



**magazin für
computer
technik**

4/84

DM 6,—
öS 52,—
sfr 6,—
hfl 6,80



Was heißt CP/M-kompatibel?

BASIC-Benchmarks

Casio FP-200

MAX 1

Für jeden Computer:

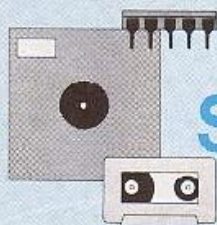
**Speichern, Lesen, Kommunizieren
mit 3600/7200 Baud**

Centronics-Schnittstellen für ZX 81, VC-20, C 64

4

April 1984

Heise



Software-Service

Ein Extra-Service des Heise-Software-Service

VICTAPE COMPUTING für VC-20



SPECTRUM COMPUTING für ZX-Spectrum



MODEL B COMPUTING für BBC-Acom, Modell B



Spezial-Magazine auf Compact-Kassetten

COMPUTING demonstriert, was Ihr Homecomputer kann:

- ★ Systemprogramme
 - ★ Spiele
 - ★ Bewegliche Grafik
 - ★ Internationale News
- auf dem Farbbildschirm

COMPUTING-Magazine — exklusiv für c't-Leser
in der internationalen Originalausgabe (in englischer Sprache)

Alle zwei Monate

Preis: 19,80 DM (zuzüglich 3 DM Versandkosten)

Für Sinclair-Fans:



Nummer 4



Nummer 5

ZX-COMPUTING

das große Spezialmagazin für ZX81 und Spectrum
(Originalausgabe in englischer Sprache)

Programme und Informationen
über Ihren Computer

Preis: 9,80 DM (zuzüglich 1,70 DM Versandkosten)

Bestellen beim

Heise-Software-Service
Postfach 27 46 · 3000 Hannover 1

Lieferung nur gegen Vorkasse. Fügen Sie Ihrer Bestellung einen Verrechnungsscheck oder den quittierten Einzahlungsbeleg Ihrer Bank bei.
Überweisungen bitte auf das Konto-Nr. 93 05-308 Postscheckamt Hannover

c't-Abonnement

Abrufkarte

GARANTIE

Wir garantieren jedem Abonnenten das Recht, seine Bestellung innerhalb einer Woche nach Abschluß schriftlich zu widerrufen.

c't-Kontaktkarte

Mit dieser Service Karte können Sie

- **Informationen** zu in c't besprochenen oder angebotenen Produkten direkt bei den genannten Firmen **abrufen**;
- **Bestellungen** bei den inserierten oder redaktionell erwähnten Anbietern **vornehmen**;
- **Platinen, Folien, Bücher, Software, bereits erschienene Hefte** beim Verlag Heinz Heise GmbH, c't-Versand, Postfach 2746, 3000 Hannover 1, **ordern**.

c't-Kontaktkarte

Mit dieser Service Karte können Sie

- **Informationen** zu in c't besprochenen oder angebotenen Produkten direkt bei den genannten Firmen **abrufen**;
- **Bestellungen** bei den inserierten oder redaktionell erwähnten Anbietern **vornehmen**;
- **Platinen, Folien, Bücher, Software, bereits erschienene Hefte** beim Verlag Heinz Heise GmbH, c't-Versand, Postfach 2746, 3000 Hannover 1, **ordern**.

c't-Abonnement

Abrufkarte

Ja, übersenden Sie mir bis auf Widerruf alle künftigen c't-Ausgaben ab Monat

(Kündigung 3 Wochen zum Jahresende möglich.)

Das Jahresabonnement kostet DM 58,— inkl. Versandkosten und MwSt.

Absender und Lieferanschrift

Bitte in jedes Feld nur einen Druckbuchstaben (ä - ae, ö - oe, ü - ue)

Vorname/Zuname

Straße/Nr.

PLZ

Wohnort

Datum/Unterschrift

Ich bestätige ausdrücklich, vom Recht des schriftlichen Widerrufs innerhalb einer Woche nach Abschluß beim Verlag Heinz Heise GmbH, Postfach 2746, 3000 Hannover 1, Kenntnis genommen zu haben.

Unterschrift

Bitte beachten Sie, daß diese Bestellung nur dann bearbeitet werden kann, wenn beide Unterschriften eingetragen sind.

c't - magazin für computer technik Kontaktkarte

Ich beziehe mich auf die in c't ____/8__, Seite ____ erschienene

- ☐ Anzeige ☐ redaktionelle Besprechung
☐ und bitte Sie, mir weitere **Informationen** über Ihr Produkt _____
☐ und gebe die nachfolgende **Bestellung** unter Anerkennung Ihrer Liefer- und Zahlungsbedingungen auf:

Menge	Produkt/Bestellnummer	à DM	gesamt DM

Absender nicht vergessen!

Datum, Unterschrift (für Jugendliche unter 18 Jahren der Erziehungsberechtigte)

c't - magazin für computer technik Kontaktkarte

Ich beziehe mich auf die in c't ____/8__, Seite ____ erschienene

- ☐ Anzeige ☐ redaktionelle Besprechung
☐ und bitte Sie, mir weitere **Informationen** über Ihr Produkt _____
☐ und gebe die nachfolgende **Bestellung** unter Anerkennung Ihrer Liefer- und Zahlungsbedingungen auf:

Menge	Produkt/Bestellnummer	à DM	gesamt DM

Absender nicht vergessen!

Datum, Unterschrift (für Jugendliche unter 18 Jahren der Erziehungsberechtigte)

c't-Abonnement

Abrufkarte

Ich wünsche Abbuchung der Abonnement-Gebühr von meinem nachstehenden Konto. Die Ermächtigung zum Einzug erteile ich hiermit.

Name des Kontoinhabers

Bankleitzahl

Konto-Nr.

Geldinstitut

Ort des Geldinstituts

Bankinzug kann nur innerhalb Deutschlands und nur von einem Giro- oder Postscheckkonto erfolgen.

Antwortkarte



Vertriebsabteilung
Verlag Heinz Heise GmbH
Postfach 2746

3000 Hannover 1

Bitte mit der jeweils gültigen Postkartengebühr freimachen

c't-Abonnement

Abrufkarte

Abgesandt am

198

zur Lieferung ab

Heft 198

Jahresbezug DM 58,—
inkl. Versandkosten und MwSt.

c't-Kontaktkarte

Anschrift der Firma, bei der Sie bestellen bzw. von der Sie Informationen erhalten wollen.

Absender
(Bitte deutlich schreiben)

Vorname/Name

Beruf

Straße/Nr.

PLZ Ort

Telefon Vorwahl/Rufnummer

Postkarte

Bitte mit der jeweils gültigen Postkartengebühr freimachen

Firma

Straße/Postfach

PLZ Ort

c't-Kontaktkarte

Abgesandt am

198

an Firma

Bestellt/angefordert

c't-Kontaktkarte

Anschrift der Firma, bei der Sie bestellen bzw. von der Sie Informationen erhalten wollen.

Absender
(Bitte deutlich schreiben)

Vorname/Name

Beruf

Straße/Nr.

PLZ Ort

Telefon Vorwahl/Rufnummer

Postkarte

Bitte mit der jeweils gültigen Postkartengebühr freimachen

Firma

Straße/Postfach

PLZ Ort

c't-Kontaktkarte

Abgesandt am

198

an Firma

Bestellt/angefordert

NEUE SUPERHITS RUND UM COMMODORE



Der COMMODORE 64 ist ein Musikgenie, und mit diesem Buch lernen Sie alles über seine musikalischen Fähigkeiten. Der Inhalt reicht von einer Einführung in die Computermusik über die Erklärung der Hardware-Grundlagen und die Programmierung in BASIC bis hin zur fortgeschrittenen Musikprogrammierung. Zahlreiche Beispielprogramme und leicht verständliche Darstellung. Erschließen Sie sich die Welt des Sounds und der Computermusik mit dem MUSIKBUCH ZUM COMMODORE 64. Ca. 200 Seiten, DM 39,-



Graphik ist eine der Hauptstärken des COMMODORE 64. Mit diesem neuen Buch lernen Sie, wie Sie die graphischen Fähigkeiten programmtechnisch optimal nutzen, von einfachen Figuren über Sprites, Zeichensatzprogrammierung, Harcoopy und IRQ-Handling bis hin zu funktionen Darstellung, Statistik, 3-D, CAD und den Geheimnissen der Actionspiele. Zahlreiche Beispielprogramme ergänzen dieses Buch, das die lernfördernde Computergrafik jedermann zugänglich macht. Ca. 250 Seiten, DM 39,-



Umfassendes Nachschlagewerk zum COMMODORE 64 und seiner Programmierung. Allgemeines Computerlexikon mit Fachwissen von A-Z und Fachwörterbuch mit Übersetzungen wichtiger englischer Fachbegriffe - das DATA BECKER LEXIKON ZUM COMMODORE 64 stellt praktisch drei Bücher in einem dar. Es enthält eine unglaubliche Vielfalt an Informationen und dient so zugleich als kompetentes Nachschlagewerk und als unentbehrliches Arbeitsmittel. Ein Muß für jeden COMMODORE 64 Anwender! Ca. 350 Seiten, DM 49,-



Nicht nur alles über Interfaces und Ausbaumöglichkeiten des COMMODORE 64 enthält dieses Buch, sondern auch über seine vielfältigen Einsatzmöglichkeiten von der Lichtorgel über Motorsteuerung, Spannungs- und Temperaturnessung bis zur programmierbaren Stromversorgung. Zehn komplette Schaltungen zum Selberbauen, vom Eprommer über den Digital-Voltmeter mit automatisch er Messbereichswahl und den Logic Analyzer bis zur preiswerten Sprachengabe-Sprachausgabe. Jeweils mit Schaltung, Layout und Softwarelisting. Ca. 220 Seiten, DM 49,- ab April 84.



64 FÜR EINSTEIGER ist eine sehr leicht verständliche Einführung in Handhabung, Einsatz, Ausbaumöglichkeiten und Programmierung des COMMODORE 64. Die keinmal Vorkenntnisse voraussetzt. Schritt für Schritt führt das Buch Sie in die Programmiersprache BASIC ein, wobei Sie nach und nach eine komplette Adressenverwaltung erstellen, die Sie anschließend nutzen können. Das Buch ist sowohl als Einführung als auch als Orientierung vor dem 64er Kauf gut geeignet. Ca. 200 Seiten, DM 29,-



Diese neue, umfangreiche Programmsammlung hat es in sich. Über 50 Optikenprogramme für den COMMODORE 64 aus den unterschiedlichsten Bereichen, vom Superpiel über Graphik- und Soundprogramme sowie Utilities bis hin zu Anwendungsprogrammen. Der Hit sind zu jedem Programm aktuelle Programmertips und Tricks der einzelnen Autoren zum Selbermachen. Also - nicht nur abtippen, sondern auch abkabeln und wichtige Anregungen für die eigene Programmierung sammeln. Ca. 250 Seiten, DM 49,-

... und natürlich die bewährten Bestseller aus bester Hand

Insgesamt über 200.000mal wurden die nachfolgenden Bücher in nur 12 Monaten verkauft. Machen auch Sie mehr als Ihrem COMMODORE mit diesen beliebten und bewährten Bestsellern aus bester Hand.



Endlich ein umfangreiches Trainingshandbuch, das Ihnen detailliert den Umgang mit SIMON'S BASIC erklärt. Ausführliche Darstellung aller Befehle und ihrer Anwendung. Zahlreiche Beispielprogramme und Programmiertricks. Dieses Buch sollte jeder SIMON'S BASIC-Anwender haben! Ca. 300 S., DM 49,-

Eine leicht verständliche Einführung in das Programmieren des C-64 in Maschinensprache und Assembler. Komplett mit vielen Beispielen sowie einem Assembler, Disassembler und einem Einzelschritt-Simulator. Und natürlich zugeschnitten auf Ihren Computer, den COMMODORE 64. Ca. 200 S., DM 39,-

64 INTERN erklärt detailliert Technik und Betriebssystem des C-64 und die Programmierung von Sound und Graphik. Ausführlich dokumentiertes ROM-Listing, zahlreiche lauffertige Beispielprogramme und 2 Original-Schaltpläne zum Ausklappen. Dieses Buch sollte jeder 64-Anwender und Interessent haben. Ca. 320 S., DM 69,-

64 TIPS & TRICKS ist eine echte Fundgrube für jeden COMMODORE 64 Anwender. Umfangreiche Sammlung von POKE's und anderen nützlichen Routinen, BASIC-Erweiterungen, Graphik und Farbe für Fortgeschrittene, CP/M, Multitasking, mehr über Anschluss- und Erweiterungsmöglichkeiten und zahlreiche lauffertige Programme. Ca. 290 S., DM 49,-

64 FÜR PROFIS zeigt, wie man erfolgreich Anwendungsprobleme in BASIC löst und verrät Erfolgsgeheimnisse der Programmierprofis. 5 komplett beschriebene, lauffertige Anwendungsprogramme (z.B. Adressverwaltung) illustrieren den Inhalt der einzelnen Kapitel beispielhaft. Mit diesem Buch lernen Sie gute und erfolgreiche BASIC-Programmierung. Ca. 320 S., DM 49,-

DAS GROSSE FLOPPY-BUCH erklärt detailliert die Arbeit mit der Floppy VC-1541, von der sequentieller Datenspeicherung bis zum Direktzugriff, für Anfänger, Fortgeschrittene und Profis. Ausführlich dokumentiertes DOS-Listing, zahlreiche lauffertige Beispielprogramme, z.B. Disk Editor und Haushaltsbuchung. Ca. 320 S., DM 49,-

VC-20 INTERN ist für jeden interessant, der sich näher mit Technik und Maschinensprache und dem VC-20 auseinandersetzen möchte. Detaillierte technische Beschreibung des VC-20, ausführliches ROM-Listing, Einführung in die Maschinensprache und 3 Original-Schaltpläne. Ca. 230 S., DM 49,-

VC-20 TIPS & TRICKS ist eine echte Fundgrube für jeder VC-20 Anwender. Sound und Graphik Programmierung, Speicherbelegung und Speichererweiterungen, POKE's und andere nützliche Routinen, zahlreiche lauffertige Beispielprogramme und Anwendungsprogramme und vieles andere mehr. Ca. 230 S., DM 49,-

Darauf haben Sie gewartet!



Die neue DATA WELT ist eine Computerzeitschrift speziell für COMMODORE-Anwender. Brandaktuell (detaillierte Informationen über die neuen COMMODORE Computer 264 und 364) und randvoll mit Berichten, Trends und interessanten Programmertips. 80 Seiten stark im Magazinformat. Gleichzeitig als Nachfolger des VC-Infos: umfassende Übersicht über aktuelle Produkte, Bücher und Programme rund um COMMODORE 64 und VC-20. Die Frühjahrsausgabe der neuen DATA WELT erhalten Sie ab Anfang März überall dort, wo es DATA BECKER BÜCHER und -PROGRAMME gibt. Am besten gleich holen oder direkt bei DATA BECKER gegen DM 4,- in Briefmarken anfordern.

IHR GROSSER PARTNER FÜR KLEINE COMPUTER

DATA BECKER

Merowingerstr. 30 · 4000 Düsseldorf · Tel. (0211) 3100 10 · im Hause AUTO BECKER

DATA BECKER BÜCHER und PROGRAMME erhalten Sie im Computer-Fachhandel, in den Computerabteilungen der Kauf- und Warenhäuser und im Buchhandel. Auslieferung für Österreich Fachbuch-Center ERB, Schweiz T-HALIAG und Benelux COMPUTERCOLLECTIEF.

BESTELL-COUPON

Ersenden an: DATA BECKER · Merowingerstr. 30 · 4000 Düsseldorf 1

☐ per Nachnahme ☐ 299 DM 5,- Versandkosten ☐ verrechnungsscheck liegt bei

☐ DATA WELT 1/84 (DM 4,-) in Briefmarken liegen

Name und Adresse bitte deutlich schreiben

8	Leserbriefe
16	Software-Service
107	Platinen-Service
12	c't-aktuell
102	c't-Buchkritik
107	Inserenten-Verzeichnis
108	Vorschau auf Heft 5/84
108	Impressum
	<i>c't-Prüfstand</i>
29	Langsam, aber gründlich Hand-held für unter 1000 DM: Casio FP 200
32	MAX aus Deutschland Vielseitiger Kleincomputer
	<i>Software-Know-how</i>
42	Grafik, die auf jedem Rechner läuft Balken und Kurven in Microsoft-BASIC
45	BASIC intern Was nicht im Handbuch steht
62	Benchmark-Test — eine Computer-Meßplatte?
64	FORTH anpassen Teil 2: Kommunikation mit FORTH
68	GENIE als ASCII-Fernschreiber Serieller Datenverkehr im Einonverfahren
72	Was heißt CP/M-kompatibel?
85	Arithmetik für 6502 und Z 80 Teil 2: Multiplizieren und Dividieren

c't-Titel

SuperTape

Vor den Programm- und Datenaustausch zwischen Computern verschiedenen Fabrikats haben die Hersteller viele Hindernisse gestellt. Eine der größten Hürden besteht im Einsatz unterschiedlicher Formate und Verfahren für die Aufzeichnung auf Datenträgern. c't will dieses Problem aus der Welt schaffen. Wir stellen ein schnelles und sicheres Kassettenformat vor, das voraussichtlich für alle populären Homecomputer verwendet werden kann. SuperTape ermöglicht eine Übertragungsrate von (wahlweise) 3600/7200 Baud. Auf eine C60-Kassette passen also bis zu 3,2 MByte. Selbstverständlich eignet sich das Verfahren auch für die direkte Datenübertragung von Rechner zu Rechner. Übrigens: Für das c't-Projekt des Monats brauchen Sie diesmal keinen Lötkolben.

Seite 21



SuperTape mit dem Z80

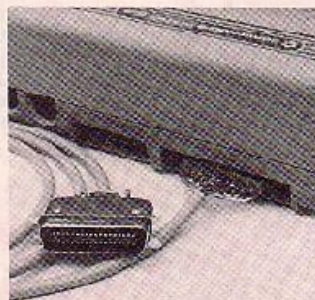
SuperTape-Basissroutinen für Z80 zum Senden und Empfangen von Daten — dargestellt am Beispiel des ZX81 — finden Sie ab

Seite 23

Centronics-Schnittstellen für VC-20, C64, ZX81

Die meistverbreiteten Homecomputer besitzen keine Schnittstellen zum Anschluß von Standard-Druckern. c't schließt diese Lücke mit einer Serie von Beiträgen. In dieser Ausgabe finden Sie Software-Lösungen für VC-20 und C64 und ein sehr preisgünstiges Interface für den ZX81 nebst Treiberprogramm für Centronics-kompatible Drucker.

Seite 92



Casio FP 200 auf dem Prüfstand

Ein Hand-Held-Computer für weniger als 1000 Mark. Kann der Casio neben seinen deutlich teureren Konkurrenten bestehen? Der c't-Test offenbart Stärken und Schwächen des FP 200.

Seite 29

Was heißt CP/M-kompatibel?

CP/M-Software läuft auf jedem CP/M-Rechner — diese Aussage gilt für fast alle kommerziell angebotenen Programme. Probleme tauchen aber dann auf, wenn man Software selbst schreibt. Dieser Bericht zeigt, wann ein Programm als kompatibel gelten darf und welche Bedingungen eingehalten werden müssen.

Seite 74

Verzeichnis

MAX aus Deutschland

Der Hirnweiser 'Made in Germany' ist sicher überflüssig bei einem Computer mit dem Namen 'MAX'. Mit MAX kann man, laut Prospekt, rechnen, schreiben, kreativ spielen, programmieren, wecken, messen, steuern und regeln. Ein wahrer Tausendsassa aus deutschen Landen also. Näheres im c't-Test auf



Seite 32

Grafik, die auf jedem Rechner läuft

Daß die grafische Darstellung oft aussagekräftiger ist als enclose Zahlenkolonnen, ist eine Binsenweisheit. Doch nicht jeder Rechner ist grafikfähig, und erst recht nicht jeder Drucker. Deshalb wurden in der EDV schon früh Techniken entwickelt, die notfalls sogar die Ausgabe auf einem Ferra-schreiber gestatten. Die Lösungen in Microsoft-BASIC ab

Seite 42

Benchmark-Test — eine Computer-Meßplatte?

Viele Leser wollten wissen, welche 'Philosophie' hinter den Benchmark-Tests steht, die auf dem c't-Prüfstand ausgeführt werden. In diesem Bericht stellen wir die Programme vor und beleuchten zugleich kritisch den Sinn solcher Geschwindigkeitsvergleiche.

Seite 62

Super-Chips

Ein Mikroprozessor mit Großrechner-Leistung: der NS32032 ist derzeit der leistungsfähigste Baustein aus der 16000-Familie von National, deren Architektur sich von herkömmlichen deutlich unterscheidet. Unseren Report über die Mikro-Avantgarde finden Sie auf

Seite 80

Einfacher A/D-Wandler

Soll ein Computer Vorgänge aus der 'analogen Welt' erfassen, überwachen oder regeln, so müssen diese in die Digitalform übersetzt werden. In diesem Praxistip stellen wir einen einfachen und sehr preisgünstigen Analog/Digital-Wandler vor, der sich zur genauen Umsetzung von Spannungswerten eignet.

Seite 78

Die neue c't-Serie: BASIC intern — Was nicht im Handbuch steht

Wie funktionieren Interpreter und Compiler? Wie kann man das 'serienmäßige' BASIC erweitern? Mit welchen Tricks arbeiten die Programmierer von Microsoft? — Eine c't-Serie für alle, die mehr als die 'Benutzeroberfläche' ihres Computers kennenlernen möchten. Also auch für Assembler-Freaks.

Seite 45



c't-Programm

WordStar als Grafik-Editor
Blockgrafik mit OSBORNE 1 und MX 80 **75**

Computer zu Hause

Zahlen Sie zuviel Steuern?
Gesamtlohnsteuertabellen 1984 **49**

Mischkulturen
Spectrum 48 K als Hobbygärtner **52**

c't-Projekte

SuperTape
Ein Kassettenformat für alle Computer **21**

SuperTape mit dem Z 80
Am Beispiel des Sinclair ZX 81 **23**

Das Betriebssystem CP/M 86
Anpassung an den c't 86, Teil 2 **36**

Nur das Kabel löten
Centronics-Schnittstelle für VC-20 und C 64 **92**

Centronics-Interface für ZX 81 **96**

c't-Praxistip

Einfacher A/D-Wandler
CA 3162 in 6502- und Z 80-Systemen **78**

c't-Report

Super-Chips
Nationals 32032 und seine Geschwister **80**

Grundlagen

I/O-Techniken
Drucken per Interrupt **57**

c't erscheint 12mal im Jahr
zur Monatsmitte.

c't 5/'84 erhalten Sie ab
12. 4. 84.

Nichts zu retten

Heute habe ich Ihre 3. Ausgabe von c't in Händen und kann schon nach kurzer Durchsicht sagen, daß es auch hier nichts mehr zu retten gibt! Es freut mich, daß ich diese Zeitschrift nicht gleich abonniert habe, obgleich ich das nach den ersten 2 Ausgaben tun wollte!

Anscheinend gibt es aber auch unter Ihren 'Schreibern' VC-20/C-64- und Sinclair-Freaks, die es nicht lassen können, weiteren 'Sabbel' darüber zu verbreiten! Wenn ich die Computer-Zeitschriften so anschau, kommt es mir vor, als gäbe es nur 2 Computer — den Commodore und Sinclair!! Nicht, daß ich verärgert bin, daß man keinen Beitrag über Genie bringt — darauf kann ich auch verzichten, aber 'Listing-Hefte' gibt es doch schon genug — oder? In Ihre Zeitschrift gehören m.E. nur Beiträge, wie z.B. der Bau des c't-86, des Terminals, Floppy usw., wie man es in der 'anderen' nicht mehr findet! Wenn Sie davon abweichen, werden Sie viele Leser verlieren — als ersten mich!

Martin Schuster, Heidenheim

Mehr Grundsätzliches

Ich möchte Ihre Zeitschrift bisher insgesamt loben. Insbesondere fand ich positiv, daß Sie bei Konstruktionsentwürfen immer einen 'Grundsatzartikel' zu einem der eventuell unbekannten Bauteile brachten (6511Q beim Terminal, 8086 beim c't 86) und möchte Sie bitten, damit nicht etwa aufzuhören, sondern dies eher noch auszuweiten. So wäre ich an einer genaueren Beschreibung des SC-01A interessiert gewesen.

Sie verweisen immer wieder auf die Datenblätter usw. der Hersteller (u.a. bei der Beschreibung des 6511Q). Leider konnte ich jedoch in den Elektronikläden der Umgebung solche Datenblätter nicht erhalten und auch keine Auskunft darüber, wer sie hier in Deutschland vertreibt. Wäre es Ihnen möglich, dies in Zukunft immer mit zu veröffentlichen?

C. Flothow, Sinsheim

Ihre Anregungen nehmen wir uns zu Herzen. Datenblätter erhalten Sie beim Hersteller oder

dessen Repräsentanten. Hier die Anschrift der deutschen Rockwell-Vertretung:

Rockwell International GmbH
Fraunhoferstr. 11
8033 München-Martinsried
(Red.)

c't-Terminal-Computer: Nur eine Werbebeilage

Um es vorwegzunehmen, Ihr o.g. Artikel erscheint mir — etwas überspitzt formuliert — als Werbebeilage der Hardware-Lieferfirma.

Mein Hauptkritikpunkt: Der Artikel ist im Anspruchsniveau entschieden zu hoch, jedenfalls nur schwer nachvollziehbar. Letzteres scheint mir beabsichtigt.

Sie haben recht damit, daß nicht jeder Artikel bei 'Adam und Eva' beginnen kann. Doch ist beispielsweise die praktische Umsetzung von Blockschaltbild 4 (c't 12/83, S. 44) in die Schaltung schlecht beschrieben. Außerdem bin ich der Meinung, daß ein Terminal für 'den anspruchsvollen Anwender' mit einer deutschen Tastatur beginnt. Die Angaben zum Anschluß einer solchen sind aber sehr mangelhaft. Die Aussage, daß ein Assembler-Listing nicht vorliegt, weil es 'bislang keinen Assembler für den ... 6511-Prozessor' gibt, ist schlicht eine Ausrede: Sie hätten die Abkürzungen aus Tab. 1 (c't 1/84, S. 89) verwenden können und so mindestens exemplarische Teile des Programms nachvollziehbar machen können. Erst so sind die von Ihnen häufig angesprochenen 'Programmanpassungen' möglich.

Der Entwickler bzw. der Artikelschreiber hat sich wahrscheinlich wochenlang mit der speziellen Materie beschäftigt. Er muß diese aber auch Leuten, die nicht hauptberuflich mit solchen Entwicklungen befaßt sind, vermitteln können — so jedenfalls verstehe ich das Konzept einer Zeitschrift wie der Ihren.

Allerdings drängt sich mir der Verdacht auf, daß dies gar nicht beabsichtigt ist. Zwar ist es durchaus legitim, den Gewinn durch das Zusammenspiel zwischen Zeitschrift und Hardware-Entwicklern (bzw. Lieferfirmen) zu maximieren, wenn dabei der Lerneffekt jedoch völlig auf der Strecke bleibt, kann ich auf den Kauf der Zeitschrift verzichten und gleich ei-

nen fertigen Computer kaufen. Ich hoffe, Sie verstehen meine Kritik als Anregung.

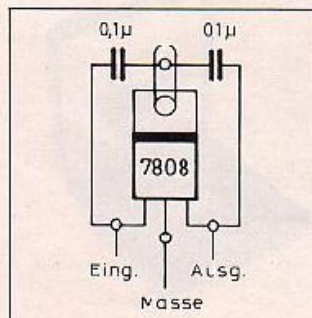
Helmut Bartz, Frankfurt

Da sind Sie auf dem Holzweg, ein Komplott zur Verschleierrung von Know-how gibt es nicht. Es liegt wohl eher in der Natur komplexer Projekte, daß sie höhere Anforderungen an das Verständnis stellen. Dem Terminal-Projekt hat c't bisher zwölf Druckseiten mit — zugegeben — konzentrierter Information gewidmet. Dies halten wir für einen tragbaren Kompromiß zwischen den Interessen der Leser, die das Terminal bauen und eventuell nach eigenen Vorstellungen verändern möchten, und denjenigen, denen andere Themen wichtiger sind. (Red.)

Den Kopf halt kühl

Ein dickes Lob an Fa. Sinclair! Sie stattete ihren Spectrum arwenderfreundlich mit einer aufwendigen, thermostatisch gesteuerten Heizungsanlage aus. Als Heizelement dient ein 5-V-Spannungsregler-IC, der so bei 50°C abschaltet. Allerdings ergibt sich ein kleiner Punkt der Kritik:

Die CPU wird hierdurch auch erwärmt. Damit wird aber nicht folgenlos gegen die Regel '... den Kopf halt kühl ...' verstoßen. Kurzum der Spectrum wird nach einiger Zeit eigenwillig (Variablen mit mehreren Buchstaben lassen sich nicht mehr eingeben). Deshalb ist es unerlässlich, die 15V des Netzteils, wie bereits für den ZX81 empfohlen, durch ein Spannungsregler-IC 7808 auf 8V herabzusetzen. Als Nebenwirkung verschwindet dadurch auch das lästige Sirren der eingebauten Spannungswandler-spule. Der Spannungsregler läßt sich dabei ganz einfach aufbauen, wie nachstehender Schaltplan zeigt:



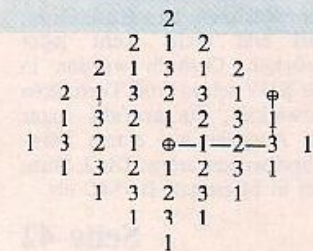
Beim Aufbau der Schaltung sind die Anschlußdrähte zu Ösen zu biegen. Die Abblockkondensatoren können an einer L31Öse befestigt werden, die an der Kühlfahne des ICs befestigt wird. Damit die ganze Mimik nicht zu heiß wird, sollte das IC auf ineinandergestellte Fingerkühlkörper (57x57 und 42x42 mm) geschraubt werden.

G. Pfeifer, Regensburg

Layout aus dem Mikro: Fehlgeroutet

An Eurem an sich sehr interessanten Artikel ist mir eine Unsauberkeit unangenehm aufgefallen:

S. 60 'Rückwärts routen'. Gelte ich wie das Programm nach dem Schema: oben, unten, links, rechts vor, so finde ich den folgenden Weg (Bild 3e, S. 59)



und nicht den von Euch im Bild 3e veröffentlichten. Der veröffentlichte Weg gefällt mir aber besser, was nun?

Bernhard Beyhl, Frankfurt

Formel falsch

In diesem Artikel hat 'Murphy' ganz fürchterlich zugeschlagen!

Man berechne die Speicheradresse für $x = 255$ und $y = 127$ und $MTX0 = 2000H$. Sie lautet: $11F7FH$! (5 Stellen!!!) Grund: die Formel ist falsch!

Sie lautet richtig:
 $XYADR = MTX0 + X - (Y * X_{max})$

Auch im Programm müssen die Register entsprechend getauscht werden.

Bitte etwas mehr Sorgfalt bei der so unangenehmen Theorie; sonst geht's meistens schief!

Uwe Kloß, Braunschweig

Beide Leser haben recht. (Bis auf die Tatsache, daß Murphy mit diesem Versehen nichts zu tun hat.) (Red.)

E.V.G.

liefert



Das Komplettsystem — VME Bus

CPU: 68000 8(10) MHz, CPU-1 Karte, 640 KByte RAM / 256 KByte EPROM, 20 MByte (form.) Platte & 1 MB Floppy

Betriebssystem: COHERENT* (Superset von UNIX*)
Multiuser / Device indep. I/O

pSOS-68h: Echtzeitbetriebssystem

Sprachen: ASSEMBLER, BASIC, C, FORTRAN77 (v. 71)

SYS68K VME Bus Karten:

CPU-1: 68000, 8(10) MHz, 128 (512) KByte RAM, 32 (256) K EPROM Uhr (1/1000 s ... Monate) mit Akku, 24 Bit Timer mit int/ext. Clock, 3 RS232 (110...38400 Bd), 16 Bit parallel I/O + 4(8) Handsch.

CPU-3: CPU Karte mit MMU für Multiprozessor

WFC-1: Winchestercontroller für 3 Drives und 4 Floppys

RR-1: 128...512 KByte CMOS RAM mit Akku und/oder ROM/EPROM/DRAM

GDC-1: Grafik, 16 Farben gleichzeitig, 1Kx1K Punkte, 512x512 Fenster, Grafikprozessor

ADDA: AD-Wandler mit 32 (10 diff.) Kanälen, 12 Bit, 60 us, Eingänge per Software einstellbar, Empf. max. ± 0 mV, DA-Wandler, 2 Kanäle, 12 Bit, 5 us

Weitere Karten auf Anfrage.

Für Einsteiger das preiswerteste 68000 Entwicklungssystem zum Anschluß an bestehende Rechner (CP/M Kommunikations- und Datentransfer Software verfügbar):

* UNIX ist Warenz. der BELL LABS, COHERENT von MARK WILLIAMS

FORCE Profi Kit 2

68000, 8 MHz, 123 (512) KByte RAM, max. 128 KByte EPROM, 2xRS232, 16 (+13) Bit parallel I/O, 3 Timer, Kassettenrecorder Interface zur Datenaufzeichnung, FORCEMON im Lieferumfang

Software:

FORCEMON:

Monitor und umfangreicher Debugger mit Tracemode, Assembler- und Disassemblerfunktion

FORCEMON/IDEAL:

komfortabler Bildschirm Editor und Assembler, die zusammenarbeiten: Interaktive Fehlerkorrektur, Labelnamen: 20 Stellen signifikant, Unterstützung blockstrukturierter Programmierung

BASIC-68K:

Editor (mit Such-&Ersetzfunktion), Compiler für erweitertes ANSI BASIC, 2 bis 10mal so schnell wie andere 16 Bit BASICs: z.B. leere FOR Schleife 10000mal in 1,31 sec, mit Hilfsmitteln für strukturiertes Programmieren: IF/THEN/ELSE/ENDIF, DO/UNTIL, WHILE/DOWHEND, Typenklärung und strenge Typenüberprüfung: REAL, DOUBLE PRECISION, INTEGER, STRING, BOOL

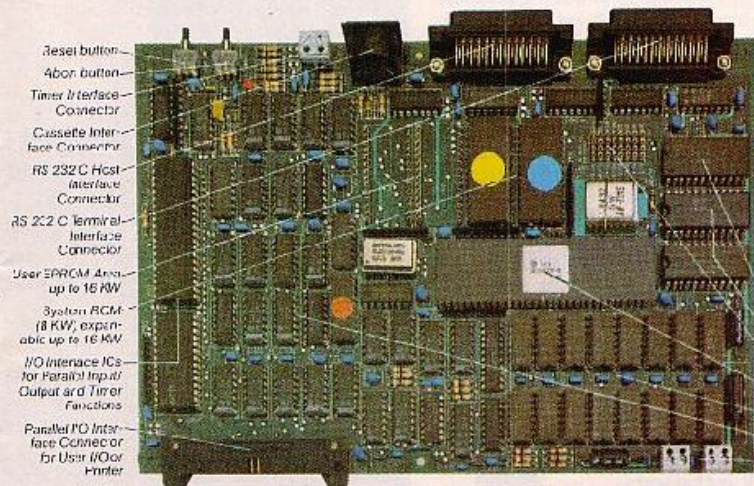
Zahlensysteme: Hex/Dec/Oct/Bin, Namen beliebig lang

Fehlersuchhilfen: ON ERROR GOTO, TRACE Mode, Assemblerprogramme können leicht eingebunden werden.

FORTH:

fig FORTH 79 Standard mit zusätzlichen Befehlen zur Unterstützung des großen Adreßraumes

Die oben genannte Software, deren Dokumentation in Deutsch oder Englisch geliefert werden kann, ist auch auf der CPU-1 Karte lauffähig, die man dann später zu einem VME System ausbauen kann.



Preise (incl. MwSt.)

PROFIKIT 2 1995,—
FORCEMON/IDEAL 324,90
BASIC-68K 387,60

FORTH-68K 387,60
SYS68K/CPU1 387,—
Handbücher einzeln 34,20

SYS68K/STAND1
System mit 20 MByte 19528,—

Technischer Support für OEMs verfügbar.

Falls Sie Ihre CPU-1 oder den Profi Kit 2 selbst erweitern wollen: WD-1002 Winchester + Floppy Controller, fertiges Modul zum Anschluß von Standardlaufwerken, schreibt/liest und formatiert automatisch für nicht mehr als DM 1024,— incl. MwSt.

Dazu passend: Slim-Line 5¼" Winchester-Laufwerk, 10 MByte formatiert DM 2595,— incl. MwSt.

Computer für die Industrie
Computer für Institute
Computer für Amateure
Bauteile + Systemteile

E.V.G.

ELECTRONIC VERTRIEBS-GMBH
Meisenweg 10 · D-3012 Hannover-Langenhagen
Tx.: 9 23 203 evghl d · Tel.: 05 11/78 2078

H A R D W A R E

ren Funktionen wie Vor- und Rücklauf usw., Nr. 122, DM 119,-.

KEYBOARDS

Diese Tastatur kann direkt nach Abnehmen der Originalblende und dardarunterliegenden Silikonstastenmatte ausgetauscht werden. Zum Lieferumfang gehört die komplette Aufsatz tastatur und die Original-SINCLAIR-Beschriftung, die auf die Tasten geklebt und mit transparenten Abdeckungen versehen wird. In dieser

DER SINCLAIR-ZX81-BAUSATZ

haben. Denn die Montageanleitung stammt:

DAS EINSTEIGER - PAKET

Für alle, die die Welt und die Faszination der Microcomputer erleben wollen und natürlich für alle Computer-Fachleute: heben wir ein SUPER-EINSTEIGER-PAKET geschnürt. Unter der Bestell Nr. C07 erhalten Sie für nur DM 498,- den kompletten ZX 81-Bausatz wie unten beschrieben und den SEIKOSHA GP 505 (Beschreibung nebenstehend). Inbegriffen ist ein 212-seitiger Basic-Kurs zugleich III-ZX 81-Handbuch. Auf die Komplett-Mini-Microcomputer-Anlage sind wir stolz, hier nochmals die wichtigsten Daten zum Bestellen: Nr. 007, DM 498,-.

aus dem Hause SINCLAIR. Für ganze DM 129,- erhaltet Siedenkompletten Bausatz mit dem Original 212-seitigen Handbuch, Netzteil, Anschlußkabel für TV und Kassetten-



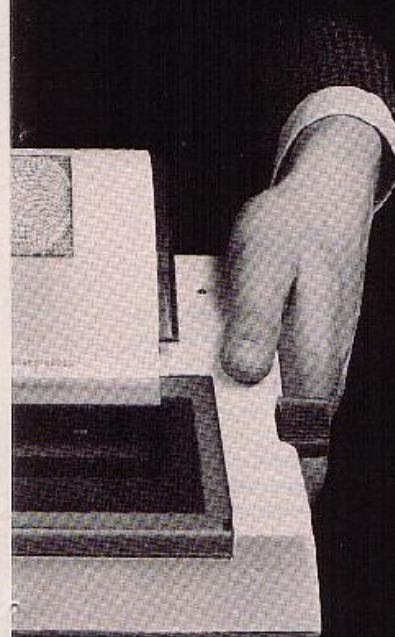
recorder. Und 8K-Byte BASIC ROM, 1K-Byte RAM und Z80A-CPU. Außerdem haben Sie auf alle Teile die Original SINCLAIR Garantie. Achten Sie auf unser SUPER-EINSTEIGER-PAKET – Drucker und Bausatz zu einem Super-Preis. Den Bausatz alleine bestellen Sie bitte Nr. 001, DM 129,-.

A black and white photograph showing a close-up of a person's hand pointing their index finger at a document. The document is partially visible, showing a fingerprint and a small logo that reads 'GP'. The background is dark and out of focus.

Der Normalpapier-Drucker mit eingebautem Interface für den SINCLAIR ZX 81 und ZX-SPECTRUM 16 und 48K. Mit Sinclair Normstecker und Netzteil. Sofort betriebsbereit. Der Friktionsantrieb gestattet die Verwendung von Rollenpapier und Einzelblatt-Papier bis zu 127 mm Breite. Modus für Grafik, einfache und doppelte Zeichenbreite innerhalb einer Zeile möglich. Voll grafikfähig, Normalschrift und doppelte Schriftbreite, Druckposition

PROGRAMM

Seikosha Graphic Printer
einer. Handlich, praktisch,
beiden, was in ihm steckt.
gebautes Interface. Ohne
nde sofort funktionsfähig.
Kleine groß. **RIESIG.**
DM 398,-



durch Zeichen oder Punkt adressierbar (Positionssteuerung).
Das Druckformat, 5 x 8 Punkt-Matrix-Druckkopf
Druckgeschwindigkeit: 40 Zeichen/s
Max. Spaltenzahl: 46 Spalten
(= 322 Punkte)
Druckarten: Standardzeichen, doppelte Zeichenbreite und Grafik
Nutzen: 1 Original und 1 Kopie
SEIKOSHA GP-50S, 1 Papierrolle, Farbband, Netzteil und Handbuch
Best.-Nr. 136. DM 398,-.

COMPUTER BÜCHER

Unentbehrliche Nachschlagewerke, faszinierende Programme und jede Menge Tips und Tricks für Anfänger und Fortgeschrittene.



Hier die ZX81 Bibliothek:

49 Explosive Spiele. Nr. 200. DM 29,80
34 1K Super-Spiele. Nr. 201. DM 19,80
Entdecken Sie die unendlichen Dimensionen Ihres ZX81. Nr. 202. DM 29,80.
Das ZX81 Buch. Nr. 203. DM 29,80.
Das ZX81 ROM-Buch. Nr. 204. DM 39,80

Und die SPECTRUM Edition:
Spectrum Spektakulär: Fifty-fifty Spaß und Nutzen. Nr. 205. DM 29,30.
Spaß & Profit SPECTRUM. 60 Spiele und nützliche Anwendungen. Nr. 206. DM 24,80.
Das Spectrum Buch. Programmieren in Maschinensprache und Spielprogramme. Nr. 207. DM 29,80.
Das SPECTRUM KUM. Nr. 209. DM 39,80.
SPECTRUM ohne Grenzen. Über 100 Programme und Routinen. Nr. 208. DM 29,80.

ZX81 - SOFTWARE

Jeder Computer ist so tüchtig wie die Software, die für ihn angeboten wird. Hier ein Elite-Angebot für der ZX81: Die 4 folgenden Programme sind Profi-Anwender-Programme:

Basic-Compiler/M-Coder. Nr. 300. DM 29,50.
VU-Calc. Kalkulationsprogramm. Nr. 322. DM 59,-.
VU-File. Dataprogramm. Nr. 321. DM 59,-.
Machine Code Test Tool. Nr. 307. DM 29,50.

Das ist unser Unterhaltungsprogramm:
Maze Death. Todesrennen. Nr. 301. DM 19,50.
Ghost Hunt. Gespensterjagd. Nr. 302. DM 19,50.
Crazy Kong. Gefährliches Abenteuer im Dschungel. Nr. 303. DM 19,50.
Iai. Invasion auf dem Staubplaneten mit viel Action. Nr. 304. DM 19,50.
Fopper. Frosch wie Frogs. Nr. 305. DM 19,50.
Cosmic Guerilla. Kosmische Danditen im Weltraum. Nr. 308. DM 19,50.
Camper/Clooper. Energie. Chaos/Fäurer & Gendarm. 2 Spiele. Nr. 309. DM 19,50.
Ocean Trader. Teufische Piraten, nicht naß werden. Nr. 310. DM 19,50.
3D black Star. Galaxis-Spiel. Nr. 311. DM 19,50.



Pioneer Trail. Wildwest Abenteuer ohne Fuzzy und Joe. Nr. 312. DM 19,50.
Asteroids. Kennt jeder. Nr. 313. DM 19,50.
Scramble. Galaxisjagd. Nr. 314. DM 19,50.
Munchos. Geister & Power Pillen steigern die Spielerpotenz. Nr. 315. DM 19,50.
Croaka-Crawla. Grüne Frösche gegen chromblitzende Trucks. Nr. 316. DM 19,50.
Defender. Kennt jeder. Nr. 317. DM 19,50.
Invaders. Die Außerirdischen mit ihren fliegenden Untertassen. Nr. 318. DM 19,50.
Galaxians & Gloops. Galaxis & Labyrinth-Spiel. 2 Spiele. Nr. 319. DM 19,50.
Schach. 6 Schwierigkeitsstufen bis zum Großmeister. Nr. 320. DM 39,50.
Flug-Simulation. Nr. 323. DM 39,50.
Weltrauminvasion. Nr. 324. DM 39,50.

SPECTRUM GALAXY-KOLLEKTION



Und hier die Galaxy-Kollektion für den SPECTRUM. Mit ausführlicher deutscher Beschreibung vom Joe. Mit diesen Programmen können Sie professional arbeiten.
Tasword II. Das einzige wirkliche brauchbare Textverarbeitungsprogramm. Über 25 Funktionen von kursiv bis Super-Lettern. und und ... Nr. 452. 69,-.
VU-3D. Entwurf. Gestaltung und Bewegung von 3 dimensionalen Körpern. Nr. 412. DM 59,50.
Adress-Manager. Adressen, Dateien, Register. Nr. 420. DM 49,-.
Machine Code Test Tool. Nr. 421. DM 49,-.
Basic-Compiler/M-Coder. Nr. 422. DM 39,50.

Editor Assembler. Nr. 425. DM 59,-.
Finance Manager. Super Finanz Programm mit vielen Funktionen. Nr. 428. DM 49,-.
Collector's Peck. Archivierungsprogramm. Nr. 433. DM 39,50.
Melbourne Draw. Das Super-Grafik-Programm, 10-fache Vergrößerung individuelle Farbgebung pro Punkt. Nr. 446. DM 49,-.
Und jetzt wird gespielt mit Super-Action, toller Grafik und Sound:
Mined out. Der gefährliche Weg über die Minenfelder. Nr. 400. DM 39,50.
Space Raiders. Banditen im Weltraum. Nr. 401. DM 29,50.
Meteor Storm. Im Kampf gegen die Astro-Wolken. Nr. 402. DM 19,50.
Space Intruders. Eindringlinge aus dem Weltall. Nr. 403. DM 19,50.
4D Time Gate. In rasendem Tempo durch die Zeit-Zonen. Nr. 404. DM 39,50.
Ghost Hunt. Geisterjagd. Nr. 405. DM 29,50.
Maze Death Race. Bei diesem Autorennen lauern tödliche Gefahren. Nr. 406. DM 29,50.
Horace goes skiing. Viel Spaß mit Horace im Schnee. Nr. 407. DM 39,50.
Hungry Horace. Katz- und Mausspiel. Nr. 408. DM 39,50.
The Chess Player. Sehr stark mit 6 Schwierigkeitsstufen. Nr. 409. DM 39,50.

Planetoids/Missile. Science Fiction. 2 Spiele auf einmal. Nr. 110. DM 29,50.
Reversi. Nr. 411. DM 39,50.
Flight Simulation. Nr. 414. DM 39,50.
Paion Chess. Nr. 415. DM 39,50.
Chess the Turk. Fast unschlagbar mit Super-Grafik. Nr. 419. DM 49,-.
Astro Blaster. Kampf um die Galaxis. Nr. 426. DM 29,50.
Horace and the Spiders. Horace in den Spinnenbergen. Nr. 427. DM 39,50.
Arcadia. All-Abenteuer. Nr. 429. DM 24,50.
Zoom. Als Abfängjäger in der unendlichen Galaxis. Nr. 430. DM 24,50.
Schizoids. Odyssee durch die Weiten der Galaxis. Nr. 432. DM 24,50.
Zip-Zap. Kolonisieren Sie die Planeten. Nr. 431. DM 24,50.
Jumping Jack. Ein lustiges und harmloses Spiel. Nr. 433. DM 24,50.
Molar Maul. Sie als todsmutiger Bakteriologe. Nr. 434. DM 24,50.
Ah Diddums. Der abenteuerlustige Teddybär im Kinderzimmer. Nr. 435. DM 24,50.
Pool. Billard. Nr. 436. DM 39,50.
Aquarius. Tauchergruppe im Einsatz gegen Mordmaschinen. Nr. 437. DM 29,50.
Magic Miner. Verschiedene Abenteuer im Wilken Westen. Nr. 438. DM 29,50.
Styx. Die abenteuerliche Reise ins Todesreich. Nr. 439. DM 29,50.
Electro Storm. Weltall-Schock. Nr. 441. DM 24,50.
Panic. Hält was der Name verspricht. Nr. 442. DM 24,50.
Light Cycle. Ein gefährliches Spiel mit dem Licht. Nr. 443. DM 24,50.
The Hobbit. Tolkien-Spiel mit phantastischer Grafik. Nr. 444. DM 78,-.
Pentratör. Wehren Sie sich gegen die Eindringlinge. Nr. 445. DM 39,50.
ZX-USER-TAPE. Die Zeitung auf Kassette, mit ganzen Programmen und brandheißen Informationen. Nr. 453. DM 19,80.

