr " pr "
type [Wieviele Steine? (max. 9)]
space 2 make "st first rl pr "
if :st > 9 [simul]
turm :st 1 2 3
pr " pr [Druecken Sie eine Taste.]
make "st rc TvH
end

to turm :m :1 :2 :3
if :m > 1 [turm :m - 1 :1 :3 :2]
ausgabe :m :1 :2
if :m > 1 [turm :m - 1 :3 :2 :1]
end

to ausgabe :s :a :b
make "i :1 + 1
type [Stein] space 1 type :s space :2
type [von Turm] space 1 type :a space 2
type [nach Turm] space 1 type :b space 2
type [()] type :i pr [()]
end

to selbst
ct
space 5 pr [** Tuerme von Hanoi **]
space 6 pr [** Selbstversuch **]
pr " pr "
type [Wieviele Steine ? (max. 9)]
space 2 make "st first rl pr "
if :st > 9 [selbst]
make "t1 []
turmliste :st
make "t [[]] make "t fput :t1 :t
ziehen :i
end

to turmliste :t
if :t = 0 [stop]
make "t1 fput :t :t1
turmliste :t - 1
end

to ziehen :i
make "i :i + 1
type [Stein] space 1 type [?]
make "st rc type char 8
type :st space 2
```

```
type [von Turm] space 1 type [?]
make "vt rc type char 8
type :vt space 2
type [nach Turm] space 1 type [?]
make "nt rc type char 8
type :nt space 2 type "( type :i pr ")
zugtest :st :vt :nt :i
listzug :vt :nt
ende
ziehen :i
end
```

```
to zugtest :s :v :n :i
(local "vt "nt "st)
make "vt item :v :t
make "nt item :n :t
if empty? :vt [err :i]
make "st first :vt
if not (:s = :st) [err :i]
if empty? :nt [stop]
if "st > first :nt [err :i]
end
```

```
to err :i
pr " space 4 pr [** Fehlerhafte Eingabe! **]
pr "
ziehen :i - 1
end
```

```
to listzug :v :n
(local "vt "nt "st)
make "vt item :v :t
make "nt item :n :t
make "st first :vt
make "vt bf :vt
make "nt fput :st :nt
cut :v :vt cut :n :nt
end
```

```
to fertig
pr " pr "
pr [Gratuliere!]
pr [Sie haben es geschafft.]
pr " pr [Druecken Sie eine Taste.]
make "st rc TvH
```

```
to space :sp
repeat :sp [type char 32]
end
```

```
to ende
if not (item 1 :t = []) [stop]
if or (item 2 :t = []) (item 3 :t = []) [Fertig]
end
```

```
to cut :c :ct
if :c = 1 [make "t fput :ct (bf :t) stop]
if :c = 3 [make "ct bl list :ct [] make "t
se (bl :t) :ct stop]
make "t1 first :t
make "t fput :ct bf bf :t
make "t fput :t1 :t
end
```

Listing 2. Die »Türme von Hanoi« bestehen aus 13 Prozeduren

Nach den etwas ungewohnten Logo-Prozeduren finden Sie nun zum Vergleich einen Vorschlag, der das Turmproblem in Basic löst. Am Anfang steht ein kurzer Initialisierungsteil, der drei Windows definiert und die Grenzen des Spiels festlegt. Der Ausgangsturm kann aus drei bis sieben Scheiben bestehen, wobei die kleinsten Scheiben zuoberst liegen. Die vom Computer durchgeführte Demonstration der optimalen Zugfolge kann ohne Verzögerung oder in Zeitlupe verfolgt werden.

Im letzten Fall (z\$="2") schiebt der Schneider nach jedem Zug eine Zeitschleife (Zeile 470) ein. Nach den Anfangsabfragen folgt jedoch erst eine Routine, die Anfangsbedingungen für die Grafik festlegt. Das Basic-Programm stellt nach jedem Zug den Zustand aller drei Türme mit Hilfe von Scheiben, die aus dem Stern (»*)« aufgebaut sind, dar. Der Inhalt aller drei Türme ist dabei in Form der Sternchenstrings in dem

Array a\$ gespeichert. Den Aufbau der Scheiben auf den verschiedenen Ebenen leistet eine einzige Zeile (Zeile 260). Um bei einer Verschiebung Scheiben löschen zu können, wird dann noch der String bl\$ benötigt. In Zeile 280 wird nun das Array, genauer der erste Turm, mit den in br\$ gespeicherten Zeilenstrings belegt, wonach die Türme durch den Aufruf der Unterroutine »Türme setzen« ausgegeben werden. Nun beginnt die eigentliche Simulation. Mit Hilfe einer Rekursion, die in Basic nur schwierig und aufwendig zu realisieren ist, werden die Züge bestimmt. Die eigentliche Zugbestimmung finden Sie dabei ab Zeile 850.

Dazu ist gegebenenfalls ein mehrfacher Aufruf dieses Programmteils durch sich selbst nötig. Dieser Rekursionsaufruf steht in Zeile 900. Nachdem der optimale Zug aus gesucht wurde, muß er nun noch ausgeführt werden. Dazu dient die Routine ab Zeile 520. Hier werden die

Verschiebungen in a\$ gemacht und danach das komplette Array durch Aufruf von Zeile 340 dargestellt. Dieser Programmablauf wiederholt sich so lange, bis der Zielturm fertig aufgebaut ist. In diesem Fall muß in der obersten Schicht des Turmes wieder die kleinste Scheibe eingetragen sein. Der Test, der das prüft, ist in Zeile 730. Falls der Turm komplett umgesetzt wurde, geht es hier in die Endabfrage ab Zeile 1050.

Als Vergleich zum Logo-Programm sollten Sie einmal die Prozedur Turmliste mit dem Rekursionsteil ab Zeile 850 vergleichen. Hier zeigen sich die Stärken von Logo als Programmiersprache: Eine sehr viel kürzere Befehlsfolge und eine klare Struktur. Aber auch das restliche Basic-Programm ist im Gegensatz zu Logo weniger klar und durchsichtig programmiert. Bei Logo dagegen sorgt das Denken in Prozeduren für Übersicht.

(Carsten Strauß)

