

ISSN 0251-0006 DM 8,-; öS 60,- Fr. 7,-

84-6



COMPUTER



**Einführung in APL**

**IBM-kompatibel: CORONA**

**Zahlentheorie mit HP-41**

**Tricom Miniplotter/Printer**

**Biegsame Kurvenlineale**

**Künstliche Intelligenz  
in BASIC**

# SEIKOSHA

## Drucker die überall passen und sagenhaft preiswert sind

Zum Beispiel:

# GP-700A

## High speed Color Graphic Printer



- für Einzelblatt und Endlospapier
- 7 Grundfarben
- Volle Graphikmöglichkeiten
- Parallel Centronics Interface
- Farbbandkassette

### Zusatzschnittstellen:

RS-232c, IEEE 488, RGB Interface

### Graphikprozessoren zu:

IBM-PC, Apple II/IIe,  
Commodore C64 usw.

**Ideal als farbiger Hardcopy-Drucker zu allen PC's sowie auch als schwarz/weiss Listing und Protokoll-Drucker.**

Endverbraucherpreis  
**Fr. 1355. — inkl. WuSt**

**SEIKOSHA-Drucker mit dem besten Preis/Leistungsverhältnis**

### SEIKOSHA - Drucker erhält man bei:

**Basel:** BD Electronic, Gundeldingerstr. 209; Computer Shop, Domacherstr. 161; Proxus GmbH, Bläsiring 160; Sysag AG, Holeystr. 87; RTV Zihlmann + Co., Rümliplatz; **Bern:** Eschenmoser AG, Laupenstr. 6; Radio TV Steiner AG, Waisenhausplatz 6; Radio TV Walz AG, Aarberggasse 55; **Bex:** Belectronic SA, Rue Central; **Biel:** EIM Computer AG, Mattenstr. 13; **Buchs/SG:** Obtron Electronic AG, Bahnhofstr. 54; **Fontainemelon:** Urs Meyer Electronic, Rue de Bellevue 17; **Frauenfeld:** Radio TV Kaiser AG, Schönhofstr. 15; **Freienstein:** Sennhauser + Co., Irchelstr. 28; **Genève:** Compustyle SA, Rue de Lyon 39; **Heerbrugg:** R.H.V. Computer; **Lausanne:** Computershop, Place de la Riponne 10; **Locarno:** MEL SA, Via ai Monti 136; **Lugano:** Computic SA, Innovation Lugano Computer Shop; **Luzern:** Dialog Computer AG, Seeburgstr. 18; **Martigny:** ECOLEX, Rossettan 6; **Münchenstein:** Geiger Microcomputer, Grabenacherstr. 15; **Neuchâtel:** Bolomey-Organisation, Ruelle Mayor 2; **Olten:** Computer + Systems H.P. Schmid, Baslerstr. 86; **Rorschach:** B. Müller Computer Systeme, St. Gallerstr. 16; **Solothurn:** Computerzentrum Solothurn, Löwengasse 2; **St. Gallen:** Computer-Schiff, Marktgasse 5; **Schaffhausen:** PIM Systems AG, Lochstr. 18; **Thun:** HMB Computer AG, Mittlerestr. 3; **Winterthur:** NOWAK AG, Technikumstr. 46; **Wohlen:** Sysag AG, alte Bahnhofstr. 7; **Zürich:** ELES Elektronik AG, Stampfenbachstr. 73; Eschenmoser AG, Birmensdorferstr. 20; Microspot AG, Sihlfeldstr. 127;  
**ganze Schweiz:** Interdiscount AG; Jelmoli SA; RTV Steiner;

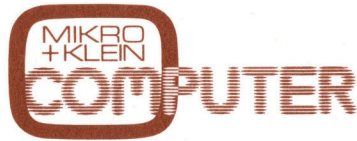
Member of Peripherals Distributors Association of Switzerland

ERNI + CO. AG Elektro-Industrie  
CH-8306 Brüttsellen (Zürich) Tel. 01/835 35 35





# Das Kleincomputer-Magazin



mit exklusiven Testberichten, praxiserprobten Anwenderprogrammen und aktuellen Informationen über Mikro- und Kleincomputer, PPC und Hand-Held-Computer für den kommerziellen und technischen Einsatz sowie für den privaten Gebrauch.

## Abo-Bestellkarte

Seit 1979 der Geheimtip für kompetente Computerinformation

bitte frankieren

Mikro+Kleincomputer  
**Informa Verlag AG**  
Postfach 1401

CH-6000 Luzern 15

bitte frankieren

### Meine Anschrift:

\_\_\_\_\_

Name

\_\_\_\_\_

Vorname

\_\_\_\_\_

Beruf

\_\_\_\_\_

Strasse

\_\_\_\_\_

PLZ/Ort

\_\_\_\_\_

Telefon

Mikro+Kleincomputer  
**Informa Verlag AG**  
Postfach 1401

CH-6000 Luzern 15

bitte frankieren

Meine/ unsere Anschrift:

\_\_\_\_\_

Name/Vorname/Firma

\_\_\_\_\_

Beruf

\_\_\_\_\_

Strasse

\_\_\_\_\_

PLZ/Ort

\_\_\_\_\_

Datum

\_\_\_\_\_

Unterschrift

**X**

Mikro+Kleincomputer  
**Informa Verlag AG**  
Postfach 1401

CH-6000 Luzern 15

## Commodore-Basic kein Problem.



2. Auflage ISBN 3-907007-01-8

Für jeden Commodore-Benutzer, der seinen CBM-Rechner noch besser kennenlernen möchte und/oder auch in Maschinensprache damit arbeiten will, ist dieses Buch eine wahre Fundgrube. Sämtliche CBM-«Spezialitäten», inkl. neue Adressen sowie Funktionen und Möglichkeiten der CBM-Betriebssysteme 3000 und 4000/8000 werden umfassend behandelt und eingehend erklärt.  
Fr./DM 49.-

**MIKRO+KLEINCOMPUTER  
INFORMA VERLAG AG**  
Postfach 1401, CH-6000 Luzern 15



**Mikro Computer Schule Basel**

Kompetente Lösungen für Schulung und Anwendung

Jetzt Aktuell

Ihre EDV-Ausbildung?

Unsere Kurse für die Praxis:

**Programmierkurse in Basic, Pascal und Assembler**

- Kurse für Mikrocomputereinsatz
- Training für Anwender in der Buchhaltung, Fakturierung, Dateiverwaltung
- Textverarbeitung mit Computersystemen

Einführungskurse für Anfänger und Anwenderkurse für Fortgeschrittene

Tages-, Abend- und Samstagskurse

**Mikro Computer Schule**  
Holeestrasse 87, 4054 Basel

Telefon **061/38 21 20**

---

Verlangen Sie bitte unser aktuelles Kursprogramm

Name: Firma: \_\_\_\_\_

Strasse: \_\_\_\_\_ PLZ/Ort: \_\_\_\_\_



Nach dem grossen Erfolg in den USA wurde der neue PC ITT XTRA auch in Europa offiziell an der Comdex-Exhibition in Amsterdam vorgestellt. Die Einführung durch STR in den Schweizer Markt erfolgte Ende November 1984. Der ITT XTRA zeichnet sich besonders durch seine Kommunikationsfähigkeit und seine Erweiterungsmöglichkeiten aus. Das Betriebssystem ist mit dem des IBM-PCs operationell kompatibel, wodurch Anwender auf praktisch alle international erhältlichen SW-Angebote zurückgreifen können. Das Foto wurde uns freundlicherweise von der Standard Telephon und Radio AG, Zürich, zur Verfügung gestellt.

Welcher Programmierer stand nicht schon einmal vor dem Problem, über das Programm Zugriff zu einem Timer oder zu einer Uhr zu haben? Diese beiden Funktionen werden immer dann benötigt, wenn ein Programm zeitgesteuert ablaufen soll, wenn Zeitintervalle gemessen werden müssen oder wenn zu bestimmten Zeiten oder Kalenderdaten bestimmte Programme ablaufen sollen.

**Seite 53**

Ausgabe Dezember 1984  
Erscheint 6mal pro Jahr  
6. Jahrgang

**KLEINCOMPUTER aktuell**

CORONA PPC-2 – die Krönung der IBM-Kompatiblen	5
Brother HR-5, klein und vielseitig	11
Die Mikroprozessoren 8086 und 8088	15
TRICOM Miniplotter CPP-114	21

**LEHRGÄNGE**

Einführung in APL (1)	27
Die Programmiersprache «C» (7)	39

**PPC/HHC**

Zahlentheorie mit dem HP-41	45
-----------------------------	----

**PRAXIS MIT MIKRO'S**

Datum/Zeit-Information und Timer für jeden PC	53
---	----

**RUND UM DEN IBM-PC**

Aktuelle Meldungen zum IBM-PC	59
Künstliche Intelligenz in BASIC	63

**GEWUSST WIE**

Splines: Biessame Kurvenlineale	69
Ein Proportional-Blocksatz-Druckprogramm	73
3D-Darstellung einer Kugel mit Längen- und Breitenkreisen mit Hidden-Line-Routine	77

**BRIEFE AN DIE REDAKTION**

Das M+K-Leserforum	83
--------------------	----

**CBM/PET NEWS**

FORTH auf Commodore C-64 (4)	89
Wie bringe ich Maschinenroutinen in den Speicher des CBM?	91

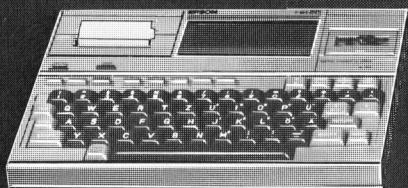
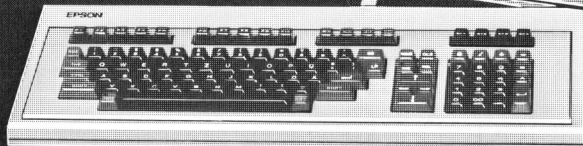
**COMPUTER-BÖRSE**

Fundgrube für günstige Occasionen	94
-----------------------------------	----

**VORSCHAU**

	98
--	----

# WIR DREI:



## Darf ich mich vorstellen: EPSON HX-20. Handheld-Computer des Jahres 1983.

Seit bald 2 Jahren werde ich als A4 kleiner netzunabhängiger Portable in der Schweiz von Computerfachleuten, Managern und in der Industrie tausendfach geschätzt. Zum Beispiel als Steuergerät (kompatibel mit dem QX-10), in der Lagerverwaltung, im Aussendienst, auf der Baustelle, in der Schule oder bei Studenten. Umfangreiche Software und leichtverständliche deutsche Handbücher erleichtern Ihnen den Einstieg in die EDV. Durch meine Schnittstelle zu diversen Peripheriegeräten bin ich sogar noch ausbaubar.

Microsoft-Basic, 32 kB RAM, 40 kB ROM, ausbaubar.

Detailinformationen über das EPSON-Computersystem und alle zugehörigen Peripheriegeräte erteilt Ihnen gerne Ihr Computerfachhändler.

## Ich bin der EPSON QX-10- Bürocomputer mit den unbegrenzten Möglichkeiten.

Wenn's um Textverarbeitung, Finanz- und Lagerbuchhaltung, Offertwesen oder Kalkulation geht, sollten Sie mich kennenlernen. Denn ich bin verblüffend einfach und ohne Computer-Vorkenntnisse zu bedienen und erweise meine Dienste auch gern im Kleinbetrieb, beim Handwerker oder beim Akademiker.

Multifont CP/M-Betriebssystem für Ausnutzung der gesamten Speicherkapazität von 256 KB RAM. Zugriff auf die komplette CP/M-Software für alle Anwendungsgebiete. Als Option zusätzlich MS-DOS-Betriebssystem-Erweiterung.

## Gestatten: EPSON PX-8 – der neue Handheld-Computer mit Riesenleistung.

Mit mir haben Sie's zu tun, wenn Sie einen mobilen Computer für den harten Alltagsinsatz – Textverarbeitung, Tabellenkalkulation, Terminplanung und Management-Aufgaben – suchen. Die entsprechenden Programme (Portable WordStar, Portable CalcStar, Portable Scheduler) werden gleich mitgeliefert und sind im Kaufpreis eingeschlossen. Mein CP/M-Betriebssystem erlaubt Zugriff zur grössten Programmauswahl in unterschiedlichsten Anwendungsbereichen.

Z80-Prozessor mit 64-182 KB RAM. Bildschirm mit 8 Zeilen zu je 80 Zeichen. Batterie/Netzbetrieb. Schnittstelle zu allen denkbaren Peripheriegeräten. Kompatibel mit dem QX-10 und anderen Personal-Computern.

<b>Informations-Coupon</b>	
<input type="checkbox"/>	Bitte senden Sie uns Unterlagen über den EPSON
<input type="checkbox"/>	<input type="checkbox"/> HX-20 <input type="checkbox"/> PX-8 <input type="checkbox"/> QX-10
<input type="checkbox"/>	Firma _____
<input type="checkbox"/>	Sachbearbeiter _____
<input type="checkbox"/>	Strasse _____
<input type="checkbox"/>	Plz/Ort _____
<input type="checkbox"/>	Coupon senden an die Generalvertretung für die Schweiz:
<input type="checkbox"/>	Excom AG Switzerland
<input type="checkbox"/>	Einsiedlerstrasse 31, 8820 Wädenswil
<input type="checkbox"/>	Telefon 01/780 74 14

# EPSON EXCOM

# CORONA PPC-2 - die Krönung der IBM-Kompatiblen

**Spricht man heute über IBM-kompatible Kleincomputer, so fällt immer häufiger ein Name: CORONA. Zu recht, wie wir meinen. Dieser Kleincomputer stellt tatsächlich eine sehr gute Anpassung an den Standard der Firma IBM dar. Ueber unsere Erfahrungen während einem zweimonatigen Testlauf mit der portablen Ausführung dieses Gerätes soll dieser Artikel einen Eindruck geben.**

Als wir den CORONA zum Testen übernahmen und er reisefertig verpackt vor uns auf dem Tisch stand, dachten wir unwillkürlich an einen, auf seiner Schmalseite aufgestellten

## Eric Hubacher

Flugreisekoffer. Dieser ganz in beige gehaltene «Flugkoffer» mit den Abmessungen 49x48x22 cm, weist an seiner oberen Schmalseite einen massiven Metallgriff auf zum Transport dieses, mit 17 kg allerdings recht schweren, Kleincomputers.

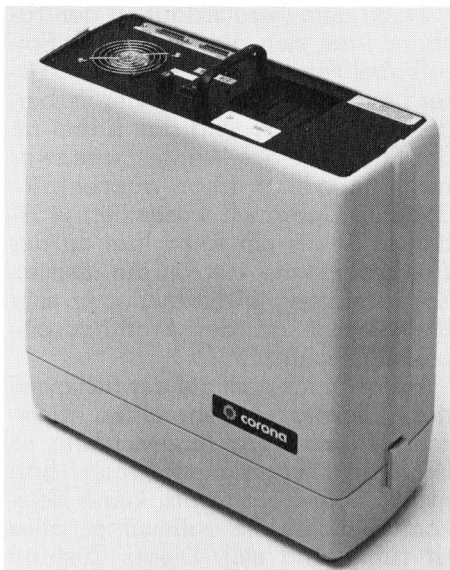
Zum Aufstellen des Gerätes klappt man unten am Gehäuse zuerst einen Aufstellbügel heraus, dann löst man mit einem Handgriff die beiden Verschlüsse aus Kunststoff, und schon kann die Haube abgenommen werden. In dieser Haube, dem abnehmbaren Kofferdeckel, ist alles Zubehör zum CORONA sehr praktisch verstaut. Sie bietet Platz für die Kabel, einen Stapel Disketten und natürlich auch für die Tastatur. Damit nichts herausfällt oder beim Transport klappert, kann mit einem patenten Klettenverschluss alles in der Haube festgehalten werden.

Nach dem Herausklauen der Tastatur lässt sich diese über ein Spiralkabel von mehr als 80 cm nutzbarer Länge mit dem Computer verbinden. Eingesteckt wird das Kabel an der Frontseite des Gerätes, was zwar nicht unbedingt schön aussieht, jedoch im Gebrauch wesentlich praktischer ist als ein Anschluss an der Geräterückseite wie ihn die meisten Konkurrenten aufweisen. Zudem sind die Stecker an der Frontbauseite besser geschützt.

Hat man mit dem auf der Rückseite des Gerätes angebrachten Netzschalter das Gerät eingeschaltet, so ist man verunsichert: Man hört nichts, man sieht nichts, keine Netzkontrolleuchte die aufleuchtet, alles ist ruhig und still. Dieser Zustand dauert etwa zehn Sekunden, aber dann plötzlich geht's los. Mit einem beeindruckend lauten Rattern beginnt die Diskettenstation die eingelegte Diskette zu lesen.

Noch einige Sekunden Rattern und das System meldet sich auf dem Bildschirm mit der Bezeichnung MS-DOS 2.0. Nach mehreren weiteren





*Der geschlossene reisebereite Koffer*

Sekunden schalten die Antriebsmotoren der Diskettenlaufwerke automatisch ab, und es kehrt wieder Ruhe ein. Unheimliche Ruhe, muss man sagen, denn der CORONA produziert kein hörbares Geräusch. Hier sind die Leute von CORONA zu loben, die es als erste fertigbrachten, einen Computer mit einem absolut lautlosen Ventilator auszurüsten.

Beim Blick auf den eingebauten monochromen Bildschirm mit einer Bildschirmdiagonalen von 9 Zoll fällt einem sofort die wesentlich bessere Zeichendarstellung gegenüber dem Vorbild IBM auf. Dies deshalb, weil beim CORONA jedes Zeichen aus 14x13 Leuchtpunkten erzeugt wird im Gegensatz zu den 7x9 Punkten des IBM. Dieser bessere Eindruck wird durch den scharfzeichnenden kleinen Bildschirm noch unterstützt. Der CORONA überzeugt übrigens ebenfalls mit einer sehr guten graphischen Auflösung.

Bei genügend grossem Speicher- ausbau bietet der CORONA auch noch die Möglichkeit, auf dem Bildschirm Text und Graphiken zu mischen und bis zu fünf Graphikseiten zu benutzen. Nebst seiner grossen Auflösung von 640x325 Punkten ist auch eine IBM-kompatible Auflösung mit 640x200 Punkten oder die mittlere Auflösung von 320x200 Punkten einstellbar. Bei einer Umschaltung zwischen verschiedenen Graphikauflösungen wird nicht die Auflösung des Bildschirms umgeschaltet, sondern es wird nur der kleinere, der Auflösung entsprechende Bildausschnitt auf dem Bildschirm dargestellt.

Das gleiche ist übrigens bei der Umschaltung vom 80-Zeichen- auf den 40-Zeichen-Modus zu beobach-

ten. Im 40-Zeichen-Modus, d.h. bei der Darstellung von 40 Zeichen pro Zeile, werden die Buchstaben nicht doppelt breit (elongiert) geschrieben wie beim IBM, sondern es wird nur die linke Hälfte des Schirmes vollgeschrieben. Dies ist eine Einschränkung gegenüber dem IBM, die jedoch unserer Meinung nach nicht wesentlich ist. Will man aber eine absolut IBM-kompatible Graphik, so muss man den CORONA mit einer Graphikkarte und einem Monitor von IBM ausrüsten. So frisiert kann dann auf dem CORONA auch mit dem Flugsimulator-Programm von Microsoft, welches ja oft als Kompatibilitätstester eingesetzt wird, gespielt werden.

Die Kontrasteinstellung für den eingebauten Bildschirm wird an einem kleinen Drehpotentiometer an der Frontseite des Gerätes vorgenommen. Eine weitere Einstellmöglichkeit für die Grundhelligkeit des Schirmes ist nicht vorgesehen.

### **IBM-kompatible Tastatur**

Die schön flache Tastatur lässt sich über Klappfüsschen auf fünf verschiedene Höhen einstellen. Die CORONA-Tastatur für sich ist tatsächlich vollständig IBM-kompatibel. Sie weist genau die gleiche, umstrittene Tastaturanordnung auf, hat jedoch trotzdem noch zwei Verbesserungen gegenüber dem IBM-Modell aufzuweisen. So sind die Gross-/ Klein-Umschalttaste und die Betriebswahltaste für den Cursor-Block mit je einer roten Kontrolleuchte versehen, so dass man immer weiss, in welchem Betriebsmodus man sich befindet. Ganz links auf der Tastatur sind wie beim IBM zehn Funktionstasten zu finden. Alle Eingabetasten für Schriftzeichen sind in weiss gehalten, während Funktionstasten mit einem vornehmen grau-beige eingefärbt sind. Die Tastatur selbst stammt, wie übrigens bei vielen Herstellern von zum IBM kompatiblen Geräten, von KEYTRONIC, einem bekannten Tastaturhersteller. Zur Verwaltung und Steuerung der Tastatur ist ein eigener Prozessor vom Typ 8048 im Tastaturgehäuse eingebaut. Dieser sorgt mit seinem Zwischenbuffer auch dafür, dass beim schnellen Schreiben nicht plötzlich einzelne Zeichen verloren gehen, nur weil der Computer mit der Verarbeitung nicht nachkommt.

Die Tastatur ist übrigens dieselbe, wie sie auch beim Tischgerät von CORONA eingesetzt wird. Viele Hersteller integrieren die Tastatur gleich

in die Abdeckung des Transportkoffers. Uns gefällt die Lösung, wie sie beim CORONA gewählt wurde, besser, da hier auch die Tastatur gut gegen mechanische Schäden oder Verschmutzungen durch Spritzwasser und Ähnliches geschützt ist.

Gleich neben dem Kontrastregler für den Bildschirm und der Steckbuchse für den Tastaturanschluss sind im rechten Drittel des Gerätes die beiden Diskettenstationen im Slimline-Format vertikal eingebaut. Die Diskettenstationen haben eine Speicherkapazität von je 360 KByte bei einer zweiseitigen Aufzeichnung. Sie arbeiten genau nach dem IBM-Diskettenformat und erwiesen sich bei all unseren Versuchen als absolut kompatibel, d.h. es war kein Problem, IBM-Disketten zu lesen und zu beschreiben.

Bei den Diskettenstationen des CORONA fiel uns erstmals an einem Computer ein sehr kluges Konstruktionsdetail auf. Der CORONA stellt ja bei Nichtgebrauch der Diskettenstationen automatisch die Laufwerksmotoren ab. Wird nun eine Diskette in die Lesestation eingeschoben, so fangen bereits beim Einschoben die Motoren zu laufen an, so dass bei vollständig eingesetzter Diskette und nach Schliessen des Diskettenfaches automatisch eine saubere Zentrierung der Diskette erfolgt. Dies vergrössert die Zuverlässigkeit der Diskette als Datenspeicher, da die Disketten immer genau gleich gut zentriert werden. Der einzige Nachteil an den Diskettenstationen - der unserer Meinung nach leicht behoben werden könnte - sind deren extreme Laufgeräusche.

Denkt man an IBM-kompatible Computer, so schaut man immer auch gleich nach, ob und wieviele IBM-kompatible Erweiterungsstecker im Computer vorgesehen sind. Beim

MUK 1	29 sec
MUK 2	35 sec
MUK 3	63 sec
MUK 4	205 sec
MUK 5	85 sec
MUK 6	21 sec
MUK 7	24 sec

*Die mit den MUK-Tests erzielten Ausführungszeiten*

MUK 3	177,1951507478859
MUK 4	189477,3807736148

*Die mit den MUK-Tests erzielten Rechenresultate*



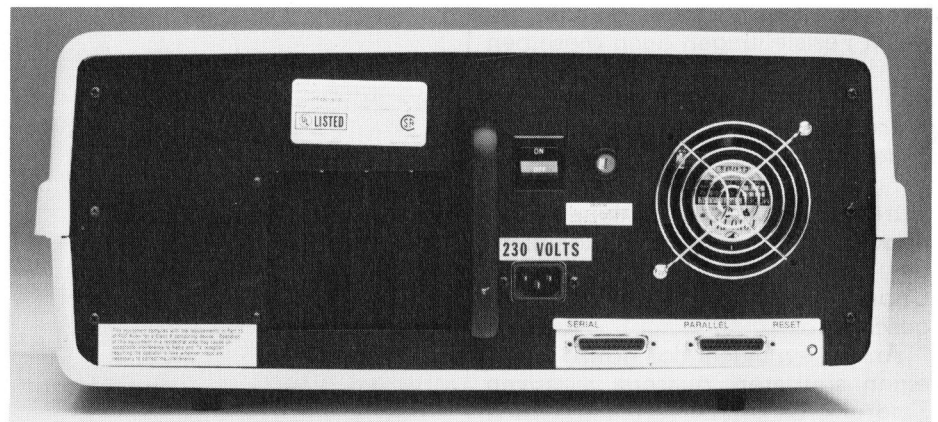
CORONA sind es vier Plätze, die der IBM-Norm entsprechen und IBM-kompatible Erweiterungskarten der langen Baugrösse aufnehmen können. Die von uns im CORONA getesteten Erweiterungskarten funktionierten alle zu hundert Prozent einwandfrei.

Vier Erweiterungsplätze scheinen nicht gerade weit zu reichen, denkt man an die Erfahrungen mit dem IBM. Dort werden ja die Erweiterungsplätze recht rasch aufgefüllt, da der Bildschirmkontroller, die Peripheriegeräte-Schnittstellen und auch der Diskettenkontroller alle als Zusatzplatinen ausgeführt sind und so rasch mindestens drei Steckplätze als Dauermieter belegen. In diesem Punkt liegt man beim CORONA besser, weil die Grundausrüstung des Gerätes bereits über zwei Peripheriegeräte-Schnittstellen, nämlich eine Parallelschnittstelle nach Centronics-Norm und eine voll über Software konfigurierbare RS232-Schnittstelle verfügt. Diese Schnittstellen belegen also keinen Erweiterungssteckplatz. Ebenfalls erfolgt die Ansteuerung des eingebauten Monitors über eine Schaltung, die sich wie der Diskettenkontroller bereits auf der Grundplatine befindet.

Auch für einen Speicherausbau des CORONA von den 256 KByte, die bei der Standardausführung bereits mitgeliefert werden, auf eine Kapazität von 512 KByte werden keine Erweiterungskarten benötigt, da auf der Hauptplatine bereits Steckerplätze für die Erweiterung auf 512 KByte RAM vorgesehen sind. Die vier zur Verfügung stehenden Erweiterungssteckplätze werden also für das Arbeiten mit dem Grundgerät nicht benötigt. Wollen Sie aber den CORONA mit einem Farbmonitor betreiben, ist eine entsprechende IBM-kompatible Farberweiterungskarte erforderlich.

Die beiden bereits erwähnten, auf der Hauptplatine untergebrachten Peripheriegeräte-Schnittstellen werden hinten am Gerät auf zwei 25-polige Miniatur-Steckern herausgeführt.

Oftmals hilfreich beim Arbeiten mit dem Rechner, vor allem wenn man selbst programmiert, ist der kleine, rechts neben den Schnittstellen-Steckern befindliche Reset-Knopf. Damit kann das Gerät bei einem Hängen des Programms, was beim IBM manchmal nur mit einem Ausschalten des Netzes und erneutem Einschalten des Gerätes behoben werden kann, auf einen kleinen Knopfdruck frisch gestartet werden.



Die Rückseite des Gerätes, die beim Transport den Kofferoberteil darstellt.

## Gut abgeschirmt

Vor unbefugten Blicken bestens geschützt sind die Innereien des CORONA. Den Blick auf die Hauptplatine des CORONA muss man sich mit dem Schraubenzieher geradezu erkämpfen. Zuerst hat man nämlich vier Schrauben zu lösen, die den oberen Teil des Gehäuses festhalten. Danach kann die obere Hälfte des recht massiven Kunststoffgehäuses abgehoben werden. Doch noch ist man nicht am Ziel, denn nun trifft man auf ein zweites Gehäuse im Gehäuse. Zur Freude jedes Benutzers, der auf eine geringe Störabstrahlung seines Gerätes angewiesen ist - denken wir nur an den Funkamateurliebling, der den Computer neben seiner Empfangsanlage betreiben wird -, ist der gesamte Computer in ein dicht verschlossenes Metallgehäuse eingebaut. Es müssen also weitere Schrauben gelöst werden. Um nur die rechte Hälfte des Gerätes mit den Diskettenstationen abzudecken, genügt das Lösen von acht Schrauben; möchte man das ganze Gerät öffnen, so sind fünfzehn Blechschrauben zu entfernen. Nach Abheben der beiden Metalldeckel hat man den Blick frei auf eine äusserst massive, amerikanisch wirkende Metallkonstruktion. Wir könnten uns vorstellen, dass der CORONA dank seinem rundherum geschlossenen Metallgehäuse eines der Geräte mit der kleinsten Störabstrahlung sein könnte, doch konnten wir dies noch nicht nachmessen.

Das Speisegerät, welches übrigens zwischen 110 und 220 Volt umschaltbar ist, wird über eine mechanisch vollständig gekapselte Netzfiltereinheit mit dem nötigen Strom versorgt. Das robust aufgebaute Schaltnetzteil erbringt gemäss den technischen Unterlagen zum CORONA eine Leistung von 110 Watt. Damit übertrifft es die eher schwachbrüstige Strom-

versorgung des IBM beinahe um das Doppelte.

Bei der Betrachtung der Hauptplatine des Computers sieht man, dass neben dem Hauptprozessor, dem 8088, auch noch ein Platz für den Co-Prozessor vom Typ 8087 vorgesehen ist. Es werden all die gleichen Typen von Komponenten eingesetzt, die auch der IBM-PC verwendet. Deshalb sind auch von dieser Seite her keine Schwierigkeiten mit der IBM-Kompatibilität zu erwarten.

## Technische Daten

### Zentraleinheit

CPU: Intel 8088 mit ?? MHz  
Taktfrequenz, Arithmetikprozessor 8087 als Option  
256 KByte RAM, Erweiterbar auf 512 KByte  
1 RS232-Schnittstelle  
1 Centronics-Parallel-Schnittstelle  
Vier voll IBM-kompatible Erweiterungssteckplätze

### Tastatur

Frei bewegliche Tastatur mit 83 Tasten  
Anschluss über Spiralkabel

### Bildschirm

Monochromer 9 Zoll-Bildschirm mit 25 Zeilen zu 80 Zeichen  
Bildschirmfarbe: Grün  
Graphikauflösung max. 640x325 Bildpunkte  
Mischen von Text und Graphik

### Speicherkapazität

2 Diskettenstationen mit einer Speicherkapazität von je 360 KByte, IBM-Aufzeichnungsformat

# KLEINCOMPUTER aktuell

ten, vor allem nachdem gemäss unseren Feststellungen auch dieselben Hardwareadressen verwendet wurden.

Die äusserst geräuschvollen Diskettenantriebe lassen sich jetzt bei genauerem Hinsehen ebenfalls erklären. Die Diskettenstationen sind nämlich direkt ohne schalldämpfende Zwischenlagen an einen am Chassis befestigten Metallwinkel angeschraubt.

Ansonsten macht das ganze Gerät einen sauberen, äusserst massiven Eindruck. Es könnte vermutlich auch einen Sturz vertragen, obwohl nirgends stossabsorbierende Elemente eingebaut sind.

## IBM-Kompatibilität

Vom Gesichtspunkt der Hardware aus ist unserer Meinung nach der CORONA vollständig zum IBM-PC kompatibel. Nicht nur verwendet er den gleichen Mikroprozessor, die gleichen Steuerbausteine für Diskettenkontroller und Graphikkontroller, sondern er hat auch in allen von uns überprüften Betriebssystem-Schnittstellen die genau gleichen Einsprungsadressen wie der IBM-PC.



Das Gehäuse im Gehäuse

Auch die Speicheraufteilung ist identisch. Als Betriebssystem wird für den CORONA das Microsoft-Betriebssystem 2.0 mitgeliefert. Dieses MS-DOS wurde von CORONA noch

so abgeändert, dass es die gleichen Einsprungsadressen wie das IBM-Betriebssystem aufweist.

Die Kompatibilität des CORONA geht sogar so weit, dass, als wir ein

ROTRONIC PERSONAL-COMPUTER

ROTRONIC PERSONAL-COMPUTER



# corona

- kompatibel
- kompakt
- komplett



Die neuen PERSONAL-COMPUTER von CORONA. Professionell vom tragbaren bis zum stationären Modell. **IBM-kompatibel.** Grosse Bildschirme von hervorragender Qualität. Graphikfähig mit 640 x 325 Punkten Auflösung. 4 Adapter-Anschlüsse - Ihre Sicherheit für individuelle Erweiterungen - heute und morgen.

Konkurrenzfähige Preise.

Wiederverkäufer: Verlangen Sie unsere Konditionen

# ro-tronic ag

8040 Zürich · Badenerstrasse 435  
Telefon 01-492 32 11 · Telex 822 530

speziell für den IBM geschriebenes Kommunikationsprogramm starten wollten und dieses unter dem CORONA-Betriebssystem nicht funktionierte, wir ein IBM-Betriebssystem luden und dann mit dem CORONA als «IBM» problemlos die Kommunikations-Software laufen lassen konnten.

Die einzige Einschränkung gegenüber dem IBM besteht darin, dass der eingebaute Bildschirmcontroller die beim IBM verwendete 40-Zeichen-Darstellung nicht unterstützt, d.h. keine doppelt breite Schrift darstellt, sondern in dieser Betriebsart nur die linke Hälfte des Schirmes benutzt. Ist man darauf angewiesen und oder will man IBM-kompatible Graphiken darstellen, so muss eine Graphikkarte von IBM eingesetzt werden.

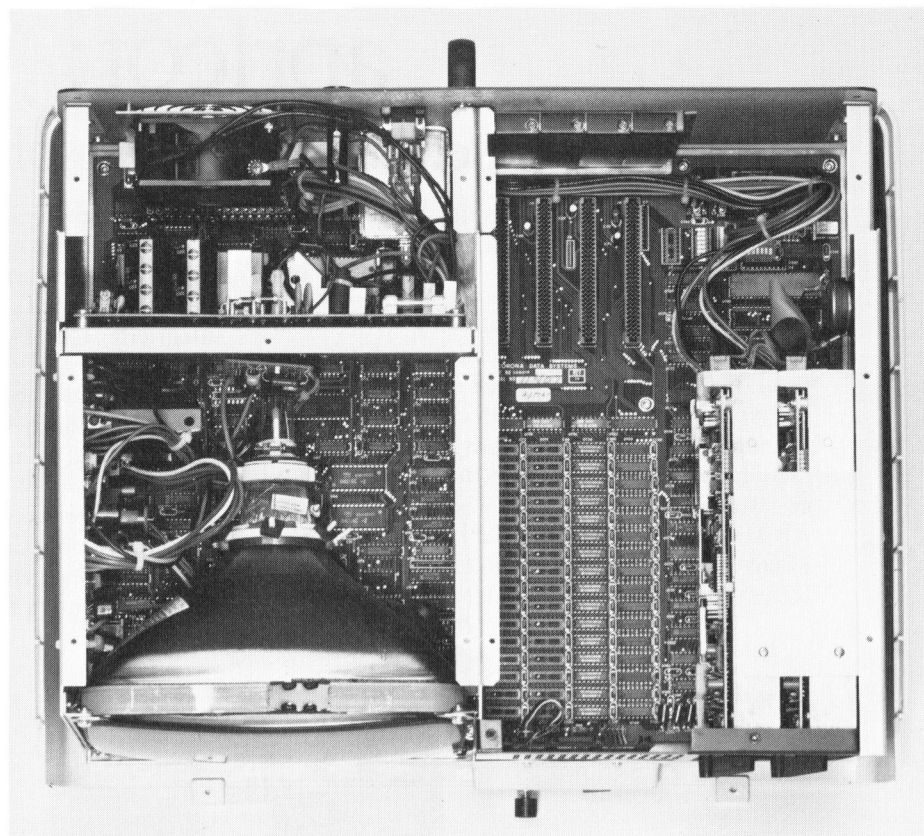
## Software

Zum CORONA wurden uns nicht nur das Betriebssystem 2.0 und 1.25 samt der entsprechenden ausführlichen Dokumentation mitgeliefert, sondern auch noch ein Lernprogramm. Dieses bietet eine interaktive Einführung in die Benutzung des Gerätes. Das Lernprogramm ist vollständig in Deutsch gehalten und erlaubt dem Erstbenutzer dieses Computers, sich leicht in dessen Eigenschaften einzuarbeiten. Der einzige Mangel war, dass das Programm nicht auf das neueste MS-DOS 2.0, sondern auf die alte Version 1.25 zugeschnitten war. So werden neue Möglichkeiten des Computers, z.B. das Erstellen von Unter-Directories, nicht erläutert. Das zum CORONA

### **CORONA PPC-2 Konfiguration und Preis der Testanlage**

- Zentraleinheit mit CPU 8088 und 256 KByte RAM-Speicher, Betriebssystem MS-DOS 2.0, Textsystem MultiMate 3.2, GW-Basic
- Zwei IBM-kompatible Diskettenstationen mit je 360 KByte Speicherkapazität.
- Monochromer Monitor mit 9 Zoll Bild diagonale, Darstellung in grün.
- Serielle Schnittstelle nach RS232-Standard, Parallele Centronics-Schnittstelle.
- IBM-kompatible Tastatur

Preis inkl. Wust Fr. 6700.--

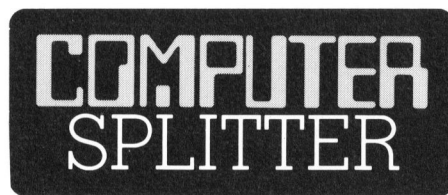


*Das Innenleben, massiv amerikanisch*

mitgelieferte grafikfähige BASIC ist vollständig IBM-kompatibel. Weiter gehört zum Lieferumfang noch ein sehr leistungsfähiges Textverarbeitungssystem, MultiMate, in der Version 3.2.

## Dokumentation

Zu unserem Testsystem wurden fünf Handbücher mitgeliefert. Zwei befassen sich mit dem Betriebssystem des CORONA, ein weiteres Ringbuch enthält die Dokumentation zum BASIC und das vierte Buch ist die Bedienungsanleitung zum Multi Mate-Textsystem. Alle diese vier genannten Bücher sind in Englisch abgefasst. Das fünfte Ringbuch enthält eine gute deutsche Uebersetzung des Microsoft-Betriebssystems, Version 2.0. □



## Wenn zwei sich streiten...

(257/eh) Der Streit vor dem amerikanischen Gericht zwischen den beiden Firmen Visi Corp und Software

Arts endete mit einem Vergleich. Software Arts ist als Entwickler des einstmals so berühmten Visi-Calc bekannt, während Visi Corp den Vertrieb dieses Programms besorgt. Der Rechtsstreit, der beide Firmen in ihrer Weiterentwicklung hemmte, sollte klären, wem die Rechte an Visi-Calc gehören. Im Vergleich wurden diese nun dem Programmierer, der Software Arts, zugesprochen. Inmitten der Klagen und Gegenklagen wurde das Wichtigste jedoch glatt vergessen: Niemand mehr wollte oder konnte sich um den Vertrieb des umstrittenen Software-Paketes kümmern, so dass dessen Verkäufe auf ein unbedeutendes Mass zurückgingen. Fazit: Operation gelungen, Patient gestorben! □

## China PC-User-Group

(256/eh) Im August tagte in Peking zum ersten Mal die «Peking IBM-PC User group». Zu dem Zeitpunkt betrug die Gesamtzahl der in China installierten IBM-PC-Geräte etwa 100 Stück. Nach Adam Riese ergibt das einen IBM-PC auf etwa 10 Millionen Chinesen. Diese Zahl gibt natürlich einen verfälschten Eindruck von der Computerdichte in China. Im letzten Jahr wurden ausserdem noch 10'000 Systeme aus chinesischer Produktion installiert. □

# apricot:

Apricot ist der  
erste Personal Computer  
der 4. Generation.

## Ein Leckerbissen unter den Personal Computern.

Zum ersten Mal ist ein Computer auf dem Markt, der nicht nur für Sie, sondern mit Ihnen arbeitet.

Der Bildschirm lässt auch für verwöhnte Augen kaum Wünsche offen. Er ist allseitig schwenkbar und lässt sich dadurch jeder beliebigen Sitzposition des Benützers optimal anpassen. Das Herz des Apricot Personal Computers besteht aus dem bekannten 16-Bit-Mikroprozessor Intel 8086. Der Hauptspeicher ist bereits mit 256 KByte ausgestattet und lässt sich bis 768 KByte ausbauen. Als Massenspeicher stehen die kompakten 3,5" Micro-Floppies von Sony mit 2 x 315 bis 2 x 720 KByte zur Verfügung. Der Harddisc-Apricot

besitzt die revolutionäre 3,5"-Festplatte mit 5 oder 10 MByte. Damit haben Sie eine Kapazität für 60 000 Adressen oder 3000 Seiten A4.

Der Clou in der Tastatur ist ein Microscreen mit integrierter Uhr, Datumanzeige und ein 10stelliger Rechner.

Eine umfangreiche und hochstehende Schweizer Software-Bibliothek steht zur Verfügung. Zudem ist der Apricot IBM- und Sirius-kompatibel.

Mit dem neuen Apricot gibt es endlich einen sehr kleinen Personal Computer mit hoher Leistung zu einem günstigen Preis. Vergleichen Sie doch selbst - bei Ihrem Fachhändler.

### brother

Verkauf durch:  
Brother Handels AG  
5405 Baden

Importeur:  
Romatron AG, 5405 Baden



### Apricot bietet mehr Leistung, ohne mehr zu kosten:

Personal-Computer:	Fr. 5 595.-
Apricot AP-1SS-N	Fr. 6 695.-
Apricot AP-2SS-N	Fr. 7 595.-
Apricot AP-2DS-N	Fr. 10 195.-
Apricot AP-5MS-N	Fr. 11 645.-
Apricot AP-10MD-N	
Monitor:	
9-Zoll-Grafik-Videomonitor	Fr. 755.-
12-Zoll-Grafik-Videomonitor	Fr. 995.-
Schönschreibdrucker:	
Zum Beispiel: Brother HR-15	Fr. 1 850.-

### Kosten Sie.

Verlangen Sie unsere Dokumentation.

- Apricot Personal Computer
- Harddisk-Apricot
- Schweizer Software

Name: \_\_\_\_\_

Firma: \_\_\_\_\_

Adresse: \_\_\_\_\_

PLZ/Ort: \_\_\_\_\_

Brother Handels AG, 5405 Baden

M+K



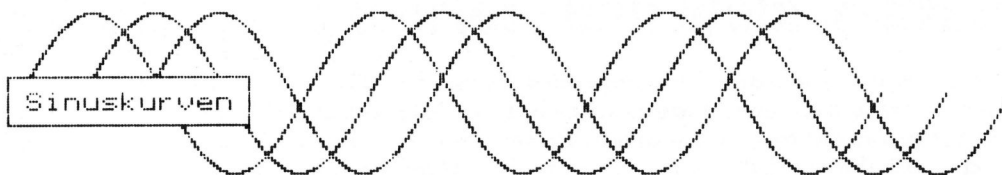


Bild 2: Screen Dump des Epson PX-8 auf HR-5

Zeichensätzen werden: ASCII, Französisch, Deutsch, Englisch, Dänisch/Norwegisch, Schwedisch/Finnisch, Italienisch oder Spanisch. Ueber zwei DIP-Schalter kann der gewünschte Zeichensatz ausgewählt werden, wobei für die letzte Vierergruppe noch eine Lötbrücke heranzuziehen ist. Da diverse Wahlmöglichkeiten auf einem 8-fach- und auf einem 6-fach-DIP-Block einzustellen sind, ist es unverständlich, warum nicht ein weiterer DIP-Schalter an Stelle der Lötbrücke eingesetzt wurde.

Die Druckgeschwindigkeit beträgt 30 Zeichen pro Sekunde, bei Fettdruck reduziert sich diese Geschwindigkeit auf die Hälfte. Miteinberechnet werden muss, dass am Zeilenende noch Zeit für den Transport des Farbbandes notwendig ist. Diese entfällt natürlich bei Verwendung von Thermopapier. Die Geschwindigkeit ist zwar nicht gross, dafür entbehrt man das Rattern und Klopfen üblicher Drucker. Nur das An- und Wegklappen des Druckkopfes unterbricht den nahezu geräuschlosen Druckvorgang. Gedruckt wird je nach Schrifttyp bidirektional mit Wegoptimierung oder nur in einer Richtung.

Unter den grafischen Zeichen befinden sich einige, die praktische Dienste etwa beim Erstellen von Tabellen leisten, aber auch solche, die man kaum brauchen wird.

## Grafikausdruck

Der HR-5 kann mit wenigen ESC-Befehlen in einen leistungsfähigen Grafikmodus gebracht werden. Die Zeilenschaltung ist von 1/72 Zoll bis 85/72 Zoll programmierbar, dabei ergibt die Programmierung auf 1/9 Zoll eine fugenlose Zeilenschaltung für Grafikausdruck, wobei acht Punkte vertikal gedruckt werden, die jeweils einem Byte entsprechen. Für Standarddruck können bis zu 480 Bilddaten pro Zeile programmiert werden, bei doppelter Punktdichte bis zu 960 Bilddaten. Ein Mischen von Zeichen und Grafik innerhalb einer Zeile ist möglich! Der Drucker interpretiert die Bytes je nach Befehl

als Zeichen- oder als Punktrasterdaten. Bei doppelter Punktdichte erhält man tiefschwarze, kontrastreiche Grafiken.

## Befehlssatz

So klein der HR-5 auch ist, er «versteht» viele Befehle und führt sie aus. Im (deutsch verfassten) Manual sind sie ausführlich erklärt und in lobenswerter Weise mit Beispielen versehen. Ausser den üblichen Standardbefehlen wie CR, LINE FEED, BACKSPACE, DEL, BEL, VERTICAL TAB, FORM FEED, HORIZONTAL TAB werden noch SHIFT OUT zur elongierten Zeichendarstellung und SHIFT IN zur komprimierten Zeichendarstellung verwendet. Gelöscht werden diese Befehle mit DC2 und DC4. DC1 und DC3 werden als Kennungen verwendet, da der HR-5 auch XON/XOFF-protokollfähig ist.

Eine Reihe von ESCAPE-Sequenzen bieten eine zusätzliche Fülle von Möglichkeiten wie: 1/9-Zoll-Schaltung, 1/6-Zoll-Schaltung; 1/72- bis 85/72-Zoll-Schaltung; Papierendetektor ein- oder ausschalten; Unterstreichen ein- oder ausschalten; Fettdruck ein- oder ausschalten; Breitdruck ein- oder ausschalten; Elite-Teilung ein- oder ausschalten; Zeilensprungfunktion ein- oder ausschalten; Hoch- oder Tiefstellen von

Zeichen; Anzahl der Druckspalten pro Zeile (1 bis 132); Einstellen der Seitenlänge in Zoll; Einstellen der Seitenlänge in Zeilen; Einstellen des Horizontaltabulators (max. 28 Pos.); Drucken in einer Richtung oder bidirektional; Einfache oder doppelte Punkt rasterdichte.

Geht man diese Tabelle durch, so sieht man, dass kaum Fähigkeiten fehlen, die herkömmliche Matrixdrucker bieten.

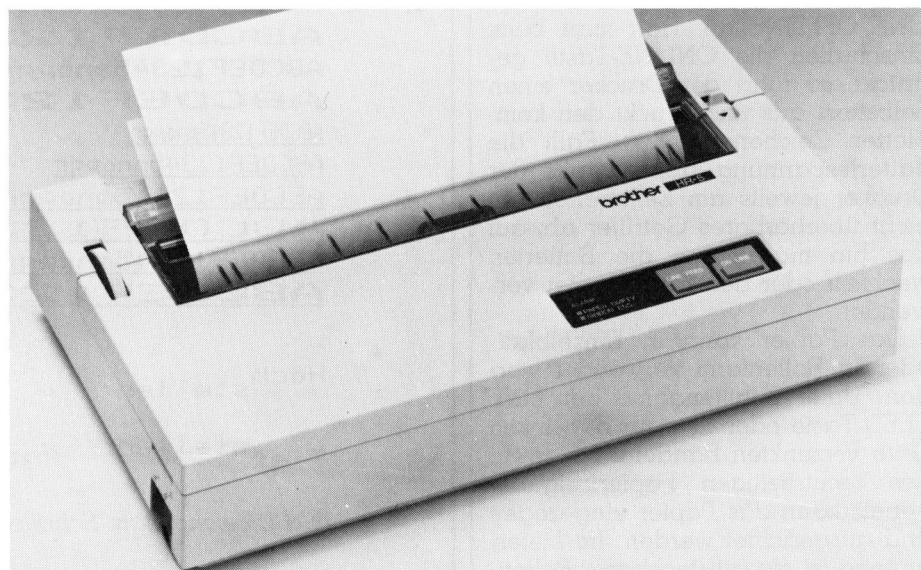
## DIP-Schalter

Von oben zugänglich, unter dem Druckmechanismus liegend, sind folgende Schalter: SLCT IN Signal, AUTO FEED Signal (automatische Zeilenschaltung), AUTO POSITION, BELL (Kontrollton durch Glocke), Nullzeichen (mit Schrägstrich oder ohne), Zeilenschaltung 1/6 Zoll oder 1/9 Zoll und der Schalter für acht internationale Zeichensätze.

Sechs weitere Schalter sind bei der RS232-Ausführung vorhanden: Centronics/RS232, Datenlänge 7 oder 8 Bits, Paritätswahl gerade oder ungerade und Baudrate 110, 150, 300, 600, 1200, 2400, 4800 oder 9600 Baud.

## Das Innenleben

Nur wenige Schrauben sind zu lösen, um Einblick in das Innenleben



zu erhalten. Ein Mikroprozessor 6303 steuert die Funktionen, ein RAM 6116 speichert die Variablen und dient als Buffer. Auf einem Sockel befindet sich das ROM HM61364, das Programm und Zeichengenerator enthält. Ausser dem ROM und einigen «konventionellen» ICs zur Schnittstellensteuerung sind die Schaltkreise in Miniaturgehäuse verpackt und einseitig aufgelötet. Diese Technik wird von den Japanern bevorzugt, um miniaturisierte Geräte zu produzieren.

## Dokumentation

Sehr wichtig, oft aber vernachlässigt, ist die Dokumentation zu Computern oder peripheren Geräten. Beim HR-5 ist dies nicht so: eine 48-seitige Anleitung, in richtigem Deutsch verfasst, gibt Auskunft über Inbetriebnahme, Einstellungen und Bedienung des Druckers, unterstützt durch instruktive Zeichnungen. Der umfangreiche Funktionscode ist gut beschrieben, Werte sind hexadezimal wie dezimal angegeben und auch Uebersichtstabellen fehlen nicht. Anwendungsbeispiele schaffen zusätzlich Klarheit. Auch die Schnittstellensignale sowie die Steckerbelegungen für Centronics- und RS232-Stecker sind hinreichend erklärt. Mit einer Darstellung aller verfügbaren Zeichen schliesst die Bedienungsanleitung, die komprimiert das Wesentliche sagt, ohne Ueberflüssiges zu bringen.

## Zusammenfassung

Der BROTHER HR-5 ist ein Gerät neuer, wegweisender Technologie, das sich am Trend zu tragbaren und netzunabhängigen Computern orientiert. Ein «Nachteil» mag sicher die Tatsache sein, dass Thermopapier oder ein Spezialthermotransferband verwendet werden muss, was auf längere Sicht höhere Kosten verursacht. Auch für den Batteriebetrieb sollten Alkalibatterien herangezogen werden, da billigere Batterien sich zu schnell erschöpfen. Im Notfall kann man noch immer auf einen Netzadapter zurückgreifen. Ein weiterer Nachteil muss zusätzlich in Kauf genommen werden: es sind keine Durchschläge möglich, ob mit Thermopapier oder Normalpapier, es wird nur jeweils ein Original erstellt. Im Bedarfsfall hilft aber ein mehrmaliges Ausdrucken.

Das Einlegen des Thermotransferbandes ist problemlos, sofern man die Anweisungen genau befolgt und

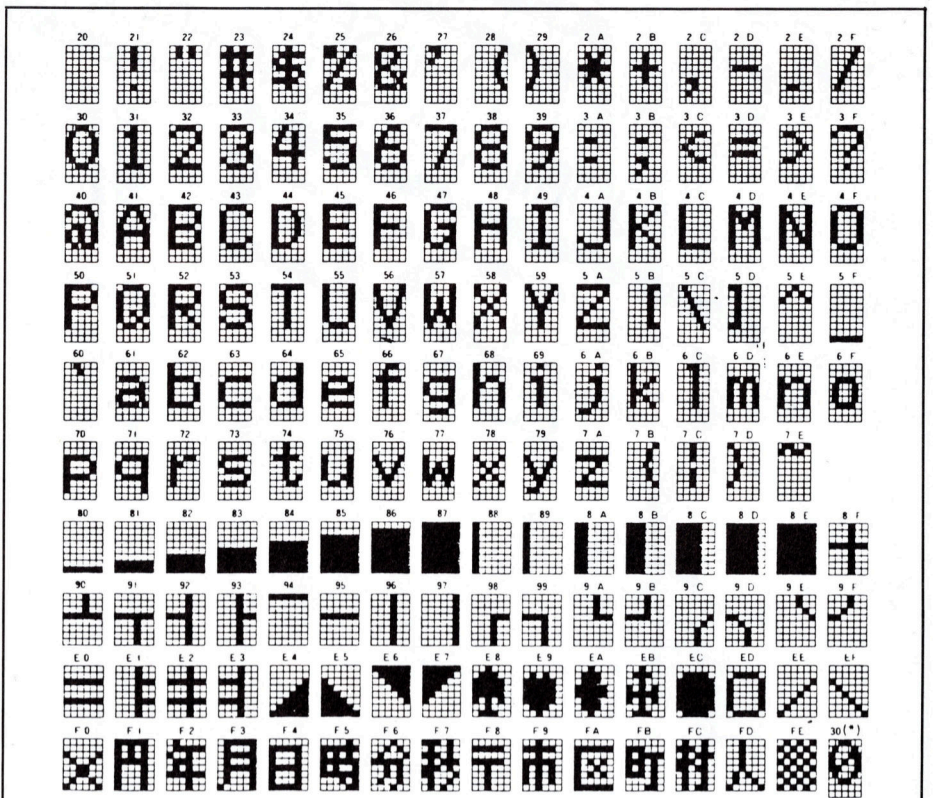


Bild 3: Zeichensatz des HR-5

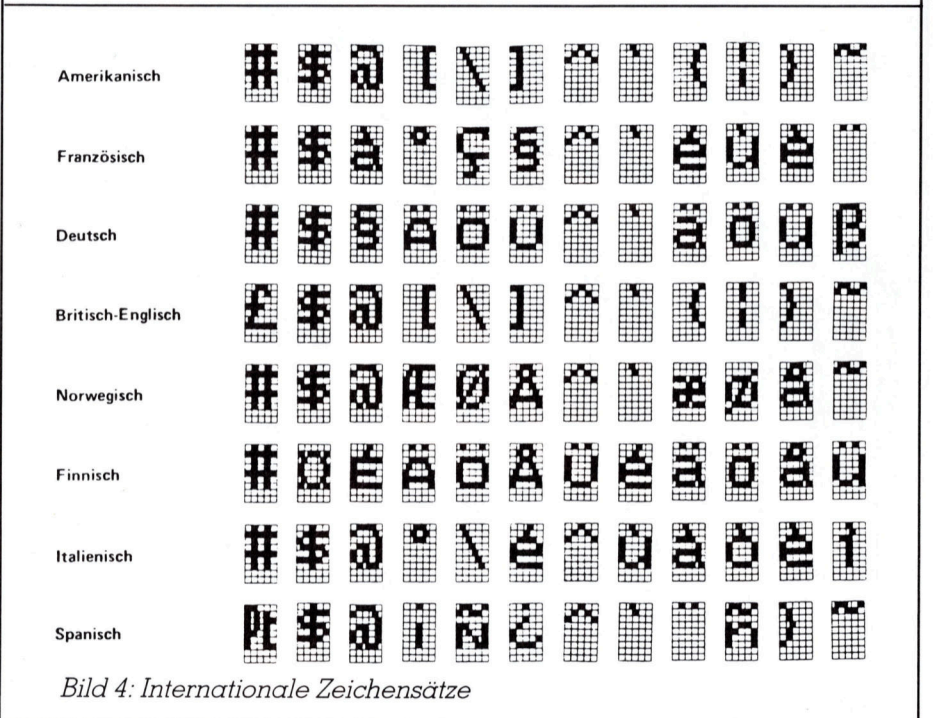


Bild 4: Internationale Zeichensätze

ein wenig übt. Da das Band sehr dünn ist, bedingt es ein vorsichtiges Hantieren.

Die Druckqualität ist zufriedenstellend, vorausgesetzt man verwendet glattes Papier, gewöhnliches Büro-papier ist schlecht geeignet. Für besonders kontrastreichen und sauberen Ausdruck ist die Fettschrift zu empfehlen.

An Vorteilen bietet der HR-5: Netzunabhängigkeit, sehr leisen Betrieb,

geringe Dimensionen und Gewicht, umfangreichen Zeichensatz und Grafikfähigkeit. Der HR-5 wurde als portabler Drucker konzipiert und die daran gestellten Anforderungen erfüllt er voll. Er ist kein Bürodruker, bewährt sich aber als Protokoll-druker und im mobilen Einsatz bestens. Als Zusatz zu Kleincomputern ist er eine ideale Ergänzung, zudem bietet er ein recht gutes Preis/Leistungs-Verhältnis. □

Jubiläum



5 JAHRE

IM DIENST DES KUNDEN

Mo: 13.30 bis 18.30  
Di bis Fr: 9.00 bis 12.00  
13.30 bis 18.30  
Sa: 9.00 bis 12.00

micomp sms ag

UNSER  
JUBILÄUMSANGEBOT

TOP 5

SOFTWARE  
MIT  
SYSTEM



**MBC-555**  
für jeden Einsatz

Der Grafik-Color-Computer für jeden Einsatz. Ausgerüstet mit 2 Laufwerken à 360 KByte, 128 KByte RAM (erweiterbar auf 256 KByte), ASCII Tastatur (dt. Tastatur in Vorbereitung), wahlweise monochromer oder Farbbildschirm.

**DAS BUCH ZUM COMPUTER**  
Auf über 230 Seiten erfahren Sie alles Wissenswerte und notwendige über den Computer und das Programmieren in BASIC. Mit vielen interessanten Beispielen und Subroutinen.  
Bestell-Nr.: S-HB01

Preis: **44.-**



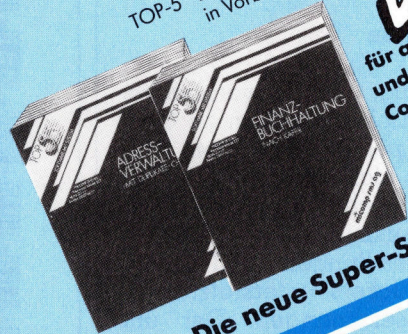
IBM-KOMPATIBEL

ab **3580.-**

- TOP-5 FINANZBUCHHALTUNG
- TOP-5 ADRESSVERWALTUNG
- TOP-5 TEXTVERARBEITUNG
- TOP-5 FAKTURIERUNG in Vorbereitung
- TOP-5 Dienstprogramme in Vorbereitung

678.-  
487.-  
478.-

225.-



für alle IBM- und kompatible Computer

NEU

Die neue Super-Software

**COMMODORE 64**

TOP-ANGEBOT



C 64 Computer 750.-  
VC-1541 Floppy 790.-  
VC-801 Drucker 650.-  
FIBUDATA Finanzbuchhaltung 350.-  
**2540.-**  
**2200.-**

TOTAL BARPREIS  
(Angebot ist nur für diese Konfiguration gültig)

Das neueste für unsere Musik-Fans  
**WERSIBOARD 64**  
inkl. Software für COMMODORE 64

548.-

**DAS NEUESTE**

**GHOSTBUSTERS**

Die Welt steht Kopf!  
Das Ghostbusters-Fieber ist ausgebrochen.  
Die neueste Gespensterkomödie aus Hollywood jetzt für Ihren COMMODORE 64



99.-  
59.-

420.-  
398.-

(Diskette)  
(Cassette)

NEU

P

24 Parkplätze direkt vor dem Laden.



Bestellen Sie direkt bei: MICOMP SMS AG, Versandabteilung, Postfach 237, 8106 Regensburg 2  
oder fordern Sie Unterlagen an bei: MICOMP SMS AG, Wehntalerstrasse 537  
8046 Zürich, Telefon 01/57 66 57

Ich interessiere mich für:

Name:

Strasse:

Ort:

PLZ:

Vorname:



## Die Mikroprozessoren 8086 und 8088

**In der letzten Zeit haben 16-Bit-Prozessoren vor allem bei grösseren Mikrocomputern Einzug gehalten. Diese Prozessoren besitzen einen 16-Bit-breiten Datenbus, der eine raschere Datenverarbeitung ermöglicht. Ein Beispiel soll dies verdeutlichen: Der Prozessor habe die Aufgabe, vier Bytes (32 Bits) in die internen Register zu laden. Ein 8-Bit-Prozessor kann pro Durchgang nur 8 Bits vom Speicher in ein Register verschieben; er benötigt also für die genannte Aufgabe vier Durchgänge, während der 16-Bit-Prozessor das Problem in nur zwei Durchgängen löst: Er ist schneller.**

Die beiden Prozessoren 8086 und 8088 besitzen die gleichen internen Register und sind softwarekompatibel, d.h. die Programme, die für den einen Prozessor geschrieben wurden, laufen auch auf dem anderen. Der wichtigste Unterschied besteht in der Anzahl Datenleitungen: Beim 8086 besteht der Datenbus aus 16,

### Stefan Ramseier

beim 8088 aus acht Leitungen. Dies hat zur Folge, dass der 8088 bei gewissen Aufgaben etwas langsamer arbeitet als sein Bruder.

Für die weiteren Betrachtungen wird nur auf den 8086 eingegangen; eine Aufzählung sämtlicher Unterschiede folgt in einem späteren Abschnitt.

### Entwicklung

Der Mikroprozessor 8086 von INTEL ist eine Weiterentwicklung des 8080 und des 8085. Seine Architektur und sein Befehlssatz basieren auf

dem 8080, sind aber viel komplexer. Die Mikroprozessoren 8080 und 8086 sind auf Assemblerebene aufwärtskompatibel. Dies bedeutet, dass ein altes, für den 8080 geschriebenes Programm nach erneutem Assemblieren auf dem 8086 lauffähig ist; die Umkehrung gilt nicht. Eine Einschränkung ist zu beachten: Das alte Programm muss in Assemblersprache verfügbar sein; bereits in Maschinensprache übersetzte Programme können nicht übernommen werden. Das gleiche gilt auch für Assemblerprogramme des 8085, wobei für die beiden Befehle RIM und SIM beim 8086 kein Äquivalent besteht.

Zwischen dem 8086 und dem Z80, der ebenfalls eine Weiterentwicklung des 8080 ist, besteht keine Kompatibilität, weil die Architektur der beiden Mikroprozessoren zu unterschiedlich ist.

In der Zwischenzeit wurden Co-Prozessoren entwickelt, die den Befehlssatz des 8086 erweitern. Es sind dies der 8087, der arithmetische Aufgaben löst, und der 8089, der Ein/Ausgabeoperationen steuert.

Der Vollständigkeit halber sei angetönt, dass auch schon Weiterentwicklungen des 8086 angeboten werden. Die beiden Prozessoren tragen die Bezeichnung 80186 und 80286; ihre Beschreibung würde den Rahmen dieses Artikels sprengen.

### Die Anschlüsse

Der Mikroprozessor 8086 steckt in einem 40-poligen Gehäuse, dessen Anschlüsse aus Bild 1 ersichtlich werden. Er kann in zwei verschiedenen Modi betrieben werden, die mit dem Pegel an Pin 33 (MN/MX') ausgewählt werden. (Der Zusatz ' bezeichnet im folgenden ein active-low-Signal.)

Bei Pin 33 = H-Pegel spricht man vom Minimum-Modus. In dieser Betriebsart, die in kleinen Systemen mit nur einer CPU gewählt wird, können maximal 1 MByte Arbeitsspeicher und 64 K Ein/Ausgabekanäle bedient werden. Die Steuerung erfolgt mit den Signalen DT/R', DEN', ALE, M/IO', RD', WR' und INTA'. Liegt an PIN 33 L-Pegel, so arbeitet der 8086 im Maximum-Modus. In dieser Betriebsart, bei der mehrere Prozessoren auf einen Systembus (Multibus) zugreifen, übernimmt ein Bus-Controller (8288) die Steueraufgaben. Dieser Controller stellt auch die Verbindung zwischen dem Mikroprozessor und dem Multibus her.

Die Anschlüsse des 8086 lassen sich in vier Gruppen unterteilen:

- Stromversorgung
- Adresssignale
- Datensignale
- Steuer- und Statussignale

Über viele der Kontaktstifte werden zwei oder drei Signale geleitet.

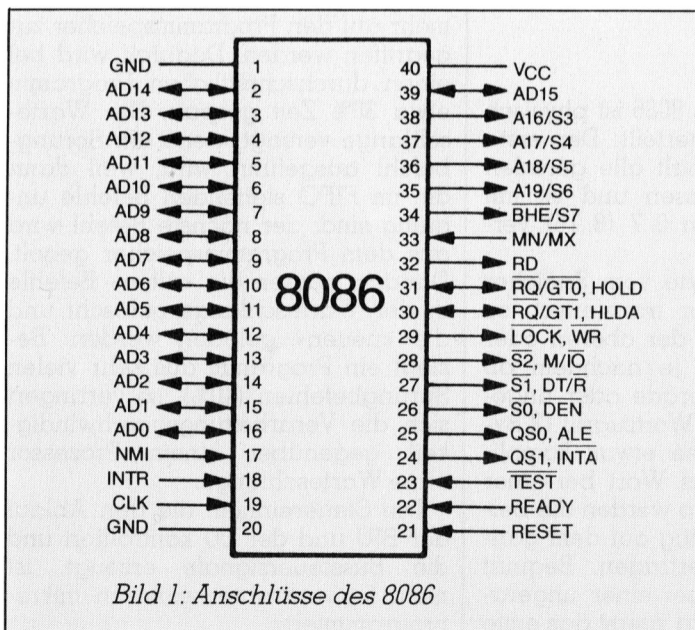


Bild 1: Anschlüsse des 8086

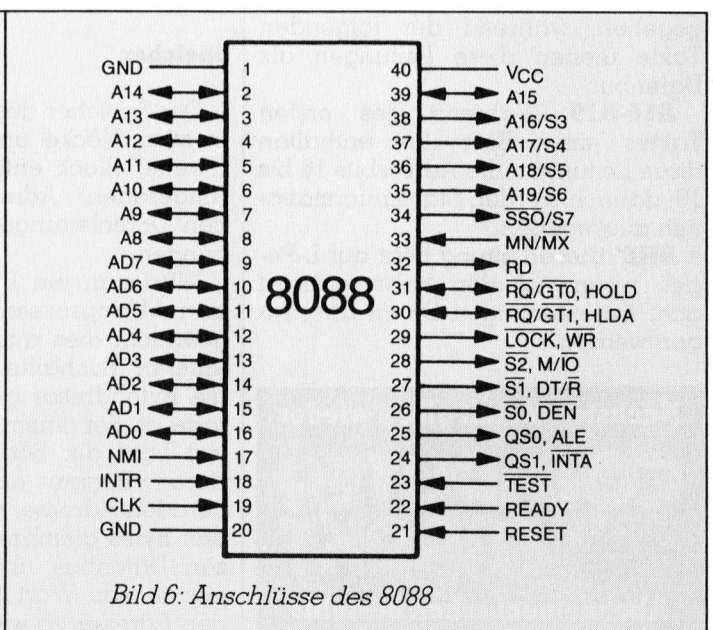


Bild 6: Anschlüsse des 8088

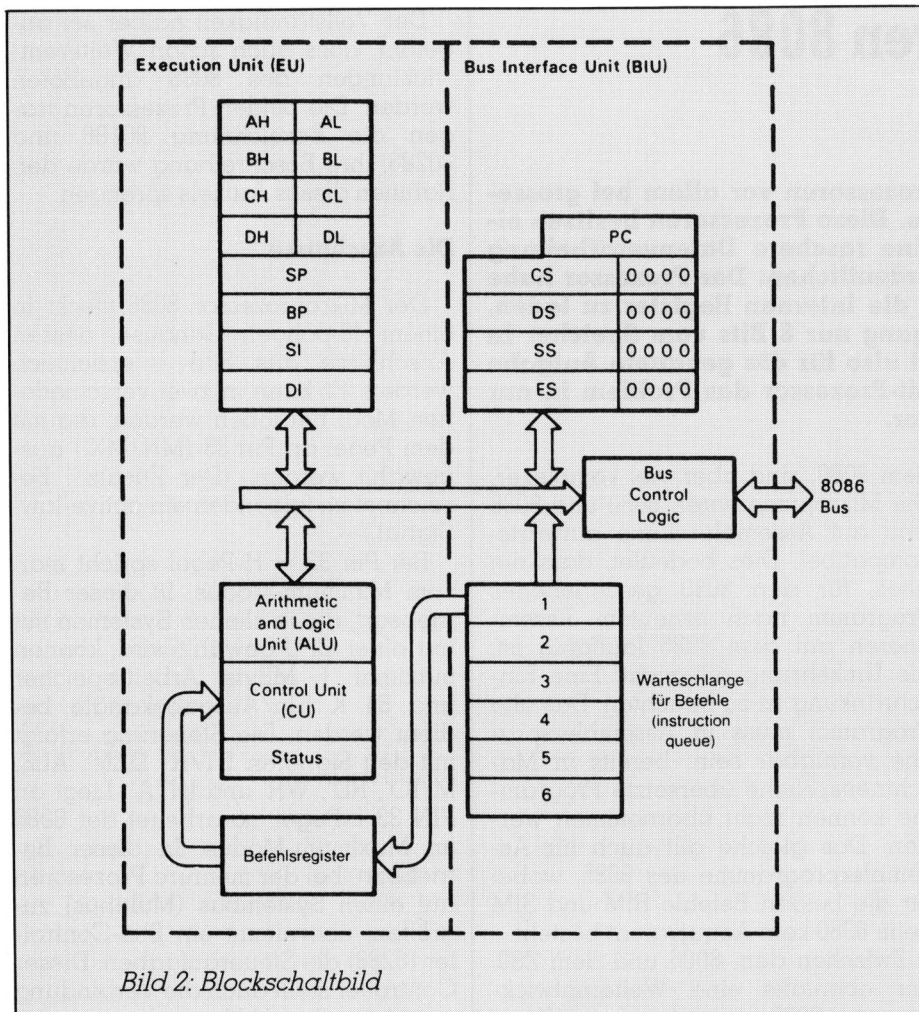


Bild 2: Blockschaltbild

Je nach der Betriebsart des 8086 haben die Signale verschiedene Bedeutung. Wir beschränken uns auf die Beschreibung der wichtigsten Minimum-Modus-Signale.

**AD0-AD15** Diese 16 Leitungen dienen als Adress- und Datenbus. Während des ersten Taktes eines Buszyklus werden 16 Adressbits ausgegeben, während der folgenden Takte dienen diese Leitungen als Datenbus.

**A16-A19** Während des ersten Taktes eines Buszyklus enthalten diese Leitungen die Adressbits 16 bis 19; danach werden Statusinformationen ausgegeben.

**BHE'** Diese Leitung liegt auf L-Pegel, wenn auf den höherwertigen acht Datenleitungen Daten übertragen werden.

**RD'** kennzeichnet einen Lese- und **WR'** einen Schreibzyklus.

**M/IO'** gibt an, ob auf den Speicher oder die Ein-/Ausgabekanäle zugegriffen wird.

**DEN'** und **DT/R'** steuern einen allfälligen Datenbustreiber; die restlichen Signale sind von anderen Prozessoren her bekannt.

## Speicher

Der Speicher des 8086 ist physisch in zwei Blöcke unterteilt: Der erste (zweite) Block enthält alle geraden (ungeraden) Adressen und ist mit den Datenleitungen 0..7 (8..15) verbunden.

Wird nun ein Byte vom Speicher zum Mikroprozessor transferiert, so geschieht dies auf der oberen oder unteren Bushälfte, je nachdem ob die Byteadresse gerade oder ungerade ist. Bei einem Wortzugriff (2 Bytes) wird die Sache etwas komplizierter: Beginnt das Wort bei einer geraden Adresse, so werden die beiden Bytes gleichzeitig auf dem ganzen Datenbus übertragen. Beginnt jedoch das Wort bei einer ungeraden Adresse, so wird zuerst das erste

Byte auf der oberen Bushälfte transportiert. Bevor das zweite Byte an die Reihe kommt muss die Adresse erhöht werden; erst dann wird das Byte auf der unteren Bushälfte transferiert. Bei einem Zugriff auf ein Wort, das an einer ungeraden Adresse gespeichert ist, werden also zwei Buszyklen benötigt.

Da der 8086 Codezugriffe immer wortweise ausführt, empfiehlt es sich, den ersten Befehl eines Programms auf eine gerade Adresse zu legen, damit die folgenden Zugriffe in einem Zyklus erfolgen.

Es ist jedoch möglich, dass durch Sprungbefehle die Sequenz geändert wird, wobei dann wieder zwei Zyklen pro Wortzugriff benötigt werden.