Außerdem führen wir Programme für alle Heim-Computer wie BBC, Commodore, Dragon, Oric und viele andere. Fordern Sie unsere Software-Liste an.

COMPUTER ACCESSOIRES INT'L
Der Computer-Ausstatter.

Hier wird bestellt:

☐ per Vorausscheck
☐ per Nachnahme (zusügl. Nachnahmegeb.)

Stück	Artikel-Nr.	Preis in DM
	Seikosha-Drucker GP 50S Nr.136	398,-
	Einstufiger-Paket Nr. 007	498,-

Name _____ CT 4

Straße _____

PLZ/Ort _____

Datum _____

Unterschrift _____

Bei Bestellungen unter DM 250,- zuzügl. Versandkosten.

COMPUTER ACCESSOIRES INT'L GMBH

Jägerweg 10 - 8012 Ottobrunn



'Wirtschaftsgipfel' in Hard- und Software

Über 6000 Aussteller
auf der Hannover-Messe erwartet

Wer technologische und handelspolitische Trends aufspüren, Chancen und Risiken neuer Märkte analysieren und vor allem Fachwissen aktualisieren will, für den ist die Hannover-Messe ein Pflichttermin. Als 'Wirtschaftsgipfel' in Hard- und Software bietet die Hannover-Messe eine hohe Informations- und Orientierungsqualität, auch 1984 (in der Zeit vom 4. bis 11. April) wieder im Hannover-typischen Profil von 'zehn Fachmessen zur selben Zeit am selben Ort'.

Die bislang vorliegenden Beteiligungsdaten unterstreichen erneut hohen Erwartungshorizont: gerechnet wird mit insgesamt über 6000 ausstellenden Unternehmen und Organisationen aus nahezu 50 Staaten. In den 23 Hallen und im Freigelände des größten Technologieparks der Welt wird sich die Netto-Ausstellungsfläche gegenüber dem Vorjahr um fast 10000 qm auf 412000 qm erhöhen. Dabei reicht das Technologiespektrum von neuen Energietechniken, dem thematischen Schwerpunkt der Hannover-Messe '84, über Sensorbildschirme oder die 'sprechende Zukunft' mit neuen Sprachausgabe-Systemen bis zu den jüngst gewonnenen Erkenntnissen für die Fabrikation im Weltraum.

Das CeBIT-Centrum der Büro- und Informationstech-

nik' — im Zentrum des Interesses der Mikrocomputerbranche — wird erneut erweitert und belegt in diesem Jahr rund 127000 qm Ausstellungsfläche. Noch attraktiver als im Vorjahr präsentiert sich die 'microtronic' — Innovationszentrum Mikroelektronik in Halle 12. Rund 300 Unternehmen aus dem In- und Ausland unterstreichen die Schubkraft der Mikroelektronik als einer Schlüsseltechnologie, an deren Einsatz die Wettbewerbsfähigkeit ganzer Nationen gemessen wird. In einer Sonderschau 'Innovative Anwendungen der Mikroelektronik' geht es darum, mit Produktbeispielen die vielfältigen Möglichkeiten der praktischen Umsetzung aufzuzeigen.

c't finden Sie übrigens gleichfalls in der microtronic-Halle 12, Stand-Nr. 3200.



Tageskarte im Vorverkauf 17 DM

Der Karten-Vorverkauf für die Hannover-Messe läuft auf vollen Touren. Im Vorverkauf kostet die Eintrittskarte für die zehn Fachmessen für einen Tag 17 Mark (Tageskasse: DM 22,—), die Dauerkarte 45 Mark (Tageskasse: DM 55,—).

64KBit E²PROM in CMOS

National Semiconductor hat jetzt die Spezifikationen des 64-KBit-microCMOS-E²PROMs bekanntgegeben. Der elektrisch löschbare Speicherchip arbeitet mit einer Versorgungsspannung von 5 V und bietet bei geringer Leistungsaufnahme eine hohe Arbeitsgeschwindigkeit. Mit der Einrichtung des 'Daten-Polling' kann das E²PROM einem Signalprozessor mitteilen, wann eine Schreiboperation ausgeführt ist.

Informationen:
National Semiconductor
Industriestraße 10
8080 Fürstfeldbruck

Ferien mit dem Computer

Für alle jugendlichen Computer-Fans ab dem 7. Schuljahr führt die GISE einen speziellen EDV-Kurs während der Oster- und Sommerferien durch. Ziel des ein- bis zweiwöchigen Kurses ist es, Kenntnisse in der Programmiersprache BASIC zu vermitteln. Eberfalls in der Ferienzeit werden Kurse für Erwachsene angeboten, deren Ziel die Vermittlung von EDV-Kenntnissen für die praktische Umsetzung im Arbeitsalltag ist.

Informationen:
GISE
Postfach 1240
8740 Neustadt/Saale

Intelligente Winchester-Drives

Die Winchester-Laufwerke der Serie 700S von Sugart vereinigen die Intelligenz einer SCSI-Schnittstelle mit der Leistungsfähigkeit von Winchesterlaufwerken. Dadurch ergeben sich, neben der Vereinfachung der Systemintegration, auch Kostenreduzierungen, da Laufwerk und Controller nicht mehr getrennt ausgetestet werden müssen. Die Laufwerke werden mit einer Kapazität von 5 und 10 MByte geliefert.

Informationen:
Sugart Associates GmbH
Drygalski-Allee 33
8000 München

Z800 — wann er kommt

Die schon vor zwei Jahren angekündigte, bis heute aber nicht lieferbare CPU Z800 soll nun im III/IV. Quartal 1984 in Mustern zur Verfügung stehen. Produktionsstückzahlen sollen ab dem ersten Quartal 1985 lieferbar sein.

Informationen:
ZILOG GmbH
Eschenstraße 8
8028 Taufkirchen

3600 Druckzeilen pro Minute

Ein neuer IBM-Zeilendrucker kann entweder mit 132 oder 168 Druckpositionen je Zeile arbeiten. Mit einer wählbaren Zeilenleistung von 2200, 3000 oder 3600 Zeilen pro Minute ist er der schnellste mechanische Drucker aus diesem Hause. Der Preis für den 'schnellen mechanischen' beträgt 210000 D-Mark zuzüglich MwSt.

Informationen:
IBM Deutschland GmbH
Postfach 800880
7000 Stuttgart 80

Mehr Speicher für IBM PC/XT

Die Speicherkarte 'ISAAK' für den IBM PC ist von 64 KByte bis 512 KByte ausbaufähig. Ein besonderes Merkmal dieser Platine ist nach Angaben des Herstellers, daß ihr Preis trotz doppelter Speicherausbaupazität dem Preisniveau der 256 KByte-Erweiterungskarten entspricht.

Informationen:
ISI Computer GmbH
Otto-Hahn-Straße 34
8012 Ottobrunn

Floppy für Apple II

Speziell für den Apple II und Apple IIe bietet die Firma nbn Floppy-Stationen an, die sich durch die halbe Bauhöhe der Apple Disk II Laufwerke und eine zwei- bis viermal so schnelle 'Track to Track'-Zeit auszeichnen.

Informationen:
nbn Elektronik GmbH
Gewerbegebiet
8036 Herrsching



Hazeltine

Esprit

EIN NEWCOMER MIT 13 JAHREN ERFAHRUNG

Aus der Hazeltine Terminal Division entstand Esprit Systems Inc. So kann dieses Unternehmen, obwohl ein neuer Name auf dem Elektronikmarkt, doch auf die Erfahrung eines bekannten Terminal-Herstellers verweisen. Aufbauend auf einer bewährten Produktpalette wird Esprit Neuentwicklungen und einen optimierten Kundendienst anbieten. Sie haben also die Auswahl unter einem breiten Spektrum leistungsfähiger Geräte: IBM-kompatible Terminals ebenso wie flexible OEM-Systeme mit farbigen oder monochromen Bildschirmen. Egal, für welche Version Sie sich letztendlich entscheiden: Sie bekommen ein Produkt, das sich durch ein bestmögliches Preis-/Leistungsverhältnis auszeichnet. So z. B. unser neues, DEC-kompatibles Terminal Executive 10/102, das eine neue Dimension ergonomischen Designs in seine Preiskategorie einbringt.

 **SYNELEC**
DATENSYSTEME GmbH

Lindwurmstr. 117 · 8000 München 2
Telefon 0 89/7 25 30 81 · Telex 5 212 289 syn d
Geschäftsstelle Düsseldorf: Telefon 02 11/35 02 36
Geschäftsstelle Frankfurt: Telefon 0 60 81/4 21 55

Autorisierter Distributor für die Bundesrepublik Deutschland

COMMODORE und APPLE

als 16-bit-Rechner mit
1024-K-Speicher

CCU 8086-Central Computer Unit

Erweiterungs- einheit

mit 16-bit μ P 8086 —
incl. bis zu 1024 KB
großem Speicher —
für alle 8-bit-
Computersysteme.

Standardausführung bestehend
aus:

CCU 8086 mit μ P8086 (5 MHz) und
128 KB Speicher (erweiterbar auf
1 MB), Interface-Karte passend für
Apple II, Commodore, IBM-PC
(TRS-80 und ECB-Bus in Vorbereitung),
Verbindungsgehäuse 34pinig
ca. 80 cm.
Gehäuse: 170 x 110 x 67 mm (geeig-
net für zweites CCU- oder Netzteil-
Karte) und Diskette mit Bootsoft-
ware für CP/M 86 mit RAM-Disk 16
bit. Preis incl. Mehrwertsteuer

DM 1360,—

— Software für RAM-Disk 8-bit (Apple DOS oder CP/M 80) DM 96,—
— CCU-RAM-Erweiterungssatz je 128 KB DM 390,—
— CCU komplett mit 512 KB Speicher DM 2390,—
— CCU komplett mit 1 MB Speicher DM 4490,—
— Handbuch MSDOS 2.0 in Deutsch DM 120,—
— Betriebssystem CP/M 86 auf Ihrem 8-bit-Format DM 998,40
— MS-DOS 2.0 auf Ihrem 8-bit-Format DM 364,80
— Umformung einer 16-bit-Software vom 8" Floppy
CP/M Standard-Format (IBM 3740) auf 5,25" 8-bit Host-Rechner-Format
(andere Formate auf Anfrage) DM 80,—
— Netzteil 5V/2,7A DM 274,—
— Baugruppe speziell für ITT 8030:
256-K-RAM-Karte (FUCKEPC) für CPJ Z80 DM 1090,—

ancon

COMPUTERTECHNIK

Ingenieurgesellschaft mbH

Auguste-Viktoria-Str. 62
1000 Berlin 33

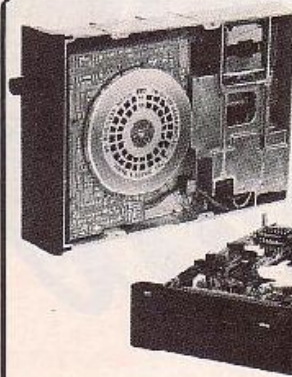


Anwendung als:

1. 16-bit-Maschine
Hauptrechner für alle 13-bit-Programme,
z.B. CP/M 86, MS-DOS 2.0 usw. Der 8-bit-
Host-Rechner (Apple, Commodore, IBM-
PC usw.) wird in diesem Falle als Terminal
mit Floppy-Disk verwendet.

2. RAM-Disk 8-bit
definiert den CCU-Speicherbereich als
Disk-Laufwerk im bisherigen 8-bit-Compu-
ter (z.Z. für Apple DOS, Apple Z80,
ITT 3030) herbar.

3. RAM-Disk 16-bit
Der Speicherbereich über 128 KB kann
auch als RAM-Disk verwendet werden.



CE-DATA®

Slimline 5,25" FD Disk
250 KB - 1 MB

Einsetzbar für alle
Systeme

CE-DATA Slimline Laufwerk 5,25"
10 Track 5C/DD, 250 KB

Laufwerke mit größerer Kapazität, ab

lager lieferbar, Preise auf Anfrage

CE-DATA Doppelfloppy im Gehäuse, komplett

betriebsfertig, à 250 KB

Floppy Disk Controller für Genie I, II

und Tandy TRS 80 Mod. I, single + double

density mit Druckerschnittstelle,

betriebsfertig im Gehäuse

Double Density Controller

Floppy Disk Controller für Apple (universal)

incl. Anschlußkabel

Apple Controller für Original-Laufwerke

Verbatim Disketten

3ASF Disketten

Siemens PT 88 Tintenstrahldrucker

StarPrinter Gemini 10 X

DM 590,—

DM 1425,—

DM 675,—

DM 198,—

DM 189,—

DM 115,—

ab DM 52,—

ab DM 55,—

UM 1.750,—

DM 1145,—

CE COMPUTER ELECTRONIC GMBH

Felchhofstr. 55 · 5840 Schwerte-Westhofen

Tele. 0 23 04 / 6 30 64-65

Komplett-Katalog gegen DM 5,— in Briefmarken

Händlerkonditionen bitte schriftlich erfragen

CE-DATA Service löst Ihre Reparaturprobleme!

c't-aktuell

Krise bei Victor

Produktion künftig in Europa?

Victor Technologies, Hersteller des 'Sirius' und des 16-Bit-Portables 'Vicki', steckt in einer tiefen Finanzkrise. Ähnlich, wie vor wenigen Monaten Osborne, hat der amerikanische Computer-Hersteller ein Verfahren nach 'Chapter 11' des US-Konkursrechts beantragt, das einen vorübergehenden Schutz vor Gläubiger-Forderungen gewährt.

Victor muß jetzt innerhalb kurzer Frist einen Finanzierungsplan vorlegen, der durch eine Kommission überprüft wird. Lehnt diese den Plan ab, bedeutet das den Konkurs. Zur Zeit werden Entwicklung und Produktion, finanziert durch die Hausbank, mit einer auf 250 Mitarbeiter geschrumpften Belegschaft weitergeführt.

Die deutsche Sirius-GmbH ist von diesen Vorgängen rechtlich nicht betroffen. 'Wir sind finanziell völlig gesund und schreiben weiterhin schwarze Zahlen', ließ es aus der Frankfurter Unternehmenszentrale. Man erwägt dort, im 'Ernstfall' die gesamte Produktion nach Europa zu verlegen.

Victor hat nach Angaben aus Frankfurt auf dem deutschen Markt bisher rund 6000 'Sirius'-Computer verkauft. Die Nachfrage nach dem Ende 1983 vorgestellten 'Vicki'-Portable sei 'ungeheuer groß'. Dieser Computer war freilich bei Redaktionsschluss noch nicht lieferbar, doch erwartete man, daß die Auslieferung noch im ersten Quartal 1984 beginnen werde.

64 KByte im EPROM

Das erste EPROM mit 512 KBit Speicherkapazität ist jetzt in Musterstückzahlen verfügbar. Der Am27512 ist in 65536 Worten zu 8 Bit organisiert und hat eine Zugriffszeit von 250 ns. Er zeichnet sich durch eine geringe Leistungsaufnahme, 132 mW in Bereitschaft, 525 mW im Betrieb, aus.

Informationen:

Advanced

Micro Devices GmbH

Rosenheimer Straße 139

8000 München 80

Transistor Array mit Uce = 50 V

Toshiba stellt ein Transistorarray vor, in dem fünf einzeln isolierte NPN-Transistoren in einem 16poligen DIP-Gehäuse untergebracht sind. Die Transistoren sind für eine Uce von 35 V oder 50 V bei einem I_{Cmax} von 200 mA ausgelegt. Alle Anschlüsse der Transistoren sind einzeln herausgeführt.

Informationen:

Bitronic GmbH

Dingolinger Straße 6

8000 München 80

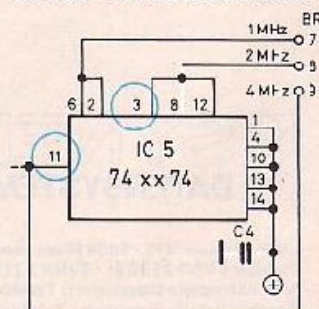
'Hacker-König' berät Hersteller

Der international bekannte 'Hacker-König' Richard Chesire macht zur Zeit als Propagandist für die 'Gegenseite' von sich reden. Chesire, der in der Bundesrepublik durch ein Spiegel-Interview bekannt wurde, tritt auf einer Fachtagung des Leuro-Seminars zum Thema 'Manipulationstechniken der Computerkriminalität und Sicherungskonzepte' auf. Er demonstriert dort live, wie man fremde Computer-Rechenzentren 'elektronisch knacken' kann. Nach Angaben der Veranstalter gibt er den versammelten Experten auch Hinweise, welche Schutzmaßnahmen von den Unternehmen getroffen werden können.

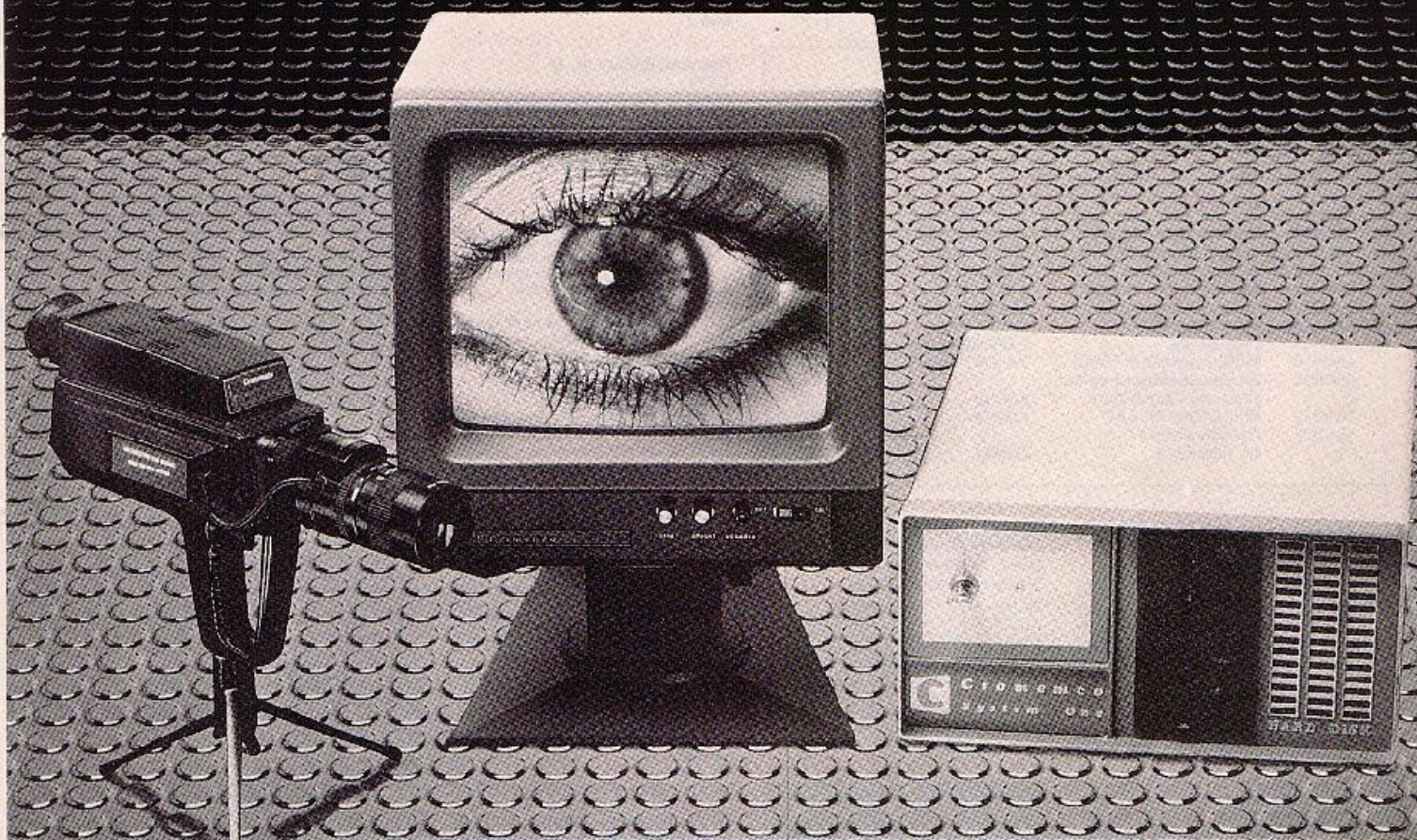
Ergänzungen + Berichtigungen

CEPAC-65 (3/84)

Im Schaltplan des c't-Projekts 'CEPAC-65' in Heft 3/84 ist uns leider ein Fehler unterlaufen. Das PlatineLayout ist jedoch korrekt. Hier also der berichtigte Teil des Schaltbildes:



»Cromemco lehrt Computer Sehen.«



Neuer Video-Digitalisierer setzt neue Maßstäbe

Die preiswerte Lösung zur aufwendiger Computergrafik heißt „Cromemco-Video-Digitalisierer“.

- Farbbilder, mit normaler Videokamera aufgenommen, lassen sich „digitalisiert“ im Rechner verarbeiten
- Bild- und Textkombinationen sind jederzeit möglich
- Archivierung und Speicherung von Bildern, z. B. für Inventuren, Bildkataloge etc.

- Durchführen strenger Sicherheitskontrollen
- Einblenden von Computergrafik in Original-Videobilder, z. B. im Training und bei Lehrgängen

Übrigens, alle anderen EDV-Anforderungen erfüllen Cromemco Systeme selbstverständlich auch. Wie können wir Ihnen helfen? Fordern Sie uns!

Cromemco®

Cromemco GmbH · Frankfurter Str. 33-35 · 6236 Eschborn 1 · Tel. 0 61 96/48 16 06

Informations Coupon

ct 4

Ich erbitte Unterlagen über Cromemco-Produkte:

☐ Mikrocomputer-Systeme ☐ Video-Digitalisierer ☐ Personal Computer C-10 ☐ S-100 Boards

Name: _____

Straße: _____

PLZ/Ort: _____

Tel.: _____

Cromemco GmbH · Frankfurter Straße 33-35 · 6236 Eschborn 1 · Telefon 06196/481606

Mittwoch, 4. - Mittwoch, 11. April
Hannover Messe '84
CeBIT Halle: 4, Stand 1903-21

Heinz Software

Superhits für VC 20 und C 64!

NEU

(Für VC 20 mit Speichererweiterung, min. 16 KByte)

Bitte bei der Bestellung Rechnerart angeben!

Programmbibliothek Nr. 10 TEXTY

Ein großes Textverarbeitungsprogramm mit folgenden Features:

- Einlesen und Speichern von Texten auf Kassette oder Diskette
- Text erfassen mit Anzeige von Zeilen- und Spaltenposition
- Neue Zeile einfügen
- Druckerausgabe
- Kopieren von Zeilen
- Suchen von Textstücken; mit der Funktion ersetzen kann der gefundene Textteil durch einen neuen, wahlweise kürzeren oder längeren Text ersetzt werden.
- Vollmenü-gesteuert

Es sind zwei verschiedene Versionen mit spezieller Druckeranpassung erhältlich:

TEXTY GP für Commodore-Drucker VC 1515, VC 1541 und Seikosha GP 80, GP 100 VC

TEXTY MX für Epson MX 80

Kassette mit Handbuch 49,— DM

Neu:

Programmbibliothek Nr. 1 ADRESSEN

Anschriften von Freunden, Verwandten, Vereinsmitgliedern, Kunden, Lieferanten werden verwaltet und in übersichtlicher Form angeordnet. In Zusammenhang mit TEXTY Adressenausdruck für Serienbriefe realisierbar.

Kassette mit Handbuch 39,— DM

Neu:

Programmbibliothek Nr. 12 KARTEIKASTEN

Dieses Programm macht alle Karteikasten überflüssig. Es erlaubt die Verwaltung beliebig großer Karteien (nur durch Speichergröße begrenzt). Möglich sind:

- Anlegen einer neuen Kartei
- Ausspeichern auf Band oder Diskette
- Einlesen bestehender Dateien von Band oder Diskette
- Sortieren nach auszuwählenden Feldern
- Druckerausgabe mit vielen Möglichkeiten

Für jede Kartei lassen sich beliebig viele List-Ausdrücke festlegen und ablesen wie die Datensätze auf Band oder Diskette speichern. Alle Funktionen werden über Menüs gesteuert.

Kassette 49,— DM

c't-Programme

Alle in den Rubriken 'c't-Programme', 'c't-Projekt' und 'Computer zu Hause' vorgestellten Programme sind auf Datenträgern erhältlich.

Zur Zeit sind lieferbar:

Nr.	Programm	Datenträger	Preis
S831241	WINIMON (Z80-Monitor)	Kassette (TRS-80)	5 DM
S831241	Terrinal-Deletersprogramm	EPROM (2732)	20 DM
S831242	Terrinal-Zeichensatz ZSO	EPROM (2732)	20 DM
S831243	Zeichensatz ZS1 (deutsch)	EPROM (2732)	20 DM
S831272	Texthausteinprogramm	Kassette (TRS-80)	5 DM
S831292	Master-Directory	5 1/4-Zoll-Floppy	15 DM
S840145	c't 88-Monitor	2 EPROMs (2732A)	39 DM
S840172	65C02-Assembler in FORTH	5 1/4-Zoll-Floppy (Apple)	15 DM
S840232	Energiekostenberechnung für Strom und Gas mit Z81	Kassette	5 DM
S840349	SPRITE-Editor für C64	Kassette	5 DM
S840324	Kfz-Kostenanalyse mit Video Genie	Kassette	5 DM

Programmbibliothek

In c't-Software-Service erhalten Sie ein Sortiment: besonders interessant und leistungsfähiger Programme für verschiedene Computersysteme, das ständig erweitert wird. Allen Programmen sind ausführliche Erläuterungen, zum Teil in Handbuchform, beigelegt.

Programmbibliothek Nr. 1

für PET 2001 (ab 8 KB), com 3001, TRS-80 Level II)

10 lehrreiche und unterhaltsame BASIC-Programme, u. a. Schnell-Lese-Training, Übung für das Fiktionsonnschreiben, Drill für das Knopfrechnen, Berechnung von Zinssätzen, der Computer als Heilseher.

Programmkassette 19,90 DM
Handbuch (56 Seiten) allein 8,30 DM

Programmbibliothek Nr. 2

für PET 2001 (ab 8 KB), com 3001, TRS-80, Level II)

10 BASIC-Programme, u. a. Drillprogramm für das Errechnen, Übung für das Geschwindigkeitsschreiben, Tilgungsplan für ein Darlehen, Reaktionszeit-Test, Gedächtnis-Training, Trainingsprogramm für die Benachteiligten, der Computer als Pest.

Programmkassette 19,90 DM

Programmbibliothek Nr. 3

RHINO

für PET 2001 (ab 8 KB), com 3001, TRS-80 Level II)

Ein spannendes Spiel für intelligente Leute. Mit vieler Variationsmöglichkeiten.

Programmkassette 19,90 DM

Programmbibliothek Nr. 4

Analogs-Uhr/Digital-Uhr

für PET 2001 (ab 4 KB) und com 3001)

Programmkassette 19,90 DM

Programmbibliothek Nr. 5

Morse-Test

für PET 2001 (ab 8 KB), com 3001)

Übungsprogramme für das Erlernen des Morse-Codes. Die akustische Ausgabe erfolgt mit Hilfe eines anschließenden Radios oder Kassettenspeichers.

Programmkassette 19,90 DM

Programmbibliothek Nr. 6

PACK/UNPACK

für PET 2001 (ab 8 KB) und com 3001)

Ein sehr nützliches Dienstprogramm zum Anlegen, Ändern/Ergänzen und Lesen von Dateien aus numerischen Daten.

Programmkassette 19,90 DM

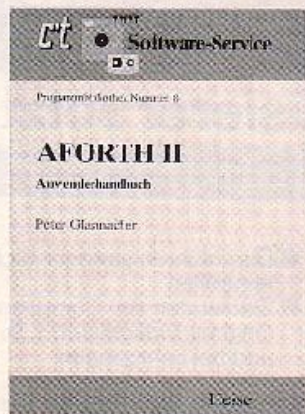
Programmbibliothek Nr. 7

Interaktive Menüplanung

(Mind. 32 KByte oder Diskette, fordern Sie unsern Spezialprospekt an.)

Geräteecken mit Computer-Hilfe. Das Programm enthält für die meisten gängigen Lebensmittel (fast 400) Informationen über Energie-, Nährstoff-, Mineralstoff- und Vitamingehalt. Es ermöglicht die Zusammenstellung von Mahlzeiten im Dialog mit dem Computer. Ideal für alle, die beruflich mit dem Erstellen von Speiseplänen zu tun haben, aber auch für den privaten Raustalt. Wahlweise Druckerausgabe.

Diskette oder Kassette 92,50 DM



Unser Bestseller:

Programmbibliothek Nr. 3

FORTH mit 65C02-Assembler

(für Apple und Apple-kompatible Computer mit Diskettenlaufwerk)

Das Programm enthält neben einem FORTH-Compiler nach dem FORTH-79-Standard einen zeilenorientierten Editor und einen Assembler für den erweiterten Befehlssatz der CMOS-CPU 65C02. Wenn das System mit einer 80-Zeilen-Karte ausgestattet ist, steht zusätzlich ein komfortabler Screen-Editor zur Verfügung.

In 64-KByte-Systemen wird FORTH in die Language-Karte geladen und belegt den Adressbereich (H) 0000...FFFF. Die Transient-Program Area (TPA) beginnt bei (H) 5C00, so daß für High-Resolution-Anwendungen noch eine Seite freibleibt. Bei anderen Systemen wird FORTH ab (H) 5000 geladen. Es steht dann mehr als 10 KByte Speicher für Anwenderprogramme zur Verfügung — wesentlich mehr als bei herkömmlichen FORTH-Systemen.

Der Compiler wird auf einer Diskette (Format: Apple Standard) geliefert, deren Rückseite das Source Listing des Assemblers und des Editors sowie nützliche Utilities wie einen FORTH-De-compiler und einen Textformattierer enthält. Es ist geplant, nach Festlegung des FORTH-83-Standards ein Anpassungsprogramm anzubieten.

Diskette mit Handbuch 98,— DM

Zwei Disketten (single sided) mit Handbuch 13,— DM

Neu:

Programmbibliothek Nr. 9 MYSTERY

(für ZX 81)

Ein Spiel, das nie langweilig wird. In Mystery-Land wimmelt es von Kobolden, Geistern und ähnlichen unangenehmen Zeitgenossen. Steuern Sie Ihre Spielfigur mit den Cursor-Steuertasten durch das Labyrinth. Der Computer wird Ihnen immer neue Aufgaben stellen, die Sie zu bestehen haben. Bei jedem Spiel ändern sich fast alle Variablen, so daß Sie immer neue Abenteuer erleben.

Kassette 19,90 DM



Neu:

Programmbibliothek Nr. 13

MICRO FORTRAN

(für TRS 80, Video Genie)

Micro Fortran ist ein Fortran-System für den TRS 80/Video Genie mit mindestens 16 K RAM und benötigt keine Diskettenstation. Da Fortran

eine sehr umfangreiche Sprache ist und der Micro Fortran schon ab 16 K RAM arbeiten soll, erhält Micro Fortran nicht alle Möglichkeiten von Fortran IV. Trotzdem versteht das System die wichtigsten Fortran-Befehle, beherrscht Realzeitanforderungen und hat einen benutzerfreundlichen Editor. Im Vergleich zu BASIC ist Fortran wesentlich schneller, strukturierte Programmierung mit Unterprogrammen ist einfacher usw. Nachteilig ist allerdings, daß das kompilierte Programm zwar sehr viel schneller ist als ein BASIC-Programm, aber dafür auch wesentlich mehr Speicherplatz verbraucht. Außerdem muß für Fortran immer der Quelltext UND das Objektprogramm im Speicher stehen.

Das gesamte Fortran-System einschließlich Editor und Laufzeitsystem benötigt knapp unter 8 K Byte, es bleibt dem Benutzer also selbst bei nur 16 Knoch genügend Platz, um einfache Programme zu schreiben. Trotzdem sind 32 K oder mehr empfehlenswert.

Das Handbuch enthält eine Einführung in den Umgang mit FORTRAN und eine ausführliche Beschreibung aller unter MICRO FORTRAN verfügbaren Befehle.

Kassette und Handbuch 7C,— DM

Neu:

Programmbibliothek Nr. 14

OTHELLO

(für Apple mit Pascal)

Das Strategiespiel Othello (3versis) in einer schnellen Pascal-Version. Drei Spielstärken sind einstellbar. Das Handbuch enthält das Listing mit sehr ausführlicher Beschreibung und ist deshalb besonders interessant für Pascal-Anfänger.

Diskette (5 1/4-Zoll) mit Handbuch 30,— DM

So können Sie bestellen:

Um unnötige Kosten zu vermeiden, liefern wir nur gegen Vorkasse. Tilgen Sie Ihrer Bestellung einen Verrechnungsscheck oder einen von Ihrer Bank quittierten Einzahlungsbeleg über die Bestellsumme zuzüglich 2 DM für Porto und Verpackung bei. Bei Bestellung aus dem Ausland muß stets eine Überweisung in DM erfolgen. Die Überweisung und Ihre Bestellung richten Sie bitte an:

c't-Versand
Verlag Heinz Heise GmbH
Bissendorfer Straße 6
3000 Hannover 61
Konto-Nr. 93 05-308,
Postcheckamt Hannover



meyer & hanelik

Microcomputer total im Regionalbereich HANNOVER

Das Einplatzsystem

Betriebssysteme: CP/M-80, CP/M-86, MS-DOS, UCSD (P)
Hauptspeicher: von 64 KB - 512 KB
Externe Speicher: 2 Diskettenlaufwerke (je 320 oder 640 KB), (integriert)
1 Platten-Laufwerk, Winchester, 5 1/4 Zoll, 10 MB (integriert) oder bis zu 3 Platten-Laufwerke je 10 MB

Das "einfache" Mehrplatzsystem

Zwei Einplatzsysteme kommunizieren unter dem Betriebssystem MP-M miteinander

Das lokale Netz (NCR DECISION NET)

Austausch von Daten u. Programmen zwischen den Benutzern von Personal-Computern (auch verschiedenen Typs)
Gemeinsamer Zugriff auf Massenspeicher und gemeinsame Nutzung leistungsstarker Peripherie
Zugriff auf zentrale Dateien von EDV-Systemen
bis zu 64 Geräte anschließbar, wobei 16 gleichzeitig kommunizieren können

Der Supermicro NCR TOWER 1632

Prozessor: 16/32-Bit-Motorola 68000 (Taktfrequenz 10 MHz)
Hauptspeicher: 512 KB bis 2 MB
Plattenspeicher: 28 MB bis 1000 MB (5,25-/8 Zoll Winchester-Platten)
Disketten: 5,25 Zoll bis zu 1 MB
Magnetbänder: Streaming oder externes Tape
Betriebssystem: UNIX™
Programmiersprachen: C, Cobol, Basic, Pascal, Fortran
Datenbank: INGRES
Anschlüsse: Multibus (IEEE-796)
Datenkommunikation: RS 232 C/V.24, ASCII TTY, BSC 2780/3780, NCR DLC, SDLC/SNA, X.21/X.25
Bildschirmarbeitsplätze: bis zu 16 (lokal und externer Verbund)

Wir sind die Hard- und Software-Spezialisten.

Unser Leistungsangebot Software:

Standardpakete (Text, Lohn- und Gehalt, FIEU, Auftragsverwaltung) umfangreiche Branchenlösungen

Datenbanken (d'Base II, Nukleus, Ingres)

Spezialanwendungen:
3270-Emulation, SNA, usw.

Sie finden uns auch auf der Hannover-Messe, NCR-Stand, CEBIT-Halle



meyer & hanelik

Tannenbergallee 4c · 3000 Hannover 1
Telefon 0511/67 10 70

Das runde Elektronik Programm

Jetzt im neuen kostenlosen Elektronik-Spezialkatalog!

kostenlosen Spezialkatalog anfordern!

Meßtechnik • Bauteile • Funk • HiFi • Modellbau • Bausätze • Industrie-Restposten • Schlagerangebote • Das alles und mehr im 28 000-Artikel-Programm von Europas größtem Elektronik-Spezialversender.

CONRAD ELECTRONIC

Fach 698
D-6452 Hirschau

Tel. 02262-4815

Endlich! Praktische EDV-Programme für den Kleinbetrieb

Angebote, Rechnungen etc. nach Textbausteinen mit Preis, Lohn, Stück, Auf- o. Abschlag einf. Vor- und Nachkalkulation Finanzbuchhaltung Lohnbuchhaltung (I.V.) Brieftext m. Adresswesen Wärmebedarf n. neuer Norm (lauffähig auf IBM, KAYPRO, GEN E u. ä.)

ab DM 980,-
DM 850,-
DM 1660,-
DM 1050,-
DM 475,-
DM 680,-

MC - SERVICE
5276 BIELSTEIN

Computersysteme

Monika Glitsch, Telefon 0 71 52 / 4 76 43

Arbeitsplatzcomputer Software
Home-Computer Literatur
Drucker Zubehör
Speichererweiterungen und und und

Hochleistungs-GROSSBILDPROJEKTOREN portable...

Ludwig-Finckh-Weg 31/1 7250 LEONBERG 1

Mikrocomputerkurs im Schulfernsehen (Nord III)

Mit Z80 und 68008, Vollgrafik und Roboter

Der NDR-Klein Computer

Broschüre „NDR“ anfordern!

Mit Begleitbuch von R. D. Klein

ELEKTRONIKLADEN · 4930 Detmold 18
Eggestraße 70 · Tel. 0 52 32/8171

TRS-80 / Genie / Colour-Genie

SEDIT DM 39,- komfortabler Bildschirmditor, volle Cursorsteuerung + Bildschirmkontrolle einfaches Einfügen - Löschen v. Zeichen	COPBIT DM 39,- Kopierprogramm. Kopiert praktisch alle Colour-Genie-Kassetten. Auch geschützte Formate, z. B.: Preloader/Baudrateumsch.
COLBAS DM 59,- Umfangreiche Basicerweiterung, NEUGE, Disk-Basic-Funktionen, RESTORE, Variablenuche, Listingformater, Packer usw. auch kombiniert mit SEDIT für nur DM 89,-!	SBUG / CBUG Monitor, Disassembler und Trace. Komfortabler Monitor mit Disassembler und progr. Trace mit Single Step durch RAM, ROM und DOS. SBUG für TRS-80/Genie DM 139,- / CBUG für Colour DM 99,- CBUG liest auch TRS-80- und Genie-Kassetten!

Ähnlich für TRS-80/Genie mit Renumer!

Alle Programme mit ausführlicher, deutscher Anleitung. Preise inkl. 14% MwSt. SBUG/SEDIT auf Disk + 10,-. Fordern Sie Info und Hardwarepreislis an: Händleranfragen erwünscht!

Ludger Röckrath Nopplustr. 10, 5100 Aachen, Telefon: (02 41) 3 40 82

RATEV ELEKTRONIK-VERTRIEBS GMBH
Postfach 16 01, 4030 Ratingen 1, Tel. 0 21 02 / 29902



Hochwertige Cherry-Tastatur, wie Titelbild mit Ausgabe 1 mit 8 Bit paralleler Datenausgang, die an jedes System oder jeden Selbstbaucomputer angepaßt werden kann, da die ursprüngliche Version mit einem EPROM für alle Anwendungen veränderbar ist. Eine ausführliche Umbauanleitung liegt vor.

Preis inkl. Gehäuse **DM 390,-**
dto., jedoch für APPLE bereits umgerüstet **DM 449,50**

MPF-1 **A/D-WANDLER-KARTE** **ECB-Bus**
16 Kanäle 8 Bit

Vielseitig verwendbar zum direkten Anschluß an das Lehrsystem Microprofessor oder Systeme mit ECB-Bus

Best.-Nr. 4601	Bausatz MPF-1	DM 170,-
Best.-Nr. 4502	Bausatz ECB-Bus	DM 170,-
Best.-Nr. 4601	fertige Karte MPF-1	DM 220,-
Best.-Nr. 4602	fertige Karte ECB-Bus	DM 220,-
Best.-Nr. 4701	Flachbandkabel	DM 30,-

MPF-1 Kalenderuhr für Anzeige des Datums und der Uhrzeit Ausgabeverstärker für den PIO-Baustein m. LED-Monitor

Best.-Nr. 4801	Uhr RTC-1 fertig (aufgebaut und getestet)	DM 80,-
Best.-Nr. 4801 A	RTC-1 mit Accus	DM 95,-
Best.-Nr. 4802	Ausgabeverstärker	DM 65,-
Best.-Nr. 4803	EPROM mit Uhrenprogramm	DM 20,-

Bitte Prospekt anfordern
Preis inkl. MwSt.,
Versand per NH.

CHR. BEHN Jakob-Kneip-Straße 107, 4000 Düsseldorf 13
Telefon: 02 11 / 70 70 98

PROFESSIONELLES DIRECT-DRIVE-LAUFWERK IN 19" 28BE / 3HE CASSETTE



- Geeignet für Compact-Cassetten
- Direkt-Drive, kein Riemen
- Verschleißfreier Antrieb
- Hohe Zuverlässigkeit
- Minimale Drift- u. Flutterwerte
- Fernsteuerbar, TTL kompatibel
- Digitalaufzeichnung bis 9,6 kbd
- Analogaufzeichnung (Hi-Fi)
- Suchlauf (Que-Review)

OPTION: 19" Cassette (Schnaff) — Leistungsteil — A-W Verstärker f. Datenaufz. — Sonderausführungen

MAGNETRONIC GmbH
Kreuzhofstr. 10, D-8000 München 71, Tel. 0 89 / 7 55 36 70

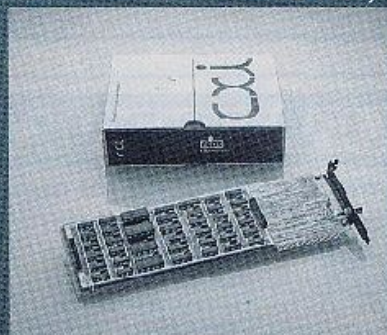
COMPU-SHACK

APPLE CP/M 3.00 Plus

6 MHz z-80 läßt Ihre Programme 2-3 mal schneller laufen

Mit CBASIC, GSX-80 (Graphic Software), CP/M 3.0 + Documentation, Platine mit zusätzlich 64k Ram; Besonderheiten: Bankselection volle 128 k, Passwortschutz; sucht Programme bis auf 4 Laufwerke; verwaltet bis zu 16 Laufwerke mit je 512 Mbyte; kein BDOS error mehr; vermerkt Datum + Uhrzeit letzten Zugriffs. **DM 1 099,-**

Programmer-Kit	DM 220,-
Eprom Programmierkarte 24 Pin	DM 299,-
Z Card z-80 mit CF/M-80 Software (2.2)	DM 375,-
80-Zeichen-Karte	DM 299,-
V24 (rs-232)	255,-
Centronics Printer Interface	DM 199,-
6 MHz z-80 b APGLI Card 64 k	
CP/M 2.2 Software	DM 899,-
APPLI Card 64 k + 128 k + CP/M 2.2 Software	DM 1499,-



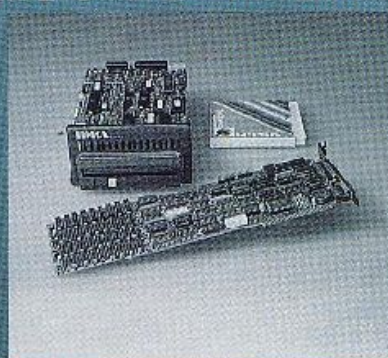
Seriell Interface	DM 240,-
80 x Zeichen für den Apple IIe + 64k Ram	DM 399,-
Diskettenlaufwerk für den APPLE	
Slim Line Laufwerk	DM 599,-
100% Apple-kompatibel, liest auch Halb-Tracks	
Vista Controller für Apple 8" Laufwerke	DM 1295,-
Teac-Laufwerk Typ FD 55A für Apple	DM 699,-
Pinball für Apple	DM 99,-

IBM Keyboard für den Apple	
11 Funktionstasten Nummern Tastatur	DM 775,-
Keyboard incl. 10er Tastatur	DM 399,-

Hercules Graphic Platine	
720 h x 384 v auf Monoch-Bildschirm	DM 1699,-

Weitere Platinen wie zum Beispiel
68000 Chip + Unix Software für den IBM/
E/irer NetLink/ 8" Controller für den IBM/
1024x1024 color Graphic Carte/Speech
Master/Voice Recognition/IEEE 488 In-
terf./ Video Platine/Mouse System

Mouse von Mouse System für den IBM	DM 1195,-
---	-----------



Big Blue. Läßt sich auf Ihrem IBM 8 Bit (CP/M) Software lesen und laufen. Besitzt außerdem Calendar, Clock, Eatory, Harddiskadapter, Parallel, Serial Port, 64k Ram. **DM 1995,-**
Baby Blue CP/M 2.2 mit Z 80 B und Bitam-Software **DM 1895,-**
8xRS232 (V24) für den IBM-PC-XT **DM 3299,-**

Accelerator mit 8088 mit 10 MHz, 128 KB aufrüstbar auf 576 KB **DM 2995,-**

Software für den IBM-PC	
Lotus 1 2 3	DM 1695,-
Access-Manager	DM 1095,-
Pascal MT + 86	DM 1095,-

Pinball von SoubLogic für IBM **DM 149,-**

Maxel-Diskette, 10er-Pack	DM 89,-
Microsoft Flugsimulator	DM 199,-

IBM 3740 Format mit 5 1/2" + 8"-Controller sowie einem 8"-Laufwerk mit 1,6 MB **DM 6995,-**

10-MB-Tape Backup für XT oder Winchester-Laufwerk **DM 7995,-**



Winchester Laufwerk für den IBM Intern (im Gehäuse)	
12.5 MB inc. Cont. Adapt.	DM 6995,-
12.5 MB 2. Laufwerk	DM 3999,-
5 MByte Wechsellaufwerk intern für den PC	DM 7995,-
Festwechsellaufwerke für IBM, Apple, removable Harddisc von DMA	
12 MB + 5 MB	DM 13999,-
20 MB + 5 MB	DM 14999,-
26 MB + 5 MB	DM 16999,-
53 MB + 5 MB	DM 22999,-

IBM Erweiterungsplatinen

Von A.S.T. MegaPlus 64 k bis 512 k
S = Serial, P = Parallel, C = Clock, G =
Game, H = Harddiskadapter, SP =
Spooler, Software, DI = Disk-Spooler

64 k, C, S, SP, DI	DM 1190,-
64 k C, 2x S, SP, DI	DM 1399,-
64 k C, S, P, SP, DI	DM 1398,-
64 k C, S, 2x P, SP, DI	DM 1550,-
Jede weiteren	
64 k bis 256 k	DM 220,-
MegaPack 256 k bis 512 k	DM 990,-

Sixpack für den IBM XT, ausbaufähig bis 384k, S, P, C, G, DI, SP

Preis auf Anfrage

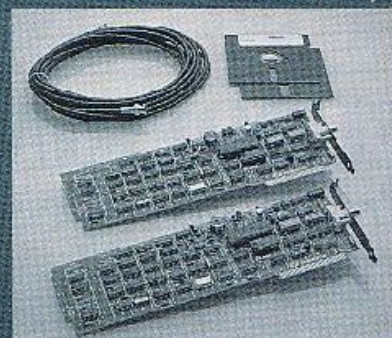
I/O Plus II

C, S, SP, DI	DM 499,-
jede weitere S, P oder G	DM 199,-
maximale Ausbaustufe C, 2x S, 1x G, 1x P, SP, DI	DM 1096,-

Drei-Com-Ethernet Board mit 10 SB Transfer pro Sekunde **DM 3995,-**

Drei-Com-EtherPrint Software **DM 1790,-**

PCnet Starter Kit, bestehend aus 2 Platinen, für 2 IBM PC oder XT, inc. Coaxial Cable, Software, Übertragungsrate 1 Mbps **DM 4990,-**
Jede weitere Platine **DM 2590,-**



Weitere Platinen von diversen Herstellern
512k Platine mit
64k bestückt **DM 799,-**
jede weiteren 64k **DM 220,-**
Combo Carte, P, S, C, G **DM 599,-**
Eprom Programmierkarte **DM 455,-**
4k-64k bit 24 Pin

IRMA, Ihr Anschluß an den
IBM 3270 Controller **DM 3995,-**
Perayal mit 3270-3780 sowie HASP
Übertragungsprotokolle mit Koaxial-
3278-Terminal **Preise auf Anfrage**
Picox 3278-Terminal mit Koaxial-An-
schluß sowie up- und download-Soft-
ware **DM 3995,-**

Weitere A.S.T. Platinen **DM 5690,-**
Adv. Communication Card **DM 799,-**
Bisync Emulation **DM 2790,-**
SNA/SDLC Emulation **DM 2790,-**

Tecmar Graphics Master
mit 640 h x 480 v. 16 Farben **DM 1999,-**
Graphic Master jetzt Lotus 1.2.3-fähig

Alle Preise für Industrie, Gewerbe und Handel.
Endverbraucher senden die Preisliste einschließ-
lich MwSt. anfordern.

