

**KLEINCOMPUTER aktuell**  
PC 1500 – Sharp's BASIC-Grafiker

**Small business JOURNAL**  
Textverarbeitung im  
arbeitspsychologischen Vergleich

**PPC/HHC**  
Barcodes für Peripheriebefehle  
Ausgefileter Datentransfer



**CENTRAL UNIT**

- 16 bit CPU
- 8x ROM
- 128k RAM
- Real Time Clock
- RS-232C Int.
- Parallel Int.

- 512kx ROM expansion
- IEEE Interface
- VIDEOKEY Interface
- RS-232C Current Loop dual channel Int.

- 200k 5.25inch Mini-Diskette
- 200k 5.25inch Mini-Diskette

200k 5.25inch  
MINI-DISKETTE

Alphanumeric  
Keyboard

**Benchmarktests  
und MUK-Testprogramme  
für Ihren Computer**

# Einen ganzen Monat lang können Sie unsere Anlage kostenlos testen!

Exklusiv von Compu Life:  
Schweizer Normtastatur!



Ganzes Bildschirm-  
Textsystem inkl.  
Wust bereits ab

**Fr. 12'330.-**

**Bearbeiten und verarbeiten Sie ab sofort Ihre Texte auf dem bequemen und bedienungsfreundlichen Ein-Platz-System von Compu Life.**

**Computer, Typ CBM 8032 - TT**  
Commodore-Computer mit exklusiv von Compu Life entwickelter Schweizer Normtastatur.

**Speichergerät (Doppel-Floppy)**  
Speicherkapazität min. 450 voll beschriebene A4-Seiten (ohne Diskettenwechsel).

**Typenrad-Schönschreibdrucker**  
Mit Schweizer Textverarbeitungs-Modul für alle deutschen, französischen, italienischen und englischen Zeichen.

**Software**  
Spezielles Software-Angebot für Industrie, Handel, Gewerbe (auch Kleingewerbe).

## System Einführung

Gründliche System-Einführung inkl. Betriebsanleitung. Falls Sie bereits ein CBM 8032-Modell besitzen, können Sie dieses ohne grossen Aufwand durch uns mit unserer speziellen Programmier- oder Schweizer Normtastatur und mit einem augenschonenden Blendschutz ausstatten lassen!

## Kommen Sie - vergleichen Sie!

Damit wir Ihnen in aller Ruhe zeigen können, was ein Commodore-Computer zu leisten vermag, ist eine telefonische Voranmeldung unerlässlich (Tel. 063 72 1113). Auf Wunsch senden wir Ihnen auch gerne detaillierte Informationen zu.

Konstruktionsänderungen vorbehalten.

Hard- + Software aus einer Hand

# COMPU LIFE

Rüfenacht AG, 4950 Huttwil, Telefon 063 72 1113

# Leserdienst-Kontaktkarte

Ich bitte Sie um weitere Informationen zu der in **Mikro+Kleincomputer Heft 82-4** 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 Heft 82-4** 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 Heft 82-4** 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.

Die **Leserdienst-Kontaktkarte** ist eine neue Dienstleistung von **MIKRO+KLEIN-COMPUTER** für seine Leser.

Die **Leserdienst-Kontaktkarte** erleichtert es Ihnen, direkt und ohne lange Umwege zusätzliche Informationen zu den in Anzeigen oder redaktionellen Besprechungen in den News...News... angebotenen Produkten und Dienstleistungen anzufordern.

Damit Ihre Anfrage bestmöglich beantwortet werden kann, kreuzen Sie bitte das zutreffende Kästchen (Informationswunsch, für welchen Einsatzbereich von Interesse, in welcher Branche und Funktion sind Sie tätig und wieviel Personen sind in Ihrer Firma beschäftigt) an. Sie helfen dadurch mit, dass die von Ihnen angefragte Firma Sie ohne unnötigen Ballast gezielt informieren kann.

Vergessen Sie nicht, die **Leserdienst-Kontaktkarte** mit der genauen Anschrift des Inserenten bzw. Anbieters und Ihre vollständige Adresse zu versehen, als Postkarte zu frankieren und natürlich abzusenden.



### Leserdienst-Kontaktkarte

Bitte genaue Anschrift des Inserenten bzw. Anbieters deutlich eintragen. Danke.



Name

Vorname

Firma/Institut

Strasse

PLZ/Ort

Telefon



### Leserdienst-Kontaktkarte

Bitte genaue Anschrift des Inserenten bzw. Anbieters deutlich eintragen. Danke.



Name

Vorname

Firma/Institut

Strasse

PLZ/Ort

Telefon



### Leserdienst-Kontaktkarte

Bitte genaue Anschrift des Inserenten bzw. Anbieters deutlich eintragen. Danke.



Name

Vorname

Firma/Institut

Strasse

PLZ/Ort

Telefon

bitte frankieren

### POSTKARTE

Firma

\_\_\_\_\_

\_\_\_\_\_

Strasse

\_\_\_\_\_

PLZ Ort

bitte frankieren

### POSTKARTE

Firma

\_\_\_\_\_

\_\_\_\_\_

Strasse

\_\_\_\_\_

PLZ Ort

bitte frankieren

### POSTKARTE

Firma

\_\_\_\_\_

\_\_\_\_\_

Strasse

\_\_\_\_\_

PLZ Ort

Die Leserdienst-Kontaktkarte ist eine neue Dienstleistung von MIKRO+KLEIN-COMPUTER für seine Leser.

Die Leserdienst-Kontaktkarte erleichtert es Ihnen, direkt und ohne lange Umwege zusätzliche Informationen zu den in Anzeigen oder redaktionellen Besprechungen in den News... News... angebotenen Produkten und Dienstleistungen anzufordern.

Damit Ihre Anfrage bestmöglich beantwortet werden kann, kreuzen Sie bitte das zutreffende Kästchen (Informationswunsch, für welchen Einsatzbereich von Interesse, in welcher Branche und Funktion sind Sie tätig und wieviel Personen sind in Ihrer Firma beschäftigt) an. Sie helfen dadurch mit, dass die von Ihnen angefragte Firma Sie ohne unnötigen Ballast gezielt informieren kann.

Vergessen Sie nicht, die Leserdienst-Kontaktkarte mit der genauen Anschrift des Inserenten bzw. Anbieters und Ihre vollständige Adresse zu versehen, als Postkarte zu frankieren und natürlich abzusenden.

# 82-4

August 1982  
Erscheint 6mal pro Jahr  
4. Jahrgang



Das schweizerische Fachmagazin für «Personal Computing» mit kompetenten Informationen über Mikroprozessoren und Kleincomputer, programmierbare Taschenrechner und Mikrocomputer für kommerzielle Anwendungen



ISSN 0251-0006

#### Verlag, Redaktion, Inserate

Mikro+Kleincomputer Informa Verlag AG  
Seeburgstrasse 12, 6006 Luzern

#### Postanschrift:

Postfach 1401, CH-6000 Luzern 15  
Telefon 041 - 31 18 46, Tx 72 227 (dclch)

#### Postcheck-Konten:

Luzern 60 - 27181  
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 (Ressort PPC/HHC), Leopold Asböck, Ernst Erb, Dr. Bruno Stanek

#### Manuskripte

Mit der Zustellung von Manuskripten anerkennt der Autor die Copyrightbestimmungen des Verlages. Mit der Annahme von Manuskripten durch die Redaktion und der Autor-Honorierung durch den Verlag hat dieser das Recht zur Veröffentlichung der entsprechenden Beiträge in anderen verlagseigenen Publikationen und zur Übersetzung in andere Sprachen erworben.

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 der Informationen erfolgt auf eigenes Risiko. Mit Verfassernamen gekennzeichnete Beiträge geben nicht unbedingt die Meinung der Redaktion wieder.

© 1982 by Informa Verlag AG, Luzern, aber Speicherung in Datenverarbeitungsanlagen für den eigenen Gebrauch erlaubt.

Nachdruck, auch auszugsweise, sowie Vervielfältigungen jedwelcher Art nur mit schriftlicher Genehmigung des Verlages und unter voller Quellenangabe.

**Bezug:** Jahresabonnement Fr. 36.-, Ausland (Europa) Fr. 44.- (inkl. Versand und Porto).  
**Einzelheftpreis** Fr. 6.50, Deutschland DM 8.-, Österreich 6S 50.

#### Auslandsvertretungen für Bezug:

##### Deutschland:

MSB-Verlag R. Nedela,  
Postfach 1420, 7778 Markdorf,  
Tel. 07544 3058, Tx 734628 msb-d

##### Österreich:

Target electronic, Abt. Zeitschriftenvertrieb,  
Maria Grüner Strasse 10, 6820 Frastanz,  
Tel. 05522 2 19 81, Tx 52300 tarel

Printed in Switzerland

## INHALT

	Der Kommentar	5
<b>KLEINCOMPUTER AKTUELL</b>	PC 1500 – Sharp's BASIC-Grafiker	7
	MP/M-Befehlsvorrat	15
	Computerneuheiten	19
	TI-88, der 15000 Byte-Texaner	21
	Speicheradressierung über 64 kBytes	23
<b>SMALL BUSINESS JOURNAL</b>	Textverarbeitung im arbeitspsychologischen Vergleich	27
<b>LEHRGÄNGE</b>	Programmieren mit Fortran VI (2)	33
<b>PPC/HHC</b>	Ausgefeilter Datentransfer	39
	Pythagoras auf Umwegen	45
	Programmgenerator und -generierung	48
	Befehlsstruktur im HP-41C/CV	53
	Barcodes für Peripheriebefehle	60
<b>HOBBY MIT MIKROS</b>	EPROMER-Programmiergerät für Parallelports	61
<b>GEWUSST WIE</b>	Benchmarktests	69
	Sortieren mit Microsoft-Basic	73
	Mühle im Endspiel (2)	77
	Sorcerer-Tips	83
	Grafik auf normalem Superbrain	85
<b>News...News...</b>	Aktuelle Meldungen aus der Welt der Mikros und Kleincomputer	89
<b>Börse</b>		96
<b>Vorschau</b>		98

Unser Titelbild, das uns freundlicherweise die Olivetti (Schweiz) AG zur Verfügung stellte, zeigt den neuen Personal Computer M20.

# Mit dem Home Computer von Texas Instruments können Sie sich in den fünf wichtigsten Sprachen unterhalten: ENGLISCH, ASSEMBLER, BASIC, PASCAL und TI-LOGO.



Wenn Sie den Home Computer von Texas Instruments mit anderen Modellen vergleichen, werden Sie schnell eine ganze Reihe von Vorteilen entdecken.

Fangen wir damit an, dass Sie mit ihm in den wichtigsten Programmiersprachen arbeiten können – was bei vergleichbaren Modellen durchaus nicht selbstverständlich ist.

Sie verfügen über eine freie Speicherkapazität von 16 K Byte (RAM), die auf 48 K erweiterbar ist. Darüber hinaus lässt sich die Gesamtkapazität bis auf 110 K Byte ausbauen.

Den TI 99/4A – angeschlossen an jeden handelsüblichen Fernseher – können Sie durch Zusatzgeräte zu einem kompletten Computer-System erweitern. Z. B. mit Ihrem Kassettenrecorder, mit Fernbedienung, Thermodrucker, RS 232-Schnittstelle für Datenübertragung mit Anschluss eines Matrixdruckers oder Plotters, mit bis zu drei Diskettenlaufwerken und einem Sprachsynthesizer.



Sie haben die Möglichkeit, vielseitig Programme zu gestalten durch eine hohe Bildauflösung mit 32 Zeichen auf 24 Zeilen und in 16 Farben, 256 x 192 Punkten, Sprache, ein Tonspektrum von 5 Oktaven und die Programmiersprachen BASIC, EXTENDED BASIC, TI-LOGO, UCSD-PASCAL und ASSEMBLER.

Um besondere Probleme zu lösen, stehen Ihnen weltweit über 600 Software-Programme zur Verfügung: als Solid State Software®-Module, auf Disketten und auf Kassetten.

Und wenn Sie zu guter Letzt seinen Preis mit dem seiner Konkurrenz vergleichen, dürfte es Ihnen nicht schwerfallen, sich für den TI 99/4A\* zu entscheiden.

Schliesslich ist es Ihr gutes Recht, von den Erfindern des integrierten Schaltkreises, des Mikroprozessors und des Mikrocomputers hohen technischen Standard zu einem vernünftigen Preis zu erwarten.

\*Erhältlich in allen Interdiscount-Filialen zu Fr. 998.– (Konsole).



Wir machen es Ihnen leichter.

TEXAS INSTRUMENTS

# Der Kommentar

VOM "HOBBYHEFT" ZUM ANERKANNTEN KLEINCOMPUTER-MAGAZIN

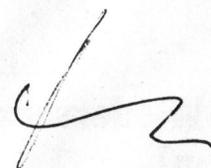
Ein herzliches Dankeschön den 921 Lesern, die sich die Zeit genommen haben, bis zum Einsendeschluss die 35 Fragen unserer ersten Leserbefragung zu beantworten. Mit höchstens 500 Teilnehmer haben wir ganz optimistisch gerechnet. Dass jetzt fast doppelt so viele Leser mitgemacht haben, hat uns mehr als überrascht. Dieses überwältigende Ergebnis zeigt eindeutig, was wir eigentlich schon lange wussten - Mikro+Kleincomputer-Leser gehören zu einer aktiven, interessierten und anspruchsvollen Leserschaft.

Um es vorneweg zu nehmen: Nach den ausgewerteten Fragebogen beziehen 80,5% unserer Leser Mikro+Kleincomputer im Abonnement, 11,5% kaufen es regelmässig am Kiosk und 8,0% kommen durch sonstige Bezugsquellen zum einzigen schweizerischen Kleincomputer-Magazin. Diese Prozentsätze widerspiegeln ziemlich genau die Verteilung der Gesamtauflage, sodass die ausgewiesenen Zahlen als repräsentativ angesehen werden können.

Danach wird Mikro+Kleincomputer von 40,2% der Leser schon seit der ersten Ausgabe anfangs 1979 gelesen. 80,8% lesen es zu Hause. Von 83,5% der Leser wird Mikro+Kleincomputer für so informativ gehalten, dass sie es länger als ein Jahr aufbewahren. 72,1% besitzen bereits ein Kleincomputersystem. Von diesen setzen 32,6% ihr System hauptsächlich für kommerzielle/administrative Zwecke ein und 58,7% benützen es für technisch/wissenschaftliche Anwendungen. 80,4% der Leser gefällt Mikro+Kleincomputer gut bis sehr gut und 61,6% finden das zweimonatliche Erscheinen gut. 81,7% lesen häufig bis immer die relativ wenigen Inserate aus denen 52,8% gelegentlich bis häufig Nutzen ziehen. Mehr als die Hälfte unserer Leser hat eine Hochschul- oder HTL-Ausbildung; 15,9% sind selbständig und 36,6% sind leitende Angestellte; 24,4% arbeiten in Kleinbetrieben mit 1-20 Beschäftigten und 28,8% sind in Unternehmen oder Institutionen mit über 500 Mitarbeitern beschäftigt. In der Elektronik-Branche sind 16,6% unserer Leser und im Sektor Handel und Dienstleistungen 18,5% zuhause. 37,5% der Mikro+Kleincomputer-Leser investieren für Kleincomputer inklusive Zubehör im Mittel pro Jahr Fr. 500-2000.--, 16,9% Fr. 1500-5000.-- und 8,0% über Fr. 10'000.--. Soviel zu den ersten Ergebnissen.

Ganz besonders gefreut haben wir uns über die zahlreichen positiven Antworten auf die Frage: "Was finden Sie an Mikro+Kleincomputer besonders gut?". Neben den vielfachen Nennungen "dass es eine schweizerische Zeitschrift ist" und "die sachliche Information" wollen wir eine Antwort herausgreifen, die stellvertretend für die mehrheitliche Zustimmung stehen soll: "Objektivität, keine "Prahler"-Artikel sowie der Versuch dem Leser ein Freund zu sein und ihn nicht einfach als zahlenden Kunden zu akzeptieren." Nur ein einziger Leser findet gar nichts gut an Mikro+Kleincomputer.

Ein grosses Dankeschön auch den vielen Lesern, die uns wertvolle Hinweise gegeben und ganz konkrete Verbesserungsvorschläge gemacht haben. Sie können versichert sein, dass wir diese Angaben sehr genau auswerten und unser Redaktionskonzept nach Möglichkeit danach ausrichten werden. Ueberrascht waren wir auch von der grossen Anzahl Leser, die sich mit sehr interessanten Themen als Autoren zur Verfügung stellen. Für Sie als Leser bedeutet dies, eine weitere Bereicherung der vielfältigen Kleincomputerinformation - vermittelt von engagierten Praktikern.



Hans-Jürgen Ottenbacher  
Verlagsleiter

# INTEGRAL DATA SYSTEMS

PRISM und PAPER TIGER Printer



- Erhältlich als Farb- oder Schwarzweiss-Drucker
- Automatische Justierung – proportionale Zwischenräume
- Hohe Auflösung als Plotter 34 x 34 Punkte/cm
- Plot 10 Graphik-Software kompatibel
- Serielle RS 232C und parallele Schnittstelle
- Korrespondenzqualität 24 x 9 Matrix
- Einzelblattzuführung
- Papierbreite bis 37 cm
- Hochstehende Qualität zu einem günstigen Preis

und vieles mehr...  
verlangen Sie unsere vollständige Dokumentation.

Verkauf und Service:

## captronix ag

20, rue de Lausanne  
1201 Genève  
Tél. 022-31 05 87



Olivengasse 11  
8032 Zürich  
Tel. 01-69 49 60

Verlangen Sie unsere Bedingungen  
für Wiederverkäufer.

## ncs sirius passt zu Ihnen. Genau nach Mass.

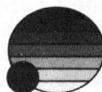


ncs sirius ist der Computer, der auf Ihren Schreibtisch passt. Er braucht nur wenig Raum. Den Bildschirm können Sie drehen, wenden und neigen. Die Tastatur ist frei beweglich. ncs sirius passt aber nicht nur äusserlich zu Ihnen. Es sind vor allem seine inneren Werte, die Sie schätzen werden. Beispielsweise sein riesiges Gedächtnis, das alle Ihre wichtigen Daten aufnehmen kann. Oder seine Sprache. Er spricht natürlich deutsch, wie Sie an seiner CH-Tastatur feststellen werden. Aber auch die von Ihnen gewünschte Computersprache – ob BASIC, COBOL, FORTRAN oder PASCAL. Und Ihre Arbeitssprache, ob Sie mit ihm nun buchhalten, konstruieren, Touren planen oder Briefe verfassen wollen.

ncs sirius ist ein Rechner der dritten Generation (16 bit). Das bedeutet technische Eigenschaften, die in dieser Grössenklasse bislang unvorstellbar waren. Zu einem äusserst niedrigen Preis. ncs sirius passt eben zu Ihnen. Genau nach Mass.

**ncs**

neue computer systeme ag  
Geissensteinring 26, 6005 Luzern



**sirius**  
COMPUTER

Generalvertretung  
Telefon 041 - 44 83 33

**Interessiert an Wiederverkauf?  
Rufen Sie uns an!**

Bitte senden Sie uns Gratisdokumentation

Name/Firma \_\_\_\_\_

Adresse \_\_\_\_\_

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

Tel. \_\_\_\_\_

# Kleincomputer aktuell



## PC 1500 - Sharp's BASIC-Grafiker

Leopold ASBÖCK

Als 1980 die japanische Firma SHARP den ersten Pocket Computer PC-1211 auf den Markt brachte, war dies ein Meilenstein in der Entwicklung BASIC-programmierbarer Kleincomputer. Das Nachfolgemodell, der SHARP PC-1500, überrascht durch Neuerungen, die gleichfalls einmalig sind - nicht zuletzt durch den kleinsten Vierfarbenplotter, dessen Konstruktion als Paradebeispiel für japanische Innovationsfreudigkeit angesehen werden kann.

Es ist nahezu unmöglich, den neuen Pocket Computer PC-1500 getrennt vom Drucker/Kassetteninterface CE-150 zu betrachten. Beide bilden funktionell und optisch eine Einheit, die trotz kleiner Dimensionen durch grosse Leistungen beeindruckt.

Der Drucker ist eigentlich ein Vierfarbenplotter, der als grafische Ausgabeinheit oder Protokollschreiber dient, Programme auslistet, Diagramme zeichnet oder in alle Richtungen plottet. Und dies in neun Schriftgrössen und in vier Farben. Bestechend einfach ist der mechanische Aufbau von Druckkopf-führung und Farbenwechsel.

Wesentlich leistungsfähiger als sein Vorgängermodell, bietet der PC-1500 dennoch alle Eigenschaften des PC-1211 (siehe Mikro+Kleincomputer 80-6). Aeusserlich sind die

Unterschiede gering: ein paar Tasten mehr, und diese bedienungsfreundlich versetzt, wie auf einer Schreibmaschine angeordnet. Innen aber steuert eine 8-Bit-CMOS-CPU das Geschehen und nicht mehr zwei 4-Bit-CPU's. Die BASIC-Programme laufen daher auch durchschnittlich zehnmal schneller ab als beim PC-1211, was sich angenehm bemerkbar macht.

Aber auch der Speicherbereich - selbstverständlich gleichfalls in CMOS ausgeführt - ist erweiterbar. Von den 3,5 kByte RAM stehen dem Benutzer 1850 Bytes zur Verfügung, das reicht durchaus für viele Anwendungen, bedenkt man, dass jeder BASIC-Befehl bei der Speicherung nur zwei Bytes belegt.

Kleine Erweiterungsmoduln mit 4 kByte oder 8 kByte werden in ein Fach auf der Rückseite des Compu-

ters eingesteckt und sorgen dafür, dass auch lange Programme ohne Speicherüberlauf ausgeführt werden können.

Der PC-1500 ist BASIC-programmierbar, sein Befehlssatz braucht den Vergleich mit dem grösserer Computer nicht zu scheuen. Dieser BASIC-Interpreter gibt dem kleinen Japaner eine grosse Ueberlegenheit gegenüber tastenprogrammierbaren Rechnern. Ohne Ueberheblichkeit darf man anmerken, dass man tastenprogrammierbare Spitzenmodelle vergisst, wenn man mit dem PC-1500 gearbeitet hat.

Selbst das Rechnen ohne Programm bietet viele Annehmlichkeiten. Die konsequente algebraische Arbeitsweise und die Eingabemöglichkeit von 80 Zeichen pro Zeile sowie die einfache Art der FormelAuswertung sind ein Komfort, der kaum zu über-



# Kleincomputer aktuell

bieten ist. Jede Ergebnisberechnung kann durch Druck auf eine Cursortaste rückgängig gemacht werden, so dass sich Neueingaben bei Eingabefehlern erübrigen. Durch die definierbaren Tasten können Funktionen mit einer Taste eingegeben werden, angezeigt werden Alphazeichen in einer optimal lesbaren Punktanzeige.

Die Stromversorgung erfolgt durch den Netzadapter EA-150, der zudem Ladegerät für die aufladbaren Batterien des Plotters ist. Da die Leistungsaufnahme des Computers mit 0,13 Watt recht gering ist, reichen vier handelsübliche 1,5-Volt-Batterien für eine lange netzunabhängige Arbeitszeit. Bei ausgeschaltetem Gerät bleiben Programme und Daten gespeichert, selbst die interne Uhr, die neben Stunden, Minuten und Sekunden auch Tag und Monat angibt, läuft weiter.

An peripheren Geräten sind eine RS232-Schnittstelle und ein Grafik-Tablett vorgesehen, eventuell gibt es auch eine Erweiterungsbox mit vier Steckplätzen, ein Videointerface und einen Sprachsynthesizer.

## DIE TASTATUR

Der PC-1500 besitzt eine übersichtlich gestaltete Tastatur in Schreibmaschinenform. Dazu kommen ein Ziffernblock sowie diverse Funktionstasten. Selbst mit grossen Fingern sind die Tasten fehlerfrei zu bedienen.

Die Buchstabentasten ermöglichen die Eingabe von Grossbuchstaben, über die SHIFT-Taste können aber auch Kleinbuchstaben eingegeben werden. Durch Drücken der SML-Taste (= small) wird diese Zuordnung umgekehrt, was die Eingabe von Text sehr erleichtert.

CURSOR- und SCROLL-Tasten vereinfachen das Editieren von BASIC-Programmen. Ueber die MODE-Taste werden drei Betriebsarten gewählt und in der Anzeige signalisiert:

- RUN-Mode zum manuellen Rechnen und Ausführen von Programmen
- PROG-Mode zur Eingabe, Ausgabe und Editieren von Programmen
- RESERVE-Mode zum Programmieren der obersten Tastenreihe in drei Ebenen.

Rechts oben befinden sich die ON- und OFF-Taste, der Computer stellt nach einigen Minuten Nichtbenützung selbsttätig ab, ohne dass Information verlorengeht. Beim neuerlichen Einschalten werden dieselben Zu-

stände aktiviert, die beim Abschalten definiert waren.

## DIE ANZEIGE

Die Flüssigkristallanzeige des PC-1500 ist eine durchgehende Matrix aus 156 x 7 Punkten, in der sich 26 Zeichen darstellen lassen. Der Eingabebuffer fasst 80 Zeichen, sodass durch Rollen über 200 Zeichen pro Zeile angezeigt werden können, da BASIC-Befehle nur zwei Bufferstellen belegen, bei der Ausgabe aber voll angezeigt werden.

Prozessor	8-bit-CMOS (SHARP)	4-bit-CMOS (2xSHARP)
ROM	16 KByte	11 KByte
RAM	1850 Bytes	1632 Bytes
Erweiterung	4 oder 8 KByte	---
Anzeige	26 Zeichen 156 x 7 Punkte	24 Zeichen 24 x 5 x 7 Punkte
Tasten	65	57
Alphazeichen	Gross- und Kleinbuchstaben	Grossbuchstaben
Defin. Tasten	18 + 3 x 6	18
Reserve Tasten	3 x 6	18
Zahleneingabe	dezimal u. hexadez.	dezimal
Characterfunkt.	ASC, VAL, LEN, CHR\$, STR\$, LEFT\$, RIGHT\$, MID\$	---
Länge der Strings	16 Zeichen	7 Zeichen
Characterarray	bis 80 Zeichen	---
Zeilennummer	1 bis 65279	1 bis 999
FOR-NEXT-Ebenen	max. 16	max. 4
GOSUB-Ebenen	max. 32	max. 4
DEBUG-Funktion	TRON / TROFF	DEBUG
WAIT-Funktion	16 msec - 17,48 min	---
INKEY\$-Funktion	ja	---
DISPLAY-Grafik	ja	---
Tongenerator	230 - 7000 Hz	BEEP
Geschwindigkeit	rund 10mal schneller	als PC-1211
Kassettenanschluss	2 Kassettengeräte	1 Kassettengerät
Drucker	4-Farben-PRINTER/ PLOTTER	Matrixdrucker
Dimensionen		
Computer	195 x 86 x 26 (mm)	175 x 72 x 16 (mm)
Drucker/K.interf.	330 x 115 x 50 (mm)	283 x 95 x 33 (mm)
Gewicht		
Computer	375 g	80 g
Drucker/K.interf.	900 g	440 g

Bild 1 Vergleich PC-1500 / PC-1211

# Kleincomputer aktuell

Der Cursor erscheint als waag-rechter Balken, wird aber in einen blinkenden Block verwandelt, sobald er in einem zeichenbelegten Feld erscheint.

INSERT- und DELETE-Tasten vereinfachen das Einfügen oder Löschen von Zeichen, zwei weitere Tasten sorgen für ein zeilenweises Auf- und Abscrollen des BASIC-Programmes.

Die vier erwähnten Tasten gehen in Dauerfunktion über, sobald sie längere Zeit gedrückt werden. Bei

Zeilenanfang oder -ende, Programm-anfang oder -ende bricht die Dauerfunktion selbsttätig ab.

Jeder Punkt der Anzeige ist mit wenigen BASIC-Befehlen einzeln ansprechbar, sodass beliebige Zeichen und Grafik erstellt werden können. Das Programmbeispiel "Zugfahrt" demonstriert mit nur wenigen BASIC-Zeilen sehr wirkungsvoll diese Grafikdarstellung: ein kleiner Zug fährt in der Anzeige mit entsprechender Geräuschkulisse hin und her und kuppelt seine Waggonen an und ab.

## PROGRAMMIERBARE TASTEN

Zehn Tasten der QWERTY-Reihe sind fest programmiert, durch Drücken der DEF-Taste und einer dieser Tasten wird in der Anzeige INPUT, PRINT, USING, ... LIST ausgegeben.

Die sechs Tasten unterhalb der Anzeige sind in drei Ebenen programmierbar, somit ergeben sich 18 frei programmierbare Tasten, die eine unschätzbare Hilfe darstellen, belegt man sie mit BASIC-Befehlen, Text oder algebraischen Ausdrücken.

Mit Hilfe der Doppelpfeiltaste werden die Ebenen I, II, III gewählt und im Display angezeigt. Mit SHIFT-MODE wird der RESERVE-Mode aufgerufen und jeder Taste kann eine beliebige Funktion zugeordnet werden (z.B.: PRINT USING, 2\*R\*PI, usw.).

Da man die Funktionen dieser 18 Tasten nicht im Kopf behalten kann, lässt sich zu jeder Taste ein Kurzkommentar speichern, der nach Betätigung der RCL-Taste im Display über der zugehörigen Taste deren Funktion angibt.

Alle übrigen Buchstabentasten sowie Gleichheitszeichen und Space können in Verbindung mit DEF als Programmaufrufe verwendet werden, falls die zugehörigen Zeichen als Kurzlabels in der entsprechenden Programmzeile stehen.

DEF Z beispielsweise startet jenes Programm, in dessen Programmzeile das Label "Z" steht. Zwei Tastaturschablonen liegen bei und erleichtern durch Beschriften das Auffinden der richtigen Tasten.

## DER VIERFARBENPLOTTER DAS KASSETTENINTERFACE

Diese unentbehrliche Einheit CE-150 wird inklusive Netzadapter EA-150, Bedienungsanleitung, Farbpatronen und Papierrollen in einem Etui geliefert, das auch den Computer samt Zubehör aufnehmen kann.

Instruktion		Funktionen	
AREAD	ARUN	ABS	ACS
BEEP	CLEAR	AND	ASC
CLS	CONT	ASN	ATN
CURSOR	DATA	CHR\$	COS
DEGREE	DIM	DEG	DMS
END	FOR..TO	EXP	INKEY\$
NEXT	GCURSOR	INT	LEFT\$
GOSUB	GOTO	LEN	LOG
GPRINT	IF..THEN	LN	MEM
INPUT	LET	MID\$	NOT
LIST	LOCK	OR	$\pi/PI$
NEXT	NEW	POINT	RIGHT\$
ON ERROR..GOTO	GOTO	RND	SGN
ON..GOSUB		SIN	$\sqrt{SQR}$
ON..GOTO	PAUSE	STATUS	STR\$
POINT	PRINT	TAN	TIME
RADIAN	RANDOM	^	VAL
READ	REM		
RESTORE	RETURN		
RUN	STEP		
STOP	THEN		
TRON	TROFF		
UNLOCK	USING		
WAIT			
Kassette		Drucker	
CHAIN	CLOAD	COLOR	CSIZE
CLOAD?	CSAVE	GLCURSOR	GRAPH
INPUT#	MERGE	LCURSOR	LF
PRINT#	RMTOFF	LINE	LLIST
RMTON		LPRINT	RLINE
		ROTATE	SORGN
		TAB	TEST
		TEXT	

Bild 2 BASIC-Befehle des Sharp PC-1500

# Kleincomputer aktuell

Aufladbare Batterien sorgen dafür, dass auch Arbeiten ohne Netzanschluss möglich ist. Eine Fehlermeldung in der Anzeige des Computers macht rechtzeitig darauf aufmerksam, dass die Batterien nachgeladen werden müssen.

```
Guten Tag!
```

```
Heute ist der  
22.Mai 1982  
Es ist zur Zeit  
16 Uhr 06 Min.
```

```
600: DIM Z$(4): T=10  
      000*TIME  
610: IF T<1E9 THEN  
      LET A$=" "+  
      STR$ T: GOTO 63  
      0  
620: A$=STR$ T  
630: FOR I=0 TO 4: Z$(  
      (1)=MID$(A$, 1  
      +2*I, 2): NEXT I  
640: TEXT : CSIZE 3:  
      COLOR 1: LPRINT  
      "Guten Tag!"  
650: CSIZE 2: COLOR  
      0: LPRINT "Heut  
      e ist der"  
660: COLOR 2: LPRINT  
      Z$(1); ". ";  
670: RESTORE 800:  
      FOR I=1 TO VAL  
      Z$(0): READ MO$  
      : NEXT I  
680: LPRINT MO$; " 1  
      982"  
690: COLOR 0: LPRINT  
      "Es ist zur Ze  
      it"  
700: COLOR 3: LPRINT  
      Z$(2); " Uhr ";  
      Z$(3); " Min.":  
      LF 2  
800: DATA "Jan.", "F  
      eb.", "Maerz", "  
      April", "Mai", "  
      Juni", "Juli", "  
      Aug."  
810: DATA "Sept.", "  
      Okt.", "Nov.", "  
      Dez."
```

Programmbeispiel 1

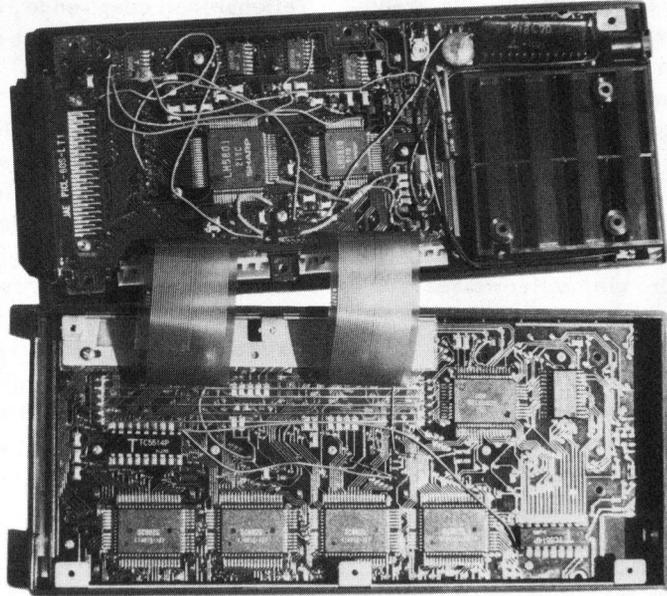


Bild 3 Das sauber gearbeitete Innenleben des Sharp PC-1500

Zwei Kassettenrekorder können zur Programm- und Datenspeicherung mit den mitgelieferten Kabeln angeschlossen werden. Das Speichern von Programmen und Daten, Verifizieren und Einlesen ist problemlos mit speziellen BASIC-Befehlen durchführbar.

REMOTE-Leitungen können über Software angesprochen werden und erübrigen ein häufiges manuelles Betätigen der Tasten beim Kassettengerät.

Der Vierfarbendrucker/-plotter besticht durch seine kleine Bauart und seine bis ins letzte Detail durchdachte einfache mechanische Konstruktion.

Auf einem nur 58 mm breiten Papierstreifen können in vier Farben (rot, grün, blau, schwarz) Text und Grafik erstellt werden. Vier Tintenpatronen in den genannten Farben werden einer Kunststoffhülse entnommen, die sie vor dem Austrocknen schützt. Durch das Betriebsprogramm gesteuert, nimmt sie der Plotterkopf auf. Auch die Entnahme der Tintenschreiber geschieht durch Hebeldruck einfach, schnell und sicher.

Der Plotter - der Arbeitsweise nach entspricht er einem Trommelplotter - bewegt den Schreibstift in x-Richtung und das Papier durch Vor- und Rückschieben in y-Richtung.

Er kann in zwei Betriebsarten - TEXT oder GRAPH - vom Programm oder durch Direkteingabe angesprochen werden.

Im TEXT-Mode werden ASCII-Zeichen in definierter Grösse und Farbe geschrieben. Die Schriftgrösse ist in neun Stufen wählbar, dadurch ergeben sich 36, 18, 12, 9, 7, 6, 5 oder 4 Zeichen pro Zeile. Am Zeilenende wird automatisch mit einer neuen Zeile begonnen.

Im GRAPH-Mode kann die Schreibrichtung nach rechts, links, oben oder unten programmiert werden. Zudem plottet der Tintenstift jede programmerzeugte Grafik in einem Raster mit 0,2 mm Auflösung. Der Koordinatenursprung lässt sich vorgeben und kann geändert werden.

Einfache Befehle sorgen für das Ziehen von Strecken - voll oder strichliert - oder das Zeichnen von Rechtecken.

# Kleincomputer aktuell

Durch einen Schalter lässt sich der Drucker auch auf Protokollausgabe stellen. Die in der Anzeige des Computers ausgegebenen Werte werden dann vom Drucker selbsttätig protokolliert.

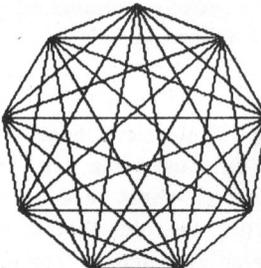
## DER BEFEHLSSATZ

Von den zahlreichen Befehlen, über die der SHARP PC-1500 verfügt, sollen einige näher erklärt werden, um die Leistungsfähigkeit dieses BASIC-Interpreters zu zeigen:

### TIME

ist eine Zeitvariable, in der ausser Stunden, Minuten und Sekunden auch Tag und Monat zur Verfügung stehen. Somit verfügt der PC-1500 neben sekundengenauer Zeit auch über das Datum. Terminplanung mit akustischem Alarm, Meldungsanzeige

Eckenzahl 9



```

10: DIM X(20), Y(20): TEXT
20: INPUT "Eckenzahl: "; N
30: IF N>20 GOTO 20
40: LPRINT TAB (4); "Eckenzahl "; N
50: LF 5: GRAPH
60: LINE (0, 0)-(110, 0), 9: SORGN
70: PHI=300/N: R=90
80: FOR I=0 TO N-1
90: X(I)=R*SIN (I*PHI)
100: Y(I)=R*ICOS (I*PHI)
110: NEXT I
120: FOR I=0 TO N-2
130: FOR J=I+1 TO N-1
140: LINE (X(I), Y(I))-(X(J), Y(J)), 0; 1
150: NEXT J: NEXT I
160: END
    
```

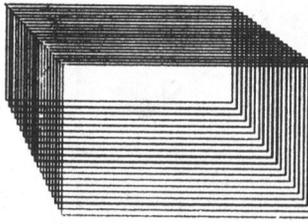
Graphikbeispiel 1  
in Originalgrösse

und Ausdruck sind leicht zu programmieren.

### BEEP

steuert einen akustischen Signalgeber, der mit BEEP ON/OFF ein- oder

ausgeschaltet werden kann. Tonzahl, -höhe und -dauer sind programmierbar, der Frequenzbereich umfasst 230 - 7000 Hz.



```

10: GRAPH
20: FOR I=1 TO 20
30: LINE (2*I, -2*I
  )-(150+3*I, -60
  -4*I), 0, RND 4-
  1, B
40: NEXT I
    
```

Graphikbeispiel 2

### ARUN

(automatic run) ist ein recht praktischer Befehl. Steht er zu Programmbeginn, so wird das Programm nach dem Einschalten des Computers automatisch gestartet.

### LOCK

blockiert die MODE-Taste, sodass ein Zerstören des Programmes durch Fehlbedienung verhindert wird. UNLOCK macht diese Verriegelung wieder rückgängig.

### TRON/TROFF

erlaubt das zeilenweise Abarbeiten des BASIC-Programms. Nach der Ausführung einer Zeile wird die Programmausführung unterbrochen, wodurch Programmfehler leichter eruiert werden können.

### WAIT

hält das Programm nach einer PRINT-Instruktion eine programmierbare Zeitspanne lang an. In Schritten von 1/64 Sekunden lassen sich Wartezeiten von 0 bis 1000 Sekunden erzielen.

### RANDOM

generiert eine Folge von Zufallszahlen. Die Funktion RND liefert eine Zufallszahl zwischen 0 und 1 oder eine positive ganze Zahl, je nach dem, ob der Argumentwert zwischen 0 und 1 liegt oder eine ganze Zahl ist.

Um eine der Zahlen 1, 2, 3, 4, 5, 6 zufällig zu wählen, reicht  
PRINT "Du hast eine "; RND 6;  
"gewürfelt!",

um eine der vier Plotterfarben "schwarz, blau, grün, rot" per Zufall zu wählen, genügt

COLOR RND 4 - 1

Speziell für die Anzeige, die aus 156 x 7 Punkten besteht, die einzeln ansprechbar sind, existieren recht wirkungsvolle Befehle:

### GCURSOR

bestimmt die Schreibposition auf der Anzeige, GPRINT gibt ein 7-bit-Muster, das dezimal, hexadezimal

```

5: REM Zugfahrt
6: REM -----
10: "Z": GRAPH :CLS
  :WAIT 0: BEEP
  ON
20: FOR I=0 TO 2
30: FOR J=110 TO 0
  STEP -1: GOSUB
  60: NEXT J
40: FOR J=0 TO 110:
  GOSUB 60: NEXT
  J: BEEP 1, 50, 12
  00
50: NEXT I: GOTO 20
60: BEEP 1, 180, 10:
  GCURSOR J:
  GPRINT "003838
  7E7E7838383F3F
  3979797F3F00";
70: FOR K=1 TO 3-I:
  GPRINT "103C7C
  7C3C3C7C7C3C00
  ";: NEXT K:
  RETURN
    
```

Programmbeispiel 2

# Kleincomputer aktuell

oder in Stringform definierbar ist, in der adressierten Punktspalte aus.

## POINT

liest den zu einer Punktspalte gehörenden 7-bit-Wert in eine Variable ein.

Mit Hilfe dieser drei Befehle lässt sich jedes Zeichen in der 156x7-Punktematrix darstellen oder bewegte Grafik erzeugen.

Die Kassettenbefehle steuern zwei Kassettengeräte (Kassette 0 und Kassette 1). Wird kein Index angegeben, so wird Kassettengerät 0 bedient, der Index -1 (z.B.: CLOAD-1, INPUT=-1, ...) aktiviert Kassettengerät 1.

Die beiden Geräte werden mit den mitgelieferten Kabeln am Drucker/Kassetteninterface angeschlossen. Es ist dann leicht möglich, Daten von einer Kassette einzulesen, auf den neuesten Stand zu bringen und auf dem zweiten Kassettengerät neu zu speichern, wodurch man gleichzeitig Vater- und Sohn-Datenbänder erhält.

Besondere Dienste leisten die Befehle RMTON und RMTOFF (remote on und remote off), die ein manuelles Starten und Stoppen der Kassetten erübrigen sowie die Befehle MERGE und CHAIN, die ein kettenförmiges Laden bzw. Ausführen mehrerer Programme gestatten.

## DRUCKERBEFEHLE

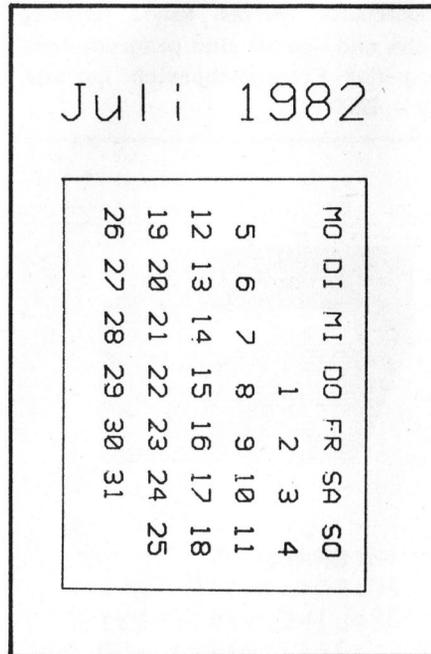
Die Druckerbefehle sind besonders effektiv - und effektiv - einzusetzen:

## COLOR

0, 1, 2, 3 lässt den Plotterkopf an den linken Rand fahren und auf die Farben schwarz, blau, grün oder rot wechseln.

## Csize

1, 2, 3 ... 9 wählt die Schriftgröße. Neun Möglichkeiten bestehen



Graphikbeispiel 3

in den Grenzen von 1,2 mm x 0,8 mm bis 10,8 mm x 7,2 mm. Bedingt durch diese Größen ergeben sich 36 bis 4 Schriftzeichen pro waagrechter Zeile.

## ROTATE

0, 1, 2, 3 bestimmt das Schreiben in vier Richtungen. Vor allem zum Beschriften von Koordinatenachsen, Figuren oder Diagrammen zieht man diesen Befehl heran.

## GLCURSOR

setzt den Plotterkopf auf eine vorgegebene Startposition, SORGN legt

diese Position als Koordinatenursprung fest. Obwohl die Streifenbreite nur 220 x-Werte zulässt, werden die x-Werte im Bereich von -2048 bis +2047 akzeptiert, eine automatische Randbegrenzung sorgt für das Abheben des Plotterstiftes bis sich die Koordinaten wieder innerhalb des Zeichenbereiches befinden.

## LINE

zeichnet eine Strecke zwischen zwei Punkten, die durch ihre Koordinatenwerte gegeben sind. Auch Farbe und Strichlängslänge - acht Möglichkeiten von 0,4 mm bis 1,8 mm - können durch Angabe je eines Argumentwertes gewählt werden.

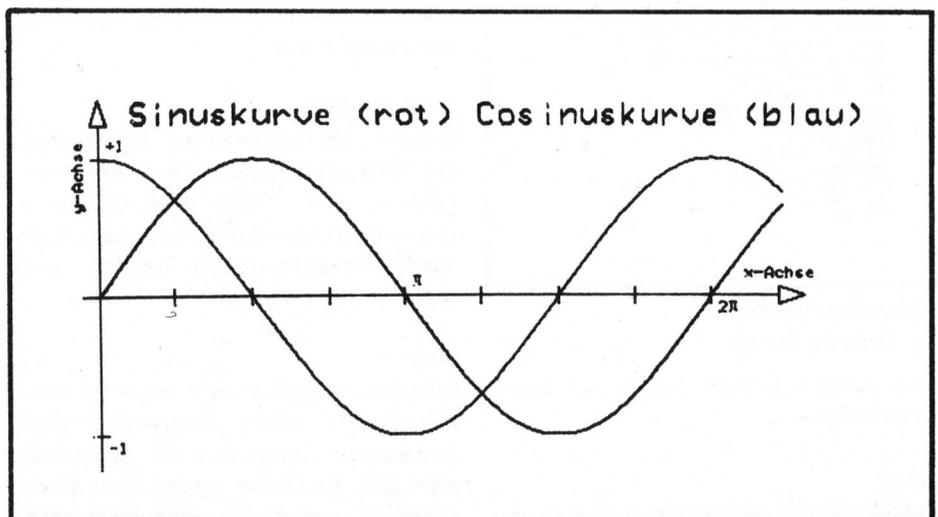
Stellt man noch ein "B" nach, so werden die beiden Punkte als Eckpunktkoordinaten eines Rechtecks aufgefasst - eine wesentliche Hilfe beim Zeichnen von Histogrammen.

## RLINE

bewirkt dasselbe wie LINE, allerdings mit relativen Koordinaten, die momentane Stiftposition wird jeweils als neuer Koordinatenursprung angesehen.

## LF

ergibt einen Zeilenvor- oder -rück Schub. Um ein Ausfädeln des Papiers zu verhindern, sorgt ein interner Zähler dafür, dass das Papier maximal um zehn Zentimeter zurückgeschoben wird.



Graphikbeispiel 4

# Kleincomputer aktuell

## LLIST

druckt eine beliebige Anzahl von Programmzeilen aus, Farbe und Schriftgrösse sind wählbar. Wird das Zeilenende erreicht, so werden eine Neuzeile begonnen und der Text eingerückt.

Rund 47 Fehlermeldungen erleichtern das Korrigieren von Programmen, innerhalb einer Zeile zeigt der Cursor die fehlerhafte Stelle an.

## DIE LITERATUR

Trotz Druck in Japan sind die Bedienungsanleitung und das Anwenderhandbuch in perfektem Deutsch abgefasst. Aber selbst die Beispiele des Anwenderhandbuches sind nicht bloss aus einem englischen Manual abgedruckt, sondern für deutschsprachige Anwender erstellt. Sogar die Literaturliste, die auf ergänzende BASIC-Literatur verweist, enthält deutschsprachige Bücher.

Zeichnungen geben Installationshinweise für Drucker und Kassetteninterface sowie für die Erweiterung mit den beiden Speichermoduln CE-151 und CE-155.

## ZUSAMMENFASSUNG

Mit dem PC-1500 ist der Firma SHARP ein grosser Wurf gelungen. Kombiniert mit dem Plotter/Kassetteninterface CE-150 ergibt sich eine Kleincomputerereinheit, die im Ingenieurbüro, im technisch-wissenschaftlichen wie auch im kaufmännischen Bereich viel Arbeitserleichterung bringen kann. Vor allem der Vierfarbenplotter stellt eine grosse Hilfe beim Zeichnen von Diagrammen, Funktionskurven oder Histogrammen dar.

Man darf den Pocket Computer PC-1500 durchaus als derzeitiges Spitzenprodukt japanischer Perfektionsarbeit ansehen, dem man grosse Investitionen an Ideenreichtum und Qualitätsdenken anmerkt.

## PC-1500 IM BENCHMARKTEST

Die Redaktion hat schon einmal darauf hingewiesen, dass es ihr Bestreben sein wird, auch für Hand Held Computer in den Testberichten Geschwindigkeitsprüfungen nach einem einheitlichen Programm zu veröffentlichen. Im Zusammenhang mit dem Artikel "Benchmarktests" in dieser Ausgabe standardisieren wir deshalb ab heute das Programm "MUKPRI" für BASIC-HHC's.

Es handelt sich dabei um das als "PRIME NUMBER CRUNCHER" (etwa: Primzahlenverschlinger) bekanntgewordene Benchmarkprogramm. Das Programm berechnet auf aufwendige Art und Weise die Primzahlen in der Menge der ersten 1000 natürlichen Zahlen. Es scheint uns tauglich für einen Vergleich der Geschwindigkeiten von FOR-NEXT-Schleifen, Grund- und Vergleichsoperationen, die zusammengefasst für HHC-Programme von hauptsächlichlicher Bedeutung sind. Mikro+Kleincomputer hat es in geänderter Fassung übernommen aus dem deutschen "Interface Age" 8/81.

Unser Benchmarkprogramm für HHC's lautet in der allgemeinen Form künftig wie folgt:

```
100 REM MUKPRI
110 REM
120 REM
130 PRINT "START"
140 FOR N = 2 TO 1000
150 FOR K = 2 TO 500
160 LET M=N/K
170 LET L=INT(M)
180 IF L=0 THEN 230
190 IF L=1 THEN 220
200 IF M>L THEN 220
210 IF M=L THEN 240
220 NEXT K
230 PRINT N;
240 NEXT N
250 PRINT "ENDE"; CHR$(7)
260 END
```

In der folgenden Anpassung auf den schnellen Sharp PC-1500 musste darauf geachtet werden, dass PRINT-Befehle normalerweise das Programm

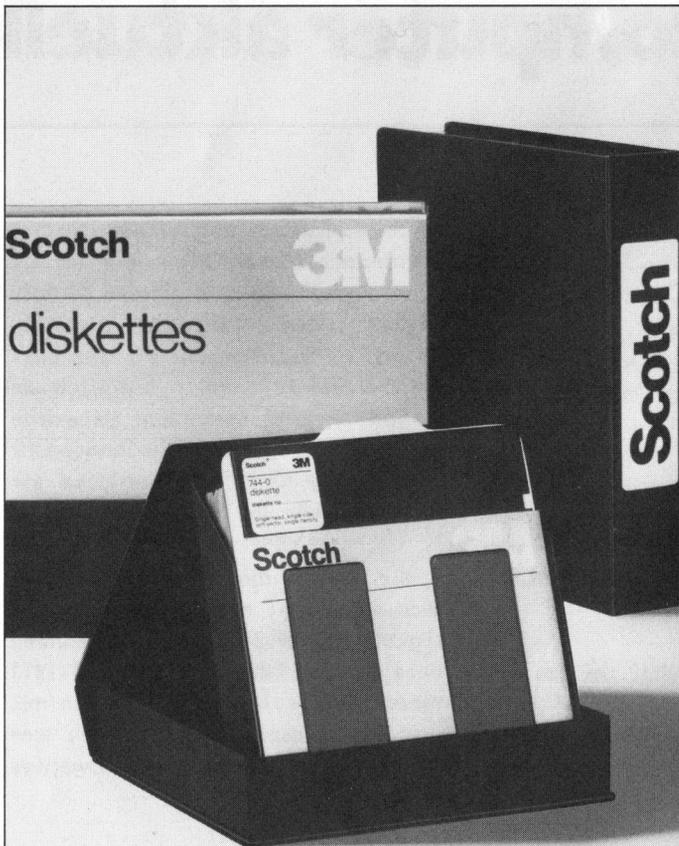
stoppen. Ein vorgeschaltetes WAIT 0 verhindert diesen Stop und ist im Programm für alle folgenden PRINTs wirksam. Der Befehl WAIT 0 verzögert das Programm für die 168 auszugebenden Primzahlen lediglich um insgesamt zwei Sekunden! Es wurde im Programm zugleich die innere Uhr verwendet, ohne ihre Funktion als Tageszeit-Uhr zu beeinträchtigen.

Wir lassen die Testresultate für sich sprechen, müssen der Gerechtigkeit halber aber noch erwähnen, dass die Zeit für den SHARP PC-1211 einen Ausdruck der Primzahlen mit dem Matrixdrucker beinhaltet, was das Resultat indessen nur um wenige Minuten verändert.

RECHNER	MUKPRI-Zeit
CBM 4032	19 min
SHARP PC-1500	1h 33 min
CASIO FX-702P	4h 50 min
SHARP PC-1211	15h 05 min

Das für den Test beim PC-1500 abgeänderte Programm lautet:

```
100:REM MUKPRI
110:REM AUF SHARP
120:REM PC-1500
130:PAUSE "START"
135:A=TIME
140:FOR N=2TO 1000
150:FOR K=2TO 500
160:M=N/K
170:L=INT(M)
180:IF L=0THEN 230
190:IF L=1THEN 220
200:IF M>LTHEN 220
210:IF M=LTHEN 240
220:NEXT K
230:WAIT 0:PRINT N
;
240:NEXT N
245:B=TIME
250:BEEP 5, 1, 10000
260:CORSOR 0:PRINT
"ENDE";
270:C=DMS (DEG B-
DEG A)
280:USING "##.###"
"
290:WAIT :PRINT "
/ ZEIT: ";C
300:USING
310:END
```



# Ihr Computer ist nur so zuverlässig wie Ihre Datenträger

Auf Scotch Disketten können Sie sich verlassen. Jede ist 100%-ig fehlerfrei. Bei neutralen Laboruntersuchungen erwies sich die Scotch Diskette als diejenige mit den optimalsten und ausgeglichensten Werten. Sie wurde als Folge zur internationalen Referenz-Diskette für die Industrie (SRM 5654) gewählt.

**Scotch Datensicherheit, damit jedes Bit erhalten bleibt.**

## COUPON

An untenstehende Adresse einsenden

Ja, ich möchte mehr über Scotch Disketten wissen; senden Sie mir Unterlagen.

Firma \_\_\_\_\_  
 Name \_\_\_\_\_  
 Strasse \_\_\_\_\_  
 Ort \_\_\_\_\_

3M (Schweiz) AG  
 Abt. DRP  
 Räfelstrasse 25, 8021 Zürich  
 Telefon 01 35 50 50



# NUMBER ONE



PIM-Systems hat sie entwickelt, die absolut konkurrenzlose

## TEXTVERARBEITUNG

- mit:
- CH-Normtastatur (äöüèääöç!) oder PIM-Tastatur (äöü)
  - Adressverwaltung (4000/8000 Adressen pro Diskette)
  - Serienbriefautomatik (Direktmailing mit Selektion)
  - TELEX-Bearbeitung mit PTT-Lochstreifenoutput
  - professionelle Offert/Devis-Verarbeitung mit Preis- und Variablenverwaltung
  - automatische Mutation der Preistabellen
  - Formularautomatik

\* keine benutzerunfreundliche, herkömmliche Textverarbeitung

**ab Fr. 1300.-**

Lernen Sie die PIM-Verarbeitungsprogramme, mit denen Sie sofort arbeiten können, kennen bei:

## PIM-SYSTEMS

Büro-Mikrocomputer & Software Lochstrasse 18  
 8200 Schaffhausen Tel. 053/4 54 50

Büro FÜRER AG, 8021 Zürich, Hardturmstr. 76, Tel. 01/42 05 05  
 Elbatex AG, 5430 Wettingen, Alb. Zwysigstr. 28, Tel. 056/ 27 01 27  
 Planal AG, 8800 Thalwil, Seestrasse 185, Tel. 01/720 35 20  
 Novak AG, 8400 Winterthur, Technikumstr. 46, Tel. 052/22 08 03  
 Sele Data, 9496 Balzers, Alte Churerstr. 710, Tel. 075/4 18 45  
 Hamatro Computer, 6442 Gersau, Sagenbachstr. 4, Tel. 041/ 84 16 39

## INTERFACES

zur Benutzung von elektronischen OLIVETTI-  
**Schreibmaschinen** als Typenrad-Drucker

Standardschnittstellen:

RS 232 (V24) / Parallel (Centronic) / IEEE für die OLIVETTI-Typen:

ET 121 / 201-221-231, Praxis 35

ab Lager lieferbar.

Alle OLIVETTI-Schreibmaschinen mit eingebautem INTERFACE zur Verwendung als DRUCKER ab Lager Zürich lieferbar.

Verlangen Sie Unterlagen!

## DERUNGS AG

**Dübendorfstrasse 335, 8051 Zürich, Telefon 01 40 33 89**

## apple II™ Europlus 48K ab Fr. 95.- pro Monat!!

Weitere Marken und ganze Systeme auf Anfrage.

Ich möchte gerne eine unverbindliche Offerte für: ➤

Name \_\_\_\_\_

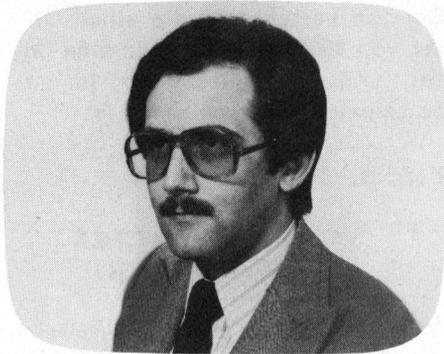
Straße \_\_\_\_\_

Plz/Ort \_\_\_\_\_ Tel: \_\_\_\_\_

einsenden an: T P C  
 Buttenustrasse 15

**TPC** 8134 Adliswil

# Kleincomputer aktuell



## MP/M-Befehlsvorrat

Roland PETER

MP/M als Abkürzung für Multiprogramming-Monitor, ist ein einfaches Time-Sharing-System und zu CP/M, dem De-facto-Standard-Betriebssystem für professionelle Mikrocomputer, voll kompatibel. In diesem Artikel wird nun der Befehlsvorrat des Betriebssystems MP/M beleuchtet. Die unter CP/M 2.2 verfügbaren Befehle werden nicht beschrieben.

Das MP/M-Betriebssystem benötigt einen frei verfügbaren Arbeitsspeicher von mindestens 32 KBytes. Unter dem MP/M-Betriebssystem können bis zu 16 Bildschirme und bis zu 16 Plattenlaufwerke mit einer Kapazität von je 8 MBytes verwaltet werden. MP/M kann bis zu acht Prozesse gleichzeitig abwickeln. Jedem Prozess ist ein fester Speicherbereich von 48 KBytes zugeordnet.

Die Funktionen die nur im Mehrbenutzersystem benötigt werden sind:

CONSOLE  
DSKRESET  
GENMOD  
SPOOL, STOPSPLR  
SCHED  
TOD  
MPMSTAT

Für MP/M-Operationen kommen noch weitere Funktionen hinzu:

ABORT  
ATTACH, CTRL D  
ERAQ  
GENHEX  
GENSYS  
MPMLDR  
PRLCOM  
USER

Beim ersten Start von MP/M erscheint folgende Meldung:

MP/M .....  
XA.

X ist die jeweilige Benutzernummer  
X = 0 für Bildschirm 0 (bei Start)  
A = Selektiertes Laufwerk (Umschalten wie unter CP/M)

### USER

Mit dem USER-Befehl ist es möglich, jedem Bildschirm eine Nummer zwischen 0 und 15 zuzuordnen. Ist ein Bildschirm auf einen Benutzerbereich geschaltet, so werden alle generierten Dateien in diesen Bereich gespeichert. Mit dem PIP-Befehl (Parameter G), können Dateien aus anderen Benutzerbereichen kopiert werden.

Format: USER  
0A.USER 1 (CR)  
1A

In einem Benutzerbereich wird so lange gearbeitet, bis ein neuer USER-Befehl eingegeben oder das System neu gestartet wird.

### DSKRESET

Mit DSKRESET wird den anderen Benutzern ein beabsichtigter Diskettentausch mitgeteilt.

Format: DSKRESET  
0A.DSKRESET (CR)  
CONFIRM RESET DISK  
SYSTEM (Y/N)?

Diese Meldung erscheint auf jedem am System angeschlossenen Bildschirm. Der Wechsel wird nur freigegeben, wenn jeder Benutzer mit Y (Yes) antwortet.

### SPOOL

Mit dem SPOOL-Befehl werden die ASCII-Dateien in eine Druckerwarteschlange gesetzt. Mit der Spooler-Routine werden sie verwaltet und gesteuert. Der SPOOL-Befehl findet dort Anwendung, wo mehrere Benutzer Daten zum Drucker senden wollen, oder ein Anwender mehrere Dateien ausdrucken will.

Format: SPOOL  
0A.SPOOL TITBL.PRN  
MENU.BAS (CR)

Die Textdateien TITBL.PRN und MENU.BAS werden hintereinander zur LST Einheit (Drucker) gesendet.

Die SPOOL Operation läuft ohne Unterbrechung solange ab bis der Puffer leer oder bis STOPSPLR eingegeben wird.

Format: STOPSPLR  
0A.STOPSPLR (CR)

SCHED

Für das Aufstarten von Programmen zu einer vorgegebenen Zeit wird der SCHED-Befehl eingesetzt.

Format: SCHED  
0A.SCHED 03/05/82 09:33  
MOVIE (CR)

Eingabe  
Datum Monat/Tag/Jahr  
Uhrzeit Stunde/Minute  
Programm Dateityp COM oder PRL

Das Programm wird gestartet unter der Voraussetzung, dass das System zu diesem Zeitpunkt in Betrieb ist. Diese Eingabe sollte nur in Übereinstimmung mit allen Benutzern

# Kleincomputer aktuell

ausgeführt werden, da von jedem Benutzer zu jedem Zeitpunkt, Zeit und Datum neu gesetzt werden können.

## TOD

Für das Abfragen von Datum und Zeit wird der TOD-Befehl verwendet (Time of Day).

Format: TOD (Abfrage)

```
0A.TOD (CR)
SAT 02/27/82 09:33:00
```

Auch das Setzen von Datum und Zeit kann mittels dem TOD-Befehl erreicht werden.

Format: TOD (Eingabe)

```
0A.TOD 03/06/82 09:09:00
(CR)
STRIKE ANY KEY TO SET
TIME (X)
SAT 03/06/82 09:09:00
0A.
```

Nach Eingabe der Zeit erscheint die Meldung "STRIKE ....". Durch Drücken einer beliebigen Taste wird die Zeit gesetzt und nochmals angezeigt.

## ABORT

Durch CTRL C kann jedes ablaufende Programm abgebrochen werden. Für Programme die über andere Bildschirm-einheiten ablaufen wird der ABORT-Befehl verwendet.

Format: ABORT

```
3C.ABORT SCK 2 (CR)
```

Von Bildschirm 3 wird das über Bildschirm 2 laufende Programm SCK abgebrochen. Im Hintergrund ablaufende Programme sind durch diese Massnahme nicht betroffen.

## CTRL D

Durch Eingabe von CTRL D wird ein laufendes Programm in den Hin-

tergrund geschaltet. Dadurch wird der Bildschirm wieder frei für andere Befehlseingaben. Ein Programm, das im Hintergrund abläuft, schaltet sich selbst in den Vordergrund, wenn durch das Programm eine Eingabe verlangt wird und der Bildschirm frei ist.

## ATTACH

Ein im Hintergrund ablaufendes Programm kann durch den ATTACH-Befehl wieder in den Vordergrund geschaltet werden. ATTACH und CTRL D müssen aber vom gleichen Bildschirm aufgerufen werden.

Format: ATTACH

```
0A.ATTACH GAUL (CR)
```

## CONSOLE

Da die USER-Nummer auf jedem Bildschirm frei wählbar ist, kann zur Kontrolle mit dem CONSOLE-Befehl die vom System zugeteilte Bildschirm-Nummer erfragt werden.

Format: CONSOLE

```
3B.CONSOLE (CR)
CONSOLE = 2
```

Dem Benutzer 3 ist der Bildschirm 2 zugeordnet.