```

10 ***** [3988]
20 ** Tueme von Hanoi ** [CE0A]
30 ** fuer Schneider ** [5F72]
40 ** unter Basic 1.0 ** [E924]
50 ***** [4590]
60 INK 0,0:INK 1,6:INK 2,26:INK 3,11 [D2C4]
70 PEN 1 [6076]
80 CLS [F8D4]
90 PRINT:PRINT" {3 SPACE}Tueme von Hanoi
   BASIC-Simulation" [5892]
100 WINDOW#3,1,40,6,25 [5982]
110 LOCATE 10,7 [E79C]
120 PRINT"Computersimulation (1)" [D274]
130 LOCATE 10,9 [95A4]
140 PRINT"Zeitlupe(2)" [3C76]
150 z$=INKEY$:IF z$<>"1" AND z$<>"2" THE
   N 150 [7EA2]
160 PRINT:PRINT:PRINT"Anzahl der Scheibe
   n auf dem Ausgangsturm(3-7)" [A296]
170 y$=INKEY$:IF y$="" THEN 170 ELSE IF
   ASC(y$)<51 OR ASC(y$)>55 THEN 170 EL
   SE mz=VAL(y$) [44EC]
180 CLS#3 [D3E2]
190 PEN 2 [34DE]
200 WINDOW#1,19,25,8,8 [EF9E]
210 IF z$="2" THEN LOCATE 16,5:PRINT"Zei
   tlupe" [BD3E]
220 ***** [6A3A]
230 ** Turmscheiben definieren ** [94CC]
240 ***** [5A3E]
250 DIM a$(7,7) [AO42]
260 FOR i=1 TO 7:br$(8-i)=SPACE$(7-i)+ST
   RING$(2*(i-1)+1,"*"):NEXT [AD8C]
270 bl$=SPACE$(13) [B25A]
280 FOR i=8-mz TO 7:a$(i,1)=br$(i):a$(i,
   2)=bl$:a$(i,3)=bl$:NEXT [C730]
290 GOSUB 340 [4BE4]
300 GOTO 750 [CD50]
310 ***** [FCF2]
320 ** Tueme setzen ** [D9DE]
330 ***** [C4F6]
340 i=1 [401C]
350 CLS#1:PRINT#1,rz [A942]
360 FOR x=20 TO 14 STEP-1 [BDB2]
370 LOCATE 1,x:PRINT a$(i,1); [1B4A]
380 i=i+1:NEXT [C93E]
390 i=1 [4326]
400 FOR x=20 TO 14 STEP-1 [1B8B]
410 LOCATE 14,x:PRINT a$(i,2); [FBAA]
420 i=i+1:NEXT [2F74]
430 i=1 [OD1C]
440 FOR x=20 TO 14 STEP-1 [E7B0]
450 LOCATE 27,x:PRINT a$(i,3); [D3BC]
460 i=i+1:NEXT [437C]
470 IF z$="2" THEN FOR t=1 TO 1000:NEXT
   t [F188]
480 RETURN [1D38]
490 ***** [8154]
500 ** Turmarray aendern ** [9810]
510 ***** [7246]
520 FOR i=7 TO 1 STEP-1 [7112]
530 IF a$(i,a)<>bl$ THEN 560 [8D52]
540 NEXT i [4602]
550 cc=0:RETURN [6C0E]
560 FOR j=1 TO 7 [326B]
570 IF a$(j,b)=bl$ THEN 590 [ABEA]
580 NEXT j [A00C]
590 FOR k=1 TO 7 [4370]
600 IF a$(i,a)=br$(k) THEN 620 [7652]
610 NEXT k [1502]
620 FOR l=1 TO 7 [3166]
630 IF a$(j-1,b)=br$(l) THEN 660 [A722]
640 NEXT l [4D0A]
650 l=1 [E12A]
660 IF k<l THEN PRINT CHR$(7):RETURN [897A]
670 a$(j,b)=a$(i,a) [A12B]
680 a$(i,a)=bl$ [3872]
690 ***** [2AAB]
700 ** ausgeben und fertig ? ** [FOEE]
710 ***** [1F9A]
720 GOSUB 340 [F6E0]
730 IF a$(7,3)=br$(7) THEN 1050 [9DBE]
740 RETURN [AC36]
750 dn=mz [EC64]
760 t1(1)=t2(1)=3:t3(1)=2 [37EC]
770 ne=1 [5CFE]
780 dd(1)=dn [C730]
790 rz=0 [8132]
800 GOSUB 850 [81EA]
810 END [BC20]
820 ***** [1352]
830 ** Rekursionsteil ** [855A]
840 ***** [2156]
850 ne=ne+1 [02F8]
860 dd(ne)=dd(ne-1)-1 [161E]
870 t1(ne)=t1(ne-1) [58DB]
880 t2(ne)=t3(ne-1) [48E0]
890 t3(ne)=t2(ne-1) [48E2]
900 IF dd(ne)=1 THEN rz=rz+1:a=t1(ne):b=
   t2(ne):GOSUB 980 ELSE GOSUB 850 [514E]
910 rz=rz+1:a=t1(ne-1):b=t2(ne-1):GOSUB
   980 [C74C]
920 t1(ne)=t3(ne-1) [CDD4]
930 t2(ne)=t2(ne-1) [8BD6]
940 t3(ne)=t1(ne-1) [71DB]
950 IF dd(ne)=1 THEN rz=rz+1:a=t1(ne):b=
   t2(ne):GOSUB 980 ELSE GOSUB 850 [A718]
960 ne=ne-1 [1400]
970 RETURN [2040]
980 [E6D0]
990 GOSUB 520 [9DF2]
1000 RETURN [9982]
1010 FOR i=7 TO 1 STEP -1 [FB68]
1020 IF a$(i,a)<>bl$ THEN RETURN [AC32]
1030 NEXT i [A358]
1040 cc=0:RETURN [7764]
1050 LOCATE 10,25:PRINT"weiter mit Taste
   ndruck" [11BE]
1060 z$=INKEY$:IF z$="" THEN 1060 [4E48]
1070 RUN [01BA]

```

Listing. »Türme von Hanoi« ist in Basic bedeutend umständlicher als in Logo



C. Strauß/H. Pick