Händleranfragen erwünscht
Alle Preise zzgl. der gesetzlichen MwSt.

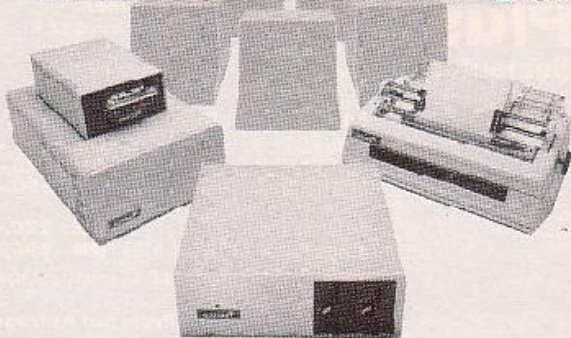
Ringstr. 56 5450 Neuwied Tel. (026 31) 290 31
Telex 867 716 celgk

COMPU-SHACK

Syscom



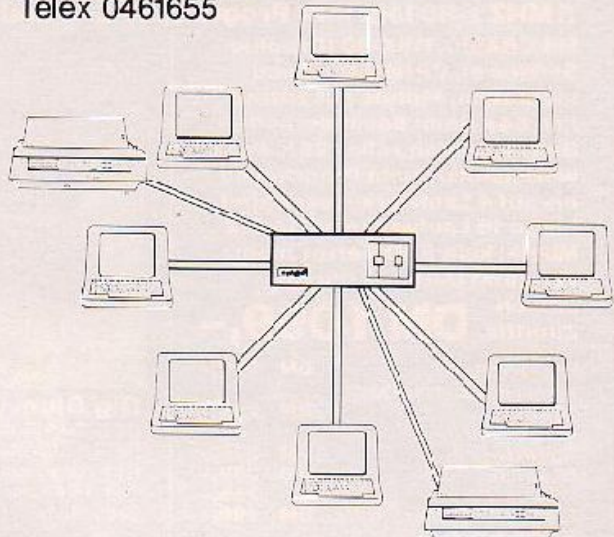
NORTH STAR HORIZON 8/16 MIT TURBO-DOS



Einer der zuverlässigsten Mikro-Computer für Schule, Büro, Wissenschaft, jetzt noch bestechender in Leistung und Vielseitigkeit durch:

- Multiprocessing — bis zu acht Benutzer, jeder mit eigener Zentraleinheit
- Northstar-Betriebssystem Turbo-DOS
- 8-Bit- und 16-Bit-Zentraleinheiten
- Große Speicherkapazität — bis zu 512K-RAM pro 16-Bit-Zentraleinheit

Winkler GmbH
Trübnerstraße 40
6900 Heidelberg 1
Tel. (06221) 4 9181
Telex 0461655



Das HORIZON 8/16 Multi-Processor-System unterstützt bis zu 8 Benutzer.

EUROPRINT *drückt den Preis*

- **MADE IN GERMANY (VDE + FTZ-Prüfzeichen)**
- Bidirektionaler Druck mit Druckwegoptimierung
- Druckgeschwindigkeit von 100 Zeichen/Sek.
- Echte Matrix mit 9x7 Punkten Druckraster daher eine hohe Auflösung von 100 Punkt/Zoll
- 9 Nadeln pro Kopf mit versetzter Anordnung zur Erhöhung der Druckdichte
- Einzelnadelsteuerung bei einer Balkenbreite von 6 Punkten vertikal hochauflösende Graphik
- 3 verschiedene Zeichenbreiten 80/100/120 Zeichen/Zeile darstellbar in doppelter Zeichenbreite (Breitschrift)
- Kursivschrift
- Unterstreichmodus
- Vollständiger Horizontal- und Vertikaltabulator beide vorwärts und rückwärts programmierbar
- Programmierbare Formularlänge
- Zeichensatz: 96 ASC II Zeichen mit Unterlängen, Standard deutscher Zeichensatz. Softwaremäßig umschaltbar auf US-Zeichensatz
- Selbsttesteinrichtung
- Ausführliche, deutsche Bedienungsanleitung

Presse
News
K6311 FT
DM 999,-

*incl. Centronics-Parallel-Interface und Formulareinzug
unverbindlicher, empfohlener Verkaufspreis
incl. MWSt., ab Lager Düsseldorf



- **Zubehör:** Paralleles Interface, Centronics (Standardausführung)
- Serielles Interface RS 232, V 24
- Rollenhalterung
- Formulareinzug

UNITRONIC®

Elektronische Bauelemente und Geräte



UNITRONIC GMBH
Münsterstraße 338
4000 Düsseldorf 30
Postfach 330 429
Telefon 0211 / *62 63 64
Telex 858 6434
Hardwarestudio
Münsterstraße 375

UNITRONIC HAMBURG GMBH & CO KG
Lindholzstraße 3
2360 Bad Segeberg
Telefon 04551 / 86 97 + 86 98
Telex 261 616

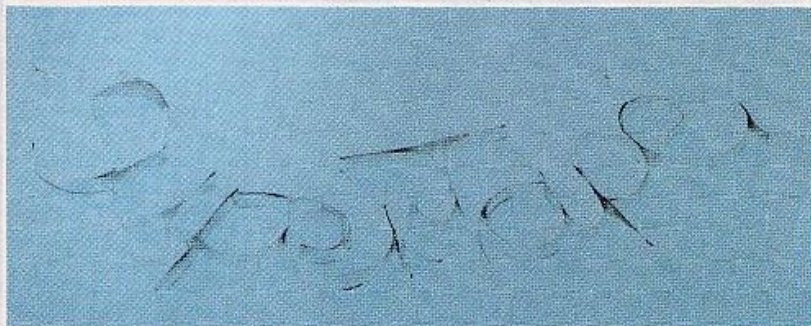
UNITRONIC VERTRIEBS GMBH
Marskestraße 29
3160 Lehrte
Telefon 05132 / 5 30 01
Telex 922 084

Händleranfragen erwünscht!



**HANNOVER
MESSE '84**
Halle 1 / CeBit
Stand 7701 / 7801
ROBOTRON

Für alle Mikrocomputer:



Kassettenaufzeichnung mit 3600/7200 Baud

Andreas Stiller/Christian Persson

Dem Austausch von Daten und Programmen zwischen Computern verschiedener Fabrikate haben die Hersteller viele Hindernisse in den Weg gestellt. Eine der größten Hürden besteht in der Wahl unterschiedlicher Verfahren und Formate für die Aufzeichnung auf Datenträgern. c't hat sich die Aufgabe gestellt, dieses Problem zu lösen. Wir stellen ein schnelles und sicheres Verfahren für die Datenaufzeichnung auf Magnetband-Kassetten vor, das sich bei praktisch allen Mikrocomputern anwenden läßt. SuperTape arbeitet wahlweise mit 3600 oder 7200 Baud und ist damit schneller als alle Verfahren, die bisher von den Herstellern populärer Homecomputer verwendet werden. Zusätzliche Hardware ist nur im Ausnahmefall erforderlich.

c't stellt in diesem Monat die SuperTape-Implementation auf Z80-Rechnern am Beispiel des ZX81 vor. Lösungen für den ZX Spectrum, für TRS80, für VC-20 sowie C 64 befinden sich im Erprobungsstadium. Weiter werden folgen, wobei die c't-Redaktion auch auf Unterstützung aus dem Leserkreis rechnet: Wer SuperTape auf einem Mikrocomputer anderen Fabrikats implementiert, ist eingeladen, seine Lösung (selbstverständlich gegen Honorar) in c't vorzustellen.

Übrigens eignet sich SuperTape, wie prinzipiell jedes Kassetten-Aufzeichnungsverfahren, auch für die direkte Datenübertragung von Rechner zu Rechner. Man benötigt dazu lediglich ein Überspielkabel mit 'gekreuzten' Send- und Empfangsleitungen. SuperTape dürfte also auch für die Besitzer von Floppy-Disk-Systemen interessant sein, die unter dem Format-Wirrwarr der Hersteller nicht minder leiden.

Ursprünglich wurde das SuperTape-Verfahren für den COBOLD-Computer unserer Schwesterzeitschrift elrad entwickelt. Schon bald stellte sich heraus, daß dieses Verfahren (bei der zunächst gewählten

Übertragungsgeschwindigkeit von 4800 Baud) außerordentlich zuverlässig arbeitet. Der COBOLD bietet die komfortable Möglichkeit, auf einen Timer im RIOT-Baustein zurückzugreifen, wodurch sich die Programmierung der SuperTape-Routinen recht einfach gestaltet. Als sich aber zeigte, daß diese auch mit einem Z80-Prozessor ohne Timer zu verwirklichen sind, lag die Weiterentwicklung zu einem universellen Standard auf der Hand. Besonders eindrucksvoll ist dabei der minimale Hardware-Aufwand, der sich bei vorhandenen Portleitungen auf nur einen Schmitt-Trigger beschränkt. Dieser ist bei allen Rechnern schon vorhanden, die mit zum Teil sehr aufwendigen — aber weniger leistungsfähigen — Kassetteninterfaces ausgestattet sind.

Eine wichtige Voraussetzung für SuperTape besteht darin, daß sämtliche Interrupts abschaltbar sein müssen. In den meisten Fällen ist diese Voraussetzung erfüllt, denn auch die anderen Kassettenverfahren müssen 'Interrupt-frei' arbeiten.

Auch eine andere Eigenschaft hat SuperTape mit den herkömmlichen Verfahren ge-

meinsam: Es ist empfindlich gegen Drop-outs. Darum sollten nur gute Bänder Verwendung finden. Außerdem darf natürlich der Kassettenrecorder kein übermäßiges Brummen oder Spikes vom Antriebsmotor einkoppeln.

Um dem Anspruch — SuperTape für jeden Rechner — gerechtzuwerden, durfte die Standard-Baudrate nicht zu

hoch gewählt werden. Zudem sollen ja auch preiswerte Rekorder verwendet werden können. Bei der Standard-Übertragungsrate (3600 Baud) ist dies gewährleistet, denn die höchste zu übertragende Frequenz entspricht der Baudrate! Wer einen etwas besseren Recorder und einen nicht allzu 'lahmen' Mikrocomputer besitzt, kann praktisch ohne Einbußen an Übertragungssicherheit auf 7200 Baud umschalten. Allerdings dürfte sich diese Baudrate für den Software-Tausch nicht so gut eignen.

Die Technik

SuperTape arbeitet, wie die meisten Magnetband-Aufzeichnungsverfahren, mit Signalen verschiedener Frequenz. Eine logische 0 wird mit einer Schwingungsperiode der höheren, eine logische 1 mit einer halben Schwingungsperiode der niedrigeren Frequenz übertragen. Daraus resultiert eine Baudrate (Bit/sec), die gleich der höchsten Signalfrequenz ist. Bild 1 zeigt, wie das ausgesendete Bitmuster des Bytes D4h aussieht.

Beim Lesen von der Kassette (Bild 2) wird ein sehr einfaches Prinzip angewendet: Die Leserroutine ermittelt den Beginn der Übertragung eines Bits (t_1) und testet über einen entsprechenden Port das Ausgangssignal am Schmitt-Trigger zum Zeitpunkt t_2 . Dann vergleicht sie den zum Zeitpunkt t_2 ermittelten logischen Pegel mit dem

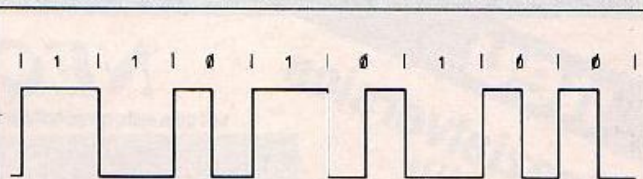


Bild 1. Sendesignal für das Bitmuster D4h

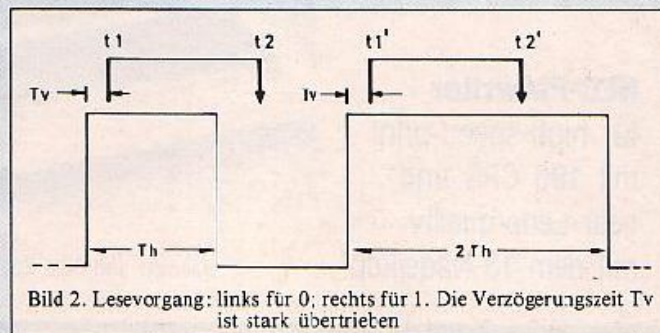
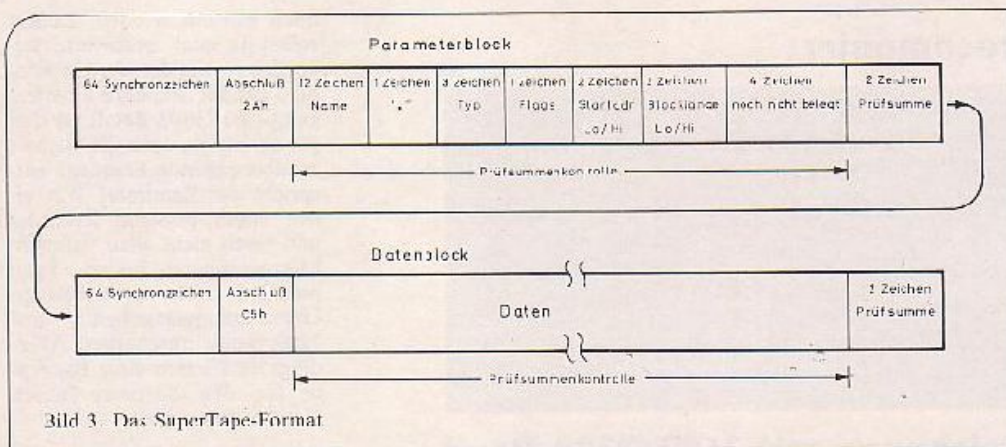


Bild 2. Lesevorgang: links für 0; rechts für 1. Die Verzögerungszeit T_v ist stark übertrieben



gespeicherten vom Zeitpunkt t_1 . Sind beide gleich, so entspricht das einer 1; sind sie verschieden, erkennt sie eine 0.

Der optimale Zeitpunkt für t_2 ergibt sich aus der Halbperiodendauer (T_h) der höheren Schwingung und der mittleren Verzögerung (T_v) bei der Erkennung der Flanke zu

$$t_2 - t_1 = 4/3 \cdot T_h - T_v$$

Er liegt also nicht in der Mitte der zweiten Periodenhälfte, sondern ist ein wenig zu kürzeren Zeiten hin verschoben. Das hängt damit zusammen, daß Geschwindigkeitsschwankungen des Kassettenscanners relative Abweichungen (in Prozent) sind. Deshalb ist der Prüfzeitraum $t_2 - t_1$ so gewählt, daß sich im Idealfall ($T_v = 0$) eine Geschwindigkeitstoleranz von $\pm 33\%$ ergibt. Falls Ihr Kassettendeck tatsächlich Abweichungen in die-

ser Größenordnung aufweist, sollten Sie vielleicht an die Anschaffung eines neuen Gerätes denken. Der Praktiker kann den optimalen Prüfzeitraum natürlich auch experimentell ermitteln, indem er die Über- und Untergrenzen für korrektes Lesen feststellt und dazwischen entsprechend mittelt.

Gegenüber Störimpulsen, die dem Signal vom Band überlagert sind und zu Mehrfach-Triggerungen führen, ist das Verfahren unempfindlich. Es kommt lediglich darauf an, daß zum Zeitpunkt t_2 ein definierter Pegel vorliegt.

Das Format

Die gesamte Datenübertragung findet bei SuperTape in zwei Blöcken statt (Bild 3). Zuerst wird ein Parameterblock gesendet, gefolgt von dem Datenblock. Der Parameterblock

wird grundsätzlich mit 3600 Baud übertragen. Er beginnt mit 64 Synchronzeichen 16h und der Synchron-Abschlußmarke 2Ah. Dann folgen der Name (maximal 12 Zeichen), ein Punkt und drei Zeichen zur Kennzeichnung des Typs der übertragenen Daten (beispielsweise 'BAS' für ein BASIC-Programm).

Üblicherweise sind diese Zeichen ASCII codiert. Das wird in dem anschließend übertragenen Flagregister durch eine Null in Flag 6 gekennzeichnet. Flag 7 gibt die Baudrate an, mit der der Datenblock übertragen wird: eine 1 steht für 7200 Baud. Die übrigen fünf Bits sind noch nicht belegt.

Es folgen die Startadresse und die Blocklänge des Datenblocks. Wie üblich, kommt auch hier das niederwertige Byte vor dem höherwertigen.

Um für die Zukunft noch einige Erweiterungsmöglichkeiten offenzuhalten (z.B. eine Autostart-Adresse), werden danach vier Bytes, sozusagen als Platzhalter, gesendet. Zum Schluß folgen die beiden Bytes der Prüfsumme — ebenfalls Low vor High —, die bei dem Sendevorgang berechnet werden. Die Prüfsumme stellt die 16-Bit-Summe sämtlicher übertragenen Einsen dar. Die Berechnung beginnt hinter der Synchronabschlußmarke und endet vor den Prüfsummen-Bytes.

Der Datenblock beginnt gleichfalls mit den 64 Synchronzeichen 16h, die aber diesmal die Abschlußmarke C5h haben. Dann folgen die auszugebenden Daten und zum Schluß die Prüfsumme über den Datenbereich.

Beim Einlesen sind einige Optionen vorgesehen. Der gesuchte Name wird mit dem gelesenen verglichen. Ein '?' im Suchnamen bedeutet, daß das an gleicher Stelle im Lesennamen stehende Zeichen ignoriert wird. Ein '*' steht stellvertretend für den ganzen Rest des Namens.

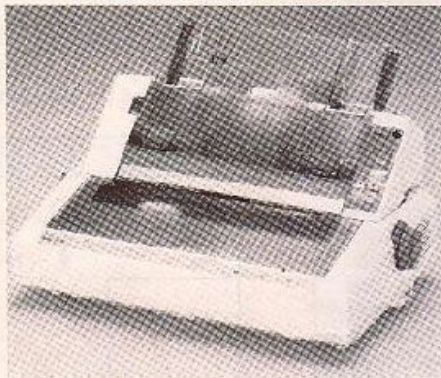
Auch der Typ kann mit '?' oder '*' maskiert werden. Deshalb dürfen diese Zeichen genauso wie der '.' nicht Bestandteil von Name/Typ sein. SuperTape bietet auch die Möglichkeit, die gelesene Startadresse zu überschreiben, so daß der gesamte Datenblock in einen anderen Speicherbereich geladen wird.

neu
Spezialversion
für
IBM PC's

NEC-Pinwriter
für high-speed-print
mit 180 CPS und
near-Letterquality
mit dem 18-Nadelkopf.

NEC

mit dem außergewöhnlichen Komfort



NEC-Typenkorb-Drucker
Spinwriter

mit 20, 35 und 55 CPS
für Briefqualität

- Ein-/Doppelschachtführung
- Einzelschachtführung
- Umschlagnadler
- Frontzuführung
- Schiebetraktor
- Bi-direktionaler Traktor

Händler- und GEM-Konditionen auf Anfrage.

NEC-HAUPTHÄNDLER:

SYSDAT GmbH
Brabantenstr. 38-40
5000 Köln 1
Tel. 02 21/23 54 44
Telex 8 882 606

DR. DOHRENBURG
Bayreuther Str. 3
1000 Berlin 30
Tel. 030/2 13 80 43-45
Telex 01 64 860

SEN COMPUTER-PERIPHERIE GERÄTE GmbH
Maybachstr. 17
6450 Hanau 7
Tel. 061 81/620 57
Telex 4 15 506

SCHWIND
DATENTECHNIK GmbH
Maria-Eich-Str. 45
8000 München 60
Tel. 089/8 34 97 15
Telex 5 213 097

SuperTape mit dem Z80

am Beispiel des ZX81/16K

Andreas Stiller

Auch mit dem einfachsten aller Z80-Rechner kann das SuperTape-Verfahren ohne jeglichen Hardwarezusatz verwirklicht werden. Die oftmals belächelte 'Schachtel' bewältigt dabei nicht nur problemlos die Übertragungsrate von 3600 Baud, sondern kann mit geeigneten Recordern bis zu 7200 Baud erreichen. Außerdem können die Programme und Daten des ZX81 problemlos von seinem größeren Bruder, dem Spectrum, gelesen werden, wenn auch dieser SuperTape-fähig ist. Am Ende dieses Beitrags schließen sich Hinweise für die SuperTape-Implementation auf anderen Z80-Rechnern an.

Da das gesamte Programm etwas zu umfangreich für eine Hefenummer ist (wer will schon 1000 Bytes auf einmal eintippen?), wurde es in zwei Teile zerlegt. Der erste Teil stellt die reinen SuperTape-Routinen vor, der zweite behandelt die komfortable Ein-/Ausgabe und die Kaltstartmöglichkeit.

Für die Standardrate von 3600 Baud eignet sich jeder 'Wald- und Wiesenrekorder', allerdings sollten Sie ein gutes Bandmaterial verwenden. Die Ansteuerung des Rekorders erfolgt wie gewohnt über die MIC- beziehungsweise EAR-Buchse. Das gesamte SuperTape-Programm ist vollständig in Maschinensprache geschrieben und beansprucht rund 1K. Damit verlieren Sie zwar etwas Speicherplatz, aber wenn Ihre Programme ungefähr zehnmal schneller als bisher zu retten und zu laden sind, werden Sie dieses Speicheropfer sicherlich gerne bringen. Hinzu kommt, daß Sie sicherlich viel öfter Zwischenspeicherungen bei der Programmentwicklung vornehmen werden, weil die nervtötenden Wartezeiten auf ein erträgliches Maß reduziert sind. Jeder, der sich schon mal nach einem Systemcrash die Haare gerauft hat, weil die letzte Zwischenspeicherung schon einige Arbeitsstunden zurücklag, wird das zu schätzen wissen.

Um das SuperTape Verfahren mit einem Z80-Prozessor ohne Timer zu verwirklichen, muß man die genaue Taktfrequenz des Rechners kennen und die Laufzeit aller Befehle berücksichtigen. Wieviel Takte jeder Befehl braucht, läßt sich dem technischen Manual des Herstellers Zilog oder der Sekundärliteratur (zum Beispiel Rodney Zaks: Programmierung des Z80) entnehmen. Wichtig ist dabei, daß die verwendeten Speicher nicht noch zusätzliche WAIT-Zyklen einstreuen, wodurch das Timing ziemlich durcheinander gerät. Glücklicherweise trifft das beim ZX81 mit handelsüblichen Speichern nicht zu. Auch der NMI-Generator läßt sich abschalten, ist also ganz im Gegensatz zu seinem Namen doch 'maskierbar'. Als Befehl dazu dient:

```
OUT 0FDH,A
```

Dieser Befehl taucht also auch in der im Programm verwendeten ROM-Routine 02E7H auf, die in eine spezielle Art des FAST-Mode, den wir hier als 'kurzfristigen FAST-Mode' bezeichnen, umschaltet.

Die Taktfrequenz des ZX81 liegt bei rund 3,25 MHz; sie ist leider nicht quartzstabilisiert und kann demnach auch etwas davon abweichen. Beim ZX des Autors ergab sich eine gemessene Taktfrequenz von 3,223 MHz. SuperTape ist je-

doch gegenüber Zeitabweichungen so großzügig, daß sich hieraus noch keinerlei Probleme für die sichere Datenübertragung ergeben. Im Testbetrieb zeigte sich, daß sogar Baudraten von 9600 Baud erreichbar sind, allerdings auf einem Spulengerät mit höherer Bandgeschwindigkeit. Üblicherweise sollte man sich ruhig auf eine Baudrate von 3600 beschränken, damit die Aufzeichnungen auch von langsameren SuperTape-fähigen Rechnern gelesen werden können.

Aller Anfang ...

Da die meisten ZX-Benutzer wohl keinen Assembler zur Verfügung haben, muß das Programm Byte für Byte mit einer kleinen Hilfsroutine eingegeben werden. Das schöne Assembler-Listing der SuperTape-Routinen ist leider auch nicht mit dem ZX erstellt worden, sondern mit einem CP/M-Rechner. Wenn auch dieser erst einmal SuperTape-fähig ist, könnte man assemblierte Programme direkt auf den Sinclair überspielen.

Als Speicherbereich ist für SuperTape der Platz ab 7C00H (31744) vorgesehen. Dieser Bereich wird vor der Eingabe und vor jedem LOAD mit

```
POKE 16389,124  
NEW
```

geschützt. Als dann können Sie folgendes Hexbyte-Eingabeprogramm eintippen:

```
1000 LET P=16538  
1100 FAST  
1200 LET P=P+6  
1300 LET X=PEEK P  
1400 IF X=227 THEN STOP  
1500 IF X=118 THEN  
    GOTO 1200  
1550 IF X=20 THEN  
    GOTO 2000  
1600 POKE R,16*X +  
    PEEK (P+1) -476  
1700 LET P=P+2  
1800 LET R=R+1  
1900 GOTO 1300  
2000 LET R=4096*PEEK  
    (P+1)+256*PEEK(P+2)+  
    16*PEEK(P-3)+  
    PEEK(P+4)-12232  
2100 LET P=P+5  
2200 GOTO 1300
```

Mit diesem kleinen Programm können Sie schon relativ komfortabel in Maschinensprache programmieren: der Hexcode wird in REM-Statements beliebiger Länge gespeichert und kann somit auch relativ einfach

mit den üblichen Editorbefehlen korrigiert werden. Das Programm erkennt das Ende eines REM-Befehls und springt dann zum ersten Hexcode des nächsten REMs — bis irgendwann ein STOP (Code 227) auftaucht. Falls als erstes Zeichen im REM-Statement ein '=' auftritt, interpretiert der 'Mikro-Assembler' die folgenden vier Bytes als hexadezimale Origin-Anweisung, also als Startadresse für den folgenden Hexcode.

Denken Sie daran, daß im Hexcode kein 'H' mehr zur Kennzeichnung der Zahlenbasis auftreten darf!

Das SuperTape-Programm beginnt also entsprechend dem Assembler-Listing (Hexcode in der zweiten Spalte) mit:

```
28 REM = 7DA8  
29 REM 21DF*F  
30 REM 011900  
31 REM 3E2A
```

und endet mit: REM STOP.

Verwenden Sie wirklich den 'STOP'-Befehl, also Shift A (und nicht etwa ausgeschriebenen). In eine Zeile können Sie natürlich auch mehrere Maschinensprachenbefehle unterbringen, doch dürfte die dem Listing folgende Aufteilung am übersichtlichsten bleiben. Wenn Sie alle Befehle eingetippt und kontrolliert haben, sollten Sie das Programm — vorerst noch im Sinclair-Verfahren — auf Band retten. Mit

```
GOTO 1000
```

beginnt dann der Mikroassembler seine Arbeit und erzeugt den Maschinencode.

Die Puffer

SuperTape benötigt zwei Pufferbereiche:

den SV-Puffer
von 7FCEh bis 7FE6h,

in den die gewünschten Werte für Save und Load geschrieben werden,

den LD-Puffer
von 7FE7h bis 7FFFh,

für die vom Band gelesenen Werte.

Die Pufferbelegung:

SV	LD	Name
7FCEh	7FE7h	Punkt
7FDAh	7FF3h	Typ
7FDBh	7FF4h	Flags
7FDEh	7FF7h	Startadr
7FDFh	7FF8h	

7FE1h	7FFAh	Länge
7FE3h	7FFCh	frei
7FE6h	7FFFh	Ende

Die Puffer sind gemäß dem SuperTape-Standard jeweils 25 Bytes lang. Zusätzlich zu den Puffern braucht der ZX81 noch zwei Register:

Flag 0: gesetzt für Save
 Flag 1: gesetzt Verify
 Flag 2: gesetzt neue Startacr.
 Flag 6: gesetzt nicht ASCII
 Flag 7: gesetzt 7200 Bd für Daten

7FCDh (32717) Baudratenregister

muß 00h für den Parameterblock
sein und wird für 7200 bd dann
80h

Das ZFlag — und das Baudratenregister, sowie die SV-Puffer müssen Sie vorerst noch 'von Hand', das heißt mittels POKE, oder besser mit Hilfe des Mikreassemblers beschicken. Für die nächste Ausgabe ist vorgesehen, dafür geeignete und komfortable Ein-/Ausgaberoutinen zu veröffentlichen. Für das mitgesendete SV-Flagregister sind bislang nur die Flags 6 und 7 interessant, die denen der ZFlags entsprechen.

Als Beispiel sollen die ersten 100h Bytes des ROM auf Band gerettet werden: im ZX81 Code, mit 3600 Baud. Der Name sei Test und vom Typ Hex.

```

1 REM = 7FCC
2 REM 4100
3 REM 392A383900000000
   0000000001B2D2A3D
4 REM 40000000010000
5 REM STOP

```

Mit GOTO 1000 wird der Puffer dann geladen. Lassen Sie den FAST-Mode bestehen. Nun endlich können Sie mit

PRINT USR 32168

das erste Mal SuperTape in Aktion 'sehen' und auch hören, sofern Sie schon einen Kassettenrekorder, auf Aufnahme geschaltet, an der MIC-Buchse angeschlossen haben.

Zum Laden wird eine andere Startadresse gewählt, so daß die ROM-Seite in den RAM-Speicher (ab 6000 h = 24576) kopiert wird. Außerdem sollten Sie noch zusätzlich eine kleine Routine verwenden, die eventuelle Fehlermeldungen ausgibt. Der Fehlercode wird von LDSUPER im A-Register gespeichert, und zwar:

00h	=	0	=	' '	fehlerfrei
27h	=	39	=	'B'	falscher
				Name oder Typ	

2Bh = 43 = 'F' Fehler
bei der Prüfsumme
12n = 18 = '>' Fehler
bei Verify

Wenn Sie diesen Code in das BC-Register bringen, erhalten Sie mit dem USR-Befehl den dezimalen Wert. Diese Aufgabe erledigt folgendes Maschinenprogramm:

```
CALL LDSUPER
LD C,A
LD B,0h
RET
```

In unserem Mikroassembler sieht das so aus:

```

1 REM = 7FCC
2 REM 4400
3 REM 392A38390000000
  000000001B2D2A3D
4 REM 400360
5 REM - 7C00
6 REM CD547E
7 REM 4F
8 REM 0600
9 REM C9
10 REM STOP

```

Die Zusatzroutine beginnt demnach bei 7C00h. Sie wird nach GOTO 1000 — weiterhin im normaler FAST-Mode — mit PRINT USR 31744 gestartet. Daraufhin wartet der Rechner auf Signale am EAR-Eingang, kann also jetzt den Datenblock laden. Wenn auf dem Schirm zum Schluß eine Null erscheint, dann ist der Ladevorgang aller Wahrscheinlichkeit nach korrekt erfolgt, wovon Sie sich mit PEEKs ab 24576 überzeugen können.

Wenn Sie beim Laden einzelne Buchstaben des Namens oder des Typs durch ein '?' (Ccde) (Fh) ersetzen, wird das eingele-
sene Zeichen an dieser Stelle

übersprungen. Ein '*' (Code 17h) überliest sogar bis zum jeweiligen Ende des Strings.

Piepshow

Die Möglichkeit, ein Ausgangssignal zu erzeugen, ist beim ZX81 durch zwei I/O-Befehle gegeben:

OUT FFh,A ;einschalten
IN FEh,A ;ausschalten

Mit dem OUT-Befehl wird das Signal an der MIC-Buchse im Zeilenfrequenzrhythmus kurzzeitig wieder abgeschaltet (16 Takte lang). Allerdings wird dieser kurze Puls durch das eingebaute Bandpaßfilter weitgehend abgefangen, so daß sich bei den benutzten Baudraten noch keine Probleme ergeben. Um den 'OUT FFh,A'-Befehl überhaupt ausführen zu können, muß zuvor mit 'OUT FDh,A' der NMI-Generator umgeschaltet werden. Wenn der Akkumulator beim IN-Befehl auf 7Fh steht, wird nicht nur das Ausgangssignal abgeschaltet, sondern mit diesem Befehl auch die untere rechte Zeile der Tastatur geloesen. Damit können Sie also zum Beispiel auf BREAK testen (Bit 0 = 0). Darüber hinaus ist dieser Befehl auch für das Einlesen der Signale vom Band verantwortlich: bei erkanntem Signal ergibt sich Bit 7 zu eins.

Taktgefühl

Die Zahl der notwendigen Takte ergibt sich aus der halben Periodendauer der höheren Schwingung. Bei 7200 Bd be-

beträgt sie etwa 69 Mikrosekunden, entspricht also 224 Takten. In Bild 1 ist das Timing-Diagramm für die SVBYTE-Routine wiedergegeben. Der Balken bedeutet die Inversion des Ausgangssignals. Zwischen zwei Inversionen liegen bei Ausgabe einer Null 147 + SVWAIT Takte, und bei einer Eins das Doppelte. Folglich muß die Verzögerungsroutine SVWAIT insgesamt $224 - 147 = 77$ Takte bei 7200 und $448 - 147 = 301$ Takte bei 3600 Baud 'vernichten'. Die INVERT-Routine ist beim ZX81 etwas umständlich, da ja zwei verschiedene Ports angesprochen werden müssen. Andere Z80-Rechner, die dafür nur ein Port verwenden, können gerade hier einige Takte sparen.

Für das Bereitstellen der auszugebenden Daten stehen den aufrufenden Routinen nur 33 Takte zur Verfügung. Um das zu gewährleisten, wird beim letzten Byte in der SVBLOC-Routine die Schleifenkontrolle eingespart. Deshalb muß der Datenblock mindestens zwei Bytes lang sein — was wahrlich keine Einschränkung darstellt.

Bild 2 gibt das Timing-Diagramm der Laderoutine LDBYTE wieder. Die Testzeit für SuperTape ist am günstigsten für:

halbe Periode bei Null * 4/3 —
mittlere Flankenerkennungszeit.

Der letzte Anteil ergibt sich in LDBYTE zu ungefähr 17 bis 18 Takten. Als Testzeit errechnet sich dann für

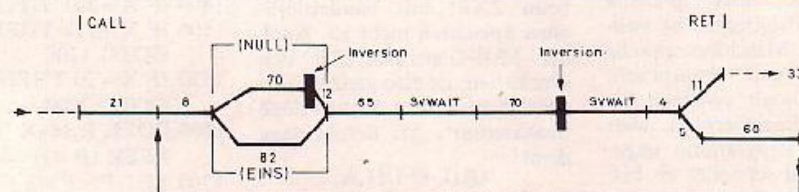


Bild 1. Timing für SVBYTE: 147T + SVWAIT zwischen zwei Inversionen

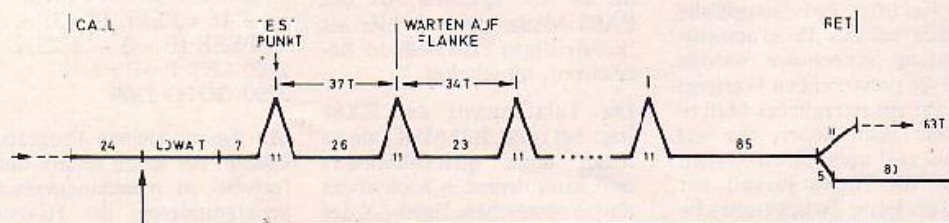


Bild 2. Timing für LDBYTE: 201T + LDWAIT für den Prüfzeitraum

7200 Baud : 280 Takte
3600 Baud : 580 Takte

Der Eigenverbrauch der Routine ist 201 Takte, so daß CALL LDWAIT noch 79 Takte bei 7200 beziehungsweise 379 Takte bei 3600 Baud abwarten muß. Für die Verarbeitung der eingelesenen Daten verbleiben den aufrufenden Routinen dann noch 63 Takte.

Die Hierarchie

Bild 3 zeigt, in welcher Weise die Save Routinen untergeordnet sind. Das Hauptprogramm SVSUPER gibt erst der SV-Puffer an das Unterprogramm SVBLOC und dann die auszugebenden Daten. SVBLOC übergibt dann die Synchronbytes, die Abschlußmarke, dann die Daten und schließlich die Prüfsumme und das Abschlußbyte an das Unterpro-

gramm SVBYTE. Hier wird dann jedes Byte Bit für Bit mit den INVERT-Routinen in die entsprechenden SuperTape-Pulse übertragen. SVWAIT sorgt dabei für die richtige Baudrate.

Ähnlich sieht die Ordnung bei den Ladeprogrammen aus (Bild 4). Das Hauptprogramm LDSUPER läßt zuerst von LBLOC den Parameterblock laden, dann wird überprüft, ob Name und Typ stimmen. Je nach gesetztem Flag wird eine ASCII-ZX81-Code-Umwandlung durchgeführt. Dann kontrolliert das Programm, ob eine neue Startadresse gesetzt werden soll und führt schließlich entweder die Lade-, oder die Verify-Routine aus.

LDBLOC ruft zunächst erst einmal die Synchronisier-Routine LDSYNC auf, die versucht, auf die Synchronbytes

einrasten und das richtige Abschlußbyte kontrolliert. Dann werden die Daten mittels LDBYTE geladen und schließlich zu LDTEST gesprungen, daß die Prüfsummen vergleicht. LDVER sieht genauso wie LDBLOC aus, nur daß es die Bytes mit denen im Speicher vergleicht, statt sie zu laden.

LDBYTE macht dann das Umgekehrte wie SVBYTE, indem es den Bandinhalt Bit für Bit einliest und diese ins E-Register schiebt. In diesem Fall ist LDWAIT für die richtige Baudrate verantwortlich.

Die einzelnen Routinen sind bis auf SVBYTE und LDBYTE so einfach strukturiert, daß sie wohl ohne Schwierigkeiten anhand des Listings verstanden werden können. Als einziger zeitsparender Trick tritt dabei die Kombination

CPI
JP PE

in den Blockübertragungsschleifen von SVBLOC (bzw. LDBLOC) auf. Der Vergleich interessiert hierbei gar nicht, sondern die Tatsache, daß CPI das Adressregister HL inkrementiert, das Zählregister BC dekrementiert und das PO/PE-Flag setzt. Dieses steht auf PE, solange BC noch nicht Null ist. Mit diesem Trick kommt man für die Schleifenüberwachung mit 26 Takten aus.

Das Schreiben ...

Die beiden Routinen SVBYTE und LDBYTE, die letztendlich die Bytes auf das Band bringen, beziehungsweise vom Band lesen, enthalten ebenfalls einen zeitsparenden Trick: ein Register übernimmt zusammen mit dem Carry-Flag sowohl die Bit-Ein-/Ausgabe als auch die Schleifenkontrolle. Wie das geschieht, zeigen für SVBYTE das Flußdiagramm (Bild 5) und die Registerdarstellung (Bild 7).

SCF setzt das Endflag, das nach und nach mit RR E durch das Ausgaberegister geschoben wird, wobei gleichzeitig das auszugebende Bit (also das niederwertige Bit zuerst) in das Carry-Flag gelangt. Hier kann es nun abgefragt und ausgewertet werden. Zum Schluß sorgt XOR A dafür, daß bei den nächsten RR E-Befehlen von links nur Nullen nachrücken. Auf diese Weise werden nacheinander alle Bits ins Carry geschoben, bis das E-Register auf

Eins steht. DEC E fragt ab, ob diese Eins schon erreicht ist. Falls ja, erfolgt der Rücksprung ins aufrufende Programm. Im anderen Fall wird mit INC E das Ausgaberegister wieder restauriert und das nächste Bit verarbeitet.

Die Auswertung

Eine Null im Carry bedingt einen Sprung ins Unterprogramm INVERT, das das Ausgangssignal invertiert. Eine Eins läßt diesen Sprung aus, inkrementiert dafür aber das Prüfsummenregister IY. Das folgende WAIT 55T bedeutet, daß der Rechner die nächsten 55 Takte nichts anderes tun soll, als nur die im Timingdiagramm (Bild 1) dargestellte Laufzeit von Call INVERT auszugleichen. Leider gibt es beim Z80 den WAIT-Befehl nicht, sondern es müssen dafür Dummy-Befehle gefunden werden, die genau diese Taktanzahl verbrauchen, ohne dabei wichtige Register zu verändern. Um diese Dummies zu kennzeichnen, sind sie im Listing als DEFB aufgeführt.

Daraufhin wartet das Programm unabhängig von dem Ausgangsbit 65 Takte, ruft die Routine SVWAIT auf, die für die Baudrate verantwortlich ist, invertiert das Ausgangssignal und ruft nochmals SVWAIT für die nächste Periodenhälfte auf.

... und das Lesen

Die Funktion von LDBYTE verdeutlicht das Flußdiagramm (Bild 6) und die Registerdarstellung (Bild 8). Hier wird eine Eins von links nach rechts durch das anfangs leere E-Register geschoben, bis sie ins Carry gelangt, woraufhin der Rücksprung erfolgt. Bei dieser Schiebaktion gelangen die eingelesenen Bits aus dem Carry ins E-Register.

Die Routine testet zunächst mit IN A, (FEh) den Zustand an der EAR-Buchse und vergleicht ihn mit dem im Register D gespeicherten. Um Zeit zu sparen, werden die gesetzten Flags auf den Stack gerettet und sofort die Flankenabfrage begonnen. Register D erhält dabei den eingelesenen Wert, und die FLWAIT-Schleife wird solange durchlaufen, bis sich ein geänderter Wert am Eingang ergibt. Dieser Wert kommt nun wieder in den Zustandsspeicher D. Ein

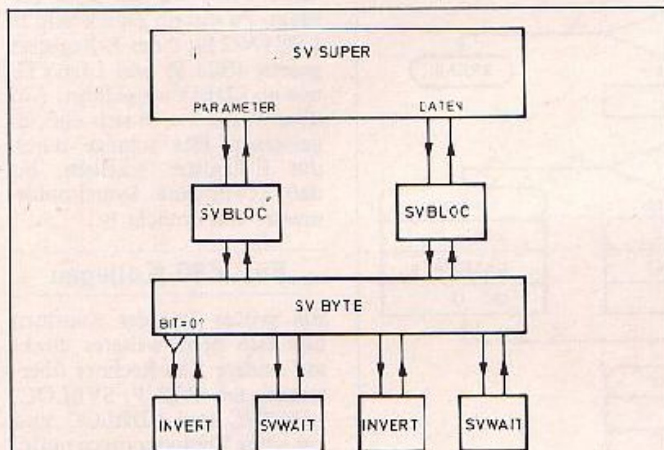


Bild 3. Die SAVE-Routinen

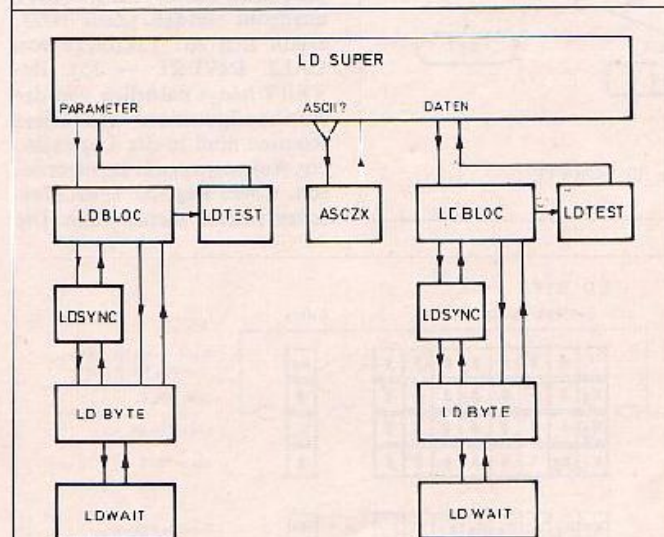


Bild 4. Die LOAD-Routinen

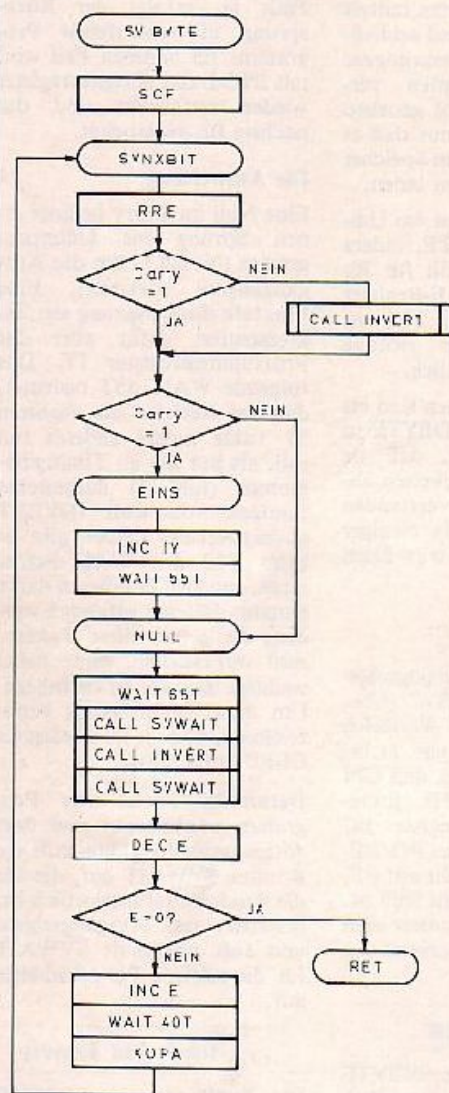


Bild 5. Flußdiagramm für SV BYTE

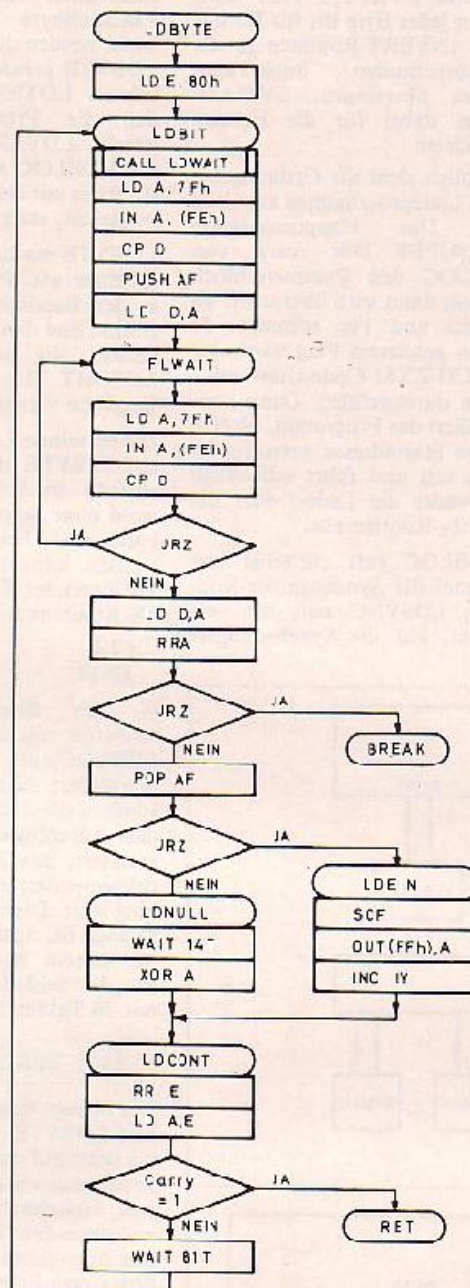


Bild 6. Flußdiagramm für LD BYTE

geänderter Wert kann sich aber auch durch BREAK ergeben haben, was mit RRA und JR NC, BREAK kontrolliert wird.

Jetzt ist genug Zeit da, sich an die Auswertung des Vergleichs zu hegen. Das Flagregister wird wieder vom Stack geholt, und das Carry-Flag wird entsprechend gesetzt:

Zwei gleiche Zustände ergeben eine Eins und bewirken eine Erhöhung der Prüfsumme in IV und eine Bildschirmausgabe (OUT (FFh), A);

ungleiche Zustände haben Carry-Flag = 0 zur Folge.

Damit ist das Eingangssignal als Bit interpretiert worden, welches nun nach dem oben beschriebenen Mechanismus in des E-Register wandert.