## ERAQ

Im Gegensatz zum ERA-Befehl unter CP/M, wird durch ERAQ jede Datei angezeigt und das Löschen erfolgt nur durch Quittieren mit Y.

Format: ERAQ

```
1A.ERAQ LMA.* (CR)
LMA.COM ? Y
LMA.PRN ? Y
```

## GENMOD

Der GENMOD-Befehl transformiert eine Datei XZY, die aus zwei HEX-

Dateien besteht und die einen Offset von 100H ausweisen, in eine Datei XXZ, die seitenweise im Arbeitsspeicher verschiebbar ist.

Format: GENMOD

```
0A.GENMOD XZY.HEX
XXZ.PRL $ssss (CR)
```

\$ssss = Benötigter Speicherbereich des Programms (Hexadezimal) (Wahlweise Eingabe)

## GENHEX

Der GENHEX-Befehl transformiert eine COM-Datei in eine HEX-Datei.

Format: GENHEX

```
0A.GENHEX LUG.COM 100 (CR)
```

100 = Offset (Wahlweise Eingabe)

Wir lösen und programmieren Ihre

### techn.-wissenschaftliche Probleme

Im weiteren stehen für CP/M-Systeme folgende Programme bereit:

- Matrizen, inverse Matrizen, Eigenwerte
- Statik, bestimmte/unbestimmte Fachwerke, Rahmen, Durchlaufträger
- Mech. Schwingungen, mehrere Freiheitsgrade, Systeme, gekoppelt, gedämpft, instationär
- Wärmeleitung, stationär/instationär.

## REUSSER

Ingenieure und Betriebsberater

Peter Reusser AG

Dipl. Ing. ETH

CH-8706 Meilen Gruebstr. 25 Tel. 01-923 29 33

### Fertige Lösungen für Ihren Klein- oder Mittelbetrieb

- Finanzbuchhaltung
- Fakturierung/Debitoren
- Liegenschaftsverw.
- Adressverwaltung
- Textverarbeitung
- Lagerverwaltung

## LOGON AG

Baslerstrasse 145  
8048 Zürich

Telefon 01 62 59 22

Konsumstrasse 1  
8630 Rüti/ZH

Telefon 055 31 72 30

### µP-STANDARDLITERATUR

- Mikrocomputer-Grundwissen
- Einführung in die Mikrocomputer-Technik
- 77 BASIC-Programme
- Programmieren in Assembler:  
Systeme 6502, 6800 und 8080A/8085
- Die 16-Bit-Generation
- CP/M und WordStar
- CBM- und Apple II-Handbücher

te-wi te-wi Verlag GmbH  
Theo-Prosel-Weg 1  
8000 München 40

# Kleincomputer aktuell

## PRLCOM

Der PRLCOM-Befehl transformiert eine PRL-Datei in eine COM-Datei. Programme die im PRL-Format abgespeichert sind, können durch das MP/M-Betriebssystem ausgeführt werden, währenddem für CP/M ein Programm im COM-Format abgespeichert sein muss.

Format: PRLCOM

```
0A.PRLCOM HAH1.PRL
HAHA2.COM (CR)
```

## GENSYS

Für die Generierung eines MP/M-Systems wird dieser Befehl angewendet. Alle notwendigen Systeminformationen werden im Dialog erfragt und danach wird eine MPM.SYS-Datei generiert.

Format: GENSYS

```
0A.GENSYS (CR)
MP/M 1.0 SYSTEM GENERATION
TOP PAGE OF MEMORY= C0 (CR)
NUMBER OF CONSOLES= 2 (CR)
BREAKPOINT RST & = 5 (CR)
ALLOCATE USER STACKS FOR
SYSTEM CALLS (Y/N) Y (CR)
MEMORY SEGMENT BASES, (FF
TERMINATES LIST)
```

```
:00 (CR)
:40 (CR)
:60 (CR)
:FF (CR)
SELECT RESIDENT SYSTEM
PROCESS: (Y/N)
TIME ? Y (CR)
SCHED ? Y (CR)
ATTACH ? Y (CR)
SPOOL ? Y (CR)
```

## MPMLDR

Durch Eingabe des MPMLDR-Befehls wird die MPM.SYS-Datei geladen und gestartet. Die wesentlichen Systemparameter werden angezeigt.

Format: MPMLDR

```
0A.MPMLDR (CR)
MP/M 1.0 LOADER
NUMBER OF CONSOLES=2
BREAKPOINT RST & =5
TOP OF MEMORY =C0FFH
MEMORY SEGMENT TABLE:
SYSTEM DAT C000H 0100
.
.
.
```

## MPMSTAT

Durch die Eingabe des MPMSTAT-Befehls werden folgende Meldungen ausgegeben:

- Prozesse die auf CPU-Zeit warten
- Prozesse die auf Meldungen von Warteschlangen (Queues) warten
- Prozesse die auf die Möglichkeit des Sendens einer Meldung warten
- Prozesse verzögert (delayed)
- Prozesse geschleift (polling)

## POLLING (ABRUFBETRIEB)

Der Abrufbetrieb kennzeichnet eine bestimmte Betriebsweise der Datenübertragung bei mehreren Datenstationen. Dabei werden die zu übertragenden Daten sendefertig in der Endstation bereitgestellt oder gespeichert. Auf ein bestimmtes Signal (fest vereinbartes Zeichen), das der Empfänger aussendet, beginnt die Endstation zu senden, oder sie antwortet, dass keine Daten zu senden sind. Beim Polling sind die Datenstationen über eine gemeinsame Leitung miteinander verbunden.

- Flags
- in Arbeit befindliche Queues
- auf den Bildschirm wartende Prozesse
- Prozesse die über den Bildschirm ablaufen
- Speicherzuordnung des ganzen Systems

Format: MPMSTAT

```
0A.MPMSTAT (CR)
```

MP/M STATUS DISPLAY

```
TOP OF MEMORY=FFFFH
NUMBER OF CONSOLES=02
DEBUGGER BREAKPOINT RESTART &
= 06
STACK IS SWAPPED ON BDOS
CALLS
Z80 COMPLEMENTARY REGISTERS
MANAGED BY DISPATCHER
READY PROCESS (ES):
```

```
.
.
PROCESS (ES) DQING:
.
.
PROCESS (ES) NQING:
.
.
DELAYED PROCESS (ES):
.
POLLING PROCESS (ES):
.
PROCESS (ES) FLAG WAITING:
.
.
FLAG (S) SET:
.
QUEU (S):
.
.
PROCESS (ES) ATTACHED TO
CONSOLES:
(0) .
(1) .
PROCESS (ES) WAITING FOR
CONSOLES:
(0) .
(1) .
MEMORY ALLOCATION:
BASE=0000H SIZE=4000H
ALLOCATED TO . (1)
BASE=4000H SIZE=2000H FREE
BASE=6000H SIZE=1100H
ALLOCATED TO . (0)
```

**59,7%**  
der Leser können sich an bestimmte  
Inserate in Mikro+Kleincomputer  
erinnern.

**Ist Ihr Inserat dabei?**

(Quelle: Leserbefragung 1982)

# Einen Computer für die Buchhaltung, für das Lager, zum Textverarbeiten, zum Planen und Forschen, zum Studieren, zum Spielen?

## Also einen Commodore.



### Commodore Volkcomputer VC20

Preis: Fr. 795.-  
 Speicher: 5k Bytes RAM, Erweiterung bis 32k RAM möglich  
 Bildschirm: 22 Zeichen, 23 Linien  
 Farben: 8farbig  
 Ton: 3 Ton-Generatoren, 3 Oktaven  
 Sprachen: Basic V 2.0, Maschinsprache 6502  
 Anschluss: an jeden Fernseher anschliessbar



### Commodore CBM 4032

Preis: Fr. 2'975.-  
 Speicher: 32k Bytes RAM  
 Bildschirm: 40 Zeichen, 25 Linien  
 Sprachen: Basic 4, Maschinsprache 6502



### Commodore CBM 8032

Preis: Fr. 3'475.-  
 Speicher: 32k Bytes RAM  
 Bildschirm: 80 Zeichen, 25 Linien  
 Sprachen: Basic 4, Maschinsprache 6502



### Commodore MMF

Preis: Fr. 4'975.-  
 Speicher: 96k Bytes RAM, 48k Bytes ROM  
 Bildschirm: 80 Zeichen, 25 Linien  
 Sprachen: Basic, Fortran, Cobol, Pascal, APL, Assembler 6502/6809

Commodore Computer gibt es für alles und jedermann. Und weil wir unseren Kunden nicht nur eine breite Hardware-Palette, sondern auch ein umfassendes Software-Programm bieten, sind weltweit schon über eine Viertelmillion Commodore im Einsatz.

Unsere Computer werden aus gutem Grund nur über erfahrene Wiederverkäufer vertrieben: sie prüfen Ihr EDV-Problem und erarbeiten auf Wunsch gleich eine Gesamtlösung.

Eines ist sicher: wir haben auch für Sie eine effiziente und kostengünstige\* Lösung. Senden Sie uns also heute noch den Coupon, damit wir Sie eingehend ins Bild setzen können.

\* weitere Geräte aus unserem Hardware-Angebot: Floppy Disk (2 x 176k Bytes) Fr. 3'175.-, Drucker ab Fr. 1'750.-.

### Informations-Gutschein:

- Ja, senden Sie mir bitte Unterlagen über Commodore.  
 Ich möchte Commodore Computer persönlich kennenlernen.

Absender: \_\_\_\_\_

Bitte an Commodore AG, Aeschenvorstadt 57, 4010 Basel, senden.

**commodore**  
 COMPUTER

Commodore AG, Aeschenvorstadt 57, 4010 Basel,  
 Tel. 061 23 78 00, Tlx 64961 cbm ch

# Kleincomputer aktuell

## Computerneuheiten

### 16 BIT-COMPUTER VON CORVUS

(101/eh) Corvus, bekannt als Hersteller von Harddisksystemen, entwickelte einen 16-bit Kleincomputer, der unter dem Namen Corvus Concept lanciert werden wird. Für die 4995 Dollars, die die Maschine auf dem amerikanischen Markt kosten soll, bietet sie ausserordentlich viel.



Das auffälligste Merkmal ist ein Bildschirm der 66 Zeilen mit je 120 Zeichen darstellen kann, ein Format, welches jeden, der den Kleincomputer für Textverarbeitungsaufgaben einsetzt, begeistern wird. Für Grafik-Anwendungen lassen sich 560\*720 einzelne Bildpunkte ansteuern. Das Format entspricht in etwa auch dem des Xerox Star und der Video Textsysteme.

Der 68'000 eine 16-bit CPU wird als Hauptprozessor eingesetzt; er kann bei der normalen Ausführung über 256 kByte RAM verfügen. Dieser grosse Speicherbereich kann im Gerät intern bis auf 512 kByte erweitert werden.

Eine originelle und praktische Lösung stellt der verstellbare Bildschirm dar, der sich um 90 Grad kippen lässt, so dass man eine

querformatige Darstellung erhält. Mit einem Knopfdruck lässt sich die Zeichenausgabe vom vertikalen auf das horizontale Bildformat umschalten. Dieses horizontale Bildformat ist praktisch für das Verarbeiten grosser Tabellen, die man jetzt auf einmal überblicken kann, während das VertikalfORMAT für die Textverarbeitung (Ganzseitenformat) ideal ist.

Zur Betriebssoftware wird noch ein CP/M-Emulator mitgeliefert, der das Laufenlassen von CP/M-Software, welche für 8-bit Maschinen erstellt wurde, auf diesem 16-bit Computer ermöglicht.

### APPLE II MIT 16-BIT

(102/eh) Coprocessors Inc., San Jose, CA stellte auf der West Coast Computer Faire in San Francisco eine Zusatzkarte für den Apple II vor, die aus dem guten alten Apple im Handumdrehen einen echten 16-bit Computer macht. Nebst dem Prozessor vom Typ 8088 enthält die Karte auch noch 64 kByte RAM, die zu dem bereits im Apple vorhandenen Speicher hinzugefügt werden. Als Betriebssystem wird PC-DOS verwendet, welches auch auf dem IBM-Kleincomputer läuft. All das erhalten Sie für 899 Dollar.

Ein ähnliches Board, jedoch ohne zusätzliche Speicher bieten Systems Research für 550 Dollar an. Zu dieser Erweiterung wird das CP/M 86 Betriebssystem geliefert.

### KRAFTWERK VON PANASONIC

(101/pf) Schon seit einem knappen Jahr wird er als kommender Geheimtip herumgeboten und ist jetzt in ersten Exemplaren auf dem Markt: Der sensationelle HHC von Matsushita, genant Panasonic- oder Quasar-HHC.

Rein äusserlich handelt es sich dabei tatsächlich eher um einen HHC als um einen Taschencomputer, er hat nämlich etwa die gleich klotzigen Ausmasse wie der Sharp PC-1500 und eine Masse von 570 Gramm. Sein Herz ist ein uP 6502 mit 1,3 MHz Taktfrequenz und seine Stärke die Geschwindigkeit: Er soll in gewissen Applikationen schneller sein als ein Kleincomputer und kündigt damit den Trend an, dass bezüglich Verarbeitungsgeschwindigkeit die Kleinen nun die Lücken zu schliessen beginnen.

Allem Anschein nach ist sein Speicher flüchtig und nicht für das eigene Programmieren ausgelegt, obwohl ihm ein kraftvolles BASIC implementiert ist. Der Panasonic- oder Quasar-HHC wird vor allem für die Verarbeitung in ROM gegossener Programmpakete eingesetzt werden können. Solche Programmmoduln sollen ihm eine ROM-Kapazität von 64 kByte gewähren.

Die Hersteller sehen den HHC vorwiegend als mobile Datenverarbeitungsanlage in einem Kofferchen. Ueber eine E/A-Buchse werden dem HHC die folgenden, ebenfalls bereits erhältlichen und im Kofferchen unterzubringenden Geräte angeschlossen werden können: Ein Video-Interface mit Farbkapazität, ein RS-232-Interface, ein 15-Spalten-Drucker, ein Kassetteninterface und ein akustisches Modem zur Fernkopplung mit einer Grossanlage.

### KONKURRENZ FUER OSBORNE?

(103/eh) Was die Fachwelt schon lange erwartete und ersehnte, jedoch erstaunlicherweise bis heute noch nicht auf dem Markt angeboten wurde, war eine echte Konkurrenz zum tragbaren Osborne I Kleincomputer. Dies soll jetzt dank NLS (Non-Linear Systems) anders werden. Vielen Elektronikern unter Ihnen wird

# Kleincomputer aktuell

NLS noch wegen ihren kleinen Digitalvoltmetern der ersten Stunde ein Begriff sein. Dieselbe alteingesessene amerikanische Firma will jetzt einen tragbaren Kleincomputer mit Z80 Prozessor und zwei eingebauten 5 1/4 Zoll Diskettenstationen auf den Markt werfen. Der Bildschirm dieses KAYCOMP II genannten Rechners soll eine 9 Zoll Bildschirmdiagonale aufweisen, dies im Gegensatz zum Osborne, der einen 5 Zoll Bildschirm besitzt.

Als Betriebssystem findet CP/M Verwendung. Das Gerät soll mit ähnlicher Software wie der Osborne und zum gleichen Preis wie der Osborne angeboten werden. Also eine echte Alternative!

## 68'000 IM KOMMEN

(104/eh) Erhält der 68'000-Prozessor für die 16-bit'er die gleiche Bedeutung wie der Z80 für die 8-bit Maschinen?

Diesen Eindruck erhält man, wenn man verfolgt, wie viele der neuen 16-bit Kompaktcomputer mit diesem Prozessor ausgerüstet werden. Sirius und Fortune, Tandy (TRS Mod. 16) und Cromenco, der von Gerüchten umwobene Apple Lisa, Wicat, Sage und viele andere mehr sind mit dieser CPU ausgerüstet worden.

Diese 16-bit Computer stammen teilweise aus Häusern, die mit dem Z80 gross geworden sind. Vom Prozessor Z8000 hört man auf dem Kleincomputer-Markt nichts und vom 8086/8088 nur sehr wenig. Im Moment wird der Intel Prozessor 8086/8088 im wesentlichen nur von IBM und DEC eingesetzt.

## CP/M FUER 68'000?

(107/eh) Digital Research, bekannt durch sein weitverbreitetes CP/M-Betriebssystem, entwickelt im Auftrag von Hitachi ein dem CP/M 86 ähnliches Betriebssystem, welches mit dem Prozessor 68'000 arbeitet.

## DER SCHNELLSTE 8086

(106/eh) Gleichzeitig wurde vom Münchner Konzern eine Version des 16 Bit Prozessors 8086 vorgestellt der mit einer Taktfrequenz von 12 MHz arbeitet und somit zu den schnellsten derzeit erhältlichen 16 Bit Prozessoren gehört.

## MASCHINENPROGRAMME AUF HP-41

(105/pf) Es scheint nicht mehr lange zu dauern, wird es für den HP-41 drei gleichwertige "Programmiersprachen" geben: Den Tastencode, Synthetische Programme und Maschinenprogramme. Unter Insidern werden nun alle Softwarekräfte in die Entwicklung und Vereinheitlichung solcher Programme verlegt. Da hierzu hardwaremässig allerdings Zusatzgeräte aus dem Eigenbau und eine profunde Kenntnis des Betriebssystems notwendig sind, wird die Maschinensprache wohl nie die Verbreitung einer Synthetischen Programmierertechnik finden. Die Vorteile sollen allerdings, vor allem bezüglich Verarbeitungsgeschwindigkeit beträchtlich sein: Maschinenprogramme werden bis zu 50 mal schneller arbeiten als ein Tastencode-Programm.

Schon geistert die Vorstellung von einem PPC-ROM II in Maschinensprache umher - bis zu dessen Entwicklung wird es allerdings noch lange Zeit brauchen (eineinhalb Jahre?). Recht detaillierte Informationen über Maschinensprache, Konfiguration, Eigenbau usw. findet man im PPCCJ V9N3 - siehe dazu die Literaturhinweise in früheren Ausgaben.

Haben Sie bereits zu wenige I/O-Ports an Ihrem HP-41? Nun es gibt die Lösungen eines Port-Extenders, bzw. eines Zusammengiessens zweier ROMs. Was diesbezüglich und zur Lösung weiterer HP-41 Probleme in amerikanischen und deutschen Landen entwickelt wurde, erfahren Sie mittels einer Anfrage unter Angabe obiger Nummer.

## NEUES VON SINCLAIR

(106/eh) Clive Sinclair, der Vater des ZX80 und ZX81 will vor Ablauf des Jahres 1982 einen weiteren Kleincomputer auf den Markt bringen. Spectrum soll sein neues Kind heissen und umgerechnet etwa Fr. 1300.-- kosten. Für diesen Preis sollen ein Z80 Basic Computer mit zwei Diskstationen zu je 100 kByte Speicherkapazität und einem CP/M Betriebssystem erhältlich sein.

Nicht diese Mitteilung schreckte die Konkurrenz auf, sondern eine Mitteilung die ganz leise mitlief. Sinclair schätzt den Preis für die von ihm produzierte Minidiskettenstation auf etwa Fr. 175.--. Ueber diese sensationell billige Speicherstation lässt sich jedoch noch nichts in Erfahrung bringen, da von Sinclair ein Informationsstop verhängt wurde, um die laufenden Patentanmeldungen nicht zu beeinflussen.

## 64 KBYTE RAM VON SIEMENS

(105/eh) Auf dem Pariser-Bauelemente-Salon stellte Siemens den ersten 64 kByte RAM Baustein aus europäischer Produktion vor.

## AUCH IHR FACHAUFSATZ GEOHRT IN MIKRO+KLEINCOMPUTER

Fachlich lehrreiche Artikel von freien Autoren sind immer willkommen. Nehmen Sie bitte mit uns Kontakt auf. Interessante Beiträge, die wir nach sorgfältiger Prüfung abdrucken, honorieren wir angemessen. Legen Sie bitte Ihren Artikeln die notwendigen Diagramme, Zeichnungen und Listings (inklusive Kassette oder Diskette) bei.

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

# Kleincomputer aktuell

## TI-88, der 15000 Byte-Texaner

Hans-Jürgen OTTENBACHER

Wenn Texas Instruments Fachjournalisten aus allen Ecken und Enden Europas zu einer Produktepräsentation an die Cote d'Azur einlädt, dann muss es sich schon um ein aussergewöhnliches Ereignis handeln. Und aussergewöhnlich war allein einmal die Tatsache, dass Texas einen neuen Rechner vor der offiziellen Ankündigung in Amerika zuerst in Europa vorstellte - dies ist eigentlich bereits eine Schlagzeile wert.

TI-88 heisst nun das jüngste Kind der Texaner und ist nach fünf langen TI-59-Jahren als ausgereiftes Nachfolgemodell eigentlich längst überfällig. Ab Herbst soll der TI-88, der in punkto Leistung Enormes verspricht, auch in der Schweiz zu einem Preis noch unter tausend Franken auf den Markt kommen. Texas zielt mit seinem TI-88-Kraftpaket eindeutig auf die bisher unangefochtene HP-41-Domäne.

Ausserlich dem Trend der neuen Texas-LCD-Modellen folgend, ist als augenfälligster Pluspunkt das Display mit einer gestochen scharfen 5x7-Punkte-Matrix zu werten; 118 alphanumerische Zeichen in Gross- und Kleinschrift sowie mathematische Charaktere lassen sich auf 16 Stellen wiedergeben, und selbst Hoch- und Tiefstellen ist perfekt möglich.

Trotz Drehens und Wendens - den Anschluss für einen Magnetkartenleser wird man vergeblich suchen. Für "Kartenfans" ein schmerzlicher Verlust, der aber durch das neuartige Unterbringen von

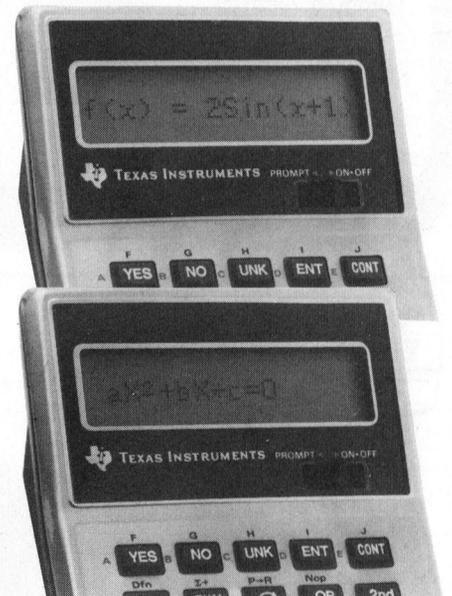
Permanentspeichern (RAM und/oder ROM) in Form von zwei Moduln im Gehäuse, mehr als wettgemacht wird.

Jedes dieser RAM-Moduln stattet den TI-88 mit zusätzlich 1184 Programmschritten aus, was ihm eine totale Programmierkapazität von 3328 Schritten gewährt. Da auch Texas' Neuer keine Programmiersprache verwendet, sondern im sogenannten Tastencode in AOS-Logik fährt, sind diese Schritte mit einzelnen Funktionsbefehlen gleichzusetzen.

Obwohl die gesamte Software, die für den TI-59 zur Verfügung steht, zum TI-88 voll kompatibel ist, muss die sehr umfangreiche Sammlung von

Programmen bei den TI-Benützern von diesen nochmals neu eingegeben werden.

Bestechend und für einen PPC völlig neuartig ist der interaktive Dialogverkehr in der jeweiligen Landessprache des Benutzers, in Englisch/Deutsch/Holländisch oder Französisch/Italienisch/Schwedisch, der durch fünf spezielle Funktionstasten YES, NO, UNK(NOWN), ENT(ER) und CONT(INUE) vorgenommen werden kann. Je drei Sprachen sind in den entsprechenden Mastermoduln enthalten.



Weiter verfügt der TI-88 über 231 programmierbare Funktionen, wovon 126 direkt über das Tastenfeld zugänglich sind, über 24 Flags (dies ist ein enormer Fortschritt zum TI-58/59) sowie über 10 verschiedene Tonsignale und eine interne Uhr. Und selbstverständlich können während eines Programmlaufes auch alphanumerische Meldungen ausgegeben werden.

Leider stand uns bis zum Redaktionsschluss noch kein Testgerät zur Verfügung. Wir werden aber sobald als möglich den TI-88 auf Herz und Nieren prüfen und einen objektiven Vergleich zu seinem anvisierten Konkurrenten HP-41 anstellen. Auf das Ergebnis dürfen wir gespannt sein.



## QUIC-N-EASI

### Wollten Sie nicht immer schon so programmieren:

Sie setzen sich an Ihr Terminal, geben auf dem Bildschirm im Dialog die Bildschirmmaske (natürlich mit allen Editiermöglichkeiten) ein, definieren schnell die Eingabefelder ... Zu jedem Eingabefeld können Sie eine kleine (oder große) Prozedur schreiben, die etwa Werte auf Plausibilität überprüft, Berechnungen durchführt oder in einer Datei nach etwas sucht. Datei ... natürlich ist ein ISAM dabei. Sie können über Schlüssel auf alle Datensätze wahlfrei zugreifen. Der Drucker ist natürlich auch einfach anzusteuern. Das Handbuch und alle Dialoge sind in Deutsch ... Quic-N-Easi ist nicht schwieriger wie Basic - nur werden die Programme viel, viel schneller fertig und die Ergebnisse sehen professionell aus. (Die Programme auch, da Quic-N-Easi zum strukturierten Programmieren zwingt - keine Angst, das ist nicht gefährlich). Als Anwendungsbeispiel bekommen Sie eine der besten Adressverwaltungsprogramme mitgeliefert, die Sie je gesehen haben - in Quic-N-Easi programmiert.

Fordern Sie die Unterlagen an - vielleicht ist Quic-N-Easi auch etwas für Sie. Handbuch DM 95,- in Deutsch; Quic-N-Easi DM 1498,- (Quic-N-Easi läuft auf CP/M Rechnern mit mind. 48K und TRS 80 Mod II). Fragen Sie Ihren Fachhändler nach Quic-N-Easi.

Wir liefern auch die passende Hardware - wir haben nur CP/M und MP/M Systeme mit wenigstens 64K RAM. Unsere Produkte:

**Osborne I** (Wir sind von Osborne autorisiert) DM 6.073,- inclusiv 4 Software Pakete: WORDSTAR, SUPERCALC, MBASIC und C BASIC.

**CS-2000 PLUS:** Z80 mit 80K (davon 62K für den Benutzer frei verfügbar), 2 Laufwerken 8" (gegen Aufpreis mit 2.4MByte), 4 seriellen und 2 parallelen Schnittstellen, CP/M, als Option MP/M und OASIS für DM 12.500,- mit 2 Jahren Garantie.

**CS-2500:** Z80 mit 64K (gegen Aufpreis mit 80K), 8" Diskettenstation mit 12MByte Kapazität und 5" Winchester mit 12MByte Platz (Preis auf Anfrage).

Dazu empfehlen wir auch Bildschirmgeräte von AMPEX mit bernsteinfarbigem Bildschirm und abgesetzter Tastatur, sowie den grafikfähigen, schnellen Drucker von NEC (CS-2000 Plus und CS-2500 sind S-100 Systeme!)

### Neu: MORROW DESIGNS: DECISION-1

Ein neuer Rechner mit IEEE-696 Bus (S-100) mit MICRONIX-Betriebssystem für Multi-USER und CP/M als Einplatz-Betriebssystem.

**OASIS** ab Juli lieferbar.

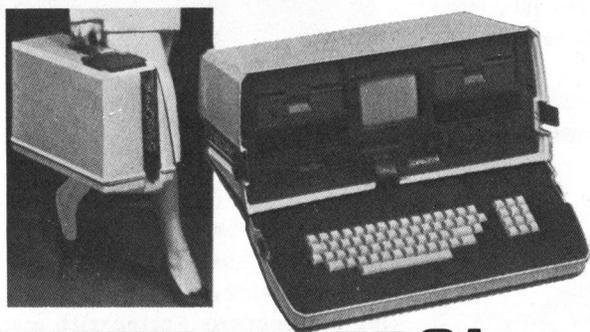
Preise incl. MwSt. - Preisänderungen vorbehalten.

## QUIC-N-EASI™

Besonders gerne liefern wir an Kunden aus Hessen oder dem Bodenseeraum und Oberschwaben. Machen Sie einfach einen Termin mit uns aus! Wir erstellen auch individuelle Software für Sie - vielleicht haben wir Ihr Problem schon gelöst.



**SOVITREL SA présente**  
L'ordinateur portable  
Der Tragbare Computer



## OSBORNE 01

- 64 K de mémoire,
- microprocesseur Z 80 A, 4 MHz,
- écran 5" + 2 unités de disquettes de 100 K octets,
- clavier QWERTY avec minuscules,
- interfaces RS-232-C, IEEE 488, MODEM,
- CP/M + CBASIC + MBASIC + SUPERCALC + WORDSTAR avec MailMerge.

**PRIX 6900.-**

**Action d'introduction en Suisse jusqu'au 15.9.82**  
**Einführungspreis für die Schweiz bis zum 15.9.82**

3000.- à la commande sur CCP 17-9086  
3000.- par remboursement à la livraison

**Total Fr. 6000.-**

LES OPTIONS: ● AZERTY accentué, ● Moniteur externe,  
● Double densité, ● HARD DISK externe de 5 Mbytes.

**RECHERCHONS DISTRIBUTEURS POUR TOUTE LA SUISSE**

Prendre contact téléphonique 037/242 282  
Sovitrel SA Av. Midi 1701 Fribourg (Suisse)

## Genie II



- Z80 A CPU
- 64 KRAM erweiterbar bis 256 KRAM
- Schreibmaschinen-Tastatur (deutsche Norm)
- Zehner-Tastenblock
- 8 frei programmierbare Funktionstasten
- Integrierter High-Resolution Datenmonitor 12" (grün, entspiegelt)
- Display-Format softwareseitig wählbar (64 x 16 oder 80 x 24)
- 2 integrierte Floppy-drives für 5 1/4", 2 x 350 Kbytes (bis ca. 1,4 Mbytes)
- Bis 80 Track, Double side, Double density
- Weitere Floppy-drives anschliessbar
- Akku-gepufferte Echtzeituhr
- Diverse Betriebssysteme einschliesslich CP/M 2.2 und NEWDOS 2.0
- **Ab SFr. 5950.-**



ZEV ELECTRONIC AG  
TRAMSTRASSE 11  
CH - 8050 ZÜRICH  
TELEPHON 01 312 22 67  
MIKROCOMPUTER

## Speicheradressierung über 64 kByte

Leopold ASBÖCK

Als vor einigen Jahren die ersten Kleincomputer auf den Markt kamen, waren Halbleiterspeicher eine teure Angelegenheit. Platinen mit 8 KByte RAM waren Sensationen, EPROMs waren teuer und zu 256x8 Bytes organisiert. Der Adressbereich der 8-bit-Mikroprozessoren schien mit 64 KByte riesig dimensioniert zu sein. Heute ist dieser Bereich für viele Anwendungen zu klein. Mit raffinierten Schaltungsdetails werden die engen Grenzen gesprengt.

Mit dem rapiden Sinken der Preise von Halbleiterspeichern - statischen und dynamischen RAMs - ist die Grösse des Arbeitsspeichers eines Kleincomputers keine Preisfrage mehr, sondern eine Frage der Ideen des Entwicklungsingenieurs.

Der neue Industriestandard von 64 Kbit in einem 16-poligen Speicher-IC löst die vor kurzem noch als sensationell empfundenen 16-Kbits-ICs ab. Inzwischen existieren aber bereits Prototypen von ICs mit 256 K Speicherplätzen. Weltweit werden 1982 schätzungsmässig rund 500 Millionen Stück von den neuen 64-Kbit-RAMs benötigt werden, eine gigantische Zahl. Mit nur acht ICs lässt sich auf wenigen Quadratzentimetern der von einem 8-bit-Prozessor adressierbare Speicherbereich voll ausbauen.

Während 16-bit-Prozessoren mehrere Megabytes adressieren können, ergeben sich bei 8-bit-Prozessoren Schwierigkeiten, wenn ein Speicherbereich von 64 KByte überschritten werden soll, da diese Grenze durch die 16 Adressleitungen gesetzt wird. Entweder begnügt man sich mit diesem Bereich und segmentiert lange Programme oder aufwendige Programmiersprachen - das bedeutet aber erhöhten Softwareaufwand und Zeitverlust durch den häufigeren Zugriff auf langsame externe Speichermedien - oder man segmentiert einen grösseren Speicherbereich, was gleichfalls erhöhten Software- und Hardwareaufwand bedingt.

Meist werden über Portadressen mehrere 64-KByte-Blöcke angespro-

chen, einer wird jeweils aktiviert, während die anderen stillgelegt werden, wobei das Refreshing der

dynamischen Speicher in allen Blöcken aber gewährleistet werden muss, um Datenausfall zu verhindern.

Das Umschalten zwischen den einzelnen Speicherblöcken verlangt einen Software-Supervisor, ein Programmsegment, das die Umschaltstelle überwacht. Dieser Supervisor muss in allen Blöcken gegenwärtig sein und darf sich nicht selbst wegschalten. Manche Computer

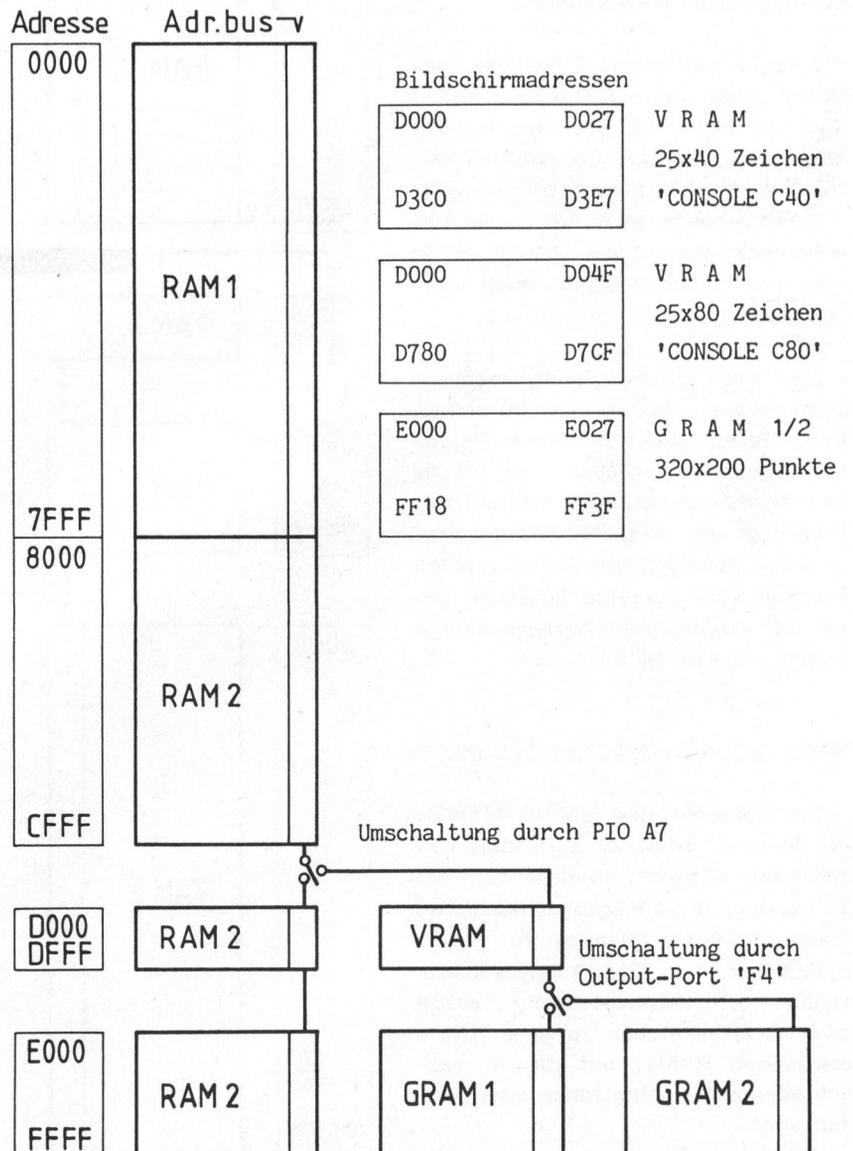


Abb.1 Zugriff aus dem Adressbereich 0000-CFFF

# Kleincomputer aktuell

verwenden auch ein "shared memory" und schalten nur Blöcke von 48 K-Byte oder 56 KByte, was den aktiven Speicherbereich weiter einengt.

Ein zusätzliches Problem ergibt sich, da gewisse Speicherabschnitte jederzeit von allen Speicherbereichen aus aktiviert werden müssen, wie zum Beispiel Bildschirmspeicher, Grafikspeicher, Diskbuffer etc. Oft umgeht man Komplikationen und trennt diese Bereiche vollständig ab, speziell dann, wenn diese Aktivitäten einen eigenen Prozessor erfordern. Zum Teil verkehren diese Mehrprozessorsysteme nur über Parallelschnittstellen oder serielle Schnittstellen miteinander.

Speicherkarten mit 256 KByte "on-board" sind keine Seltenheit mehr. Was bleibt ist das Problem der Organisation. Manche Computerhersteller finden recht elegante Lösungen, während andere es nicht einmal der Mühe wert finden, das Urlader-PROM oder den vollen EPROM-Bereich wegzuschalten.

Eine vorbildliche Lösung, mehrere Speicherbereiche parallel zu schalten, bietet SHARP in seinem Personal Computer MZ80B. Auf Grund leichter Ueberschaubarkeit soll gezeigt werden, wie drei Speicherbereiche einen einzigen Adressbereich belegen oder derselbe Speicher sogar auf verschiedene Adressbereiche gelegt werden kann.

## SPEICHERORGANISATION DES MZ80B

Die Speicher des SHARP MZ80B, die in m+k computer 81-5 kurz beschrieben wurden, gliedern sich bei Vollausbau in 64 KByte dynamisches RAM, 2KByte Urlader in einem EPROM vom Typ 2716, 2 KByte statischen Bildschirmspeicher, sowie zwei Grafikspeicher zu je 8 KByte statischen RAMs, auf die in verschiedenen Kombinationen zugegriffen wird.

Unberücksichtigt bleibt der Zeichengenerator mit 2 KByte, auf den der Prozessor nicht zugreift und

der somit auch den Adressbereich nicht belastet.

Zum besseren Verständnis werden folgende Bezeichnungen gewählt:

IPL-ROM (Initial Program Loader) ist das Urlader- oder Bootstrap-Programm, das in einem EPROM gespeichert ist und 2 KByte umfasst.

RAM1 ist das serienmässig vorhandene dynamische RAM von 32 KByte

RAM2 bezeichnet das zum vollen Speicherausbau zusteckbare dynamische RAM von 32 KByte.

VRAM ist der 2 KByte umfassende Bildschirmspeicher, in dem die Codes der auf dem Bildschirm darzustellenden Zeichen abgelegt werden.

GRAM1 ist die erste Grafiksseite, die 8 KByte umfasst und zur Darstellung von 320x200 Punkten dient.

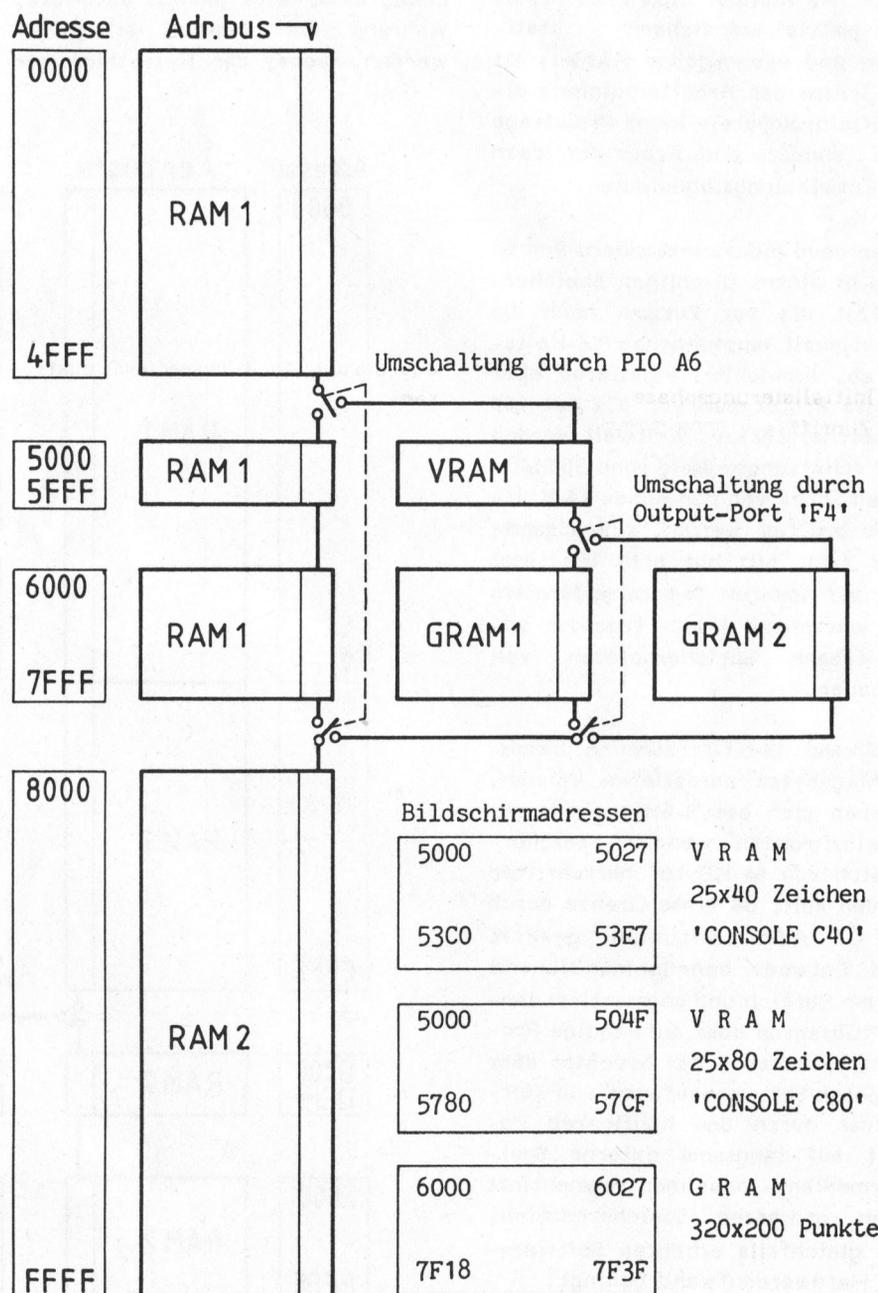


Abb.2 Zugriff aus dem Adressbereich 8000-FFFF

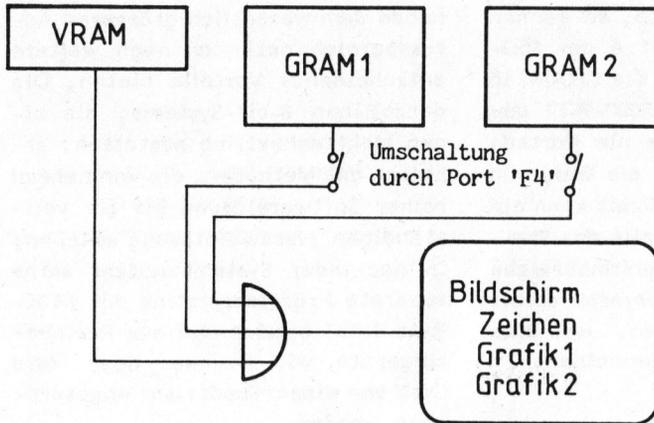


Abb.3 Ausgangsschaltung der V/G-RAMs

erzwingt, während die Monitor-Reset-Taste einen Warmstart bei 0000 bewirkt und die IPL-Prozedur überspringt. Mit dem Warmstart kann ein definierter Ausstieg aus einer endlosen Programmschleife erfolgen, ohne das Programm zu zerstören oder einen Neubeginn zu bewirken. Der Computer bringt hierauf die Monitormeldung und steht zur Entgegennahme von Befehlen bereit.

Da während eines Programmablaufes 64 KByte RAM voll zur Verfügung stehen, bleibt kein Adressbereich frei, um auf das Video-RAM (VRAM) und die Grafikseiten (GRAM1, GRAM2) zuzugreifen. Auf diese Speicher greift zudem noch der Videocontroller zu. Da die beiden Grafikseiten gleichzeitig oder wechselweise, der Video-RAM-Inhalt immer angezeigt wird, ergibt sich ein kompliziertes Umschalten zwischen allen Speicherbereichen, das von den SHARP-Technikern mit typisch japanischer Findigkeit elegant gelöst wurde.

GRAM2 ist die zweite Grafikseite mit analoger Funktion wie GRAM1.

Drei Phasen des Zugriffs auf die Speicher sind zu unterscheiden:

1. Initialisierungsphase
2. Zugriff aus 0000-CFFF
3. Zugriff aus 8000-FFFF

Während der Initialisierungsphase benötigt das Umlader-EPROM den Adressbereich ab 0000. Dort liegt aber jener RAM-Bereich, in den Programme ab Diskette oder Kassette eingelesen werden sollen.

Manche Firmen behelfen sich durch Abschalten des ersten 16-KByte-Blocks, laden in den Restbereich und springen in ein Programm in diesem Bereich, das das EPROM wegschaltet und aus dem höheren Bereich oder von externen Speichermedien das vollständige Betriebsprogramm in den ersten 16-KByte-Block ladet.

SHARP geht anders vor: der Bereich RAM1 (0000-7FFF hexadezimal) wird über die Adressleitung A15 auf die Adressen 8000-FFFF geschaltet. Nun wird vom IPL-Programm, das von 0000-07FF residiert, der Computer initialisiert und von Diskette, Kassette oder sogar von einem externen EPROM das Betriebsprogramm in die Speicherplätze ab 8000 geladen. Ist diese Operation beendet,

werden per Software zwei Bits eines Ausgabeports aktiviert, die das EPROM wegschalten, RAM1 in den Anfangsbereich zurückschalten und einen Prozessor-RESET mit Programmausführung ab Adresse 0000 ausführen.

Damit nimmt der Computer seinen regulären Betrieb mit der eingelesenen Software auf.

Derselbe Ablauf findet statt, falls man mit der IPL-Taste auf der Computerrückseite einen Kaltstart

Nehmen wir an, wir befinden uns im Programmablauf irgendwo im Speicherbereich von 0000 bis CFFF, also in den unteren 52 KByte des Adressbereiches und es sollen Daten im Video-RAM oder in einem der beiden

Ausgabe an Port 'F4'	GRAM1		GRAM 2	
	Eingabe	Ausgabe	Eingabe	Ausgabe
00	x	-	-	-
01	-	-	x	-
02	x	x	-	-
03	-	x	x	-
0C	x	-	-	x
0D	-	-	x	x
0E	x	x	-	x
0F	-	x	x	x

Abb.4 Ein-/Ausgabe-Schaltung der GRAMs

# Kleincomputer aktuell

Grafikspeicher zur Anzeige auf dem Bildschirm abgelegt werden.

Mit einem einzigen Bit (Bit 7 des PIO-Ports A) lässt sich der RAM-Bereich D000-FFFF wegschalten (Bild 1) und der VRAM-Bereich sowie der GRAM-Bereich im gleichen Adressbereich zuschalten, wobei ein Byte, das über die Portadresse F4 ausgegeben wird, noch als Schalter wirkt, ob GRAM1 oder GRAM2 zur Eingabe aktiviert wird. Gleichzeitig wird durch dieses Byte auch der Ausgabemodus (Bild 4) festgelegt, da der Videokontroller alle drei RAM-Bereiche (VRAM, GRAM1, GRAM2) erfasst und durch entsprechende Ausgangsverknüpfungen (Bild 3) beliebige Anzeigekombinationen ermöglicht werden.

Befinden wir uns im Programm im Speicherbereich von 8000-FFFF, also

in den oberen 32 KByte, so können (Bild 2) durch das Bit 6 des PIO-Ports A das VRAM und die GRAMs in den Adressbereich von 5000-7FFF geschaltet werden, über die Portadresse F4 bleibt wieder die Wahl von GRAM1 oder GRAM2. Somit kann ein Programm aus jeder Stelle des Speichers auf die Bildschirmbereiche zugreifen ohne sich vorerst selbst verschieben zu müssen, um eine Selbstabschaltung zu vermeiden.

## MULTI-USER-SYSTEME

Vor allem bei Multi-User-Betrieb traten Probleme mit 8-bit-Prozessoren auf, die eine Vielfalt von Lösungen zur Folge hatten. Es ist deshalb naheliegend, dass bei der Entwicklung von Multi-User-Systemen in Zukunft vor allem 16-bit-Prozessoren ins Auge gefasst werden, die

neben dem wesentlich grösseren Adressbereich natürlich noch weitere entscheidende Vorteile bieten. Die derzeitigen 8-bit-Systeme, die einen Mehrfachbetrieb gestatten, arbeiten mit Methoden, die von nahezu reiner Softwarelösung bis zur vollständigen Hardwarelösung reichen, in der jeder Systembenutzer seine separate Prozessorplatine mit 64 K-Byte RAM besitzt und nur Peripheriegeräte wie Drucker oder Hard Disk von einem Supervisor angesprochen werden.

Alle Lösungen zeigen Ideenreichtum und tiefen Systemdurchblick der Entwicklungsingenieure, Leistung und Preis differieren aber dennoch und können nur von einem Fachmann eingestuft werden. Ein unerfahrener Anwender sollte nur Komplettlösungen bevorzugen, die auch seine Softwarewünsche voll abdecken.



# Kleincomputer

...und alles, was dazugehört

## 8 Bit

- **commodore** COMPUTER
- **apple computer**
- **DCT-SUPERBRAIN**
- **DCT Compustar**
- **HEWLETT PACKARD**
- **MONROE**  
Litton Systems For Business

## Peripherie

- Drucker/Plotter
- Floppies
- Hard-Disk

## Zubehör

- Disketten
- Farbbänder
- Bücher

## 16 Bit



- **sirius** COMPUTER
- **IBM-PC**

Fordern Sie den ausführlichen Katalog an beim

# Computer Shop Luzern

DIALOG COMPUTER  
TREUHAND AG  
Seeburgstrasse 18  
6002 Luzern

Telefon 041 - 31 45 45

## Textverarbeitung im arbeitspsychologischen Vergleich

Prof. Dr. Alfred Lang

**Im Hinblick auf die Einführung der Textverarbeitung in einem Universitätsinstitut hat der Autor zwei der meistverbreiteten Textprogramme für die Verwendung auf universellen CP/M-Mikrocomputern einer detaillierten vergleichenden Bewertung unterzogen: «WordStar» nach dem Modell der Grosscomputer konstruiert und für Mikros adaptiert, und «Spellbinder», eher für kleinere, autonome Computer entwickelt.**

Die meistverbreitete und bestbekannte Textverarbeitungs-Software auf CP/M-Mikrocomputern scheint der WordStar von Micropro zu sein. Neben dem Studium des Handbuches hatte ich Gelegenheit, das Programm für kurze Zeit praktisch zu erproben. Meine Bemerkungen hat ein Kollege, der über bald zwei Jahre Erfahrung mit dem WordStar verfügt, überprüft.

Seit über zwei Jahren arbeite ich fast täglich an einem Exidy Sorcerer, hauptsächlich mit der Textverarbeitung. Anfänglich benutzte ich das WP ROM PAK, seit Sommer 1982 den Spellbinder von Lexisoft, einen CP/M-Word-Processor, von dem das ROM PAK einen Auszug darstellt.

Ich schicke dies voraus, um deutlich zu machen, dass ich mit Spellbinder wesentlich mehr Erfahrung habe als mit WordStar. Vergleichende Bewertungen sind unvermeidlich von persönlichen Vorlieben abhängig. Die nachfolgende Übersicht über die Besonderheiten der beiden Programme wird deutlich machen, dass meine Bevorzugung von Spellbinder auf mehr als persönlicher Gewohnheit beruht.

### Die Super-Textverarbeiter

Beide Programme sind sogenannte «Super-Textverarbeiter». Sie erlauben das bildschirmorientierte Editieren von beliebigen Texten (auch Programmen in Assembler oder höheren Sprachen), das «Verwalten» von Texten auf Disk (Floppy oder Hard Disk) sowie das formatierte Drucken der Texte auf Matrix- oder Typenraddruckern. Beide Programme verfügen über nahezu 150 Befehle, mit denen man weitgehend die gleichen Leistungen vollbringen kann. Beide Programme

bieten zusätzlich zur eigentlichen Textverarbeitung eine einfache «Programmiersprache», mit der man fast beliebige textorientierte Verarbeitungsprobleme lösen kann, darunter Robot-Briefe, Adress-Verwaltung usw. MAIL-MERGE (beim WordStar) muss allerdings extra gekauft werden; M-SPEAK (MACRO) beim Spellbinder ist im Grundprogramm enthalten (gebrauchsfertige MACROs werden bei einigen Lieferanten gratis mitgeliefert, bei andern muss man sie extra kaufen).

Die beiden Programme realisieren recht unterschiedliche «Philosophien» der Textverarbeitung. Dies betrifft insbesondere die Behandlung der «Mensch-Maschine-Schnittstelle». Damit meine ich hier weniger den ergonomisch-technischen Aspekt (der je nach Bildschirm und Tastatur bei beiden Programmen schlecht oder recht gelöst sein kann), sondern die ergonomisch-psychologische Seite. Die Frage richtet sich also nach der Art und Weise der Beziehung zwischen Gerät und Bediener, die der Autor des Programms implizit eingebaut hat. Letzten Endes steht hier die Alternative zur Diskussion, ob der Benutzer gezwungen wird, sich den Eigenschaften des Geräts anzupassen, oder ob das Gerät unter Berücksichtigung der Eigenschaften des Benutzers konstruiert worden ist. Generell hat jede neue Technik zumeist einen «Bias» zuungunsten des Menschen; mit zunehmender Entwicklung wird ein mehr oder weniger günstiger Kompromiss erreicht.

Fazit meines Vergleichs von WordStar und Spellbinder ist die Feststellung, dass Spellbinder einer Reihe von psychologischen Kriterien besser ent-

spricht und daher bei gleicher Leistungsfähigkeit WordStar vorzuziehen ist. Aber ich will das im Einzelnen belegen.

Die grundlegenden Unterschiede kann ich in zwei Aussagen zusammenfassen:

1) Im WordStar schreibt man Texte in einem bestimmten Druckformat in ein Diskfile; im Spellbinder schreibt man gewissermassen «reinen» Text, den man nach Bedarf in einem Diskfile ablegen und/oder in einem (auch nachträglich) wählbaren Format ausdrucken kann.

2) WordStar geht von einem relativ komplexen Fall von Druckformat aus und man muss mit besonderen Befehlen zum einfachen Fall «abbauen»; Spellbinder setzt den Benutzer zunächst vor den einfachen Fall und lässt ihn die komplizierteren Sachen durch Zusatzbefehle «aufbauen».

### Grundsätzliches

Um diese zwei Aussagen zu verstehen, betrachten wir zunächst, was unmittelbar nach dem Laden des Programmes geschieht. Beim Spellbinder erscheint zusammen mit der Sign-on-Message die Frage, ob das «alte File», welches sich von einer allfälligen vorausgehenden Bearbeitung noch im RAM befindet, weiterbearbeitet werden soll. Die Antwort «Y» macht dieses File direkt zugänglich; das ist nicht nur praktisch beim wiederholten Kompilieren von kleinen Programmen, sondern höchst nützlich nach einem allfälligen System-Crash. Ein beliebiger Tastendruck statt «Y» bringt den Benutzer unmittelbar in den EDIT-Modus, d.h. den Zustand, in welchem Text zunächst in den leeren Arbeitsspeicher geschrieben und dann modifiziert wird. Der Benutzer kann einen Text herstellen, welcher je nach RAM-Grösse im Umfang von 7 bis 40 kBytes (entsprechend 2 bis 20 Schreibmaschinenseiten; über längere Texte später mehr) direkt zugänglich ist. Spellbinder (19K einschliesslich

M-Speak) und CP/M (6K) nehmen zusammen etwa 25 kBytes ein. Wird Hilfe benötigt, so können mit dem Befehl HE diverse Befehlslisten abgerufen werden; man kann im Unterschied zu WordStar auch eigene «Gedächtnis-krücken» auf den Bildschirm holen.

Beim WordStar ist die Sache etwas umständlicher, wie nach dem Umfang von Manual und Programm zu erwarten ist: WordStar umfasst ohne CP/M mehr als 40 K, obwohl zwei overlay-files verwendet werden und Mail-Merge ebenfalls als overlay noch dazu kommt. Nach dem Sign-on kommt ein Haupt-Menu, das sogenannte NO-FILE-MENU, gefolgt vom File-Verzeichnis des aktuellen Disks auf den Bildschirm. Vor Beginn der Arbeit muss man einen oder mehrere aus 12 Befehlen geben. Beispielsweise kann man das File Directory ausschalten oder einen von drei Help-Levels setzen (keine Hilfen bis viel Hilfe, wobei ca. ein Drittel des Bildschirms ständig von Befehlslisten beansprucht wird). Man muss dann den Typus des Textes vorwählen, entweder Document, d.h. gewöhnlichen Text, der als fortlaufende Zeichenkette gespeichert wird, oder Non-Document, z.B. ein Programm, wobei im File jede Zeile mit CRLF abgeschlossen wird. In beiden Fällen wird man nach einem Filenamen gefragt. Dann prüft WordStar, ob das betreffende File auf dem Disk schon existiert. Im zutreffenden Fall wird das File eingelesen und ist nun zur Bearbeitung verfügbar, andernfalls kann das Schreiben und Modifizieren im noch leeren Arbeitsspeicher beginnen. Der Benutzer braucht sich nicht darum zu kümmern, ob sein Text kurz oder lang ist, die Koordination von RAM und Disk Memory ist weitgehend automatisiert.

Der Benutzer arbeitet also im WordStar stets mit einem benannten File, das primär auf dem Disk beheimatet ist, aber natürlich im RAM bearbeitet wird. Im Spellbinder hingegen arbeitet man stets mit einem Text, der unmittelbar im RAM befindlich ist. Will man im Spellbinder den Text aufbewahren, so muss man einen geeigneten Speicherbefehl geben und wird dann auch nach einem Filenamen gefragt. Der Unterschied mag oberflächlich geringfügig aussehen, er ist aber grundlegend. Spellbinder ist für den Benutzer transparenter und eher auf kleinere, autonome Computer gedacht. WordStar ist nach dem Modell

der Grosscomputer konstruiert und für Mikros adaptiert; d.h. dem Benutzer wird die Sorge um die Speicherung seines Textes abgenommen, er verliert dadurch aber auch einen Teil seines Handlungsspielraumes, er verliert an Verfügungsfreiheit über seinen Text. Mit andern Worten und bewusst überspitzt formuliert: Im WordStar ist der Benutzer ein bisschen mehr der Sklave des Computers, im Spellbinder ist der Computer eher der Sklave des Benutzers. Ich komme darauf zurück.

## Drucken

Das wird noch deutlicher, wenn es ans Drucken geht, was ja doch das unmittelbare Ziel der Textverarbeitung darstellt. Und damit belege ich auch meine zweite Aussage. Hat der Benutzer von Spellbinder einen Text im RAM, so bewegt er den Cursor an den Beginn der zu druckenden Stelle, wechselt in den COMMAND-Mode und gibt bei bereitgestelltem Drucker den Befehl P. Ausgedruckt wird dann der Text, im wesentlichen identisch mit der Anordnung auf dem Bildschirm. Nach 60 Zeilen unterbricht Spellbinder, um dem Benutzer Gelegenheit zu geben, das nächste Blatt einzuspannen. Das ist der einfache Fall. Der Benutzer hat natürlich die Möglichkeit, vor dem Ausdrucken eine Reihe von Druck-Parametern zu setzen (oder auch direkt im Text einzufügen), welche ein seinen Wünschen besser gerecht werdendes Produkt hervorbringen. So will er vielleicht nur 40 Zeilen pro Seite und dafür einen grösseren Zeilenabstand und eine höhere Zeichendichte bei 75 Zeichen pro Zeile, oder er druckt auf Endlospapier und macht von der Form Feed Möglichkeit Gebrauch. Die Verwendung von Titelseiten und Paginierung oben oder unten, verschiedene Formen der Auszeichnung wie Unterstreichung, Fettdruck etc., Blocksatz, Proportional-schrift usw. sind unter Voraussetzung eines geeigneten Druckers möglich. Der Benutzer baut sein Wunsch-Satzbild durch Veränderung von Parametern oder Hinzufügen von Befehlen ausgehend vom einfachen Fall auf. Die Anordnung der Druckbefehle in der Y-Tabelle dürfte allerdings besser durchdacht sein. Doch lässt sich mit Leichtigkeit eine personalisierte Spellbinder-Version herstellen, derart dass der «einfache Fall» eigenen Wünschen gerecht wird.

Nicht so im WordStar. WordStar druckt grundsätzlich nur Files, die auf Disk gespeichert sind. Gibt man den Druckbefehl P, so wird man zunächst vor eine Reihe von sieben Fragen gestellt, die, wenn man sie ignoriert, zu einem Satzbild führen, welches einem verhältnismässig «komplexen» Fall entspricht: Blocksatz (d.h. links und rechts gerader Rand) mit paginierten Seiten. Das ist unpassend bei Korrespondenz oder bei Manuskripten für Artikel und dergleichen. Natürlich ist es möglich, den «komplexen Fall» auszuschalten. Und ebenso ist es möglich, mit weiteren Befehlen letztlich (praktisch) dasselbe Satzbild zu erzielen wie mit Spellbinder.

Der Unterschied besteht im Vorgehen: beim Spellbinder schreibt man Text im wesentlichen als fortlaufende Zeichenkette in einem Standard-Bildschirm-Format (der «einfache Fall») und lässt ihn zum Drucken durch eine geeignet zusammengestellte «Druck-Maske» laufen; die Druck-Maske kann ein (nichtdruckender) Bestandteil des Textes sein.

Beim WordStar hingegen schreibt man den Text in einem vorausbestimmten Druckformat, welches auf dem Bildschirm so gut wie möglich «simuliert» wird (man kann natürlich nicht auf dem Bildschirm proportional oder in Zeilenlängen grösser als Bildschirmbreite darstellen). Der Text wird dann stets auch in diesem Format gespeichert; und wenn man ihn in einem anderen Format drucken will (z.B. mit um acht Zeichen verbreiterten Zeilen, damit man nicht für ein paar Zeilen noch eine neue Seite anfangen muss), dann muss man den ganzen Text neu formatieren und auch ein neues File anlegen bzw. das alte überschreiben.

Nur ein kleiner Unterschied besteht im Resultat: Beim Blocksatz-Proportionaldruck, welcher vom Computer aus gesteuert wird (nicht im Drucker selbst erzeugt) befolgt Spellbinder das von den Typographen bevorzugte Verfahren, den überflüssigen Leerraum gleichmässig auf die Wortzwischenräume aufzuteilen. WordStar hingegen verteilt den Leerraum auf die Buchstabenbreiten, was ein eher unruhiges Schriftbild erzeugt. Natürlich kann man durch Silbentrennung an optimalen Stellen das Bild verbessern. Beide Programme bieten dafür eine Trennungshilfe an. Auch hier ziehe ich das Verfahren von Spellbinder vor. Ich halte es für benutzerorientier-

ter, weil es erlaubt, gewissermassen strengere oder largere Ansprüche zu stellen, indem man minimale und maximale Breiten der zulässigen Wortzwischenräume setzt. Im übrigen habe ich aber bezüglich Druckformatierung nichts gefunden, was nur im einen und nicht auch im andern Programm hergestellt werden könnte.

## Editieren

Kehren wir zurück zum Editieren auf dem Bildschirm. Beide Programme puffern die Eingabe: Beim schnellen Tippen gehen in der Regel keine Zeichen verloren. Beide Programme bieten die üblichen Cursorbewegungsbefehle an. Im WordStar sind sie mit Control-Tasten (ctrl-S,D,E,X im Koordinatenkreuz) zu bedienen (eine Umstellung auf die Cursorbewegungstasten des Terminals scheint ausgeschlossen zu sein), den Spellbinder kann man je nach Terminal für die ergonomisch praktischen Tasten konfigurieren. Cursorbewegungen sind in erster Linie zeichenorientiert, wie ja generell beide Programme auf dem Manipulieren von Zeichen (Bytes) basieren. Beide Programme bieten aber Super-Cursorbewegungen an: Springen von Wort zu Wort, ans rechte Ende der Zeile oder zurück an den Zeilenbeginn, auf vorher irgendwo gesetzte Marken, an den Beginn oder das Ende des ganzen Textes usw. Aber dann ist da wieder der folgenreiche kleine Unterschied in der «Philosophie» der Textverarbeitung, nämlich dem Benutzer alles Automatisierbare abzunehmen oder ihn stets noch als denkenden Menschen vorzusetzen.