## Architektur

Wie aus dem Blockschaltbild ersichtlich ist (Bild 2), besteht der 8086 aus zwei verschiedenen Baugruppen: der Execution Unit (EU) und der Bus Interface Unit (BIU). Die BIU enthält die Schnittstelle zum Adress- und Datenbus, den Programmzähler, die Segmentregister, die Adressierlogik und eine sechs Bytes fassende Warteschlange. Die EU enthält Register für Daten und Adressen sowie die Logik für die arithmetischen und logischen Operationen.

Die beiden Blöcke arbeiten zeitlich unabhängig voneinander. Während die EU z.B. noch mit der Verarbeitung von Daten beschäftigt ist, lädt die BIU bereits die nächsten Befehle aus dem Programmspeicher in die Warteschlange (FIFO-Speicher). Verlangt nun die BIU einen neuen Befehl, so steht dieser bereits in einem internen Register; es muss nicht mehr auf den Programmspeicher zugegriffen werden. Dadurch wird bei einem durchschnittlichen Programm etwa 30% Zeit gespart. Die Warteschlange versagt, wenn ein Sprungbefehl ausgeführt wird, weil dann die im FIFO stehenden Befehle ungültig sind; der nächste Befehl wird aus dem Programmspeicher geholt. Danach müssen die «alten» Befehle in der Warteschlange gelöscht und die «neuen» geladen werden. Besteht ein Programm aus sehr vielen Sprungbefehlen (40%), so verringert sich die Verarbeitungsgeschwindigkeit gegenüber einem Prozessor ohne Warteschlange.

Die Steuereinheit, die den Ablauf der BIU und der EU kontrolliert und die Bussteuersignale erzeugt, ist nicht festverdrahtet, sondern mikroprogrammiert.

Nächsten Monat gibt's wieder

**COMPUTER  
MARKT**

Abonnement schon bestellt?

## Die Register

Der 8086 verfügt über vier Hauptregister, zwei Zeigerregister, zwei Indexregister, einen Programmzähler, vier Segmentregister und ein Statusregister, die alle 16 Bit breit sind (Bild 3).

**Die Hauptregister AX, BX, CX und DX** dienen zum Speichern der Operanden und Resultate von arithmetischen und logischen Operationen. Jedes dieser Register stellt eine Kombination aus zwei 8-Bit-Registern dar und kann entweder als 16-Bit- (z.B. AX) oder getrennt als zwei unabhängige 8-Bit-Register (AH und AL) adressiert werden. Dadurch ist es möglich, 8-Bit-Operationen mit 8-Bit-Registern durchzuführen, die weniger Zeit benötigen als 16-Bit-Operationen.

Jedes der Hauptregister hat daneben noch eine oder mehrere Sonderfunktionen:

- **AX** dient als Akkumulator; Ein/Ausgabeoperationen sowie einige spezielle Operationen werden darüber abgewickelt.
- **BX** wird u.a. bei der Bildung von Speicheradressen verwendet.
- **CX** dient als Zählerregister, mit dem einfache Programmschleifen aufgebaut werden können.
- **DX** nimmt Operanden und Resultate von Multiplikation und Division auf und enthält bei Ein/Ausgabeoperationen die Adresse des Ein/Ausgabekanals.

Über die beiden **Zeigerregister BP und SP** werden Speicherplätze im Stacksegment adressiert (BP = Base Pointer, SP = Stack Pointer). Ebenso können diese beiden Register zum Speichern von 16-Bit-Operanden benutzt werden.

Die beiden **Indexregister SI und DI** werden zur Adressierung von Stringelementen und zum Speichern von Operanden verwendet (SI = Source Index, DI = Destination Index).

Der **Programmzähler** enthält die Adresse des nächsten auszuführenden Befehls. Er wird mit PC (Program Counter) oder mit IP (Instruction Pointer) abgekürzt.

Die vier **Segmentregister CS, DS, SS und ES** werden zur Berechnung der Adressen herangezogen. Jedes dieser Register definiert im Speicher des 8086 einen 64 KBytes grossen Bereich.

Das CS-Register (Code Segment) bestimmt zusammen mit dem Programmzähler die Adresse des nächsten Befehls. Das DS-Register (Data Segment) wird zur Berechnung der

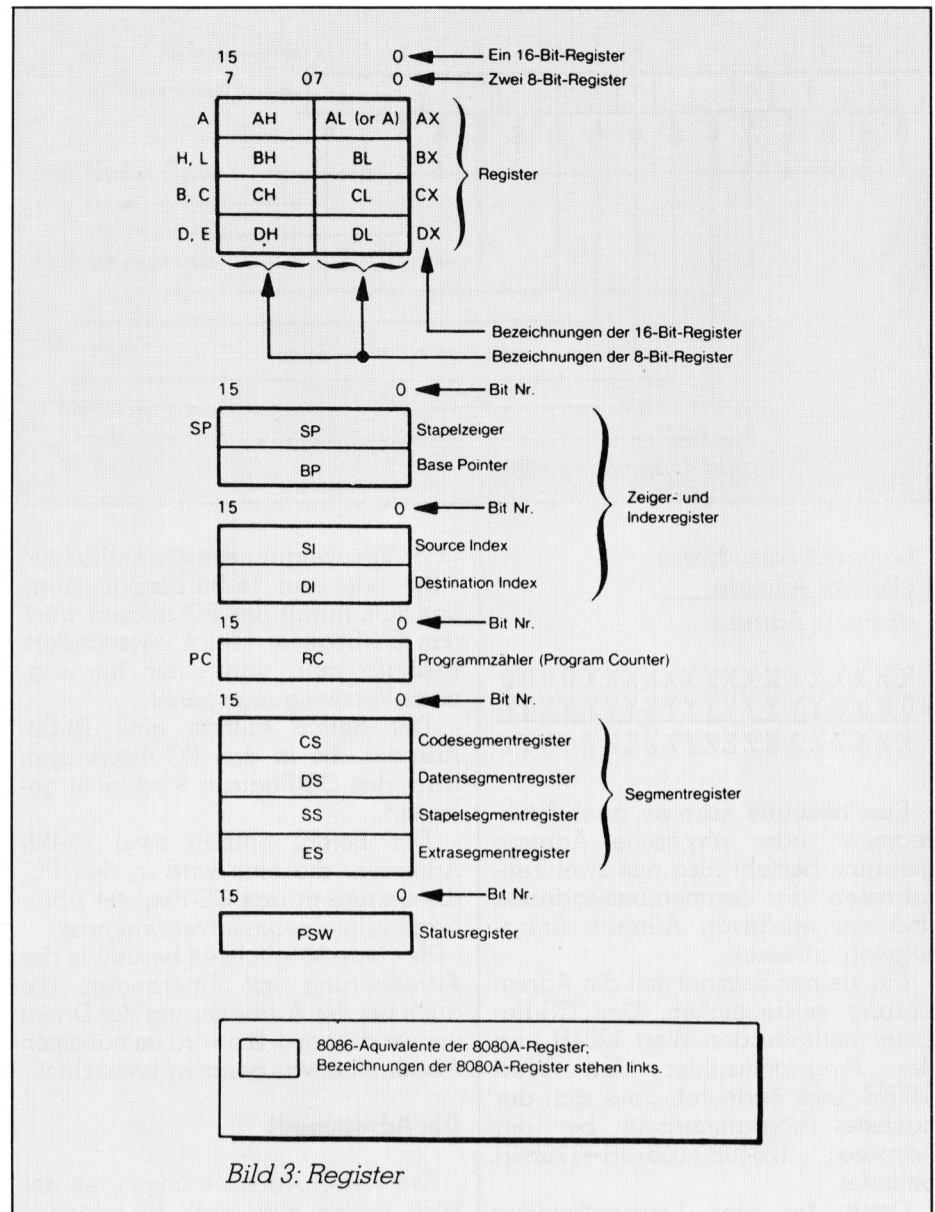


Bild 3: Register

Datenadresse und das SS-Register (Stack Segment) zusammen mit SP und BP zur Berechnung der Stackadresse verwendet. Das ES-Register (Extra Segment) dient zusammen mit dem DI-Register zur Adressierung der Operanden von Stringoperationen.

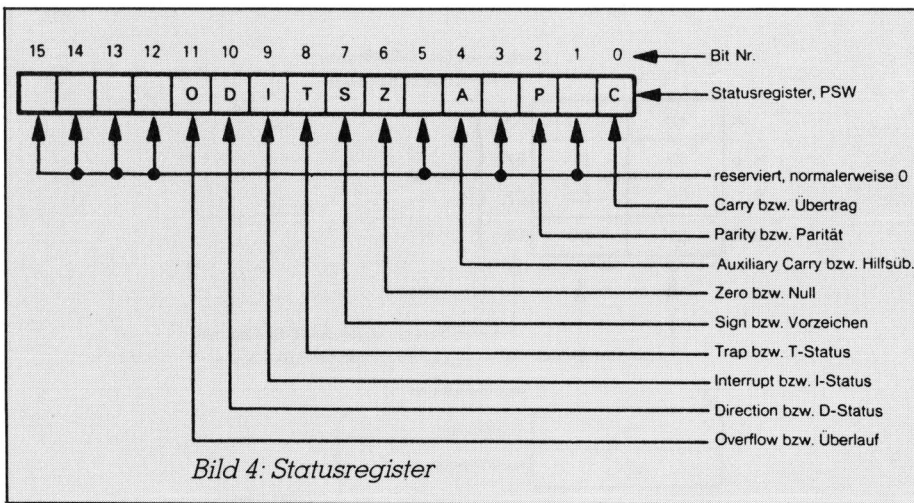
Das **Statusregister oder Prozessor-Status-Wort PSW** enthält Information über den Zustand des Prozessors (siehe Bild 4). Die niederwertigen acht Bits stimmen mit dem Flagregister des 8080 überein und benötigen keine weitere Erklärung.

Das I-Statusbit (Interrupt Flag) gibt an, ob maskierbare Interrupts verarbeitet werden sollen (I=1) oder nicht (I=0). Das T-Statusbit (Trap Flag) leistet beim Testen von Assemblerprogrammen eine grosse Hilfe, weil damit eine Art Einzelschrittausführung eingeschaltet werden kann (T=1). Der Wert des D-Statusbit

(Direction Flag) bestimmt, ob bei der wiederholten Ausführung von Stringbefehlen die Inhalte der SI- und/oder DI-Register automatisch aufwärts (D=0) oder abwärts (D=1) gezählt werden.

## Die Adressierung

Der Mikroprozessor 8086 kann einen Speicherbereich von  $2^{20} = 1'048'576$  Bytes adressieren, wozu 20 Adressleitungen benötigt werden. Die Adresse eines Speichers wird immer mit Hilfe eines Segment- und eines weiteren Registers berechnet. Dazu wird der Inhalt des Segmentregisters mit 16 multipliziert (d.h. die Dualzahl wird um vier Stellen nach links geschoben oder der Hex-Zahl wird eine Null angehängt) und zur Adresse des weiteren Registers dazugezählt:



Segmentbasisadresse  
+ effektive Adresse  
-----  
absolute Adresse

```
XXXXXXXXXXXXXXXXXXXX 0000
+ 0000YYYYYYYYYYYYYY YYY
XXXXXXXXZZZZZZZZZZZZ YYY
```

Eine absolute Adresse, auch Intersegment- oder physische Adresse genannt, besteht also aus zwei Teildressen: der Segmentbasisadresse und der effektiven Adresse (Intra-segmentadresse).

Ein kleines Beispiel soll die Adressierung verdeutlichen: Das CS-Register enthalte den Wert 1234H und der Programmzähler den Wert 5678H. Dies bedeutet, dass sich der nächste Programmschritt bei der Adresse 12340H+05678H=179B8H befindet.

Dank der vier Segmentregister kann der 8086 vier verschiedene 64 KByte-Blöcke (Segmente) adressieren, die sich auch überlappen können. Jedes der Segmentregister gibt also die Untergrenze eines 64 KByte-Blocks an, der irgendwo im adressierbaren Speicherbereich liegen kann. Daher ist der Speicher des 8086 nicht in feste 64 KByte-Blöcke (pages) unterteilt, wie dies bei den meisten anderen Mikroprozessoren der Fall ist.

Programme, die den Inhalt der Segmentregister nicht verändern, werden als dynamisch verschiebbar bezeichnet. Ein solches Programm kann beliebig im Speicherbereich verschoben und mit neuen Segmentregister-Inhalten gestartet werden.

Wie bereits erwähnt wurde, wird beim 8086 ein Befehl immer mit dem CS-Register und dem PC adressiert. Bei einer Programmverzweigung gibt es nun vier Adressierungsmöglichkeiten:

- Der Verzweigungsbefehl enthält ein 8-Bit- oder ein 16-Bit-Displacement, das zum Inhalt des PC addiert wird; das CS-Register bleibt unverändert, weshalb man von einer Intra-segment-Verzweigung spricht.

- Der Befehl enthält eine 16-Bit-Adresse, die in den PC übertragen wird; das CS-Register wird nicht geändert.

- Der Befehl enthält zwei 16-Bit-Adressen: die eine wird in den PC, die andere in das CS-Register übertragen (Intersegmentverzweigung)

- Die vierte Möglichkeit besteht in der Adressierung mit Operanden, die auch bei der Adressierung der Daten verwendet wird. Sie wird im nächsten Abschnitt etwas genauer betrachtet.

## Die Adressmodi

Der vielen Adressierungsarten des 8086 lassen sich grob in folgende sieben Gruppen unterteilen:

1. Register  
MOV AX,BX
2. unmittelbar  
MOV AX,1234H
3. direkt  
MOV AX,[1234H]
4. indirekt indiziert ohne Displacement  
MOV AX,[SI]
5. indirekt indiziert mit Displacement  
MOV AX,[SI]1234H
6. relativ zum BX-Register  
MOV AX,[BX]1234H
7. relativ zum BP-Register  
MOV AX,[BP]1234H

Beim ersten Beispiel wird AX mit dem Inhalt von BX, beim zweiten mit der Zahl 1234H und beim dritten mit dem Inhalt der Speicherstelle 1234H (relativ zum DS-Register) geladen. Beim vierten Beispiel wird die effektive Adresse vom Inhalt des SI-Reg-

isters gebildet, beim fünften wird das Displacement (1234H) zum Inhalt des SI- bzw. DI-Registers addiert. Das sechste Beispiel funktioniert gleich wie das fünfte, nur wird hier das BX-Register verwendet. Die bisher erwähnten Adressierungsarten berechneten die Adresse immer aus der effektiven Adresse und dem Inhalt des DS-Registers. Das siebte Beispiel verwendet nun das SS-Register als Basisadresse, womit man direkten Zugriff zum Stack-Segment hat.

Als weitere Adressierungsmöglichkeit kann der Anwender wählen, welches Segmentregister als Basisadressenregister dienen soll:

MOV AX,CS:BX

lädt den durch das BX-Register adressierten Wert im Codesegment in das AX-Register.

## Der Befehlssatz

Der 8086 versteht 106 verschiedene 1- bis 6-Byte-Befehle, die in sechs funktionelle Gruppen unterteilt werden können:

1. Datentransfers
2. Arithmetische Befehle
3. Logische Befehle
4. Stringmanipulation
5. Programmsteuerbefehle
6. Prozessorsteuerbefehle

Zur ersten Gruppe gehören u.a. MOV, PUSH, POP, IN und OUT. Der 8086 beherrscht die Addition, Subtraktion, Multiplikation und Division von 8-Bit- oder 16-Bit-Zahlen mit oder ohne Vorzeichen.

Bei der 16-Bit-Multiplikation resultiert eine 32-Bit-Zahl, bei der Division kann der Dividend ebenfalls 32 Bits lang sein. Die logischen Befehle funktionieren ähnlich wie bei anderen Prozessoren; als Besonderheit kann im CL-Register bei Mehrfach-schiebeoperationen die Anzahl Schiebezyklen festgelegt werden:

SHL AX,1  
schiebt AX um eine,

MOV CL,7  
SHL AX,CL

um sieben Stellen nach links.

Mit den Stringbefehlen können einfache Manipulationen an Byte- oder Wordstrings vorgenommen werden. Geht diesen Befehlen ein Wiederholungspräfix voraus, so werden die Operationen mehrfach aus-

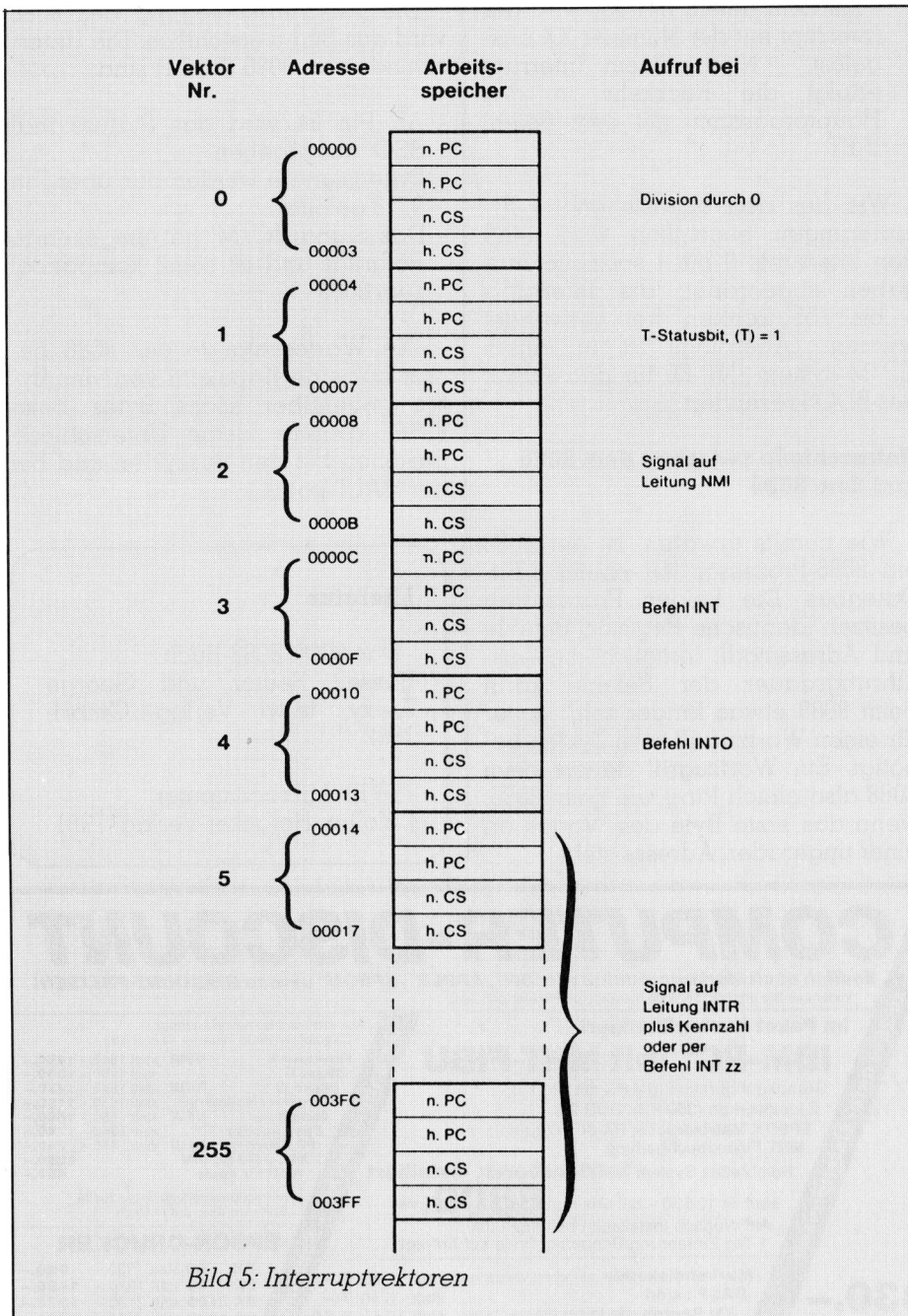


Bild 5: Interruptvektoren

geführt. Als Adressregister für den Quelloperanden dient das SI- und DS-, für den Zieloperanden das DI- und ES-Register.

```
MOV CX, 0FFFFH ;64 KWords
CLD ;aufwärts verschieben
```

```
MOV DX, 2000H
MOV DS, DX
MOV SI, 412CH
;Startadresse : 2412CH
```

```
MOV DX, 8000H
MOV ES, DX
MOV DI, 487CH ;Zieladresse : 8487CH
```

```
REP
MOVSW ;Verschiebe Wort von
      Start nach Ziel
      Dekrementiere CX,
```

zähle zu DI und SI zwei dazu und wiederhole MOVSW bis CX=0

Dieses kleine Programm verschiebt 64 KWords (128 KBytes) von 2412CH bis 4412BH nach 8487CH bis A487BH.

Nebst dem Blocktransfer können mit Stringbefehlen auch Blöcke verglichen oder mit einer Konstante geladen werden. Es ist auch möglich, in einem Block ein bestimmtes Element zu suchen.

Zu den Programmsteuerbefehlen gehören die Call-, Return- und Sprungbefehle, Interrupts, bedingte Sprünge und Wiederholbefehle. Mit Hilfe der letztgenannten Befehle können einfache Schleifen aufgebaut werden:

```
MOV CX, 1000
LABEL ... ;es folgen ein oder mehrere
          Befehle
```

```
LOOP LABEL ;dekrementiere CX und
           springe nach LABEL,
           falls CX nicht 0
```

Die letzte Gruppe bilden die Prozessorsteuerbefehle. Dazu gehören die Flagmanipulationen, sowie HLT, ESC, WAIT und LOCK. HLT bewirkt einen Haltzustand des Prozessors, der nur durch einen Interrupt oder Reset aufgehoben werden kann. Der ESC-Befehl dient dazu, Co-Prozessoren zu steuern (z. B. den 8087) und WAIT veranlasst den Prozessor zu warten, bis der TEST'-Eingang (PIN 23) aktiv wird. Der LOCK-Präfix wird in Systemen mit mehreren Prozessoren benützt; er bewirkt, dass der gemeinsame Datenbus während des folgenden Befehls für die anderen Prozessoren gesperrt ist.

## Interruptstruktur

Beim 8086 können maximal 256 Interruptvektoren gespeichert werden. Jeder Vektor besteht aus vier Bytes: die ersten beiden werden bei einem Interrupt in den Programmzähler, die anderen beiden in das CS-Register geladen. Die ursprünglichen Inhalte dieser zwei Register werden auf dem Stack gespeichert und nach Beendigung des Interrupts wieder geladen. Für die Interruptvektoren sind die Speicherplätze 00000H bis 003FFH reserviert (siehe Bild 5). Die Adresse des ersten Bytes eines Vektors erhält man, indem die Nummer des Vektors mit vier multipliziert wird.

Interrupts können wie folgt ausgelöst werden:

- 1) Durch Hardware:
  - Der Pegel an Pin 21 (RESET) wird high: sämtliche Bits des Statusregisters werden gelöscht, das CS-Register wird mit FFFFH und der Programmzähler mit 0000H geladen; das Programm wird bei der Adresse FFFF0H fortgesetzt.
  - Bei Division durch Null wird der Interrupt 0 ausgelöst
  - Wenn das T-Statusbit gesetzt ist, wird nach jedem Programmschritt Interrupt 1 ausgelöst. Dadurch kann ein Single-Step-Ablauf realisiert werden.
  - Wird der Pegel an Pin 17 high (NMI, Non Maskable Interrupt), so wird Interrupt 2 verarbeitet
  - Wenn das I-Statusbit gesetzt ist und der Pegel an Pin 18 (INTR, Interrupt Request) high wird, so

# KLEINCOMPUTER aktuell

bedeutet dies, dass von einem externen Gerät ein Interrupt angefordert wird. Der Mikroprozessor liest dann die Nummer des Interrupts von der unteren Hälfte des Datenbus und führt ihn aus. Diese Art von Interrupt ist sperrbar, indem das I-Statusbit gelöscht wird.

## 2) Durch Software

- Der Befehl INTO löst den Interrupt 4 aus, falls das Overflow-Flag gesetzt ist. - Der Befehl INT bewirkt das Verarbeiten des Interrupts 3

- Mit dem Befehl INT XX wird der Interrupt mit der Nummer XX ausgelöst. Nach einem Interrupt erfolgt die Rückkehr in das Hauptprogramm mit dem Befehl IRET.

Wie aus den obenstehenden Erläuterungen ersichtlich wird, sind den Interrupts 0 bis 4 spezielle Aufgaben zugeordnet; die Interrupts 5 bis 255 können frei verwendet werden. (Allerdings ist in einem CP/M-System INT E0 für den Aufruf des BDOS reserviert.)

### Unterschiede zwischen dem 8086 und dem 8088

Wie bereits erwähnt, ist der 8088 ein 8086-Prozessor mit einem 8-Bit-Datenbus. Die beiden Prozessoren besitzen identische Register, Befehle und Adressmodi; lediglich die Ausführungsdauer der Befehle kann beim 8088 etwas länger sein, da er für einen Wortzugriff zwei Zyklen benötigt. Ein Wortzugriff dauert beim 8088 also gleich lang wie beim 8086, wenn das erste Byte des Wortes an einer ungeraden Adresse steht.

Die Anschlussbelegung des 8088 wird aus Bild 6 ersichtlich. Die Unterschiede zum 8086 (Bild 1) sind:

- An Pin 34 wird das Statussignal SSO' ausgegeben
- Datensignale werden nur über Pin 9..16 geleitet
- Das Signal IO/M' hat umgekehrte Polarität und ist somit kompatibel zum 8085

Die Warteschlange des 8088 besitzt nur eine Kapazität von vier Bytes gegenüber sechs Bytes beim 8086. Weitere kleine Unterschiede bestehen bei den Buszyklen und bei der HALT-Logik. □

# 23.01.85

## Inserateschluss für die Februar- Ausgabe

### Literatur

- Das 8086/8088-Buch  
Russell Rector und George Alexy, te-wi Verlag GmbH, 1982

- 16 Bit Mikrocomputer  
J. Koller, Hofacker Verlag, 1981

**BERATUNG  
EINFÜHRUNG  
DEMONSTRATION**

**IBM-PC®**

Komplettes System mit 128 KB inklusive 2 Laufwerken (à 360 KB), Monitor, Tastatur DIN, DOS 2.0, mit allen Kabeln usw.

**betriebsbereit installiert**

statt Fr. 7148.- im Paket **6350.-**

**IBM XT®** mit 10 MB Harddisk und 1 Laufwerk (360 KB) Asynchr. Adapter, DOS 2.0 Komplettes System, wie oben **betriebsbereit installiert**

statt Fr. 12 030.- bei uns nur **10 830.-**

Wir lassen Sie auch nach dem Kauf nicht alleine! Auf Wunsch schulen wir Sie und Ihre Mitarbeiter bei uns oder an Ihrem Domizil. Preise inkl. Wust Änderungen/irrtümer vorbehalten. \* Apple und IBM sind geschützte Warenzeichen der Apple Computer Inc. resp. der International Business Machines Corp.

# COMPUTER-DISCOUNT

**Endlich auch Markenprodukte wie IBM®, APPLE®, EPSON® usw. zu DISCOUNT-PREISEN!**  
Verlangen Sie unsere Preisliste mit weiteren Paketangeboten.

**Im Paket noch günstiger!**

**IBM-PC® mit MRT-FIBU**

Grundkonfiguration IBM PC mit 128 KB, 2 Laufwerken (360 KB), DOS 2.0, EPSON Matrixdrucker RX-80 mit Kabel, MRT Finanzbuchhaltung

Komplettes System **betriebsbereit installiert**

statt Fr. 10 530.- bei uns nur **8800.-**

Auf Wunsch Installation bei Ihnen inkl. 1 Tag Einführung/Schulung. Preis auf Anfrage.

**Markendisketten** ss/dd mit Verstärkungsring ab **6.-**

**BASF** ss/dd statt 8.40 **7.-**

**3M Scotch** ds/dd für IBM-PC usw. statt 10.50 **8.50**

**EPSON-DRUCKER**

<b>RX 80</b>	statt 990.-	<b>920.-</b>
<b>FX 80</b>	statt 1690.-	<b>1480.-</b>
<b>FX 100</b>	statt 2190.-	<b>1970.-</b>
<b>LQ 1500</b>	statt 4290.-	<b>3860.-</b>

Aus unserem Software-Angebot: (Original verpackt, neueste Versionen!)

<b>Framework</b>	<b>NEU!</b>	statt 1848.-	<b>1450.-</b>
<b>dBase II</b>		statt 1290.-	<b>1050.-</b>
<b>dBase III</b>	<b>NEU!</b>	statt 1848.-	<b>1450.-</b>
<b>Wordstar / Mailmerge</b>		statt 1680.-	<b>1190.-</b>
<b>Symphony</b>	<b>NEU!</b>	statt 1850.-	<b>1690.-</b>
<b>Open Access</b>		statt 2340.-	<b>1780.-</b>
<b>PC + Master</b>	<b>NEU!</b>	statt 450.-	<b>380.-</b>
<b>MRT-FIBU für IBM</b>			<b>2250.-</b>
<b>FIBU für Apple</b>			<b>450.-</b>

Auftragsbearbeitung, individuelle Software usw. auf Anfrage.

**Garantie - Service - Schulung**

**Bis zu 25% unter Listenpreisen**

**BOROX-DATA AG** Schöneggstrasse 5, (5. Stock) 8004 Zürich, Tel. 01/241 61 26

Showroom offen:  
Montag-Donnerstag 9.00-12.00 13.30-18.00 Uhr  
Freitag durchgehend 9.00-15.00 Uhr



## EDV Kurse + Beratung

- Praxisnahe Ausbildung auf modernsten Computern APRICOT / HP 150 / ALPHATRONIC / KAYPRO
- Programmierkurse in BASIC (mit Fachausweis)
- Anwenderkurse in Textverarbeitung FIBU / LAGER / FAKTURIERUNG DATENBANK / SUPERCALC usw.

Die neuen Kurse beginnen am 2. 1. 85  
8 x 2 Std. = Fr. 350.- (eine Person pro Computer)  
Kleine Gruppen

**Beratung** für den Einsatz von Mikrocomputern im Büro und Gewerbe.

## MP Software

Rixheimerstr. 17, 4009 Basel  
Telefon 061/43 75 76, ab 18.30 Uhr 061/61 49 79

## TRICOM Miniplotter CPP-114

**Der Sharp PC-1500 war der erste Computer, der einen Vierfarbendrucker/-plotter in Miniaturausführung als periferes Gerät aufweisen konnte. Derzeit bietet nahezu jeder Hersteller von Pocket Computern so ein praktisches Zeichengerät an. Ohne grossen Aufwand sind diese Printer-Plotter, die recht sehenswerte Leistungen erbringen, an jeden Kleincomputer anschliessbar. Ein solches Gerät soll hier vorgestellt werden.**

Ist der Plotter des PC-1500 noch recht schmal, so bieten Canon oder Olivetti beispielsweise Modelle für doppelte Papierbreite an. Aber auch Casio, Tandy und Commodore haben solche Zeichenzwerg im Programm, die im wesentlichen alle das gleiche können - und dies ist nicht wenig.

Alle genannten Geräten haben eines gemeinsam: sie verwenden den Plotmechanismus von ALPS. ALPS ist

### Leopold Asböck

eine japanische Firma, die OEMs (Original Equipment Manufacturers) beliefert und das recht einfach konstruierte Plotwerk hat sich zu einem Bestseller entwickelt. Es wird in verschiedenen Breiten von ca 5 bis 20 cm produziert und besitzt eine Trommel, in die vier kleine Farb-Druckminen in Art von Kugelschreiberminen eingelegt werden können. Durch Programmbefehl lassen sich diese vier Farben wählen.

Der TRICOM CPP-114 misst nur 20x17,5x6 cm und ist mit 1,1 kg ein Leichtgewicht. Er ist in zwei Versionen erhältlich: die Standardversion besitzt einen Parallelschnittstellenanschluss und ist durch NiCd-Batterien netzunabhängig. Zum Laden der Batterien wird ein Netzadapter mitgeliefert. Eine zweite Version verfügt über Parallel- und Serienschnittstellenanschluss, kann aber nur über einen Netzadapter betrieben werden.

Der Printer/Plotter verwendet handelsübliches Rollenpapier von 114 mm Breite, auf die er im Textmo-

de 40 bzw. 80 Zeichen pro Zeile zu schreiben vermag. Durch einen DIL-Schalter auf der Geräteunterseite lässt sich der 40/80-Zeichenmodus einstellen. Ein zweiter DIL-Schalter ermöglicht die Wahl einer automatischen LINE FEED-Funktion.

Auf der Vorderseite befinden sich drei Folientasten, die auf Druck Stiftwechsel, Farbwechsel oder Papieranschub bewirken. Hält man beim Einschalten des Plotters die Papieranschubtaste gedrückt, so wird ein Selbsttest ausgeführt, wobei ein Teil des Zeichensatzes in allen vier Farben ausgedruckt wird.

Eine grüne Leuchtdiode zeigt den Betriebszustand an. Sie beginnt zu blinken, falls die Ladung der eingebauten Batterien zu Ende geht.

Die Papierrolle wird durch einen Drahtbügel vor dem Herausfallen gesichert, zudem sorgt der Bügel dafür, dass das Papier, das aus dem Plotter kommt, nicht versehentlich hinten wieder hineingezogen wird.

Auf der rechten Seite ist der Parallelschnittstellenanschluss für acht Datenbits, ein Strobosignal, Acknowledge und Busy. Da der Anschluss 36-polig ist,

nimmt er viel Platz neben dem Ein/Aus-Schalter des Gerätes ein.

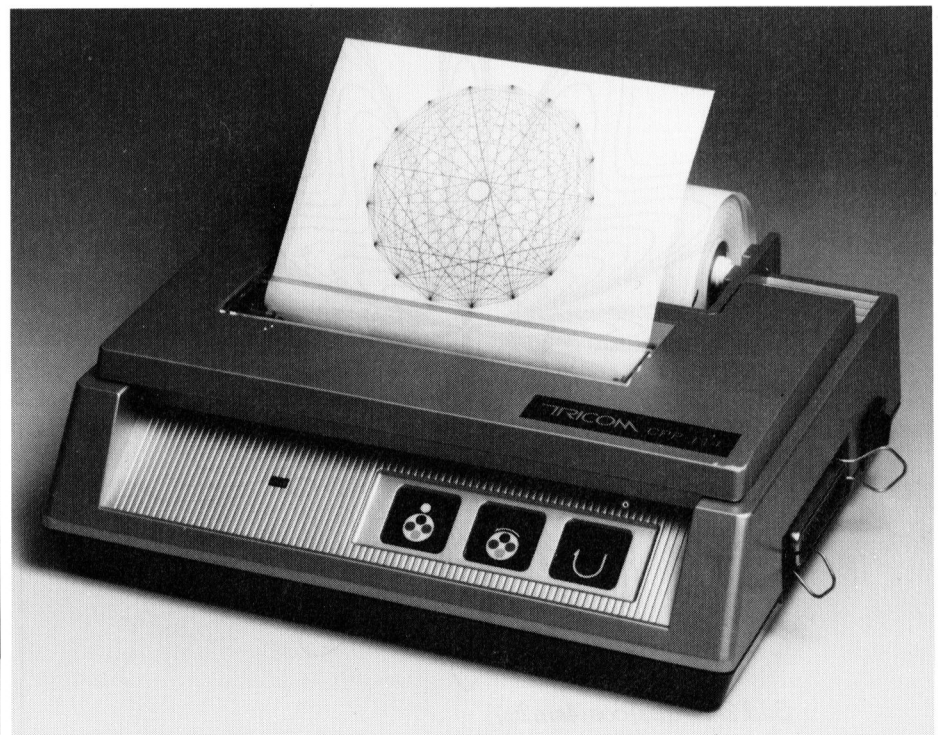
Dass in diesem kleinen Gerät noch intelligente Elektronik Platz findet, ist verwunderlich. Die Platine ist entsprechend klein, auf ihr befinden sich nur wenige Bauteile: die Interfaceschaltkreise, die Komponenten für die Batterieladung - und der Einchipcomputer. Es handelt sich um einen Hitachi HD44860A22, einen Winzling im quadratischen Gehäuse, der seine 53 Beinchen von sich streckt. Er steuert den Vierfarbplotter, und dass in seinem Programm viel Hirn steckt, zeigt der Befehlssatz:

### Printermode

Der CPP-114 arbeitet als Printer oder als Grafikplotter. Beim Einschalten ist er auf Printerbetrieb gestellt und je nach Stellung des schon erwähnten DIL-Schalters schreibt er 40 oder 80 Zeichen pro Zeile, wobei man bei 80 Zeichen etwas Mühe mit dem Lesen hat.

Er akzeptiert den ASCII-Code für alphanumerische Zeichen und Sonderzeichen, an Controlcodes verarbeitet er BACK SPACE (08), LINE FEED (0A), LINE FEED-BACK (0B), sowie 12 (hexadezimal) um in den Grafikmode zu wechseln. Mit 11 (hexadezimal) kehrt er aus dem Grafikmode in den Textmode zurück.

Da der Printer 8-bit-Daten verarbeitet, ist auch noch ein Satz japanischer Zeichen greifbar, mit denen der europäische Anwender aber recht wenig anfangen wird. Der ge-



**C/NORM**  
Versand  
Melonenstrasse, 9001 St.Gallen

**ENORME HITPREISE**

**FÜR COMPUTER**

**071 22 63 22**

```

! " # $ % & ' ( ) * + , -
. / 0 1 2 3 4 5 6 7 8 9 :
; < = > ? @ A B C D E F G
H I J K L M N O P Q R S T U
V W X Y Z [ \ ] ^ _ ` a b
c d e f g h i j k l m n o
p q r s t u v w x y z { | }
~ ☒
    
```

Bild 1: Zeichensatz des CPP-114

samte Zeichensatz ist in Bild 1 dargestellt. Um die diversen Anwendungen zu demonstrieren, wurden alle Abbildungen und Listings mit dem TRICOM erstellt.

Im Textmode verhält sich der Printer wie ein gewöhnlicher Drucker, er «malt» mit ca. 12 Zeichen pro Sekunde den gewünschten Text auf das Papier.

## Grafikmode

Im Grafikmode werden die Grafikbefehle im ASCII-Code an den Plotter übermittelt, dies gestattet ein recht einfaches Arbeiten mit dem Computer, vor allem in BASIC-Programmen. Die 32-seitige Broschüre demonstriert an kurzen Beispielen die Wirkung der Befehle. Jeder Befehl wird als Textstring übermittelt, wobei der erste Buchstabe als Befehl interpretiert wird:

### ALL INITIATION «Aabc...»

Befindet man sich im Grafikmode, so bewirkt dieser Befehl die Rückkehr in den Textmode und druckt den nachfolgenden Text aus.

### COLOR CHANGE «Cn»

C0 wählt den schwarzen, C1 den blauen, C2 den grünen und C3 den roten Stift. Dazu fährt die Plottertrommel an den linken Rand und über einen Magneten wird die Trommel gedreht, bis der gewünschte Farbstift in die oberste (Schreib-)Lage kommt.

### DRAW ABSOLUT «Dx,y,x,y...»

Der Schreibstift verbindet die Punkte, deren absolute Koordinaten durch die Paare x,y gegeben sind. Der Koordinatenbereich beträgt in x- wie in y-Richtung von -999 bis +999, wobei aber in x-Richtung nur 480 Werte ansprechbar sind. Liegt eine

Koordinate ausserhalb dieses Bereiches, so führt der Plotter die Bewegungen mit abgehobenem Stift am Rand durch, d.h., es wird eine automatische Randbegrenzung ausgeführt, ohne am Rand hin und her zu schmierern.

### DRAW RELATIVE «Jx,y,x,y,...»

Hier gelten dieselben Voraussetzungen wie für das Zeichnen mit absoluten Koordinaten, ausser dass die x,y-Werte als relative Koordinaten - also immer bezogen auf den letzten Punkt als Ursprung - betrachtet werden.

### ABSOLUT MOVEMENT «Mx,y»

setzt den Zeichenstift ohne eine Linie zu ziehen auf den Punkt x,y des Koordinatenbereiches.

### RELATIV MOVEMENT «Rx,y»

bewegt den Zeichenstift ohne eine Linie zu ziehen um die relativen Koordinaten x,y weiter.

### HOME «H»

führt den Zeichenstift auf den Ursprung.

### INITIATION «I»

wählt die momentane Zeichenstiftposition als Ursprung.

### LINE TYPE «Ln»

wählt eine aus 16 verschiedenen Linienarten, n steht für eine Zahl von 0 bis 15.

### PRINT «Pabc...»

Um im Grafikmode schreiben zu können, verwendet man den Befehl P, gefolgt vom gewünschten Text. Zuvor kann der Zeichenstift (mit MOVE) zu dem Punkt bewegt werden, ab dem die Zeichen geschrieben werden sollen.

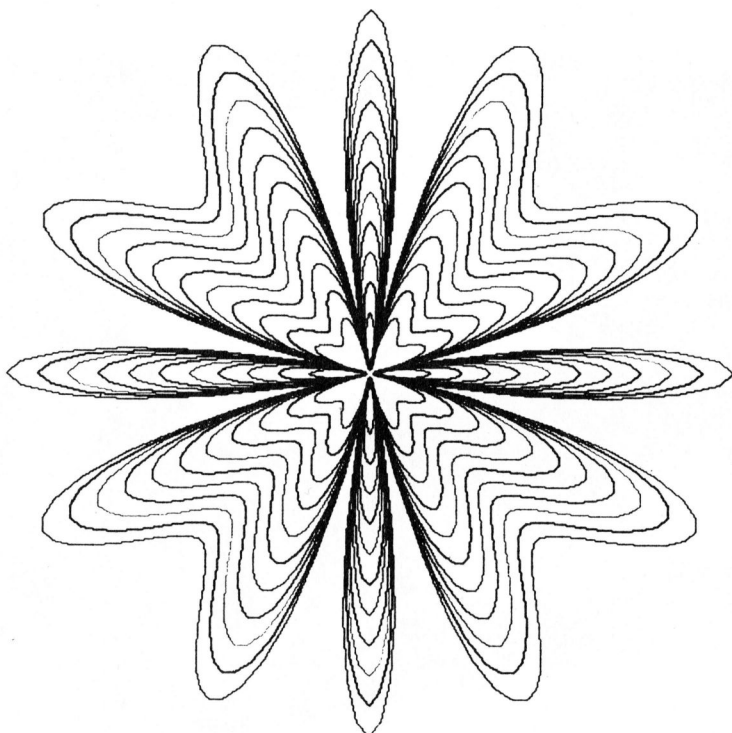


Bild 2:  $r = \text{abs}[\cos(4\sin 2\phi)]$

## Hat im März dieses Jahres ...

die PTT mit einer kräftigen Taxerhöhung hingelangt, so steht uns jetzt eine weitere und sehr massive Produktionsverteuerung ins Haus (allein das Papier ist in diesem Jahr um 20% teurer geworden). Das zwingt uns, den seit 1979 unveränderten Abopreis leicht anzupassen. Ab 1. 1. 1985 kostet das M+K-Jahresabo neu Fr. 42.--, inkl. Porto und Versandkosten. Wir zählen auf Ihr Verständnis und danken Ihnen für Ihre Treue zu M+K.

Redaktion und Verlag



## SCALE SET «Sn»

Die Zeichen, die mit dem PRINT-Befehl geschrieben werden, können in 16 Grössen definiert werden. Dazu wird S mit einer Zahl von 0 bis 15 an den Plotter ausgegeben.

## ROTATION «Qn»

Jeder Text kann im Grafikmode nicht nur in vier Farben und in sechzehn Grössen, sondern auch in vier Richtungen (jeweils um 90 Grad gedreht) geschrieben werden. Q0 schreibt von links nach rechts, Q1 von oben nach unten, Q2 von rechts nach links und Q3 von unten nach oben.

## Zusammenfassung

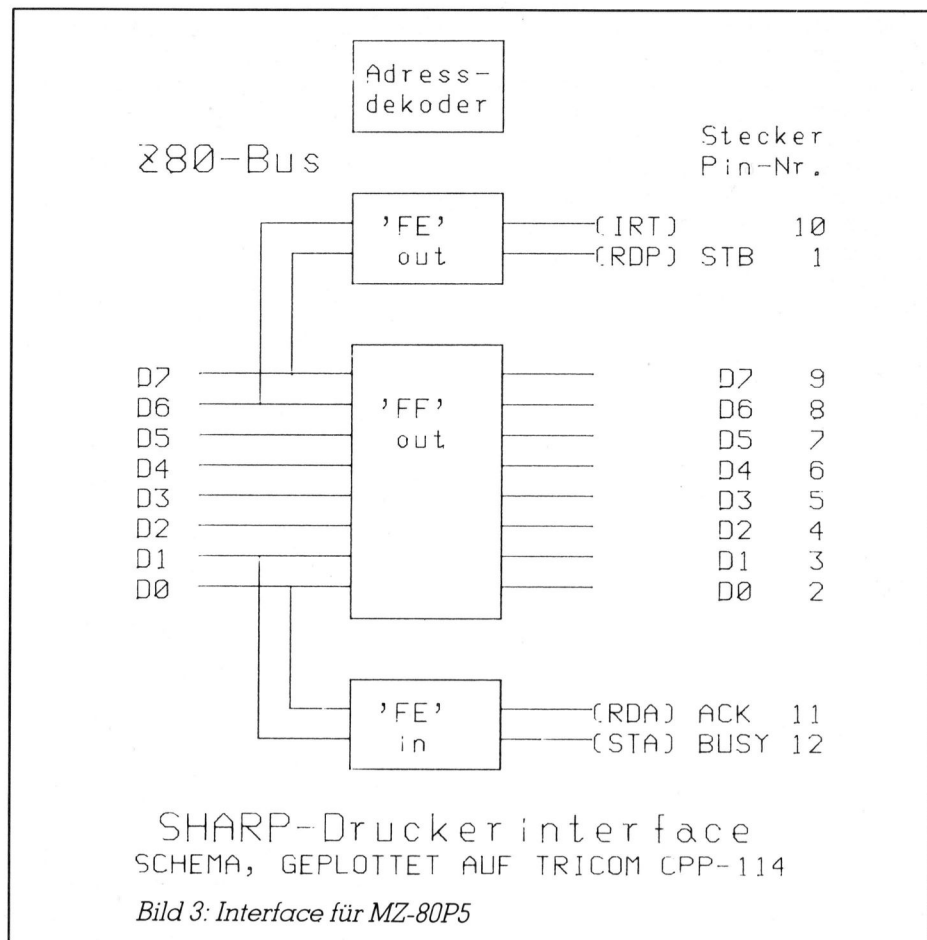
Die Netzunabhängigkeit des Printer/Plotters ist besonders bei tragbaren, batteriebetriebenen Computern ein nicht zu unterschätzender Vorteil, hat man dadurch auch ohne Steckdose stets einen Drucker für Programmlistings oder andere Hardcopies bei der Hand und dazu erst noch ein flottes Zeichengerät, das in vier Farben plottet. Bei einer Zeichengeschwindigkeit von 52 mm bis 73 mm pro Sekunde ist häufig der rechnende Computer der langsamere Partner, wie zum Beispiel in Bild 2, wenn Grafikkordinaten berechnet werden müssen.

Bei 2,2 Watt Leistungsaufnahme reicht die Batterieladung für eine volle Stunde intensiver Zeichen- oder Schreibearbeit. Papier und vier Farbstifte liegen dem Plotter bei und sind im Fachhandel leicht erhältlich. Ein Ladegerät und eine «possierliche» Staubschutzhaube sowie ein kleines Manual mit instruktiven Beispielen sind gleichfalls im Kaufpreis enthalten.

## Anschluss an einen Computer

Der TRICOM besitzt einen Centronics-ähnlichen Parallelanschluss. Im folgenden sollen einige Möglichkeiten gezeigt werden, wie man diesen Plotter/Printer an einen Computer anschliessen kann. Konkret soll er an den Sharp MZ80B adaptiert werden, doch ist eine Lösung für andere Geräte leicht zu finden.

Der Anschluss an einen Computer setzt eine Parallelschnittstelle voraus. Entweder ist eine solche vorhanden oder kann durch Kauf oder Eigenbau leicht ergänzt werden. Elf Signale sind notwendig: 8 Datenbits D7 bis D0, ein Strobosignal STR, als Rückmeldungen das Acknowledge-signal ACK (Daten wurden ange-



nommen) und BUSY (Gerät ist beschäftigt, Datenannahme ist nicht möglich). Für die neun Ausgangssignale benötigt man zwei Ausgabeports: ein 8-bit-Port für die Daten und ein 1-bit-Port für das STROBE-Signal. Will man sparsam sein, so kann man sich mit 7 Datenbits begnügen (ausser man will die japanischen Zeichen auch drucken) und Datenbit 7 auf logisch 0 legen. Dafür kann bei der Ausgabe das STROBE-Signal auf Bit 7 des Ausgabeports gelegt werden und man spart ein Ausgabeport für dieses einzelne Bit. Zur Rückmeldung benötigt man ein Port für die beiden Signale ACK und BUSY.

Wie schliesst man den Plotter an den Computer an? Konkreter Fall: Z80-Bus, noch konkreter: Sharp MZ80B.

Ein Parallelinterface für den Z80-Bus ist keine Hexerei, wer ein bisschen Hardwarekenntnisse besitzt, bringt es aus einigen 74LS...-Schaltkreisen leicht zustande. Als Hilfestellung sei die Schaltung in M+K 82-2, EPROM-Lesegerät, angeführt. Zwei Input- und zwei Output-Ports reichen bei weitem für einen Anschluss. Fertig gibt es dieses Interface als Universalinterface MZ80IO2 für den Sharp MZ80B. Wer allerdings bereits einen Drucker (z.B. MZ80P5) am

MZ80B laufen hat, kann auch das Druckerinterface heranziehen, da Plotter und Drucker mit Parallelanschluss gleiche Bedingungen voraussetzen.

Das Druckerinterface für den MZ80P5 hat allerdings etwas anders geartete Signale, da diverse Druckerfunktionen rückgemeldet werden. Für den Betrieb des Plotters wird aber dieses Interface verwendet und die Software adaptiert.

Kabel zwischen MZ80P5-Printerkabel und TRICOM

25-pol		36-polig	
STB	1	-----	1 STB
D0	2	-----	2 D0
D1	3	-----	3 D1
D2	4	-----	4 D2
D3	5	-----	5 D3
D4	6	-----	6 D4
D5	7	-----	7 D5
D6	8	-----	8 D6
D7	9	-----	9 D7
	10		
ACK	11	-----	10 ACK
BUS	12	-----	11 BUSY
GND	13	-----	19 GND
GND	14	-----	20 GND

*Bild 4: Kabel für den Plotter*

Das Druckerkabel besitzt zwei 25-polige Stecker, der Plotter allerdings einen 36-poligen anderer Bauart. Ein passendes Kabel ist aber schnell gelötet: man braucht einen 25-poligen Stecker wie sie auch für RS232-Anschlüsse üblich sind und einen 36-poligen Stecker (Amphenol), dazu ein Kabel mit mindestens 12 Adern, eventuel ein 14-adriges Flachbandkabel von geeigneter Länge.

```
60000 REM Druckerdriver-Aenderung zum
60010 REM Anschluss eines
60020 REM -----
60030 REM TRICOM CPP-114 - Plotters
60040 REM -----
60050 REM an das Druckerinterface
60060 REM des SHARP MZ-80B
60070 REM
60080 REM L.Asboeck, 12.8.1984
60090 REM
60100 RESTORE 60250: GOTO 60110
60101 RESTORE 60300: REM Zeilennummer!
60110 ADRESSE=18100: I=0
60120 READ A$
60130 IF A$="FINITO" GOTO 60240
60140 A1$=LEFT$(A$,1)
60150 A2$=RIGHT$(A$,1)
60160 A1=ASC(A1$)-48
60170 IF A1>9 THEN A1=A1-7
60180 A2=ASC(A2$)-48
60190 IF A2>9 THEN A2=A2-7
60200 A=16*A1+A2
60210 POKE ADRESSE+I, A
60220 I=I+1
60230 GOTO 60120
60240 END
60250 DATA 3E,0D,F5,DB,FE,CB,4F,20
60260 DATA FA,F1,D3,FF,3E,80,D3,FE
60270 DATA DB,FE,CB,47,28,FA,AF,D3
60280 DATA FE,C9, FINITO
60290 REM
60300 DATA 3E,1B,F5,CD,CA,46,F1,D3
60310 DATA FF,3E,80,D3,FE,3E,01,CD
60320 DATA CB,46,AF,D3,FE,C9,AF,C5
60330 DATA D5,4F, FINITO
60340 REM
60350 REM --- Wiederherstellung des
60360 REM --- Originaldruckerdrivers
60370 REM --- mit 'RUN 60101'
```

Listing 1: Plotterdrive für Sharp-BASIC 5510

```
46B4 3E 0D      LD  A,0D      ;Accumulator mit Byte laden
46B6 F5        PUSH AF      ;zwischenspeichern
46B7 DB FE BUSY:IN  A,(FE)    ;testen auf BUSY
46B9 CB 4F     BIT  1,A      ;(Bit 1)
46BB 20 FA     JR   NZ,BUSY  ;sonst Schleife
46BD F1       POP  AF      ;rueckspeichern
46BE D3 FF     OUT  (FF),A   ;Ausgabe
46C0 3E 80     LD  A,80H    ;Strobeimpuls laden (Bit 7)
46C2 D3 FE     OUT  (FE),A   ;und ausgeben
46C4 DB FE ACK: IN  A,(FE)    ;Acknowledge abwarten
46C6 CB 47     BIT  0,A      ;(Bit 0)
46C8 28 FA     JR   Z,ACK    ;sonst Schleife
46CA AF       XOR  A        ;Accu=0, Strobeimpuls
46CB D3 FE     OUT  (FE),A   ;beenden
46CD C9       RET          ;Ruecksprung
```

Listing 2: Z80-Code für den Plotterdriver

Die Ausgangssteckerbelegung des Druckerinterfaces ist aus Bild 3 ersichtlich. Die Verbindungen für das neue Anschlusskabel sind in Bild 4 dargestellt.

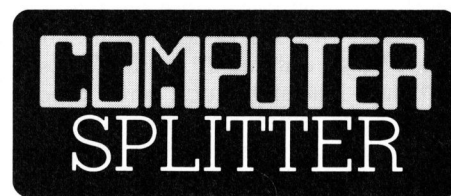
## Software

Die Software des MZ80B fragt die Druckerbereitschaft recht aufwendig ab (Busy, Papierende, Störung etc.). Für den Plotter ist dies nicht notwendig, es genügt die Abfrage der Signale ACK und BUSY. Dies lässt sich durch INs und OUTs in BASIC machen, das ist aber eine unelegante Methode und bedingt den dauernden Aufruf von Unterprogrammen, Programm listings können zudem keine erstellt werden. Besser ist es, die Original-Maschinenprogrammroutine durch ein neues, kurzes Programm zu ersetzen, das auf den Plotter zu rechtgeschnitten ist. In M+K 84-3, Controlcodes für SHARP-Drucker, wurde schon näher auf diese Ausgaberroutine des BASIC-Interpreters SB-5510 eingegangen. Ab Adresse 46B4 befinden sich Unterprogramme im Speicher, die für den Verkehr mit dem Drucker verantwortlich sind. Auf diese greifen die Befehle PRINT/P und LIST/P (in anderen BASIC-Versionen LPRINT und LLIST) zu.

Setzt man hier ein neues Programm ein, so führen die genannten Befehle den richtigen Transfer zum Plotter durch. Im Listing 1 ist ein kurzes BASIC-Programm dargestellt, das eine entsprechende Maschinensprachroutine lädt. Es muss, solange man den Plotter betreibt, nur einmal laufen (RUN 60100). Dann wird die Originalroutine für den Drucker von der neuen Routine für den Plotter überschrieben. Will man wieder den Drucker betreiben, wird mit RUN

60101 der ursprüngliche Zustand wieder hergestellt. In Listing 2 ist die Maschinensprachroutine ausführlich aufgelistet, sie kann für eigenentwickelte Interfaces oder an jeden Z80-Computer leicht übertragen werden.

Wie im angesprochenen Artikel über Sharp-Drucker-Codes bereits erwähnt, «schluckt» der BASIC-Interpreter einige Controlcodes, ohne sie weiterzugeben. Man behilft sich dann mit «direkter» Uebertragung, also beispielsweise statt PRINT/P CHR\$(18) verwendet man POKE 18101,18:USR(18100):POKE 18101,13. Dadurch wird die Maschinensprachroutine direkt aufgerufen, ohne dass der Interpreter die Controlcodes zensurieren kann. □



## Hardware-Tips

(288/fp) ... für HP-Geräte gibt eine alle zwei Monate erscheinende Broschüre «Bench Briefs». Die enthaltenen Informationen beziehen sich auf Service-Tips, Hinweise für Geräte-Modifikationen, neue Testmethoden und Hilfsmittel zur Fehlersuche. Sie richten sich in erster Linie an Service-Techniker, dürften aber auch für den einen oder anderen Hardware-Freak von Interesse sein. Die «Bench Briefs» können unentgeltlich bei HP bezogen werden. □

## Astro-look für HHC aus Japan

(280/fp) Der japanische Hersteller Ampere hat einen HHC mit etwas futuristischem Design unter dem Namen Big.APL vorgestellt. Das Gerät wird gesteuert von einem HD68000 Mikroprozessor (!), der Arbeitsspeicher beläuft sich in der Grundversion auf 64 KBytes und das ROM auf 128 KBytes. Das Betriebssystem verfügt über einen APL-Interpreter, Echtzeit-Uhr, Kalender, Datei-Handhabung, Text-Editor und, und, und ... Es ist multitasking und multiuser-fähig und hat als solches mehrere Fenster mit voller Editierbarkeit! Das Kassettenlaufwerk (300 KBytes) verwaltet Daten, Programme und analoge Informationen. Wir vermuten, dass neben dem Design noch etwas anderes astronomisch ist ... □

# SANYO PC

## Der logische Weg in die Zukunft



### SANYO Computer MBC-555 ... auf Leistung und Wachstum programmiert!

Wenn Sie sich heute für einen Computer entscheiden, so denken Sie doch auch an die Zukunft! Dass der Computer auch nach Jahren noch ausgebaut und erweitert werden kann.

Bei Sanyo müssen Sie sich nicht heute den Kopf darüber zerbrechen, ob Sie morgen vielleicht Farbgrafik benötigen. Die interne Speichererweiterung, den Farbmonitor, die zweite Diskettenstation oder gar das Hard Disk-Laufwerk setzen Sie erst dann ein, wenn Sie es wirklich brauchen.

Entscheiden Sie sich also für eine flexible und leistungsfähige Lösung. Für Sanyo PC MBC-555.

Die für alle Systemvarianten gleiche Grundausstattung umfasst den modernen 16-Bit Mikroprozessor 8088, 128 KBytes interne Speicherkapazität (ausbaufähig auf 256 KBytes), ein oder zwei Disketten-Laufwerke mit je 360 KBytes Aufzeichnungskapazität, hochauflösende Grafik mit 640 x 200 Bildpunkten (wahlweise mit 8 Farben) sowie das moderne Betriebssystem MS-DOS (Version 2.11).

Ein serielles Interface und eine parallele Schnittstelle (seriell erweiterbar) erlauben den Anschluss unzähliger Peripheriegeräte.

Die moderne und leistungsfähige Hardware wird mit massgeschneiderter Software ergänzt, die auch für EDV-Laien leicht verständlich ist. Das von Sanyo speziell für die Schweiz

entwickelte reichhaltige Software-Angebot für privaten und kommerziellen Einsatz ist in deutscher oder französischer Sprache erhältlich. Dank dem IBM®-kompatiblen Betriebssystem MS-DOS kann ausserdem ein reichhaltiges Angebot an Standardsoftware aus dem freien Markt problemlos eingesetzt werden. Und eine über die ganze Schweiz verteilte Beratungs- und Wartungs-Organisation garantiert hohe Einsatzsicherheit.

#### SANYO PC MBC-555 Das ideale System für anspruchsvolle Einsteiger und Profis.

Preisbeispiele: SANYO PC, MBC-555: Interner Speicher 128 kB, 2 Diskettenlaufwerke mit je 360 kB, Monochrom-Monitor 12":

mit Farbmonitor

Fr. 3990.-

Fr. 4990.-



#### Autorisierte Wiederverkäufer SANYO MBC 555

**Aarau:** Rediffusion AG/**Aigle:** Maflioly SA/**Allschwil:** Thürlimann-Discount AG/**Andwil SG:** Allelectronic/**Bachenbühlach:** Rediffusion AG/**Basel:** BD-Electronic, Rediffusion AG, Thürlimann-Discount AG, Zihlmann & Co/**Bern:** Eschenmoser AG, Jelmolli SA, Rediffusion AG, R. Von May/**Biel:** Rediffusion AG/**Bulle:** Maflioly SA/**Burgdorf:** Rediffusion AG/**Château d'Oex:** Delay SA/**Clarens:** Maflioly SA/**Dübendorf:** Tele Locher AG/**Effretikon:** Rediffusion AG/**Emmen:** Léon Nordmann & Cie/**Fribourg:** La Plac. Nordmann & Cie, Delay SA/**Genf:** Bug Electronique, Filippone A., Gd. Mag. La Placette SA, Gd. Passage/**Haag:** Rediffusion AG/**Heerbrugg:** Rediffusion AG/**lbach-Schwyz:** Rediffusion AG/**Jona:** Hans Brunner/**Langendorf:** Rediffusion AG/**Lausanne:** Gd. Mag. Innovation SA, Horba Micro-Computer SA, Maflioly SA, Delay SA/**Luzern:** Rediffusion AG/**Martigny:** Delay SA/**Mels-Sargans:** Rediffusion AG/**Monthey:** Gd. Mag. La Placette SA, Delay SA/**Montreux:** Maflioly SA/**Morges:** Maflioly SA/**Neuchâtel:** Delay SA/**Rapperswil:** Rediffusion AG/**Schönenwerd:** TV Lehmann AG/**Sion:** Clausen-Discount/**St. Gallen:** Rediffusion AG/**Vallorbe:** Delay SA/**Vevey:** Delay SA, Maflioly SA/**Villars:** Delay SA/**Volketswil:** Rediffusion AG/**Wädenswil:** Rediffusion AG/**Wül:** Rediffusion AG/**Winterthur:** Rediffusion AG/**Yverdon:** Delay SA/**Yvonand:** Delay SA/**Chur:** Rediffusion AG/**Zürich:** Eschenmoser AG, Jelmolli SA, Micomp SMS AG, Radio Münsterstern AG, Rediffusion AG, Service Technik, Kaufhaus Vilan AG.

# Der Neugeborene ist ein Erwachsener!

Auf dem Gebiet der professionellen Mikrocomputer lanciert Sumicom den PRO-16, einen 16-Bit-Computer, dessen Verarbeitungskapazität weit über den meisten Systemen seiner Kategorie liegt. Mikroprozessor INTEL 80186, 256 kB-Hauptspeicher, ausbaubar bis 896 kB.

Variante: der PRO-16 X mit einer 5 1/4-Zoll-Diskette (720 kB) und einem Festplattenspeicher (Kapazität 10 MB, formatiert).

16-Farben-Bildschirm mit hoher Auflösung von 640 x 400 Punkten oder Monochrom-Bildschirm in speziell augenschonender Bernstein-Farbe. Ultramodernes Design, ergonomische Flachtastatur, Bildschirm verstellbar.

MS-DOS<sup>R</sup> 2.0-Betriebssystem für den Zugang zu einer umfangreichen Programm- und Anwendungsbibliothek. Zudem lässt sich der PRO-16 an das «Omninet»-Rechner-Netzwerk anschliessen. Die Sumicom Mikrocomputer kommen gleich als Erwachsene zur Welt!

## Mit offenen Karten

Wenn Sie alles über die Sumicom-Systeme, -Programme und unseren Dienst am Kunden wissen möchten...



...so senden Sie diesen Coupon an Sumicom (Europe) SA, 10, rue de Veyrot,  
1217 Meyrin/GE. Telefon (022) 82 77 00.

Schicken Sie mir bitte Ihre Dokumentation  
 Rufen Sie mich bitte an, ich habe Fragen.

Name/Vorname: \_\_\_\_\_  
Firma/Branche: \_\_\_\_\_ Tel.: \_\_\_\_\_  
Adresse: \_\_\_\_\_

**Sumicom**

MK 6

# Einführung in APL (1)

1962 erschien im Wiley-Verlag, New York, das Buch «A Programming Language» von Dr. Kenneth E. Iverson. Iverson, Dozent für angewandte Mathematik in Harvard, definierte darin eine neue mathematische Formulierungssprache - nach dem Titel des Werkes kurz APL genannt -, die er zur Formulierung von Algorithmen und zur Beschreibung von Computeroperationen im Unterricht entwickelt hatte. Im Gegensatz zu anderen Programmiersprachen, war der Anwendungsbereich von APL sehr viel grösser. Er reicht von der Bit-Manipulation über numerische Algorithmen bis in die reine Textverarbeitung. Am IBM-Laboratorium Yorktown hat Iverson in Zusammenarbeit mit A.D. Falkoff und L.M. Breed APL weiterentwickelt, so dass bis 1966 für das IBM-System /360 die erste APL-Implementation fertiggestellt werden konnte. Nachdem APL intern bei IBM grosse Verbreitung gefunden hatte, wurde es 1969 auch von der Aussenwelt entdeckt. Inzwischen existieren APL-Systeme für die verschiedensten Datenverarbeitungsanlagen und sogar schon für einige Mikrocomputer.

## 1. Was ist APL?

APL ist...

- eine moderne Notation für die Darstellung mathematischer Sachverhalte
- eine Programmiersprache zur Beschreibung von Algorithmen
- ein interaktives Time-Sharing-System

## Thomas Gutekunst

### 1.1. APL als Notation

Bei APL handelt es sich eigentlich um eine stark verfeinerte Schreibweise für mathematische Zusammenhänge. Gegenüber der klassischen Notation präsentiert sich APL als konsequent und völlig konsistent, d.h. ausnahmsfrei.

APL erlaubt den Umgang mit komplexeren Objekten (Vektoren, Matrizen, Folgen) in ebenso einfacher Weise wie mit einzelnen Zahlen. Die dadurch bewirkte Entlastung von Unwesentlichem gestattet einsichtigeres Arbeiten auf höherer Stufe der Mathematik.

### 1.2. APL als Programmiersprache

Es ist ein charakteristisches Merkmal von APL, dass sich der Übergang von der Notation zur Computercodierung fast unbemerkt vollzieht. So ist ein einzelner Ausdruck in Iverson-Notation eigentlich schon ein Programm.

Gegenüber der blossen Notation von Termen kommen lediglich hinzu: 1) die Möglichkeit zur Programmstrukturierung (Verzweigung, Schleife, Unterprogramm),

- 2) die Möglichkeit der Benennung eines Programmes mit einem Namen und
- 3) die Möglichkeit Programme zu edieren, d.h. zu erstellen, zu korrigieren, abzuspeichern etc.

### 1.3. APL als Time-Sharing-System

Dass APL eine Time-Sharing-Sprache ist, gilt heute - im Zeitalter der Mikrocomputer - nur noch bedingt. Ursprünglich wurde APL jedoch für Time-Sharing-Betrieb geschaffen. Time-Sharing-Betrieb findet man bei Mehrbenutzersystemen, z.B. Grossrechenanlagen, die über Telefon- oder feste Leitungen mit mehreren Terminals (Benutzerstationen) verbunden sind (Bild 1).

Beim Time-Sharing, was man etwa mit «Zeit verteilen» übersetzen könnte, hat jeder Benutzer des Systems den Eindruck, dass der Rechner nur ihm zur Verfügung stehe. Dies kommt daher, dass das System fortwährend allen Benutzern abwechslungsweise die Rechnerleistung in kleinen Portionen (time slices) anbietet. Wer also gerade seine nächsten Schritte überlegt und das Terminal nicht benutzt, belastet den Rechner nicht.

Das Time-Sharing im APL-System verläuft so, dass immer nur die aus Programmen und Daten bestehenden Arbeitsbereiche (work spaces) weniger Benutzer in den Hauptspeicher des Rechners geladen werden, während der Rest vorübergehend auf einem Magnetplattenspeicher zwischengespeichert ist (swapping area). Da das Heraus- und Hereinholen (swapping) der Benutzerworkspaces infolge der Überlappung und Pufferung sehr schnell vor sich geht, ohne den Rechner wesentlich zu belasten, kommen alle Benutzer

kurzfristig wieder an die Reihe, so dass eine scheinbar ständige Verbindung mit dem System besteht.

## 2. Grundlagen von APL

APL ist ein interpretatives System, d.h. jede eingegebene APL-Anweisung wird sofort in eine system-interne tabellarische Form umgewandelt und ausgeführt. Eine Ausnahme bilden die Programmzeilen der sogenannten definierten Funktionen. Solche Zeilen werden bei der Eingabe nicht ausgeführt, sondern lediglich zur späteren Verwendung abgespeichert. Bei Feststellung eines Fehlers gibt das APL-System eine Fehlermeldung aus.

### 2.1. Die APL-Tastatur

Bild 2 zeigt eine typische APL-Tastatur der älteren Generation. Auffallend ist die Vielzahl an Sonderzeichen. Es handelt sich hierbei grösstenteils um Funktionszeichen für die verschiedenen APL-Funktionen. Einige Funktionszeichen ergeben sich zwar erst durch Uebereinandertippen zweier Symbole, doch findet man bei neueren Tastaturen auch für diese Zeichen eigene Tasten. Bild 3 zeigt den APL-Zeichensatz.

### 2.2. Die Ein- und Ausgabe

APL kennt keine speziellen Ein- und Ausgabeanweisungen. Die Eingabedaten werden zusammen mit den Operatoren eingegeben. Alphanumerische Daten werden durch Hochkommata begrenzt.

Nach dem Drücken der Eingabetaste (Return) liefert das System auf der nächsten Zeile das Resultat.

Die Unterscheidung von Ein- und Ausgabezeilen ergibt sich dadurch, dass Ausgaben beim Zeilenanfang beginnend geschrieben werden, während Eingaben stets um sechs Positionen nach rechts eingerückt sind. Fortsetzungen von Ausgabezeilen beginnen ebenfalls um sechs Positionen eingerückt.

### 2.3. Einfache Daten

#### 2.3.1. Numerische Daten

Zur Darstellung von Zahlen werden folgende Zeichen verwendet:

0 1 2 3 4 5 6 7 8 9 . - E.

Die Ziffern 0 bis 9 und der Dezimalpunkt haben die übliche Bedeu-

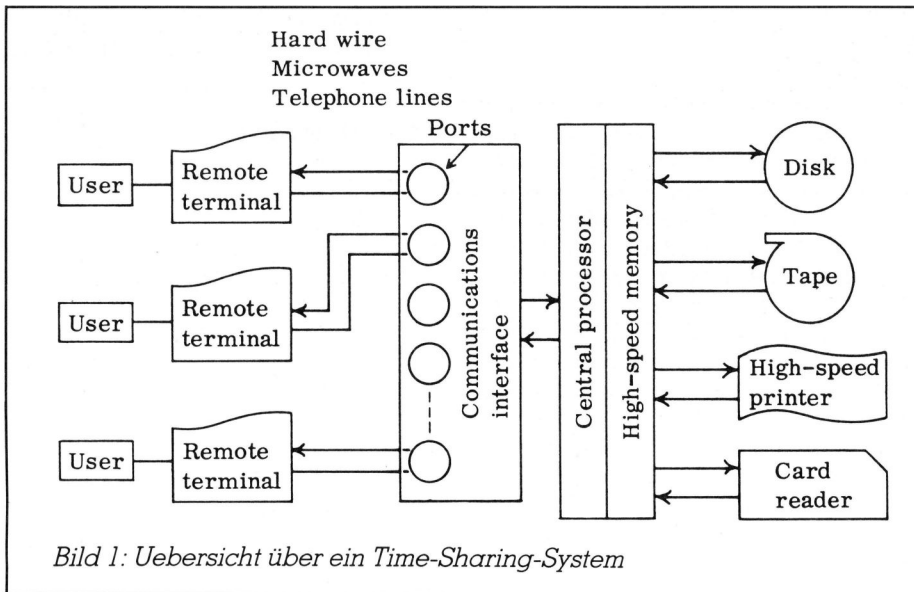


Bild 1: Uebersicht über ein Time-Sharing-System

Die operationellen Bedeutungen von (a) und (b) wurden in APL beibehalten, während die statische Bedeutung (c) durch ein hochgestelltes Minuszeichen ausgedrückt wird.

Hier kommt nun eine kleine APL-Besonderheit zum Vorschein. In der üblichen Notation hat das Minuszeichen drei Bedeutungen:

- (a) Subtraktion zweier Zahlen
- (b) Umklappung des Vorzeichens
- (c) Kennzeichnung einer negativen Zahl

Die operationellen Bedeutungen von (a) und (b) wurden in APL beibehalten, während die statische Bedeutung (c) durch ein hochgestelltes Minuszeichen ausgedrückt wird.

Mit dem Buchstaben *E* wird der Zehnerexponent in der Potenzdarstellung einer Zahl gekennzeichnet. Vor und hinter dem Buchstaben *E* darf kein Zwischenraum stehen. Diese Darstellungsart wird oft auch als halblogarithmische Darstellung bezeichnet.

Eine Zahl kann nach Belieben in der halblogarithmischen oder in der üblichen Form eingegeben werden; die Form der Ausgabewerte hängt von diesen Werten selbst ab.