CPC 464 für Ein- und Umsteiger
 Februar 1985, 260 Seiten
 Über die hervorragende Qualität des CPC 464 sind sich alle Experten einig. Aber der Computerneuling wie auch der alte Hase, der bereits ein anderes System kennt, benötigt Hilfe, die ihm dieses Buch gibt. Eine BASIC-Einführung (an den Beispielen Dateiverwaltung und Textverarbeitung) ist selbstverständlich. Darüber hinaus konzentrieren sich die Autoren auf die besonderen Eigenschaften des 464ers, seine mächtigen Befehle, die andere BASICs ganz schön alt aussehen lassen und seine begeisternden Grafik- und Musikmöglichkeiten. Ein eigenes Kapitel ist der Echtzeitverarbeitung gewidmet.
Best-Nr. MT 801
ISBN 3-89090-090-9 DM 46,-



C. Strauß
CPC 464 Programmieren in Maschinensprache
 Juli 1985, 276 Seiten
 Dieses Buch weilt in die Arbeitsweise des BASIC-Interpreters ein und erklärt die Funktionsweise der Bauteile des Geräts und deren Zusammenwirken. So ergeben sich auch für reine BASIC-Programmierer bereits viele Änderungen- und Eingriffsmöglichkeiten in die Maschine. Die Einführung in die Z80-Maschinensprache ist speziell auf den CPC464 zugeschnitten und ermöglicht tiefe Einblicke in den Aufbau des Betriebssystems. Das Buch macht Sie autonom: ein Monitor- und Disassemblierprogramm wird entwickelt, wodurch Sie nicht nur die Funktionsweise dieser wichtigen Programmierhilfen kennenlernen, sondern auch alle nötigen Werkzeuge für die Arbeit mit dem Buch an der Hand haben.
Best-Nr. MT 829
ISBN 3-89090-166-2 DM 46,-

Markt & Technik-Fachbücher erhalten Sie bei Ihrem Buchhändler.

Markt & Technik
BUCHVERLAG

Hans-Pinsel-Str. 2, 8013 Haar bei München

Depot-Händler

Tragen Sie Ihre Buchbestellung auf eine Postkarte ein und schicken diese an einen Depothändler in Ihrer Nähe oder an Ihren Buchhändler.

Buchhandlung Harder, Kurfürstendamm 69
 1000 Berlin 15, Tel. (030) 8835002,
 BTX: +921782 #
 Computare Fachbuchhandlung, Keithstraße 18
 1000 Berlin 30, Tel. (030) 2139021
 Thalia Buchhaus, Große Bleichen 19
 2000 Hamburg 36, Tel. (040) 3005060
 Boysen + Maasch, Hermannstraße 31
 2000 Hamburg 1, Tel. (040) 30050515
 Electro-Data, Wilhelm-Heldsiek-Straße 1
 2100 Cuxhaven, Tel. (04721) 51288
 Buchhandlung Muehlau, Holtzauer Straße 116
 2300 Kiel, Tel. (0431) 85085
 ECL, Nordstraße 94-98
 2390 Flensburg, Tel. (0461) 28181
 Buchhandlung Weiland, Königstraße 79
 2400 Lübeck, Tel. (0451) 74006-09
 Buchhandlung Storm, Langenstraße 10
 2800 Bremen 1, Tel. (0421) 321523
 Buchhandlung Lohse-Eissing, Marktstraße 38
 2940 Wilhelmshaven, Tel. (04421) 41687
 Buchhandlung Schmorl u. v. Seefeld, Bahnhofstraße 13
 3000 Hannover 1, Tel. (0511) 327651
 Buchhandlung Graff, Neue Straße 23
 3300 Braunschweig, Tel. (0531) 49271
 Deuerlich'sche Buchhandlung, Weender Straße 33
 3400 Göttingen, Tel. (0521) 58866
 Buchhandlung an der Hochschule, Holländische Straße 22
 3500 Kassel, Tel. (0561) 83807
 Stern Verlag, Friedrichstraße 24-28
 4000 Düsseldorf, Tel. (0211) 373033
 Buchhandlung Baedeker, Kettwiger Straße 33-35
 4300 Essen 1, Tel. (0201) 221381
 Regensberg'sche Buchhandlung, Alter Steinweg 1
 4400 Münster, Tel. (0251) 40541-5
 Buchhandlung Acker, Johannisstraße 51
 4500 Osnabrück, Tel. (0541) 28488
 Buchhandlung Lensing, Westenhellweg 86-88
 4600 Dortmund, Tel. (0231) 169805
 Buchhandlung Brockmeyer, Querenburger Höhe 281/Unicenter
 4630 Bochum, Tel. (0234) 701360