WordStar erlaubt das Springen von Seite zu Seite, vorwärts und rückwärts sowie das Zurückspringen an die Stelle der vorausgehenden Operation, was manchmal sehr praktisch ist und beim Spellbinder fehlt (bzw. durch Setzen einer Marke vorausgeplant werden müsste). Im WordStar kann man sogar mit einstellbarer Geschwindigkeit zeilen- oder seitenweise durch seinen Text «rollen» und an der gesuchten Stelle unterbrechen. Beim Spellbinder ist hingegen das seitenweise Springen oder Rollen nicht möglich, weil der Seitenumbruch ausser in besonderen Fällen erst beim Drucken erfolgt; dafür erlaubt aber Spellbinder das Springen von Satz zu Satz und von Abschnitt zu Abschnitt, vorwärts und rückwärts. Zudem ist

Spellbinder «zeilenorientiert»: eine praktische Einrichtung, die beim WordStar fehlt. Befehle wie F (forward), B (back), D (delete), R (read from disk), W (write to disk), P (print), V (verify) und H (hold, vgl. unten) können mit einem Zeilen-Koeffizienten von 1 bis n versehen werden: Beispielsweise heisst «f200»: springe 200 Zeilen vorwärts; das bringt den Cursor sehr rasch in näher oder entfernter liegende Textbereiche.

WordStar versucht seinen Beweglichkeits-Nachteil durch ein Angebot von mehreren verschiedenen Marken wettzumachen. Man kann also gezielt an jede einzelne von 10 markierten Textstellen springen; leider bleiben die Marken im Disk-File nicht erhalten. Spellbinder erlaubt beliebig viele und speicherbare Marken; man springt mit wiederholtem f oder b rasch an eine gewünschte Stelle.

Cursorbewegungsbefehle bestehen im WordStar aus einem Kontrollzeichen (die einfachen, z.B. ctrl-A für Cursor ein Wort nach links) oder einem Kontrollzeichen plus einem Buchstaben (die Superbewegungen, z.B. ctrl-Q R für Cursor nach Beginn des Files). Im Spellbinder kann man die einfachen Befehle mit einem einzelnen Tastendruck auslösen; je nach Terminal sind dabei die Funktions- oder Extratasten (beim Sorcerer z.B. das numerische Tastenfeld) verwendbar. Die Superbefehle erfordern ein Umschalten vom Edit- in den Command-Modus (ein Tastendruck) oder ein Umschalten zwischen Wort-, Satz- oder Abschnitt-Modus.

Im WordStar schreibt man seinen Text stets in einem bestimmten Format. Das hat zur Folge, dass jede Änderung des Formats von Befehlen begleitet sein muss, welche die Anpassung des schon bestehenden Textes im Rest des Files bewirken (ctrl-B für jeden Abschnitt). Im Spellbinder hingegen schreibt man gewissermassen «reinen» Text, natürlich in Abschnitte gegliedert, der erst hinterher für die eine oder andere Druckweise formatiert wird. Änderungen an einer Stelle haben demnach geringere Folgen für den nachfolgenden Text, so dass die Anpassung automatisch erfolgen kann. Will man beispielsweise die Druckbreite von den üblichen 65 Zeichen auf 55 reduzieren, so muss man im WordStar die Ränder neu setzen und den ganzen Text durchformatieren. Im Spellbinder hat man die Wahl:

Entweder ändert man nur die Druckmaske im Parameter Zeilenbreite, oder man will den neuen Text auch vorher auf dem Bildschirm sehen und befiehlt L55, was sehr rasch den ganzen Text neu anordnet, doch nur auf dem Bildschirm, das File bleibt unbeeinflusst.

Die Zeilenorientiertheit des Spellbinder bringt eine enorme Flexibilität. Die Zeilenlänge ist von 17 Zeichen bis doppelte Bildschirmbreite anpassbar; bei überlangen Zeilen wird jeweils nur ein Teil dargestellt. Vorübergehend kann man auch den linken Rand auf jede Tabulatorstelle setzen und so eingerückte Abschnitte darstellen. Beim WordStar setzt man hingegen den linken und den rechten Rand. Die Zeilenlänge ist theoretisch beliebig, doch werden Zeichen, die auf der Bildschirmbreite nicht Platz finden, als weitere (markierte) Zeile(n) angefügt. Man kann den Rand jederzeit umstellen, muss aber dann die betreffenden Abschnitte neu formatieren. Wie bei einer Schreibmaschine kann man dafür bei einzelnen Zeilen links und rechts über den Rand hinaus schreiben.

Der kleine Unterschied zwischen Zeilen- oder Seitenorientierung wirkt sich auch auf die Verfahren zur Behandlung von Textblöcken aus. Was der Redaktor früher umständlich mit Kopie, Schere und Leim hantierte, das Verschieben oder Verdoppeln von Sätzen, Abschnitten oder sonstigen Textteilen, das macht der Wortprozessor elektronisch sofort und das Resultat ist jedesmal eine «Reinschrift» auf dem Bildschirm. WordStar verlangt das Markieren des Blockanfangs (mit ctrl-K B) und des Blockendes (mit ctrl-K K), worauf der Block an die neue Cursorstelle entweder verschoben (unter Entfernung an der ursprünglichen Stelle mit ctrl-K V) oder kopiert (mit ctrl-K C) wird. Die Marken sind spezifisch für Blockanfang und -ende; man kann einen Block auch löschen oder in ein eigenes Diskfile schreiben. Letzteres empfiehlt sich für Blockbewegungen in überlangen Files.

Im Spellbinder scheint mir die Blockbehandlung effizienter und wiederum für den Benutzer transparenter: Man verfügt über einen Puffer, genannt «Hold», von beliebiger Grösse soweit es der im RAM vorhandene Text jeweils zulässt. Der Befehl H im Command-Modus bewirkt das Einlagern des dem Cursor nachfolgenden

Textes in den Puffer; der Puffertext geht bis zur nächsten gesetzten Marke oder, wenn keine folgt, bis zum Textende; hn lagert n Zeilen ein (bei kleinen Blöcken sind die Zeilen rasch gezählt, das Setzen einer Marke und etliche Cursorbewegungen bleibt einem erspart!). Soll der Text kopiert werden, so befiehlt man h/u, wodurch der eingelagerte Block unmittelbar anschliessend wieder eingesetzt wird, im Puffer aber bestehen bleibt. Der Befehl u allein kopiert den Block an der neuen Cursorstelle. Das kann beliebig oft wiederholt werden (z.B. schreibt 25u den Pufferinhalt 25 Mal nacheinander in den Text), solange nicht der Puffer mit h0 (null) ausdrücklich entleert wird. Der Pufferinhalt bleibt auch erhalten, wenn man den übrigen Text abspeichert, den Arbeitsspeicher leert und einen neuen Text einliest; das ist sehr praktisch zum Transfer von Blöcken von File zu File. Die Sache wäre noch vollkommener, wenn in der Statuslinie stets ein Vermerk vorhanden wäre, solange der Puffer besetzt ist. Auch die Verfügbarkeit von zwei oder drei separaten Puffern wäre manchmal erwünscht; doch würden einem zuviele davon wieder die Übersicht nehmen.

Ich habe insgesamt den Eindruck, dass die Befehle beim Spellbinder geschickter gruppiert wurden. Der Wechsel zwischen Edit- und Command-Modus teilt zunächst alle Befehle in zwei Gruppen, eine kleinere von direkt verfügbaren Eintasten-Editierbefehlen, und eine grössere, ihrerseits unterteilte, von spezielleren Befehlen. Beim WordStar sind 26 Befehle direkt (Buchstabe mit ctrl-Taste zusammen) verfügbar; fünf davon erschliessen weitere Befehlslisten, und dazu kommen noch spezielle Editierbefehle für die Antworten und Optionen. Spellbinder kommt mit insgesamt rund 10 Befehlen weniger aus; auch sind praktisch alle Befehle mit sinnreichen Buchstaben auszuführen. Im WordStar gibt es nahezu 100 Zweibuchstabenbefehle, im Spellbinder nur 20. Beim WordStar finde ich psychologisch besonders ungeschickt, dass es eine ganze Reihe von Befehlszeichen gibt, welche je nach gültigem Modus recht unterschiedliche Bedeutungen haben: z.B. haben mehrere Buchstaben 5, 6 oder gar 7 Bedeutungen, je nach vorausgegangenem Zeichen, und sie sind nur zum Teil mnemonisch sinnvoll. Zwar ist der gültige

Modus jeweils auf der obersten Zeile (Statuszeile) angezeigt; doch können einem die WordStar-Befehle leicht durcheinandergeraten, wenn man nicht regelmässig mit dem Programm arbeitet. Bei beiden Programmen steht in der Statuszeile auch immer die gegenwärtige Zeilen- und Spaltennummer des Cursors, beim WordStar zusätzlich die Seitennummer.

### Weitere Funktionen

Nicht grundlegend unterscheiden sich die Funktionen Tabulator (im Spellbinder fehlen ein Rückwärts- und Dezimaltabulator) und Search-and-replace (im Spellbinder fehlen einige Optionen und der sehr praktische Befehl ctrl-L des WordStar: wiederhole die Suche mit dem letzten Such-(und-Ersetze-)Begriff. Bei beiden Funktionen bietet WordStar etwas mehr Komfort.

Mehr ins Gewicht fallen im täglichen Gebrauch die Unterschiede in der Einfüpfungsfunktion: Im WordStar ist die klassische «Insert»-Funktion implementiert, d.h. jedes eingefügte Zeichen schiebt den nachfolgenden Text ein Zeichen weiter. Im Spellbinder wird eine Textlücke geöffnet («Expand»), in welche man seine Ergänzung hineinschreiben kann. Die Lücke wächst nach Bedarf, doch ist auf der letzten Bildschirmzeile stets der Anschluss text sichtbar. Bei den Editierbefehlen bestehen einige Einschränkungen. Die geöffnete Lücke ist «einladender» als das Vor-sich-her-Schieben des Textes.

Wichtig sind auch die Unterschiede im Umgang mit dem Massenspeicher. WordStar bringt den Vorteil, dass man zusätzlich zum bearbeiteten File je ein weiteres Read- und Writefile öffnen kann; da ist Spellbinder beschränkter. Aber genau genommen gibt es im WordStar eigentlich gar keine Disk-Befehle für das File in Bearbeitung, da ja der Floppy wie ein erweitertes Memory behandelt wird. Vielmehr gibt es mehrere Befehle zum Beenden des Editierens: mit oder ohne Aufbewahren des Textes und unter Verbleiben im WordStar, mit dem alten oder einem neuen File, oder Ausgang ins CP/M. Das hat zur Folge, dass ein Diskettenwechsel während des Editierens nicht in Frage kommt. Man muss ins No-File-Menu oder ins CP/M aussteigen. Im Spellbinder kann man immer Diskwechsel vornehmen, solange write-file offen ist, was

zur Uebernahme von Textteilen aus anderen Files oft recht praktisch ist. Man kann im Spellbinder auch einen im Arbeitsspeicher vorhandenen Text nacheinander auf mehrere Disketten schreiben: praktisch für Sicherheitskopien ohne PIP.

Aus beiden Programmen kann man das Directory der Diskette abrufen, beim WordStar allerdings nur zeilenweise. Beim Spellbinder ist es ganz und zudem mit Angaben über den Umfang der Files auf der Diskette einschliesslich des Totals versehen, während man im WordStar dafür eigens das CP/M-STAT.COMmand abrufen muss. Die von beiden Programmen automatisch erstellten Backup-Files kann Spellbinder auch direkt wieder lesen, im WordStar muss man sie erst umbenennen. Andererseits hat WordStar eine Einrichtung, die das Abrufen von Drittprogrammen direkt erlaubt, sofern das Zurückkommen in den WordStar geregelt ist; das fehlt im Spellbinder. WordStar gestattet auch das direkte Löschen, Kopieren und Umbenennen von Files auf der Diskette, Spellbinder nur das Löschen.

Überlange Files, d.h. Texte, die den Umfang des Arbeitsspeichers übersteigen, werden von beiden Program-

Der vorliegende Vergleich bezieht sich auf Spellbinder (Lexisoft Inc. in Davis, Ca.) Version 5.03 und WordStar (Micropro Inc., San Rafael, Ca.) Version 2.10, beide aus dem Jahr 1981. Für beide Programme sind inzwischen neue Versionen erschienen. Beispielsweise bietet der neue Spellbinder nun den erwünschten Komfort bezüglich Suchbefehlen, Dezimaltabulator und Filenamens-Repetition; ferner bringt er eine stark verbesserte Trennhilfe und eine elegante Lösung des Umlaute-Druck-Problems. WordStar bringt ebenfalls eine bessere Trennhilfe sowie die Möglichkeit, Text horizontal zu verschieben (z.B. eine ganze Spalte in einer Tabelle zu verlegen; Spellbinder hat dafür einen Macro); die kritisierte unschöne Blocksatz-Proportionaldruckweise kann nun wahlweise umgeschaltet werden auf die schönere Verteilung des Leerraums in die Wortzwischenräume.

men geschickt behandelt. Theoretisch sind Files bis zur Länge der Diskkapazität möglich; praktisch wird man in der Regel besser mehrere kürzere Stücke separat bearbeiten und zum Druck zusammenhängen. Eigentlich müsste man hier den Vorteil des diskorientierten WordStar erwarten. Das Handbuch weist aber mit Recht auf eine Reihe von Risiken und Gefahren hin. Beide Programme speichern in Form von «sequential files». Beim WordStar ist das nicht evident, weil das File-Management automatisch erfolgt; dem Benutzer wird ein «random access file» vorgemacht. Beim Spellbinder muss der Benutzer alle wesentlichen Diskaktivitäten selber auslösen, so dass er stets genötigt ist, den Überblick zu behalten. Es gibt den praktischen «Global»-Befehl, der bei überlangen Files alles Nötige regelt. Wünschbar wäre eine Einrichtung zum Repetieren von Filenamen, wie sie WordStar hat.

## Besondere Merkmale

CP/M-Standard-files enthalten in Zeilen geordneten Text; nach jeder Zeile folgt CRLF (Carriage Return, Line Feed). In Textverarbeitungsprogrammen ist das unbequem, weil man zum Umformatieren zwischen «harten» Zeilenenden, d.h. den vom Schreiber gesetzten Abschnittenden, und «weichen» Zeilenenden unterscheiden muss, die vom Programm eingesetzt und nach Bedarf wieder entfernt werden. WordStar und Spellbinder lösen das Problem nicht auf gleiche Weise. Um das «weiche» Zeilenende zu markieren, setzt WordStar das 7. Bit des letzten Buchstabens, während Spellbinder den letzten

Wortzwischenraum jeder Zeile mit besonderen Zeichen (0E oder 0C statt 20hex) füllt.

In Assembler-Programmen oder in Sprachen wie Basic, Pascal, C usw. muss aber die CP/M-Standardform eingehalten werden. Beide Textverarbeiter sind sehr viel praktischer für das Schreiben von Programmen als die speziellen Editoren. Im WordStar wählt man vor dem Editieren «Non-Document»; leider scheint es unmöglich, nachträglich zwischen den beiden Filetypen zu wechseln. Im Spellbinder wählt man erst vor dem Abspeichern, indem man dem Filenamen «/1» anfügt; das setzt die nötigen CRLFs ein.

Das Handbuch finde ich beim WordStar überlang und unnötig repetitiv. Wie die Editier- und Druckbefehle könnte es geschickter organisiert sein. Das Spellbinder-Handbuch ist wesentlich kürzer und durchaus klar. Es wird zudem begleitet von einem «Tutorial», eine vom Einfachen zum Komplizierteren fortschreitende Anleitung für den Beginner, mit dem sich ein englisch lesender durchschnittlich intelligenter Schreibmaschinenschreiber ohne weiteres selber in die Geheimnisse der elektronischen Textverarbeitung einweihen können sollte.

Der wirkliche «Hit» von Spellbinder liegt aber im Autocommand - einer Einrichtung, der im WordStar nichts Vergleichbares entspricht. Praktisch jeder Befehl kann nämlich mit mehreren ändern zu einer Art Mikro-Programmen zusammengestellt werden, indem man sie bis zu einer Zeilenlänge einfach aneinanderreihet. Sie werden nacheinander ausgeführt, natürlich unter Berücksichtigung der Koef-

fizienten. Also Ad-hoc-Programmierung, die viel Zeit erspart und das Denken herausfordert! Beispielsweise will ich aus einer langen Namensliste alle Zeilen, die einen bestimmten Code (z.B. «\*PSY») aufweisen herausdrucken. Der Autocommand sieht dann so aus: «999s/\*PSY//b0/pl», und bedeutet: 999 Mal (d.h. sehr viel) - suche «\*PSY» und nimm heraus - spring auf Zeilenanfang - drucke eine Zeile - beginn den nächsten Such.

M-Speak, die MACRO-Programmiersprache in Spellbinder ist übrigens nichts anderes als eine Erweiterung der Autocommand-Einrichtung auf beliebig lange und speicherbare MACROs, ergänzt mit Dingen wie Variablen, Bedingungstests und Sprungbefehlen usw.

Hier tritt der psychologische Vorteil von Spellbinder besonders deutlich hervor. Es ist die wichtigste Erkenntnis der modernen Arbeitspsychologie, dass Verfügbarkeit, Durchschaubarkeit und Beeinflussbarkeit von Arbeitsvorgängen die entscheidende Waffe sind gegen Stress am Arbeitsplatz (vgl. etwa E. Ulich: Möglichkeiten autonomieorientierter Arbeitsplatzgestaltung. Kap. 6. in: M. Frese (Hrsg.): Stress im Büro. Bern, Huber, 1981).

Die vorstehende ausführliche Beschreibung macht deutlich, dass WordStar und Spellbinder ganz vorzügliche Textverarbeitungsprogramme sind. Im Vergleich mit dedizierten Textverarbeitern (IBM, WANG, CTC, Philips usw.) würden sie nur wenig zurückfallen. Spellbinder ziehe ich gegenüber WordStar vor, weil die Vorgänge für den Benutzer transparenter und verfügbarer sind.

## ALTOS Serie 5

Multi User Floppydisk und Harddiskysteme

ACS 5-15D	2 MBy Floppy Disk 3 User, 192 KBy Memory	US\$ 4300.—
ACS 5-5D	1 MBy Floppy Disk 3 User, 192 KBy Memory 5 MBy Harddisk	US\$ 7300.—
ACS 1	Bildschirmterminal grüne Schrift, separate Tastatur, Split Screen usw.	US\$ 1100.—
MP/M II	gratis zu jedem System	

Ringler Computer Systeme

Gotthardstr. 36 8800 Thalwil Tel. 01 - 720 64 50

**Die Spezialzeitschrift für Commodore-Benutzer**  
**CBM**  
**PET**  
**NEWS**

befasst sich herstellerunabhängig mit allen auftauchenden Fragen der Commodore-Computer (PET 2001, CBM 3000, 4000, 8000 und neu VC 20) und wird dadurch für unentbehrlichen Pflichtlektüre für jeden ernsthaften CBM/PET-Benutzer. Die CBM/PET NEWS sind eine Fachzeitschrift, die im Gegensatz zu den unzähligen Fachbüchern unverzüglich auf neue Erkenntnisse eingeht und die Anregungen aus einer grossen Leserschaft sofort weitergibt. Verlangen Sie kostenlos ein Ansichtsexemplar bei Mikro+Kleincomputer Informa Verlag AG,  
 Postfach 1401, CH-6000 Luzern 15

**Mit dem Sharp PC-1500 stecken Sie so ziemlich alles in die Tasche:** dieser zeichnerisch hochbegabte und Basic-sprechende Kleincomputer kann sehr viel, was sonst nur die Grossen können. Geschäftsleuten ist er Kalkulator und Kontrolleur, Studenten eine universelle Hilfe, Managern Analysator und Agenda, Ingenieuren Hilfe zur Lösung mathematischer und mechanischer Berechnungen. Die Speicherkapazität liegt bei 16 Kbyte ROM und 3,5 Kbyte RAM. Sie kann durch den Einsatz von Speichermoduln um bis 8 Kbyte erweitert werden. Formeln und Befehle, die häufig eingegeben werden müssen, können direkt über Reservetasten abgerufen werden. 17 Tasten lassen sich als Programmnamen verwenden. Der Druck nach oben und unten, nach links und rechts erfolgen.

**Zur gleichen Familie der logischen**

**Denker** im Taschenformat gehört der Basic-Computer PC-1211. Grösser und damit noch vielseitiger sind die Tischmodelle von Sharp: Bürocomputer für kleinere und mittlere Betriebe, Techniker und Wissenschaftler, aber auch für Vereine. Und nicht zuletzt fürs Hobby. Grösster von allen ist der Facit 6520. Wir sagen Ihnen gern, welcher für Sie der ideale ist.

- Bitte senden Sie mir Ihre Dokumentation über den
- Taschencomputer PC-1500 und PC-1211
  - Bürocomputer MZ-80,  PC-3201
  - Facit 6520  m+k 82-4

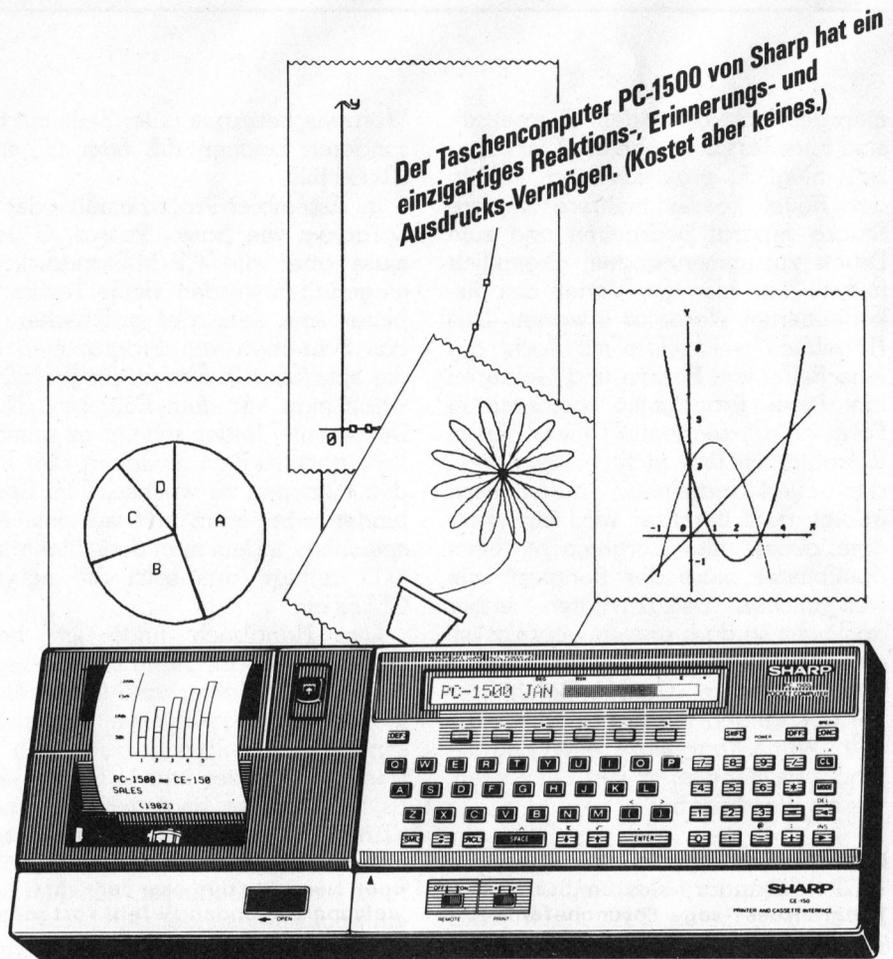
Sachbearbeiter \_\_\_\_\_

Firma \_\_\_\_\_

Strasse \_\_\_\_\_

PLZ, Ort \_\_\_\_\_

P.S. Was von Facit und Sharp kommt, kann auch gemietet oder geleast werden.



**Was hier schwarz erscheint, drucken Sharp-Taschencomputer auch rot, blau und grün auf Weiss.**

**FACIT SHARP**

Sharp-Büromaschinen-Generalvertretung Schweiz/Liechtenstein. Mit Filialen in Bern, Lausanne und Genf. Und mit vielen Fachhändlern.

Facit-Addo AG  
Badenerstrasse 587, 8048 Zürich  
Telefon 01/52 58 76

Battis und Rüeegg BSW

# Lehrgänge



## Programmieren mit FORTRAN VI (2)

Oliver ROSENBAUM, dipl. Ing. FH

### 2. Teil

In dieser zweiten Folge unserer Einführung in die FORTRAN-Programmierung erfahren Sie mehr über Steuer- und Formatanweisungen sowie über das Programmieren von Programmschleifen.

### 5. STEUERANWEISUNGEN UND ANWEISUNGEN FUER EIN- UND AUSGABE

Programme werden in der Reihenfolge der Zeilen bzw. Anweisungen ausgeführt. Wünscht man einen anderen Ablauf, z.B. Wiederholung einzelner Programmteile mit evtl. geänderten Variablenwerten, muss man sich besonderer Steueranweisungen bedienen. Diese Steueranweisungen brechen den sequentiellen Programmablauf an der Stelle, an der sie stehen, ab und verzweigen das Programm zu einer anderen Stelle des Programmes (sog. Sprungbefehle).

#### 5.1. SPRUNGBEFEHL

Der einfachste Sprungbefehl ist der GOTO-Befehl. Ihm muss eine Anweisungsnummer folgen, damit der Steuerung des Programmes die Stelle mitgeteilt wird, an der mit der Programmabarbeitung fortgefahren werden soll.

```
GO TO 100
...
100 A = X + Y
```

Trifft das Programm auf eine GOTO-Anweisung, so überspringt es die folgenden Programmzeilen bis zu der Anweisungsnummer, die hinter dem GOTO-Befehl angegeben wurde (hier 100). Die Anweisung mit der entsprechenden Nummer kann, wie im Beispiel hinter dem GOTO-Befehl stehen, aber auch davor. In diesem Fall springt das Programm zurück, aber immer genau auf die Zeile mit der entsprechenden Nummer.

Taucht diese Anweisungsnummer an keiner Stelle im Programm auf, so wird das Programm mit einer Fehlermeldung abgebrochen. Diese Art der GOTO-Anweisung nennt man die "unbedingte GOTO-Anweisung", weil hier keine Alternativen möglich sind. Das GOTO kann in einem oder in zwei Worten geschrieben werden.

Neben diesem GOTO gibt es noch den "berechneten" oder "bedingten" GOTO Befehl, der es ermöglicht, in Abhängigkeit der Grösse einer Variablen zu bestimmten Adressen zu springen.

```
GO TO (100, 800, 50) A
```

In diesem Beispiel wird dann bei der Adresse 100 fortgefahren, wenn die Variable den Wert 1 hat, bei 800 wenn  $A = 2$  und bei  $A = 3$  fährt das Programm mit Zeile 50 fort. Ist der Wert der Variablen weder 1, 2 oder 3 so wird mit dem der GOTO Anweisung folgenden Befehl fortgefahren, also mit der nächsten Zeile. Die Anzahl der angegebenen Adressen ist beliebig, jedoch wird beim Wert der Variablen = 1 immer zur ersten, bei 2 immer zur zweiten Adresse gesprungen usw.

Beispiel:

```
INTEGER X
2 X = 3
GO TO (8,2,6,1) X
A = A + 1
6 X = 1
1 GO TO 2
8 ...
```

Da in der zweiten Zeile (hier mit der Adresse 2),  $X = 3$  gesetzt wird, werden die Zeilen nach dem GOTO-Befehl übersprungen bis zu der Zeile mit der Anweisungsnummer 6, also der dritten Adresse in den Klammern des GOTO-Befehls. Hätte X z.B. den Wert 9, so würde nach GOTO mit der Zeile  $A = A + 1$  fortgefahren.

Eine andere Art der bedingten Sprunganweisung ist die IF-Anweisung. Sie hat die Form

IF (arithmetischer Ausdruck) a,b,c

a, b und c sind Adressen die in Abhängigkeit des arithmetischen Ausdruckes angesprungen werden.

Hat der arithmetische Ausdruck ein negatives Resultat, so wird

nach a gesprungen, ist das Resultat = 0, so wird mit der Anweisungsnummer b fortgefahren, bei einem positiven Ergebnis fährt das Programm bei c fort.

Der arithmetische Ausdruck kann entweder aus einer oder mehreren Variablen verknüpft durch mathematische Operatoren bestehen. Hier können im Gegensatz zum berechneten GOTO-Befehl nur drei Adressen angegeben werden, da es nur die drei Zustände gibt:

- kleiner Null
- gleich Null
- grösser Null

Beispiel 1:

```
REAL I, K
I = 3.0
K = 4.0
IF (K - I) 60,100,80
```

Da  $K = 4,0$  und  $I = 3,0$  ist, wird das Ergebnis des Ausdruckes  $K - I = 1,0$  also positiv: Das Programm fährt mit der Anweisungsnummer 80 fort.

Beispiel 2:

```
INTEGER X
X = 0
7 X = X + 1
IF (X - 10) 7,7,8
8 ...
```

Zunächst wird X als INTEGER vereinbart (ohne Vereinbarung würde der Variablenname X eine REAL-Variable ausweisen).

In der zweiten Programmzeile wird X der Wert 0 zugewiesen, was nicht automatisch der Fall ist. Zeile drei besagt, dass sich der Wert von X und 1 erhöhen soll, in diesem Fall also auf  $0 + 1 = 1$ .

Zeile vier:  $X - 10 = 1 - 10 = 9$  also negativ. Das Programm fährt daraufhin mit der Anweisungsnummer

7 fort. Die Anweisung 7 erhöht die Variable X wieder um 1. In X steht nun der Wert 2. Wieder die Abfrage  $X - 10 = 2 - 10 = -8$ , also wiederum zur Anweisung 7. Dies geschieht solange, bis X den Wert 11 hat, erst dann erzeugt der Them in der Klammer einen positiven Ausdruck.

Eine andere Form der IF-Anweisung ist die logische IF-Abfrage:

IF (.vergleichende logische Operation.) Anweisung

Die vergleichenden logischen Operationen wurden unter 2.1. beschrieben. Ist der logische Vergleich in der Klammer wahr, so wird die Anweisung die hinter der Klammer, in der gleichen Zeile der IF-Abfrage steht, ausgeführt.

Führt der logische Vergleich zu einem "falsch", so wird die Anweisung ignoriert, und es wird mit der nächsten Zeile fortgefahren.

Beispiel 3:

```
I = 5
J = I + 1
IF (J.EQ.4) GO TO 100
...
```

Da  $J = 6$  und nicht gleich vier ist (.EQ. = equal) wird nicht nach 100 gesprungen, sondern mit der nächsten Programmzeile fortgefahren.

Beispiel 4:

```
A = 7
B = A + 1
IF (B.GT.A) A = 0
...
```

Nach diesen drei Zeilen enthält A den Wert 0 und B den Wert 8. Da B grösser (.GT. = greater than) ist als A in Zeile 3 (A zunächst noch 7), wird die Anweisung hinter der Klammer, also  $A = 0$ , ausgeführt. Erst danach führt das Programm die nächsten Zeilen aus.

## 5.2. EINGABE- UND AUSGABE-ANWEISUNGEN

Diese Anweisungen werden im folgenden als E/A-Anweisungen bezeichnet.

Sollen Werte von peripheren Geräten oder Datenträgern in den Kernspeicher gelesen werden, wo sie dann dem Programm direkt zur Verfügung stehen in Form von z.B. Variablen bedient man sich der READ-Anweisung:

READ (a,b,END=c, ERR=d) Liste von Variablen

a ist entweder eine Konstante oder eine Variable. Der Zahlenwert von a ist der Code des Gerätes, von dem eingelesen werden soll. Bei manchen Rechnern haben die Geräte wie Kartenleser, Plattenlaufwerk, Terminal usw. feste Gerätenummern, sie sind den Handbüchern dieser Anlagen zu entnehmen. Anderen Rechnern muss man vor Beginn des Programms mitteilen, welche Geräte welche Nummern erhalten sollen. Unter diesen Nummern sind die Geräte dann im Programm ansprechbar. Auch diese Information findet sich in dem Handbuch des jeweiligen Rechners.

b ist eine Anweisungsnummer. In der Anweisung mit dem Label b muss eine Vereinbarung über das Format gemacht werden, in dem die Daten in diesem Falle (READ) eingelesen werden sollen (siehe auch das Kapitel über FORMAT-Anweisung).

c ist ebenfalls eine Anweisungsnummer mit der fortgefahren werden soll, wenn das Einlesen der Daten beendet ist. Soll mit der nächsten Zeile des Programms fortgefahren werden, so kann  $END=c$  entfallen.

d bezeichnet die Anweisungsnummer zu der das Programm im Fehlerfall springt. Auch sie kann entfallen, wenn dies nicht gewünscht wird, es ist aber zu beachten, dass wenn keine Angabe  $ERR=d$  gemacht wird, das Programm unter Umständen bei einem Fehler abgebrochen wird und das Betriebssystem eine Fehlermeldung ausgibt.

Die Variablenliste bezeichnet die Variablen, in denen die eingelesenen Werte gespeichert werden sollen. Die eingelesenen Werte sind dann innerhalb des Programmes unter diesen Namen zu finden (vgl. auch Kapitel: COMMON-Anweisung).

Beispiel:

```
READ (5,100) A, B, C
```

Hier werden vom Gerät mit der Nummer 5 drei Werte gelesen und in den Variablen A, B und C abgespeichert. In der Zeile mit der Anweisungsnummer 100 (also nicht die 100. Zeile) muss eine FORMAT-Anweisung stehen.

Das Gerät, von dem gelesen werden soll, im Beispiel die Nr. 5, muss ein Gerät sein, von dem gelesen werden kann. Der Drucker kann z.B. nur Information vom Computer auf

das Papier bringen, nicht umgekehrt, er ist also ein Ausgabegerät. Der READ-Befehl kann sich nur auf Eingabegeräte beziehen.

Eingabegeräte:

- Kartenleser
- Konsolschreibmaschine
- Tastatur
- Terminal
- Lochstreifenleser
- Magnetplatte
- Floppy-Laufwerke
- Magnetband-Laufwerke
- Digitizer

Ausgabegeräte:

- Kartenstanzer
- Konsolschreibmaschine
- Bildschirm
- Terminal
- Lochstreifenstanzer
- Magnetplatte
- Floppy-Laufwerke
- Magnetband-Laufwerke
- Plotter
- Drucker

Mit der READ-Anweisung hat man also Zugriffsmöglichkeiten zu den Eingabegeräten. Um Daten auch ausgeben zu können, benötigt man die WRITE-Anweisung. Mit ihr liest man Daten aus dem Kernspeicher des Rechners und gibt sie an ein peripheres Gerät (hier nur Ausgabegerät) weiter.

WRITE (a,b,ERR=d)  
Liste von Variablen

Diese Anweisung hat also die selben Elemente wie die READ-Anweisung mit einer Ausnahme, es fehlt eine  $END=c$  Angabe. Diese ist hier nicht von Bedeutung, da die Variablen, die ausgegeben werden sollen, im Programm belegt werden mussten und daher die Anzahl der Variablen nur endlich ist. Zu beachten ist: a darf nur die Nummer eines Ausgabegerätes sein.

Für Datenübertragung vom und zum Bildschirmterminal gibt es zwei besondere Befehle: ACCEPT und DISPLAY.

ACCEPT (a,b,LINE=c, POSITION=d, ERASE, PROMPT, ERR=e) Variable

a und b sind wie vor beschrieben, die Gerätenummer (hier die des Bildschirmterminals) und die Anweisungsnummer der Zeile, welche die FORMAT-Anweisung enthält.

c ist die Nummer der Zeile auf dem Bildschirm, von wo der einzuge-

bende Wert angenommen werden soll. c muss zwischen 1 und der maximal möglichen Anzahl von Zeilen auf dem entsprechenden Bildschirm sein, was von Anlage zu Anlage verschieden ist. Mit d wird die Spalte, in welcher der einzugebende Wert angenommen werden soll, angegeben. d kann also zwischen 1 und der maximalen Anzahl an Spalten des entsprechenden Bildschirms liegen (meist 80).

Mit ERASE wird der Bildschirm vor der Eingabe gelöscht. Wird dies nicht gewünscht, so kann das Wort ERASE einfach weggelassen werden.

PROMPT druckt am Bildschirm Sterne, die genau an der Stelle stehen, wo die Eingabe stattfinden soll. Bei Eingabe von Werten werden die Sterne überschrieben. PROMPT kann ebenfalls weggelassen werden.

ERR=e ermöglicht im Fehlerfall den Sprung zur Adresse e. Diese Angabe kann ebenfalls weggelassen werden.

Beispiel:

```
ACCEPT (8,150, LINE=10,
POSITION=40, PROMPT) A
```

Diese Zeile bewirkt, dass in Zeile 40, Spalte 10 auf dem Bildschirm (vorausgesetzt dieser hat die Geräte Nummer 8) auf die Eingabe eines Wertes gewartet wird. Die Stelle, wo der Wert erwartet wird, ist mit einem Stern gekennzeichnet. Das Programm wird solange unterbrochen, bis ein Wert angegeben wurde. Dieser ist dann im Programm über die Variable A abrufbar. In der Anweisung 150 steht eine FORMAT-Anweisung, die bestimmt, welchen Typ und Länge die Variable haben muss.

Die Ausgabe über Bildschirm erfolgt mit dem Befehl:

```
DISPLAY (a,b,LINE=c, POSITION=d,
ERASE, ERR=e) Variable
```

Die einzelnen Elemente dieser Anweisung sind äquivalent zu denen des ACCEPT-Befehles. PROMPT ist hier nicht möglich und auch nicht sinnvoll. ERASE und ERR können wieder wahlweise weggelassen werden. Auch die Reihenfolge der Parameter ist beliebig, mit Ausnahme von a und b. Die Geräte- und die Anweisungsnummer der FORMAT-Anweisung müssen immer die erste und zweite Stelle in der Klammer belegen.

Beispiel:

```
DISPLAY (8,899, POSITION=1,
LINE=1, ERASE) A
```

Der Wert der Variablen A wird im Format der FORMAT-Anweisung 899 auf dem Gerät Nr. 8 (hier Bildschirm) in der oberen linken Ecke ausgegeben. Vor Ausgabe wird der Bildschirminhalt gelöscht, d.h. etwa vorher vorhandene Zeichen auf dem Bildschirm werden gelöscht. Hat der Wert mehr als eine Stelle, so bezeichnet POSITION und LINE die Stelle, wo das erste Zeichen des Wertes steht.

Beispiel:

Hat im obigen Beispiel die Variable den Zahlenwert 5000 und soll sie in die rechte obere Ecke des 80-Spalten breiten Bildschirms gedruckt werden, so wird LINE=1 angegeben. Die POSITION errechnet sich aus der Anzahl der Spalten am Bildschirm vermindert um die Anzahl der Zeichen aus der die Zahl besteht plus 1.  $80 - 4 + 1 = 77$  Im DISPLAY-Befehl steht also POSITION=77.

```
Spalte: ... 74 75 76 77 78 79 80
           5  0  0  0
Zeile 1
      2
      3
      4
      .
      .
```

Bild 1

## 6. DIE FORMAT-ANWEISUNG

Wie schon im Kapitel für die Ein- und Ausgabe von Daten auf, bzw. von peripheren Geräten beschrieben, müssen die Daten, die von einem Speichermedium zum anderen transportiert werden sollen, formatiert werden, d.h. es muss eine Vereinbarung getroffen werden über Länge, Art und Anzahl der Variablen. Diese Aufgabe übernimmt die FORMAT-Anweisung. Jede FORMAT-Anweisung muss eine Anweisungsnummer haben, damit sich die E/A-Befehle auf dieses FORMAT beziehen können.

a FORMAT (b, c, d, ...)

Mit a ist die Anweisungsnummer gemeint, sie wird beliebig gewählt, auch die Reihenfolge der Anweisungen im Programm sind unabhängig von ihr. Auf eine FORMAT-Anweisung können sich beliebig viele E/A-Befehle beziehen. Die FORMAT-Anweisung kann an beliebiger Stelle im Programm stehen.

b, c, d usw. sind sogenannte Feldspezifikationen. Durch sie wird die eigentliche Formatierung vorgenommen. Es können beliebig viele Feldspezifikationen in einer Formatanweisung stehen.

### 6.1. I-FELD-SPEZIFIKATION

Diese Feldspezifikation bezeichnet einen Wert vom Typus INTEGER. Sollen also Daten vom Typ INTEGER ausgegeben oder eingelesen werden, so muss sie in der FORMAT-Anweisung stehen, auf die sich die E/A-Anweisung bezieht.

Die I-Feld Spezifikation hat folgende Form in einer FORMAT-Anweisung:

a FORMAT (Iw)

a ist wieder die Anweisungsnummer und das I in der Klammer bedeutet INTEGER. Die Variable hinter der E/A-Anweisung ist also vom Typ ganzzahlig.

w bezeichnet die maximale Anzahl der Stellen dieser Variablen. (I 3) bedeutet also eine dreistellige INTEGER Zahl.

### 6.2. F-FELD-SPEZIFIKATION

F bedeutet in diesem Zusammenhang Floating-Point, also Fließkomma. Diese Spezifikation bedingt also REAL-Zahlen.

a FORMAT (F w.d)

w ist wieder wie beim I-Format die maximale Gesamtzahl der Stellen inklusive des Vorzeichens und hier des Dezimalpunktes.

d besagt, wieviele Stellen davon hinter dem Komma stehen können. Eine Zahl muss nicht alle möglichen Stellen ausfüllen, sie kann auch kleiner sein.

Beispiel 1:

```
75 FORMAT (F 10.3)
```

Diese Formatierung gibt eine REAL-Zahl mit insgesamt maximal 10 Stellen (oder weniger) an:

3 Stellen hinter dem Dezimalpunkt  
1 Stelle für den Dezimalpunkt selbst  
6 Stellen vor dem Dezimalpunkt inklusive Vorzeichen

Beispiel 2:

```
109 FORMAT (F 4.0)
```

Diese REAL-Zahl hat maximal 4 Stellen, davon:  
0 Stellen hinter dem Dezimalpunkt  
1 Stelle für den Dezimalpunkt selbst  
3 Stellen vor dem Dezimalpunkt inklusive Vorzeichen

Für die F-Feld-Spezifikation aus Beispiel 1 (F 10.3) werden einige Werte als Beispiel aufgeführt:

Wert der Variablen	Ausgabe
500.23	500.230
-1234.123456	-1234.123
6000000.05	???????????
.5	0.500
- 16.0	-16.000

Es ist also zu beachten:

- zuviele Nachkommastellen werden abgeschnitten
- die angegebenen Nachkommastellen werden immer bis zur angegebenen Länge mit Nullen aufgefüllt, falls keine oder weniger Nachkommastellen vorhanden sind als in der FORMAT-Anweisung angegeben worden sind
- bei zuvielen Vorkommastellen wird eine Fehlermeldung in Form von Fragezeichen ausgegeben
- das Vorzeichen bleibt erhalten, es ist bei der Länge der Zahl mit zu berücksichtigen

### 6.3. E-FELD SPEZIFIKATION

Sie hat die Form:

a FORMAT (E w.d)

E steht hier für Exponent. Zahlen werden also hier in exponentieller Form dargestellt.

w ist wieder die Gesamtstellenanzahl und

d stellt die Anzahl der Stellen hinter dem Dezimalpunkt dar. Bei der Wahl der Gesamtstellenanzahl ist zu berücksichtigen, dass ausser den gewünschten Nachkommastellen noch mindestens sechs Stellen zusätzlich zur Verfügung stehen müssen, da der Exponent, der Dezimalpunkt und gegebenenfalls das Vorzeichen Platz finden müssen. Es ist also immer für w eine Zahl grösser oder gleich d + 7 zu wählen.

Beispiel bei einer Spezifikation von (E 10.3):

Wert der Variablen	Ausgabe
0.00000000005	0.500E-10
6666.0	0.666E 04
- 33.459	-0.334E 02

Die Schreibweise bedeutet:

0,5 wird multipliziert mit  
10 hoch -10 = 0,00000000005  
0,666 wird multipliziert mit  
10 hoch 4 = 6660,0  
-0,334 wird multipliziert mit  
10 hoch 2 = - 33,400

Bei der Rückwärtsrechnung erkennt man die Rechengenauigkeit.

### 6.4. Gemischte Feld-Spezifikationen

Sollen z.B. in einem WRITE oder DISPLAY-Befehl mehrere Variablen unterschiedlichen Datentyps ausgegeben werden, so muss in der zu der E/A-Anweisung gehörenden FORMAT-Anweisung für jeden Variablentyp eine Formatierung vorgenommen werden.

Beispiel 1:

```
INTEGER I
REAL Z, X
WRITE (10,16) I, Z, X
16 FORMAT (I3, F5.2, E10.2)
...
```

In der WRITE-Anweisung stehen mehrere Variablen unterschiedlichen Datentyps. Die E/A-Anweisung kann sich immer nur auf eine FORMAT-Anweisung beziehen, also müssen die unterschiedlichen Feld-Spezifikationen hier untergebracht werden. Dies geschieht durch Trennung mit Komma. Auf diese Weise ist es möglich, beliebig viele Feld-Spezifikationen in einer einzigen FORMAT-Anweisung unterzubringen. Die Reihenfolge der Spezifikationen im FORMAT muss mit der Reihenfolge der entsprechenden Variablen in der E/A-Anweisung übereinstimmen.

Beispiel 2:

```
READ (12,3) I, J, K
3 FORMAT (3I2)
```

In diesem Beispiel werden alle drei Variablen im Format I2, also INTEGER mit zwei Stellen gelesen. Die Voranstellung einer Konstanten bewirkt somit das mehrmalige Wiederholen der Feld-Spezifikation. Anzahl der Variablen und Grösse des Faktors müssen übereinstimmen!

Beispiel 3:

```
READ (12,4) A, B, C, I, J
4 FORMAT (3F10.2, 2I5)
WRITE (10,4) A, B, C, I, J
```

Hier werden zunächst von Gerät Nr. 12 fünf Werte in die Variablen A, B und C im Format F10.2, dann

zwei weitere Werte in die Variablen I und J im Format I5 gelesen.

Die WRITE-Anweisung gibt auf das Gerät Nr. 10 die Werte der Variablen im gleichen Format aus, in dem sie eingelesen wurden. Anzahl und Reihenfolge müssen in der E/A-Anweisung und in der FORMAT-Anweisung übereinstimmen.

Beispiel 4:

```
READ (12,7) A, B, C
READ (12,8) I, J
WRITE (10,7) A, B, C
WRITE (10,8) I, J
8 FORMAT (2I5)
7 FORMAT (3F10.2)
```

Hier wird das gleiche erreicht wie in Beispiel 3. Es soll noch einmal gezeigt werden, dass

- es gleichgültig ist, an welcher Stelle die FORMAT-Anweisung im Programm steht
- auf eine FORMAT-Anweisung sich beliebig viele E/A-Anweisungen beziehen können.

### 6.5. TEXT-FELD-SPEZIFIKATIONEN

Bisher wurden nur Zahlenwerte aus Variablen ausgegeben. Mit Hilfe der FORMAT-Anweisung ist es aber auch möglich Texte auszugeben und auch eine Zeilensteuerung des Druckers (bzw. Bildschirms) vorzunehmen.

Ein Text kann auf zwei verschiedene Arten ausgegeben werden:

- mit der H-Feld-Spezifikation (H = Hollerith)
- mit Hilfe des Hochkommazeichens ('')

```
WRITE (10,99)
99 FORMAT (8H computer)
```

8H besagt, dass nun acht Textzeichen folgen. Die Ausgabe wird hier auf Gerät Nr. 10 durch eine WRITE-Anweisung vorgenommen. Hierbei ist keine Variable erforderlich im E/A-Befehl.

```
WRITE (10,98)
99 FORMAT ('computer')
```

Diese beiden Zeilen bewirken das Gleiche: das Wort computer wird auf dem Gerät Nr. 10 z.B. dem Drucker oder dem Bildschirm ausgegeben.

Im folgenden Beispiel sollen Variablenwerte und Text gemischt ausgegeben werden. Das Programmierbeispiel und den Bildschirmausdruck finden Sie in Bild 2.

```

I = 20
J = 16
DISPLAY (4,61, POSITION=3, LINE=2) I, J
61 FORMAT (I2, 9H PKW und , 12, 4H LKW)
    
```

Ausgabe auf dem Bildschirm:

```

.....
20 PKW und 16 LKW.....
.....
    
```

Bild 2 Programm und Bildschirmausgabe

Bei der H-Feld-Spezifikation ist also zu beachten, dass nicht alleine die Zeichen, sondern auch die Leerstellen mitgezählt und auch mitgeschrieben werden in der FORMAT-Anweisung.

Der Buchstabe H hat noch eine andere Bedeutung in der FORMAT-Anweisung: die Druckersteuerung (siehe 6.7).

## 6.6. X-FELD-SPEZIFIKATION

Mit ihr werden Leerstellen, sogenannte Blanks, erzeugt. Die X-Feld-Spezifikation hat die Form:

(aX)

a ist eine ganze Zahl und gibt die Anzahl der Leerstellen an. Beispiel:

```

WRITE (6,6) B
6 FORMAT (10X, F10.5)
    
```

Auf Gerät Nr. 6 (Drucker o.ä.) werden zunächst 10 Leerstellen ausgegeben, dann der Wert der Variablen B im Format F10.5. Man erreicht hier das gleiche Ergebnis wie bei POSITION=10 im DISPLAY-Befehl, der aber nur für den Bildschirm zugelassen ist.

## 6.7. DRUCKERSTEUERZEICHEN

Die Druckersteuerung erfolgt in der FORMAT-Anweisung. Sie kann wie die Feldspezifikationen gemischt mit diesen und mehrfach auftreten und hat die Form:

(lHa)

lH ist konstant und bezeichnet das folgende Zeichen als Drucksteuerzeichen.

Folgende Drucksteuerzeichen können für a gesetzt werden:

Leerzeichen (blank)

einfacher Zeilenvorschub

0 (Null)

doppelter Zeilenvorschub

1 nächster Blattanfang

+ kein Zeilenvorschub, die letzte Zeile wird überschrieben

/ (slash)

einfacher Zeilenvorschub, hierbei kann der Vorsatz lH entfallen, ebenso die trennenden Kommas im Format.

Beispiel:

```

K = 4
WRITE (6,7) K
7 FORMAT
(lH1, lH0 / 100X, 'Seite', l1, lH0)
    
```

Dabei bedeuten im einzelnen:

```

lH1 neuer Blattanfang
lH0 doppelter Zeilenvorschub
/ einfacher Zeilenvorschub
(also 3 Leerzeilen)
100X 100 Leerstellen
'Seite' das Wort Seite wird gedruckt
l1 die Zahl 4 wird gedruckt
lH0 doppelter Zeilenvorschub
    
```

In der rechten oberen Ecke des Blattes steht nun in Zeile 3, mit der 101. Druckposition beginnend, das Wort: Seite 4. Der Druckkopf steht dann in der 6. Zeile des Blattes.

## 7. DIE DO-SCHLEIFE

Wie der Name Schleife schon sagt, ist es mit der DO-Anweisung möglich, eine Programmschleife zu bilden. Diese Programmschleife wird so oft durchlaufen bis bestimmte, vorgegebene Bedingungen erfüllt sind. Mit der DO-Schleife können somit

iterative Prozesse ausgeführt werden. Allgemeine Form:

DO a I=b,c,d

a ist eine Anweisungsnummer, die das Ende der Schleife anzeigt. In der Zeile mit dieser Adresse steht der letzte auszuführende Befehl der Schleife.

I ist die Laufvariable. Bei jedem Durchlauf der Schleife wird sie um den Wert d erhöht. I muss eine INTEGER-Variable sein, die beliebig gewählt werden kann.

b und c sind die untere und obere Grenze der Laufvariablen I. b und c müssen beide grösser Null sein und c grösser als b.

d wird als Schrittweite bezeichnet. Bei jedem Schleifendurchlauf wird I um die Grösse von d erhöht. d muss ganzzahlig und positiv sein und ungleich der Null. Die Angabe von d kann auch weggelassen werden, dann wird automatisch I um eins erhöht pro Durchlauf.

Beispiel 1:

```

A = 5
DO 12 I=1,9,2
12 A = A + 3
...
    
```

Diese Schleife besteht nur aus zwei Zeilen. Die Zeile mit der Adresse 12 ist die letzte ausführbare zu der DO-Schleife gehörende Anweisung.

Die Laufvariable I soll von 1 beginnend mit der Schrittweite 2 solange erhöht werden, bis I=9 ist (= 5 Durchläufe). Im ersten Durchlauf wird zu dem Wert von A der Wert 3 hinzuaddiert und wieder in A abgelegt. Dort steht jetzt die Zahl 8. Beim zweiten Durchlauf der Schleife wird zur 8 wieder der Wert 3 addiert, usw. Nach dem letzten Durchlauf enthält die Variable A den Wert 20, I ist gleich 10, und es wird mit der nächsten Programmzeile fortgefahren.

Beispiel 2:

```

INTEGER SUM
SUM = 0
DO 900 K=10,20
...
...
900 SUM = SUM + K
16 GO TO 20
    
```

In dieser Schleife durchläuft die Variable K die Werte von 10 bis 20 mit der Schrittweite = 1.

Im ersten Durchlauf wird zu der 0 in SUM die 10 hinzuaddiert, beim zweiten zu der nun vorhandenen 10 die 11, im dritten Durchlauf der Schleife wird zu dem Wert 21 die 12 addiert usw. Hat K den Wert 20 erreicht, wird der letzte Durchlauf bis Zeile 900 ausgeführt, dann aber nicht mehr zur DO-Anweisung zurückgesprungen, sondern hier mit Zeile Nr. 16 fortgefahren.

Beispiel 3:

```
DIMENSION A(100)
DO 2, INDEX=1,100
A(INDEX)=0
2 CONTINUE
...
```

In diesem Beispiel sollen die 100 Elemente des Feldes A gleich Null gesetzt werden. INDEX ist hier der Name der Laufvariablen. Hier ist die letzte Anweisung eine sogenannte Leeranweisung: die CONTINUE-Anweisung. Sie bewirkt keine Ausführung von Befehlen. Sie ist aber bei der Programmierung von DO-Schleifen oft sinnvoll,

- da manche Befehle nicht als letzte Anweisung in einer DO-Schleife stehen dürfen. Diese Befehle sind: FORMAT, IF, GOTO DO, PAUSE, STOP, RETURN

- wird die CONTINUE-Anweisung stets als letzte Schleifenanweisung verwandt, so sieht man leichter, wo die Schleife endet. Dies dient nur zur Uebersichtlichkeit der Programme für den Programmierer. Auf den Programmablauf hat dies keine Folgen.

Beispiel 4:

```
DIMENSION ALPHA (100,50)
DO 2 I=1,100
DO 1 J=1,50
ALPHA(I,J) = 0
1 CONTINUE
2 CONTINUE
```

In diesem Beispiel liegt nun eine geschachtelte Schleife vor. Mit ihrer Hilfe kann das Feld ALPHA mit insgesamt 500 Elementen gleich Null gesetzt werden.

Zunächst wird das Feld ALPHA dimensioniert. Die erste DO-Anweisung setzt die Laufvariable I = 1, die zweite DO-Anweisung die Laufvariable J = 1. Das bedeutet, dass in der nun folgenden Zeile ALPHA(1,1) steht, also das erste Feldelement.

Beim Erreichen der Zeile 1 wird zur inneren DO-Schleife zurückgesprungen und J um 1 erhöht. Mit I=1

und J=2 wird nun ALPHA(1,2) = 0 gesetzt. Dies wird solange fortgesetzt bis J=50 ist, darauf wird die innere Schleife beendet und in die nächste Zeile gesprungen, die zweite CONTINUE-Anweisung. Sie ist das Ende der äusseren Schleife und da I den Wert 100 noch nicht erreicht hat, wird wieder zur ersten DO-Anweisung gesprungen und I um 1 erhöht. Nun wird wieder die innere Schleife 50mal durchlaufen usw. Auf diese Weise wird die 100 x 50 Matrix ALPHA zeilenweise gleich Null gesetzt.

Bei der Programmierung der DO-Schleifen ist folgendes zu beachten:

- innerhalb von DO-Schleifen können wieder DO-Schleifen auftreten, siehe Beispiel 4; man spricht hier von geschachtelten Schleifen. Verboten ist das Ueberlappen von Schleifen.

Schachtelung von Schleifen (erlaubt):

```
DO 1 .....
.....
DO 2 .....
.....
2 CONTINUE
1 CONTINUE
```

Ueberlappung von Schleifen (nicht erlaubt!):

```
DO 1 .....
.....
DO 2 .....
.....
1 CONTINUE
2 CONTINUE
```

- Es ist erlaubt, dass sich mehrere DO-Schleifen auf die gleiche Endadresse beziehen. Es handelt sich hier nicht um eine Ueberlappung.

Schachtelung von Schleifen (erlaubt):

```
DO 1 ....
DO 1 ....
.....
1 CONTINUE
```

- Das Herausspringen aus DO-Schleifen ist erlaubt, nicht aber das Hineinspringen!!

Herausspringen (erlaubt):

```
DO 1 ....
.....
GO TO 2
1 CONTINUE
2 .....
```

Hineinspringen (nicht erlaubt!):

```
GO TO 2
.....
DO 1 ....
2 .....
1 CONTINUE
```

- die letzte Anweisung einer DO-Schleife darf nicht sein: FORMAT, DO, IF, GOTO, PAUSE, STOP, RETURN

- der Sprung aus einer DO-Schleife in ein Unterprogramm und die Rückkehr aus diesem in die Schleife ist erlaubt. Es handelt sich hier nicht um ein Hineinspringen.

- bei verschiedenen FORTRAN-Versionen ist es gestattet, die Laufvariable innerhalb der Schleife zu ändern (jeweiliges Compiler-Handbuch beachten)

## Wichtiger Hinweis

für alle Leser, die

Mikro+Kleincomputer

nicht im Abonnement

erhalten!

Dass Sie Mikro+Kleincomputer lesen, zeigt Ihr Interesse an der «Kleincomputerei».

Wenn Sie noch mehr aktuelle Informationen erhalten wollen, geben Sie uns doch ganz einfach Ihre genaue Adresse bekannt. Benützen Sie dazu die in dieser Ausgabe beigeheftete «Info-Karte». Sie erhalten dann wie die Abonnenten von Mikro+Kleincomputer in regelmässigen Abständen kostenlos weiteres Informationsmaterial zugestellt.

Auf bald.

# PPC/HHC - Die Programmierbaren



## Ausgefileter Datentransfer

Peter FISCHER

Der Besitzer des HP-IL Vollsystems sieht sich bei der File-Verwaltung mit nicht wenigen Hemmnissen konfrontiert, wie der obligatorischen Dimensionierung von Datenfiles auf den Speichermedien, der nicht einfachen Uebersicht über die sehr vielfältigen File-Transfer-Befehle oder dem Eingriff in den Hauptspeicher. Im hier vorgestellten, grosszügig ausgelegten Programm sind die entsprechenden Manipulationen weitestgehend automatisiert und alle Eventualitäten abgedeckt. Weil es einen vollständigen Dialog mit dem Benutzer führt, ist es auch für den HP-IL-Neuling ein Einfaches, Uebersicht über seine Files, deren Transfer und über die Speichermedien zu bewahren.

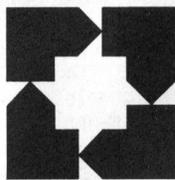
Files sind Dateneinheiten wie Programme, Tastenzuordnungsblöcke, Datenregisterauszüge, die, versehen mit einer Alpha-Etikette, als geschlossene Datenpakete in einem Massenspeicher abgelagert werden können. Von den insgesamt sieben auf dem Interface Loop (IL) verfügbaren File-Typen - siehe Handbuch zum IL-Modul - behandelt unser Programm die Verwaltung und den Transfer der drei weitaus häufigsten: ASCII-Files (A), Daten-Files (D) und Programm-Files (P). Im HP-IL System stehen mittlerweile vier Speichermedien zur Verfügung, in denen sich Files befinden können: Der erweiterte Speicher oder Extended Memory (E), der Hauptspeicher (H), die Digitalkassette im Interface Loop (I) und die Magnetkarte (L). Die in Klammern verwendeten Abkürzungen werden auch vom vorgestellten Programm in den Menüs verwendet.

### RUF NACH AUTOMATISIERUNG

Wer die angebotenen Möglichkeiten zum vielfältigen Manipulieren und Abspeichern von Files rege ausnutzen will, sieht sich einer Fülle von entsprechenden Funktionen gegenüber, über die erst einmal Uebersicht bewahrt werden muss. Auch ihre Eingabe ist nicht sehr komfortabel, haben wir doch schon einmal darauf hingewiesen, dass die neuen Befehle fast durchwegs aus mindestens fünf Alpha-Zeichen bestehen und dass es HP offensichtlich nicht für nötig befunden hat, die Güte ihrer neuen Produkte gleich auch mit Befehls-Bar-Codes zu vervollkommen. Eine erste Herausforderung ergibt sich aus dem obligatorischen Dimensionieren von A- und D-Files an ihrem Zielort. Mit Hilfe des erweiterten Funktio-

nen-Moduls (EFM) wird dieses Dimensionieren zwar stark erleichtert, aber erst in einem Programm selbständig ausgeführt.

Des weiteren sollte man von einem Verwaltungsprogramm verlangen können, dass es die Files am Zielort sofort überprüft und nötigenfalls nochmals einspeichert. Für diese Prüfroutinen verlangt der Rechner die Namen der Zielfiles. Diese können bei A- und P-Files unter Umständen aber hinter einem Komma stehen und müssen an den Anfang des Alpha-Registers geschoben werden.



### KONFIGURATION

Je nach Quelle und Ziel der Files sind fakultativ notwendig: HP-41, Kartenleser, IL-Peripherie mit Kassettenlaufwerk, Drucker, EFM/EM.

Die kraftvollen Alpha-Ketten-Manipulationen des EFM rufen auch hier nach einer Automatisierung.

Eine letzte Herausforderung ergibt sich daraus, dass der Transfer von Files, ausser den A-Files, den Hauptspeicher benützt. Es ist nun wünschenswert, die Anzahl Datenregister nach dem Transfer wieder herzustellen oder diese gar wieder mit ihrem ursprünglichen Inhalt zu belegen. Wir wollen auch nicht vergessen, dass der Hauptspeicher vor dem Transfer von Datenfiles richtig dimensioniert wird, da bei einer allfällig zu geringen Anzahl Datenregister ohne Fehlermeldung einfach

die verfügbaren belegt werden und der restliche File verlorengeht. Die hätte einen zu "kleinen" File am Zielort zur Folge!

### WAS DAS PROGRAMM ALLES NICHT KANN

Unsere Bequemlichkeit hindert uns oftmals daran, in den Massenspeichern reine zu machen, veraltete Files zu ersetzen oder zu löschen. In den Massenspeichern häufen sich überflüssige Files an und führen zu einer Unübersichtlichkeit bezüglich ihrem Inhalt. Auch hier soll das Programm Abhilfe schaffen, indes es nach einem File-Transfer ein Löschen des Quellfiles erlaubt.

Wollen wir allen obgenannten Wünschen gerecht werden, sind es der Aufgaben für ein aufwendiges Programm genug. Um sie mit einem höchstmöglichen Bedienungskomfort zu meistern und doch für den File-Transfer im Hauptspeicher genügend Platz frei zu halten, mussten dem Programm auch einige Beschränkungen auferlegt werden:

- Es können nur ganze Files verschoben werden
- die weniger häufigen File-Typen bleiben unberücksichtigt und sind weiterhin manuell zu transferieren
- da keine Datenregister belegt werden dürfen - theoretischer SIZE 000 - kann die Abfrage nach dem Zielnamen erst nach dem Laden des Quellfiles in den Hauptspeicher erfolgen (Ausnahme: A-Files)
- es werden nur Standardgeräte angesteuert
- es wird ausschliesslich das erste Kassettengerät im Interface-Loop angesteuert.

## DIE MENUS

Beim Start des Programm mit (XEQ) "FLTRNS" werden die Directories des E- und I-Speichers ausgegeben und bei Druckerstellung NORM auch auf einem Zettel dokumentiert. Anschliessend bietet das Programm das Hauptmenu an, indem es nach dem File-Typ fragt. Mit GETKEY wartet es auf den Tastendruck des entsprechenden Typs, wonach es sofort zum zugehörigen Programmteil springt.

## Bild

Das zweite Menu ist - ebenfalls mit GETKEY - die Abfrage nach dem Quell- und Zielmedium. Erfolgt innert nützlicher Frist (ca. 12 sec.) bei einer GETKEY-Frage keine Antwort, springt das Programm zurück zum Hauptmenu, das es ohne Unterbruch laufend neu anbietet. R/S ist für die GETKEY-Funktion eine völlig gleichberechtigte Taste, wird im Programm aber dazu verwendet, dass dieses doch mit geringer Verzögerung aus dem Menu herausspringt und abbricht. Auch die obersten vier Tasten reagieren auf GETKEY: ON wurde im Programm dazu verwendet, aus dem Menu heraus sofort die Ausgabe der Directories zu veranlassen. Ausser diesen Tasten und den in den Menus angebotenen sollten keine verwendet werden. Aus dem Programmaufbau erklärt sich dieses Verbot sofort: Es könnte sich eine sinnlose Sprungadresse ergeben.