Für die Synchronisierung ist das bitweise Einlesen erforderlich, da der Rechner ja noch nicht weiß, wo ein Byte anfängt. Zu diesem Zweck wird in LDSYNC Bit 0 des E-Registers gesetzt (Bild 9) und LD BYTE erst ab LDBIT ausgeführt. Auf diese Weise lassen sich die eingelesenen Bits solange durch das E-Register schieben, bis daß gewünschte Synchronbitmuster 16h erreicht ist.

Für Z80-Kollegen

Ein großer Teil der Routinen läßt sich ohne weiteres direkt auf andere Z80-Rechner übertragen. Bei SVSUIP, SVBLOC, LDSYNC und LDBLOC sind gar keine Veränderungen nötig, in SVBYTE muß die Ausgleichszeit für CALL INVERT angepaßt werden (Zeile 0093, ergibt sich zu: Taktlänge von CALL INVERT = 15). INVERT hängt natürlich von der Portkonfiguration ab. Diese Routine muß in der Lage sein, ein Ausgangssignal zu invertieren, wobei Register D als Zwischenspeicher dienen kann. Die

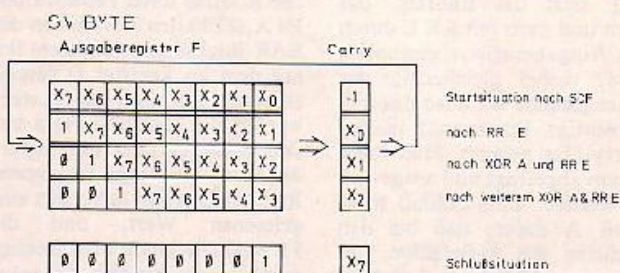


Bild 7. Bei SV BYTE wird eine 1 von Carry bis Bit 0 geschoben

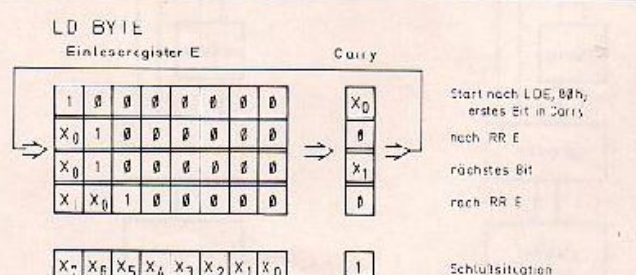
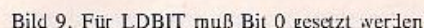


Bild 8. Bei LD BYTE wandert die 1 von Bit 7 bis zum Carry-Flag



In LDSUPER ist nur die Fehlermeldung in Zeile 186 durch einen anderen Code (normalerweise hoffentlich ASCII) zu ersetzen. Zeilen 175, 183 und 184 können dann entfallen. Auch in LDVER und LDTEST sind nur Änderungen in den Fehlermeldungen nötig (Zeile 238 und 256). Auf die Codewandlungsroutinen ab Zeile 371 können Sie natürlich verzichten.

Vergessen Sie nicht, alle Interrupts abzuschalten, bevor Sie die SuperTape-Routinen benutzen!

```

0001
0002
0003
0004
0005
0006
0007
0008
0009
0010
0011
0012
0013
0014
0015
0016
0017
0018
0019
0020
0021
0022
0023
0024
0025
0026
0027
0028
0029
0030
0031
0032
0033
0034
0035
0036
0037
0038
0039
0040
0041
0042
0043
0044
0045
0046
0047
0048
0049
0050
0051
0052
0053
0054
0055
0056
0057
0058
0059
0060
0061
0062
0063
0064
0065
0066
0067
0068
0069
0070
0071
0072
0073
0074
0075
0076
0077
0078
0079
0080
0081
0082
0083
0084
0085
0086
0087
0088
0089
0090
0091
0092
0093
0094
0095
0096
0097
0098
0099
0100
0101
0102
0103
0104
0105
0106
0107
0108
0109
0110
0111
0112
0113
0114
0115
0116
0117
0118
0119
0120
0121
0122
0123
0124
0125
0126
0127
0128
0129
0130
0131
0132
0133
0134
0135
0136
0137
0138
0139
0140
0141
0142
0143
0144
0145
0146
0147
0148
0149
0150
0151
0152
0153
0154
0155
0156
0157
0158
0159
0160
0161
0162
0163
0164
0165
0166
0167
0168
0169
0170
0171
0172
0173
0174
0175
0176
0177
0178
0179
0180
0181
0182
0183
0184
0185
0186
0187
0188
0189
0190
0191
0192
0193
0194
0195
0196
0197
0198
0199
0200
0201
0202
0203
0204
0205
0206
0207
0208
0209
0210
0211
0212
0213
0214
0215
0216
0217
0218
0219
0220
0221
0222
0223
0224
0225
0226
0227
0228
0229
0230
0231
0232
0233
0234
0235
0236
0237
0238
0239
0240
0241
0242
0243
0244
0245
0246
0247
0248
0249
0250
0251
0252
0253
0254
0255
0256
0257
0258
0259
0260
0261
0262
0263
0264
0265
0266
0267
0268
0269
0270
0271
0272
0273
0274
0275
0276
0277
0278
0279
0280
0281
0282
0283
0284
0285
0286
0287
0288
0289
0290
0291
0292
0293
0294
0295
0296
0297
0298
0299
0300
0301
0302
0303
0304
0305
0306
0307
0308
0309
0310
0311
0312
0313
0314
0315
0316
0317
0318
0319
0320
0321
0322
0323
0324
0325
0326
0327
0328
0329
0330
0331
0332
0333
0334
0335
0336
0337
0338
0339
0340
0341
0342
0343
0344
0345
0346
0347
0348
0349
0350
0351
0352
0353
0354
0355
0356
0357
0358
0359
0360
0361
0362
0363
0364
0365
0366
0367
0368
0369
0370
0371
0372
0373
0374
0375
0376
0377
0378
0379
0380
0381
0382
0383
0384
0385
0386
0387
0388
0389
0390
0391
0392
0393
0394
0395
0396
0397
0398
0399
0400
0401
0402
0403
0404
0405
0406
0407
0408
0409
0410
0411
0412
0413
0414
0415
0416
0417
0418
0419
0420
0421
0422
0423
0424
0425
0426
0427
0428
0429
0430
0431
0432
0433
0434
0435
0436
0437
0438
0439
0440
0441
0442
0443
0444
0445
0446
0447
0448
0449
0450
0451
0452
0453
0454
0455
0456
0457
0458
0459
0460
0461
0462
0463
0464
0465
0466
0467
0468
0469
0470
0471
0472
0473
0474
0475
0476
0477
0478
0479
0480
0481
0482
0483
0484
0485
0486
0487
0488
0489
0490
0491
0492
0493
0494
0495
0496
0497
0498
0499
0500
0501
0502
0503
0504
0505
0506
0507
0508
0509
0510
0511
0512
0513
0514
0515
0516
0517
0518
0519
0520
0521
0522
0523
0524
0525
0526
0527
0528
0529
0530
0531
0532
0533
0534
0535
0536
0537
0538
0539
0540
0541
0542
0543
0544
0545
0546
0547
0548
0549
0550
0551
0552
0553
0554
0555
0556
0557
0558
0559
0560
0561
0562
0563
0564
0565
0566
0567
0568
0569
0570
0571
0572
0573
0574
0575
0576
0577
0578
0579
0580
0581
0582
0583
0584
0585
0586
0587
0588
0589
0590
0591
0592
0593
0594
0595
0596
0597
0598
0599
0600
0601
0602
0603
0604
0605
0606
0607
0608
0609
0610
0611
0612
0613
0614
0615
0616
0617
0618
0619
0620
0621
0622
0623
0624
0625
0626
0627
0628
0629
0630
0631
0632
0633
0634
0635
0636
0637
0638
0639
0640
0641
0642
0643
0644
0645
0646
0647
0648
0649
0650
0651
0652
0653
0654
0655
0656
0657
0658
0659
0660
0661
0662
0663
0664
0665
0666
0667
0668
0669
0670
0671
0672
0673
0674
0675
0676
0677
0678
0679
0680
0681
0682
0683
0684
0685
0686
0687
0688
0689
0690
0691
0692
0693
0694
0695
0696
0697
0698
0699
0700
0701
0702
0703
0704
0705
0706
0707
0708
0709
0710
0711
0712
0713
0714
0715
0716
0717
0718
0719
0720
0721
0722
0723
0724
0725
0726
0727
0728
0729
0730
0731
0732
0733
0734
0735
0736
0737
0738
0739
0740
0741
0742
0743
0744
0745
0746
0747
0748
0749
0750
0751
0752
0753
0754
0755
0756
0757
0758
0759
0760
0761
0762
0763
0764
0765
0766
0767
0768
0769
0770
0771
0772
0773
0774
0775
0776
0777
0778
0779
0780
0781
0782
0783
0784
0785
0786
0787
0788
0789
0790
0791
0792
0793
0794
0795
0796
0797
0798
0799
0800
0801
0802
0803
0804
0805
0806
0807
0808
0809
0810
0811
0812
0813
0814
0815
0816
0817
0818
081
```



```

7E1A ED4BA7F 0200 LD BC, (LOLEN)
7E1B ACCDF 0201 LD A, (ZPLAKR)
7E1C C857 0202 BIT 2, A ;NELE STARTADR?
7E1D 2B03 0203 JR 1, SKADR
7E1E 2B0F 0204 LD HL, (SVSTART)
7E1F 0B4F 0205 SKADR BIT 1, A ;LADEN ODER VERIFY?
7E20 3C08 0206 LD A, (INCOUT)
7E21 2012 0207 JR NZ, LDVER
7E22 0208
7E23 0209
7E24 0210
7E25 0211
7E26 0212
7E27 0213
7E28 0214
7E29 0215
7E2A 0216
7E2B 0217
7E2C 0218
7E2D 0219
7E2E 0220
7E2F 0221
7E30 0222
7E31 0223
7E32 0224
7E33 0225
7E34 0226
7E35 0227
7E36 0228
7E37 0229
7E38 0230
7E39 0231
7E3A 0232
7E3B 0233
7E3C 0234
7E3D 0235
7E3E 0236
7E3F 0237
7E40 0238
7E41 0239
7E42 0240
7E43 0241
7E44 0242
7E45 0243
7E46 0244
7E47 0245
7E48 0246
7E49 0247
7E4A 0248
7E4B 0249
7E4C 0250
7E4D 0251
7E4E 0252
7E4F 0253
7E50 0254
7E51 0255
7E52 0256
7E53 0257
7E54 0258
7E55 0259
7E56 0260
7E57 0261
7E58 0262
7E59 0263
7E5A 0264
7E5B 0265
7E5C 0266
7E5D 0267
7E5E 0268
7E5F 0269
7E60 0270
7E61 0271
7E62 0272
7E63 0273
7E64 0274
7E65 0275
7E66 0276
7E67 0277
7E68 0278
7E69 0279
7E6A 0280
7E6B 0281
7E6C 0282
7E6D 0283
7E6E 0284
7E6F 0285
7E70 0286
7E71 0287
7E72 0288
7E73 0289
7E74 0290
7E75 0291
7E76 0292
7E77 0293
7E78 0294
7E79 0295
7E7A 0296
7E7B 0297
7E7C 0298
7E7D 0299
7E7E 0300
7E7F 0301
7E80 0302
7E81 0303
7E82 0304
7E83 0305
7E84 0306
7E85 0307
7E86 0308
7E87 0309
7E88 0310
7E89 0311
7E8A 0312
7E8B 0313
7E8C 0314
7E8D 0315
7E8E 0316
7E8F 0317
7E90 0318
7E91 0319
7E92 0320
7E93 0321
7E94 0322
7E95 0323
7E96 0324
7E97 0325
7E98 0326
7E99 0327
7E9A 0328
7E9B 0329
7E9C 0330
7E9D 0331
7E9E 0332
7E9F 0333
7FA0 0334
7FA1 0335
7FA2 0336
7FA3 0337
7FA4 0338
7FA5 0339
7FA6 0340
7FA7 0341
7FA8 0342
7FA9 0343
7FAB 0344
7FAC 0345
7FAD 0346
7FAE 0347
7FAF 0348
7FB0 0349
7FB1 0350
7FB2 0351
7FB3 0352
7FB4 0353
7FB5 0354
7FB6 0355
7FB7 0356
7FB8 0357
7FB9 0358
7FBA 0359
7FBB 0360
7FBC 0361
7FBD 0362
7FBE 0363
7FBF 0364
7FC0 0365
7FC1 0366
7FC2 0367
7FC3 0368
7FC4 0369
7FC5 0370
7FC6 0371
7FC7 0372
7FC8 0373
7FC9 0374
7FCA 0375
7FCB 0376
7FCC 0377
7FCD 0378
7FCE 0379
7FCF 0380
7FD0 0381
7FD1 0382
7FD2 0383
7FD3 0384
7FD4 0385
7FD5 0386
7FD6 0387
7FD7 0388
7FD8 0389
7FD9 0390
7FDA 0391
7FDB 0392
7FDC 0393
7FDD 0394
7FDE 0395
7FDF 0396
7FE0 0397
7FE1 0398
7FE2 0399
7FE3 0400
7FE4 0401
7FE5 0402
7FE6 0403
7FE7 0404
7FE8 0405
7FE9 0406
7FEA 0407
7FEB 0408
7FEC 0409
7FED 0410
7FEE 0411
7FEF 0412
7FF0 0413
7FF1 0414
7FF2 0415
7FF3 0416
7FF4 0417
7FF5 0418
7FF6 0419
7FF7 0420
7FF8 0421
7FF9 0422
7FFA 0423
7FFB 0424
7FFC 0425
7FFD 0426
7FFE 0427
7FFF 0428

```

```

7FAA 3F 0334 LDE NS SCF
7FAB 09FF 0335 CUI
7FAC FD23 0336 INC
7FAD 0B1D 0337 LOCNT AR
7FAE 7B 0338 LD
7FAF 0B 0339 RET C
7FB0 0340 DEFE
7FB1 0341 DEFE
7FB2 0342 DEFE
7FB3 0343 JR
7FB4 0344
7FB5 0345
7FB6 0346
7FB7 0347
7FB8 0348
7FB9 0349
7FBA 0350
7FBB 0351
7FBC 0352
7FBD 0353
7FBE 0354
7FBF 0355
7FC0 0356
7FC1 0357
7FC2 0358
7FC3 0359
7FC4 0360
7FC5 0361
7FC6 0362
7FC7 0363
7FC8 0364
7FC9 0365
7FCA 0366
7FCB 0367
7FCC 0368
7FCD 0369
7FCE 0370
7FCF 0371
7FD0 0372
7FD1 0373
7FD2 0374
7FD3 0375
7FD4 0376
7FD5 0377
7FD6 0378
7FD7 0379
7FD8 0380
7FD9 0381
7FDA 0382
7FDB 0383
7FDC 0384
7FDD 0385
7FDE 0386
7FDF 0387
7FE0 0388
7FE1 0389
7FE2 0390
7FE3 0391
7FE4 0392
7FE5 0393
7FE6 0394
7FE7 0395
7FE8 0396
7FE9 0397
7FEA 0398
7FEB 0399
7FEC 0400
7FED 0401
7FEE 0402
7FEF 0403
7FF0 0404
7FF1 0405
7FF2 0406
7FF3 0407
7FF4 0408
7FF5 0409
7FF6 0410
7FF7 0411
7FF8 0412
7FF9 0413
7FFA 0414
7FFB 0415
7FFC 0416
7FFD 0417
7FFE 0418
7FFF 0419
7FA0 0420
7FA1 0421
7FA2 0422
7FA3 0423
7FA4 0424
7FA5 0425
7FA6 0426
7FA7 0427
7FA8 0428
7FA9 0429
7FAB 0430
7FAC 0431
7FAD 0432
7FAE 0433
7FAF 0434
7FB0 0435
7FB1 0436
7FB2 0437
7FB3 0438
7FB4 0439
7FB5 0440
7FB6 0441
7FB7 0442
7FB8 0443
7FB9 0444
7FBA 0445
7FBB 0446
7FBC 0447
7FBD 0448
7FBE 0449
7FBF 0450
7FC0 0451
7FC1 0452
7FC2 0453
7FC3 0454
7FC4 0455
7FC5 0456
7FC6 0457
7FC7 0458
7FC8 0459
7FC9 0460
7FCA 0461
7FCB 0462
7FCC 0463
7FCD 0464
7FCE 0465
7FCF 0466
7FD0 0467
7FD1 0468
7FD2 0469
7FD3 0470
7FD4 0471
7FD5 0472
7FD6 0473
7FD7 0474
7FD8 0475
7FD9 0476
7FDA 0477
7FDB 0478
7FDC 0479
7FDD 0480
7FDE 0481
7FDF 0482
7FE0 0483
7FE1 0484
7FE2 0485
7FE3 0486
7FE4 0487
7FE5 0488
7FE6 0489
7FE7 0490
7FE8 0491
7FE9 0492
7FEA 0493
7FEB 0494
7FEC 0495
7FED 0496
7FEE 0497
7FEF 0498
7FF0 0499
7FF1 0500
7FF2 0501
7FF3 0502
7FF4 0503
7FF5 0504
7FF6 0505
7FF7 0506
7FF8 0507
7FF9 0508
7FFA 0509
7FFB 0510
7FFC 0511
7FFD 0512
7FFE 0513
7FFF 0514

```

Warte mal ...

Wenn Sie den Z80 X Takte warten lassen wollen, und X liegt zwischen 9 und 4113, dann können Sie die Warteroutine nach folgendem Schema bestimmen:

```

LD A,N      3E ...
LOOP DEC A  3D
JR NZ LOOP 20 FD
(DUMMY)

```

N und die Dummy-Befehle müssen dann bestimmt werden. Aus

$$(X-2)/16 = Z \text{ Rest } R$$

ergibt sich der ganzzahlige Anteil Z und der Rest R.

R	Dummies	N
0	—	Z
1	A6D8D8	Z-1
2	A6A6A7	Z-1
3	A6A6D8	Z-1
4	A7	Z
5	C0	Z
6	3435	Z-1
7	A6	Z
8	A7A7	Z
9	A7D8	Z
10	C0C0	Z
11	A6A7	Z
12	A6D8	Z
13	A7A7D8	Z
14	A6A6	Z
15	A6A7A7	Z

Die Dummies und die Schlicke sind so ausgelegt, daß nur das AF-Register beeinflusst wird.

Langsam, aber gründlich

Casio FP 200: Hand-Held mit Statistik-Funktionen

Peter Hagemann

Der Anwenderkreis von Hand-Held-Computern wächst ständig. Nun bietet Casio mit dem FP 200 für unter 1000 D-Mark ein wohl speziell für den Einsteiger interessantes Gerät an. Kann dieser Computer neben seinen deutlich teureren Konkurrenten (EPSON HX-20, TRS 80 Modell 100) bestehen? Der c't-Test offenbart die Besonderheiten des FP 200.



In der Grundausstattung bekommt man den FP 200 ohne Koffer, ohne Tasche und ohne Netzteil. Zum Lieferumfang gehört allerdings ein Batteriesatz, eine Bedienungsanleitung, ein Bezugshandbuch und eine Kommando-Liste. Die in deutsch gehaltene Literatur erklärt auch dem ungeübten Anwender, wie er seinen FP 200 zum Leben erweckt.

Der Computer ist mit seinen Maßen von 310x220x56mm in allen drei Dimensionen geringfügig größer als der HX-20; das Format 'DIN A4' trifft also nicht mehr zu. Trotzdem behält das stabile Gehäuse 'Hand-Held-Format'.

Als erstes fällt am FP 200 auf, daß das asymmetrisch eingebaute LCD eigentlich auch doppelt so groß sein könnte. Die Anzeige hätte dann mit 40x8 Zeichen der des TRS 80 Modell 100 entsprechen. Aber auch so ist es erfreulich, daß mit 20 Spalten in 8 Zeilen doppelt so viele Zeichen dargestellt werden können wie auf dem LCD des HX-20.

Jedes Zeichen stellt der FP 200 in einem 8x8-Feld dar. Die dazu benötigten 160x40-Punkte lassen sich darüber hinaus auch einzeln setzen, löschen oder prüfen.

Wie bei seinen Mitbewerbern ist beim FP 200 ein Steller vorhanden, mit dem sich der Kontrast des LCDs, entsprechend dem Betrachtungswinkel, anpassen läßt.

Tastatur

Hier wurde am verkehrten Ende gespart. Die mechanisch sehr einfach geratene Konstruktion besitzt weder 'Auto Repeat' noch eine 'Roll-Over'-Funktion; das heißt bevor ein neuer Tastendruck angenommen wird, muß die zuletzt gedrückte Taste losgelassen werden. Selbst für 'mittelschnelle' Schreiber ist dieses System zu langsam.

Zur Verfügung stehen der ASCII-Zeichensatz und einige Grafikzeichen. Die deutschen Umlaute und das 'ß' sind nicht darstellbar. Ansonsten werden Sonderfunktionstasten durch Größe oder Farbe vom eigentlichen Tastenfeld abgehoben.

Fünf Funktionstasten sind doppelt (SHIFT) mit bis zu je 15 Zeichen frei belegbar. Wenn das durch Umschaltung erreichbare, integrierte Zehnerfeld nicht ausreicht, kann eine zusätzliche, externe Zehnertastatur anschließen (Option).

Schnittstellen

Ein Blick auf die Rückseite des Gerätes läßt folgende Schnittstellen erkennen:

Kassettenrecorder-Interface (CMT)

Über die 8polige Buchse läßt sich ein handelsüblicher Rekorder anschließen. (Der Kauf eines Spezialgerätes ist also nicht erforderlich.)

RS-232 C-Interface

Der Anschluß erfolgt ebenfalls über eine 8polige Buchse. Das ist zwar nicht unbedingt normgerecht, spart aber (wie beim HX-20) Steckerkosten.

Selbst CASIO befürchtet (laut Handbuch) Verwechslungen der CMT und der RS-232 C-Buchse. Es ist daher unverständlich, warum für den Rekorder keine andere Buchse vorgesehen wurde. Konstruktiv müßte das leicht möglich sein, da beim CMT trotz REMOTE nur fünf Stifte belegt sind.

CENTRONICS-kompatible Parallelschnittstelle

Hierzu läßt sich nicht viel sagen, außer, daß die Haltebügel für den Stecker störend vom Gerät abstecken (nicht einklappbar).

Anschluß für ein CASIO-Floppy-Disk-Laufwerk

Diese Buchse befindet sich hinter dem Batteriekasten, der vor dem Anschließen der Floppy entfernt werden muß.

Für diejenigen, die sich über das RS-232 C-Interface freuen, haben, hier nun der Wermutstropfen: Sowohl die CMT- als auch die RS-232 C-Schnittstelle arbeiten mit einer festgelegten Übertragungsgeschwindigkeit von 300 Baud. Eine Umstellung ist nicht möglich. Auch eine so komfortable Einstellung der Übertragungsdaten wie beim HX-20 oder beim TRS 80 Modell 100 ist beim FP 200 nicht durchführbar.

Alle Interfaces, bis auf die Recorder-Schnittstelle, arbei-

ten nur bei angeschlossenem Netzgerät. Hierdurch wird die Mobilität des Systems doch etwas eingeschränkt.

In der Grundausstattung wird der FP 200 zwar mit 32K ROM, aber nur mit 8K RAM geliefert. Wem das zu wenig ist, der kann den Speicher in 8K-Schritten aufrüsten. CASIO bietet hierzu Module an, die ohne Mithilfe eines Fachhändlers vom Anwender eingesetzt werden können. Geplant ist auch eine zur Zeit nicht lieferbare 8K-ROM-Erweiterung (nach gleichem System).

Wer einen zünftigen Steckplatz für die Speicher-Module erwartet hat, der wird enttäuscht. Die Module werden über Leitgummi-Kontakte auf die vergoldeten Leiterbahnen gedrückt und dort arretiert (Bild 1).

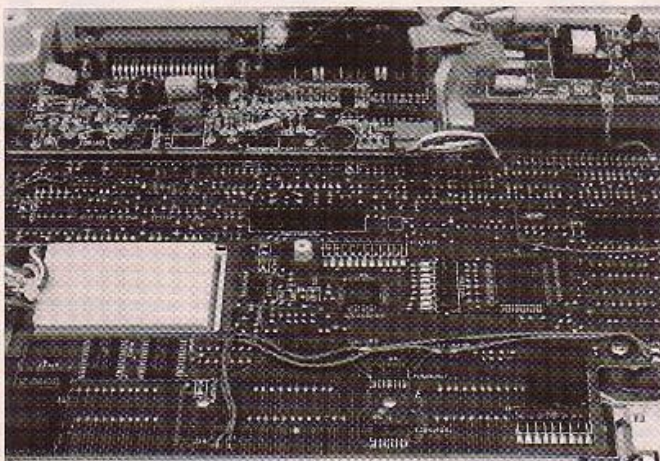
Die Hauptstromversorgung des Gerätes erfolgt entweder durch das Netzteil (Option) oder durch vier in den Batteriekasten eingelegte Mignon-Batterien. Im letzteren Fall beträgt die Betriebsdauer laut Herstellerangaben bei Alkali-Batterien und 32K RAM-Ausbau nur circa elf Stunden. Diese Tatsache und die Betriebseinschränkung der Schnittstellen machen das Netzgerät zu einer dringend erforderlichen Anschaffung.

Etwas riskant erscheint die Art der Speicherpufferung, da CASIO eine Methode ohne NiCd-Akku gewählt hat. Getrennt vom Haupt-Batteriekasten existiert eine weitere Batteriehalterung für zwei 'Speicherschutz-Batterien'.

c't-Prüfstand

Bild 1: Eine eigenwillige Lösung für Speichererweiterungen: Die Kontaktierung erfolgt beim Einsetzen der Module über Leitummii, das auf vergoldete Leiterbahnen drückt.

Bild 2: Bemerkenswert an der Platine: Erge Leiterbahnführung und Flat-Pack-ICs.



Hierdurch wird der RAM-Inhalt auch bei abgeschaltetem FP 200 erhalten.

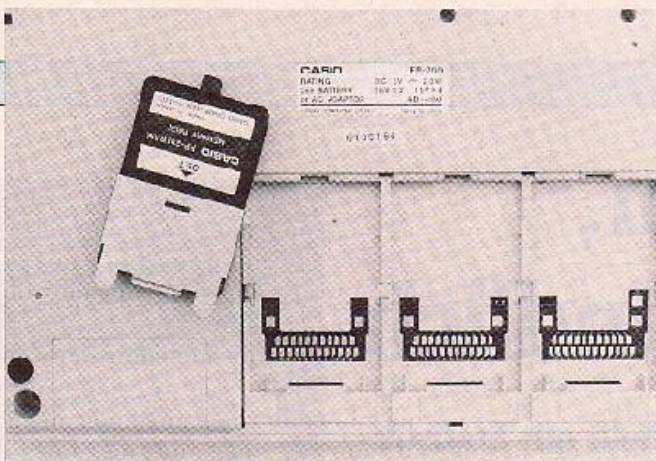
Leider gibt es keine Anzeige und kein Meßgerät, das etwas über den Zustand der Schutz-Batterien aussagt. Man muß also gelegentlich (hoffentlich rechtzeitig) Messungen anstellen oder einen Austausch auf Verdacht durchführen. CASIO spricht von einer 6monatigen Lebensdauer der Batterien.

Ein Wechsel der Speicher-Schutz-Batterien zerstört den RAM-Inhalt. Nach dem Wechsel beziehungsweise bei der ersten Inbetriebnahme muß das RAM durch Eingabe von 'RESET' <RETURN> initialisiert werden. Der FP 200 'merkt' hierbei gleichzeitig, wieviel Speicher-Module in seinem Bauch eingebaut sind.

Die 'Datenverarbeitung' wird beim FP 200 von einer 80C85-CPU durchgeführt. Die Quarzfrequenz beträgt 6,144 MHz, was einem Systemtakt von rund 3 MHz entspricht.

CETL ...

Im Gegensatz zu anderen Rechnern kann man beim FP 200 durch einen 'MODE'-Schalter festlegen, ob man mit BASIC oder mit CETL arbeiten will. Bei CETL handelt es sich um



ein einfaches Tabellenkalkulationsprogramm. Dadurch wird in gewissem Umfang eine Geräteanwendung ohne Programmierkenntnisse ermöglicht. CETL läßt sich durch 16 leicht erlernbare Befehle handhaben. Die Eingaben erfolgen im Dialog mit dem Rechner (Tabelle 1).

<A>uto	ank	<C>omp
<D>elet	<F>ind	<G>ct
<I>nscri	<J>ump	<K>ill
<L>ist	<M>ove	<N>ew
<P>ut	<R>ename	<S>ort
<T>able		

Tabelle 1: Befehlssatz von CETL

Der Umfang numerischer oder alphanumerischer Tabellen hängt nur von der jeweiligen Speicherkapazität ab. Die Berechnung der maximal möglichen Datensätze ist in der Bedienungsanleitung verständlich beschrieben.

CETL verwaltet im RAM bis zu zehn verschiedene Dateien. Als Ergebnis bekommt man einen formatierten Ausdruck. Jede einzelne Dateiinformation läßt sich aber auch auf dem LCD betrachten. Korrekturen und Eingaben sind genauso leicht möglich wie Verknüpfungen der Datenfelder durch die Grundrechenarten. CETL kann komplizierte Bearbeitungsaufträge an BASIC über-

geben. Die geänderten Daten können anschließend wieder von CETL genutzt werden.

... und BASIC

Wer einen FP 200 in der Grundversion gekauft hat: und BASIC startet, wird mit der Meldung: '1902 Bytes Free' überrascht. Wo ist der Rest von den 8K RAM geblieben?

Circa 2,3K werden unwiderruflich an das Betriebssystem abgetreten. Die restlichen 5,7K teilt der FP 200 automatisch zwischen BASIC, Stringvariablenbereich und CETL auf.

Mit 'CLEAR' und 'AREA' lassen sich Bereichsänderungen vornehmen. Trotzdem erhält man nie mehr als etwa 5,7K für den Betrieb mit BASIC. Neben dem Netzteil zeichnet sich hier also eine Speichererweiterung als unbedingt erforderlich ab.

Das BASIC selbst weist einige Besonderheiten auf. So sind zum Beispiel Variablenamen mit 255 signifikanten Stellen möglich. Wer jetzt noch Schwierigkeiten bei der Na-

mensgebung hat ist selber Schuld.

Felder lassen sich aber leider nur dreidimensional festlegen. Der FP 200 ist sehr genau! Bei Variablen einfacher Genauigkeit werden intern neun Stellen verwendet, von denen sechs angezeigt werden können. Bei doppelter Genauigkeit werden intern sogar 19 Stellen verarbeitet. Der Speicherplatzbedarf steigt dadurch aber auf sechs beziehungsweise elf Bytes pro Wert.

Bei so viel Genauigkeit gibt sich der FP 200 mit einfachen 'Integers' gar nicht erst ab. Eine Integerdefinition ist ebenso wie die im Handbuch beschriebene 'erweiterte' Genauigkeit nicht möglich.

Der hohe Speicherplatzbedarf der Variablen und die fehlenden Integer-Verarbeitung macht sich natürlich deutlich im Benchmark-Test bemerkbar (Tabelle 2).

Obwohl die Benchmark-Tests nicht das 'A & O' der Computer-Bewertung darstellen, werden extreme Werte des FP 200 vermutlich manchen Kaufinteressenten abschrecken.

Daß der FP 200 gut mit Zahlen umgehen kann, steht außer Frage. Als einer der wenigen Rechner bietet er sogar implementierte Statistik-Befehle (Tabelle 3). Dadurch wird der Programmieraufwand für diese Problemstellung erheblich reduziert.

Überhaupt ist der Befehlssatz recht umfangreich. Alle Befeh-

Rechner	BMT1	BMT2	BMT3	BMT4	BMT5	BMT6	BMT7	BMT8
alphaTronic PC	2,2	5,3	15,4	16,7	18,1	31,0	42,6	17,8
Apple II Plus	1,4	8,4	15,8	17,6	19,0	28,4	45,0	10,4
TRS 80 M1 L2	2,8	11,2	27,0	27,8	31,0	50,6	78,0	11,8
TRS 80 M100	3,7	9,8	26,6	29,7	31,4	46,8	52,8	20,9
CASIO FP 200	4,6	17,6	55,4	56,4	60,7	95,5	137,0	56,9

Tabelle 2: Benchmark-Test (Zeitangabe in Sekunden)

Funktion	Bedeutung
CNT	Anzahl der Eingabepunkte für statistische Verarbeitung
SUMX	Summe der X-Daten
SUMY	Summe der Y-Daten
SUMX 2	Summe der Quadrate der X-Daten
SUMY 2	Summe der Quadrate der Y-Daten
SUMXY	Summe der Produkte der X- und Y-Daten
MEANX	Mittelwert der X-Daten
MEANY	Mittelwert der Y-Daten
SDX	Probenstandardabweichung der X-Daten
SDY	Probenstandardabweichung der Y-Daten
SDXN	Standardabweichung von der Gesamtheit der X-Daten
SDYN	Standardabweichung von der Gesamtheit der Y-Daten
LRA	Koeffizient für lineare Regression
LRB	Konstantenausdruck für lineare Regression

Tabelle 3: Statistik-Funktionen des FP 200

	HX-20	Modell 100	FP200
Stromversorgung	Akku	Batt/Netz	Batt/Netz
Datensicherung	Akku	Batt/Akku	Batterie
LCD	20 x 4	40 x 8	20 x 8
Tastatur	deutsch	ASCII	ASCII
Programmspeicherung	5	19	10 + 10 Data
Programmre BASIC +	Monitor	Text, Adressen	CETL
		Termin, TFI.COM	
Dokumentation	deutsch	englisch	deutsch
Prozessor	2 x 6301	80C85	80C85
Editieren	Bildschirm	extra Modus	extra Modus
Variablenidentifikation	16	2	255
Variablendef. String +	INT, SNG, DBL	INT, SNG, DBL	SNG, DBL
RS-232C-Interface	einstellbar	einstellbar	fest
CENTRONICS	nein	ja	ja

Tabelle 4. Der FP200 im Vergleich

le für die Vorbereitung von RANDOM-Dateien bei Floppy-Betrieb (FIELD, LSET, RSET, usw.) sind ebenso wie RENUM, OPTION BASE und DEFFN bereits im ROM enthalten. Es verwundert aber, daß eine so gebräuchliche Funktion wie INSTR nicht vorgesehen ist.

BASIC verwaltet, ähnlich wie CETL, zehn verschiedene Programmbereiche. Allen Programmbereichen stehen sämtliche Variablenwerte zur Verfügung, denn ein 'PROG-Wechsel' oder ein 'RUN' zerstört die Variableninhalte nicht. Auch nach dem Aus- und Einschalten bleiben alle Werte erhalten. Dadurch lassen sich Funktionen leicht auf verschiedene Programme aufteilen. Man muß nur aufpassen, daß der Variablen Speicherbedarf nicht zu groß wird.

Alle Programmebenen können gemeinsam durch ein Passwort geschützt werden. Ein LIST, LOAD oder SAVE wird dann unmöglich.

Betrüblich ist es, daß man mit den Cursor-Tasten auf dem LCD zwar beliebig 'hin- und herfahren' kann, aber trotzdem

eine Korrektur nicht möglich ist. Um zu korrigieren, muß der 'EDIT-Modus' aufgerufen werden.

Wer bei der Taste 'INSERT' an ein komfortables Einfügen denkt, wird ebenfalls enttäuscht. Es können lediglich Leerzeichen eingefügt werden, die man in einem weiteren Arbeitsschritt überschreiben kann.

Fazit

Die gewonnenen Erkenntnisse führen zu folgendem Ergebnis: Für unter 1000,- DM erhält man relativ 'viel' Computer.

Durch CETL wird auch dem Anfänger eine sofortige Anwendungsmöglichkeit geboten. Die sicherlich verbesserte Hardware sollte unter Berücksichtigung des Preises mit der Konkurrenz verglichen werden.

Die Aufstellung (Tabelle 4) soll dabei helfen. Zeikritische Anwendungen dürfen mit dem CASIO nicht zu lösen sein. Die Tastatur ist für eine schnelle Eingabe nicht geeignet. □

Ergebnisse auf einen Blick:

- ⊕ günstiger Preis
- ⊕ integriertes Tabellenkalkulationsprogramm
- ⊕ CENTRONICS und RS-232 C
- ⊕ Speicher kann ohne Werkstatt erweitert werden
- ⊕ integrierte Statistik-Befehle
- ⊕ lange Variablenamen
- ⊖ zu einfache ASCII-Tastatur
- ⊖ nicht überwachter Speicherschutz
- ⊖ langsame Ausführungszeit
- ⊖ nur 300 Baud bei CMT und RS-232 C
- ⊖ Grundversion nur 5,7K freies RAM
- ⊖ Schnittstellen nur mit Netzteil anzusprechen (außer CMT)
- ⊖ kein Bildschirmeditor

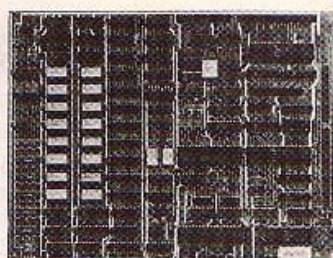
BULLET — das CP/M-Geschoß



Einer der schnellsten und leistungsfähigsten CP/M-Rechner für CP/M 3,0 (Plus) und 2,2

System-Lieferung möglich:

mit 2 mal 5,25-Zoll-Laufwerken je 1 MB oder 2 mal 5,25-Zoll-Laufwerken je 1,6 MB oder 5,25-Zoll-Laufwerk und eingeb. Harddisk 10 MB oder 2 mal 8-Zoll-Laufwerke DS/DD je 1,6 MB oder 8-Zoll-Laufwerk plus Harddisk ab 10 MB



20 x 27 cm

Universell einsetzbar für:

- OEM's
- Praxis und Hobby
- Entwicklung
- Universitäten und Institute

Fachhändler gesucht

- Verwaltungen und Anschluß für vier Floppys 5,25-Zoll (40, 80 oder 77 Spur) und Anschluß für vier Floppys 8-Zoll (SS/SD oder DS/DD) gleichzeitig, Anschluß über Flachbarkabel Harddisk-Anschluß serienmäßig
- Kopieren der Software von 5,25-Zoll auf 8-Zoll und Kopieren der Software von 8-Zoll auf 5,25-Zoll-Diskette
- alle gängigen Programmiersprachen lauffähig
- Lieferung als Platine oder als Komplettsystem — Sie bestimmen, wir liefern —
- Kundenspezifische Applikationen möglich
- Anschluß-Platine für 8086 für MS-DOS und RAM-Floppy bis 1 MB vorgesehen.
- Convertierungsbulet zum Lesen und Kopieren von Daten fast aller Disketten anderer Computer.

Ausführliche Unterlagen und Anwendungsbeispiele bei:



M. Mandt
Ilmspaner Straße 29, D-6971 Großrinderfeld
Telefon (0 93 49) 271-12 71, Telex 689 549 EPS D

Detlef Grell

In puncto Kleincomputerpatriotismus sind uns die Briten vermutlich uneinholbar voraus. Hat zumindest unsere Konsumer-Industrie Geräte, die man an Stelle japanischer Produkte kaufen kann, so sind Computer aus Deutschland immer noch rar. Und je länger die deutsche Industrie herumtrödelte, desto schwieriger wurde es natürlich, noch eine 'ökonomische Nische' auszumachen. Mit dem gesamten Konzept, das hinter MAX 1 steckt, scheint tatsächlich eine Marktlücke gefunden zu sein.



MAX aus Deutschland

Vielseitiger Kleincomputer, auch zum Selbstbau

Auf den ersten Blick ist MAX ein Hand-Held-Computer. Komfortable deutsche Schreibmaschinentastatur und zweizeiliges LC-Display drängen einem diesen Eindruck auf. BASIC ist resident vorhanden, ebenso ein Kassetten-Interface und eine Centronics-Schnittstelle, und das RAM ist gepuffert.

Das Format hingegen tendiert deutlich mehr zu DIN A3 als zum obligaten Hand-Held-A4. Und wenn man ihn in der Alu-Blech-Gehäuse-Version in der Hand 'held', ist die Verwirrung komplett: rund drei Kilogramm massive Solidität. (Erinnerungen an deutsche Wertarbeit werden wach.)

Vielseitig

Mit MAX kann man (laut Prospekt) rechnen, schreiben, kreativ spielen, programmieren, wecken, Zeit anzeigen, messen, steuern und regeln. Dem Leser bleibe überlassen, wie er MAX nun bezeichnen möchte.

MAX gibt es als Bausatz und als Fertiggerät. Außerdem kann er durch spezielle Steckmodule einfach erweitert werden. Man bekommt ihn auch im Kunststoffgehäuse und in verschiedenen Ausbaustufen, sowohl was Hardware als auch Software anbelangt.

Und es gibt etwas ganz Neues: Der Kauf eines Bausatzes ist mit einem eintägiger Selbstbau-Seminar gekoppelt. Nach dem Motto 'morgens gebracht, abends gemacht' wird dem Käufer garantiert, ein fertiges und funktionierendes Gerät mit nach Hause zu nehmen.

Alle, für die nur zwei Prozessor-

ren Weltgeltung besitzen, werden den 5803, der den MAX

MAX' Anatomie

beseelt, mit einer gewissen Mißbilligung betrachten. Nun, auch wenn wir durchaus 'anderen' Prozessoren tolerant gegenüberstehen: der Weg zu CP/M und APPLE-Software ist erstmal verbarrikadiert. Allerdings liegen MAX' Hauptanwendungen auch ganz woanders, so daß man ohnehin mehr auf spezielle Software oder Hausgemachtes zurückgreifen wird.

Der eigentliche Rechner befindet sich auf einem Mother-Board, und somit gibt es natürlich auch die geliebten SLOTS, allerdings ganz anders. Erweiterungen werden von außen als Steckmodule in das Gehäuse eingeschoben, und zwar ohne Öffner des Hauptgerätes. Diese kleinen Baugruppenträger mit Steckerleiste für Wrap-Pfosten haben eine eigene Frontplatte, und — wo erforderlich — eine Canon-Buchse für den Kontakt nach außen.

Da es MAX als Selbstbaugrät gibt, ist ein luftiger Aufbau, wie er hier praktiziert wird, sicher günstiger als eine vollgequatschte Multilayer-Platine. Derzeit sind auf der Platine maximal 16K statisches RAM unterzubringen, drei PIAs (6821), drei EPROMs (2764), ein Uhrenbaustein (der standardmäßig dazugehört) und diverse 'kleine' ICs. Nicht zu vergessen die Puffer-Akkus, die allerdings nicht den Betrieb des Rechners unabhängig vom Netz ermöglichen, sondern lediglich der Daten- und Pro-

grammsicherung dienen. Auf der Hauptplatine befindet sich auch die Elektronik für das Kassetten-Interface (über fünfpolige DIN-Buchse herausgeführt) und für die Centronics-Schnittstelle.

Zusätzlich zum Mother-Board ist ein Netzteil vorhanden. An der Rückseite des Gehäuses befindet sich eine Kaltgeräte-Steckdose zum Anschluß eines Netzkabels. Auf der Netzteilplatine ist auch das Relais zum Ein-/Aus-switchen (im Schaltuhrbetrieb oder per BASIC) einer in der Rückwand eingebauten Schuko-Netzsteckdose untergebracht (siehe auch Bild 1).

Kennenlernen

MAX kommt dem Einsteiger

entgegen. Nach dem Einschalten meldet er sich mit 'Hallo, alles klar, ich bin bereit ...', und in der Zeile darunter findet man auch schon das Hauptmenü mit Abkürzungen, die in ihrer Reihenfolge unmittelbar den acht Funktionstasten zugeordnet sind. Drückt man auf eine dieser Tasten, erscheint ein Untermenü, daß sich ebenfalls wieder an den Funktionstasten orientiert.

Menügeführt (Funktionstaste F2) kann man die interne Echtzeituhr stellen, beziehungsweise deren Anzeige aufrufen. Mit F3 erreicht man die Zeitschaltuhr. Über die eingebaute Netzsteckdose lassen sich externe Geräte bis 400 Watt Leistungsaufnahme schalten, man kann sich

BMM-BASIC mit Standard-BASIC-Befehlssatz:

BEEP	DIM	NEW
CALL	END	PLIST
CLEAR	FOR ... NEXT	POKE
CLOAD	GOTO	PRINT
CMERGE	GOSUB ... RETURN	PRINTER
CSAVE	IF	RAD
CONSOLE	INPUT	REM
DELETE	LET	RUN
DEG	LIST	WAIT

Problembezogene Erweiterungen:

DOUT	— Digitale Ausgabe
DIN	— Digitale Eingabe
AOUT	— Analoge Ausgabe
AIN	— Analoge Eingabe
NETZ	— Steuern der 220-V-Steckdose
GEN	— Erzeugen einer Impulsfolge
XLES/YLES	— Für Tracer (in Vorbereitung)

Funktionen:

LN, LOG, SQR, EXP	ABS, INT, RND
SIN, COS, TAN	PEEK
ASN, ACS, ATN	CHR\$
	LEFT\$, RIGHT\$

Tabelle 1

aber auch mit einem 1000Hz Piepton begnügen.

Mit F6 wird der BMM-BASIC-Interpreter aktiviert. MAX' BASIC kann zwar nicht als üppig bezeichnet werden, aber alle notwendigen Befehle sind vorhanden (Tabelle 1).

Dafür sind seine Editormöglichkeiten erfreulich komfortabel. Es gilt der Grundsatz, daß das, was man auf dem Display sieht (und per RETURN abschickt), auch in den Programmspeicher übernommen wird. Diverse Cursor-Funktionen sind vorhanden, auch Insert und Delete ist möglich.

Darüber hinaus erfolgt nach der Eingabe einer jeden Programmzeile ein ausgiebiger Syntax-Check. Der Cursor springt in die unmittelbare Nähe des Fehlers, und der Rechner weigert sich strikt, eine syntaktisch falsche Zeile anzunehmen.

MAXens BASIC hält aber auch Überraschendes bereit: es verfügt über ein paar spezielle Be-

fehle für die bereits eingangs angedeuteten Steuerungsaufgaben.

Unmittelbare Freude hat man als Besitzer der Grundversion zunächst aber nur am Befehl NETZ und GEN. Mit ersterem schaltet man die Steckdose in den jeweils anderen Zustand um, mit GEN kann man Rechtecksignale erzeugen (dieser Befehl ist auch über das Hauptmenü (F4) zugänglich). Unter Angabe von Periodendauer und Anzahl der Schwingungen darf man das Ergebnis über den eingebauten Piezowandler live erleben und kann es über Pin 5 des Kassetten-Interfaces abgreifen.

Wer mehr steuern, regeln oder messen will, der muß nunmehr zu den Steckmodulen greifen (und zur Brieftasche, versteht sich).

(Ein-)Schubweise

Zum Test verfügten wir über folgende Steckmodule:



Bild 1. Eine Netzsteckdose für Schaltuhrbetrieb ist eingebaut

- DIN/DOUT Acht Relais, einzeln als Ausgänge ansprechbar. Die Schaltkontakte sind über eine Canon-Buchse herausgeführt. Acht digitale Eingänge.
- AIN Acht analoge Eingangskanäle (mit AD 7581) über Canon-Buchse.
- AOUT Zwei analoge Ausgabekanäle (mit ZN 426) über Canon-Buchse.
- EPROM Zwei leere, aber beschaltete Stecksocket für EPROMs.

Erst mit diesen Modulen kom-

men die Vorzüge des MAX und seiner steuerungsspezifischen BASIC-Befehle zum Tragen. Normalerweise erfordern auch so einfache Aufgaben, wie das Abfragen von einzelnen Leitungszuständen (bitweise!) oder das Betätigen von Relais, recht umfangreiche Maschinenprogramme. In BMM-BASIC braucht man nur einen Befehl.

Mit DOOUT Kanalnummer:
Schaltzustand
ist alles bereits erledigt. Mit

DIN Kanalnummer:
Numerische Variable
holt man sich den Wert einer Eingangsleitung in eine Varia-

DER KMT ODER DAS KMT?

Beides: Der Klein-Modul-Träger ist ein zweiteiliges Gehäusemagazin mit einem neuen Konzept – Funktion vorne. Wärme hinten.

Das KMT-System, für kompakte Elektronik mit kleinem Volumen in Stecktechnik, umfaßt Frontplatten, Karten, Stecker, Busplatten, Netzteile u.v.m. Bei erstaunlichem Preis-Leistungs-Verhältnis bietet es dem Elektroniker viel Zeit fürs Eigentliche.