### 2.3.2. Vektorkonstanten

Eine numerische Vektorkonstante ist ein Vektor, der aus mehreren Zahlen besteht, die durch eines oder mehrere Leerzeichen voneinander getrennt sind.

Eine Vektorkonstante wird als eine Grösse aufgefasst, sie braucht deshalb in einem Ausdruck auch nicht in Klammern gesetzt zu werden.

Eine einzelne Zahl wird nicht als Vektorkonstante mit einem Element, sondern als Skalar aufgefasst!

Beispiel:  $6.4^{-7} 246 981E^{-2}$

### 2.3.3. Alphanumerische Daten

Alphanumerische Daten werden als eine Folge von Zeichen, eingeschlossen in Hochkommata, eingegeben. Bei der Ausgabe werden die Hochkommata weggelassen.

Beispiel: 'VIERWALDSTAETTERSEE'  
'ABC~Δ↓123⊞'

Wenn ein Hochkommata innerhalb einer Zeichengruppe auftritt, muss es doppelt eingegeben werden:

'O'BRIEN'S STRING'  
O'BRIEN'S STRING

## 2.4. Skalare Funktionen

### 2.4.1. Monadische und dyadische Funktionen

Mathematische Funktionen können danach unterschieden werden, wieviele Argumente zur Berechnung benötigt werden. So benötigen die vier Grundrechenarten jeweils zwei Argumente, während das Wurzelziehen oder die trigonometrischen Funktionen nur eines haben.

Funktionen mit zwei Argumenten nennt man dyadisch, Funktionen mit einem Argument monadisch. Bei dyadischen Funktionen werden in APL die Argumente links und rechts von den Funktionssymbolen geschrieben, während monadische Funktionen ihr Argument immer rechts haben.

In der Programmiersprache APL kann ein und dasselbe Zeichen als Operator völlig verschiedene Auswirkungen haben, je nachdem, ob es monadisch oder dyadisch gebraucht wird.

Die skalaren Funktionen sind zunächst für skalare Argumente definiert und liefern in diesem Fall als Ergebnis wiederum einen Skalar. Die vollständigen Definitionen der Skalaren Funktionen finden Sie in Tabelle 1. Die Beispiele sollen die Definitionen etwas veranschaulichen.

Die Skalaren Funktionen lassen sich in folgende Gruppen unterteilen:

#### 2.4.1.1. Arithmetische Operationen

a) Die vier Grundrechenarten

Diese Operationen sind aus dem täglichen Gebrauch allgemein bekannt. APL verwendet als Operatoren die bei Taschenrechnern üblichen Symbole:  $+ - \times \div$ .

Monadische Verwendung dieser Operatoren bewirkt:

- $+$  : nichts (Vorzeichen beibehalten)
- $-$  : Negation (Vorzeichen umkehren)
- $\times$  : Signum (Vorzeichen testen)
- $\div$  : Reziprokwert (Kehrwert bilden)

b) Potenzieren, Wurzelziehen, Logarithmieren

In konventioneller mathematischer Notation wird eine Exponentiation ausgedrückt, indem der Exponent in kleinen Ziffern rechts über die Basis geschrieben wird. «2 hoch 3» wird beispielsweise als  $2^3$  geschrieben.

Eine solche Schreibweise ist für einen Computer alles andere als praktikabel, und eigentlich ist es auch ein wenig verwunderlich, dass die Exponentiation kein eigenes Symbol hat, wie es zum Beispiel Addition und Multiplikation haben. Deshalb verwendet APL ein spezielles Symbol für die Exponentiation, dass zwischen Basis und Exponent gesetzt wird. Das Zeichen ist  $*$  (Sternchen) und befindet sich auf Tastatur über dem Buchstaben P (P für engl. power).  $2^3$  schreibt sich also in APL:  $2*3$ .

Für das Wurzelziehen sieht APL kein spezielles Symbol vor. Das Wurzelziehen lässt sich jedoch durch Potenzieren des Radikanden mit dem Kehrwert des Wurzelexponenten ausdrücken:

$$\sqrt[x]{Y} = Y^{\frac{1}{x}}$$

In APL schreiben wir nun, um die X-te Wurzel aus Y zu ziehen:  $Y^* \div X$ .

Aus ähnlichen Gründen wie bei der Exponentiation verwendet APL für das Logarithmieren ebenfalls ein spezielles Symbol:  $\circledast$  (Kreis mit Sternchen). Der Logarithmus von X



Bild 2: Tastatur des System APL\360

zur Basis  $Y$  schreibt sich in APL wie folgt:  $Y \otimes X$ .

Bei monadischer Verwendung der Operatoren für Potenzieren und Logarithmieren wird als Basis die Euler'sche Zahl  $e$  angenommen.

### c) Minimum und Maximum

Oft ist es nützlich, die grössere von zwei Zahlen auswählen zu können. APL stellt hierfür eine Funktion zur Verfügung:  $Y \uparrow X$  ergibt die grössere der beiden Zahlen  $Y$  und  $X$ .

Ein Beispiel soll eine Nutzenanwendung dieser Funktion zeigen: Nehmen wir an, Sie arbeiten in einem Dienstleistungsbetrieb und müssen am Ende jedes Monats für jeden Kunden berechnen, in welchem Wert er Leistungen in Anspruch genommen und wieviel er dafür bereits bezahlt hat; wir bezeichnen diese beiden Beiträge mit *LEISTUNGEN* bzw. *GELDBETRAG*. Anhand dieser Angaben müssen Sie dem Kunden dann eine Rechnung stellen; Sie berechnen die Differenz *LEISTUNGEN - GELDBETRAG*.

Für noch offene Beträge verrechnen Sie Spesen von 1% pro Monat:  $.01 \times (\text{LEISTUNGEN} - \text{GELDBETRAG})$ , oder da in APL Ausdrücke - wie wir noch sehen werden - ohne Verwendung irgendwelcher Prioritätsregeln von rechts nach links ausgewertet werden, können wir die Klammern auch weglassen:  $.01 \times \text{LEISTUNGEN} - \text{GELDBETRAG}$ .

Da aber einige Kunden bereits mehr bezahlt haben könnten, als sie Leistungen in Anspruch genommen haben, ergäbe sich bei diesen Kunden ein negativer Spesenbetrag. Wenn Sie nun die Spesen in eben gezeigter Weise berechnen, würde das bedeuten, dass Sie dem Kunden einen Zins für vorausbezahlte Beträge zahlen würden. Ihr Betrieb zieht es aber vor, die Spesen als 1% entweder vom geschuldeten Betrag oder von Null zu berechnen,

je nachdem welches grösser ist! In APL drückt man dies wie folgt aus:

$.01 \times 0 \uparrow \text{LEISTUNGEN} - \text{GELDBETRAG}$ .

In gleicher Weise gibt es eine APL-Funktion, welche die kleinere von zwei Zahlen auswählt:  $Y \downarrow X$ .

### d) Auf- und Abrunden, Absolutwert und Rest bilden

Sind von einer Zahl die Nachkommastellen für eine weitere Verwendung der Zahl störend oder unnötig, so rundet man diese Zahl auf eine ganze Zahl, entweder auf die nächstkleinere oder auf die nächstgrössere ganze Zahl.

Für beide Arten ist in APL ein Funktionszeichen vorhanden: es sind die gleichen wie bei Minimum und Maximum, nur werden sie hier monadisch gebraucht:

$\uparrow X$ : aufrunden

$\downarrow X$ : abrunden

Um den Absolutbetrag einer Zahl zu bilden, verwendet man monadisch den senkrechten Balken:  $|X$ . Mit dem gleichen Symbol wird, wenn man es dyadisch verwendet, die Modulo-Funktion ausgedrückt:  $Y | X$  ergibt den Rest von  $X$  bei der Division durch  $Y$ . Wie in APL die Modulo-Funktion für negative Argumente definiert ist, geht aus Tabelle 1 hervor!

### e) Fakultät, Gamma-Funktion und Binomialkoeffizienten

In konventioneller mathematischer Notation wird die Fakultät von  $X$  geschrieben als  $X!$ . Dies widerspricht jedoch den allgemeinen Syntax-Regeln von APL, deshalb heisst es in APL:  $!X$ .

$!X$  ist auch für gebrochene Zahlen definiert. Zur Berechnung wird dann die Gamma-Funktion herangezogen:  $(!X) = \Gamma(X+1)$ .

Wenn das Symbol  $!$  dyadisch gebraucht wird, so dient es zur Berechnung von Binomialkoeffizienten.  $Y ! X$  ergibt die Anzahl der möglichen

Kombinationen, wenn aus  $X$  verschiedenen Objekten  $Y$  herausgenommen werden.

### f) Trigonometrische Funktionen

In APL werden die trigonometrischen Funktionen (oder Kreisfunktionen) mit dem Kreissymbol ausgedrückt. Um die nötige Vielfalt an Funktionen mit einem Symbol zu erreichen, wird wie folgt ein Linksargument verwendet, das die auf das Rechtsargument anzuwendende Funktion angibt:

$\sin X$ : $1 \circ X$	$\arcsin X$ : $-1 \circ X$
$\cos X$ : $2 \circ X$	$\arccos X$ : $-2 \circ X$
$\tan X$ : $3 \circ X$	$\arctan X$ : $-3 \circ X$

All diese Funktionen sind für Winkel im Bogenmass definiert, also nicht für das Gradmass!

Das Linksargument 0 ergibt die Beziehung zwischen Sinus- und Cosinusfunktion eines Winkels:  $0 \circ X$  entspricht  $(1 - X^2)^{.5}$ . Es gibt noch weitere Kreisfunktionen mit entsprechenden Linksargumenten; sie sind aus Tabelle 1 zu entnehmen.

Das Kreissymbol kann auch ohne Linksargument verwendet werden (monadisch); das Rechtsargument wird dann mit der Kreiszahl  $\pi$  multipliziert.

#### 2.4.1.2. Vergleichsoperationen

Die Vergleichsoperatoren  $<$ ,  $\leq$ ,  $=$ ,  $>$ ,  $\geq$ ,  $\neq$  vergleichen Links- und Rechtsargument in Bezug auf die angegebene Relation. Ist die Aussage richtig, so ist das Ergebnis 1, andernfalls 0.

#### 2.4.1.3. Logische Operatoren

Die logischen Operatoren «und» ( $\wedge$ ), «oder» ( $\vee$ ), «nicht» ( $\sim$ ) etc. sind nur auf die Argumente 0 («falsch») und 1 («wahr») anwendbar:

# LEHRGÄNGE

$Y \wedge X$  ergibt «wahr», wenn sowohl  $Y$  als auch  $X$  «wahr» sind.  
 $Y \vee X$  ergibt «wahr», wenn zumindest  $Y$  oder  $X$  «wahr» ist.  
 $\sim X$  ergibt «wahr» genau dann, wenn  $X$  «falsch» ist.  
 $Y \wedge \sim X$  ergibt «wahr», wenn zumindest  $Y$  oder  $X$  «falsch» ist.  
 $Y \nabla X$  ergibt «wahr», wenn sowohl  $Y$  als auch  $X$  «falsch» sind.

```
(01)          7×4
(02) 28
(03)          1 2 3 4×5 6 7 8
(04) 5 12 21 32
(05)          6 8 10 12-9
(06) -3 -1 1 3
(07)          60÷2 3 4 5
(08) 30 20 15 12
(09)          9 8 7 6+1 2 3
(10) LENGTH ERROR
(11)          9 8 7 6+1 2 3
(12)          ^
(13)          A+5
(14)          A+'HALLO'
(15)          A+51 7.3 -15.2 0.03 -1
(16)          A×-20
(17) -1020 -146 304 -0.6 20
(18)          A
(19) 51 7.3 -15.2 0.03 -1
(20)          B
(21) VALUE ERROR
(22)          B
(23)          ^
(24)          63÷3+7
(25) 6.3
(26)          1+2-3×4÷5
(27) 0.6
(28)          (((1+2)-3)×4)÷5
(29) 0
```

Listing 1: Grundregeln in APL

```
(01)          2 3ρ2 3 5 7 11 13
(02) 2 3 5
(03) 7 11 13
(04)          3 4ρ0
(05) 0 0 0 0
(06) 0 0 0 0
(07) 0 0 0 0
(08)          5ρ1 0
(09) 1 0 1 0 1
(10)          5 7ρ'ALPHA BETA GAMMA DELTA EPSILON'
(11) ALPHA
(12) BETA
(13) GAMMA
(14) DELTA
(15) EPSILON
(16)          D+2 2ρ1 2 3 4
(17)          E+2 2ρ8 -1 13 5
(18)          D×E
(19)          8 -2
(20) 39 20
(21)          E-7
(22) 1 -8
(23) 6 -2
```

Listing 2: Arbeiten mit Matrizen

```
(01)          16
(02) 1 2 3 4 5 6
(03)          10
(04)
(05)          □IO←0
(06)          16
(07) 0 1 2 3 4 5
(08)          □IO←1
(09)          V←'ABCDEFGHJKLMNOPQRSTUVWXYZ'
(10)          S←'BASEL'
(11)          V1S
(12) 2 1 19 5 12
(13)          'ABCDEF'1'DEFA'
(14) 4 6 7 1
(15)          L←2 3ρ'TIPTOP'
(16)          L
(17)          TIP
(18)          TOP
(19)          'TAP'1L
(20) 1 4 3
(21) 1 4 3
(22)          ,L
(23)          TIPTOP
(24)          H←,9
(25)          ρρH
(26) 1
(27)          ,3 3ρ13
(28) 1 2 3 1 2 3 1 2 3
(29)          (13),5
(30) 1 2 3 5
(31)          0,(16),(13)*2
(32) 0 1 2 3 4 5 6 1 4 9
(33)          'HUNDE','KUCHEN'
(34)          HUNDEKUCHEN
(35)          ρV
(36) 26
(37)          ρρV
(38) 1
(39)          ρL
(40) 2 3
(41)          ρρL
(42) 2
(43)          5 6ρS
(44)          BASELB
(45)          ASELBA
(46)          SELBAS
(47)          ELBASE
(48)          LBASEL
```

Listing 4: Die Operatoren  $\iota$ ,  $\rho$ ?

```
(01)          V←11 22 33 44 55 66
(02)          V[4]
(03) 44
(04)          V[1 4 3 1 2]
(05) 11 44 33 11 22
(06)          V[6 2 3]+60 20 30
(07)          V
(08) 11 20 30 44 55 60
(09)          V[9]
(10) INDEX ERROR
(11)          V[9]
(12)          ^
(13)          M←2 3ρ1 2 3 4 5 6
(14)          M
(15) 1 2 3
(16) 4 5 6
(17)          M[1;3]
(18) 3
(19)          M[2;1 3]
(20) 4 6
(21)          M[;2]
(22) 2 5
(23)          M[;1 3]
(24) 1 3
(25) 4 6
(26)          M[;1 3]+M[;3 1]
(27)          M
(28) 3 2 1
(29) 6 5 4
```

Listing 3: Indizierung in APL



Im normalen Sprachgebrauch versteht man unter «oder» meist «entweder das eine oder das andere, aber nicht beides gleichzeitig». Genau genommen ist dies die Exklusiv-Oder-Funktion. APL hat hierfür kein spezielles Symbol, da ja der «Ungleich»-Operator für die Argumente 0 und 1 genau die gewünschte Funktion hat:

$Y \neq X$  ergibt «wahr», wenn entweder  $Y$  oder  $X$  «wahr» ist.

#### 2.4.1.4. Zufallsoperatoren

Mit dem Operator  $?$  kann man Zufallszahlen erzeugen. Er hat zwei Formen, die eine ist monadisch, die andere dyadisch. Der monadische Operator  $?$  erzeugt voneinander unabhängige ganze Zufallszahlen, wie dies zum Beispiel beim Würfeln der Fall ist. Die Simulation eines Wurfes mit drei Würfeln wird in APL wie folgt ausgedrückt:  $?6\ 6\ 6$ . Dabei kann es durchaus vorkommen, dass nicht alle der drei gelieferten Zufallszahlen voneinander verschieden sind!

Der dyadische Operator  $?$  erzeugt eine Folge von Zahlen, die zufällig aus einer Folge aufeinanderfolgender Zahlen ausgewählt werden. Beispiel: Ziehen von Lottozahlen inkl. Zusatzzahl (hier kann und darf keine Zahl zweimal vorkommen!) würde ausgedrückt als:  $?7?49$ .

Eigentlich ist der dyadische Operator  $?$  keine skalare Funktion, da die Verknüpfung einen Vektor ergibt! Er sei hier im Zusammenhang mit Zufallszahlen von der Vollständigkeit halber erwähnt, um den Unterschied zum monadischen Operator  $?$  aufzuzeigen.

#### 2.4.2. Grundregeln in APL

Zunächst sollen einige Grundregeln von APL unter Bezug auf Listing 1 vorgestellt und demonstriert werden.

##### 2.4.2.1. Verknüpfungen

Die erste Rechnung wird in Zeile (1) angezeigt; das kann auch jeder Taschenrechner. Doch schon die nächste Rechnung (3) übersteigt die Möglichkeiten des Taschenrechners. Mit einem einzigen Multiplikationszeichen werden hier vier Multiplikationen ausgeführt. Auf beiden Seiten des Operationszeichens stehen hier Vektoren, die aus je vier Elementen bestehen; das Ergebnis ist ebenfalls ein Vektor der Länge 4. Die Multiplikation geschieht so, dass das erste

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	$\Delta$
a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	$\underline{\Delta}$
0	1	2	3	4	5	6	7	8	9																	

..	dieresis	$\alpha$	alpha	$\nabla$	nor
-	overbar	$\Uparrow$	upstile	$\nabla$	nand
<	less	$\Downarrow$	downstile	$\Delta$	del stile
$\leq$	not greater	$\bar{\_}$	underbar	$\Delta$	delta stile
=	equal	$\nabla$	del	$\phi$	circle stile
$\geq$	not less	$\Delta$	delta	$\phi$	circle slope
>	greater	$\circ$	null	$\ominus$	circle bar
$\neq$	not equal	$\uparrow$	quote	$\otimes$	log
$\vee$	or	$\square$	quad	$\text{I}$	l-beam
$\wedge$	and	$($	left paren	$\nabla$	del tilde
-	bar	$)$	right paren	$\emptyset$	base null
$\div$	divide	$[$	left bracket	$\nabla$	top null
+	plus	$]$	right bracket	$\nabla$	slope bar
$\times$	times	$\subset$	left shoe	$\nabla$	slash bar
$?$	query	$\supset$	right shoe	$\cap$	cap null
$\omega$	omega	$\cap$	cap	$\square$	quote quad
$\epsilon$	epsilon	$\cup$	cup	$\cdot$	quote dot
$\rho$	rho	$\perp$	base	$\boxplus$	domino
$\sim$	tilde	$\top$	top	$ $	stile
$\uparrow$	up arrow	$;$	semicolon	$*$	star
$\downarrow$	down arrow	$:$	colon	$\iota$	iota
$\rightarrow$	right arrow	$,$	comma	$\backslash$	slope
$\leftarrow$	left arrow	$\cdot$	dot	$/$	slash
$\circ$	circle	$\text{space}$	space	$\underline{\Delta}$	delta underbar

Bild 3: APL-Zeichensatz beim IBM-PC

Element des linken Vektors mit dem ersten Element des rechten Vektors multipliziert wird. Das Produkt hiervon ergibt dann das erste Element des Ergebnisvektors. In gleicher Weise werden die zweiten und die folgenden Elemente miteinander verknüpft. So werden in APL mit einem Operationszeichen oft viele Rechnungen durchgeführt. Die erste Grundregel lautet:

**Jedes Element der linken Seite wird mit dem entsprechenden Element der rechten Seite verknüpft.**

Diese Grundregel gilt ohne Ausnahme auch für Argumente, die komplizierter als Vektoren strukturiert sind. Wir werden bald Matrizen als nächsthöhere Struktur kennenlernen!

Soll ein Vektor elementweise nur mit einer Zahl (Skalar) verknüpft werden, so ist dies wie in (5) und (7) möglich. Ist dagegen die Länge der beiden Vektoren wie in (9) verschieden, so ist bei elementaren Operationen keine sinnvolle Verknüpfung möglich; es erfolgt eine Fehlermeldung des APL-Systems (10-12): Neben dem eigentlichen Fehlertext

bringt der Computer die Eingabe nochmal und kennzeichnet diejenige Stelle, an welcher der Fehler auftrat.

Die zweite Grundregel lautet also:

**Strukturen (Vektoren, Matrizen etc.) müssen bei elementaren Verknüpfungen entweder zusammenpassen oder ein-elementig sein.**

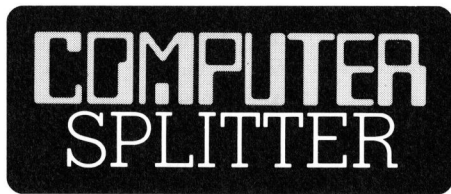
##### 2.4.2.2. Variablen

In APL wird eine Variable dadurch definiert, dass man eine Zuweisung vornimmt (13). Das Zeichen für die Zuweisung ist der Pfeil nach links ( $\leftarrow$ ). Im Gegensatz zu vielen anderen Programmiersprachen ist die Variablendefinition dynamisch, d.h. man kann nicht nur den Inhalt, sondern auch die Struktur jederzeit neu definieren (14,15).

Variablenamen bestehen aus einem einzelnen Buchstaben ( $A$  bis  $Z$  oder  $\underline{A}$  bis  $\underline{Z}$ ; bei manchen Systemen anstelle  $\underline{A}$  bis  $\underline{Z}$  auch  $a$  bis  $z$ ), dem Zeichen  $\Delta$  oder einer Kombination von Buchstaben, Ziffern und  $\Delta$ . Zulässig sind alle Kombinationen, die nicht mit einer Ziffer oder mit  $S\Delta$  oder  $T\Delta$  beginnen. Zwischenräume in Namen sind nicht erlaubt.

Namen dürfen im allgemeinen nicht beliebig lang sein. Ausserdem sind meist nicht alle Zeichen für die Identifikation einer Variablen relevant; bei der APL-Implementation des IBM-PC beispielsweise werden nur die ersten zwölf Zeichen beachtet.

Mit definierten Variablen kann wie mit Konstanten gerechnet werden (16), zur Abfrage einer Variablen



## TI ohne Fortune

(272/tp) Wir lesen es, und dann in Bezweiflung unseres Auffassungsvermögens nochmals, sicherheitshalber nochmals, um dann definitiv das Handtuch zu werfen. Da meldet Texas Instruments dem US-Verteidigungsministerium am 4. September 1984, dass gewisse ihrer «militärischen Halbleiterprodukte möglicherweise nicht ganz in Übereinstimmung mit den Kundenspezifikationen getestet worden» seien. Am 10. September informiert das Ministerium die Presse und am 11. lädt TI selbst zu einer Pressekonferenz. Schon am 12. September werden auch die Schweizer Medien informiert. Dies halten wir einerseits für seriöse Informationspolitik, angesichts der Brisanz des Themas aber auch für eine dringende Notwendigkeit. Nun, so prompt wie die Mitteilung an die hiesige Öffentlichkeit gelangt, so lückenhaft ist sie. Erstens: Da wird wörtlich festgestellt: «Das Problem ist vollständiges Testen, nicht fehlerhafte Chips.» Interessieren die Öffentlichkeit etwa die Testmethoden einer Firma? Würde mit einwandfrei funktionierenden Chips einfach so zum Spass ein so am Image nagender Wirbel veranstaltet? Zweitens: die NASA habe die Auswirkungen dieses Testfehlers auf den verschobenen Discovery-Start untersucht. Wir zitieren am besten wieder das Communiqué: «IBM hat selber gesagt, dass sie darin einen Zufallsfehler sehen, der in keinem Zusammenhang mit dem TI-Testproblem stand.» Alle Klarheiten beseitigt? Als Normalverbraucher interessiert mich noch folgendes: Wo finden die «fehlerhaft getesteten» Mikrochips im militärischen Bereich Anwendung? □

gibt man einfach den Namen ein (18). Bei Verwendung nicht definierter Variablen erfolgt eine Fehlermeldung (21). Wir merken uns:

**Variablen werden durch Zuweisung neu- bzw. undefiniert. Nicht zugeordnete Ergebnisse werden ausgegeben.**

### 2.4.2.3. Die Rechts-Links-Regel

In APL werden alle Ausdrücke gleichartig verarbeitet: Wenn man von links nach rechts liest, so hat jedes Funktions- und Zuweisungszeichen, den gesamten rechts davon stehenden Ausdruck (bzw. wenn das Zeichen in Klammern steht, den Ausdruck bis zur schliessenden Klammer) als Argument (24,26).

Demnach wird zuerst der Ausdruck am rechten Ende der Zeile ermittelt, dann der davorstehende Ausdruck usw. Mit anderen Worten:

**Die Ausführung erfolgt von rechts nach links.**

Durch Klammersetzung (28) kann eine andere als die durch die Schreibweise eines Ausdrucks vorgegebene Reihenfolge der Verarbeitung erzwungen werden. Die Reihenfolge richtet sich dann nach den Klammern, die stets paarweise auftreten müssen.

Das Fehlen von Prioritätsregeln (z.B. «Punkt vor Strich») mag anfänglich etwas seltsam anmuten. Doch nach dem Kennenlernen weiterer APL-Funktionen werden wir den riesigen Vorteil der Rechts-Links-Regel, bei der alle Rechenoperationen gleich behandelt werden, erst richtig einschätzen können und das anfängliche Umdenkenmüssen nicht mehr bedauern.

Im APL Primer fand ich eine bemerkenswerte Illustration zur Rechts-Links-Regel, die ich dem Leser nicht vorenthalten möchte:

In Bezug auf die Reihenfolge entspricht die Struktur eines APL-Ausdruckes derjenigen eines gewöhnlichen Satzes der englischen Sprache. Mit deutschen Sätzen klappt's nicht immer, doch der folgende Satz ist hierfür geeignet: «Frau Meier kauft eine Büchse Hundefutter für den Hund ihrer Tochter».

Um die Bedeutung dieses Satzes zu finden, stellen wir nun einige Fragen:

Was kauft Frau Meier?  
**eine Büchse** Hundefutter für den Hund ihrer Tochter

Was ist in der Büchse?

**Hundefutter** für den Hund ihrer Tochter

Für wen ist das Hundefutter?

**für den Hund** ihrer Tochter

Wem gehört der Hund?

**ihrer Tochter**

In diesem Satz bezieht sich jeder Satzteil auf alles, was hintendran steht. Dieselbe Struktur finden wir in APL:

$A+BxC-D\div E$

Was wird zu  $A$  addiert?  $BxC-D\div E$

Womit wird  $B$  multipliziert?  $C-D\div E$

Was wird von  $C$  subtrahiert?  $D\div E$

Wodurch wird  $D$  dividiert?  $E$

Nehmen wir an, Sie müssten obigen Satz jemandem erklären, der zwar die Grammatik beherrscht, aber nicht weiss, worauf sich die einzelnen Satzteile beziehen. Sie werden feststellen, dass die Antwort auf jede der Fragen noch alles umfasst, was später im Satz kommt. Werden die Fragen in der Reihenfolge gestellt, wie wir es vorher taten, also von links nach rechts, so ist die Antwort auf die erste Frage nicht sofort brauchbar, da sich die Antwort noch auf den bisher noch nicht bestimmten Rest des Satzes bezieht. Um die Bedeutung des Satzes logisch aufzubauen, müssen Sie mit dem letzten Satzteil beginnen: «ihrer Tochter». Sie schreiten nun nach links weiter zu «für den Hund» und erhalten die Bedeutung «für den Hund ihrer Tochter». So fahren Sie weiter, immer weiter nach links, bis die Bedeutung des ganzen Satzes gefunden ist.

Aehnlich wird ein Ausdruck wie  $A+BxC-D\div E$  ausgewertet: Der Computer beginnt damit, den Wert der Variablen  $E$  herauszusuchen. Dann sucht er den Wert von  $D$ ; damit sind alle nötigen Voraussetzungen geschaffen, um « $D$  geteilt durch  $E$ » zu berechnen. Als nächstes sucht der Computer den Wert von  $C$ , nun kann die Differenz zwischen  $C$  und « $D$  geteilt durch  $E$ » berechnet werden.

## 2.5. Strukturdaten

Oft ist es sinnvoll, viele Daten, die gleichartig verarbeitet werden sollen, als eine Einheit zu betrachten. In APL nennt man diese Gebilde Strukturdaten.

### 2.5.1. Rang und Dimension

Als Strukturdaten kennen wir bereits Vektoren. Es handelt sich hierbei gewissermassen um eine eindimensionale Struktur, da die Elementen

te eines Vektors linear angeordnet sind.

Die nächsthöheren Strukturdaten, bei denen die Elemente in einem Rechteck (zweidimensional) angeordnet sind, bezeichnen wir als Matrizen. Für drei- und höherdimensionale Strukturen gibt es keine speziellen Bezeichnungen; sie werden allgemein als Tensoren bezeichnet.

Der Rang einer Strukturgröße gibt an, wieviel-dimensional diese ist. Vektoren haben also den Rang 1, Matrizen den Rang 2 etc. Skalare haben den Rang 0.

Obwohl die meisten Strukturdaten aus mehr als einem Element bestehen, gibt es auch welche mit nur einem und solche ohne Element. Eine Struktur ohne Element wird als leere Struktur bezeichnet. Diese wird bei der Ausgabe als eine Leerzeile dargestellt.

Der Dimensionsvektor gibt die Anzahl Elemente in jeder Richtung der Struktur an. Bei Skalaren ist der Dimensionsvektor ein Leervektor. Allgemein entspricht die Anzahl Elemente im Dimensionsvektor dem Rang der Struktur.

Mit dem Operator  $\rho$  (rho) kann bei monadischer Verwendung die Struk-

tur von Strukturdaten ermittelt werden. Das Ergebnis ist der Dimensionsvektor. Für einen Vektor mit fünf Elementen ergibt sich zum Beispiel der Vektor 5, während der Vektor 2 3 für eine Matrize bestehend aus 2 Zeilen und 3 Spalten steht. Lautet der Dimensionsvektor 3 5 8, so ist die zugehörige Strukturgröße ein dreidimensionaler Tensor, dessen Grundfläche aus einer Matrix von 5 Zeilen und 8 Spalten gebildet wird, über der in zwei weiteren Ebenen gleichdimensionierte Matrizen liegen.

Bei zweifacher monadischer Verwendung von  $\rho$  wird die Dimension des Dimensionsvektors gezeigt, die ja gleich dem Rang der Struktur ist.

### 2.5.2. Generierung von Matrizen und Tensoren

Um Strukturen zu bilden verwendet man ebenfalls den Operator  $\rho$ , allerdings dyadisch. Auf der rechten Seite des Operators steht die zu strukturierende Größe, während links in Form einer Dimensionsvektors die Struktur angegeben wird.

Beispiele hierzu sind in Listing 2 (1-15) gegeben. Reichen die Elemente im Rechtsargument nicht aus, um

die neue Struktur zu füllen, so wird es noch so viele Male vervielfacht bis genügend Elemente vorhanden sind.

Analog wie unter 2.4.2.1. beschrieben, können zwei Matrizen (oder Tensoren) mit gleichen Dimensionen durch die skalaren Funktionen elementweise miteinander verknüpft werden (16-20). Natürlich kann jede Struktur auch mit einem Skalar verknüpft werden, wobei der Skalar dann mit jedem Element der Struktur verknüpft wird (21-23).

### 2.5.3. Indices

Um einzelne Elemente einer Struktur verarbeiten bzw. verändern zu können, müssen diese gezielt ansprechbar sein. Dies wird durch Indizierung ermöglicht; jedes Element der Struktur erhält eine Art Koordinaten, mit denen die Stellung innerhalb der Größe angegeben wird.

Die Art und Weise der Indizierung in APL soll anhand von Listing 3 gezeigt werden: Will man zum Beispiel das vierte Element eines Vektors  $V$  herausgreifen, so wird  $V$ , versehen mit der Zahl 4 in der eckigen Indexklammer, aufgerufen ( $V[4]$ ). Es können

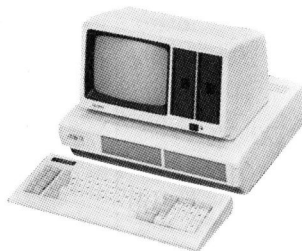
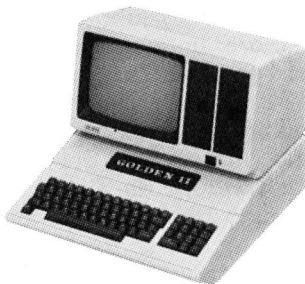
## Computer-<sup>\*</sup>Delicious-Versand

\* Delicious = Köstlich

Wir importieren das Beste aus Fernost

**Golden-64**

**Zeus 3005**



CPU 6502, 64k RAM  
8 Slots, Schaltnetzteil  
Zahlenblock, Funktionstasten  
Gross- + Kleinschrift  
Apple II voll kompatibel  
**Superpreis Fr. 1095.-**

Dual CPU (Z80+6502)  
2 Disk Drives 143kB  
64k RAM, 8 Slots  
Zahlenblock, Funktionstasten  
CP/M + Apple II kompatibel  
**Superpreis Fr. 2850.-**

- Disk Drive Mitac AD-3 Slimline Spitzendrive zu Apple II/III 525.-
- Gratis 10 Disketten dazu!
- Disk Controller 100.-
- Ergonomische Tastatur Apple II 300.-
- Grappler parallel + Graphic 180.-
- Microbuffer parallel 32k Buffer 280.-
- Super Serial Int. 50-19200 B 280.-
- Pkaso-Epson Graphic-Interface 260.-
- 16k RAM-Karte 150.-
- 128k RAM-Karte + Software 550.-
- 80-Zeich. Karte mit Softswitch 220.-

- Z-80 Karte CP/M 180.-
- 8088 Karte CP/M-86 695.-
- EPROM Programmierkarte 220.-
- AD/DA Karte 16 Kanal/12 Bit 595.-
- Wild-Karte 150.-

### Ihre Vorteile:

- \* 10 Tage Rückgaberecht
- \* 6 Monate Garantie
- \* Eigener Reparaturservice

Verlangen Sie Unterlagen oder bestellen Sie direkt bei:



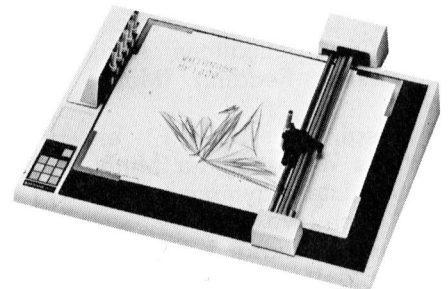
**Computer-Delicious-Versand**

Postfach 2769, 4002 Basel  
Neu Direkter Auskunftsdienst: Tel. 061/35 29 00



vormals WATANABE Instr.

ab Fr. 2053.-  
netto  
ohne Wust



## Intelligenter Mehrfarben-Plotter Modell MP 1000

- Papierformat DIN A3
- 6 Schreibstifte, Faser- oder Tusch-Schreibfedern
- Schnittstellen: IEEE-488, RS 232 C, 8-Bit parallel
- Betriebsarten: Zeichnen, Drucken, Test
- Funktionen: Charaktergenerator (ASCII), Charaktergröße, Schriftrichtung, ganze und unterbrochene Linien, Koordinaten absolut und relativ, Markierungen
- Schraffur, deutscher Schriftsatz und griechische Buchstaben als Option
- An alle bekannten Kleincomputer anschliessbar



**SEYFFER + CO. AG.**

**8048 ZÜRICH**

Abteilung Messtechnik Tel. 01/62 82 00

auch mehrere Elemente auf einmal herausgegriffen werden (4).

Soll dagegen  $V$  an bestimmten Stellen geändert werden, so geschieht die Zuweisung auf den entsprechend indizierten Vektor (6).

In der Indexklammer brauchen keine einfachen Indizes zu stehen, sondern es können auch beliebige APL-Ausdrücke sein, die gültige Indizes liefern. Falls in der Indexklammer Zahlen auftauchen, die keinen möglichen Index darstellen, antwortet APL mit einer Fehlermeldung (10-12).

## Literatur

- (1) Iverson, Kenneth E.; A Programming Language; John Wiley and Sons; New York 1962 (vergriffen)
- (2) Smillie, K.W.; STATPACK 2: An APL Statistical Package; Department of Computing Science; University of Alberta; Publ. No. 17; Alberta 1969
- (3) APL\360 Primer; IBM-Form. GH20-0689-1; 1970
- (4) APL\360 User's Manual; IBM-Form. GH20-0683-1; 1970
- (5) Stoop, K.; APL\360 Time Sharing; IBM Schweiz Zürich (Form. 3267-2); Zürich 1971
- (6) Iverson, Kenneth E.; Algebra: An Algorithmic Treatment; Addison-Wesley; Reading (Mass.) 1972
- (7) Grey, Louis D.; A Course in APL With Applications; Addison-Wesley; Reading (Mass.) 1973
- (8) Haegi, Hans R.; APL - ein geistiges Werkzeug für Mittelschulen; Literargymnasium Zürich; Zürich 1975
- (9) Pakin, Sandra & Lochner, Hans; APL-Handbuch; Science Research Associates GmbH; Stuttgart 1976
- (10) Specht, Joseph; APL-Praxis; B.G. Teubner; Stuttgart 1983
- (11) Zaks, Rodney; A Microprogrammed APL-Implementation; Sybex; Berkeley (Ca.) 1983
- (12) IBM PC APL Manual; IBM Madrid Scientific Center (Form. 1502219); 1983
- (13) Bensimon, Jacques; STSC APL\*PLUS and IBM PC APL; BYTE 3-84, pp. 246-264
- (14) Masterson, Fred. A.; Languages for Students; BYTE 6-84, pp. 233-238

Die Indizierung bei Matrizen und Tensoren verläuft prinzipiell gleich wie bei Vektoren, nur hat man es hier mit zwei Indices für Zeilen und Spalten (bei Tensoren mit drei oder mehr Indices) zu tun. Die entsprechenden Indexausdrücke müssen durch Semikolon voneinander abgetrennt werden. Einige Beispiele:

- (17) gibt das Element in der 1. Zeile und 3. Spalte.
- (19) gibt die Elemente der 2. Zeile, die in der 1. und 3. Spalte stehen.
- (21) gibt die ganze 2. Spalte.
- (23) gibt die 1. und 3. Spalte.
- (26) Die 3. und 1. Spalte werden herausgegriffen und auf die 1. und 3. Spalte abgespeichert, was einer Spaltenvertauschung gleichkommt!

## 2.6. Die Operatoren $\iota$ , $\rho$ ?

### 2.6.1. Indexvektor bilden ( $Z \leftarrow \iota X$ )

Das Ergebnis dieser monadischen Funktion ist ein Vektor, dessen  $X$  Komponenten die ersten  $X$  natürlichen Zahlen (in aufsteigender Reihenfolge) sind. Für das Argument 0 liefert die Funktion den Leervektor.

Normalerweise ist 1 der Indexanfang, mit  $\square IO \leftarrow 0$  ( $IO =$  index origin) kann man den Indexanfang jedoch zu Null machen. Näheres hierzu folgt in einem späteren Abschnitt. Bei Indexanfang Null ist 0 die erste Komponente von  $\iota X$ , und  $X-1$  die letzte Komponente. Beispiele hierzu sind in Listing 4 (1-8) gegeben.

### 2.6.2. Index zeigen ( $Z \leftarrow \iota X$ )

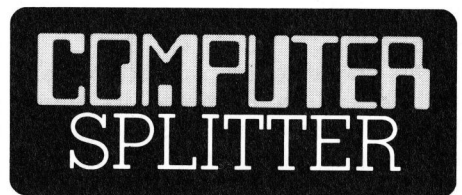
Bei dyadischer Verwendung von  $\iota$  (iota) ist  $Z[I]$  gleich dem Index von  $X[I]$  in  $Y$ , wobei  $Y$  ein Vektor sein muss. Beispiel: (9-12).

Wenn zu einem Element von  $X$  kein passendes Element von  $Y$  gefunden wird, wird das entsprechende Element des Ergebnisses gleich dem Index des letzten Elementes von  $Y$  plus 1 (13). Wenn  $Y$  ein leerer Vektor ist, wird jedes Element des Ergebnisses zu  $\iota 1$ . Weiteres Beispiel: (15-21).

### 2.6.3. Autreihen ( $Z \leftarrow ,X$ )

Das Ergebnis ist ein Vektor, dessen Komponenten die Elemente des Arguments sind. Ihre Reihenfolge ist gegeben durch die Indexfolge des Arguments.

Mit dieser Operation lassen sich beliebige Strukturen, aber auch Ska-



## Süsse und saure Äpfel

(273/tp) Ein Riesenerfolg auch in Europa scheint der Macintosh zu sein. Die erwarteten Verkaufszahlen des Deutschen Generalvertreters wurden gleich um 40 Prozent überboten. Es gibt zur Zeit Wartefristen. In den USA stürzen sich die Zubehör-Hersteller auf Mac: So werden in den USA massenweise Werkzeuge für die Bildverarbeitung vorgestellt. Es gibt schon Zubehör für grosse Netzwerke, und seit einigen Wochen auch Compiler für C, Logo und LISP. Sogar IBM stellt für den Junior ein Programm ColorPaint vor, das, wie böse Zungen behaupten, MacPaint nachempfunden ist. Weniger erfreulich geht der Ilc über den Ladentisch. Fachleute machen dafür die fehlende Software verantwortlich.  $\square$

lare zu Vektoren umformen. Beispiele: (22-28).

### 2.6.4. Verketteten ( $Z \leftarrow Y, X$ )

Mit dem Operator  $,$  (Komma) können dyadisch beliebig lange Vektoren und Skalare untereinander zu neuen Vektoren zusammengekettet werden (29-34).

Diese Funktion ist auch für Strukturdaten als Argumente definiert, doch soll uns vorläufig die Definition für Vektoren und Skalare genügen! Wir werden in einem späteren Abschnitt noch genauer auf diese Funktion eingehen.

### 2.6.5. Dimension zeigen ( $Z \leftarrow \rho X$ ) und Strukturieren ( $Z \leftarrow \gamma \rho X$ )

Sowohl die monadische als auch die dyadische Bedeutung des Operators  $\rho$  wurde in Abschnitt 2.5.1. bzw. 2.5.2. vorweggenommen, weshalb sie an dieser Stelle nicht nochmals erklärt werden. Einige Beispiele (35-48) sollen uns die beiden Funktionen jedoch wieder in Erinnerung rufen!

In der nächsten Folge werden wir die wichtigsten Befehle zur Bedienung des APL-Systems kennenlernen, und die ersten APL-Programme schreiben.  $\square$

Operator	monadisch		dyadisch	
	Definition	Beispiele	Definition	Beispiele
+	<b>Vorzeichen beibehalten</b> $Z \leftarrow +X$  $Z=X$	$+9$ $+6 \ 8.2 \ 17$ $-6 \ 8.2 \ 17$	<b>Addieren</b> $Z \leftarrow Y+X$  $Z=Y+X$	$7 \ 8 \ 9+1 \ 2 \ 3$ $8 \ 10 \ 12$ $2+3.4 \ -5.1 \ 10.2$ $5.4 \ -3.1 \ 12.2$
-	<b>Vorzeichen umkehren</b> $Z \leftarrow -X$  $Z=-X$	$-7$ $-5 \ 6 \ -3.9$ $-5 \ 6 \ -3.9$	<b>Subtrahieren</b> $Z \leftarrow Y-X$  $Z=Y-X$	$5 \ 7 \ 3-4 \ 8 \ -2$ $1 \ -1 \ 5$ $10-5.3 \ -3 \ 11$ $4.7 \ 13 \ -1$
×	<b>Vorzeichen testen (Signum)</b> $Z \leftarrow \times X$  $Z=+1$ , für $X>0$ $Z=0$ , für $X=0$ $Z=-1$ , für $X<0$	$\times 5$ $1$ $\times^{-3} \ 8 \ 0$ $-1 \ 1 \ 0$	<b>Multiplizieren</b> $Z \leftarrow Y \times X$  $Z=Y \times X$	$5 \ 3 \ 2 \times 7 \ 6 \ 1$ $35 \ 18 \ 2$ $3 \times 7.1 \ 12 \ -5$ $21.3 \ 36 \ -15$
÷	<b>Kehrwert bilden</b> $Z \leftarrow \div X$  $Z = \frac{1}{X}; X \neq 0$	$\div 4$ $0.25$ $\div 5 \ 2 \ .1111111111$ $0.2 \ -0.5 \ 9$	<b>Dividieren</b> $Z \leftarrow Y \div X$  $Z = \frac{Y}{X}; X \neq 0$ für $Y=X=0$ wird $Z=1$ , weil $\lim_{x \rightarrow 0} \frac{x}{x} = 1$	$6 \ 20 \ 4 \div 3 \ 6 \ 5$ $2 \ 3.333333333 \ 0.8$ $0 \div 5$ $0 \div 0$ $1$
*	<b>Potenzieren zur Basis e</b> $Z \leftarrow *X$  $Z=e^X$  e ist die Basis der natürlichen Logarithmen  $e \approx 2.7182818284590451$	$*1$ $2.718281828$ $*^{-3} \ 5$ $0.04978706837 \ 1.648721271$	<b>Potenzieren</b> $Z \leftarrow X^*Y$  $Z=Y^X$  $Y^0=1$  X beliebig, $X \geq 0$ , für $Y > 0$ für $Y = 0$ $X = \frac{m}{2n-1}$ mit $m, n \in \mathbb{N}$ , für $Y < 0$ für $Y=X=0$ wird $Z=1$ , weil $\lim_{x \rightarrow 0} x^x = 1$	$5 \ 3 \ 4^*0 \ 4 \ -2$ $1 \ 81 \ 0.0625$ $0^*6$ $0$ $-32 \ -125^*6 \ 2$ $-8 \ -2.626527804$ $-243^*-2$ $-0.3333333333$ $0^*0$ $1$
⊗	<b>Logarithmieren zur Basis e</b> $Z \leftarrow \otimes X$  $Z = \ln X; X > 0$	$\otimes 2.7182818284$ $1$ $\otimes .04978706837$ $-3$	<b>Logarithmieren</b> $Z \leftarrow Y \otimes X$  $Z = \log_Y X; X > 0, Y > 0$  Für $Y=1$ muss $X=1$ sein!	$5 \ 3 \ 4 \otimes 1 \ 81 \ .0625$ $0 \ 4 \ -2$ $10 \otimes 1 \ 10 \ 25 \ 100$ $0 \ 1 \ 1.397940009 \ 2$
⌈	<b>Aufrunden</b> $Z \leftarrow \lceil X$  Z ist die (algebraisch) kleinste ganze Zahl, die grösser oder gleich X ist.	$\lceil 5.2$ $6$ $\lceil 5.7 \ 8.1 \ -3.3 \ -8.7$ $6 \ 9 \ -3 \ -8$ $\lceil 7 \ -7$ $7 \ -7$	<b>Maximieren</b> $Z \leftarrow Y \lceil X$  $Z=Y$ , für $Y > X$  $Z=X$ , für $Y \leq X$	$6 \lceil 78$ $78$ $4 \lceil 3 \ 5 \ -7 \ 8$ $4 \ 5 \ 4 \ 8$ $2 \ 6.5 \ -5 \lceil 4 \ 1 \ -9$ $4 \ 6.5 \ -5$
⌊	<b>Abrunden</b> $Z \leftarrow \lfloor X$  Z ist die (algebraisch) grösste ganze Zahl, die kleiner oder gleich X ist.	$\lfloor 5.2$ $5$ $\lfloor 5.7 \ 8.1 \ -3.3 \ -8.7$ $5 \ 8 \ 4 \ -9$ $\lfloor 7 \ -7$ $7 \ -7$	<b>Minimieren</b> $Z \leftarrow Y \lfloor X$  $Z=Y$ , für $Y \leq X$  $Z=X$ , für $Y > X$	$6 \lfloor 78$ $6$ $4 \lfloor 3 \ 5 \ -7 \ 8$ $3 \ 4 \ -7 \ 4$ $2 \ 6.5 \ -5 \lfloor 4 \ 1 \ -9$ $2 \ 1 \ -9$
	<b>Absolutbetrag bilden</b> $Z \leftarrow  X$  $Z= X $	$ ^{-5.73}$ $5.73$ $ ^{-7} \ -1.2 \ 9 \ 8.7$ $7 \ 1.2 \ 9 \ 8.7$	<b>Rest bilden</b> $Z \leftarrow Y   X$  $X \equiv Z \pmod{Y}$  für $Y=0$ gilt: $Z=X$ für $Y \neq 0$ gilt: $Z=X-tY; t \in \mathbb{Z}$ $0 \leq  Z  <  Y  \wedge Y \cdot Z \geq 0$	$3 5$ $2$ $3 \ -3 \ -3 \ 0 ^{-5} \ 4 \ -4 \ -4$ $1 \ -2 \ -1 \ -4$ $2.5 7.24 \ 6$ $2.24 \ 1$ $ 3.25 \ 3 \ -6$ $0.25 \ 0 \ 0$

Tabelle 1: Skalare Funktionen

# LEHRGÄNGE

Operator	monadisch		dyadisch																																																	
	Definition	Beispiele	Definition	Beispiele																																																
!	<b>Fakultät bilden</b> $Z \leftarrow !X$ $Z = X!$ , für $X \in \mathbb{N}_0$ für andere Argumente $X$ gilt: $Z = \Gamma(x+1)$ (Gammafunktion), wobei $x$ keine negativ ganzzahligen Werte annehmen darf.	$!6$ 720 $!5 \quad !5 \quad 0$ 0.8862269254 1.772453851 1	<b>Binomialkoeffizienten bilden</b> $Z \leftarrow Y!X$ $Z = \binom{X}{Y}$ Allgemein gilt: $Z = \frac{X!}{Y!(X-Y)!}$	$4!8$ 70 $2.5 \quad 2.4!4 \quad 6.7$ 5.432488724 24.39985591 $0 \quad 1 \quad 2 \quad 3!3$ $1 \quad 3 \quad 3 \quad 1$ $!2.1 \quad !6!2 \quad 6$ 0.007370511731 0.1426763772																																																
○	<b>Mit <math>\pi</math> multiplizieren</b> $Z \leftarrow \circ X$ $Z = X \cdot \pi$ $\pi \approx 3.141592653589793$	$0!1$ 3.141592654 $0.25 \quad .5$ 0.7853981634 1.570769327	<b>Kreisfunktion bilden</b> $Z \leftarrow \circ X Y$ Y wählt die auf X anzuwendende Funktion aus: <table border="1" style="margin-top: 10px;"> <thead> <tr> <th>Y</th> <th>Funktion</th> <th>Definitionsbereich</th> </tr> </thead> <tbody> <tr> <td>-7</td> <td>artanh X</td> <td><math> X  \leq 1</math></td> </tr> <tr> <td>-6</td> <td>arcosh X</td> <td><math>X \geq 1</math></td> </tr> <tr> <td>-5</td> <td>arsinh X</td> <td></td> </tr> <tr> <td>-4</td> <td><math>\sqrt{X^2-1}</math></td> <td><math> X  \geq 1</math></td> </tr> <tr> <td>-3</td> <td>arctan X</td> <td></td> </tr> <tr> <td>-2</td> <td>arccos X</td> <td><math> X  \leq 1</math></td> </tr> <tr> <td>-1</td> <td>arcsin X</td> <td><math> X  \leq 1</math></td> </tr> <tr> <td>0</td> <td><math>\sqrt{1-X^2}</math></td> <td></td> </tr> <tr> <td>1</td> <td>sin X</td> <td></td> </tr> <tr> <td>2</td> <td>cos X</td> <td></td> </tr> <tr> <td>3</td> <td>tan X</td> <td></td> </tr> <tr> <td>4</td> <td><math>\sqrt{X^2+1}</math></td> <td></td> </tr> <tr> <td>5</td> <td>sinh X</td> <td></td> </tr> <tr> <td>6</td> <td>cosh X</td> <td></td> </tr> <tr> <td>7</td> <td>tanh X</td> <td></td> </tr> </tbody> </table> Winkel werden im Bogenmass dargestellt!	Y	Funktion	Definitionsbereich	-7	artanh X	$ X  \leq 1$	-6	arcosh X	$X \geq 1$	-5	arsinh X		-4	$\sqrt{X^2-1}$	$ X  \geq 1$	-3	arctan X		-2	arccos X	$ X  \leq 1$	-1	arcsin X	$ X  \leq 1$	0	$\sqrt{1-X^2}$		1	sin X		2	cos X		3	tan X		4	$\sqrt{X^2+1}$		5	sinh X		6	cosh X		7	tanh X		$1 \quad 1 \quad 2 \quad 3 \circ 0.5 \quad 2 \quad .25$ $1 \quad 1 \quad 1$ $4 \quad 0 \quad !4 \circ 5 \quad .25 \quad 1$ 5.099019514 0.9682458366 ○ $5 \quad 6 \quad 7 \circ 0$ $0 \quad 1 \quad 0$ $!1 \quad !2 \quad !3 \circ 1$ 1.570769327 0 0.7853981634 $!5 \quad !6 \quad !7 \circ 0 \quad 1 \quad 0$ $0 \quad 0 \quad 0$
Y	Funktion	Definitionsbereich																																																		
-7	artanh X	$ X  \leq 1$																																																		
-6	arcosh X	$X \geq 1$																																																		
-5	arsinh X																																																			
-4	$\sqrt{X^2-1}$	$ X  \geq 1$																																																		
-3	arctan X																																																			
-2	arccos X	$ X  \leq 1$																																																		
-1	arcsin X	$ X  \leq 1$																																																		
0	$\sqrt{1-X^2}$																																																			
1	sin X																																																			
2	cos X																																																			
3	tan X																																																			
4	$\sqrt{X^2+1}$																																																			
5	sinh X																																																			
6	cosh X																																																			
7	tanh X																																																			
<	--		<b>Prüfen auf 'kleiner'</b> $Z \leftarrow Y < X$ $Z=1$ , für $Y < X$ $Z=0$ , für $Y \geq X$	$1 \quad 4 \quad !4 < 3 \quad 2 \quad !1$ $1 \quad 0 \quad 1$																																																
≤	--		<b>Prüfen auf 'kleiner-gleich'</b> $Z \leftarrow Y \leq X$ $Z=1$ , für $Y \leq X$ $Z=0$ , für $Y > X$	$1 \quad 4 \quad !2 \leq 3 \quad 4 \quad !5$ $1 \quad 0 \quad 1$																																																
=	--		<b>Prüfen auf 'gleich'</b> $Z \leftarrow Y = X$ wenn Y und X numerische Grössen sind: $Z=1$ , für $Y=X$ $Z=0$ , für $Y \neq X$ wenn Y oder X alpha-numerisch ist: $Z=1$ , wenn Y und X identisch $Z=0$ , wenn nicht identisch	$3 \quad 4 \quad !3 = 3 \quad !4 \quad !3$ $1 \quad 0 \quad 1$ 'ABC'='ADC' $1 \quad 0 \quad 1$ 'ABC'=19 $0 \quad 0 \quad 0$																																																
>	--		<b>Prüfen auf 'grösser'</b> $Z \leftarrow Y > X$ $Z=1$ , für $Y > X$ $Z=0$ , für $Y \leq X$	$5 \quad !1 \quad 6 > 5 \quad 1 \quad !9$ $0 \quad 0 \quad 1$																																																

Tabelle 1: Skalare Funktionen

②

Operator	monadisch		dyadisch							
	Definition	Beispiele	Definition	Beispiele						
≧	--		<b>Prüfen auf 'grösser-gleich'</b> $Z \leftarrow Y \geq X$ $Z=1$ für $Y \geq X$ $Z=0$ für $Y < X$	$1 \ 0 \ 1$ $5 \ 1 \ 6 \geq 5 \ 1 \ 9$						
≠	--		<b>Prüfen auf 'ungleich'</b> $Z \leftarrow Y \neq X$  wenn Y und X numerische Grössen sind:  $Z=1$ , für $Y \neq X$ $Z=0$ , für $Y = X$  wenn Y oder X alpha-numerisch ist:  $Z=1$ , wenn Y nicht identisch mit X $Z=0$ , wenn identisch	$0 \ 1 \ 0$ $3 \ 4 \ 3 \neq 3 \ 4 \ 3$  $0 \ 1 \ 0$ 'ABC' ≠ 'ADC'  $1 \ 1 \ 1$ 'ABC' ≠ 19						
∧	--		<b>Mit 'und' verknüpfen</b> $Z \leftarrow Y \wedge X$  $Y, X \in \{0,1\}$	<table style="border-collapse: collapse; margin: auto;"> <tr><td style="border-right: 1px solid black; padding: 5px;">∧</td><td style="padding: 5px;">0 1</td></tr> <tr><td style="border-right: 1px solid black; padding: 5px;"></td><td style="padding: 5px;">0 0 0</td></tr> <tr><td style="border-right: 1px solid black; padding: 5px;"></td><td style="padding: 5px;">1 0 1</td></tr> </table>	∧	0 1		0 0 0		1 0 1
∧	0 1									
	0 0 0									
	1 0 1									
∨	--		<b>Mit 'oder' verknüpfen</b> $Z \leftarrow Y \vee X$  $Y, X \in \{0,1\}$	<table style="border-collapse: collapse; margin: auto;"> <tr><td style="border-right: 1px solid black; padding: 5px;">∨</td><td style="padding: 5px;">0 1</td></tr> <tr><td style="border-right: 1px solid black; padding: 5px;"></td><td style="padding: 5px;">0 0 1</td></tr> <tr><td style="border-right: 1px solid black; padding: 5px;"></td><td style="padding: 5px;">1 1 1</td></tr> </table>	∨	0 1		0 0 1		1 1 1
∨	0 1									
	0 0 1									
	1 1 1									
⊗	--		<b>Mit 'nicht gleichzeitig' verknüpfen</b> $Z \leftarrow Y \otimes X$  $Y, X \in \{0,1\}$	<table style="border-collapse: collapse; margin: auto;"> <tr><td style="border-right: 1px solid black; padding: 5px;">⊗</td><td style="padding: 5px;">0 1</td></tr> <tr><td style="border-right: 1px solid black; padding: 5px;"></td><td style="padding: 5px;">0 1 1</td></tr> <tr><td style="border-right: 1px solid black; padding: 5px;"></td><td style="padding: 5px;">1 1 0</td></tr> </table>	⊗	0 1		0 1 1		1 1 0
⊗	0 1									
	0 1 1									
	1 1 0									
⊕	--		<b>Mit 'weder – noch' verknüpfen</b> $Z \leftarrow Y \oplus X$  $Y, X \in \{0,1\}$	<table style="border-collapse: collapse; margin: auto;"> <tr><td style="border-right: 1px solid black; padding: 5px;">⊕</td><td style="padding: 5px;">0 1</td></tr> <tr><td style="border-right: 1px solid black; padding: 5px;"></td><td style="padding: 5px;">0 1 0</td></tr> <tr><td style="border-right: 1px solid black; padding: 5px;"></td><td style="padding: 5px;">1 0 0</td></tr> </table>	⊕	0 1		0 1 0		1 0 0
⊕	0 1									
	0 1 0									
	1 0 0									
~	<b>Boole'sch negieren</b> $Z \leftarrow \sim X$  $Z=1$ , für $X=0$ $Z=0$ , für $X=1$  $X \in \{0,1\}$	$\sim 1$  $0$  $\sim 1 \ 0 \ 1 \ 0 \ 1$ $0 \ 1 \ 0 \ 1 \ 0$	--							
?	<b>Zufallszahl auswählen</b> $Z \leftarrow ?X$  Z ist eine ganze Zahl, die «zufällig» aus den Zahlen des Indexvektors ?X (s. Abschnitt 2.6.1.) ausgewählt wird. Alle Zahlen dieser Reihe haben die gleiche Auswahlchance.	$?8$ $5$  $?8 \ 8 \ 8 \ 8 \ 8 \ 8 \ 8 \ 8$ $7 \ 4 \ 5 \ 2 \ 8 \ 1 \ 6 \ 4$  $?6 \ 12 \ 25 \ 87 \ 8$ $6 \ 5 \ 13 \ 73 \ 3$	(s. Gemischte Funktionen)							

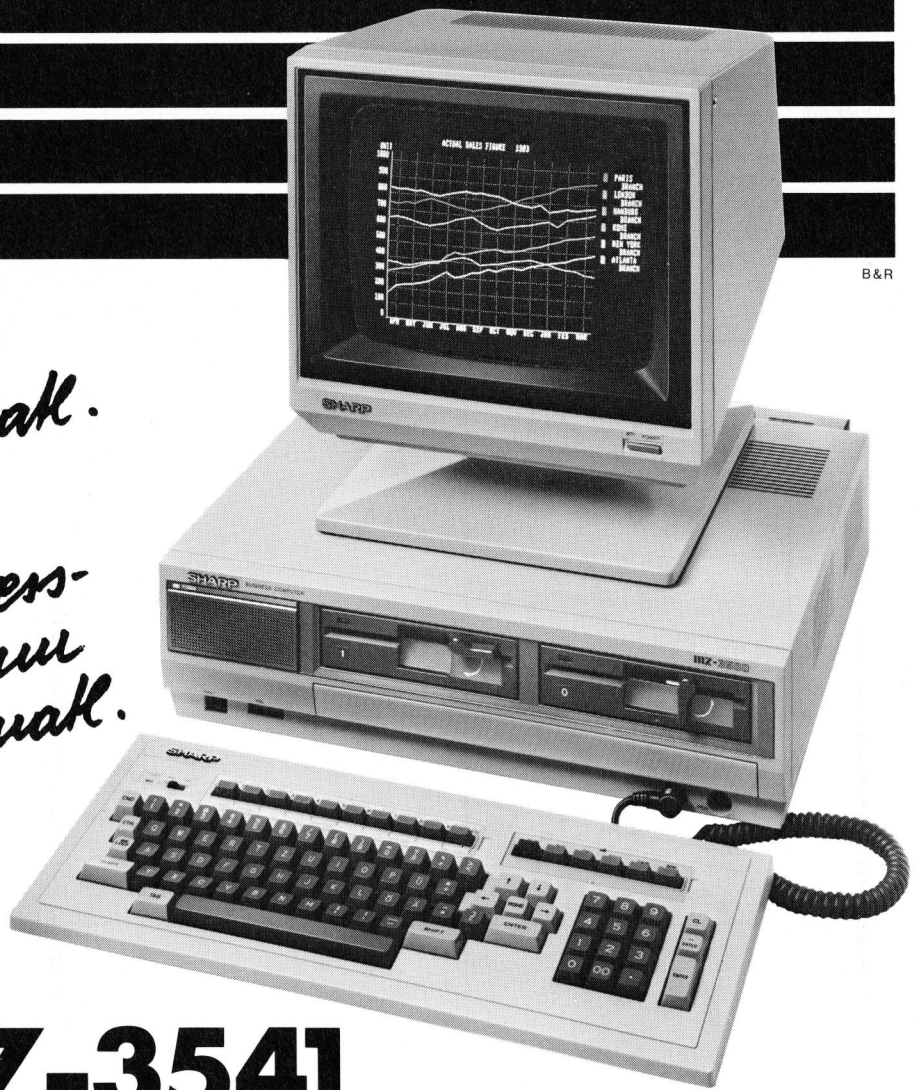
Tabelle 1: Skalare Funktionen

**SHARP**

B&R

\* a.  
mit Finanzbuch-  
haltungsprogramm  
nur Fr. 307.25 monatl.

\* b.  
mit Text- und Adress-  
verwaltungprogramm  
nur Fr. 284.60 monatl.



Die Gelegenheit!

# Sharp MZ-3541 Bürocomputer, komplett: \* mit Bürofach-Leasing

- 128 KByte Arbeitsspeicher
- 2 Diskettenlaufwerke à 390 KByte
- Betriebssystem EOS (CP/M-kompatibel)
- Drucker: Facit 4512 140 Zeichen/Sek., A4 quer
- schwenkbarer Bildschirm
- frei bewegliche, sehr flache Tastatur mit Umlauten
- weitere Sharp-Lizenzprogramme für den ganzen Bürobereich und viele andere CP/M-Programme wie: D-Base, Multiplan, Wordstar usw.

Generalvertretung für die Schweiz + FL:

**Facit-Addo AG**  
Badenerstrasse 587  
8048 Zürich  
Telefon 01/491 42 60

Coupon für Gratis-Dokumentation MZ-3541/Leasing

Firma

Adresse

Sachbearbeiter

PLZ, Ort

Telefon

bitte einsenden an nebenstehende Adresse

M+K 84-6

**Durch Nachdenken vorn.**

**SHARP**



# Die Programmiersprache C

**Nebst der Möglichkeit, Funktionszeiger in Strukturen einzubetten, können Funktionen universell anwendbar gemacht werden, indem man beim Aufruf Funktionszeiger als Parameter übergibt. Innerhalb einer Struktur darf ein Zeiger auf die gleichnamige Struktur stehen, was als «Rekursivität» bezeichnet wird. Mit einer «union» kann man einen gemeinsamen Speicherbereich für verschiedene Variablentypen definieren.**

## 6.7 Zeiger auf Funktionen als Funktionsparameter

### 6.7.1 Problemstellung

Das Beispiel 25 ist nicht ernst gemeint, es soll lediglich auf einfache Weise zeigen, wie man Funktionen verallgemeinern kann. Die Problemstellung lautet wie folgt: Es werden hintereinander Messungen für den diastolischen und systolischen Blutdruck und für das Körpergewicht eines Menschen ausgeführt und diese drei Werte als ganze Zahlen eingegeben.

Nach jeder Eingabe ist eine Prüfung durchzuführen, wobei:

- der diastolische Blutdruck mindestens 70 sein muss
- die Differenz zwischen dem systolischen und diastolischen höchstens 100 sein darf und
- das Gewicht mindestens 50 sein muss und höchstens 120 sein darf.

## Prof. Dr. Erwin Nievergelt

Wie bereits bemerkt hat diese Problemstellung keine medizinische Bedeutung.

### 6.7.2 Bildung einer Universalfunktion

Für die drei Prüfungsarten bilden wir die drei Funktionen «diastolisch(n)», «systolisch(n)» und «gewicht(n)». Alle drei Funktionen haben einen Parameter «n», welcher den Messwert aufnimmt. Hält ein Wert der Prüfung nicht stand, so wird -1 als Funktionswert zurückgegeben, andernfalls der Messwert selbst.

Die Eingabe eines Messwertes übernimmt die Funktion «eingabe()». Damit diese nicht eine Fallunterscheidung durchführen und je nach Resultat eine der drei Funktionen aufrufen muss, kann man eine sogenannte Universalfunktion bilden. Wir nennen Sie:

(\*fkt)()

Die Universalfunktion ist gleichzeitig ein Parameter der Funktion «ein-

gabe()». Beim Aufruf von «eingabe()» wird an dessen Stelle als Argument jeweils die entsprechende Funktion eingesetzt. Das zweite Argument ist ein Zeichenzeiger auf den Text.

Die drei Aufrufe lauten wie folgt:

eingabe(diastolisch, «diastolischer Blutdruck ?»);

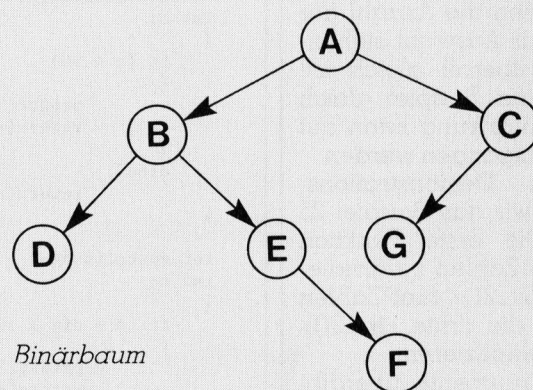
eingabe(systolisch, «systolischer Blutdruck ?»);  
 eingabe(gewicht, «Gewicht ?»);

«diastolisch», «systolisch» und «gewicht» sind Zeiger, welche auf die entsprechenden Funktionen weisen. Damit der Compiler weiss, was diese Worte bedeuten, müssen sie vorher als Funktionen, welche ganzzahlige Werte liefern, deklariert werden:

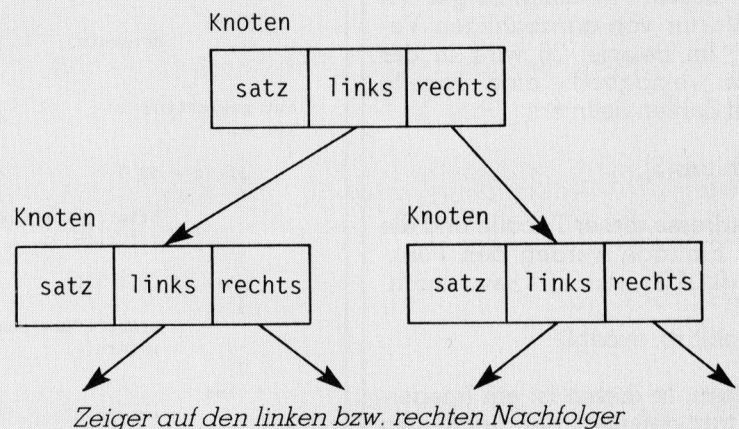
```
int diastolisch();
int systolisch();
int gewicht();
```

Erfolgt beispielsweise der Aufruf

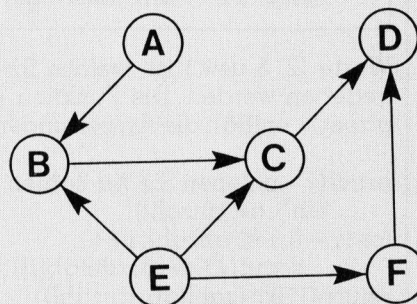
eingabe(diastolisch, «diastolischer Blutdruck ?»);



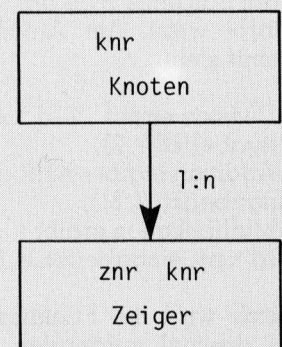
Binärbaum



Zeiger auf den linken bzw. rechten Nachfolger



Netzwerk-Struktur



1:n Beziehung zwischen Knoten und Zeiger

Bild 14

so wird «(\*fkt)» in der Funktion «eingabe()» durch «diastolisch» ersetzt. Dieser wird der eingegebene Wert «n» als Argument übergeben.

Die drei Funktionen «diastolisch(n)», «systolisch(n)» und «gewicht(n)» haben alle einen einzigen Parameter, nämlich «n». Hier könnten natürlich mehrere Parameter stehen, z.B. diastolisch(r,s,t). In der Universalfunktion müssten dann auch drei Argumente erwähnt werden, z.B.

```
int x1, x2, x3;
(*fkt)(x1, x2, x3);
```

### 6.7.3 «Variable» Anzahl von Parametern

Bei oberflächlicher Betrachtung hat man nun den Eindruck, mit dieser Methode liessen sich nur Probleme lösen, bei denen die Anzahl der Argumente der zur Auswahl stehenden Funktionen überall gleich ist, also wie im letzten Beispiel gleich drei. Diese Einschränkung kann auf elegante Weise umgangen werden.

Wiederum zu Demonstrationszwecken ändern wir das Beispiel 25 wie folgt ab: Die erste Funktion «fkt\_1()» hat zwei Zahlen zu dividieren, die zweite «fkt\_2()» fünf Zahlen zu addieren und die dritte «fkt\_3()» drei Zahlen zu multiplizieren.