## ASCII-FILES (A)

ASCII-Files sind Textfiles, die nicht an die engen Grenzen von Datenregistern gebunden sind. Die einzelnen Zeilen (Records) können eine Länge bis zu 256 Zeichen haben. Anstelle eines Zeichens kann auch der Code für end-of-line (carriage return und line-feed) auftreten, der einen Ausdruck des Puffers und einen Zeilenvorschub bewirkt. Die ASCII-Files sind vor allem im Zusammenhang mit Steuer- und Messgeräten oder mit Fremddruckern interessant. Der Byte-Bedarf dieser Files ist in erster Line abhängig von der Anzahl Zeichen.

Dies ist ein  
ASCII - File  
mit 6 12Zeichen  
Records  
fuer Sperr -  
schrift.

ASCII-Files werden im EFM zusammengestellt und können nur dort, allerdings sehr einfach mit Hilfe von zweidimensionalen Zeigern, ma-

nipuliert oder abgerufen werden. Die Digitalkassette kann ASCII-Files lediglich speichern.

Das Verwaltungsprogramm begnügt sich mit der Abfrage nach der Herkunft, womit auch das Ziel bestimmt ist, sowie mit der Frage, ob der Quellfile zu löschen sei. Die Namen der Quell- und Zielfiles werden durch ein Komma getrennt in eine Zeile ins Alpha-Register gebracht und ab geht die Post. Es schliessen sich eine automatische Ueberprüfung mit Quittung derselben sowie ein Rücksprung zum Hauptmenu an.

## DATEN-FILES (D)

Weil das Programm für den Transfer von Datenfiles weitaus die anspruchsvollsten Aufgaben übernimmt, sei es an dieser Stelle anhand eines Beispiels etwas erläutert. Wir gehen aus von folgender Annahme: Im E-Speicher befindet sich ein Datafile "DEMO" mit 25 Registern, der aus Platzgründen dort entfernt, dafür aber auf einer Digitalkassette unter dem Namen "DAT 1" abgespeichert werden muss. Nach diesem Vorgang sei der Inhalt des Hauptspeichers mit 17 Registern wiederherzustellen.

Wir starten das Programm mit (XEQ)"FLTRNS", es erfolgt die Ausgabe der Directories - diejenige des E-Speichers wird nur in Druckerstellung NORM ausgedruckt. Der Rechner fragt: "TYP: A/D/P", wir drücken (LOG) für (D). Frage: "HER: E/H/I/L", wir drücken (E); nächste Frage: "HIN: E/H/I/L", Taste (I). Der Rechner erkundigt sich weiter: "E-FL WEG: J/N", Druck auf (J); jetzt will er noch wissen: "H RETTEN: J/N", (J) für ja. Etwa drei Sekunden Pause dienen ihm nun dazu, automatisch den SIZE des Hauptspeichers abzufragen, diesen im E-Speicher zu initialisieren und zu sichern. Danach fragt er: "E-NAME ?", wir tippen ein "DEMO" und (R/S), nach abermals drei Sekunden, in denen er das E-File abgerufen hat, erkundigt er sich: "I-NAME ?". Wir antworten ihm mit dem Namen des Files auf der Kassette: "DAT 1" und (R/S). Sofort beginnt das Kassettenlaufwerk vorwärts und rückwärts zu schnurren, nach ziemlich genau 30 Sekunden zeigt der Display an: "DAT 1, TEST", nach nochmals zwei Sekunden: "DAT 1, TEST GUT". Mit einer kleinen Verzögerung von wenigen Sekunden, die dem Rechner dazu dient, den ursprünglichen SIZE wiederherzustellen und den Hauptspeicher zurückzurufen, bietet er wieder das Hauptmenu an.

Misstrauische Leser werden nun diese Abfrage mit (R/S) verlassen und in den Handbüchern überprüfen, was uns dieser Vorgang an Erleichterungen gebracht hat und ob in allen Medien der gewünschte Zustand hergestellt ist. Wir übrige wollen an dieser Stelle noch einige Hinweise zu Kenntnis nehmen. Dienen I oder L als Quellmedien, wird noch nach der Anzahl Register des Files auf diesen Medien abgefragt, da dies nicht automatisch wie im E-Medium geschehen kann. Die Antwort findet sich für I auf dem Directory-Zettel. L als Zielmedium verlangt in der Anzeige in der gewohnten Form des Magnetkartenlesers die Karten zum Abspeichern und anschliessend sofort nochmals mit "CARD" zur Ueberprüfung. Muss der Inhalt des Hauptspeichers nicht gerettet werden, dann wird wenigstens dessen früherer SIZE wiederhergestellt, es sei denn: H ist selber Zielmedium.

Mit Hilfe unseres Programms gelingt es auch, zwei Datenfiles in E und H in einem Streich zu vertauschen: Datafile von E verlangen und H retten lassen. H wird in der kleinstmöglichen Dimensionierung mit der Etikette "M+K" in E gelagert.

## PROGRAMM-FILES (P)

Auch der Programmtransfer läuft über den Hauptspeicher. Das Verwaltungsprogramm beansprucht selber einen Teil seines Platzes. Es bleibt bei einem HP-41CV immerhin noch genügend Platz für den Transfer eines 1550-Byte-Programms oder eines 222-Register-Files. Wegen der grossen Erleichterung, die das Programm anbietet, kann bedenkenlos empfohlen werden, dass es sich als einziges im Hauptspeicher befindet.

Ist das letzte Programm im Hauptspeicher nicht mit einem END abgeschlossen, wird es bei einem Programm-File-Transfer gelöscht, also auch "FLTRNS" selber. Programme werden grundsätzlich von allen Medien ohne END übernommen, weshalb sich bei einer längeren Transferisierung die hintersten Programme immer gegenseitig löschen und den Hauptspeicher laufend frei halten.

Ist L Zielmedium, ist nur ein manuelles Uebertragen des Programms möglich. Dies wird vom Rechner auch so befohlen, sobald er über das entsprechende Programm im Hauptspeicher verfügt. Um auch Programm-Magnetkarten zu überprüfen, springe man nach dem Uebertragen des Programms auf Karte zurück mit (XEQ)

"VERY", wonach sofort die Aufforderung "CARD" erscheint und das Verwaltungsprogramm gewohnt weiterläuft. Für die Zielmedien E und I kann nach dem Programmnamen und einem Komma ein Filename angegeben werden. Für die Ueberprüfung dieses Files am Zielort werden die Namen automatisch zyklisch vertauscht und die Ueberprüfung wird sonst wie gewohnt in der Anzeige quittiert. Programme mit gesetztem Flag 11 werden normal transferiert.

aber als Subroutine angerufen und ohne STOP den Rücksprung veranlassen würde.

- Um von ausserhalb des Programms jederzeit einfach in dieses hineinspringen und es ohne Ausgabe der Directories starten zu können, taste man einfach (XEQ) "T" ein. Im Programm erreichen wir einen solchen Neustart, indem wir (XEQ) 00 eingeben oder eine GETKEY-Abfrage nicht beantworten.

- Die VER-Funktion des Kartenlesers ist nicht programmierbar. Der unsynthetische Trick stammt nicht von uns und ist folgender: Tastenzuordnung von VER bei eingestecktem Kartenleser. Diesen bei ausgeschaltetem Rechner entfernen. Beim Drücken der entsprechenden Taste im USER- und PRGM-Modus wird die Funktion programmiert.

- Kann ein File mit NO ROOM nicht in ein Zielmedium übertragen werden und wurde er vom Quellmedium schon entfernt, befindet er sich noch im Hauptspeicher!

- Das Programm beansprucht nur das Vorhandensein der Medien, die am Transfer beteiligt sind. Bei fehlender IL-Peripherie starte man es mit (XEQ) "T".

Unser Programm belegt 683 Bytes, Flags 00 bis 11 und nur den Stack zur Datenspeicherung. Im Listing sind alle Befehle im Klartext ausgedruckt. Wer über ein Bar-Code-Programm mit den XROM-Adressen verfügt, kann auf der Redaktion gratis ein Listing mit diesen verlangen. Bespielte Magnetkarten sind erhältlich nach Zusendung von Fr. 10.-- oder durch Giro auf PC 60-27 181, Informa Verlag AG, Vermerk "Fischer 82-4".

## LISTING

```

01*LBL "FLTRNS"
PWRUP AUTOIO

04*LBL 01
SF 21 ENDIR ADV ADV
DIR

10*LBL "T"

11*LBL 00
CF 21 , X<F CF 08
CF 09 "TYP: A/D/P"

18*LBL 05
TONE 0 AVIEW GETKEY
GTO IND X

23*LBL 33
ISG Y

25*LBL 24
ISG Y

27*LBL 23
ISG Y

29*LBL 15

30*LBL 41
CF 10 RTN
    
```

## ZUM AUFBAU DES PROGRAMMS

- Beim Listing sind die einzelnen Kommas unbedingt zu beachten, sie entsprechen einem sehr schnellen (CLX) - mit Stack-Lift.
- Versuchen Sie keinesfalls, mit den LBL-Nummern "Ordnung" schaffen zu wollen. Die GETKEY-Funktion wird Ihnen diesen Versuch in ein gewaltiges Chaos verwandeln.
- Der STOP-Befehl am Ende ist unerlässlich, weil LBL 84 (GETKEY-R/S) das Programm stoppen soll,

## LETZTE HINWEISE UND TRICKS

- Sind Programme zwischen zwei HP-Rechnern auszutauschen, kann dies als File mit einem raschen Umstecken des EFM-Moduls geschehen. Dies ist aber nur dann möglich, wenn keine EM-Moduln angeschlossen oder diese "leer" sind. Rechner ausschalten!
- Schnelle Kassettenkopien erhält man mit dem Transferprogramm, I als Quell- und Zielmedium und Kassettentausch beim Namensprompt.

C - CV - IL - EFM - EM - SP

Red. Die Finanzspannweite des HP-41-Systems bewegt sich mittlerweile zwischen Fr. 600.-- und Fr. 5000.--. Die Anwendungsspannweite reicht vom HP-41 als Taschenrechner bis zu seinem System in der IL-Grosskonfiguration, die Programme decken von der Lösung einer Proportion bis zu Mess- und Regelaufgaben, vom einfachen Tastencode bis zu hochverschlüsselten Synthetischen Programmen alles ab. Diese Verschiedenheit bezüglich Ansprüchen und Konfiguration wird sich wahrscheinlich auch in Inhalt und Zweck kommender Artikeleinsendungen widerspiegeln. Die Redaktion wird deshalb bestrebt sein, künftigen Veröffentlichungen zum HP-41-System schon äusserlich eine möglichst grosse Transparenz betreffend Inhalt, Zweck, Programmiertechnik und verlangter Systemkonfiguration zu verleihen. Sie bittet deshalb die Autoren, ihr bei diesem Bestreben zu helfen:

- Vermerken Sie auf einem separaten Blatt die minimal verlangte Systemkonfiguration für Ihren Programmvorschlag.
- Um auch Lesern mit kleiner Konfiguration möglichst gerecht zu werden, bitten wir Sie, Ihre Programme für den geringsten Geräteaufwand auszulegen. Es ist Ihnen selbstverständlich überlassen, auf weitere Lösungen, zum Beispiel mit dem EFM, hinzuweisen (siehe Tonprogramm im Artikel von Erwin Gosteli in diesem Heft).
- Programmieren Sie entweder konsequent "normal" oder synthetisch und kennzeichnen Sie Programme mit Synthetischen Bestandteilen schon im Titel als solche. Geben Sie die Dezimaläquivalente bei Synthetischen Textzeilen am Ende des Listings an. Drucken Sie Synthetische Listings möglichst mit einem "alten" Drucker.
- Legen Sie allen Programmen Magnetkarten und evtl. die Bar-Codes bei. Sie ersparen uns damit viel Arbeit. Bespielte Digitalkassetten werden unabhängig vom Veröffentlichungstermin sofort nach Eingang kopiert und zurückgeschickt.
- Artikel, die auf das leistungsfähige IL-System zugeschnitten sind, werden künftig das weiter oben zu findende Signet tragen.

# PPC/HHC - Die Programmierbaren

```

33*LBL 11
SF 11 XEQ 22 FS? 02
XEQ 13 XEQ 26 FS? 00
"E" FS? 02 "I" 40
XTOA FS? 00 "I,I"
FS? 02 "I,E" 41 XTOA
XEQ 06 FS? 02 GTO 08
FLSIZE XEQ 07 RCL Z
CREATE XEQ 07 SAVERS
FS? 09 XEQ 27 XEQ 07
XEQ 28 GTO 00

65*LBL 08
R↑ XEQ 07 RCL Z
CRFLAS XEQ 07 GETAS
FS? 09 XEQ 27 GTO 00

75*LBL 06
TONE 0 "I-NAME ?" AON
PROMPT AOFF AVIEW RTN

83*LBL 07
"↑," 44 POSA AROT
ATOX RTN

90*LBL 14
XEQ 21 "H RETTEN: J/N"
FC? 01 XEQ 05 SIZE?
ENTER↑ ENTER↑ ENTER↑
FS? 08 XEQ 09 FS? 00
XEQ 10 FS? 02 XEQ 12
FS? 03 XEQ 13 FS? 09
XEQ 27 FS? 05 GTO 00
FS? 04 XEQ 18 FS? 06
XEQ 16 FS? 07 XEQ 17
R↑ PSIZE FC? 08
GTO 00 "M+K" GETR
PURFL GTO 00

125*LBL 22
"HER: E/" FS? 11 "I"
FC? 11 "H/I/L" ,
XEQ 05 SF IND Y
FS?C 11 RTN
"HIN: E/H/I/L" 4
XEQ 05 SF IND Y RTN

141*LBL 09
"M+K" CRFLD SAVER RTN

146*LBL 10
"E" XEQ 06 FLSIZE
PSIZE GETR RTN

153*LBL 12
XEQ 13 PSIZE "I"
XEQ 06 READR RTN

160*LBL 13
TONE 0 "ANZ. " FS? 02
"↑I" FS? 03 "H"
"↑-REG ?" PROMPT
FS? 02 RTN PSIZE RDTA
RTN

174*LBL 25
FC? 10 SF 08 FS?C 10
SF 09 RTN

180*LBL 18
"E" XEQ 06 CRFLD
SAVER RTN

186*LBL 16
"↑I" XEQ 06 CREATE
WRTR XEQ 28 FC?C 25
GTO 16 RTN

195*LBL 17
CLD WDTA VER RTN

200*LBL 21
XEQ 22 FS? 00 XEQ 26
FS? 02 XEQ 26 RTN

207*LBL 43
XEQ 21 FS? 01 GTO 19
CLD FS? 03 RSUB
FS? 03 GTO 19 FS? 02
"↑I" FS? 00 "E" XEQ 06
FS? 02 READP FS? 00
GETP FS? 09 XEQ 27

227*LBL 19
FS? 05 GTO 00 FS? 07
GTO 20 "P" 40 XTOA
"↑,FL" 41 XTOA XEQ 06
FS? 04 SAVEP FS? 04
GTO 00 WRTP XEQ 07
XEQ 28 FC?C 25 GTO 19
GTO 00

249*LBL 26
SF 10 FS? 00 "E"
FS? 02 "I"
"↑-FL WEG: J/N" XEQ 05
RTN

258*LBL 27
FS? 00 PURFL FS? 02
PURGE RTN

264*LBL 28
SF 25 "↑, TEST" AVIEW
VERIFY FC? 25 "NICHT"
"↑ GUT" AVIEW RTN

274*LBL 20
"MANUELL" PROMPT

277*LBL "VERY"
VER GTO 00

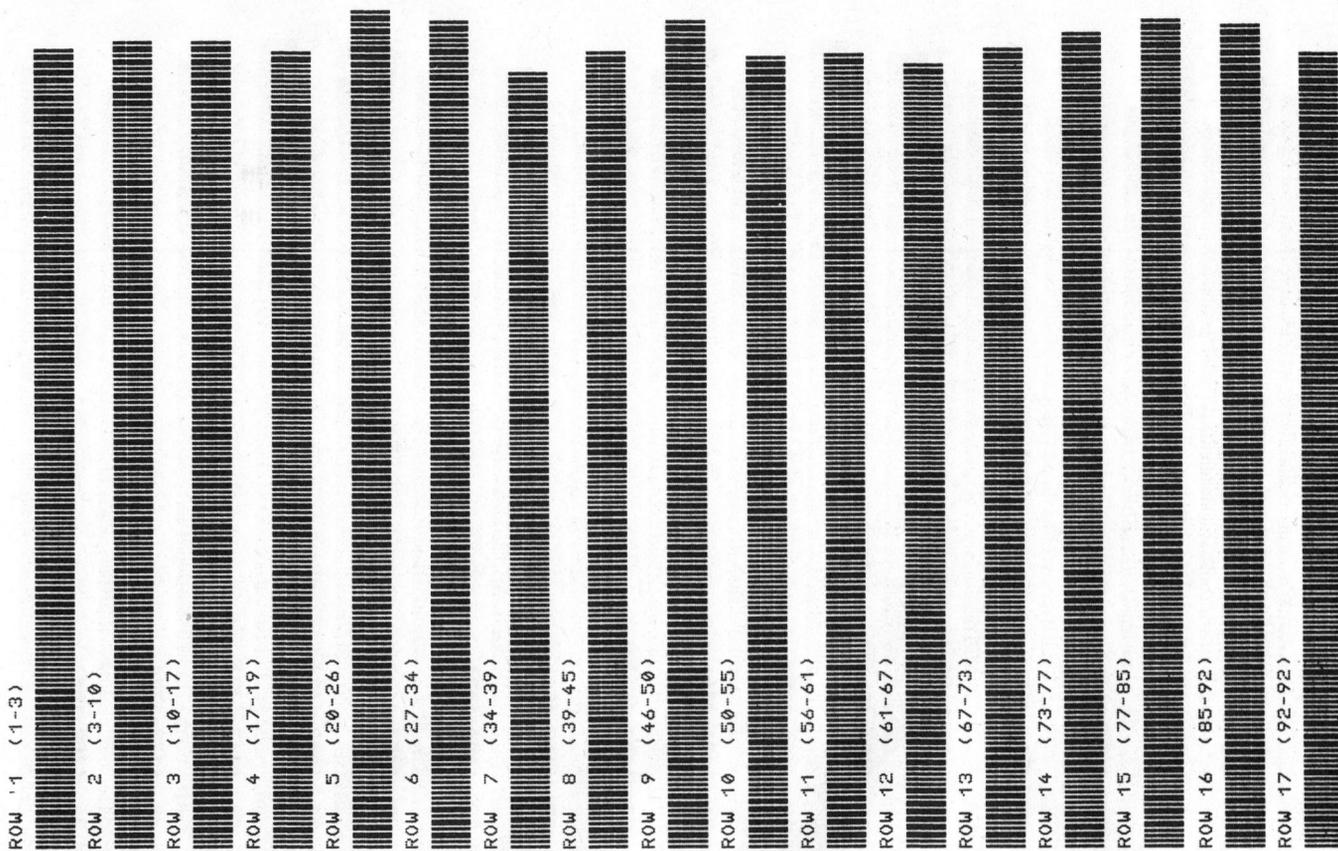
280*LBL 84
CLD STOP END

```

## Überwinden Sie bestehende Grenzen...



Bar-Codes gedruckt von Peter LAEDRACH



## ...mit Ihrem HP-Rechner!

Über den Hewlett-Packard Interface Loop – kurz HP-IL genannt – lassen sich bis zu 30 kompatible Geräte wie z.B. Thermodrucker, Kassettenlaufwerke und zukünftige Peripheriegeräte gleichzeitig an den HP-41 anschliessen.

keiten des neuen Druckers können Dokumentationen noch problemloser und übersichtlicher gestaltet werden.

Das ist HP-IL – ein neuer Standard. Natürlich von Hewlett-Packard.



Digital-Kassettenlaufwerk. Grosse Datenmengen und eine Vielzahl von Programmen können heute bequem und schnell auf einer Minikassette mit einer Speicherkapazität von 131000 Bytes gespeichert werden. Thermodrucker. Durch vielfältige Formatiermöglich-

Ich interessiere mich für den Hewlett-Packard Interface Loop.

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

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

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

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

Tel.: \_\_\_\_\_

Senden an:

Hewlett-Packard (Schweiz) AG

Abt. Information, Allmend 2, 8967 Widen

# PPC/HHC - Die Programmierbaren

ROW 18 (93-100)



ROW 19 (100-105)



ROW 20 (105-110)



ROW 21 (111-116)



ROW 22 (116-121)



ROW 23 (122-126)



ROW 24 (126-130)



ROW 25 (130-135)



ROW 26 (136-136)



ROW 27 (137-143)



ROW 28 (143-150)



ROW 29 (150-157)



ROW 30 (157-162)



ROW 31 (162-167)



ROW 32 (167-172)



ROW 33 (172-179)



ROW 34 (180-186)



ROW 35 (186-191)



ROW 36 (192-198)



ROW 37 (199-204)



ROW 38 (205-210)



ROW 39 (210-216)



ROW 40 (217-222)



ROW 41 (223-228)



ROW 42 (229-234)



ROW 43 (235-239)



ROW 44 (239-245)



ROW 45 (245-250)



ROW 46 (251-255)



ROW 47 (255-257)



ROW 48 (258-264)



ROW 49 (265-268)



ROW 50 (269-271)



ROW 51 (271-275)



ROW 52 (276-279)



ROW 53 (280-283)



## Pythagoras auf Umwegen

Felix DANERS

In MIKRO- UND KLEINCOMPUTER 81-4 wurde ein Programm zur Berechnung rechtwinkliger Dreiecke veröffentlicht. Der Autor des Beitrags hielt die interessante Verzweigungsmethode vor allem für HP-Rechner mit numerischen Labels geeignet. Dieser Beitrag vermochte nun einen jungen Leser dazu herauszufordern, die Idee auch für seinen TI-58C zu verwirklichen und damit gegen den Umstand anzukämpfen, dass man auf TI-Rechnern keine numerischen Labels kennt.

Die erwähnte Verzweigungsmethode sei eingangs nochmals erläutert: Für die Berechnung eines rechtwinkligen Dreiecks sind in der Regel neben dem rechten Winkel zwei Größen verlangt, um alle weiteren berechnen zu können. Wie merkt der Rechner nun sehr schnell, welche der Größen eingegeben wurden? Die Idee war die, dass jeder Grösse bei deren Eingabe ein Codezahl zugeordnet wird. Nach der zweiten Eingabe mit Identifikation über eine Label-Taste stellt sich der Rechner einen zweistelligen Steuercode her, der als Adresse eines indirekt angerufenen Labels dient.

Dieses Verzweigungssystem hat gegenüber den üblicherweise verwendeten Programmteilen wesentliche Vorteile:

- Der Rechner ist sehr einfach zu bedienen
- Daten können nicht zum falschen Zeitpunkt eingegeben werden
- es ist sehr schnell
- es benötigt nicht viel Speicherplatz.

Bei der Umsetzung für TI-Rechner werden folgende Codezahlen für die Größen in Bild 1 verwendet:

0 für a, 2 für b, 4 für c,  
6 für p, 8 für q.

Die Höhe h wird uns später beschäftigen.

Aus den beiden gegebenen Größen setzt der Rechner nun die zweistellige Codezahl zusammen, und zwar so, dass immer die kleinere der beiden vorangeht. Sind zum Beispiel die Seite b und der Hypotenusenabschnitt p eines rechtwinkligen Dreiecks gegeben, entsteht so der Code 26. HP-Rechner haben numerische Labels (z.B. LBL 26) und können auch nur auf Labels verzweigen, also nicht auf Zeilennummern wie die TI-Rechner. Das numerische LBL 26 aus unserem Beispiel müsste nun noch indirekt aufgerufen werden.

### AUF UMWEGEN ZUM ZIEL

Auf TI-Rechnern kann man leider nicht so direkt zur Lösungsformel springen. Der Ausschnitt aus dem Flussdiagramm (Bild 2) zeigt, wie man mit einem Zwischenschritt dennoch zum Ziel gelangt: Die Codezahl dient als Adresse und ist eine Zeilennummer. In dieser Zeilennummer nimmt der Rechner den Befehl zum Anspringen der Lösungsformel auf, in der die Berechnung der nicht gegebenen Größen eingeleitet werden kann.

### PLATZMANGEL

Aus Platzgründen konnte für die Höhe h kein Code mehr verwendet werden. Der Höhe wird deshalb die gleiche Codezahl zugeteilt, wie der Seite a. Zusätzlich wird aber noch ein Flag gesetzt, welches das Anspringen einer anderen Formel veranlasst.

Ebenfalls aus Platzmangel musste mit den Rechnern TI-58/58C auf eine zweite, bessere aber langsamere Methode verzichtet werden. Da sie aber als Anregung zu anderen Programmvarianten aufgenommen werden könnte, sei sie dennoch kurz beschrieben.

Die Codezahl wird in das T-Register geschoben. Im Programm wird nun jede mögliche Codezahl in das X-Register gelesen und mit T verglichen. Ist die Codezahl gefunden, wird verzweigt. Besteht keine Lösung zum gestellten Problem, kann eine Fehlermeldung erzeugt werden, was bei der ersten Variante nicht möglich ist. Der Rechner setzt im Falle eines "falschen" Codes irgendwo im Programmspeicher ein. Es erscheint damit in den meisten Fällen eine Fehlermeldung, da mit grosser Wahrscheinlichkeit eine Formel falsch angewendet wird.

### WICHTIGE BENUTZERHINWEISE

Folgende Punkte sollten beim Gebrauch der hier vorgestellten Programmvariante unbedingt beachtet werden:

- Vor der Eingabe des zweiten Wertes darf das T-Register nicht gelöscht werden, da sonst der Code zerstört wird
- wenn die erste Zahl falsch eingegeben wurde, kann durch Drücken von RST eine Korrektur erfolgen, d.h. der richtige Wert kann eingegeben werden
- wenn ein Fehler vorliegt, blinkt die Anzeige
- die Ausgabe kann mit R/S unterbrochen und später wieder fortgesetzt werden, solange der Programmzeiger nicht verstellt wird und die Register 00-06 nicht gelöscht werden

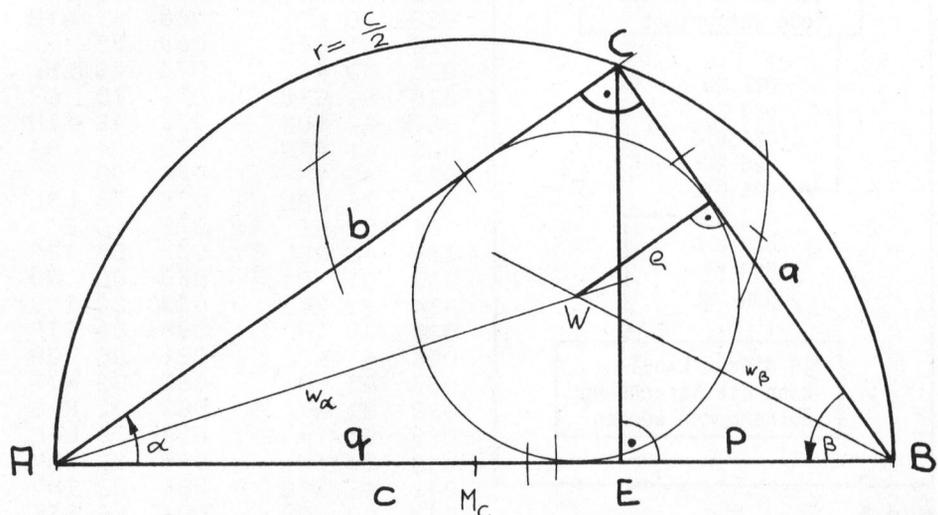


Bild 1: Grössen im rechtwinkligen Dreieck

# PPC/HHC - Die Programmierbaren

- die Ergebnisse können später aus den folgenden Registern abgerufen werden:

a	00
b	01
c	02
p	03
q	04
h	05
$\alpha$	06
$\beta$	wird nicht gespeichert
Code	09

- möglich sind sämtliche Kombinationen ausser "c, h"

- bei fehlerhaften Ausgaben RST drücken und neu beginnen

- Flags 0 und 1 sind belegt

- Labels: A für a, B für b, C für c, D für p, E für q, A' für h.

## ANWENDUNGSBEISPIEL

Gegeben sind der Hypotenusenabschnitt p mit 1,8 cm und die Seite a mit 3 cm.

Tastenfolge	Kommentar	Anzeige
1.8		
D	Initialisieren von p	T-Register
3		
A	Initialisieren von a	3
R/S	Seite a	4
R/S	Seite b	5
R/S	Seite c	1.8
R/S	Hyp'abschnitt	3.2
R/S	Hyp'abschnitt	2.4
R/S	Höhe h	36.869 Grad
R/S	Winkel $\alpha$	53.131 Grad
R/S	Winkel $\beta$	

## LISTING

```

000 61 GTD      044 55 ÷      089 22 INV      134 61 GTD
001 42 STD      045 68 NOP      090 77 GE       135 43 RCL
002 61 GTD      046 61 GTD      091 23 LNX      136 00 00
003 32 X!T      047 53 (       092 32 X!T      137 33 X²
004 61 GTD      048 61 GTD      093 76 LBL      138 33 X²
005 33 X²       049 54 )       094 23 LNX      139 55 ÷
006 61 GTD      050 76 LBL      095 65 ×       140 43 RCL
007 34 FX       051 12 B        096 01 1       141 03 03
008 61 GTD      052 42 STD      097 00 0       142 33 X²
009 35 1/X      053 01 01      098 85 +       143 75 -
010 76 LBL      054 02 2       099 32 X!T      144 43 RCL
011 11 A        055 10 E'       100 54 )       145 00 00
012 42 STD      056 76 LBL      101 42 STD      146 33 X²
013 00 00       057 13 C        102 09 09      147 95 =
014 00 0        058 42 STD      103 83 GO*     148 34 FX
015 10 E'       059 02 02      104 09 09      149 42 STD
016 76 LBL      060 04 4       105 76 LBL      150 01 01
017 16 A'       061 10 E'       106 32 X!T      151 61 GTD
018 42 STD      062 76 LBL      107 87 IFF      152 95 =
019 05 05       063 14 D        108 01 01      153 76 LBL
020 86 STF      064 42 STD      109 52 EE       154 35 1/X
021 01 01      065 03 03      110 61 GTD      155 87 IFF
022 00 0        066 06 6       111 95 =       156 01 01
023 10 E'       067 10 E'       112 76 LBL      157 65 ×
024 61 GTD      068 61 GTD      113 33 X²      158 43 RCL
025 43 RCL      069 55 ÷       114 87 IFF      159 04 04
026 61 GTD      070 76 LBL      115 01 01      160 65 ×
027 44 SUM      071 15 E        116 68 NOP      161 53 (
028 61 GTD      072 42 STD      117 43 RCL      162 24 CE
029 45 YX       073 04 04      118 02 02      163 85 +
030 76 LBL      074 08 8       119 33 X²      164 53 (
031 52 EE       075 76 LBL      120 75 -       165 24 CE
032 43 RCL      076 10 E'       121 43 RCL      166 33 X²
033 01 01      077 87 IFF      122 00 00      167 85 +
034 33 X²       078 00 00      123 33 X²      168 04 4
035 75 -        079 22 INV      124 95 =       169 65 ×
036 43 RCL      080 86 STF      125 34 FX       170 43 RCL
037 05 05      081 00 00      126 42 STD      171 00 00
038 33 X²       082 32 X!T      127 01 01      172 33 X²
039 95 =        083 91 R/S      128 61 GTD      173 54 )
040 34 FX       084 76 LBL      129 95 =       174 34 FX
041 42 STD      085 22 INV      130 76 LBL      175 54 )
042 00 00       086 22 INV      131 34 FX       176 65 ×
043 61 GTD      087 86 STF      132 87 IFF      177 93 .
044 61 GTD      088 00 00      133 01 01      178 05 5
    
```

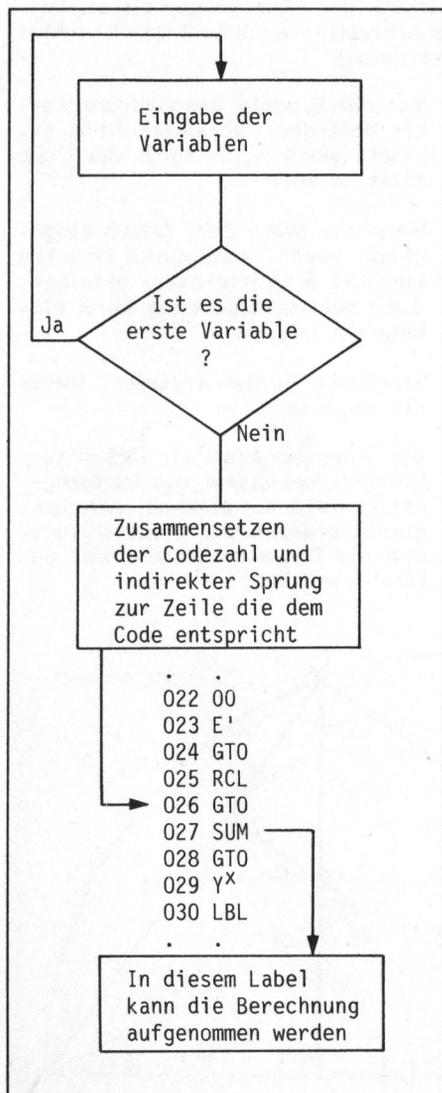


Bild 2: Entstehung der Sprungadressen

# PPC/HHC - Die Programmierbaren

179	95	=	225	24	CE	319	04	04	272	65	x	366	02	02	<b>Namen und Position der verwendeten Labels</b>
180	34	FX	226	85	+	320	33	X <sup>2</sup>	273	43	RCL	367	75	-	
181	42	STD	227	53	(	321	85	+	274	03	03	368	43	RCL	
182	01	01	228	24	CE	322	43	RCL	275	95	=	369	03	03	
183	61	GTD	229	33	X <sup>2</sup>	323	05	05	276	34	FX	370	95	=	
184	95	=	230	85	+	324	33	X <sup>2</sup>	277	42	STD	371	42	STD	
185	76	LBL	231	04	4	325	95	=	278	00	00	372	04	04	
186	42	STD	232	65	x	326	34	FX	279	61	GTD	373	91	R/S	
187	22	INV	233	43	RCL	327	42	STD	280	33	X <sup>2</sup>	374	43	RCL	
188	87	IFF	234	01	01	328	01	01	281	76	LBL	375	04	04	
189	01	01	235	33	X <sup>2</sup>	329	61	GTD	282	54	)	376	65	x	
190	68	NOP	236	54	)	330	45	YX	283	43	RCL	377	43	RCL	
191	43	RCL	237	34	FX	331	76	LBL	284	02	02	378	03	03	
192	00	00	238	54	)	332	95	=	285	65	x	379	95	=	
193	33	X <sup>2</sup>	239	65	x	333	22	INV	286	43	RCL	380	34	FX	
194	75	-	240	93	.	334	86	STF	287	04	04	381	42	STD	
195	43	RCL	241	05	5	335	01	01	288	95	=	382	05	05	
196	05	05	242	95	=	336	43	RCL	289	34	FX	383	91	R/S	
197	33	X <sup>2</sup>	243	34	FX	337	00	00	290	42	STD	384	43	RCL	
198	95	=	244	42	STD	338	91	R/S	291	01	01	385	06	06	
199	34	FX	245	00	00	339	43	RCL	292	61	GTD	386	91	R/S	
200	42	STD	246	61	GTD	340	01	01	293	43	RCL	387	09	9	
201	03	03	247	95	=	341	91	R/S	294	76	LBL	388	00	0	
202	61	GTD	248	76	LBL	342	43	RCL	295	55	+	389	75	-	
203	34	FX	249	45	YX	343	01	01	296	43	RCL	390	43	RCL	
204	76	LBL	250	43	RCL	344	32	XIT	297	03	03	391	06	06	
205	43	RCL	251	01	01	345	43	RCL	298	65	x	392	95	=	
206	43	RCL	252	33	X <sup>2</sup>	346	00	00	299	43	RCL	393	29	CP	
207	02	02	253	33	X <sup>2</sup>	347	22	INV	300	04	04	394	91	R/S	
208	33	X <sup>2</sup>	254	55	+	348	37	P/R	301	95	=	395	76	LBL	
209	75	-	255	43	RCL	349	42	STD	302	34	FX	396	68	NOP	
210	43	RCL	256	04	04	350	06	06	303	42	STD	397	22	INV	
211	01	01	257	33	X <sup>2</sup>	351	32	XIT	304	05	05	398	86	STF	
212	33	X <sup>2</sup>	258	75	-	352	42	STD	305	76	LBL	399	01	01	
213	95	=	259	43	RCL	353	02	02	306	61	GTD				
214	34	FX	260	01	01	354	91	R/S	307	43	RCL				
215	42	STD	261	33	X <sup>2</sup>	355	43	RCL	308	05	05				
216	00	00	262	95	=	356	00	00	309	33	X <sup>2</sup>				
217	61	GTD	263	34	FX	357	33	X <sup>2</sup>	310	55	+				
218	95	=	264	42	STD	358	55	+	311	43	RCL				
219	76	LBL	265	00	00	359	43	RCL	312	03	03				
220	44	SUM	266	61	GTD	360	02	02	313	95	=				
221	43	RCL	267	95	=	361	95	=	314	42	STD				
222	03	03	268	76	LBL	362	42	STD	315	04	04				
223	65	x	269	53	(	363	03	03	316	76	LBL				
224	53	(	270	43	RCL	364	91	R/S	317	65	x				
			271	02	02	365	43	RCL	318	43	RCL				

Das Programm belegt beide Spuren einer Magnetkarte. Eine bespielte Magnetkarte kann gegen Zusendung von Fr. 5.-- bestellt werden. Bei PC-Ueberweisung bitte den Vermerk "Daners 82-4" auf dem Girozettel anbringen.

Verwandeln Sie Ihren Tischrechner in ein leistungsfähiges Datenerfassungssystem

HP-85, APPLE II, IBM, PDP 11

## ANALOG DEVICES

9, rue de Berne  
CH-1201 GENÈVE  
Tél.: 022 / 31 57 60

Micromac 4000 bietet Ihnen:

- Mess- und Steuerungssysteme
- Anwendungen
- Datenerfassung
- Datenregistrierung
- Prozesssteuerung
- Aufbereitung der Fühlersignale
- Flexibilität um sich der Anwendung anzupassen
- Entlastet Ihren Computer von Routinearbeiten
- Gebrauchsfertige Kommunikationsware

Bitte, verlangen Sie detaillierte Unterlagen



## Programmgenerator und -generierung

Gerfried TATZL, dipl. Ing. WIV

### Teil 3, Schluss

Im vorliegenden letzten Teil unserer Serie erläutert der Autor die Hilfsprogramme für die Kalkulation und die Stücklistenauflösung. Der Aufbau der Programme und deren Handhabung werden detailliert erklärt und an einem Beispiel vertieft. Zum Schluss wird ein genauer Hinweis gegeben, wie ein Speicherüberlauf einfach korrigiert werden kann. Jeder Leser sollte nun aus den Programmen die für seine Anwendung wichtigen Teile herauschälen und damit wertvolle Programmschritte sparen können.

Das Programm für die automatisch ablaufende Kalkulation muss so konzipiert werden, dass es von der Bedienungsseite her gesehen, folgendes Aussehen hat:

- Datenkarte mit den Verrechnungsdaten einlesen
- Steuerprogramm S 005 SERKALK einlesen
- Programm mit Betätigung der Taste E starten
- Tasten (f) (merge) drücken; Anzeige Crd
- Datenprogramm S 004 PRODAT ANR einlesen
- Verarbeitung mit Betätigung der R/S-Taste einleiten. Für den Fall, dass der Rechner vor dem Ausdruck der Herstellkosten anhält, sind die Sonderkosten der Fertigung einzugeben und die R/S-Taste zu drücken; das gleiche gilt sinngemäss bei Anhalten des Rechners vor Ausdruck der Gesamtkosten für die Sonderkosten der Verwaltung und des Vertriebes.

Die Verarbeitung ist mit Ausdruck der Herstellkosten (Grenz- und Vollkosten sowie Variator) und dem Ausdruck der Gesamtkosten (Grenz- und Vollkosten sowie Variator) beendet. Die nächste Verarbeitung kann mit der Betätigung der Tasten (f) (merge) in Angriff genommen werden.

Dieses Steuerprogramm für die Kalkulation besteht im wesentlichen aus der Summe der Unterprogramme, die von einem Datenprogramm aus aufgerufen werden, wobei der mit dem LBL E in Programmzeile 077 eingeleitete Zweig etwas herausfällt: Dieser ist nämlich dafür erforderlich, dass nach einer Verarbeitung automatisch das nächste Datenprogramm nach Drücken der Tasten (f) und (merge) eingeschoben und in gleicher Weise wie die zuvor eingelesenen verarbeitet werden kann.

### AUFBAU DES PROGRAMMS

Das Operationsprogramm für die Kalkulation besteht also im einzelnen aus folgenden Unterprogrammen:

- Ausdrucksroutine für die Artikelnummer (LBL C)
- Ermittlung der Einzelkosten mit nachfolgender Aufsummierung zu Gesamtkosten für Kostenarten, für die die zugehörigen Verrechnungsdaten im linken Teil eines Datenspeichers zu finden sind (LBL 1)
- Ermittlung der Einzelkosten mit nachfolgender Aufsummierung zu Gesamtkosten für Kostenarten, für

die die zugehörigen Verrechnungsdaten im rechten Teil eines Datenspeichers zu finden sind (LBL 2)

- Aufsummierung von Sonderkosten (LBL 6)
- Ausdrucksroutine für Grenz- und Vollkosten auf der Basis von Herstell- bzw. Gesamtkosten (LBL 8)
- Teilung einer Zahl durch 100 (LBL 9).

Das Entpacken der Verrechnungsdaten ist speicherteilgerecht auszuführen und in den betreffenden Unterprogrammen gelöst.

### PROGRAMMLISTUNG

001	LBL 0	Ausdruck der Artikelnummer ANR
002	SPC	Leerzeile
003	CF 3	da das Flag 3 durch das Einlesen der Datenkarte sowie durch Eingaben von Sonderkosten gesetzt wird, ist dieses Flag vor einer Verarbeitung zu löschen, da es im Programm anderweitig Verwendung findet
004	PRT X	Ausdruck der Artikelnummer
005	DSP 2	Einstellen auf eine Zahlenausgabe mit zwei Dezimalstellen
006	1	
007	STO I	
008	0	gleichzeitig Einstellen der ersten möglichen Leermeldung
009	STO 0	löschen des Speichers für Grenzkosten...
010	STO 1	... und für Vollkosten
011	RTN	Ende des mit dem LBL 0 eingeleiteten Unterprogramms
012	LBL 1	Zerlegung Speicherteil 1 links
013	ISZ I	
014	x = 0?	
015	RTN	Abkürzung einer Verarbeitung im Fall einr Leermeldung; kein artikelbezogener Mengenwert vorhanden
016	RCL(i)	Indirekter Abruf eines Speicherinhalts
017	INT	Isolierung des linken Speicherteils
018	GSB 9	Teilung durch 100
019	GTO 3	Fortsetzung bei gemeinsamer Verarbeitung in Programmzeile 028
020	LBL 2	Zerlegung Speicherteil 2 rechts
021	x = 0?	
022	RTN	Abkürzung der Verarbeitung im Fall einer Leermeldung
023	RCL(i)	
024	FRAC	Isolierung des rechten Speicherteils
025	EEX	
026	3	
027	*	
028	LBL 3	Gemeinsame Verarbeitung für beide Speicherteile 1 links und 2 rechts
029	FRAC	Isolierung des Variators
030	LAST X	

# PPC/HHC - Die Programmierbaren

031	INT	Isolierung des Verrechnungssatzes
032	EEX	
033	0	Platzhalter für unterschiedliche Zahl von Dezimalstellen
034	:	
034	RCL E	
035	+	Verrechnung von Grenz- und Vollkosten nur bei 18.2 wirksam
036	R ↑	Stammdatum im X-Speicher (Anzeige)
038	F ?	
039	GSB 9	Teilung durch 100 wegen Prozentverrechnung
040	*	Berechnung der Einzel-Vollkosten
041	RND	Runden auf zwei Dezimalstellen
042	STO+1	Neue Zwischensumme Vollkosten
043	F ?	
044	RTN	Stop für Gewinnverrechnung; Gewinn nur auf Vollkosten
045	X<>Y	Variator in der Anzeige, Vollkosten im Y-Speicher
046	x ≠ 0?	
047	GTO 4	
048	x<>y	Vollkosten in die Anzeige
049	1	Variator 0 ist Variator 100 %
050	LBL 4	
051	*	Ermittlung der Einzel-Grenzkosten
052	RND	Runden auf zwei Dezimalstellen
053	STO+0	Neue Zwischensumme Grenzkosten
054	0	Einsparung Leermeldung auf Anlageblatt
055	RTN	Ende Unterprogramm 1 und 2 und Rückkehr ins Hauptprogramm mit der Zahl 0 im X-Speicher
056	LBL 6	Aufsummierung Sonderkosten
057	R/S	Stop für Eingabe Grenzkosten (ENTER) Vollkosten. Für unser Beispiel ist im Fall der Fertigungskosten die Zahl 45 einzutasten und die ENTER-Taste zu betätigen, worauf die R/S-Taste gedrückt werden kann
058	STO+1	Aufsummierung Sonder-Vollkosten
059	R ↓	
060	STO+0	Aufsummierung Sonder-Grenzkosten
061	CF 3	
062	RTN	Ende des Unterprogramms für die Sonderkosteneingabe
063	LBL 8	Kostenausdruck
064	SPC	Zeilenvorschub
065	RCL 0	
066	PRT X	Ausdruck Grenzkosten
067	RCL 1	
068	PRT X	Ausgabe Vollkosten
069	:	Berechnung Variator geteilt durch 100
070	PRT	Ausdruck Variator
071	RTN	Ende der Druckroutine
072	LBL 9	Teilung durch 100
073	EEX	
074	9	
075	:	
076	RTN	Ende des Unterprogramms für die Teilung durch 100
077	LBL E	Startroutine zum Overlay für das nächste Datenprogramm
078	R/S	
079	DSP 0	
080	GTO E	Diese Anweisung wird nach Betätigung der Tasten (f) (merge) und nachfolgendem Einlesen eines neuen Datenprogramms überschrieben

nach dem Ausdruck der Artikelnummer bewerkstelligt werden. Die Anzahl der zusätzlichen Anweisungen muss ohne Beeinträchtigung sämtlicher Datenprogramme möglich sein. Sie können auch noch andere Varianten einbauen; hier sind Ihrer Phantasie keine Grenzen gesetzt.

## BEISPIEL

Das im vorigen Heft erwähnte Datenprogramm eines Blumentrogs würde nun wie folgt kalkuliert, wobei die nachfolgenden Verrechnungssätze unterlegt seien:

Schlüssel	Verrechnungssatz	Variator
2.1	30.--	70
2.2	45.--	85
3.1	720.--	100
3.2	780.--	100
4.1	850.--	100
4.2	200.--	100
5.1	80.--	100
5.2	75.--	100
6.1	70.--	100
6.2	180.--	100
7.1	150.--	100
7.2	78.--	100
8.1	52.--	100
8.2	500.--	100
9.1	800.--	100
9.2	840.--	100
10.1	750.--	100
10.2	15.--	100
11.1	120.--	95
11.2	105.--	95
12.1	112.--	95
12.2	20.--	60
13.1	134.--	75
13.2	555.--	64
14.1	210.--	72
14.2	58.--	88
15.1	25.--	55
15.2	15.--	95
16.1	48.--	45
16.2	25.--	25
17.1	700.--	38
17.2	84.--	65
18.1	20.--	40
18.2	15.--	5
19.1	112.--	95
19.2	22.--	60
20.1	12.--	0

Die Verarbeitung des Datenprogramms S 004 PRODAT 29500 ergibt:

Herstellkosten:	
Grenzkosten	1196.19
Vollkosten	1333.35
Variator	0.90

Gesamtkosten:	
Grenzkosten	1227.75
Vollkosten	1653.05
Variator	0.74

Die Durchführung der Kalkulation ist im wesentlichen auf die beiden mit den LBL 1 und LBL 2 eingeleiteten Unterprogramme beschränkt. Die Verrechnungsdaten sind im übrigen auf alle drei Kalkulationsvarianten anwendbar, die wir im ersten Abschnitt zu diesem Thema skizzenhaft angedeutet hatten. Daher sind sich auch die gewählten Kostenartenschlüssel gleich. Wenn wir mit dem angebotenen Datenspeichervolumen das Auslangen finden, könnten wir in den Steuerteil verschiedene zu-

sätzliche Programmzweige vorsehen, so den Einbau einer Datumsautomatik, nach der vor oder nach dem Ausdruck der Artikelnummer ein Daten in der Form JJJJ.MMTT oder TTMM.JJJJ zusätzlich ausgedruckt wird. Das Tagesdatum könnte beispielsweise im Speicher C(22) abgespeichert werden und mit der Programmfolge

```
RCL C    DSP 0
DSP 4    X<>Y
PRT X
```

Dabei vergessen wir nicht, dass bei einer Verarbeitung die Sonderkosten der Fertigung mit 45.00 einzutasten sind, worauf die Tasten (ENTER) und (R/S) gedrückt werden müssen; diese Eingabe wird vom Datenprogramm noch vor dem Ausdruck der Herstellkosten verlangt.

Im nächsten und letzten Abschnitt zur Frage von Programmgeneratoren bei Verwendung programmierbarer Taschenrechner befassen wir uns mit dem Steuerprogramm für eine Stücklistenauflösung.

## STUECKLISTENAUFLOESUNG

Grundsätzlich gleichartig ist auch das Operationsprogramm für die Stücklisten- oder Mengenauflösung S 006 MATAUF aufgebaut. Das ein Kalkulationsschema enthaltene Generatorprogramm ändert sich dabei verständlicherweise gar nicht, sondern nur die im Steuerenteil enthaltenen Unterprogramme, die bekanntlich von einem Datenprogramm aufgerufen werden. Auf diese Weise können die gleichen Artikel-Stammdaten sowohl für eine Kalkulation in Verbindung mit den Verrechnungsdaten, als auch für eine Stücklistenauflösung herangezogen werden. Das Datenprogramm ist somit zum Datenträger für verschiedenfarbige Einsätze geworden.

Nachdem nun die einzelnen Kostenarten so verteilt sind, dass jeweils zwei von ihnen zusammen in Form von Verrechnungsdatenpaaren in einem Datenspeicher abgestellt sind, müssen im Falle einer Stücklistenauflösung auch die Massensummen der verschiedenen Kostenarten paarweise abgespeichert werden. Das bedeutet aber, dass diese Summen ohne Berücksichtigung der jeweiligen Stellung des Dezimalpunkts bei maximal 10-stelliger Anzeige nicht grösser als 99999 sein dürfen, ohne dass sie sich gegenseitig beeinflussen. Im folgenden Programmbeispiel für einen Operationsteil einer Stücklistenauflösung wählen wir die Summengrösse je Kostenart mit 9999,9. Nun haben wir aus Platzgründen auf die Kontrolle der Summengrösse verzichtet; an sich ein gefährlicher Umstand, denn jede Ueberschreitung dieser Grenze bedeutet im allgemeinen für beide Speicherteile, dass die darin enthaltenen Summen unbrauchbar geworden sind. Eine Abprüfung derselben müsste automatisch zu einer vorzeitigen Summenlistung führen und damit zu einem Neubeginn der Verarbeitung. Man kann aber auch aus der Erfahrung abschätzen, in welchen Grössenordnungen sich solche Sum-

menbildungen abspielen und aus dieser Ueberlegung heraus bewusst vorzeitige Summenabrufe von aussen und nicht vom Programm aus vorsehen. Ausserhalb der folgenden Programmdiskussion wollen wir aber dennoch die Konsequenzen einer solchen Massnahme des Einbaues einer Summenkontrolle erörtern.

Der Steuerenteil für eine Stücklistenauflösung unter Benutzung der bereits besprochenen Datenprogramme muss folgende Funktionen erfüllen:

- Steuerprogramm S 006 MATAUF einlesen
- Programm mit Betätigung der Taste E starten
- Tasten (f) (merge) drücken; Anzeige Crd
- Datenprogramm S 004 PRODAT ANR einlesen
- Stücklistenauflösung mit Betätigung der R/S-Taste starten
- bei Anhalten des Rechners ist für den betreffenden Artikel eine Mengeneingabe vorzunehmen und die Verarbeitung mit der Betätigung der R/S-Taste weiterzuführen
- werden weitere Artikel aufgelöst, ist die Verarbeitung mit dem Drücken der Tasten (f) (merge) bei nachfolgender Eingabe des nächsten Datenprogrammes fortzusetzen
- sind keine weiteren Artikel in der genannten Weise zu verarbeiten, ist der Summenabruf mit dem Drücken der Taste (A) einzuleiten. Ausgedruckt werden vor einer Summe die dieser Kostenart zugehörige Schlüssel. Nicht benötigte Kostenarten werden bei der Summenlistung übergangen, es erfolgen also keine Null-Ausdrucke.

## PROGRAMMTEILE

Die Gestaltung der Unterprogramme im Steuerenteil einer Stücklistenauflösung wird analog zum Steuerenteil

001	LBL A	Auflistung der Massengesamtsummen der einzelnen Kostenarten
002	2	
003	STO I	Index auf ersten Doppelsummenspeicher einstellen
004	LBL a	Beginn der Auflistungsschleife
005	.	
006	1	
007	+	Kostenartenschlüssel II.1
008	RCL(i)	Indirekter Abruf Speicherinhalt
009	INT	Isolierung des linken Speicherteils (1)
010	GSB c	Ausdruck Kostenartenschlüssel und Ausdruck der kostenartenbezogenen Massengesamtsumme
011	RCL I	Indexabruf
012	.	
013	2	
014	+	Kostenartenschlüssel II.2
015	RCL(i)	Indirekter Abruf Speicherinhalt
016	FRAC	Isolierung des rechten Speicherteils (2)

für die Kalkulation vorgenommen und wie folgt dargestellt auszusehen haben:

- LBL 0: Ausdruck der Artikelnummer und Eingabe der Menge mit nachfolgendem Kontrollausdruck dieser eingegebenen Zahl
- LBL 1 und LBL 2: Berechnung der Einzelmassen je Kostenart und Bildung der jeweils neuen Gesamt-massenzwischensummen im entsprechenden Speicherteil eines Datenregisters
- LBL 6 und LBL 8: Diese Unterprogramme werden hier nicht benötigt und werden daher als sogenannte Leerroutinen gestaltet. Man könnte diese Programmteile auch sinn-gemäss als MOP-Routinen (nop = no operation) bezeichnen
- LBL 9: Ebenso wird die Teilung einer Zahl durch 100 benötigt
- LBL E: Auch hier ist analog zum Steuerenteil für die Kalkulation der Finstieg in eine folgende Verarbeitung für das Einlesen des betreffenden Datenprogramms entsprechend zu gestalten
- LBL A: Zusätzlich ist aber im Falle einer Stücklistenauflösung, bei der im Bereich der Massen der Ausdruck der Summen mit einer Dezimalstelle erfolgt, ein entsprechender Summenabruf zu programmieren. Den einzelnen Summen geht der Ausdruck der zugehörigen Kostenartenschlüssel voran; nicht in Anspruch genommene Kostenarten werden beim Ausdruck nicht berücksichtigt
- LBL c: Für die Summenlistung ist die Einrichtung einer eigenen Druckroutine erforderlich

## PROGRAMMLISTUNG

In der nachfolgenden Programmlistung ist wie bereits erwähnt eine allfällige Summenkontrolle unberücksichtigt geblieben; eine solche wird nach der Programmlistung erörtert werden.

# PPC/HHC - Die Programmierbaren

017 EEX  
 018 5  
 019 \*  
 020 GSB c Zum Ausdruck Kostenartenschlüssel und Ausdruck der kostenartenbezogenen Massengesamtsumme  
 021 ISZ I  
 022 1  
 023 9 Indexgrenzwert  
 024 RCL I Abruf aktueller Speicherindex  
 025  $x \leq y?$   
 026 GTO a Sprung zurück in die Schleife, wenn der obere Grenzwert des Speicherindex noch nicht erreicht ist  
 027 CL REG  
 028 P<>S  
 029 CL REG Löschen sämtlicher Summenspeicher für neue Rechnung  
 030 GTO E Beginn einer neuen Auflösung mit Einlesen eines weiteren Datenprogramms  
 031 LBL c Druckroutine  
 032  $x = 0?$   
 033 RTN Kein Ausdruck von Null-Ergebnissen  
 034 EEX  
 035 1  
 036 : Alle Summen werden mit einer Dezimalstelle ausgegeben  
 037  $x < y$   
 038 SPC Zeilenvorschub  
 039 DSP 1  
 040 PRT X Ausdruck des Kostenartenschlüssels  
 041  $x < y$   
 042 PRT X Ausdruck der kostenartenbezogenen Massengesamtsumme  
 043 RTN Ende des Unterprogramms für den Ergebnisausdruck  
 044 LBL 0  
 045 SPC  
 046 PRT X Ausdruck der dezimalstellenlosen Artikelnummer  
 047 R/S Stop zur Mengeneingabe  
 048 DSP 2  
 049 PRT X Kontrollausdruck dieser Menge mit zwei Dezimalstellen  
 050 STO 1 Sicherstellung der eingegebenen Menge  
 051 1  
 052 STO I Der Index ist mit der Zahl 1 zu setzen, da bei der ersten Verarbeitung nach einer Eingabe im mit dem LBL 1 eingeleiteten Unterprogramm eine automatische Indexerhöhung eingebaut ist  
 053 GTO 6 Dieser Sprung nach Schritt 074 ist deswegen erforderlich, wenn nach der Kostenart 2.1 kein Stammwert vorliegt und die erste Auflösung daher mit der Zahl 0 zu erfolgen hat  
 054 LBL 1 Unterprogramm für die Auflösung eines Wertes, dessen Einzelmassensumme im linken Teil eines Datenregisters abgespeichert wird  
 055 SF 2 Steuerflag Speicherteil 1 links  
 056 ISZ I  
 057 LBL 2 Beginn des Unterprogramms für die Auflösung eines Wertes, dessen Einzelmassensumme im rechten Teil eines Datenregisters abgespeichert wird  
 058 F 0? Für einige Kostenarten kann die Summenbildung ausgesetzt werden  
 059 RTN Ist der Auflösungsflag 0 gesetzt, wird für die augenblicklich eingestellte Kostenart keine Summenbildung vorgenommen und in das Datenprogramm zurückgesprungen  
 060 RCL 1 Abruf der eingegebenen Menge  
 061 \* Berechnung der Einzelmassensumme  
 062 DSP 1  
 063 RND Rundung auf eine Dezimalstelle, um Summenüberschneidungen in den Datenspeichern zu verhindern  
 064 EEX  
 065 1  
 066 \*  
 067 F 2?  
 068 GTO 3 Abspeicherung der Einzelmassensumme im linken Teil eines Datenregisters  
 069 EEX  
 070 5

071 : Abspeicherung der Einzelmassensumme im rechten Teil eines Datenregisters  
 072 LBL 3  
 073 STO+(i) Indirekte Aufsummierung  
 074 LBL 6  
 075 0 Einsparung einer Leermeldung im Datenprogramm  
 076 LBL 8 Leerroutine wegen Steueranteil Kalkulation erforderlich  
 077 RTN Gemeinsames Ende der mit LBL 1, LBL 2, LBL 6 und LBL 8 eingeleiteten Unterprogramme  
 078 LBL 9 Teilung eines Wertes durch die Zahl 100  
 079 EEX  
 080 2  
 081 :  
 082 RTN  
 083 LBL Z Startroutine zum Overlay des nächsten Datenprogramms nach Betätigung der Tasten (f) (merge)  
 084 DSP 0  
 085 GTO E Diese Anweisung wird im Falle des Einlesens eines neuen Datenprogramms von demselben überschrieben

Nun wollen wir aber auch noch die Konsequenzen durchdenken, die im Zusammenhang mit dem Einbau einer Abprüfung gegen Ueberschreitung einer Massensumme über den Grenzwert 99999 zu ziehen sind. Ohne Berücksichtigung der augenblicklichen Stellung des Dezimalpunktes war klar, dass bei Vorhandensein einer maximal 10-stelligen Anzeige eine Teilgesamtsumme 5-stellig sein muss und ohne Beeinträchtigung des Nachbarwerts den mit 99999 angegebenen Grenzwert nicht überschreiten darf. Bevor nun eine neue Einzelmassensumme zur Bildung der neuen Zwischengesamtsumme herangezogen werden darf, ist zu prüfen, ob durch diese Operation eine Ueberschreitung des oberen Grenzwerts erfolgt. Ist dies nicht zu erwarten, kann die Aufrechnung erfolgen und programmgemäß fortgesetzt werden.

Wenn aber der Grenzwert von 99999 auch nur um 1 überschritten wurde, muss etwas geschehen. Den Gedanken, eine generelle Summenlistung anzusteuern, müssen wir nach einiger Ueberlegung fallen lassen, weil es ja nicht sicher ist, dass die Grenzwertüberschreitung schon bei der ersten Kostenart erfolgt. Bei später folgenden Kostenarten müsste sich der Rechner merken, ab welcher

# PPC/HHC - Die Programmierbaren

Position, bzw. ab welchem Kostenartenschlüssel die Verarbeitung fortzusetzen ist. Wir wählen aber einen anderen, weniger komplizierten Weg, weil auch das Anhalten der Verarbeitung zum Zweck einer Entscheidung über den weiteren Fortgang über das Tastenfeld Probleme mit sich bringt.

## FAKULTATIVE EINSCHUEBE

Wir wählen eine dritte Möglichkeit und wollen jene Summe samt dem dazugehörigen Kostenartenschlüssel ausdrucken, für den sich die Grenzwertüberschreitung ergeben hat und setzen daraufhin den in Frage kommenden Datenspeicherteil auf den Inhalt 0, um damit für diese Kostenart eine neue Summenbildung zu ermöglichen und so die Verarbeitung fortsetzen zu können. Mit Sicherheit lässt sich schon jetzt sagen, dass eine diesen Umstand entsprechende Abänderung des vorliegenden Programmes zu viele Speicherplätze kosten wird und daher auf diesem Rechnertyp nicht zu realisieren ist. Trotzdem wird diese Lösungsmöglichkeit diskutiert, um auch Lesern mit anderen Rechnern einen Hinweis auf eine problemanalytische Optimierung geben zu können.

Um diese Kontrolle der Grössenordnung einer Massenzwischensumme im Programm vorzusehen, wären folgende Programmeinschübe vorzunehmen:

- Zwischen den Programmzeilen 066 und 67:  
RCL (i) Abruf des Inhaltes eines Doppelsummenspeichers
- Zwischen den Programmzeilen 068 und 069:

FRAC	Isolierung des rechten Speicherteiles	5	Erzeugung des Grenzwertes mit 100000
STO-(i)	Löschen des entsprechenden Teilsummenspeichers	$x > y?$ GTO 7	Ist die neue Summe noch kleiner 100000, kann die Verarbeitung wie bisher fortgesetzt werden
EEX 5 *	Die Massenzwischensumme wird ganzzahlig gemacht	R↓	Umspeicherung im Stack; die neue Massenzwischensumme gelangt in die Anzeige
2	Erzeugung des Speicherteilindex für den rechten Speicherteil	X<>Y R↓	Durch Umspeicherungen wird im Stack jene Speicherordnung hergestellt, die für einen Summenausdruck im mit dem LBL c eingeleiteten Unterprogramm erforderlich ist
GSB 7	Kontrolle auf allfällige Summenüberschreitung	GSB c	Ausdruck der letzten Massenzwischensumme grösser oder gleich 100000
- Zwischen den Programmzeilen 071 und 072: GTO 4	Sprung zur gemeinsamen Fortsetzung	CL X	Die Anzeige wird gelöscht, um die Löschung der Teilsummenspeicher auch optisch anzuzeigen
- Zwischen den Programmzeilen 072 und 073: INT	Isolierung des linken Speicherteils	RTN	Ende des Unterprogrammes bei Auftreten einer Grenzwertüberschreitung
STO-(i)	Löschen des entsprechenden Teilsummenspeichers	LBL 7 R↓	Die neue Massenzwischensumme kleiner gelangt wieder in die Anzeige
1	Erzeugung des Speicherteilindex für den linken Speicherteil	RTN	Ende des Unterprogrammes für den Fall, dass keine Grenzwertüberschreitung zu verzeichnen war
GSB 7	Kontrolle auf allfällige Summenüberschreitung		Nun würde der Einbau dieser Programmänderungen bedeuten, dass unser verfügbarer Programmspeicher für die Datenprogramme um 34 Zeilen verkürzt werden würde, was im Falle, dass wir den HP 97, bzw. HP 67 verwenden, nicht akzeptabel wäre. Mit einer Anpassung des Programms auf die eigenen Bedürfnisse oder der manuellen Durchführung einiger Teile wird dieser Platz aber zu sparen sein.
LBL 4	Gemeinsame Fortsetzung		
- Das mit dem LBL 7 einzuleitende Unterprogramm wird am vorteilhaftesten zwischen die Programmzeilen 077 und 078 eingeschoben: LBL 7	Unterprogramm zur Kontrolle einer allfälligen Ueberschreitung des Grenzwertes 99999 für eine Massenzwischensumme		
RCL I +	Abruf des Speicherindex Bildung des Kostenartenschlüssels		
R↓ +	Berechnung der neuen Massenzwischensumme		
EEX			



Drucksachen jeder Art  
und verschiedener Ausführung!  
Unionsdruckerei Luzern  
Kellenstrasse 6, Telefon 44 24 44



## Befehlsstruktur im HP-41C/CV

Erwin GOSTELI, El. Ing.