Zwei Schrauben und ein Profil halten wärmeerzeugende Leistung auf störfreiem Abstand zur Funktionselektronik und bieten besten Zugang zur Verdrahtungszone.

Steckmodule mit Raslverriegelung im E- oder 1/2 E-Format in Karten oder Chassis-Ausführung.



KMT erhalten Sie im Fachhandel zum vernünftigen Preis

Anwendungen



ble. Jede Variable repräsentiert dann allerdings nur ein Bit.

Sehr einfach lassen sich auch die A/D- beziehungsweise D/A-Wandler ansprechen. Mit

AOUT Kanalnummer:
Numerischer Ausdruck

lassen sich beliebige 8-Bit-Werte ausgeben. Wenn in einer Schleife, die Sinuswerte errechnet, jeweils eine Analogausgabe eingebaut wird, verfügt man über einen programmierbaren Sinusgenerator.

Der achtkanalige Analogeingang läßt auf die gleiche einfache Art die Durchführung von mehreren Meßvorgängen zu. Was es also auch sei, einfache Steuer-, Meß- und Regelprobleme sind mit wenigen BASIC-Befehlen abgetan.

Diese Merkmale machen MAX auch für industrielle Anwendungen interessant, wo er zur Entwicklung von Steuerungen oder zum schnellen Ausprobieren bestimmter Sachverhalte ausgezeichnet geeignet ist. Und selbstverständlich erleichtert seine Konzeption auch gerade dem Anfänger den Umgang mit hardwaremäßigen Anwendungen seines Computers.

Aber nicht nur eitel Freude gab es speziell mit den Steckmodulen. Sie haben keine Verriegelung zum Hauptgehäuse, so daß man beim Abziehen des Caron-Steckers üblicherweise das ganze Modul herauszieht. Bei laufendem Gerät kann das mißliebige Folgen haben. Der Mangel wurde aber herstellenseitig erkannt und wird künftig durch eine Schraube behoben. Auch mit den Führungsschienen für die Module machten wir sogleich Bekanntschaft. Nur auf Lack geklebt, fielen uns gleich zwei in die Hände. Im Kunststoffgehäuse sicher kein Problem, beim Metallgehäuse sind aber Verschraubungen nötig.

MAX ist noch sehr jung, er wird noch wachsen. Seine Produktion läuft gerade ein halbes Jahr. Alles, auch die gesamte Betriebssoftware, ist eine Eigenentwicklung der Herstellerfirma. Inzwischen wird bereits ein Video-Interface angeboten, und ein Digitizer ist in Vorbereitung. Was man natürlich schmerzlich vermißt und was auch ganz sicher bald angeboten werden wird, ist ein EPROM-Programmierer.

Die Selbstbauseminare sind zweifellos eine bemerkenswerte Neuerung. Sie gehören nzw-

Schulung und Hilfestellung

schen obligat zum Konzept des MAX. Das heißt ganz konkret: MAX gibt es nur fertig oder als Bausatz einschließlich Seminar. Der 'Zwangs'-Charakter dieser Seminare wird zwar nicht jeden Selbstbauer erfreuen, aber viele Probleme eines Bausatzanbieters, wie Rücksendungen angeblich defekter Bauteile, treten gar nicht erst auf.

Das Handbuch vor MAX umfaßt derzeit knapp 70 DIN-A4-Seiten Text. Nach acht Seiten 'Erste Schritte' folgen gut 20 Seiten vertiefende Informationen über den Rechner und den Umgang mit ihm. Weitere 16 Seiten beschreiben das BMM-BASIC (jeder Befehl wird einzeln mit mehreren Beispielen erklärt), und auf 15 Seiten wird auf die Steckmodule eingegangen. Dazu sind vier konkrete Anwendungsbeispiele angegeben, wie beispielsweise eine Innenwiderstandsbestimmung.

Bestückungspläne und Steckerbeziehungsweise IC-Anschlußbelegungen sind derzeit noch in 'ASCII-Grafik' dargestellt, damit sie (laut Hersteller) per Textverarbeitungssystem leicht änderbar bleiben. Gelegentlich geht das etwas zu Lasten der Übersichtlichkeit.

Lästiger aber sind so kleine Unterlassungsünden wie beim GEN-Befehl, wo man bei Angabe einer Periodendauer von unterhalb einer Millisekunde zwar die Meldung erhält 'Generator läuft', der Piezo aber schweigt. Man könnte ja meinen, daß die Ausgangsleitung Signal führt, und hier nur der Schallwandler (und die Nerven des GENERATORS) geschont werden sollen. Aber dem ist nicht so. Folglich lassen sich wohl keine Schwingungen oberhalb 1000 Hz erzeugen.

Solche Probleme mit der Dokumentation werden vermutlich durch die Seminare weitgehend aufgefangen, über deren Qualität wir allerdings keine Aussagen machen können, da wir nur über ein Fertiggerät verfügten. Aber schließlich wird ja garantiert, daß man nicht ohne funktionierendes Gerät nach Hause gehen muß.

Wirklich ärgerlich ist die bewußt betriebene 'Verschleierraktik' bezüglich der Schaltpläne. Da immerhin alle Schnittstellen mit Pin-Belegungen von Steckern und Port-ICs 'preisgegeben' werden, könnte sich der Fertiggeräte-Besitzer viele Anfragen sparen, wenn zumindest ein Blockschaltbild mit eben diesen Schnittstellenbezeichnungen vorhanden wäre.

Auf Anfragen, warum man denn einem Selbstbau-Computer kein Schaltbild beilege, wurden Bedenken wegen Raubkopien à la Apple geäußert (was von Selbstbewußtsein künde). Ausdrücklich ausgeschlossen wurde eine künftige Veröffentlichung allerdings nicht.

Was man anlegen muß

Die abgemagerte Grundversion (4K RAM, geringfügig verringertes Betriebssystem/BASIC, kein Centronics-Stecker und kein Akku, Kunststoffgehäuse) kostet einschließlich Selbstbauseminar unter 800 D-Mark. Über die Hälfte davon machen (laut Herstellerangabe) LC-Display, Schreibmaschinen-tastatur, Gehäuse und Seminar aus. Zumal MAX nicht aus fernöstlicher Massenproduktion stammt, kann man den Preis wohl als angemessen bezeichnen.

Rund 1000 D Mark kostet die voll ausgestattete Selbstbauversion (ohne Steckmodule), ebenfalls im Kunststoffgehäuse. Etwa 1900 D-Mark muß man allerdings hinblättern, will man die uns vorliegende Version komplett mit Steckmodulen erwerben. Und es ist kein Seminar im Preis enthalten.

Fazit

Wer sich als Einsteiger in die Computerei einen MAX zulegt,

verfügt über ein Gerät mit komfortabler Tastatur, magerem, allerdings auch narrensicherem BASIC und — wenn es gar nicht klappt mit dem Computern — über eine intelligente Digitaluhr mit Schaltvorrichtung.

Aber auch Profis brauchen sich nicht zu genieren, einen MAX zu erwerben. Wenn auch manchen eher maschinensprachorientierten Interessenten der Prozessor 'zu exotisch' sein mag, um sich darauf einzuarbeiten: für die meisten Probleme wird man mit dem Steuerungs-BASIC auskommen.

Dennoch, ein Bausatzgerät, zu dem kein Schaltplan geliefert wird, stellt schon ein Kuriosum dar. Nicht nur, daß es das grundsätzliche Verständnis der Hardware erschwert, auch eventuell anfallende Reparaturen, die beim Basteln nicht ausbleiben werden, sind fast unmöglich oder müssen vom Hersteller (Kosten!) durchgeführt werden.

Die kleinen Fehler an der Mechanik sind zwar unerfreulich, lassen aber ganz sicher keine Rückschlüsse auf gravierende konzeptionelle Fehler zu. Der optische Eindruck von Innenleben und Gehäuse ist arsonsten ausgesprochen positiv.

Man sollte sich allerdings genau überlegen, ob sich die Anschaffung des Fertiggerätes lohnt: Erstens ist es beachtenswert teurer als die Selbstbauversionen, zum anderen bietet das Seminar eine derzeit wohl noch unverzichtbare Ergänzung zur Dokumentation.

Wenn MAX all das hält, was sich sein Hersteller von ihm verspricht, wird dieser Beitrag sicher nicht der letzte über ihn bleiben. In diesem Sinne: MAX vobiscum! □

Ergebnisse auf einen Blick:

- | | |
|---|--|
| ⊕ Ergonomische deutsche Tastatur | ⊖ Kein Schaltbild für Selbstbauer |
| ⊕ Insgesamt solide verarbeitet | ⊖ Lücken im Handbuch |
| ⊕ Selbstbau-Unterstützung durch Seminar | ⊖ Schlecht verklebte Modulführungsschienen |
| ⊕ Sehr einfach erweiterbar (Steckmodule) | ⊖ Module bisher noch nicht verriegelt |
| ⊕ Spezielle BASIC-Befehle für Steueraufgaben | ⊖ Mageres BASIC |
| ⊕ Komfortabler Editor, gründlicher Syntax-Check | |

NEU

HX-20 & CP/M®

HX-20-Video-Adapter jetzt

Auch in Farbe!

die komfortable Verbindung zum Monitor!

7x10 Punkt-Matrix, gestrichelte scharfe Anzeige mit Unterlängen. Visueller Bildschirm: 80 Zeilen x 24 Spalten. Alle nationalen Zeichensätze wie HX-20 (alle Editierfunktionen).

STOP

Kompletter HX-20-Zeichensatz (inkl. Grafik + zusätzl. Zeichen), sämtliche Steuerbefehle, umschaltbar per Programm und Tastatur. Nahezu alle Programme am Monitor ohne Änderung lauffähig.

HX-20-Floppy-Set (bis 1,6 MB)

1—2 Laufwerke, je 380—730 KB, voller HX-20-Befehlssatz, Video-Adapter und Floppy in gleichen oder separatem Gehäuse. CP/M®-Betriebssystem, zusätzlich CP/M®-Programme einsetzbar.

CP/M ist ein Warenzeichen der Digital Research, Inc.

time-soft-edu®

Sophienstraße 32 · 7000 Stuttgart 1 · Telefon: 071/22 84 71/72
Programme + Computer für zeitgemäße Anwendungen

Columbia/IBM-PC/Sirius/Victor/Wang

Die

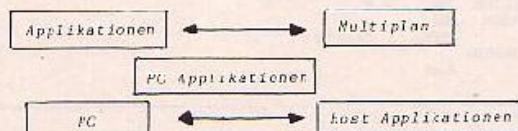
UNIVERSELLE

Software

SCHNITTSTELLE

für PCs

FILE MASTER ermöglicht den problemlosen Datentransfer zwischen



Einfachste Bedienung dank der integrierten Bildschirm-Benutzerführung.
Deutscher Dialog und Dokumentation.

H+I Informatik Aktiengesellschaft
Schwefelstraße 33, FL-9490 Vaduz, Telefon 075/2 42 94

MICROCOMPUTER



ATARI 600XL
ATARI 800XL

Für dieses Computersystem gibt es bereits heute über 2000 Programme!

Wir sind autorisierter Händler für die gesamte ATARI Computer Palette. Guter Geräteservice vorhanden. Groß-Servicestelle v. Atari in nächster Nähe.

Die neue SUPERPALETTE von ATARI

ATARI 600XL, 16K RAM, 24K ROM, (inkl. ATARI BASIC), Netzgerät, dt. Dokumentation 549,— DM

ATARI 54 MODUL, Spielereiweiterung f. Atari 600XL 349,— DM

Technische Daten wie unten.

ATARI 800XL, 64K RAM, inkl. Atari-Basic, Netzgerät, dt. Beschreibung 899,— DM

Technische Daten: 24K ROM, eingeb. BASIC, 256 Farben, davon 16 in BASIC gleichzeitig darstellbar, 320 x 192 Bildpunkte, 21 Zeilen mit 40 Zeichen, 5 Textmodi und 11 Graphikmodi, 4 voneinander unabhängige Tonkanäle (Tonumfang: 3 1/2 Oktaven), Schreibmaschinenartatur mit 4 speziellen Funktions- und internationalen Zeichensätzen (11 Um- laute auf Bildschirm sichtbar mit Graphik- zeichen, Map-Taste für Rückfragen und Zwischenliste in verschiedenen Programmen, CPU: 6502 C, 3 spezielle Prozessoren für Graphik, Ton und Bild)

ATARI 1050 Diskettenstation 999,— DM

Gle. 127 KB pro Diskette (ca. 100 Schreib- machinenformat), dt./engl. DOS II, Güteschein für DOS III, Wird Betriebsbereit geliefert (Netzgerät, Datenkab., dt. Besch.) Anschluß über eingebauten Interface.

Lernset 1011 (inkl. Facorier, 2 Lernprogr., kompl. mit Anschluß (Netzteil, Kabel) 249,— DM

ATARI 1027 Drucker 899,— DM

Schreibendrucker mit Interface & Kabel

ATARI 1025 Matrixdrucker 1.399,— DM

Inkl. Kabel, Netzteil, Interface

ATARI 1020 Farbdruker 769,— DM

Dieser Drucker ermöglicht Superlatgraphik. Inkl. Interface, Kabel, Netzteil, Progr. Cass.

Track-Ball f. schnelles präzises Spielen 179,—

AUS UNSERER BUCHERECKE

ATARI BASIC (deutsch) Das ideale Buch für den Computer-Neuling mit vielen Programmbeispielen. 39,— DM

Das große Spielbuch für ATARI 600/800 XL 20 Spiele nur 29,80

Min ATARI-Computer 56,— DM

Das beste Buch über den ATARI: Die ATARI wird von allen Seiten her durchleuchtet. Hardware, Peripherie, BAS C-Einführung, fortgeschrittene Programmertechniken, Graphik, Sound u. vieles mehr. Auch neue XL-Typen werden behandelt!

Die Trickkiste 39,— DM

Neue Tips, Tricks und Programme für ATARI

ATARI BASIC spielerisch lernen 10,80 DM

101 ATARI Computer Programming 29,— DM

Tips & Tricks

31 New ATARI Computer Programs 29,— DM

2 Bücher m. einer Vielzahl interessanter Progr.

Einführung in LOGO 42,— DM

Eine Programmiersprache speziell für Graphik- darstellung auf dem Bildschirm (bald auch für ATARI erhältlich)

Der ATARI-Assembler 36,— DM

Durchsichtige Buch über Assemblerpro- grammierung auf dem ATARI.

The Best of Software 70,— DM

Die besten Programme aus dem Computer- magazin "Software".

ATARI Sound und Graphics 49,— DM

Sound und Graphik für jeden ATARI-User

Mapping the ATARI 58,— DM

Alle wichtigen Adressen und Vektoren Ihres ATARI-Computers. Eines der wichtigsten Bücher um den Computer voll zu beherrschen.

ATARI Graphics 49,— DM

Viel Wissenswertes über Graphikprogram- mierung auf dem ATARI.

Machinelanguage for Beginners 49,— DM

Hervorragendes Buch zum Erlernen der 6502- Maschinensprache (330 Seiten 9).

ATEXT — Wortprozessor f. ATARI mit dt. Anleitung. Einer der besten Wortprozessoren weltweit! 148,— DM

(D) 159,— DM (ROM-Modul) 199,— DM

MÜNZENLOHER GMBH

Tölzer Straße 5
D-8150 Holzkirchen / Obb.
Telefon: (0 30 24) 18 14

Direkt an der Autobahn München - Salzburg.
Gerätebau, Computersysteme — SOFTWARE
Lieferung per NN od. Vorkasse a. Postcheck- Kto. 2845 58-807 München od. Eurocheck.

NEU SHARP MZ-731 NEU



Technische Daten:

Z-80A 8,5 MHz, 64 KB, ASCII-Textatur, Cursor-Steuertasten, Definierbare Funktions- tasten, Tasten zum Einfügen und Löschen (INS, DEL), Verwendung eines Fernseh- gerätes (PAL-System) oder einer Monitoran- zeige möglich. RGB-Anschluß, Lautsprecher u. Uhr-Funktion eingebaut, zusätzlicher Drucker und Floppy-Disk-Anschluß möglich.

Mit Sharp BASIC Interpreter, mit 2 Cassetten u. auf dt. Handbuch. Mit Cassetten-Facolor Vierfarbdrucker u. 10 Progr. nur 1395,—

NEUE BÜCHER

Schritt für Schritt mit MZ-700 von Prof. Dr. Günther O. Harmsen 29,80

Programmier-Handbuch für SHARP 49,—

Programme für SHARP MZ-700

PC-Graphiksystem (Modul) 360,— DM

Erlaubt Ansteuerung aller Bildschirmaspek- te und Modifiz. der Zeichensätze.

Miri HUBASIC-Compiler (C) 120,— DM

Urukt-700 (C) 120,— DM

Sehr komfortables Textverarbeitungsprogr.

Urbis (C) 120,— DM

Flexibles Datenverwaltungsprogramm

FORTAN-Compiler (C) 95,— DM

Programmauf. 50 mal schneller als in BASIC

Superdisassembler (C) 75,— DM

Super-ML C100 bzw. 5000 (C) 55,— DM

Zum Programmieren in Maschinensprache

Adressverwaltung (C) 68,— DM

Lagerverwaltung (C) 59,— DM

Terminverwaltung (C) 60,— DM

Börsenprogramm - wurde v. einem Börsen- fachmann entwickelt u. erstellt. Es ermöglicht Ihren Kauf- u. Verkaufstermine von Aktien zu bestimmen. Sie geben laufend u. stunden Aktienkurse ein. Der Rechner wertet diese Daten aus, druckt Graphiken aus, liefert Trendmeldungen, Kauf- u. Verkaufstermine u. sortiert alle eingegebenen Aktien nach ver- schiedenen Kriterien. 99,— DM

Über 100 Spielprogramme zwischen 20,— und 100,— DM f.d. SHARP stehen zur Verfügung.

HARDWARE

Original SHARP-Laufwerke m. 1 od. 2 Lauf- werken 5 1/4 Zoll, Doppelseitig, doppelte Schreiblichts, je 250 kb mit Betriebssystem.

Formschöner SHARP-Monitor MZ-1D05, RGB-Eingang m. brillanten FarbBild 1.425,—

Die 2 speziellen Hefts f. ATARI (ANTIC u. ANALOG) sowie COMPUTE a' 13,80 DM

BUSPACK

Das Super-Geschäftsprogramm für den Ver- triebsmann u. den Kleinbetrieb. Lager- (1000 Artikel) u. Adressenverwaltung (500 Adressen).

Auf Wunsch 1000 Adressen u. 500 Artikel. Rechnungsschreib-Program. Lager wird mitge- liefert mit Beispielen f. ATARI 499,— DM

Bücher für den C-64

Wirtschaft auf den C-64 38,— DM

40 Progr. aus d. Bereichen Finanzmathem., Unternehmensforschung u. Betriebswirtschaftl.

Graphik auf dem Commodore 64 38,— DM

BASIC auf dem Commodore 64 38,— DM

BASIC-Einführung u. Erläuterung spezifischer Eigenschaften.

Bedieneranl. Sie Ihren Commodore 64 19,80

64 Programme für den C-64 39,—

Hardware Erweiterung f. C-64 39,—

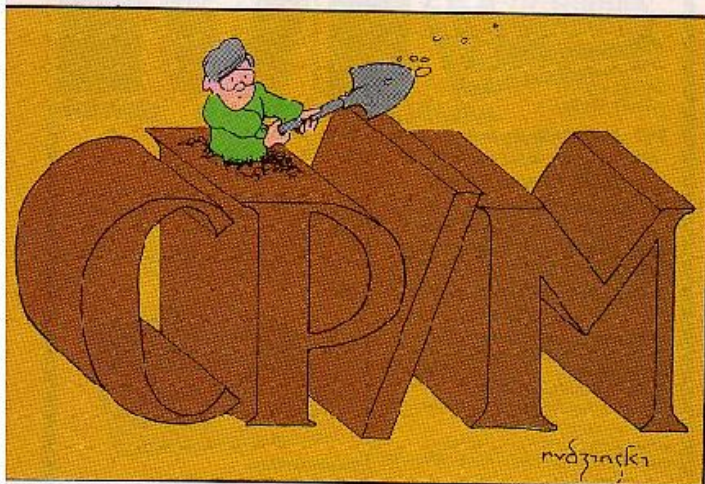
Progr. i. Maschinensprache m. C-64 29,80

Mehr als 29 Programme f. C-64 29,80

NEU — BLIZTEXT 1.1 für C-64 199,—

NEU — BUSPACK f. C-64 — Lager- Adressenverw. u. Fakturierung 299,— DM

Fordern Sie noch heute unseren Katalog speziell für den SHARP bzw. ATARI an! Wir freuen uns auch auf Ihren Besuch



Das Betriebssystem CP/M-86

Anpassung, Teil 2

Kurt Werner

Im ersten Teil dieses Artikels wurde gezeigt, wie das BIOS des CP/M-86 an die Hardware des c't 86 anzupassen ist. Damit sind die Vorarbeiten für den ersten Praxistest abgeschlossen. Dieser Artikel zeigt, wie ein lauffähiges Betriebssystem erstellt werden kann. Die dazu notwendigen Hilfsprogramme werden ebenfalls beschrieben.

Neben dem angepassten BIOS sind zur Erstellung einer bootfähigen Diskette noch drei Hilfsprogramme nötig: Ein Urlader, der das eigentliche CP/M von der Diskette in den Speicher lädt, ein Programm zum Kopieren der Systemspuren einer Diskette und ein Formatierprogramm, das eine Diskette erst beschreibbar macht. Mit diesen Programmen kann man eine Diskette mit einem lauffähigen CP/M-86 erstellen.

Der Urlader

Dieses Ladeprogramm steht auf der ersten Spur der Diskette. Beim Bootvorgang sorgt der Lader dafür, daß die ersten beiden Spuren gelesen werden. Dazu muß das Programm an das verwendete Laufwerksformat angepaßt sein. Die Definition des Programms kann für 5¼"-Drives und für 8"-Laufwerke vorgenommen werden.

Soll der Urlader für ein System mit 5¼"-Laufwerken generiert werden, ist im Source-Listing die Variable 'FUENF' auf 'TRUE' zu setzen. Für 8"-Drives muß die Variable den Wert 'FALSE' erhalten.

Die zweite Anpassung des Programms an die Laufwerke betrifft die Aufzeichnungsformate. Das Programm lädt immer die ersten beiden Spuren der Diskette in den Speicher, egal, ob es sich um ein einseitiges oder doppelseitiges Laufwerk handelt. Dieses Format kann durch den Wert der Variablen 'MAXSEC' bestimmt werden. Dabei wird lediglich die Anzahl der maximal verfügbaren Sektoren festgelegt: Bei 8"-Laufwerken (einseitig) 27 und bei zweiseitigen 8"-Drives 53. Diese Zahlen stellen die reale Sektorenanzahl minus eins (bedingt durch eine Inkrementation der Sektorenanzahl beim Programmaufruf) dar. Für 5¼"-Drives muß der Urlader an das Format des 'Boot-Laufwerks' angepaßt werden. Normalerweise kann bei einseitigen Laufwerken die Variable 'MAXSEC' auf 11, und bei doppelseitigen Drives auf 21 gesetzt sein.

Um den Lader mit den korrekten Laufwerksangaben zu generieren, muß folgende Befehlsfolge in den Rechner eingegeben werden:

```
ASM86 BOOT $PX
GENCMD BOOT
```

```
; B O O T - BOOTER FÜR CP/M-86
;
TRUE EQU -1
FALSE EQU 0
;
FLENF EQU TRUE
ACIT EQU NOT FUENF
;
; MONITORAUFRUF
;
HOME EQU 0A1
SETDMA EQU HOME+2
SETTRK EQU CTDMA+2
SETDRV EQU SETTRK+2
CTDSEC EQU SETDRV+2
SETMAS EQU SETSEC+2
READ EQU SEUMAS+2
;
; CSEG - 0F000H
; ORG 0E000H
;
; MOV LUN:
;
; CPMST EQU 0B10H ; STARTADRESSE VOM LOADER CP/M-86
; BOOTR EQU 0B000H ; STARTADRESSE DES INITIAL BOOTERS
; CPMST EQU 0H ; STARTSEGMENT VOM CP/M-86
; MAXTRK EQU 2H ; MAXIMAL 24H - 128 TRACKS
;
; IF FUENF
; CPMLO EQU 8200H ; STARTADRESSE DES ZU LADENDEN CODES
; OFSET EQU 200H ; 512 BYTES / SEKTOR
; MAXSEC EQU 21 ; 20 SEKTOREN AUF DER BOOTDISKETTE
; ENDF
;
; IF ACIT
; CPMLO EQU 8080H
; OFSET EQU 80H
; MAXSEC EQU 18H
; ENDF
;
; ; START DES CODES
;
; CSEG 0
; ORG 0
;
; MOV AX,CB
; MOV DS,AX ; DATENSEGMENT = CODESEGMENT
; MOV DI,ADDRESS,CPMLO ; INITIIEREN
;
; ; DMAADRESSE SETZEN
; MOV CX,CPMST
; MOV AL,SEDMAS
; CALLF MONITOR
;
; ; DIE ANDEREN PARAMETER SIND SCHON AUS
; ; DEM EPROM HERAUS GEBSETZT
;
; MOV TRACK,0 ; BEGIN : TRACK 0,SEKTOR 2
; MOV SECTOR,2H
; JNFB BOOTR
; MOV CX,SECTOR
; MOV AL,SETSEC
; CALLF MONITOR ; SEKTOR SETZEN
; MOV CX,ADDRESS
; MOV AL,GETDMA
; CALLF MONITOR ; UEBERTRAGUNGSBEREICH SETZEN
; MOV AL,READ
; CALLF MONITOR ; LEBEN
; ADD ADDRESS,OFSET ; NÄCHSTEN UEBERTRAGUNGSBEREICH
; ; FESTLEGEN
;
; JNC SEKTOR
; CMP SECTOR,MAXSEC ; LEITER SEKTOR ?
; JNZ BOOTL
; INC TRACK ; LETZTER TRACK ?
; CMP TRACK,MAXTRK
; JZ BOOTR
; MOV SECTOR,1 ; NÄCHSTEN TRACK SETZEN
; MOV CX,TRACK
; MOV AL,SETTRK
; CALLF MONITOR
; JMPF BOOTL
; JNFB DATA
; DATA EQU OFFSET $
;
; ; DSEG - 0F000H
; ; ORG 0F000H
; ;
; ADDRESS DW 0
; SECTOR DW 0
; TRACK DW 0
; CSEG CPMST
;
; LOADER:
; CND
```

SCOPY-Programm

Dieses Programm ermöglicht es, Dateien auf die Systemspuren einer Diskette zu kopieren. SCOPY entspricht somit dem SYSGEN-Programm des CP/M-80. Da der c't 86 auch

mit 5¼"-Laufwerken betrieben werden kann, ist das Programm 'LDCOPY' von DIGITAL RESEARCH nicht verwendbar: LDCOPY funktioniert nur bei 8"-Drives.

Das Programm 'SCOPY' muß, bevor es eingesetzt wird, an die


```

1 SYSTEMCOPY
2
3 TIME EQU -1
4 FALSE EQU NOT TRUE
5
6 FUENT EQU TRUE
7 ACHT EQU TRUE
8
9 CS=0
10 DPC 05CH
11
12 ; DEFINITION DES UEBERGEHENDEN PARAMETERS
13 ;
14 PER:
15 ;
16 ; BEGINN DES PROGRAMMS
17
18 ORG 0100H
19
20 ; DS=CS
21 MOV AX,CS
22 MOV DS,AX
23 MOV DX,OFFSET STAR
24 MOV CL,9
25 INT 274
26 MOV WRITE,0
27 MOV BX,OFFSET FCB
28 MOV AL,10BH
29
30 ; PARAMETER VORHANDEN ?
31 CMP AL,0CH
32 JZ READTO
33
34 ; JA, EINLESEN EINES FILES
35 = CALL READFILE
36 LOOP: MOV CL,9
37
38 ; SCHREIBEN AUF DISKETTE
39
40 MOV DX,OFFSET DESTH
41 INT 274
42 MOV CL,1
43 INT 274
44 CMP AL,05H
45 JNZ NOABORT
46
47 ; ENDE
48 FINI: MOV DL,0
49 MOV CL,0
50 INT 274
51
52 NOABORT: CMP AL,45H
53 JA LOOP
54 CMP AL,41H
55 JB LOOP
56 SUE AL,41H
57 MOV DRIVE,AL
58 MOV WRITE,1
59
60 ; AUF AUSGEWAELTEM DRIVE SCHREIBEN
61 CALL WRITETO
62 JMP LOOP
63
64 ; READTO: CALL READT
65 JMP LOOP
66
67 ; SCHREIBEN ODER LESEN AUF SYSTEMSPUREN
68 ;
69 READT: MOV DX,OFFSET SOURCE
70 MOV CL,9
71 INT 274
72
73 ; DRIVE EINLESEN
74 MOV CL,1
75 INT 274
76 CMP AL,3
77 JNZ NOAB
78 MOV DL,0
79 MOV CL,0
80 INT 274
81
82 NOAB: CMP AL,41H
83 JB READTO
84 CMP AL,45H
85 JA READTO
86 SUB AL,41H
87 MOV DRIVE,AL
88
89 ; BERECHNUNG DER KLEINSTEN UND DER GROSSTEN SPUR
90 WRITETO: MOV AL,DRIVE
91 ADD AL,AL
92 MOV BX,OFFSET DRVTABLE
93 MOV AH,0
94 ADD DX,AX
95 MOV CX,0BH
96 MOV MINSEC,CL
97 MOV CURSEC,0
98 MOV MAXSEC,CH
99
100 ; AUFWERK SETZEN
101 MOV C,DRIVE
102 MOV AL,9
103 CALL BIOS
104 MOV CURTRACK,0
105 MOV AL,8
106
107 ; HUNT
108 CALL BIOS
109 MOV CX,CS
110 MOV AL,17
111
112 ; DATENSEGMENT IST DAS CURRENT CODE SEGMENT
113 CALL BIOS
114 MOV AX,OFFSET DATN+80H
115 MOV ADDRESS,AX
116 READT1: MOV AL,11
117 MOV CL,CURSEC
118
119 ; SEKTOR AUSGEESEN
120 CALL BIOS

```

```

121 MOV CX,ADDRESS
122 MOV AX,10
123 ; DMA-ADRESSE AUSGEESEN
124 CALL BIOS
125 MOV AL,13
126 ; LECCH ODER GOINCIDEN
127 CMP WRITE,0
128 JZ READTO
129 MOV AL,14
130 READT2: MOV CL,0
131 CALL BIOS
132 JN AL,AL
133 JZ NERROR
134 ; FEHLEN BEIM LESEN ODER SCHREIBEN
135 NEXT1: MOV CL,9
136 MOV DX,OFFSET ERROR
137 INT 274
138 MOV CL,1
139 INT 274
140 CMP AL,"N"
141 JNZ M2
142 JMP FINI
143 M2: CMP AL,"I"
144 JNZ NEXT1
145
146 NERROR: MOV ATD ADDRESS,80H
147 INC CURSEC
148 MOV AL,CURSEC
149 CMP AL,MAXSEC
150 JLC READT1
151
152 ; ZWEITE SPUR ?
153 CMP CURTRACK,0
154 JNZ RETUP
155 INC CURTRACK
156 MOV CX,1
157 MOV AL,10
158
159 ; SPUR 1 SETZEN
160 CALL BIOS
161 MOV AL,MINSEC
162 MOV CURSEC,AL
163 JMP READT1
164
165 RETUP: RET
166
167 ; READ SYSTEM FROM FILE
168 ;
169 READFILE: MOV CL,15
170 MOV DX,OFFSET FCB
171 MOV BX,9
172
173 ; FILE EXTENSION - CHD
174 MOV ECX,43H
175 INC BX
176 MOV ECX,40H
177 INC BX
178 MOV ECX,44H
179
180 ; OPEN FILE
181 INT 274
182 CMP AL,255
183 JZ END
184 MOV CL,51
185 MOV DX,CS
186
187 ; DMA DECMONT ADDRESS - CS
188 INT 274
189 MOV AX,OFFSET DATN
190 MOV ADDRESS,AX
191 READB: MOV DX,ADDRESS
192
193 ; DMA-ADRESSE SETZEN
194 MOV CL,26
195 INT 274
196 MOV CL,20
197 MOV DX,OFFSET FCB
198
199 ; LESEN
200 INT 274
201 OR AL,AL
202 JNZ READB
203 ADD ADDRESS,80H
204 JMP READB
205
206 ; FILE ENDE
207 READC: MOV CL,14
208 MOV DX,OFFSET FCB
209
210 ; CLOSE FILE
211 INT 274
212 RET
213
214 ;
215 ; ERROR : FILE NICHT DA
216 ;
217 END: MOV DX,OFFSET FILENICHTDA
218 MOV CL,9
219 INT 274
220
221 ; AUSGANG ZUM BETRIEBSSYSTEM
222 MOV DL,0
223 MOV CL,0
224 INT 274
225
226 ; AUSGABE AN DAS BIOS
227 ;
228 BIOS: MOV TABLE,AL
229 MOV CX,CS
230 MOV DX,DX
231 MOV CL,50
232 MOV DX,OFFSET TABLE
233 INT 274
234 RET
235
236 ; DATENSEGMENT
237 ;
238 DATA EQU OFFSET 4

```



```

;-----
; CSEG
; ORG DATA
;
; DRIVE DE 0
; CURSEC DE 0
; MINSEC DE 0
; MAXSEC DE 0
; JFF DW 0
; CURTRACK DD 0
; WRITE DE 0
; TABL E BP 0
; CXDES DW 0
; CXDES DW 0
; ADDRESS DW 0
; DRIVETABLE RW 0
;
; MINIMALE UND MAXIMALE COCKTODEN
;
; IF ACHT
; DE 1,1AH
; DE 1,1BH
; ENDIF
;
; IF FUENF
; DE 0,79
; DE 0,79
; DE 0,63
; DE 0,63
; ENDIF
;
; START DE 13,0
; DE "SYSTEMGENERIERUNG VERSION 1.0"
; DE 13,0,0
; SOURCE DE "BITTE SOURCE DRIVE AUSWAHL (A-F)?"
; DE 13,10,0
; DESTW DE 13,0
; DE "BITTE DESTINATION DRIVE AUSWAHL (A-F)?"
; DE 13,10,0
; FILENICHIA DE "FILE IST NICHT AUF DER DISKETTE"
; DE 13,10,0
; ERROR DE "SCHREIB- ODER LESEFEHLER. WEITER (Y/N)?"
; DE 13,10,0
; DATEN RS 5100H
; DE 0
; END

```

vorhandene Systemkonfiguration angepaßt werden. Dazu sind im Source-Listing die Variablen 'FUENF' und 'ACHT' dem System entsprechend zu setzen (siehe auch Urlader). Danach ist die Drivetabelle an das System anzupassen. In der Tabelle stehen jeweils die kleinste und die größte Sektornummer, auf der das Betriebssystem steht. Die angegebenen Werte gelten für zweiseitige 5¼"-Laufwerke und einseitige 8"-Drives. Bei anderen Laufwerken sind die Werte durch zwei zu teilen (5¼"-Drives), beziehungsweise zu verdoppeln (8"-Drives). Nach der Anpassung muß das Programm assembliert werden. Dazu dient folgende Befehlsfolge:

```

ASM86 SCOPY SPX
GENCMD SCOPY 8080

```

Danach steht das lauffähige Programm bereit.

SCOPY bietet nun zwei Möglichkeiten, den zu kopierenden File zu bestimmen. Zum einen kann beim Programmaufruf der Name einer Datei angegeben werden. Es muß sich dabei um eine Systemdatei mit der Extension '.CMD' handeln. Diese Datei wird dann eingelesen und anschließend in die ersten beiden Spuren der Diskette geschrieben. Dieser Vorgang ist notwendig, wenn ein neues Betriebssystem generiert werden

soll. Verfügt man über ein lauffähiges Betriebssystem auf einer Diskette, kann es eingelesen und dann in die Spuren einer anderen Diskette kopiert werden. Um diesen Vorgang auszulösen reicht es aus, SCOPY ohne Angabe eines Parameters aufzurufen. Das Programm fragt dann nach dem Laufwerk von dem kopiert werden soll, anschließend nach dem Bestimmungslaufwerk. Dabei kann der Programmablauf jederzeit durch die Eingabe von 'Control C' abgebrochen werden.

FORMAT

Da wohl die Mehrzahl der c't-86-Interessierten 5¼"-Laufwerke einsetzen wird, stellen wir ein Formatierprogramm für dieses Floppyformat vor. Bei 'FORMAT' entfällt zwar die Auswahl des verwendeten Laufwerktyps, eine Anpassung an das System muß trotzdem erfolgen. Dazu sind die Variablen SINGLE, DOUBLE, SEC MAX, TRK MAX und STEP entsprechend dem System zu setzen. SINGLE und DOUBLE bestimmen, ob einseitige oder zweiseitige Laufwerke bedient werden sollen. Zur Anpassung ist die entsprechende Variable auf den Wert 'TRUE' zu setzen, die andere auf den Wert 'FALSE'.

SEC MAX gibt die Anzahl der

```

;-----
; FORMATIERPROGRAMM
;
; TRUE EQU -1
; FALSE EQU NOT TRUE
;
; SINGLE EQU FALSE
; DOUBLE EQU NOT SINGLE
;
; KONSTANTEN, SYSTEMABHÄNGIG
;
; SEC MAX EQU 10
; TRK MAX EQU 80
; STEP EQU 0
;
;
; DREHZAHL EQU 066H
;
; KONSTANTEN, SYSTEMUNABHÄNGIG
;
; KAE EQU 04EH
; KFE EQU 0F6H
; KFS EQU 0F5H
; KFC EQU 0FCH
; KF7 EQU 0F7H
;
; FLOPI EQU 0BAH
; FLOPD EQU 0B6H
; INT2 EQU 03EH
; REFPRT EQU 0BCH
; FLOPD EQU 0B9H
; FLOPC EQU 0B0H
;
;
; CSEG
; ORG 0100H
;
; START: MOV DX,OFFSET STARTMESS
; MOV CL,5
; INT 224
;
; AUSWAHL DER OPTIONEN
; MAXIMALER SEKTOR
; IF SEC MAX EQ 8
; MOV DX,OFFSET SEKS
; ENDIF
; IF SEC MAX EQ 9
; MOV DX,OFFSET SEKS
; ENDIF
; IF SEC MAX EQ 10
; MOV DX,OFFSET SEKS
; ENDIF
;
; MAXIMALE TRACK
; MOV CL,5
; INT 224
; IF TRK MAX EQ 40
; MOV DX,OFFSET TR40
; ENDIF
; IF TRK MAX EQ 80
; MOV DX,OFFSET TR80
; ENDIF
;
; ANZAHL DER SEITEN DER DISKETTE
; MOV CL,5
; INT 224
; IF SINGLE
; MOV DX,OFFSET S11
; ENDIF
; IF DOUBLE
; MOV DX,OFFSET S12
; ENDIF
; MOV CL,5
; INT 224
; MOV DX,OFFSET STARTME
; MOV CL,5
; INT 224
;
; DRIVENET: MOV CL,1
; INT 224
; CMP AL,C ;CONTROL C?
; JNZ FINI1
; JMP FINI
;
; DRIVE ?
; FINI1: CMP AL,41H
; JNGE DRIVENET
; CMP AL,44H
; JNGE DRIVENET
;
; DRIVEFOUND: SUB AL,41H
; MOV AH,0
; MOV DRIV,AX
; MOV EX,AX
; MOV AL,TRUE
; OUT FLOPD,AL
; ANFANG: MOV DX,OFFSET ORLE
; MOV CL,5
; INT 224
;
; HOME
; DECIN: MOV AL,C
; OUT FLOPD,AL
; MOV TRACK,0
;
; SPUR SETZEN
; LOOP: MOV AL,TRACK
; OUT FLOPD,AL
; MOV AL,10H
; OUT FLOPD,AL
;
; SPURNUMMER AN TERMINAL AUSGEBEN
; MOV AL,TRACK
; MOV DL,10
; DIV DL
; PUSH AX
; CALL B1HEX
; POP AX

```



```

MOV AL, AH
CALL BIHEX
; CUPSOR ZURUECKSETZEN
MOV CL, 2
MOV DL, 8
INT 224
MOV DL, 0
MOV CL, 2
INT 224
; FLOPPYCONTROLLER FERTIG ?
WAIT1: IN AL, FLOPC
TEST AL, 1H
JNZ WAIT1
CALL MAKETRACK
CALL WRITETRACK
INC TRACK
CMP TRACK, TRKMAX
JNZ LOOP
; ZWEIFLASEIGE DISKETTE
IF DOUBLE
CMP S.DE, 1
JZ ENDE
MOV S.DE, 1
JMP BEGIN
ENDIF
; ENDE
ENDE: MOV DX, OFFSET ENDEMESS
MOV CL, 9
INT 224
MOV S.DE, 0
MOV CL, 1
INT 224
CMP AL, 04AH
JZ ANFANG
CMP AL, 'N'
JNZ ENDE
FINI: MOV CL, 0
MOV DL, 0
INT 224 ; ENDE
; AUSGABE EINER ZIFFER AN TERMINAL
;
BIHEX: AND AL, 0FH
CMP AL, 04H
JL BIHEX
ADC AL, 7
BIHEX: ADD AL, 30H
MOV DL, AL
MOV CL, 2
INT 224
RET
; AUFRUFEN EINER SPUR
; FORMATAUFBAU SIEHE DATENSLATT 2797
;
MAKETRACK: MOV CURSEC, 1
MOV BX, 0
MOV AL, K4E
MOV CX, 80
CALL DUP
MOV AX, 0
MOV CX, 12
CALL DUP
MOV AX, KFA
MOV CX, 3
CALL DUP
MOV SEKANF[BX], KFC
INC BX
MOV AL, K4E
MOV CX, 80
CALL DUP
MOV AL, 0
MOV CX, 12
CALL DUP
MOV AX, KFE
MOV CX, 3
CALL DUP
MOV SEKANF[BX], 0FH
INC BX
MOV AL, TRACK
MOV SEKANF[BX], AL
INC BX
MOV AL, SIDE
MOV SEKANF[BX], AL
INC BX
MOV AL, CURSEC
MOV SEKANF[BX], AL
INC BX
MOV SEKANF[BX], 2
INC BX
MOV SEKANF[BX], KF7
INC BX
MOV AL, 04EH
MOV CX, 22
CALL DUP
MOV AL, 0
MOV CX, 12
CALL DUP
MOV AL, 0FCH
MOV CX, 3
CALL DUP
MOV SEKANF[BX], 0FBH
INC BX
MOV AL, 0ECH
MOV CX, 512
CALL DUP
MOV SEKANF[BX], 07H

```

```

INC BX
MOV AL, 04EH
;
; VARIABLE IST DER ABSTAND DER SEKTOREN
; VONEINANDER AUF DER DISKETTE
; 10 SEKTOREN = 24 BYTES
; 9 SEKTOREN = 66 BYTES
; 8 SEKTOREN = 120 BYTES
;
MOV CX, 22*SECMAX*SECMAX - 450*SECMAX+2406
CALL DUP
INC CURSEC
CMP CURSEC, SECMAX
JLE MLOOP
MOV AL, 04EH
MOV CX, 1098
CALL DUP
REI
;
; DUPLIZIEREN VON SPURHEADBEREICHEN
;
DUP: MOV SEKANF[BX], AL
INC BX
LOOP DUP
RET
;
; WRITETRACK:
; WRITE PROTECTED DISK ?
IN AL, FLOPC
TEST AL, 4CH
JZ WRYO
MOV DX, OFFSET WRITE
MOV CL, 9
INT 224
MOV CL, 0
INT 224
; DRIVE READY
WRYO: TEST AL, 80H
JZ WRY1
MOV DX, OFFSET READY
MOV CL, 9
INT 224
MOV CL, 0
INT 224
; WARTEN AUF INDEXLEDCH
WRY1: IN AL, FLOPI
TEST AL, B
JNZ WRY1
IN AL, FLOPI ; ENTPRELLEN
TEST AL, 0
JNZ WRY1
; WARTEN AUF NACHSTER INDEXLEDCH
MOV AX, DREHZAH
HL1: CALL WIMS
DEC AX
JNZ HL1
MOV DL, 2
MOV DH, 1
MOV BX, DTU
MOV AL, DRIVES[BX]
LEA BX, SEKANF
AND AL, 7FH
MOV DI, 0
CMP S.DE, 0
JZ NODR
MOV DI, 2
NODR: OR AL, 4CH
OUT FLOPI, AL
AND AL, 3FH
OUT FLOPI, AL
CLI
MOV AL, 0FBH
OUT INT2, AL
MOV AL, 0F0H
STI
XOR AX, DI
OUT FLOPC, AL
OUT REPORT, AL
WLOOP: IN AL, FLOPC
TEST AL, DI
JZ WREND
TEST AL, DL
JZ WLOOP
MOV AL, [BX]
OUT FLOPI, AL
INC BX
JMP WLOOP
WREND: PUSH AX
PUSH DS
; INTERRUPT FREISCHALTEN
MOV AL, 0F00H
MOV DS, AX
MOV AL, INTERRU
PUSH DS
OUT INT2, AL
OUT REPORT, AL
POP AX
TEST AL, 1CH
JZ NODR
MOV DX, OFFSET ERROR
MOV CL, 9
INT 224
MOV AX, 80
CALL WIMS
DEC AX
JNZ WAIT
JMP WRITETRACK
NODR: RET

```



```

#IMS: MCV EX,290
#IMS9: LCOP #IMS0
RET
ENDE ECU OFFSET 4
$
DEEG
DRIVES DE ENDE
DE 1H ;DOUBLE DENSITY
DE 2H ;DOUBLE DENSITY
DE 4H
DE 8H
DRIV DW 0
TRACK DE 0
CURSEC DD 0
SIDE DB 0
STARTMESS DE 13,10
DB "FORMATIERPROGRAMM FUER 5 1/4 ZOLL DISKETTEN"
DB 13,10,"$"
SEK8 DB "ACHT SEKTOREN PRO TRACK"
DB 13,10,"$"
SEK9 DB "NEUN SEKTOREN PRO TRACK"
DB 13,10,"$"
SEK10 DB "ZEHN SEKTOREN PRO TRACK"
DB 13,10,"$"
TR40 DD "40 TRACKS PRO DISKETTE"
DB 13,10,"$"
TR80 DD "80 TRACKS PRO DISKETTE"
DB 13,10,"$"
CI1 DD "EINGECITIDE DISKETTE"
DB 13,10,"$"
CI2 DD "ZWISCHENCITIDE DISKETTE"
DB 13,10,"$"
STARTME DE "DISKETTE EINLEGEN UND LAUFWERK AUSWAHLEN"
DB 13,10,"$"
ENDLESS DB 13,10,"$"
DB "FORMATIERUNG BEENDET. NOCH EINE DISKETTE"
DB "EINGABEN"
DB 13,10,"$"
WRITE DB 13,10,"$"
DB "DISKETTE SCHREIBGESCHUTZT"
DB 13,10,"$"
READY DB 13,10,"$"
DB "DRIVE NOT READY"
DB 13,10,"$"
ERROR DB 13,10,"$"
DB "SCHREIBFEHLER"
DB 13,10,"$"
CRLF DB 13,10,"$"
$
SEKANS DB 18000
DB 0
$
LEIZES BYTE IM MONITOR -
FREIFABEYTE FUER INTERRUPT
$
ORG OFFSET
INTERR DB 0
END

```

Sektoren je Track an. Dieser Variablen können die Werte Acht, Neun oder Zehn zugeordnet werden.

Die Variable TRKMAX gibt die maximale Anzahl der Spuren auf der Diskette an. Hier können die Werte 40 oder 30 eingetragen werden.

Die letzte zu definierende Variable heißt STEP und gibt die Steprate des Laufwerks an. Die für das verwendete Laufwerk relevanten Werte können der Tabelle 1 entnommen werden (siehe auch c't 2/84, Seite 86, Tabelle 5). In der Tabelle entspricht R1 dem Bit 1, R0 der Variablen. In STEP darf kein Wert eingetragen werden, der größer als drei ist.

Praxis

Sind alle Systemanpassungen vorgenommen, ist der c't86 einsatzbereit. Doch mancher stolze Besitzer wird Schwierigkeiten haben, die notwendigen Anpassungen der Software

selbst vorzunehmen. In diesem Fall bieten sich zwei Auswege an: Zum einen kann ein sachkundiger Bekannter Hilfestellung geben. Neben seinen Fachkenntnissen sollte er auch mindestens über ein lauffähiges CP/M-80-System und einen 8086-Cross-Assembler verfügen. Eine andere Lösung ist, sich eine bootfähige Diskette zu kaufen. Dieser Weg zum Erfolg ist sicher der schnellste und einfachste, aber wohl auch der kostspieligste. Damit c't-86-Anwender trotzdem zu einem preiswerten CP/M-86 kommen, haben wir uns etwas einfallen lassen (siehe Kasten).