Anstelle der Argumente übergibt man in diesem Fall einen Zeiger auf einen Vektor von ganzzahligen Variablen. Im Beispiel 26 wird in der Funktion «eingabe()» eine Tabelle von fünf Zahlen definiert:

```
int zahltab[5];
```

Die Adresse dieser Tabelle und die Anzahl Einträge werden den Funktionen «fkt\_1()», «fkt\_2()» usw. durch

```
(*fkt)(zahltab, anzahl)
```

übergeben. In diesen ist ein Integerzeiger tab definiert, der nach dem Aufruf auf das erste Byte des Vektors «zahltab[]» weist. Am Anfang des Programms steht

```
printf(«Division ergibt : %ld\n»,
    eingabe(fkt_1, 2));
printf(«Addition ergibt : %ld\n»,
    eingabe(fkt_2, 5));
printf(«Multiplikation ergibt :
    %ld\n», eingabe(fkt_3, 3)).
```

Dadurch wird die Funktion «eingabe()» dreimal aufgerufen, wobei ihr mitgeteilt wird, welche Funktion (fkt\_1, fkt\_2 usw.) sie ihrerseits aufrufen soll und wie gross die Anzahl der

```
/* Pruefung einer eingegebenen Zahl */
int dia, syst, gew;

main()
{
    int    diastolisch();
    int    systolisch();
    int    gewicht();

    while ((dia = eingabe(diastolisch, "diastolischer Blutdruck ?") < 0);
    while ((syst = eingabe(systolisch, "systolischer Blutdruck ?") < 0);
    while ((gew = eingabe(gewicht, "Gewicht ?") < 0);
    printf("\nDie Eingabewerte %d %d %d sind in Ordnung\n",
        dia, syst, gew);
}

int eingabe(fkt, text)
int (*fkt) ();
char    *text;
{
    int    n;
    printf("\n%s\n", text);
    scanf("%d", &n);
    return((*fkt)(n));
}

int diastolisch(n)
int n;
{
    if (n < 70)
    {
        printf("Der diastolische Blutdruck ist zu tief\n");
        return(-1);
    }
    else
        return(n);
}

int systolisch(n)
int n;
{
    if (n - dia > 100)
    {
        printf("Die Differenz ist zu gross\n");
        return(-1);
    }
    else
        return(n);
}

int gewicht(n)
int n;
{
    if (n < 50 )
    {
        printf("Das Gewicht ist unwahrscheinlich klein\n");
        return(-1);
    }
    else if (n > 120)
    {
        printf("Das Gewicht ist unwahrscheinlich gross\n");
        return(-1);
    }
    else
        return(n);
}
```

Beispiel 25: Funktionszeiger als Funktionsparameter

Werte (2, 5 usw.) ist, welche ihr eingegeben werden. Die Funktion «eingabe()» enthält die Anweisungen

```
printf(«\nGeben Sie %d Zahlen
    ein\n», anzahl);
for (i = 0; i < anzahl; i++)
    scanf(«%d», &zahltab[i]);
return((*fkt)(zahltab, anzahl));
```

Die printf-Anweisung besagt, dass 2, 5 usw. Zahlen einzugeben sind, während die scanf-Anweisung dafür

sorgt, dass die eingegebenen Zahlen jeweils in den Variablen «zahltab[0]», «zahltab[1]» usw. gespeichert werden. Der Aufruf der entsprechenden Funktion erfolgt in der return-Anweisung, weil ja der Funktionswert seinerseits der main-Funktion mitgeteilt werden muss.

```
*(tab + i)
```

ist der Wert der i-ten Variablen des Vektors «zahltab[]». Durch

```

/* verschiedene arithmetische Funktionen */

int i, n;

main()
{
    long   fkt_1();
    long   fkt_2();
    long   fkt_3();
    long   eingabe();

    printf("Division ergibt : %ld\n", eingabe(fkt_1, 2));
    printf("Addition ergibt : %ld\n", eingabe(fkt_2, 5));
    printf("Multiplikation ergibt : %ld\n", eingabe(fkt_3, 3));
    printf("\nAlle drei Funktionen ausgefuehrt\n");
}

long   eingabe(fkt, anzahl)
long   (*fkt) ();
int   anzahl;
{
    int   zahltab[5];
    printf("\nGeben Sie %d Zahlen ein\n", anzahl);
    for (i = 0; i < anzahl; i++)
        scanf("%d", &zahltab[i]);
    return((*fkt)(zahltab, anzahl));
}

long   fkt_1(tab, anz)
int   *tab;
int   anz;
{
    return(*tab / *(tab + 1));
}

long   fkt_2(tab, anz)
int   *tab;
int   anz;
{
    for (n = 0, i = 0; i < anz; i++)
        n += *(tab + i);
    return(n);
}

long   fkt_3(tab, anz)
int   *tab;
int   anz;
{
    for (n = 1, i = 0; i < anz; i++)
        n *= *(tab + i);
    return(n);
}

```

Beispiel 26: «Variable» Anzahl von Parametern

return(\*tab / \*(tab + 1));

wird «zahltab[0]» durch «zahltab [1]» dividiert, durch

```
for (n = 0, i = 0; i < anz; i++)
    n += *(tab + i);
```

werden die ersten «anz» Werte von «zahltab» zu n addiert und durch

```
for (n = 1, i = 0; i < anz; i++)
    n *= *(tab + i);
```

wird n das Produkt der ersten «anz» Werte von «zahltab» zugewiesen.

## 6.8 Rekursivität von Strukturen

Verschiedene Anwendungen benötigen sogenannte rekursive Datenstrukturen. Darunter sind die Baumstrukturen wohl die wichtigsten. Um solche Anwendungen ele-

gant programmieren zu können, wurde die Möglichkeit geschaffen, in Strukturen Zeiger einzubauen, welche auf den gleichen Strukturtyp weisen, also quasi auf sich selbst.

### 6.8.1 Definition und Darstellung von Binärbäumen

Bei der Verwendung von Baumstrukturen spielen wiederum die Binärbäume eine besonders grosse Rolle. Für Freunde abstrakter Denk-

prozesse geben wir die rekursive Definition von Binärbäumen:

Unter einem Binärbaum versteht man eine Menge von Knoten, welche entweder leer ist oder aus einer Wurzel und zwei disjunkten Binärbäumen besteht, die man linker und rechter Teilbaum nennt.

Zum leichteren Verständnis haben wir im Bild 14 einen Binärbaum dargestellt. Links unterhalb der Wurzel «A» befindet sich ein Teilbaum mit der Wurzel «B». Dieser wiederum hat einen linken Teilbaum, welcher nur aus dem Knoten «D» besteht und einen rechten, welcher die Wurzel «E» hat. Der rechte Teilbaum von «A» hat die Wurzel «C» und einen linken Teilbaum «G». Die Endknoten «F» und «G» (auch «Blätter» genannt) sind rechts bzw. links von ihren Vorgängern angeordnet.

Einem Knoten kann ein Satz einer Datei (Relation) zugeordnet werden. Dadurch wird innerhalb der zugeordneten Entitätsmenge ein 1:n Beziehung (Baumstruktur, Hierarchie) definiert. Im C-Programm wird der Datensatz wie üblich als Struktur deklariert, z.B. wie in Abb. 1.

Die Beziehung zwischen den Sätzen kann damit aber noch nicht dargestellt werden. Dazu benötigt man eine Struktur, welche wir «knoten» nennen wollen und noch Zeiger auf diese Struktur.

```

struct knoten
{
    struct satz   psatz;
    struct knoten *links;
    struct knoten *rechts;
};

```

Man beachte, dass lediglich Zeiger auf die Struktur «knoten» eingebettet werden können, weil deren Länge implementations- aber nicht strukturabhängig ist. Die Einbettung einer Struktur würde den Compiler überfordern, weil er Platz für eine Struktur reservieren müsste, deren Länge er noch nicht kennt.

Ein Binärbaum von z.B. 100 Knoten würde nun wie folgt definiert:

```
struct knoten   bibaum[100];
```

```

struct satz
{
    long   pnr;           /* Personalnummer */
    char   pname[30];     /* Name und Vorname */
    char   pstrasse[20];  /* Strasse und Hausnummer */
    char   plz_ort[25];   /* PLZ und Wohnort */
    usw.
};

```

Abb. 1

Beim Aufbau müssen die Adressen der Nachfolger in den Knoten eingetragen werden.

## 6.8.2 Netzwerk-Strukturen

Im Gegensatz zu einer Baumstruktur darf ein Knoten in einem Netzwerk *mehrere Vorgänger* haben (vgl. Bild 14). Dies bedeutet, dass ein Knoten eine variable Anzahl von Zeigern besitzt. In einem relationalen Datenmodell muss deshalb ein Normalisierungsprozess durchgeführt, d.h. der Satz «knoten» in die erste Normalform gebracht werden. Dazu benötigt man einen neuen Entitätstyp, den wir «zeiger» nennen wollen. Die Deklaration der Strukturen sieht dann wie in Abb. 2 aus. Die 1:n Beziehung ist im Bild dargestellt.

```

struct knoten
{
    long   knr;    /* Schlüssel */
    struct satz   ksatz;
};

struct zeiger
{
    long   znr;    /* Schlüssel */
    long   knr;    /* Fremdschlüssel */
    struct knoten *zknoten; /* Zeiger auf
                           die Struktur knoten */
};

```

Abb. 2

## 6.9 Der Datentyp «union»

### 6.9.1 Zweck und Darstellung

Bei der Systemprogrammierung kommt es vor, dass man den gleichen Speicherplatz auf zwei oder mehr Arten benützen möchte. Beispielsweise möchte man in einem Bereich einmal mit einer Struktur «a», das andere mal mit einer Struktur «b», einmal mit «c» usw. arbeiten. Dafür wurde der Datentyp «union» geschaffen, welcher auf deutsch «Variante» genannt wird.

```

union bereich
{
    struct a  a_bereich;
    struct b  b_bereich;
    struct c  c_bereich;
};

```

Den dadurch festgelegten gemeinsamen Speicherbereich kann

man einmal mit den Augen der Struktur «a», das andere mal mit denjenigen der Struktur «b» usw. sehen. Der Compiler reserviert Speicherplatz für die grösste Struktur.

### 6.9.2 Speicherung von Knoten in einem festen Bereich

Das Problem der Ablage von grossen Binärbäumen auf dem Plattenspeicher führt dazu, dass die Knoten in Teilmengen unterteilt und diese festen Speicherbereichen, «Seiten» genannt, zugeteilt werden müssen. Im einen Fall haben die Sätze der Knoten z.B. die Länge 38, im andern Fall 114. Die Länge variiert also von Baum zu Baum. Um dieses Problem zu lösen, kann man eine «union» bilden, welche den Speicherbereich von z.B. 1000 Bytes einmal als Zeichenvektor, das andere Mal als Zahlenvektor sieht:

```

/* Gemeinsamer Bereich fuer Zeichen und Zahlen */

main()
{
    union   seite
    {
        char   c[1000];
        long   z[250];
    } u;

    int     satz_laenge;
    int     knoten_laenge;
    int     anzahl;

    /* Eingabe der Satzlaenge */

    printf("Satzlaenge ?\n");
    scanf("%d", &satz_laenge);

    /* Berechnung der Anzahl Knoten */

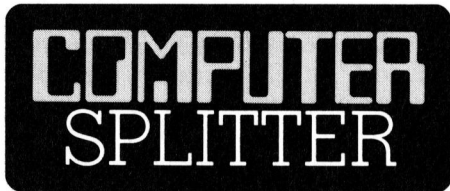
    knoten_laenge = satz_laenge + 2 * sizeof(long);
    anzahl = 1000 / knoten_laenge;

    /* Zugriff zum 1. Byte des 15. Datensatzes */

    u.c[14 * knoten_laenge] = 'X';
    printf("Knotenlaenge = %d\tZeichen = %c\n",
           knoten_laenge, u.c[14 * knoten_laenge]);
}

```

Beispiel 27: Organisation einer Binärbaumstruktur



## Victor in neuen Händen

(283/fp) Der Leidensweg von Victor Technologies Inc., dem Produzenten des Sirius/Victor I, findet ein Ende. Seit April dieses Jahres wird wieder mit Gewinn produziert und nun hat der schwedische Software-Jumbo Datatronic die Firma käuflich übernommen. Geht der Stern Sirius wieder auf? □

## Informationen zum HP-150

(271/fp) Das kürzlich an die Europäischen Empfänger gelangte Augustheft des Hewlett-Packard Journal ist voll und ganz dem HP-150 (M+K 84-3) gewidmet. Die acht Artikel geben detailliert und doch einfach verständlich Auskunft über die Philosophie des HP-150, das durch PAM aufgerüstete MS-DOS 2.11, die Firmware, den Aufbau des Kontakt-schirms, die Anwendersoftware mit besonderer Berücksichtigung der Grafik, über den eingebauten Drucker und die zum HP-Standard bestimmte Tastatur. Das HP-Journal ist gratis bei HP erhältlich und kann auch kostenlos abonniert werden. A propos HP-Journal: Schon die Juli-Nummer enthielt reichlich Informationen zu einem Kleinen von - dem HP-71B. □

```
union seite
{
    char c[1000];
    long z[250];
};
```

In diesem Beispiel sind auf modernen Computern die beiden Bereiche gleich lang, was aber bei einer union nicht unbedingt der Fall sein muss. Beispiel 27 zeigt einen Programmausschnitt, der die Organisation einer Binärbaumstruktur veranschaulicht. Zur Illustration wird im ersten Byte des 15. Datensatzes der Buchstabe «X» gespeichert und anschliessend wieder gelesen.

```
union seite
{
    char c[1000];
    long z[250];
} u;
```

deklariert die Variante «seite» und reserviert unter dem Namen «u» Speicherplatz für ein Vorkommnis. Der Einfachheit halber wird die Satzlänge interaktiv erfragt:

```
printf(»Satzlänge ?\n»);
scanf(»%d», &satz_länge);
```

Nun müssen die Knotenlänge und die Anzahl der Knoten, die in der Seite Platz haben, berechnet werden. Da die Länge einer long-Zahl theoretisch von einem Computer zum andern variieren könnte, ist es vorsichtig, nicht 4 einzusetzen, sondern diese mit «sizeof(long)» zu bestimmen.

```
knoten_länge = satz_länge + 2 *
                sizeof(long);
anzahl = 1000 / knoten_länge;
```

Der Zugriff zum 1. Byte des 15. Datensatzes gestaltet sich wie folgt:

```
u.c[14 * knoten_länge] = 'X';
printf(«Knotenlänge = %d\tZeichen
      = %c\n»,
      knoten_länge, u.c[14 *
      knoten_länge]);
```

Bis anhin haben wir uns immer in der Zentralspeicher «aufgehalten». In der kommerziell-administrativen Datenverarbeitung müssen aber grössere Datenmengen in peripheren Medien gespeichert und gezielt wiederaufgefunden werden können. Wie das gemacht wird, zeigt die nächste Folge, wo die Ein- und Ausgabe von Datensätzen behandelt wird. □

# COMPUTER SPLITTER

## Entlassungen bei Atari

(285/fp) Bei Atari wird mit brutalen Methoden «gesundgeschrumpft»: Allein bei der deutschen Niederlassung des Unternehmens wurden 42 Personen entlassen, genau die Hälfte der Belegschaft. Dies geschieht im Rahmen eines Sanierungskonzepts unter dem neuen Besitzer von Atari, Jack Tramiel. □

## Bücher von der Quelle

(284/fp) Ab 1985 gibt es auch in deutsch Bücher von Microsoft zu haben. Die Firma will ihre Wissens- und Erfahrungsvorsprünge dem Anwender in Buchform zugute kommen lassen: 60000 Stück eines Buchs zum Macintosh wurden in den USA innert zweier Monate verkauft. Die anderen Themen der ersten drei Bände: MS-DOS und der PC-jr. □

# Satz tippen nach Druckerei Art! Auf Ihrem eigenen Computer.

Der technische Fortschritt ermöglicht Ihnen heute, auf einfache Weise Ihren Satz selbst herzustellen.

In gewissen Bereichen der Verwaltung und der Industrie müssen laufend Texte gesetzt und gedruckt werden. Hier bietet die direkte Umsetzung von Informationen aus der EDV deutliche Vorteile:

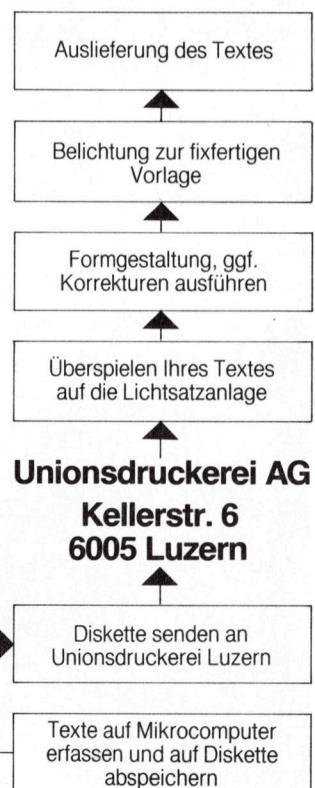
## Reduktion des Zeitaufwandes Kostensparende Fertigung

Sie tippen also Ihren Text auf Ihrem Mikrocomputer (z.B. Sirius). Die Diskette mit dem gespeicherten Satz und mit den Angaben über Schriftart, Schriftgrösse, Spaltenbreite usw. senden Sie an uns. Das ist alles! Das übrige besorgen wir resp. unsere Lichtsatzanlage.

Die von Ihnen erfassten Daten werden auf diese Anlage übertragen, in die von Ihnen gewünschte Form gebracht und belichtet. Ihren reprofähigen Satz, auf Papier oder Film, erhalten Sie umgehend.

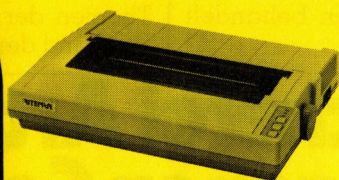
Rufen Sie uns doch an!  
**Telefon (041) 44 24 44**

Wir orientieren Sie gerne über nähere Details.



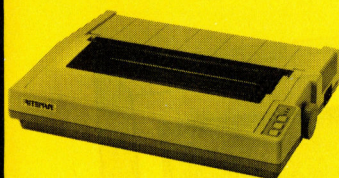
# :RITEMAN:

## Die Professionellen für Professionelle



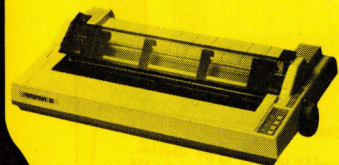
### Riteman A1 Personal Printer

120 Zeichen/sek.  
Endlos- und Einzelblattführung  
internationale Zeichensätze  
A4  
Fr. 1100.-



### Riteman II Professional Printer

160 Zeichen/sek.  
Endlos- und Einzelblattführung  
256 programmierbare Zeichen  
A4  
Fr. 1415.-



### Riteman 15 Business Printer

160 Zeichen/sek.  
Endlos- und Einzelblattführung  
256 programmierbare Zeichen  
A4 quer  
Fr. 2100.-

Verkauf durch den Fachhandel

# LOGOTRON AG

Datacom · Instruments · Peripherals

8805 Richterswil 01 784 22 26  
1052 Le Mont s/Lausanne 021 32 30 22

# Zahlentheorie mit dem HP-41

**Eigentlich sollte es ein Aufsatz über Algorithmen zu den Primfaktoren werden. Herausgeschaut hat ein Exkurs in die Zahlentheorie und «The Art Of Computer Programming». Das Standardwerk von D. E. Knuth mit diesem Titel, Euklid und Fermat, haben dem Artikel Pate gestanden.**

Wir meinen, dass mit der Weiterführung des im folgenden beschriebenen Konzepts die Primfaktorisierung von Zahlen  $> 10^{10}$  in akzeptablen Zeiten (Tage?) möglich sein sollte. Dazu würde das Werk von D. E. Knuth «The Art Of Computer Programming», insbesondere Vol. II, noch einen erheblichen Beitrag lei-

## Friedrich Hillebrandt

sten können. Auch die folgenden Ausführungen beruhen weitgehend auf den Arbeiten von Knuth (Vol. II, Ch. 4.5.4.).

Unser Programm kombiniert zwei unterschiedliche Algorithmen, die in ihrer Primitivform zunächst kurz vorgestellt werden sollen.

### Algorithmus E

Dies ist die bekannte konventionelle Teilerstestmethode. Wir vermuten, dass sie auf Euklid zurückgeht (Die Elemente, 9. Buch § 14).

Es wird nacheinander geprüft, ob 2, 3, 4, 5, 6, ... Teiler der gegebenen Zahl n sind. Der erste so erhaltene Teiler ist der kleinste Primteiler von n. Ist bis  $\sqrt{n}$  kein Teiler gefunden, so ist n eine Primzahl, denn ein Teiler  $p > \sqrt{n}$  ( $p \neq n$ ) kann nicht kleinster Primteiler sein, da mit ihm sein Komplementärteiler  $n/p < \sqrt{n}$  vorher als Teiler erkannt worden wäre (siehe Bild 1).

### Algorithmus F

Diese Methode geht auf Fermat zurück. n sei die zu faktorisierende Zahl. Ist  $n = F_1 \cdot F_2$  ( $F_1 \leq F_2$ ), so kann man für  $F_1$  und  $F_2$  den Ansatz  $F_1 = x - y$  und  $F_2 = x + y$  für geeignete (nicht notwendig ganze) Zahlen x und y machen. Dann ist  $x = (F_2 + F_1) / 2$  und  $y = (F_2 - F_1) / 2$ . Insbesondere sind x und y ganzzahlig, falls  $F_1$  und  $F_2$  beide ungerade sind.

Mit dem Fermat'schen Algorithmus werden für ungerades n diese Werte x und y gesucht: Ausgehend von  $x = \lceil \sqrt{n} \rceil$  (kleinste ganze Zahl  $\geq \sqrt{n}$ ) wird x erhöht und jeweils getestet, ob  $x^2 - n$  eine Quadratzahl (nämlich  $y^2$ ) ist. Man beachte:  $n = (x - y)(x + y) =$

$x^2 - y^2$ . Mit Algorithmus F wird der grösste Faktor (nicht notwendig Primfaktor)  $\leq \sqrt{n}$  gefunden (siehe Bild 2).

Es versteht sich, dass die beiden abgedruckten Programme «Euklid» und «Fermat» nicht für den praktischen Gebrauch bestimmt sind, sondern nur zur Charakterisierung der Algorithmen dienen sollen.

Dazu seien zwei Beispiele mit den jeweiligen Rechenzeiten angegeben (Bild 3).

Ist nur die Alternative zwischen Algorithmus E und F gegeben, so entscheidet man sich ohne Zweifel für Algorithmus E - ca.  $\sqrt{n}$  Tests gegenüber  $n/4 \cdot \sqrt{n}$  ( $\gg n$  bei grossem n) Tests - zumal damit auf einfache Weise der Restfaktor weiter faktorisiert werden kann.

Beim Auswerten der gemessenen Rechenzeiten fällt aber doch etwas auf: Der Abstand vom Startwert ( $\sqrt{n}$  bei F und 2 bei E) bis zum ersten gefundenen Faktor ist bei beiden Beispielen ungefähr gleich:

$$\sqrt{1001423} - 887 = 113,7 \approx 113;$$

d.h., es war zu erwarten, dass Algorithmus F für 1001423 ungefähr die gleiche Zeit (oder mehr, da der je-

weilige Test aufwendiger ist) wie Algorithmus E für 1001293 benötigte. Aber offensichtlich ist hier der Fermat'sche Algorithmus nahezu siebenmal so schnell. Das verwundert nicht, wenn man beachtet, dass in F mit der ersten Erhöhung von

$$x = \lceil \sqrt{n} \rceil = \lceil \sqrt{1001423} \rceil = 1001$$

um 1 der «Testfaktor»  $p = x - y$  einen «Sprung» von

$$(x - \sqrt{x^2 - n}) - ((x+1) - \sqrt{(x+1)^2 - n}) = 25,8$$

macht, während der Testfaktor im Algorithmus E konstant um 1 wächst.

Die Veränderung des Testfaktors p in F wird mit wachsendem x allerdings immer geringer, was sich in der unverhältnismässig hohen Rechenzeit beim Beispiel 1001293 ausdrückt. Man kann aber mit Recht erwarten, dass mit Algorithmus F grosse Faktoren wesentlich schneller gefunden werden als mit E.

Es liegt nahe, die beiden Verfahren zu kombinieren und den Uebergang so zu wählen, dass eine optimale Rechengeschwindigkeit erreicht wird. Selbstverständlich wird zunächst Algorithmus E bis zu einem grössten Testfaktor  $p_0$  angewandt, da damit schnell kleine Faktoren gefunden werden und somit in der Regel nur noch ein Restfaktor weiterfaktorisiert werden muss. Ausserdem brauchen bei anschliessender Anwendung von Algorithmus F nur solche x mit  $x - y$  zwischen  $\sqrt{n}$  und  $p_0$  getestet zu werden. Die Berechnung eines optimalen Wertes  $p_0$  wird wei-

```

01♦LBL "EUK
                    LID"
02 STO 00
03 " IST PRI
                    M"
04 VIEW 00
05 SQRT
06 STO 02
07 1
08 STO 01

09♦LBL 00
10 1
11 ST+ 01
12 RCL 02
13 RCL 01
14 X>Y?
15 GTO 01
16 RCL 00
17 RCL 01
18 MOD
19 X≠0?
20 GTO 00
21 "HAT TEI
                    LER "
22 ARCL 01

23♦LBL 01
24 BEEP
25 PROMPT
26 END
                
```

```

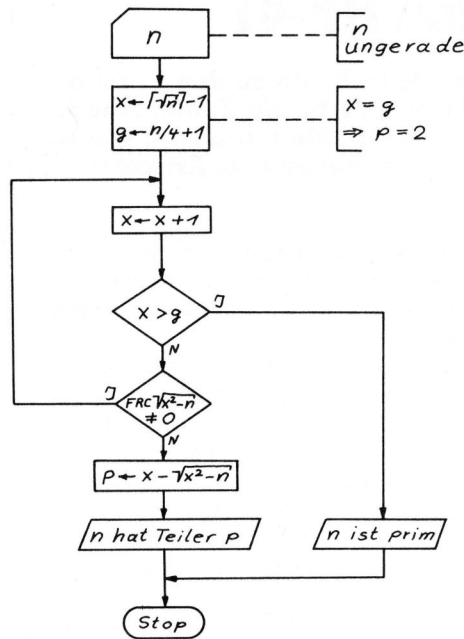
graph TD
    n[n] --> g["g ← √n"]
    g --> p["p ← 1"]
    p --> pinc["p ← p + 1"]
    pinc --> pgtg{"p > g"}
    pgtg -- J --> prim["n ist prim"]
    pgtg -- N --> modp{"n mod p = 0"}
    modp -- J --> teiler["n hat Teiler p"]
    modp -- N --> pinc
    prim --> stop((Stop))
    teiler --> stop
    
```

Bild 1: Algorithmus E mit Listing

```

01 *LBL "FER" 23 *LBL 00
    MAT"      24 1
02 STO 00    25 ST+ 01
03 "IST PRI  26 RCL 02
    M"        27 RCL 01
04 VIEW 00   28 X>Y?
05 2         29 GTO 01
06 MOD      30 RCL 01
07 1/X      31 X↑2
08 RCL 00   32 RCL 00
09 SQRT     33 -
10 RCL X    34 SQRT
11 -1       35 FRC
12 MOD      36 X≠0?
13 -        37 GTO 00
14 1        38 RCL 01
15 -        39 LASTX
16 STO 01   40 -
17 RCL 00   41 "HAT TEI
18 4        LER "
19 /        42 ARCL X
20 1        43 *LBL 01
21 +        44 BEEP
22 STO 02   45 PROMPT
            46 END
    
```

Bild 2: Algorithmus F mit Listing



ter unten durchgeführt. Da dieser Wert immer grösser als  $\sqrt[3]{n}$  ist (wobei n der Restfaktor ist), kann n nur noch aus höchstens zwei Faktoren bestehen. Somit liefert Algorithmus F dann auch in jedem Falle Primfaktoren.

### Algorithmus E im Programm

Analysiert man Algorithmus E, so fällt auf, dass es überflüssig ist, weiterhin gerade Faktoren p zu testen, wenn die Teilbarkeit durch 2 nicht (mehr) gegeben ist. Damit wird die Anzahl der Tests praktisch halbiert; dementsprechend auch die Rechenzeit. Die analoge Ueberlegung gilt für den Testfaktor 3. Ist die zu faktorisierende Zahl nicht (mehr) durch 3 teilbar, so reicht es, den Testfaktor von 5 ausgehend abwechselnd um 2 oder 4 zu erhöhen. Damit werden alle Vielfachen von 2 und 3 übersprungen.

Eine Verminderung der Teilterests auf diese Weise lässt sich mit den Primfaktoren 5, 7, 11, ... fortführen. Jedoch zeigt die letzte Zeile der folgenden Tabelle, dass der zusätzliche Nutzen (Einsparung an überflüssigen Tests) immer geringer wird, während der Aufwand unverhältnismässig ansteigt (Bild 4).

Da für jeden Test mindestens ein Unterprogrammaufruf mit Eingangsparameterübergabe notwendig ist, wären bei f) mindestens  $480 \cdot 3 = 1440$  Bytes nur für die Schleife notwendig (dabei wurde der relativ langsame indirekte Unterprogrammaufruf zugrunde gelegt). Bei e) sind mindestens 144 Bytes in der Schleife not-

wendig, was durchaus realisierbar ist.

Für das folgende Programm wurde die Alternative e) zu d) untersucht, wobei allerdings in d) die Unterprogramme abgerollt wurden, d.h. nur im Erfolgsfall (Teiler gefunden) wird hier ein Unterprogramm aufgerufen. In diesem Fall ist der indirekte Unterprogrammaufruf nicht nur jeweils um ein Byte kürzer, sondern auch etwas schneller, (d.h. das Ueberspringen ist schneller).

Es zeigte sich, dass das Vorgehen nach d) die günstigsten Rechenzeiten liefert. (Ein Schleifendurchlauf Zeile 054 bis 112 dauert 1,160 sec).

### Algorithmus F im Programm

Während alle mir bekannten Realisierungen des Algorithmus E mindestens die Inkrementierung nach b) der obenstehenden Tabelle benutzen, ist es nicht ganz so offensichtlich, dass man im Algorithmus F die

Anzahl der Tests ebenfalls (sogar effektiver) drastisch reduzieren kann.

Der Algorithmus wurde von Fermat benutzt, und es ist nicht anzunehmen, dass er sich für ein Verfahren entschied, bei dem er wesentlich häufiger die Wurzel aus einer Zahl ziehen, als im anderen Falle eine Division durchführen musste.

Fermats Vorgehensweise bestand darin, dass er die beiden Endziffern des Radikanden  $x^2 - n$  betrachtete, um so viele Fälle auszuschliessen; er arbeitete modulo 100. Soll nämlich  $x^2 - n$  ein Quadrat sein, so müssen die Endziffern 00, a1, a4, 25, b6 oder a9 sein, wobei a gerade und b ungerade ist. 100 wurde sicher deshalb als Modul gewählt, weil man n mod 100 bei dezimal dargestelltem n direkt ablesen kann. Für die Anwendung mit einem Rechner ist das kein Vorteil mehr; daher kann man einen günstigeren Modul suchen.

Nach vielen Tests, die weitgehend mit Hilfsprogrammen durchgeführt wurden, haben wir uns für den Modul 144 entschieden. Eine grössere Zahl wurde nicht gewählt, um den Programmumfang in Grenzen zu halten.

Eine Tabelle, welche die Funktion des Moduls 144 bei der Inkrementierung von x im Algorithmus F veranschaulicht, wäre fast 1 m<sup>2</sup> gross, so dass sie hier aus naheliegenden Gründen nicht wiedergegeben werden kann ...

Stattdessen soll unsere Vorgehensweise mit dem Modul 9 verdeutlicht werden. Voraussetzung ist, dass n schon soweit reduziert wurde, dass 3 kein Teiler mehr ist. In der Tabelle (Bild 5) sind in der Kopfspalte alle in Frage kommenden Reste n mod 9 aufgeführt. Das sind  $\phi(9) = 6$  Stück, denn 0, 3 und 6 scheiden aus, da n nicht durch 3 teilbar ist. In der Kopfreihe sind die Werte x mod 9 aufgeführt; das sind natürlich 9 Stück. Die Eintragungen in der Tabelle sind je-

		X MOD 9								
X↑2 - N MOD 9	*	0	1	2	3	4	5	6	7	8
*****										
N MOD 9	*									
1	*	8	0	3	8	6	6	8	3	0
2	*	7	8	2	7	5	5	7	2	8
4	*	5	6	0	5	3	3	5	0	6
5	*	4	5	8	4	2	2	4	8	5
7	*	2	3	6	2	0	0	2	6	3
8	*	1	2	5	1	8	8	1	5	2
*****										

Bild 5: Reduktion der Teiler-Tests mit Modul 9



weils die Werte  $x^2-n \pmod 9$ , wobei diese Werte mit doppelter Zeichenbreite gedruckt wurden, falls sie ein Quadrat mod 9 darstellen.

Es ist im Algorithmus F also beispielsweise unsinnig, einen  $x$ -Wert zu testen, für den gilt:  $x \pmod 9 = 3$  und gleichzeitig  $n \pmod 9 = 4$ , denn der zu untersuchende Radikand  $x^2-n$  ist keine Quadratzahl, da er bei Teilung durch 9 den Rest 5 ergibt, und 5 ist kein Quadrat mod 9. Unter Berücksichtigung der Ergebnisse dieser Tabelle ergeben sich folgende Inkremente für  $x$ :

$n \pmod 9 = 2, 5$  oder  $8$ :

$x \leftarrow \lfloor \sqrt{n/9} \rfloor \cdot 9$  und dann jeweils um 3 erhöhen

$n \pmod 9 = 1$ :

$x \leftarrow \lfloor \sqrt{n/9} \rfloor \cdot 9 + 1$  und dann abwechselnd um 7 und 2 erhöhen

$n \pmod 9 = 4$ :

$x \leftarrow \lfloor \sqrt{n/9} \rfloor \cdot 9 + 2$  und dann abwechselnd um 5 und 4 erhöhen

$n \pmod 9 = 7$ :

$x \leftarrow \lfloor \sqrt{n/9} \rfloor \cdot 9 + 4$  und dann abwechselnd um 1 und 8 erhöhen

Bei dieser Inkrementierung werden also mit einem Test mindestens 3, meistens sogar 4,5  $x$ -Werte abgedeckt, im Gegensatz zu nur einem bei der Primitiv-Version.

Wie man sieht, sind jedoch leider die Inkremente abhängig von der Eingabe. Das ist beim Algorithmus E nicht der Fall. Die Berechnung der Inkremente erfolgt aber nur einmal und hat deshalb keine Rechenzeitverlängerung (im Programm, bei Modul 144, nur ca. zwei Sek.) zur Folge, sie verlängert das Programm jedoch erheblich; Zeilen 124 bis 221.

An dieser Stelle sei eine zusätzliche Schwierigkeit genannt, die Algorithmus F bereitet: Da im Ausdruck  $x^2-n$  der  $x^2$ -Wert bei grossem  $n$  grösser als  $10^{10}$  wird, ist keine korrekte Berechnung mehr möglich, wenn wie im obenstehenden Primitivprogramm inkrementiert wird. (Man versuche, damit 98947·101063 zu faktorisieren.) Diese Schwierigkeit lässt sich aber umgehen, indem man ausnutzt, dass die folgende Rekursion gilt:

$$(x+i)^2-n = (x^2-n) + (i^2+2ix)$$

Im Gegensatz zu Algorithmus E, bei dem ein erhöhter Aufwand einen immer geringeren zusätzlichen Nutzen bietet, führt beim Algorithmus F die Wahl eines geeigneten grösseren Moduls (mit mehr Primzahlpotenzen)

	E	F
1001423 = 887:1129	4 min 17 sec	4,8 sec
1001293 = 113:8861	33,5 sec	24 min 12 sec

Bild 3: Aufwand und Ertrag beim Optimieren

zu wesentlich besseren Ergebnissen. Nach Knuth ist beispielsweise bei Berücksichtigung der ersten 30 Primzahlen (mit dem Taschenrechner natürlich nicht realisierbar) nur noch einer von  $2^{30}$  Fällen zu testen.

Die Arbeitsweise von Programmen, die auch Zahlen  $> 10^{10}$  faktorisieren, stellen wir uns so vor, dass beim Uebergang zum Algorithmus F das Programm stoppt und dann nur noch anzeigt, welche von möglicherweise sehr vielen Datenkarten mit den richtigen Inkrementen eingelesen werden muss. Das Berechnen der Inkremente im Programm wäre nicht allzu zeitaufwendig, jedoch würde es den Programmumfang so stark aufblähen, dass für die eigentlichen Testroutinen, die dann ja wesentlich komplizierter sind, kaum Platz bliebe.

Bei unserem Beispiel, dem Modul 144 werden je nach Beschaffenheit des Restfaktors  $n$  entweder 72 ( $n \pmod 3 = 1$ ) oder 48 ( $n \pmod 3 = 2$ )  $x$ -Werte mit vier Tests überdeckt. Das bedeutet, dass bei  $n$  in der Grössenordnung  $10^{10}$  mit den ersten vier Tests 3720 bzw. 3050 Testfaktoren  $p$  erfasst werden. Dieser Bereich für  $p$  nimmt mit wachsendem  $x$  bis auf ca. 44 in beiden Fällen ab; bis dahin ist die Testmethode nach Algorithmus E schneller.

Zum Abschluss sei noch die Vorgehensweise erläutert, mit der die Uebergangsstellen von E nach F gefunden wurden: Die zeitintensiven

Schleifen im Programm sind die Programmsequenzen 054 bis 112 für Algorithmus E und 264 bis 326 für Algorithmus F. Die Schleife für E wird in  $\alpha = 1,160$  Sek. durchlaufen, während für die Schleife in F die Zeit  $\beta = 1,669$  Sek. benötigt wird. Bei der folgenden Konstantenberechnung geht nur das Verhältnis  $\beta/\alpha$  ein, so dass unterschiedliche Rechnerlaufzeiten kompensiert wären.

Insgesamt muss vom Programm jeder in Frage kommende Faktor  $p$  im Bereich 0 bis  $\sqrt{n}$  getestet werden, wobei das in folgender Reihenfolge geschieht: von 0 bis  $p_0$  mit Algorithmus E und von  $\sqrt{n}$  bis  $p_0$  mit Algorithmus F. Die folgende Funktion  $f(p,n,B)$  gibt die Differenz des Verhältnisses der von den beiden Algorithmen überstrichenen Testfaktorbereiche und dem Verhältnis der Rechenzeiten für diese Bereiche an. Dabei ist  $B$  der Bereich der  $x$ -Werte, der mit einem Schleifendurchlauf überdeckt wird (bei Modul 144 ist  $B = 72$  oder  $48$ ).

$$f(p,n,B) = (p(x)-p(x+B))/30 - \beta/\alpha$$

Im Algorithmus F gilt:

$$n = p \cdot q = (x-y)(x+y) = x^2-y^2$$

$$\text{d.h. } p(x) = x - \sqrt{x^2-n}$$

$$\text{bzw. } x(p) = (p^2+n)/2p$$

	Primitiv-	a)	b)	c)	d)	e)	f)	g)
	version							
	allgem.							
Testfaktoren enthalten keine Primteiler bis inkl.	p		2	3	5	7	11	13
Bereich, den eine Schleife überdeckt	$L = \prod_{q \text{ prim}} \frac{p}{q}$	1	2	6	30	210	2310	30030
Anzahl der Tests pro Schleife (Programmlänge)	$\varphi(L)$	1	1	2	8	48	480	5760
Bereich, der mit einem Test überdeckt wird	$L/\varphi(L)$	1	2	3	3,75	4,38	4,81	5,21
Gewinn gegenüber vorherigem Fall			100%	50%	25%	16,8%	9,8%	8,3%
Dabei ist $\varphi$ die Eulersche $\varphi$ -Funktion: $\varphi(n) =$ Anzahl der zu $n$ teilerfremden Zahlen kleiner $n$								

Bild 4: Rechenzeiten der beiden Primitiv-Algorithmen

*Listing*

```

01*LBL "PF"
SF 12 CF 25 CF 29
FIX 0 RND STO 00
STO 07 10 Y+X RCL 07
LN 1/X 2 STO 02 X+2
STO 04 6 STO 06 11
STO 08 12 STO 09 CLA
ASTO 11 ASTO 12 " "
ARCL 00 SF 21 SF 25
ADV PRA CF 21 AVIEW
"- " XEQ 00 CLX RCL 02
MOD X=0? XEQ IND 04
CLX 3 MOD X=0?
XEQ IND 04 CLX 5 MOD
X=0? XEQ IND 04 SIGN
SIGN

```

```

54*LBL 00
CLX RCL 06 LASTX +
MOD X=0? XEQ IND 04
CLX RCL 04 LASTX +
MOD X=0? XEQ IND 04
CLX RCL 02 LASTX +
MOD X=0? XEQ IND 04
CLX RCL 04 LASTX +
MOD X=0? XEQ IND 04
CLX RCL 02 LASTX +
MOD X=0? XEQ IND 04
CLX RCL 04 LASTX +
MOD X=0? XEQ IND 04
CLX RCL 06 LASTX +
MOD X=0? XEQ IND 04
CLX RCL 02 LASTX +
MOD X=0? XEQ IND 04
DSE 10 GTO 08 CLX
LASTX X+2 X>Y? GTO 97
ST+ Y CLX LASTX ST+ X
/ STO 10 CF 06 CF 07
RCL 07 3 MOD
GTO IND X

```

```

130*LBL 02
RCL 07 4 MOD 3 X=Y?
GTO 03 18 STO 01 6
SF 06 GTO 02

```

```

142*LBL 03
12 STO 01 SF 07
GTO 02

```

```

147*LBL 01
RCL 07 12 MOD ST+ X
RCL 07 9 MOD +
XEQ IND X GTO 02

```

```

158*LBL 03
16 STO 01 2 RTN

```

```

163*LBL 06
18 STO 01 14 RTN

```

```

168*LBL 09
8 STO 01 46 RTN

```

```

173*LBL 15
20 STO 01 16 RTN

```

```

178*LBL 18
4 STO 01 32 RTN

```

```

183*LBL 21
28 STO 01 8 RTN

```

```

188*LBL 02
STO 02 RCL 05 RCL 01
STO 03 ST+ X - RCL 02
- STO 04 RCL 07 16
MOD GTO IND X

```

```

202*LBL 03
203*LBL 11
CF 07 RCL 01 X<> 02
STO 01 RCL 03 X<> 04
STO 03

```

```

211*LBL 01
212*LBL 13
FS?C 06 GTO 11 RCL 02
X<> 04 STO 02

```

```

218*LBL 05
219*LBL 07
220*LBL 09
221*LBL 15
RCL 07 SORT RCL X
RCL 05 MOD - STO 06
X+2 RCL 07 - RCL 04
2 / FS? 07 CLX
XEQ 07 X=0? GTO 01
CLX RCL 01 XEQ 07
X=0? GTO 01 CLX
RCL 02 XEQ 07 X=0?
GTO 01 CLX RCL 03
XEQ 07 X=0? GTO 01
CLX RCL 10 RCL 06 -
RCL 05 / INT STO 10
ISG 10

```

```

264*LBL 12
CLX RCL 04 X+2 RCL 06
LASTX ST+ 06 ST+ X *
+ + ENTER+ SORT FRC
X=0? GTO 01 CLX
RCL 01 X+2 RCL 06
LASTX ST+ 06 ST+ X *
+ + ENTER+ SORT FRC
X=0? GTO 01 CLX
RCL 02 X+2 RCL 06
LASTX ST+ 06 ST+ X *
+ + ENTER+ SORT FRC
X=0? GTO 01 CLX
RCL 03 X+2 RCL 06
LASTX ST+ 06 ST+ X *

```

```

+ + ENTER+ SORT FRC
X=0? GTO 01 DSE 10
GTO 12
327*LBL 01
RCL 06 LASTX - FRC
X=0? XEQ 04

```

```

334*LBL 97
RCL 07 ENTER+ FRC
DSE Y XEQ 04 XEQ 03
BEEP

```

```

342*LBL B
CLX CLD RTN SF 12
SF 21 " " ARCL 00
AVIEW RCL 09 1 E3 /
11 +

```

```

356*LBL 14
CLA ARCL IND X ISG X
ARCL IND X AVIEW ISG X
GTO 14 GTO B

```

```

365*LBL 07
X+2 RCL 06 LASTX
ST+ 06 ST+ X * + +
ENTER+ X<0? RTN SORT
FRC RTN

```

```

380*LBL 04
STO 10
382*LBL 10
LASTX ST/ 07 SIGN
ST+ 10 RCL 07 LASTX
MOD X=0? GTO 10
RCL 00 LASTX ARCL X
X=Y? "IS PRIME" SIGN
RCL 10 X>Y? "++" X>Y?
ARCL X FS? 25 SF 21
FS? 00 AVIEW CF 21
ASTO 01 ASHF ASTO 03
CLA ARCL IND 08
ARCL IND 09 ARCL 01
ARCL 03 ASTO T ASHF
ASTO Z ASHF ASTO Y
CLA ASTO X X=Y?
GTO 13 XEQ 03 RCL 01
RCL 03 2 ENTER+
ST+ 08 ST+ 09

```

```

432*LBL 13
RDN RDN STO IND 09
X<>Y STO IND 08
ARCL IND 08 ARCL IND 09
FC? 00 AVIEW "*"

```

```

443*LBL 00
LASTX STO 10 RCL 07 3
MOD XEQ IND X RCL 07
SORT RCL 10 - * RND
X<> 10 ABS RTN

```

```

459*LBL 00
460*LBL 01
72 STO 05 RCL 07 ,016
RTN

```

```

466*LBL 02
48 STO 05 RCL 07
,01856 RTN

```

```

472*LBL 03
FC? 00 FC? 25 RTN CLA
SF 21 ARCL IND 08
ARCL IND 09 AVIEW CLA
CF 21 END

```

SIZE= 017 PRGM= 096

BEISPIELE :

953  
IS PRIME

1517  
=37\*41

1591  
=37\*43

2183  
=37\*59

2627  
=37\*71

3071  
=37\*83

461761  
=409\*1129

746269  
=661\*1129

999731  
=599\*1669

SF 00

5907942612

=2+2  
\*3+2  
\*11+2  
\*13  
\*17+2  
\*19+2

RUN  
5907942612

=2+2\*3+2  
\*11+2\*13  
\*17+2\*19+2

Eingesetzt ergibt das:

$$f(p,n,B) = (\sqrt{((n+p^2/2p+B)^2 - n)} + (p^2-n)/2p - B)1/30 - \beta/\alpha$$

Eine Nullstelle  $p_0$  dieser Funktion bei festem  $n$  ist der optimale Uebergangswert für diesen Wert  $n$ . Durch lineare Approximation dieser Nullstellen  $p_0$  in Abhängigkeit von  $\sqrt{n}$  ergibt sich:

für  $B = 72$  ( $n \bmod 3 = 1$ ):  
 $p_0 = 0,48 \cdot \sqrt{n}$

und für  $B = 48$  ( $n \bmod 3 = 2$ ):  
 $p_0 = 0,557 \cdot \sqrt{n}$

Im Programm werden diese Faktoren benutzt, um den Schleifenzähler für Algorithmus E zu bestimmen, d.h. die Werte sind noch durch 30 (= Periodenlänge in E) zu dividieren. Die Berechnung der Nullstellen von  $f$  erfolgte mit SOL des MATHE I-Moduls und die lineare Approximation mit dem Kurvenanpassungsprogramm der Standard-Programmsammlung.

Die längste Rechenzeit ergibt sich für  $p = 10^{10-71}$  (grösste Primzahl

$p < 10^{10}$  mit  $p \bmod 3 = 2$ ): ca. 46 Min. dagegen für  $10^{10-33}$  (grösste Primzahl unter  $10^{10}$ ): ca. 42 Min.

Ein «ungünstiger» Fall ist sicher 55697·179533 mit ebenfalls 46 Min., denn hier würde der kleinere Faktor mit Algorithmus E direkt gefunden, wenn nicht zu F übergegangen würde. 55681·179591 mit ca. 36 Min. liegt kurz vor diesem Uebergang. Diese «Ungereimtheit» liegt einfach daran, dass mit Algorithmus F von der Wurzel an abwärts getestet wird; dafür wird dann ja auch beispielsweise 99991·100003 in ca. 31 Min. gefunden. (Alle Rechenzeiten ohne angeschlossenen Drucker.) Diese Beispiele zeigen, dass mit Algorithmus F für 52 % des gesamten Testbereiches nur 26,3 % bzw. für 44,3 % nur 22,8 % der Gesamtrechenzeit benötigt wird.

Einige Anmerkungen zur Benutzung des Programms:

- SIZE  $\geq 107$  einstellen und Programm einlesen bzw. eintasten
- Programmstart: n XEQ «PF» (n wird zur nächsten Ganzzahl gerundet, liegt diese nicht im Bereich

$2 \leq n < 10^{10}$ , so erfolgt eine Fehlermeldung).

- Ist die Faktorisierung beendet, so ertönt BEEP und 0 wird angezeigt. Die Anzeige von n und der gefundenen Faktoren kann jederzeit durch R/S (oder durch XEQ «B» und anschliessendes R/S) wiederholt werden.

Ist der Drucker während der Faktorisierung zur Zeitersparnis nicht angeschlossen bzw. eingeschaltet, so kann nach dem Programmstopp und Einschaltung des Druckers wie bei der Anzeigewiederholung ein Ausdruck erreicht werden.

Die Anzeige bzw. der Ausdruck erfolgt grundsätzlich so, dass so viele Primzahlpotenzen wie möglich in einer Zeile untergebracht werden. Ist jedoch Flag 0 gesetzt, so werden bei der ersten Anzeige die Primzahlpotenzen einzeln angezeigt, bzw. ausgedruckt.

Nach Verwendung des Programms als Unterprogramm müssen die Sonderfunktionsflags 12, 21, 25 und 29 sowie das Anzeigeformat gegebenenfalls neu gesetzt werden. □

## Brillante «Apfel»-Farben!

Darauf haben Sie (und Ihre Farbgrafik) schon lange gewartet, die neue Supercolor Karte SCC 2000! Hier ist die Farbkarte, nach PAL-Norm für Apple II und alle kompatiblen Rechner (nicht IIe), welche wirklich keine Wünsche mehr offenlässt. Einige Merkmale der neuen Farbkarte SCC 2000:

- Bis zu 16 klare, leuchtende und richtige Farben
- Normgerechtes PAL-Signal für beste Wiedergabe
- Keine Streifen, keine «Buntsäume» und ähnliches
- HR-Grafik und 80-Zeichen Wiedergabe (RGB)
- Anschliessbar an jedem TV-Gerät oder Monitor
- Einfache Anpassung mittels Lötbrücken
- Automatische Umschaltung für TV/AV-Betrieb
- Schweizer Entwicklung, Produktion und Qualität

Die SCC 2000 gibt es in 2 Versionen: SCC 2000/RGB und SCC 2000/Video. Die RGB-Version zum Betrieb an hochwertigen Farbmonitoren und die Video-Version zum Betrieb an jedem Farbfernsehgerät. Mit der SCC 2000 kommt ein mehrseitiges Manual, mit allen notwendigen Angaben über Installation, Anpassung an verschiedene Monitoren/CTV, Inbetriebnahme.

SCC 2000/RGB SFR. 295.-  
 SCC 2000/Video SFR. 315.-

**Gleich bestellen oder Prospekt anfordern!**

### ELECTRONIX VERSAND

Postfach A-123, CH-8052 Zürich, Tel. 01/301 29 23  
 aus der BRD, 0041/13 01 29 23,  
 (24 Stunden per Anrufbeantworter!)

## \* \* IBM-PC \* \* PREIS-SENSATION!

**IBM-PC mit 10 MB** Harddisk  
 (bootfähig), 1x360 K Floppy, DOS 2.0  
 (betriebsfertig mit Garantie)

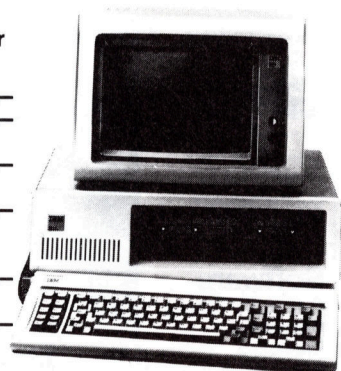
**Fr. 9990.-**

Einbausätze mit Controller  
 für IBM + Kompatible

- 10 MB Harddisk Fr. 3800.-
- 20 MB Harddisk Fr. 4950.-
- 2,6 MB Floppy 5¼" Fr. 2790.-
- 25 MB Back-up Tape mit Netzteil Fr. 3950.-
- 20 MB Harddisk und 20 MB Back-up Tape Fr. 8800.-
- 64 K RAM Erweiterung Fr. 220.-

Alle Geräte mit Garantie

IBM NCR VICTOR apricot EPSON



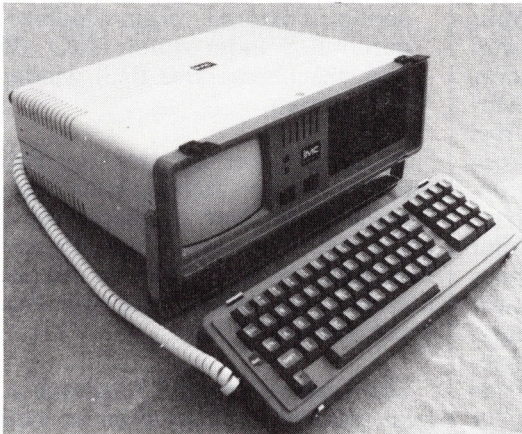
## Computer Team

MICRO COMPUTER SYSTEME  
 7500 St. Moritz, Hauptstrasse 41  
 Telefon 082/3 14 42

# IMC (Voll All®-kompatibel plus einiges mehr)

Generalvertretung: **beltronic, Im Chapf, 8455 Rüdlingen, Tel. 01/867 19 87** (Regionalvertreter in BE, ZH)

**Ein Portabler, der diesen Namen auch verdient!**



★ ★ ★  
**Weihnachts-**  
**Aktion** ★  
**5% auf alle Preise** ★

(Bei Bestellungen bis 20. 12. 84)



Z80+6502 Proc./5" Amber Schirm  
 2 Slim Line Floppy/64K RAM  
 1 Centronics + 1 RS 232 Interface  
 40/80-Zeichen-Taste/Freie Tastatur

**Fr. 3200.-**

Z80 + 6502 Proc./12" Amber Schirm/2 Slim Line Floppy  
 TEAC/Centronics Interface/40/80-Zeichen-Taste  
 (Softswitch)/64K RAM  
 Freie Tastatur mit Funktionstasten usw. **Fr. 3670.-**

Beispiele aus unserem Zubehör: 128K RAM-Karte Fr. 529.-/PAL-Karte Fr. 167.-/Grappler 32K Fr. 321.-  
 Disk Box (für 100 Disc) mit Schloss Fr. 48.- usw.

**Gerne senden wir Ihnen weitere Informationen** (Anfragen für weitere Regionalvertretungen erwünscht)



Bitte senden Sie mir Unterlagen über  Portable  Tischmodell 2001  Apple II® und IMC Zubehör

Meine Adresse: \_\_\_\_\_

ADCOMP  
 PC Vertretungen:

**EPSON**  
 Druckerserie

**COLUMBIA**

Personal Computer  
 16 bit, 8088 CPU,  
 128 - 1024 kB RAM,  
 IBM-PC kompatibel  
 Winchester Versionen

**Mountain**  
 Disk- und  
 Backup-Systeme

**GRID**  
 Hand Held Computer  
 16 bit, 8086 CPU,  
 256 kB RAM  
 Bubble Memory,  
 ROM Software

**Software**  
 Finanzbuchhaltung,  
 Debitoren,  
 Kreditoren,  
 Stammverwaltung,  
 Lohn,  
 Auftrag- und  
 Textverarbeitung

**VISICORP**



## Anschlussfreude

EPSON Drucker repräsentieren Anschlussfreude, Qualität und Auswahl. Sie wählen aus einer Palette von 10 Grundmodellen. Wählbar sind Schnittstellen, verschiedene Buffer, Charakter-ROMs, Tractors oder automatische Einzelblatt-Einzüge. Sie wählen Kompatibilität.

**DX 100** (Bild links oben)  
 Typenraddrucker für perfekte Korrespondenz, auch IBM- und DEC-kompatibel

**HI 80** (Bild links unten)  
 Plotter-Printer  
 230 mm/s, A4 und B5, benötigt keine speziellen Printer Escape-Funktionen, auch HP kompatibel

**JX 80**, hochqualitativer Farbmatrixdrucker mit gleichem Bedienungskomfort wie die FX-Serie, 7 Farben



Offizieller EPSON Importeur seit 6 Jahren

☆☆☆☆  
**TOP 5**

# DIE NEUE SUPER-SOFTWARE!

## Speziell für IBM und Kompatible Computer.

### TOP-5: Die neue Super-Software von EUCOTECH

Das sind die allgemeinen Merkmale dieser unglaublich preisgünstigen Software-Linie, die wir speziell für IBM und andere Kompatible Computer entwickelt haben:

- Voll editierbare Eingabemaschinen und Auswahlmenüs
- Speziell für MS-DOS entwickelt
- Teilweise mit Fenstertechnik
- Einsetzbar mit 2 Diskettenlaufwerken
- komfortable Bedienung, daher sofort einsetzbar
- ausführliches Handbuch (min. 100 Seiten) wird mitgeliefert

### TOP-5: FINANZBUCHHALTUNG

Unsere Finanzbuchhaltung nach Käfer setzt neue Maßstäbe. Füllen Sie einfach das Journal Seite für Seite aus. Die Gegenbuchungen und alle weiteren Berechnungen werden vom Programm selbst vorgenommen. Auch die Verarbeitungsgeschwindigkeit überzeugt.

Bestell-Nr.: SMS-T2

678.-

### TOP-5: FAKTURIERUNG

Für alle die auch das Schreiben von Rechnungen automatisieren wollen ist dieses Programm genau das Richtige.

Bestell-Nr.: SMS-T4

in Vorbereitung

### TOP-5: DIENSTPROGRAMME

Das ist ein Hilfsmittel für alle Programme der TOP-5 Serie. Es dient Ihnen z.B. zum Sortieren der Adressen oder zum Löschen von Buchungen und vieles mehr.

Bestell-Nr.: SMS-T5

225.-

### TOP-5: TEXTVERARBEITUNG

Unsere Textverarbeitung kann nicht nur Schreiben, Speichern und Drucken sondern auch RECHNEN. Natürlich können Sie dieses Programm auch mit der Adressverwaltung kombinieren, z.B. für Werbebriefe.

Bestell-Nr.: SMS-T3

478.-

### TOP-5: ADRESSVERWALTUNG

Diese Adressverwaltung verfügt über eine traumhafte Duplikate-Kontrolle. Auf Tastendruck erfahren Sie, ob die Adresse bereits einmal erfasst wurde. Ausserdem können Sie die Eingabemaske frei definieren und damit z.B. auch Ihre Videothek verwalten.

Bestell-Nr.: SMS-T1

487.-

**EUCOTECH**

**TOP!**

Einsenden an: Eucotech AG, Ahornweg 5, 8155 Niederhasli, Tel. 01-57 51 14

Ja, Ihr Angebot überzeugt mich, ich bestelle:

Stück Dokumentation / Händlernachweis  
 Stück Adressverwaltung  
 Stück Finanzbuchhaltung  
 Stück Textverarbeitung  
 Stück Fakturierung  
 Stück Dienstprogramme

**P.S. Diese Bestellung wird durch den Fachhändler ganz in Ihrer Nähe ausgeführt.**

Name: \_\_\_\_\_ Strasse: \_\_\_\_\_ Ort: \_\_\_\_\_ PLZ: \_\_\_\_\_

**LQ 1500**, der Leader der EPSON Parade mit Near-Letter-Quality, 9 Grafik Modi, 11 Zeichensätze, 24 Nadeln usw.

**FX 80**, der bewährte Drucker, 160 Zeichen in der Sekunde, 136 verschiedene

Schriftarten, serielle oder parallele Schnittstellen, 80 Zeichen pro Zeile, komprimiert bis 137

**EPSON**



**Neu RX 80 F/T Plus** mit Vertikal-Tabulator, 9 bit Image Mode für Zeichen-Generierung

**FX 100**  
136 Zeichen pro Zeile  
komprimiert 255

**ADCOMP**  
Advanced Computer-Systems  
-Components -Software -Education

Branch-Offices: 3015 Bern 1227 Carouge-Genève  
Tel. 031-44 11 11 Tél. 022-43 13 60

ADCOMP AG, 8953 Dietikon  
Tel. 01-741 41 11, Telex 58 657

**Qualität zu**

**vernünftigem Preis**

- voll IBM-PC-Kompatibel
- Anschluss an fast alle PC's
- Normal- und Korrespondenzschrift



## RADIX 10+15 LQ

Technische Daten:

- 200 CPS Normalschrift
- 50 CPS Korrespondenzschrift
- 136 oder 80 Kolonnen
- 16 KB Buffer reicht für 10 Seiten
- Autom. Blatteinzug Standard
- Stapelinzug Option
- Doppelinterface seriell + parallel

Radix 15 LQ:

**Fr. 2985.-**

Radix 10 LQ:

**Fr. 2515.-**

IBM-Versionen: plus Fr. 200.-

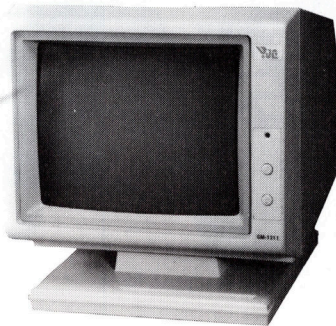


**die erfolgreiche Printerfamilie**

PECO AG • Personal Computer Products • 5000 Aarau • Telefon 064/22 63 63



## 12" Monochrome Monitor



- 20 MHz Video Bandbreite (18 MHz für PC-Typ)
- 15.75 kHz Scan Frequency
- Composite Video bzw. TTL-Eingangssignal
- grüne oder amber Bildröhre
- Drehfuss

**Fr. 298.-** inkl. WUST

**R. Schmid** / elektronische Bauteile und Geräte  
Tumigerstrasse 84, 8606 Greifensee, Tel. 01 / 41 16 33

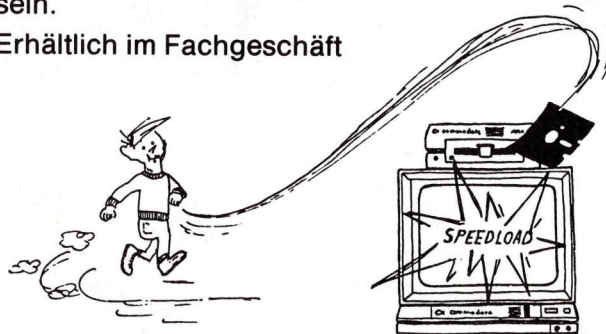
## Das Floppy 1541 – 4mal schneller!

Dank **SpeedLoad** laden Sie Ihre BASIC- oder Maschinensprach-Programme 4mal schneller als üblich.

**SpeedLoad** ist ein sog. Linkerprogramm, das selbständig die Laderoutine an Ihr Programm hängt und neu auf Diskette abspeichert.

**Speed-Load** wird ein unentbehrliches Hilfsprogramm für Ihre tägliche Arbeit mit dem Floppy sein.

Erhältlich im Fachgeschäft



**MICROTRON**  
COMPUTERPRODUKTE

Brunnenweg 5 Postfach 40 CH-2542 Pieterlen

Verlangen Sie nähere Informationen. Wir haben auch andere interessante Produkte für den C-64.

## Informatik-Schule New Data AG Olten

offeriert:

Programmier- und Anwenderkurse

- BASIC 1 + 2
- PASCAL 1 + 2
- ASSEMBLER 1 + 2
- INFORMATIK GEWERBE
- MULTIPLAN
- TEXTVERARBEITUNG 1 + 2

Tageskurse, Abendkurse, Kurse am Samstag

Max. Teilnehmerzahl pro Kurs: 6 Personen

Unterrichtscomputer:

COMMODORE VC 64,  
EPSON QX 10  
mit CP/M und MS-DOS

Verlangen Sie bitte das Kursprogramm für 1985 bei:

**NEW DATA AG**

Informatik-Schule +  
Softwareservice

Terrassenweg 20, 4600 Olten, Tel. 062 / 26 40 04

# Datum/Zeit-Information und Timer für jeden PC

**Welcher Programmierer stand nicht schon einmal vor dem Problem, über das Programm Zugriff zu einem Timer oder zu einer Uhr zu haben? Diese beiden Funktionen werden immer dann benötigt, wenn ein Programm zeitgesteuert ablaufen soll, wenn Zeitintervalle gemessen werden müssen oder wenn zu bestimmten Zeiten oder Kalenderdaten bestimmte Programme ablaufen sollen.**

Der Timer entspricht in seiner Funktion einer Stopuhr. Sein Anwendungsgebiet ist hauptsächlich die Erzeugung von Zeitverzögerungen, die Realisierung von zeitabhängigen Programmverzweigungen und die Aufsummierung von Zeitintervallen.

Als Beispiel sind etwa denkbar: Steuerung einer Modellbahnanlage, hier werden Zeitverzögerungen eingesetzt, die den Zugverkehr steuern; Spielprogramme erwarten oft innerhalb einer bestimmten Zeit eine Ein-

**Walter Schmid**

gabe oder lassen für einen Entscheid nur eine bestimmte Zeit zu; Spielzeiten bei einem Schachprogramm werden aufsummiert usw.

Eine Uhr wird dort gebraucht, wo im weitesten Sinne ein Protokoll erstellt wird. Denkbare Beispiele dazu wären etwa die Protokollierung bei einer Sportveranstaltung, die Erfassung von Wetterdaten zu bestimmten Tageszeiten, die Nachbildung einer Schaltuhr oder die Protokollaufnahme bei Beobachtungen für wissenschaftliche Zwecke.

Bei vielen PC's fehlt nun aber ein Timer. Eine Uhr, die neben der genauen Zeit auch das Datum liefert ist praktisch nie vorhanden. Der Grund ist hauptsächlich der, dass die Uhr nach jedem Betriebsunterbruch wieder neu gerichtet werden muss. Der Trend läuft nun zwar dahin, das auch Kleincomputer mit einer Uhr ausgerüstet werden. Siehe dazu den Artikel in M+K 84-3 über den APRI-COT. Dort wird eine Uhr mit einer Batterie ständig in Betrieb gehalten. Die Genauigkeit der Zeit ist abhängig von der Quarzgenauigkeit. Der Uebergang auf Sommerzeit und zurück muss aber auch da noch immer von Hand eingestellt werden.

Nachfolgend wird nun eine Schaltung vorgestellt, die es erlaubt, jeden PC mit den Funktionen «Uhr» und «Timer» auszurüsten. Die Datum/Zeit-Information wird selbständig aufbereitet und auf Anforderung des

Computers über eine serielle Schnittstelle übergeben. Durch die Verwendung der genormten seriellen Uebertragungsart ist die Uhr an jedem Computersystem einsetzbar. Zusätzlich kann noch auf zwei unabhängig voneinander arbeitende Timer zugegriffen werden. Um das mühsame Richten der Uhr nach einem Stromausfall oder einer gewollten Abschaltung zu umgehen, wird das Prinzip der «Funkuhr» angewendet. D.h., dass das Richten der Uhr nicht manuell erfolgt, sondern dass die dazu notwendige Information von einem Zeitzeichensender drahtlos empfangen wird und das Richten automatisch erfolgt.

## Zeitzeichensender

Diese werden für zwei Verwendungszwecke eingesetzt. Einerseits eignet sich die hochgenaue Trägerfrequenz als Eichnormal für Frequenzmesser, andererseits wird die aufmodulierte Information zum Richten oder Betreiben von Uhren verwendet. Im Gebiet der Schweiz sind vor allem zwei Sender von Interesse. Der eine, HBG, befindet sich in Prangins am Genfersee, er sendet auf der Langwelle 75 kHz, der andere, DCF77, steht in Mainflingen, in der Nähe von Darmstadt, er sendet auf 77.5 kHz.

Für die hier beschriebene Anwendung eignet sich nur der Sender DCF77, weil bei HBG die Datumsinformation (noch) fehlt. Beim DCF77 werden dem Träger Sekundenimpulse in der Form aufmoduliert, dass die Trägeramplitude jeweils für die Dauer von 0.1 oder 0.2 Sekunden auf 25% abgesenkt wird. Der Beginn der Absenkung markiert den Beginn der Sekunde. Der 59. Sekundenimpuls innerhalb jeder Minute wird unterdrückt um anzuzeigen, dass mit dem folgenden Impuls eine neue Minute beginnt. Die Uebertragung der Datum/Zeit-Information geschieht nun so, dass den unterschiedlich langen Sekundenimpulsen die binären Werte 0 und 1 zugeordnet werden. Ein

kurzer Impuls entspricht dem Binärwert 0, ein langer dem Wert 1. So ist es möglich, innerhalb jeder Minute ein Telegramm von 59 Bits Länge zu übermitteln. Dieses Telegramm beginnt mit der Sekunde 0 und enthält alle Informationen, die notwendig sind um eine Uhr mit Datumsanzeige zu richten. In Abb. 1 ist das Codierschema dargestellt. Die Impulse 0...14, 18 und 19 enthalten keine Information, sie haben den Wert 0.

Die Bits R, A, SZ, S, P1..P3 haben folgende Bedeutung:

- R der Sender arbeitet mit Ausweichantenne, von Bedeutung als Frequenznormal
- A Ankündigung Uebergang auf Sommerzeit und zurück, kaum von Bedeutung
- SZ Zeitangabe erfolgt in Sommerzeit, 1 = Sommerzeit
- S Beginn Datum/Zeit-Information, immer auf 1
- P1 Prüfbit für Minuten
- P2 Prüfbit für Stunden
- P3 Prüfbit für Datum und Wochentag

Die Prüfbits sind so gesetzt, dass die Parität (Anzahl der log. 1) im zu prüfenden Block gerade ist.

Es ist zu beachten, dass die Information im sogenannten BCD-Format (Binary Coded Decimal) und nicht im reinen Binärformat übertragen wird. Es wird während der Minute N jeweils diejenige Information übermittelt, die dann zu Beginn der Minute N+1 gültig ist.

Der Empfang von Zeitzeichensendern ist seit dem 1.1.84 nicht mehr konzessionspflichtig. Jedermann darf diese Sender ohne Formalitäten empfangen und deren Information auswerten.

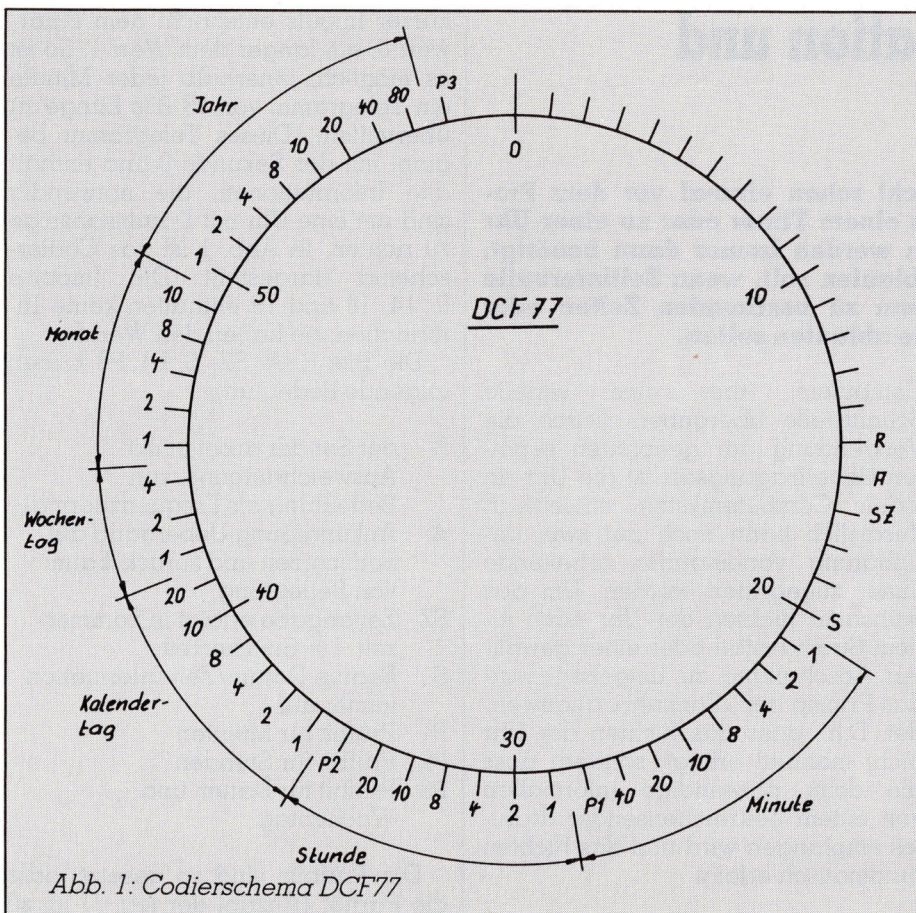
## Empfangseinheit für DCF77

Die Empfangseinheit besteht aus zwei Teilen, einer aktiven Ferritantenne und dem davon räumlich getrennten Empfänger.

Im folgenden soll die Funktion dieser beider Schaltungen kurz vorgestellt werden, ohne aber auf Details einzugehen. Auch ohne genaue Kenntnisse aller Schaltungsfunktionen ist es möglich die Schaltung sicher nachzubauen und in Betrieb zu nehmen.

### Aktive Ferritantenne

In Abb. 2 ist das Schema dieser Antenne dargestellt. Die Spule auf dem Ferritstab bildet zusammen mit



dem 390 pF-Kondensator einen Schwingkreis, der auf die Senderfrequenz von 77.5 kHz abgestimmt sein muss. Nur diese Frequenz wird dann ungeschwächt empfangen, alle übrigen Frequenzen werden mehr oder weniger abgeschwächt. Im nachfolgenden Verstärker wird das Empfangssignal verstärkt und niederohmig an den Ausgang gegeben. Die Speisespannung für den Verstärker wird auf der Signalleitung zugeführt, so dass für den Anschluss nur ein abgeschirmtes Kabel beliebiger Länge und beliebiger Typs benötigt wird.

## Empfänger

Ein Zeitzeichenempfänger zeichnet sich dadurch aus, dass er eine sehr geringe Bandbreite besitzt, in der Regel nicht mehr als einige 10 Hz. Um diese Eigenschaft zu erreichen, muss entweder ein Quarzfilter eingesetzt werden, oder es wird das Empfangsprinzip des Superhetempfängers angewendet. Das zweite Verfahren ist für einen sicheren und einfachen Nachbau besser geeignet und wird darum hier vorgestellt.

Beim Superhetprinzip, nach dem übrigens jeder Rundfunkempfänger arbeitet, wird das Empfangssignal mit Hilfe einer Oszillatorfrequenz auf

eine normalerweise tiefere Zwischenfrequenz umgesetzt. In unserem Fall liegt diese bei 700 Hz und ist damit so tief, dass mit einem Aktivfilter die gewünschte geringe Bandbreite leicht erreicht werden kann.