### 5. Teil, Schluss

Wir beschliessen unsere Serie über das Synthetische Programmieren mit einer Diskussion über die HP-41 Befehlsstruktur. Inhaltlich gehört der folgende Artikel eigentlich an den Anfang einer Serie. Es ging unserem Autor jedoch darum, den Lesern möglichst bald Werkzeuge für die Erzeugung Synthetischer Befehle zu vermitteln. Also: Zum Schluss jetzt noch etwas trockene Theorie.

Normalerweise sieht der Programmierer des HP-41 keinen Anlass, sich ernsthaft mit der Befehlsstruktur auseinanderzusetzen. Der synthetische Programmierer jedoch oder derjenige, der ein Computerprogramm zur Erzeugung von Barcodes schreiben möchte, hat ein vitales Interesse daran. Letztlich verhilft es in gewissen Fällen auch zu einer Erhöhung der Byteökonomie, ein nicht zu unterschätzender Vorteil in einem so begrenzten Arbeitsspeicher wie ihn der HP-41 - trotz des Extensions-Modules - aufweist.

Die HP-41 Codetabelle (MIKRO-UND KLEINCOMPUTER 81-6) stellt die Basis für die nachfolgenden Ausführungen dar. In dieser Tabelle sind die Bytewerte gemäss ihrer Aufteilung in zwei Nibbles matrixförmig in Spalten und Zeilen angeordnet. Die 16 Zeilen sind jeweils mit den höherwertigen, die 16 Spalten mit den niederwertigen Nibbles (Hex 0...F) bezeichnet. So hat die Funktion % den Bytewert Hex 4C und befindet sich somit im Schnittpunkt von Zeile 4 und Spalte C.

Der HP-41 Befehlssatz setzt sich aus Ein-, Zwei- und Mehrbyte-Instruktionen zusammen. In der Codetabelle ist die Befehlsgruppen-Aufteilung am rechten Rand angegeben.

### EINBYTE-BEFEHLE

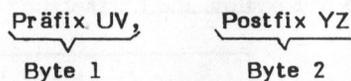
Allen Einbyte-Befehlen sind die Werte Hex 00...8F zugeordnet. Dabei zeigt eine genauere Betrachtung, dass Befehle, die maschinenintern gleiche oder ähnliche Verarbeitungszyklen beinhalten müssen, jeweils in einer Zeile mit dem gleichen höherwertigen Nibble zusammengefasst worden sind. Eine solche Organisation spart ROM-Speicherplatz und erhöht die Verarbeitungsgeschwindigkeit.

Der Befehl Hex 00, das Nullbyte, ist nicht direkt als solches im Programmmodus eingebbar, es ergibt

sich aber beim Programmeditieren. Jedesmal, wenn ein eingegebener Befehl gelöscht wird (DEL), wird der Platz der gelöschten Instruktion durch Nullbyte(s) überschrieben. Packen lässt diese Nullbytes wieder verschwinden, indem nachfolgende Programminstruktionen in diesen durch die Nullbytes geschaffenen Leerraum geschoben werden.

### ZWEI-BYTE-BEFEHLE

Diese Instruktionen sind folgendermassen strukturiert:



Mit Ausnahme der lokalen Zweibyte-Sprungbefehle, definiert das Präfix UV die auszuführende Operation, das Postfix YZ ist durch das Funktionsargument gegeben. Für letzteres gilt die Konvention, dass Postfixwerte aus der oberen Hälfte der Codetabelle direkte Argumente, solche aus der unteren Tabellenhälfte die indirekten Argumente darstellen. So weist z.B. der direkte Registerspeicherbefehl STO 27 den Wert Hex 91 1B auf, das indirekte Abspeichern über Register 27, STO IND 27, ist jedoch durch Hex 91 9B gegeben. Eine Ausnahme bilden die beiden indirekten Befehle GTO IND... und XEQ IND..., die beide das gleiche Präfix-Byte AE haben. Die Unterscheidung erfolgt durch Wahl des Postfixes und zwar so, dass ein solches aus der oberen Hälfte (Hex 00-7F) ein indirekter Sprungbefehl, ein Wert aus dem Bereich (Hex 80-FF) hingegen ein indirekter Subroutinenaufruf bedeutet. (Vgl. GTO IND 2 = HEX AE 3E und XEQ IND 62 = HEX AE BE.)

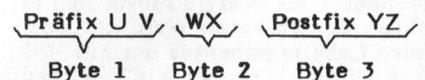
Die Präfixbytes für die meisten Zweibyte-Instruktionen stammen aus den Zeilen 9 und A. Eine Ausnahme bilden die Vertauschfunktionen X< und die lokalen Labels; ihre Präfixe sind in Zeile Hex C zu finden.

Hex 00 bis 63 als Postfix werden vom Display und Drucker als äquivalente dezimale Argumente 00 bis 99 angezeigt. Das gleiche gilt für die indirekten Argumente Hex 80 bis E3. Postfixe, in deren Fortsetzung in Zeile 6 (bis 6F) und in Zeile E (bis E7) bedeuten die dezimalen Argumente 100, 101, 102...111. Diese werden als 00, 01, A...J angezeigt und gedruckt. Entsprechende Registerbefehle wie STO+..., ISG... usw. führt der Rechner in der erwarteten Art aus. Es sind also nicht nur die nach Handbuch möglichen 100 Datenregister direkt ansteuerbar, sondern durch synthetische Mittel deren 112! Postfixe aus Zeile 7 ergeben die bekannten Statusbefehlsargumente.

Für die Displayformatierbefehle FIX, ENG, SCI sowie für den Tonbefehl TONE ergeben sich in Verbindung mit unerlaubten, synthetisch aber erzeugbaren Argumenten, interessante Befehle, wenn auch mit unvorhergesehenen Eigenschaften. Diese sind nur durch genaues Beobachten und Experimentieren erfass- und dokumentierbar. So toggelt z.B. ein FIX e die Flags von No. 43 abwärts in binärer Zählweise bis hinunter in den Bereich der anwenderzugänglichen Flagpositionen. Ton-Befehle mit beliebigen Postfix ergeben neue Tonfrequenzen, aber auch Töne unterschiedlicher Dauer.

### DREI-BYTE-BEFEHLE

Diese Instruktionen zeigen folgenden Aufbau:



Die sog. lokalen "Langform"-Sprungbefehle GTO 15-99 sowie lokale Subroutinenaufrufe (XEQ 00-99) und globale END's sind alle aus drei Bytes gemäss obigem Schema aufgebaut. Desgleichen gilt für die lokalen Sprung- und Subroutinen-Auf-

rufbefehle inkl. die mit synthetischen Postfixen aus Zeilen 6 und 7 (z.B. GTO P). Im Gegensatz zu den vorgehenden Zweibyte-Instruktionen weisen diese Befehle wegen ihrer reduzierten Vielfalt ein Präfix, bestehend aus nur einem Nibble auf. In den Nibbles V, W, X setzt der Prozessor nach dem Adressen-Compilervorgang die Sprungdistanz zum adressierten Label ein, beim globalen END die Sprungdistanz zum nächsten globalen Befehl (GTO oder END) weiter oben im Speicher. Das Argument des anzuspringenden Labels bzw. der Programmfilestatus im Falle eines END sind im Postfix YZ enthalten.

## MEHR-BYTE-BEFEHLE

Diese Klasse umfasst alle globalen Marken (Zeile C), Sprungbefehle, z.B. (GTO "ABC", Zeile D) aber auch globale Subroutineanrufe (Zeile E) und die Zeichenketten (Zeile F).

Wie schon in der ersten Folge erwähnt, besteht jeder String von n Zeichen aus n Bytes sowie einem Kopfbyte Hex Fn. Das erste Nibble F besagt, dass die nachfolgenden Bytes als Zeichen zu behandeln sind, das zweite Nibble n gibt die Länge dieser Kette an. Da n nur die Werte Hex 0-F durchlaufen kann, können Programmtextzeilen höchstens 16 Charaktere aufnehmen.

## LOKALE LABELS

Man unterscheidet zwischen lokalen und globalen Labels (Marken). Lokale Labels sind nur innerhalb eines globalen Programms aufrufbar. Lokale Labels werden weiter unterschieden in Kurzform- (ein Byte) und Langform- (zwei Bytes) Labels. Das erste höherwertige Nibble eines Kurzformlabels ist immer Hex 0, das zweite (niederwertige) Nibble weist den äquivalenten hexadezimalen Wert des Labelarguments +1 an: Label 00 → Hex 01, Label 14 → Hex 0F, etc.

Das Präfix-Byte eines Langform-Labels ist mit dem Wert CF gekennzeichnet. Das Postfix nimmt den direkten äquivalenten Wert des dezimalen Labelargumentes an: z.B. LBL 15 → CF 0F; LBL 99 → CF 63 etc. Die direkte Fortsetzung bis Hex CF 7F ergibt die lokalen und über die oberen beiden Tastenreihen direkt aufrufbaren Alpha-Labels LBL A - LBL J, LBLa - LBLj oder die synthetischen Labels wie LBL M oder LBL Postfixe von Hex 80-FF ergeben indirekte Labels, z.B. CF AB → LBL

IND43. Diese sind aber nur unter ganz bestimmten Umständen aufrufbar.

## GLOBALE LABELS /END

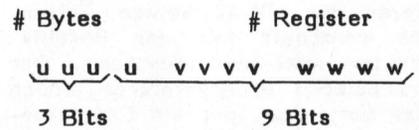
Globale Labels und END's sind untereinander verkettet, indem in jede dieser Instruktionen die Sprungdistanz zum nächsten globalen Label oder END weiter oben im Speicher codiert ist. Ausgangspunkt dieser Kette ist das permanente .END., dessen absolute Registerposition in den drei niederwertigen Nibbles von Buchhaltungsregister c vorliegt. Die Suche nach einer globalen Instruktion fängt immer von diesem Ausgangspunkt an.

Der Aufbau eines globalen Labels ist

Hex Cu vw Fn+1 xy z1...zn  
 n-Bytes  
 (max 7)

Fn+1 xy z1...zn ist die den globalen Namen kennzeichnende Zeichenkette einschliesslich des Tastencodes xy für die Tastenzuordnung des Programmfiles. Als Tastencode wird der gleiche Aufbau wie in den Tastenzuweisungsregistern verwendet (MIKRO- UND KLEINCOMPUTER 82-3, Bild 3). Liegt keine Tastenzuordnung vor, so nimmt xy den Wert 00 an.

Die Nibbles uvw nehmen die Linkdistanz zum nächsten globalen Label oder END auf und zwar in folgender Codierung:



Die Linkdistanz ist jeweils von Anfang zu Anfang der verketteten globalen Labels oder END definiert. Die niederwertigen neun Bits nehmen die Distanz m, ausgedrückt in der

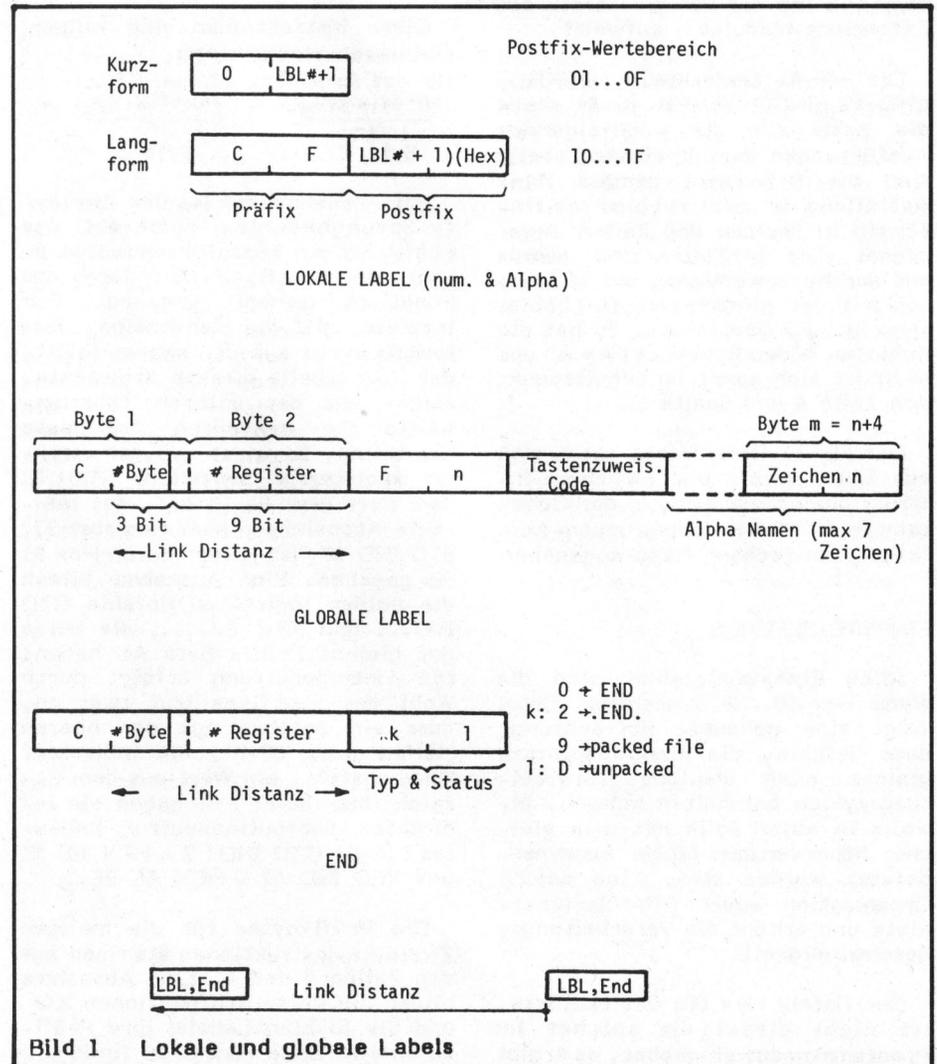


Bild 1 Lokale und globale Labels

Anzahl vollständig zu überspringender Register, die 3 höherwertigen Bits die Anzahl Bytes des letzten teilweise zu überspringenden Registers auf.

Die maximal codierbare Sprungweite beträgt  $7 \times 2^9 = 3584$  Bytes.

Die globalen ENDS sind ähnlich strukturiert, indem auch insgesamt drei Nibbles zur Aufnahme der Linkdistanz zur Verfügung stehen. Anstelle des Fn-Bytes tritt jedoch ein Byte kl zur Charakterisierung des END-Typs und des Programmfile-Status mit folgenden Werten:

k (Typ):	0	END
	2	.END.
l (Status):	9	File packed
	D	File unpacked

Wird während eines Redigiervorgangs ein Programmschritt eingefügt oder gelöscht, so ändert der END-Status in 0D bzw. 2D.

Globale Labels oder END können auch synthetisch erzeugt werden. Allerdings sind solcherart erzeugte Instruktionen nicht a priori Teil der globalen Kette. Ein PACK oder GTO muss diese Verkettung erzwingen. Bei der synthetischen Einfügung von END in Programmfile ist unbedingt darauf zu achten, dass nur der Typ des Programm-END und nicht der eines Schlussendes .END. erzeugt wird: (1) eine .END.-Instruktion innerhalb eines Programmfiles ist nicht löscherbar, (2) ist dieses .END. noch nicht Teil der globalen Kette und wird gepackt, so wird es zum neuen Schlussende und alle Instruktionen unterhalb dieses Schlussendes gehen verloren, d.h. sind nicht mehr zugänglich. Bild 1 zeigt die Strukturen der verschiedenen lokalen und globalen Labelinstruktionen.

## LOKALE GTO/XEQ

Kurzform-Sprungbefehle GTO 00 bis GTO 14 sind 2-Byte-Befehle mit dem Aufbau Bx yz. Kurzform-Subroutineaufrufe existieren keine. Nibble x bezeichnet das anzuspringende Label, die Nibbles yz nehmen die Sprungdistanz zum adressierten Label auf:

Sprungdistanz  $y_0$   $\frac{yyy}{\# \text{ Bytes}}$   $\frac{zzzz}{\# \text{ Reg}}$

$y_0$  ist das Richtungsbyte: Es nimmt bei einem Vorwärtssprung den

Wert  $y_0 = 0$  an, bei einem Rückwärtssprung wird es  $y_0 = 1$  gesetzt. Die Sprungdistanz ist definiert als die Distanz in Bytes vom Ende des GTO-Befehls bis zum Anfang des anzuspringenden Labels. Wie man sieht, lässt obiges Codierschema eine max. Sprungdistanz von  $\pm 2$  hoch 5 Bytes =  $\pm 112$  Bytes zu.

Langform-Sprungbefehle (inkl. lokale Alpha), aber auch lokale Subroutineaufrufe sind 3-Byte-Befehle und haben die Form Du vw xy bzw. Eu vw xy. Die Sprungdistanz ist in den Nibbles uvw codiert und zwar mit dem gleichen Codierschema wie für die globalen Labels. Das letzte Byte xy nimmt das Argument des anzuspringenden Labels und die Sprungrichtung zu diesem auf:

$x_0$   $\frac{xxx \ yyyy}{\text{Argument}}$

mit  $x_0 = 0$  für einen Vorwärts- und  $x_0 = 1$  für einen Rückwärtssprung. Die Sprungdistanz in Bytes ist definiert vom Ende des ersten Bytes Du bzw. Eu bis zum Anfang des anzuspringenden Labels. Mit sieben Bits für das Argument können  $2^7 = 128$  verschiedene Labels angesprungen werden, d.h. neben dem normalen numerischen von 00 bis 99, auch alle synthetischen!

Wird bei der Programmeingabe ein lokales GTO/XEQ erstmalig eingegeben, so nehmen die Distanz-Nibbles die Werte 0 an. Während des erstmaligen Programmablaufs sucht der Prozessor das im Argument des GTO/XEQ-Befehls angegebene Label und steckt nach dessen Finden die Sprungdistanz in die dazu vorgesehenen Nibblepositionen des aufrufenden Befehls (Adressen-Compilation). In weiteren Programmläufen werden dann die aufgerufenen Labels direkt angesprungen. Bild 2 zeigt zusammenfassend die verschiedenen Sprungbefehlsstrukturen.

## GLOBALE GTO/XEQ

Diese sind durch den Präfix Hex 1D (globales GTO) bzw. Hex 1E (globales XEQ) gekennzeichnet und haben den Aufbau:

GTO global: 1D Fn  $\frac{z1...zn}{n \text{ Zeichen}}$

XEQ global: 1E Fn  $\frac{z1...zn}{n \text{ Zeichen}}$

Fn ist das Kopfbyte des nachfolgenden n Zeichen (max. sieben) umfassenden globalen Programmnamens.

Globale Sprung- bzw. Subroutineaufrufbefehle compilieren die Sprungadresse nicht! Jedesmal, wenn ein globales Programmlabel aufgerufen wird, muss der Prozessor dieses über die globale Labelkette suchen. Sind viele kleine Programme im Arbeitsspeicher, so kann diese Suche u.U. sehr lange dauern. Es ist daher schlechte Programmierpraxis z.B. in einem iterativen Schleifenblock, ein globales Label anzuspringen oder aufzurufen.

## XROM

Werden Befehle eines peripheren ROM, z.B. des Druckers, bei eingestecktem ROM in den Programmspeicher eingegeben und anschließend das ROM entfernt, so zeigt bekanntlich der Rechner einen aufgerufenen Befehl als XROM U, V an. U identifiziert das den Befehl enthaltende ROM, V die in diesem ROM implementierte Funktion. Alle peripheren ROM-Befehle sind aus zwei Bytes aufgebaut, wobei der Präfix immer im Bereich Hex A0...A7 liegt. Jedem dieser Präfix sind 256 verschiedene Postfixe, entsprechend Hex 00...FF zuordbar. Insgesamt sind  $8 \times 256 = 2048$  verschiedene periphere Befehlscodes möglich.

Die angezeigten XROM-Zahlen U, V sind die direkte Interpretation der dem Nibble Hex A nachfolgenden 3 Nibbles (= 12 Bit). Der Prozessor unterteilt die 12 Bits in zwei Gruppen zu je 6 Bits und errechnet die dezimalen Äquivalente U, V dieser beiden Gruppen:

Präfix Postfix  
AX YY: 1010  $\frac{0XXX \ YY \ YY \ YYYY}{U \ V}$

X, Y = binäre Bits  
U = Modul Identifikation  
V = Funktionsidentifikation

So hat z.B. WRTP (Kassettenlaufwerk) den Befehlscode A7 17 und wird als XROM 28, 19 angezeigt:

WRTP = Hex A7 17 =  
Bin 1010 | 0111 00 | 01 0011 = XROM 28, 19  
U = 28 V = 19

Allgemein, wenn die periphere Funktion den Befehlscode Hex Am np aufweist, so errechnet sich U, V zu

$$U = 4 m_{(10)} + \text{Int } n_{(10)} / 4$$

$$V = 16 (n_{(10)} \text{ Mod } 4) + p_{(10)}$$

# PPC/HHC - Die Programmierbaren

Umgekehrt lassen sich  $m$ ,  $n$  und  $p$  aus  $U$ ,  $V$  berechnen zu

$$\begin{aligned} m_{(10)} &= \text{Int}(U/4) \\ n_{(10)} &= 4U \text{ Mod } 4 + \text{Int}(V/16) \\ p_{(10)} &= V \text{ Mod } 16 \end{aligned}$$

Mit obigen Beziehungen ist es ein leichtes, die Codes von ROM-Befehlen mit synthetischen Mitteln direkt in den Programmspeicher einzugeben, ohne dass das betreffende ROM zur Verfügung stehen müsste. Für eine Eingabe mittels "BL" (MIKRO- UND KLEINCOMPTUER 82-2) oder "MZ" (MIKRO- UND KLEINCOMPUTER 82-3) sind allerdings die dezimalen Äquivalente notwendig. Bezeichnet  $A$  das dezimale Präfix und  $B$  das dezimale Postfix des einzelnen Befehls, so berechnet sich

$$\begin{aligned} A &= 160 + \text{Int}(U/4) \\ B &= 64U \text{ Mod } 16 + V \end{aligned}$$

## PXROM

Bekanntlich stehen in den Tastenzuweisungsregistern für jede Befehlszuordnung zwei Bytepositionen zur Verfügung (MIKRO- UND KLEINCOMPUTER 82-3). Üblicherweise belegen jedoch nur periphere ROM-Befehle die beiden Bytepositionen. Für maschineninterne Befehle lässt der Rechner nur Einbyte-Funktionen oder die Präfixe von Zweibyte-Instruktionen zu. In den letzten beiden Fällen belegt der Prozessor die erste Byteposition mit dem Füllwert Hex 04.

Wird im USER-Modus eine belegte Taste gedrückt, so zeigt das Display kurz den Namen der zugeordneten maschineninternen Funktion oder bei eingestecktem ROM den peripheren Befehl an. Bei herausgezogenem ROM erfolgt die Kurzvoranzeige wiederum mittels XROM  $U$ ,  $V$ . Bei der Zuordnung synthetischer Befehle mit beliebigen Präfixen und Postfixen erfolgt die Voranzeige ebenfalls durch ein XROM. Im Unterschied zu wirklichen XROM's, die einen Präfix aus  $A0...A7$  voraussetzen, bezeichnen wir diese Voranzeige jedoch als PXROM (pseudo XROM). Offensichtlich überprüft der Prozessor nicht, ob das höherwertige Nibble des Präfixes Hex A aufweist. Er nimmt bei einem höherwertigen Präfix-Nibble Hex 0, einfach an, dass es sich um eine periphere Zweibyte-Instruktion handelt und berechnet aus den nachfolgenden Nibbles die Zahlen  $U$ ,  $V$  entsprechend.

Zweibyte-Befehle mit dem gleichen Postfix und mit Präfixen aus der gleichen Spalte ergeben die glei-

chen Werte für  $U$  und  $V$ . So sind z.B. die vorangezeigten PXROM's für ASTO 70 und FS?70 gleich und ergeben XROM 41,06.

Es kann nun ohne weiteres vorkommen, dass bei der synthetischen Erzeugung von Tastenzuordnungen XROM-Zahlen erzeugt werden, die mit solchen von existierenden ROM's übereinstimmen. In einem solchen Fall zeigt der Prozessor bei eingestecktem ROM die periphere Funktion mit ihren mnemotechnischen Namen an, der Prozessor führt aber - da das erste Nibble des Präfixes  $\neq$  Hex A

ist - die zugehörige maschineninterne Funktion aus. Als Beispiel sei der in der letzten Folge gewählte Bytemaskierer mit dem Bytecode Hex F7 8E erwähnt. Dieser Bytecode ergibt ein XROM 30, 14 und die Voranzeige erscheint bei eingestecktem Kartenleser als 7DSP2.

## SYNTHETISCHE TOENE

Das Tonpräfix ist Hex 9F. Es ist naheliegend, dieses Präfix mit allen möglichen Postfixen der oberen Hälfte der Codetabelle zu verbind-

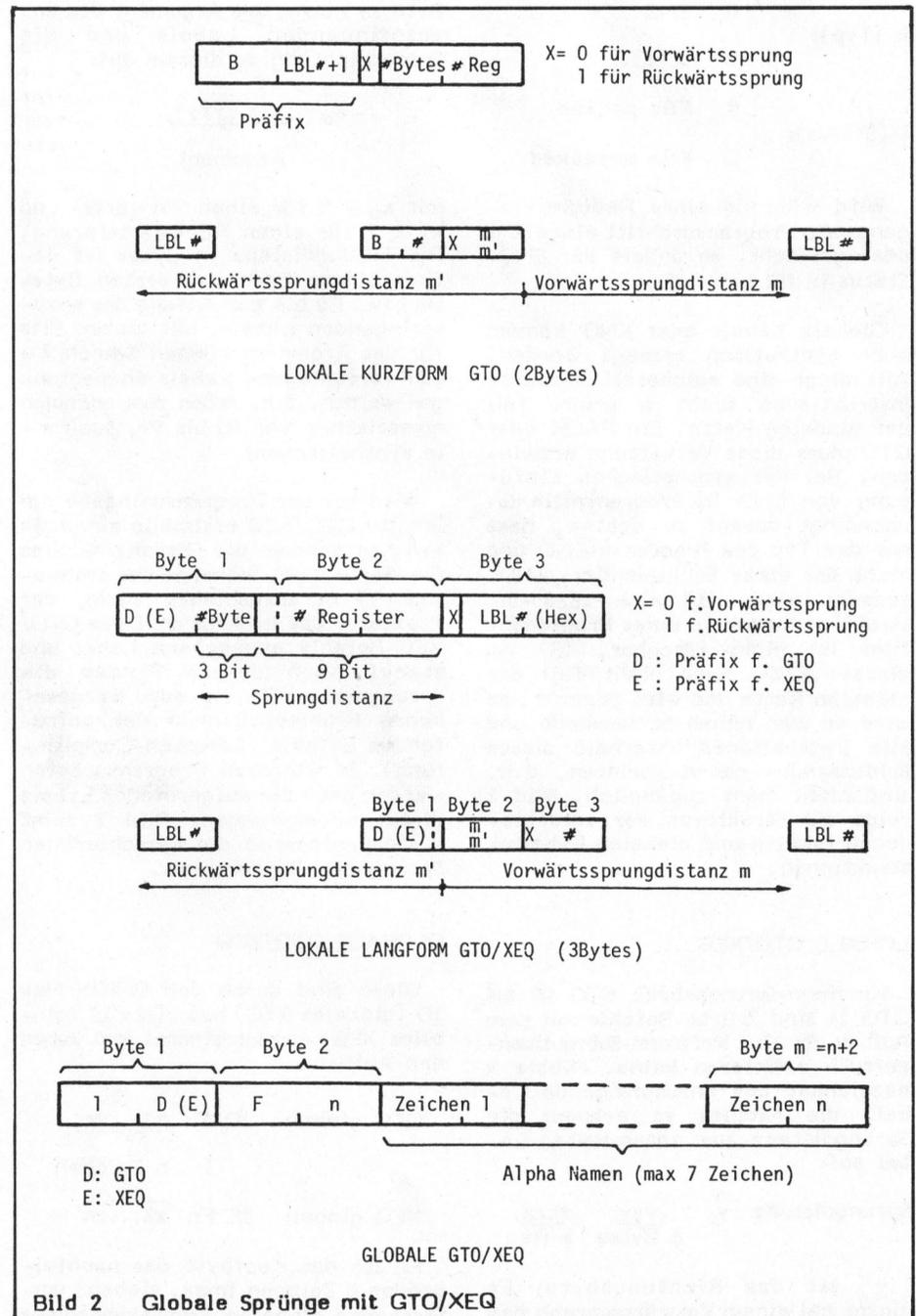


Bild 2 Globale Sprünge mit GTO/XEQ

# PPC/HHC - Die Programmierbaren

den. Die Erfahrung lehrt, dass damit dem HP-41 neue Töne entlockt werden können. Postfixe aus der gleichen Spalte der Codetabelle ergeben die gleichen Tonfrequenzen; die Tondauer ist jedoch verschieden und variiert von 0,023s (z.B. für die Postfixe Hex 26 bzw. 27) bis ca. 5s für Hex 1A. Es sind folgende Tonfrequenzen erzeugbar:

Niederwertiges Postfix		Niederwertiges Postfix	
Nibble Hex	Freq. (Hz)	Nibble Hex	Freq. (Hz)
0	175	8	788
1	197	9	1051
2	225	A	1051
3	263	B	113
4	315	C	121
5	394	D	131
6	525	E	143
7	629	F	158

} neu

Die Anzeige des TONE-Arguments im Display ergibt sich aus Arg. = Dez.Postfix Modulus 10 (z.B. Postfix Hex 4A = Dez 74 → TONE 4).

Von Hex 66 an werden die Argumente in bekannter Manier durch Buchstaben angezeigt.

Mit dem kurzen Programm "TO" können Sie die Musikalität Ihres Rechners testen. Geben Sie das dezimale Argument X des gewünschten Tones (128) ein, XEQ "TO" und der Ton ertönt. R/S erzeugt den nächsten Ton X+1. Das Programm erzeugt in Alpha eine Textzeile mit der gewünschten Ton-Präfix/Postfix-Kombination. Nach Setzen des Programmzeigers in das Alpha-Register wird dieser "Text" als Programminstruktion ausgeführt.

Damit schliesst die Artikelserie über das synthetische Programmieren ab. Diese Programmieretechnik wird auch nach Einführung des erweiterten Funktionenmoduls interessant bleiben. Mit ihrer Hilfe lassen sich weiterhin vertiefte Einblicke in die Arbeitsweise des HP-41 gewinnen, es lassen sich schnellere, Byte-sparendere und bedienerfreundlichere Programme erstellen.

## LISTINGS

```

01*LBL "TN"
02 128
03 MOD
04 ""
05 XTOA
06 "FZ"
07 ES
08 X<> b
09 SIGN
10 +
11 END

01*LBL "TO"
02 INT
03 128
04 MOD
05 ENTER↑
06 ""
07 LASTX
08 +
09 OCT
10 X<> d
11 FS?C 11
12 SF 12
13 FS?C 10
14 SF 11
15 FS?C 09
16 SF 10
17 FS? 07
18 SF 09
19 X<> d
20 X<> [
21 "F+"
22 STO \
23 "FZ"
24 CLX
25 E7
26 X<> b
27 SIGN
28 +
29 END

04 : F1 9F
06 : F3 7F CE 7C

06 : F1 9F
23 : F3 7F CE 7C
    
```

ALTRAC AG



Fr. 945.-

## DOT MATRIX PRINTER DP-8480

- Impact-Matrix, 9 x 7 Zeichen, 6 x 6 Graphik
- 80 Zeichen pro Sekunde
- Tractor & Friction Feed Type
- Centronics-Schnittstelle
- Option: RS 232-C / 20 mA current loop

Bezugsquellen-Nachweis auf Anfrage

ALTRAC AG    Zürcherstrasse 90  
CH-8953 DIETIKON  
Tel 01 / 741 46 44 Telex 56 169

ALTRAC AG

ALTRAC AG

## Neu! Das VT-100 compatible Videoterminal von Lear Siegler ADM-36



Einige technische Merkmale:

- RS-232, RS-422 und 20 mA Current Loop Schnittstelle
- Getrennte Tastatur
- Grosser 15"-Bildschirm
- 80 x 24 oder 132 x 24 Zeichen Bildschirmdarstellung
- Horizontal Split Screen
- Linien- und Punktgraphik (132 x 120)
- 8 umschaltbare Zeichengeneratoren
- Drucker-Ausgang
- Smooth oder Jump Scroll
- Video-Ausgang RS-170 kompatibel

## COMICRO AG

CH-8045 Zürich, Eichstrasse 24, Tel. 01 / 461 04 66, Telex 813617

Wehntalerstrasse 537  
 (Am Zehntenhausplatz)  
 8046 Zürich  
 Tel. 01-57 66 57

**micomp sms**

**BRANDNEU !**

**olivetti PRAXIS 30**  
 mit paralleler Schnittstelle



1'680.--

**DEMNÄCHST:** An alle Architekten und Bauunternehmer:  
 Wir sind gerade dabei, ein System zusammenzustellen, welches alle 184.000 Einträge im NPK des CRB jederzeit und ohne wechseln von Disketten für Sie zur Verfügung hält. Das komplette System wird unter 25'000 Franken kosten, inkl. 10MB Hard-Disk. Haben Sie Interesse, dann rufen Sie uns noch heute an.

**ITT 3030**  
 Im Architekturbüro

**Die Sensation auf dem Schweizer Markt**

**EUROCOMP  
 ALS  
 LADENKASSE**

In jedes Detailgeschäft

**DIE LEISTUNG:**

Erfassen der Kundenadressen mit Geburtsdaten, Quittungsausdruck, Lagerverwaltung. Artikel können direkt beim Quittungsausdruck abgerufen werden und werden automatisch aus dem Lager ausgebucht. Ausserdem Umsatzstatistik, Verkäuferstatistiken, Debitorenkontrolle, Kreditkartensystem .....

Oder wollen Sie vielleicht auch das altbewährte Rabattsystem wieder einführen, das bis jetzt so viel Mühe gemacht hat ?

Dieses Kassensystem, das bis jetzt nur grossen Ladenketten vorbehalten war gibt es jetzt auch für Klein-bis Mitteldetailisten. Bieten Sie Ihren Kunden den Service eines Grossen.

Preis für das komplette System : ab Fr. 23'865.--

**ITT 3030**  
 Im Treuhandbüro

Mit dem neuen ITT 3030 haben Sie endlich ein System, das mit den an Sie gestellten Anforderungen mitwächst. Heute verwalten Sie vielleicht nur 5 Kundenbuchhaltungen, aber morgen können es viele sein. Heute sind es vielleicht nur 4000 Buchungen, bald aber schon 15000 Buchungen im Jahr. Das einsatzfähige Basissystem kostet unter Fr. 9000.--



**FIBU II ms**

Ein Finanzbuchhaltungsprogramm, das die Probe schon lange bestanden hat.



Die einfache Handhabung dieses Programmes erlaubt es nun auch branchenfremden, die Buchhaltung im Haus zu erstellen. Eine Hilfsstufe kann jederzeit eingeschaltet werden und verschiedene Hilfsprogramme, die natürlich zum Programm gehören helfen eine fehlerfreie Buchhaltung zu erstellen.

Preis komplett: Fr. 1'295.--



Verzichten Sie nicht länger auf das wichtigste Informationsmittel Ihrer Firma !

**TRIUMPH ADLER  
 TYPENRADDRUCKER & SCHREIBMASCHINE**



Brandneu auf dem Schweizer Markt und bereits von Triumph Adler zugelassen ist dieses Interface, mit dem Sie aus Ihrer gewöhnlichen Schreibmaschine einen vollwertigen EDV-Drucker machen können.

**SEINE LEISTUNG:** Bidirektionaler Druck mit Druckwegoptimierung, Schreibgeschwindigkeit bis zu 25 Zeichen / Sek. Proportionalschrift und Schattenschrift je nach Modell und als Option. Plotten ebenso

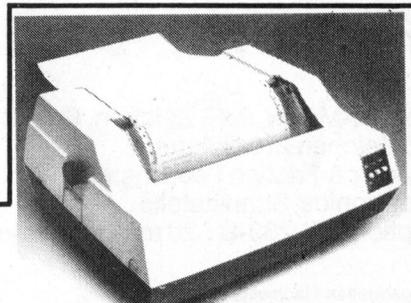
Preis: ab Fr. 3'495.--

Das Interface ist auch einzeln, zumnachträglichen Einbau, erhältlich.

**Matrixdrucker MP-125 von C.I.TOH**

- 3 Schriftarten - 125 Zeichen/Sek. - Traktorführung  
 Centronics Interface,

(nur solange Vorrat)



Unser Preisschlag Fr. 990.--

CP/M  
Software

neu: Fr. 1'295.--  
(komplett)

Regionalvertreter gesucht

# das neue Programm FiBu II<sup>ms</sup>

Jetzt gibt es endlich ein Buchhaltungsprogramm zu einem vernünftigen Preis, mit dem auch Sie ohne grosse Buchhaltungskennnisse arbeiten können. Jede Buchung wird auf ihre Richtigkeit hin überprüft, somit werden nur richtige Buchungen akzeptiert. Das Sortieren nach Datum übernimmt der Computer jetzt ebenfalls und ausserdem sagt er Ihnen, wo etwa Sie den zur Buchung gehörenden Beleg in Ihrem Ordner finden können.

Mit diesem Programm können auch Sie jetzt ein Informationsmittel im Haus haben, auf das man heute nicht mehr verzichten darf. Fast alle Fehler, die man beim Führen der Buchhaltung machen kann werden automatisch wieder behoben. Auch ein plötzlicher Stromausfall richtet keinen Schaden an, denn das Programm regeneriert sich von selbst.

Selbstverständlich ist dieses Programm mandatsfähig und bietet Ihnen völlige Flexibilität beim Kontenrahmen. Lieferbar ist dieses Programm für alle CP/M Rechner mit min. 200 KB Disk-Kapazität, wie z.B. ITT 3030, Superbrain, Sesam oder EUROCOMP.

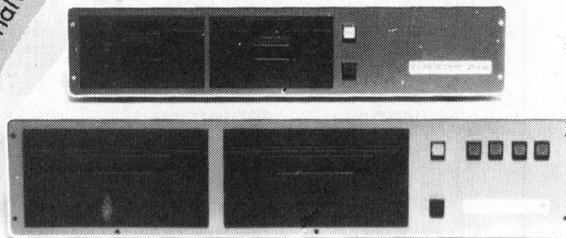
Kein Direktverkauf. Bezugsquellennachweis bei:

**EUCOTECH AG**

Tel.: 01-57.51.14

Regionalvertreter gesucht

# Multi-User



Die neuen EUROCOMP Systeme bieten Ihnen jetzt noch mehr Flexibilität und Leistungsreserven.

Mit dem neuen Modell 2008 haben Sie zum Beispiel die Möglichkeit bis zu 4 unabhängige Arbeitsplätze anzuschliessen, wobei selbst dann noch Verarbeitungsgeschwindigkeiten erreicht werden, die sich mit jedem 'Mini' vergleichen können.

Für die Speicherung Ihrer Daten stehen Massenspeicher von 2.4 / 4.8 / 20 oder 40 MB zur Verfügung. Damit kann man dann schon rund 400'000 Adressen verwalten.

Angetrieben wird das ganze von CP/M im Einplatz - betrieb oder vom sensationellen TURBODOS im Mehrplatzbetrieb, Turbodos gestattet die vollständige Übernahme aller CP/M Programme. Sicher.

Den 2008 gibt es bereits unter 14'000 Franken.

Kein Direktverkauf. Bezugsquellennachweis bei:

**EUCOTECH AG**

Tel.: 01-57.51.14

Wehntalerstrasse 537  
(Am Zehntenhausplatz)  
8046 Zürich  
Tel. 01-57 66 57

**micomp sms**

**OCCASION** ★ ★ ★  
ITT 2020  
1'920.--

**AUGUST-AKTION!**  
★ ★ ★ ★ ★

Disketten zu einem  
sensationalen Preis  
10 Stück 62.--  
double - density  
single - sided  
soft ★

## SINCLAIR

### Erweiterungen

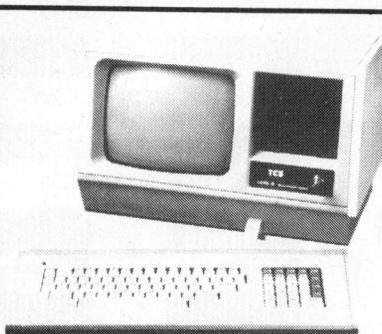
Normal-Tastatur	Fr.	295.--
64 KB RAM Erweiterung	Fr.	395.--
80 Zeichen Drucker	Fr.	1'295.--

### Neueste Software:

Textprogramm, Roulette, Invasion,  
Graphik-Hilfsprogramm, Laufschrift,  
Squash und vieles mehr.

### Grosse Bücherauswahl:

z.B. Programmiertricks, Umschreiben von  
ZX-80 Programmen auf ZX-81, Bücher  
mit Programm-Listings, technische Bücher...



## Genie III

Mit dem GENIE III gibt es wieder einen echten Preisbrecher bei den professionellen Kleinsystemen:

- 64 KB Arbeitsspeicher
- Z-80 CPU 4 Mhz
- grüner Bildschirm
- deutsche Tastatur
- 80 x 24 Zeichen
- Graphik 640x288 (Option)
- bis zu 1.4 MB Floppy

Überall einfach ideal einsetzbar, und das für weniger als 7'000 Franken, komplett.

## AKTUELLES

### Textprogramm für Video-Genie

Texteingabe, Korrektur und Umgestaltung des Textes ist selbst durch eine ungeübte Schreibkraft effektiv durchzuführen. Automatisches Übernehmen einer zu langen Zeile auf die nächste. Häufig verwendete Wendungen, z.B. "Mit freundlichen Grüßen" können durch Tastendruck aufgerufen werden, ebenso wie man Textblöcke beliebig verschieben kann. Verschiedene Schriften des Druckers lassen sich innerhalb des Textes umschalten. Direktes Unterstreichen möglich. Graphikzeichnungen können mit Text kombiniert werden.

Preis: nur Fr. 495.--

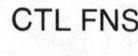
### SPIELE:

z.B. Spacewar, Flugsimulator, Steuern eines Rennwagen und vieles mehr.

### HP-82180A Extended Function/Memory Module

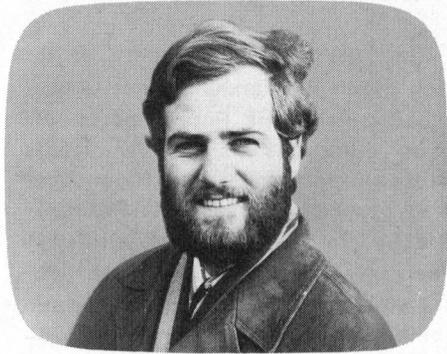
EXT FCN					
	CLFL	GETAS	INSCHR	RCLFLAG	SAVERX
					
ALENG	CLKEYS	GETKEY	INSREC	RCLPT	SAVEX
					
ANUM	CRFLAS	GETP	PASN	RCLPTA	SEEKPT
					
APPCHR	CRFLD	GETR	PCLPS	REGMOVE	SEEKPTA
					
APPREC	DELCHR	GETREC	POSA	REGSWAP	SIZE?
					
ARCLREC	DELREC	GETRX	POSFL	SAVEAS	STOFLAG
					
AROT	EMDIR	GETSUB	PSIZE	SAVEP	X<>F
					
ATOX	FLSIZE	GETX	PURFL	SAVER	XTOA

### HP-82160A HP-IL Interface Module

MASS ST					
	READP	SEEKR	WRTR	FINDID	OUTA
					
CREATE	READR	UNSEC	WRTRX	INA	PWRDN
					
DIR	READRX	VERIFY	WRTS	IND	PWRUP
					
NEWM	READS	WRTA	ZERO	INSTAT	REMOTE
					
PURGE	READSUB	WRTK		LISTEN	SELECT
					
READA	RENAME	WRTP	CTL FNS	LOCAL	STOPIO
					
READK	SEC	WRTPV	AUTOIO	MANIO	TRIGGER

### HP-82182A Time Module

TIME					
	ATIME24	CLOCK	DMY	RUNSW	STOPSW
					
ADATE	CLK12	CORRECT	DOW	SETAF	SW
					
ALMCAT	CLK24	DATE	MDY	SETDATE	T+X
					
ALMNOW	CLKT	DATE+	RCLAF	SETIME	TIME
					
ATIME	CLKTD	DDAYS	RCLSW	SETSW	XYZALM



## EPROMER-Programmiergerät

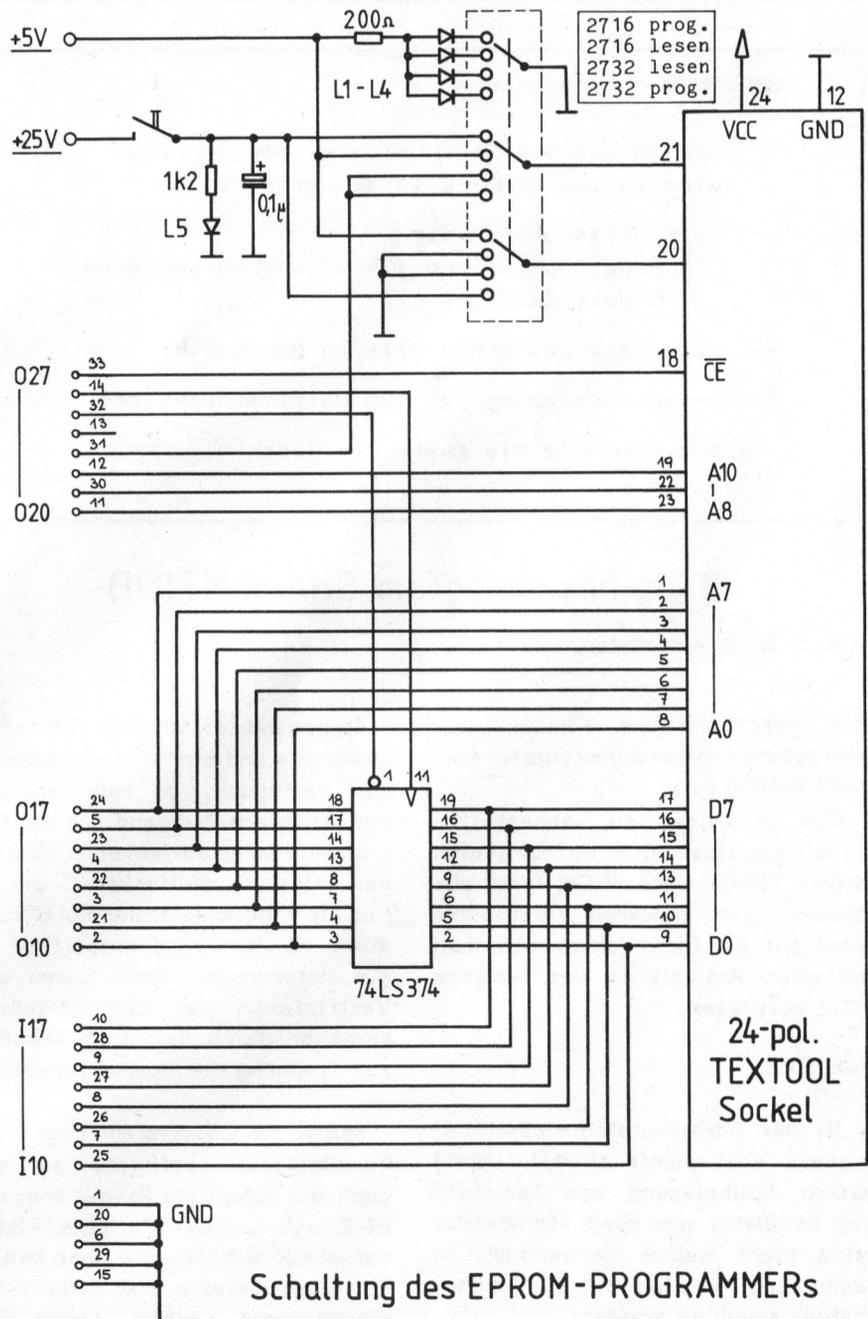
Leopold ASBÖCK

In MIKRO- UND KLEINCOMPUTER 82-2 wurden einige Schaltungsvarianten vorgestellt, die speziell in Z80-Systemen trotz Speichervollausbau von 64 KByte das Lesen von ROMs oder EPROMs erlauben. Bei Vorhandensein von Parallelschnittstellen beschränkt sich der Hardware-Aufwand auf einen 24-poligen Sockel. Heute stellen wir nun eine Schaltung vor, die durch Hinzufügen eines einzigen ICs ein komfortables Programmiergerät darstellt, das im Dialogbetrieb das Programmieren der EPROM-Typen 2716 und 2732 erlaubt.

Programmiergeräte sind eine teure Angelegenheit: Meist besitzen sie eine eigene Mikroprozessorsteuerung und verfügen über einen grossen Speicherbereich. Entweder werden im "stand alone"-Betrieb Daten eingegeben und hierauf in ein EPROM eingebrannt, oder ein Computer steuert im "slave"-Betrieb das Programmiergerät über eine RS232-Schnittstelle. Wenige Anweisungen und die Daten werden übertragen, dann besorgt das Programmiergerät die korrekte Programmierung des EPROMs oder PROMs. Geräte der oberen Preisklasse verfügen über austauschbare Module, während jene der mittleren Preisklasse mit Wahlschaltern das Programmieren der gebräuchlichsten Speicherschaltkreise erlauben.

In manche Kleincomputer können preisgünstige Platinen eingesteckt oder extern angeschlossen werden, die die Programmierung von EPROMs ermöglichen. Meist erstreckt sich die Typenwahl aber nur auf EPROMs mit 2 KByte oder 4 KByte Speicherinhalt wie 2716, 2516, 2732, 2532 oder ähnliche. Da diese Festwertpeicher aber recht häufig zu finden sind, leisten diese kleinen Zusatzplatinen wertvolle Dienste.

Ein besonderer Anreiz liegt darin, den Hardwareaufwand auf ein Minimum zu beschränken und die zur Programmierung wesentlichen Grundaufgaben der Software zu übertragen. Dies ist ja eine Grundidee der gesamten Mikroprozessortechnologie,



Schaltung des EPROM-PROGRAMMERS

## E P R O M - P r o g r a m m e r

- 1 - 2716 programmieren
- 2 -        verifizieren
- 3 -        einlesen
  
- 4 - 2732 programmieren
- 5 -        verifizieren
- 6 -        einlesen
  
- 9 - Programmende

Druecken Sie die gewünschte Taste !

## 2716 programmieren

Der Inhalt des Speicherbereiches von 7000 - 77FF wird in das EPROM 2716 gespeichert.

- Kontrollieren Sie die  
    Schalterstellung ( ♦ 2716 programmieren )  
    5 Volt Spannung
- Legen Sie das EPROM 2716 in den Sockel
- Programmiervoltage (+25 Volt) einschalten
- Druecken Sie die Taste P (oder M...Menu)

## Bildschirmdialog am SHARP MZ80B

Bild 2: Bildschirmdialog

die wesentlich zur Flexibilitätssteigerung prozessorgesteuerter Geräte beiträgt.

Die im folgenden vorgestellte Schaltung lässt sich auf beliebige andere EPROM- und PROM-Typen abändern. Zum besseren Verständnis wird auf ein Durchdenken des oben zitierten Artikels in der Ausgabe 82-2 verwiesen.

### DIE IDEE

Ist der Adressbereich eines Computers voll ausgeschöpft, etwa durch Vollbelegung des Speichers mit 64 KByte, oder durch Memorymapping nicht weiter verwertbar, so kann über Parallelports ein Adresszähler simuliert werden.

Stehen mindestens zwei 8-bit-Ausgabeports und ein 8-bit-Eingabeport zur Verfügung, so reduziert sich der Hardware-Aufwand für ein Programmiergerät auf einen IC 74LS374, der eine Doppelfunktion erfüllt: Für die Programmierung der EPROMs dient er als Zwischenspeicher für die Datenwerte, beim Lesen oder Verifizieren der EPROM-Inhalte sorgt er durch Tri-State-Ausgänge zur Trennung der Busrichtungen.

Sollte Ihr Z80-System über keine Parallelports verfügen, so kann nach der Schaltung in m+k computer 82-2 mit sieben ICs eine Platine aufgebaut werden, die über zwei 8-bit-Ausgangsports und zwei 8-Bit-Eingangsports verfügt. Ueber diese

Ports können Sie auch Drucker, Plotter oder andere periphere Geräte anschliessen.

Eine weitere Möglichkeit ergibt sich durch Verwendung eines Grossschaltkreises mit Parallelports wie etwa eines ICs 8255 (oder 9555). Diese 40-poligen ICs verfügen über drei programmierbare 8-bit-Parallelports. Je nach Bedarf sind die Ports als Ein- oder Ausgänge programmierbar. Meist sind jedoch Treiberstufen nachzuschalten, so dass sich keine Reduzierung des Hardware-Aufwandes ergibt.

### DIE AUSFUEHRUNG

Anschliessend wird eine konkrete Schaltungsvariante vorgestellt, die hard- und softwaremässig für den Personal Computer SHARP MZ80B mit Parallelportinterface MZ80I02 realisiert wurde, aber an jeden Kleincomputer mit dem Mikroprozessor Z80 übernommen werden kann. Es sind nur zwei Details zu beachten: Die Software wurde als Hybridprogramm abgefasst, das heisst der Dialogteil wurde in SHARP-BASIC geschrieben und kann mit geringfügigen Änderungen in jede andere BASIC-Version übertragen werden.

Der Maschinenprogrammteil kann für Z80-Systeme ungeändert übernommen werden, es ist nur darauf zu achten, dass die Zeitschleife "T50MS", die die Länge des Programmierimpulses (50 Millisekunden) bestimmt, für einen Systemtakt von 4 MHz (Z80A) ausgelegt ist. Im Assemblerlisting sind die Befehlsausführungszeiten (in Mikrosekunden) angegeben. Bei anderem Systemtakt sind diese Werte proportional und der Schleifenzählerwert in BC ist dementsprechend zu ändern.

Zu beachten sind noch die Portadressen (Output \$00, \$01, Input \$00). Das Maschinenprogramm benötigt weniger als 256 Bytes, zusätzlich werden 2 KByte bzw. 4 KByte als Speicher für die zu programmierenden Daten benötigt.



```

330 REM
340 GOSUB 400: IF T=9 GOTO 1340
350 REM
360 ON T GOTO 610, 860, 1110, 610, 860, 1110, 340, 340
370 REM
380 REM --- Hauptmenu anzeigen
390 REM
400 PRINT CHR$(6): C=20
410 PRINT TAB(C); STRING$( "█",37)
420 PRINT TAB(C); "      EPROM-Programmierer"
430 PRINT TAB(C); STRING$( "█",37): C=24
440 CURSOR C,6 : PRINT "1 - 2716 programmieren"
450 CURSOR C,8 : PRINT "2 -      verifizieren"
460 CURSOR C,10: PRINT "3 -      einlesen"
470 CURSOR C,13: PRINT "4 - 2732 programmieren"
480 CURSOR C,15: PRINT "5 -      verifizieren"
490 CURSOR C,17: PRINT "6 -      einlesen"
500 CURSOR C,20: PRINT "9 - Programmende": C=20
510 CURSOR C,23: PRINT "Drücken Sie die gewünschte Taste !";
520 GET T$: IF (T$<"1")+(T$<"9") GOTO 520
530 T= ASC(T$)-ASC("0")
540 IF T<4 THEN EPROM$="2716": E$="2716": A$="77FF": POKE LAENGE+1,$08
550 IF T>3 THEN EPROM$="2732": E$="2732": A$="7FFF": POKE LAENGE+1,$10
560 PRINT CHR$(6)
570 RETURN
580 REM
590 REM --- Programm `programmieren`
600 REM
610 POKE FAL123,1
620 CURSOR C,1 : PRINT EPROM$: " programmieren"
630 CURSOR C,2 : PRINT "-----"
640 CURSOR C,5 : PRINT "      Der Inhalt des Speicherbereiches von 7000 - "; A$
650 CURSOR C,7 : PRINT "      wird in das EPROM "; E$: " gespeichert."
660 CURSOR C,10: PRINT "---- Kontrollieren Sie die"
670 CURSOR C,12: PRINT "      Schalterstellung ( ♦ "; E$: " programmieren )"
680 CURSOR C,14: PRINT "      5 Volt Spannung"
690 CURSOR C,17: PRINT "---- Legen Sie das EPROM "; E$: " in den Sockel"
700 CURSOR C,20: PRINT "---- Programmspannung (+25 Volt) einschalten"
710 CURSOR C,23: PRINT "---- Drücken Sie die Taste P (oder M...Menu)"
720 GET T$: IF (T$<"P")*(T$<"M") GOTO 720
730 IF T$="M" GOTO 820
740 CURSOR C,23: PRINT "      EPROM";EPROM$;" wird programmiert !"
750 IF T=4 THEN POKE $6F30,$A0: POKE $6FB1,$BF: POKE $6FB9,$FF
760 USR(PSTART)
770 IF T=4 THEN POKE $6F30,$20: POKE $6FB1,$FF: POKE $6FB9,$BF
780 MUSIC "+G1": CURSOR C,17: PRINT STRING$("-",50)
790 CURSOR C,20: PRINT "      Programmierung fertig !"
800 CURSOR C,23: PRINT "      +25 Volt abschalten !" Drücken Sie M (Menu)"
810 GET T$: IF T$<"M" GOTO 810
820 GOTO 340: REM Menu
830 REM
840 REM --- Programm `verifizieren`
850 REM
860 POKE FAL123,2
870 CURSOR C,1 : PRINT EPROM$: " verifizieren"
880 CURSOR C,2 : PRINT "-----"
890 CURSOR C,6 : PRINT "      Der Inhalt des EPROMs ";E$;" wird mit dem Inhalt"
900 CURSOR C,8 : PRINT "      des Speicherbereiches 7000 - ";A$;" verglichen."
910 CURSOR C,12: PRINT "---- Kontrollieren Sie die"
920 CURSOR C,14: PRINT "      Schalterstellung ( ♦ "; E$: " lesen )"
930 CURSOR C,16: PRINT "      5 Volt Spannung"
940 CURSOR C,19: PRINT "---- Legen Sie das EPROM "; E$: " in den Sockel"
950 CURSOR C,23: PRINT "---- Drücken Sie die Taste V (oder M...Menu)"
960 GET T$: IF (T$<"V")*(T$<"M") GOTO 960
970 IF T$="M" GOTO 1070
980 CURSOR C,23: PRINT "      Verifizieren des EPROMs";EPROM$;"
990 REM
1000 USR(PSTART)
1010 REM
1020 A=PEEK(FEHLER+1): B=PEEK(FEHLER): F=256*A+B: MUSIC "+G1"
1030 CURSOR C,19: PRINT STRING$("-",50)
1040 CURSOR C,23: PRINT "      fertig !" "; F: " Fehler ";
1050 PRINT "---- Drücken Sie M (Menu)"
1060 GET T$: IF T$<"M" GOTO 1060
1070 GOTO 340: REM Menu
1080 REM
1090 REM --- Programm `einlesen`
1100 REM
1110 POKE FAL123,3
1120 CURSOR C,1 : PRINT EPROM$: " einlesen"
1130 CURSOR C,2 : PRINT "-----"
1140 CURSOR C,6 : PRINT "      Der Inhalt des EPROMs "; E$: " wird in den"
1150 CURSOR C,8 : PRINT "      Speicherbereich von 7000 - "; A$: " eingelesen."
1160 CURSOR C,12: PRINT "---- Kontrollieren Sie die"
1170 CURSOR C,14: PRINT "      Schalterstellung ( ♦ "; E$: " lesen )"
1180 CURSOR C,16: PRINT "      5 Volt Spannung

```

74LS374 ist ein 8-bit-Register mit Tri-State-Ausgängen. Beim Lesen oder Verifizieren eines EPROMs werden die Ausgänge inaktiviert (High-Z), indem über Output-Port 2 Bit 5 auf logisch 1 gelegt wird.

Beim Programmiervorgang dient dieses Register als Speicher für die zu programmierenden Datenbytes. Ein Low-high-low-Clock-Impuls über Bit 6 des Ports 2 bewirkt die Datenübernahme des Bytes, das über Port 1 ausgegeben wurde.

Bit 7 des Ports 2 ist für den Programmierimpuls verantwortlich. Die EPROM-Typen 2716 und 2732 sind (siehe Datenblatt) sehr einfach zu programmieren: wenn Adresse und Datenbyte anliegen und die Programmspannung von 25 Volt am entsprechenden Pin anliegt, wird ein TTL-Impuls von 50 Millisekunden Länge an Pin 18 ausgegeben. Die einzelnen Bytes können in aufsteigender oder in beliebiger Reihenfolge programmiert werden.

Falls man das Programmiergerät an die Parallelportplatine des SHARP MZ80B anschließen will, empfiehlt sich die Realisierung der Hardware in Form einer kleinen Platine, die über einen 37-poligen D-Stecker direkt an der Computerrückseite angesteckt wird. Führt man die Verdrahtung auf einer Lochrasterplatine in Wire-wrap-Technik aus, so ist der Aufwand recht gering, eine spätere Umverdrahtung für andere EPROM-Typen ist leicht ausführbar. Als EPROM-Sockel empfiehlt sich ein 24-poliger TEXTOL-Sockel oder ein gleichwertiges Fabrikat mit Nullkraftkontakten.

## DIE SOFTWARE

Das zur Programmierung gehörende Programm ist in zwei Abschnitte gegliedert: das Dialogprogramm, es ist in SHARP-BASIC abgefasst (Listung 1) und meldet sich mit einem Menu, das aus Bild 2 ersichtlich ist. Nach dem Drücken der entsprechenden Taste folgen die Anweisun-

gen zum Programmieren, Verifizieren (Vergleichen) oder Lesen eines EPROMs 2716 oder 2732.

Die Eingabe der Hexadezimalwerte, die in das EPROM geschrieben werden sollen, erfolgt mit dem Monitor in den Speicherbereich von 7000 bis 77FF (bzw. 7FFF). Natürlich besteht auch die Möglichkeit, ein kurzes BASIC-Unterprogramm für hexadezimal Dateneingabe anzufügen.

Der zweite Teil (im DATA-Bereich) enthält ein Z80-Maschinenprogramm (Listing 2), das vom BASIC-Programm geladen wird und durch USR(PSTART) von diesem aufgerufen wird.

Es führt in Sekundenbruchteilen das Einlesen oder Verifizieren eines EPROMs aus. Das Programmieren dauert rund zwei Minuten (EPROM 2716) oder vier Minuten (EPROM 2732), da pro Byte 50 Millisekunden Programmierdauer eingehalten werden müssen.

Dem Anwender steht es frei, das Programm zu ergänzen. Praktisch ist ein "empty check", das heisst, es wird überprüft, ob das EPROM vollständig gelöscht ist, ob alle Bytes auf hexadezimal FF gesetzt sind. Dies lässt sich auch erzielen, indem man alle Bytes des Bufferspeichers auf FF setzt und das EPROM verifiziert. Auch das Programmieren einzelner Bytes kann ergänzt werden, wird in den meisten Anwendungsfällen aber von untergeordneter Bedeutung sein.

Im vorliegenden Programm werden für die EPROMs 2716 oder 2732 folgende Befehle ausgeführt:

## PROGRAMMIEREN

Der Speicherinhalt von 7000 bis 77FF (bzw. 7FFF) wird in das EPROM geschrieben.