 Buchhandlung Meier + Weber, Warburger Straße 98
 4790 Paderborn, Tel. (05251) 63172
 Buchhandlung Phonix GmbH, Oberntorwall 25
 4800 Bielefeld 1, Tel. (0521) 69071
 Buchhandlung Gonski, Neumarkt 24
 5000 Köln 1, Tel. (0221) 210528
 Mayer'sche Buchhandlung, Ursulinerstraße 17-19
 5100 Aachen, Tel. (0241) 48142
 Buchhandlung Behndt, Am Hof 5a
 5300 Bonn 1, Tel. (0228) 858021
 Buchhandlung Cusanus, Schloßstraße 12
 5400 Koblenz, Tel. (0261) 36239
 Akad. Buchhandlung Interbook, Fleischstraße 61-65
 5500 Trier, Tel. (0651) 43596
 Buchhandlung W. Finke, Kipdorf 32
 5600 Wuppertal 1, Tel. (0202) 454220
 Buchhandlung Balogh, Sandstraße 1
 5900 Siegen, Tel. (0271) 55298-9
 Buchhandlung Naacher, Steinweg 3
 6000 Frankfurt 1, Tel. (069) 298050
 Buchhandlung Wellnitz, Lautenschlagerstraße 4
 6100 Darmstadt, Tel. (06151) 76545
 Buchhandlung Feller + Gecks, Friedrichstraße 31
 6200 Wiesbaden, Tel. (06121) 304911
 Ferber'sche UNI-Buchhandlung, Seltenweg 83
 6300 Gießen, Tel. (0641) 12001
 Sozialwissenschaftliche Fachbuchhandlung, Friedrichstraße 24
 6400 Fulda, Tel. (0661) 75077
 Gutenberg Buchhandlung, Große Bleiche 29
 6500 Mainz, Tel. (0631) 37011
 Buchhandlung Bock + Seip, Futterstraße 2
 6600 Saarbrücken, Tel. (0681) 30677
 Buchhandlung Wilhelm Hofmann, Bismarckstraße 98
 6700 Ludwigshafen, Tel. (0621) 518001
 Buchhandlung Loeffler, S. 15
 6800 Mannheim 1, Tel. (0621) 28912
 Buchhandlung Stehn, Bahnhofstraße 13
 7000 Stuttgart 50, Tel. (0711) 561476
 Buchhandlung am Markt, Kramstraße 6
 7100 Heilbronn, Tel. (07131) 68882
 PCB Micro-Computer, Oskar-Kalb-Platz 8
 7410 Reutlingen, Tel. (07121) 270443
 UNI Buchhandlung Kallner + Moessner, Kaiserstraße 18
 7500 Karlsruhe, Tel. (0721) 691436
 Buchhandlung Roth, Hauptstraße 45
 7600 Offenburg, Tel. (0781) 22097
 Rombach Center, Bertholdstraße 10
 7800 Freiburg, Tel. (0761) 49091
 Fachbuchhandlung Hofmann, Hirschstraße 4
 7900 Ulm, Tel. (0731) 60949
 Schautlas Elektronik, Bachstraße 52
 7980 Ravensburg, Tel. (0751) 26138
 Buchhandlung Hugendubel, Marienplatz
 8000 München 2, Tel. (089) 2389-9
 Computerbücher am Obelisk, Barerstraße 32-34
 8000 München 2, Tel. (089) 282383
 Pele's Computerbücher, Schillerstraße 17
 8000 München 2, Tel. (089) 555229
 Universitätsbuchhandlung Lachner, Theresienstraße 43
 8000 München 2, Tel. (089) 521340
 Buchhandlung Schönhuber, Theresienstraße 6
 8070 Ingolstadt, Tel. (0841) 33146/47
 Computerstudie Gertrud Friedrich, Ludwigstraße 3
 8220 Traunstein, Tel. (0891) 14767
 Buchhandlung Puster, Ki. Exerzierplatz 4
 8390 Passau, Tel. (0851) 56945
 Buchhandlung Puster, Gesandtenstraße 6
 8400 Regensburg, Tel. (0941) 53061
 Buchhandlung Dr. Büttner, Adlerstraße 10-12
 8500 Nürnberg, Tel. (0911) 232318
 STB Computer Vertrieb, Werner-Siemens-Straße 19
 8580 Bayreuth, Tel. (0921) 62320
 Computer-Center-Burger, Leimitzer Straße 11-13
 8670 Hof, Tel. (09281) 40076
 Sortiments- u. Bahnhofsbuchh. J. Strykowski, Bahnhofplatz 4
 8700 Würzburg, Tel. (0931) 54389
 Buchhandlung Puster, Grottenau 4
 8900 Augsburg, Tel. (0821) 35437
 Kempfner Fachsortiment, Salzstraße 30
 8960 Kempten, Tel. (0831) 14413
 Belgien:
 Eicher Micro & Personal Computer, Hünningen 56-58
 B-4780 St. Vith, Tel. (080) 227393
 Luxemburg:
 Librairie Promoculture, 14, rue Duchesne (Pl. de Paris)
 L-1011 Luxembourg-Gare, Tel. 480691, Telex 3112