Aus Abb. 3 geht das Schema des Empfängers hervor. Am Anschluss 5 wird das Antennensignal abgenommen und über den Transistor BC 107 einem Schwingkreis zugeführt, wo eine weitere Vorselektion stattfindet. Es wird dann dem Empfänger-IC TCA 440 über die Pins 1 und 2 zugeführt. Am Pin 4 wird die für die Frequenzumsetzung benötigte Oszillatorfrequenz eingespeist. Diese wird aus dem Quarzoszillator abgeleitet, der auf 3.686400 MHz schwingt. Mit dem IC 4024 erfolgt eine Frequenzteilung durch 8, mit dem 4018 eine solche durch 6, sodass eine Frequenz von 76.8 kHz zur Verfügung steht.

Am Pin 16 des TCA 440 kann das auf 700 Hz (Differenz 77500-76800) umgesetzte Eingangssignal abgenommen werden. Es wird im nachfolgenden aktiven Bandpassfilter auf eine Bandbreite von 50 Hz begrenzt. Das Signal wird über Pin 12 wieder in den TCA 440 gegeben, wo es im regelbaren ZF-Verstärker weiter verstärkt und an den Pin 7 geführt wird. Im IC LM324/c erfolgt die Demodulation des Signals, so dass über dem

2.2 µF-Kondensator die interessierenden Sekundenimpulse anliegen. Aus der Spannung an diesem Punkt wird auch die Regelspannung für den TCA 440 abgeleitet, die an den Pins 3 und 9 eingespeist wird. Diese Regelung ist nötig, damit bei unterschiedlichen Empfangsbedingungen (Fading) immer eine konstante Spannung am Demodulator anliegt.

Der LM 324/d arbeitet als Komparator. Er bringt das Nutzsignal auf die logischen Pegel 0/5 Volt und sorgt für die notwendige Flankensteilheit. Nach dem Durchlaufen einer Inverterstufe steht das Zeitzeichensignal am IC 4069, Pin 8, für die Weiterverarbeitung zur Verfügung. Der Pegel an diesem Punkt ist 0, er springt jeweils für die Dauer des Sekundenimpulses auf 1.

## Auswertung des Zeitzeichensignals

Grundsätzlich sind verschiedene Wege der Auswertung denkbar. So etwa eine reine Hardwarelösung, bei der die 59 Bits des Datum/Zeit-Telegramms in ein Schieberegister eingelesen werden. Dieses wird dann in ein Anzeigeregister übertragen, sobald die Information vollständig eingelesen ist. Die Daten können dann entweder auf einem Display angezeigt oder von einem Computer abgefragt werden. Der Nachteil dieser Methode ist der, dass eine solche Uhr bei schlechten Empfangsverhältnissen, z.B. bei Gewittern, eine falsche Anzeige geben kann. Bei Gewittern in der Nähe des Senders wird dieser abgeschaltet, ebenso für periodische Kontrollen. Solche aussergewöhnlichen Bedingungen muss die Uhr erkennen und die Zeit dann selbständig weiterführen. Diese Funktion wird als Gangreserve bezeichnet.

Die Zeitzeichenimpulse lassen sich auch mit einem PC verarbeiten. Er übernimmt die Funktion des Schieberegisters und des Speicherregisters. Allerdings ist der PC dann für andere Programme blockiert. Für Demonstrationen oder Versuche ist

**C/NORM**  
Versand  
Melonenstrasse, 9001 St.Gallen

**ENORME HITPREISE**

**FÜR COMPUTER**

**071 22 63 22**



dieser Weg denkbar. Es wäre aber möglich, auf diesem Weg eine bereits vorhandene Uhr zu richten.

Bei der hier vorgestellten Lösung kommt ein Mikroprozessor zur Anwendung, dessen Programm alle notwendigen Funktionen übernimmt. Verwendet wird der MC68705P3 von Motorola. Bei diesem IC sind auf einem Chip ein 8 Bit Prozessor, ein RAM, ein EPROM, ein Timer und eine I/O-Schnittstelle aufgebaut. Das Programm führt folgende Funktionen aus:

- Nachbilden einer Uhr, Auflösung 1/10 Sekunde
- Nachbilden zweier 16-Bit-Timer, Auflösung 1/10 Sekunde
- Auswerten des Zeitzeichensignals
- Richten der Uhr
- Kommunikation mit einem PC über serielle Schnittstelle
- Testroutine für Prüfzwecke

### Programm

Als Hilfe bei der Entwicklung eines eigenen Auswerteprogramms ist in Abb. 4 das Flussdiagramm für die Impulsauswertung und Uhrfunktion dargestellt. Oben links wird geprüft, ob der Impulseingang eine positive Flanke und somit den Beginn eines Sekundenimpuls erhalten hat. Beim ersten detektierten Impuls wird ein Timer auf Null gesetzt. Dieser misst die Periodenzeit von einem Impuls zum nächsten. Dann wird die Minutenlücke von 2 Sekunden Länge gesucht. In diesem Moment beginnt dann die Sekunde 0 und es kann damit begonnen werden, das Datum/Zeit-Telegramm einzulesen. Bei jedem ankommenden Impuls wird geprüft, ob die Periodendauer gültig sei, also 1 Sekunde betrage. Eine Toleranz von etwa +/- 50 ms kann hier zugestanden werden. Wenn die Periodenzeit nicht stimmt, muss die Empfangsroutine neu aufgestartet werden. Falls nicht die Impulse der 58. oder 0. Sekunde ankommen, wird eine Zeitverzögerung «DELAY» mit 150 ms gestartet. Bei der 58. Sekunde ist das ganze Telegramm eingelesen und es kann eine Auswertung der in ein Schieberegister eingelesenen Daten erfolgen. Es wird ein Flag gesetzt, das vermerkt, dass ein Datum/Zeit-Telegramm fehlerfrei empfangen wurde. Bei der 0. Sekunde werden die ausgewerteten Daten in die Uhr übertragen. Die Uhr ist gerichtet, was ebenfalls mit einem Flag vermerkt wird. Sobald die Zeitverzögerung «DELAY» abgelaufen ist, wird der Pegel am Eingang gelesen. Bei

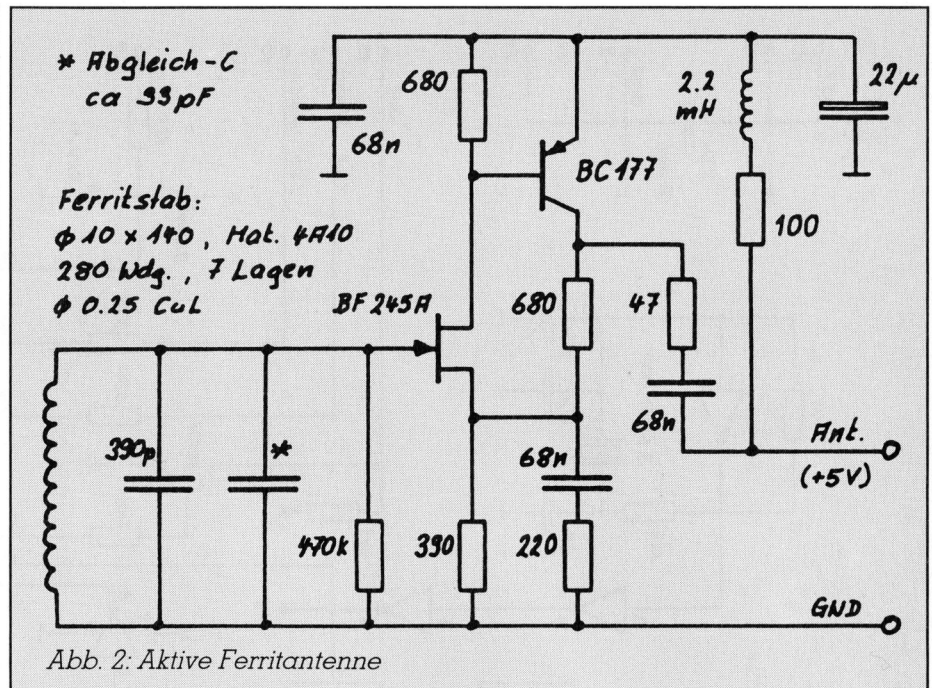


Abb. 2: Aktive Ferritantenne

einem langen Impuls wird man in diesem Moment log. 1, bei einem kurzen log. 0 messen. Das eingelesene Bit wird in ein 58 Bit langes Schieberegister gebracht, unabhängig davon, ob es sich um ein Daten- oder Prüfbit handelt, die Auswertung und Aufbereitung erfolgt dann in der 58. Sekunde. Sollte die Periodenzeit den Wert 2.05 Sekunden überschritten haben, so muss angenommen werden, dass der Empfänger keine Impulse mehr liefert, es wird dann ebenfalls neu aufgestartet.

Eine nicht gezeichnete Interruptroutine liefert alle 1/10 Sek. eine Marke, die zur Weiterschaltung der Uhr dient. Falls die Uhr noch nicht gerichtet ist, werden diese Marken ignoriert, sonst wird die Uhr weiterschaltet. Wenn die Sekunden auf 59.9 stehen, wird geprüft, ob in der soeben abgelaufenen Minute ein gültiges Telegramm empfangen wurde. Trifft das zu, so wird der Zeitimpuls ignoriert, die Weiterschaltung der Uhr erfolgt dann gleichzeitig mit dem Richten zu Beginn der neuen Minute. Andernfalls wird die Uhr weiterschaltet, sie läuft dann als normale Quarzuhr in der Betriebsart «Gangreserve» bis sie wieder durch gültige Daten neu gerichtet werden kann. Die Uhr wird also immer gerichtet, wenn ein gültiges Telegramm empfangen wurde, im Normalfall einmal pro Minute.

Noch ein Wort zur Genauigkeit. Die Genauigkeit von DCF77 kann als absolut bezeichnet werden, es gibt keine genauere Zeit. Hingegen wird das Empfangssignal durch diverse Filter im Empfänger verzögert. Der

am Empfängerausgang gelieferte Impuls ist gegenüber dem Original beim Sender um ca. 50 ms verzögert.

### Programmfunktionen

Über den seriellen Eingang, Anschluss 28 können dem Prozessor im RS232-Format (8-Bit-Wort, 2 Stop-Schritte, kein Parity-Bit, 1200 Baud) folgende Befehle erteilt werden:

- Befehl 0 sende Bits 13..20 des Empfangstelegramms
- Befehl 1 sende Statusregister
- Befehl 2 sende Sommerzeit (1 = Sommerzeit, 0 = Normalzeit)
- Befehl 3 sende Zeit in MEZ (Stunden, Minuten, Sekunden, 1/10-Sekunden)
- Befehl 4 sende Zeit in GMT (Stunden, Minuten, Sekunden, 1/10-Sekunden)
- Befehl 5 sende Datum (Kalendertag, Monat, Jahr)
- Befehl 6 sende Datum und Zeit MEZ (7 Bytes)
- Befehl 7 sende Wochentag (1 = Montag, 2 = Dienstag usw.)
- Befehl 8 reset Timer 1
- Befehl 9 start Timer 1
- Befehl 10 stop Timer 1



- Befehl 11  
  sende Timer 1 (2 Bytes, MSB, LSB)
- Befehl 12  
  reset Timer 2
- Befehl 13  
  start Timer 2
- Befehl 14  
  stop Timer 2
- Befehl 15  
  sende Timer 2 (2 Bytes, MSB, LSB)

Die verlangten Daten können am Anschluss 19 ebenfalls im RS232-Format, mit gleichem Format und in derselben Geschwindigkeit wie bei den Befehlen, entgegengenommen werden. Die Nummer des Befehls ist gleichzeitig der Inhalt des Befehls. Wird z.B. vom Computer binär 3 (nicht ASCII-3, dezimal 51!) gesendet, so werden 4 Bytes in der Reihenfolge Stunden, Minuten, Sekunden, Zehntelsekunden zurückgesendet. Diese Werte sind in Binärform codiert und nicht im BCD-Format wie vom Sender empfangen!

Die beiden Timer haben einen Bereich von 16 Bits, d.h., dass sie bis 65535 zählen können, entsprechend 6553.5 Sekunden oder etwa 110 Minuten.

Die genaue Reihenfolge und der Aufbau der vom Prozessor gesendeten Bytes ist dem Datenblatt des Prozessors zu entnehmen.

Der Statusbyte kann die folgenden Werte enthalten:

- 0 Uhr nicht gerichtet
- 1 Uhr gerichtet, läuft auf Gangreserve
- 3 Uhr gerichtet, DCF77 wird ungestört empfangen

Mit dem Befehl 0 kann der Teil des Empfangstelegramms gelesen werden, der die Bits R, A, SZ und S enthält.

## Zusammenbau

Dieser sollte keine Probleme bieten. Auf folgende Punkte sei aber speziell hingewiesen:

Die Wicklung der Ferritantenne wird am besten auf einen Spulenkörper aufgebracht, der dann auf den Stab geschoben wird. Dieser kann direkt auf einer kleinen Printplatte montiert werden, auf der sich auch die Komponenten des Verstärkers befinden. Die Montage des Ferritstabes darf nicht mit Metallteilen erfolgen, nur Kunststoff verwenden. Wird die Schaltung in ein Gehäuse eingebaut, so muss dieses ebenfalls aus Kunststoff bestehen.

Der Empfänger kann auf derselben Printplatte wie der Prozessor

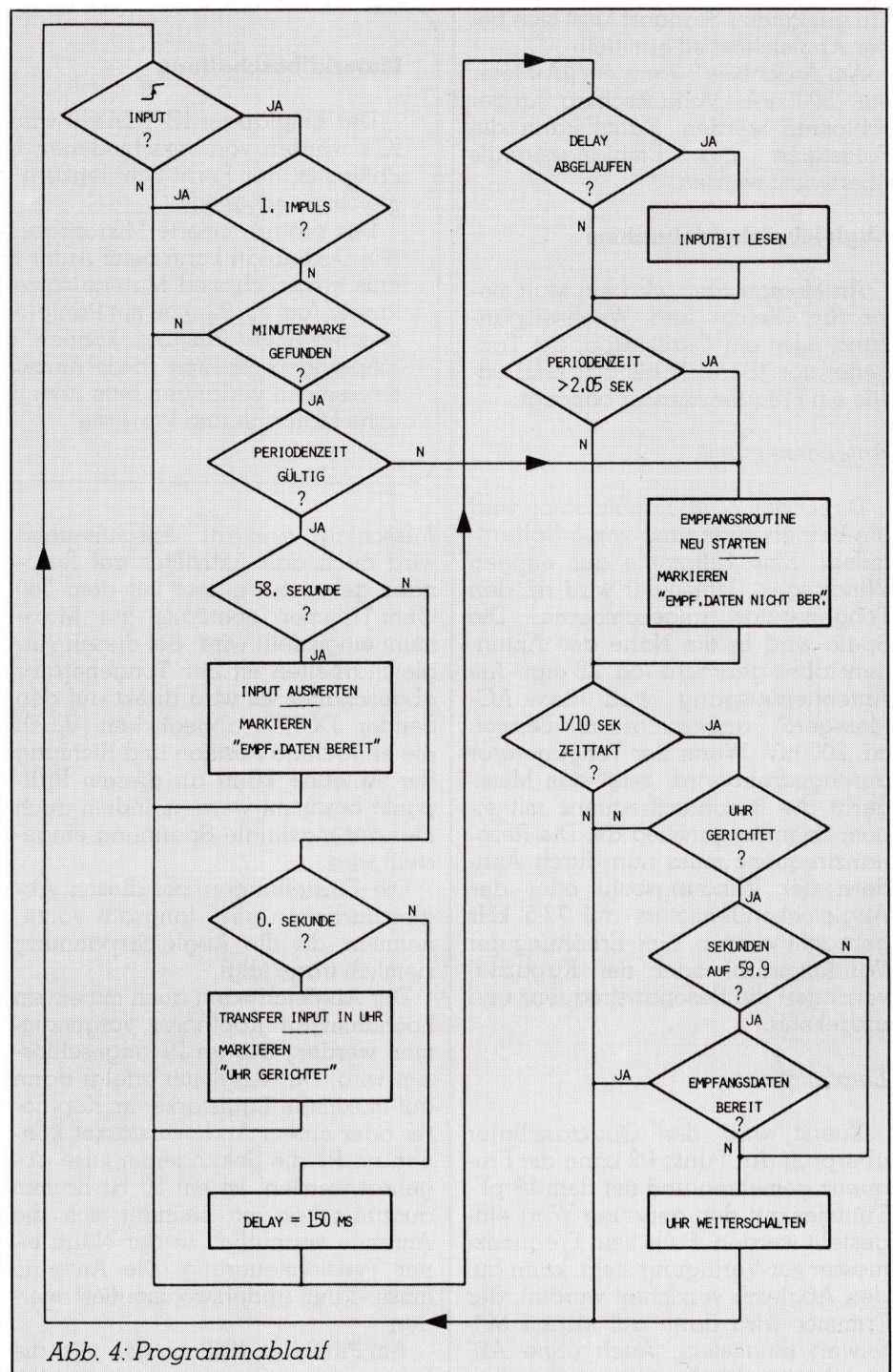


Abb. 4: Programmablauf

montiert werden. Es ist aber strikt darauf zu achten, dass die Stromversorgungsleitungen für den Empfangsteil mit den IC's TCA440 und LM324 separat geführt werden von denjenigen des Logikteils. Nur am Tantal-Elko 33 µF erfolgt der Zusammenschluss. Der 10 nF-Kondensator am Eingangskreis und die beiden 18 nF-Kondensatoren im Aktivfilter sollten nicht mehr als 2% Toleranz aufweisen, die Abgleicharbeit wird dadurch erleichtert. Besonders gut eignen sich Kondensatoren vom Typ «Micropoco» von Philips. Die Spule am Eingang wird mit einem Ferrit-schalenkern mit Abgleichstift reali-

siert, genaue Angaben sind dem Schema zu entnehmen.

Bei den Widerständen können Kohle- oder Metallfilmtypen verwendet werden. Eine Toleranz von 5% ist ausreichend. Bei den Kondensatoren können Keramik- oder Folientypen eingesetzt werden, die Toleranz sollte nicht über 10% liegen.

Es ist zu empfehlen, die ganze Schaltung in ein Metallgehäuse einzubauen, die Störstrahlung des Prozessors und des Quarzoszillators wird dann zurückgehalten, was besonders im Amateurfunkbetrieb wichtig ist. Die Ferritantenne braucht nicht im Freien montiert zu werden.

Ein geeigneter Standort lässt sich bei der Abgleicharbeit ermitteln.

Am Anschluss 3 kann ein  $\mu\text{A}$ -Meter mit 500  $\mu\text{A}$  Vollausschlag angeschlossen werden. Damit kann die Feldstärke des Empfangssignals überwacht werden.

## Abgleich, Inbetriebnahme

An Messgeräten wird ein Multimeter für Gleich- und Wechselspannung oder ein Oszilloskop, ein Tongenerator (Bereich bis 100 kHz) sowie ein Frequenzmesser benötigt.

### Antenneneinheit

Direkt am 22  $\mu\text{F}$ -Kondensator wird die Betriebsspannung von 5 Volt angelegt. Eine Hilfsspule aus einigen Windungen (Luftspule) wird an den Tongenerator angeschlossen. Die Spule wird in die Nähe des Antennenstabes gebracht (ca. 20 cm). Am Antennenausgang wird das AC-Messgerät angeschlossen, Bereich ca. 200 mV. Wenn der Tongenerator durchgedreht wird, zeigt das Messgerät die Resonanzfrequenz mit einem Spannungsanstieg an. Die Resonanzfrequenz muss nun durch Ändern der Windungszahl oder des Abgleichkondensators auf 77.5 kHz gebracht werden. Eine Erhöhung der Windungszahl oder der Kapazität verringert die Resonanzfrequenz und umgekehrt.

### Empfänger

Zuerst wird der Quarzoszillator überprüft. An Punkt P2 kann die Frequenz gemessen und mit dem 18 pF-Trimmer auf den genauen Wert eingestellt werden. Falls kein Frequenzmesser zur Verfügung steht, kann auf den Abgleich verzichtet werden, der Trimmer wird dann auf seinen Mittelwert eingestellt. Auch ohne Abgleich wird die Frequenzgenauigkeit innerhalb der zulässigen Toleranz liegen. Am Pin 1 des 4018 muss eine Frequenz von 76.8 kHz vorhanden sein, dann arbeitet auch der Frequenzteiler richtig. Die abgegliche Antenne wird an günstiger Empfangslage (Fensterhöhe oder dort wo guter MW-Empfang möglich ist) in Ost-West Richtung montiert und an den Empfänger angeschlossen. Die Verbindung kann über ein beliebiges abgeschirmtes Kabel erfolgen. An den Anschluss 3 wird ein DC-Messgerät gelegt (Messbereich ca. 500 mV). Der Abgleichkern im Spulenkörper wird nun soweit eingedreht, bis das Messgerät maximalen

## Materialbeschaffung

Der Empfänger-IC TCA440 wird von Philips hergestellt, die übrigen IC's werden von verschiedenen Herstellern angeboten. Ebenfalls von Philips ist der Ferrit-Schalenkern P14/8 mit Montagematerial und Abgleichstift erhältlich.

Der programmierte Mikroprozessor MC68705P3/DCF77 mit detailliertem Datenblatt kann beim Autor bezogen werden (ca. Fr. 110.-). Ebenfalls erhältlich sind Materialsätze für Antenne (ca. Fr. 28.-) und Empfänger (ca. Fr. 92.-), je mit Printplatten und Bestückungszeichnungen. In beschränktem Umfang können auch abgegliche Antennen- und Empfängereinheiten sowie einzelne Printplatten bezogen werden. Interessenten verlangen bitte über die Redaktion beim Autor die ausführliche Material- und Preisliste.

Ausschlag anzeigt. Anschliessend wird auch das Aktivfilter auf Resonanz gebracht, indem mit dem 500 Ohm-Trimmer ebenfalls auf Maximum eingestellt wird. Bei diesen Abgleicharbeiten ist der Tongenerator abgeschaltet, es wird direkt auf den Sender DCF77 abgeglichen. Auch die endgültige Position und Richtung der Antenne kann an diesem Prüfpunkt bestimmt werden, indem auch hier auf maximale Spannung eingestellt wird.

Die Einstellungen bei diesen Abgleicharbeiten sind langsam vorzunehmen, da die Abgleichspannung ziemlich träge folgt.

Der Abgleich kann auch mit einem hochohmigen Kopfhörer vorgenommen werden, der an P1 angeschlossen wird. Der Abgleich erfolgt dann auf maximale Lautstärke. Im Kopfhörer oder einem Audioverstärker können an P1 die Sekundenimpulse abgehört werden. Ist ein 50 Hz-Brumm auszumachen, so befindet sich die Antenne vermutlich in der Nähe einer Thyristorsteuerung. Die Antenne muss dann anderswo montiert werden.

Am Pin 8 des 4069 müssen jetzt die Sekundenimpulse gemessen werden können, z.B. mit einem Logiktester oder mit einer LED und Vorwiderstand.

An Orten mit extrem schlechten Empfangsverhältnissen ist es möglich, dass DCF77 nur sehr schwach hereinkommt und die Sekundenimpulse nicht detektiert werden. In diesem Fall kann der Emitterwiderstand des BC107 im Eingangskreis von 8.2 auf 2.7 kOhm reduziert werden. Die Vorverstärkung nimmt dann um den Faktor 3 zu. Diese Massnahme sollte aber nur dann ergriffen werden, wenn eindeutig feststeht, dass die Signalstärke von DCF77 sehr schwach ist.

### Mikroprozessor

Der Prozessor wird auf einen Sockel montiert. Prüfen, ob richtig eingesetzt! Die Brücke B1 wird eingelötet, B2 und B3 haben keine Bedeutung. An den Anschlüssen 15 und 17 wird je eine LED angeschlossen, Kathode an diese Anschlüsse, Anode direkt an 5 Volt.

Die Speisespannung wird jetzt eingeschaltet. Die Impuls-LED muss im Takt der ankommenden Impulse aufleuchten. Unmittelbar auf die zweite Minutenlücke muss die Status-LED aufleuchten (Status 3). Die Uhr ist jetzt gerichtet. Wird nun die Antennenzuleitung unterbrochen, ohne aber die Speisespannung abzuschalten, so geht die Status-LED in Blinken über (Status 1). Die Uhr läuft auf Gangreserve.

Wird die Speisespannung bei offener Brücke B1 angelegt, so wird am seriellen Ausgang für Prüfzwecke ein endloser ASCII-String ausgegeben. An P3 können für Prüfzwecke 10 ms-Impulse mit einer Frequenz von 10 Hz abgenommen werden.

Die Stromaufnahme der Empfangseinheit mit Prozessor, Antenne und LED's liegt bei maximal 150 mA.

## Zusammenfassung

Der Zeitzeichensender DCF77 strahlt rund um die Uhr die Datum/Zeit-Information aus. Mit relativ geringem Aufwand lässt sich eine Schaltung bauen, die diese Signale empfängt und decodiert. Ueber eine serielle Schnittstelle können die aufbereiteten Daten von jedem PC abgerufen werden. Die Decodierlogik, die aus einem Single Chip Prozessor besteht, verfügt zusätzlich noch über eine Quarzuhr als «Gangreserve» sowie über zwei Timer. □

## Telefon Management System mit IBM-PC

**Die Telefonistin ist die Visitenkarte der Unternehmung. Nur - Freundlichkeit und guter Wille sind nicht immer ausreichend, um Anrufer schnell und ohne lange Rückfragen zu verbinden. Ferienabwesenheiten und Neueintritte, Bürowechsel und Umzüge, ergeben bei den Telefonistinnen immer wieder neue Probleme und Unsicherheiten. Das muss nicht sein, denn unnötige Telefonkosten und lange Wartezeiten sowie stets gut und schnell bediente Anrufer sind keine Hexerei mit dem DCT Telefon Management System DCT-TMS.**

Zur Hilfe stehen modernste IBM Personal Computer, die mit einem Netzwerk verbunden sind. Das gesamte Telefonverzeichnis ist auf einem Festplattenlaufwerk im Hauptgerät gespeichert. Jeder weitere angeschlossene Arbeitsplatz hat ebenfalls Zugriff auf diese Datei.

Das DCT-TMS ermöglicht das rasche Auffinden nach Betriebszweigen, Abteilungen, Mitarbeitern, Kurzzeichen der Mitarbeiter und nach Telefonnummern. Die Bedienerin des Hauptgerätes hat jederzeit die Möglichkeit Neuzugänge zu erfassen, Mutationen oder Löschungen vorzunehmen. Und selbstverständlich kann mit einem Schönschreiber das ganze Telefonverzeichnis, d.h. eine Namens- oder eine Nummernliste gedruckt werden.

Ein DCT-TMS mit vier Arbeitsplätzen steht seit mehreren Monaten in der Telefonzentrale der Kantonalen Verwaltung Luzern im praktischen Einsatz. Info: DCT Dialog Computer Treuhand AG, Seeburgstrasse 18, 6002 Luzern. □

## PC-Integration mit «Sidekick»

**Immer neue Produkte für den IBM-PC drängen auf den Markt, welche die Integration des PC's als Arbeitsplatz-Computer weiter vorantreibt. Eines dieser neuen Produkte heisst «Sidekick» und**

**ist als elektronischer Schreibtisch zu verstehen.**

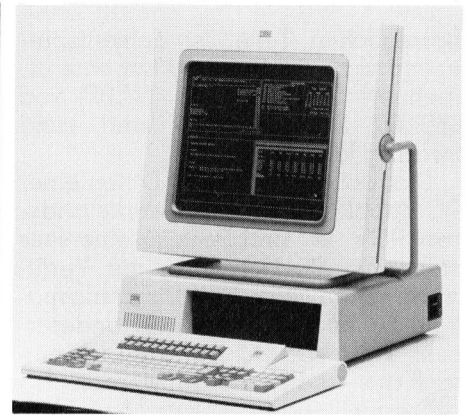
Sidekick besteht aus fünf Modulen, die per Window-Switch aufgerufen und bedient werden können, inkl. Datenübergabe an das gerade unterbrochene Programm. «Sidekick» kann immer parallel zu anderen Programmen permanent im Speicher sein. «Sidekick» besteht aus: PC-Taschenrechner inkl. Speicher sowie hexadezimalen und binären Rechenfunktionen; Notizbuch mit Datum-Funktion und Disk-Speicherung; Kalender für Termine etc. inkl. Printerfunktion; ASCII-Tabelle; Telefonverzeichnis, inkl. automatischer Telefonnummern-Wahl. Info: CS Data-Disc Lassahn + Co., Zeppelinstrasse 18, 8042 Zürich. □

## Grosser Plasmabildschirm für den IBM 3270 Personal Computer

**Die IBM Schweiz hat für den IBM 3270 Personal Computer einen Plasmabildschirm mit grosser Kapazität angekündigt, ebenso eine Erweiterung des Kontrollprogramms, das die Unterteilung aller 3270 PC-Bildschirme in zehn sogenannte «Fenster» gestattet, die voneinander unabhängig und zu verschiedenen Zwecken zur Verfügung stehen.**

Der neue Plasmabildschirm ist bemerkenswert kompakt und leicht gebaut, braucht er doch sehr wenig Platz auf einem Arbeitstisch. Da er mit der sogenannten Plasmatechnik (Mischung von Neon und Argon) gebaut ist, kann er 9'920 alphanumerische Zeichen anzeigen und dies in gestochener Schärfe, da die bei den herkömmlichen Kathodenstrahlbildschirmen erforderliche Wölbung an der Oberfläche wegfällt. Die neuen Modelle des 3270 Personal Computer sind mit anderen Terminals des Systems IBM 3270 kompatibel, sei es für die Verarbeitung von Daten oder von Grafiken.

Version 2 des Kontrollprogramms des IBM 3270 Personal Computer ge-



stattet die Unterteilung aller Bildschirme in bis zehn voneinander unabhängige «Fenster». Vier davon können für Anwendungen zur Verfügung gestellt werden, deren Programme sich in einem oder mehreren andern Computern befinden, zwei nehmen Notizen auf und ein bis zehn Fenster (in Abhängigkeit von der verbliebenen Verfügbarkeit) stehen für lokale PC-Anwendungen zur Verfügung. Info: IBM Schweiz, General Guisan-Quai 26, 8002 Zürich. □

## Bubble Disk für IBM-PC/XT

**Die GEI Systeme AG erweitert ihre Angebotspalette für den PC-Markt um ein innovatives und qualitativ hochwertiges Produkt in Bubblespeicher-Technologie: die PC Bubble Disk der Helix Laboratories.**

Die 512 KB-Bubble-Disk benötigt einen Einschub im IBM-PC/XT/Portable und macht konventionelle Diskettenlaufwerke auch dort überflüssig, wo raue Umweltbedingungen herrschen: Staub, Schmutz, grosse Temperaturschwankungen, sowie Erschütterungen (bis 200G) hindern nicht mehr den Einsatz von PC's ausserhalb des Bürobereiches.

Die PC Bubble Disk emuliert bei einer mittleren Zugriffszeit von 48ms eine Mini-Winchester und arbeitet völlig geräuschlos. Ein leicht zugänglicher Write-Protect-Schalter ermöglicht, dass die gespeicherten Daten von Ueberschreiben oder Löschen geschützt werden.

Durch den Einbau weiterer drei Platinen beträgt die maximale Bubble-Speicherkapazität 2 Megabyte pro PC, wobei nur eine Disk Drive simuliert wird.

## RUND UM DEN IBM-PC

Die PC Bubble Disk enthält keine beweglichen Teile, ist dementsprechend wartungsfrei, und hat eine errechnete Betriebsdauer (MTBF) von 180.000 Stunden, das sind rund zwanzig Jahre.

Das Überspielen von Daten einer PC Bubble Disk auf Laufwerke anderer PC's ist problemlos: einerseits stehen lokale Netzwerke zur Verfügung und diverse Übertragungspakete auf asynchroner Basis, andererseits kann die PC Bubble Disk einfach aus- und eingebaut werden.

Die vielfältigen Einsatzmöglichkeiten sind vor allem durch einen hohen Anspruch an Zuverlässigkeit des Massenspeichers gekennzeichnet. Info: GEI Systeme AG, Täfernstrasse 29, 5405 Baden-Dättwil. □

## Grafik-System für den IBM-PC

**Die Genigraphics AG, Zürich, stellte kürzlich ihr Grafik-System der Serie 1000 vor, das speziell für den Einsatz mit dem IBM-PC/XT oder kompatiblen Rechnern konzipiert wurde. Mit Hilfe des Programmes und einer Erweiterungsplatine wird aus dem PC-XT ein leistungsstarkes Grafiksystem. Mit einem hochauflösenden Farbmonitor kann die Anlage mehr als 4000 unterschiedliche Farbtöne darstellen. Professionelle Strich-, Balken- oder Kreisdiagramme mit einer Vielzahl von Schrifttypen lassen sich ohne Mühe zeitsparend herstellen.**

Die Serie 1000 besteht aus einem Grafiksoftware-Paket, einer Erweiterungsplatine mit Zubehör sowie einer elektronischen Zeichenplatte. Über eine Kommunikationsschnittstelle lassen sich alle Darstellungen auf ein leistungsfähiges Reproduktionsgerät übertragen. Mit ihm können gestochen scharfe Dias und Aufsichtsvorlagen mit einer Auflösung von 2000x2000 Punkten hergestellt werden.

Die erforderlichen Rechnerkonfiguration ist ein IBM-PC/XT mit 320 KB RAM, Monochrom- oder Farbmonitor, einem Floppylaufwerk mit 320 KB und einer 10 MB Festplatte sowie einer seriellen Schnittstelle. Info: Genigraphics AG, Thurgauerstrasse 40, 8050 Zürich. □

## Professionelle Grafik auf dem IBM-PC

**Der Quintar Graphic Controller füllt die Lücken zwischen der niedrig auflösenden IBM-Grafik und extrem hochauflösender Grafikdisplays. Quintar bringt eine neue Klasse von Grafik-Controllern, die komplett BUS-unabhängig sind und zusammen mit dem PC ein leistungsfähiges Display System mit einer Auflösung von 832x630 Punkten mit jeweils sechzehn Farben aus einer Palette von 4'096 Farben ermöglicht.**

Diese Eigenschaften machen den Quintar ideal für den Einsatz im Personal Computer CAD-Bereich oder Businessgrafik, wobei mit der guten Auflösung direkt Hardcopies auf Kamerasystem oder Inkjetprinter ausgegeben werden können.

Der Controller ist ausgerüstet mit einer Z-80B (8 MHz) CPU und dem NEC 7220 Display Controller Chip. Der Befehlssatz für die Ansteuerung umfasst die Emulation verschiedener Industriestandards, wobei schon verschiedene Grafiksoftware-Hersteller Driver für die Ausnutzung aller Eigenschaften anbieten. Info: Computer Graphix AG, Giessereistrasse 1, 8620 Wetzikon. □

## Systemfamilie für Bürokommunikation mit IBM-PC, mittleren und grossen IBM-Systemen

**Die IBM Schweiz hat eine Systemfamilie für Bürokommunikation angekündigt, die erlaubt, Texte zu verarbeiten sowie Kalender und Verteilungslisten zu führen. Diese Programme arbeiten auf dem IBM-PC, dem IBM-System/36 und den Systemen mit /370-Architektur. Ebenso mög-**

**lich ist die elektronische Dokumentenverteilung unter diesen Systemen.**

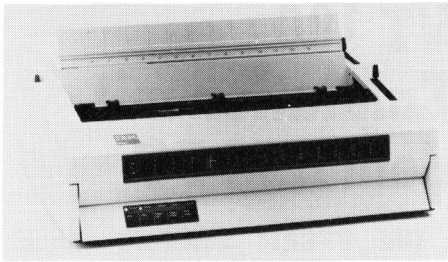
Diese Systemfamilie hat den grossen Vorteil, auf allen beteiligten Computersystemen mit ähnlichen Funktionen und einheitlichen Bildschirmformaten zu arbeiten. Die Programme dienen dem Bürobetrieb im weitesten Sinne, also auch Managern, Fachspezialisten und selbstverständlich Sekretärinnen und Sachbearbeitern. Ein Teil dieser Programme ist in englischer Sprache.

Mit der Ankündigung hat die IBM eine Absichtserklärung gegeben, nach der Verbesserungen bei den Verbindungsfähigkeiten zwischen den verschiedenen Systemen für Bürokommunikation und dem DISOSS (Distributed Office Support System/370) geplant sind. Info: IBM Schweiz, General Guisan-Quai 26, 8002 Zürich. □

## Lautloser Drucker für den IBM-PC

**Die IBM Schweiz hat neue Drucker für Korrespondenzqualität und für die IBM Personal Computer-Familie angekündigt sowie weitere Modelle für die IBM-Systeme im mittleren und grossen Leistungsbereich. Bei den ersten handelt es sich um den «Quietwriter», einen Drucker mit Thermodrucktechnologie, und den «Wheel Printer» Typenraddrucker, bei den letzteren Modellen um die Matrixdrucker IBM 4214 Modell 1 und 2.**

Die neue Thermodrucktechnologie ist erstmals am 16. Oktober im Zusammenhang mit der neuen Thermoerschreibmaschine IBM 6750 vorgestellt worden. Dieses Druckverfahren arbeitet ohne mechanischen Anschlag, und das Papier wird dabei nicht berührt. Da die Farbe mittels elektronisch gesteuertem Wärmeprozess auf das Papier gebracht wird, ist der «Quietwriter» der leiseste Drucker aller IBM-PC-Drucker. Er druckt mit einer Geschwindigkeit von 40-60 Zeichen/Sekunde und Proportionalschrift mit Schrittschaltungen von 10, 12, 15 Zeichen. Es sind mehrere Zeichensätze zu 252 Schriftzei-



chen lieferbar, von denen jeweils zwei gleichzeitig aktiv sind. Dies ermöglicht es dem Benutzer, zwei verschiedene Schriftbilder im gleichen Dokument verwenden zu können, ohne den Druckvorgang unterbrechen zu müssen. Der «Quietwriter» ist mit allen IBM-PCs kompatibel, welche die deutsche oder französische Schweizer Normtastatur verwenden.

Der Typenraddrucker «Wheel Printer» hat eine Geschwindigkeit von 25 Zeichen/Sekunde und verfügt über Schrittschaltungen von 10, 12, 15 Zeichen und eine grosse Vielfalt von Proportionalschrift-Typenrädern mit 96 Schriftzeichen. Die Typenräder, welche mit jenen der kürzlich angekündigten Schreibmaschinen IBM 6746 und 6747 identisch sind, können auch bei zahlreichen anderen IBM Druckern verwendet werden.

Der Matrixdrucker IBM 4214 kann sowohl für Datenverarbeitung als auch Korrespondenz eingesetzt werden, je nachdem beträgt seine Geschwindigkeit bis zu 200 Zeichen/Sekunde. Er bietet eine grosse Auswahl an horizontalen und vertikalen Schrittschaltungen und kann maximal 220 Zeichen pro Zeile drucken. Ausserdem verfügt er sowohl über Vorrichtungen für die Verarbeitung von Endlos- als auch Einzelblatt-Formularen. Das Modell 1 kann an die Kontrolleinheit des IBM 3270 Bildschirmsystems, das Modell 2 direkt an das IBM System/36 bzw. /38 oder auch an die entfernt eingesetzte Kontrolleinheit IBM 5294 angeschlossen werden. Info: IBM Schweiz, General Guisan-Quai 26, 8002 Zürich. □

## PC-Tastatur-Schablone

**Von Praktikern für Praktiker wurde die PC-Tastatur-Schablone entwickelt. Sie erleichtert die Arbeit beim Eingeben von Befeh-**

**len erheblich. Zusammen mit der Tastatur liegen die Funktionen jetzt klar und übersichtlich vor Ihren Augen. Die Schablone hilft Ihnen, die richtigen Tastenanschläge für das entsprechende Programm schnell zu finden, bzw. in Ihr Gedächtnis zurückzurufen. Das spart Zeit und reduziert das umständliche Hantieren mit Handbüchern auf ein Minimum.**

Die PC-Tastatur-Schablone ist für alle PC-Anwender geeignet. Für Anfänger und Fortgeschrittene genau so wie auch für Schulungsleiter und Selbstlerner. Die Schablone ist beidseitig bedruckt und aus widerstandsfähigem Kunststoff gefertigt.

Zur Zeit liegen PC-Schablonen mit folgenden Programmen für IBM- und COMPAQ-PC vor: dBase II, DOS/BASIC 2.00, LOTUS 1-2-3, WordStar, «DO IT YOURSELF» (eine Blanko-Schablone für eigene Programme, die mit einem wasserlöslichen Stift beschriftet werden kann, welcher mitgeliefert wird) und WORD- und OPEN ACCESS. Die Schablonen für die Programme von MULTIPLAN und dBASE III sind in Vorbereitung. Info: micom Micro Computer Systeme AG, Zugerstrasse 64, 8810 Horgen. □

## Corvus Bank für IBM-PC

**20 IBM-XT-Disk oder mehr als 1'250 IBM-PC-DDS-Disketten auf einmal zur Verfügung - das haben Sie mit The Bank von Corvus Systems. Ein 200 MB Back-up Bandspeicher mit Random Access erlaubt das Abspeichern und Zurückladen von einzelnen Volumes (Teil einer Disk), wie auch das Abspeichern einer ganzen Disk (Back-up).**

The Bank eröffnet neue Möglichkeiten der Datenspeicherung, da die auswechselbaren Bandkassetten einen direkten Zugriff auf die Daten erlauben. Der Benutzer kann über das Netzwerk Omninet oder Omni-share auf die Bank zugreifen und seine Daten auf die Winchesterdisk laden, um sie von dort aus zu verarbeiten. Nach der Verarbeitung lassen sich die Daten auf einfache Weise wieder auf die Bank auslagern und die Disk steht für neue Aufgaben frei.

## Kombinationskarte mit Netzwerk zum IBM-PC

**Orchid Blossom ist eine neue Erweiterungskarte für den IBM-PC und kann als Kombinationskarte der 2. Generation bezeichnet werden.**

Die Basisplatine umfasst Speichererweiterungen von 64 bis 384 KByte RAM, eine serielle Schnittstelle (RS232), eine parallele Schnittstelle (IBM, Centronics) sowie eine Echtzeituhr. Die mitgelieferte Software besteht aus einer RAM-Disk, einer besonders leistungsfähigen Cache-Software (25% Beschleunigung bei Disk- und Diskettenzugriff) sowie einer Printerspooler-Software. Das Basisboard enthält ausserdem die COAX-Verbindungs-Schnittstelle für ein PCnet.

Das von Orchid Technology ebenfalls angebotene PCnet Tochterboard wird auf die Blossom-Karte aufgesteckt. Mit dieser Zusatzkarte wird aus Blossom eine Netzwerkkarte mit allen Funktionen des bekannten PCnet-Boards (sowie erstmals ein Softwarepaket zur Unterteilung der XT-Platte in mehrere Laufwerke).

Das PCnet Profikit enthält ein PCnet Blossom (Orchid Blossom plus Tochterkarte), ein PCnet-Board (Standard) sowie alle notwendigen Netzwerkteile wie Software, Handbuch, BNC-Teile und ein 6m-Kabel. Die Inbetriebnahme des Netzwerkes ist äusserst einfach und benötigt bei der ersten Installation ca. 1 Stunde.

PCnet Diskless ist ein Standard PCnet Board mit Boot-ROM. Diese Karte ermöglicht das «Booten» eines IBM-PC's ohne eigenes Laufwerk. Der Bootvorgang wird über das Netzwerk auf dem angeschlossenen Shared PC ausgeführt. Info: Computer 2000 AG, Lettenstrasse 3, 6343 Rotkreuz. □

The Bank hat eine Datentransfer-Rate von 1 MBit/Sek. Die mittlere Access-Zeit bei einer 100 MB-Kassette beträgt 4,8 Sek. Es sind 100 MB- und 200 MB-Kassetten erhältlich. Info: Cosendai Computer Products SA, En Budron C, 1052 Le Mont-sur-Lausanne. □



## STAR goes IBM

**STAR konnte in der ganzen Welt erstaunlich schnell einen anständigen Marktanteil im hartumkämpften PC-Druckergeschäft erringen. Mit breiter Front wird jetzt der IBM und IBM-kompatible Markt angegriffen.**

STAR steigt gleich mit sechs Matrixdruckermodellen in den Kampf, dazu kommt noch ein Typenrad-drucker, der dank Downloading für IBM wie auch andere PC's ohne spezielles Interface eingesetzt werden kann. Die sechs Typen setzen sich aus drei Serien zusammen, die jeweils schmal oder breit (A4 hoch oder quer) erhältlich sind.

Alle Drucker enthalten sämtliche IBM-Steuer-codes für Druck-Output, Grafik sowie Screen-Dump. Zusätzlich ist der normale Zeichensatz durch Schalter in «Griffnähe». Somit können die STAR-Drucker nicht nur an den IBM-PC, sondern auch an alle anderen MS-DOS oder kombinierte MS-DOS/CPM-Systeme angeschlossen werden. Info: Peco AG, Laurenzenvorstadt 119, 5000 Aarau. □

## DESQ = Fensterln mit dem PC

**Die DESQ Window Software verkörpert eine neue Generation von Window Technik für Personal Computer. Mit DESQ lassen sich alle bestehenden Anwendungspakete wie Multiplan, Lotus 1-2-3, WordStar, dBase II usw. in eine gemeinsame Hülle integrieren. Der Benutzer kann die Integration selbst ohne Softwareänderungen vornehmen.**

Jederzeit kann der Benutzer von einem zum anderen Window umschalten, wobei nur jeweils ein Pro-

gramm aktiv sein kann. Die Bedienung kann über Tastatur oder Maus erfolgen.

Wesentlich beim Arbeiten mit DESQ ist, dass jederzeit Datentransfer von einem zum anderen Programm durchgeführt werden kann, ähnlich wie bei integrierten Systemen.

Mit DESQ lässt sich ein integriertes System erzeugen, das optimal auf den Benutzer zugeschnitten werden kann. Voraussetzung ist ein PC mit Harddisk und mindestens 512 KB Memory. Info: Computer Graphix AG, Giessereistrasse 1, 8620 Wetzikon. □

## Multifunktionskarten für den IBM-Portable

**IDEAssociates Inc., einer der führenden Produzenten von PC-Add-On-Produkten für den IBM-PC, präsentiert zwei neue Multifunktionskarten für den IBM-Portable.**

IDEAminimax, entwickelt für den kurzen Slot des IBM-Portable PC erlaubt eine Speichererweiterung bis 384 KB RAM-Speicher in einer speziellen «Doppeldeckerversion». Mit 128 KB auf der Basiskarte und zusätzlichen 256 KB auf einer Huckepackkarte benötigt diese Neuentwicklung von IDEA nur einen kurzen Slot im IBM-Portable PC oder natürlich auch im XT.

IDEAmini, diese schon bekannte Multifunktionskarte erfuhr eine Anpassung auf die neuen Platzverhältnisse im IBM-Portable PC. Somit können maximal zwei serielle, und eine parallele Schnittstelle sowie eine batteriegepufferte Uhr auf einer einzigen Karte realisiert werden.

Beide Karten, die IDEAminimax, wie die IDEAmini, werden mit einer Auswahl an Software angeboten: IDEAMenü ermöglicht dem Benutzer eine Boot-Diskette zu erstellen, ohne System-Konfigurationsfiles editieren zu müssen; RAMFloppy erlaubt dem Benutzer einen Teil des RAM-Speichers wie eine Floppydisk zu verwenden. Dadurch wird der Datentransfer zur RAMFloppy bis zu 50 mal schneller als zu einer physikalischen

## VT 100 Emulation mit dem IBM-PC

**Die VTERM Software macht aus dem IBM-PC ein hundertprozentiges DEC VT 100 kompatibles Terminal. Die Emulation des Terminals umfasst die komplette Tastatur inkl. Funktionstasten, alle Videoattribute wahlweise monochrom oder farbig. Die verschiedenen Editoren die EDT, TECO, WORDM oder EFACS lassen sich problemlos einsetzen, anstelle der LED Anzeige steht als 25. Zeile eine Statusanzeige zur Verfügung.**

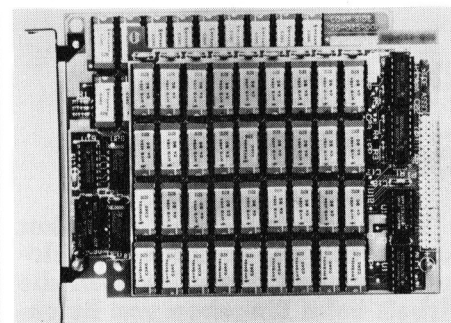
Ueber den SET-UP Mode lassen sich alle Parameter setzen, die in einem Konfigurations-File abgelegt werden. Printer Support ist ebenso gewährleistet wie 132-Zeichen/Zeile über ein spezielles Displayboard.

Wesentlich ist natürlich der Filetransfer zwischen DEC-Rechner und PC mit dem sowohl ASCII als auch Binär-Files übertragen werden können, wobei zusätzlich eine Hostsoftware benötigt wird.

VTERM bleibt memoryresident, der Benutzer kann jederzeit durch Tastendruck in PC-Mode und retour umschalten. Info: Computer Graphix AG, Giessereistrasse 1, 8620 Wetzikon. □

Floppy-Disk; Selektion paralleler Drucker, dieses Programm erlaubt ein einfaches Umschalten auf verschiedene parallele Drucker; Print Spooler; Diagnostics; Clock routine.

IDEA offeriert eine integrierte Linie an Kommunikations-, Grafik-, Massenspeicher- und Back-up-Produkten für den IBM-PC, XT und Portable. Info: W. Stolz AG, Täferstrasse 15, 5405 Baden-Dättwil. □





# Künstliche Intelligenz in BASIC

**Ueber Künstliche Intelligenz, im folgenden KI genannt, wird viel geredet und philosophiert. Die einen versuchen, die Grenzen der KI zu erforschen und mögliche neue Einsatzgebiete aufzuzeigen. Andere, vielfach nicht weniger vernünftige Leute, weisen auf die Gefahren hin, die mit dem Einsatz «intelligenter» Maschinen verbunden sind. Selten jedoch werden grundlegende Methoden der KI behandelt und noch seltener werden diese in einer Form behandelt, die auch vom KI-Laien nachvollzogen werden kann.**

Anhand eines den meisten Lesern wohl bekannten Problems sollen hier einige grundlegende Techniken der KI aufgezeigt und gleich auch in ein Programm umgesetzt werden. Das Programm ist absichtlich in einem Minimal-BASIC geschrieben, damit es auf den üblichen Personal Computern ohne Aenderungen lauffähig ist.

Was die KI an diesem Beispiel ausmacht, ist nicht so sehr das Pro-

## Urs Bürge

gramm an sich und der damit dargestellte Algorithmus, sondern die Art, wie von einem Problem ausgehend das Programm entsteht.

Als Beispiel dient uns folgendes Problem: Ein Bauer steht mit einem Wolf, einer Ziege und einem Kohl vor einem Fluss und möchte diesen überqueren. Fragen Sie mich nicht, was der Bauer mit dem Wolf vorhat, schliesslich ist das ja seine Sache. Für das Uebersetzen steht ihm ein Boot zur Verfügung, in welches er allerdings nur sich selber und einen Gegenstand auf einmal laden kann. Er muss nun zusätzlich darauf achten, dass er niemals den Wolf mit der Ziege oder die Ziege mit dem Kohl allein an einem Ufer lässt, da sonst

die ersteren die letzteren fressen - entsprechend ihrem Naturell.

Sie werden nun sofort Ihre grauen Zellen bemühen und in verhältnismässig kurzer Zeit eine gültige Lösung finden. Das zeigt eben, dass auch Sie intelligent sind.

Wir möchten nun eine Maschine, in diesem Fall die BASIC-Maschine, dazu bringen, die Lösung selber zu finden. Dazu müssen wir einerseits die für die Problemlösung relevanten Elemente der *realen Welt* in der Maschine irgendwie darstellen können und andererseits die Maschine mit einer geeigneten *Strategie* ausstatten.

Die entscheidenden Merkmale der *Problemwelt* sind die drei Gegenstände und der Ort, wo sich jedes dieser Objekte zu einem bestimmten Zeitpunkt befindet, diesseits oder jenseits des Flusses. Diese drei Informationen zusammen wollen wir einen «*Zustand*» nennen. In BASIC lässt sich das leicht in einer eindimensionalen Matrix  $Z()$  mit drei Elementen darstellen, indem wir festlegen, dass das erste Feld immer den Ort des Wolfes anzeigt, das zweite Feld den Ort der Ziege und das letzte demzufolge den Ort des Kohls. Zudem wird diesseits des Ufers mit -1 dargestellt und jenseits des Ufers mit 1. Der *Ausgangszustand* wird also wie folgt dargestellt: -1 : -1 : -1 :

Mit diesen Angaben können alle in unserer Problemwelt möglichen Zustände dargestellt werden. In der gleichen Art können wir jetzt auch den Zielzustand darstellen, nämlich : 1 : 1 : 1 :. Wenn dieser Zustand erreicht ist, ist das Problem gelöst.

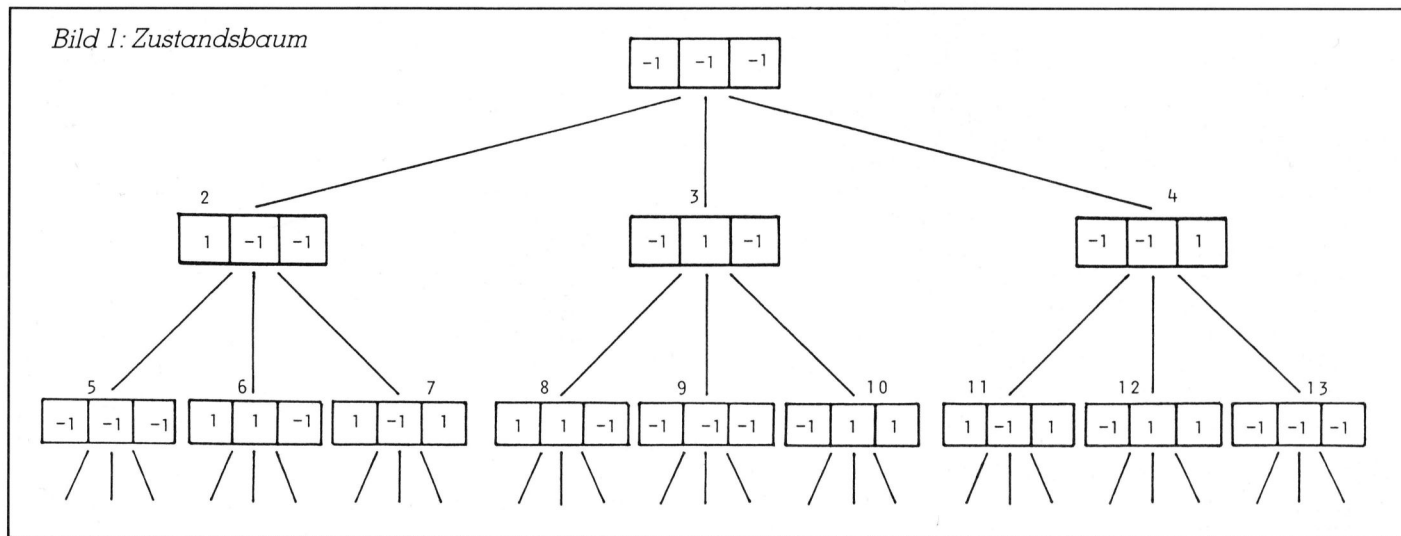
Von irgendeinem Zustand ausgehend wollen wir uns jetzt einmal überlegen, welche Möglichkeiten der Bauer hat, um seinem Zielzustand näherzukommen. Er hat oberflächlich gesehen jedesmal drei Möglichkeiten: Er kann entweder den Wolf, die Ziege oder den Kohl von einem Ufer zum andern bringen. Eventuell muss er dazwischen noch mit dem Boot allein zum andern Ufer hinüberwechseln, was wir jedoch für die Lösung beiseitelassen können, da sich das implizit aus der Angabe, welches Objekt er verschieben soll, ergibt.

Wenn wir jetzt einmal ausgehend vom Ausgangszustand alle Möglichkeiten darstellen, die der Bauer jeweils hat, so gibt das den *Zustandsbaum*, wie er in Bild 1 ansatzweise dargestellt ist.

Von einem Zustand zum nächsten kommen wir, indem wir eine *Operation* ausführen, nämlich eine der drei Zahlen invertieren. In BASIC: LET  $Z(X) = -Z(X)$ . Das Bestimmen aller möglichen Zustände, die man von einem Zustand aus mit einer Operation erreichen kann, nennen wir «*einen Zustand expandieren*».

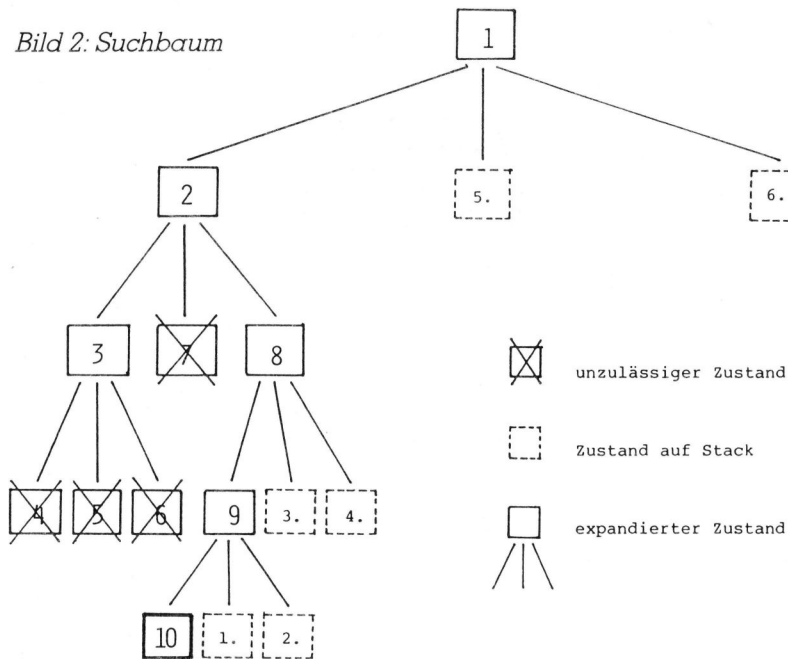
Das Programm müsste jetzt eigentlich nur diesen Baum durchsuchen, bis es einen Weg vom Ausgangszustand zum Zielzustand findet; dieser Weg wäre dann bei gewissen Verschiebungen ein *unzulässiger Zustand*, der eine Fressituation ergibt; so z.B., wenn der Kohl verschoben wird, und sich der Wolf und die Ziege am selben Ufer befinden.

Bild 1: Zustandsbaum



# RUND UM DEN IBM-PC

Bild 2: Suchbaum



stellt: 1 : -1 : -1 : 1 ;, der Zustand 10 entsprechend : -1 : 1 : 1 : 3 :. Für jeden Zustand kann nun rückblickend getestet werden, ob dabei eine Fressituation eintrat; in diesem Fall ist es ein unzuverlässiger Zustand. In unserem Problem gibt es zwei Gruppen von unzulässigen Zuständen:

1. Wolf und Ziege am gleichen Ufer, während Kohl verschoben wurde.
2. Ziege und Kohl am gleichen Ufer, während Wolf verschoben wurde.

Entsprechend testen wir in BASIC:

1. IF Z(1)=Z(2) AND Z(4)=3 THEN ...
2. IF Z(2)=Z(3) AND Z(4)=1 THEN ...

Fassen wir einmal kurz zusammen, wo wir bei der Umsetzung unseres Problems in ein Programm stehen:

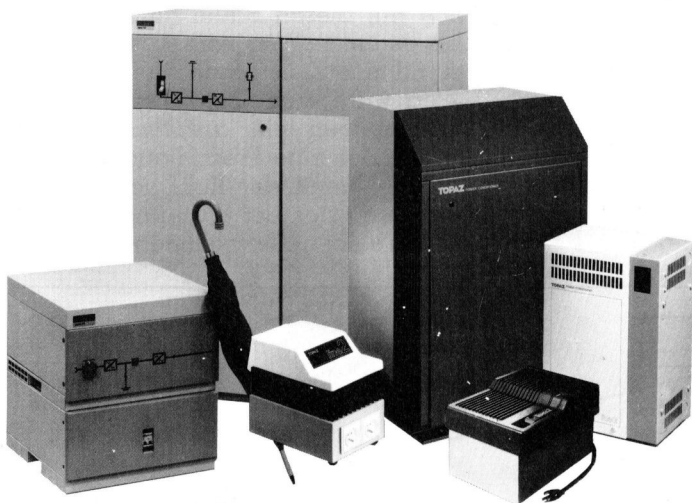
- Wir können jeden möglichen Zustand in BASIC darstellen.
- Wir kennen den Ausgangszustand.
- Wir kennen die drei möglichen Operationen, die von irgendeinem Zustand ausgehend zu einem nächsten möglichen Zustand führen.
- Wir kennen die Bedingungen, die auf einen Zustand angewandt

Um diese Bedingung vom Programm testen lassen zu können, fügen wir für die Darstellung eines Zustandes ein weiteres, viertes Feld hinzu. Dieses vierte Feld gibt an,

welcher der drei Gegenstände verschoben wurde, um diesen Zustand zu erreichen, der Wolf(1), die Ziege(2) oder der Kohl(3). Der Zustand 2 in Bild 1 wird also neu so darge-

## Bockt Ihre Elektronik?

- Computer
- Textverarbeitungssysteme
- Registrierkassen
- Überwachungsanlagen



**Störpulse aus dem Netz und Spannungsschwankungen sind meistens die Ursache, wenn elektronische Geräte ausflippen.**

Netzfilter, Netzstabilisatoren, Notstromgeräte und -anlagen sind die Lösung! Unser Verkaufsprogramm umfasst vom

kleinsten 100-VA- bis zum 100-KVA-Gerät oder mehr.

Profitieren Sie von unserer langjährigen Erfahrung auf dem Stromversorgungsbereich, denn ein elektronisches Gerät an der unaufbereiteten Netzspannung betreiben, ist das gleiche Risiko, wie die Einnahme einer Mahlzeit mit unkontrollierten Pilzen: Es kann gut gehen – oder auch nicht!

Für unsere heutigen Lösungen bürgen wir auch morgen!

**Dr. K. Witmer Elektronik AG**

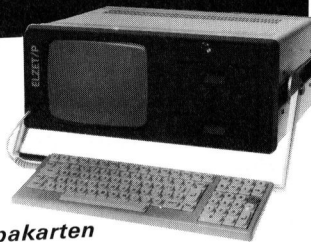
Seestrasse 141  
CH-8703 Erlenbach/ZH  
Tel. 01/915 35 61

Bureau de vente Suisse Romande  
Rue Martigny 3, 1005 Lausanne  
Tel. 021/22 85 37

## ELZET 80

das universelle Mikrocomputer-System

- CP/M 2.2-Betriebssystem
- 2x 800 KByte Minifloppy
- 128 K RAM, Z 80 A-CPU
- 80x25 Video-Display oder Vollgrafik 512x256
- RS-232; IEC-Bus
- inkl. deutsche Tastatur
- erweiterbar mit ELZET-Europakarten



Generalvertretung Schweiz:

**Bernhard-Elektronik**  
CH-5734 Reinach Aarauerstr. 20 Tel. 064/71 69 44



- Memotech MTX-512, der Professionelle Computer, mit DIN-Tastatur, für hohe Ansprüche.
- Memotech FDx, das CP/M 2.2 System für höchste Ansprüche im Büroalltag. CH-Tastatur, NewWord Textverarbeitungsprogramm und SuperCalc Tabellenkalkulierungsprogramm sind im Lieferumfang enthalten.

Generalvertretung für die Schweiz:

**URWA ELECTRONIC**

Computer Hard- und Software  
Lindenweg 24, 2503 Biel

werden können und uns sagen, ob es ein zulässiger Zustand ist, bzw. ob er auf zulässige Weise erreicht wurde.

- Wir kennen den Zielzustand.

Was uns jetzt noch fehlt, ist ein Suchalgorithmus, der uns in einem Zustands-Diagramm einen gültigen Weg vom Ausgangszustand zum Zielzustand findet.

Zuerst ein wenig Terminologie. Den in einem bestimmten Augenblick vom Programm schon durchsuchten Teil des Zustands-Baumes bezeichnen wir als *Suchbaum*. Der Suchbaum - wie auch andere Bäume - kann in *Ebenen* unterteilt werden. Der Ausgangszustand befindet sich in Ebene 0, die direkt vom Ausgangszustand aus erreichbaren Zustände in Ebene 1 und alle direkt von diesen aus erreichbaren Zustände in Ebene 2 usw.

Es gibt nun verschiedene Strategien, wie solche Zustands-Bäume durchsucht werden können. Die beiden einfachsten Strategien werden «Suchen in die Breite» und «Suchen in die Tiefe» genannt.

Beim «Suchen in die Breite» werden die Zustände in der Reihenfolge untersucht, wie sie in Bild 1 nummeriert sind. Jede Ebene wird vollständig durchsucht, bevor mit der nächsten Ebene begonnen wird. Der Suchbaum wächst also, mit Ausnahme der unzulässigen Zustände, schön symmetrisch an.

Der Vorteil der «Suche in die Breite» besteht darin, dass auf jeden Fall der kürzeste Weg zum Zielzustand gefunden wird. Andererseits, wenn viele Wege zum Zielzustand führen, aber alle erst über viele Operationen erreichbar sind, ist die Suche ineffizient.

Für jeden Fall ist das «Suchen in die Tiefe» effizienter. Bei dieser Suche wird jede Möglichkeit, in eine tiefere Ebene vorzustossen gleich ausgenutzt. Die auf diesem Weg zusätzlich errechneten Zustände werden vorläufig alle einmal beiseite gelassen; auf sie wird erst zurückgegriffen, wenn es sonst nicht mehr weitergeht. Sie sehen einen Suchbaum nach 10 Operationen für diese Suche in Bild 2.

Als *Datenstruktur* für die Steuerung dieser Strategie eignet sich das *STACK* (Stapel, Kellerspeicher, filo). Jeder neu berechnete Zustand wird zuoberst auf einen Stapel gelegt, und bei Bedarf wird der oberste Zustand des Stapels genommen und weiterverarbeitet.

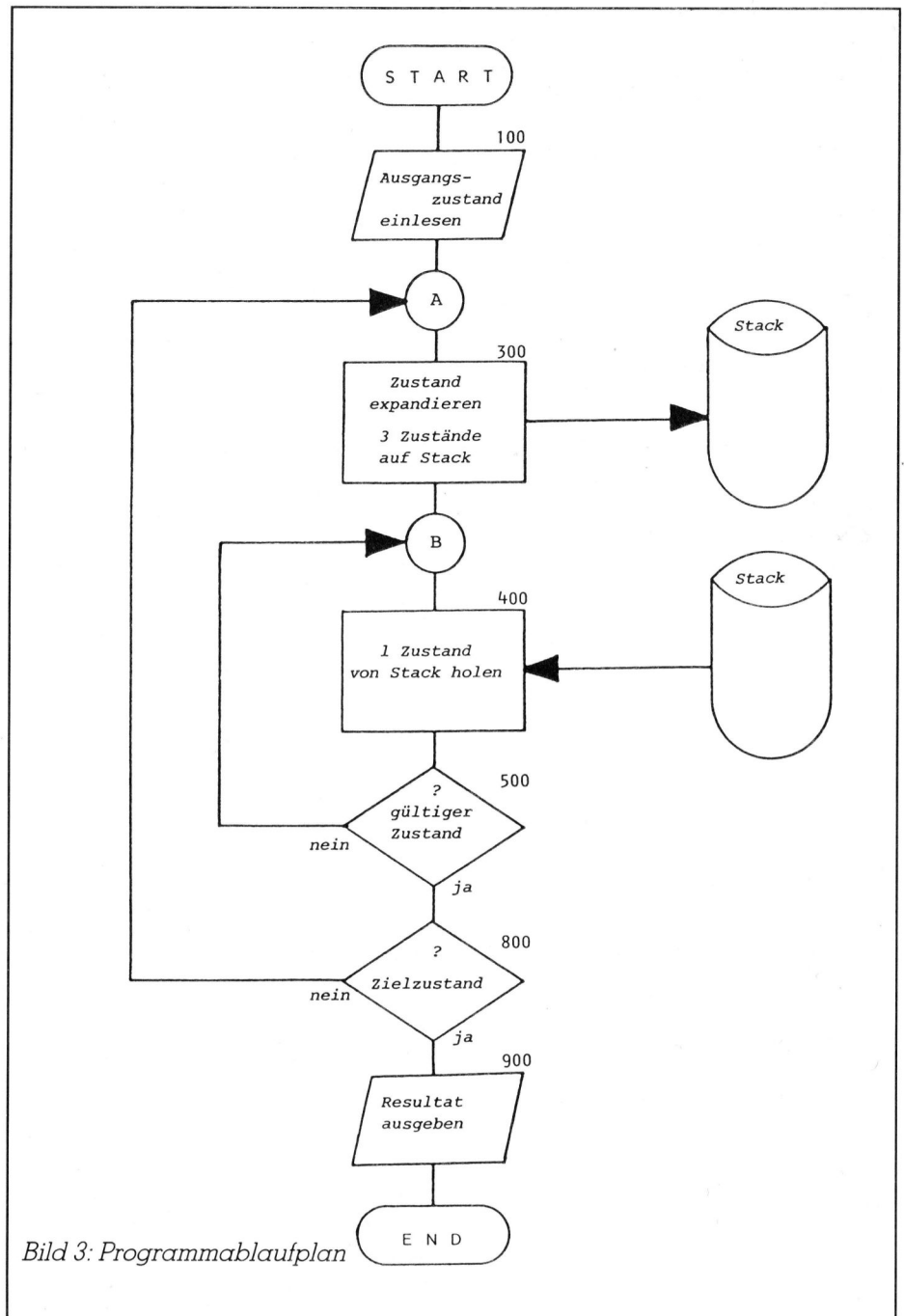


Bild 3: Programmablaufplan

Die Strategie des «Suchens in die Tiefe» werden wir für unser Beispiel verwenden. Den grundsätzlichen Programmablaufplan sehen Sie in Bild 3. Die Umsetzung dieses Ablaufplans in ein lauffähiges BASIC-Programm ist einfach. Es wurde darauf geachtet, dass das Programm möglichst Dialekt-unabhängig ist. Einzig bei der Stringmatrix A, welche den Ausgangszustand in komfortabler Form entgegennimmt, könnten sich Probleme ergeben, wenn Sie in einem BASIC arbeiten, das keine variable Länge der Strings kennt. In diesem Fall müssten Sie diese Matrix mit zwei Dimensionen versehen.

Bei der Darstellung eines Zustandes wird in einem fünften Feld ange-

geben, in welcher Ebene er erreicht wurde. Die Kontrolle über die Ebenen wird mit der Variable E geführt (Zeilen 300, 380, 460, 760). Um am Schluss den ganzen Lösungsweg abrufbar zu haben, wird die Resultate-Matrix R(10,4) definiert. Jeder Zustand der expandiert wird, wird in die vier Stellen (RE,1) bis R(E,4) geschrieben (Zeilen 750-770), wobei E die Ebene dieses Zustandes ist. Am Schluss befindet sich also für jedes E der zuletzt in dieser Ebene expandierte Zustand in der Matrix. Dieser Zustand ist gleichzeitig der Zustand dieser Ebene, der sich auf dem Lösungsweg befindet.

Das STACK kann für dieses einfache Problem im Hauptspeicher auf-

# RUND UM DEN IBM-PC

gebaut werden. Es wird dazu die Matrix S(15,5) mit der dazugehörigen Zeiger-Variable S7 definiert. Der Zeiger befindet sich zu Beginn auf Null. Jeder neu errechnete Zustand wird an die Stelle S(S7,X) geschrieben, wobei S7 vorher um 1 erhöht wurde. Soll ein Zustand vom Stack geholt werden, so wird S(S7,X) gelesen und anschliessend S7 um 1 erniedrigt (Zeilen 410, 450).

Als Letztes muss dafür gesorgt werden, dass ein Zustand, der genau gleich ist wie ein schon einmal expandierter, nicht noch einmal expandiert wird. Zu diesem Zweck wird jeder expandierte Zustand in eine Matrix geschrieben (Zeilen 200-220, 710-730) und für einen neu zu untersuchenden Zustand wird in dieser Matrix nachgeschaut, ob er schon einmal expandiert wurde (Zeilen 500-520). Dafür werden die Matrix A(30,3) und die Zeiger-Variable A7 definiert.

Die Zeile 420 dient einzig dazu, alle vom Programm untersuchten Zustände auf dem Bildschirm zu protokollieren. Sie kann weggelassen werden. Ab Zeile 900 wird der gefundene Weg in verständlicher Form ausgegeben.

Wenn Sie das Programm starten, erwartet es von Ihnen, dass Sie für Wolf, Ziege und Kohl jeweils mit «L» oder «R» angeben, ob sie sich zu Beginn links oder rechts des Flusses befinden. Unser ursprünglich angenommener Ausgangszustand ist dabei L-L-L. Das Programm findet aber die Lösung für jeden anderen Ausgangszustand auch.