## VERIFIZIEREN

Der Inhalt des EPROMs wird mit dem Speicherinhalt von 7000 bis

```

1190 CURSOR C,19: PRINT "---- Legen Sie das EPROM "; E$; " in den Sockel"
1200 CURSOR C,23: PRINT "---- Druecken Sie die Taste L (oder M...Menu)"
1210 GET T$: IF (T$<>"L")*(T$<>"M") GOTO 1210
1220 IF T$="M" GOTO 1300
1230 REM
1240 USR(PSTART)
1250 REM
1260 MUSIC "+61": CURSOR C,19: PRINT STRING$("-",50)
1270 CURSOR C,23: PRINT "Eingabe EPROM";EPROM$;"Angelesen ";
1280 PRINT "Druecken Sie M (Menu)"
1290 GET T$: IF T$<>"M" GOTO 1290
1300 GOTO 340: REM Menu
1310 REM
1320 REM --- Programmende
1330 REM
1340 CURSOR C,1 :PRINT "Programmende"
1350 CURSOR C,2 :PRINT "-----"
1360 CURSOR C,5 :PRINT "Programmierspannung (+25 Volt) abschalten!"
1370 CURSOR C,10:PRINT "Auf Wiedersehen!": PRINT: PRINT
1380 LIMIT MAX : REM maximaler Speicherbereich
1390 END
1400 REM
1410 REM
1420 REM --- Maschinenprogramm laden (ab %6F00)
1430 REM Programmstart bei PSTART (= %6FOA)
1440 REM
1450 RESTORE 1630
1460 N=0
1470 READ U$: IF U$="FINITO" GOTO 1520
1480 GOSUB 1560: REM Hex-Dez umwandeln
1490 POKE BUFFER+N,U
1500 N=N+1
1510 GOTO 1470
1520 RETURN
1530 REM
1540 REM --- HEX-DEZ
1550 REM
1560 U1= ASC(LEFT$(U$,1))-48: IF U1>9 THEN U1=U1-7
1570 U2= ASC(RIGHT$(U$,1))-48: IF U2>9 THEN U2=U2-7
1580 U= 16*U1+U2
1590 RETURN
1600 REM
1610 REM --- Maschinenprogramm
1620 REM
1630 DATA 20,00,00,70,00,08,00,00, 00,00,E5,D5,C5,F5,CD,2F
1640 DATA 6F,2A,02,6F,ED,4B,04,6F, 3A,08,6F,FE,01,2B,1F,FE
1650 DATA 02,2B,3B,FE,03,2B,56,CD, 2F,6F,F1,C1,D1,E1,C9,3E
1660 DATA 20,32,00,6F,D3,01,3E,00, 32,01,6F,D3,00,C9,3A,00
1670 DATA 6F,CB,AF,32,00,6F,D3,01,  CD,9A,6F,CD,CB,6F,7B,B1
1680 DATA 20,F6,3A,00,6F,CB,EF,32,  00,6F,D3,01,1B,C9,11,00
1690 DATA 00,ED,53,06,6F,CD,8A,6F,  BE,2B,09,E5,2A,06,6F,23
1700 DATA 22,06,6F,E1,CD,CB,6F,7B,  B1,20,EA,1B,AA,CD,8A,6F
1710 DATA 77,CD,CB,6F,7B,B1,20,F5,  1B,9D,3A,01,6F,D3,00,3A
1720 DATA 00,6F,D3,01,00,00,00,DB,  00,C9,7E,D3,00,3A,00,6F
1730 DATA CB,F7,D3,01,CB,B7,D3,01,  3A,01,6F,D3,00,3A,00,6F
1740 DATA CB,FF,F3,D3,01,CD,BE,6F,  CB,BF,D3,01,FB,C9,F5,C5
1750 DATA 01,30,25,0B,7B,B1,20,FB,  C1,F1,C9,E5,21,01,6F,34
1760 DATA AF,BE,20,04,21,00,6F,34,  E1,23,0B,C9, FINITO
    
```

## Listing 1 Programm für EPROM-PROGRAMMIER

77FF (bzw. 7FFF) verglichen. Sollten Abweichungen auftreten, so werden die fehlerhaften Bytes gezählt.

## LESEN

Der Inhalt des EPROMs wird in den Speicherbereich von 7000 bis 77FF (bzw. 7FFF) eingelesen.

## DER PROGRAMMABLAUF

Im Speicherbereich werden folgende Werte abgelegt (Bild 3). P2BYTE und P1BYTE enthalten die EPROM-Adressen, die über Port 2 und Port 1 ausgegeben werden. In ANFANG ist die Anfangsadresse 7000 des Speicherbereichs abgelegt. In LAENGE steht die Blocklänge, entweder 0800 (=2048 dezimal) für EPROM 2716

oder 1000 (=4096 dezimal) für EPROM 2732.

Die Speicherstelle FAL123 wird vom BASIC-Programm mit 1, 2 oder 3 belegt, je nachdem, ob das EPROM programmiert, verglichen oder eingelesen wird.

Bei der Programmierung wird ein Datenbyte in das 8-bit-Register geschrieben, die Adresse an Port 2 und Port 1 ausgegeben und schliesslich der Programmierimpuls von 50 Millisekunden Dauer ausgegeben. Um die Zeitschleife nicht zu stören, wird für diese 50 ms Dauer die Interruptmöglichkeit abgeschaltet und anschliessend wieder eingeschaltet.

Nach jeder Byteausgabe werden die Adresszeiger für den Speicher und das EPROM erhöht, bis alle Bytes programmiert sind.

Nach der Fertigmeldung am Bildschirm empfiehlt sich das Verifizieren des EPROMs, wobei der Schalter auf "LESEN" umzustellen ist. Beim Verifizieren werden die programmierten Bytes mit dem Speicherinhalt verglichen. Sollten Unstimmigkeiten auftreten, so wird die Anzahl der fehlerhaften Bytes angezeigt.

## WICHTIGE TIPS

Kontrollieren Sie die Verdrahtung Ihrer Platine, lassen Sie es nicht auf einen "Rauchtest" ankommen. Die Numerierung der Portlinien am linken Rand des Schaltplanes bezieht sich auf den 37-poligen Stecker der SHARP MZ80I02-Platine.

Achten Sie auf die richtige Eingabe der Hexwerte in den DATA-Zeilen des BASIC-Programms. Ein einziger falscher Wert kann zur Selbstzerstörung des gesamten Programms führen (vor dem Testen speichern!).

Achten Sie beim Programmieren auf die richtige Schalterstellung, damit die 25-Volt-Spannung nicht am falschen Pin anliegt!

## LISTING 2

```

*****
*
*      Maschinenspracheprogramm zu EPROMMER
*
*****
Leopold Asböck                                -200282-

Das folgende Programm im Z80-Code ist im BASIC-Programm
enthalten und wird von diesem geladen und ausgeführt.
Die Zeitschleife für den 50-Millisekunden-Programmier-
impuls ist für 4-MHz-Takt (Z80A) ausgelegt.

ORG      6F00H

0000      PORT1 EQU 00H
0001      PORT2 EQU 01H
0000      IPORT EQU 00H

6F00      20      P2BYTE: DEFB 20H
6F01      00      P1BYTE: DEFB 00H
6F02      7000'   ANFANG: DEFW 7000H
6F04      0800'   LAENGE: DEFW 0800H
6F06      0000'   FEHLER: DEFW 0000H
6F08      0000'   FAL123: DEFW 0000H

6F0A      E5      PSTART: PUSH HL          ;Register im Stack speichern
6F0B      D5      PUSH DE
6F0C      C5      PUSH BC
6F0D      F5      PUSH AF
6F0E      CD 6F2F' CALL INIT          ;Initialisierung der Output-Ports
6F11      2A 6F02' LD HL,(ANFANG)       ;Blockbeginn
6F14      ED 4B 6F04' LD BC,(LAENGE)     ;Blocklänge
6F18      3A 6F08' LD A,(FAL123)        ;Fallunterscheidung 1/2/3:
6F1B      FE 01      CP 01H
6F1D      28 1F      JR Z,PROGR          ;Programmieren
6F1F      FE 02      CP 02H
6F21      28 3B      JR Z,VERIF          ;Verifizieren
6F23      FE 03      CP 03H
6F25      28 56      JR Z,LESEN          ;Lesen
6F27      CD 6F2F'   RUECK: CALL INIT
6F2A      F1      POP AF                ;Register rückspeichern
6F2B      C1      POP BC
6F2C      D1      POP DE
6F2D      E1      POP HL
6F2E      C9      RET

6F2F      3E 20      INIT: LD A,20H          ;Initialisierung der Output-Ports
6F31      32 6F00'   LD (P2BYTE),A
6F34      D3 01      OUT (PORT2),A        ;Byte 0010 0000
6F36      3E 00      LD A,00H
6F38      32 6F01'   LD (P1BYTE),A
6F3B      D3 00      OUT (PORT1),A        ;Byte 0000 0000
6F3D      C9      RET
6F3E      3A 6F00'   PROGR: LD A,(P2BYTE)  ;Programmieren von 2K/4K-Byte
6F41      CB AF      RES 5,A            ;Bit 5 = 0
6F43      32 6F00'   LD (P2BYTE),A
6F46      D3 01      OUT (PORT2),A        ;Register-Output aktivieren
6F48      CD 6F9A'   PROG1: CALL WRITBY   ;Programmieren eines Bytes
6F4B      CD 6FCB'   CALL NEXTAD        ;Adressen erhöhen, Zähler dekrem.
6F4E      78      LD A,B
6F4F      B1      OR C
6F50      20 F6      JR NZ,PROG1         ;fertig? - nein, nochmals
6F52      3A 6F00'   LD A,(P2BYTE)
6F55      CB EF      SET 5,A            ;Bit 5 = 1
6F57      32 6F00'   LD (P2BYTE),A
6F5A      D3 01      OUT (PORT2),A        ;Register-Output inaktivieren
6F5C      18 C9      JR RUECK           ;Rücksprung

6F5E      11 0000'   VERIF: LD DE,0000H   ;Verifizieren
6F61      ED 53 6F06' LD (FEHLER),DE    ;Fehlerzähler = 0000
6F65      CD 6F8A'   VER1: CALL READBY   ;ein Byte von EPROM lesen
6F68      BE      CP (HL)              ;und vergleichen mit Speicher
6F69      28 09      JR Z,VER2          ;identisch? - ja
6F6B      E5      PUSH HL              ;nein - Fehlerzähler erhöhen
6F6C      2A 6F06'   LD HL,(FEHLER)
6F6F      23      INC HL
6F70      22 6F06'   LD (FEHLER),HL
6F73      E1      POP HL
6F74      CD 6FCB'   VER2: CALL NEXTAD   ;Adresszeiger erhöhen
6F77      78      LD A,B
6F78      B1      OR C
6F79      20 EA      JR NZ,VER1         ;fertig? - nein, nochmals
6F7B      18 AA      JR RUECK           ;ja - Rücksprung

```

Fortsetzung des Programmausdruckes auf der nächsten Seite

```

6F7D CD 6F8A'   LESEN: CALL READBY   ;Lesen - ein Byte einlesen
6F80 77        LD      (HL),A   ;und speichern
6F81 CD 6FCB'   CALL  NEXTAD   ;Adresszeiger erhöhen
6F84 78        LD      A,B
6F85 B1        OR      C
6F86 20 F5     JR      NZ,LESEN ;fertig? - nein, nochmals
6F88 18 9D     JR      RUECK   ;ja - Rücksprung

; READBY liest ein Byte des EPROMs (Adresse in
; P2BYTE/P1BYTE) in den Accu ein

6F8A 3A 6F01'   READBY: LD      A,(P1BYTE) ;Adresse ausgeben
6F8D D3 00     OUT     (PORT1),A
6F8F 3A 6F00'   LD      A,(P2BYTE)
6F92 D3 01     OUT     (PORT2),A
6F94 00        NOP        ;Verzögerung
6F95 00        NOP
6F96 00        NOP
6F97 DB 00     IN      A,(IPORT) ;Byte einlesen
6F99 C9        RET        ;Rücksprung

; WRITBY schreibt das Byte in (HL) in das EPROM
; (Adresse in P2BYTE/P1BYTE)

6F9A 7E        WRITBY: LD      A,(HL) ;Byte nach Accu
6F9B D3 00     OUT     (PORT1),A ;Ausgabe an Port1
6F9D 3A 6F00'   LD      A,(P2BYTE)
6FA0 CB F7     SET     6,A ;Bit 6 = 1
6FA2 D3 01     OUT     (PORT2),A ;Register-Clock LOW-HIGH
6FA4 CB B7     RES     6,A ;Bit 6 = 0
6FA6 D3 01     OUT     (PORT2),A ;Register-Clock HIGH-LOW
6FA8 3A 6F01'   LD      A,(P1BYTE) ;EPROM-Adresse LOW-Byte
6FAB D3 00     OUT     (PORT1),A ;an Port1 ausgeben
6FAD 3A 6F00'   LD      A,(P2BYTE) ;EPROM-Adresse HIGH-Bits
6FB0 CB FF     SET     7,A ;Bit 7 = 1, Progr.impuls
6FB2 F3        DI        ;Interrupt abschalten
6FB3 D3 01     OUT     (PORT2),A ;Progr.impuls einschalten
6FB5 CD 6FBE'   CALL  T5OMS   ;Verzögerung 50 Millisekunden
6FB8 CB BF     RES     7,A ;Bit 7 = 0
6FBA D3 01     OUT     (PORT2),A ;Progr.impuls abschalten
6FBC FB        EI        ;Interrupt ermöglichen
6FBD C9        RET        ;Rücksprung

; T5OMS ist eine Verzögerungsschleife
; für 50 Millisekunden (4 MHz-Takt!)

6FBE F5        T5OMS: PUSH  AF ;2.75 Mikrosekunden
6FBF C5        PUSH  BC ;2.75
6FC0 01 2530'   LD      BC,2530H ;2.5
6FC3 0B        LOOP: DEC  BC ;1.5 / 5.25 us * 9520
6FC4 78        LD      A,B ;1 /
6FC5 B1        OR      C ;1 /
6FC6 20 FB     JR      NZ,LOOP ;1.75 /
6FC8 C1        POP   BC ;2.5
6FC9 F1        POP   AF ;2.5
6FCA C9        RET        ;2.5

; NEXTAD erhöht den Inhalt der EPROM-Adress-
; speicher P2BYTE/P1BYTE um eins
; erhöht HL und dekrementiert BC

6FCB E5        NEXTAD: PUSH HL ;EPROM-Adresse erhöhen
6FCC 21 6F01'   LD      HL,P1BYTE ;Adresse LOW-Byte erhöhen
6FCF 34        INC   (HL)
6FD0 AF        XOR   A
6FD1 BE        CP   (HL) ;wenn 00
6FD2 20 04     JR   NZ,NEXT1
6FD4 21 6F00'   LD      HL,P2BYTE ;dann Adresse HIGH-Bits erhöhen
6FD7 34        INC   (HL)
6FD8 E1        NEXT1: POP  HL
6FD9 23        INC  HL ;Speicherzeiger erhöhen
6FDA 0B        DEC  BC ;Bytezähler dekrementieren
6FDB C9        RET        ;Rücksprung

END

Symbols:
ANFANG 6F02 FAL123 6F08 FEHLER 6F06 INIT 6F2F
IPORT 0000 LAENGE 6F04 LESEN 6F7D LOOP 6FC3
NEXT1 6FD8 NEXTAD 6FCB P1BYTE 6F01 P2BYTE 6F00
PORT1 0000 PORT2 0001 PROG1 6F48 Progr 6F3E
PSTART 6FOA READBY 6F8A RUECK 6F27 T5OMS 6FBE
VER1 6F65 VER2 6F74 VERIF 6F5E WRITBY 6F9A

```

No error(s)

## Listing 2 Assemblerprogramm



**BUEFA 1982**  
Ausstellung für Informatik,  
Büroelektronik und Büro-  
einrichtungen  
13. - 17. September 1982  
Genf

**MINI/MICRO COMPUTER  
CONFERENCE  
AND EXPOSITION**  
14. - 16. September 1982  
Anaheim, Californien/USA

**SICOB**  
Int. Salon für EDV,  
Telematik, Kommunikation,  
Büro-Organisation und  
Bürotechnik  
22. Sept. - 1. Okt. 1982  
Paris

**HOBBY ELEKTRONIK 82**  
Fachausstellung für  
Praktische Elektronik  
6. - 10. Oktober 1982  
Stuttgart

**INTERBIRO**  
Int. Ausstellung für Daten-  
verarbeitungsmittel und  
Bürowirtschaft  
11. - 15. Oktober 1982  
Zagreb

**BUEROFA '82**  
Büro- und Datentechnik  
15. - 18. Oktober 1982  
Kiel

**EDUCATA '82**  
Schweizerische Ausstellung  
für berufliche Aus- und  
Weiterbildung  
22. - 27. Oktober 1982  
Zürich

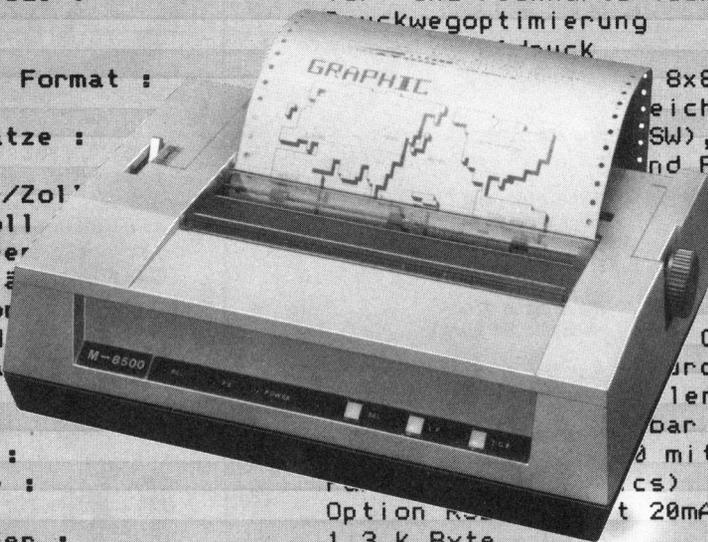
**ORGATECHNIK**  
Int. Büromesse  
26. - 31. Oktober 1982  
Köln

**ELECTRONICA '82**  
10. Intern. Fachmesse für  
Bauelemente und Baugruppen  
der Elektronik  
9. - 13. November 1982  
München

# Der Matrixprinter 8510 von C.Itoh setzt neue Masstäbe

## Technische Daten:

Druckgeschwindigkeit : 100 Zeichen/Sek.  
 Druckmethode : Vor- und rückwärts (schaltbar)  
 Druckwegoptimierung  
 Druckwegoptimierung  
 Charakter Format : 8x8(Graphic) 9 Nadelkopf  
 Zeichen, Unterlängen,  
 Zeichensätze : (SW), Griechisch,  
 und Punktdarstellung  
 Charakter/Zoll \* 5 \* und proportional  
 Zeilen/Zoll Dot-Graphik  
 Tabulatorer n/Zeilen programmierbar  
 Formularl Original plus 3 Kopien)  
 Formularb durch Schrittmotor  
 Formulard lenpapier und Einzelblatt  
 Papiertra dar  
 Farbband : (mit Spulenfarbband)  
 Interface : (cs)  
 Datenbuffer : Option RS-232C  
 1,3 k Byte t 20mA Current Loop



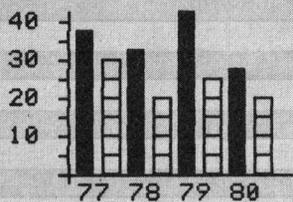
Modell 8500 80-stelliger Dot Matrix Drucker

**C. Itoh**

## Schriftmuster :

10 CPI PICA : THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG  
 5 CPI Elongated : THE QUICK BROWN FOX JUMPS  
 OVER THE LAZY DOG  
 Proportional : THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG  
 Elongated : THE QUICK BROWN FOX JUMPS  
 OVER THE LAZY DOG  
 17 CPI Compressed : THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG  
 8,5 CPI Elongated : THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG  
 12 CPI Elite : THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG  
 6 CPI Elongated : THE QUICK BROWN FOX JUMPS  
 OVER THE LAZY DOG

## Graphic :



Dieser 80stellige Dot-Matrix-Serial-Drucker bietet dem Systemdesigner eine Fülle von Möglichkeiten in der Standardversion, die gewöhnlich nur als kostspielige Optionen erhältlich sind.

Durch eine Druckwegoptimierung wird die Druckleistung gegenüber früheren Druckermodellen erheblich gesteigert. Der zusätzliche grosse Zeichenpuffer entlastet das Computer-System. Ein 9-Nadel-Kopf mit Unterlängen und echtem Unterstreichen, Grafikmöglichkeiten mit Ansteuerung einzelner Nadeln sowie Schrittmotoren für den horizontalen und vertikalen Antrieb runden das Bild ab. Vier Printer lassen sich, einzeln adressier-

bar, an ein Computer-System anschliessen. Als Option kann kundenspezifische Firmware betrieben werden. Anschlussmöglichkeit an alle Systeme über die Schnittstellen Standard Parallel, RS 232C oder 20 mA. Das Gerät ist steckerkompatibel mit fast allen Druckertypen. Es wird mit einer 6monatigen Garantie geliefert. Für allfällige Probleme steht Ihnen ein vielseitiges, gut ausgebildetes und technisch versiertes Kundendienstpersonal zur Verfügung. Die Qualität, Arbeitsleistung und Wirtschaftlichkeit dieses Matrixprinters lässt ihn jedem anderen vergleichbaren Modell seiner Preisklasse voranstellen. Verlangen Sie detaillierte Unterlagen.

# GEWUSST WIE!



## Benchmarktests

Eric HUBACHER

Sicher haben auch Sie schon das Wort Benchmarktest ausgesprochen oder es mindestens irgendwo gehört? Doch wissen Sie auch, was es bedeutet, und was man darunter versteht? Im Englischen versteht man unter Benchmark eine Markierung, die irgendwo angebracht wird um daran andere, ähnliche Objekte vergleichen zu können.

Stellen Sie sich eine Gruppe von Kindern vor. Der Anführer stellt sich an einen Türrahmen und markiert seine Grösse (Länge!) mit einem Strich am Türrahmen. Jeder seiner Freunde und Mitspieler kann sich nun ebenfalls an den Rahmen stellen und nachschauen, ob er grösser oder kleiner als der Anführer ist. Soviel zum Begriff BENCHMARK in seiner ursprünglichen Bedeutung.

Dieses einfache Beispiel zeigt uns viele Erkenntnisse, die auch bei den Computer-Benchmarktests gültig sind.

Die Marke lässt zwar erkennen, dass Peter grösser ist als Max. Es leuchtet Ihnen aber bestimmt ein, dass diese Feststellung nichts über das Spurtvermögen von Peter im Vergleich zu Max aussagt. Auch erlaubt diese Marke keinen Vergleich der geistigen Fähigkeiten.

Gleiche Verhältnisse bestehen nun auch bei Computer-Benchmarktests.

Die bekanntesten Benchmarktests im Kleincomputer-Bereich sind Geschwindigkeits-Vergleiche; doch es gibt auch solche, mit welchen sich die Rechengenauigkeit oder die Grösse des für ein "Normprogramm" benötigten Speicherbereiches bestimmen lassen.

### KILOBAUD-PROGRAMME

Die ersten Benchmark-Programme für Kleincomputer wurden vor etwa

fünf Jahren von der amerikanischen Computer-Zeitschrift "Byte" (damals noch "Kilobaud Microcomputing") vorgestellt. "Byte" publizierte acht verschiedene Testprogramme,

numeriert von BM1 bis BM 8 (BM bedeutet Benchmark). Dem damaligen Stand der Kleincomputertechnik entsprechend erlaubten sie hauptsächlich, die Geschwindigkeit der Pro-

```
10 REM BM1
300 PRINT "S";CHR$(7)
400 FOR K=1 TO 1000
500 NEXT K
700 PRINT "E";CHR$(7)
800 END
```

```
10 REM BM2
300 PRINT "S";CHR$(7)
400 K=0
500 K=K+1
600 IF K<1000 THEN 500
700 PRINT "E";CHR$(7)
800 END
```

```
10 REM BM3
300 PRINT "S";CHR$(7)
400 K=0
500 K=K+1
510 A=K/K*K+K-K
600 IF K<1000 THEN 500
700 PRINT "E";CHR$(7)
800 END
820 RETURN
```

```
10 REM BM4
300 PRINT "S";CHR$(7)
400 K=0
500 K=K+1
510 A=K/2*3+4-5
600 IF K<1000 THEN 500
700 PRINT "E";CHR$(7)
800 END
```

```
10 REM BM5
300 PRINT CHR$(7)
400 K=0
500 K=K+1
510 A=K/2*3+4-5
600 IF K<1000 THEN 500
700 PRINT CHR$(7)
800 END
```

```
10 REM BM6
300 PRINT "S";CHR$(7)
400 K=0
430 DIM M(5)
500 K=K+1
510 A=K/2*3+4-5
520 GOSUB 820
530 FOR L=1 TO 5
540 NEXT L
600 IF K<1000 THEN 500
700 PRINT "E";CHR$(7)
800 END
820 RETURN
```

```
10 REM BM7
300 PRINT "S";CHR$(7)
400 K=0
430 DIM M(5)
500 K=K+1
510 A=K/2*3+4-5
520 GOSUB 820
530 FOR L=1 TO 5
535 M(L)=A
540 NEXT L
600 IF K<1000 THEN 500
700 PRINT "E";CHR$(7)
800 END
820 RETURN
```

```
10 REM BM8
300 PRINT "S";CHR$(7)
400 K=0
500 K=K+1
530 A=K^2
540 B=LOG(K)
550 C=SIN(K)
600 IF K<100 THEN 500
700 PRINT "E";CHR$(7)
800 END
820 RETURN
```

Bild 1 Die Testprogramme von Kilobaud

grammauführung durch den Basic-Interpreter zu vergleichen.

Diese Programme - abgebildet sind sie in Bild 1 - sind aufeinander aufgebaut. Das bedeutet, dass nach Eingabe des Benchmark-Programmes BM1 das nächste durch einfaches Einfügen weiterer Programmschritte erzielt wird.

Nach diesem Prinzip entsteht BM3 aus BM2 durch Einfügen des Programmschrittes 510. Dieses Konzept wurde jedoch nicht konsequent durchgehalten (z.B. Uebergang von BM1 nach BM2).

Die in Bild 2 dargestellten Ergebnisse dieser BM1 bis BM8 sind mit Vorsicht zu betrachten. Bei geschickter Beeinflussung der Programme lassen sich bessere Ausführungszeiten erreichen als beim einfachen Eintippen der Programme. So kann z.B. im Beispiel BM1 die Variable K als Integerzahl definiert werden (entweder durch Schreiben von K %, oder indem am Programmfang eine Instruktion 10 DEF INT K eingegeben wird). Lassen Sie die so abgeänderten Programme auf Ihrem Computer ablaufen, und vergleichen Sie die Ausführungszeiten.

Die Byte-Programme definieren also nicht, ob die Variablen als Integer-, Single-Precision- oder Double-Precision-Werte behandelt werden (Kästchen).

Die Testprogramme von Byte erlauben auch keine Aussagen über die Zugriffszeiten auf externe Datenspeicher (wie z.B. Floppy Diskettenstationen). Trotz der aufgezählten Nachteile muss man sie kennen, da sie noch sehr verbreitet sind, besonders in der amerikanischen Literatur.

## MUK TESTPROGRAMME

Die von Mikro+Kleincomputer verwendeten Tests - im Folgenden werden sie MUK-Tests genannt - stellen wir Ihnen nachstehend detailliert vor. Die Zusammenstellung der Programme können Sie aus Bild 3 entnehmen, während in Bild 4 einige bereits gemessene Ausführungszeiten zusammengestellt sind.

Falls das von Ihnen getestete System eine eingebaute Echtzeituhr besitzt, können Sie in allen MUK-Testprogrammen (MUK 1..6) die Zeile 100 durch einen Start der internen Zeituhr ersetzen und in Schritt 400 einen Stopp der Echtzeituhr sowie ein Auslesen der benötigten Ausführungszeit veranlassen. Auf diese Weise benötigen Sie keine Stoppuhr, die Sie jeweils zur rechten Zeit von Hand starten müssen.

## MUK 1

Der Test MUK 1 entspricht dem BM1. Er weist jedoch eine grössere

## INTEGER VARIABLEN

Unter Integer Variablen versteht man ganzzahlige Variablen (ohne Nachkommastellen) im Bereich von -32'768 bis +32'767. Sie benötigen einen Speicherplatz von 2 Bytes pro Zahl.

"Single Precision" Variablen sind sechsstellig ausgedruckte Fließkommazahlen die mit einer Genauigkeit von 7 Stellen abgespeichert werden. Sie benötigen einen Speicherplatz von 4 Bytes.

"Double precision" Variablen sind Fließkommazahlen die gegenüber "Single-Precision" Variablen mit einer verdoppelten Genauigkeit verarbeitet werden. Ausgedruckt und abgespeichert werden sie mit einer Genauigkeit von 16 Stellen. Jede Zahl benötigt einen Speicherplatz von 4 Bytes.

Diese Angaben gelten für Microsoft-Basic.

Zahl von Schleifendurchläufen auf, wodurch eine grössere Ausführungszeit erreicht wird. Dies erlaubt eine genauere Messung, da der Fehler, der durch die Reaktionszeit verursacht wird, relativ kleiner ist. Die Laufvariable (I) wird ausserdem durch die Anweisung in Schritt 10 als Integer-Zahl deklariert.

```
10 REM MUK1
20 DEFINT I
100 PRINT"START"
200 FOR I = 1 TO 30000
300 NEXT I
400 PRINT "ENDE";CHR$(7)'Beep
500 END
```

Durch einfaches Abändern der Instruktion in Zeile 10 können Sie die Laufvariable auch als Single Precision (DEFSNG I) oder Double

Computer Typ	BM 1	BM 2	BM 3	BM 4	BM 5	BM 6	BM 7	BM 8
Commodore CBM 3032	1.6	9.7	18.1	20.1	21.7	32.3	50.9	11.8
Tandy TRS-80	2.5	18.0	34.5	39.0	45.0	67.0	109.0	-
Apple II	1.3	8.5	16.0	17.8	19.1	28.6	44.8	10.7
Sorcerer	1.8	10.0	20.7	22.2	24.3	37.6	53.7	9.6
Tandy TRS-80 Color	2.5	11.0	22.0	23.0	26.5	40.5	59.0	-
Sinclair ZX 81	1.7	4.6	9.1	8.0	12.1	25.4	38.5	-
Philips P2000	1.7	7.4	17.3	18.5	20.1	32.0	49.6	8.9
HP 85	2.2	4.5	17.0	17.0	18.0	30.0	45.0	12.5
Commodore VC 20	1.5	8.9	16.4	18.1	19.8	29.1	46.2	10.7
DCT-Superbrain	1.6	5.3	14.0	13.9	14.8	26.1	43.1	4.6

Bild 2: Ergebnisse der BM-Testprogramme in Sekunden

Precision (DEFDBL I) definieren. Diese Definitionen gelten für Microsoft Basic; in anderen Dialekten müssen diese manchmal leicht geändert werden; im Prinzip sind sie jedoch überall möglich.

Die Ausführungszeiten für eine Programmschleife mit einem Schlaufenzähler (I), der als Variable mit doppelter Präzision abgespeichert wird, sind bei manchen Basicdialekten dramatisch grösser.

Mikro+Kleincomputer publiziert jeweils nur die Zeit, welche für einen Test nach MUK 1 erzielt wurde.

Der Test MUK 2 ermittelt die Ausführungsgeschwindigkeit eines Programmes, in welchem Floating-Point Operationen vorkommen.

## MUK 2

Anstatt die Variablen I und Z im Programmkopf zu definieren, kann für die Integervariable I ein I% und für die Floating Point Variable Z einfach nur Z geschrieben werden.

```
10 REM MUK2
20 DEFINT I
30 DEFSNG Z
40 Z=1!
100 PRINT"START"
200 FOR I = 1 TO 10000
250 Z=Z*1.001
300 NEXT I
400 PRINT "ENDE";CHR$(7)
500 END
```

Die Definition im Programmkopf soll Sie jedoch ausdrücklich daran erinnern, dass in anderen Programmiersprachen (z.B. PASCAL) eine Variablendeklaration erforderlich ist.

## MUK 3

```
10 REM MUK3
20 DEFINT I
30 DEFDBL A
100 PRINT"START"
200 FOR I = 1 TO 2000
250 A=A+(I^2.2)/EXP
(SQR (I)+1)
300 NEXT I
400 PRINT "ENDE";CHR$(7)
450 PRINT A
500 END
```

MUK 3 und MUK 4 erlauben auch einen Vergleich der Rechengenauigkeit verschiedener Programmiersprachen. Die mit diesem Test erzielten Rechenresultate sind in der Tabelle in Bild 5 aufgelistet.

## MUK 4

Den Test MUK 4 ersetzen wir durch den in der Kundenzeitschrift BIT 68 (Digicomp) gefundenen modifizierten Ashley-Grayson-Test.

```
10 REM MUK 4
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 END
```

Beim Vergleich von Rechnern sind die erzielbaren Anführungsgeschwindigkeiten von Rechenprogrammen jedoch nur ein Teilaspekt. Bei den meisten im Small Business-Bereich eingesetzten Rechnern ist die erforderliche Zugriffszeit, um Daten von einem externen Speicher zu lesen und auf dem externen Speicher zu sichern, mindestens ebenso wichtig. Von den bis jetzt vorgestellten Tests BM1..8 und MUK 1..4 werden diese Zeiten aber nicht überprüft. Sie werden von den Programmen MUK 5...8 ausgetestet.

## MUK 5

Das Testprogramm MUK 5 erstellt ein sequentielles File (TEST5.DAT) mit einer Länge von 100 kByte, bestehend aus 2000 Records mit je 50 Bytes. Am Anfang jedes Records steht seine Nummer. Dieses Textfile wird von MUK 6 und 7 benötigt.

```
10 REM MUK1
20 DEFINT I
100 PRINT"START"
200 FOR I = 1 TO 30000
300 NEXT I
400 PRINT "ENDE";CHR$(7)'Beep
500 END

10 REM MUK2
20 DEFINT I
30 DEFSNG Z
40 Z=1!
100 PRINT"START"
200 FOR I = 1 TO 10000
250 Z=Z*1.001
300 NEXT I
400 PRINT "ENDE";CHR$(7)
500 END

10 REM MUK3
20 DEFINT I
30 DEFDBL A
100 PRINT"START"
200 FOR I = 1 TO 2000
250 A=A+(I^2.2)/EXP
(SQR (I)+1)
300 NEXT I
400 PRINT "ENDE";CHR$(7)
450 PRINT A
500 END

10 REM MUK4
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 END

10 REM MUK5
20 DEFINT I
40 T$=STRING$(50,"-")
70 OPEN "O",#1,"TEST5.DAT"
100 PRINT"START";CHR$(7)
200 FOR I = 1 TO 2000
250 PRINT #1,T$
300 NEXT I
400 PRINT "ENDE";CHR$(7)
500 END

10 REM MUK6
20 DEFINT I
40 OPEN "R",#1,"TEST5.DAT",53
50 FIELD #1,50 AS A$
100 PRINT"START";CHR$(7)
200 FOR I = 1 TO 100
250 GET #1,INT(RND*2000)
300 NEXT I
400 PRINT "ENDE";CHR$(7)
500 END

MUK 7 (für ein CP/M-
Betriebssystem):
A> PIP <CR>
* B:=A:TESTS.DAT <CR>
```

Bild 3 Die MUK-Testprogramme

Computer Typ	MUK 1	MUK 2	MUK 3	MUK 4	MUK 5	MUK 6	MUK 7
DCT-Superbrain*	24	37	164	604	117	181	71
Altos ACS 8000*	26	39	146	530	170	175	75
Apple II Plus <sup>1)</sup>	41	157	264	260	330	60	53
Apple III <sup>2)</sup>	51	120	181	201	50	34	40
Commodore CBM 8032 <sup>3)</sup>	46	174	292	267	103	-	-
Monroe OC 8820*	28	41	176	651	114	149	73
Sirius <sup>4)</sup>	MS-DOS	42	41	52	176	86	27
	CP/M*	36	36	66	325	109	82
HP-87 <sup>5)</sup>	148	156	206	269	119	23	-
VT-180 Robbin*	21	31	132	488	58	99	50

\*) CP/M-Betriebssystem  
<sup>1)</sup> Apple II-Basic und Apple II Copyprogramm  
<sup>2)</sup> Apple III-Businessbasic und Apple III Copyprogramm  
<sup>3)</sup> CBM-Basic mit 8050 Discdrive  
<sup>4)</sup> 16-Bit Processor  
<sup>5)</sup> HP-87 Basic

Bild 4: Ausführungszeiten der MUK-Programme in Sek. (Alle Tests in interpretiertem BASIC)

```

10 REM MUK5
20 DEFINT I
40 T$=STRING$(50,"_")
70 OPEN "O",#1,"TEST5.DAT"
100 PRINT"START";CHR$(7)
200 FOR I = 1 TO 2000
250 PRINT #1,T$
300 NEXT I
400 PRINT "ENDE";CHR$(7)
500 END
    
```

Die Ausführungsgeschwindigkeit dieses Programmes ist praktisch nur von jener Zeit abhängig, die der Computer benötigt, um die 2000 Records sequentiell auf seiner Diskette unterzubringen. Die Einflüsse durch die Rechengeschwindigkeit und Programmiersprache können vernachlässigt werden.

Wichtig ist es, darauf zu achten, dass dieses Programm für Microsoft-Basic geschrieben wurde, bei andern Disketten und Sprachen können sich die Befehle zur Fileerstellung beträchtlich unterscheiden.

#### MUK 6

Dieses Testprogramm testet die Geschwindigkeit des Zugriffs auf die Disketten sowie die Transferate.

Computer Typ	MUK 3	MUK 4
DCT-Superbrain	177,1954690415149	189477.39527548
Apple II Plus	177,195135	189477.396
Commodore CBM 8032	177,195135	189477.396
Monroe OC 8820	177,1951690415149	189477.3952527584
Sirius	MS-DOS	177,1951507478859
	CP/M	177,1951545384132
HP-87	177,195136899	189477.387828
VT-180 Robbin	177,1951690415149	189477.3952527584

Bild 5: Die mit MUK 3 und 4 erzielten Rechenresultate

```

10 REM MUK6
20 DEFINT I
40 OPEN "R",#1,"TEST5.DAT",53
50 FIELD #1,50 AS A$
100 PRINT"START";CHR$(7)
200 FOR I = 1 TO 100
250 GET #1,INT(RND*2000)
300 NEXT I
400 PRINT "ENDE";CHR$(7)
500 END
    
```

#### MUK 7

Beim Geschwindigkeitstest Nummer 7 wird das 100 kByte grosse File "TEST5.DAT" von einer Diskettenstation auf die andere übertragen. In CP/M erfolgt dies mit dem Befehl PIP. Die Instruktion wird ohne die Option V (verify) eingegeben. V bedeutet, dass nach dem Schreiben das neue File nochmals gelesen und mit dem ursprünglichen verglichen werden muss. Für das CP/M Betriebssystem lautet die Befehlsfolge:

MUK 7 (für ein CP/M-Betriebssystem):

```

A> PIP <CR>
* B:=A:TESTS.DAT <CR>
    
```

Beim Drücken von Return gleichzeitig die Stoppuhr starten.

Sobald das Prompt wieder erscheint, müssen Sie die Uhr stoppen.

Diese sieben Testprogramme erlauben also einen Vergleich für alle verschiedenen Kleincomputer in Bezug auf Ausführungsgeschwindigkeit und Rechengenauigkeit.

Es muss hier jedoch nochmals darauf hingewiesen werden, dass diese Geschwindigkeitsvergleiche nur einen von vielen Aspekten darstellen, welche beim Kauf eines Computersystems berücksichtigt werden müssen.

Wir wünschen Ihnen beim Arbeiten mit diesen Testprogrammen viel Vergnügen. Sollten Sie Zugang zu einer Maschine besitzen, über die wir noch keine Messdaten (Tabellen in den Bildern 2 und 5) veröffentlichten, so freuen wir uns über eine Zusendung dieser Testwerte. Eine damit erweiterte Benchmarktabelle werden wir dann in einer der nächsten Ausgaben von Mikro+Kleincomputer veröffentlichen.

NUR NOCH WENIGE EXEMPLARE...

Achtung! Die Nachfrage nach Mikro+Kleincomputer, dem Schweizer Kleincomputer-Fachmagazin, hält unvermindert an. Folgende Ausgaben sind vergriffen: 79-3, 79-4, 80-1, 80-2, 80-6 und 81-1. Auch von den übrigen bisher erschienenen Nummern sind nur noch wenige Exemplare vorrätig. Ein Nachdruck vergriffener Ausgaben ist nicht vorgesehen. Sichern Sie sich sofort die Ihnen noch fehlende Hefte.

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

## Sortieren mit Microsoft-BASIC

Leo SCHWERI

In Mikro+Kleincomputer 81-1 wurden verschiedene Sortierverfahren beschrieben und miteinander verglichen. Sieht man sich in der Literatur etwas um, so stellt man fest, dass unter den Verfahren, deren Effizienz sich nicht auf irgendeiner speziellen Datenfeld-Eigenschaft abstützt, Quicksort am besten abschneidet. Leider fehlt Quicksort in der genannten Arbeit. Es wird daher im vorliegenden Beitrag gezeigt, wie Quicksort in Microsoft-BASIC umgeschrieben und für bestimmte Anwendungen optimiert werden kann.

Die vier in Microsoft-BASIC geschriebenen Programme wurden auf einem Superbrain ausgetestet, für den auch die Zeitmessungen gelten. Für andere Rechner dürften die Zeitverhältnisse zwischen den einzelnen Varianten ähnlich bleiben. Die Optimierung von Quicksort basiert auf vier Grundideen:

- Das Programm soll wenig verändert werden, gegenüber dem optimierten PASCAL-Original in (1).
- Verwenden von Integer-Variablen, wo immer dies sinnvoll ist.
- Verwenden von SWAP für den Elemententausch.
- Umgehen von Operationen die zu "Garbage-Collections" (Speicherbereinigungen) führen.

### DAS PRINZIP

Quicksort beruht auf folgender Erkenntnis: Ein Sortieralgorithmus ist dann besonders effizient, wenn der Austausch von Elementen über grössere Distanzen im Datenfeld erfolgt. Man wählt willkürlich ein Element x, durchsucht dann den Array von links nach rechts, bis man ein Element findet, das grösser als x ist. Dann durchsucht man den Array von rechts her, bis man ein Element findet, das kleiner als x ist. Nun werden diese beiden Elemente vertauscht.

Durchsuchen und Vertauschen werden solange weitergeführt, bis man

lich. Wir müssen somit eine iterative Lösung suchen. Der Schlüssel dazu liegt im Führen von Listen mit den Kenngrössen für die Zerlegungen. Das hat Quicksort den Ruf eingetragen, in der Programmierung etwas aufwendig zu sein. Von der Rechenzeit her ist Quicksort jedoch dermassen effizient, dass dieses

	:	100	:	1000	:	Elemente sortiert	:
-----							
Quicksort 1 :	13	:	175	:	sec :	Zahlen	:
Quicksort 2 :	301	:	2700	:	sec :	10 Zeichen/\$	:
Quicksort 3 :	15	:	540	:	sec :	"	:

sich beim Durchsuchen von beiden Seiten her irgendwo in der Mitte trifft. Unser Datenfeld ist nun zerlegt in einen linken Teil mit Elementen kleiner als das Schlüsselement x und einen rechten Teil mit Elementen die grösser sind.

Diesen Zerlegungsprozess wenden wir nun auf beide Seiten des Datenfelds an und wiederholen diese Zerlegung für immer kleinere Teilfelder, bis jedes Teilfeld nur noch ein Element umfasst. Dies führt auf ein rekursives Verfahren, d.h. auf ein Vorgehen, bei dem sich ein Programmteil selber wieder aufruft. Bei den meisten BASIC-Dialekten sind aber Rekursionen nicht mög-

Mehr an Speicherplatz bei weitem aufgewogen wird durch seine Leistung, etwa gegenüber einem Bubble-Sort.

Weiter braucht Quicksort etwas Speicherplatz für die Listen. Sie werden umso länger, je mehr Elemente sortiert werden sollen. Hat das Datenfeld etwa 1000 Elemente, so entstehen Listen mit je 13 - 14 Elementen, während es bei einer Million Datenfeld-Elementen etwa 30 sein werden. In diesem Sinne ist Quicksort etwas vom zu lösenden Problem abhängig, doch lässt sich die Maximallänge der Listen meist genügend genau abschätzen und die Listenlänge entsprechend gross dimensionieren.

```

90 DEFINT I
91
92
93
100 NMAX=1000
110 DIM MAT(NMAX)
120 DIM IQR(20),IQL(20)
130
140 *HIER MUSS DER ARRAY MAT(I) MIT DEN DATEN GEFUELLT WERDEN
150
1860 *----- QUICKSORT 1
1880 IS=1:IQL(1)=1:IQR(1)=N
1890 IL=IQL(IS):IR=IQR(IS):IS=IS-1
1900 I=IL:IJ=IR:X=MAT((IL+IR)/2)
1910 WHILE MAT(I)<X:I=I+1:WEND
1920 WHILE X<MAT(IJ):IJ=IJ-1:WEND
1930 IF I<=IJ THEN SWAP MAT(I),MAT(IJ):I=I+1:IJ=IJ-1
1940 IF I<=IJ THEN 1910
1950 IF I<IR THEN IS=IS+1:IQL(IS)=I:IQR(IS)=IR
1960 IR=IJ
1970 IF IL<IR THEN 1900
1980 IF IS<>0 THEN 1890
1990 * ENDE QUICKSORT

```

Bild 2 Quicksort 1

```

2000 DEFINT I           'I.. SIND INTEGERVERARIABLE
2001 NMAX=1000         '1000 DATENELEMENTE
2002 DIM TXT$(NMAX)    'DATENFELD
2003 DIM IQR(20),IQL(20) 'LISTEN
2004 '
2005 ' HIER DAS DATENFELD TXT$(I) AUFFUELLEN
2006 '
2010 REM ----- QUICKSORT 2
2020 IS=1:IQL(1)=1:IQR(1)=N
2030 IL=IQL(IS):IR=IQR(IS):IS=IS-1
2040 I=IL:IJ=IR:X#=TXT$((IL+IR)\2)
2050 WHILE TXT$(I)<X#:I=I+1:WEND
2060 WHILE X#<TXT$(IJ):IJ=IJ-1:WEND
2070 IF I<=IJ THEN Y#=TXT$(I):TXT$(I)=TXT$(IJ):TXT$(IJ)=Y#:I=I+1:IJ=IJ-1
2080 IF I<=IJ THEN 2050
2090 IF I<IR THEN IS=IS+1:IQL(IS)=I:IQR(IS)=IR
2100 IR=IJ
2110 IF IL<IR THEN 2040
2120 IF IS<>0 THEN 2030
2130 ' ENDE QUICKSORT

```

Bild 3 Quicksort 2

Das Programmstück in Bild 2 ist eine direkte Uebersetzung des in (1) angegebenen PASCAL-Programms in BASIC 80. Sortiert wird der Array MAT(I). Durch Zeile 90 sind alle mit dem Buchstaben I beginnenden Variablen zu Integervariablen geworden. SWAP A, B vertauscht A und B.

Quicksort kann auch zum Sortieren von Strings gebraucht werden. Das nächste Programm in Bild 3 dient dazu.

In Quicksort 2 wurde auf die Verwendung von SWAP verzichtet. Welch signifikante Verbesserung die Sortierzeit SWAP bringt, lässt sich durch Vergleich der Zeiten von Quicksort 2 und Quicksort 3 ermes- sen (Tabelle 1). Die Laufzeiten von Quicksort 2 und 3 hängen sehr stark von der Grösse des Restspeichers und der Stringlänge ab.

Es interessiert vielleicht, warum diese dramatischen Zeitunterschiede entstehen. In MICROSOFT-BASIC wird die Länge eines Strings nicht festgelegt. Dadurch ist es auch nicht möglich einem String einen festen Speicherbereich zuzuordnen, wie etwa einer Zahl, von der man weiss, dass sie aus beispielsweise vier Bytes besteht (bzw. 2 Bytes bei Integerzahlen oder 8 Bytes bei doppelt genauen Zahlen). Stattdessen steht in der Variablenliste, die der Rechner intern anlegt, nur die Adresse eines Speicherbereichs, auf

dem der String abgelegt ist. Der Speicherbereich mit dem eigentlichen String heisst "Halde" (2), im Gegensatz etwa zum "Kellerspeicher", dem Stack. Wird ein String geändert, so wird er neu auf der Halde, am Anfang des freien Speicherbereichs gespeichert und im Variablenverzeichnis eine neue Anfangsadresse vermerkt. Diese Art der Stringspeicherung hat nun einige Konsequenzen:

1) Um zwei Strings gegeneinander auszutauschen genügt es, statt die einzelnen Zeichen des Strings, nur die Anfangsadressen im Variablenverzeichnis, also die Pointer die auf den String weisen, zu tauschen. So ein Pointer besteht aus zwei Bytes, während ein String aus tausenden von Buchstaben bestehen kann. Genau diesen Tausch von zwei Speicheradressen bewirkt SWAP, dessen Ueberlegenheit gegenüber dem naiv

durchgeführten Austausch von zwei Variablen damit verständlich ist.

2) Weil bei jeder Aenderung eines Strings ein neuer String auf der Halde abgelegt wird, überfließt die Halde früher oder später. Die alten, ungültigen Strings bleiben nämlich auf der Halde. Wenn nun die Halde überfließt, und kein freier Speicherplatz mehr verfügbar ist, so tritt automatisch eine Speicherbereinigung - die "Garbage-collection" - in Funktion, die alle ungültigen Variablen entfernt, und die gültigen Variablen zusammenschiebt, sodass wieder ein freier Speicherbereich an einem Ende der Halde entsteht. Eine derartige Garbage-collection ist eine sehr zeitraubende Tätigkeit, obwohl der Benutzer, ausser der Wartezeit, davon nichts merkt.

Bei Quicksort 3 (Bild 4) tritt in Zeile 2180 eine Stringzuweisung auf. Beim Sortieren von 1000 Strings wurde festgestellt, dass diese Zuweisung 890 mal erfolgte (die genaue Zahl hängt von den zu sortierenden Strings ab). Je nach Stringlänge werden daher mehrere Garbage-Collections im Verlaufe des Sortierens erforderlich sein. Bei der Variante Quicksort 4 (Bild 5) wurde dieses Problem behoben. Es wird aber ein neuer, grosser Array IO(NMAX) nötig. Die Idee besteht darin, dass nicht mehr die Strings sortiert werden, sondern dass wir jedem String eine Nummer zuweisen und diese Nummern entsprechend einem Sortierschlüssel, der sehr wohl

```

2140 DEFINT I           'I.. SIND INTEGERVERARIABLEN
2141 NMAX=1000         '1000 ELEMENTE WERDEN SORTIERT
2142 DIM TXT$(NMAX)    'ZU SORTIERENDER ARRAY
2143 DIM IQR(20),IQL(20) 'SORTIERLISTEN
2144 '
2145 ' HIER MUSS DER ARRAY TXT$(I) MIT DATEN GEFUELLT WERDEN
2146 '
2150 ' ----- QUICKSORT 3
2160 IS=1:IQL(1)=1:IQR(1)=N
2170 IL=IQL(IS):IR=IQR(IS):IS=IS-1
2180 I=IL:IJ=IR:X#=TXT$((IL+IR)\2)
2190 WHILE TXT$(I)<X#:I=I+1:WEND
2200 WHILE X#<TXT$(IJ):IJ=IJ-1:WEND
2210 IF I<=IJ THEN SWAP TXT$(I),TXT$(IJ):I=I+1:IJ=IJ-1
2220 IF I<=IJ THEN 2190
2230 IF I<IR THEN IS=IS+1:IQL(IS)=I:IQR(IS)=IR
2240 IR=IJ
2250 IF IL<IR THEN 2180
2260 IF IS<>0 THEN 2170
2270 ' ENDE QUICKSORT

```

Bild 4 Quicksort 3

ein String sein kann, sortieren. Wir machen also dasselbe, wie MICROSOFT in der Stringspeicherung. Unsere Nummer ist ein Pointer, der auf einen String weist, oder auf ein höheres Datenobjekt, wie ein Randomfile. Falls wir nicht Strings sortieren, dann müssen wir vorerst die Suchschlüssel in die Stringma-

trix TXT\$(I) umspeichern. Der Array IO(I) ist im Falle der Randomfiles eine Liste mit den Recordnummern.

Mit diesen Massnahmen dürfte der Spielraum beim Sortieren mit BASIC weitgehend ausgeschöpft sein. Wohl hört man immer wieder von sagenhaften Sortierleistungen in noch kürzerer Zeit. Doch handelt es sich

dann entweder um Routinen in compiliertem BASIC, oder um Assembler-routinen die vom Interpreter aufgerufen werden. Eine weitere Kategorie von Sortierprogrammen nützt spezielle Eigenschaften von Datenfeldern aus, wie die, dass das Feld bis auf ein einziges Element bereits sortiert ist, weil bereits bei der Dateneingabe vorsortiert wurde.

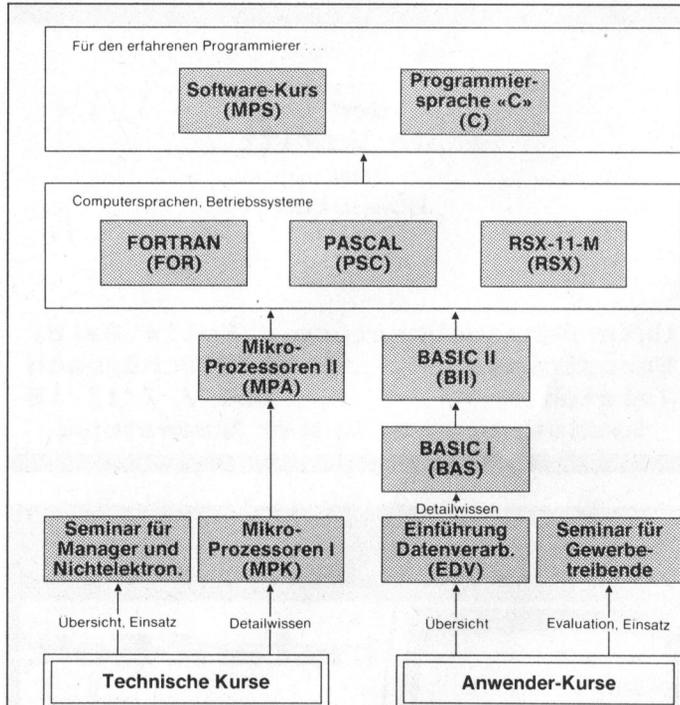
```

2281 DEFINT I          ' I.. SIND INTEGERVARIABLE
2282 NMAX=1000        ' 1000 ZU SORTIERENDE ELEMENTE
2283 DIM TXT$(NMAX)   ' SORTIERSCHLUESSEL
2284 DIM IQR(20), IQL(20) ' LISTEN
2285 DIM IO(NMAX)     ' ORDNUNGSNUMMERN
2286 '
2290 '-----QUICKSORT 4
2300 FOR I=1 TO N:IO(I)=I:NEXT I
2310 IS=1:IQL(1)=1:IQR(1)=N
2320 IL=IQL(IS):IR=IQR(IS):IS=IS-1
2330 I=IL:IJ=IR:IX=IO((IL+IR)\2)
2340 WHILE TXT$(IO(I))<TXT$(IX):I=I+1:WEND
2350 WHILE TXT$(IX)<TXT$(IO(IJ)):IJ=IJ-1:WEND
2360 IF I<=IJ THEN SWAP IO(I),IO(IJ):I=I+1:IJ=IJ-1
2370 IF I<=IJ THEN 2340
2380 IF I<IR THEN IS=IS+1:IQL(IS)=I:IQR(IS)=IR
2390 IR=IJ
2400 IF IL<IR THEN 2330
2410 IF IS<>0 THEN 2320
2420 ' ENDE QUICKSORT 4
    
```

Bild 5 Quicksort 4

## LITERATUR

- (1) Algorithmen und Datenstrukturen  
Niklaus Wirth  
Teubner, Stuttgart 1979
- (2) Compiler - Aufbau und Arbeitsweise  
Hans-Jürgen Schneider  
Walter de Gruyter, Berlin, New York 1975



## Dabei sein . . .

setzt eine entsprechende Ausbildung voraus. Unsere Lehrkräfte zeigen Ihnen gerne, was Mikroprozessoren und Mikrocomputer sind, was sie können, wo sie eingesetzt werden sollten, wo Probleme liegen.

Unser Kursangebot ist strukturiert: Neben Grundkursen (MPK, BAS) führen wir laufend diverse Aufbaukurse durch. Die Kursdaten teilen wir Ihnen gerne mit.

### Technische Kurse

- **Mikroprozessoren I (MPK):**  
Fachkurs für Elektroniker (14 Abende oder 5 Tage). Voraussetzung: Digitaltechnik
- **Mikroprozessoren II (MPA):**  
Fortsetzungskurs zu MPK (14 Abende oder 6 Tage). Voraussetzung: Kenntnisse entsprechend Grundkurs
- **Seminar für Manager und Nichtelektroniker (MMA):**  
1-Tages-Seminar, Orientierung über Mikroprozessoren

### Anwender-Kurse

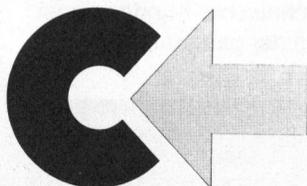
- **Einführung in die Datenverarbeitung (EDV)**  
Einführungskurs in die Prinzipien und Methoden der modernen Datenverarbeitung für EDV-Neulinge.
- **BASIC-I (BAS):**  
Einführungskurs für EDV-Anfänger. Erlernung der Programmiersprache BASIC.
- **BASIC-II (BII):**  
Fortgeschrittenen-Kurs für Anwender, welche BASIC bereits gut kennen.
- **Seminar für Gewerbetreibende (SKC):**  
Orientierung über Möglichkeiten des praktischen Einsatzes heutiger Kleincomputer.

### Fortgeschrittenen-Kurse

(setzen Vorkenntnisse voraus, nicht für Anfänger geeignet, Englischkenntnisse vorteilhaft)

- **PASCAL (PSC):**  
Programmiersprache PASCAL in Theorie und Praxis
- **FORTRAN (FOR):**  
Programmiersprache FORTRAN für technische Anwender
- **RSX-11-M (RSX):**  
PDP-11-Betriebssystem RSX-11-M für System-Spezialisten
- **Software-Kurs (MPS):**  
Software-Engineering, Software-Verfahren, Methodik, Organisation (für den fortgeschrittenen Anwender resp. Programmierer)
- **Programmiersprache 'C' (C)**  
Grundlagen und Anwendung der Programmiersprache 'C' der Bell Labs (USA) für Steuerungen und Systemprogrammierung

Fordern Sie unser Kursprogramm, Detail-Beschreibungen obiger Kurse sowie Anmeldekarten bei unserem Sekretariat (Tel. 01 / 461 12 13) an.



**Computerschule  
Zürich**

**Digicomp AG**

Birmensdorferstr. 94, 8003 Zürich  
Tel. 01 / 461 12 13, Telex 812035  
Informatik-Kurse seit 1976

## THE SYSTEM 2800 FAMILY OF MICROCOMPUTERS



### FEATURES

- IEEE S100 Bus Compatibel, Z80A Basiert
- 8" Drives, 8" Winchester 20 Mbyte
- 8 Slot Motherboard
- Standard 2 Serielle Schnittstellen
- System Software CP/M, MP/M oder OASIS
- Single User oder Multi User Facilitaeten
- Bis 12 Benutzer unter MP/M, High Throughput
- Exzellent für Datenbank wie dBase
- Full System Support unter Business Express

Ausgezeichnete Merkmale, wie ein superschnelles Exekutieren, machen das SYSTEM 2800 zu einem deutlichen «LEADER». Tatsache ist, dass SYSTEM 2800 eines der schnellsten Z80 Systeme ist, das momentan auf dem Markt ist.

**Weitere Vorteile:** Boot von jedem Drive oder Harddisk, ausgezeichnete Error Recovery, IEEE-S100-BUS, high Throughput unter Multi-User, DMA Floppy Controller, extrem einfacher Service mit Kontrollmöglichkeiten usw.



ZEV ELECTRONIC AG  
TRAMSTRASSE 11  
CH - 8050 ZÜRICH  
TELEPHON 01 312 22 67  
MIKROCOMPUTER



**Neu! Jetzt auf Deutsch!**

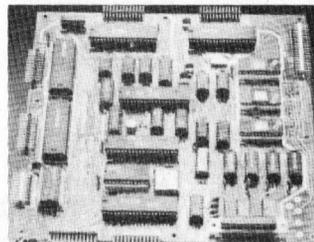
Die unentbehrliche Zusammenfassung aller CP/M-Befehle, die zu jedem CP/M-Computer gehört!

# CP/M Befehlsliste

Die CP/M-Befehlsliste im praktischen Taschen-Format erhalten Sie **gegen Voreinsendung einer 10-Franken-Note.**

DIALOG COMPUTER  
TREUHAND AG  
Seeburgstrasse 18  
6002 Luzern  
Telefon 041 - 31 53 33

## LOW COST Z-80 SINGLE Board Computer



- 72 Parallel I/O Lines
  - 12 K EPROM 2708/16/32
  - 2 K RAM 2114
  - 8 16 Bit Counter Timer
  - 2 Serial I/O Ports
  - 1 High Speed Arithmetic Processor AMD 9511/12
- Platine mit Monitor-ROM kann einzeln bezogen werden

**ANCRONA**   
Militärstrasse 8

Postfach 8021 Zürich  
Tel. 01/242 30 77

## Mini-Disketten 5 1/4"

- 40 + 77 Spur, einzeln getestet
- speziell abriebfest = lange Lebensdauer
- mit Verstärkungsring

**Günstige  
Preise!**



## 8"-Disketten

- 100% fehlerfrei

**Plastikboxen und weitere  
Ablegesysteme für Disketten**

**KONTVA AG**

Gotthardstrasse 40, 8800 Thalwil, Telefon 01 / 720 10 26

der  
erste Schritt  
zum  
Computer

**Unternehmensberatung Felix Beldi**  
Bernstrasse 64 CH-3535 Schüpbach  
Telefon 035 / 7 17 18

Dienstleistungen auf dem Gebiet der Datenverarbeitung

**commodore**  
COMPUTER



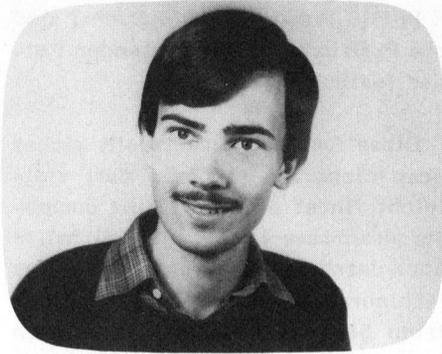
...und  
Instant-Soft  
lösen  
administrative  
Probleme.

**Instant-Soft**

... das Systemhaus:  
Hardware-Software-Schulung  
technischer Kundendienst  
in der ganzen Schweiz.



Instant-Soft AG  
Stetterstrasse 25 Seftigenstrasse 17  
5507 Mellingen 3007 Bern  
Tel. 056 - 912 021 Tel. 031 - 452 141



## Mühle im Endspiel

Stefan RAMSEIER

### 2. Teil

Nachdem in der letzten Folge die Theorie des Endspiels beschrieben wurde, sollen nun noch einige Gedanken zur Realisierung eines Pascal-Programms erörtert werden. Weiter werden einige Grafik-Prozeduren beschrieben, die das Mühle-Brett auf dem Bildschirm des Apple darstellen.

### PROGRAMMABLAUF

Bild 12 zeigt einen Teil des vereinfachten Flussdiagramms des Mühle-Endspiels, Bild 11 den entsprechenden Ausschnitt aus dem PASCAL-Programm.

Die Wörter "Mühlegeschlossen", "Mühleschliessen", "Mühledroht", "Doppelmühle", "Mühleverhindern", "Drohendedoppelmühle" und "Um..Ecken" bezeichnen Funktionen der Form

Function Mühlegeschlossen:Boolean;

Steht der Programmzeiger auf der ersten Zeile von Bild 11, wird die Funktion "Mühlegeschlossen" ausgeführt. Dort überprüft der Computer, ob der Gegner mit seinem letzten Zug eine Mühle geschlossen hat. Ist dies der Fall, erhält "Mühlegeschlossen" den Wert "True" (wahr), andernfalls den Wert "False" (falsch).

Durch die Möglichkeit, solche Funktionen zu benützen, wird der Programmaufbau wesentlich übersichtlicher.

Ist nämlich eine solche Funktion einmal programmiert und getestet, kann sie "en bloc" weiterverwendet werden. Der Programmierer muss dann nicht mehr wissen, was sich in dieser Funktion genau abspielt; es genügt die Kenntnis, unter welchen Bedingungen das Resultat "wahr" oder "falsch" ist.

Die meisten Funktionsnamen sprechen für sich, so dass nicht mehr auf sie eingegangen wird.

Die Prozedur "Gegnerreaktion" verdient jedoch noch besondere Beachtung: In ihr werden die Reaktionen des Gegners berücksichtigt, wie dies im Abschnitt "weitere Gewinnmöglichkeiten" beschrieben worden ist.

Findet nun der Computer keinen Gewinnzug, wählt er denjenigen Zug aus, bei dem sein menschlicher Gegner am ehesten einen Fehler begeht!

Dabei stützt sich der Computer auf die Tatsache, dass z.B. die Stellung "um 4 Ecken" schwieriger zu erkennen ist als die "Doppelmühle".

Diese Taktik hat sich in der Praxis sehr gut bewährt: Selbst der Autor des Programms verliert mit den schwarzen Steinen in etwa 20 Zügen; ungeübte Spieler halten kaum 10 Züge durch.

Hingegen ist es sehr schwierig und zeitaufwendig, mit den weissen Steinen den Computer zu besiegen, weil sich dieser fast immer optimal verteidigt. Oft kommt auch der Fall vor, dass der angreifende Spieler

einen Fehler begeht und der Computer dann sogar mit den schwarzen Steinen gewinnt!

### DATENSTRUKTUR

Das Spielfeld wird in diesem Programm wie folgt gespeichert:

```
VAR
M:ARRAY[1..24] OF INTEGER;
D:ARRAY[0..4,1..24] OF INTEGER;
```

Die Variable M entspricht dem Spielbrett: M[1] bezeichnet den Punkt A, M[2] den Punkt B usw., M[24] den Punkt X.