 Schweiz:
 Buchhandlung Meissner, Bahnhofstraße 41
 5000 Aarau, Tel. (064) 247151
 Bücher Balmer, Neugasse 12
 6300 Zug, Tel. (042) 214141
 Buchhandlung Enge, Bleicherweg 58
 8002 Zürich, Tel. (01) 2012078
 Buchhandlung Orali Füssli, Pelikanstraße 10
 8022 Zürich, Tel. (01) 2118011
 Freihofer AG, Wissenschaftliche Buchhandlung, Universitätsstr. 11
 8033 Zürich, Tel. (01) 3834282
 Buchhandlung am Rössli, Webergasse 5
 9001 St. Gallen, Tel. (071) 228726

Markt & Technik
BUCHVERLAG

Impressum

Herausgeber: Carl-Franz von Quadt, Otmar Weber
Chefredakteur: Michael Scharfenberger (sc)
Leitender Redakteur: Michael Lang (lg)
Redakteure: Andreas Hagedorn (hg; Inhalt)
 Petra Wängler (wg; Koordination)
Redaktionsassistent: Monika Lewandowski (222)
Fotografie: Jens Jancke
Layout: Leo Eder (ltg.)

Auslandsrepräsentation:
Schweiz: Markt & Technik Vertriebs AG,
 Kollerstrasse 3, CH-6300 Zug,
 Tel. 042-415656, Telex: 862329 mut ch
USA: M & T Publishing, 2464 Embarcadero Way, Palo
 Alto, CA 94303; Tel. 415-424-0600; Telex 752351

Manuskripteinsendungen: Manuskripte und Programmings werden gerne von der Redaktion angenommen. Sie müssen frei sein von Rechten Dritter. Sollten sie auch an anderer Stelle zur Veröffentlichung oder gewerblichen Nutzung angeboten worden sein, muß dies angegeben werden. Mit der Einsendung von Manuskripten und Listings gibt der Verfasser die Zustimmung zum Abdruck in von der Markt & Technik Verlags AG herausgegebenen Publikationen und zur Vervielfältigung der Programmings auf Datenträger. Mit der Einsendung von Bauanleitungen gibt der Einsender die Zustimmung zum Abdruck in von Markt & Technik Verlag AG verlegten Publikationen und dazu, daß Markt & Technik Verlag AG Geräte und Bauteile nach der Bauanleitung herstellen läßt und vertreibt oder durch Dritte vertreiben läßt. Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte und Listings wird keine Haftung übernommen.

Produktionsleitung: Klaus Buck (180)
Anzeigenverkauf: Brigitta Fiebig (211)
Anzeigenverwaltung und Disposition:
 Patricia Schiede (172)
Vertriebsmarketing: Hans Hörll (114)
Vertriebsleitung: Helmut Grünfeldt (189)
Verlagsleiter M & T Buchverlag: Günther Frank (212)
Vertrieb Handelsauflage: Inland (Groß-, Einzel- und Bahnhofsbuchhandel) sowie Österreich und Schweiz:
 Pegasus Buch- und Zeitschriften-Vertriebs GmbH,
 Hauptstätter Str. 96, 7000 Stuttgart 1, Tel. (0711) 6483-0
Bezugsmöglichkeiten: Leser-Service: Telefon 089/4613-249. Bestellungen nimmt der Verlag oder jede Buchhandlung entgegen.

Bezugspreis: Das Einzelheft kostet DM 14,-.
Druck: R. Oldenbourg GmbH, Hürderstraße 4, 8011 Kirchheim.
Urheberrecht: Alle im Schneider-Sonderheft erscheinenden Beiträge sind urheberrechtlich geschützt. Alle Rechte, auch Übersetzungen, vorbehalten. Reproduktionen gleich welcher Art, ob Fotokopie, Mikrofilm oder Erfassung in Datenverarbeitungsanlagen, nur mit schriftlicher Genehmigung des Verlages. Anfragen sind an Michael Scharfenberger zu richten. Für Schaltungen, Bauanleitungen und Programme, die als Beispiele veröffentlicht werden, können wir weder Gewähr noch irgendwelche Haftung übernehmen. Aus der Veröffentlichung kann nicht geschlossen werden, daß die beschriebenen Lösungen oder verwendeten Bezeichnungen frei von gewerblichen Schutzrechten sind. Anfragen für Sonderdrucke sind an Peter Wagstyl zu richten.