Tabelle 1. Step-Raten des Floppy-Controllers

Step-Rate		Zeit in ms	
R1	R0	5 1/4"	8"
0	0	6	3
0	1	12	6
1	0	20	10
1	1	30	15

Preisgünstig zu CP/M 86

Ein fertig angepaßtes CP/M 86 für den c't 86 ist nicht billig. Das liegt daran, daß die relativ hohen Lizenzkosten bei geringen Stückzahlen voll zu Buche schlagen. Es gibt aber einen Weg, preisgünstig zu einem angepaßten Betriebssystem zu kommen: c't-86-Anwender können eine CP/M-86-Version für den IBM PC kaufen, die im Handel schon für rund 200 Mark erhältlich ist. Beim Software-Service des Heise-Verlags kann man das BIOS für den c't 86 sowie das Boot-Programm, das Formatierungsprogramm und das Programm Systemcopy auf die Original-Diskette kopieren lassen. Ergebnis: Ein CP/M-86-Betriebssystem für den c't 86. Der Kopierdienst kostet 20 Mark.

Wer diesen Service nutzen will, sollte die System-Diskette(n) in ausreichend fester Verpackung an den Heise-Software-Service, Bissendorfer Straße 8, 3000 Hannover 61, senden. Legen Sie bitte einen Verrechnungsscheck über den Betrag von 20 DM und einen Zettel mit genauer Angabe des Diskettenformats bei. Erlaubt sind folgende Formate:

5 1/4-Zoll, 40 Track, SS oder DS, DD, 8 oder 9 Sektoren pro Track und

8-Zoll, SS, SD (IBM-Standard-Format 3740).

Das Einkopieren der genannten Programme erfolgt ohne Gewähr für etwaige Beschädigungen.

Das Superangebot: NEC PC-8000

Computersystem
komplett mit
Keyboard,
Monitor, Doppel-
diskettenlaufwerk
sowie einem Text-
und einem
Kalkulations-
programm.



Komplettpreis: **5 699,—** inkl. MWST

V. Linde elektronik · Neue Straße 18 · 7170 Schwäb. Hall · Tel. 07 91/73 18

Und hier bekommen Sie:

Händler gesucht

Prolicomputer mit zwei CPUs und 64 K RAM + 12 K ROM
CPU 6502 + 280 A auf dem Motherboard, Int-Basic + Monitor prog.,
7 Slots, alle IC's gesockelt, betriebsbereit
Harte / Grafik apple-compatibel = CP/M-fähig **DM 1320,—**
Light-Pen: High-Resolution, Interface + Software nur **DM 519,—**
Graphic-Table: Software + Hardware, Plott-II kompl. **DM 269,—**
Slimline Disc-Laufwerk 5 1/4 Zoll, FDD-810 ein Qualitätslaufwerk
von TEAC. Im Gehäuse, voll apple-compatibel **DM 888,—**
Disc-Controller Card **DM 169,—**
Disketten höchster Qualität, 1S/125K-40Tr. 10 Stk. nur **DM 49,50**
RGB-Monitor 12", Color, 18 MHz, unbegrenzte Farben **DM 950,—**
Color-Monitor 14", NTSC-Eingang, für Apple-
Computer **nur DM 690,—**
Monitor 9" grün, High-Resolution 20 MHz, bei uns nur **DM 279,—**
IBM-compatible in Kürze lieferbar.

Weiteres COMP/CTP-Zubehör finden Sie in unserer SONDERLISTE, kostenlos anfordern!!!
Preise incl. MwSt. + Versand. Apple ist eingetrag. Warenz. der Apple Inc.

HÖSCH Elektronik Bruchstr. 43 4000 Düsseldorf 1 Tel. 02 11/678214

68008 ECB CPM/68K

- Single-Euro-Karte mit ECB-Bus
- 2 RS232 Vollduplex-Schnittstellen
- 2 Byte-wide Sockel für max. 64 kB
- Power fail detect
- 16 kB Monitor: stand-alone lauffähig
- Wartungsfeld kann angeschlossen werden
- Wait synchronisation
- Bustiming: MREQ einstellbar 300 ns – 1 µs
- Interr-Level: NMI, INT, Real-time-clk
- Adressen A0 – A19 ohne Banking
- Refresh-Generierung (ohne Adressen)
- Single 5V-Versorgung
- gepuffertes Uhrenchip
- Baugruppe: DM 1.699 inkl. MWST
- kompl. Rechner: DM 14.799 inkl. MWST
- CPM/68K für IHREN FD-Controller a.A.

dr. neuhaus

Hochallee 39 2000 Hamburg 13
Tel.: 040 / 44 17 02 Tx: 217 35 13 drn d

Assemblerkurs ASEM-4



komplett nur
DM 98,-

Mehr rausholen aus dem „Micro“ mit schnelleren und leistungsfähigeren Maschin-Programmen. – Lernen Sie Assembler-Programmierung und Computer-Aufbau verstehen. In direktem Bildschirm-Dialog und begleitet von zahlreichen Übungsaufgaben, wird der Stoff in leichtverständlicher Form vorgestellt.

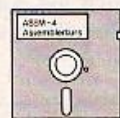
Im Preis von DM 98,- enthalten ist der komplette Kurs, bestehend aus:

- 2 deutschen Handbüchern (ca. 400 Seiten)
- Lösungsblätter und Referenzkarte
- Software auf Diskette oder Cassette

Wählen Sie für Ihr System passend aus:



CASSETTE
CBM 30xx / 4040
CBM 40xx
VC-20, C-64
ZX-31, ZX-Spectrum
Genie I, II, III
TRS-80, TI-99



DISKETTE
CBM 30xx / 4040
CBM 40xx / 4040
CBM 80xx / 8050
VC-20, C-64 / VC 1541
Genie I, II, III
APPLE II + IIe

... andere Systeme auf Anfrage!

Neu: ASEM-4 jetzt auch für CP/M-Systeme

Software für CP/M Systeme auf Cassette. Die Übertragung auf Ihren Computer erfolgt über die serielle Schnittstelle (V24) mit einem normalen Cassetten-Recorder und dem mitgelieferten Cassetten-Interface.

ASEM-4 komplett für CP/M, incl. Cassetten-Interface: nur **DM 148,-**

Versand per NN., Preise incl. MwSt.

Ing.-Büro Wilke

Postfach 1727 - 5100 Aachen - Telefon (0241) 3 681 + 87 0208

LECH-TECHNICS

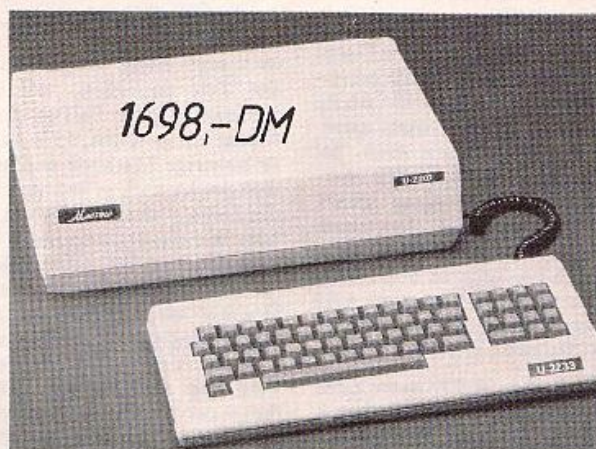
Gesellschaft zur Herstellung und Vertrieb von elektronischen Geräten und Microcomputern mbH.

Nußbaumallee 6
5014 Kerpen-Türnich
West-Germany

Telefon: 0 22 37/81 71
Telex: 899 103 wcr d

MICROCOMPUTER PC 301
voll IBM PC kompatibel

Das System enthält:
Mainboard
Disk Drive Carc, Multifunktion Card
Color Display Card, Schaltkreis: 8237
CPU 8086 (4.77 Mhz, DMA Contr. 8237
Interrupt Controller 8259 A
Sockel für Mathematik Chip 8087
128K RAM, aufrüstbar bis 512K
Uhr mit Kalender (Batterie gelad.)
zwei Diskettenlaufwerke 3.5" 320K
(double-sided, double-density)
1 seriellen Port RS 232C
1 Parallel Port (Centronic)
Text Mode:
40 Zeichen x 25 Zeilen Farbe/sw
80 Zeichen x 25 Zeilen Farbe/sw
Graphik Mode:
320 Punkte x 200 Linien Farbe/sw
640 Punkte x 200 Linien Farbe/sw
Betriebssystem
M-DOS 1.1 mit ext. BASIC, M-DOS 2.0
CP/M 86, ausführliche Handbücher
EINFÜHRUNGSPREIS: 7 999,- DM



Microcomputer "Atlas 2" (U 2200), voll APPLE kompatibel

CPU 6802 und Z 80 A, 64 K RAM, 24 K ROM (softswitch), 4 APPLE II kompatible Slots, eingebaute 16K Card, eingebaute Z 80 Card, eingebauter Disk Controller (DOS 3.3) für 2 Diskettenlaufwerke, Schaltkreis: +5V/5A, +12V/2.5A, -5V/0.5A, -12V/0.5A, abgesetzte Tastatur mit 12er Block, Belegung und Funktion voll APPLE II kompatibel.

EINFÜHRUNGSPREIS: 1 698,- DM

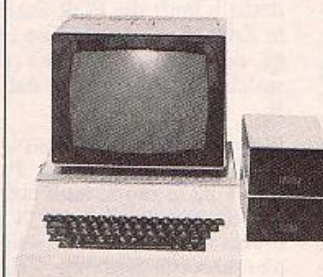
Zubehör für APPLE II und kompatibel:

Disk Drive Shugart SA 390	598,- DM
Disk Drive 5 1/4 Type	598,- DM
Monitor "Prince" 24 MHz grün	395,- DM
Monitor "Prince" 24 MHz bern.	445,- DM
Disk Controller APPLE komp.	148,- DM
80 Zeichen Karte softswitch	198,- DM
2 00 Karte (Microsoft)	148,- DM
Eprmer Karte (2716, 2732)	128,- DM
PAL-Karte mit Modulator	178,- DM
16K Language Card	148,- DM
256K Karte (Pseudo Disk)	998,- DM
Graphik Karte für Drucker	148,- DM

Paketangebote:

"Atlas 1", 16K Card, Disk Controller,	1900,- DM
Disk Drive	
"Atlas 1", 16K Card, Disk Controller,	2300,- DM
Disk Drive, Monitor grün	
"Atlas 1", 16K Card, Disk Controller,	2500,- DM
2 Diskettenlaufwerke	
"Atlas 1", 16K Card, Disk Controller,	2650,- DM
Disk Drive, 80 Zeichen Karte, Z 80 Karte,	
Monitor grün	

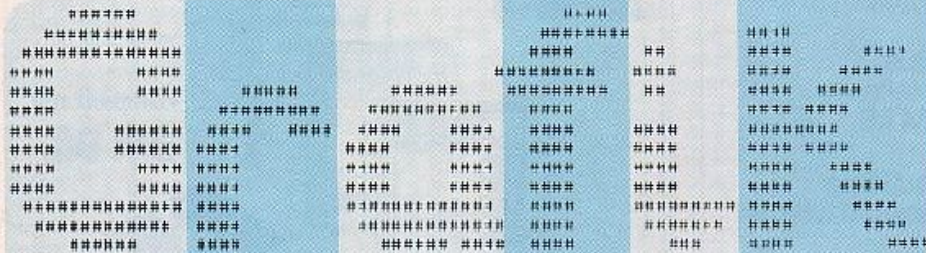
Alle Preise incl. 14% MwSt. zzgl. Porto und Verpackung.
Ausführliche Informationen und Preislisten gegen DIN A 5 Freumschlag und 1,30 DM Rückporto.



Microcomputer "Atlas 1"
voll APPLE kompatibel

betriebsbereit im Aluminiumgehäuse mit UIF-Modulator, 48K RAM, Schaltkreis, Anschlüsse für Monitor, Cassettenrecorder und Fern-seiter, Groß/Kleinschreibung.

EINFÜHRUNGSPREIS: 1 098,- DM



Eine Grafik sagt bekanntlich mehr als 1000 Worte. Doch nicht jeder Rechner ist grafikfähig und erst recht nicht jeder Drucker. Deshalb wurden in der EDV schon früh einige Techniken entwickelt, die notfalls auch noch die Ausgabe auf einem Fernschreiber gestatten. Hier die Lösungen in Microsoft-Basic.

Am häufigsten findet man Histogramme (Balkendiagramme) in allen möglichen Varianten. Das Prinzip ist — wie immer — ganz einfach. Der Teufel steckt — auch wie immer — im Detail und in einigen Tricks, die sich Programmierer im Laufe der Zeit haben einfallen lassen. Arbeiten wir uns langsam vor.

Bild 1 zeigt ein Listing mit zugehörigem Ergebnis. Die Daten werden in einem Array gespeichert, unabhängig davon, ob sie von der Tastatur, einem externen Gerät oder aus einer Berechnung stammen. Wir werden noch sehen, warum Arrays so vorteilhaft sind. In diesem Beispiel lesen wir die Daten aus der DATA Zeile in den Array A ein. Damit repräsentiert A(1) Balken 1 mit einer Länge von 2, A(4) Balken 4 (Länge=7) usw.

Jeder Balken soll in einer Zeile gedruckt werden. Wir haben 8 Daten = 8 Balken, deshalb die FOR-Schleife 1 TO 8 in Zeile 30. Die innere Schleife (Zeile 40) druckt nun bis 'TO A(I)' so viele 'X', wie das I-te Element von A zulässt. Zeile 50 schließlich hebt den mit Semikolon unterdrückten Zeilenvorschub wieder auf, und mit 'NEXT I' wiederholt sich das Ganze achtmal.

Im allgemeinen muß man die Balken mit Informationen versehen. Wir beschränken uns hier auf die laufende Nummer und den Wert. Wie Sie in Bild 2 sehen, erledigt das Zeile 35 recht gut. In Zeile 55 ist noch ein PRINT hinzugekommen, das eigentlich eine Leerzeile zwischen den Balken bringt.

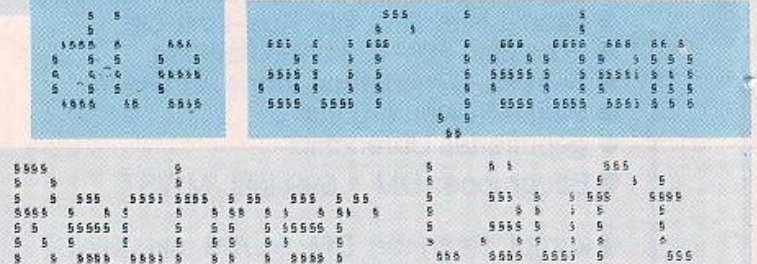
Es zeigt hier keine Wirkung, und das liegt am Drucker. Sollten Sie auch über so ein Exemplar verfügen, müssen Sie da den Vorschub erzwingen, in dem Sie einen String des Inhalts von 81 Blanks ausgeben, wenn die Zeilenlänge 80 ist.

Balken senkrecht

Will man einen Vorgang über der Zeit darstellen, sollte es besser sein, die Zeit auf der X-Achse zu sehen. Wir müssen also unser Histogramm um 90° drehen (Bild 3). Wir unterstellen, daß Drucker und Bildschirm nur Zeilenvorschub können, was sehr oft die Praxis ist. Jetzt wird die Anzahl der Zeilen durch den Größtwert von A(I), der Länge des längsten Balkens, bestimmt. Wie man den ermittelt, wird später noch gezeigt; wir notieren erst einmal die Zeile 30 G=10 und gleich daneben die Anzahl Balken mit B=8. A(G) muß zuerst gedruckt werden, die Zeile(n) mit dem kleinsten Wert zuletzt, deshalb läuft die FOR-Schleife in Zeile 40 rückwärts. Die Länge der inneren Schleife entspricht der Anzahl der Balken B.

Wenn jetzt der Druckkopf innerhalb einer Zeile von Position 1 bis B läuft, muß entweder ein Blank oder ein X gedruckt werden. Blank, wenn I (der abwärts laufende Zeilenzähler) größer ist als A(J),

Peter Wollschläger



```
XX
XXX
XXXX
XXXXXX
XXXXXXXXXX
XXXXXXXXXX
XXXXXX
XXXX
Ready
10 DATA 2,3,5,7,10,8,6,4
20 FOR I=1 TO 8:READ A(I):NEXT I
21
30 FOR I=1 TO 8
40 FOR J=1 TO A(I):PRINT "X":NEXT J
50 PRINT
60 NEXT I
```

Bild 1. Die einfachste Art, ein Balkendiagramm zu erzeugen.

```
1 2 XX
2 3 XXX
3 5 XXXXX
4 7 XXXXXXX
5 10 XXXXXXXXXX
6 8 XXXXXXXXX
7 6 XXXXXX
8 4 XXXX
Ready
10 DATA 2,3,5,7,10,8,6,4
20 FOR I=1 TO 8:READ A(I):NEXT I
21
30 FOR I=1 TO 8
35 PRINT I;TAB(5);A(I);TAB(9);
40 FOR J=1 TO A(I):PRINT "X":NEXT J
50 PRINT
55 PRINT
60 NEXT I
```

Bild 2. Der nächste Schritt. Das Histogramm wird beschriftet.

```
X
X
X X
X X X
X X X X
X X X X X
X X X X X X
X X X X X X X
X X X X X X X X
X X X X X X X X
Ready
10 DATA 2,3,5,7,10,8,6,4
20 FOR I=1 TO 8:READ A(I):NEXT I
21
30 G=10:B=0
40 FOR I=B TO 1 STEP -1
50 FOR J=1 TO B
60 IF I>A(J) THEN PRINT " ";GOTO 90
PRINT "X ";
80 NEXT J
90 PRINT
100 NEXT I
```

Bild 3. So wird das Histogramm um 90° gedreht.

```
10 DATA 2,3,5,7,10,8,6,4
20 FOR I=1 TO 8:READ A(I):NEXT I
21
30 B=10:B=0
40 FOR I=B TO 1 STEP -1
50 FOR J=1 TO B
60 IF I>A(J) THEN PRINT " "; ELSE PRINT "X ";
80 NEXT J
90 PRINT
100 NEXT I
```

Bild 3a. Mit IF...THEN...ELSE läßt sich das Problem einfacher schreiben.

spricht, die Balkenlänge in dieser Spalte. Die Zeilen 60 und 70 zeigen das, nur ist in beiden Zeilen noch je ein Blank angehängt, um die Abstände zwischen den Balken zu erzeugen. Bild 3a ist nur eine Variante für die Glücklichen, deren Basis: IF...THEN...ELSE kennt.

In der Praxis ist es so, daß man zwar die Anzahl Balken weiß, z.B. 12 für 12 Monate, nicht aber ihre Länge. In der Lösung nach Bild 1 kann das dazu führen, daß Balken über den rechten Rand hinausschießen. In der zweiten Lösung besteht die Gefahr zwar nicht, aber wenn der Drucker plötzlich 10 Meter Papier auswirft, ist das ja auch nicht schön.

Um das zu verhindern, wird normiert (Bild 4). Die Zahlen in der DATA-Zeile sind etwas größer geworden, wir wollen aber unseren Ausdruck auf maximal 13 Zeilen begrenzen. Der größte Wert ist 100, daraus muß 13 werden. 13/100 ergibt 0.13. Wenn wir also alle Werte mit dem Faktor 0.13 multiplizieren, ist das Ziel erreicht. In den Zeilen 21/22 finden Sie die Standardroutine zum Suchen eines Größtwertes in einem Array, in 23 den schon erwähnten Faktor. Sollten Sie über einen HP- oder Tek-Rechner verfügen, reicht für alle 3 Zeilen 'F=13/MAX(A)'.

In Zeile 24 wird der Array mit F multipliziert, und dann folgt alles wie gehabt, bis auf Zeile 45, die jetzt den neuen Skalierungswert ausgibt. Die Zahlen sind evtl. etwas nichtssagend, weshalb üblicherweise der Array vorab in Prozent umgerechnet wird. Dann kann man entweder akzeptieren, daß die Länge des Ausdrucks vom größten Prozentsatz abhängt, oder man normiert den auch noch.

Der nächste Schritt: Oft sollen Daten-Paare, zum Beispiel Umsatz und Erlös, zusammen dargestellt werden. Das eine als Reihe von 'X', das andere als '*'. Wie Bild 5 zeigt, muß dazu nur die innere Schleife zweimal gefahren werden, 'no problem'.

Schwieriger wird es schon, wenn wir das Bild drehen wollen. Dann gibt es nämlich je Druckposition 4 Möglichkeiten, und zwar (B steht für Blank): BB, BX, *B und *X. Bild 6 zeigt die Lösung. Im Prinzip doch ganz einfach: Die

```

100      X
92.3077  X
84.6154  X
76.9231  X X
69.2308  X X X
61.5385  X X X
53.8462  X X X X
46.1530  X X X X X
38.4615  X X X X X X
30.7692  X X X X X X
23.0769  X X X X X X X
15.3846  X X X X X X X
7.69231  X X X X X X X
Ready

10 DATA 20,30,50,70,100,80,60,40
20 FOR I=1 TO 8:READ A(I):NEXT
21 B=A(1):B=B
22 FOR I=2 TO 8:IF A(I)>B THEN B=A(I):NEXT
23 F=B/6
24 FOR I=1 TO 8:A(I)=A(I)*F:NEXT
25
40 FOR I=13 TO 1 STEP -1
45 PRINT I/F,
50 FOR J=1 TO B
60 IF I>A(J) THEN PRINT " ";:GOTO 80
70 PRINT "X ";
80 NEXT J
90 PRINT
100 NEXT I

```

Bild 4. Durch Normierung erreicht man konstant große Ausdrücke.

4 Möglichkeiten werden mit 3 'IF...THEN' getestet, die vierte ergibt sich dann zwangsläufig.

Kurven angedeutet

Warum immer nur Balken? Kurven kann man zwar nicht

```

XXX
*****
XXXXXXXXX
*****
XXXXXXXXXX
*****
XXXXXXXXXX
*****
XXXXXXXXXX
*****
Ready

10 DATA 3,7,11,14
11 DATA 5,8,12,11
20 FOR I=1 TO 4:READ A(I):NEXT
21 FOR I=1 TO 4:READ V(I):NEXT
30 FOR I=1 TO 4
40 FOR J=1 TO A(I):PRINT "X":NEXT
41 PRINT
42 FOR J=1 TO V(I):PRINT "*":NEXT
50 PRINT:PRINT
60 NEXT I

```

Bild 5. Daten-Paare zugleich dargestellt.

```

      * X
      * X
    X * X
  X X X X
X X X X X
X X X X X X
X X X X X X
X X X X X X X
X X X X X X X X
X X X X X X X X
X X X X X X X X
Ready

```

```

10 DATA 2,3,5,7,10,8,6,4
11 DATA 3,2,6,8,7,10,5,2
20 FOR I=1 TO 8:READ A(I):NEXT
21 FOR I=1 TO 8:READ V(I):NEXT
30 B=10:B=B
40 FOR I=6 TO 1 STEP -1
50 FOR J=1 TO B
60 IF I>A(J) AND I<V(J) THEN PRINT " ";:GOTO 80
70 IF I>A(J) AND I<V(J) THEN PRINT "X ";:GOTO 80
71 IF I<A(J) AND I>V(J) THEN PRINT "* ";:GOTO 80
72 PRINT "XX ";
80 NEXT J
90 PRINT
100 NEXT I

```

Bild 6. Bild 5 um 90° gedreht und mit Zwischenräumen: sieht schon besser aus.



```

1 SYSTEM "ROUTE *ID *PR"
100 SI=1:MY=5:H=B:PI=3.14
110 K1=H/SI*SQR(2*PI)
120 DEF FN GS(X)=INT(K1*EXP(-((X-MY)/SI)^2/2)+.5)
130 FOR I=MY-3*SI TO MY+3*SI STEP .5
140 PRINT TAB(FN GS(I)):"*"
150 NEXT
160 FOR I=FN GS(MY) TO 0 STEP -1
170 FOR J=MY-3*SI TO MY+3*SI STEP .2
180 IF FN GS(J)=I THEN PRINT "X":GOTO 200
190 PRINT " ";
200 NEXT J
210 PRINT
220 NEXT I
221 SYSTEM "RESET *DD"

```

Bild 7. So druckt man Kurven, der Drucker verzerrt das Bild.

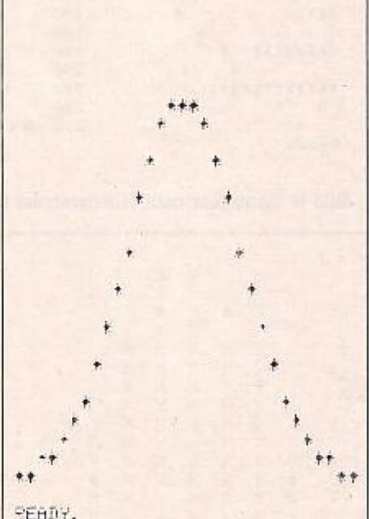


Bild 8. Das Programm aus Bild 7, aber mit einem anderen Drucker.

durchziehen, aber immerhin ganz gut andeuten. In Bild 7 sind in den Zeilen 100—110 ein paar Konstanten definiert, die Funktion in Zeile 120 ist die der Gauß'schen Glockenkurve, die sog. Normalverteilung. Wir kommen noch drauf, erst einmal zum Drucken von Funktionen überhaupt. Die Kurve von links nach rechts abzubilden, ist höchst simpel, die Zeilen 130—150 erledigen das. Als TAB-Parameter wird einfach der Funktionswert eingesetzt.

Die zweite Kurve ist schon schwieriger. Vorab: Wenn Sie da die Glockenform kaum erkennen, so liegt das nicht am Programm, sondern am Drucker. Dieser hat (solange man ihn nicht besonders einstellt) ein unsymmetrisches Zeilen/Spaltenverhältnis, was leider unsere Kurve linearisiert. Bild 8 zeigt das gleiche Programmergebnis auf einem Commodore-Drucker.

Wie müssen wieder vom Größt-
wert rückwärts drucken. Die

Maxima-Suche können wir hier sparen, das ist bei dieser Funktion GS(MY), so Zeile 160. Die innere Schleife muß in ziemlich kleinen Schritten laufen (hier Step 0.2), sonst haben wir nämlich geringe Chancen, einen Funktionswert zu erwischen, der gleich 1, also unserer Zeilennummer, ist. Die Zeilen 1 und 221 können Sie ignorieren, sie dienen lediglich bei dem hier eingesetzten Rechner zur Umleitung der Bildschirm-ausgabe auf den Drucker.

Bleibe last not least, wie drückt man Histogramm und Kurve gleichzeitig? In Forschung und Technik tägliche Praxis ist das Auswerten von Meßreihen. Dazu werden die Meßwerte in Klassen sortiert (gezählt). Die Klassenhäufigkeit kann man als Balkendiagramm darstellen. Man kann aber aus den Meßwerten die Verteilungsfunktion errechnen. Letzteres setzt immer voraus, daß ein Modell, wie z. B. die Gauß-Verteilung, angenommen wird. Stimmt die Annahme nicht, sind alle er-

rechneten Werte wie etwa Mittelwert und Streuung falsch. Es gibt zwar zahllose Programme, die automatisch die Ist-Verteilung gegen verschiedene Modelle rechnen und Unmengen von Zahlen ausdrucken, aber wer will das alles lesen?

Mehr als alle Zahlen sagt eine Grafik, die die Ist-Verteilung ausdrückt und darüber als Kurve das Modell, von dem das Programm meint, daß es das am besten korrelierende sei. Mit einem Plotter ist das einfach, da zeichnet man Histogramm und Kurve nacheinander. Hat man nur einen Drucker, muß man sich an ein Programm wie das in Bild 9 halten. Wie man die Parameter der Kurve rechnet, wird nicht vorgeführt. Die DATA-Zeile entspricht schon der Klassenhäufigkeit, in Zeile 40 stehen schon die Rechenergebnisse in Konstanten, Zeile 50 rechnet den Array in Prozent um, und in 60 schließlich finden Sie wieder die Funktion der Gauß-Laplace-Verteilung. Wir wollen

nun den Array Inhalt als Balken mit 'X' drucken und die Kurve mit '*'. Dabei hat ein '*' Vorrang, das heißt schneidet die Kurve einen Balken, soll kein 'X' sondern '*' gedruckt werden.

Letztierer Fall bedeutet, es müssen erst einige 'X', dann ein 's' und schließlich die restlichen 'X' gedruckt werden. Die Zeilen 140 bis 170 erledigen das. Es kann aber auch sein, daß die Kurve über den Balken liegt, dann sind erst die 'X', dann einige Blanks und dann der 's' zu drucken. Diesen Teil erledigen die Zeilen 190—210.

Der Trick in dem Programm ist, daß die Endwerte der FOR-Schleifen vom Index abhängige Variable sind. In Zeile 190 steht z. B. 'FOR J=1 TO T+(T=A(I))'. '(T=A(I))' ist ein logischer Ausdruck, der als wahr -1 ergibt. (So alle Commodore- und Tandy-Rechner. Beim Apple ergibt sich +1, da sind dann die Vorzeichen zu ändern.) Diese Technik erspart diverse IF-THEN zu verschiedenen Schleifen, die sich nur in der Endbedingung unterscheiden.

In Bild 10 wird die Gauß-Verteilung wieder aufgerichtet. (Die Daten sind da andere.) Wie Sie sehen, ist die Lösung wesentlich einfacher als die aus Bild 9. Daß man dennoch so oft die 'falsche' Darstellung findet, liegt daran, daß die wesentlich schneller ist. Der Nachteil des Programms von Bild 10 ist, daß die Funktion in der inneren Schleife gerechnet wird, also sehr oft, und das kostet Zeit.

Zum Schluß: Es gibt Programme, die entsprechend dem späteren Druckbild eine Maske generieren. Die würden also, um bei unserem letzten Beispiel zu bleiben, einen 2-dimensionalen Array mit Blanks, Kreuzen und Sternchen laden. Die Technik ist sehr einfach, da man ja wie beim Plotter alle Vorgänge nacheinander erledigen kann. Sie empfiehlt sich, wenn man weitere Balken mit anderem Muster und/oder mehr als eine Kurve ins Bild bringen will. Wie auch immer, diese Art der Grafik ist zwar nicht hochauflösend, aber höchst universell. Das gilt für die Hardware ebenso wie die Software. In den Grafik-Befehlen unterscheiden sich die Rechner sehr stark, PRINT können sie alle. □

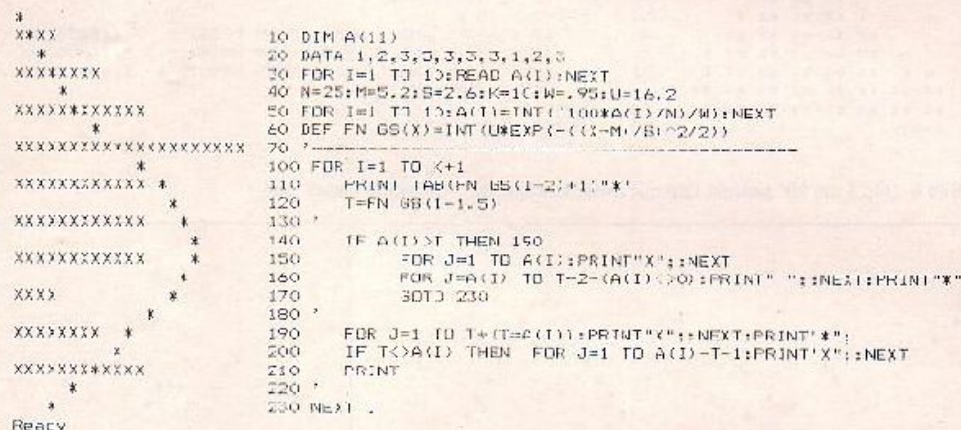


Bild 9. So mischt man Histogramme und Kurven.



Bild 10. Das Histogramm gedreht. Besser, kostet aber mehr CPU-Zeit.

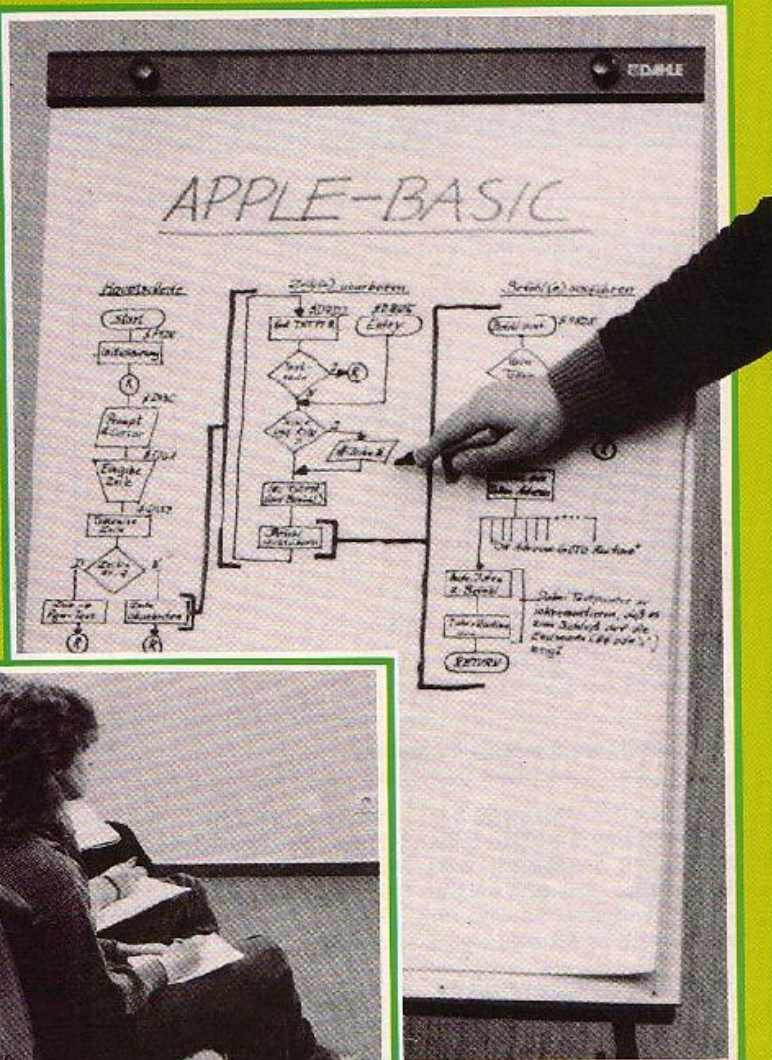
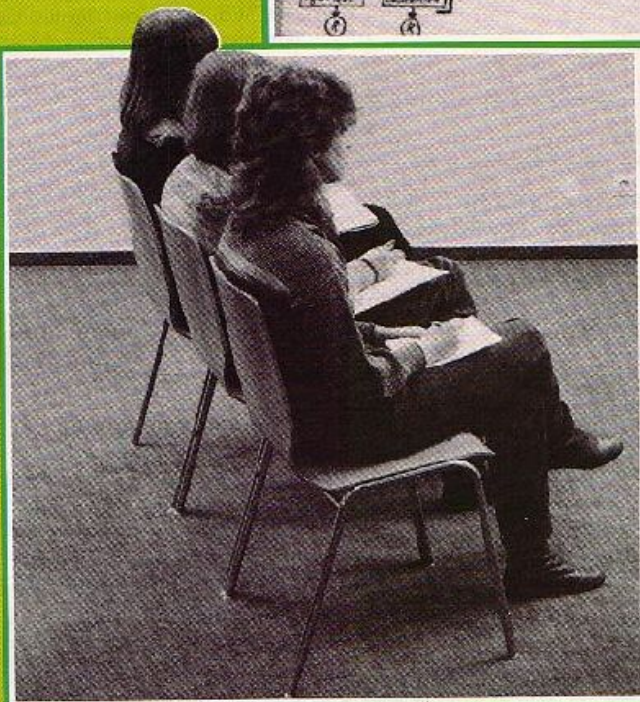
BASIC intern

Teil 1

Was nicht im Handbuch steht

Heinz-Peter Heidinger

Wie funktionieren Interpreter und Compiler? Wie kann man das 'serienmäßige' BASIC erweitern? Mit welchen Tricks arbeiten die Programmierer von Microsoft? — Eine c't-Serie für alle, die mehr als die 'Benutzeroberfläche' ihres Computers kennenlernen wollen. Also auch für Assembler-Freaks.



BASIC zu erlernen, stellt meist kein großes Problem dar. Manche Dinge sind allerdings nicht auf Anhieb verständlich. Als Beispiel sei hier nur VARPTR-Funktion genannt. Diese Funktion liefert Ergebnisse, die die Art und Weise reflektieren, wie BASIC sich Variablennamen und -inhalte intern im Speicher ablegt. Ohne spezielle Kenntnisse über die Strukturen der Speicherbereiche, die BASIC intern benutzt, fällt es schwer, mit diesen Werten etwas anzufangen. Dieser Artikel liefert das nötige Know-how.

Zum Verständnis sollten Sie lediglich etwas Übung im Umgang mit binären und hexadezimalen Zahlen mitbringen. Es wäre auch von Vorteil, wenn Ihnen ein Monitor-Programm zur Verfügung stünde. Ein solches ist zum Beispiel in c't 12/83 für Z80-Systeme vorge-

gestellt worden. Wer keinen Monitor zur Verfügung hat, wird die folgenden Strukturuntersuchungen an BASIC 'trocken' nachvollziehen können. Die Untersuchungen im Rahmen dieses Artikels orientieren sich am BASIC-Interpreter des TRS-80/Modell-1, sind aber prinzipiell auf die weiteren Interpreter aus dem Hause Microsoft übertragbar.

Bevor wir auf die ersten grundsätzlichen Strukturen von BASIC zu sprechen kommen, einige Worte zum SETUP-Programm in Tabelle 1. Dieses Programm dient einerseits dazu, für unsere Untersuchungen einen PROGRAMMTEXT im PROGRAMM-SPEICHER-BEREICH abzulegen. Es enthält alle möglichen Datenstrukturen (Datentypen), die in einem realen Programm in Be-

scheinung treten könnten — ganzzahlige, einfache- und doppeltgenaue sowie Zeichenketten-Variablen und Konstanten,

benutzerdefinierte Funktionen, BASIC-Anweisungen, -Funktionen und Befehle. Andererseits hat es die Aufgabe, durch

1	Programtext
1	einfache Variable
1	Feldvariable
1	freier Speicher
1	Stapel (Stack)
1	Zeichenkettenvariable (String spare)
1	fuer Maschinensprache-Programme reservierter Speicherplatz sofern dieser durch MEM SIZE geschuetzt wurde

Bild 1. Grobstruktur des Speichers für BASIC


```

100 REM *** Program-Anfang ***
110 REM *** Speicher-Setup fuer: "DASIC Intern!"
120 CLEAR 1000
130 DEFINT I
140 DEFINT J
150 DEFINT K
160 DEFSTR S
170 DEF FN I(A,B) = ALB
180 DIM A(7), C1(7), B1(7), G1(7)
190 DIM A2(7,7)
200 DIM A3(7,7)
210 A=32000: BZ=20000
220 C=45678.9: D=54789.1
230 E=123456789.1234567: F=23456789.12345678
240 B="String - vordefiniert": H="String - explizit definiert"
250 INPUT "ganzzahliger Wert: " I
260 INPUT "einfach genauer Wert: " J
270 INPUT "doppelt genauer Wert: " K
280 INPUT "String: " S
290 FOR X=0 TO 7: A(X) = A + X: NEXT X
300 FOR X=0 TO 7: C1(X) = C + X + .1: NEXT X
310 FOR X=0 TO 7: E(X) = E + X + .000001: NEXT X
320 FOR X=0 TO 7: G1(X) = "Feld-" + B + STR$(X): NEXT X
330 FOR Y=0 TO 7
340 FOR X=0 TO 7
350 A2(X,Y) = 21X + 21Y
360 NEXT X
370 FOR Y=0 TO 7
380 FOR X=0 TO 7
390 A3(X,Y,Z) = 21X + 21Y + 21Z
400 NEXT X
410 NEXT Y
420 PRINT FN I(5,2)

```

Tabelle 1. Speicher-Setup-Programm

den PROGRAMMLAUF den VARIABLEN-SPEICHERBEREICH mit bestimmten Werten zu füllen. Das eine wie das andere dient als Grundlage für unsere späteren Betrachtungen.

Speicher-Grobstruktur

Wenn wir einmal die Speicherbereiche richtig beachten, die durch den BASIC-Interpreter selbst, der Bildwiedergabespeicher und sonstige rechnerabhängige Gegebenheiten belegt werden, gliedert sich der für BASIC verfügbare Speicherbereich schematisch, wie in Bild 1 dargestellt. Die Pfeile geben dabei jeweils an, in welcher Richtung sich die Belegung der einzelnen Speichersegmente beim Schreiben des Programmes beziehungsweise beim Programmlauf ändert.

Die Unterscheidung zwischen 'Schreiben (oder Ändern) des Programmes' und 'Programmlauf' ist natürlich nicht willkürlich. Wie aus Bild 1 leicht zu folgern ist, 'schrumpft' der Speicherabschnitt 'Programtext' in Richtung niedrigerer Speicheradressen, wenn Sie Zeilen löschen, oder er breitet sich in Richtung höherer Speicheradressen aus, wenn Sie Programmzeilen hinzufügen, und überschreibt damit die Speicherbereiche, in denen die Variablenwerte abgelegt sind, wenn das Programm schon einmal gelaufen ist. Dagegen ist dieser Speicherabschnitt statisch, wenn das Programm läuft.

Vielleicht haben Sie schon einmal versucht, ein abgebroche-

dann wieder auf, wenn Sie das Programm neu starten. In seinem 'Workspace' (Arbeitsbereich) des BASIC-Interpreters, in dem sich BASIC alles 'Wichtige' merkt, siehe Tabelle 2), legt sich der Interpreter unter vielen anderen auch die Adressen für Anfang und Ende des Programmtextes ab. Man nennt diese beiden Adressen auch HEAD- und TAIL-Pointer (wörtlich übersetzt: 'Kopf'- und 'Schwanz-Zeiger' und nachfolgend immer als HP und TP abgekürzt). Zum HP erfahren Sie später noch einiges. Wir gehen erst einmal davon aus, daß er sich nicht ändert. Interessant ist zunächst der TP, dem gleichzeitig noch eine weitaus wichtigere Aufgabe zukommt. In Bild 1 sehen wir, daß sich BASIC alles 'Wissenswerte' über einfache Variable direkt hinter dem Programmtext ablegt. Wo dieser Bereich im Speicher beginnt, markiert der TP ebenso. Anders gesagt: der Zeiger auf das Programmtextende ist gleichzeitig Zeiger auf den Anfang der Variablen-tabelle.

Direkt auf das Speichersegment für einfache Variable folgt der Abschnitt, in dem sich BASIC die Namen und korrespondierenden Werte für Feld-Variablen ablegt. Auch für diesen Speicherabschnitt ist im Workspace

ein Zeiger vorgesehen: der Feldvariablen-Pointer (FPA). Dieser Zeiger ändert sich immer dann, wenn in die Tabelle für einfache Variable eine Variable neu aufgenommen wird. Dann muß der Feldvariablen-Bereich auch in Richtung des freien Speichers verschoben werden. Damit BASIC auch 'weiß', wie groß der Feldvariablen-Block ist, 'merkt' es sich außerdem, wo dieser Block im Speicher endet. Demzufolge gehören zu diesem Speichersegment zwei Pointer: der Zeiger auf den Anfang (FPA) und der auf das Ende (FPE).

Im weiteren wollen wir die Speicherschematik 'von unten her' (bezogen auf Bild 1) betrachten. Gehen wir einmal davon aus, daß kein Speicherplatz für ML-Routinen (ML = 'machine language') reserviert wurde. Dann befindet sich am oberen Speicherende, das heißt, auf den höchsten Speicheradressen, der 'Stringspace'. Dieser Speicherbereich enthält die Zeichenfolgen, die Stringvariablen zugewiesen wurden. Schaut man mit einem Monitor in diesen Abschnitt, stellt man fest, daß sich hier ohne erkennbare 'Trennbytes' und völlig beziehungslos alle möglichen Zeichenfolgen tummeln. Daß hier eine Beziehung zu der entsprechenden Stringvariablen besteht und Trennzeichen gar nicht notwendig sind, werden wir sehen, wenn wir die Variablen-tabelle näher betrachten.

Wichtig ist jedoch zu wissen, daß der Stringspace der Speicherbereich ist, dessen Umfang Sie mit der CLEAR-Anweisung festlegen. Damit ist auch dieses Speichersegment, wie das oberste (Programmtext), statisch, was allerdings nicht bedeutet, daß sich sein Inhalt nicht ändert. Ganz im Gegenteil! In diesem Speicherbereich regt sich am meisten, denn hier findet die berühmte 'garbage collector' statt. Was das ist, erfahren Sie bei näherer Betrachtung der Variablen-Tabelle.

Prozessor-Stack und BASIC-Stack

Über dem Stringspace ist der BASIC-Stack im Speicher angeordnet. Wie üblich dehnt er sich in Richtung niedrigerer Speicheradressen aus, wenn er zum Abstackeln von Programm-

```

# DUMP 4000 4110
4000 15 00 FF FF 00 00 00 CA 04 01 00 00 00 00 00 02 .....Feld.....
400C 00 FE FF FF 05 40 00 29 19 02 57 0E 00 00 00 00 00 .....S...M.....
401C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
402C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
403C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
404C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
405C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
406C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
407C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
408C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
409C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
40AC 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
40BC 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
40CC 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
40DC 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
40EC 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
40FC 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
410C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
4110 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....

```

1. Anfang des 'String space'
2. Aktuelle Zeilennummer, die bei Rückkehr zur Interpreter-Hauptschleife auf 5535 (FFFF) gesetzt! => Direkt-Modus!
3. HEAD-Pointer, zeigt auf den Anfang des Programmtextes (entspr. Betriebssystem!)
4. Höchste Speicheradresse, die für BASIC verfügbar ist (TOPMEM). Ab dieser Adresse beginnt der für Maschinen-Programme reservierte Speicherbereich sofern dieser durch den Stack festgelegt wurde.
5. Stack-Basis-Adresse (immer 2 Byte über dem String space).
6. TAIL-Pointer / Anfang der Variablen-Tabelle
7. Anfang Feld-Variablen
8. Ende Feld-Variablen
9. DATF-Pointer, zeigt auf das Trennzeichen hinter dem zuletzt mit READ genommen Datum. Bei Programmstart (RUN) wird er auf den Anfang des Programmtextes initialisiert (Zeilen-Trennzeichen '00').
10. Typcode-Tabelle. Enthält für jeden Anfangsbuchstaben der Variablen den entsprechenden Typcode. Durch NEW werden alle Tabelleneinträge auf 00 gesetzt (Typcode für einfache Genauigkeit SNG). Die Befehle zur Typcodeerstellung (DEFINT, DEFSTR, DEFDBL, DEFSTP) manipulieren die Tabelle. Für Variablen, die keine Typbezeichnung (Z, I, F, S) mitführen, wird der Typcode dieser Tabelle entnommen.

Tabelle 2. Microsoft-BASIC's Workspace (Auszug) / hier: TRS-80/M1/L3 (NEWDOS/80)

hex	dez.	ASCII	hex	dez.	ASCII	hex	dez.	ASCII
20	32	Space	40	64	@	60	96	`
21	33	!	41	65	A	61	97	a
22	34	"	42	66	B	62	98	b
23	35	#	43	67	C	63	99	c
24	36	\$	44	68	D	64	100	d
25	37	%	45	69	E	65	101	e
26	38	&	46	70	F	66	102	f
27	39	'	47	71	G	67	103	g
28	40	(48	72	H	68	104	h
29	41)	49	73	I	69	105	i
30	42	*	4A	74	J	6A	106	j
31	43	+	4B	75	K	6B	107	k
32	44	,	4C	76	L	6C	108	l
33	45	-	4D	77	M	6D	109	m
34	46	.	4E	78	N	6E	110	n
35	47	/	4F	79	O	6F	111	o
36	48	0	50	80	P	70	112	p
37	49	1	51	81	Q	71	113	q
38	50	2	52	82	R	72	114	r
39	51	3	53	83	S	73	115	s
40	52	4	54	84	T	74	116	t
41	53	5	55	85	U	75	117	u
42	54	6	56	86	V	76	118	v
43	55	7	57	87	W	77	119	w
44	56	8	58	88	X	78	120	x
45	57	9	59	89	Y	79	121	y
46	58	:	5A	90	Z	7A	122	z
47	59	;	5B	91	[7B	123	{
48	60	<	5C	92	\	7C	124	
49	61	=	5D	93]	7D	125	}
50	62	>	5E	94	^	7E	126	~
51	63	?	5F	95	_	7F	127	DEL

Tabelle 3. ASCII-Code-Tabelle

daten gebraucht wird. Neben seiner Prozessor-üblichen Funktion dient er dem Interpreter dazu, die Rückkehradressen für Unterprogrammaufrufe abzuheben. 'Das ist doch normal', mögen Assembler-Freaks jetzt gedacht haben. Nun, ganz so 'normal' ist es eben nicht, denn als Rückkehradresse für Unterprogrammaufrufe durch GOSUB, ist der Prozessor-interne 'programm counter' (PC) völlig irrelevant! Der Interpreter muß 'wissen', an welcher Stelle im Programmtext der UP-Aufruf stattgefunden hat, und die Speicheradresse dieser Stelle muß daher auf dem Stapel abgelegt sein.