## Schlussbemerkungen

Lassen Sie sich vom Protokoll-Ausdruck nicht täuschen. Dieser zeigt nur die «Ueberlegungen» des Programms auf. Wenn das Programm einen Roboter steuern müsste, der die gefundene Lösung in der realen Welt ausführt, so würde dieser erst ab Zeile 900 angesteuert; alles Vorgängige ist nur Planung und ist vergleichbar mit den Ueberlegungen, die der Bauer hoffentlich anstellen würde, bevor er mit den Flussüberquerungen beginnt.

Wenn Sie diese Art von Computer-Anwendungen interessiert, so versuchen Sie jetzt, das gleiche Problem mit der Strategie des «Suchens in die Breite» zu lösen; Sie brauchen dabei nicht sehr viel am Programm zu ändern. Anschliessend können sie den gleichen Algorithmus auf andere Probleme anwenden. □

```
LIST
10 DIM S(15,5),A(30,3),R(10,5)
20 DIM A$(3)
50 S7=1
60 A7=1
70 E=0
100 PRINT "WOLF   L/R ";
105 INPUT A$(1)
110 PRINT "ZIEGE  L/R ";
115 INPUT A$(2)
120 PRINT "KOHL   L/R ";
125 INPUT A$(3)
130 FOR I=1 TO 3
135   IF A$(I)="R" THEN GOTO 150
140   Z(I)=-1
145   GOTO 155
150   Z(I)=1
155 NEXT I
200 FOR I=1 TO 3
210   A(A7,I)=Z(I)
220 NEXT I
300 E=E+1
310 FOR I=1 TO 3
320   S7=S7+1
330   FOR M=1 TO 3
340     S(S7,M)=Z(M)
350   NEXT M
360   S(S7,I)=-S(S7,I)
370   S(S7,4)=I
380   S(S7,5)=E
390 NEXT I
400 FOR I=1 TO 5
410   Z(I)=S(S7,I)
420   PRINT Z(I);
430 NEXT I
440 PRINT
450 S7=S7-1
460 E=Z(5)
500 FOR I=1 TO A7
510   IF Z(1)=A(I,1) AND Z(2)=A(I,2) AND Z(3)=A(I,3) THEN GOTO 400
520 NEXT I
600 IF Z(1)=Z(2) AND Z(4)=3 THEN GOTO 400
610 IF Z(2)=Z(3) AND Z(4)=1 THEN GOTO 400
700 A7=A7+1
710 FOR I=1 TO 3
720   A(A7,I)=Z(I)
730 NEXT I
750 FOR I=1 TO 5
760   R(E,I)=Z(I)
770 NEXT I
800 IF Z(1)=1 AND Z(2)=1 AND Z(3)=1 THEN GOTO 900
810 GOTO 300
900 FOR I=1 TO E
910   G=R(I,4)
920   IF G=1 THEN PRINT "  WOLF  ";
930   IF G=2 THEN PRINT "  ZIEGE  ";
940   IF G=3 THEN PRINT "  KOHL   ";
950   IF R(I,G)=1 THEN PRINT "  <---->";
960   IF R(I,G)=-1 THEN PRINT " <----"
970 NEXT I
999 END
Ok
```

## COMPUTER SPLITTER

### Eindeutig daneben!

(277/fp) Zugegeben, auch die Titelseite von M+K werden in den meisten Fällen von holder Weiblichkeit (mit)geziert. Gelegentlich sind es auch die dynamischen Geschäftsmänner, zu deren Aeusserem selbstverständlich die Krawatte und eine

ernste Miene mit leicht verdunkelten Augenbrauen gehören. Und eingestanden, die Titelsujets führen auch auf der Redaktion hin und wieder zu Diskussionen. Kein Diskussionsthema ist indessen für uns die Verwendung weiblicher Nacktheit für die Werbung von Datenträgern. In gleicher Weise verwerflich ist für uns ein fudlibluttes, doppelseitiges Mannsbild, das die Aufmerksamkeit der Leserinnen einer Frauenzeitschrift auf die Vorzüge eines Commodore Heimcomputers lenken soll! Hoffentlich bleibt die Abbildung anatomischer Details nicht das einzige Argument für die erworbenen Geräte... □

**(258/eh) Das Konzept der offenen Architektur hat IBM dazu verholfen, ihr Kleincomputerprodukt, den IBM-PC, als Industriestandard zu etablieren. Dadurch, dass IBM die technischen Daten ihrer Kleincomputer publizierte und zudem ein frei erhältliches Lizenzprodukt als Betriebssystem verwendet, kann jeder im Computergeschäft mitreden, indem er Zubehörplatinen zum IBM-PC oder gar IBM-kompatible Geräte, ausgerüstet mit dem Betriebssystem MS-DOS von Microsoft, entwickelt. Ob diese schönen Zeiten aber noch länger anhalten werden, ist zweifelhaft. Verschiedene Anzeichen deuten darauf hin, dass IBM die Tür zur offenen Architektur ein wenig zustossen möchte.**

So wurde von IBM kürzlich das erste vollständig von ihr entwickelte Softwareprodukt für ihren Kleincomputer, TOPVIEW, vorgestellt. Obwohl diese Ankündigung im Rummel um den IBM-PC AT etwas unterging, kann dieses Produkt langfristig auf den Markt einen prägenden Einfluss ausüben. Was bietet dieses Programm denn besonders?

TOPVIEW ist eine Erweiterung zum IBM-Betriebssystem, welches dieses stellenweise ersetzt. TOPVIEW erlaubt das gleichzeitige Ablaufenlassen mehrerer Programme und das parallele Darstellen der Resultate in verschiedenen Bildschirmausschnitten (Windowing). Das können andere auch schon, werden Sie hier mit Recht einwenden und dabei an das bereits seit langem angekündigte Microsoft Window, Visi On von Visi-Corp oder Desq von Quarterdeck denken. Stimmt - nur kann TOPVIEW wesentlich mehr.

All die aufgezählten «Window»-Programme verlangen eine Anpassung der Anwenderprogramme. Unter TOPVIEW jedoch sollen alle Programme betrieben werden können, die auf einem normalen IBM-PC lauffähig sind. Dies erlaubt es den unabhängigen Softwareherstellern, sofort ohne zeitraubende und teure Anpassungen auf die «Fenster»-Technik umzusteigen. Dass in dieser Betriebsart nicht alle Fähigkeiten von TOPVIEW voll ausgenutzt werden können, nimmt anfänglich fast jeder in Kauf. Der Appetit kommt bekanntlich beim Essen - ähnlich wird die Situation auch hier sein. Der Softwareproduzent wird sich bemühen, alle Möglichkeiten von TOPVIEW auszuschöpfen, d.h., er wird seine neu auf den Markt kommenden Produkte ganz dem TOPVIEW anpassen - und schon ist die Kompatibilitäts-Falle ein klein wenig zugechnappt. Die Programme laufen jetzt nämlich nur noch auf den Geräten aus dem Hause IBM, denn es darf mit an Sicherheit grenzender Wahrscheinlichkeit angenommen werden, dass IBM an ihre Konkurrenten keine Lizenzen für TOPVIEW vergeben wird. Jeder kann seiner

Vorstellungskraft an dieser Stelle selbst freien Lauf lassen und sich überlegen, was sich die IBM-Strategen sonst noch ausgedacht haben mögen.

Zur Untermuerung dieser meiner Gedanken sollen noch folgende Fakten dienen:

1. IBM verkauft TOPVIEW zu einem sehr günstigen Preis von 149 Dollar. Dies sicher, um rasch einen bestimmten Marktanteil zu erreichen und nicht um einen grossen kurzfristigen Profit zu erzielen.

## IBM quo vadis?

2. Ein IBM-Vizepräsident selbst bezeichnete TOPVIEW als ein für IBM wichtiges strategisches Product (strategic product). Warum so viel Ehre für ein einfaches Window-Programm?

3. IBM hat in einer ersten Kampagne bereits Arbeitskopien von TOPVIEW zweihundert unabhängigen Softwareherstellern in Amerika zur Verfügung gestellt.

Lässt man sich die Geschichte der Einführung des IBM-PC, die vor drei Jahren begann, einmal in Ruhe durch den Kopf gehen, so vermeint man plötzlich schemenhaft eine wahrhaftig clevere Strategie zu erkennen. Wie waren wir doch alle darüber verwundert, dass IBM damals ein in manchen Punkten technisch veraltetes Gerät auf den Markt brachte. Der erste IBM-PC hatte eine Speicherkapazität von nur 64 KByte. Die von IBM dazu angebotenen Speichererweiterungskarten waren auch nicht besonders attraktiv. Die Folge davon war, dass sich viele amerikanische Hardware-Hersteller darum rissen, Erweiterungskarten für den IBM-PC auf den Markt zu werfen. Jeder machte für sein Produkt kräftig Wirbel und im gleichen Atemzug auch für den IBM-Kleincomputer Werbung. Bei einer Durchsicht amerikanischer Kleincomputer-Zeitschriften stellten wir damals fest, dass von 100 Inseraten, die direkt oder indirekt für den IBM-PC war-

ben, nur knapp ein Fünftel von IBM stammten. Beinahe ist man versucht zu sagen, dass die Mängel des IBM-PC diesem zum Industriestandard verhalfen, da jeder durch seine Zusatzprodukte zur Leistungssteigerung Mama Blues Jüngsten erst so richtig bekannt machte.

Was, glauben Sie, empfinden dieselben Hardware-Hersteller heute, wenn sie feststellen müssen, dass die Grundgeräte von IBM jetzt plötzlich serienmässig mit immer grösseren RAM-Speichern ausgerüstet werden? So erhielt der IBM-PC in einer ersten Ueberarbeitung 128 KByte auf der Hauptplatine, und bei der zweiten Erneuerungskur wurde der «On Board»-Speicherbereich sogar auf 256 KByte ausgebaut. Der nun vorgestellte IBM-PC AT wird jetzt sogar vom Werk mit 512 KByte serienmässig ausgeliefert. Wer kauft denn da noch Speicherkarten?

Oder wie fühlen sich heute die Hersteller von IBM-kompatiblen Kleincomputern? Wie werden diese die nun schon zweimalige massive Preissenkung von IBM überleben?

Will IBM nun auch den Markt für Applikationssoftware an sich reißen, indem sie ihr leistungsfähiges Textverarbeitungsprogramm «DISPLAYWRITER», welches bis heute nur auf den teureren IBM-Systemen lief, in Amerika nun für nur 300 Dollar auf den Markt bringt?

Ashton Tate und Lotus sind zwei auch bei uns recht bekannte amerikanische Softwarehersteller. Vor kurzem gab's Bewegung bei deren Aktienkurse - nach unten! Warum wohl? IBM kündigte eine ganze Serie von integrierten Anwendungsprogrammen an, zu denen ein Datenbankmanagement-Programm für nur 250 Dollar, ein «Report Writer» für 150 Dollar, ein Programm zur Erstellung von Geschäftsgraphiken für 200 Dollar und ein weiteres Textsystem, «Words Edition», für nur 150 Dollar gehören. Alle diese Programme nebst einigen Buchhaltungsprogrammen mit dem Label IBM werden durch den bekannt guten Service von IBM unterstützt.

Versiegt nun langsam die Goldmine «Software herstellen»? Bekommt jetzt auch dieser Bereich, der lange mit enormen Umsatzzuwachszahlen brillieren konnte, den harten Wind der Konkurrenz zu spüren?

Auf die in diesem Text aufgeworfenen Fragen, gibt Ihnen der Schreiber - nach Ablauf der in der Computertechnik auch so langen Zeitspanne von einem Jahr - gerne an dieser Stelle wieder Auskunft. □

MCC

**Micro Computer Center**  
Ihr zuverlässiger Partner

KLYBECKSTR. 76 4057 BASEL 061/ 32 12 92

**commodore**  
COMPUTER



<b>HARDWARE:</b>			
COMMODORE	C-64	Home-Computer	Fr. 580.—
COMMODORE	SX-64	mit Floppy	Fr. 2250.—
COMMODORE	1541	Floppylaufwerk	Fr. 680.—
COMMODORE	1530	Kassettengerät	Fr. 80.—
COMMODORE	801	Matrixdrucker 50 Z/Sek.	Fr. 590.—
COMMODORE	1520	4-Farben-Plotter	Fr. 360.—
COMMODORE	1701	Farbmonitor 14"	Fr. 750.—
COMMODORE	1311	JOYSTICK	Fr. 20.—
COMMODORE	1312	PADDLE	Fr. 30.—
<b>SOFTWARE:</b>			
COMMODORE	180A	Textverarbeitung	Fr. 110.—
COMMODORE	181A	Adressverwaltung	Fr. 70.—
COMMODORE	0110	Finanzbuchhaltung	Fr. 350.—
COMMODORE	4003	BASIC-Kurs Kass.	Fr. 35.—
COMMODORE	4013	BASIC-Kurs Disk.	Fr. 45.—

Bestellung von: .....

Stück	Artikel	Preis
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____

Bitte ausschneiden und an obige Adresse senden.  
Versand inkl. Porto und Verpackung gegen Nachnahme.

**SKC® Der Augenblick kommt, wo nur das Beste zählt**



**Ihr Vorteil:**

- 100% fehlerfrei
- Nach DIN, ECMA, IBM und Shugart-Norm entwickelt
- enorm günstig

<b>5 1/4</b> Zoll	SS SD 40 TPI	1S	SS/SD	<b>5.85</b>
	SS DD 48 TPI	2D	DS/DD	<b>7.95</b>
	SS DD 96 TPI	<b>NEU</b>	1D 96tpi	<b>8.25</b>
	DSDD 48 TPI		2D 96tpi	<b>9.75</b>
	DSDD 96 TPI			
3 1/2 Zoll Disketten in Vorbereitung				

SKC Disketten sind 100% ERROR FREE d. h. jede einzelne Diskette wird sorgfältig geprüft. Damit wird gleichbleibende zuverlässige Qualität garantiert.

Verlangen Sie Unterlagen! (Händler willkommen)

**EM ELECTRONIC MARKETING AG**  
Your Swiss distributor for high technology.  
Jurastr. 45, 4002 Basel, Telefon 061 / 35 57 00

**Nutzen Sie Ihren Apple-Computer auch zum Lernen**

INTUS-Lernprogramme sind attraktiv und motivierend für vergnüglichen und leichtes Lernen mit hohem Lernerfolg. Alle Programme in deutscher Sprache für Apple IIe und IIc.

- Maschineschreiben wie der Blitz. Didaktisch ausgezeichnet. Garantiert erfolgreich in 20 Lektionen. Fr. 145.—
- Basic-Lernprogramm. Bestes Lernprogramm 1982 in den USA. Fr. 225.—
- Computer-Simulator. Mit Lehrgang in 5 Lektionen. Fr. 112.—
- LOGO deutsch. «Das neue Lernen» – Computersprache für Anfänger und «Profis». (Probierdisk Fr. 25.—) Fr. 380.—
- Deutsche Grammatik mit Spass. Für Sekretärinnen, Schüler und «Jedermann». Einschl. Rechtschreibe-Training mit Vortest/Nachtest. 12 Bereiche. Je Disk Fr. 125.—
- Wortschatztrainer, Franz./Deutsch, 800 Wörter Fr. 108.—  
Englisch/Deutsch, 800 Wörter Fr. 108.—  
Business English, 1200 Wörter Fr. 125.—  
Attention please, 2400 Wörter Fr. 216.—
- An der Börse jonglieren. Praxisgerechtes, aufregendes, lehrreiches Börsengeschehen. Fr. 82.—
- Unternehmens-Planspiel «Kartellbrüder und Halsabschneider». Professionelles, lehrreiches Strategiespiel. Vorkenntnisse notwendig. Umfangreiche Dok. Fr. 380.—
- Schulprogramme wie Rechnen/Mathe, Sprachen, Physik, Chemie, Informatik, Vorschule. Bestens für die Nachhilfe geeignet. Vermietung von Programmen.
- Testmiete von Apple IIc-Computer, Fr. 240.— pro Monat. Anrechnung bei Kauf. Günstige Vorführ- und Leihgeräte.

\*\*\*\* Mit Ihrer ersten Bestellung erhalten Sie gratis das Programm «Mein Freund, der Apple» mit interessanten Denk-, Strategie- und Entspannungsspielen.



INTUS LERN-SYSTEME AG

6981 Astano, Tel. 091 - 73 25 51 und 042 - 22 31 13

# Splines: Biegsame Kurvenlineale

Wenn man vor der Aufgabe steht, durch Punkte eine Kurve zu legen, die einerseits dem Auge angenehm erscheint, andererseits sich aber möglichst eng an die gegebenen Punkte anschmiegt, so benutzt man zweckmässigerweise ein biegsames Kurvenlineal. Im Laufe dieses Jahrhunderts ist es nun der Mathematik gelungen, Funktionen zu definieren, die sich wie ein biegsames Kurvenlineal benehmen. Man spricht von sogenannten «nähernden kubischen Splines». Hinter diesem etwas geheimnisvollen Namen verbirgt sich jedoch im Wesentlichen hauptsächlich Schulmathematik.

Zunächst ist «Spline» die englische Bezeichnung für das oben erwähnte biegsame Kurvenlineal. Ursprünglich entsprach dem Spline die «Stracklatte», eine Latte, die dazu diente, die Form des Schiffsrumpfes zwischen den Spanten (Querstützen eines Rumpfes) herzustellen. Sie gab dann dem Rumpf die gewünschte glatte Form. Die Stracklatte ist gewissermassen der Vorläufer des

## Dr. Andreas Heertsch

biegsamen Kurvenlineals. Man hat ursprünglich die erwähnten Funktionen auch «Strack-Funktionen» genannt. Allerdings hat sich schliesslich doch der englische Ausdruck «Spline» durchgesetzt (1).

Ehe wir zu Programmen kommen, die solche Spline-Funktionen durch Punkte legen, wollen wir zunächst untersuchen, wie man einen solchen kubischen Spline erhält.

### Mathematik der Splines

Nehmen wir an, wir sollten eine glatte Kurve durch drei beliebig liegende Punkte legen. Zunächst kann man sie einfach durch Geraden verbinden (Abb. 1). Das führt zum Polygonzug (siehe dazu auch M+K 84-2). Um jetzt im mittleren Punkt die Ecke zu vermeiden, muss die Linie rechts des Punktes in der Nähe des Punktes die gleiche Richtung haben, wie in der Nähe links des Punktes (Abb. 2). Im betrachteten Punkt müssen demnach die Kurven von links und von rechts die gleiche Steigung haben. Mathematisch formuliert: Die Ableitungen der Kurven müssen im betrachteten Punkt gleich sein.

Um nun Punkte in beliebiger Anordnung durch glatte Kurven verbinden zu können, müssen die Kurven die Fähigkeit haben, Wendepunkte durchlaufen zu können (Abb. 3). Die Kurve zwischen den beiden mittleren Punkten hat einen solchen Wendepunkt: zunächst links eine «Linkskurve», dann rechts eine «Rechtskurve».

Die einfachsten Kurven, die das können, sind Parabeln 3. Grades (eben «kubische» Polynome, Abb. 4):

$$y = a \cdot x^3 + b \cdot x^2 + c \cdot x + d$$

Eine Spline-Funktion ist nun nichts weiter als eine Sammlung solcher kubischer Polynomstücke, die so zusammengesetzt werden, dass sie glatte Anschlüsse in den Stützpunkten haben.

Allerdings müssen wir noch eine weitere Bedingung fordern, sonst könnten die Kurven überschwingen (Abb. 5). Dies kann man vermeiden, indem man fordert, dass in den Stützpunkten auch die Änderung der Steigung (2. Ableitung) nach beiden Seiten gleich ist, dass also aus einer Rechtskurve nicht plötzlich eine Linkskurve wird, sondern der Punkt weiter in einer Rechtskurve durchlaufen wird. Ausnahme: der Stützpunkt liegt genau im Wendepunkt der Kurve.

Es besteht also eine solche Spline-Funktion aus lauter einzelnen Versatzstücken zwischen je zwei Stützpunkten. Demnach müssen für  $n$  Punkte  $n-1$  Versatzstücke berechnet werden. Jedes dieser Versatzstücke wird durch vier Koeffizienten ( $a, b, c, d$ ) definiert, also für die Verbindung zwischen Punkt  $i$  und Punkt  $i+1$ :

$$y(i, x) = a(i) \cdot x^3 + b(i) \cdot x^2 + c(i) \cdot x + d(i)$$

Diese Koeffizienten berechnet man, indem man ein kompliziertes Gleichungssystem (aus  $4 \cdot (n-1)$  Gleichungen mit ebensovielen Unbekannten) löst. Sind alle Koeffizienten berechnet, so lässt sich aus ihnen die Spline-Funktion plotten.

### Nähernder Spline

Wenn man Messwerte, die mit Fehlern behaftet sind, plotten will, die also nicht genau auf einer Kurve liegen müssen, so entsteht die Notwendigkeit, die Kurve nicht genau

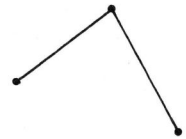


Abb. 1: Polygonzug (durch Geraden verbundene Punkte)

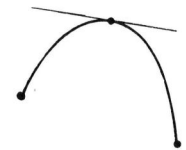


Abb. 2: Die Steigung der Kurve links des mittleren Punktes ist gleich der Steigung des rechten Punktes: Beide Kurvenstücke haben im betrachteten Punkt eine gemeinsame Tangente.

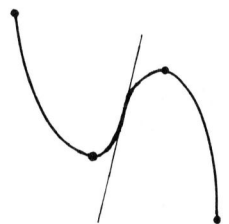


Abb. 3: Die Kurve zwischen den beiden mittleren Punkten hat einen Wendepunkt.

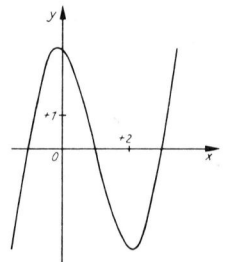


Abb. 4: Kubische Parabel; nach (4)

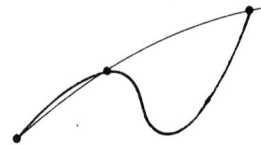


Abb. 5: Ueberschwingende Kurve

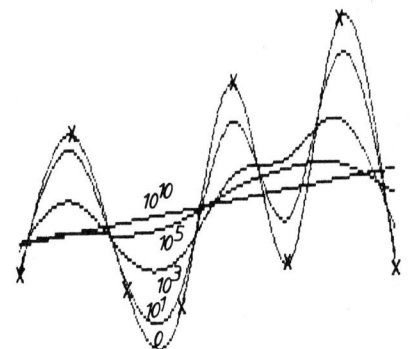


Abb. 6: Verschiedene Gesamtgewichte  $s$  für die gleichen vorgegebenen Punkte.

```

80 rem"smooth1.bas
90 data132,174,127,122,189,135,8,133
100 xa=0:xe=159:d=1:n=8:dimx(n),y(n),a(n),b(n),c(n),d(n),h(n),x0(n)
105 n2=n:dimr(n2+1),r1(n2+1),r2(n2+1),t(n2+1),t1(n2+1),u(n2+1),v(n2+1)
110 m=550:dimxt(m),yt(m):ms=0
120 fori=1ton:x(i)=40#i:ready(i):next:gosub4000
125 fori=1ton:x=x(i):y=y(i):gosub2000:next
130 n1=1:input"s";s:fori=n1ton2:dy(i)=10:next:gosub60203:ifie<>0then200
140 xa=40:xe=n*40:gobsub43870:ifie<>0then200
150 x=xt(0):y=yt(0):gosub2500:fori=0tol-1:y=yt(i):x=xt(i):gosub3000:next:end
200 print"error";ie;" in line";ln:end
2000 rem"plot kreuz
2010 ifgg=0thengg=1:dimkrx(11):forkr=0to11:readkrx(kr):next
2020 ifx>y>0then:gobsub5000:!p,x,y:!t,5,krx(0):return
2030 data2,1,-4,-2,2,1,2,-1,-4,2,2,-1
2500 rem"plot punkt
2510 gosub5000:!p,x,y:return
3000 rem"plot linie
3010 gosub5000:!m,x,y:return
4000 rem"plot kasten
4010 print"█":!s:!a,3:!o,5,5,505,205:!a,3:return
5000 rem"plot grenzen
5010 x=x+5:ifx>505thenx=505elseifx<5thenx=5
5020 y=y+5:ify>210theny=205elseify<5theny=5
5030 return

```

Abb. 7: Hauptprogramm smooth.bas

```

43800 rem"sptab.bas
43870 ie=0:ifxa=xethenie=1:ln=43870:return
43890 ifd=0thenie=2:ln=43890:return
43900 l=0:i=1:k=n
43930 m=int((i+k)/2):ifm=i then44000
43950 ifxa<x(m)thenk=melseifm
43990 goto43930
44000 ia=i:k=n
44020 m=int((k+i)/2):ifm=i then44090
44030 ifxe<x(m)thenk=melseifm
44040 goto44020
44090 je=i:ifia=jethen44210
44100 i=ia:x2=xa:x3=x(ia+1):gosub44330:ifje-ia=1then44210
44160 fori=ia+1toje-1:x2=x(i):x3=x(i+1):gosub44330:next
44210 i=je:x2=x(i):x3=xe:gobsub44330:l=1-1:ifxt(1)=xethenreturn
44270 l=1+1:xt(1)=xe:x1=xe-x(je):yt(1)=((d(je)*x1+c(je))*x1+b(je))*x1+a(je)
44310 return
44330 forx0=x2tox3stepd:x1=x0-x(i):xt(1)=x0
44360 yt(1)=((d(i)*x1+c(i))*x1+b(i))*x1+a(i):l=1+1:next:return

```

Abb. 8: Unterprogramm sptab.bas (Spline tabellieren)

durch die Punkte zu legen, sondern so, dass die Kurve möglichst durch die Mitte der «Punktwolke» geht: Man approximiert die Messwerte durch die Kurve. Dabei können ausserdem gewisse Messwerte besonders berücksichtigt werden, da man von ihnen beispielsweise weiss, dass sie besonders genau gemessen sind: Die einzelnen Punkte sollen mit verschiedenem Gewicht eingehen.

Man wird also jedem Wert ein Gewicht geben. Damit sind die Werte in ihrer «Gewichtigkeit» untereinander bestimmt. Um noch festzulegen, wie sehr die Gewichte der einzelnen Punkte insgesamt für die Auslegung der Kurve ins Gewicht fallen sollen, muss man zusätzlich noch eine Art Gesamtgewicht angeben, dem die Kurve entsprechen soll. Ist das Gesamtgewicht gross, so «kümmert» sich die Kurve kaum um die einzelnen Punkte. Sie legt sich im Extremfall als Gerade durch die Punktwolke. Wird das Gesamtgewicht relativ

klein, so folgt die Kurve den Punkten immer stärker, sie wird immer welliger. Ist das Gesamtgewicht 0, so benimmt sich die Kurve wie ein normaler Spline: Sie geht genau durch die Punkte. Auf diese Weise kann man durch eine Punktwolke ganz nach Belieben einen Spline immer mehr anschmiegen (Abb. 6). Voraussetzung ist allerdings bei allen Spline-Berechnungen, dass die Werte der x-Koordinate in aufsteigender Grösse sortiert sind (wie man diese Bedingung umgehen kann, werden wir im Abschnitt «Parametrisierung» noch sehen).

### Spline-Programmbeschreibung

Das in Abb. 9 angegebene Spline-Programm (smoothing.bas) ist nach (3) für BASIC umgeschrieben. Es berechnet approximierende kubische Splines. Weiterhin ist ein Tabellierprogramm (sptab.bas) angegeben (Abb. 8). Es ist der sehr zu empfeh-

lenden Algorithmensammlung (2) entnommen. Das gesamte Programm besteht aus drei Teilen:

- Hauptprogramm (80-5030), das die Bilder erstellt und die beiden Unterprogramme aufruft. Dieses Programm muss der Leser seinen Anforderungen entsprechend gestalten (Abb. 7).
- sptab.bas (43800-44360). Es tabelliert aus den Spline-Koeffizienten eine Spline-Funktion. Sie wird dann in Zeile 150 geplottet (Abb. 8).
- smoothing.bas (60000-60420). Es berechnet die vier Spline-Koeffizienten (a,b,c,d) für jedes Punktepaar.

### Hauptprogramm (Abb. 7)

#### Zeile 90

Definition der Y-Koordinaten der Stützpunkte.

#### Zeile 100-110

Definition der Felder dabei ist:

n = Anzahl der Punkte

d = Abstand zwischen zwei

Tabellenwerten von sptab.bas

m = Anzahl der Tabellenwerte in sptab.bas

#### Zeile 120

Koordinaten der Stützpunkte berechnen

#### Zeile 125

Punkte plotten

#### Zeile 130

Aufruf von smoothing.bas

n1 = Index des ersten Stützpunktes

n2 = Index des letzten Stützpunktes

#### Zeile 140

Aufruf von sptab.bas

xa = Anfangswert der Tabelle

xe = Endwert der Tabelle

#### Zeile 150

Funktion plotten

### Literatur

(1) Böhmer, K.; Spline-Funktionen; Täubner-Verlag; Stuttgart 1974

(2) Engeln-Müllges, G./Reutter, F.; Formelsammlung zur numerischen Mathematik mit BASIC-Programmen; B.I.; Mannheim 1983

(3) Reinsch, H.C.; Smoothing by Spline Functions; Numer. Math. 10; p. 177-183 (1967) und 16, p. 451-454 (1971)

(4) Kleine Enzyklopädie Mathematik; Hrsg. Gallert, W. et al.; Frankfurt; Zürich 1972



**Zeile 1000**

File-Handling (für run1000)

**Zeile 2000-3010**

Plotroutine für die Plotsymbole «Kreuz», «Punkt» und «Linie»

**Programm sptab.bas**

sptab.bas tabelliert die Funktionswerte in die Felder xt(i), yt(i) beginnend mit xa im Abstand d bis xe. Entsprechend müssen die Bereiche xt(i), yt(i) mit  $m=(xe-xa)/d+n-2$  dimensioniert sein.

l = Anzahl der Tabellenwerte (Ausgabe)

**Zeile 43870**

Einsprung:  $\alpha < xe$  sonst error ie = 1

**Zeile 43890**

Abstand  $d > 0$  sonst error ie = 2

**Zeile 43900-44210**

Anzahl der Tabellenwerte zwischen zwei Stützpunkten bestimmen

**Zeile 44270-44310**

Letzten Tabellenwert berechnen

**Zeile 44330-44360**

Tabellenwerte mit den zugehörigen Koeffizienten bestimmen:  $yt(i) = d(i) \cdot xt(i) + c(i) \cdot xt(i) + b(i) \cdot xt(i) + \alpha(i)$

**Programm smoothing.bas**

smoothing.bas bestimmt die Koeffizienten a(i),b(i),c(i),d(i) mit  $i = n1 \dots n2-1$ , n1 = Index des 1. Stützpunktes, n2 = Index des letzten Stützpunktes. Die Koeffizienten mit dem Index i definieren die Ausgleichsparabel zwischen den Punkten x(i),y(i) und x(i+1),y(i+1).

Eingabe:

- n1 = Index des 1. Stützpunktes
- n2 = Index des letzten Stützpunktes
- x(i),y(i) = x,y-Koordinaten der Stützpunkte,  $x(i) < x(i+1)$
- dy(i) = Gewichtsfunktion (Gewicht der einzelnen Stützpunkte)
- s = Gesamtgewicht (Anpassungsgrad),  $s=0$ : Spline,  $s > > 1$ : Gerade

Ausgabe:

- a(i),b(i),c(i),d(i) = Koeffizienten der Ausgleichsparabeln
- i = n1...n2-1
- Fehler:
- ie = -1, falls  $n1 > n2$
- ie = -2, falls  $s < 0$
- ie = Index der ersten nicht monoton steigenden Koordinate

Für den mathematischen Hintergrund siehe (3).

**Glättung von Kurven**

Dass Spline-Funktionen nicht nur ästhetischen Wert haben, zeigt Ta-

```

60000 rem"smoothing.bas
60010 rem smoothing routine benutzt kubischen spline zur naeherung durch
60020 rem eine punktwolke.
60030 rem
60040 rem dimensionen im rufenden programm:
60050 rem dimr(n2+1),r1(n2+1),r2(n2+1),t(n2+1),t1(n2+1),u(n2+1),v(n2+1)
60060 rem dimx(n2),y(n2),dy(n2),a(n2),b(n2),c(n2),d(n2)
60070 rem
60080 rem eingabe - parameter:
60090 rem n1,n2   nr. des 1. und letzten datenpunktes
60100 rem x,y,dy  felder mit x(i),y(i),dy(i)↑,5 als abszisse, ordinate und
60110 rem                rel. gewicht des i-ten datenpunktes (i=n1(1)n2). die
60120 rem                die x-werte müssen streng monoton steigen.
60130 rem s      nicht-negativer parameter, bestimmt den grad der anpassung
60140 rem                mit s) = (f(x(i)) - y(i)/dy(i))↑2 ; f(x(i)) beschreibt
60150 rem                eine gerade
60160 rem
60170 rem ausgabe - parameter:
60180 rem a,b,c,d  felder, die die kubischen spline-koeffizienten enthalten
60190 rem                a(n2)=c(n2)=0, b(n2),d(n2) nicht definiert
60191 rem
60192 rem"benutzte variablen:
60193 rem"n1,n2,s,sy,x(i),y(i),dy(i),a(i),b(i),c(i),d(i)
60194 rem"i,m1,m2,e,f,f2,g,h,p,r(n),r1(i),r2(i),t(i),t1(i),u(i),v(i)
60200 sy=0:ifn1=1thenln=60203:ie=-1:return
60203 m1=n1+1:m2=n2-1:p=0:r(m1)=p:r(n1)=p:r2(n2)=p:r2(m2)=p:u(m1)=p:u(n1)=p
60205 fori=n1ton2:sy=sy+dy(i):ifx(i-1)<x(i)thennext:elseifln=60205:ie=i:return
60207 sy=s*sy:ifs<0thenln=60207:ie=-2:return
60210 u(n2)=p:u(m2)=p:r1(n2)=p:h=x(m1)-x(n1):f=(y(m1)-y(n1))/h
60220 f2=2/3:fori=m1tom2:g=h:h=x(i+1)-x(i):e=f:f=(y(i+1)-y(i))/h:a(i)=f-e
60230 t(i)=f2*(g+h):t1(i)=h/3:r2(i)=dy(i-1)/g:r(i)=dy(i+1)/h
60240 r1(i)=-dy(i)/g-dy(i)/h:next
60250 fori=m1tom2:b(i)=r(i)*r(i)+r1(i)*r1(i)+r2(i)*r2(i)
60260 c(i)=r(i)*r1(i)+r1(i)*r1(i)+r2(i)*r2(i):d(i)=r(i)*r2(i+2):next:f2=-sy
60270 fori=m1tom2:r1(i-1)=f*r(i-1):r2(i-2)=g*r(i-2)
60280 r(i)=1/(p*b(i)+t(i)-f*r1(i-1)-g*r2(i-2))
60290 u(i)=a(i)-r1(i-1)*u(i-1)-r2(i-2)*u(i-2)
60300 f=p*c(i)+t1(i)-h*r1(i-1):g=h:h=d(i)*p:next
60310 fori=m2tom1step-1:u(i)=r(i)*u(i)-r1(i)*u(i+1)-r2(i)*u(i+2):next:e=0:h=0
60320 fori=n1tom2:g=h:h=(u(i+1)-u(i))/(x(i+1)-x(i))
60330 v(i)=(h-g)*dy(i)*dy(i):e=e+v(i)*(h-g):next
60340 g=-h*dy(n2)*dy(n2):v(n2)=g:e=e-g*h:g=f2:f2=e*p*p
60350 iff2=0syorf2<gthen60400
60360 f=0:h=(v(m1)-v(n1))/(x(m1)-x(n1))
60370 fori=m1tom2:g=h:h=(v(i+1)-v(i))/(x(i+1)-x(i))
60380 g=h-g-r1(i-1)*r(i-1)-r2(i-2)*r(i-2):f=f+g*r(i)*g:r(i)=g:next
60390 h=e-p*f:ifh<0thenp=(sy-f2)/((sqr(sy/e)+p)*h):goto60270
60400 fori=n1tom2:a(i)=y(i)-p*v(i):c(i)=u(i):next
60410 fori=n1tom2:h=x(i+1)-x(i):d(i)=(c(i+1)-c(i))/3/h
60420 b(i)=(a(i+1)-a(i))/h-(h*d(i)+c(i))*h:next:return
    
```

Abb. 9: Unterprogramm smoothing.bas (nähernden Spline berechnen)

```

80 rem"param.bas
90 data01,1,02,2,01,1,04,1,03,2,04,1,04,4,03,3,04,4
91 data01,4,02,3,01,4,01,1,02,2,01,1,04,1
100 n=16:xa=2.9:xe=15:d=.1:dimx(n),y(n),a(n),b(n),c(n),d(n),h(n),x0(n),dy(n)
105 n2=n:n1=1:dimr(n1),r1(n+1),r2(n+1),t(n+1),t1(n+1),u(n+1),v(n+1),xx(n)
110 m=(xe-xa)/d+n:dimxt(m),yt(m),ty(m),xm(m),ym(m):ms=0
120 fori=1ton:readx(i),y(i):next:fori=1ton:x(i)=100*x(i):y(i)=50*y(i)-25:next
125 gosub4000:fori=1ton:x=x(i):y=y(i):gosub2000:xx(i)=x(i):x(i)=i:next
130 s=0:fori=n1ton2:dy(i)=10:next:gosub60200:ifie<0then200
140 gosub43870:ifie<0then200:"y-komponente"
145 fori=0tol:ty(i)=yt(i):next:fori=1ton:y(i)=xx(i):next:gosub60200:ifie<0then200
146 gosub43870:ifie<0then200:"x-komponente"
148 fori=0tol:xt(i)=yt(i):yt(i)=ty(i):next
150 x=xt(1):y=yt(1):gosub2500:fori=1tol-1:y=yt(i):x=xt(i):gosub3000:next:end
200 print"error";ie;" in line";ln:end
    
```

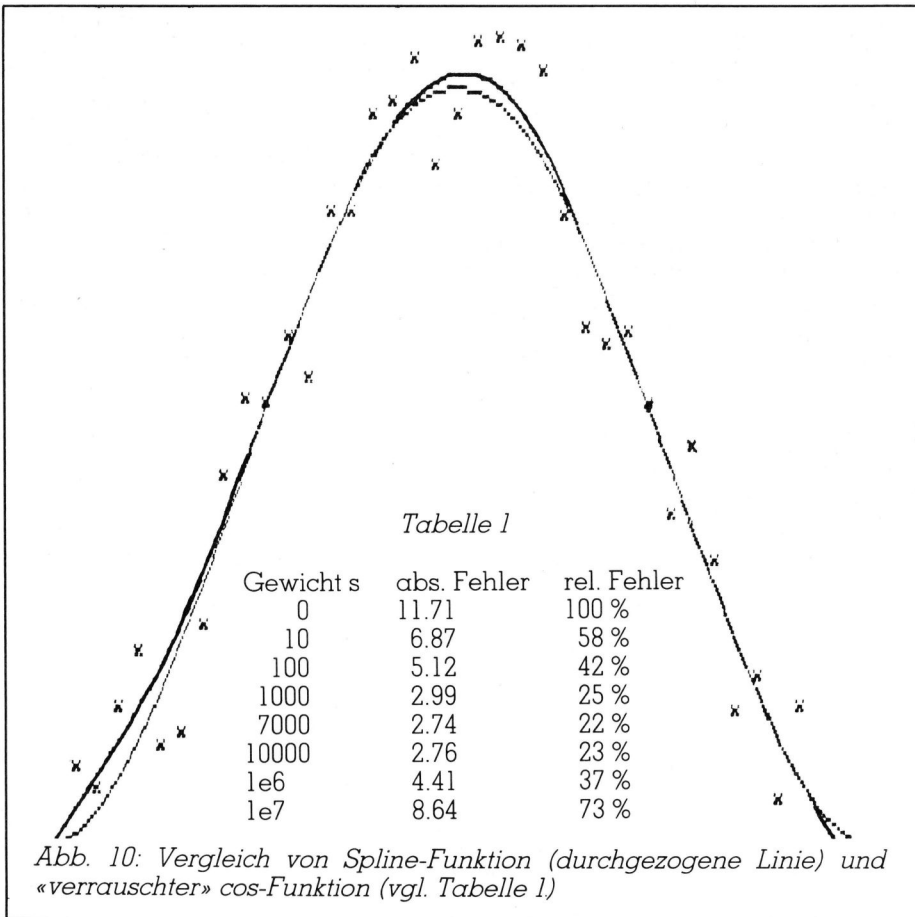
Abb. 13: Hauptprogramm param.bas erstellt Abb. 12. Aufruf von sptab.bas, smoothing.bas

belle 1 und Abb. 9. Dort ist eine cos-Funktion aufgetragen, die zusätzlich mit Zufallszahlen «verrauscht» wurde:

$$f(x) = 1 - \cos(x) + (\text{rnd}(0) - 0.5) \cdot 0.1$$

Tabelle 1 zeigt nun die mittleren Differenzen der glättenden Spline-

funktion gegenüber der erzeugenden, ungestörten cos-Funktion. Dabei sind verschiedene Glättungsgewichte s benutzt. Man sieht, dass man durch eine geeignete Gewichtung zu einer guten Näherung der Kurve kommen kann. (Allerdings muss man wissen, wie die anzunähernde Kurve aussieht, um genau



das richtige Glättungsgewicht zu finden; sonst muss man sich auf sein Augenmass verlassen.)

## Parametrisierung von zwei-dimensionalen Splines

Die Bedingung, dass alle Werte der Grösse nach geordnet sein müssen, ist sehr einschränkend, wenn

man z.B. Punkte miteinander verbinden will, die ein Quadrat bilden.

Keine geschlossene Kurve (bei der Anfangs- und Endpunkt zusammenfallen) lässt sich durch monoton steigende x-Koordinaten beschreiben (irgendwo muss sie ja zum Ausgangspunkt zurückkehren).

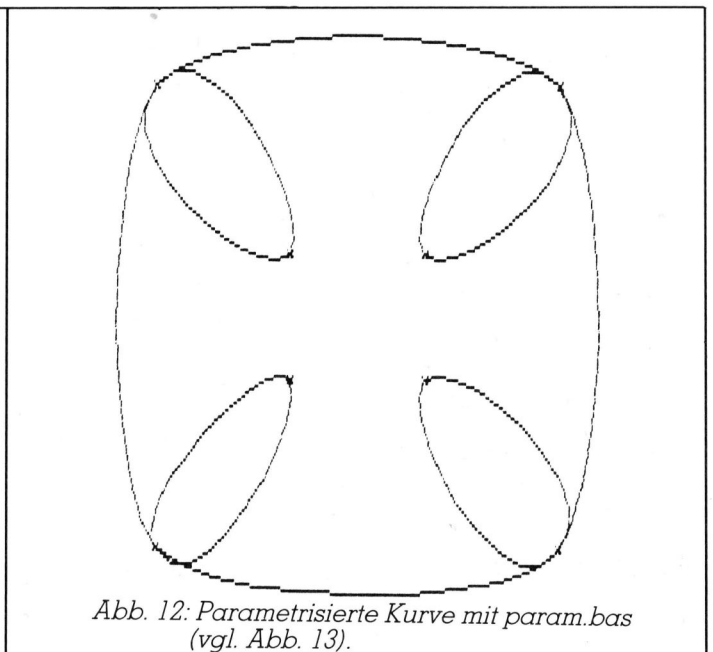
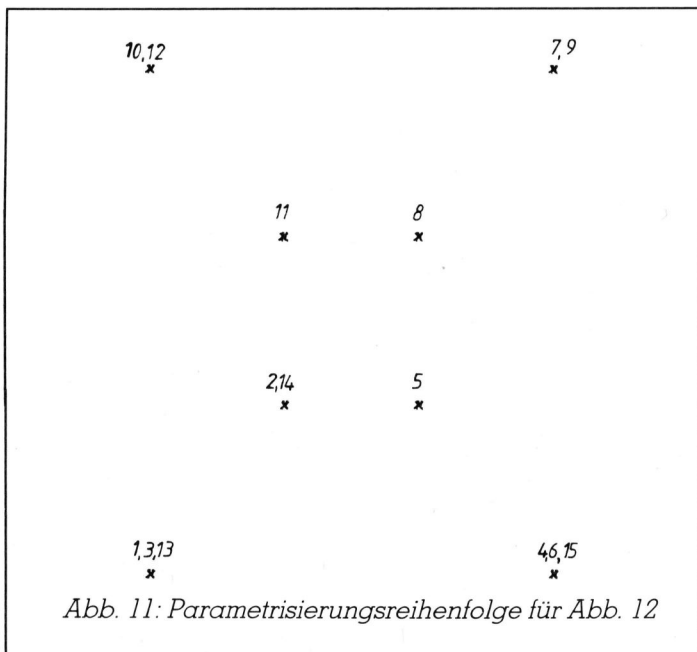
Wenn man solche Kurven trotzdem durch Splines darstellen will, muss

man sich durch einen Trick behelfen: Man berechnet die x- und y-Koordinate der Funktion getrennt. Das geschieht, indem man sich vorstellt, man fährt die Kurve mit einem Stift ab und ordnet als neue Koordinate die Länge des Kurvenstriches jeweils der x- und der y-Koordinate zu. So entstehen zwei Funktionen, deren gemeinsame Längenkoordinate monoton wächst. Das nennt man «eine Kurve durch ihre Bogenlänge parametrisieren». Beim Plotten der Kurve wird die x-Funktion für die Auftragung der x-Achse und entsprechend die y-Funktion für die Auftragung der y-Koordinaten verwendet.

Nehmen wir als Beispiel die acht Punkte in Abb. 11: Sie sind in der Reihenfolge numeriert, wie sie in Abb. 12 durchlaufen werden (Programm param.bas, Abb. 13).

Man sollte nun erwarten, dass die inneren vier Punkte je einmal und die äusseren vier je zweimal durchlaufen werden. Die beiden unteren Punkte und der linke untere innere werden jedoch noch einmal mehr durchlaufen. Dies geschieht, damit sich insgesamt eine geschlossene, glatte Kurve ergibt; andernfalls hätte sie am Anfangs- bzw. Endpunkt einen Knick, da ja die Krümmung eines Verbindungspolynoms immer auch von den beiden Nachbarpolynomen bestimmt wird. Bei genauer Betrachtung zeigt sich trotzdem, dass die linke untere Schlaufe etwas schlanker ist als die übrigen.

Die lässt sich vermeiden, wenn man die Steigungen im Anfangs- und Endpunkt vorgeben kann. Programme, bei denen das möglich ist, sind in (2) angegeben. □



# Ein Proportional-Blocksatz-Druckprogramm

**Die wenigsten der für Personal Computer geschriebenen Textverarbeitungs-Programme unterstützen den proportionalen Blocksatz, obwohl die Mehrzahl der auf dem Markt erhältlichen Drucker das Drucken mit einem proportionalen Zeichensatz erlauben. Dem Anwender, der auf den proportionalen Blocksatz nicht verzichten möchte, bleibt deshalb häufig nichts anderes übrig, als selbst ein solches Programm zu entwickeln, was verständlicherweise viel kostbare Zeit in Anspruch nimmt.**

Im folgenden stelle ich deshalb ein Programm vor, welches das Formatieren eines beliebigen Textes auf elegante Art und Weise erledigt. Als Besitzer eines Druckers von C. Itoh (Modelle 8510 und 1550) können Sie das Programm «as is» direkt übernehmen, - vorausgesetzt natürlich, dass Sie im Besitze eines Pascal-

eingeleitet, welche einige Besonderheiten aufweist, die beim Uebersetzen des Programmes in andere Pascal-Dialekte beachtet werden sollten.

Ein Merkmal, das beim Betrachten der Konstanten-Liste besonders auffällt, ist die zweimal angewandte typenbezogene Konstantendeklaration (Zeilen 11, 31), welche vom Turbo-

Pascal-Compiler ausdrücklich unterstützt wird und die eine rationelle Implementation von Tabellen und Listen ermöglicht. Ein weitere Eigenart von Turbo-Pascal tritt auf den Zeilen 35..38 zu Tage. Hier werden spezielle Konstanten vom Typ CHAR durch Angabe ihres ASCII-Codes direkt initialisiert. Dieser kann normalerweise nur durch die Funktion CHR(<Code>) einem Zeichen zugewiesen werden.

## Variablen

Der Uebersichtlichkeit halber sind die meisten Variablen global deklariert (VAR-Tabelle auf den Zeilen 45..62). In den Prozeduren selbst sind nur wenige Hilfsvariablen zusätzlich lokal deklariert und es erfolgt auch keine Parameterübergabe in die Prozeduren.

## Andreas Widmer

Compilers sind, der ähnliche Operationen erlaubt wie der zur Entwicklung des Programms gebrauchte TURBO-Pascal-Compiler von Borland, welcher auf den Betriebssystemen CP/M und MS-DOS verfügbar ist und sich durch ein bisher noch nie dagewesenes Preis-Leistungs-Verhältnis auszeichnet.

Grosse Vorteile verschafft das Programm besonders auch Benützern des UCSD-p-Systems (mit zugehörigem Pascal-Compiler), da der im System enthaltene Text-Editor keinerlei Formatierungsmöglichkeiten bietet. Die im Turbo-Pascal-Quelltext direkt enthaltene Liste der Zeichenbreiten muss bei der UCSD-Version allerdings auf andere Weise erzeugt werden. Dies geschieht mit Vorteil durch das Einlesen der entsprechenden Daten von einer Diskettendatei. Alle nötigen Anweisungen hierzu entnehmen Sie bitte dem UCSD-Pascal-Handbuch.

Im übrigen lässt sich das Programm auch auf diverse andere Drucker anpassen, wobei nur minimale Aenderungen vorgenommen werden müssen (Zeichenbreite, Drucker-Steuer-Codes). Das Prinzip bleibt das gleiche.

## Beschreibung des Programmes im einzelnen

### Konstanten

Das Programm wird durch eine umfangreiche Konstanten-Tabelle

```
#P+(44)
```

```
\Beispiel für Proportionalzeichen-Blocksatz\
```

```
Die wenigsten der für Personal Computer geschriebenen
Textverarbeitungs-Programme unterstützen den proportionalen
Blocksatz, obwohl die Mehrzahl der auf dem Markt erhält-
lichen Drucker das Drucken mit einem proportionalen Zeichen-
satz erlauben. Dem Anwender, der auf den proportionalen
Blocksatz nicht verzichten möchte, bleibt deshalb häufig
nichts anderes übrig, als selbst ein solches Programm zu
entwickeln, was verständlicher Weise viel kostbare Zeit in
Anspruch nimmt.
```

### Normaler «Schreibmaschinen»-Textausdruck

```
Beispiel für Proportionalzeichen-Blocksatz
```

```
Die wenigsten der für Personal Computer geschriebenen
Textverarbeitungs-Programme unterstützen den proportionalen
Blocksatz, obwohl die Mehrzahl der auf dem Markt erhält-
lichen Drucker das Drucken mit einem proportionalen Zeichen-
satz erlauben. Dem Anwender, der auf den proportionalen
Blocksatz nicht verzichten möchte, bleibt deshalb häufig
nichts anderes übrig, als selbst ein solches Programm zu
entwickeln, was verständlicher Weise viel kostbare Zeit in
Anspruch nimmt.
```

### Blocksatzausdruck mit dem Druckprogramm

```

1: (*****
2: (* Proportional-Blocksatz-Druckprogramm fuer C. Itoh 8510/1550 *)
3: (* Version 2.2 / Datum 02.07.84 *)
4: (* TURBO-Pascal-Version / Apple-CP/M *)
5: (* Autor: A. Widmer *)
6: (*****
7:
8: PROGRAM ProportionalPrint (* PPRINT *);
9:
10: CONST
11:   chwidth: ARRAY (..'.'..'B'.) OF INTEGER =
12:
13: (*****
14: (* Die folgende Tabelle enthaelt die (proportionale) Breite der einzelnen *)
15: (* Zeichen in Punkten (Dots) ausgedrueckt. *)
16: (*****
17:
18: (* ! " # $ % & / ( ) * + , - . / 0 *)
19: (* 7, 10, 14, 12, 16, 13, 7, 7, 7, 12, 12, 7, 12, 7, 12, 12, *)
20: (* 1 2 3 4 5 6 7 8 9 : ; < = > ? s *)
21: (* 12, 12, 12, 12, 12, 12, 12, 12, 7, 7, 12, 12, 12, 12, *)
22: (* A B C D E F G H I J K L M N O P *)
23: (* 16, 15, 14, 15, 15, 15, 14, 15, 9, 13, 12, 13, 17, 16, 15, 13,

```

# GEWUSST WIE

## Haupt-Sequenz

Das Programm startet mit Proportional-Modus ausgeschaltet (Zeile 165). Der Benutzer wird zuerst nach dem Namen der zu druckenden Textdatei gefragt (172). Diese wird - sofern gefunden - geöffnet (175). Anschliessend wird der in der Datei enthaltene Text Zeichen für Zeichen eingelesen, wobei jedesmal gleich der zugehörige ASCII-Wert geprüft wird (182..183). Ergibt sich ein Wert, der grösser als 127 ist, so wird dieser um 128 erniedrigt (184). Danach werden alle druckbaren Zeichen zu Text-Zeilen zusammengestellt und durch Aufruf der Prozedur PrintLine ausgedruckt (185..190). Das Einlesen einzelner Zeichen ermöglicht somit das einfache Entfernen oder Umwandeln störender Zeichen, welche von Editoren wie WordStar in den Text eingestreut werden.

## Prozeduren PrintLine, CountCharacters

Enthält die eingelesene Zeile die Zeichenkette \$P+, so wird der Ausgleichalgorithmus aktiviert. Die dem «Switch» folgende, in Klammern gefasste Zahl bezeichnet die Blockbreite in Pica-Schrift (10 Zeichen/inch). Ist keine Angabe vorhanden, so wird der in der Konstante <deflinewidth> gespeicherte Wert als Blockbreite übernommen (114..123). Der Proportional-Blocksatz Modus wird durch \$P- wieder ausgeschaltet (113).

Befindet sich das Programm im Blocksatz-Modus, so werden zuerst alle am Anfang einer Zeile auftretenden Leerzeichen gezählt und anschliessend entfernt. Alle Leerzeichen die am Schluss der eingelesenen Zeile auftreten, werden ebenfalls entfernt, aber nicht gezählt (80..81, 84).

Der übriggebliebene, eigentliche Block wird nun Zeichen für Zeichen durchsucht, wobei alle Leerschläge gezählt werden. Bei den übrigen Zeichen wird die zugehörige Breite in Punkten (dots, pixel) in der Tabelle nachgeschlagen und aufsummiert. Die daraus resultierende Summe bezeichnet die Anzahl der durch druckbare Zeichen belegten Pixel in der Zeile (85..89).

Die noch freien Pixel werden nun auf die einzelnen Leerzeichen verteilt, wobei den Leerzeichen ein Durchschnittswert zugeordnet wird; der Rest wird vorläufig gespeichert (90..92).

```
24: (* Q R S T U V W X Y Z Ae De Ue ^ _ \ *)
25: 16, 15, 12, 14, 15, 16, 17, 11, 14, 11, 13, 13, 13, 12, 12, 0,
26: (* a b c d e f g h i j k l m n o p *)
27: 12, 12, 10, 12, 12, 10, 12, 12, 8, 7, 10, 8, 16, 12, 12, 12,
28: (* q r s t u v w x y z ae oe ue ß *)
29: 12, 10, 12, 10, 12, 12, 16, 12, 12, 10, 13, 13, 14, 0 );
30:
31: okset: SET OF CHAR = (' ' '..'β'.); (* druckbare Zeichen *)
32: deflinewidth = 64; (* vorgegebene Zeilenbreite *)
33: ignorechar = 'β'; (* wird nicht gedruckt *)
34: boldswitch = '^'; (* schaltet Fettschrift ein/aus *)
35: eolnchar = #13; (* ASCII-Code fuer <CR> *)
36: eofchar = #26; (* ASCII-Code fuer <Ctrl-Z> *)
37: nul = #0; (* ASCII-Code fuer <NUL> *)
38: escape = #27; (* ASCII-Code fuer <Escape> *)
39: boldon = '!'; (* schaltet Fettdruck ein *)
40: boldoff = '"'; (* schaltet Fettdruck aus *)
41: propon = 'P'; (* schaltet Proportionaldruck ein *)
42: propoff = 'N'; (* schaltet Proportionaldruck aus *)
43: graphicon = 'S'; (* schaltet Graphik ein *)
44:
45: VAR
46: line: STRING<.255.>; (* enthaelt den Zeilentext *)
47: inputchar: CHAR; (* enthaelt 1 Zeichen von infile *)
48: inputstring: STRING<.1.>; (* gleicher Inhalt wie inputchar *)
49: linewidth, (* Blockbreite in Zeichen *)
50: dotsperline, (* Blockbreite in Dots *)
51: numberofdots, (* durch Zeichen belegte Dots *)
52: spacesperline, (* Anz. Leerschlaege in der Zeile *)
53: guidingspaces, (* Leerschl. bis Blockbeginn *)
54: freespace, (* nicht belegte Dots *)
55: avspacewidth, (* Leerschlag-Breite Durchschnitt *)
56: spacecounter, (* Zaehler fuer Leerzeichen *)
57: spacerepeat, (* Repetitionszaehler fuer Dots *)
58: spaceoverhead: INTEGER; (* restliche Dots *)
59: boldmode, (* Schalter fuer Fettdruckmodus *)
60: propmode: BOOLEAN; (* Schalter fuer Proportionalmodus *)
61: filename: STRING<.14.>; (* Datei-Name *)
62: infile: TEXT; (* Sequentielle Textdatei *)
63:
64:
65: (*****
66: (* Folgende Prozedur schneidet bei der zu druckenden Textzeile hinten und *)
67: (* vorne alle ueberflussigen Leerzeichen ab, wobei alle einleitenden Leer- *)
68: (* schlaege gezaehlt werden. Anschliessend werden von allen uebrigen *)
69: (* Zeichen die entsprechenden Zeichenbreiten aufsummiert und am Schluss die *)
70: (* Durchschnittsbreite (in Punkten) eines Leerzeichens berechnet. *)
71: (*****
72:
73: PROCEDURE CountCharacters;
74: VAR
75: i: INTEGER;
76: c: CHAR;
77: BEGIN
78: guidingspaces := 0; numberofdots := 0; spacesperline := 0;
79: IF Length(line) > 0 THEN
80: WHILE Copy(line,1,1) = ' ' DO
81: BEGIN Delete(line,1,1); guidingspaces := guidingspaces + 1 END;
82: IF Length(line) > 0 THEN
83: BEGIN
84: WHILE Copy(line,Length(line),1) = ' ' DO Delete(line,Length(line),1);
85: FOR i := 1 TO Length(line) DO
86: BEGIN c := line(i.);
87: IF c = ' ' THEN spacesperline := spacesperline + 1
88: ELSE numberofdots := numberofdots + chwidth(c.);
89: END;
90: freespace := dotsperline - numberofdots;
91: avspacewidth := freespace DIV spacesperline;
92: spaceoverhead := freespace MOD spacesperline;
93: END;
94: END;
95:
96:
97: (*****
98: (* Die Prozedur PrintLine druckt jeweils eine Linie , wobei sie auch zur *)
99: (* Aufgabe hat, die einzelnen Modi ein- und auszuschalten. *)
100: (* Ist das zu druckende Zeichen ein Leerschlag, so wird auf den Graphik- *)
101: (* modus umgeschaltet und ein entsprechend grosser Abstand in Form von *)
102: (* CHR(0) gesendet. Beim Auftauchen eines Fettdruck-Control-Zeichen wird *)
103: (* eine Escape-Sequenz zum Drucker gesendet, die den Fettdruck ein- oder *)
104: (* ausschaltet. Das 'Ignore'-Zeichen wird nicht gesendet. *)
105: (*****
106:
107: PROCEDURE PrintLine;
108: VAR
109: i, j, mark1, mark2, code: INTEGER;
110: c, h: CHAR;
```

```

111: p: STRING(4.); (* zur Drucker-Initialisierung *)
112: BEGIN
113: IF Pos('$P-',line) > 0 THEN BEGIN propmode := FALSE; line := '' END;
114: IF Pos('$P+',line) > 0 THEN
115: BEGIN mark1 := Pos('(',line); mark2 := Pos(')',line);
116: IF (mark1 = 0) OR (mark2 = 0) OR (mark1 > mark2)
117: THEN linewidth := deflinewidth
118: ELSE BEGIN Val(Copy(line,mark1+1,mark2-mark1-1),linewidth,code);
119: IF code <> 0 THEN linewidth := deflinewidth;
120: END;
121: dotsperline := linewidth * 17;
122: propmode := TRUE; line := '';
123: END;
124: IF NOT propmode THEN WriteLn(1st,line)
125: ELSE
126: BEGIN spacecounter := 0;
127: CountCharacters;
128: FOR i := 1 TO guidingspaces DO Write(1st,' ');
129: Write(1st,escape,propoff);
130: FOR i := 1 TO Length(line) DO
131: BEGIN c := line(i.);
132: IF c = boldswitch THEN
133: IF boldmode
134: THEN BEGIN boldmode := FALSE; Write(1st,escape,boldoff) END
135: ELSE BEGIN boldmode := TRUE; Write(1st,escape,boldon) END;
136: IF (c <> ignorechar) AND (c <> boldswitch) THEN
137: IF c <> ' ' THEN Write(1st,c)
138: ELSE
139: BEGIN spacecounter := spacecounter + 1;
140: IF spacecounter > spaceoverhead
141: THEN spacerepeat := avspacewidth
142: ELSE spacerepeat := avspacewidth + 1;
143: Str(spacerepeat,4,p);
144: FOR j := 1 TO 4 DO
145: BEGIN h := p(j.);
146: IF NOT (h IN ('0'..'9')) THEN p(j.) := '0';
147: END;
148: Write(1st,escape,graphicon,p);
149: FOR j := 1 TO spacerepeat DO Write(1st,nul);
150: END;
151: END;
152: WriteLn(1st); Write(1st,escape,propoff);
153: END;
154: END;
155:
156:
157: (*****
158: (* In der Hauptsequenz wird die Datei geöffnet und alle vorhandenen Zeilen *)
159: (* eingelesen und durch Aufruf von PrintLine ausgedruckt. *)
160: (*****
161:
162: (* Main *)
163: BEGIN
164: ClrScr;
165: propmode := FALSE; boldmode := FALSE;
166: inputstring := '';
167: WriteLn('*****');
168: WriteLn('* Proportional Print Program for C. Itoh 8510/1550 (V2.2) *');
169: WriteLn('*****');
170: REPEAT
171: WriteLn; WriteLn;
172: Write('Filename (</E> to End): '); ReadLn(filename);
173: IF filename <> '/E' THEN
174: BEGIN Assign(infile,filename);
175: ((*I-*) Reset(infile) ((*I+*);
176: IF IOResult <> 0 THEN WriteLn(' File not Found!')
177: ELSE
178: BEGIN Write('Working...');
179: WHILE NOT Eof(infile) DO
180: BEGIN line := '';
181: REPEAT
182: Read(infile,inputchar);
183: IF ORD(inputchar) > 127 THEN
184: inputchar := CHR(ORD(inputchar)-128);
185: IF inputchar IN okset THEN
186: BEGIN inputstring(1.) := inputchar;
187: line := Concat(line,inputstring);
188: END;
189: UNTIL inputchar IN (.eolnchar,eofchar.);
190: PrintLine;
191: END;
192: Close(infile);
193: END;
194: END;
195: UNTIL filename = '/E';
196: WriteLn; WriteLn('End Run!');
197: END.

```

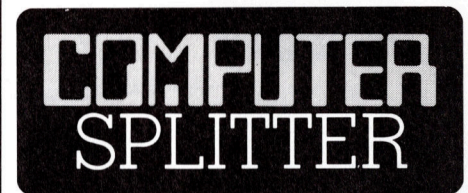
In der Folge wird in der Prozedur PrintLine die vollständig aufbereitete Textzeile ein letztes Mal zeichenweise durchgearbeitet. Dabei werden, je nach Zeichen, verschiedene Funktionen aktiviert:

Stösst das Programm auf ein «boldswitch»-Zeichen, so wird der angeschlossene Drucker in einen temporären Fettdruck-Modus versetzt, welcher ein erneutes Auftreten des gleichen Zeichens wiederum ausgeschaltet wird (132..135).

Wird ein «ignore»-Zeichen im Text gelesen, so wird dieses kurzerhand übersprungen, denn dieses Zeichen dient ausschliesslich der Rand-Justage, d.h. es dient dazu, Leerzeichen am Block-Anfang und am Block-Ende vor dem Löschen zu schützen (136).

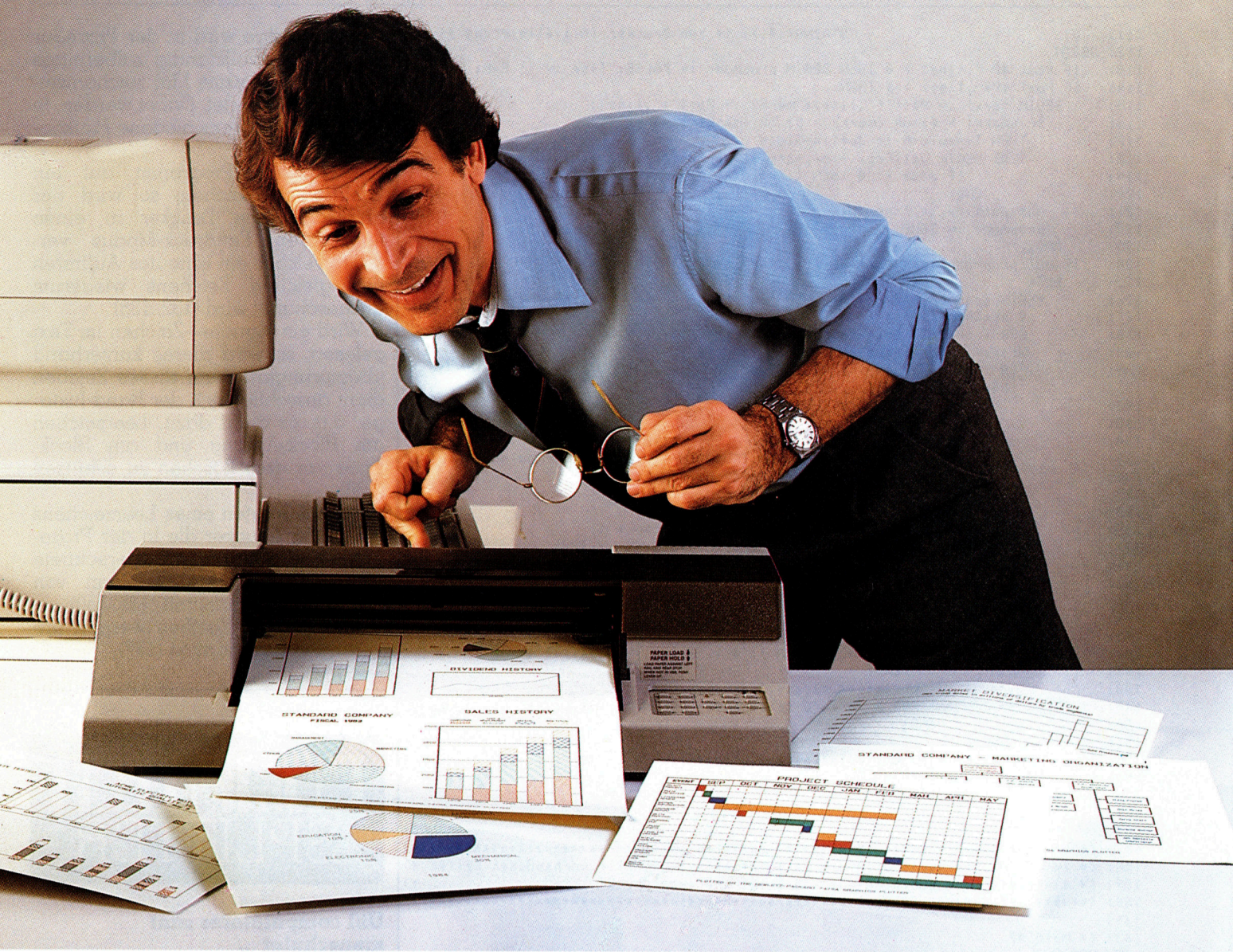
Beim Auftreten eines Leerzeichens erhält der Drucker die in der Prozedur CountCharacters berechnete Anzahl Leerschritte in Form von Grafikdaten übermittelt (139..150).

Alle übrigen Zeichen werden unverändert ausgedruckt (137). □



## Ubi computatores sunt menschulat ...

(286/fp) 1984: Ist es nicht tröstlich, auch voll computerisierte Firmen sind in vielen Bereichen noch weit weg von der Perfektion, die Computer noch weit weg von der Machtübernahme über uns Menschen. Da meldet der Schreibende auf einem wohlvorbereiteten Formular und unter Beifügung aller möglichen - bis zu 20-stelligen - Codes seinen Adresswechsel: Behörden, Banken, Versicherungen, Versandhäuser für Baby-Wäsche und Gartenbau-Bücher, die edlen Spender persönlicher Glücksnummern und diverse Computer-Firmen sind die Empfänger von weit mehr als hundert solcher Meldungen. Beim Sanitär-Monteur mit seiner Handkartei klappt die Mutation ohne Umstände, andere brauchen etwas Nachdruck: Den Rekord hält IBM mit bisher erfolglosen vier schriftlichen Meldungen und einem noch hängigen Telefongespräch, dicht gefolgt von time-life-Books, Olivetti und HP mit drei Meldungen ... □



## Jetzt kann Ihr Personal Computer zeigen, was er zu sagen hat.

Denn mit den neuen Grafik-Plottern von Hewlett-Packard können Sie Grafiken und Schaubilder Ihres PC mehrfarbig erstellen. Dadurch können auch komplizierte Fakten und Zahlen deutlich und eindrucksvoll dargestellt werden – sauber, mühelos und vor allem schnell.

So werden Trends und Zusammenhänge leichter erkennbar, Daten können genauer analysiert werden, Berichte, Präsentationen und Dokumentationen werden übersichtlicher – sei es auf Papier oder auf Folie.

Sie können wählen zwischen einem 6-Farb-Plotter (bis zum Format A3) und einem 2-Farb-Plotter (A4). Beide Modelle sind ausgesprochen preisgünstig. Dabei bieten sie die weltweit bekannte, hervorragende Leistung und Qualität von Hewlett-Packard.

Beide Plotter lassen sich an alle gängigen Personal Computer anschliessen. Egal, welchen PC Sie verwenden, ein Plotter von Hewlett-Packard zeigt und dokumentiert, was wirklich in ihm steckt – und in Ihrer Arbeit. Indem er die Ergebnisse einfach zeichnet.

Ich möchte wissen, was wirklich in meinem PC steckt – bitte schicken Sie mir Informationsmaterial über die neuen Grafik-Plotter von Hewlett-Packard.

(Name) \_\_\_\_\_ (Tel.) \_\_\_\_\_

(PLZ/Ort) \_\_\_\_\_ (Strasse) \_\_\_\_\_

(Firma/Branche) \_\_\_\_\_ (Funktion) \_\_\_\_\_

Hewlett-Packard (Schweiz) A.G., Abteilung Information,  
Allmend 2, CH-8967 Widen. **M+K 84-12**

# 3D-Darstellung einer Kugel mit Längen- und Breitenkreisen mit Hidden-Line-Routine

In M+K 83-4 wurde im Artikel «Programmieren mit hochauflösender Grafik» ein Programm vorgestellt, mit dem eine Kugel samt Längen- und Breitenkreisen in schiefer Parallelprojektion dargestellt werden kann. Es ist dies das Programm 25 in meinem Buch «Programmieren mit hochauflösender Grafik».

Das Kugelprogramm hat leider zwei Schwächen: 1.) Bei der angewendeten Parallelprojektion ergeben sich für den Projektionswinkel  $\alpha \neq 90^\circ$  unschöne Verzerrungen. 2.) Die nicht sichtbaren Teile der Längen- und Breitenkreise sind ausgezogen, obwohl sie unterdrückt sein sollten.

**Marcel Sutter**

Eine dreidimensionale Darstellung einer Kugel ist erst dann befriedigend, wenn sie aus jeder Blickrichtung unverzerrt und mit einer Hidden-Line-Routine versehen ist.

Da das Kugelprogramm offenbar auf grosses Interesse stösst, will ich ein neues Programm vorstellen, das alle Schwächen des Vorgängers beseitigt. Es basiert allerdings auf einem anderen Algorithmus und erfordert weitergehende mathematische Kenntnisse als das erste Programm. Daher will ich zuerst die theoretischen Grundlagen bereit stellen.