Dabei kann jeder M[Punkt] drei verschiedene Werte annehmen, je nachdem ob sich darauf kein, ein eigener oder ein gegnerischer Stein befindet.

Die Geometrie des Spielbrettes ist im Array D festgehalten (siehe Bild 13).

D 0,Punkt gibt an, auf wieviele Felder man von "Punkt" aus direkt fahren kann (2,3 oder 4).

```

IF NOT MUEHLEGESCHLOSSEN
THEN IF NOT MUEHLESCHLIESSEN
    THEN IF NOT MUEHLEDROHT
        THEN BEGIN IF NOT DOPPELMUEHLE
                    THEN IF NOT UM2ECKEN
                        THEN IF NOT UM3ECKEN
                            THEN IF NOT UM4ECKEN
                                THEN GEGNERREAKTION
                            END
                    ELSE IF MUEHLEVERHINDERN
                        THEN IF NOT DROHENDEDOPPELMUEHLE
                            THEN VERTEIDIGEN
        END
    END
END

```

Bild 11

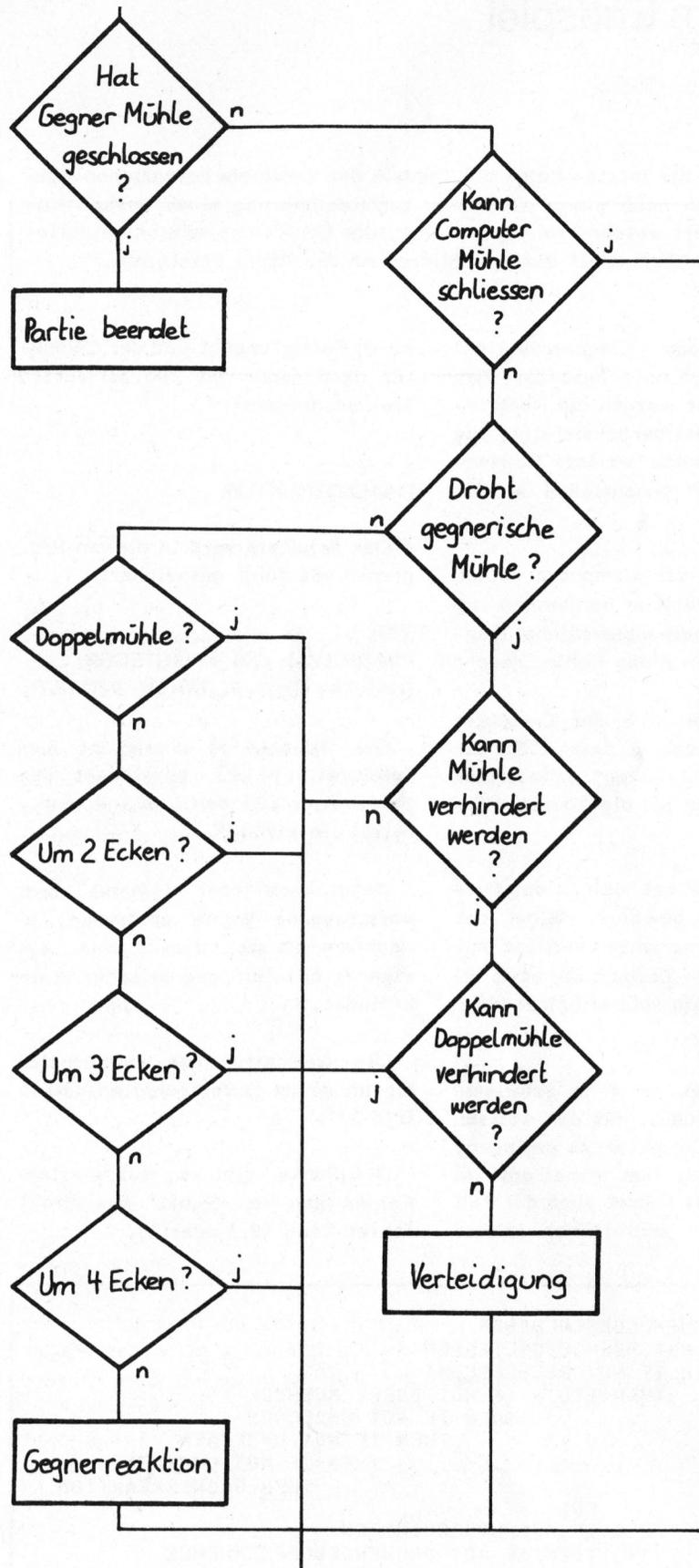


Bild 12 Vereinfachtes Flussdiagramm des Mühlespiels

In D[1,Punkt] bis D[4,Punkt] sind die Positionen der umliegenden Felder festgelegt.

Diese Datenstruktur stellt sicher noch nicht das Optimum dar; vielleicht findet ein Leser eine computergerechtere Lösung. Dabei sollte auch daran gedacht werden, dass das Mühlebrett 16 verschiedene Symmetrien besitzt. D.h. es kann durch Drehung, Spiegelung und "Verzerrung" (Ring A-B-C-O-X-W-V-J wird nach innen und Ring G-H-I-M-R-Q-P-L nach aussen gebogen) auf sich selbst abgebildet werden.

Es sollte noch vermerkt werden, dass die Wahl der Datenstruktur keinen Einfluss auf die Spielstärke, sondern nur programmtechnische Bedeutung hat (Rechenzeit, Programmieraufwand).

#### DAS PROGRAMM

Das Dreistein-Mühle-Programm, das sämtliche in der letzten Folge beschriebenen Taktiken beherrscht,

#### KORRIGENDA

(Red.) In unserer letzten Ausgabe haben sich im ersten Teil von "Mühle im Endspiel" zwei sinnentstellende Fehler eingeschlichen. Sorry, auch wir kochen nur mit Wasser, richtig muss es heissen:

1. p. 70, linke Spalte, zweitunterster Abschnitt sollte heissen: "Es droht also die Mühle S-T-U, die Schwarz mit B-T oder V-T verhindern kann. Nun kann Weiss aber mit S-F eine Doppelmühle öffnen und gewinnt die Partie."

2. p. 71 linke Spalte unten: Variante 2 (Schwarz zieht J-E) statt Z-E

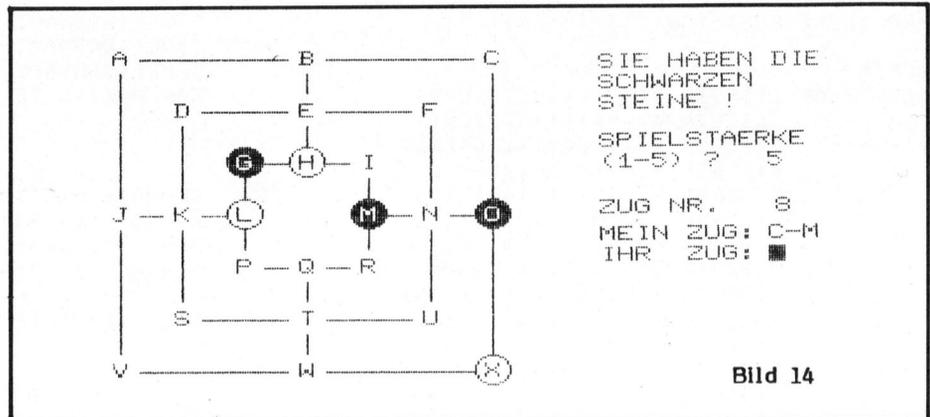
1. G-B J-E
2. H-A W-J Doppelmühle
3. V-C

D :	0	1	2	3	4
1	2	2	10	3	22
2	3	1	5	3	8
3	2	2	15	1	24
4	2	5	11	6	19
5	4	4	2	6	8
6	2	5	14	4	21
7	2	8	12	9	16
8	3	7	5	9	2
9	2	8	13	7	18
10	3	1	11	22	12
11	4	4	10	19	12
12	3	7	11	16	10
13	3	9	14	18	15
14	4	6	13	21	15
15	3	3	14	24	13
16	2	12	17	7	18
17	3	16	20	18	23
18	2	17	13	16	9
19	2	11	20	4	21
20	4	19	17	21	23
21	2	20	14	19	6
22	2	10	23	1	24
23	3	22	20	24	17
24	2	23	15	22	3

Bild 13

und mit dem auch Endspielstudien gelöst werden können, läuft auf einem Apple und benötigt etwa 27K (P-Code) Speicherplatz. Es besitzt 5 Spielstärken; die Rechenzeit pro Zug beträgt etwa 10 Sekunden bis Stufe 4 (ohne "Gegnerreaktion") und ca. 100 Sekunden auf Stufe 5 (mit "Gegnerreaktion").

Als Demonstration sie eine kurze Dreistein-Partie notiert, welche die Spielstärke des Computers eindrücklich aufzeigt.



SIE HABEN DIE SCHWARZEN STEINE !

SPIELSTÄRKE (1-5) ? 5

ZUG NR. 8  
MEIN ZUG: C-M  
IHR ZUG: ■

Bild 14

Der Computer hat die weissen Steine:

1. E D Schwarz muss B, D, F oder H setzen
2. H B
3. G I
4. H-P B-L Schwarz könnte auch D-L ziehen
5. P-B I-H besser wäre D-H
6. E-C D-A einzig richtiger Zug von Schwarz
7. B-O A-X einzig richtiger Zug von Schwarz
8. C-M! X-N nach C-M ist Schwarz verloren (s. Bild 14)
9. G-X H-C um 4 Ecken (X-W-Q-R)
10. O-W C-V um 3 Ecken (W-Q-R)
11. X-Q V-T um 2 Ecken (Q-R)
12. W-R T-P Doppelmühle
13. G-I Weiss gewinnt

In Bild 14 ist die Stellung nach dem 8. Zug von Weiss dargestellt. Aus drucktechnischen Gründen sind die Farben Weiss und Schwarz vertauscht.

und werden bei der Besprechung der einzelnen Prozeduren erklärt.

#### PROZEDUR POSITION

In dieser Prozedur wird berechnet, welche Koordinaten ein Punkt des Spielfeldes auf dem Apple-Bildschirm hat. Der Nullpunkt des Koordinatensystems liegt in der linken unteren Bildschirmcke. Der Parameter dieser Funktion kann die Werte 1..24 annehmen, entsprechend den Feldern A..X auf dem Mühle-Brett.

#### PROZEDUREN STUFF, STEINWEISS UND STEINSCHWARZ

Diese Prozeduren erzeugen zwei Shapes, die dem weissen und dem schwarzen Stein entsprechen. Die Variablen "Weiss" und "Schwarz" bilden je ein Quadrat von 15\*15 Punkten, wobei "Weiss" einen ausgefüllten Kreis und "Schwarz" eine geschlossene Kreislinie darstellen.

#### GRAFIK-PROZEDUREN

Die auf den folgenden Seiten abgedruckten Prozeduren stellen das Mühle-Brett und die Spielsteine grafisch dar.

Die benötigten Variablen haben die Form

```
Type Picture=Packed
Array [L.15,L.15] of Boolean;
Var Weiss,Schwarz:Picture;
X,Y,ROW:INTEGER;
```

Die in den Prozeduren Steinweiss und Steinschwarz in Form von Strings enthaltene Information über die Spielsteine wird in der Prozedur Stuff in das gewünschte "Packed Array of Boolean" umgewandelt. Dieses kann später mit "Drawblock" an jede beliebige Stelle des Bildschirms gezeichnet werden (siehe Bild 14).

Die Spielsteine müssen nur zu Beginn des Programms, in der Initialisierungsphase, erzeugt werden.

# GEWUSST WIE!

```
PROCEDURE POSITION(I:INTEGER);
```

```
BEGIN
CASE I OF 1:BEGIN X:=20;Y:=170;END;
2:BEGIN X:=95;Y:=170;END;
3:BEGIN X:=170;Y:=170;END;
4:BEGIN X:=45;Y:=145;END;
5:BEGIN X:=95;Y:=145;END;
6:BEGIN X:=145;Y:=145;END;
7:BEGIN X:=70;Y:=120;END;
8:BEGIN X:=95;Y:=120;END;
9:BEGIN X:=120;Y:=120;END;
10:BEGIN X:=20;Y:=95;END;
11:BEGIN X:=45;Y:=95;END;
12:BEGIN X:=70;Y:=95;END;
13:BEGIN X:=120;Y:=95;END;
14:BEGIN X:=145;Y:=95;END;
15:BEGIN X:=170;Y:=95;END;
16:BEGIN X:=70;Y:=70;END;
17:BEGIN X:=95;Y:=70;END;
18:BEGIN X:=120;Y:=70;END;
19:BEGIN X:=45;Y:=45;END;
20:BEGIN X:=95;Y:=45;END;
21:BEGIN X:=145;Y:=45;END;
22:BEGIN X:=20;Y:=20;END;
23:BEGIN X:=95;Y:=20;END;
24:BEGIN X:=170;Y:=20;END;
```

```
END;
END; (* POSITION *)
```

```
PROCEDURE STUFF(VAR BITMAP: PICTURE; S:STRING);
VAR J: INTEGER;
    BIT:BOOLEAN;
```

```
BEGIN
FOR J:=1 TO LENGTH(S) DO
BEGIN
BIT:=(S[J]<>' ');
BITMAP[ROW,J]:=BIT;
BITMAP[ROW,16-J]:=BIT;
BITMAP[16-ROW,J]:=BIT;
BITMAP[16-ROW,16-J]:=BIT;
END;
ROW:=ROW+1;
```

```
END; (* STUFF *)
```

```
PROCEDURE STEINWEISS;
```

```
BEGIN
ROW:=1;
STUFF(WEISS,' XX');
STUFF(WEISS,' XXXX');
STUFF(WEISS,' XXXXX');
STUFF(WEISS,' XXXXXX');
STUFF(WEISS,' XXXXXX');
STUFF(WEISS,' XXXXXX');
STUFF(WEISS,' XXXXXX');
FOR ROW:=1 TO 15 DO BEGIN
WEISS[ROW,8]:=TRUE;
WEISS[8,ROW]:=TRUE;
END;
```

```
END; (* STEINWEISS *)
```

```
PROCEDURE STEINSCHWARZ;
```

```
BEGIN
ROW:=1;
STUFF(SCHWARZ,' XX');
STUFF(SCHWARZ,' XX ');
STUFF(SCHWARZ,' X ');
STUFF(SCHWARZ,' X ');
```

```
STUFF(SCHWARZ,' X ');
STUFF(SCHWARZ,'X ');
STUFF(SCHWARZ,'X ');
FOR ROW:=2 TO 14 DO BEGIN
SCHWARZ[ROW,8]:=FALSE;
SCHWARZ[8,ROW]:=FALSE;
END;
```

```
SCHWARZ[8,1]:=TRUE;
SCHWARZ[8,15]:=TRUE;
SCHWARZ[15,8]:=TRUE;
SCHWARZ[1,8]:=TRUE;
END; (* STEINSCHWARZ *)
```

```
PROCEDURE ZEICHNE(X,Y,I1:INTEGER);
```

```
VAR B:BOOLEAN;
    K:INTEGER;
BEGIN
DRAWBLOCK(B,2,0,0,15,15,X-7,Y-6,0);
PENCOLOR(NONE);
MOVETO(X-3,Y-3);
CH:='A';
K:=0;
WHILE K < I1 DO BEGIN CH:=SUCC(CH);
K:=K+1;
```

```
END;
```

```
WCHAR(CH);
```

```
END; (* ZEICHNE *)
```

```
PROCEDURE ZEICHNECHAR;
```

```
VAR I1:INTEGER;
BEGIN
FOR I1:=1 TO 24 DO BEGIN POSITION(I1);
ZEICHNE(X,Y,I1-1);
END;
```

```
END; (* ZEICHNECHAR *)
```

```
PROCEDURE ZEICHNEBRETT(COLOR:SCREENCOLOR);
```

```
BEGIN
INITTURTLE;
TEXTMODE;
MOVETO(20,20);
PENCOLOR(COLOR);
MOVETO(20,170);MOVETO(170,170);MOVETO(170,20);
MOVETO(20,20);
PENCOLOR(NONE);
MOVETO(45,45);
PENCOLOR(COLOR);
MOVETO(45,145);MOVETO(145,145);MOVETO(145,45);
MOVETO(45,45);
PENCOLOR(NONE);
MOVETO(70,70);
PENCOLOR(COLOR);
MOVETO(70,120);MOVETO(120,120);MOVETO(120,70);
MOVETO(70,70);
PENCOLOR(NONE);
MOVETO(20,95);
PENCOLOR(COLOR);
MOVETO(70,95);
PENCOLOR(NONE);
MOVETO(120,95);
PENCOLOR(COLOR);
MOVETO(170,95);
```

Dies geschieht mit den Befehlen Steinweiss; Steinschwarz; die

## PROZEDUR ZEICHNEBRETT

zeichnet das Spielbrett in einer der Farben "white", "black", "reverse", "green", "violet", "orange" oder "blue". Danach werden die

## PROZEDUREN ZEICHNE UND ZEICHNECHAR

aufgerufen, welche das Spielbrett mit den Buchstaben A..X beschriften. Das Resultat dieses Vorgangs ist in Bild 1 gezeigt (Mikro+Kleincomputer 82-2 p. 69).

## PROZEDUR ZEICHNESTEIN

Diese Prozedur zeichnet einen Spielstein auf das Spielbrett. Die Farbe des Steins und seine Koordination werden mit den Parametern "Farbe" und "POS" festgelegt.

Der betreffende Stein wird mit "Drawblock" fünf Mal "EXOR" auf den Bildschirm gezeichnet. Dadurch scheint der Stein dreimal zu blinken und bleibt dann stehen.

```
FENCOLOR(NONE);
MOVETO(95,20);
FENCOLOR(COLOR);
MOVETO(95,70);
FENCOLOR(NONE);
MOVETO(95,120);
FENCOLOR(COLOR);
MOVETO(95,170);
ZEICHNECHAR;
GRAFMODE;

END; (* ZEICHNEBRETT *)
```

```
PROCEDURE ZEICHNESTEIN(FARBE:STRING;POS:INTEGER);
VAR I,J:INTEGER;
BEGIN
POSITION(POS);
FOR I:=1 TO 5 DO BEGIN
IF FARBE='WEISS'THEN DRAWBLOCK(WEISS,2,0,0,15,15,X-7,Y-6,6)
ELSE DRAWBLOCK(SCHWARZ,2,0,0,15,15,X-7,Y-6,6);
FOR J:=1 TO 500 DO;
END;
END; (* ZEICHNESTEIN *)
```

1. M-G! F-I oder U-I
  2. N-L \*-P
  3. G-J \*-K
  4. L-A \*-V
  5. J-B \*-C
  6. A-E
- oder
1. M-G! O-I!
  2. H-L \*-P
  3. G-J \*-K
  4. L-V \*-A
  5. J-X \*-W
  6. V-O \*-M
  7. N-C

"Drawblock" ist übrigens im Apple Pascal Language-Reference-Manual auf Seite 96 f. ausführlich beschrieben.

Mit diesen Prozeduren kann auf dem Apple mit kleinem Aufwand eine recht ansprechende Grafik erzeugt werden.

## AUFLOESUNG DER AUFGABE AUS MIKRO+KLEINCOMPUTER 82-2

Weiss: H, M, N  
Schwarz: F, O, U  
Gewinnzug ist M-G.

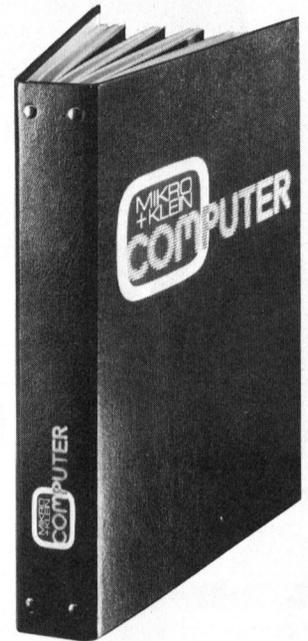
"\*" bezeichnet einen beliebigen schwarzen Stein.

## LITERATUR

Apple Pascal Language Reference Manual P. 90-100 (Turtlegraphics)

Es lohnt sich auch, das File Grafdemo.Text, welches sich auf der Diskette Apple3: befindet, zu studieren.

**Jetzt  
wieder  
erhältlich!**



## der praktische Sammelordner

mit einem strapazierfähigen Kunststoffüberzug in ansprechender blauer Farbe und bequemer Stabmechanik für jeweils sechs Nummern **Mikro + Kleincomputer** (also ein ganzer Jahrgang), damit jedes einzelne Heft unbeschädigt bleibt.

Diesen praktischen Sammelordner können Sie für nur Fr. 14.50 inkl. Versandkosten bestellen. Übrigens, zwei Exemplare kosten bei gleichzeitiger Bestellung nur noch Fr. 27.-. Zahlen Sie bitte den entsprechenden Betrag auf unser **Postkonto Luzern 60-27181** ein und vermerken Sie auf der Rückseite Ihres Einzahlungsscheins «Sammelordner».

**Mikro+Kleincomputer  
Informa Verlag AG  
Postfach 1401  
CH-6000 Luzern 15**

**ERNI**

## Raffiniert und Leistungsstark



### Standard Ausrüstung:

- Bidirektionaler Nadeldrucker mit Farb-  
bandkassette
- Anpresswalze und Formtraktor
- 100 Charakter/Sek. Druckgeschwindigkeit
- 8 verschiedene Schriftgrößen
- 96 Zeichen ASCII Zeichensatz
- Format Steuerung
- Serielle Schnittstelle (RS 323c)
- 1K Puffer Speicher
- Parallele Schnittstelle (Centronics)

**Fr. 950.—**

Der E 88 T wird auch mit **Graphic-** und weiteren Optionen als Version E 88 G geliefert.

**Fr. 1150.—**

**ERNI + Co. Elektro-Industrie**  
CH-8306 Brütisellen Tel. 01/833 33 33

## Bächler-Sidler AG Luzern

Endlosformular-, Block- und Garniturenfabrik, Druckerei Sagenmattstrasse 7 6002 Luzern Telefon 041-22 71 71 Telex 78 218

### Endlosformulare

### Snapband

### Einzelgarnituren

### Snapout



### Kassenblöcke

### Durchschreibeblocke

### Geschäftsdrucksachen

**BSL**

## PET/CBM/VC 20-Besitzer

Kennen Sie SYNTAX – das Programm-Magazin auf Kassette?

Es bringt jeden Monat 5 neue Programme in deutscher Sprache aus allen Bereichen. Zum Beispiel Dateisysteme, Textverarbeitung, Lehrgang Maschinensprache, User-Programme usw. Kenner der SYNTAX-MAGAZINE loben Leistung und Preis.

Ab Januar 1982 erscheint das erste SYNTAX-PROGRAMM-MAGAZIN auf Kassette für VC 20.

Fordern Sie gleich heute noch kostenlose Informationen von



Soft- und Hardware GmbH

P. B. 1609  
D-7550 Rastatt  
Tel. 07222/34296

## HOCHQUALIFIZIERTE Hardware und Software

### 16 bit Computer:

Prozessor:	8088
Bildschirm:	25*80 oder 50*132, Vollgrafik
Interner Speicher:	128 - 1024 Kbyte
2 Mini-Floppies:	1.2 - 2.4 Mbyte
Betriebssystem:	CP/M oder MDOS

ab Fr. 11 500.—

### Standard-Software:

Finanzbuchhaltung mit WUST und Fremdwährungen ab	Fr. 2400.—
Adressverwaltung mit Roboterbrief	Fr. 900.—
Benutzerfreundliche Textverarbeitung	Fr. 1100.—

Individuelle Programmanfertigungen.  
Leichtverständliche deutsche Dokumentation.

Roman Kaiser	Würzenbachstrasse 62
EDV Beratung/Software	6006 Luzern, Telefon 041 - 31 49 69

## Sorcerer-Tips

In loser Folge bringen wir unter dem Titel "Sorcerer-Tips" Auszüge aus dem Informationsblatt der aktiven Sorcerer-Gruppe. Auch der folgende Beitrag dürfte bei vielen unserer Leser auf starke Interesse stossen. Wer sich speziell für Sorcerer interessiert, kann sich an Herrn Werner Gribi, Industriestrasse 9, CH-3254 Büren a.A. wenden.

Mit diesem Programm kann ein Teil des Bildschirms für Ausgaben gesperrt werden. Nur in einem vom Benutzer definierten Bereich wird auf den Bildschirm geschrieben, bzw. wird der Bildschirm gelöscht. Das Programm arbeitet mit dem Basic-Rom-Pac zusammen. Es kann aber auch in reinen Maschinenprogrammen verwendet werden.

### ARBEITSWEISE

Nach dem Aufruf der Initialisierungsroutine in 330H sind die Input- und Outputvektoren so verändert, dass sie nicht mehr direkt in den Monitor sondern zuerst in das Textfensterprogramm führen.

Der grosse Vorteil dieses Programms liegt darin, dass das Text-

fenster kann irgendwo auf dem Bildschirm definiert werden kann und nicht nur am unteren Bildschirmrand, wie dies beim Apple II der Fall ist.

### INSTALLATIONSANLEITUNG

Befolgen Sie genau die Anleitung, da ein kleiner Fehler die Sache zum Scheitern verurteilen kann!

- 1) S-Basic-Pac einstecken, mit "BYE" in den Monitor gehen und das Programm ab 01D0 Hex eingeben.

## SORCERER - Bildschirmatlas

POKE A,Z / A = 64 \* Y + X - 3968 / Z = ASCII - CODE

o Mitte 14/31 = - 3041

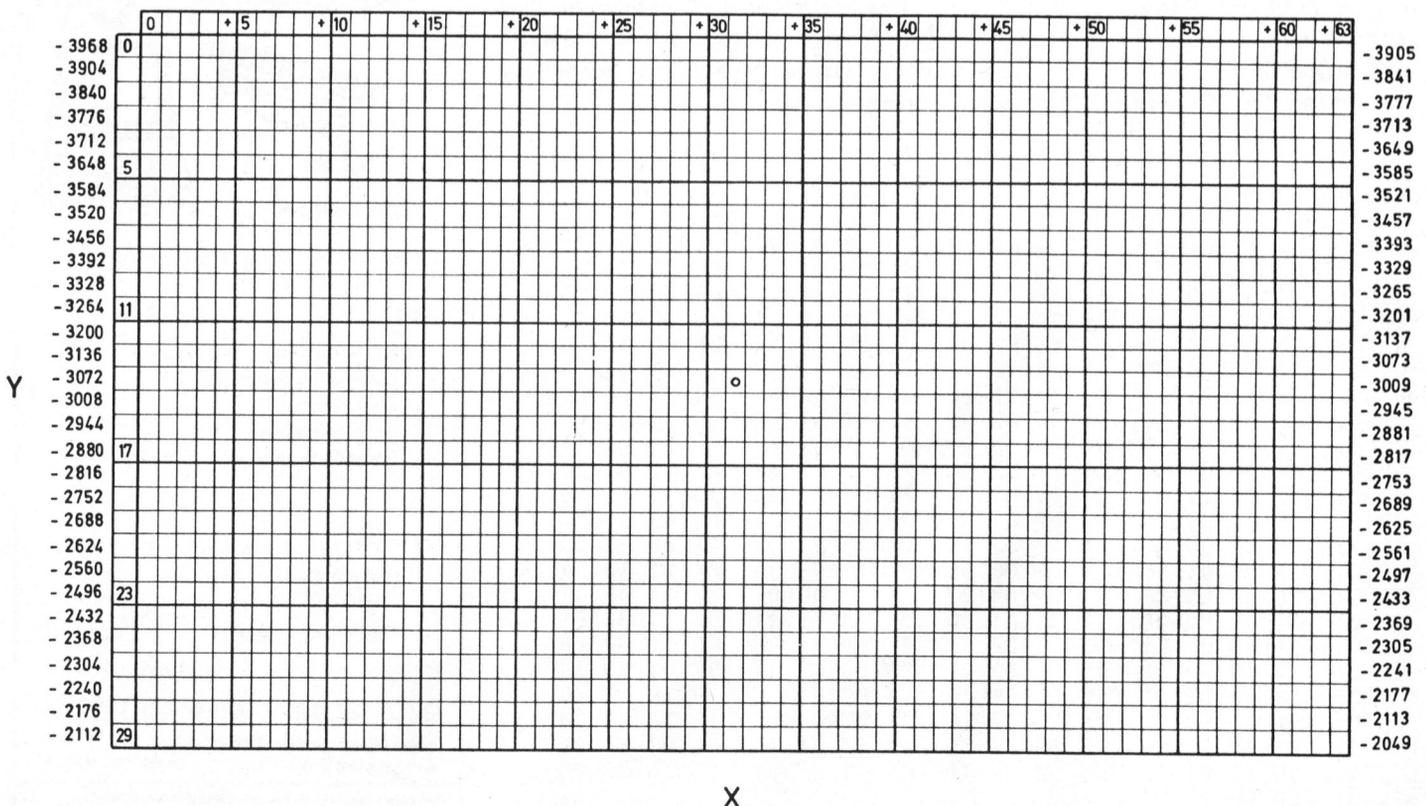


Bild 1 Jeder Bildpunkt hat seine eigene Adresse

2) EN 0149 CR 40 03 / CR :  
Schützen des Programms  
PP CR  
NEW CR

3) BYE CR  
SA WIND 0140 0342 CR :  
Erstellen einer Programmkopie  
PP CR

4) Programm eingeben.

## PROGRAMMIERUNG

Nun können Sie das Textfenster programmieren. POKEN Sie hierzu die folgenden Speicherstellen mit den durch den Kommentar festgelegten Werten.

- 473 (Links)  
Abstand vom linken Rand
- 474 (Breite)  
Breite des Textfensters
- 475 (Oben)  
Abstand der ersten Linie vom oberen Rand
- 476 (Unten)  
Abstand der letzten Linie vom oberen Rand
- 477 (Links)
- 478 (Oben)
- 322 muss mit 255 gePOKEt werden.

Der Programmanfang sieht zum Beispiel so aus:

```
10 POKE 473,10:POKE 477,10
20 POKE 474,30:POKE 476,20
30 POKE 475,10:POKE 478,10:
   POKE 322,255
```

Dies ergibt ein Textfenster, das 10 Zeichen vom linken Bildschirmrand beginnt, 30 Zeichen breit ist und sich von der zehnten bis zur zwanzigsten Linie erstreckt. Jetzt folgt die Initialisierungslinie:

```
40 POKE 260,48:POKE 261,3:X=USR
   (0):PRINT"";
```

Die Programmierung des Textfensters ist nun fertig und es folgt das normale Programm.

```
50 LIST
```

Geben Sie nun RUN ein und beobachten Sie die Auswirkungen!

VORSICHT

Das Programm darf nicht mehr über CSAVE abgespeichert werden! Gespeichert wird es im Monitor folgendermassen:

```
DU 01B7 01B8 CR
```

Es resultiert die folgende Ausgabe:

```
ADDR 0 1 2 3 4 5 6 7 8
01B0: XX XX XX XX XX XX XX YY ZZ
```

Geben Sie nun ein:

```
SA Name 0140 ZZYY CR
```

Geladen wird es wie jedes andere Basic-Programm mit CLOAD.

In Bild 1 sehen Sie ein vereinfachtes Abbild des Sorcerer-Bildschirmes. Den Wert A für jeden beliebigen Bildpunkt erhält man durch eine einfache Addition der zur entsprechenden Zeile gehörenden Zahl Y (am rechten Bildrand notiert) und der Spaltenabhängigen Zahl X.

## Listing

```
01D0: E6 11 DF 01 00 80 F0 80 F0 00 40 00 1E 00 00 F5
01F0: DD 01 35 F0 3A DA 01 3D 32 DD 01 3A DB 01 21 DE
0200: 01 BE D0 35 3A DE 01 CD E8 02 21 D9 01 8E 32 D5
0210: 01 C9 FE 0C 28 27 FE 0D 28 47 FE 0A 28 49 FE 17
0220: 28 D9 FE 1A 28 41 FE 01 28 C5 FE 13 CA D6 02 FE
0230: 08 28 B4 FE 11 28 1E FE 20 D2 C4 02 C9 21 DD 01
0240: 46 3A DB 01 F5 CD 07 02 CD AE 02 06 00 F1 3C 21
0250: DC 01 BE 38 EF 3A DB 01 32 DE 01 3E 00 32 DD 01
0260: C9 AF 32 DD 01 18 9D 21 DE 01 34 7E EB 21 DC 01
0270: BE 38 91 EB 35 3A DB 01 F5 CD 07 02 2A D5 01 22
0280: D7 01 21 DA 01 46 F1 3C 21 DC 01 BE 30 14 F5 C5
0290: CD 07 02 C1 2A D7 01 48 06 00 EB 2A D5 01 ED B0
02A0: 18 DA 06 00 CD AE 02 D2 04 02 21 DD 01 46 3E 20
02B0: 2A D5 01 5F 48 06 00 09 77 41 04 3A DA 01 3D B8
02C0: 7B 30 ED C9 F5 CD 04 02 F1 21 DD 01 46 2A D5 01
02D0: 48 06 00 09 77 41 21 DD 01 34 7E 21 DA 01 BE F5
02E0: D4 61 02 F1 D2 67 02 C9 47 04 21 40 F0 11 40 00
02F0: 19 10 FD 22 D5 01 7D C9 E5 D5 C5 3A DE 01 CD E8
0300: 03 3A DD 01 16 00 5F 19 3A D9 01 5F 19 7E F5 36
0310: 5F 11 50 00 D5 CD 18 E0 D1 B7 20 0E 1B 7A B3 20
0320: F3 7E C1 C5 B8 28 E8 70 18 E7 C1 70 C1 D1 E1 C9
0330: CD A2 E1 11 F8 02 CD 31 E6 11 DF 01 C3 15 E6 00
0340: 00 00
```

Bild 2 Kompakte Darstellung des Maschinenprogrammes

MikroComputerSchule **MCSB**

**AUSBILDUNGS ANGEBOT**

Spezialkurs



Dateienverwaltung mit Floppy-Disk

**Inhalt:**

- . Dateiaufbau / -organisation
- . Datenübertragung
- . Zugriffsarten
- . Datei-Mutation
- . Geschützte Felder

**Dateien:**

- . sequentiell
- . relativ
- . sequentiell indexiert

**Dauer:** 6 Halbtage

**Durchführung:** September 1982

MikroComputerSchule  
Holestrasse 87, 4054 Basel Telefon 061/38 21 20

Verlangen Sie bitte unser aktuelles Kursprogramm D

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

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

## Grafik auf «normalem Superbrain»

Ing. Ferdinand HAIJSACKL

Die grosse Mehrzahl der auf dem Markt angebotenen Mikrocomputer für den kommerziellen Einsatz sind serienmässig ohne Bildschirmgrafik ausgerüstet. Im nachstehenden Beitrag zeigen wir Ihnen, wie man mit wenig Aufwand einen "normalen" SUPERBRAIN mit einer guten Semigrafik ausrüsten kann.

Dank einer kleinen technischen Aenderung lässt sich dem "normalen Superbrain" auf seinem Bildschirm eine beachtliche, mit den Tastaturzeichen mischbare Grafik entlocken. Die Punktauflösung ist zwar recht bescheiden (160x72), die Setzgeschwindigkeit aber relativ gross, da die übliche Funktion des Gerätes mit seiner optimalen Bildschirmsteuerung erhalten bleibt.

### DER GRUNDGEDANKE

Wie vielfach üblich, soll das MS-BIT eines Zeichenbytes folgendes bewirken:

**MS-BIT=0:** dem restlichen Zeichenbyte (7 Bit) soll ein ASCII-Zeichen entsprechen. Diese Funktion wird nicht geändert.

**MS-BIT=1:** dem restlichen Zeichenbyte soll ein "Grafik-Punktfeld" entsprechen. Im Normalfall wird damit ein blinkendes ASCII-Zeichen dargestellt.

An die Stelle des Zeichenattributes "Blinken" soll jetzt neu die Funktion "Grafik" treten. Der Videogenerator ist prinzipiell geeignet, die angestrebten "Grafik-Punktfelder", folgend kurz Punktfelder genannt, zu erzeugen (2x3 kleine Rechtecke pro Zeichenfeld).

### GERAETESPEZIFISCHE VORGABEN

Um die durchgeführten Aenderungen verstehen zu können, wollen wir die Funktionsweise anhand des Schaltungsausguges in Bild 1 kurz erläutern.

Zum Aufbau einer Zeichenreihe am Bildschirm zirkuliert die im Shiftregister MM 5053 enthaltene Information (80 Byte entsprechen 80 Zeichen pro Reihe) synchron mit der Bildschirmabtastung. Insgesamt 10 mal, entsprechend 10 Zeilen pro Reihe, wird dabei dem Videogenerator für den gleichen Zeilenort auch das gleiche Zeichenbyte zugeführt. Die Neuladung des Shiftregisters geschieht durch einen direkten Speicherzugriff des CRT-Controllers (Kennzeichen: RECirculate Enable = LOW, daher ist das Shiftre-

gister mit dem Datenbus verbunden; der CRT-Controller liefert während einer Zeilendauer 80 Adressen für einen zugehörigen Datenstrom in das Shiftregister.) Wichtig ist nun die zeitliche Lage dieser DMA-Phase in Bezug auf die 10 Zeilen einer Zeichenreihe. Abhängig vom zugeführten TTL-Pegel an Pin 11 des CRT-Controllers (Charakter Generator Program Input - CGPI) gibt es die beiden Betriebsarten:

**CGPI HIGH:**  
DMA während d.1.Zeile ... (1)

**CGPI LOW:**  
DMA während d.10.Zeile ... (2)

Der CRT-Controller arbeitet im SUPERBRAIN normalerweise nur in

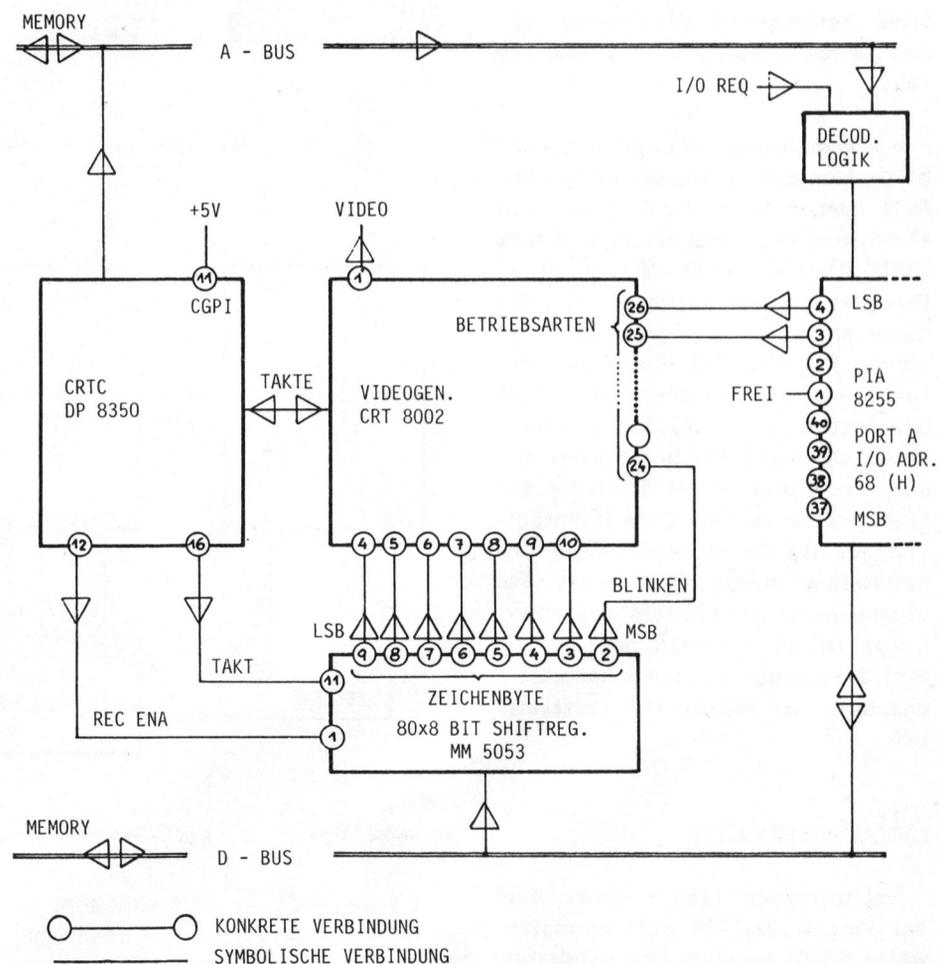
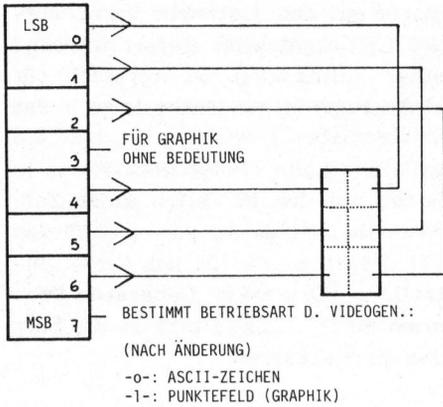


Bild 1: Schema des unveränderten Superbrains

ZEICHENBYTE

PUNKTEFELD BZW;  
ASCII-ZEICHENFELD



Das im Zeichenbyte gesetzte Bit erhält den über Pfeil erreichbaren Teil des Punktefeldes.

**Bild 2: Graphikbetrieb des Videogenerators**

der Betriebsart (1). Der Grafikbetrieb bedingt die Betriebsart (2) und zwingt entsprechend einzugreifen.

Wie angedeutet, muss auch die Arbeitsweise des Videogenerators CRT 8002 dem Wunsch, Punktefelder zu erzeugen, angepasst werden. Diese Grafik-Betriebsart des Videogenerators erreicht man durch einen LOW-Pegel an Pin 25 und 26 seiner Eingänge für die Betriebsartenwahl. Die genannten Eingänge sind über den Port-A der PIA 8255 (I/O Adresse (H)68) steuerbar und können somit versuchsweise per Software auf LOW gesetzt werden. Ohne jeden Eingriff in das Gerät kann so die Arbeitsweise dieser Betriebsart des Videogenerators beobachtet werden. Dabei ist im wesentlichen der im Bild 2 gezeigte Zusammenhang Zeichenbyte zu Punktefeld feststellbar.

## DIE ÄNDERUNGEN

Nachzutragen bleibt noch, dass der Port-A des PIA 8255 normalerweise (H)03 ausgibt. Die Änderung nimmt auch darauf Rücksicht und der

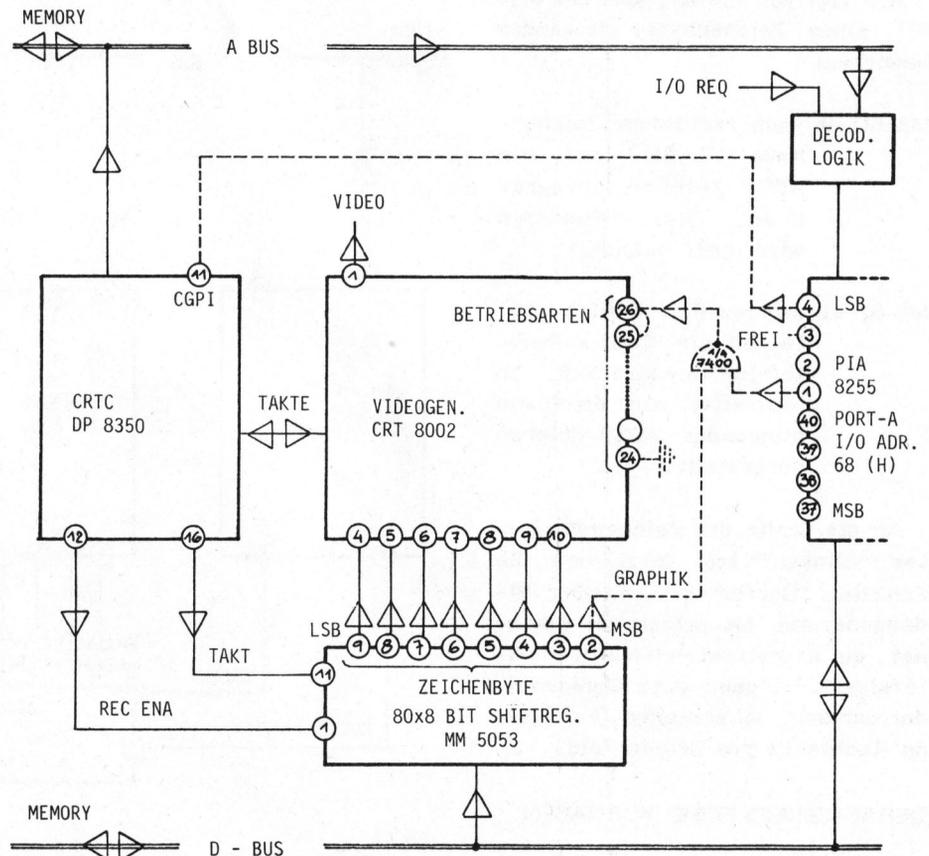
SUPERBRAIN funktioniert also nach einem Kaltstart wie gewohnt (ausgenommen "Blinken"). Will man zusätzlich "Grafik", so ist eine kleine Initialisierungsroutine zweckmässig, wie z.B. PROCEDURE GRAPH im angeführten PASCAL Demonstrationsprogramm.

Die gezeichneten Änderungen können ohne Eingriff in die Geräteplatine realisiert werden. Die notwendigen Verbindungspunkte bzw. Leitungsunterbrechungen lassen sich nämlich alle durch Zwischenschaltung entsprechend präparierter Sockel zwischen den angesprochenen Bauteilen und ihren Platinensockeln gewinnen. Für den Videogenerator sind zwei übereinandergesteckte Zwischensockel zweckmässig. Neben der verbesserten Zugänglichkeit gewinnt man im Innern der Sockel genügend Raum, um das notwendige NAND-Glied einzubauen.

Bei dem PIA und dem CRT-Controller können versuchsweise auch passende Drähtchen neben den zugehörigen Anschlüssen in die Sockelkontakte gesteckt werden. Dann ist allerdings die Verbindung zu Pin 11 des CRT-Controllers auf der Platine zu unterbrechen. Zur Versorgung des NAND-Gliedes stehen an Pin 18 des Videogenerator +5V und an Pin 17 0V zur Verfügung.

## SCHLUSSBEMERKUNGEN UND DEMONSTRATIONSPROGRAMM

Ueber eine schnelle Bildschirmgrafik zu verfügen, ist sicher der geheime Wunsch vieler Computer-Amateure. Ist nun die prinzipielle Möglichkeit geschaffen, wie hier beim SUPERBRAIN durch die aufgezeigte Änderung, so soll die verwendete Programmiersprache das neue Talent entsprechend unterstützen.



- — ○ KONKRETE VERBINDUNG
- SYMBOLISCHE VERBINDUNG
- - - - - ÄNDERUNG

**Bild 3: Schaltung des geänderten Superbrains**

Die Optimierung der Geschwindigkeit, um verblüffende Effekte zu erzeugen, ist dann aber immer noch eine grosse Herausforderung für den Programmierer.

Das angeführte Demonstrationsprogramm "UHR" stellt sicher keine Hochentwicklung in diesem Sinne dar. Auch soll es nicht eine voll-

kommene und genaue Analoguhr nachbilden. Es ermöglicht vielmehr einen ersten Einblick in eine "Grafik-Werkzeugkiste", wie sie z.B. durch PASCAL MT+ bereitgestellt werden kann. Als Besonderheit wird die Möglichkeit auffallen, den "Bildschirm" als absolute Variable zu vereinbaren (Speicherplatzzuteilung durch den Programmierer). Wei-

ters ist auf die Standardprocedure MOVE (SOURCE, DESTINATION, NUMBYTES) hinzuweisen. Es ist dies eine äusserst schnelle Transferroutine für Daten aller Art innerhalb eines Programmes. Sie wird hier eingesetzt, um ein möglichst schlagartiges Weiterrücken der "Uhrzeiger" zu bewirken.

Pascal/MT+ Release 5:2 Copyright (c) 1980 MT MicroSYSTEMS Page # 1  
 Compilation of: B:UHR

```

Stmt  Nest  Source Statement
  1      0  PROGRAM UHR;
  2      0  ä.....ü
  3      0  CONST  RAD=    0.01745;
  4      1          PI=    3.14159;
  5      1  ä.....ü
  6      1  TYPE    ZEIT=  RECORD  H:0..60;
  7      1                          M:0..59;
  8      1                          S:0..59;
  9      1          END;
 10      1
 11      1          FELD=  ARRAY  Å0..23,0..79Ü OF BYTE;
 12      1  ä.....ü
 13      1  VAR    BILDSCHM:  ABSOLUTE  Å$F800Ü FELD;
 14      1          STEMPEL:  FELD;
 15      1          ZEICHNUNG:  FELD;
 16      1          Z:          ZEIT;
 17      1          I:          0..159;
 18      1  ä.....ü
 19      1  PROCEDURE GRAPH;
 20      1  VAR    I:0..23;
 21      2  BEGIN
 22      2  WRITE (CHR(12));          ä^Lü
 23      2  OUT  Å$68Ü:=8;          ä8255 PORT Äü
 24      2  FOR I:=1 TO 23 DO WRITELN (I);  äAUFHEBUNG ZEILENDUNKELSTEUERUNGÜ
 25      2  WRITE ('24',CHR(1));          ä^Äü
 26      2  FILLCHAR(ZEICHNUNG,24*80,CHR(128));  äBIT 7=128 <-> GRAHIKZEICHENÜ
 27      2  END;
 28      1          äPARAM.STANDARDPROC.FILLCHAR
 29      1          (DESTINATION, LENGTH, CHAR)ü
 30      1  ä.....ü
 31      1  PROCEDURE SET_PKT ( X,Y:INTEGER);
 32      1  BEGIN
 33      2  SETBIT (ZEICHNUNG  ÅY DIV 3,X DIV 2Ü, Y MOD 3 +4 -4*(X MOD 2));
 34      2  END;
 35      1  ä.....ü
 36      1  PROCEDURE SET_STR (X,Y:INTEGER; STR:STRING);
 37      1  VAR I:1..80;
 38      2  BEGIN
 39      2  FOR I:=1 TO LENGTH (STR) DO
 40      2  ZEICHNUNG  ÅY,X-3+IÜ:= STRÄIÜ
 41      2  END;
 42      1  ä.....ü
 43      1  ä$Pü
  
```

# GEWUSST WIE!

```

Stmnt  Nest  Source Statement
44      1  PROCEDURE ZIFFERBLATT;
45      1  VAR      J,HX,HY:      INTEGER;
46      2  ZIFFERN:      STRING;
47      2  BEGIN
48      2  FOR J:=10 TO 149 DO
49      2  BEGIN  SET_PKT (J,1);
50      3  SET_PKT (J,70)
51      3  END;
52      2  FOR J:=1 TO 70 DO
53      2  BEGIN  SET_PKT (10,J);
54      3  SET_PKT (12,J);
55      3  SET_PKT (147,J);
56      3  SET_PKT (149,J)
57      3  END;
58      2  SET_PKT (80,37);          äMITTELPUNKTÜ
59      2  SET_STR (26,0,' S U P E R B R A I N   T i m i n g ');
60      2
61      2
62      2  ZIFFERN:= 'XII XI X IX VIII VII VI V IV III II I ';
63      2  FOR J:= 0 TO 11 DO
64      2  BEGIN  HX:= ROUND (20 *COS(30 *RAD *J +PI/2));
65      3  HY:= ROUND (20 *0.4 *SIN(30 *RAD *J +PI/2));
66      3  ä0.4 -- VERKUERZUNG IN Y-RICHTUNGÜ
67      3  SET_STR (HX+40,-HY+12, COPY (ZIFFERN,J *4 +1,4));
68      3  END;
69      2  END;
70      1  ä.....ü
71      1  PROCEDURE TICK (VAR Z:ZEIT);
72      1  BEGIN WITH Z DO
73      2  BEGIN IF S<59 THEN S:=S+1
74      3  ELSE BEGIN S:=0;
75      4  IF M<59 THEN M:=M+1
76      4  ELSE M:=0;
77      4  CASE M OF 11,23,35,47,59: H:=H+1 END;
78      4  IF H=60 THEN H:=0;
79      4  END
80      4  END
81      3  END;
82      1  ä.....ü
83      1  PROCEDURE ZEIGER (RADIUS, WINKEL: INTEGER);
84      1  VAR X,Y: INTEGER;
85      2  BEGIN
86      2  X:= ROUND (RADIUS *COS (6 *RAD *WINKEL -PI/2));
87      2  Y:= ROUND (RADIUS *0.65 *SIN (6 *RAD *WINKEL -PI/2));
88      2  ä0.65 -- VERKUERZUNG IN Y-RICHTUNGÜ
89      2  SET_PKT (X-1+80,Y+37);
90      2  SET_PKT (X+1+80,Y+37);
91      2  SET_PKT (X+80,Y-1+37);
92      2  SET_PKT (X+80,Y+1+37);
93      2  END;
94      1  ä.....ü
95      1  ä.....ü
96      1  BEGIN
97      1  GRAPH;
98      1  ZIFFERBLATT;
99      1  MOVE (ZEICHNUNG, STEMPEL, 1920);
100     1  REPEAT MOVE (STEMPEL, ZEICHNUNG, 1920);
101     2  TICK (Z);
102     2  ZEIGER (28, Z.H);
103     2  ZEIGER (46, Z.M);
104     2  ZEIGER (46, Z.S);
105     2  MOVE (ZEICHNUNG, BILDSCHM, 1920);
106     2  UNTIL FALSE
107     2  END.

```

B>

## EUROCOM II VERSION 7 (E II V7)

Der neue EUROCOM II V7 gestattet es, ein vollständiges Floppy-Disk-System, mit nur einer Doppel-Europa-Karte aufzubauen. Es sind alle Shugart-kompatiblen 5" und 8" Laufwerke direkt an die Prozessorplatine anschliessbar (5" über Adapter). Ferner beinhaltet die Karte einen Videocontroller mit hervorragenden Grafikeigenschaften, bis zu 64 k-Byte RAM, max. 8 kByte ROM, 2 PIA's und eine V24-Schnittstelle. Die I/O-Adressen sind durch programmierbare-Array-Logik (PAL) an Kundenwünsche anpassbar.

Der EUROCOM II V7 besitzt die wohl zur Zeit modernste 8-Bit-CPU MC 6809. Sie verfügt über zwei Akkumulatoren, zwei Indexregister und zwei Stackpointer. Vielfältige Adressierungsarten, hardwaremässige Multiplikation und interne 16-Bit-Operationen unterstützen die Programmierung. Zwei auf der Platine befindliche Parallel-Interface-Adapter (PIA 6821) bieten 40 frei programmierbare Ein/Ausgabe Leitungen. Die V24-Schnittstelle, aufgebaut mit dem ACIA-Baustein MC 6850, ermöglicht die Einstellung von Baudraten von 50 bis 19'200 Bit/s. An dem Floppy-Controller WD 1794 lassen sich max. vier 5" bzw. 8" Laufwerke anschliessen; Double Side und bei 5" auch Double Density sind möglich. Durch den Vollgrafik-Videocontroller, mit einer Auflösung von 512 H x 256 V Bildpunkten, sind Text und Grafik beliebig mischbar, ebenso sind Gestalt und Grösse eines jeden Zeichens per Software frei definierbar. Die Adressdekodierung, mit PAL's realisiert, ist jederzeit kundenspezifischen Anforderungen anpassbar. Alle CPU-Signale sind sowohl nach aussen sowie zur Prozessorplatine hin gebuffert. Die Gesamtleistungsaufnahme beträgt 11 Watt.

Eine Reihe von Erweiterungskarten ermöglicht die Anpassung an unterschiedliche Anforderungen. Der Speicher lässt sich mit 96 KByte umfassenden RAM-Karten bis auf 256 KByte ausbauen, der sowohl als Daten und Programmspeicher als auch zur Erweiterung des Bildwiederhol-speichers mit max. 64 Farben bzw. Graustufen verwendbar ist. An den Floppy-Controller (mit DMA-Controller und IEC-BUS-Interface) sind bis zu vier Floppy-Disk-Laufwerke (5" oder 8") Single Density oder Double-Density anschliessbar. Der EPROM-Programmer gestattet die Programmierung aller handelsüblichen EPROM's (auch 8 kByte Typen). Eine Parallel-I/O-Karte, aufgebaut mit

8 VIA's (6522), bietet 128 Ein-Ausgabe-Kanäle. Ausserdem steht eine universelle I/O Karte wahlweise mit DART oder SIO (2xV24) zur Verfügung. Ein Interface für das Philips-Mini-DCR eröffnet den Zugang zu einem kostengünstigen Massenspeicher auf Digitalkassette.

Zur Bildbearbeitung dienen folgende Zusatzkarten: AD-Wandler 30 MHz 6 Bit zur Bilddigitalisierung, Video-look-up-table mit RGB-Mischer zur Ansteuerung von Analog-RGB-Monitoren (max. 64 Farben), Zusatzkarte für die Fremdsynchronisation auf ein externes Video-Synchronraster, Doppelte Auflösung mit 512 H x 511 V Bildpunkten, Interlaced-mode und Videobild-DMA. Mit diesen Karten lässt sich z.B. ein Bildverarbeitungssystem, mit Uebernahme von Standard-TV-Bildern in den Bildwiederhol-speicher des EUROCOM II und anschliessender Bildbearbeitung, z.B. Falschfarbendarstellung realisieren.

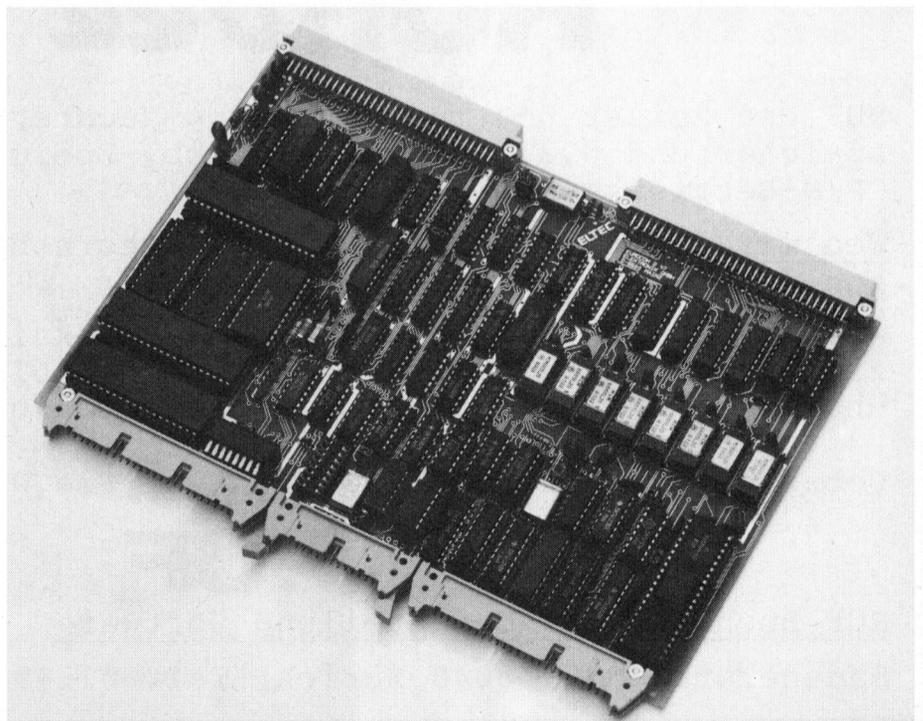
Es steht umfangreiche Software auf 5" und 8" Disk, Digitalkassette und EPROM zur Verfügung. Für Floppy-Disk-Systeme wird das FLEX Betriebssystem angeboten. Als Hochsprache PASCAL mit P-Code-Compiler oder Compiler wobei letzterer einen echten, sehr effizienten und prombaren Maschinen-Code generiert. FORTH, eine interaktive Sprache, ist besonders für Steuerungsaufgaben geeignet. Ferner befindet sich ein "C" Compiler, Extended Basic,

## NEWS...NEWS...

sind der Redaktion zugestellte Pressemitteilungen aus Industrie und Handel über neue oder wesentlich verbesserte Produkte. Diese Rubrik will und kann deshalb nicht die Meinung der Redaktion wiedergeben. Sie als Leser haben aber dadurch die Möglichkeit, sich zusätzlich zu den Fachartikeln in Mikro+ Kleincomputer ein umfassendes Bild zu machen, über die ungebremsame Entwicklung im Bereich der Mikroprozessoren, Kleincomputer, Peripherie und alles was so dazu gehört.

Editor (Bildschirmorientiert), Text Prozessor, Assembler, Disassembler, Cross-Assembler und -Disassembler für alle 68xx-CPU's sowie für 8080/8085/Z80 und 6502, Debug, Sort/Merge, Hardcopy (vom Bildschirm auf Drucker) und komfortable Terminalprogramme für Text und Grafik im Angebot.

SPECTRALAB  
Brunnenmoosstr. 7, 8802 Kilchberg  
Tel. 01 - 715 56 40





## PREISWERT PROGRAMMIEREN

Der PKW 3000 ist ein Programmiergerät das neue Massstäbe setzt. Ohne Zubehör bewältigt das Gerät 2716 bis 2764 inklusive den EEPROM's.

Der eingebaute 8085 uP ermöglicht eine grosse Anzahl von Befehlen (load, erase, program, compare etc.) und Kontrollfunktionen (sum check, self check or faults in RAM, Vcc marginal checking etc.). Die Buffer RAM-Kapazität von 8 kByte gibt darüber hinaus genügend Spielraum.

Die eingebaute serielle Schnittstelle (RS232-C) erlaubt eine direkte Kommunikation zwischen dem PKW 3000 und anderen Systemen wie CPU, Terminals und Entwicklungssystemen. Die Baudraten sind einstellbar zwischen 110 .... 4800 Baud wie die gebräuchlichen Formate der Maschinensprache (Intel-Hex, Binary, Motorola MIBUG und ASCII Hex). Mit den mech. Dimensionen von 280x190x65 mm und einem Gewicht von 1,8 kg, wird der PKW 3000 zu einem echten Portableprogrammer.

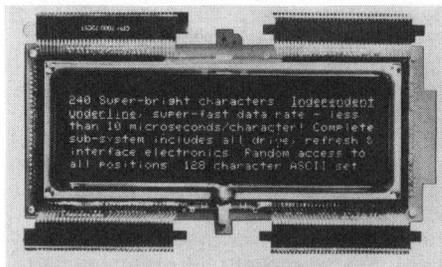
ALTRAC AG  
Zürcherstrasse 90, 8953 Dietikon  
Tel. 01 - 741 46 44

## BITRONIC STELLT EIN INTERESSANTES DISPLAY VOR

Die Firma Bitronic GmbH liefert ein neues Fluoreszenz-Display mit 6x40 Charaktern des Herstellers DECO. Neuheit bei diesem Produkt ist die Erzeugung aller Betriebsspannungen auf dem Modul selbst, so dass der Anwender nur noch eine 5 V-Spannungsversorgung benötigt. Der

Betrieb in Mikroprozessorsystemen wird dadurch erheblich vereinfacht.

Das Einschreiben der Zeichen erfolgt nicht über Cursorabfrage und Programmierung, sondern über einfache Adressierung mittels Adressleitungen, die dem Adressbus des Systems entsprechen. Somit wirkt das Display als RAM im Schreibbetrieb und der gesamte Anzeigeninhalt kann innerhalb von maximal 2,4 ms ausgetauscht bzw. geändert werden.



Technische Daten: Typ M6400, Versorgungsspannung +5 V DC, Stromaufnahme 3,6 A max., Zeichengrösse 5,05x3,5 mm, Zeilengrösse 188 mm.

BITRONIC GMBH  
Einsteinstrasse 127,  
D-8000 München 80

# W A N T E D

Software-Partner für

# RUF *bee*



RUF bee heisst unser neuer Z80 A-Rechner mit 64 KB Anwenderspeicher und 0,3 bis 1,2 MB Floppy-Speicherkapazität unter CP/M-Betriebssystem.

Wen wir suchen: Softwarehäuser, Unternehmensberater mit einschlägigen Lösungen.

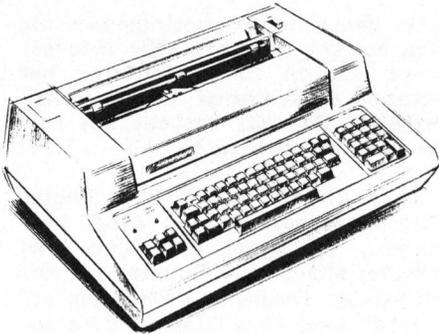
Was wir bieten: Sie haben die Möglichkeit, Ihre (kommerziellen) Programme, Branchenlösungen, Dienstprogramme usw. über den "RUF bee"-Software-Katalog anzubieten.

Ueber alles Nähere informieren wir Sie gerne.

## RUF

RUF Buchhaltung AG, Abteilung Software

Badenerstr. 595, 8048 Zürich, Tel. 01-491 64 00, Frau Hämmig



OMNI 840 MATRIXPRINTER

Mit dem neuen mikroprozessorgesteuerten Drucker - OMNI 840 KSR / RO - bietet Texas Instruments einen kostengünstigen Drucker mit modernem Design, umfangreicher Grundausstattung und hoher Zuverlässigkeit an.