Folgender Satz ist daher fundamental für BASIC-Interpreter auf 8080/8085- und Z80-Rechnern:

Zeiger auf den Programmtext ist immer das HL-Register. Sie können in Maschinensprache-Routinen mit den Registern machen, was Sie wollen (ausgenommen den Prozessor-Stackpointer zu ändern, ohne ihn wiederherzustellen) — unter der Bedingung, daß das IL-Register bei Rückkehr zum Interpreter unmittelbar hinter die Aufrufstelle zeigt!

Diesem Satz zufolge muß also bei UP-Verzweigungen der Inhalt des HL-Registers (BASIC-PC) auf dem Stapel abgelegt werden.

Auf dem BASIC-Stack werden ferner die Programmtext-

Adressen der ersten Anweisung nach Öffnen einer FOR-NEXT-Schleife sowie Name und Wert der korrespondierenden Laufvariablen abgelegt, damit BASIC weiß, ab wo weiter zu interpretieren ist, wenn eine NEXT-Anweisung auftritt.

Zwischen den beiden Speicherbereichen, deren Grenzen sich beim Programmablauf dynamisch verändern (Stack und einfache/Feld-Variablen-tabelle), liegt als Pufferzone der freiverfügbare Speicherraum ('spare memory'). Wenn dieser aufgezehrt ist (sei es durch zahlreiche Variablengebrauch, viele und/oder groß dimensionierte Felder, vielfältig verschachtelte FOR-NEXT-Schleifen oder auch zahlreiche UP-Aufrufe), erzeugt der BASIC-Interpreter einen 'OUT-OF-MEMORY ERROR'.

Blick ins 'Kleingedruckte'

Nach dem vorangegangenen Überblick wollen wir jetzt die einzelnen Speicherabschnitte detailliert betrachten. Beginnen wir mit dem Abschnitt 'Programmtext'. Einen Auszug dieses Abschnittes sehen Sie in Tabelle 5. Auf den ersten Blick wird man erstaunt sein, keine BASIC-Schlüsselwörter zu finden. Nun, sie sind dennoch vorhanden, aber in verschlüsselter Form.

Bekanntlich wird auf (fast) allen Rechnern zur Darstellung

von Zeichen der ASCII-Zeichensatz (American Standard Code for Information Interchange) verwendet. Eine Code-Tabelle dieses Zeichensatzes finden Sie in Bild 2. Diese Codeliste endet bei 127 (7Fh). Mit acht Bit Datenbreite ist es aber möglich, 128 weitere Codes für andere Zwecke zu vergeben.

Exakt dies wird auch mit den BASIC-Schlüsselwörtern gemacht. Daher finden Sie auch keine Keywords mehr im Speicher: Der Interpreter hat selbige in einzelne Bytes übersetzt, die 'Token' genannt werden (token <engl.> = Kennzeichen, Kennzeichner), bevor er die Programmzeile im Speicher ablegt. Eine Token-Liste, wie sie für den TRS-80 gilt, zeigt Tabelle 4.

Das 'Tokisieren' hat entscheidenden Einfluß auf die Arbeitsgeschwindigkeit des Interpreters, denn über den Code des Schlüsselwortes kann er sehr schnell die Adresse der zugehörigen Interpreter-Routine in einer Sprungtabelle finden. Außerdem erübrigt sich die zeitraubende Unterscheidung zwischen Variablennamen und Schlüsselwörtern während des

Programmlaufes, denn alles, was dann kein Token ist, wird als Variable interpretiert. Unterschieden wird schon bei der Übersetzung (bevor die Zeile im Programmspeicherbereich abgelegt wird) und nicht erst beim Programmlauf.

Inspizieren wir nun den Aufbau einer Programmzeile: Dem Memory-Dump in Tabelle 5 entsprechend finden wir das im Speicher befindliche Programm gemäß Tabelle 1 an Speicheradresse 6A46h, wie durch den HP (in Tabelle 2) ausgewiesen ist. Tatsächlich beginnt das BASIC-Programm aber schon ein Byte vorher, und dieses Byte muß immer '00' sein. Zunächst scheinen die ersten sechs Bytes bedeutungslos zu sein. Ab Byte sieben finden wir dann den Text, der in Zeile 100 hinter dem REM-Befehl abgelegt wurde, und an den sich wieder ein '00'-Byte anschließt. Die Funktion dieses '00'-Bytes ist jetzt augenfällig: es dient BASIC als Trennzeichen zwischen zwei Programmzeilen.

Betrachten wir jetzt die nächsten fünf Bytes der Zeile: Die Bytes 2 und 3 lauten 64h und

80 - 12B END	90 - 144 RESTORE	A0 - 160 OUT
81 - 129 FOR	91 - 145 GOSUB	A1 - 161 ON
82 - 130 RESET	92 - 146 RETURN	A2 - 162 OPEN
83 - 131 SET	93 - 147 REM	A3 - 163 FIELD
84 - 132 LLS	94 - 148 STOP	A4 - 164 GET
85 - 133 CHR	95 - 149 ELSE	A5 - 165 PUT
86 - 134 RANDOM	96 - 150 TRON	A6 - 166 CLOSE
87 - 135 NEXT	97 - 151 TROFF	A7 - 167 LOG
88 - 136 DATA	98 - 152 DEFSTR	A8 - 168 MERGE
89 - 137 INPUT	99 - 153 DEFINT	A9 - 169 NAME
90 - 138 DIM	9A - 154 DEFNG	AA - 170 KILL
91 - 139 READ	9B - 155 DEFDBL	AB - 171 LSET
92 - 140 LET	9C - 156 LINE	AC - 172 RSET
93 - 141 GOTO	9D - 157 EDIT	AD - 173 SAVE
94 - 142 RUN	9E - 158 FRAGR	AE - 174 SYSTEM
95 - 143 IF	9F - 159 RESUME	AF - 175 LPRINT

80 - 176 DEF	90 - 192 VARPTR	D0 - 208 /
81 - 177 POKE	91 - 193 USR	D1 - 209 L
82 - 178 PRINT	92 - 194 ERL	D2 - 210 AND
83 - 179 CONT	93 - 195 ERR	D3 - 211 OR
84 - 180 LIST	94 - 196 SCREENS	D4 - 212 *
85 - 181 LLIST	95 - 197 INSTR	D5 - 213 -
86 - 182 SELECT	96 - 198 POINT	D6 - 214 <
87 - 183 AUTO	97 - 199 TIME%	D7 - 215 SGN
88 - 184 CLEAR	98 - 200 MEM	DA - 216 INT
89 - 185 CLOAD	99 - 201 INKEY%	D9 - 217 ABS
90 - 186 CSAVE	9A - 202 THEN	DA - 218 FRE
91 - 187 NEW	9B - 203 NOT	DB - 219 TNP
92 - 188 TAB	9C - 204 STEP	DC - 220 POS
93 - 189 TO	9D - 205 +	DD - 221 SGN
94 - 190 FN	9E - 206 -	DE - 222 RND
95 - 191 USING	9F - 207 *	DF - 223 LOG

80 - 224 EXP	90 - 240 CSNG	
81 - 225 COS	91 - 241 CDBL	
82 - 226 SIN	92 - 242 FIX	
83 - 227 TAN	93 - 243 LEN	
84 - 228 ATN	94 - 244 STR	
85 - 229 PEEK	95 - 245 VAL	
86 - 230 CVI	96 - 246 AGC	
87 - 231 CVS	97 - 247 CIRC	
88 - 232 CVD	98 - 248 LEFT	
89 - 233 CDF	99 - 249 RIGHT	
90 - 234 LOC	9A - 250 MID	
91 - 235 LOF	9B - 251	
92 - 236 MKI	9C - 252	
93 - 237 MKS	9D - 253	
94 - 238 MKD	9E - 254	
95 - 239 CINT	9F - 255	

Tabelle 4. BASIC-Tokens (TRS-80 M1)

6Ah. Aha, das könnte eine Adresse innerhalb des betrachteten Speicherbereiches sein – 6A64h. Auf 6A63h liegt das Trennbyte '00'. Die beiden nächsten Bytes sind wieder eine Adresse, die auf den Anfang der folgenden Programmzeile zeigt usw. ...

BASIC führt also in jedem Informationsblock (hier: Programmzeile) eine Adresse (Zeiger) mit, die auf den Anfang des nächsten Datenblockes verweist. Eine solche Datenstruktur nennt man übrigens eine '(einfach) verkettete Liste'. Jetzt wird auch die Ein-Byte-Differenz zwischen tatsächlichem Programmtext-Anfang und Inhalt des HP deutlich — der HP ist der Zeiger auf das erste Lister-Element:

Wenn Sie die Zeiger-Systematik einmal weiterverfolgen, finden Sie irgendwann als Zeiger auf die nächste Zeile die Bytefolge 00 00. Daran erkennt der Interpreter, daß hier der Programmtext endet — er hat also die letzte Anweisung abgearbeitet und führt jetzt automatisch END (impliziertes END) aus. GOTO oder RETURN ändern den Zeiger auf die nächste Zeile (Anweisung). (GOTO durch die angegebene Zeilennummer und RETURN durch die Adresse auf dem Stapel).

Bleiben nun noch drei Bytes übrig, die sondiert werden müssen. Sie werden vielleicht schon die Zeilennummern vermißt haben: Diese finden sich auf den Plätzen 4 und 5 der im Speicher niedergelegten Programmzeile. Für unser Beispiel lauten diese beiden Bytes 64h und 00h — also 0064h, was die Zeilennummer 100 repräsentiert.

Da sich mit zwei Bytes in vorzeichenloser Darstellung 65536 Zahlen darstellen lassen, ergibt sich als größte Zeilennummer (theoretisch) 65535. Ihrem Rechnerhandbuch werden Sie aber schließlich schon entnommen haben, daß diese größtmögliche Zahl tatsächlich aber gar nicht mehr als Zeilennummer akzeptiert wird, sondern daß der Zeilennummernbereich irgendwo vorher begrenzt ist, beispielsweise bei 65529. Wozu diese Begrenzung?

Nun, sie reserviert einige Zeilennummern für den Interpreter, so daß dieser intern bestimmte Sonderzustände kenn-

zeichnen kann. So zum Beispiel, daß er sich im Direkt-Modus befindet, oder im Falle der ERL-Funktion, daß es keine Programmzeile gibt, in der ein Fehler aufgetreten ist.

Kommen wir zum letzten ungeklärten Byte des speicherinternen Formates unserer Programmzeile 100. Zu finden ist hier 93h — von der REM-Anweisung in Zeile 100 aber keine Spur . . . Was sagt Tabelle 4? Aha, 93h ist das Token von 'REM', also beginnt mit dem sechsten Byte der Zeile unser Programmtext. — Hier in Form eines tokisierten Schlüsselwortes.

Bevor ich Ihnen noch einige
 Tips und Tricks zu den zum
 Programmtext-Segment gehö-
 rigen Pointern (HP und TP) ge-
 be, eine kurze Zusammenfas-
 sung über die Struktur des hier
 besprochenen Speicherab-
 schnittes:

Byte 1 : immer '00', markiert den Anfang jeder folgenden Programmzeile.

Bytes 2,3: Zeiger auf den Anfang der nächsten Programmzeile im Rechnerspeicher. Wenn '0000', liegt Programtextende vor.

Bytes 4,5: binäres Äquivalent
der Zeilennummer:

Bytes 6 — n: Programmtext in
tokisierter Form.

Nun zu den Pointern: In Tabelle 2 sehen Sie einen Auszug des BASIC-Workspace (MICROSOFT-BASIC auf TRS-80/M1), in dem einige Adressen durch Kästchen markiert sind. Die im ersten Kästchen gekennzeichnete Adresse enthält in Übereinstimmung mit Tabelle 5 den HP, die zweite Markierung zeigt den TP. Der eine oder andere Leser (und TRS-80-Anwender) mag sich nun wundern, weil er ganz andere Adressen vorfindet (z. B. 42E9h im TRS-LEVEL2-BASIC), wenn er sich mit dem Monitor denselben Speicherbereich ansieht. Wer seinen TRS-80 mit DISK-BASIC unter NEWDOS/80 fährt, wird die gemachten Angaben dagegen bestätigen, andererseits hat der TRSDOS-Anwender schon wieder Einwände. Waren wir nicht von der Voraussetzung ausgegangen, daß sich der HP nicht ändert? Nun, prinzipiell hat das seine Richtigkeit, nur ist es — erstens — vom Betriebssystem abhängig

(LEVEL-2, TRSDOS 1.1/1.3/2.3/2.3B, NEWDOS/40, NEWDOS/80, DOSPLUS, LDOS, CP/M und was es sonst noch alles für den TRS-80 gibt), auf welchen Wert der HP beim Einstieg ins BASIC gesetzt wird. Zweitens ist beim DISK BASIC entscheidend, wie viele Puffer (0—15, je 290 Bytes) für das diskettenorientierte Input/Output beim Hochfahren des BASIC-Interpreters reserviert werden.

Es bedarf wohl keiner Diskussion, daß der HP nicht mehr geändert werden darf, wenn sich bereits ein Programm in Programmspeichersegment befindet. Gegen eine Verschiebung des HP spricht aber nichts, solange Sie noch kein Programm geladen haben. Durch 'Umpoken' des HP auf eine höhere Speicheradresse können Sie sich zum Beispiel einen bestimmten Speicherbereich reservieren, den Sie für Maschinenprogramme nutzen wollen. Welche Vorteile sich daraus allerdings im Vergleich

zeigt, ist dieses Programm zunächst für den BASIC-Interpreter verdeckt. Zwischen HP und TP steht nur die Endmarkierung des Programmtextes — 'kein Programm vorhanden!' heißt das für den Interpreter. In dieser Situation laden wir für gewöhnlich ein Programm, und es spricht nichts dagegen, daß wir es nun, nach Verschiebung des HP, gleichfalls tun. Das neue Programm wird dann ab IIP im Speicher abgelegt und der TP während des Ladens entsprechend nachgeführt. Dabei wird auch die alte Endmarkierung durch das nachgeladene Programm überschrieben, weil der 'verhogene' HP darauf zeigte. Ist das zweite Programm nun im Speicher, müssen wir den HP noch auf seinen ursprünglichen Wert zurücksetzen. Anzumerken ist allerdings, daß die erste Zeilennummer des zweiten Programmes größer sein muß als die letzte des ersten Programmes, das heißt, die Zeilennummern dürfen sich nicht überschneiden.

```

DUMP 6A45
00000000 00 64 61 64 00 73 20 28 2A 2A 20 *      .djd. ***
0A50 58 72 6F 67 72 61 6D 2D 41 6F 6A 61 5E 67 26 *Program=Antarg
2A00 2A 2A 2A 00 62 6E 00 37 2B 2A 2A 20 53 76 ****.j.. ** Sp
0A70 6E 69 63 68 72 7D 53 55 74 75 70 6A 76 75 65 *eicher=Setup fun
0A80 72 3A 20 22 42 41 53 49 13 26 6A 6F 74 5E 77 6F * "RAS" intern
0A90 21 22 00 7E 7A 00 80 20 31 36 30 30 6A 6A 6A *"=====
0AA0 87 00 70 20 41 0E 6A 3E 00 9F 20 45 80 B6 6F *   A.....
0AB0 8E 00 7D 3A 55 0E 6A 30 00 9F 70 47 80 D3 6F *   E.....
0AC0 8A 00 50 20 BE 20 49 28 41 20 47 29 20 35 20 41 *   I(A,B) .A
                                     .
0A80 9F 0A 00 61 41 32 20 32 32 32 32 2C 5A 29 30 05 20 *.m..3(X,Y,Z) .
0A90 01 01 50 20 D3 20 5B 01 59 20 3C 02 01 5A 00 AB *.X . Y . Z .
0DA0 6D 0A 01 87 3A 20 87 3A 00 07 6A 6A 0A 01 BE *.X . Y . Z .
0DB0 20 BE 20 49 28 35 2C 32 29 00 00 00 00 00 00 00 *.Y . Z .

```

Tabelle 5. BASIC-Programm im Speicher

zum standardmäßigen Reservieren von Speicherplatz am oberen Speicherende ergeben, mag dahingestellt bleiben.

Ein Trick zum Schluß

Mit den vielen Giften ist es so: Richtig dosiert, sind sie ein gutes Heilmittel. Wenn also oben gesagt wurde, daß der HP nicht mehr geändert werden darf, wenn sich ein Programm im Rechner befindet, so ist aus diesem 'Gift' für BASIC-Interpreter, die keinen Befehl zum Nachladen (Anhängen) von Programmzeilen kennen, jedoch eine 'Theorie' entwickelbar. Wenn Sie bereits ein Programm geladen haben und dann den HP so ändern, daß er zwei Speicherplätze vor den TP

Dieser kleine Trick zeigt, daß Kenntnisse über die speicherinternen Strukturen zu Verfahren im Umgang mit BASIC führen können, über die Ihr BASIC-Handbuch keine Aufschlüsse (mehr) gibt. In der nächsten Ausgabe von Ct wollen wir die anderen Speicher-Segmente eingehender durchleuchten. Zur Übung mag der geeignete Leser auf der BASIS der hier vermittelten Kenntnisse einmal versuchen, ein Programm zu schreiben, das die Programmzeilen eines bestehenden BASIC-Programmes umnummert. Eine gute Problemlösung sollte GOTO- und GOSUB-Anweisung berücksichtigen. Auch sollten Anfangszeile und Schrittweite festzulegen sein.

(Fortsetzung folgt)

Zahlen Sie zuviel Steuern?

Gesamtlohnsteuertabellen auf dem Spectrum 16K/48K

Walter Grotkasten

'Der Nettolohn ist nur die Provision für die Erarbeitung der Steuern' — so seufzen allerorts die Steuerzahler. Was Sie zu zahlen haben und ob die Berechnungen vom Finanzamt richtig durchgeführt wurden, das überprüft dieses Programm, das alle notwendigen Steuertabellen enthält. Es gibt Ihnen darüber Auskunft, wieviel Lohnsteuer der Arbeitgeber vom laufenden monatlichen Arbeitslohn einbehält, wieviel Kirchensteuer und Investitionshilfesonderabgabe Sie zu zahlen haben.



Je nach Auszahlungsart können Sie zwischen einer Monats-, Wochen- und Tageslohnsteuertabelle wählen; diese Tabellen werden aus den Einkommensteuertabellen (Grund- und Splittingtarif) entwickelt. Ihre individuellen Freibeträge, die in der Lohnsteuerkarte eingetragen sind, werden mit berücksichtigt.

Ferner haben Sie die Wahl zwischen einer 'Allgemeinen Lohnsteuertabelle' (z. B. Angestellte, Arbeiter) und einer 'Besonderen Lohnsteuertabelle' (z. B. Beamte, Richter, Berufssoldaten). Die Höchstwerte der drei Tabellen liegen bei:

- Monatslohnsteuertabelle 30 000 DM
- Wochenlohnsteuertabelle 19 000 DM
- Tageslohnsteuertabelle 2 700 DM.

Als Berechnungsgrundlage für die Tabellen sind folgende Freibeträge und Pauschbeträge vorausgesetzt:

— der Grundfreibetrag in den Steuerklassen 1, 2 und 4	DM 4 212
— der Grundfreibetrag in der Steuerklasse 3	DM 8 424
— der Arbeitnehmer-Freibetrag in den Steuerklassen 1 bis 5	DM 480
— der Werbungskosten-Pauschbetrag in den Steuerklassen 1 bis 5	DM 564
— der Sonderausgaben-Pauschbetrag in den Steuerklassen 1, 2 und 4	DM 270
— in der Steuerklasse 3	DM 540
— Vorsorgespauschale bis zu den jeweiligen Höchstbeträgen abhängig von der Anzahl der Kinder, Steuerklasse und Beitragsbemessungsgrenze zur Rentenversicherung (1984)	DM 62 400)
— der Haushaltsfreibetrag in der Steuerklasse 2 bei Arbeitnehmern mit mindestens einem Kind	DM 4 212
— der Kinderfreibetrag je Kind in den Steuerklassen 2 und 3	DM 432
— in der Steuerklasse 4	DM 216
— ein Rundungsbetrag in der Steuerklasse 6	DM 18

Die Kirchensteuer wird in einem bestimmten, in den einzelnen Bundesländern verschiedenen Prozentsatz von der Lohnsteuer erhoben. Sind Kinder zu berücksichtigen, so kann man die Kirchensteuer nicht unmittelbar aus der Lohnsteuer berechnen. Es werden je nach Anzahl der Kinder von der Lohnsteuer noch bestimmte Beträge abgezogen.

Die Investitionshilfesonderabgabe wird ab einer bestimmten Höhe der im Jahr angefallenen zu zahlenden Einkommensteuer erhoben. Sie beträgt grundsätzlich 5%. Zur Vermeidung von Härten wurden jedoch bestimmte Übergangsbereiche gebildet, in denen der Prozentsatz geringer ist.

Menü und Microdrive

Nach Eingabe des Listings muß das Programm mit GOTO 2000 gestartet werden. Wird das Programm nun mit LOAD 'LOHN' von der Kassette in

Kurzbeschreibung

Zeile	
30:	Initialisierung der Variablen
35:	Aufbau der Menü-Karte
50:	Auswahl der Tabellensart (Monats-, Wochen- oder Tageslohnsteuertabelle)
70:	Allgemeine Hinweise für den Programmablauf (Auswahl durch die Menü-Karte)
93:	Eingabe, ob die 'Allgemeine Lohnsteuertabelle' oder die 'Besondere Lohnsteuertabelle' berechnet werden soll
100:	Eingabe der Steuerklasse 1 bis 6
130:	Ausgabe bei falscher Eingabe
140:	Eingabe der Anzahl der Kinder (max. 13 Kinder)
165:	Ausgabe bei falscher Eingabe
170:	Eingabe des Freibetrages (in der Steuerkarte eingetragen)
200:	Eingabe des Anfangswertes der Tabelle, die berechnet werden soll
265:	Eingabe des Endwertes der Tabelle, die berechnet werden soll
290:	Die letzten 3 Eingaben werden auf ihren Betrag überprüft
310:	Eingabe des Bundeslandes wo Sie wohnen
340:	Initialisierung für die Monatslohnsteuertabelle
360:	Initialisierung für die Wochenlohnsteuertabelle
380:	Initialisierung für die Tageslohnsteuertabelle
400:	Berechnung der Pauschbeträge (abhängig von der Wahl der Steuerklasse)
440:	Berechnung der Kirchensteuer
480:	Berechnung der Vorsorgespauschale
505:	Ausgabe für den Tabellenkopf
605:	Aufruf des Unterprogrammes: Berechnung der Einkommensteuertabelle
640:	Berechnung der Lohnsteuer
685:	Berechnung der Investitionshilfesonderabgabe
800:	Ausgabe der Daten für die Tabelle
810:	Überprüfung, ob Tabellenende erreicht wurde
825:	Überprüfung, ob Bildschirmseite gefüllt ist
1000:	Unterprogramm: Überprüfung der Eingaben
1500:	Unterprogramm: Berechnung der Einkommensteuertabellen

den Rechner geladen, so erscheint nach dem erfolgreichen Ladevorgang die Menü-Karte auf dem Bildschirm. Durch Aufruf des Menüpunktes 'Hinweise' erhält man eine Kurzbeschreibung des Programmes. Es werden nun im Dialogverfahren alle Daten abgefragt, die zur Berechnung nötig sind. Soll das Programm auf ein

Microdrive gespeichert werden, so muß nach Anschluß des Interface und des Microdrive die Zeile 2000 wie folgt geändert werden:

SAVE*"m";1;"LOHN" LINE 0
Mit dem Befehl
LOAD*"m";1;"LOHN" LINE 0
wird das Programm von dem Microdrive in den Rechner geladen.




```

405 IF EA=EI THEN LET GD=1314
410 IF EA=2 THEN LET GD=5525+EB
*432
415 IF EA=3 THEN LET GD=1584+EB
*432
420 IF EA=4 THEN LET GD=1314+EB
*215
425 IF EA=FU THEN LET GD=1044
430 LET GG=EI
435 IF EA=3 OR EA=FU OR EA=5 TH
EN LET GG=NL
440 LET GH=NL
445 IF EB=EI THEN LET GH=600
450 IF EB=2 THEN LET GH=1560
455 IF EB=2 THEN LET GH=1560+(E
3-2)*1000
460 IF EA=4 THEN LET GH=GH/2
465 IF EA=EI THEN GO TO 475
470 LET GE=S1: LET GF=32
475 IF EA=2 THEN GO TO 485
480 LET GE=S1+EB+600: LET GF=52
+EB*300
485 IF EA=3 THEN GO TO 495
490 LET GE=2*S1+EB+600: LET GF=
2*32+EB*300
495 IF EA=4 THEN GO TO 505
500 LET GE=S1+EB+300: LET GF=52
+EB*150
505 PRINT "ARBEITS- /LOHNSTEUER
/ SONDERAB- LOHN /
/ GABE"
510 PRINT
515 PRINT " /TABELLEN-
/ KIRCHEN- /ART
/ STEUER"
520 PRINT
525 LET HA=(GA-EC)*GC
530 IF EA=5 OR EA=6 THEN GO TO
520
535 LET HB=HA
540 IF HB>62400 THEN LET HB=624
00
545 LET HB=HB*0.09
550 LET HC=HB
555 IF HB>GE THEN LET HC=GE
560 IF HB>GF THEN GO TO 575
565 LET HC=HC+HB
570 GO TO 580
575 LET HC=HC+GF
580 LET HC=(INT (HC/54))*54
585 IF GG=EI AND HC<300 THEN LE
T HC=300
590 IF GG=NL AND HC<600 THEN LE
T HC=600
595 LET HA=HA-HC-GD
600 LET KA=HA
605 GO SUB EIN
610 LET HD=KE
615 GO TO 660
620 LET HA=HA-GD
625 LET HA=(INT (HA/54))*54
630 LET KA=HA*1.5
635 GO SUB EIN
640 LET HD=KE
645 LET KA=HA*2.5
650 GO SUB EIN
655 LET HD=KE-HD
660 IF HD<NL THEN LET HD=NL
665 LET HE=(HD-GH)*EF/100
670 IF HE<NL THEN LET HE=NL
675 LET HD=(INT (HD/GC*100))/10
0
680 IF B$="M" THEN LET HD=(INT
(HD*10))/10
685 LET HF=NL
690 IF B$(">"M" THEN GO TO 725
695 IF EA=3 THEN GO TO 715
700 IF HD>1250 THEN LET HF=(INT
(HD-1250))*0.35
705 IF HD>1458.3 THEN LET HF=HD
*0.05
710 GO TO 725
715 IF HD>2500 THEN LET HF=(INT
(HD-2500))*0.35
720 IF HD>2916.6 THEN LET HF=HD
*0.05
725 IF B$(">"W" THEN GO TO 760
730 IF EA=3 THEN GO TO 750
735 IF HD>291.66 THEN LET HF=(I
NT ((HD-291.66)/0.2))*0.07

```

```

740 IF HD>340.2 THEN LET HF=HD*
2.05
745 GO TO 760
750 IF HD>583.33 THEN LET HF=(I
NT ((HD-583.33)/0.2))*0.07
755 IF HD>880.6 THEN LET HF=HD:
2.05
760 IF B$(">"T" THEN GO TO 795
765 IF EA=3 THEN GO TO 785
770 IF HD>44.53 THEN LET HF=(HD
-41.56)*0.35
775 IF HD>48.6 THEN LET HF=HD*0
.05
780 GO TO 795
785 IF HD>80.33 THEN LET HF=(HD
-33.33)*0.35
790 IF HD>97.21 THEN LET HF=HD:
3.05
795 LET HF=(INT (HF*100))/100
800 PRINT GA; TAB 10;HD; TAB 20
HF
802 PRINT TAB 10;E5;"/";B$;"/";
GA;"/";EB; TAB 20;HE
805 PRINT
810 IF GA=EE THEN GO TO 855
815 LET GA=GA+GI
820 LET COU=COU+3
825 IF COU<21 THEN GO TO 525
830 PRINT AT 21,NL;"(D)RUCKEN
(W)EITER (M)ENUE"
835 IF INKEY$="D" THEN GO SUB 1
700: COPY
840 IF INKEY$="M" THEN CLS : PA
USE 50: RUN 30
845 IF INKEY$="W" THEN LET COU=
NL: CLS : GO TO 525
850 GO TO 830
855 PRINT AT 21,NL;"(D)RUCKEN",
(M)ENUE"
860 IF INKEY$="D" THEN GO SUB 1
700: COPY
865 IF INKEY$="M" THEN CLS : PA
USE 50: RUN 30
870 GO TO 855
1000 LET AR=NL
1005 FOR X=EI TO LEN A$
1010 IF A$(X)>="0" AND A$(X)<="9
" THEN GO TO 1025
1015 PRINT AT 15,NL;"FALSCH EIN
GABE. BITTE NEU EIN-","GEBEN!"
: BEEP 1.5
1020 LET AR=EI
1025 NEXT X
1030 IF A$="" THEN LET AR=EI
1035 RETURN
1040 IF KA<=18000 THEN GO TO 159
5
1500 IF GG=EI THEN GO TO 1530
1505 IF EA=3 THEN GO TO 1520
1510 LET KA=(54*INT (KA/54))/2
1515 GO TO 1535
1520 LET KA=(108*INT (KA/108))/2
1525 GO TO 1535
1530 LET KA=54*INT (KA/54)
1535 IF KA<=4212 THEN GO TO 1605
1540 IF KA<=18000 THEN GO TO 159
5
1545 IF KA<=59999 THEN GO TO 158
0
1550 IF KA<=129999 THEN GO TO 16
55
1555 LET KE=INT (0.56*KA-14837)
1560 GO TO 1610
1565 LET KE=(KA-60000)/10000
1570 LET KE=INT (((0.09*KE-5.45
)*KE+88.13)*KE+5040)*KE+20018)
1575 GO TO 1610
1580 LET KE=(KA-18000)/10000
1585 LET KE=(((3.05*KE-73.76)*K
E+695)*KE+2200)*KE+3034)
1590 GO TO 1610
1595 LET KE=INT (0.22*KA-926)
1600 GO TO 1610
1605 LET KE=NL
1608 IF EA=5 OR EA=6 THEN LET KP
=INT (0.22*KA)
1610 IF GG=NL THEN LET KE=KE+KE
1615 RETURN
1700 PRINT AT 21,NL,""
1705 RETURN
9000 SAVE "LOHN" LINE 0

```


Mischkulturen

Spectrum 48 K als Hobby-Gärtner

Udo Bartz

Daß Erdbeeren besonders gut munden, wenn zwischen ihnen Knoblauch oder Zwiebeln angepflanzt werden, ist versierten Hobby-Gärtnern schon länger bekannt. Diese sogenannten Mischkulturen bieten aber noch erheblich mehr Vorteile. Bei geschickter Kombination der Pflanzen halten diese sich

sogar gegenseitig die Schädlinge vom Hals, man spart also Schädlingsbekämpfungsmittel. Trotz dieser verdächtig gesunden Anbaumethode braucht man aber keine Ertragseinbußen hinzunehmen. Der gezielte Einsatz von Mischkulturen erfordert allerdings einen hohen 'Verwaltungsaufwand', wenn man alle Vorteile optimal nutzen will. Hier ist der Computer in seinem Element.



Foto: Eduard Dietl, BAVARIA

Wenn Mischkulturen richtig geplant werden, kann man Ermüdungserscheinungen des Bodens, wie sie bei Monokulturen zwangsläufig auftreten, weitgehend vermeiden. Dadurch ist man nicht gezwungen, den Boden mit immer größeren Mengen Kunstdünger vollzupumpen, was ganz sicher allen Beteiligten, nämlich Mensch, Pflanze und Boden, zugute kommt.

Die richtige Mischung

Mischkulturen sind der Natur abgucken, denn dort gibt es kein Monopol einer einzigen

Pflanzenart. Die Pflanzen, die sich in der Wildnis an ihrem Standort behaupten, müssen sich aber auch miteinander 'vertragen' können. Sie bilden eine Biozönose (griech.: Lebensgemeinschaft). Solche Harmonie wird auch in der Biozönose 'Garten' angestrebt. Sollen die oben genannten Vorteile genutzt werden, muß man auch im Garten auf gute Verträglichkeit der Pflanzen untereinander achten.

Die Pflanzen dürfen sich mit ihren Früchten und Blättern nicht gegenseitig behindern. Sie müssen auch und gerade im

VORSCHLAG Nr. 1 (Beetbr. = 1,20 m)

*****	24/39	RAD./KR.
*****	21	KOHLRAB I
*****	23	KOPFSAL.
*****	24/39	RAD./KR.
*****	20	KOHL
*****	24/39	RAD./KR.
*****	23	KOPFSAL.
*****	21	KOHLRAB I
*****	24/39	RAD./KR.

SPÄTER KOENNEN NOCH WINTERSALATE ODER SPINAT NACHGESÄET WERDEN

Wurzelbereich miteinander harmonisieren, und ihre Nährstoffentnahme muß berücksichtigt werden. Pflanzen geben auch eine Vielzahl von Substanzen an den Boden zurück, die von ihren Nachbarn nicht immer gut vertragen werden. So spielen die Däfte und Wurzelabscheidungen bei der Wahl ebenfalls eine wichtige Rolle.

Zu diesem Thema gibt es mittlerweile eine ganze Wissenschaft, die sich, gestützt auf jahrzehntelange Untersuchungen und Beobachtungen, mit diesen Wechselwirkungen beschäftigt. Die Ergebnisse sind verblüffend und einfach zugleich. Manche Pflanzen können sich buchstäblich nicht 'reiben' und gehen ein, wenn sie zu entsprechender Nachbarschaft gezwungen werden, andere Kombinationen blühen förmlich auf und wirken so günstig aufeinander ein, daß man nur staunen kann. 'Artenfreundliche' Pflanzen verflechten sich im Wurzelbereich sogar, während 'artenfeindliche' Nachbarn sich zumeist auf einen engen, isolierten Wurzelbereich zurückziehen und unter Umständen verkümmern.

Diese Beobachtungen sind gar nicht neu. Schon den alten Chinesen und anderen Kulturvölkern waren diese Zusammenhänge bekannt und wurden bei der Planung und Gestaltung der Gärten und Felder berücksichtigt. In das hier vorgestellte Programm sind diese Erkenntnisse eingebracht worden und können für 62 Pflanzenarten ausgewertet werden.

Mischen — Säen — Ernten

Die Struktur des Programms ist bewußt einfach und übersichtlich gestaltet, um individuelle Erweiterungen recht einfach vornehmen zu können. Bei der Planung Ihres Gartens gehen Sie am besten wie folgt vor:

Lassen Sie sich zunächst die Liste aller hier benutzten Pflanzen ausdrucken, und suchen Sie sich diejenigen heraus, die Sie anbauen wollen. (Falls Sie keinen Drucker haben, sehen Sie einfach im Listing — Zeile 8000 bis 8030 nach.) Danach lassen Sie sich vom SPECTRUM zeigen, welche Pflanzen Sie auf keinen Fall auf ein gemeinsa-

mes Beet setzen dürfen. Unter Berücksichtigung der verschiedenen Pflanz- und Erntezeiten können Sie nur mit Hilfe des Programms Ihre Beete zusammenstellen. Sollten Sie noch ungeübt in diesem Metier sein, dann sehen Sie sich vielleicht zuerst die in drei ausgewählten Beispielen im Programm vorhandenen Computer-Vorschläge an.

Pflanz- und Erntezeiten sind im Programm selbst nicht berücksichtigt worden, da dies wegen der Vielfalt des erhältlichen Saatgutes kaum möglich ist.

Alle Jahre wieder

Denken Sie bitte daran, daß eine Mischkultur im Gegensatz zur herkömmlichen zeitlichen Fruchtfolge nicht ein Nacheinander, sondern ein Miteinander in einer Pflanzengemeinschaft, also eine räumliche Fruchtfolge, darstellt. Ein einmal festgelegtes Schema können und sollten Sie daher immer wieder benutzen, denn das bunte Gemisch ist nicht wahllos zusammengewürfelt, sondern folgt einem wohlgedachten Plan, um die wechselseitig schützenden, vorbeugenden und fördernden Eigenschaften der Pflanzen bestmöglich auszunutzen.

Ein gutes Beispiel ist das Zusammenleben von Möhren und Zwiebeln: beide vertreiben sich gegenseitig mit ihren Ausscheidungen die Schädlinge (Möhren- und Zwiebelfliege). Aber auch auf den Geschmack der geernteten Früchte wirken sich Mischkulturen positiv aus.

Erweiterungen

Das Programm benötigt fast 21 KByte an Speicherplatz, und ist so aufgebaut, daß alle wichtigen Angaben in DATA-Zeilen vorhanden sind. Dies läßt den SPECTRUM sehr schnell reagieren und macht das Einfügen Ihrer eigenen Pflanzkombinationen in das Programm sehr einfach. Dazu ein Beispiel:

Zwei neue Pflanzen sollen aufgenommen werden. Ihre Namen werden zuerst an das DATA-Feld ab Zeile 8000 angehängt, dann wird die Laufvariable der beiden Schleifen in den Zeilen 9 und 50 um zwei erhöht, da jetzt 64 Einträge vor-

handen sind. Die Querverweiseroutine in den Zeilen 6300 bis 6360 sollte komplett verschoben werden. Dazu eignet sich der Bereich 8300 bis 8360. Die Variable 'qv' in Zeile 11 wird entsprechend in 8300 geändert. So haben Sie Platz für sieben weitere Pflanzen. Da der Zugriff auf die einzelnen Pflanzen in Hundertschritten erfolgt, (Zeile 52 und 67) verfahren Sie nach dem gleichen Muster und legen ähnliche Routinen in den Zeilen 6300 und 6400 für die beiden neu hinzugekommenen Pflanzenarten an, wie Sie sie in den anderen Hunderterblöcken erkennen können. Sie dürfen auch nicht vergessen, das Feld für AS (Zeile 8) um die Anzahl der Neuzugänge zu erweitern, desgleichen die Variablen 'i' in der Zeile 35.

Grünes Menue

Über ein Menue haben Sie nach dem Programmstart die Möglichkeit, sich entweder die zueinander passenden Pflanzen herauszusuchen, oder die Vorschläge des SPECTRUM anzusehen, Ihre eigenen Kreationen zu entwickeln und ausdrucken zu lassen, oder sich eine komplette Liste der im Programm vorhandenen Pflanzen vom Drucker ausgeben zu lassen. Bei der Pflanzenauswahl können Sie kreuz und quer durch die Kombinationen springen, Sie brauchen nur die entsprechenden Codezahlen für die jeweilige Pflanze einzugeben.

Haben Sie den Menuevorschlag 'Eigene Beetgestaltung' gewählt, fragt der SPECTRUM nach Beetnummer oder Lagebezeichnung und Pflanzzeit. Dann ist die Anzahl der zu bepflanzen Reihen einzugeben. Der Computer zeichnet daraufhin ein entsprechend großes Beet und links daneben einen Cursor, mit dem Sie herauf- und herunterfahren können, um die gewünschten Reihen mit den Pflanzennamen zu kennzeichnen oder um Korrekturen vorzunehmen. Mit den Tasten '6' und '7' ist der Cursor zu bewegen, und mit 'P' teilen Sie mit, daß Sie in die rechts neben dem Cursor stehende Reihe etwas eintragen wollen. Dieser Eintrag darf maximal 19 Zeichen lang sein, was ausreichen sollte, um auch zwei Namen gleichzeitig in eine Reihe zu bekommen.

Wenn Sie fertig sind, tippen Sie auf die Frage nach der Pflanzenart einfach ENTER. Nunmehr stellt der SPECTRUM Ihnen nacheinander die sieben ungünstigsten Kombinationen vor, wie sie in einer Mischkultur auf jeden Fall vermieden werden sollten, und fragt, ob diese in Ihrer Kreation vorhanden sind. Wird dies in irgendeinem Fall bejaht, erhalten Sie die Möglichkeit zur Korrektur. Ist alles okay, dann können Sie sich Ihr Beet ausdrucken lassen, um es in Ihren Gartenplan einzukleben. Vorsichtshalber sehen Sie aber noch einmal im Menuevorschlag '1' nach, ob alle von Ihnen gewählten Pflanzen sich miteinander vertragen.

Listing

Im Programmlisting bedürfen einige Passagen einer Erläuterung, da der verwendete Drucker nicht alle Zeichen des SPECTRUM kennt und es ihm daher beliebige, eigene Merkwürdigkeiten auszugeben:

Bei der gestrichelten Linie in der Zeile 20 handelt es sich nicht um das Minuszeichen, sondern um den roten Strich auf der Taste 0, der eine durchgehende Linie erzeugt. In der Variablen p\$ in Zeile 7280 steht ein \$, das durch das Copyright-Zeichen unter der Taste 'P' zu ersetzen ist. Gleiches gilt für die Zeile 7410. POKE 23658,8 sorgt dafür, daß der SPECTRUM ständig im Großbuchstabenmodus bleibt. Dies darf nicht geändert werden. In diesem Sinne also: 'Frohe Ernte'!

Als weiterführende Literatur sei empfohlen:

DER BIO-GARTEN
von Marie-Luise Kreuter,
Büchergilde Gutenberg,
München 1981.