© 1986 Markt & Technik Verlag Aktiengesellschaft, Redaktion »Happy-Computer«.

Verantwortlich: Für redaktionellen Teil:
 Michael Scharfenberger
 Für Anzeigen: Ralph Peter Rauchfuß (126).
Vorstand: Carl-Franz von Quadt, Otmar Weber
Anschrift für Verlag, Redaktion, Vertrieb, Anzeigenverwaltung und alle Verantwortlichen:
 Markt & Technik Verlag Aktiengesellschaft,
 Hans-Pinsel-Str. 2, 8013 Haar bei München,
 Telefon 089/4613-0, Telex 5-22052

Telefon-Durchwahl im Verlag:
 Wählen Sie direkt: Per Durchwahl erreichen Sie alle Abteilungen direkt. Sie wählen 089-4613 und dann die Nummer, die in Klammern hinter dem jeweiligen Namen angegeben ist.

Aktionäre, die mehr als 25% des Kapitals halten:
 Otmar Weber, Ingenieur, München; Carl-Franz von Quadt, Betriebswirt, München; Aufsichtsrat: Dr. Robert Dissmann (Vorsitzender), Karl-Heinz Fanselow, Eduard Heilmayr

COMPUTER-ZEITSCHRIFTEN

VON PROFIS FÜR PROFIS

COMPUTER PERSÖNLICH

Das aktuelle Fachmagazin für Personal-Computer.

- ★ Wenn Sie jetzt den Schritt vom Heim-Computer zur professionellen Anwendung eines Personal Computers planen
- ★ Wenn Sie beruflich oder privat bereits einen Personal Computer benutzen
- ★ Wenn Sie regelmäßig Informationen über das aktuelle Produktangebot benötigen
- ★ Wenn Sie selbst programmieren
- ★ Wenn Sie professionelle Hard- und Softwaretests suchen
- ★ Wenn Sie Ihr eigenes System möglichst effizient einsetzen wollen

dann ist »Computer persönlich«, das aktuelle Fachmagazin für Personal Computer, genau Ihre Zeitschrift.

Die konsequente Ausrichtung auf professionelle Anwendungen bietet Ihnen alle wichtigen Informationen.

Von Profis für Profis!

»Computer persönlich« gibt es alle 14 Tage neu bei Ihrem Zeitschriftenhändler oder im Computer-Fachgeschäft.

PC MAGAZIN

Einzig Wochenzeitung für Personal Computer im IBM-Standard.

Sie beschäftigen sich beruflich oder privat mit dem Einsatz und der Anwendung von Personal Computern?

Sie sind an aktuellen, professionellen Informationen über IBM-PCs, kompatible Systeme und deren professionellen Einsatz interessiert? Dann ist das PC Magazin genau auf Ihre persönlichen Bedürfnisse zugeschnitten.

Es wird von anerkannten und erfahrenen Fachjournalisten für professionelle Anwender und Fachleute geschrieben.

Es berichtet jede Woche ausschließlich über Computer im IBM-Standard und kompatible Systeme, über Hard- und Softwareneuheiten. Es bringt ausführliche Testberichte und gibt Ihnen wichtige Informationen über Netzwerke sowie die PC/Host-Verbindung.

Nur diese Spezialisierung ermöglicht eine gezielte Berichterstattung und bietet genügend Raum, um auf Anwenderprobleme spezifisch eingehen zu können.

Von Profis für Profis!

Und das jeden Mittwoch neu bei Ihrem Zeitschriftenhändler oder im Computer-Fachgeschäft.

GUTSCHEIN

für ein kostenloses Probeexemplar

Senden Sie mir die neueste Ausgabe der von mir angekreuzten Zeitschrift kostenlos als Probeexemplar:

COMPUTER PERSÖNLICH

Wenn mir Computer persönlich zusagt und ich es regelmäßig weiterbeziehen möchte, brauche ich nichts zu tun: Ich erhalte Computer persönlich dann regelmäßig alle 14 Tage per Post frei Haus geliefert und bezahle pro Jahr nur DM 98,—. Zustellung und Postgebühren übernimmt der Verlag.

PC-MAGAZIN