## Theorie

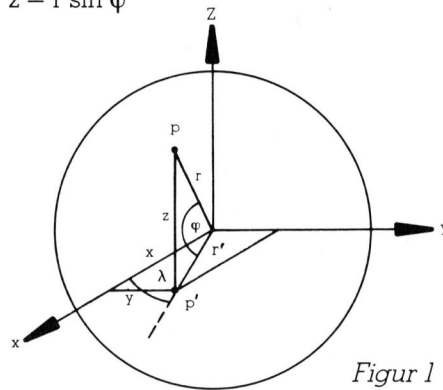
Die Kugel mit dem Radius  $r$  habe ihren Mittelpunkt im Nullpunkt eines räumlichen Koordinatensystems. Ein Punkt  $P(x/y/z)$  der Kugeloberfläche erfüllt die bekannte Kugelgleichung

$$x^2 + y^2 + z^2 = r^2$$

Wie ich schon im Artikel M+K 83-4 gezeigt habe, ist es für die Programmierung der Kugeldarstellung vorteilhafter, räumliche Polarkoordinaten, sogenannte Kugelkoordinaten einzuführen. Drehen wir den Vektor  $r$ , dessen Anfangspunkt im Nullpunkt und dessen Spitze auf der  $x$ -Achse liegt, im positiven Sinn um den Winkel  $\lambda$  ( $0^\circ \leq \lambda \leq 360^\circ$ ) in der  $xy$ -Ebene und danach um den Winkel  $\varphi$  ( $-90^\circ \leq \varphi \leq 90^\circ$ ) aus der  $xy$ -Ebene heraus, dann zeigt die Spitze des

Vektors auf einen Punkt  $P(x/y/z)$  der Kugeloberfläche. Wie man leicht aus Figur 1 abliest, gelten die Formeln

$$\begin{aligned} x &= r' \cos \lambda = r \cos \varphi \cos \lambda \\ y &= r' \sin \lambda = r \cos \varphi \sin \lambda \\ z &= r \sin \varphi \end{aligned}$$



Figur 1

Dabei bedeutet  $r'$  die senkrechte Projektion von  $r$  auf die  $xy$ -Ebene. Im nachfolgenden BASIC-Programm erscheinen die Formeln in der Form

$$\begin{aligned} X &= R * \text{COS}(P * \text{BM}) * \text{COS}(L * \text{BM}) \\ Y &= R * \text{COS}(P * \text{BM}) * \text{SIN}(L * \text{BM}) \\ Z &= R * \text{SIN}(P * \text{BM}) \end{aligned}$$

$\text{BM} = \pi/180$  ist der Umrechnungsfaktor vom Gradmass ins Bogenmass.

Den Nordpol wählen wir auf der positiven  $z$ -Achse. Alle Längenkreise sind Grosskreise durch den Nord- und Südpol, während die Breitenkreise parallel zur  $xy$ -Ebene liegen.

Dreht man jetzt die Kugel um den Winkel  $\alpha$  um die  $x$ -Achse, danach um den Winkel  $\beta$  um die  $y$ -Achse und zum Schluss um den Winkel  $\gamma$  um die  $z$ -Achse, dann hat sich die Kugel im

Raum um den Nullpunkt gedreht. Der Nordpol liegt daher nicht mehr auf der  $z$ -Achse, sondern irgendwo auf der Kugeloberfläche. Ebenso sind jetzt die Breitenkreise nicht mehr parallel zur  $xy$ -Ebene.

Ein Punkt  $P(x/y/z)$  geht bei dieser «dreifachen Drehung» in den Punkt  $P'(x'/y'/z')$  über.

## Transformationsformeln für die Drehung

$$\begin{aligned} x' &= \alpha_{11} x + \alpha_{12} y + \alpha_{13} z \\ y' &= \alpha_{21} x + \alpha_{22} y + \alpha_{23} z \\ z' &= \alpha_{31} x + \alpha_{32} y + \alpha_{33} z \end{aligned}$$

Dabei berechnen sich die Koeffizienten  $\alpha_{jk}$  wie folgt:

$$\begin{aligned} \alpha_{11} &= \cos \beta \cos \gamma \\ \alpha_{12} &= -\cos \beta \sin \gamma \\ \alpha_{13} &= \sin \beta \\ \alpha_{21} &= \cos \alpha \sin \gamma + \sin \alpha \sin \beta \cos \gamma \\ \alpha_{22} &= \cos \alpha \cos \gamma - \sin \alpha \sin \beta \sin \gamma \\ \alpha_{23} &= -\sin \alpha \cos \beta \\ \alpha_{31} &= \sin \alpha \sin \gamma - \cos \alpha \sin \beta \cos \gamma \\ \alpha_{32} &= \sin \alpha \cos \gamma + \cos \alpha \sin \beta \sin \gamma \\ \alpha_{33} &= \cos \alpha \cos \beta \end{aligned}$$

Wer mit der Matrixschreibweise vertraut ist, erkennt unschwer die elegantere Darstellung:

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} \\ \alpha_{21} & \alpha_{22} & \alpha_{23} \\ \alpha_{31} & \alpha_{32} & \alpha_{33} \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} \text{ oder } R' = A \cdot R$$

$R$  ist der Ortsvektor von  $P(x/y/z)$  und  $R'$  ist der Ortsvektor von  $P'(x'/y'/z')$ .  $A$  heisst die sogenannte Rotationsmatrix.

Auf eine Herleitung der Formeln verzichte ich. Sie können diese aber zum Beispiel in der kleinen Enzyklopädie der Mathematik nachlesen. Von dort habe ich auch obige Formeln übernommen.

Im nachfolgenden BASIC-Programm erscheinen die Transformationsformeln in folgender Form:

$$\begin{aligned} S1 &= \text{SIN}(A * \text{BM}) : S2 = \text{SIN}(B * \text{BM}) \\ &: S3 = \text{SIN}(C * \text{BM}) \\ C1 &= \text{COS}(A * \text{BM}) : C2 = \text{COS}(B * \text{BM}) \\ &: C3 = \text{COS}(C * \text{BM}) \end{aligned}$$

$$\begin{aligned} AX &= C2 * C3 : AY = -C2 * S3 : AZ = S2 \\ BX &= C1 * S3 + S1 * S2 * C3 \\ &: BY = C1 * C3 - S1 * S2 * S3 : BZ = S1 * C2 \\ CX &= S1 * S3 - C1 * S2 * C3 \\ &: CY = S1 * C3 + C1 * S2 * S3 \\ &: CZ = C1 * C2 \end{aligned}$$

$$\begin{aligned} XX &= AX * X + AY * Y + AZ * Z \\ YY &= BX * X + BY * Y + BZ * Z \\ ZZ &= CX * X + CY * Y + CZ * Z \end{aligned}$$

Nächsten Monat gibt's wieder

**COMPUTER  
MARKT**

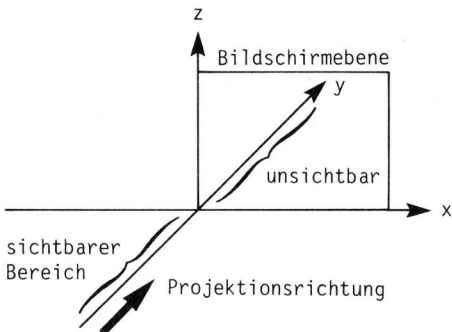
Abonnement schon bestellt?

# GEWUSST WIE

A, B, C sind die eingelesenen Drehwinkel  $\alpha$ ,  $\beta$ ,  $\gamma$  im Gradmass und XX, YY, ZZ die Koordinaten  $x'$ ,  $y'$ ,  $z'$  des gedrehten Punktes P.

Nachdem wir die Kugel gedreht haben, müssen wir sie nur noch auf die Bildelebene projizieren.

Damit keine Verzerrungen eintreten, wählen wir als Bildelebene die xz-Ebene und als Projektionsrichtung die y-Achse (siehe dazu die Figur 2). In der Zeichnung ist das Koordinatensystem um  $90^\circ$  gegenüber Figur 1 verdreht gezeichnet.



Figur 2

Wir sehen sofort ein, dass bei dieser speziellen Wahl der Projektionsrichtung die XX- und die ZZ-Koordinaten des gedrehten Punktes unverändert bleiben. Auf dem Bildschirm ist allerdings  $YY=0$ .

Seien U und V wie üblich die Mittelpunktswinkel des HRG-Bildschirms (beim C-64 ist  $U=160$  und  $V=100$ ), dann lauten in BASIC die Bildschirmkoordinaten XB, YB des projizierten Punktes P(XX/YY/ZZ), nachdem sie noch auf ganze Zahlen gerundet werden

$$XB = \text{INT}(U+XX+0.5)$$

$$YB = \text{INT}(V-ZZ+0.5)$$

## Hidden-Line-Routine

Wie unterdrücken wir jetzt die nicht sichtbaren Linienteile der Längen- und Breitenkreise?

Das ist bei dieser speziellen Projektionsart äusserst einfach. Hat ein gedrehter Punkt P(XX/YY/ZZ) eine negative YY-Koordinate, so liegt er vor der Bildelebene und ist somit sichtbar (siehe wieder Figur 2). Kriterium:

$$YY < 0 \quad \text{Punkt ist sichtbar}$$

$$YY > 0 \quad \text{Punkt ist unsichtbar}$$

Nur der Vollständigkeit halber sei erwähnt, dass man auch längs der x-Achse auf die yz-Ebene oder längs der z-Achse auf die xy-Ebene projizieren kann. In der neuen Branche «Computer aided Design» wird meist

die zuletzt erwähnte Projektionsart verwendet.

Damit sind alle theoretischen Grundlagen für das Verständnis des Programms bereit gestellt. Halten wir nochmals den unterschiedlichen Algorithmus der beiden Programme fest:

### 1. Kugelprogramm

Die Kugel steht im Raum fest. Die Projektionsrichtung wird frei gewählt.

### 2. Kugelprogramm

Die Kugel wird beliebig im Raum um ihren Mittelpunkt gedreht. Die Projektionsrichtung ist fest, nämlich parallel zur y-Achse.

Das folgende Programm (Listing 1) ist für den C-64 geschrieben. Für die hochauflösende Grafik verwende ich das weit verbreitete SIMONS BASIC. Das Programm ist strukturiert und top-down geschrieben und kann daher leicht auf andere Mikrocomputer, z.B. Apple umgeschrieben werden.

Im Programm treten zwei spezielle Variable F1 und F2 auf. Es sind sogenannte Flaggen, die zur Steuerung des Zeichnens dienen:

F1=0  
P(XB/YB) ist Anfangspunkt einer Strecke

Listing 2: Programm für den Sharp PC-1500

```

10: TEXT : COLOR 0:           ";D
   CSIZE 1
20: LPRINT "KUGEL           120: GRAPH ;
   MIT HIDDEN-LIN           GLCURSOR (108,
   E-ROUTINE"              -120):SORGN
30: LF 2
40: INPUT "DREHUNG           130: DEGREE :R=100:
   UM X-ACHSE=";           H=0.5
   A
50: INPUT "DREHUNG           140: S1=SIN A;S2=
   UM Y-ACHSE=";           SIN B;S3=SIN C
   B                       :C1=COS A;C2=
60: INPUT "DREHUNG           150: AX=C2*C3;AY=-C
   UM Z-ACHSE=";           *S3;AZ=S2
   C                       160: BX=C1*S3+S1*S2
70: INPUT "LINIENA           *C3;BY=C1*C3-S
   BSTAND=";D              1*C2*S3;BZ=-S1
80: LPRINT "DREHUN           *C2
   G UM X-ACHSE =
   ";A
90: LPRINT "DREHUN           170: CX=S1*S3-C1*S2
   G UM Y-ACHSE =
   ";B                       *C3;CY=S1*C3+C
100: LPRINT "DREHUN          180: FOR W=0TO 360
   G UM Z-ACHSE =
   ";C                       STEP 8
110: LPRINT "ABSTAN          190: X=INT (R*COS W
   D DER LINIEN =          +H);Y=INT (R*
                               SIN W+H)
                               200: IF W=0THEN
                               GLCURSOR (X,Y)
                               210: IF W>0THEN

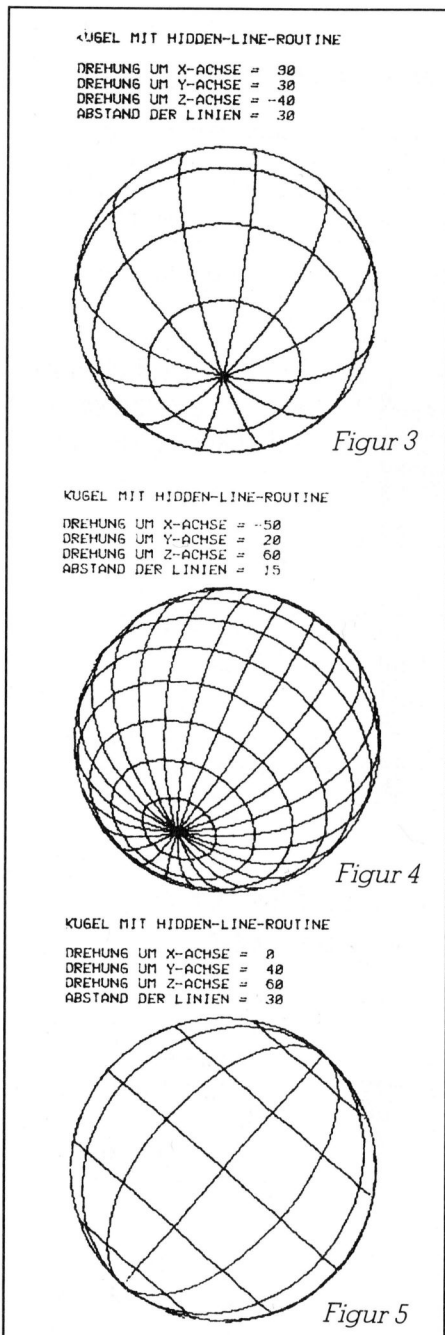
```



F1=1  
P(XB/YB) ist Endpunkt einer Strecke  
F2=0  
P(XB/YB) ist unsichtbar  
F2=1  
P(XB/YB) ist sichtbar

Wie man mit diesen Flaggen steuern kann, wann Strecken, d.h. Liniensegmente gezeichnet und wann unterdrückt werden, habe ich in den Artikeln in M+K 83-2 und 83-3 sowie in meinem Buch ausführlich erklärt.

Da der PC-1500 von Sharp offenbar weit verbreitet ist und ich überdies alle hier abgebildeten Kugeldarstellungen auf dem Miniplotter dieses Gerätes hergestellt habe, will ich auch das Programm für den PC-1500 herauslisten. (Listing 2).



Listing 1: Programm für den C-64

```

100 PRINT CHR$(147)
105 :
110 PRINT "KUGEL MIT LAENGEN- UND BREITENKREISEN"
120 PRINT "-----"
130 PRINT:PRINT
200 INPUT "DREHWINKEL UM X-ACHSE":A:PRINT
210 INPUT "DREHWINKEL UM Y-ACHSE":B:PRINT
220 INPUT "DREHWINKEL UM Z-ACHSE":C:PRINT
225 INPUT "LINIENABSTAND (10-45)":D
230 U=160:V=100:R=98:BM=PI/180:H=0.5
240 S1=SIN(A*BM):S2=SIN(B*BM):S3=SIN(C*BM)
250 C1=COS(A*BM):C2=COS(B*BM):C3=COS(C*BM)
255 :
260 REM ROTATIONSMATRIX BERECHNEN
270 AX=C2*C3:AY=-C2*S3:AZ=S2
280 BX=C1*S3+S1*S2*C3
290 BY=C1*C3-S1*S2*S3:BZ=-S1*C2
300 CX=S1*S3-C1*S2*C3
310 CY=S1*C3+C1*S2*S3:CZ=C1*C2
315 :
320 REM FARBEN WAELHEN, HRG EINSCHALTEN
325 :
330 POKE 53280,3:POKE 53281,3:HIRES 0,3
335 :
400 REM KUGELUMRISS ZEICHNEN
410 CIRCLE U,V,R,R,1
415 :
500 REM LAENGENKREISE ZEICHNEN
510 FOR L=0 TO 180-D STEP D:F1=0
520 :   FOR P=0 TO 360 STEP 5
530 :   GOSUB 1000:REM XX,YY,ZZ BERECHNEN
540 :   IF YY>0 THEN F2=0:F1=0:GOTO 580
550 :   XB=INT(U+XX+H):YB=INT(V-ZZ+H):F2=1
560 :   IF F1=0 THEN X1=XB:Y1=YB:F1=1:GOTO 580
570 :   LINE X1,Y1,XB,YB,1:X1=XB:Y1=YB:F1=F2
580 :   NEXT P
590 NEXT L
595 :
600 REM BREITENKREISE ZEICHNEN
610 FOR P=-90+D TO 90-D STEP D:F1=0
620 :   FOR L=0 TO 360 STEP 5
630 :   GOSUB 1000:REM XX,YY,ZZ BERECHNEN
640 :   IF YY>0 THEN F2=0:F1=0:GOTO 680
650 :   XB=INT(U+XX+H):YB=INT(V-ZZ+H):F2=1
660 :   IF F1=0 THEN X1=XB:Y1=YB:F1=1:GOTO 680
670 :   LINE X1,Y1,XB,YB,1:X1=XB:Y1=YB:F1=F2
680 :   NEXT L
690 NEXT P
695 :
700 GET A$:IF A#="" THEN 700
710 END
715 :
999 REM KUGELKOORDINATEN -> KARTESISCHE KOORDINATEN
1000 X=R*COS(P*BM)*COS(L*BM)
1010 Y=R*COS(P*BM)*SIN(L*BM)
1020 Z=R*SIN(P*BM)
1025 :
1030 REM P(X/Y/Z) IN P(XX/YY/ZZ) DREHEN
1040 XX=AX*X+AY*Y+AZ*Z
1050 YY=BX*X+BY*Y+BZ*Z
1060 ZZ=CX*X+CY*Y+CZ*Z
1100 RETURN

```

# GEWUSST WIE

Die Figuren 3, 4 und 5 zeigen Kugeln in dreidimensionaler Darstellung, die mit obigem Programm erzeugt wurden. Die Drehwinkel  $\alpha$ ,  $\beta$  und  $\gamma$  um die entsprechenden Koordinatenachsen sind angegeben.

## Ausbau des Programms

Sie können, wenn Sie sehr fleissig sind, das Kugelprogramm ausbauen.

Nehmen Sie einen geeigneten Atlas zur Hand und lesen Sie die geografische Länge  $\lambda$  und die geografische Breite  $\varphi$  von sehr vielen Punkten längs der Grenze von Eurasien, Nord- und Südamerika, Afrika, Australien und den grossen Inseln heraus. Diese Zahlenpaare schreiben Sie nacheinander in DATA-Zeilen.

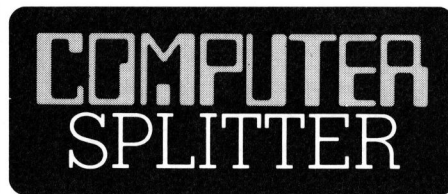
Nachdem der Computer die gedrehte Erdkugel samt Längen- und Breitenkreisen (Schrittweite  $30^\circ$ ) gezeichnet hat, liest er der Reihe nach die Zahlenpaare ( $\lambda$ ,  $\varphi$ ) ein, rechnet sie in XX, YY, ZZ und dann in XB, YB um. Falls von zwei aufeinanderfolgenden Punkten  $P_1$  und  $P_2$  jeweils  $YY <= 0$  ist, wird die Strecke  $P_1P_2$  gezeichnet.

Auf diese Art erhalten Sie eine Darstellung der Erde mit den sicht-

baren Erdteilen von jenem Punkt aus dem Weltraum, der sich auf Grund der Wahl von  $\alpha$ ,  $\beta$  und  $\gamma$  ergibt.

Dieses Programm samt den umfangreichen DATA-Zeilen werde ich Ihnen in einer folgenden M+K-Ausgabe vorstellen.

Wenn Sie dann noch das Meer in Blau und die Kontinente in anderer Farbe zeichnen, was beim C-64 mit SIMONS BASIC sehr leicht ist, erhalten Sie ein Top-Programm, welches Ihnen und Ihren Zuschauern viel Freude bereitet.  $\square$



## Mikrocomputer-Jahrbuch 1985

(287/fp) Systematischer, dicker, teurer ist es geworden, das Mikrocomputer-Jahrbuch von Harald Schumny beim Vieweg-Verlag. Und trotz des massiven Preises von Fr. 38.- ist seine Lektüre schon ein Muss.

Inhaltlich finden wir einleitend die fundierten Entwicklungsübersichten und -perspektiven, gefolgt von Marktübersichten. Es folgen einführende oder auf solidem Grundwissen basierende Artikel zu Software-Themen, so z.B. ein exquisiter Aufsatz zur Programmiersprache LOGO. Ein eigenes Kapitel ist dem Computer-Einsatz in der Ausbildung gewidmet. Die Programmsammlung enthält im Gegensatz zu früheren Jahren besser dokumentierte Programme zu unterschiedlichen Themen. Es folgen interessante Knobelthemen und abschliessend finden wir die wertvollen Datensammlungen und Bibliografien. Konzeptionell immer noch nicht ausgereift finden wir die thematische Gruppierung, die Aufteilung sowie Kriterien zur Auswahl der Artikel.  $\square$

## Philips baut auf MSX

(291/fp) Philips beginnt in Frankreich mit der Produktion von mindestens zwei Modellen Heimcomputer mit dem MSX-Betriebssystem. Die Firma hat sich damit als erster Europäer für den MSX-Standard entschieden.  $\square$



**C/NORM**  
Versand  
Melonenstrasse, 9001 St.Gallen

ENORME HITPREISE

C	Commodore 64	750.-	655.-
	Floppy 1541	790.-	720.-
brother	CE-50	948.-	795.-
	Schreibmaschine		
EPSON	RX-80	995.-	895.-
	Matrix-Drucker		

Bei C/Norm erhalten Sie Apricot, Brother, Commodore, Epson, DEC, Sirius/Victor und Disketten, Papier, Etiketten, Farb-bänder usw. zu vorteilhaften Versandpreisen. Die Lieferung erfolgt per NN oder Vorkasse ab Lager St.Gallen

071 22 63 22



Mit Programm-Linien für VC-20, C-64, PC-1500 und Apple II

Ein leicht verständlicher Lehrgang anhand von 40 Grafik-Programmen

288 Seiten, 72 Abb., Fr./DM 45.-  
ISBN 3-907007-02-6

Der interessierte Computer-Anwender wird schrittweise in das Programmieren mit HRG eingeführt. Die vorgestellten BASIC-Programme umfassen meist weniger als 30 Zeilen, sind strukturiert, können top-down gelesen werden, sind selbsterklärend und lassen sich für jedes Computer-System adaptieren.

Bestellungen über die nächste Buchhandlung oder direkt beim Verlag

MIKRO+KLEINCOMPUTER  
INFORMA VERLAG AG

Postfach 1401, CH-6000 Luzern 15

## V 24/RS 232 Interface-Tester/Konfigurator



- «Tri-State»-LEDs
- beliebige Kreuzverbindungen mit Minischaltern und Brücken
- Doppelstecker positiv und negativ auf jeder Seite, usw.

TELTEC HESS, 3250 Lyss

Knospweg 4, Tel. (032) 84 42 40, Telex 34 446

**DIE PERSONAL COMPUTER MIT DEM EUROPÄISCHEN KNOW-HOW.**



## **FREUNDLICH & KOMPATIBEL.**

Auch wenn Sie sie noch nicht kennen: die neuen Personal Computer von Olivetti kennen sich bereits in Ihrem Unternehmen aus, mit Ihren Aufgaben, Ihren Problemen und damit, wie man sie am besten löst.

Wie das? In ihnen steckt die Erfahrung, die Olivetti als grösster europäischer Computerhersteller im Kontakt mit tausenden von Unternehmen gewinnt, Tag für Tag, beim Nachdenken über Lösungen, die eben speziell auf die Wünsche, Gewohnheiten und Erwartungen europäischer Anwender eingehen.

Olivetti M24 und M21 sind europäische Personal Computer. Hardware und Software sind kompatibel zum

Industriestandard, bieten aber mehr. Mehr Leistung, weil sie Daten und Texte schneller verarbeiten und von den Zahlen schneller zur Grafik kommen.

Mehr Spielraum, weil sie nicht nur für jede Aufgabe die richtige Kapazität bieten, sondern auch in ein lokales Netz eingefügt oder dazu ausgebaut werden können.

Mehr Zukunft, weil Sie mit den neuen Personal Computern von Olivetti einfach weitermachen können, wenn Ihr Unternehmen wächst. Weiter aufbauen. Und weiterkommen.

Mit Olivetti.  
Der europäischen Alternative.

# **olivetti**

Bitte schicken Sie diesen Coupon an  
Olivetti Verkaufs-AG, PC-Informations-Center, Kernstrasse 37,  
8004 Zürich (Tel. 01 241 83 09)  
Senden Sie mir bitte mehr Information über die  
neuen Olivetti Personal Computer.  
Name: \_\_\_\_\_  
Firma: \_\_\_\_\_  
Strasse: \_\_\_\_\_  
PLZ, Ort: \_\_\_\_\_

# micom ag micro computer systeme

Zugerstrasse 64 8810 Horgen Telefon 01 725 5010



**commodore**  
COMPUTER



**IBM**

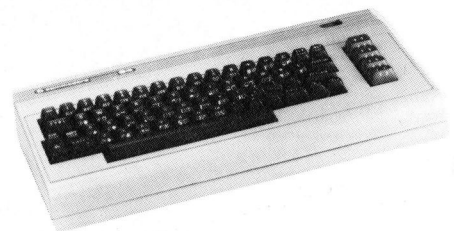
SEIKOSHA

**TAXAN**

**olivetti**



TEXAS  
INSTRUMENTS



IHR  
PARTNER  
FÜR INDIVIDUELLE  
COMPUTER-  
LÖSUNGEN

**hp** HEWLETT  
PACKARD

**RAIR**



**star**

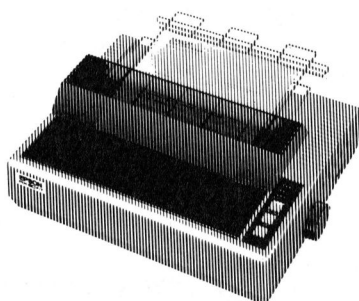
EPSON

**SVI**<sup>TM</sup>  
SPECTRAVIDEO

**brother**

COLUMBIA

**SANYO**



Ausserdem erhalten  
Sie bei uns:

- Büromöbel
  - Verbrauchsmaterial
- und natürlich auch die
- fachmännische Beratung  
dazu



## MUK-Test's (M+K 82-4 und 84-1)

Durch eine einfache Aenderung im Test MUK4 liess sich mein ins Pensionsalter kommender Superbrain zu nahezu CDC-Cyber-Genauigkeit aufstacheln. Das Rezept: Man muss dafür sorgen, dass in der Rechenformel nur mit Doppel-Präzision gearbeitet wird. Für die Variablen ist im Original-Testprogramm dafür gesorgt. Neu werden auch die Zahlenkonstanten (Zeile 250) mit der Definition # versehen. Das Resultat spricht für sich: Ohne grösseren Zeitaufwand stimmt das Resultat bis auf die letzte Stelle mit Cyber überein! In MUK3 bringt das Rezept keinen Erfolg. Das Resultat ist genau so schlecht wie in M+K 84-1 publiziert. Der Grund dafür liegt darin, dass die sogenannten «transzendenten» Funktionen (SIN, COS, TAN, ATN, LOG, EXP, SQR sowie Potenzen) vom Interpreter nur mit einfacher Genauigkeit berechnet werden können.

Der Microsoft-Compiler rechnet doppelt genau, wie das ausgedruckte Resultat beweist. Der Superbrain erreicht wieder fast Cyber-Präzision - ist er doch noch nicht pensionsreif? Ernüchternd wirkt allerdings der enorme Zeitbedarf der sonst so schnellen Compiler-Rechnung. Sie arbeitet siebeneinhalbmal langsamer als der Interpreter. Die höheren Funktionen doppelter Genauigkeit sind also harte Brocken für den Superbrain.

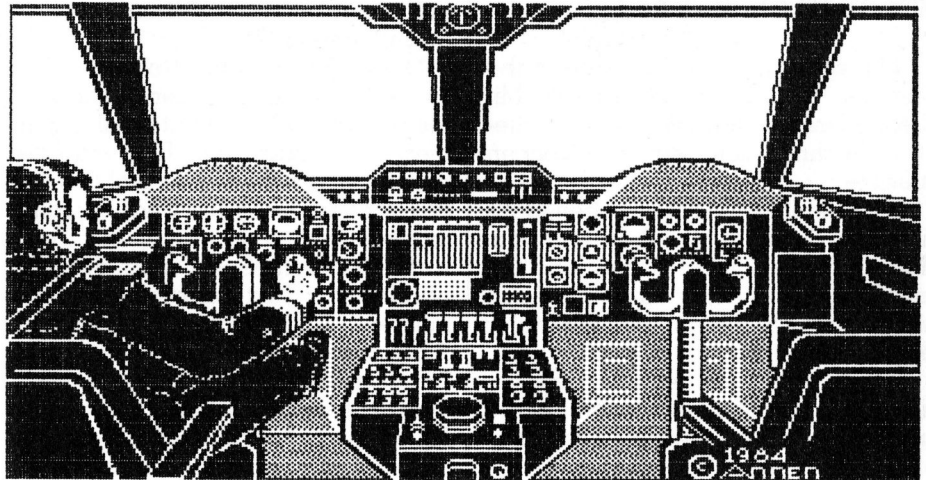
Die vorgestellten Tests verdichten den Verdacht, dass die jetzt gültigen MUK3- und MUK4-Rechnungen nur eine Scheingenauigkeit überprüfen. Die Anzeige von 16 Stellen ist kein Beweis, dass der Computer auch wirklich mit 16 Stellen rechnet. Deshalb sollten die Tests 3 und 4 auch mit andern PC's geprüft und neu definiert werden.

Von besonderem Interesse wäre sicher auch die Publikation von Tests

### Manuskript-Einsendungen

Fachliche lehrreiche Artikel von freien Autoren sind immer willkommen. Die Zustimmung des Verfassers zum Abdruck wird vorausgesetzt. Interessante Beiträge, die wir abdrucken, honorieren wir angemessen.

**Mikro+Kleincomputer**  
**Informa Verlag AG**  
**Postfach 1401, 6000 Luzern 15**



mit Geräten, welche über den schnellen 8087-Prozessor verfügen.

```
10 'MUK4B'
20 DEFINT I,K
30 DEFDBL A,X
40 '
100 PRINT «START»;CHR$(7)
150 FOR I=1 TO 100
170 FOR K=1 TO 100
200 X=I
250 A=((X-1#)/(X+1#))*X*.39#+A
300 NEXT K
350 NEXT I
400 PRINT «ENDE»;CHR$(7)
450 PRINT «RESULTAT= »;A
500 INPUT «DRUCK Y/N »;C$
550 IF C$=«Y» THEN LPRINT «A=»;A
600 END
```

A = 189477.387723603  
 Zeit: 656 Sek.

```
10 'MUK3C'
20 DEFINT I
30 DEFDBL A,D
100 PRINT «START»
200 FOR I=1 TO 2000
210 D=I
250 A=A+(D↑2.2#)/EXP(SQR(D)+1#)
300 NEXT I
400 PRINT «ENDE»;CHR$(7)
450 PRINT «RESULTAT= »;A
500 INPUT «AUSDRUCK Y/N »;C$
550 IF C$=«Y» THEN LPRINT «A=»;A
600 END
```

A = 177.1951690415149  
 Zeit: 187 Sek.  
 MUK3C compiliert:  
 A = 177.1951369259303  
 Zeit: 1409 Sek.

Anmerkung: Doppelpräzision bei Konstanten scheint nur bei Operationen höher als +,- nötig zu sein. Das Setzen des #-Zeichens auch in diesen Fällen bringt aber keine Nachteile.

Dr. Werner Fisch

## CRT-SOFT-COPY (M+K 84-4)

Besten Dank für Ihre guten Artikel in M+K über den Sharp-Computer.

Ich besitze privat einen MZ-80B, voll ausgebaut. Mein Interesse gilt deshalb ganz besonders Ihren Artikeln, die ich mit Erfolg anwende. In meinen Programmen verwende ich viel hochauflösende Grafik. Ihr «CRT-SOFT-COPY» ist mir dabei eine unschätzbare Hilfe. Da ich selbst fast ausschliesslich in BASIC programmiere, fehlen mir jedoch die Grundlagen um CRT-SOFT-COPY für Diskettenbetrieb umzuschreiben. Haben Sie eine Diskettenversion zur Verfügung haben?  
 Werner Annen

Leider kann ich Ihnen keine Diskettenversion bieten, ich nehme aber an, dass sich unter den Lesern bestimmt einige dieses praktische Verfahren zum Speichern einer Grafikseite für Diskette adaptiert haben. Der Vorgang ist nicht schwer - analog zur Kassettenversion, bedingt aber doch Kenntnisse über Maschinensprache.

Speichert man Bilder in hochauflösender Grafik, so lässt sich mit schnellem Zugriff jedes Bild von Diskette auf den Bildschirm holen. Bei Betrieb mit Kassette ist nur ein sequentieller Zugriff möglich, zudem benötigt das Einlesen wesentlich mehr Zeit, das Bild wird nach dem SOFT-COPY-Verfahren in rund 30 Sekunden auf den Bildschirm gebracht. Um ein bisschen Anregung zu geben, ist im Bild das Cockpit einer BOEING 747 dargestellt. Wer hilft Herrn Annen bei der Realisierung seines Flugsimulators durch ein Disketten-SOFT-COPY weiter?  
 Leopold Asböck

## EPSON PX-8 in M+K 84-5

Mit sehr grossen Interesse habe ich den obigen Artikel gelesen. Mit besonderem Interesse, weil ich diesen Rechner seit einigen Wochen besitze.

Zu Anfang muss ich zugeben, dass ich den Befehl zum Uebergang vom Menü auf «A³» nicht kannte und ihn mit Freude zur Kenntnis nahm. Auch mein recht gut informierter Verkäufer kannte ihn nicht.

Auch möchte ich betonen, dass ich mir diesen Rechner auch heute nochmal kaufen würde. Aber dies darf nicht bedeuten, dass man ihn für perfekt hält. Ich möchte mir daher erlauben einige Kritik, und wie ich sie verstehe, helfende Kritik anzubringen.

Mit der positiven Betrachtung Ihres Artikels über die äussere Form bin ich voll inhaltlich einverstanden. Diese hervorragende Aussenform, die deutsche Tastatur, die inneren «Werte», alles ganz prima. Aber bei der Software sieht man leider, dass man zu sehr an den «Manager» und fast nicht an den «Ingenieur» gedacht hat. Oder will man etwa damit andeuten, dass einem Ingenieur in der Praxis ein so fantastisches Gerät gar nicht zusteht?!?

### Zur Tastatur:

Sehr schön, Tastatur in Ordnung und angenehm in der Bedienung, aber ...

Wenn schon die Möglichkeit der Verwendung als Taschenrechner (mit sh num), warum sind die Klammern (, ) und das Potenzzeichen ^ nicht mit auf der Tastatur angegeben? Warum denkt man nicht an die wissenschaftliche Verwendung? Wo sind die Tasten für die Winkelfunktionen, die Logarithmen, statistischen Funktionen usw. Das wäre doch echt ein erheblicher Fortschritt! Könnte man dann nicht auch den print-Befehl «?» einsparen? Eine Zeile im «NUM»-Modus eine Rechnung! Es sind doch soviel Tasten frei für so etwas. Man könnte doch die Tastenvorderseite für diese Bezeichnungen verwenden!

### Zum Textsystem:

Es ist ja anscheinend nur ein Vorläufer der später nachzuliefernden PROM-Version. Mit den Kassetten-Programmen hatte ich verschiedene Schwierigkeiten beim Einlesen, die aber jetzt behoben sind. Bei Ihnen

las ich, dass es ein WordStar-Programm sein soll?

Meine Kritik: ein Textprogramm sollte das Schieben einer Restzeile in die nächste Zeile gestatten, dies geht hier nicht; auch das Rückwärtsziehen beim Löschen geht nicht, schade, schade, schade. Und es sollte möglich sein, die formatierte Form abzuspeichern. Sonst sind die vorhandenen Text-Funktionen nach meiner Auffassung ausreichend. Aber warum leuchtet das «insert»-Lämpchen im Textmodus nicht? Dazu vielleicht die Angabe, dass ich den PX-8 für Entwürfe unterwegs verwende, und diese Texte zuhause auf eine grössere Anlage überspiele. Und warum funktioniert die «auto»-Abschaltung beim Textmodus nicht? Nur eine Kleinigkeit, aber etwas was ärgert! Und warum nicht die Möglichkeit die oberen zwei Zeilen im Display abzuschalten, dann hätte man doch wirklich acht Zeilen zur Verfügung! Sie schreiben von einer Help-Funktion im WordStar. Bei meinem Textprogramm funktioniert kein «help». Also habe ich wahrscheinlich kein WordStar!

### Zum Calc:

Und wieder das Thema, Ingenieur Anwendungen vergessen! Was soll dieses Primitiv-Programm (33K!) in dem man nur die vier Grundrechnungsarten verwenden kann. Mit Mühe kriegt man eine Schwerpunktsberechnung eines zusammengesetzten Querschnittes hin, eine Berechnung von einfachen Trägheitsmomenten geht gerade noch, aber Hauptträgheitsmomente oder Dehnungsrechnungen oder Hauptspannungen bei denen man Winkelfunktionen und Potenzen braucht, sind leider unmöglich. Warum melden sich die Ingenieure nicht zu solchen Problemen? Die Funktionen sind doch alle im Rechner vorhanden, es kann also kaum am Programmumfang liegen. Mit 33K lässt sich doch mehr machen! Besonders schön wäre natürlich eine «if»-Abfragemöglichkeit eines Feldelementes, das hat ja nicht mal das legendäre «Visicalc»! Und noch eine letzte Randbemerkung zum «Calc», die Beschreibung ist sehr schlecht. Auf einer A4-Seite kann man auch dem Anfänger mehr sagen, als mit diesen Blättern.

Der Befehl «wahl» funktioniert zwar weder im Text- noch beim Calc-Programm. Aber dies dürfte wohl nur ein Fehler in den Kassettenversionen sein.

Ihrer Auffassung der «exzellente» geschriebenen Handbücher stimme ich nicht zu. Die BASIC und das USER's Manual sind ganz ordentlich, aber warum kein Schlagwortverzeichnis, wie in der englischen Ausgabe? Man hätte es doch bloss übersetzen müssen! Warum bringt man solche Abschnitte, wie die sehr wichtige Angabe der Hierarchie der RESET-Befehle nicht in der deutschen Übersetzung?

Ein Programm «SCHEDULE» war bei mir nicht vorhanden und auch mein Händler kennt es nicht. Die Schweiz ist eben soweit von «Old Germany» entfernt. Vermutlich hat EPSON bei Ihnen eine andere Zusammenstellung ausgeliefert.  
Prof. H.L.Jochen Kux

*Ein ganz herzliches Dankeschön für Ihren umfassenden Fragekatalog. Weder wir von der Redaktion noch unsere Autoren, nehmen für sich in Anspruch in Sachen Computer wirklich allwissend zu sein. Und Ihre sehr konkreten und vernünftigen Fragen sind ja wohl eher an Epson gerichtet als an die M+K-Redaktion. Es würde uns deshalb auch in Ihrem Interesse freuen, wenn Epson auf Ihre Kritik reagiert, die wir gerne an dieser Stelle abdrucken.*

## Toolkits zum PC-1500

Den Artikel «Toolkits zum PC-1500 in M+K 84-3 haben wir mit grossem Interesse gelesen. Man kann den Autoren den geglückten Versuch bestätigen, die wesentlichen Punkte dieser Erweiterungen klar verständlich dargelegt zu haben.

Die berechtigte Kritik an der Beschreibung des RWEbasic hat uns zu einer nochmaligen Ueberarbeitung veranlasst. Die vermissten genaueren Angaben zum Speicherverbrauch bei lokalen Angaben sind jedoch im wesentlichen in der Anleitung enthalten:

Die als Variable «getarnte» Tabelle für die zum Löschen der lokalen Variablen notwendigen Informationen belegt 8 Bytes sowie 4 Bytes je Referenz im Variablen-Bereich (nur bei GSB-Aufruf notwendig, nicht bei FN).

Die Referenz-Variablen benötigen darüber hinaus keinen Speicherplatz. Alle übrigen lokalen Variablen unterscheiden sich nicht von gleichartigen globalen Variablen und belegen damit auch den dort notwendigen Speicherplatz.

Dipl. Ing. Werner Eckstein

## Probleme mit Schleifen

Antwort auf Leserbrief in M+K 84-3: Hier ist nicht die FOR-NEXT-Schleife das Problem, sondern die Stringverwaltung. Wenn der String-speicher voll ist, müssen die Strings wieder gepackt werden («Garbage Collection»), was bei 1600 Strings ca. 3-4 Minuten dauert! Haben Sie schon einmal versucht, mit einem in BASIC geschriebenen Editor Assemblerprogramm von über 1000 Zeilen zu editieren? Die Wartezeiten sind wirklich sehr eindrucksvoll.

Durch das STOP/RESTORE wird die Garbage-Collection unterbrochen, d.h. die Stringzeiger sind durcheinander, neue Strings werden in die alten geschrieben.

Mögliche Abhilfe: Weniger Strings verwenden (Felder eines Datensatzes zusammenfassen); Maschinensprache; Relativer File.

Uebrigens haben fast alle Microsoft-BASIC's diese Art von Stringverwaltung; schon mache Programmierer haben sich darüber gewundert, dass ihr Programm bei grossen Datenmengen fast zum Stillstand kam...  
Pascal Dornier

## Programme in TURBO-Pascal

Mit grossem Interesse habe ich die Pascal-Spielprogramme Ballon, Tigerjagd und Turm von Hanoi in den Ausgaben M+K 81-2, 81-5 und 81-6 gelesen.

Als Besitzer eines Sirius-1 und des neuen TURBO-Pascals habe ich nun versucht, diese Spiele zu implementieren. Das Programm Hanoi läuft ausgezeichnet und begeistert meine ganze Familie. Leider bereitet mir die Anpassung der beiden anderen Spiele einige Probleme, da ich die Bedeutung und Funktion der Befehle «inport(x,y)» und «outport(x,y)» nicht kenne.  
Marco Rampone

*Die erwähnten Befehle bewirken nichts anderes, als dass ein Byte von einem Port (mit der Adresse x) eingelesen, bzw. ein Byte an diese Portadresse ausgegeben wird. In den Programmen, die in Pascal-M geschrieben sind, wird damit eigentlich nur die Tastatur des Superbrain abgefragt, um festzustellen, welche Taste gedrückt wurde. Im Basic entspricht dem etwa der INKEY\$-Befehl, bei manchen Interpretern der GET-Befehl. Aus Zeitmangel konnte ich mich noch nicht mit dem TURBO-*

*Pascal-Compiler beschäftigen, doch nehme ich an, dass es hier gleichfalls eine einfache Möglichkeit geben wird, um eine einzelne Tasteneingabe zu realisieren.*

*Sollten Sie Ballon und Tigerjagd übertragen, so empfehle ich, die hervorragende Grafik des Sirius einzusetzen, da die Spiele am Superbrain nur mit den ASCII-Standardzeichen ausgeführt wurden.*  
Leopold Asböck

## Automatisches Schraffieren von Flächen (M+K 84-1)

Auf der Suche nach einem Schraffurprogramm für Flächen fand ich den Beitrag von Prof. Dr. W. Bachmann in M+K 84-1.

Es stellte sich jedoch heraus, dass dieses Programm nur für einfache Flächen funktioniert. Sobald eine Fläche mehrere einspringende Ecken aufweist, werden diese von den Schraffurlinien, anstatt ausgespart, grösstenteils überzeichnet.

Bei der Ueberprüfung der errechneten Schnittpunkte stellt ich fest, dass zwar alle Werte richtig waren, jedoch ihre Reihenfolge mit steigender Anzahl der einspringenden Ecken immer willkürlicher wurde. Die im Programm gewählte Ausgabeart (Zeile 520-560) kann diesem Verhalten jedoch nicht gerecht werden.

Durch folgende Aenderung des Programms lassen sich beliebige ebene Flächen schraffieren:

```

410 m=m+1
420 u(m)=x(i)+t*(x(i2)-x(i))
430 v(m)=y(i)+t*(y(i2)-y(i))
470 nexti
475 ifm=0then585
480 ifm<3then570
485 j1=m:ifw>45andw<135then530
490 j1=int(j1/2):ifj1=0then570
495 k=m-j1:j=1
500 i=j
505 ifu(i)<=u(i+j1)then520
510 i2=u(i):j2=v(i):u(i)=u(i+j1)
      :v(i)=v(i+j1):u(i+j1)=i2
      :v(i+j1)=j2:i=i-j1
515 ifi>=lthen505
520 j=j+1:ifj>kthen490
525 goto500
530 j1=int(j1/2):ifj1=0then570
535 k=m-j1:j=1
540 i=j
545 ifv(i)<=v(i+j1)then560
550 i2=v(i):j2=u(i):v(i)=v(i+j1)
      :u(i)=u(i+j1):v(i+j1)=i2
      :u(i+j1)=j2:i=i-j1
555 ifi>=lthen545
560 j=j+1:ifj>kthen530
    
```

```

565 goto540
570 fori=1tom-1step2
575 !ls,u(i),v(i),u(i+1),v(i+1),l
580 nexti
585 nextd
590 end
    
```

Auf die Berechnung der Extremschnittstellen (Zeile 440-460) wird verzichtet. Ist die Anzahl m der Schnittpunkte einer Schraffurgeraden mit der Flächenumrandung grösser als 2, werden sie in Zeile 490-525 bzw. 530-565 der Grösse nach sortiert und dann erst ausgegeben (Zeile 570-580).

Das Kriterium, ob nach aufsteigenden X- oder Y-Werten sortiert wird, bildet der Schraffurwinkel w (Zeile 485).

Rainer Eberenz

## NEC 8201A

Ich habe einen NEC 8201A. Bei einem Besuch in Kanada habe ich mir kürzlich eine Programm-Kassette «personal finance» für den Tandy Mod. 100 gekauft, in der Annahme, dass ich sie für meinen NEC verwenden kann. Die beiden Apparate sollen ja praktisch identisch sein (M+K 83-5). Leider geht es aber nicht. Können Sie mir einen Trick verraten?  
Walther Howald

*Da ich zur Zeit weder auf einen NEC 8201A noch auf einen Tandy Mod. 100 Zugriff habe, ist es mir nicht möglich, den festgestellten Unterschied zu untersuchen; ich vermute ihn jedoch eher in der Betriebssoftware als in der Hardware.*

*Doch vielleicht hat einer unserer Leser das Problem bereits gelöst? Schreiben Sie uns, wir drucken Ihre Antwort gerne in dieser Rubrik ab.*  
Eric Hubacher

## MZ-700

Beim Arbeiten mit meinem MZ-731 bin ich auf Probleme gestossen. Beim Programmieren in Hisoft-Pascal gelingt es mir nicht, Zeichen in das V-RAM zu poken und auf dem Monitor darzustellen. Da ich zu wenig Erfahrung habe, möchte ich nicht mit Maschinensprache arbeiten. Gibt es einen Weg, mit POKE?  
Reto Buchli

*(Red.) Welcher Leser hat damit bereits Erfahrung und weiss eine Antwort?*

# BRIEFE AN DIE REDAKTION

## Interface für CPC 464 parallel Centronics auf Brother C-50

Ich bin Besitzer eines Schneider Personal Computer CPC 464 und einer Brother Schreibmaschine C-50 mit Anschluss seriell. Gibt es dazu ein Interface, das diese beiden Geräte kompatibel zu einander macht?

Auf der einen Seite haben wir also den Computer mit einer standardisierten Centronics Paralleldrucker-Schnittstelle, auf der andern Seite die Schreibmaschine (Drucker) mit einer 5-poligen Anschlusseinheit seriell (vorbereitet, also lauffähig für C-64).

Jetzt suche ich ein Interfacekabel für den CPC 464, damit ich mit der Brother-Schreibmaschine drucken kann.

Johann Wyss

*(Red.) Ein Kabel alleine genügt nicht. Sie brauchen dazu Interface parallel/seriell, von denen es verschiedene auf dem Markt gibt. Am besten fragen Sie Ihren Computerhändler in Ihrer Nähe. Vielleicht kann Ihnen aber auch ein Mitleser weiterhelfen?*

## Mikrocomputer-Markt in der Schweiz

Als verzweifelter Schweizer Mikrocomputer-Zeitschriftenleser, -Sucher, -Käufer, -Benutzer, stelle ich fest, dass wir in dieser Sache immer noch recht «mikrig» informiert werden.

Bloss eine Zeitschrift - M+K - berichtet ausschliesslich über diesen Markt. Wer aber die Lage in Deutschland betrachtet, der muss hier verzweifeln. Dort gibt es offenbar die wundervollsten Mikros zu kaufen, die hier keiner kennt, die hier keiner verkaufen will, geschweige denn reparieren kann. Oder ist das Täuschung?

Da fragt man sich beim Evaluieren eines Gerätes immer: «Wer ist der Schweizer Importeur? Wo sitzt er? Wo oder wie finde ich die Vertretung? Muss ich wirklich für meinen Wunsch-Computer nach München, wenn ich mich nicht auf die hier bekannten Marken beschränken will?»

Sie werden einwenden, dass halt jeder, der was verkaufen will, inserieren muss. Das stimmt aus Ihrer Sicht und aus der Sicht Ihrer Inserenten. Aber es genügt nicht als Dienstleistung einer Fachzeitschrift, die beinahe allein und regelmässig über dieses Gebiet berichten kann.

Beispiel: In Ihrer letzten Nummer (M+K 84-5) haben Sie kurz darüber berichtet, dass die Informations-passive Firma A von der aktiven Firma B aufgekauft worden sei. Die Interessenten und Besitzer dieses Computers, so lese ich weiter, dürfen aufatmen.

Es war möglicherweise das erste Mal, dass in M+K ein Hinweis auf ein Produkt jener Marke zu finden war. Dabei handelt es sich um einen der ersten und grössten Mikro-Vertreiber der Welt. In jeder grösseren Stadt Deutschlands betreibt er seinen «Shop». In der Schweiz ist er nicht-existent(?).

Ich verlange nicht, dass Sie nun Produkte propagieren sollen, welche in unserem Land gar nicht zu haben sind. Das besorgen andere Verlage über die Grenzen. Aber Sie könnten zur Förderung der Vielfalt beitragen, indem Sie regelmässig eine Marken-Importeurliste veröffentlichen. So weiss der Leser, wo er sich über den nächstgelegenen Vertreter erkundigen kann. Sollte die Marke nicht in unserem Land vertreten werden, so geben Sie die europäische Generalvertretung an. Vielleicht kennt man dort eine Schweizer Firma, die bei Bedarf das Gerät importiert.

Im Moment noch sind wir computerologischen Hinterland. Die hier ungefragten Marken veranlassen die Europa-Vertretungen zu glauben, dass hier kein genügender Markt vorhanden sei. Kein Wunder, wenn niemand weiss, wer dafür angefragt werden kann.

Peter Freiburghaus

*Tja, da haben Sie nun tatsächlich den Nagel auf den Kopf getroffen. So wie Ihnen ergeht es uns ebenfalls. Auch wir bekommen hin und wieder nasse Augen, wenn wir sehen, was es so alles im «grossen Kanton» zu haben geben soll - nur, ein bedrucktes Blatt Papier (und wenn es noch so schön farbig ist) muss noch lange keine Garantie dafür sein, dass man das so beworbene Gerät auch wirklich bekommen kann. Wir haben nun Ihre Anregung, für die wir uns herzlich bedanken, zum Anlass genommen, eine Markenliste vorzubereiten. Die ersten Adressen sind bereits bei uns eingetroffen. Trotzdem werden wir aufgrund der im Moment noch zögernden Resonanz, den Eindruck des «computerologischen Hinterlandes» nicht ganz los. In einer kommenden Ausgabe veröffentlichen wir die «Markenliste» Schweiz. Haben Sie also noch ein wenig Geduld. Ihre M+K-Redaktion*

## Replik auf die Kritik zum Pascal-Programm (M+K 84-3)

Es freut mich, dass mein Pascalprogramm Herrn Debrunner aus der Reserve gelockt hat. Das war genau das, was ich im Stillen erhoffte, nämlich dass über Pascal etwas mehr geschrieben wird.

Es war nicht mein Ziel, ein möglichst einfaches Programm zu demonstrieren, sondern mit den Möglichkeiten von Pascal etwas zu spielen. So wurde z.B. das INLINE-Statement absichtlich belassen, wie auch das GOTO.

Uebrigens habe ich die Erfahrung gemacht, dass kurze Pascal-Programme mit GOTO eher übersichtlicher sein können als wenn mehrere Boolean'sche Variablen verwendet werden müssen. GOTO sollte aber nur für Fehlersprünge gebraucht werden.

Herr Debrunner hat in meinem Pascal-Programm in Zeile 60 einen Fehler gefunden. Die Bemerkung dazu ist völlig richtig. Wie die Nachprüfung ergab, hatte es hier glücklicherweise keine Konsequenzen. Der Compiler reservierte nämlich in der Variablen-tabelle zwischen der Variablen «EINBUCHSTABE» aus der Procedure «ZAHL\_ZU\_STRING» und der nächsten Variablen «N» aus der Procedure «BUCHSTABE» ein zusätzliches Byte. So erklärt es sich, dass das Programm fehlerlos läuft.

Die übrigen Bemerkungen von Herrn Debrunner sind ebenfalls richtig und entsprechen der Tatsache, dass verschiedene Wege nach Rom führen. Redundanz braucht zwar Speicherplatz, ist aber für das Funktionieren des Programms unwesentlich und schafft beim Programmieren oft Klarheit (Zeile 46/51). Die Procedure «ZAHL\_ZU\_STRING» wurde aus einem anderen Programm übernommen. Tatsächlich könnte sie im Prinzip eine zehnstellige Integerzahl in einen String umwandeln, sofern die Maschine eine solche Integerzahl bewältigen kann. Sie wird hier in Zeile 175 aufgerufen, wobei jeweils immer nur eine einzige Zahl verarbeitet wird. So arbeitet die Procedure also wiederum mit einer Redundanz, welche hier eine Fähigkeit vortäuscht, die das gesamte Programm nicht hat. Im übrigen steht nirgends, dass vom Programm eine zehnstellige Zahl berechnet werden könne.

Die Verwendung von INLINE sollte gemäss Handbuch keine Probleme bieten, steht doch ausdrücklich: «This feature allows the user to insert data in the middle of a Pascal/MT+



procedure or function.» (User's Guide, Release 5, Febr. 82, p. 114)

Die Verwendung einer FOR-Schleife anstelle von REPEAT ist selbstverständlich überall möglich. Ich verwende in der Regel REPEAT, weil der Compiler einen kürzeren Object-Code erzeugt (54-68).

Das READ in Zeile 305 bringt keine Probleme, weil in beiden Fällen der Auswahl ein WRITELN folgt.

Für die Bildschirmausgabe auf meinem System (Altos-Computer) hatte ich für die Integerzahlen zwischen 32768 und 65535 schon verschiedene Probleme. Dies führte zu den Kunstgriffen für die negativen Zahlen.

So berechnet er für 8000H = -32768, für 8001H = -32767 etc., für FFFFH schliesslich -1. Dies geschieht bei mir auch mit dem von Herrn Debrunner angegebenen Programm.

Letzteres hat übrigens den Nachteil, dass bei jeder Berechnung ein (unnötiger) Diskzugriff erfolgt, was dieses Programm verlangsamt.

Dr. Alfons Fässler

## Nochmals FORTH-Kurs in M+K 84-4 und Korrektur in M+K 84-5

Ein Leser bemängelt die fehlerhafte BASIC-Programmierung des Befehls MOD, bringt aber selbst mit seinem Vorschlag

```
67 MOD 3 = (67/3-INT(67/3))*3
```

eine Implementierung, die wegen der nicht ganz genauen Berechnung von 67/3 eine nachfolgende aufwendige Rundung zum Ausgleich der Rechenungenauigkeit erfordert. Es geht in BASIC aber auch viel einfacher und ohne Rundung:

```
67 MOD 3 = 67-(INT(67/3)*3
```

Hier fällt erst noch eine REAL-Division weg, so dass neben der Genauigkeit auch noch die Rechengeschwindigkeit höher ist.

Mathias Eugster

**Nächsten Monat gibt's wieder**

**COMPUTER  
MARKT**

**Abonnement schon bestellt?**

## Turbo Source Lister 1.00A

Use this program to make a listing of your programs.  
To format your print you may use the following directives  
in your source code:

Directive	Explanation	Example
{.PL <int>}	sets the pagelength to <int> lines.	{.PL66}
{.CP <int>}	new page if not space for next <int> lines	{.CP10}
{.PA}	immediate page break	{.PA}
{.PO <int>}	<int> spaces in left margin	{.CP8}
{.HE <text>}	place <text> as heading on each page	{.HEProgram test}
{.FO <text>}	place <text> as footing on each page	{.FOContinued..}
{.L-}	turn listing off	
{.L+}	turn listing on again	

Include files will only be listed if the {\$I directive is in column one.  
If you use the HE or FO directive then you can use the character «#»  
to indicate where in the HEading or FOoting you want the page number.

## Neue Befehle auf SHARP PC-1500 (M+K 83-3)

Warum kommt es bei meinem Rechner bei Anwendung des Programms AUTOR.+OFF jedesmal zu einem Absturz (auch wenn ich in 40C7H anstelle von 4AH 43H einsetze)? Für einen guten Tip bin ich dankbar.  
A. Coray

*Der Fehler liegt bei dem in älteren PC-1500 eingebauten ROM. Es gibt bis jetzt drei Versionen des SHARP PC-1500, die sich im ROM unterscheiden. Sie tragen die Bezeichnung A01, A03 und A04. Fehler im Betriebssystem der Rechner der Version A01 wurden in den späteren Versionen korrigiert. So funktioniert die Tastaturumleitung zum Beispiel nur bei den Versionen A03 und A04. Da diese im Autorepeat-Programm (M+K 83-4) zur Verwendung kommt, kann es auf A01-Rechnern nicht eingesetzt werden. Mit zwei PEEKs können Sie bei Ihrem Rechner feststellen, welche Ausführung Sie besitzen:*

```
PEEK &C443 PEEK &C5BD
A01 56      129
A03 59      129
A04 59      74
```

*Welche Fehler die alten Versionen auch noch aufweisen, ist in der 1. Clubzeitschrift des PC-1500-Clubs in Chur beschrieben. Die Kontaktadresse lautet: SHARP PC-1500 Club, Marco Feusi, Giacomettistrasse 33, 7000 Chur.  
Markus Golder*

## Die Turbo-Sensation (M+K 84-5)

Der Artikel über das neue Turbo-Pascal ist Ihnen gut gelungen. Obwohl ich Turbo-Pascal bereits seit einiger Zeit verwende, konnte ich aus Ihrem ausführlichen Artikel noch einige zusätzliche Informationen gewinnen. Der Vergleich von Turbo-Pascal mit Pascal MT+ zeigte ausserdem, wie komfortabel und schnell ein Pascal-Compiler werden kann, wenn man genügend «brainware» in die Entwicklung eines solchen Produktes steckt.

Aber anscheinend hat der Turbo-Test doch ein wenig zu schnell stattgefunden. Die Aussage, dass die Punktbefehle für die Gestaltung eines Programmlistings mittels TLIST nirgends beschrieben sind, ist leider falsch und somit keine offene Frage beim Turbo-Pascal.

Startet man TLIST, so erscheint die Anweisung

Enter name of file to list (?) for help  
<RETURN> to skip:

Drückt man danach die Taste [?], so erscheint am Bildschirm eine Auskunft aller zur Verfügung stehender Punktkommandos. Man sieht also, dass die Kommandos {.FO text}, {.L-} und {.L+} in der letzten Uebersicht fehlten und dass die Kommandos {.LH nn}, {.MT nn} und {.MB nn} nicht angeführt sind.  
Gerd Platl

# Dieses Netzwerk paßt nahtlos: PCnet.

Produkte von  
Computer 2000 gibt  
es nur im guten Fachhandel.  
Hier eine Auswahl:

## Schweiz:

- 3013 Bern**  
Hannes Keller AG, 0 31/41 22 45
- 4051 Basel**  
Globudata, 0 61/23 31 60
- 4053 Basel**  
Kubli + Eicher AG, 0 61/35 05 17
- 5405 Baden-Daettwil**  
GEI-Systeme AG, 0 56/83 30 85
- 6002 Luzern**  
Dialog Computer Treuhand AG, 0 41/ 31 45 45
- 6280 Hochdorf**  
Computer Center Hochdorf, 0 41/88 33 91
- 7000 Chur**  
Rampa Comp. Eng., 0 81/24 32 44
- 8003 Zürich**  
Globudata, 01/4 63 60 40
- 8006 Zürich**  
Gesmarco AG, 01/3 63 13 05
- 8032 Zürich**  
Hannes Keller AG, 01/69 36 33
- 8048 Zürich**  
Logon AG, 01/62 59 22
- 8810 Horgen**  
Micom AG, 01/7 25 50 10
- 9495 Triesen**  
Micomp AG, 0 75/2 79 97

## Deutschland:

- 6000 Frankfurt**  
Henneveld KG, 01 11/28 14 82-83  
Computerhaus Kegelmann, 01 11/44 60 16  
NATIC + EDV Vertrieb, 01 11/7 24 05 68-9
- 6200 Wiesbaden**  
Henneveld KG, 0 61 21/30 70 91  
NATIC + EDV Vertrieb, 0 61 21/51 7 45
- 6374 Steinbach**  
Renthal GmbH, 0 61 71/70 11
- 6500 Mainz**  
Henneveld KG, 0 61 31/8 50 91
- 6600 Saarbrücken**  
COS GmbH, 06 81/5 20 55  
Pfeiffer KG, 06 81/5 27 11
- 6700 Ludwigshafen**  
DatSERVICE, 06 21/58 18 73
- 6740 Landau**  
Data-Service GmbH, 0 63 41/8 30 72
- 6750 Kaiserslautern**  
Data-Service GmbH, 06 31/1 60 81
- 6901 Dossenheim**  
G.D.O., 0 62 21/8 50 34
- 7000 Stuttgart**  
Kübler GmbH, 07 11/61 06 51  
Häussler GmbH, 07 11/7 83 30
- 7500 Karlsruhe**  
DatSERVICE, 07 21/37 59 57
- 8000 München**  
PCM, 0 89/28 58 60  
Startcomputer Center, 0 89/28 22 09 + 72
- 8011 Baldham**  
UDV GmbH, 0 81 06/76 93
- 8400 Regensburg**  
EPA KG, 09 41/4 50 58-9
- 8700 Würzburg**  
Computer Martin, 09 31/1 65 58
- 8900 Augsburg**  
Böwe MASCHINENFABRIK GmbH, 08 21/5 70 21

## Osterreich:

Hayward Computer & Peripherie  
A-5033 Salzburg  
Tel. 06 62/2 00 26

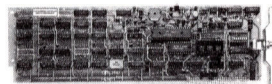
## Schweiz:

Computer 2000 AG  
CH-6345 Rotkreuz  
Tel. 0 42/64 20 22

## Computer 2000 GmbH

Klenzestraße 38  
8000 München 5  
Tel. 0 89/26 80 70

Wenn Sie bereits 2 IBM-PC's besitzen, wird es Zeit, Ihr **lokales Netzwerk (LAN)** zu knüpfen. Mit PCnet von Orchid können alle angeschlossenen Computer die teuren Peripheriegeräte wie **Drucker, Plotter, Modems** etc. gemeinsam benutzen und haben Zugriff auf gemeinsame, aktuelle Daten. Über 10.000 Installationen, lange Erfahrung mit lokalen Netzen und modernste Technologie verhalfen PCnet zum Erfolg. Davon können Sie jetzt profitieren. Damit Ihr Netzwerk sitzt wie angegossen, brauchen Sie nur eine **PCnet-Erweiterungskarte** oder **PCnet-Blossom-Karte** für jeden anzuschließenden PC sowie ein ganz normales Koax-Kabel für die Verbindung der PC's. So schnell und einfach kann heute ein Netzwerkbetrieb sein: Sie erzielen sofortige Produktivitätssteigerungen, Umlernen ist nicht notwendig. Dabei sind alle angeschlossenen PC's völlig normal weiter zu benutzen. Sie brauchen weder Spezialkabel noch Spezialzusätze (Servergeräte etc.).



Vereinbaren Sie einfach bei einem der nebenstehenden Fachberater einen Termin. Er informiert Sie auch über Multiuser-Datenbanken, wie **LAN: Datastore, Electronic Mail, Diskless PCnet, Multifunktions PCnet (Blossom), 80186 Coprozessoren** oder **Telexnetz**-Verbindungen. Oder fordern Sie von uns eine komplette Händlerliste an.

# COMPUTER 2000

**Wir wissen, was läuft.**

# FORTH auf Commodore C-64 (3)

**Der C-64 ist durch seine einfache Editiermöglichkeit bekannt, wobei man unter Editieren die Veränderung eines Textes oder eines Programmes versteht. Nach dem Laden des Programmes aus dem RAM in den Bildschirmspeicher kann mit LIST die Veränderung mit dem Cursor vorgenommen werden und mit der RETURN-Taste wieder in das RAM überführt werden.**

Nicht ganz so einfach ist das Editieren in FORTH. Die Veränderungen werden, wie auch in vielen anderen Programmiersprachen, mittels eines speziellen Editors vorgenommen. Das Quellprogramm eines FORTH-Systems ist in einer Reihe von Blöcken, die «Screens» genannt werden, auf Disk oder Tape gespeichert. Soll ein solches Screen editiert werden, so wird zuerst das RAM auf die Existenz dieses Screen hin überprüft; ist es nicht vorhanden, wird es vom Tape oder der Disk geladen. Da die Veränderung des Screens beim Editieren nur auf dem Bildschirm erfolgt, nicht aber im RAM, muss vor dem Verlassen des Editors das geänderte Screen in den Speicher zurückgeschrieben und eventuell auf Tape oder Disk abgespeichert werden.

Der Aufruf eines Screens zum Editieren erfolgt mit

```
<Screen #> EDIT
```

Sofern sich das Screen noch nicht im Arbeitsspeicher befindet, wird es von der Disk geladen. Die ersten 1000 Charakter, dies entspricht 25 Zeilen, werden auf dem Bildschirm angezeigt. Die verbleibenden 24 Zeichen werden nicht generell weggelassen, sondern sind, wenn nötig ebenfalls zugänglich. Die Art des Editierens eines Screens unterscheidet sich dann nicht mehr vom Editieren in BASIC. Auch hier kann mittels Cursor die zu verändernde Stelle angefahren werden und durch Ueberschreiben oder Einfügen mit der INSERT-Taste bzw. Löschen mit der DELETE-Taste gearbeitet werden. Der gesamte Bildschirm ist mit dem Cursor zugänglich. Die Cursor-, Revers-, Home- und Farbtasten arbeiten wie gewohnt. Der Bildschirm kann nicht mit CLR-Taste gelöscht werden, da diese ebenfalls die HOME-Funktion ausübt. Verschiedene spezielle Optionen machen die Arbeit mit dem Editor relativ einfach.

## Länge einer Screen-Linie

Jede Linie des angezeigten Screens enthält normalerweise 40 Charakter. Das Löschen oder Einfü-

gen von Zeichen in der Mitte einer Linie hat in der Regel die Verschiebung der Zeichen dieser Linie zur Folge. Hierbei müssen zwei Ausnahmen gemacht werden.

Mit der Funktionstaste I wird die Länge der Linie auf 80 Charakter gesetzt, eine Ausnahme bildet die letzte Linie des Bildschirms. Wird die Taste F1 gedrückt oder wird eine Zeile mittels Cursorbewegung verändert, wird die Zeilenlänge auf 80

## Heinz Kastien

Zeichen gesetzt. Jegliches Löschen oder Einfügen von Zeichen in dieser Linie hat eine Verschiebung der Charakter zur Folge.

Bei Benutzung der «DEL»-Taste beim ersten Zeichen einer Linie wird die Linie als eine Erweiterung der vorhergehenden Linie betrachtet und setzt sie auf 80 Zeichen.

## Buffer im Edit-Mode

Verschiedene Funktionen des Editors benutzen Buffer bzw. eröffnen dieselben. Es wird der freie Speicherplatz oberhalb des PAD bis zum Beginn des Disk-Buffers belegt. Weiterhin wird ein 80 Byte Charakterbuffer von DELETE-Charakter und SAVE-Funktion benutzt. Ein Linebuffer wird von der DELETE-Linie und SAVE-Funktion belegt.

## Einfügen

Es gibt verschiedene Möglichkeiten, Text innerhalb eines Screen zuzufügen. Die INSERT-Taste bewirkt eine Verschiebung des gesamten Textes einer Linie von der Position des Cursors bis zum Ende der Zeile, es wird ein Leerzeichen eingefügt. Der Text am Ende der Linie geht verloren. Ein INSERT-Mode wird durch CTRL-I aufgerufen, jedes alphanumerische Zeichen bewirkt eine Verschiebung des Textes nach rechts und Einfügen an der Cursorposition. Dieser INSERT-Mode wird mit CTRL-O wieder aufgehoben. Die F4-Funktion bewirkt das Einfügen einer Leerzeile. Der gesamte Text wird aus der

bestehenden Position um eine Zeile nach unten verschoben. Sofern die 25 Zeilen des Bildschirms überschritten werden, verschwindet der Text aus dem Screen.

## Löschen von Zeilen

Mit der Funktion CTRL-C wird die Linie, auf welcher sich der Cursor befindet, gelöscht und der Cursor auf den Anfang der Leerzeile gesetzt. CTRL-V bewirkt das Löschen der Zeile ab der definierten Cursorposition bis zum Ende der Zeile. Beide Funktionen arbeiten mit der momentanen Zeilenlänge. Die STOP-Taste hat die gleiche Funktion wie CTRL-C.

## DELETE-Funktion

Zum Löschen von Texten sind ebenfalls verschiedene Möglichkeiten offen. Die DEL-Taste des Rechners hat die gleiche Funktion wie im BASIC. Mit CTRL-D wird ein Charakter gelöscht, anstelle des gelöschten Zeichens wird ein Space eingefügt. Die Funktionstaste F3 löscht die Linie.

Während mit den bisher beschriebenen Funktionen der gelöschte Text aus dem Screen entfernt wird und unwiederbringlich verloren ist, ist mit im nächsten Block aufgeführten Befehlen auch eine Ueberführung in ein anderes Screen oder die Speicherung von Charaktern und ganzen Zeilen möglich.

## CUT- und PASTE-Funktionen

Es stehen fünf Befehle zur Verfügung, die mittels vier Funktionstasten und einem Controllcommand ausgelöst werden. In einem Buffer, der eine Kapazität von 80 Byte hat, werden die Charakter oder Zeilen zwischengespeichert; ist der Buffer voll, werden die entsprechenden Funktionen blockiert. Einzelne Charakter werden mit der Funktionstaste F7, ganze Zeilen mit F5 aus dem Screen gelöscht und zwischengespeichert. Mit den Funktionstasten F8 und F6 werden diese Daten wieder aus dem Buffer gelesen und an einer beliebigen Stelle eines frei gewählten Screen an der momentanen Cursorposition eingefügt. Mit der POP-A-LINE-Funktion CTRL-P werden die Daten des Buffers gelesen und in die Linie vor der Cursorposition eingefügt. Die Ueberführung von Daten aus dem Buffer in ein anderes Screen ist nur möglich, solange die Zeilenlänge nicht verändert worden ist. Die

gespeicherten Charakter werden nur in ihrem Code, nicht aber in ihrer Farbe gebuffert.

## Die 26. Zeile

Und wo ist die Differenz aus den 1000 Zeichen des Bildschirms und der Speicherkapazität eines Screens von 1024 Zeichen? Die Antwort: in einer «MINI-Linie» gerade unterhalb des Screen. Wenn immer eine LINE-INSERT-Funktion ausgelöst wird, rutschen alle folgenden Zeilen nach unten in die 26. Zeile. Wird die LINE-DELETE-Funktion betätigt, geschieht der umgekehrte Weg. Es ist lediglich noch zu bemerken, das immer die rechten 16 Zeichen der 26. Zeile verloren gehen und mit Space aufgefüllt werden, da diese Zeile nur 24 Charakter enthalten kann.

## Abschluss des EDIT-Mode und Abspeicher

Auch hier bestehen vier Varianten zum Verlassen des EDIT-Mode. Mit CTRL-Z wird der Editor verlassen und die Daten im Diskbuffer abgelegt. Es erfolgt keinerlei Hinweis auf das Updating. Mit CTRL-X erfolgt der gleiche Arbeitsgang, jedoch wird das Screen als «updated» markiert. CTRL-F verlässt ebenfalls den Editor, legt das Screen im Buffer ab und schreibt alle Screen auf Disk, die als «updated» markiert sind. CTRL-L verlässt den Editor, sichert das Screen im Arbeitsspeicher, speichert alle markierten Screen auf Disk und lädt das Screen ab Disk.

## Eigene Begriffe und Befehle

Die Definition neuer Begriffe und Befehle erfolgt durch Voransetzen eines Doppelpunktes vor die eigentliche Definition. Hiermit wird die Definition mittels Compiler im Wörterbuch abgelegt und es erfolgt ein Eintrag in den Header, der Name wird der Definition zugeordnet und alle folgenden numerischen Werte und bereits vorher definierten Begriffe werden in den neuen Begriff eingebaut. In den neuen Begriff einbauen heisst, dass ein Pointer dem neuen Begriff zugeordnet wird, dass mit der Ausführung andere Begriffe oder Befehle verbunden sind, oder dass zum Zeitpunkt der Ausführung der Definition die numerischen Werte auf den Stack gelegt werden. Nach Abschluss der Definition mit dem Semikolon (;) wird der Compile-Mode abgeschlossen und der Interpreter-Mode tritt wieder in Kraft. Alle

Eingaben werden nun wieder direkt ausgeführt.