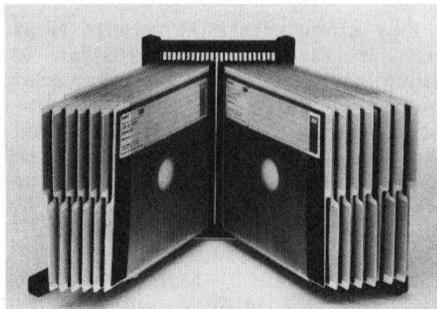
Der OMNI 840 verfügt über eine RS-232C/V24 Schnittstelle, 9x7 Punktmatrix, 132 Zeichen Schreibbreite, Übertragungsgeschwindigkeit von 110-9600 Baud, 2000 Zeichen Puffer, acht verschiedene Charactersätze wie z.B. Deutsch, Englisch, Französisch, sechs verschiedene Schriftarten, wahlweise einstellbar, vollen ASCII-Zeichensatz, 128 Zeichen.

Der OMNI 840 Drucker kann dank seiner Flexibilität in den verschiedensten Anwendungsbereichen eingesetzt werden wie z.B. im naturwissenschaftlichen, technischen und kaufmännischen Bereich. Ganz speziell geeignet ist der OMNI 840 zusammen mit dem Home Computer TI 99/4A.

FABRIMEX AG  
Kirchenweg 5, 8032 Zürich  
Tel. 01 - 47 06 70

## PRAKTISCHE ABLAGE- UND ORGANISATIONSSYSTEME FUER DISKETTEN

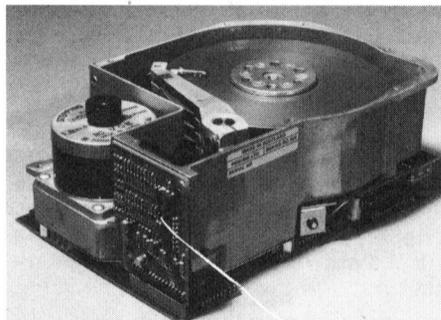
Die praktischen 3M Ablage- und Organisationssysteme für Disketten basieren auf einer Tasche, die zwei



Disketten aufnimmt. Die Tasche wird in zwei Grössen (für 8-Zoll- und Minidisketten) geliefert und schützt die Disketten vor Staub und Schmutz. Eine solide Trennwand verhindert eine Beschädigung der Disketten durch Verbiegen. Da die Taschen aus einem durchsichtigen, reflexfreien Material hergestellt sind, ist die ganze Diskette gut sichtbar und die Information auf den Etiketten jederzeit lesbar. Die in vier Farben mitgelieferten Etiketten erlauben eine übersichtliche Anordnung und einen schnellen Zugriff. Die Ablagesysteme werden kompakt und einzeln verpackt geliefert.

Die Abbildung zeigt das Disketten-Tischmodul, das 50 Mini- oder 8-Zoll-Disketten aufnimmt. Das Ablagesortiment umfasst im weiteren Archivboxen, Diskettenmappen, Diskettenkarussells und Data-Cartridge-Boxen.

3M (SCHWEIZ) AG  
Räffelstrasse 25, 8021 Zürich  
Tel. 01 - 35 50 50



## 5 1/4"-WINCHESTER DISK DRIVES MIT SPEICHERKAPAZITAET BIS ZU 27 MBYTES

RODIME Ltd. aus Schottland stellt als absolute Neuheit, vier verschiedene Modelle der Serie R0 200 vor. Die maximale Kapazität beträgt 26,67 MBytes unformatiert (typisch 21 MBytes formatiert).

Die Serie R0 200 basiert grösstenteils auf der früheren Serie R0 100 und besteht aus vier Grundmodellen, angefangen mit dem Modell R0 201 bestückt mit einer Festplatte bis zum Modell R0 204, mit vier Festplatten. Sie weist nicht nur die gleiche Abmessungen wie jene der Serie R0 100 und andere Drives auf, sondern es können auch die selben Speisungen (+ 5V/+ 12V DC) verwendet werden. Dadurch können diese Drives untereinander ausgetauscht werden.

Die höhere Kapazität gegenüber den Modellen der Serie R0 100 wird erreicht, indem 1) die Bitdichte von 8060 auf 8720 bpi, 2) die Anzahl Spuren/Oberfläche von 192 auf 220, 3) die Anzahl Spuren/Inch von 260 und 360 und schliesslich 4) die Flux-Wechsel/Inch von 8060 auf 8720 erhöht wurden.

Die hohen Zugriffsgeschwindigkeiten werden durch einen mikroprozessorgesteuerten Schrittmotor erreicht. Die durchschnittliche Zugriffszeit beträgt 90 ms. Sämtliche Festplattenspeicher von RODIME weisen die Besonderheit auf, dass sie in zwei Kammern unterteilt ist; dadurch werden die beweglichen Teile von den Plattenoberflächen getrennt. Eine gefilterte Luftzirkulation zwischen beiden Platten sichert eine gleichmässige Verteilung der Temperatur.

Wie ihr Vorgänger, benötigt die Serie 200 keine mechanische Justierungen und weist ebenfalls einen Mikroprozessor für die Fehlerdiagnostik auf. Zehn (10) verschiedene Fehlerquellen werden durch LED-Signalisierungslampen auf der Frontplatte angezeigt. Der MTBF beträgt über 12'000 Stunden.

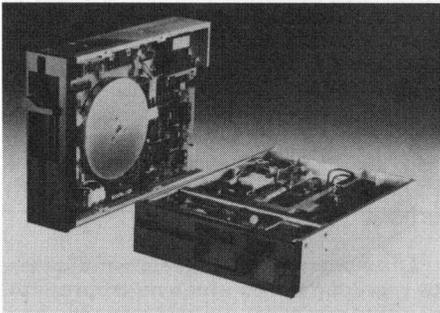
Prototypen sind bereits erhältlich und fabrikationsreife Laufwerke ab anfangs September 1982.

MULTICAP SA  
Av. General Guisan 58, 1800 Vevey  
Tel. 021 - 52 80 40

## MAGNETBANDSTATION FUER IBM-PERSONAL COMPUTER

INNOVATIVE DATA TECHNOLOGY (IDT), in der Schweiz vertreten durch STOLZ AG, hat neu ihre weitverbreitete und zuverlässige Magnetbandstation 1050 auch an den IBM-Personal Computer angeschlossen. Dadurch wird es möglich 1/2 Zoll Magnetbänder in ANSI, ECMA, IBM und ISO Standard zu verarbeiten. Die Daten werden auf 9 Spuren in 800 bpi (NRZI) oder 1600 bpi (PE) aufgezeichnet. Ein spezieller Hardware-Datenconverter ermöglicht dem Benutzer auch den Datenaustausch im Binär, ASCII oder EBCDIC Code. Das Bandgerät verarbeitet Standard-Magnetbänder mit Spulen bis zu 10 1/2 Zoll. Das Subsystem wird komplett mit Controller, Kabel, Dokumentation und Softwarelieferant geliefert.

W. STOLZ AG  
Täferstr. 15, 5405 Baden-Dättwil  
Tel. 056 - 84 01 51



NEUE MINIFLOPPY DRIVES MIT HALBER EINBAUHOEHE

Als erster Hersteller der Welt hat TEAC Corporation eine komplette Serie von SLIM-LINE 5 1/4"-Floppy-Laufwerken angekündigt.

Seine Höhe von nur 41 mm gestattet den Einbau von zwei solchen FD-55 anstelle eines einzigen herkömmlichen Laufwerkes und verdoppelt auf diese Weise die Speicherkapazität der Anlage.

Die neuen Laufwerke sind mit dem bekannt zuverlässigen bürstenlosen Motor, hier als Direktantrieb, ausgerüstet und besitzen die gleichen Anschlüsse und Funktionen wie die normalhohen Geräte - auch Head-Load!

Die technischen Daten: einseitig/doppelseitig, 48/96 tpi, single/double density, Kapazität 125... 1000 kB, Track-to-track Access Time 3 ms, MTBF 10'000 Stunden.

WENGER DATENTECHNIK  
Im Kägen 23/25, 4153 Reinach  
Tel. 061 - 76 87 87

## "C"-COMPILER FUER UNIFLEX-MEHRBENUETZER-SYSTEM

Für die Mehrplatz-Mikrocomputer-Systeme des amerikanischen Herstellers "Southwest Technical Products Corp" (SWT) ist nun eine volle Implementation der Programmiersprache "C" lieferbar.

In letzter Zeit gewann die von den "Bell Labs" in den USA entwickelte Computersprache zusehends an Bedeutung. Insbesondere wird sich "C" mit der Verbreitung des Unix-Betriebssystem (welches für alle neueren Mikroprozessoren erhältlich ist, bzw. sein wird) in einem breiten Feld der Applikationsprogrammierung durchsetzen. "C" ist durch das Buch "The C Programming Language" von Brian W. Kernighan und Dennis M. Ritchie (beide von Bell

Labs) "de-facto"-standardisiert und heute auf den meisten Computersystemen verfügbar.

Gegenüber herkömmlichen blockorientierten Sprachen wie PASCAL, PL/M, PL/Z, MPL usw. bietet "C" einige Vorteile. Einer der Hauptvorteile von "C" ist die Effizienz des generierten Codes, sowohl was die Speicherausnutzung anbetrifft wie auch in Bezug auf die Rechengeschwindigkeit. Auch als zeitkritisch eingestufte Programme wie z.B. das Betriebssystem Unix wurden erfolgreich in "C" realisiert. Das separate Uebersetzen von Modulen ist in "C" ebenfalls möglich. "C" besitzt die Vorteile von PASCAL, ohne aber dessen Einschränkungen zu unterliegen. Die Generalität von "C" macht diese neuere Sprache für ein sehr weites Anwendungsgebiet einsetzbar.

Die nun für SWT-Rechner verfügbare Version ist eine volle Implementation der Sprache. Der Compiler erzeugt relocative Module, welche mit dem Linker zu einem ausführbaren 6809-Programm zusammengefügt werden können. Das Erstellen von "shared"-Modulen, d.h. von Programmen, die auch bei mehreren Benutzern nur einmal im Hauptspeicher vorhanden sein müssen, ist möglich. Die generierten Programme sind reentrant.

Eine System-Bibliothek vervollständigt das Software-Paket. Alle Systemaufrufe an das UniFLEX-Betriebssystem (eine Unix-Variante für Mikroprozessoren) werden unterstützt.

DIGICOMP AG  
Birmensdorferstr. 94, 8003 Zürich  
Tel. 01 - 461 12 13

## DESIGN KIT FUER DIE EVALUATION VON SINGLE-CHIP MIKROPROZESSOREN

Die Preistendenz für Motorola Single-Chip Mikroprozessoren mit EPROM ist stark sinkend. Dies erlaubt, diese Prozessoren nicht nur als Prototypen für Single-Chips, sondern auch in Applikationen mit kleinen bis mittleren Stückzahlen einzusetzen.

Die 6805 Familie umfasst verschiedenste Versionen mit 20-32 I/O Leitungen, 64-112 Byte RAM und 1,8 - 3,8K EPROM. Ebenso sind Typen mit integriertem 4 Kanal A/D-Wandler oder PLL-Schaltung erhältlich. Alle Prozessoren sind voll softwarekompatibel.

Um dem Entwicklungsingenieur die Einarbeitung in diese sehr interessante Familie zu erleichtern, hat Motorola einen Design Kit zusammengestellt. Der Kit besteht aus folgenden Komponenten:

Je 1 IC: MC 68705P3L/S Single-Chip Mikroprozessor mit 20 I/O Leitungen, 112 Bytes RAM und 1,8K EPROM; MC 14040P Counter für den Aufbau der Programmierschaltung; MC 146818P Real Time Clock und Kalender; MC 14447P A/D Converter; MC 144110P Hex D/A Converter (6 Bit); MCM 144102P 16 x 16 Bit Backup RAM; MCM 2801 16 x 16 Bit EPROM; MC 14499P LED Display Driver; MC 144115 LCD Display Driver.

Datenblätter über alle Motorola Single-Chip Mikroprozessoren; Programmier-Faltkarten für alle Typen; Datenblätter von allen mitgelieferten IC's; Applikationsvorschläge mit Schematas; Softwarelistings für die Ansteuerung von allen mitgelieferten Bauteilen; als Zusatz kann auch ein Print für den Aufbau der Schaltungen geliefert werden.

Der Kit ist ebenso für die Evaluation wie für die Ausbildung auf Single-Chip Mikroprozessoren geeignet und zu einem äusserst attraktiven Preis erhältlich.

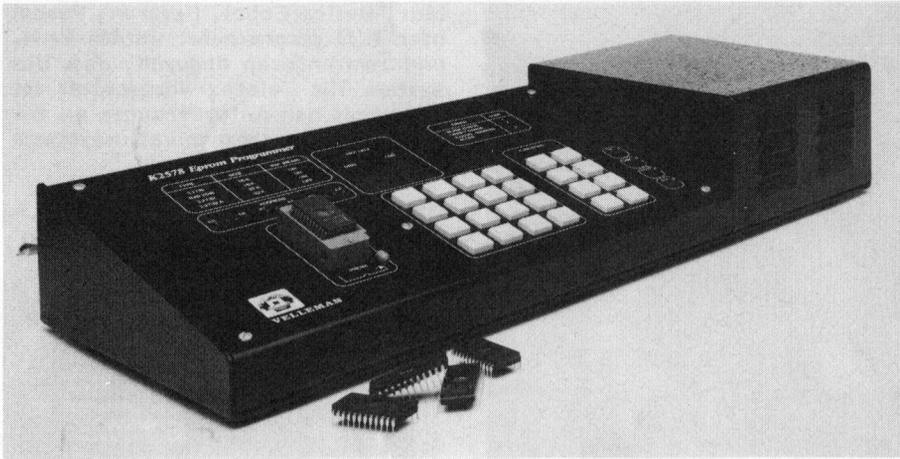
ELBATEX AG  
Hardstrasse 72, 5430 Wettingen  
Tel. 056 - 27 01 27

## GRAFIMATE, DAS KOMPAKTESTE GRAPHICSYSTEM

Grafmate ist ein Mikroprozessor gesteuerter Plotter für Zeichenformate bis A4 auf Papier und Overheadfolie. Im Gegensatz zu anderen Plottern benötigt aber Grafmate keinen Computer mehr. Sämtliche Funktionen lassen sich über das integrierte Flüssigkristall-Display und die Tastatur im Dialog programmieren. Für die hauptsächlichsten Darstellungen, wie Balkendiagramme, Kreissektor-Darstellungen, Hystogramme, sind Funktionstasten vorhanden.

Das eingetastete Programm lässt sich in bis zu 1000 Schritten in einem nichtflüchtigen Speicher oder in der als Option erhältlichen Microdisk-Einheit abspeichern, wobei die jeweils darzustellenden Daten separat gelöscht und eingegeben werden können.

COMPUTER GRAPHIX AG  
Giessereistrasse 1, 8620 Wetzikon  
Tel. 01 - 932 34 82



VELLEMAN K 2578 UNIVERSAL EPROM-PROGRAMMIERGERÄT

Seit einigen Jahren sind EPROM-Programmierer im Handel. Allerdings waren die Preise für solche sehr hoch und viele hatten Funktionen die nie gebraucht werden oder es konnten Typen programmiert werden die nur selten benötigt werden.

Neu ist ein Bausatz der Firma Velleman aus Belgien, der hier in eine Lücke springt. Alle üblichen 2 und 4 K EPROM's können programmiert und dupliziert werden. Das Gerät wird auch in Bausatzform geliefert, wodurch ein erstaunlich günstiges Programmiergerät unter 1000 Franken entsteht.

Das Gerät hat einen eigenen Speicher von 2K-Byte, welcher mit 4 IC's problemlos auf 4K erweitert werden kann. Die meisten Arbeiten im Gerät übernimmt ein Mikroprozessor wodurch ein einfacher Aufbau resultiert. Der Bausatz kann in ca. 3-4 Stunden problemlos zusammengebaut werden.

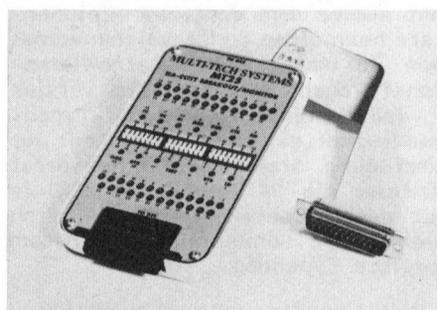
Technische Daten: Speisung 220 Volt; RAM 2K (4K); 12 LED; Datenanzeige 2 HEX-Anzeigen; HEX-Tastatur; Funktionstasten 8; Programmierbare EPROM's INTEL 2716 etc., TEXAS TMS 2516, INTEL 2732 etc., INTEL 2732A (HMOS).

Der Mikroprozessor wird auf einen Sockel gesteckt, welcher aus Endloskontakten besteht. Ansonsten gibt der Bausatz keine Probleme. Die beiden Prints und der Ringkerntrafo werden in das stabile, gut beschriftete Stahlblechgehäuse montiert, und fertig ist ein leistungsfähiger EPROM-Programmierer! Auf der Frontplatte befindet sich eine gut lesbare 2-stellige HEX-Anzeige der Daten und eine Reihe von 12 LED's zeigt die Adresse an. Mittels einer Tastatur können Daten

eingetragen oder modifiziert werden. Automatisch wird eine Verifikation vorgenommen. Nicht gelöschte IC's werden vor dem Schiessen als solche erkannt und zurückgewiesen. Ein TEXTOL-Stecker ermöglicht ein müheloses Wechseln der EPROM's.

Besonders hervorzuheben ist die gute Anleitung die mit vielen Bildern und Skizzen das Zusammenbauen erleichtert. Ebenfalls soll hier angeführt werden, dass sich der Betrieb als sehr zuverlässig erwiesen hat.

MUNDWILER ELECTRONIC  
Postfach, 8134 Adliswil  
Tel. 01 - 710 22 22



V24 INTERFACE-TESTER OHNE BATTERIE

Das Taschengrät MT25 erlaubt eine rasche Überprüfung von V24- und RS232-Schnittstellen.

24 Minischalter ermöglichen das Auftrennen der einzelnen Signalleitungen. Auf jeder Seite der Minischalter sind 25 Testbuchsen angeordnet, die zusammen mit kleinen Verbindungskabeln, beliebige Kreuzverbindungen und Kontrollen gestatten. 11 Leuchtdioden zeigen die massgebenden Zustände der Schnittstelle an. Zwei weitere Leuchtdio-

den können jeder Testbuchse für die Anzeige von positiven und negativen Signalen angeschlossen werden. Das Gerät ist 13,3x7,6x1,9 cm gross, wiegt 143 g und benötigt keine Batterie oder externe Speisung.

TELTEC HESS  
Knospweg 4, 3250 Lyss  
Tel. 032 - 84 42 40

## PERFEKTE FINANZBUCHHALTUNG AUF CP/M-MIKROCOMPUTER

Einfachheit für den Bediener und echte Dialogverarbeitung waren das Ziel bei der Programmentwicklung der DCT-Finanzbuchhaltung. Für den Anwender bedeutet dies - Buchhaltungskennnisse vorausgesetzt - das problemlose und komfortable Führen einer perfekten Finanzbuchhaltung, nach einer Einarbeitungszeit von nur zwei bis drei Stunden.

Die DCT-Finanzbuchhaltung ist auf dem Kontenrahmen von Dr. Karl Käfer aufgebaut. Gebucht wird auf dem Unterkonti (4stellige Kontonummern). Durch Eröffnung von Kumulierungskonti (2- oder 3stellige Kontonummern) kann eine Bilanz und/oder Erfolgsrechnungsstruktur erreicht werden. Durch das automatische Errechnen der Saldi und Umsätze sind Additionsfehler ausgeschlossen und die Bilanz und Erfolgsrechnung "stimmen" immer. Ohne Mehrarbeit können jederzeit Kontoauszüge, Kontoplan, Journal, Bilanz und Erfolgsrechnung ausgedruckt werden; durch die zeitliche Begrenzung lässt sich eine unnötige Papierflut verhindern. Bei jeder Buchung ist es ausserdem noch möglich, nebst dem Soll- und Habenkonto noch eine Kostenstelle zu erfassen; somit benötigt man für die Betriebsbuchhaltung keine separate Kostenerfassung mehr.

Die DCT-Finanzbuchhaltung verwendet als Programmiersprache BASIC und läuft auf allen CP/M-fähigen Mikrocomputern. Als Arbeitsspeicher werden 64 kBytes benötigt und für die externe Speicherung sollten mindestens 2x340 kBytes zur Verfügung stehen. Je nach Version können 350 bis 500 Konti geführt und 3000 bis 6000 Buchungen vorgenommen werden. Das Programm bietet folgende Unterprogramme und Funktionen: Initialisieren, Kontoeröffnung, Buchen, Kontostand, Buchungsjournal, Kontoauszug, Kontoplan, Erfolgsrechnung und Saldierung.

DIALOG COMPUTER TREUHAND AG  
Seeburgstr. 18, 6002 Luzern  
Tel. 041 - 31 53 53

## Beispiel: Laplace-Transformation

$$Y(p) = \int_{t=0}^{\infty} y(t) e^{-pt} dt = \int_{t=0}^{\infty} y(t) e^{-j\omega t - j\omega t} dt = L[y(t)]$$

$$Y(t) = \frac{1}{2\pi j} \int_{\sigma - j\infty}^{\sigma + j\infty} Y(p) e^{pt} dp = L^{-1}[Y(p)]$$

### SCITEX - MULTIFUNKTIONALES TEXTSYSTEM MIT KOMPLETTEM GRIECHISCH-MATHEMATISCHEN ZEICHENSATZ

SCITEX ist ein kombiniertes Text- und EDV-System, das basierend auf einem Standardtextsystem mit WordStar, speziell für die Erstellung mathematischer und naturwissenschaftlicher Texte gedacht ist. Neben dem deutschen Zeichensatz mit Umlauten und dem ISO-7-Bit-Code (ASCII-Code), über die Tastatur umschaltbar, besitzt SCITEX noch einen mathematisch-naturwissenschaftlichen Zeichensatz mit allen griechischen Buchstaben und mathematischen Zeichen, wie z.B. Summenzeichen und Integral. Wissenschaftliche Veröffentlichungen und Gutachten können somit inklusiv aller Formeln druckreif auf dem Bildschirm erstellt werden.

Besonders interessant ist SCITEX für wissenschaftliche Institute, da, ausser der Erstellung technisch-wissenschaftlicher Texte, mit dem gesondert lieferbaren kameralistischen Finanzbuchhaltungsprogramm INSTFIBU die Verwaltung der staatlichen bzw. privaten Geldmittel und Verteilung auf die verschiedenen Titel und Titelgruppen gesteuert werden kann.

In der Grundversion ist SCITEX ein Einzelplatzsystem, bestehend aus Bildschirm mit Tastatur, Drucker und einer Zentraleinheit mit zwei Diskettenlaufwerken für 5"- oder 8"-Disketten mit einfacher oder doppelter Schreibdichte, ein- oder beidseitig beschreibbar. Der Arbeitsspeicher - Zykluszeit 250 ns

- hat eine Minimalkapazität von 64 kB und kann je nach Bedarf in Stufen von 64 kB entsprechend erweitert werden. An jedes System kann darüber hinaus noch eine Reihe von Peripherie-Einheiten, wie Fest-/Wechselplattenlaufwerke zwischen 16 MB und 96 MB, 5"- oder 8"-Winchesterlaufwerke und weitere Drucker angeschlossen werden.

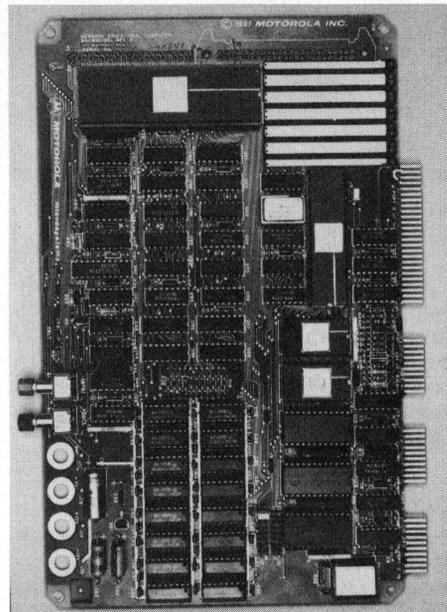
Geliefert wird die Grundversion in zwei verschiedenen Ausführungen. Version 1 wird mit dem Typenkorbdrucker NEC 5500D geliefert. Sie hat ausser dem deutschen Zeichensatz noch einen Sonderzeichenvorrat von 30 mathematisch-naturwissenschaftlichen Zeichen. Die Version 2 wird mit einem Doppelkopf-Typenraddrucker geliefert. Der damit zur Verfügung stehende Zeichenvorrat umfasst die 96 Zeichen des ISO-7-Bit mit Umlauten sowie zusätzlich weitere 96 Sonderzeichen auf dem zweiten Typenrad.

Durch das verwendete Betriebssystem, eine Multiuser-, Multitasking-, Multiprozessorversion, kann ein Einzelplatzsystem jederzeit zu einer Mehrplatzversion aufgerüstet werden, bei der jeder Bildschirmarbeitsplatz über einen eigenen Prozessor und 64 kByte Arbeitsspeicher verfügt. Durch die CP/M-Kompatibilität des Betriebssystems können sämtliche unter CP/M geschriebenen Programme ebenfalls auf diesem System eingesetzt werden.

Besonders unterstrichen wird die Flexibilität dieses multifunktionalen Informationssystems zum einen dadurch, dass SCITEX zur Lösung spezieller Anforderungen in Assem-

bler, Basic, Cobol, Fortran, Pascal oder PL/1 programmiert werden kann, und zum anderen dadurch, dass das System für Teletex vorbereitet ist und somit den Anforderungen an zukünftige Bürokommunikationssysteme Rechnung getragen wurde.

ARCUS DATENTECHNIK GMBH  
Löwenwall 6, D-3300 Braunschweig



MC68000 LERNCOMPUTER BIETET ASSEMBLER UND DISASSEMBLER

Der MEX68KECB Lerncomputer von MOTOROLA erlaubt es dem Anwender, ohne grosse Investitionen mit dem 16-Bit Mikroprozessor zu arbeiten. Der auf einer einzigen Platine aufgebaute Computer hat einen Arbeitsspeicher von 32 kBytes sowie 8 kBytes Firmware in ROM. Zwei serielle RS-232-Schnittstellen können mit Datenübertragungsraten von 110 bis 9600 Baud arbeiten.

Für den Betrieb des Computers wird an eine der seriellen Schnittstellen ein Terminal angeschlossen. Es bestehen zusätzliche Anschlussmöglichkeiten für ein Kassettengerät, einen Drucker und einen Host-Computer. Ein Wirewrap-Feld erlaubt die Erweiterung mit kundenspezifischer Hardware.

Die Firmware umfasst Betriebsprogramme für das Eingeben, Assemblieren, Disassemblieren und das Austeuen von Anwenderprogrammen. Darüberhinaus unterstützt sie die Kommunikation mit einem Host-Computer über die zweite serielle Schnittstelle.

ELBATEX AG  
Hardstrasse 72, 5430 Wettingen  
Tel. 056 - 27 01 27



## DAS PASCAL-HANDBUCH

Dieses Nachschlagewerk enthält alle PASCAL-Symbole, Bezeichner, reservierte Wörter und Operator für alle wichtigen PASCAL und PASCAL/MT+ Versionen. Sie werden in alphabetischer Reihenfolge präsentiert samt Definitionen, SYNTAX-Diagramm, Implementierung, Variationen und einem Programm-Beispiel.

Das Buch beinhaltet eine komplette Abhandlung des ISO Standards für PASCAL und behandelt alle Charakteristika und Variationen der folgenden PASCAL-Versionen: STANDARD, HP 100, J#W/CDC, OMSI, PASCAL/Z und UCSD.

Dieses Handbuch ist besonders wichtig für: a) PASCAL-Programmierer, die alle implementierungsabhängigen Besonderheiten im direkten Zugriff haben möchten; b) PASCAL-Programmierer, die ein Programm für eine andere Implementierung umschreiben möchten und c) all diejenigen, die nach einer Bibliothek von PASCAL Programmen suchen (beinhaltet als Implementierungsbeispiele der einzelnen Einträge).

SYBEX-VERLAG GMBH  
Heyestrasse 22  
D-4000 Düsseldorf 22

## IST DER TMS99000 NACHFOLGER DES TMS9900?

Die ersten zwei Mitglieder der dritten Generation der 16-Bit Mikroprozessoren, TMS99105 und TMS99110, sind angekündigt worden.

Die neuen Bausteine arbeiten mit 24 MHz Quarz und erreichen dadurch mehr als die doppelte Geschwindigkeit im Vergleich zu anderen auf dem Markt befindlichen 16-Bit Mikroprozessoren. Beide kommen mit ei-

nem umfangreichen Software Angebot auf den Markt wie z.B. Memory-mapped Real-Time-Executive (RX), das die Software-Entwicklung für TMS9900 Applikationen wesentlich vereinfacht und die Basis-Software der TMS99000 Familie bildet. Die Software, RX 3.0 oder RXMAP genannt, erlaubt auf einfache Art und Weise durch Rekompilierung von Programmen, welche ursprünglich für ältere Prozessoren gedacht waren, volle Kompatibilität. Die speicherintensive Architektur kündigt neue und charakteristische Eigenschaften dieser Familie an "Makrospeicher" an. Hier handelt es sich um einen superschnellen, vom Hauptspeicher unabhängigen adressierbaren und auf dem Chip integrierten Speicher.

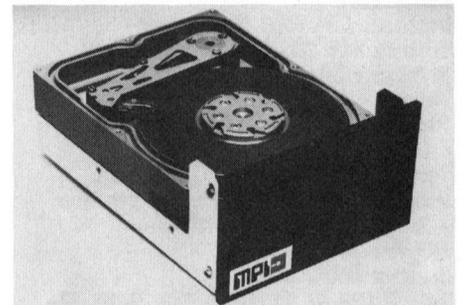
Der TMS99105 ist die Basis-Zentraleinheit und schlägt eine Brücke zwischen dem leistungsfähigen TMS9995 und dem TMS99000. Er verarbeitet 82 Befehle, die unter anderem die TMS9995 und TMS9900 Instruktionen beinhalten. Zusätzliche Befehle sind: "double precision" Arithmetik, "stack operation", parallele Ein-/Ausgabe und Speicher-Bit-Manipulationen.

Der noch leistungsfähigere TMS99110 ist weltweit der erste Fließkomma-Arithmetik-Prozessor. Seine Fließkomma-Arithmetik ist intern mit der IBM Standard Excess-64-Notation Methode auf dem Makro-Store realisiert worden. Er ist für hohe numerische Genauigkeit und computerintensive Applikationen bestimmt. Ausser den 82 Befehlen des TMS99105, enthält er 11 Fließkomma-Instruktionen wie Load-, Store-, Negate-, Add-, Subtract-, Multiply, Divide-real, convert integer to real und vice-versa, convert extended integer to real und vice-versa. Beide Prozessoren liefern eine 16 x 16 vorzeichenbehaftet Multiplikation mit einem 32-Bit Ergebnis in nur 3,6  $\mu$ s.

Den Einsatzschwerpunkt für den TMS99110 sieht man im industriellen und im Prozesssteuerungsbereich, wo hohe numerische Genauigkeit verlangt wird. Ebenfalls bestens ausgestattet ist er für die neue Generation von intelligenten Instrumenten. Die vorgestellten Mikroprozessoren bieten zudem umfangreiche Ein-/Ausgabe-Möglichkeiten. Sie verfügen über 16 priorisierte, vektorisierte Interrupts, DMA (direct-memory-access), einen Instruktionsprozessor/Computer Interface und serielle/parallele Ein-/Ausgabefunktionen. Die E/A Daten können bit-, byte- oder wortweise adressiert werden, das Context-Switching dauert ca. 2  $\mu$ s.

Sowohl der TMS99105 als auch der TMS99110, basierend auf einer Scaled N-Kanal Silicon-Gate MOS-Technologie, sind im 3-Mikron Design ausgeführt. Um den Chip-Bereich möglichst ökonomisch auszunutzen, wurde die bekannte SCAT-Methode (Strip-Chip-Architecture-Topology) verwendet. Die Chips sind alle TTL-kompatibel und ergänzen das schon sehr komplette TMS9900 Spektrum. Sie sind in einem 40-Pin DIP Plastik- oder Keramik-Gehäuse mit multiplexten Daten- und Adress-Leitungen untergebracht.

FABRIMEX AG  
Kirchenweg 5, 8032 Zürich  
Tel. 01 - 47 06 70



## 5 1/4"-WINCHESTER MIT SCHNELLEM DATENZUGRIFF

Mit den Abmessungen eines Mini-floppy-Laufwerkes, bringt MPI Micro Peripherals Inc. das Micro Winchester Laufwerk Modell 10 auf den Markt, welches eine Speicherkapazität von 10 Megabytes (formatiert) bietet.

Die hervorragende mittlere Zugriffszeit von 25ms, wird durch eine intelligente Kopfpositionierung mit gesteuerter Beschleunigung und Verzögerung erreicht. Der Schreib-/Lesekopf wird mit Hilfe von Mikroschritten über der gesuchten Spur genau positioniert. Das angewendete Verfahren ergibt hohe Positioniergenauigkeit und erübrigt eine mechanische Temperaturkompensation.

Das integrierte Filtersystem versorgt das geschlossene Diskgehäuse mit gereinigter Kühlluft; die ganze Einheit ist vollkommen wartungsfrei. Die Schnittstelle ist kompatibel zu den meisten angebotenen 5 1/4" Winchester Laufwerken. Als Speisespannung werden nur +5 Volt und +12 Volt benötigt.

ELBATEX AG  
Hardstrasse 72, 5430 Wettingen  
Tel. 056 - 27 01 27

**MIKRO  
+ KLEIN  
COMPUTER  
BÖRSE**

Zu verkaufen wegen Systemwechsel:  
Commodore 4032 mit Grossbildschirm  
inkl. Recorder. Neupreis  
Fr. 3175.— Verhandlungsbasis  
Fr. 2000.—. Geräte nur 4 Monate  
alt!  
Tel. 041 45 46 09 ab 19 Uhr

Zu verkaufen: Superbrain QD,  
neuwertig, max. 100 Std. gelaufen  
Fr. 7500.— inkl. CP/M und MBASIC.  
C. Märki, 8800 Thalwil-ZH  
Tel. B 01 221 26 22  
Tel. ab 19.00 Uhr P 01 720 29 47

Wer hilft?  
Problem: CBM 8032 Databecker-  
Graphik mit TCL-Pascal ansteuern?  
Erfolgspremie nach Vereinbarung.  
Heiner Audilet, Photogrammeter,  
Sirapuit 26, A-6460 Inst/Tirol

Verkaufe: uP-Lehrgang Christiani  
kompl. mit Drucker + Kass.Interface  
+ Kursunterlagen Fr. 1000.—  
(NP Fr. 2400.—).  
Stephan Noth, Biel  
Tel. P 032 42 41 81 G 032 21 32 37

Gelegenheit! TI59 + PC100A, nur  
Fr. 500.— \*\*\* TI-99/4A, 1 Monat  
alt mit RS-232 Interface und  
Kassettenrekorder für nur Fr.1600.—  
(Neupreis zusammen Fr. 1900.—)  
mit Garantie. Evtl. mit Drucker.  
Tel. 044 6 52 06

Verkaufe HP85, tadelloser Zustand,  
mit Koffer, 32K RAM, I/O-ROM,  
Standard-, Math.- und Games-PAL,  
div. Zubehör, aktueller Katalog-  
preis Fr. 9000.—, Verhandlungs-  
basis Fr. 4000.—  
Tel. 01 302 19 66

Dialog-Computer DG CS/10  
mit Hard-Disk 12.5 MB, Floppy 1.2  
MB, Bildschirm 24 Zeile x 80  
Zeichen, Programmiersprache: COBOL.  
Günstig abzugeben  
Tel. 01 737 33 20

Günstig zu verkaufen! ABC80 LUXOR  
Kleincomputer Grundausrüstung mit  
Kassetten-Speicher  
W. Baumgartner,  
Rank, 9221 Schweizerholz  
Tel. 072 42 24 13 ab 18.00 Uhr

Zu verkaufen: Commodore Computer  
4032 mit Toolkit; Centronics  
Metallpapierprinter + 5 Rollen  
Papier; Datenrekorder C2N; Div.  
Kass. alles zus. nur Fr. 2200.—  
J. P. Kousz  
Tel. 01 945 27 45

\*\*\*\*\*  
\* zu kaufen gesucht: \*  
\* 8-KANAL- \*  
\* LOCHSTREIFEN-LESER-STANZER \*  
\* Tel. 053 5 50 58 abends \*  
\*\*\*\*\*

SUPER-BIORHYTHMUS (für 1 oder  
mehrere Tage, besondere und  
kritische Tage für 1 oder mehrere  
Jahre) auf HP-41CV. Ausführliche  
Gratisinformation von DI Gerfried  
Tatzl, Mörikestrasse 17,  
A-8052 Graz/Oesterreich

Zu verkaufen Mikrocomputer ROCKWELL  
AIM 65 4K RAM + Basic-Expansion 4K  
PROM. Netzteil u. Gehäuse, Kabel  
für Rekorder. 1 Jahr alt. Diverse  
Bücher. Preis Fr. 1600.—  
R. Stoffel, Zürich  
Tel. 01 35 16 43 ab 18.30 Uhr

Verkaufe: CASIO FX-702P mit Drucker  
FP-10 und Kassetten-Interface FA-2,  
noch 8 Monate Garantie, en bloc  
Fr. 500.—. Verkaufe: SHARP 4KB-RAM-  
Modul zu PC-1500 für Fr. 80.—  
Beat Dörr  
Tel. 01 481 80 17

Zu verkaufen:  
Mikro-Computer ITT 2020 48K,  
1 Floppy Apple mit Controller,  
1 Printer Epson MX-80 F/T  
Tel. 042 31 28 73 ab 19.00 Uhr

Zu verkaufen:  
ITT 2020, 16K, fast neu (7.81)  
nur Fr. 2000.—  
Tel. 032 51 73 37 ab 18.00 Uhr

Zu verkaufen:  
Heathkit Floppy-Station H17  
(1 Drive) + MDOS 1.6 + H17 ROM  
Source-Listing. VB Fr. 1500.—  
Tel. 032 22 50 18 ab 20 Uhr

Verkaufe PET 3032 mit Toolkit und  
Kassette: Fr. 2000.—, dazu Printer  
CBM 3022: Fr. 950.—, HRG-Soft  
Tool-EPROM: Fr. 100.—. Alles en  
bloc Fr. 2900.—  
R. Baumgartner, Freie Strasse 36,  
4632 Trimbach, Tel. 062 21 17 53

Verkaufe: Apple II Europlus, 48 K,  
Floppy-Disk + Controller, 16 Sekto-  
ren, Pal-Color-Card, Applesoft  
Firmware-Card. Autostart-ROM, Pro-  
grammierhilfe A1 ROM, 2 Paddles,  
div. Manuals. Verhandlungspreis  
ca. Fr. 5400.—  
Tel. 036 22 32 47

Zu verkaufen: 2 Kleincomputer,  
32K RAM, 2x200K Floppy, ASCII,  
1000Z/Bildschirm, in einem Gehäuse,  
inkl. Centronics-Drucker 779 und  
1 Karteiprogramm je Fr. 5000.—  
Tel. 01 201 22 11  
Hr. Mülli verlangen

Verkaufe Superboard m. Gehäuse  
u. Kassettengerät Fr. 600.—  
Christiani-uP-Lehrgang inkl.  
Hardware (Drucker usw.) Fr. 450.—  
Tausche Apple I/O Karte gegen 80  
Zeichen-Karte  
Tel. 054 9 69 96 ab 18 Uhr

Zu verkaufen: Neuer Osborne 1 -  
Computer, original verpackt mit  
sämtlicher Software: CP/M 2.2,  
CP/M-Utilities, CBASIC, MBASIC,  
WordStar, Supercalc. Mit ausführ-  
lichem Manual. Fr. 4'200.—  
Tel. 041 31 48 10

Zu verkaufen: Apple-Zubehör,  
1 serielles Interface Fr. 350.—,  
1 Epson Parallel-Interface  
Fr. 150.— beide neuwertig  
1 Drucker Heath H14 wenig  
gebraucht Fr. 1200.—  
Tel. 081 51 29 58 ab 18 Uhr

## BASIC-KURSE

- Es sind keinerlei Vorkenntnisse nötig.
- Für jeden, der mit einem erfahrenen Lehrer und im Dialog mit dem Computer programmieren lernen möchte.
- Für je zwei Teilnehmer ist ein eigener Computer vorhanden.
- Kleine Gruppen mit individueller Beratung.
- Kursinhalt: Standard-BASIC (unabhängig von Computersystem).
- Kursdauer: 6 Abende (1x pro Woche) in Zürich.
- Beginn der nächsten Kurse: Ende August und Anfang Oktober.
- Kursgeld: Fr. 360.- inkl. sämtliches Kursmaterial.
- Fortsetzungskurse: Kaufm. Bereich/Statistik/Num. Mathematik/Lernalgorithmen (Themenkreis künstl. Intelligenz) u. a.

Verlangen Sie kostenlos Kursunterlagen

**Computer Training Zürich, Tel. 01 - 363 30 57**  
**Die Computerschulung mit der 12jährigen**  
**Computer- und Unterrichtserfahrung.**

## Aus Konkursmasse zu verkaufen

### 1 Computer-System, bestehend aus

- 1 Zentraleinheit CBM Commodore, Mod 8032
- 1 Speichereinheit CBM Commodore, Mod. 8050
- 1 elektr. Schreibmaschine HERMES top-tronic 20 mit Interface, mit Einzelblatt-Einführung

### 1 Computer-System, bestehend aus

- 1 Zentraleinheit CBM Commodore, Mod. 8032
- 1 Speichereinheit CBM Commodore, Mod. 8050
- 1 Typenrad-Printer DAISY M 50

Besichtigung nach Vereinbarung, Verkauf wie besichtigt ab Platz.

Auskunft erteilt das Konkursamt Zug, Tel. 042 - 25 34 25/26

Zu verkaufen: Sorcerer 32K mit Printer MP125, Rekorder für 1200 Bd, techn. Manuals und versch. Software. Neuwertiger Zustand. Für Fr. 3200.— (Neupreis Fr. 4500.—) Tel. 031 32 28 16

Zu verkaufen: Neuwertiger Apple 2 Plus 48K mit Language-Karte 16K, Disk 2 mit DOS3.3 Platte. Sanyo-Monitor 5912. Deutsches Handbuch. Wenig gebraucht. Preis Fr. 4650.— Tel. 01 984 08 47

Zu verkaufen: Div. VC-20 mit Zubehör sowie CEM 8032 von Programmier-Schule. Alle Geräte sind neuwertig. Tel. 01 930 60 58

Zu verkaufen: Sharp PC-1211 inkl. Printer/Kassetteninterface CE-122 Verkaufspreis: Fr. 535.— Tel. 056 44 13 83

\*\*\* WordStar Version 3.0 \*\*\*  
Einzigartige Steigerung der Effizienz durch Ausnutzung des "Memory mapped"-Betriebs auf DCT-Superbrain. Systems Analysis, Hintergasse 17, 8803 Rüslikon

Gelegenheit für Einsteiger! Wegen Systemwechsel verkaufe ich meinen PET 2001 (1/2 jährig) mit grosser Tastatur, externem Bandgerät, vielen Programmen und reichlicher Dokumentation für Fr. 1200.— Tel. 056 22 05 82 (ab 19.00 Uhr)

Zu verkaufen: ITT 2020, 48K, ca. 1 Jahr alt, für Fr. 1950.— (inkl. diverse Programme und Manuals) Tel. 042 21 06 87 (MI oder abends)

Günstig zu verkaufen:  
VC20 mit EW-16K. Neuwertig.  
Tel. 055 38 11 54 ab 18.00 Uhr

Drucker CEM 8026, Abholpreis Fr. 1940.—  
Volker Hagen, 3510 Häutlingen  
Tel. 031 99 18 23

\*\*\*\*\* Suche \*\*\*\*\*  
\* Tischrechner HP-9830 \*  
\* Thermodrucker HP-9866 \*  
\* Mass Memory HP-9880 (ev.) \*  
\* günstige Angebote an: \*  
\* Tel. 071 31 20 48 (abends) \*  
\*\*\*\*\*

Zu verkaufen: CEM 3032 mit Floppy 4040. Intel Software 6502 Assembler, TCL Pascal, Schach, Toolkit, Newtim Monitor. Komplet für nur Fr. 4600.—  
T. Koeflerli, Feldi, 8479 Altikon  
Tel. 052 38 14 61 (18 - 21.00 Uhr)

\*\*\*\* Verkäufe Printer \*\*\*\*  
AXIOM IMP-2; 110 cps; 80/96/132K, 96 ASCII; 50-1200 bd; Par/serial Interface; Input buffer; tractor + friction feed; 7x7 dot matrix; Graphics 7xN bit; nur Fr. 1100.—  
A. Stucki, Dübendorf  
Tel. 01 821 14 55

Verkäufe wegen Systemwechsel:  
CEM/PET 32KB inkl. Toolkit und Kassettengerät. Fr. 2000.—  
Tel. 061 99 52 77

Verkäufe wegen Systemwechsel:  
Superbrain A, Printer Heath H14, - Floppyabschaltung DCT, - Europa-tastatur mit ä, ö, ü, wenige Betriebsstunden. Statt Fr. 12'025.— nur Fr. 7'900.—  
Tel. 037 43 15 33

Printer CENTRONICS 779  
3 Monate alt, wenig gebraucht, Fr. 950.—: R. Höltschi, Hinterdorf, 5735 Pfeffikon über Mittag Tel. 064 71 29 64  
Geschäft Tel. 064 71 44 19

Osborne-1 + BASCOM, Pascal/M, Fortran + Garantie Fr. 3900.—, CEM 3032 Fr. 1700.—, Comprint-912 Printer mit IEEE zu PET, Metallpapier Fr. 350.—  
Heiner Weber, CH-4500 Solothurn  
Tel. 065 22 90 39

Anfänger sucht Software, Hardware und Ratschläge für Bedienung, Ausbau usw. für TI 99/4A  
Miller Hans, Krummackerstr. 3, 8953 Dietikon

Verkäufe: CEM 8032 mit Tool-Kit und div. Programmen evtl. mit Single-Floppy oder Dual-Floppy 8050. Neuwertig. Tel. 01 930 60 58

# OUTPUT

Die schweizerische EDV-Zeitschrift für den Manager und den Fachmann

Output vermittelt Wissen und Erfahrung und erleichtert somit Führungsentscheide. Natürlich brauchen Sie Output nicht zu abonnieren, wenn Sie es sich leisten können, von neuestem EDV-Wissen nichts zu wissen. Aber können Sie es sich wirklich leisten? Abonnieren Sie Output.



Output ist nicht am Kiosk erhältlich –  
aber bei Frl. Annemarie Jaeckli  
unter Tel. (071) 41 66 11

## OUTPUT

Eine Publikation der  
FACHPRESSE GOLDACH  
Hudson & Co., 9403 Goldach

...endlich Fertigergerät mit EUROCOM-II/V7!  
sFr. 6950.— exkl. WUST, exkl. Monitor



- Im Preis enthalten:
- programmierbare Tastatur mit Cursor-, Nummernblock und Funktionstasten
  - zwei Mini-Floppies, je 330 kB formatiert
  - EUROCOM-II/V7-Computer mit Feingrafik und 60-kB-RAM, Anschluss von gewöhnlichen Videomonitoren möglich, V 24-Interface, Floppy-Controller, Druckeranschluss
  - getaktetes Netzgerät 100 Watt
  - zwei freie Steckplätze für Doppeleuropakarten (Zusatz-RAM, I/O-Karten, EPROM-Programmer)
  - Betriebssystem, Bildschirmeditor und Textprozessor, mit deutschen Handbüchern
  - vollständige Hardwaredokumentation mit Schema
  - geeignet für OMEGASOFT-Pascal-Compiler mit Grafikusatz; für allgemeinen und industriellen Einsatz (EPROM-Erstellung)

SPECTRA  
LAB

Brunnenmoosstrasse 7, CH-8802 Kilchberg, Tel. 01/715 56 40

# Vorschau

Im nächsten Heft stellen wir Ihnen zwei neue interessante Computersysteme vor. Beides sind Kleincomputer sowohl für den Small Business-Einsatz als auch für den Privatgebrauch. Trotzdem unterscheiden sie sich sehr stark voneinander: das eine ist eine 16 Bit Maschine, das andere ein 8 Bit'ler. Jede mit einem eigenen Charakter. Mehr wollen wir Ihnen hier jedoch noch nicht verraten.

Kleincomputer werden, obwohl immer leistungsfähiger, immer erschwinglicher. Der gleiche Trend lässt sich auch bei den Peripheriegeräten beobachten, doch leider, für den Käufer, nicht in demselben Ausmasse. Floppystationen, Schön-schreibdrucker und Plotter sind auch heute noch teuer. Betreibt jemand mehrere Kleincomputer, so lohnt sich die Ueberlegung wie die

teuren Peripheriegeräte von allen Benutzern genutzt werden können. Im nächsten Heft befassen wir uns mit den Grundlagen lokaler Netzwerke und zeigen, dass nicht nur Systeme wie Ethernet, SWAN und wie sie alle heissen mögen, die einzige Lösung bieten.

Ausserdem möchten wir Ihnen die Kriterien, die bei der Auswahl eines Computersystems berücksichtigt werden müssen, erläutern. Lesen Sie auch, was ein OCR-Leser ist und wie man ihn einsetzt.

Für die Anfänger unter uns sowie zur allgemeinen Wiederauffrischung, haben wir eine kleine Einführung in die Bedienung des für Mikrocomputer nach wie vor am meisten eingesetzten CP/M-Betriebssystems zusammengestellt.

Zugegeben, die Rubrik PPC/HHC war auch schon abwechslungsreicher

- und sie wird es in der folgenden Ausgabe auch wieder sein: Wir wollen u.a. den neuen TI-88 auf Herz und Nieren prüfen und ausführlich darüber berichten. Daneben ist ein Beitrag in Vorbereitung, der sehr eindrücklich die Grafikmöglichkeiten des Sharp PC-1500 demonstriert.

Die Fortsetzung der im letzten Heft begonnenen Anleitung zum Aufbau eines "selbstgestrickten" Textverarbeitungssystems für den Apple mussten wir aus technischen Gründen auf die nächste Ausgabe verschieben.

Mikro+Kleincomputer Nr. 82-5 erscheint am 8. Oktober 1982. Haben Sie Ihr Abonnement schon bestellt? Für nur Fr. 36.- (Ausland Fr. 44.-) erhalten Sie ein Jahr lang aktuelle Kleincomputer-Informationen.



Die DCT ist offizieller Software-Distributor der Data Center Luzern AG

## Ihr Mikro leistet mehr mit DCL- Software

für CP/M, MP/M, CP/M-86, MS-DOS (PC-DOS) ... ausgereift und Hardware-unabhängig!

- Finanzbuchhaltung
- Debitoren/Kreditoren
- Lohn und Gehalt
- Lagerbewirtschaftung
- Auftragsbearbeitung
- Fakturierung
- Adressverwaltung
- Statistiken
- usw.



programmierte Qualität seit 1963

*...sprechen Sie mit uns,  
bevor Sie irgendeine Software kaufen*

DIALOG COMPUTER  
TREUHAND AG  
Seeburgstrasse 18  
6002 Luzern

Telefon 041 - 31 53 33

Wenn's um Kleincomputer geht...



Das einzige schweizerische Kleincomputer-Fachmagazin bringt alle zwei Monate aktuelle und kompetente Informationen, Testberichte und Problemlösungen.

Profitieren Sie von diesem einmaligen Erfahrungsschatz, wenn Sie mehr wissen und verstehen wollen, was Mikroprozessoren sind, wie Kleincomputer funktionieren und was man alles mit ihnen machen kann.

...wenn's um Kleincomputer geht!



Bitte senden Sie mir ab der nächstfolgenden Ausgabe regelmässig und bis zur Abbestellung



das einzige schweizerische Fachmagazin für Kleincomputer, Mikrocomputer für kommerzielle Anwendungen und programmierbare Taschenrechner zum Jahresbezugspreis von  SFr. 36.- im Inland für 6 Hefte pro Jahr. Im Ausland (nur Europa):  SFr. 44.-  DM 49.-  öS 375



die Commodore-Anwenderzeitschrift zum Jahresbezugspreis von  SFr. 48.-  DM 55.-  öS 400 für 6 Hefte pro Jahr. **Abonnenten von «Mikro- und Kleincomputer» bezahlen für das CBM/PET NEWS-Abonnement nur  SFr. 24.-  DM 29.-  öS 200.** Die CBM/PET NEWS sind nur im Abonnement pro Kalenderjahr erhältlich. Bereits erschienene Ausgaben des laufenden Jahres werden automatisch nachgeliefert.

# Bestellkarte

für ein Jahresabonnement 82-4

In den genannten Abonnementpreisen sind sämtliche Nebenkosten, inkl. Porto, enthalten. Die Kündigung ist jeweils 8 Wochen vor Ablauf des laufenden Bezuges möglich. Die Abonnementgebühr ist nach Erhalt der Rechnung fällig.

Der angekreuzte Betrag wurde bereits auf Ihr Postkonto  
 Luzern 60-27181  Stuttgart 3786-709 (BLZ 60010070)  
 Wien PSK 7975.035 einbezahlt.  Eurocheck liegt bei

\_\_\_\_\_  
Name/Vorname

\_\_\_\_\_  
Beruf

\_\_\_\_\_  
Strasse

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

\_\_\_\_\_  
Datum

\_\_\_\_\_  
Unterschrift

## Info-Karte

### Für alle Leser, die Mikro+Kleincomputer nicht im Abonnement erhalten

Bitte notieren Sie meine Adresse, damit Sie mir in regelmässigen Abständen völlig kostenlos weiteres Informationsmaterial über den Kleincomputerbereich zustellen können.

- CH
- D
- A
- 

\_\_\_\_\_  
Name/Vorname

\_\_\_\_\_  
Beruf

\_\_\_\_\_  
Strasse

\_\_\_\_\_  
PLZ

\_\_\_\_\_  
Ort

\_\_\_\_\_  
Datum

\_\_\_\_\_  
Unterschrift

82-4

## AUFTRAGSKARTE FÜR EIN KLEININSERAT IN DER BÖRSE



Bitte veröffentlichen Sie in der nächsterreichbaren Ausgabe den nachstehenden Inseratetext: 82-4

\_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

(Bitte jeweils 30 Buchstaben pro Zeile – einschliesslich Satzzeichen und Wortzwischenräumen)

Kleininserate in der Börse werden **nur gegen Vorauszahlung** aufgenommen. Der Betrag von  Fr. 20.- ( Fr. 40.- für Nichtabonnenten) für ein **privates** Kleininserat  Fr. 100.- für ein **kommerzielles** Kleininserat liegen bei  wurde auf Ihr Postkonto einbezahlt  Luzern **PC 60-27181**  Stuttgart **3786-709**  Wien **PSK 7975.035**  Eurocheck liegt bei **Bitte vergessen Sie nicht, umseitig Ihre Adresse einzutragen! Vielen Dank!**

bitte  
frankieren

Mikro+Kleincomputer  
Informa Verlag AG  
Postfach 1401

CH-6000 Luzern 15

**Auch Sie  
finden  
Ihr Ziel-  
publikum  
dort, wo  
Interessierte  
seit 1979  
sich regel-  
mässig  
informieren!**

bitte  
frankieren



Mikro+Kleincomputer  
Informa Verlag AG  
Postfach 1401

CH-6000 Luzern 15

**hat  
entschluss-  
freudige  
und kauf-  
kräftige  
Leser . . .  
das sind  
Ihre Kunden  
von heute  
und morgen.**

bitte  
frankieren

Mikro+Kleincomputer  
Informa Verlag AG  
Postfach 1401

CH-6000 Luzern 15

**Sprechen  
Sie mit uns,  
wenn's um  
Klein-  
computer  
geht.  
Gerne  
senden wir  
Ihnen die  
Media-  
Unterlagen.**

# ALL THE PROGRAMS YOU'LL EVER NEED.

## FOR Frs. 1350.-

Oubliez les programmes coûteux et fastidieux: maintenant The Last One® est partout disponible.

Unique par sa conception et sa simplicité, The Last One® est un programme, générateur de programmes. Des programmes qui fonctionnent chaque fois au premier essai.

The Last One® dialogue avec vous non pas dans un quelconque jargon technique, mais en bon français. Il s'informe de vos besoins et élabore, sur la base de vos réponses, un programme *Basic* garanti sans erreurs et prêt à l'usage.

De plus, The Last One® vous permet de modifier ou d'adapter votre programme aussi souvent que vous le désirez, avec une facilité déconcertante. Vos besoins changent, les programmes suivent.

Vous avez hésité à acheter un ordinateur parce que les programmes vous semblaient trop onéreux, trop compliqués, voire impossibles à adapter aux conditions spécifiques de votre entreprise? N'attendez plus!

Vous trouverez The Last One® chez les meilleurs spécialistes informatiques. Pour le commander, présentez cette annonce à votre fournisseur et demandez-lui de plus amples renseignements. Vous pouvez également nous écrire directement.

*The Last One®: le seul programme dont vous avez besoin.*

Manuel d'utilisation sFr. 30.-  
déductible en cas d'achat.

Plébiscité par la presse informatique.

Vorbei die Kosten und Frustrationen, die mit der Herstellung von Software verbunden waren: nun gibt es The Last One®!

The Last One® ist ein Programm, das Computerprogramme schreibt. Es übertrifft in seiner Verständlichkeit und seinem Konzept alles Herkömmliche. Die Programme klappen jedes Mal auf Anhieb.

The Last One® befragt Sie in perfektem Deutsch nach Ihren Bedürfnissen und bedient sich der Antworten, um ein vollkommen fehlerfreies, sofort gebrauchsfertiges *Basic*-Programm zu erstellen.

Noch besser! Mit The Last One® können Sie Ihre Programme abändern und umschreiben, so oft Sie wollen, ohne Mühe, Aufregung oder zusätzliche Kosten. Je nach Ihren Bedürfnissen verändern Sie auch ganz leicht Ihre Programme.

Wenn Sie bis jetzt vom Kauf eines Computers abgehalten wurden, weil Ihnen die Programmerstellung zu kostspielig, kompliziert, und die Anpassung an Ihre Bedürfnisse zu schwierig erschienen, brauchen Sie nun nicht länger zu zögern.

The Last One® ist bei Ihrem Computerspezialisten erhältlich. Wenn Sie es bestellen wollen, gehen Sie mit dieser Anzeige zu Ihrem Lieferanten und erkundigen Sie sich nach weiteren Einzelheiten. Sie können auch gerne direkt an unsere Adresse schreiben.

*The Last One®: Sie können auf alle anderen Programme verzichten.*

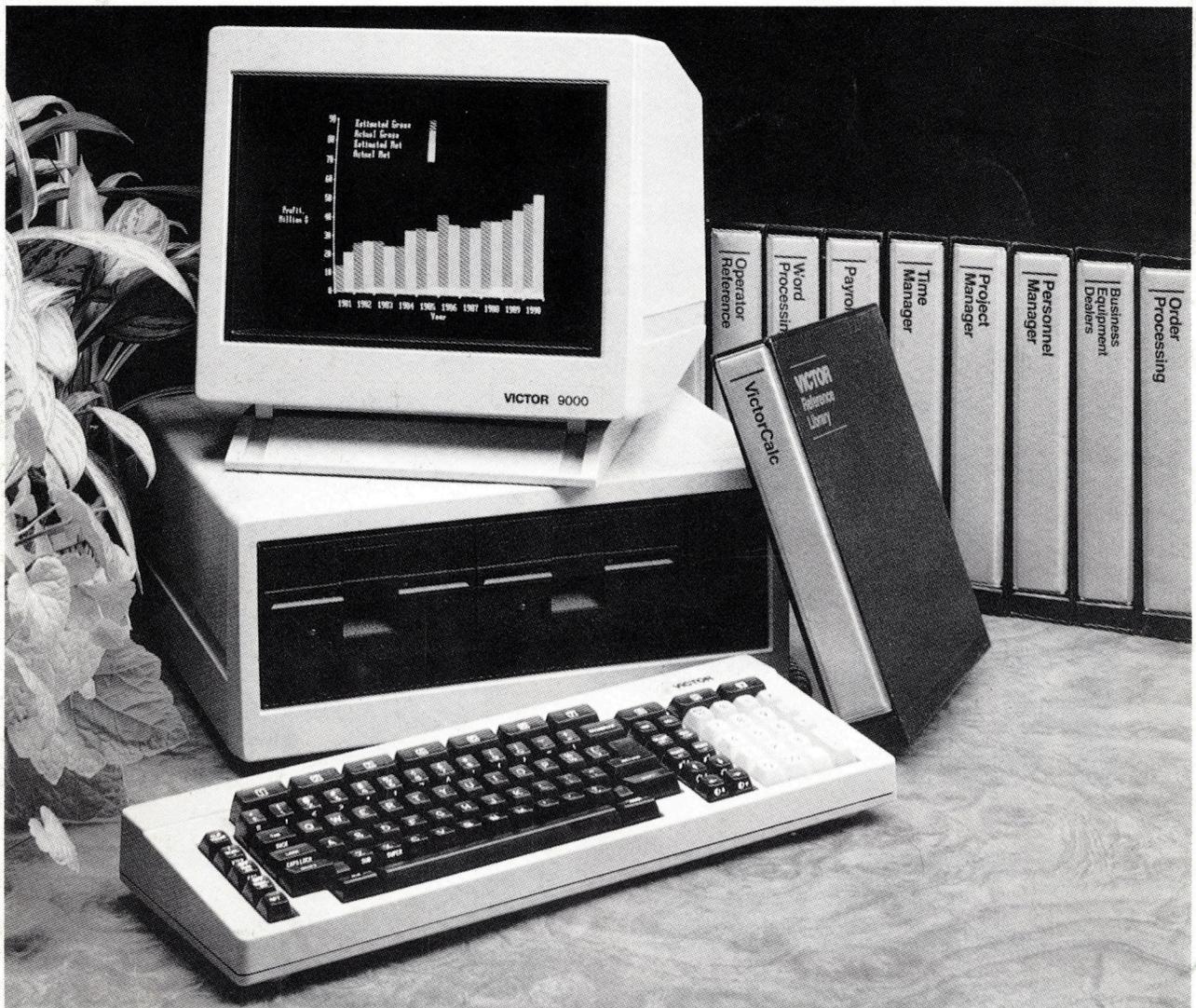
Anwender-Manual sFr. 30.-  
wird beim späteren Kauf angerechnet.

Durch die Fachpresse angepriesen.

# THE LAST ONE®

**SIVICO SA**  
Société informatique de vente et conseils  
19a, bd de Grancy CH-1006 Lausanne  
Tél. 021/27 44 32 et 26 35 18  
Télex 26 371 INCM CH

# VICTOR 9000



## Der neueste Klein-Computer

der 3. Generation mit dem besten Preis-Leistungs-Verhältnis.

16-bit CPU 8088. Wie der IBM-PC. CP/M und MS/DOS (IBM-PC-DOS) Betriebssysteme. 128 k RAM, ausbaubar bis 1 MByte. 2 Floppy total 1.2 MByte oder 2.4 MByte. Deutsche Tastatur / getrennt. Tastatur und Zeichensatz voll programmierbar. Vollgrafik 800 x 400 Punkte. Bildschirm: 80 x 25 Zeichen und 132 x 50. Digitalisierte Sprache. Einbau-Lautsprecher.

Ausbau mit Harddisk und Netzwerk. 100 % kompatibel mit den SIRIUS-Computern.

**KOMPLETT: Fr. 11 950.-**

inkl. WUST und Verpackung.

Ihr VICTOR-Händler bietet Ihnen seine eigene Arbeit und zudem das ganze Potential der VICTOR-Generalvertretung.

Garantierter Top-Service. 23 Jahre Computer-Erfahrung. Die riesige CP/M Programmbibliothek.

Exklusiv für VICTOR: Programme für höchste Anforderungen:

VICTORCALC: Wie »VISICALC«, aber 3-dimensional. VICTORWRITER: Die ganz einfache Textverarbeitung. TOMCAT: Die neue Computer-Buchhaltung von Hannes Keller mit totaler Transaktions-Kontrolle, FIBU/DEBI/KREDI, Fremdwährungen. KELLER-DATABASE: Beliebig verknüpfte Datei-Systeme.

**HANNES KELLER COMPUTER-ZENTRUM AG ZÜRICH**  
Eidmattstrasse 36, 8032 Zürich. Tel. (01) 69 36 33 Telex 58766 und 53808

**HANNES KELLER COMPUTER-ZENTRUM AG BERN**  
Quartiergasse 16, 3013 Bern. Tel. (031) 41 22 45

**HANNES KELLER HI-TECH INC**  
1050 Edwards Road, Burlingame, California USA