Wenn mir das PC-Magazin zusagt und ich es regelmäßig weiterbeziehen möchte, brauche ich nichts zu tun: Ich erhalte mein PC-Magazin dann regelmäßig jede Woche per Post frei Haus geliefert und bezahle pro Jahr nur DM 155,—. Zustellung und Postgebühren übernimmt der Verlag.

Vorname/Name _____

Straße _____ PLZ/Ort _____

Datum _____ 1. Unterschrift _____

Mir ist bekannt, daß ich diese Bestellung innerhalb von 8 Tagen bei der Bestelladresse widerrufen kann und bestätige dies durch meine zweite Unterschrift. Zur Wahrung der Frist genügt die rechtzeitige Absendung des Widerrufs.

Datum _____ 2. Unterschrift _____

Bücher zu Schneider CPCs

J. Hückstädt
CP/M 2.2 Anwenderhandbuch
CPC 464/664/6128
Dezember 1985, ca. 250 Seiten

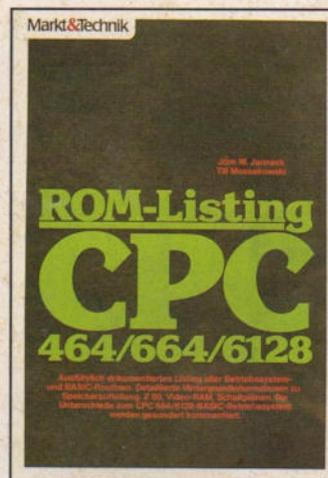
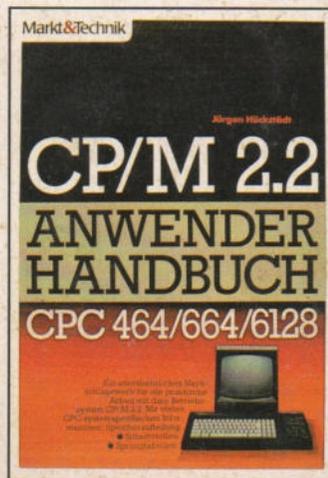
Wenn Sie glücklicher Besitzer eines Schneider-Computers sind und mehr wissen wollen über das leistungsstarke Betriebssystem CP/M 2.2, dann ist dieses Buch genau das Richtige für Sie! Es behandelt CP/M 2.2 nicht nur in seiner allgemeinen Form, wie sie für sämtliche CP/M-Computer gültig ist, sondern bezieht auch die Hardware der CPC-Computer mit ein.

Best.-Nr. MT 859
ISBN 3-89090-204-9
DM 46,-/sFr. 42,30/6S 358,80

J. Hückstädt
CP/M Plus Anwenderhandbuch
CPC 6128
Dezember 1985, ca. 250 Seiten

Ein unentbehrliches Nachschlagewerk für die praktische Arbeit mit CP/M-Plus und seinen Hilfsprogrammen. Mit zahlreichen Beispielen.

Best.-Nr. MT 627
ISBN 3-89090-2
DM 46,-/sFr. 42,30/6S 358,80



T. Mossakowski/J. Janneck
ROM-Listing CPC 464/664/6128
Dezember 1985, ca. 450 Seiten

Dieses Buch enthält in konzentrierter Form umfassende Informationen über den Aufbau Ihres Computers. Es kann sich daher schnell zu einem unentbehrlichen Arbeitsbuch für die Programmierung entwickeln. Um es optimal nutzen zu können, sollte man mit dem Schneider-BASIC vertraut sein und erste Erfahrungen in der Maschinensprache des Z80 besitzen. Zu jeder Routine im Listing sind die Übergabeparameter aufgeführt. Verschiedene Tabellen erleichtern das Auffinden einer bestimmten Routine.

Best.-Nr. MT 711
ISBN 3-89090-134-4
DM 64,-/sFr. 58,90/6S 499,20

Th. Erpel
CPC BASIC-Kurs
November 1985, 376 Seiten

Ein Buch für den Einstieg in die Bedienung und Programmierung der Schneider-Computer.

Best.-Nr. MT 828
ISBN 3-89090-167-0
DM 46,-/sFr. 42,30/6S 358,80



C. Strauch
Schneider CPC Grafik-Programmierung
Dezember 1985, ca. 300 S.

Dieses Buch wendet sich an die Schneider CPC-Besitzer, die alles über die Grafikfähigkeiten ihres Computers wissen wollen. Es bietet einen umfassenden Überblick über die verschiedenen Anwendungsbereiche der Grafikprogrammierung: zwei- und dreidimensionale Diagrammdarstellungen, Definition und Bewegung von Sprites, Entwurf von Titelgrafiken, Einsatz der Grafik bei der Unterstützung anderer Programme.

• Besonders interessant: ein Sprite-Generator, ein Mailprogramm für hochauflösende Grafik, ein Programm zur Erstellung von Titelgrafiken sowie ein universelles Darstellungsprogramm.
Best.-Nr. MT 782
ISBN 3-89090-182-4
DM 46,-/sFr. 42,30/6S 358,80



J. Hückstädt
Der Schneider CPC 6128
September 1985, 273 Seiten