Die Verschachtelung von neuen Begriffen kann bis zu einer beliebigen Tiefe erfolgen, d.h. definierte Begriffe können in neuen Definitionen beliebig weiter verwendet werden. Beispiel:

```
: CLRSCREEN 147 EMIT ;  
: 3RDLINE CLRSCREEN CR CR ;  
: SHOWNEST 3RDLINE .«Beispiel  
  der Verschachtelung»
```

Die erste Definition CLRSCREEN bewirkt das Löschen des Bildschirms, er wurde in den zweiten definierten Begriff 3RDLINE integriert, welcher nun das Löschen des Bildschirm und ein zweifaches Linefeed bewirkt, dieser wiederum wird im Ausdruck SHOWNEST verwendet.

Die meisten Wörter des FORTH-Vokabulars können in neuen Definitionen verwendet werden, Ausnahme bilden lediglich die Begriffe, die als «immediate» klassifiziert sind. Diese Begriffe setzen den COMPILER-Mode ausser Kraft und führen den Befehl sofort aus. Beispiele solcher Begriffe sind: ASSEMBLER, EDITOR und Verzweigungen wie IF...THEN und BEGIN...UNTIL.

Obwohl nicht die Regel, ist es effizient, FORTH-Definitionen möglichst kurz zu halten. Eine Limite besteht in der Anzahl der Charakter pro Zeile, hier können nicht mehr als 80 Zeichen verwendet werden. Da FORTH eine freie Formatwahl hat, kann eine neue Definition auch aus mehreren Zeilen bestehen. Wird eine neue Definition mit mehr als 80 Zeichen pro Zeile eingegeben, so antwortet der Rechner nach dem Drücken der RETURN-Taste nicht mit OK. Wird eine neue Definition mit dem Semikolon abgeschlossen, so geht das Programm wieder in den Interpreter-Mode und der Rechner gibt ein OK aus.

Wird ein neues Wort definiert, so geht beim Ablegen im Wörterbuch der Quellcode verloren. Es ist daher nicht einfach, nachträglich die verwendeten Wörter, Konstanten, Variablen und Felder in ein Screen zu schreiben. Der LOAD-Befehl eines Screen, wie oben beschrieben, behandelt die Befehle innerhalb des Screen so, als ob sie in einer Linie einer Definition stehen würden. Der Vorteil dieses Verfahrens ist die Möglichkeit der permanenten Sicherung auf Disk.

Wird ein solches Screen mit dem Befehl <SCR> # LOAD geladen, so beginnt der FORTH-Interpreter mit

der Abarbeitung des Screens, so als behandle er eine 1024 Byte-Eingabe. Normalerweise gibt der Rechner nach dem Einlesen der 1024 Byte ein OK aus und kehrt zum Eingabebuffer zurück um auf neue Eingaben zu warten. Findet der Interpreter in einem Screen das Wort «--->» beendet er die Bearbeitung des Screens, sucht das nächste Screen im Arbeitsspeicher oder auf der Disk und speichert dieses Screen im Input-Buffer. Tritt während des Ladevorganges ein Fehler auf, stoppt der Prozess und es wird eine Fehlermeldung ausgegeben. Nach der Ausgabe einer solchen Fehlermeldung kann mit dem Befehl

```
WHERE <RETURN>
```

die Zeile auf dem Bildschirm ausgegeben werden, in der sich der Fehler befindet. Mit einem Pfeil hinter dem Wort, das fehlerhaft ist, wird dieses markiert. Dies funktioniert allerdings nur solange nach der Fehlermeldung keinerlei andere Befehle via Tastatur gegeben werden. Wird z.B. eine Zeile aus dem Screen 20 gelesen, bei dem ein Wort nicht definiert ist

```
: FEHLERMELDUNG CR FEHLER ;
```

und ist das Wort FEHLER im Wörterbuch nicht definiert, so erscheint auf dem Bildschirm

```
20 LOAD 20 ? CAN'T FIND
```

```
WHERE
```

```
: FEHLERMELDUNG CR FEHLER ↑ ;
```

Rechts neben dem Wort LOAD wird die Nummer des Screens wiederholt, das geladen wird.

Ganze Programme oder längere Wörter können nun mit Hilfe des Editors und eines Screens definiert werden. Auf der Originaldisk befinden sich bereits 59 Screen mit einem Assembler, HRG und anderen Utilities. Die beste Art das Arbeiten mit Screens und auf dem Editor zu lernen, ist die Variation bestehender Screens mit dem Editor.

In der nächsten Folge behandeln wir vor allem die Eingabebefehle und die Verarbeitung von Strings. □

Nächsten Monat gibt's wieder

COMPUTER  
MARKT

Abonnement schon bestellt?

# Wie bringe ich Maschinenroutinen in den Speicher des CBM?

**Die bekannteste Art, Maschinenroutinen in den Speicher des CBM zu bringen und in Basic-Programme einzubauen, ist die Möglichkeit, die Codes in DATA-Statements unterzubringen und dann mit Hilfe einer FOR...NEXT-Schleife in den Speicher zu POKEn. Diese Möglichkeit wurde bis anhin in M+K dargestellt. Sie frisst aber sehr viel Speicher, denn jede Ziffer in den DATA-Statements benötigt ein Byte; hinzu kommen die Kommata, die zur Trennung der Zahlen benötigt werden. Das POKEn mittels Zählschleife (FOR...NEXT) ist langsam, und ausserdem stehen die gePOKEten Werte nach dieser Prozedur zweimal im Speicher: einmal in Dezimalzahlen-Form in den DATA-Statements und noch einmal in Form echten Maschinen-Codes dort, wohin sie durch POKE gebracht wurden.**

Es gibt elegantere Möglichkeiten, Maschinenroutinen in den Speicher zu bringen. Der TIM-Monitor, welcher durch SYS54386 (oder SYS1024) aufgerufen wird, eignet sich nur im Direkt-Modus. Soll eine Maschinenroutine zu einem Basic-Programm geladen werden, bietet sich die Anweisung LOAD an. Im CBM-Bedienner-Handbuch steht: «Durch LOAD

## Niklaus Weiss

können auch Maschinenprogramme geladen werden.» Damit dies richtig funktioniert, sind jedoch einige Hinweise zu beachten.

Anweisungen, die nach LOAD (oder DLOAD) stehen, werden nicht ausgeführt. Ferner wird durch LOAD auch der Stapel (Stack) geleert. Die Variablen bleiben jedoch erhalten, wenn LOAD (oder DLOAD) in einem Programm steht. Wird LOAD jedoch im Direktmodus benützt, gehen die Variablen verloren. Die Tatsache, dass LOAD im Programm-Modus die Variablen stehen lässt und auch die Speicherzeiger in den Zellen (40) bis (53) nicht verändert, kann zum Laden von Maschinenroutinen durch Basic-Programme ausgenützt werden.

Grundsätzlich kann eine Maschinenroutine überall im Speicher stehen. Sehr beliebt ist die Möglichkeit, solche Routinen in einem der Kassetten-Puffer zu speichern. Grössere Routinen stehen oft im oberen Bereich des Arbeitsspeichers. Schliesslich kann man ein Maschinenprogramm auch unmittelbar hinter ein Basic-Programm hängen. Zuerst soll jedoch gezeigt werden, wie eine Maschinenroutine in den oberen Bereich des Arbeitsspeichers oder den Kassettenpuffer gebracht wird. Dazu

müssen am Anfang eines Programms die folgenden Anweisungen stehen:

```
10 ifpg=0thenpoke52,10:poke53,hi:clr
20 pg=pg+1:onpggoto30,40,50
30 dload»routine.1»
40 dload»routine.2»
50 rem beginn des basic-programms
```

Sollen die Maschinenroutinen für ein ganzes Programm-Paket gebraucht werden, das mit [shift] und [run] gestartet wird, speichert man auf der Disk folgendes kleine Programm «init», das im Directory an erster Stelle steht:

```
10 ifpg=0thenpoke52,10:poke53,hi:clr
20 pg=pg+1:onpggoto30,40,50
30 dload»routine.1»
40 dload»routine.2»
50 dload»basic-programm»
```

Am Anfang des zu ladenden Basic-Programms müssen in diesem Fall folgende Anweisungen stehen:

```
100 ifpeek(42)=peek(44)and
    peek(43)=peek(45)then120
100 poke42,peek(201):poke43,
    peek(202):clr
120 rem programmbeginn
```

Erklärung: Nach dem Kaltstart durch RUN sind alle Variablen gelöscht. Die Anweisung in Zeile 10 (IFpg=0THEN...) wird also ausgeführt. Mit POKE52,10:POKE53,hi wird der für die Maschinenprogramme benötigte Platz reserviert. Dabei steht 10 für das niederwertige Byte (low) und hi für das höherwertige Byte (high). Da nach RUN die Variablen gelöscht sind, müssen anstelle von 10 und hi die entsprechenden Konstanten stehen. Die folgende Anweisung CLR stellt die Zeiger in den Zellen (42) bis (49) auf den so definierten Speicherbereich ein. Wird

zur Speicherung von Maschinenroutinen nur der Kassettenpuffer verwendet, so erübrigt sich diese Zeile.

In Zeile 20 wird zu der Variable pg, die nun auf NULL steht, eins dazugezählt. Der folgende Sprungverteiler ONpgGOTO... sorgt dafür, dass zuerst die Maschinenroutinen geladen werden. Nach jedem DLOAD wird das Basic-Programm wieder von vorn begonnen. Der Wert der Variablen pg bleibt aber erhalten, so dass die Anweisungen in Zeile 10 nicht mehr ausgeführt werden. Korrekt kommt nun das erste LOAD an die Reihe. Danach werden - falls vorhanden - die weiteren LOADs ausgeführt. Zuletzt wird noch das Basic-Programm gestartet oder geladen. Im zweiten Fall - wenn also das Programm «init» verwendet wird, das sich nur n seiner letzten Zeile vom anderen unterscheidet - müssen im geladenen Basic-Programm die in Zeilen 100 bis 120 dargestellten Anweisungen stehen. Die Erklärungen dazu stehen auf Seite 181 im CBM-Handbuch.

Alle diese zusätzlichen Programmzeilen können eingespart werden, wenn die Maschinenroutinen direkt hinter das Basic-Programm gehängt werden. Dazu sollte aber ein NEWTIM-Monitor vorhanden sein. Das Basic-Programm wird normal erstellt, und für die Start-Adresse werden hinter den SYS-Befehl oder in die Zuweisung für die hinter dem SYS-Befehl stehende Variable vorläufig vier oder fünf Nullen geschrieben. Danach wird das Programm zur Sicherheit auf Disk gespeichert.

Jetzt ruft man den NEWTIM-Monitor auf. Mit dem Kommando H,0401,nnnn,%00,00,00% [return] wird das Ende des Basic-Programms gesucht. Hierbei ist nnnn das Ende des Such-Bereichs; [return] symbolisiert die RETURN-Taste. Die auf Disk vorhandene Maschinen-Routine hat eine Startadresse, die grösser ist als die Endadresse des Basic-Programms. Sie wird nun geladen mit L«d:Routine»,08 [return] (d steht für die Laufwerknummer, Routine für den Programmnamen und 08 für die Gerätenummer. Mit dem Kommando CU oooo,pppp,qqqq wird die Maschinenroutine ans Basic-Programm angehängt. Dabei ist oooo die Anfangsadresse, pppp die Endadresse +1 des Maschinenprogramms und qqqq die Zieladresse, wohin das Maschinenprogramm kopiert werden soll. Da die Maschinen-Routine ans Basic-Programm gehängt werden soll, wird als Zieladresse die Stelle hinter den drei Nullen angegeben,

die das Ende des Basic-Programms kennzeichnen.

Sollten in der Maschinenroutine Tabellen vorhanden sein, darf das Kommando CU nur verwendet werden, falls diese Tabellen am Ende der Routine sind und mit einem Code beginnen, der kein 6502-Befehl ist (die Erklärung dazu steht im NEWTIM-Handbuch). Nun rechnet man die neue Anfangsadresse der Maschinenroutine von hexadezimal in dezimal um mit dem Kommando \$qqqq, wobei qqqq die umzurechnende Hex-Zahl ist. Den nun auf dem Bildschirm erscheinen Dezimalwert notiert man sich. Mit dem Kommando M,xxxx,yyyy kann man das Ende des so zusammengesetzten Programmes suchen. Dabei ist xxxx die Anfangs- und yyyy die Endadresse. Nach dem Programmieren folgen auf dem Bildschirm nur noch Monitorzeilen mit lauter aa.

Zum Speichern wird nun das Kommando S@d:Name»,08,0401,zzzz [return] verwendet, wobei zzzz die Endadresse +1 des Gesamtprogrammes ist. Somit befindet sich das Gesamtprogramm auf der Diskette. Mit dem Monitor-Kommando G fd16 [return] (CBM8032) wird ein Total-Reset ausgelöst und hernach das Gesamtprogramm neu normal mit DLOAD«Name» geladen. Dadurch wird der Programm-Ende-Zeiger hinter das Ende des Gesamt-Programmes gerichtet.

Das kann man aber auch wie folgt erreichen: Nachdem man das Ende des Gesamt-Programmes gesehen hat, rechnet man nieder- und höherwertiges (low und high) Byte der Endadresse +1 separat aus: \$ 00hi [return] \$ 0010 [return]. In allen Hex-Zahlen stellen die vorderen zwei Stellen das high-Byte und die hinteren Stellen das low-Byte dar. Hier wird auch high wie low in die hinteren Stellen gegeben, um zwei Acht-Bit-Werte zu erhalten. Der Monitor wird danach mit X [return] verlassen. Die beiden Dezimalwerte werden nun mit POKE42,10:POKE43,hi:clr in den Programm-Ende-Zeiger gebracht und mit CLR auch die übrigen Zeiger richtig eingestellt.

Anstelle der Nullen in der Startadresse im SYS-Befehl wird die notierte Startadresse eingegeben und [return] gedrückt. Das Programm darf jetzt keinesfalls grösser oder kleiner werden, weil sonst die Maschinenroutine nicht mehr lauffähig ist. Aus diesem Grunde muss der Platz für die Startadresse im SYS-Befehl mit Nullen vorbelegt werden, denn Änderungen, die sich nicht auf die Länge des Programmes aus-

wirken, dürfen vorgenommen werden. Ist die Start-Adresse vierstellig und hat man fünf Nullen vorgegeben, lässt man die vorderste Null stehen. Anschliessend muss das Programm natürlich wieder auf Disk gespeichert werden. Dies geschieht mit dem Kommando DSAVE«@Name»: Der «Klammeraffe» vor dem Dateinamen bewirkt, dass die alte Programm-Version mit dem gleichen Namen überschrieben wird. Mit dem Kommando SAVE oder dem Monitor-Kommando S ist dieses Uberschreiben ebenfalls möglich. Man gibt SAVE«@d:NAME»,8 oder S«@d:NAME»,08,... ein. Hierin ist d die Laufwerknummer, vor welcher der «Klammeraffe» steht. So kann man sich das Kommando SCRATCH ersparen. Nach dem letzten Speichern startet man das Programm. Hat man alles korrekt gemacht, wird man sich vom richtigen Funktionieren des Gesamtprogrammes bestens überzeugen können.

Auch im Direktmodus kann LOAD oder DLOAD zum Laden von Maschinenroutinen gebracht werden. In diesem Fall müssen die Maschinenroutinen immer vor dem Basic-Programm geladen werden. Kommen die Maschinen-Routinen in den oberen Bereich des Arbeitsspeichers zu stehen, muss wiederum Platz reserviert werden mit POKE52,10:POKE 53,hi. Danach folgt das Kommando NEW, denn nun muss auch der Programm-Anfangs-Zeiger wieder richtig eingestellt werden. Werden mehrere Maschinen-Routinen geladen, muss man nach jedem Ladebefehl NEW eingeben. Zuletzt wird das Basic-Programm geladen, aber diesmal folgt natürlich kein NEW!

Wer Maschinenroutinen auf die oben beschriebenen Arten mit LOAD in den Speicher bringt, wird begeistert sein. Anstelle des langen Wartens, das bei der Methode mit DATA-Statements nötig ist, rauscht nur kurz das Floppy-Laufwerk, und schon ist (sind) die benötigte(n) Maschinenroutine(n) im Speicher. Wird das (oder werden die) Maschinenroutine(n) direkt ans Basic-Programm gehängt, geschieht das Laden von Basic-Programm und Maschinenroutine(n) sogar in nur einem Arbeitsgang. Ausserdem spart man den durch die DATA-Anweisungen benötigten kostbaren Speicherplatz. Die Erklärungen dieses Artikels dürften sinngemäss für alle CBM-Computer sowie auch für den VC-20 und den C-64 gelten. Alle hier angegebenen Adressen beziehen sich jedoch auf den CBM 8032. □

## COMPUTER SPLITTER

### Herbst

(261/eh) Herbst wird es offenbar auch im Computerblätterwald. Nicht nur in Deutschland, wo sich die Bilanzen diverser einschlägigen Publikationen rot zu verfärben beginnen, sondern vor allem auch in den USA hat der Blätterfall jetzt richtig eingesetzt. Diesen Sommer und Herbst traf es den gut entwickelten Baum der Computerzeitschriften besonders hart, verlor er doch so auflagestarke Blätter wie «Personal Software», «List», «Computers and Peripherals», «Softalk for the IBM», «Personal Computer», «St. Mac», «St. Game», «Microcomputing» und «Microsystems». Würde man beim Zusammenkehren alle Blätter einzeln umdrehen und genau anschauen, könnte man sicher noch andere, auch bei uns recht bekannte Zeitschriftentitel finden. Auch in Europa - und da besonders in Deutschland - hat sich die Zahl der Computerpublikationen in den letzten eineinhalb Jahren ungeheuer vermehrt und mit überdimensionierten Redaktionen wurde Jagd auf den verunsicherten Inserenten gemacht. Sowohl alteingesessene Verlagshäuser als auch geschäftstüchtige Newcomer wollten am Mikroboom teilhaben. Doch die tatsächlich verkauften Zeitschriften hinken den gewünschten Absatzerwartungen ganz gewaltig hinterher. Wie stark bei uns der Herbstwind die Blätter schütteln wird, lässt sich wahrscheinlich schon bald erkennen. □

### Farbiger, fetter Macintosh

(264/eh) Wird Apple-Computer an der nächsten Aktionärsversammlung im Januar einen Macintosh mit Farbbildschirm präsentieren? Wird auf diesem Gerät eine Mac-Version von Symphonie, dem Software-Paket von Lotus, laufen? Offene Fragen, die durch ein Gerücht genährt werden, wonach die erste Vorversion von Mac-Symphonie an Apple ausgeliefert worden sein soll. Symphonie ist ein Software-Paket, das bis heute nur für den IBM-PC und seine Verwandten verfügbar war. □

Integration  
richtig  
verstanden

Software-Probchen für Sie. MULTIPLAN, die Planungshilfe; MS-CHART, der neue Standard für Präsentationsgrafiken, und MS-WORD, das überragende Textverarbeitungsprogramm: Sie alle kommen zum Ausprobieren zu Ihnen über Ihre Händler Computertechnik für Manager AG und Industrade AG. Probieren geht eben über Studieren. Viel Spass!

So sparen Sie 256 KB. Sind 640 KB wenig oder viel für Ihren PC? Da gibt es Mainframes und Minicomputer, die niemals eine derartige Kapazität hatten und trotzdem recht komplexe Aufgaben lösen konnten. Das Geheimnis bestand in der optimalen Ausnutzung des vorhandenen Speicherplatzes. Und genau da hapert es heute bei vielen Software-Paketen. Sicherlich gibt es einige, vor allem im kommerziellen Bereich, die nicht für PC's, sondern für Grossrechner konzipiert sind und nie richtig für die neue Umgebung optimiert wurden. Der Software-Markt kennt jedoch Schlimmeres: Programme, die eigens für die PC-Umgebung geschrieben werden und dann allein für ihre Existenz volle 384 KB verschwenden. Die restlichen 256 KB sind dann – wie grosszügig – für die Applikation da. Was bedeutet, dass die Anwendungen, die noch möglich sind, oft keinen realen Praxistest überstehen. Denn die in die 384 KB eingearbeitete sogenannte Integration bläht jedes Modell um ein Vielfaches mit Verwaltungsballast auf. Fazit: Integration um der Integration willen – und das bei äusserst mangelhafter Software-Ergonomie. Wir glauben, dass Funktionalität und Ergonomie wichtiger sind als die heute angebotene Pseudo-Integration. Denn so werden bei Programmen wie MS-WORD, MS-CHART, MS-PROJECT und MULTIPLAN pro PC bis zu 256 KB gespart. Diese Programme und ihre Modelle sind immer noch vielfältiger, als es in der sogenannten integrierten Umgebung möglich wäre. Ausserdem sind die Einzelpakete von Microsoft leistungsfähiger, weil auf einen einzigen Zweck spezialisiert. Daran wird übrigens auch der PC AT so schnell nichts ändern, denn nur im 286-Modus geschriebene Programme können mehr als 640 KB ausnutzen. Unser Rat also: Bevor Sie die nächste Speichererweiterung kaufen, überlegen Sie bitte, ob Sie sie wirklich brauchen.

Ausbildung zum PC-Profi. Microsoft will Ihnen nicht einfach etwas verkaufen, sondern auch das nötige Know-how vermitteln. Als Software-Hersteller haben wir uns dazu einen geeigneten Partner gesucht, und natürlich haben wir Aufbau und Qualität der Seminare kritisch unter die Lupe genommen. Aufgrund dessen können wir Ihnen nun die qualitativ hochwertigen Lehrgänge anbieten, die Sie benötigen.

Sind Sie an Seminaren interessiert? Dann wenden Sie sich bitte an unsere Distributoren: Computertechnik für Manager AG oder Industrade AG. Dort können Sie sich auch anmelden. Ein weiterer Microsoft-Dienst am Kunden!

Die einzig autorisierten Microsoft-Distributoren für die Schweiz:

**CfM Computertechnik für Manager**

Computertechnik für Manager AG  
Baarerstr. 45 · CH-6301 Zug

**industrade ag**

Industrade AG  
Hertiestr. 31 · CH-8304 Wallisellen

**MICROSOFT**

Microsoft GmbH  
Erdinger Landstraße 2  
D-8011 Aschheim-Dornach

## Zu verkaufen

**Sharp MZ-80B**, 64 K, Grafik 1 und 2, einjährig, Drucker P5B, Basic und Pascal, Handbücher. NP Fr. 6200.-, VP Fr. 4300.-.  
☎ 055/63 41 65

**Drucker Brother-HR15** mit Traktor und Tastatur, neuwertig, alles zusammen: Fr. 2000.-. D. Aebi, Storchengasse 2, 8001 Zürich, ☎ 01/211 94 75 abends

**HP-75C, HP-41CV**, Kartenleser und Akku wegen Nichtgebrauchs günstig abzugeben. Alle Geräte neuwertig.  
☎ 056/22 49 62

**Mikro-Computer «Basis-208»** mit NEC-Spinwriter-Schnelldrucker, Software: DB=MDBS, PL/I, Pascal, WordStar, Magic-Wand, Supersort, DataStar etc., Verhandl.basis Fr. 4500.-. ☎ P 061/72 63 50, G 061/27 83 86 (N. Ness)

**1 Sharp PC-1500** mit Drucker-Plotter Fr. 500.-. 1 Heath H8 mit Drucker H14, 24 K und div. Interfacekarten und Terminal ADM 3A Fr. 1000.- (VHB). Ch. Specker, ☎ 01/725 00 31



Melonenstrasse, 9001 St.Gallen

**ENORME HITPREISE**

**FÜR COMPUTER**

**071 22 63 22**

Commodore: Dual-Drive Floppy **CBM 4040**. Sehr wenig gebraucht, VP Fr. 1900.-, ☎ 062/76 27 22 abends

**Apple:** 50 Disk. mit 63 Prog. Darunter: Business-, Sprach-, Betriebs- und Spielprogramme. VP Fr. 500.-. ☎ 053/7 75 76 ab 18 Uhr

**Sorcerer** 48 K mit Floppy 308 K und Monitor wegen Systemwechsel zu verk. für Fr. 1000.- (Neupreis ca. Fr. 5700.-). Alles in gutem Zustand, wenig gebraucht.  
☎ 061/72 27 87 abends

**CBM 8032** mit Epson-Interface und Essex-HRG-Platine + CBM 8050-Floppy + div. Software + Literatur: Fr. 3000.-.  
☎ G 031/62 62 75, P 031/24 25 32

**Kaypro 4**, 64 K, 6 MHz, 2x400 K Floppies, CP/M, WordStar, Supercalc, dBase, MBasic etc. Praktisch neu. Fr. 6275.-.  
☎ 01/825 51 61, M. Frauenfelder

**Apple II+** compatible 64 K, mit Z-80 und 2 Drives integriert, separate Tastatur mit Funktionstasten. 6 Mt. Garantie Fr. 3380.-. 192 K-Virtual-Disk mit Software Fr. 650.-. Epson-Drucker MX-82 Typ III mit Epson-Interface für Apple wenig gebraucht Fr. 850.-. ☎ 01/69 11 08 abends

**HX-20**, Mikrokassette, Hand- und Lehrbuch mit Kassette, Techn. Manual, Forth-Modul mit Handbuch, div. Software und Zubehör, Preis Fr. 1900.-.  
☎ 055/27 48 17 ab 19 Uhr

### Mini-Disketten

Maxi-Qualität

Micro-Preise

Art.-Nr.	Typ	10	20	50	100
5251S	ss/sd	5.20	5.05	4.95	4.80
5251D	ss/dd	6.25	6.05	5.95	5.75
5252S	ds/sd	5.40	5.25	5.15	4.95
5252D	ds/dd	7.10	6.85	6.75	6.55

**Electronix Versand, Postfach A-123  
8052 Zürich, Telefon 01/301 29 23**

**Basis 108** für Apple u. CP/M Progr. 40/80 Zeichen, par. u. ser. Interf., ext. Tastatur, 2 Diskdrives 2seitig mit Trackcounter, Standard Software, VHB Fr. 3500.-. ☎ 061/76 42 86

**Apple //e**, 128 K, 80 Zeichen Monitor BMC, 2 Disk, Joystick, Epson FX-80 mit Interface. Software: Multiplan, Applewriter, PFS-File, Games. A. Brunner, ☎ 01/810 69 94

**AIM 65** (Geh; Spg; 36 K-RAM; Basic; ASM) + HRG-Terminal (512x256x4; Text, Graphic, Tektronix, Logo Modus; Geh; Spg) + 12 Zoll Monitor + Lit. Gesamtpreis: Fr. 2000.-! ☎ G 031/65 36 91, P 031/24 58 79

**PC-1500 Systemerweiterung** Floppy-Disk-Interface VC-1541, A/D-Wandler: 8-Kanal, 12-Bit Parallel-Interface: 16-Bit. TRAMsoft Ambühler & Müller, Rümelbachstr. 52, 8153 Rümlang

**Kugelpopf-Schreibmaschine IBM-60** mit RS-232-Interface. ☎ G 031/97 11 21 int. 16



Seeburgstrasse 18 (Nähe Verkehrshaus) 6002 Luzern  
Telefon 041-31 45 45

Die Nr. 1 in der Zentralschweiz,  
für Mikrocomputer und alles was dazu gehört ...

z. B. offizieller Vertragshändler für



DCT Computer Shop - ein Unternehmen der Data Center Luzern AG

**HP-41CV** plus Drucker (82143A), Barcode- + Kartenleser mit 360 Karten; Module/ROMs: X-Memory, X-Functions, Time, PPC, Math 1, Stat 1 für (NP über Fr. 4000.-) Fr. 2500.-. ☎ 071/41 93 86 abends

## MCC MICRO COMPUTER CENTER

Ihr zuverlässiger Computerpartner.

### Systeme:

APRICOT - COMMODORE - HP -  
MAD - SIRIUS - TELEVIDEO

### Drucker:

BROTHER - EPSON - STAR  
Hardware/Software-Service und Verkauf  
Umfassende Fachliteratur-Auswahl

KLYBECKSTRASSE 76 4057 BASEL  
Telefon 061/32 12 92

**Dualdisk Cumana** mit Netzteil, einzeln umschaltbar 35/80 Tracks, Konvertierprogramm, neu, mit Interface für Apple mit DOS 3.3 + CP/M Fr. 1250.-.  
☎ G 061/55 70 62, P 061/47 51 41

**CBM 8032** + Exbasic II mit Discdrive 2031 LP, Tractor-Printer 4022, Kassettengerät, Floppies und Tisch. Preis kompl. 2900.-. D. Fischer, SH-Str. Diessenhofen, ☎ 053/7 68 12, ab 18 Uhr

Grafikfähiger **Star Delta 15** Matrix-Drucker 160 Z/Sec., Druckbreite 132 Zeichen, 8 K Buffer, 2x96 freiladbare Zeichen inkl. Centronics + RS-232 Schnittstelle. Nur Fr. 1540.-, ☎ 031/24 54 62

**HX-20** mit 32 K-Bytes RAM, Mikrokasstentendrive und geregelterm Netzgerät. Preis: Fr. 1600.- (NP 2600.-),  
☎ 01/821 03 54 abends

## FOTOSATZ ab Ihren Floppys



**Meier+Cie AG Schaffhausen**

Graphisches Unternehmen, Vordergasse 58  
Telefon 053-8 81 11, Werner Bärtschi

**CBM 8032 SK**, CH-Normtastatur, CBM MPP 1361 A4 quer, 150 z/sek., IEEE-488 Kabel P-P, wie neu! VP ca. Fr. 2700.-.  
☎ 032/42 09 83 mittags, 032/25 45 53 abends

**Occasion für IBM-PC:** Software Finanzer II (Privatfinanzen) Fr. 200.-. Auditor für Lotus-Spreadsh. (druckt alle Formeln + Namen) Fr. 150.-. ☎ 01/858 28 87

**1 Floppy Drive** BASF 6106, neu, 250 K Fr. 250.-. 1 Doppel-Floppy Drive Canon in Gehäuse, inkl. Power-Supply, geeignet für CP/M-Rechner usw. 2x320 K, Fr. 1100.-. ☎ 01/945 40 40

**Sharp MZ 721** (64 K) mit Handbuch. Wenig gebraucht. Fr. 850.-.  
☎ 063/46 15 65 (Lothar Jäggi)

**CBM MMF 9000** (6809/6502: Super Pet) mit Floppy 8050, Herbst 1983, NP Fr. 6400.-, Verhandlungsbasis Fr. 5200.-. G. Wettstein, unterer Imm, 9050 Appenzell, ☎ 071/87 27 66



**HP-75**, 10 Monate alt, guter Zustand: Fr. 1600.-. Nicolas Guillet, St. Josef, 6460 Altdorf

**Verbatim-, Verex-, Xidex-** und andere Markendisketten schon ab Fr. 4.-! In allen Formaten für alle Computer! Preisliste sofort anfordern bei M. Rogivue, Schulstr. 7, 8802 Kilchberg, ☎ 01/715 12 10

**Quadboard für IBM PC/XT**, Occasion, 6 Funktionen, komplett mit 256 K, Schworer, 8174 Stadel, ☎ 01/858 28 87

**HP-86** (200 K) inkl. Monitor, Disc-Drive, Epson RX80, Adv. Progr.-ROM, Matrix-ROM, Visi-Calc. 3 Jahre alt. Fr. 4800.- (Neuwert: ca. 13000.-). ☎ 031/98 10 58

## Endlich

– Aus dem grossen Angebot der EDV-Literatur (nebst Randgebieten) erstellen wir für Sie persönlich eine individuelle Übersicht. Sie brauchen uns nur Ihr EDV-Gerät und/oder das spezielle Interessengebiet zu nennen. Freiumschlag erbeten.

**M+C MICRO-COMPUTER GmbH**  
Karlst. 17d, D-4018 Langenfeld K

**CBM 8032** mit Floppy 4040 und Drucker 4022 inkl. Software und Bücher VP Fr. 3500.-. ☎ P 038/51 24 08, G 065/52 25 55

**HP-75C** + 8 K Erw. + Math ROM + HP-IL-Videointerface + div. Solutions-Books: Games, I/O Utilities. Nur Fr. 2000.-. Kein Einzelverkauf. NP ca. Fr. 4500.-. ☎ 031/52 01 25 mittags + abends

### Apple II - IBM PC/XT

Branchen-Lösungen vom Fachmann: FIBU, FAKT, AKTIEN, ABO-SERV, KRED-, DEBI-, LAGER-, ADR-, HAUS-, MITGL-VERW etc. Kompl.-Systeme. >STOCKER-SOFT< ☎ 01/940 04 29

## DCT Computer Shop

Seeburgstrasse 18 (Nähe Verkehrshaus) 6002 Luzern  
Telefon 041-31 45 45

Die Nr. 1 in der Zentralschweiz,  
für Mikrocomputer und alles was dazu gehört ...

z. B. offizieller Vertragshändler für

**IBM PC**

DCT Computer Shop - ein Unternehmen der Data Center Luzern AG

**HP-41CV** komplett mit Magnetkartenleser, Drucker 82143A, Akku, Ladegerät, 120 Magnetkarten und Software. Fr. 1600.- (NP ca. Fr. 2620.-)  
☎ G 052/81 69 56, R. Sauter

**Bis zu 50% Rabatt** auf Sharp Taschencomputer: PC-1500A + CE 150 Fr. 690.- statt 1225.-; CE 161 16 K Mod. Fr. 289.- statt 390.-; PC-1401 4x2 K Fr. 195.- statt 299.-. URSoft, Urs Ribl, ☎ 052/36 19 44

**CBM 3032**, CBM 8050, CBM 4022, Datensette, Software, viel Zubeh. Fr. 3500.- MPF-1 neuw. Fr. 250.-, IEEE Epson Interface Fr. 120.-. ☎ 062/35 16 21 ab 18 Uhr

**Cromemco C-10**, 64 K RAM, CP/M, 390 K Floppy, Textverarb., dt. Handbücher, dBase II, Basic, Lisp und weitere Software, kompl. nur Fr. 4000.-. ☎ 057/44 10 02

**OSBORNE 1**, 64 K, 2x185 Floppy, 5-Zoll-Monitor, CP/M, WordStar, Supercalc, MBasic und CBasic. Versch. Schnittstellen. Verkaufsbasis Fr. 3000.-.  
☎ 056/96 11 27

**Sirius/Victor SBS 1**, 512 K RAM, 2x600 K Floppy, 10 Mte. alt, Fr. 7500.-, Multiplan Fr. 200.-, Rechentext mit Adressverwaltung Fr. 1000.- (NP 3000.-), Matrixdrucker DRE 8910 (220 Z/sec.) Fr. 1900.- (NP 4800.-), ☎ 071/83 25 28 abends ab 18 Uhr

**Sharp MZ80B**, 64 K mit Graphik I Printer MZ80P5, Basic Interpreter mit div. Software. Anlage neuwertig. Preis: Fr. 3000.-  
☎ 033/45 58 35 M. Avondet ab 18 Uhr

**CBM 3032** (32 K gr. Tast.) mit Doppelfloppy (Computhink). Software: RTTY, Dateiverw., Textver., Buchhalt. nur Fr. 1300.-. ☎ 056/96 44 93

## DCT Computer Shop

Seeburgstrasse 18 (Nähe Verkehrshaus) 6002 Luzern  
Telefon 041-31 45 45

Die Nr. 1 in der Zentralschweiz,  
für Mikrocomputer und alles was dazu gehört ...

z. B. offizieller Vertragshändler für

**commodore**

DCT Computer Shop - ein Unternehmen der Data Center Luzern AG

**1 Herculeskarte** zu IBM-PC Fr. 700.-, 1 Lotus 1-2-3 Original, Programm mit Handbuch, für IBM-PC Fr. 700.-. Beides wenig gebraucht. ☎ 071/67 33 66

**Tulip System** wie neu, 256 K, Floppy 750 K, Harddisk 10 MB, deutsche Tastatur.  
☎ 022/20 01 62

**Sirius 1**, 256 K mit 10.7 MB Harddisk, 1x1,2 MB Floppy, CH-Tastatur, 12 Zoll Grafik-Monitor, Matrix-Drucker MT 140S mit Tractor (160 Z/sec.). Software: MS-Basic 86, GW-Basic, Basic-Compiler, Pascal, Cobol, Assembler, Fabs, Autosort, Sysselect, PMate, Pulsar-File-Transfer (IBM/Apricot), WordStar/MailMerge, Multiplan, Graphics Tool-Kit, SI-Rechentext. 1 Jahr alt - wenig gebraucht - Software auf Original-Disketten inkl. Original-Dokumentation. ☎ 01/493 02 21

**Epson HX-20** (32 K, Microcass.) mit Video-Interface (Text + Graphik), Monitor, Zubehör, neuwertig, NP Fr. 3700.-, VP Fr. 1800.-. ☎ 041/41 71 83

## Gesucht

**Handbuch/Literatur** zu Apple Z80-Software bzw. CP/M 2.2. ☎ 072/44 15 76 abends

Gebrauchte, günstige **MICRO-PROFESSOREN MPF-I, MPF-IP** (Microprozessor-System). H. Siegrist, Bruggmatte, 5615 Fahrwangen, ☎ 057/27 12 27

COMACON

## Computer Market

Ankauf und Verkauf von gebrauchten Kleincomputern.

Donnerstag 17.00 – 21.00  
Samstag 10.00 – 16.00

Meinrad-Lienert-Strasse 15  
beim Lochergut, 8003 Zürich  
Tel. 01 462 19 57

**Hilfe!** Besitzer eines guten PC (HP-150) möchte mit einer guten Sprache arbeiten lernen: LOGO. Wer gibt mir einen Tip über existierende Implementationen (wenn möglich DR. LOGO oder IWT-LOGO). Wer kennt LOGO-Lehrgruppen? Bitte anrufen: ☎ 041/47 23 37. Der erste Tip für ein lauffähiges LOGO auf meinem 150er wird mit einem Abo von M+K belohnt!

## Verschiedenes

**Disketten 3M** und Cartridges für PC und komm. Anwendungen. Günstige Preise. Feiertags- und Notfallservice Region Zürich. ☎ 01/825 51 61 M. Frauenfelder

**Grafiksoftware**, Zeichenprogramme für Apple, Epson PX-8, HX-20 und div. Plotter und Matrixdrucker nur Fr. 500.-. Dr. Ulrich Walther AG, Oberallenbergstr. 19, 8708 Männedorf

## Kontakte

**Pocket-Computer-Club** sucht weitere Mitglieder zwecks Informationsaustausch. Clubheft und Treffen. Info gegen Rückporto bei PCC, Postfach 716, CH-8046 Zürich-Schauenberg

Suche Computer-Sachverständigen mit Computer und Plotter zwecks Graphischer Arbeiten. Schriftliche Angebote an: W. Mesaric, Jennershausweg 19, 3098 Köniz

## M+K 84-1

Olympia's PEOPLE grundsolide  
 MODULA-2  
 TI's Profi - der PROFESSIONAL  
 MUK-Test's auf neuem Stand  
 Programmieren mit HRG (7)  
 Automatisches Schraffieren  
 von Flächen  
 Die Programmiersprache C (2)  
 Die Hardware des Sharp PC-1500  
 Frei definierte Zeichen auf HX-20  
 MICRO Z8000 - Die Hardware  
 Aktuelle Meldungen zum IBM-PC  
 Print-Programm in Pascal  
 Hidden Lines  
 Bilddigitalisierung mit C-64  
 Random Access Programmierung (1)



## M+K 84-2

RAINBOW 100+ von DEC - ein  
 verkanntes Genie  
 ALPHATRONIC-PC - klein,  
 aber recht fein  
 SORD BASIC II  
 Epson FX-80 Drucker im Einsatz  
 Spectravideo - MSX zum ersten  
 Die Programmiersprache C (3)  
 Flächenwert eines  
 geschlossenen Polygonzuges  
 Synthetisches Programmieren  
 auf dem HP-41  
 MICRO Z8000 - Die Software  
 Aktuelle Meldungen zum IBM-PC  
 BASIC-Schutz geknackt  
 Das Nadelproblem von Buffon  
 Dateien kopieren mit  
 einer Single-Drive-Floppy  
 Random Access Programmierung (2)



## M+K 84-3

APRICOT made in Europe  
 Der Graphtec-Plotter MP-1000  
 HP-Touch ist mehr als ein Gag  
 ...dicht gefolgt von HP-110  
 Die Programmiersprache C (4)  
 Das Primzahlensieb von Sundaram  
 Toolkits zum PC-1500  
 Radioaktivität für TI-58/59, 66, 99  
 HX-20 löst Quadr. Gleichungen  
 Aktuelle Meldungen zum IBM-PC  
 RENUMBER für MBASIC-Programme  
 Controlcodes für SHARP-Drucker  
 PASCAL zur Berechnung von  
 Hexadezimalzahlen  
 6502-Assembler in BASIC  
 Universal-Plotroutine  
 3D-Rotation eines freidefinierbaren  
 Objektes auf CBM 30XX mit HRG

## M+K 84-4

P2000C - der «Portabel» von Philips  
 DEC-Rainbow 100+, ein erster  
 Zwischenbericht  
 MAD-1: nomen est omen?  
 ZILOG Z8000 - die 32-Bit-Maschine  
 die Programmiersprache C (5)  
 Biorhythmen auf dem HP-41CX  
 PC-1500 Systemsubroutinen  
 Aktuelle Meldungen zum IBM-PC  
 CRT-SOFT-COPY  
 Fourier-Transformation für Praktiker  
 Comic-Figuren mit HRG  
 FORTH auf Commodore C-64 (1)  
 Random Access Programmierung (3)  
 Schreibmaschinenkurs auf VC-20



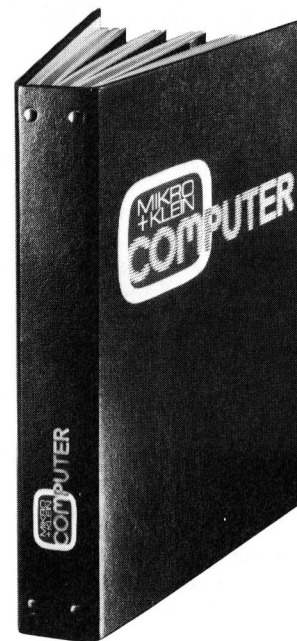
## M+K 84-5

Die Turbo-Sensation  
 EPSON PX-8: Kleiner Wolf  
 im Schafspelz  
 Der FM7, Fujitsu's neuer Mikro  
 FUTURE - System der Zukunft  
 MSX - der Normcomputer  
 Die Jackson-Methode  
 Die Programmiersprache C (6)  
 Casio FP-200 und Canon X-07  
 im Vergleich  
 Kurvendiskussion mit dem HP-41  
 Aktuelle Meldungen zum IBM-PC  
 Datensicherung auf Kassette  
 BASIC-Käfer  
 Springertour auf dem NxN-Brett  
 FORTH auf Commodore C-64 (2)  
 Barcode drucken mit CBM 3022  
 Random Access Programmierung (4)  
 VC-20 löst kubische Gleichungen



## M+K 84-6

CORONA PPC-2 - die Krönung  
 der IBM-Kompatiblen  
 Brother HR-5, klein und vielseitig  
 Die Mikroprozessoren 8086 und 8088  
 TRICOM Miniplotter CPP-114  
 Einführung in APL (1)  
 Die Programmiersprache C (7)  
 Zahlentheorie mit dem HP-41  
 Datum/Zeit-Information und  
 Timer für jeden PC  
 Aktuelle Meldungen zum IBM-PC  
 Künstliche Intelligenz in BASIC  
 IBM quo vadis?  
 Splines: Biagsame Kurvenlineale  
 Ein Proportional-Blocksatz-  
 Druckprogramm  
 3D-Darstellung einer Kugel mit  
 Längen- und Breitenkreisen  
 mit Hidden-Line-Routine  
 FORTH auf Commodore C-64 (3)  
 Wie bringe ich Maschinenroutinen  
 in den Speicher des CBM?



## M+K im praktischen Sammelordner

mit bequemer Stabmechanik  
 für jeweils sechs Ausgaben  
 (also ein ganzer Jahrgang)  
 damit jedes Heft unbeschädigt  
 bleibt. Stabile Ausführung  
 mit einem strapazierfähigen  
 Kunststoffüberzug in blauer  
 Farbe.

Den praktischen Sammelordner  
 erhalten Sie für Fr. 14.50 (inkl.  
 Versandkosten). Bei gleich-  
 zeitiger Bestellung von zwei  
 Exemplaren zahlen Sie nur noch  
 Fr. 27.-. Und so bestellen Sie:  
 Zahlen Sie bitte auf unser  
**Postkonto Luzern 60-27181-0**  
 den entsprechenden Betrag  
 ein und vermerken Sie auf der  
 Rückseite Ihres Einzahlungs-  
 scheins «Sammelordner».

**Mikro+Kleincomputer  
 Informa Verlag AG  
 Postfach 1401  
 CH-6000 Luzern 15**



Im Rahmen unserer Artikelserie «Programmieren mit HRG» hat Ihnen unser Autor Marcel Sutter in M+K 83-6 einige LOGO-Programme, die sogenannte Turtlegrafik erzeugen, in BASIC übersetzt. Eigentlich ist das ein Rückschritt, denn LOGO ist eine weitaus mächtigere und vor allem bessere Sprache als BASIC. Man sollte daher LOGO-Programme nicht in BASIC simulieren. LOGO wurde 1967/68 entwickelt und zunächst nur auf Grossrechenanlagen implementiert. LOGO wurde als geeignete Computersprache für Kinder und Computerneulinge, nicht aber als «kindliche Sprache» entworfen. Sie sollte einerseits dem Anfänger, wie jung auch immer er sein möge, einen einfachen Einstieg mit raschem Anfangserfolg ermöglichen und andererseits dem professionellen Programmierer optimale Möglichkeiten im Entwurf komplexer Programme und im Handling von Datenstrukturen bieten. Wir beginnen in M+K mit einer neuen Artikelserie über LOGO.

Wenn man in Pascal programmiert, ist es immer wieder aufwendig, die zu editierende Zeile je nach Strukturtiefe einzurücken. Ein weit aus grösserer Aufwand ist das nachträgliche Einfügen bzw. Löschen von Strukturbefehlen in bestehenden Pascal-Dateien. Alle Zeilen innerhalb des geänderten Strukturblockes müssten mit dem Editor um eine Strukturtiefe verschoben werden. Wir stellen Ihnen das PSCT (Pascal-Source-Code-Transformer)-Programm vor, das eine wesentliche Erleichterung ist, um Pascal-Dateien normiert und strukturiert einzurücken. Die sich daraus ergebenden Vorteile sind: geringerer Zeitaufwand beim Erstellen von neuen Pascal-Dateien; einheitliche und übersichtliche Gestaltung; leichtes Auffinden von Strukturfehlern und schnelles Einfügen

## Programmierberichtigung M+K 84-5

Im Programm «Lösung kubischer Gleichungen mit dem VC-20» sind zwei Zeilen fehlerhaft, da der verwendete Durcker das Zeichen  $\pi$  nicht ausgedruckt hat. Es handelt sich um die Zeilen 1620 und 1660 (Seite 92). Die Zeilen müssen wie folgt lauten:

$$1620 \text{ W}=\text{ATN}(\text{TW})\cdot 180/\pi$$

$$1660 \text{ :X(I)}=\text{COS}(((\text{W}+\text{IM})/3\cdot\pi/180))\cdot \text{R}$$

bzw. Löschen von neuen Strukturbefehlen.

PEARL gehört zur Gruppe der höheren Programmiersprachen und wird seit 1969 in Deutschland entwickelt. Der Name PEARL (Process and Experiment Automation Realtime Language) lässt auf das Haupt Einsatzgebiet dieser Programmiersprache schliessen: Echtzeitaufgaben und Prozess-Steuerungen jeder Art, wie z.B.: Kommunikationsprozesse zu Informationssystemen, Steuerung wissenschaftlicher Experimente oder Steuerung und Ueberwachung industrieller Fertigungsprozesse. PEARL hat gegenüber anderen Systemim-

Nächsten Monat gib's wieder

# COMPUTER MARKT

Abonnement schon bestellt?

plementierungssprachen wie Concurrent-Pascal, Modula oder ADA erhebliche Vorteile in der Echtzeitprogrammierung. Im Gegensatz zum Prozess-Fortran besitzt PEARL direkte Sprachelemente zur Prozess-E/A (Steuerung) und zeitlichen Programmabwicklungen. Diese Vorzüge garantieren der PEARL-Software ein hohes Mass an Portabilität.

Unter dem Namen CE-153 Software-Board existiert ein Peripheriegerät zum Sharp PC-1500, von dem recht wenig bekannt ist. Dies erstaunt, könnte es doch dem Anwender als universelles Menü-Auswahlgerät, als Maus, Kontaktschirm oder Grafiktablett dienen. Unser Erfahrungsbericht zeigt mögliche Anwendungsgebiete und erwähnt auch einige mögliche Gründe für die geringe Popularität dieser System-Erweiterung.

Machen Sie Bekanntschaft mit einer neuen Programmiersprache. BOOLEMAC ist ein Programmiersystem für Maschinensteuerungen, welches auf dem EPSON HX-20 wie ein normales Basic-Programm - mit lauter Kommentarzeilen - erstellt wird. So lassen sich die Möglichkeiten des Basic-Editors (z.B. Renum, Auto, Load, Merge usw.) alle ausnützen. Das so erstellte Programmfile wird dann vom BOOLEMAC-Compiler in das entsprechende Maschinenprogramm umgesetzt und via HX-20 auf die Prozessorkarte übertragen und dort ausgeführt. Der gesteuerte Prozessor lässt sich über den HX-20 überwachen.



## Das Kleincomputer-Magazin

6. Jahrgang

ISSN 0251-0006

## IMPRESSUM

### Verlag, Redaktion, Inserate

Mikro+Kleincomputer Informa Verlag AG  
Seeburgstrasse 12, 6000 Luzern 15  
Telefon 041 - 31 18 46, Tx 72 227 (dcl ch)

### Postanschrift:

Postfach 1401, CH-6000 Luzern 15

### Postcheck-Konten:

Luzern 60 - 27181-0  
Stuttgart 3786-709 (BLZ 600 100 70)  
Wien PSK 7975.035

### Verlagsleitung

Hans-Jürgen Ottenbacher

### Redaktion

Eric Hubacher, El.-Ing. HTL (verantwortlicher Redaktor), Peter Fischer, Leopold Asböck, Ernst Erb, Dr. Bruno Stanek, Heinz Kastien, Dipl.-Ing.

### Manuskripte und Copyright

Manuskripte werden von der Redaktion entgegengenommen. Die Zustimmung zum Abdruck wird vorausgesetzt. Für unverlangt eingesandte Manuskripte wird keine Haftung übernommen. Mit der Zustellung von Manuskripten anerkennt der Autor die Copyrightbestimmungen des Verlages. Mit der Annahme von Manuskripten durch die Redaktion und der Bestätigung durch den Verlag hat dieser das Recht zur exklusiven Veröffentlichung der entsprechenden Beiträge auch in anderen verlagseigenen Publikationen sowie zur Übersetzung in andere Sprachen erworben. Veröffentlichte Beiträge werden Eigentum des Verlages. Presstexte werden nicht bestätigt. Die Publikation von Pressemitteilungen über neue oder wesentlich verbesserte Produkte ist eine Dienstleistung des Verlages. Ueber die Auswahl der Texte und Bilder, Kürzungen und Umformulierungen sowie deren Präsentation entscheidet die Redaktion. Ein Recht auf Veröffentlichung besteht nicht. Für die Veröffentlichung wird keine Gewähr oder Garantie übernommen, auch nicht dafür, dass die verwendeten Schaltungen, Firmennamen und Warenbezeichnungen usw. frei von Schutzrechten Dritter sind. Die Verwendung von Information erfolgt auf eigenes Risiko. Mit Verfassernamen gekennzeichnete Beiträge geben nicht unbedingt die Meinung der Redaktion wieder.

Jeder Nachdruck, auch auszugsweise, sowie Vervielfältigungen oder sonstige Verwertung von Texten aus MIKRO+KLEINCOMPUTER nur mit schriftlicher Genehmigung des Verlages und unter voller Quellenangabe.

© Mikro+Kleincomputer Informa Verlag AG

Im gleichen Verlag erscheint auch der aktuelle schweizerische COMPUTERMARKT

### Erscheinungsweise:

zweimonatlich (gerade Monate)

### Bezug:

Jahresabonnement Fr. 42.-- (inkl. Versand und Porto). Ausland (Europa) Fr. 48.--. Abbestellung ist durch schriftliche Kündigung jeweils 8 Wochen vor Ablauf des laufenden Bezuges möglich. Der Abonnementsbetrag ist nach Erhalt der Rechnung zur Zahlung fällig.

**Nachbezug:** SFr. 8.-- pro Heft

**Inserate:** nach Tarif Nr. 6 gültig ab 1.1.85

**Inserateservice:** Markus Kappeler

**Auflage:** 14'000 Exemplare

**Druck:** Unionsdruckerei AG Luzern

Printed in Switzerland

# Leserdienst-Kontaktkarte

Ich bitte Sie um weitere Informationen zu der in MIKRO+KLEINCOMPUTER 84-6 auf Seite \_\_\_\_\_ erschienenen  Anzeige  redaktionellen Besprechung über Ihr Produkt:

**Ich wünsche:**

- Prospekt/Datenblatt
- Preisliste
- schriftliches Angebot
- telefonische Kontaktaufnahme
- technisches Gespräch

**Einsatzbereich**

- Industrie
- Handel
- Ingenieurbüro/Labor
- Selbständiger Beruf
- Hochschule/Institute
- Behörde/öffentliche Verwaltung

**Branche**

- Elektronik
- Elektrotechnik
- Maschinen- und Fahrzeugbau
- Forschung/Entwicklung
- Chemische Industrie
- Verkehrs- und Nachrichtenwesen
- Energie- und Wasserversorgung
- Feinmechanik/Optik
- Ingenieurbüro
- Handel/Dienstleistung
- Hochschule/Institute
- Behörde/öffentliche Verwaltung

**Funktion im Betrieb**

- Unternehmungsleitung
- Forschung/Entwicklung
- Konstruktion/Labor
- Produktion/Service
- Einkauf
- Sonstige

**Betriebsgrösse**

- 1 - 20 Beschäftigte
- 21 - 50 Beschäftigte
- 51 - 100 Beschäftigte
- 101 - 500 Beschäftigte
- über 500 Beschäftigte
- Behörde/Institute/usw.

Bitte vergessen Sie nicht, umseitig Ihre Adresse sowie die gewünschte Firmenanschrift einzutragen. Danke.

# Leserdienst-Kontaktkarte

Ich bitte um weitere Informationen zu der in MIKRO+KLEINCOMPUTER 84-6 auf Seite \_\_\_\_\_ erschienenen  Anzeige  redaktionellen Besprechung über Ihr Produkt:

**Ich wünsche:**

- Prospekt/Datenblatt
- Preisliste
- schriftliches Angebot
- telefonische Kontaktaufnahme
- technisches Gespräch

**Einsatzbereich**

- Industrie
- Handel
- Ingenieurbüro/Labor
- Selbständiger Beruf
- Hochschule/Institute
- Behörde/öffentliche Verwaltung

**Branche**

- Elektronik
- Elektrotechnik
- Maschinen- und Fahrzeugbau
- Forschung/Entwicklung
- Chemische Industrie
- Verkehrs- und Nachrichtenwesen
- Energie- und Wasserversorgung
- Feinmechanik/Optik
- Ingenieurbüro
- Handel/Dienstleistung
- Hochschule/Institute
- Behörde/öffentliche Verwaltung

**Funktion im Betrieb**

- Unternehmungsleitung
- Forschung/Entwicklung
- Konstruktion/Labor
- Produktion/Service
- Einkauf
- Sonstige

**Betriebsgrösse**

- 1 - 20 Beschäftigte
- 21 - 50 Beschäftigte
- 51 - 100 Beschäftigte
- 101 - 500 Beschäftigte
- über 500 Beschäftigte
- Behörde/Institute/usw.

Bitte vergessen Sie nicht, umseitig Ihre Adresse sowie die gewünschte Firmenanschrift einzutragen. Danke.

# Leserdienst-Kontaktkarte

Ich bitte Sie um weitere Informationen zu der in MIKRO+KLEINCOMPUTER 84-6 auf Seite \_\_\_\_\_ erschienenen  Anzeige  redaktionellen Besprechung über Ihr Produkt:

**Ich wünsche:**

- Prospekt/Datenblatt
- Preisliste
- schriftliches Angebot
- telefonische Kontaktaufnahme
- technisches Gespräch

**Einsatzbereich**

- Industrie
- Handel
- Ingenieurbüro/Labor
- Selbständiger Beruf
- Hochschule/Institute
- Behörde/öffentliche Verwaltung

**Branche**

- Elektronik
- Elektrotechnik
- Maschinen- und Fahrzeugbau
- Forschung/Entwicklung
- Chemische Industrie
- Verkehrs- und Nachrichtenwesen
- Energie- und Wasserversorgung
- Feinmechanik/Optik
- Ingenieurbüro
- Handel/Dienstleistung
- Hochschule/Institute
- Behörde/öffentliche Verwaltung

**Funktion im Betrieb**

- Unternehmungsleitung
- Forschung/Entwicklung
- Konstruktion/Labor
- Produktion/Service
- Einkauf
- Sonstige

**Betriebsgrösse**

- 1 - 20 Beschäftigte
- 21 - 50 Beschäftigte
- 51 - 100 Beschäftigte
- 101 - 500 Beschäftigte
- über 500 Beschäftigte
- Behörde/Institute/usw.

Bitte vergessen Sie nicht, umseitig Ihre Adresse sowie die gewünschte Firmenanschrift einzutragen. Danke.

## An alle Noch-nicht- Abonnenten, Zweitleser und Am-Kiosk- Käufer

Haben Sie sich schon einmal überlegt, welche Vorteile Ihnen ein persönliches M+K-Abonnement bringt?



### Einfacher geht's nicht.

M+K erhalten Sie ohne langen Umweg frei Haus und druckfrisch per Post. Versandkosten und Porto übernehmen wir.



### Bis zum 31. 12. 1984

Sparen Sie erst noch gegenüber dem Einzelverkaufspreis. Im Abonnement erhalten Sie sechs Hefte zum Preis von fünf.



### Bequemer geht's nicht.

Prompte Lieferung. Sie erhalten alle Hefte lückenlos und sicher an Ihre Postanschrift und erst noch früher als am Kiosk oder im Firmenumlauf.



Mikro+Kleincomputer  
Informa Verlag AG  
Postfach 1401  
CH-6000 Luzern 15

### Leserdienst-Kontaktkarte

Bitte genaue Anschrift der Firma angeben, von der Sie weitere Informationen wünschen. Danke.



Name

Vorname

Firma/Institut

Strasse

PLZ/Ort

Telefon

bitte  
frankieren

### POSTKARTE

Firma

Strasse

PLZ Ort

Mit Programm-Listings für  
Commodore VC-20, C-64,  
Sharp PC-1500 und Apple II

## Programmieren mit HRG

Ein leicht verständlicher Lehrgang mit 40 Grafik-Programmen ist jetzt als Buch erhältlich (288 Seiten, A5 Paperback Fr./ DM 45.-)

Wer kennt sie nicht, die raffinierten Demo-Programme, die in jedem Computershop oder auf Computer-Ausstellungen die Aufmerksamkeit auf sich ziehen. Meist handelt es sich um grafikfähige Spielprogramme oder um wunderschöne dreidimensionale Darstellungen von Funktionen. Wer dann aber den Wunsch hat, ähnliche Programme auf seinem Computer selbst zu entwickeln, kommt sehr rasch in Schwierigkeiten und gibt seine Bemühungen vermutlich bald einmal auf. Selbst erhältliche Programm-Listings für hochauflösende Grafik nützen oft wenig, da diese Programme immer nur für ein ganz bestimmtes Grafik-System geschrieben sind und die Programmautoren immer alle Tricks ausnützen, die in ihrem System drin liegen. Ein Umschreiben auf sein eigenes Grafik-System ist mühsam und allzu oft gar nicht möglich.

Im Buch «Programmieren mit hochauflösender Grafik» wird nun jeder interessierte Computer-Anwender anhand von vierzig Kurzprogrammen schrittweise in das Programmieren mit HRG eingeführt. Die vorgestellten Programme umfassen meist weniger als 30 Zeilen, sind alle in Microsoft-BASIC geschrieben und verwenden nur die geläufigsten BASIC-Befehle. Alle Programme sind strukturiert, können top-down gelesen werden, sind selbsterklärend und verwenden immer die gleichen Variablen. Es werden darin nur zwei Grafik-Befehle verwendet, die sich für jedes Computer-System adaptieren lassen.

Mikro+Kleincomputer  
Informa Verlag AG  
Postfach 1401  
CH-6000 Luzern 15

### Leserdienst-Kontaktkarte

Bitte genaue Anschrift der Firma angeben, von der Sie weitere Informationen wünschen. Danke.



Name

Vorname

Firma/Institut

Strasse

PLZ/Ort

Telefon

bitte  
frankieren

### POSTKARTE

Firma

Strasse

PLZ Ort

### Leserdienst-Kontaktkarte

Bitte genaue Anschrift der Firma angeben, von der Sie weitere Informationen wünschen. Danke.



Name

Vorname

Firma/Institut

Strasse

PLZ/Ort

Telefon

bitte  
frankieren

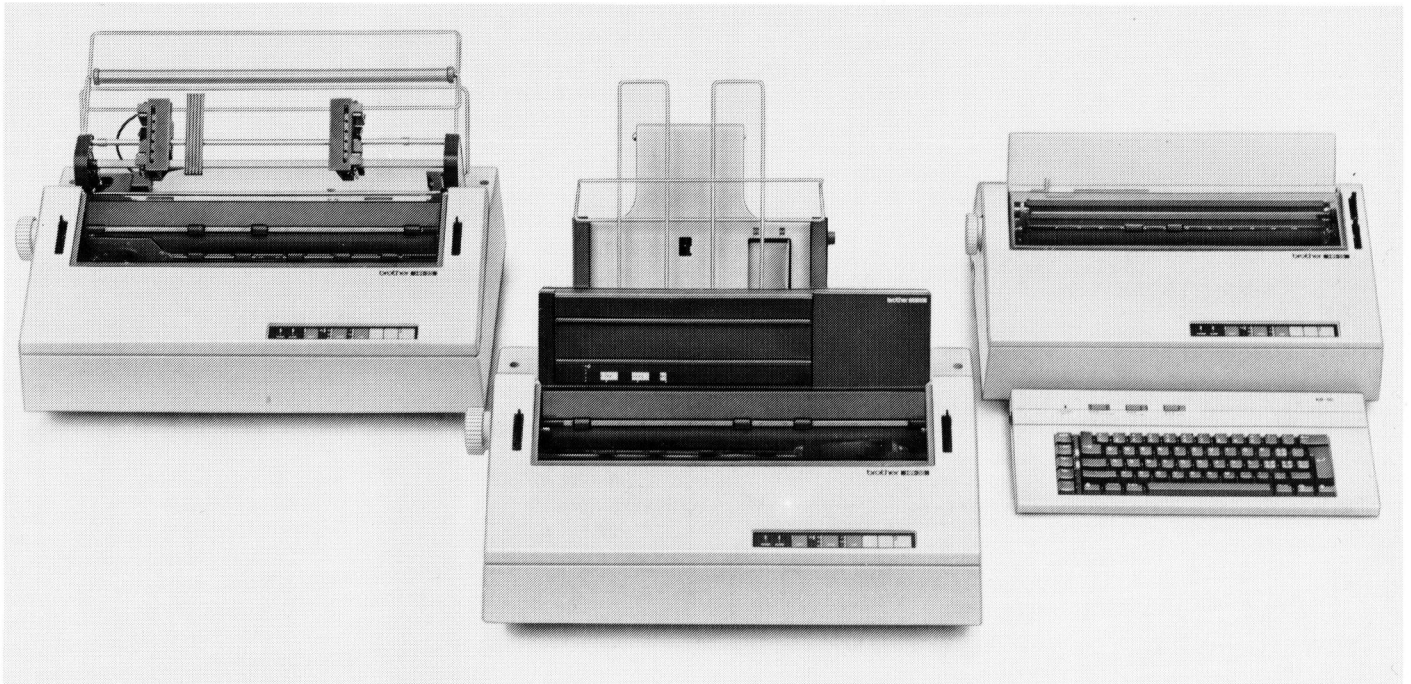
### POSTKARTE

Firma

Strasse

PLZ Ort

# Schön schnell, schön scharf, schön preisgünstig: drei von vielen Vorteilen der neuen Typenraddrucker von Brother.



Die Vorteile der neuen Druckerfamilie von Brother sind offensichtlich. Schon die Bedienungselemente sind sehr übersichtlich an der Frontpartie angeordnet. Einzigartig ist die direkte Programmiermöglichkeit der wichtigsten Druckparameter durch Tastendruck. Besonders augenfällig sind die gestochen scharfen Schriften der Brother-Typenräder, die schon bei den elektronischen Büroschreibmaschinen Begeisterung hervorgerufen haben. Je nach Modell drucken Brother-Schönschreibdrucker fett, hoch und tief, proportional, rot – und unterstreichen automatisch. Ein Pufferspeicher bis zu 7 K sorgt dafür, dass Ihr Bildschirm zum Arbeiten immer frei bleibt. Mit dem Kopierspeicher – eine Brother-Exklusivität – können Sie zum Beispiel Serienbriefe durch Druck auf die Kopiertaste ohne Computerhilfe beliebig duplizieren. Zu den Vorzügen gehören auch Druckwegoptimierung, Bi-Direktionaldruck, Papierdurchlass bis A3 quer und ein abgestimmtes Zubehörprogramm. Schnittstellen: V24 (RS-232C) oder Centronics Parallel – andere auf Anfrage. Was zudem für die Schönschreibdrucker von Brother spricht, sind die IBM PC-Kompatibilität, die lange Lebensdauer von 100 Millionen Funktionen, die extreme Zuverlässigkeit und die optimale Relation von Druckgeschwindigkeit und Schriftqualität. Brother Schönschreibdrucker, wenn Sie ganz schön wirtschaftlich drucken wollen.

## HR-15 – der Lowcost-Printer.

Ein Schönschreibdrucker, den sich jeder leisten kann. Papierbreite bis A4 quer (+ Perforation) 5-K-Pufferspeicher, Zweifarbendruck, Fett- und Proportional-schrift, 15 Z/Sek. Ausbaubar mit Original Brother-Zubehör. Mit der Zusatzastatur zum Beispiel wird der HR-15 im Handumdrehen zur Typenradschreibmaschine mit Korrekturspeicher.  
Fr. 1850.-

## HR-25 – der Printer mit den unbegrenzten Möglichkeiten.

Ein Schönschreibdrucker, den man wirklich überall einsetzen kann, 25 Z/Sek., Papierbreite bis A3 quer, 5-K-Pufferspeicher. Druck in allen Variationen: 2farbig, fett und proportional. Ausbaubar mit Original-Brother-Zubehör.  
Fr. 2950.-

## HR-35 – der High Speed-Printer.

Die technischen Spezifikationen sind gleich wie beim HR-25, jedoch mit einer Druckgeschwindigkeit von 35 Z/Sek. und einem Pufferspeicher von 7 K.  
Fr. 3650.-

## Elektronische Schreibmaschinen mit Interface.

Brother bietet ebenfalls verschiedene elektronische Büroschreibmaschinen mit Schnittstellen an, die sich für den Einsatz als Drucker eignen.

## Original Brother-Zubehör.

Das ist massgeschneidertes Zubehör zu Konfektionspreisen:

- Automatischer Einzelblatteinzug zu HR-15 Fr. 550.-
- Autom. Einzelblatteinzug zu HR-25/HR-35 Fr. 570.-
- Endlospapierzuführung (Traktor) zu HR-15 Fr. 280.-
- Endlospapierzuführung (Traktor) zu HR-25/HR-35 Fr. 305.-
- Tastatur KB-50 zu HR-15 Fr. 395.-

Verkauf durch den Fachhandel

M + K

## Coupon

Bitte senden Sie uns Unterlagen über die Brother-Schönschreibdrucker mit Bezugsquellennachweis.

Name: \_\_\_\_\_

Firma: \_\_\_\_\_

Adresse: \_\_\_\_\_

PLZ/Ort: \_\_\_\_\_

Bitte einsenden an: Brother Handels AG, 5405 Baden

**brother**  
Qualität zu fairem Preis.  
Brother Handels AG, 5405 Baden



## ZEV ELECTRONIC AG COMPUTER DIVISION

Tramstrasse 11, 8050 Zürich, ☎ 01 312 22 67



### MICROCOMPUTER APPLE //e

(bewährt und beliebt)

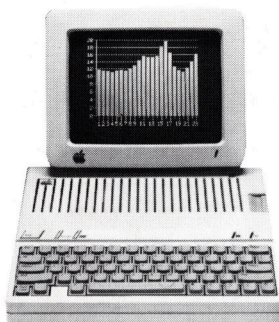
A//e 64K, Grundgerät	2189.-
A//e 64K, 80 Z. Karte, 1 Disk, Monitor	4110.-
A//e 64K, 80 Z. Karte und 64K, Duodisk, Monitor	5325.-



### MICROCOMPUTER APPLE //c

(klein an Grösse –  
aber gross an Leistung)

A//c 128K, Grundgerät	3079.-
Monitor 9"	565.-
Externer Diskdrive	935.-



# Macintosh vom Apple

### MICROCOMPUTER APPLE MACINTOSH

MAC SYSTEM 128K, deutsch, inkl. MacWrite/MacPaint	6500.-
MAC SYSTEM 512K, deutsch, inkl. MacWrite/MacPaint	9438.-
MAC BUNDLE 128K, deutsch, inkl. MacWrite/MacPaint + Drucker Imagewriter	8220.-

MAC numerisches Keyboard	380.-
MAC externer Diskdrive	1450.-
Disketten 3,5" (10 Stück)	145.-

### MACINTOSH-SOFTWARE

MS-Basic	415.-
Multiplan	540.-
Chart	345.-
PFS File/Report	490.-
MegaMerge	285.-
MegaFiler	475.-
Filevision	490.-
MacForth, Level 1	370.-
MacForth, Level 2	615.-



Grosse Auswahl an Literatur!

### Tragbarer Olivetti Personal Computer M 21 Operativ kompatibel mit dem Industrie-Standard

#### MICROCOMPUTER OLIVETTI

M21-Systemeinheit	ab 5600.-
M24-Systemeinheit	ab 4595.-

#### Tastaturen PREH

für A//	ab 335.-
für IBM-PC	598.-
Wordstar-Tastatur für IBM-PC	800.-



Ja, ich möchte mehr wissen über:

Apple//e//c Macintosh Columbia IBM-Zubehör  
Epson/Star Monitoren Bücherkatalog Zubehör

Name .....

Adresse .....

Stadt .....

aus-  
füllen  
und einsenden

#### MONITOREN (color, RGB)

für A//, A//e:	
TAXAN vision-II (12", 510 Pte. Auflösung)	998.-
TAXAN vision-III (12", 640 Pte. Auflösung)	1295.-
SANYO CD-3235 (14", 560 Pte. Auflösung)	1590.-
RGB-Karten	ab 239.-

#### MONITOREN (monochrom)

SANYO DM-2112, grün	275.-
SANYO DM-2212, amber	295.-
TEKO, mit Toneingang grün	335.-
amber	335.-
ZENITH ZVM-123, grün	315.-
ZENITH ZVM-122, amber	335.-
TAXAN KX 1201, grün	359.-
TAXAN KX 1203, amber	380.-
TAXAN KX 1212, f. IBM	479.-

#### Grosse Auswahl an Zubehör:

für IBM-PC:	
TAXAN vision-PC (12", 640 Pte. Auflösung)	1485.-
SANYO CD-3240 (14", 720 Pte. Auflösung)	1850.-
A///-Farbmonitor	999.-
5,25"-Disketten	ab 90.-
Disketten-Ablage-Kästen	ab 39.-
Druckerpapier (1000)	24.50
Etiketten (1000)	ab 12.-
Memory-Erweiterungskarten für IBM-PC	ab 917.-

#### MATRIX-DRUCKER STAR

Gemini 10X	1080.-
Radix-10, NLQ	2515.-
Radix-15, NLQ	2985.-
IBM-Zeichensatz: Aufpreis Gemini	100.-
Aufpreis Radix	200.-

#### MATRIX-DRUCKER EPSON

RX-80 F/T	1250.-
FX-80	1590.-
JX-80 (Farbdrucker)	2390.-
FX-100	2190.-
Zeichensätze für FX-80/100	
IBM/DEC	135.-
Imagewriter-Emul.	135.-
Ständer f. RX/FX-80	145.-
Ständer f. RX/FX-100	155.-

#### TYPENRAD-DRUCKER

STAR Powertype	1595.-
EPSON DX-100	1790.-
BROTHER HR-15	1850.-
BROTHER HR-25	2950.-
IBM/DEC-Zeichensätze für DX-100/HR-15/HR-25	100.-

Schallschluckgehäuse: für RX/FX-80	655.-
für RX/FX-100	700.-
IBM-Zeichensatz: Aufpreis Gemini	100.-
Aufpreis Radix	200.-



ELECTRONIC

# computers