Dieses Buch ist für jeden CPC 6128-Besitzer eine wertvolle Hilfe, die vielfachen Möglichkeiten dieses bisher einmaligen Computers kennenzulernen und anzuwenden. Der Computerneuling wird Schritt für Schritt in den Umgang mit dem Computer und die BASIC-Programmierung eingeführt, bis er alle notwendigen Kenntnisse besitzt, die mancher Profi bereits mitbringt. Aber an dieser Stelle wird das Programmieren mit dem CPC 6128 erst interessant, nämlich dann, wenn es darum geht, eine eigene Dateiverwaltung aufzubauen oder Grafik und Sound zu programmieren. Weiterhin erfahren Sie alles über CP/M Plus auf dem CPC 6128.

Best.-Nr. MT 849
ISBN 3-89090-192-1
DM 46,-/sFr. 42,30/6S 358,80



C. Strauch
DR. LOGO auf dem Schneider CPC
Januar 1986, ca. 250 Seiten

Speziell auf die Schneider Computer anwendbar finden Sie in diesem Buch eine strukturierte Anleitung für die praktische Arbeit mit der Programmiersprache LOGO. Mit zahlreichen Beispielen zur Grafik- und Soundprogrammierung. Das letzte Kapitel enthält nützliche Utilities (z.B. SORT-Routinen). Informationen über die Aufteilung des Speichers (Speicheranalyse und Tastendefinition), Erklärungen zu den Editorkommandos über deutsche LOGO-Befehle sowie Lösungsvorschläge zu den Aufgaben.
Best.-Nr. MT 865
ISBN 3-89090-210-3
DM 46,-/sFr. 42,30/6S 358,80



H. Tischer
Programm-entwicklung unter CP/M 2.2 auf dem CPC 464/664
Dezember 1985, ca. 250 S.

Dieses Buch vermittelt alle Informationen, die zum selbstständigen Entwickeln von CP/M 2.2-Programmen nötig sind. Besprochen wird sowohl die grundlegende Funktionsweise des CP/M Betriebssystemes als auch alle dem Anwender schon zur Verfügung stehenden Systemroutinen, die diesem viel Arbeit ersparen. Zwei Kapitel beschäftigen sich ausschließlich mit den zusätzlichen Möglichkeiten, die nur die Computer CPC 464/664 bieten.

Kenntnisse der 8080- oder Z80-Assemblersprache sind erforderlich.
Best.-Nr. MT 864
ISBN 3-89090-209-X
DM 52,-/sFr. 47,80/6S 405,60



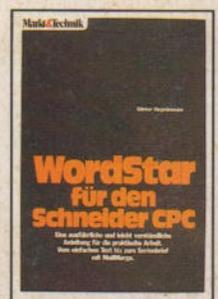
C. Strauch
CPC 464 - Programmieren in Maschinensprache
Juli 1985, 276 Seiten

Dieses Buch weilt in die Arbeitsweise des BASIC-Interpreters ein und erklärt die Funktionsweise der Bauteile des Geräts und deren Zusammenwirken. So ergeben sich auch für reine BASIC-Programmierer bereits viele Änderung- und Eingriffsmöglichkeiten in die Maschine.

Best.-Nr. MT 829
ISBN 3-89090-166-2
DM 46,-/sFr. 42,30/6S 358,80

C. Strauch/H. Pick
CPC 464 für Ein- und Umsteiger

Februar 1985, 260 Seiten
Starthilfe für den Anfänger; Orientierungshilfe für den Umsteiger.
Best.-Nr. MT 801
ISBN 3-89090-090-9
DM 46,-/sFr. 42,30/6S 358,80



G. Jürgensmeier
WordStar 3.0 mit MailMerge für den Schneider CPC
September 1985, 435 Seiten

Das unentbehrliche Zusatz-Handbuch für die Arbeit mit dem Schneider CPC.
Best.-Nr. MT 779
ISBN 3-89090-180-8
DM 49,-/sFr. 45,10/6S 382,20

Dr. P. Albrecht
dBASE II für den Schneider CPC
September 1985, 280 Seiten

Best.-Nr. MT 837
ISBN 3-89090-188-3
DM 49,-/sFr. 45,10/6S 382,20

Dr. P. Albrecht
MULTIPLAN für den Schneider CPC
September 1985, 226 Seiten

Best.-Nr. MT 835
ISBN 3-89090-186-7
DM 49,-/sFr. 45,10/6S 382,20

Markt & Technik
BUCHVERLAG

Hans-Pinsel-Straße 2, 8013 Haar bei München
Schweiz: Markt & Technik Vertriebs AG,
Kollerstrasse 3, CH-6300 Zug, ☎ 042/41 56 56
Österreich: Rudolf Lechner & Sohn,
Heizwerkstraße 10, A-1232 Wien, ☎ 02 22/67 75 26

Markt & Technik-Fachbücher
erhalten Sie bei Ihrem Buchhändler!

Bestellkarten bitte an Ihren Buchhändler
oder an einen unserer Depot-Händler.
Adressenverzeichnis am Ende des Heftes.

www.mts.computer-technik.de