GESUNDER GARTEN
DURCH MISCHKULTUR
von Gertrud Franck,
Südwest-Verlag 1980


```

1 GO TO 8
2 SAVE "FARMER" LINE 1
3 DIM A$(62,15): POKE 23656,0
4 RESTORE 8300: FOR n=1 TO 62
  READ A$(n): NEXT n
5 LET qv=0: LET p=0: LET l
  $=""
15 CLS: PRINT "BITTE WAERLEN:"
16 LET p=0
20 PRINT "1 MISCHKULTURPFLAN
ZEN AUSSUCHEN
-----2 COMPUTERVORSCHEIDN
EGE
-----3 EIGENE BEETGESTA
LTUNG
-----4 LISTE ALLER PFLA
NZENK(PIRTER)"
25 LET e$=INKEY$: IF INKEY$=""
THEN GO TO 25
30 IF e$<"1" OR e$>"4" THEN G
O TO 25
31 IF e$="1" THEN GO TO 40
32 IF e$="2" THEN GO TO 7000
34 IF e$="3" THEN GO TO 9000
35 IF e$="4" THEN FOR i=1 TO
62
36 LPRINT i;"-">"A$(i): NEXT i
38 GO TO 15
40 CLS: INPUT "PFLANZE ? ".B$
  LET B$=B$+" "
  TO 15)
53 RESTORE 8000: FOR N=1 TO 62
  READ A$(N)
52 IF A$(N)=B$ THEN PRINT AT
0,1;"Zu ".B$:"Passen in der Misc
hkultur: ". LET p=100*B$
65 NEXT N
66 IF p=0 THEN PRINT "KEIN EI
NTRAG VORHANDEN: ". PAUSE 100: GO
TO 15
67 GO SUB p
99 GO TO 15
100 PRINT 20: PAPER 4: INK 9:a$
(20): GO SUB qv: RETURN
200 RESTORE 290
205 READ bk: IF LEN STR$ bk=1 T
HEN PRINT "0":
206 PRINT bk: PAPER 4: INK 9:a$
(bk): IF bk<>55 THEN GO TO 205
208 PRINT "weitere Pflanzen:":
210 INPUT "weiter mit ENTER:":z
$
215 FOR i=2 TO 21: PRINT AT i,0
1$: NEXT i
217 PRINT AT 1,0:1$:AT 1,0:"wei
ter Passen:"
219 READ bk: IF LEN STR$ bk=1 T
HEN PRINT "0":
220 PRINT bk: PAPER 4: INK 9:a$
(bk): IF bk<>22 THEN GO TO 219
222 PRINT "nicht dazupassend:":
225 READ bk: PRINT bk: PAPER 2:
INK 9:a$(bk): IF bk<>60 THEN G
O TO 235
240 GO SUB qv: RETURN
250 DATA 61,62,9,13,16,23,27,30
,37,46,39,40,43,48,52,55,15,8,36
,1,25,22
292 DATA 10,19,60
300 RESTORE 390
305 READ bo: IF LEN STR$ bo=1 T
HEN PRINT "0":
310 PRINT bo: PAPER 4: INK 9:a$
(bo): IF bo<>29 THEN GO TO 305
330 PRINT "Aber nicht:":
340 READ bo: IF LEN STR$ bo=1 T
HEN PRINT "0":
350 PRINT bo: PAPER 2: INK 5:a$
(bo): IF bo<>50 THEN GO TO 340
360 GO SUB qv: RETURN
390 DATA 13,21,23,29,9,12,27,60
400 RESTORE 490
405 FOR i=1 TO 2: READ bk: PRI
N T bk: PAPER 4: INK 9:a$(bk): NE

```

```

1
423 GO SUB qv: RETURN
493 DATA 60,61
500 PRINT 10: PAPER 4: INK 9:a$
(10)
513 GO SUB qv: RETURN
603 PRINT 12: PAPER 4: INK 9:a$
(12)
610 GO SUB qv: RETURN
700 PRINT 16: PAPER 4: INK 9:a$
(16)
710 GO SUB qv: RETURN
800 RESTORE 890
820 READ di: IF LEN STR$ di=1 T
HEN PRINT "0":
830 PRINT di: PAPER 4: INK 9:a$
(di): IF di<>60 THEN GO TO 820
850 GO SUB qv: RETURN
890 DATA 9,13,20,23,33,37,46,43
,60
900 RESTORE 990
910 READ er: IF LEN STR$ er=1 T
HEN PRINT "0":
920 PRINT er: PAPER 4: INK 9:a$
(er): IF er<>3 THEN GO TO 910
930 PRINT "Nicht aber:":
940 READ er: IF LEN STR$ er=1 T
HEN PRINT "0":
950 PRINT er: PAPER 2: INK 9:a$
(er): IF er<>50 THEN GO TO 940
960 GO SUB qv: RETURN
990 DATA 12,13,20,23,33,39,40,5
8,29,8
995 DATA 61,19,27,55,60
1000 RESTORE 1050
1005 READ eb: IF LEN STR$ eb=1 T
HEN PRINT "0":
1010 PRINT eb: PAPER 4: INK 9:a$
(eb): IF eb<>5 THEN GO TO 1005
1015 PRINT "Unpassend sind:":
1020 READ eb: IF LEN STR$ eb=1 T
HEN PRINT "0":
1030 PRINT eb: PAPER 2: INK 9:a$
(eb): GO SUB qv: RETURN
1090 DATA 19,23,27,52,60,38,61,5
,20
1100 PRINT 12: PAPER 4: INK 9:a$
(12):"--WINTERZIEBELN "
1110 GO SUB qv: RETURN
1200 RESTORE 1290
1210 READ fe: IF LEN STR$ fe=1 T
HEN PRINT "0":
1220 PRINT fe: PAPER 4: INK 9:a$
(fe): IF fe<>46 THEN GO TO 1210
1230 PRINT "NICHT KOMBINIEREN MI
T:":
1240 READ fe: PRINT fe: PAPER 2:
INK 9:a$(fe): IF fe<>61 THEN G
O TO 1240
1250 GO SUB qv: RETURN
1290 DATA 13,11,57,59,6,38,37,23
,62,9,44,46,55,61
1300 RESTORE 1390
1310 READ gu: IF LEN STR$ gu=1 T
HEN PRINT "0":
1320 PRINT gu: PAPER 4: INK 9:a$
(gu): IF gu<>51 THEN GO TO 1310
1325 PRINT "ES PASSEN NICHT:":
1330 INPUT "weiter mit ENTER:":z
$: FOR i=1 TO 23: PRINT AT i,0:1
$: NEXT i
1340 PRINT AT 1,0:"PASSEN NICHT:
"
1350 READ gu: PRINT gu: PAPER 2:
INK 9:a$(gu): IF gu<>55 THEN G
O TO 1350
1360 GO SUB qv: RETURN
1390 DATA 29,3,9,8,48,60,27,43,2
3,20,12,25,22,61,19,50,51
1395 DATA 16,39,40,55
1400 RESTORE 1490
1410 READ hi: IF LEN STR$ hi=1 T
HEN PRINT "0":
1420 PRINT hi: PAPER 4: INK 9:a$
(hi): IF hi<>9 THEN GO TO 1410
1430 GO SUB qv: RETURN
1490 DATA 19,61,9

```

```

1500 RESTORE 1590
1510 READ ka: PRINT ka: PAPER 4:
INK 9:a$(ka): IF ka<>60 THEN G
O TO 1510
1520 GO SUB qv: RETURN
1590 DATA 20,27,48,60
1600 RESTORE 1690
1610 READ kt: IF LEN STR$ kt=1 T
HEN PRINT "0":
1620 PRINT kt: PAPER 4: INK 9:a$
(kt): IF kt<>61 THEN GO TO 1610
1630 PRINT "nicht zu KARTOFFELN
Passen:":
1640 READ kt: IF LEN STR$ kt=1 T
HEN PRINT "0":
1650 PRINT kt: PAPER 2: INK 9:a$
(kt): IF kt<>55 THEN GO TO 1640
1660 GO SUB qv: RETURN
1690 DATA 20,21,52,22,7,31,54,17
,36,25,19,61
1695 DATA 43,50,48,13,9,55
1700 RESTORE 1790
1710 READ kk: PRINT kk: PAPER 4:
INK 9:a$(kk): IF kk<>58 THEN G
O TO 1710
1720 GO SUB qv: RETURN
1790 DATA 16,39,40,55,58
1800 PRINT 23: PAPER 4: INK 9:a$
(23)
1810 GO SUB qv: RETURN
1900 RESTORE 1990
1910 READ kn: PRINT kn: PAPER 4:
INK 9:a$(kn): IF kn<>16 THEN G
O TO 1910
1920 PRINT "keinesfalls aber:":
1930 READ kn: IF LEN STR$ kn=1 T
HEN PRINT "0":
1940 PRINT kn: PAPER 2: INK 9:a$
(kn): IF kn<>51 THEN GO TO 1930
1950 GO SUB qv: RETURN
1990 DATA 41,20,56,10,24,14,55,4
3,33,13,16
1995 DATA 20,9,61,51
2000 RESTORE 2090
2010 READ ko: PRINT ko: PAPER 4:
INK 9:a$(ko): IF ko<>30 THEN G
O TO 2010
2020 INPUT "weiter mit ENTER:":z
$
2030 FOR i=1 TO 21: PRINT AT i,0
1$: NEXT i
2040 PRINT AT 1,0:"weitere Pfla
nzen:":
2050 READ ko: IF LEN STR$ ko=1 T
HEN PRINT "0":
2055 PRINT ko: PAPER 4: INK 9:a$
(ko): IF ko<>61 THEN GO TO 2050
2060 PRINT "jedoch nicht:":
2065 READ ko: PRINT ko: PAPER 2:
INK 9:a$(ko): IF ko<>19 THEN G
O TO 2065
2070 GO SUB qv: RETURN
2090 DATA 16,40,43,55,52,37,23,6
2,27,15,36,25,22,39,40,46,30,9,8
,1,13,61
2095 DATA 60,10,19
2100 RESTORE 2190
2105 READ ki: IF LEN STR$ ki=1 T
HEN PRINT "0":
2110 PRINT ki: PAPER 4: INK 9:a$
(ki): IF ki<>62 THEN GO TO 2105
2120 GO SUB qv: RETURN
2190 DATA 3,52,51,48,47,43,39,40
,27,20,10,9,21,62
2200 RESTORE 2290
2210 READ k1: PRINT k1: PAPER 4:
INK 9:a$(k1): IF k1<>43 THEN G
O TO 2210
2220 GO SUB qv: RETURN
2290 DATA 13,20,43
2300 RESTORE 2390
2310 READ ks: PRINT ks: PAPER 4:
INK 9:a$(ks): IF ks<>27 THEN G
O TO 2310
2315 INPUT "weiter mit ENTER:":z
$
2320 FOR i=1 TO 21: PRINT AT i,0

```



```

115: NEXT i
2325 PRINT AT 1,0;"aussenden Pas
st er zu."
2330 READ ks: IF LEN STR# ks=1 T
HEN PRINT "0":
2335 PRINT ks: PAPER 4: INK 9:a#
(ks): IF ks<>52 THEN GO TO 2330
2340 PRINT "nicht aber zu."
2345 READ ks: PRINT ks: PAPER 2:
INK 9:a#(ks): IF ks<>48 THEN G
O TO 2345
2350 GO SUB 9v: RETURN
2390 DATA 35,40,24,18,33,13,10,6
0,57,55,51,47,43,29,27,3,21,20,1
2,61,36,8,9,52
2395 DATA 35,48
2400 RESTORE 2490
2405 READ ke: PRINT ke: PAPER 4:
INK 9:a#(ke): IF ke<>40 THEN G
O TO 2405
2410 GO SUB 9v: RETURN
2490 DATA 23,39,40
2500 RESTORE 2590
2505 READ ku: PRINT ku: PAPER 4:
INK 9:a#(ku): IF ku<>43 THEN G
O TO 2505
2510 GO SUB 9v: RETURN
2590 DATA 13,16,20,43
2600 RESTORE 2690
2605 READ kb: PRINT kb: PAPER 4:
INK 9:a#(kb)
2610 PRINT "nicht aber:"
2615 READ kb: PRINT kb: PAPER 2:
INK 9:a#(kb)
2620 GO SUB 9v: RETURN
2690 DATA 29,10
2700 RESTORE 2790
2705 READ la: PRINT la: PAPER 4:
INK 9:a#(la): IF la<>13 THEN G
O TO 2705
2710 PRINT "ER VERTRAGT SICH NI
CHT MIT:"
2715 READ la: IF LEN STR# la=1 T
HEN PRINT "0":
2720 PRINT la: PAPER 2: INK 9:a#
(la): IF la<>61 THEN GO TO 2715
2730 GO SUB 9v: RETURN
2790 DATA 63,48,33,55,47,25,21,2
0,10,62,15,13
2795 DATA 43,9,61
2800 PRINT 19: PAPER 4: INK 9:a#
(19)
2810 GO SUB 9v: RETURN
2900 RESTORE 2990
2905 READ ma: IF LEN STR# ma=1 T
HEN PRINT "0":
2910 PRINT ma: PAPER 4: INK 9:a#
(ma): IF ma<>9 THEN GO TO 2905
2915 PRINT "Nicht jedoch:"
2920 READ ma: PRINT ma: PAPER 2:
INK 9:a#(ma): IF ma<>43 THEN G
O TO 2920
2930 GO SUB 9v: RETURN
2990 DATA 55,23,3,13,58,26,32,16
,9
2995 DATA 48,43
3000 RESTORE 3090
3005 READ mn: PRINT mn: PAPER 4:
INK 9:a#(mn): IF mn<>33 THEN G
O TO 3005
3010 GO SUB 9v: RETURN
3090 DATA 39,40,20,33
3100 RESTORE 3190
3105 READ ne: PRINT ne: PAPER 4:
INK 9:a#(ne): IF ne<>34 THEN G
O TO 3105
3110 GO SUB 9v: RETURN
3190 DATA 16,34
3200 PRINT 29: PAPER 4: INK 9:a#
(29)
3210 GO SUB 9v: RETURN
3300 RESTORE 3390
3310 READ mo: IF LEN STR# mo=1 T
HEN PRINT "0":
3315 PRINT mo: PAPER 4: INK 9:a#
(mo): IF mo<>23 THEN GO TO 3310

```

```

3320 GO SUB 9v: RETURN
3390 DATA 63,55,57,39,40,30,27,1
9,8,46,45,47,42,44,23
3400 RESTORE 3490
3405 READ ob: PRINT ob: PAPER 4:
INK 9:a#(ob): IF ob<>52 THEN G
O TO 3405
3420 GO SUB 9v: RETURN
3490 DATA 31,19,52
3500 RESTORE 3590
3505 READ pe: PRINT pe: PAPER 4:
INK 9:a#(pe): IF pe<>55 THEN G
O TO 3505
3510 PRINT "nicht aber zu."
3515 READ pe: PRINT pe: PAPER 2:
INK 9:a#(pe)
3520 GO SUB 9v: RETURN
3590 DATA 51,55,23
3600 RESTORE 3690
3605 READ pf: PRINT pf: PAPER 4:
INK 9:a#(pf): IF pf<>23 THEN G
O TO 3605
3610 GO SUB 9v: RETURN
3690 DATA 16,20,23
3700 RESTORE 3790
3705 READ ps: IF LEN STR# ps=1 T
HEN PRINT "0":
3710 PRINT ps: PAPER 4: INK 9:a#
(ps): IF ps<>61 THEN GO TO 3705
3720 GO SUB 9v: RETURN
3790 DATA 33,55,51,47,43,39,43,2
0,12,8,61
3800 PRINT 12: PAPER 4: INK 9:a#
(12)
3810 GO SUB 9v: RETURN
3900 RESTORE 3990
3905 READ ra: IF LEN STR# ra=1 T
HEN PRINT "0":
3910 PRINT ra: PAPER 4: INK 9:a#
(ra): IF ra<>55 THEN GO TO 3905
3915 PRINT "sie gehoert nicht z
wischen:"
3920 READ ra: PRINT ra: PAPER 4:
INK 9:a#(ra)
3925 GO SUB 9v: RETURN
3990 DATA 24,3,20,9,10,21,23,30,
33,37,46,52,55,10
4000 PRINT "siehe unter RADIESCH
EN (Nr.39)"
4020 GO SUB 9v: RETURN
4100 PRINT 19: PAPER 4: INK 9:a#
(19): AT 2,13:"utd"
4102 PRINT 54: PAPER 4: INK 9:a#
(54)
4105 GO SUB 9v: RETURN
4200 RESTORE 4290
4205 FOR i=1 TO 2: READ rm: PRIN
T rm: PAPER 4: INK 9:a#(rm): NEX
T i
4220 GO SUB 9v: RETURN
4290 DATA 33,44
4300 RESTORE 4390
4305 READ rb: IF LEN STR# rb=1 T
HEN PRINT "0":
4310 PRINT rb: PAPER 4: INK 9:a#
(rb): IF rb<>55 THEN GO TO 4305
4320 PRINT "Nicht jedoch zu:"
4325 READ rb: PRINT rb: PAPER 2:
INK 9:a#(rb): IF rb<>29 THEN G
O TO 4325
4330 GO SUB 9v: RETURN
4390 DATA 60,20,21,37,19,13,8,25
,22,61,23,52,55
4395 DATA 27,29
4400 RESTORE 4490
4410 FOR i=1 TO 3: READ sa: PRIN
T sa: PAPER 4: INK 9:a#(sa): NEX
T i
4420 GO SUB 9v: RETURN
4490 DATA 12,33,12
4500 PRINT "aehnlich wie LAUCH,
siehe Nr.27:"
4505 GO SUB 9v: RETURN
4600 PRINT "siehe bei PFUECKSA
L AT Nr.37 nach !!!"
4605 GO SUB 9v: RETURN

```

```

4700 RESTORE 4790
4705 READ sw: PRINT sw: PAPER 4:
INK 9:a#(sw): IF sw<>33 THEN G
O TO 4705
4720 GO SUB 9v: RETURN
4790 DATA 27,23,37,21,33
4800 RESTORE 4890
4805 READ se: IF LEN STR# se=1 T
HEN PRINT "0":
4810 PRINT se: PAPER 4: INK 9:a#
(se): IF se<>15 THEN GO TO 4805
4820 PRINT "sie soll nicht zu:"
4825 READ se: PRINT se: PAPER 2:
INK 9:a#(se): IF se<>29 THEN G
O TO 4825
4830 GO SUB 9v: RETURN
4890 DATA 55,61,27,21,20,2,13,15
,16,23,29
4900 PRINT 20: PAPER 4: INK 9:a#
(20)
4905 GO SUB 9v: RETURN
5000 RESTORE 5090
5010 READ so: PRINT so: PAPER 4:
INK 9:a#(so)
5020 PRINT "ABER NICHT ZU:"
5025 READ so: PRINT so: PAPER 2:
INK 9:a#(so)
5030 GO SUB 9v: RETURN
5090 DATA 13,16
5100 RESTORE 5190
5105 READ sp: PRINT sp: PAPER 4:
INK 9:a#(sp): IF sp<>23 THEN G
O TO 5105
5110 PRINT "Passt nicht zu:"
5120 FOR i=1 TO 2: READ st: PRIN
T st: PAPER 2: INK 9:a#(st): NEX
T i
5130 GO SUB 9v: RETURN
5190 DATA 55,35,13,37,23,19,60
5200 RESTORE 5290
5205 READ si: PRINT si: PAPER 4:
INK 9:a#(si): IF si<>34 THEN G
O TO 5205
5220 GO SUB 9v: RETURN
5290 DATA 10,43,55,53,39,40,21,2
0,16,34
5300 PRINT "siehe (mit Einschnae
n kaufen bez.Hoehe und Platz) unt
er BUCHEN (Nr.03) nach"
5310 GO SUB 9v: RETURN
5400 RESTORE 5490
5405 PRINT AT 8,0:" AT 2,0:"a
ls Schaedlingsbekaempfer gut zu"
5410 FOR i=1 TO 3: READ ta: PRIN
T ta: PAPER 4: INK 9:a#(ta): NEX
T i
5420 GO SUB 9v: RETURN
5490 DATA 41,16,55
5500 RESTORE 5590
5505 READ to: PRINT to: PAPER 4:
INK 9:a#(to): IF to<>17 THEN G
O TO 5505
5510 PRINT "nicht Passend sind:"
5515 INPUT "weiter mit ENTER:":z
#
5520 FOR i=1 TO 21: PRINT AT 1,0
,15: NEXT i
5525 PRINT AT 1,0:"Passen nicht:
"
5530 READ to: IF LEN STR# to=1 T
HEN PRINT "0":
5535 PRINT to: PAPER 2: INK 9:a#
(to): IF to<>9 THEN GO TO 5530
5540 GO SUB 9v: RETURN
5590 DATA 33,52,48,43,39,40,57,3
7,35,23,29,27,21,20,19,17
5595 DATA 16,13,12,9
5600 PRINT 19: PAPER 4: INK 9:a#
(19): PRINT "zur Schaedlingsabwe
hr"
5605 GO SUB 9v: RETURN
5700 RESTORE 5790
5705 READ zi: PRINT zi: PAPER 4:
INK 9:a#(zi): IF zi<>17 THEN G
O TO 5705
5710 GO SUB 9v: RETURN

```



```

5790 DATA 55,53,23,33,12
5800 RESTORE 5690
5805 FOR i=1 TO 4: READ zw: PRINT
      T zw: PAPER 4: INK 0: a$(zw): NEXT
      i
5820 GO SUB 3v: RETURN
5890 DATA 60,53,29,17
5900 PRINT "gehört zu den ZIC
      HORIE-Salaten siehe unter Nr.57:
      "
5910 GO SUB 3v: RETURN
6000 RESTORE 6090
6005 READ zw: IF LEN STR$ zw=1 T
      HEN PRINT "3";
6010 PRINT zw: PAPER 4: INK 0: a$(
      zw): IF zw<4 THEN GO TO 6005
6015 PRINT "nicht zusammenpflanzen
      er mit:"
6020 READ zw: IF LEN STR$ zw=1 T
      HEN PRINT "3";
6025 PRINT zw: PAPER 2: INK 9: a$(
      zw): IF zw<9 THEN GO TO 6020
6040 GO SUB 3v: RETURN
6090 DATA 33,43,57,23,13,10,8,15
      ,4
6095 DATA 20,3,9
6100 PRINT "GRUNDSÄTZLICH GILT
      GLEICHES WIEZU BOHNEN (Nr.03) U.
      STANGEN- BOHNEN (Nr.53) GESAG
      T WURDE!"
6110 GO SUB 3v: RETURN
6200 RESTORE 6290
6210 FOR i=1 TO 3: READ en: PRIN
      T en: PAPER 4: INK 9: a$(en): NEXT
      i
6220 GO SUB 3v: RETURN
6290 DATA 12,27,20
6299 STOP
6300 INPUT "SOLL ICH UNTER DEN G
      ERNANTEN AR-TEN WEITER NACHSEHEN
      (J/N)?": z$: IF z$="N" THEN RET
      URN
6310 INPUT "unter welcher Nummer
      (2-stellig):": LINE f$: IF LEN f
      $<2 THEN GO TO 6310
6320 IF f$(1)<"0" OR f$(1)>"9" T
      HEN GO TO 6310
6322 IF f$(2)<"0" OR f$(2)>"9" T
      HEN GO TO 6310
6325 IF VAL f$>62 THEN GO TO 63
      10
6330 CLS: PRINT "Zu "a$(VAL f$
      )"Pflanzen:"
6350 GO SUB VAL f$*100
6360 RETURN
7000 CLS: PRINT "ES FOLGEN 3 DE
      ISPIELE FUER GLTE MISCHKULTUREN.
      DIE NUMMERN EN
      TSPRECHEN DEN CODEZAHLEN IM PROC
      RAMM."
7010 INPUT "weiter mit ENTER:": z
      $
7020 CLS: PRINT "VORSCHLAG Nr.
      1 (Beetbr.=1,20 m)"
7030 PLOT 6,160: DRAW 180,0: DRA
      W 0,-140: DRAW -180,0: DRAW 0,14
      0
7040 PRINT AT 20,0:"PFLANZEN S: E
      IM FRUEHJAHR IN DERMITTE DES BE
      ETES EINE REIHE KOHL": PAUSE 100
7042 LET end=20: LET r1=10: LET
      r2=r1: LET p$="*20": GO SUB 7900
7055 PRINT AT 20,0:"RECHTS U. LI
      NKS DAVON JE EINE GEMISCHTE RE: H
      E RADIESCHEN/KRESSE": PAUSE 200
7057 LET end=17: LET r1=8: LET r
      2=12: LET p$="*24/39": GO SUB 79
      00
7065 PRINT AT 20,0:"DANEBEN JE E
      INE REIHE KUFERLAT": 1$: PAUSE
      200
7070 LET end=20: LET r1=6: LET r
      2=14: LET p$="*23": GO SUB 7900
7080 PRINT AT 20,0:"ES FOLGEN JE
      EINE REIHE KOHLRABIU. GEMISCH":
      KRESSE/RADIESCHEN "

```

```

7085 LET end=20: LET r1=4: LET r
      2=16: LET p$="*21": GO SUB 7900
7090 LET end=17: LET r1=2: LET r
      2=18: LET p$="*24/39": GO SUB 79
      00
7100 LET d$="RAD./KR.": PRINT AT
      2,24:d$:AT 8,24:d$:AT 12,24:d$:
      AT 16,24:d$
7110 PRINT AT 4,24:"KOHLRABI":AT
      16,24:"KOHLRABI"
7115 PRINT AT 8,24:"KOPFSAL.":AT
      14,24:"KOPFSAL."
7120 PRINT AT 10,24:"KOHL"
7130 PRINT AT 20,0:"SFAETER KOEN
      NEN NOCH WINTERSALA-TE ODER SPIN
      AT NACHGESAET WERDEN"
7140 INPUT "VORSCHL. DRUCKEN?(J
      /N)": z$: IF z$<"J" AND z$<"N"
      THEN GO TO 7140
7150 IF z$="J" THEN COPY: FOR
      i=1 TO 5: LPRINT: NEXT i
7200 CLS: PRINT "VORSCHLAG Nr.2
      (Beetbr. 1,20 m)"
7210 PLOT 6,160: DRAW 160,0: DRA
      W 0,-64: DRAW -160,0: DRAW 0,64
7220 LET end=19: LET r1=4: LET r
      2=0: LET p$="*": GO SUB 7900
7230 LET r1=2: LET r2=6: LET p$=
      "": GO SUB 7900
7240 PRINT AT 2,21:"MOEHREN(33)"
      AT 6,21:"MOEHREN(33)"
7245 PRINT AT 4,21:"ZIEBEL(6E)"
      AT 8,21:"ZIEBEL(6E)"
7250 PRINT AT 10,0:"DIESE KOMEIN
      ATION SCHUETZT SICH GEGENSEITIG
      VOR SCHADELNGEN. IM SPAETSCHE
      ER WERDEN DANN AUF DEM GLEICHEN
      DEET AUSGESAET."
7260 PAUSE 200: LET p$="0": GO S
      UB 7900
7270 PRINT AT 2,21:"FELD(11)"
      AT 6,21:"FELD(11)"
7280 LET r1=4: LET r2=8: LET p$=
      "": GO SUB 7900
7290 PRINT AT 4,21:"SPINAT (5E)"
      AT 8,21:"SPINAT (5E)"
7300 INPUT "VORSCHLAG DRUCKEN (J
      /N)": z$
7305 IF z$="J" THEN PRINT AT 2,
      1:"FRUEH:MOEHR,SFAET->":AT 4,1:"
      FRUEH:ZIEB,SPAT->": COPY: FOR
      i=1 TO 5: LPRINT: NEXT i
7310 IF z$<"J" AND z$<"N" THEN
      GO TO 7300
7350 CLS: PRINT "AUCH MIT KARTO
      FFELN LASSEN SICH GUTE MISCHKULT
      UREN BILDEN ""VORSCHLAG Nr.3 (
      EBENFALLS FUER EINE BEETBREITE
      VON 1,20 MTR.):""FRUEH:KARTOFFE
      LN, SPINAT, RADIESCHEN, KAPUZIN
      ERKRESSE, GRUENKOHLEN LAUCH:"
7360 INPUT "ENTER TIPPEN:": z$
7400 CLS: PRINT "VORSCHLAG Nr.
      3:"
7410 PRINT AT 2,0: PAPER 4: INK
      0:"KAPUZINERKRESSE(17)*****
      *****":1$: "RADIESCHEN(3S).....
      ....":1$: "KARTOFFELN(15).....
      0000000000000000":1$: "SPINAT(5E)
      *****":1$: "KARTO
      FFELN(16)0000000000000000":1$:
      "RADIESCHEN(39).....
      .":1$: "KAPUZINERKRESSE(17)*****
      *****"
7450 PRINT "->KARTOFFELN BRAUCHE
      N NICHT AN- GEHAEUFELT WERDEN.
      <- Nr. 17 IST SALAT UND KARTOF
      FELNSCHUTZ GLEICHZEITIG.NACH ER
      NTE VON 16 KANN GRUENKOHLE UND L
      AUCH GE- PFLANZT WERDEN."
7455 INPUT "VORSCHLAG DRUCKEN (J
      /N)": z$: IF z$="J" THEN COPY:
      FOR i=1 TO 5: LPRINT: NEXT i
7899 GO TO 15
7900 FOR i=1 TO end: PRINT AT r1
      ,1: PAPER 4: INK 0: p$:AT r2,1: p$
      : PAUSE 20: NEXT i

```

```

7910 INPUT "ENTER TIPPEN": z$: RE
      TURN
8000 DATA "BEIFU3S","BLUMENKOHLE"
      ,"BOHNEN","BOHNENKRAUT","BORETSCH"
      ,"CHICOREE","DICKE BOHNEN","DI
      LI","FARSEN","ERDAFFERN","FELDSAL
      LAT","FENCHEL","GURKEN","HIMBEER
      EN","KAMILLE","KARTOFFELN","KAPU
      ZINERKRESSE","KERBEL","KNOBLAUCH"
      ,"KOHLLARTEN","KOHLRABI","KORIAN
      DER"
8010 DATA "KOPFSALAT","KRESSE","K
      UEBIMEL","KUEBIS","LAUCH","LILI
      EN","MAIS","MANGOLD","MEERRETTIC
      H","MELONEN","MOEHREN","OBSTBAE
      UME","PETERSILIE","PFEFFERMINZE",
      "PFLUECKSALAT","RADICCHIO","RADI
      ESCHEN","RETTICH","ROSEN"
8020 DATA "ROSMARIN","ROTE BEETE"
      ,"SALBEI","SCHNITTlauch","SCHNIT
      Tsalat","SCHWARZWURZELN","SELLER
      IE","SENF","SONNENELUMEN","SPAR
      GEL","SPINAT","STANGENBOHNEN","TA
      GETES","TOMATEN","TULPEN","ZICHU
      RIE","ZUCCHINI","ZUCKERHUT","ZWI
      EBELN","BUECHBOHNEN"
8030 DATA "ENDIVIE"
9000 CLS: PRINT "VERMEIDEN SIE
      AUF JEDEN FALL FOLGENDE KOMBI
      NATIONEN:"
9010 PRINT "a$(12); "<->";a$(55);
      a$(61); "<->";a$(50);a$(20); "<->";
      a$(60);a$(55); "<->";a$(9);a$(9);
      "<->";a$(3);a$(16); "<->";a$(50);
      a$(20); "<->";a$(45)
9020 INPUT "ENTER TIPPEN": z$
9030 CLS: INPUT "EET Nr. ODER
      LAGE?": s$: PRINT "EET: ";s$
9040 INPUT "PFLANZZEIT?": h$: PR
      INT "("h$;")"
9050 INPUT "WIEVIELE REIHEN (max
      . 15)?": y$
9052 FOR i=1 TO LEN y$: IF y$(i)
      <"0" OR y$(i)>"9" THEN GO TO 90
      50
9053 NEXT i
9054 LET y=VAL y$
9055 IF y<1 OR y>15 THEN GO TO
      9050
9060 PLOT 20,175-((y*8)+19): DRA
      W 160,0: DRAW 0,(8*y+4): DRAW -1
      60,0: DRAW 0,-(8*y+4)
9070 LET u$="": LET x=2
9080 PRINT #0:AT 0,0:"REIHENWUHL
      mit:6/7 (F=FFERTIG)"
9090 LET x$=INKEY$
9095 LET z=x
9100 LET x=x+(INKEY$="6" AND x<=
      y)-((INKEY$="7" AND x>=3)
9105 PRINT AT z,0: "AT x,0,u$
9110 IF x$="P" THEN GO TO 9150
9115 PAUSE 5: GO TO 9090
9150 INPUT "WELCHE PFLANZE(N)?"
      ,s$: IF s$="" THEN GO TO 9200
9160 LET s$=s$+"
      "X TO 19): PRINT AT x,3:s$
9170 GO TO 9080
9200 PRINT AT 10,0:"BITTE PRUEFE
      H,OB FOLGENDE FEHL- KOMBINATIONE
      N VORHANDEN SIND:"
9250 RESTORE 9500
9260 READ fa,fb: PRINT AT 21,0:
      PAPER 2: INK 9: a$(fa);a$(fb)
9270 INPUT "VORHANDEN?(J/N)": z$
      IF z$="J" THEN GO TO 9070
9280 IF z$<"J" THEN GO TO 9250
9290 INPUT "BEET DRUCKEN?(J/N)":
      z$
9300 IF z$="J" THEN FOR i=1 TO 18
      TO 21: PRINT AT i,0:1$: NEXT i: CO
      PY: FOR i=1 TO 5: LPRINT: NEXT
      i
9350 PRINT AT 10,0:"PROGRAMMENDE
      ":AT 11,0: "(RUN=RESTART)"
9500 DATA 12,55,51,60,20,60,55,9
      ,9,3,16,50,20,49
9510 STOP

```


Rolf Keller

I/O-Techniken:

Drucken per Interrupt

Die Verwendung von Interrupts ist eine äußerst elegante Methode, Peripheriegeräte schnell und effizient zu bedienen. Die CPU verdrängt keine Zeit in Warteschleifen oder durch Statusabfragen, wenn sie aber gebraucht wird, dann ist sie da. Da fragt man sich doch, warum diese wundervolle Methode nicht ausschließlich angewendet wird. Nun, sie stellt bestimmte Anforderungen an die verwendete Hardware und Software. Auch mit Interrupts bestens vertraute Programmierer gewinnen dabei des öfteren noch 'völlig neue' Erkenntnisse.

Ein Interrupt (Unterbrechung) ist eine Programmverzweigung, üblicherweise in Form eines Unterprogrammrufes. Ähnlich wie bei einem GOSUB in BASIC-Programmen oder einem CALL in Assembler, wird ein Unterprogramm ausgeführt und in das übergeordnete, normalerweise aufrufende Programm, mit einem RETURN genau dorthin zurückgesprungen, wo es verlassen wurde. Eine derartige Verzweigung erfolgt per Interrupt aber, ohne daß im laufenden Programm ein diesbezüglicher Aufruf programmiert worden ist. Wie das?

Die CPU verfügt hardwaremäßig über (mindestens) eine Interrupt-Request-Leitung (Interrupt-Anforderung). Wird diese Leitung auf das (je nach Prozessor) für eine Interruptanforderung vorgeschriebene Potential gesteuert, führt die CPU eine Programmverzweigung auf eine ganz bestimmte Adresse durch. Diese Adresse ist prozessorabhängig, bei manchen lassen sich mehrere, verschiedene einstellen. Dann kann man seine Interrupt-Service-Routine direkt dort ablegen und für verschiedene Peripherie-Geräte geeignete Service-Routinen direkt anspringen. Laufen alle Interrupts über nur eine Verzweigungsadresse, kann man dort eine Suchroutine unterbringen, die dann her-

austüftelt, welches Peripherie-Gerät bedient werden möchte. Dann verzweigt man von dieser Suchroutine zur eigentlichen Service-Routine. Und wenn alles erledigt ist, wird das unterbrochene Programm, aus dem ohne sein Wissen verzweigt wurde, still und heimlich dort fortgesetzt, wo es verlassen wurde.

Die Methode, per externer Hardware zu einer bestimmten Adresse zu verzweigen, wenden Sie übrigens stets dann an, wenn Sie bei Ihrem Rechner einen RESET durchführen. Dabei wird allerdings nicht in ein übergeordnetes Programm zurückgekehrt, sondern nur die RESET-Adresse angesprungen und das Programm, das dort liegt, ausgeführt.

Was macht den Umgang mit Interrupts nun so knifflig? Interrupts treten im allgemeinen asynchron auf, das heißt, man weiß nicht genau wann. Darin liegt letztlich ja der Sinn der Interrupttechnik: auf zeitlich nicht exakt vorhersehbare Ereignisse umgehend zu reagieren, ohne durch Warteschleifen und dauernd eingestreute Abfragen, ob das Ereignis wohl eingetreten ist, Zeit für die Ausführung eines anderen Programmes zu vergeuden. Diese Unvorhersehbarkeit ist aber nicht unproblematisch. Da die Service-Routine irgend etwas

tut (sonst gäbe es sie wohl nicht), verändert sie in bestimmter Weise auch das unterbrochene Programm. So könnte in einem Ein-/Ausgabepuffer 'plötzlich' ein Zeichen mehr oder weniger enthalten sein.

Wenn dies das unterbrochene Programm nicht mitbekommt, ergeben sich ganz sicher Probleme. In ähnlicher Weise muß man auch Sorgfalt walten lassen, wenn man (auf Maschinenebene) durch Interrupt Routinen Registerinhalte verändert. Hier sollte man grundsätzlich alle Register, auf die in der Interruptroutine zugegriffen wird, vorübergehend auf dem Stack ablegen.

Und so gibt es noch eine Reihe weiterer Fehlermöglichkeiten, die die Interrupt-Technik sozusagen in Verruf gebracht haben. Aber Fehler sind schließlich dazu da, daß man sie macht, findet, behebt und daraus lernt. Und da nachweislich viele, viele Interrupt-Programme erdacht und zum Laufen gebracht wurden, kann es so schlimm nun auch wieder nicht sein. Besonders einfach wird es, wenn man über BASIC-Versionen verfügt, die die Interruptprogrammierung unterstützen.

Einfachste Lösung

In BASIC-Versionen, die das Arbeiten mit Interrupts unterstützen, kann man beispielsweise ein Programm wie in Bild 1 schreiben.

In Zeile 11 wird die Zeile 100 zur Anfangsadresse der Interruptroutine erklärt. Außerdem wird die Interruptursache vorab definiert: es soll eine Interruptanforderung gestellt werden, wenn die Taste (KEY) 3 betätigt wird. Es ist klar, daß

```
10 REM Hauptprogramm
11 ON KEY $2 GOSUB 100
12 . . .
22 . . .
23 . . .
. . .
99 GOTO . . .
100 REM Interruptroutine
101 PRINT "TASTE 3"
102 RETURN
103 END
```

Bild 1. Interrupts zur Tastaturabfrage in BASIC

die Schaltung des Rechners diese Möglichkeit vorsehen muß.

In Bild 2 wird das Prinzip einer solchen Schaltung gezeigt. Die Tastatur wird nicht per Programm laufend abgefragt. Das Tastaturinterface verfügt über 'eingebaute Intelligenz' und kann per Software so eingestellt werden, daß es die Betätigung einer bestimmten Taste (hier: 3) ohne CPU-Hilfe selbst

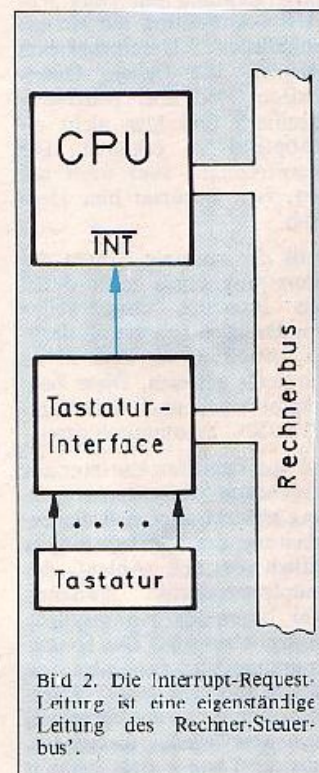


Bild 2. Die Interrupt-Request-Leitung ist eine eigenständige Leitung des Rechner-Steuersbus.

erkennen. Es meldet diese Betätigung 'hinter dem Rücken' des laufenden Programms über die Leitung INT (Interrupt-Request) direkt an die CPU.

Die Anweisung in Zeile 11 hat nur eine Initialisierungsfunktion; sie fragt die Tastatur nicht ab und bewirkt auch keine Verzweigung. Aber die Taste 3 ist ab sofort 'scharf gemacht'. Nach Bearbeitung der Zeile 11 läuft das Programm weiter in den Zeilen 12 bis 99, dem 'Hauptprogramm'. Das Hauptprogramm kann beliebig lange laufen.

Falls nun, während das Hauptprogramm läuft, die Taste 3 betätigt wird, so wird von dem Tastaturinterface durch Aktivierung der Leitung INT eine Interruptanforderung an die

CPU gestellt. Nehmen wir an, das Hauptprogramm wäre zu diesem Zeitpunkt gerade mitten in Zeile 22. Die Zeile 22 wird dann noch fertig bearbeitet werden, danach wird automatisch der Sprung (GOSUB 100) zur Interruptroutine ausgeführt.

Die Interruptroutine gibt den Text 'Taste 3' aus, danach läuft sie auf RETURN. Die RETURN-Anweisung, die eine Interruptroutine beendet, ist nicht identisch mit einer RETURN-Anweisung, die ein gewöhnliches Unterprogramm beendet. Die 'feinen Unterschiede' sind aber prozessor-spezifisch und hier nicht erschöpfend zu erklären. Die Auswirkungen sind nach außen, zum Benutzer hin, identisch.

Falls der normale Ablauf des Hauptprogramms nicht durch den Interrupt unterbrochen worden wäre, so wäre als nächste Anweisung die Zeile 23 an der Reihe gewesen. Diese Zeile kommt nun, nach Interrupt-RETURN, automatisch dran.

Die von dem Tastaturinterface ausgehende Interruptanforderung bewirkt also, daß die Bearbeitung der Interruptroutine zeitlich in den Ablauf des Hauptprogramms zwischen zwei beliebigen Anweisungen eingeschoben wird. Das Hauptprogramm wird für einige Zeit (Bearbeitungszeit der Interruptroutine) unterbrochen, dann aber wieder fortgesetzt. Falls die Taste 3 nicht betätigt wird, so wird die Interruptroutine 100 niemals durchlaufen.

Interruptquellen

Die möglichen Interruptquellen sind sehr vielfältig und hängen stark von der Rechnerhardware ab. Außer den durch Interfaces ausgelöst werden kann man noch Interrupts, die durch Timer-Bausteine, Single-Step-Schaltungen, Netzausfall, Fehler im Programmablauf oder durch spezielle Programmbefehle (Softwareinterrupts) ausgelöst werden.

Prinzipiell kann jedes Interface einen Interrupt auslösen. Unter welchen Bedingungen es dies tut, wird durch seinen Typ, durch die Schaltung des Rechners und durch die softwaremäßige Initialisierung im voraus bestimmt. Ob und wann diese Bedingungen eintreten, ist, wie

bereits erwähnt, nicht genau bekannt. Im Programm von Bild 1 (Tastaturinterrupt) hängt das beispielsweise vom freien Willen des Bedieners ab.

Der einfachste Einsatz von Interrupts für I/O-Zwecke läßt sich mit Timern verwirklichen.

Pseudo-I/O-Interrupts

Ein Timer (im Rechner eingebauter Zeitgeber) kann nach entsprechender Initialisierung durch ein Programm in regelmäßigen Zeitabständen einen Interrupt auslösen.

Da der Timer unabhängig von der CPU exakt wie eine Uhr läuft, weiß man, wann er das Hauptprogramm unterbricht. Aber wo innerhalb des Programms dies geschieht, bleibt trotzdem praktisch unbekannt. Um es herauszufinden, müßte man die Bearbeitungszeiten aller Anweisungen genau kennen und darüber eine Buchführung anlegen. Bei bedingten Sprüngen (IF ... THEN GOTO ...) wäre dabei die unterschiedlichen möglichen Wege zu berücksichtigen. Bei Rechenoperationen müßte bedacht werden, daß sie je nach Wert der Operanden verschieden lange dauern können. Aber all dieses würde für eine Vorhersage immer noch nicht ausreichen, falls der Rechner auf Ereignisse von außen (Peripheriegeräte, Bedieneraktionen usw.) reagiert.

Der Einsatz von Timer-Interrupts ermöglicht es, trotz der eben erwähnten Ungewißheiten, bestimmte Aktionen (die in der Timer-Interruptroutine programmiert werden) in regelmäßigen Zeitabständen durchzuführen.

Im ersten Teil des Beitrags (Heft 3/84) wurde die Notwendigkeit dargelegt, bei der programmgesteuerten I/O-Technik mit Pufferung ein bestimmtes Unterprogramm (dort mit UP 2100 bezeichnet) in gewissen Zeitabständen unabhängig vom Hauptprogramm aufzurufen. Dies läßt sich mit Timer-Interrupts gut bewerkstelligen. Man legt einfach den erwähnten UP-Aufruf in eine Timer-Interruptroutine, die extra für diesen Zweck eingerichtet wird. Die Interrupts bei dieser Methode werden 'Pseudo-I/O-Interrupts' genannt. Einerseits

dienen sie zu I/O-Zwecken, andererseits werden sie aber nicht durch einen I/O-Baustein, sondern durch einen Timerbaustein erzeugt.

Die entsprechende Weiterentwicklung eines Programms aus dem ersten Teil des Beitrags zeigt Bild 2. Das Prinzip der Pufferung bleibt dabei vollständig erhalten. Der Aufruf 'GOSUB 2100' im UP 2000 ist entfallen, da er ja (in Form der Timer-Interruptroutine UP 3100) sowieso laufend ausgeführt wird.

Bei allen uns bekannten BASIC-Versionen, die Interrupts in dieser Form kennen, kann die Interruptroutine nur zwischen zwei Programmzeilen durchlaufen werden. Zwischen zwei Anweisungen, die in derselben Programmzeile stehen, gibt es keine Unterbrechung. Die Warteschleife in Zeile 3001 besteht nur aus einer einzigen Zeile. Man könnte nun denken, daß dies während des Ablaufs der Schleife den Ablauf der Interruptroutine verhindert, da es ja keine zwei Zeilen gibt, zwischen denen sie eingeschoben werden könnte. Dies ist zum Glück nicht der Fall: BASIC betrachtet den Sprung GOTO 3001 so, als führe er zu einer neuen Zeile. Diese neue Zeile ist nun zufällig wieder dieselbe Zeile.

Eine weitere Änderung findet sich in der Initialisierungsroutine UP 3300. Hier ist mit Zeile 3304 die Initialisierung für die Timer-Interruptroutine ergänzt worden: der Timer-Interrupt soll alle 35 msec erfolgen und zur Interruptroutine 3100 führen. Verwendet wird der Timer Nr. 1. Wichtig ist, daß die Timer-Initialisierung erst am Ende der gesamten Initialisierung erfolgt. Es könnte sonst passieren, daß der allererste Timer-Interrupt erfolgt, bevor noch die gesamte Initialisierung abgeschlossen ist. Die Timer-Interruptroutine greift aber sowohl auf das Interface als auch auf den Puffer zu. Sind diese beiden dann noch nicht initialisiert, so kann ein Fehler entstehen. Bei diesem Programm ist vorausgesetzt, daß der Timer sich nach Ablauf des Zeitintervalls selbst neu startet.

Ein ganz wichtiger Punkt ist, daß die Interruptroutine das Hauptprogramm überall unterbrechen kann; dies ist im Programmlisting aber nicht direkt

zu sehen. Sie benutzt und verändert teilweise Variablen, die im Hauptprogramm verwendet werden. Falls man dies bei der Programmierung übersieht, kommt es zu den 'unerklärlichen' Fehlern, von denen eingangs bereits gewarnt wurde.

Ein Beispiel:

```
200 FOR Z=1 TO 50
201 FELD(Z)=0
202 NEXT Z
```

Dieses Programmstück ist, für sich betrachtet, vollkommen in Ordnung. Wird es aber in einem Hauptprogramm verwendet, das die Interruptroutine 3100 (Bild 3) enthält, so kommt es zu Fehlern. Wenn die Interruptroutine zwischen den Zeilen 200 und 202 eingeschoben wird (was man nicht verhindern kann), so wird Z in unzulässiger Weise verändert. Um dies von vornherein zu vermeiden, ist es gute Programmier-technik, in jeder Interruptroutine (und in den von ihr aufgerufenen UP's!) keine Variablen zu verwenden, die an anderer Stelle im Hauptprogramm verwendet werden. Dies sollte man selbst dann tun, wenn man genau weiß (zu wissen glaubt!), daß kein Fehler zu befürchten ist.

Es gibt jedoch Variablen, die zwangsläufig sowohl in der Interruptroutine als auch im Hauptprogramm verwendet werden müssen. Im Beispiel von Bild 3 ist dies die Variable PUFFERS\$. In irgendeiner Variablen muß man ja Daten austauschen. Obwohl in diesem Fall PUFFERS\$ in beiden Programmteilen exakt die gleiche Bedeutung hat, gibt es trotzdem Fehlermöglichkeiten. Nehmen wir dazu an, im UP 3000 von Bild 3 hätte man Zeile 3002 in zwei Zeilen codiert:

```
3002 B$ = PUFFERS$
3003 PUFFERS$ = B$ + A$
```

Wenn dann die Interruptroutine zwischen diesen beiden Zeilen durchlaufen wird, kann sie den String PUFFERS\$ durch eine vollführte Ausgabe verkürzen. B\$ bleibt davon aber unberührt, so daß in Zeile 3003 ein Fehler entsteht, der zu Zeichenverdopplungen bei der Ausgabe führt. Dieses Beispiel erscheint zwar gewaltsam konstruiert (ist es auch!), aber in der Praxis sind ja die Programme nicht immer so übersichtlich. Wenn man dann nach zwei Monaten die üblichen Programmverbesserungen vornimmt, sollte man


```

3000 REM Ausgabe AS -> PUFFER#
3001 IF LEN(AS) <= LEN(PUFFER#) > 255 THEN GOTO 3001
3002 PUFFER# = PUFFER# + AS
3004 RETURN

```

```

3100 REM Timer-Interrupt: PUFFER# -> Drucker
3101 IF LEN(PUFFER#) = 0 THEN GOTO 3106
3102 Z = ASC(PUFFER#)
3103 GOSUB 3200
3104 IF S=0 THEN GOTO 3106
3105 PUFFER# = RIGHT$(PUFFER#, LEN(PUFFER#)-1)
3106 RETURN

```

```

3200 REM Drucker-Treiber
3201 S = 0
3202 IF (INP(0) AND 810) <> 4081 THEN GOTO 3205
3203 S = 1
3204 OUT 4,2
3205 RETURN

```

```

3300 REM Initialisierung
3301 OUT 6,0 : OUT 6,0 : OUT 5,0 : OUT 6,8H40
3302 OUT 6,8H0F : OUT 6,8H27
3303 PUFFER# = ""
3304 ON TIMER #1.55 GOSUB 3100
3305 RETURN

```

Bild 3. I/O mit Pufferung und Timer-Interrupt

```

4000 REM Ausgabe AS -> PUFFER#
4001 IF LEN(AS) <= LEN(PUFFER#) > 255 THEN GOTO 4001
4002 PUFFER# = PUFFER# + AS
4003 IF MMIO# = 0 THEN GOTO 4007
4004 DISABE INTR
4005 IF LEN(PUFFER#) > 0 THEN GOSUB 4300
4006 ENABE INTR
4007 RETURN

```

```

4100 REM I/O-Interrupt: PUFFER# -> Drucker
4101 IF LEN(PUFFER#) > 0 THEN GOTO 4104
4102 READY = 1
4103 GOTO 4105
4104 GOSUB 4300
4105 RETURN

```

```

4200 REM Drucker-Treiber
4201 OUT 4,2
4202 RETURN

```

```

4300 REM Hilfs-UP: PUFFER# -> Drucker
4301 S = ASC(PUFFER#)
4302 PUFFER# = RIGHT$(PUFFER#, LEN(PUFFER#)-1)
4303 READY = 0
4304 GOSUB 4200
4305 RETURN

```

```

4400 REM Initialisierung
4401 READY = 1
4402 OUT 6,0 : OUT 6,0 : OUT 5,0 : OUT 6,8H40
4403 OUT 6,8H0F : OUT 6,8H27
4404 PUFFER# = ""
4405 ON INTR #6 GOSUB 4100
4406 RETURN

```

Bild 4. I/O-Interrupteinsatz

also zuerst wieder 'Variablen-Lernen'.

'Pseudo'-Praxis

Zunächst muß die Größe des Timer Zeitintervalls geeignet gewählt werden. Macht man es zu klein, so wird die Interrupt-routine sehr oft durchlaufen; das Hauptprogramm wird durch die häufigen Unterbrechungen stark verlangsamt. Beliebiger klein läßt sich das Zeitintervall in Interpreter-BASIC ohnehin nicht machen. Zwar kann man oft den Timer in Schritten von einer Millisekunde einstellen, doch nützt das wenig. Die laufende BASIC-Zeile wird ja, wie oben beschrieben, beim Auftreten eines Interrupts zuerst fertig bearbeitet, was häufig länger als eine Millisekunde dauert. Auch wird meist die Bearbeitung der Interruptroutine selbst länger als eine Millisekunde dauern; bevor sie aber nicht zum RETURN gelangt, kann sie nicht neu aufgerufen werden.

Macht man das Zeitintervall zu groß, so wartet oft der Drucker einige Zeit, bis der nächste Interrupt kommt. Er wird also verlangsamt.

Trotz dieser Schwierigkeiten beim Einstellen des 'richtigen' Zeitintervalls ist die Pseudo-Interrupt-Technik der einfachen I/O-Technik mit Pufferung überlegen. Die Zeitintervalle sind ja in jedem Fall wegen des Pufferungsprinzips nötig und auch tatsächlich vorhanden. Bei der einfachen Technik werden sie durch Programmlaufzeiten usw. geildet; dabei lassen sie sich schlecht festlegen und noch schlechter verändern. Außerdem können die einzelnen Intervalle sehr verschieden lang ausfallen. Das bedeutet aber, daß durch diese Schwankungen zwangsläufig von der optimalen Intervalllänge abgewichen wird.

Bei Verwendung von Timer-Interrupts läßt sich sehr einfach ein Zeitintervall mit einigermaßen genau bekannter Größe erzeugen. Änderungen zum Ausprobieren des optimalen Intervalls sind sehr einfach, da nur eine einzige Anweisung (Zeile 3304 in Bild 3) geändert werden muß. Außerdem erspart man sich das Einfügen der Aufrufe des Ausgabe-UP's überall im Programm. Da das nur zeitlich gesehen erfolgt, ohne daß tatsächlich überall die Aufrufe stehen, wird das Programm kürzer und übersichtlicher.

'Echte' I/O-Interrupts

Eine weitere Verbesserung ergibt sich, wenn man die Interrupts unmittelbar von einem I/O-Baustein auslösen läßt.

Der Timer-Baustein 'weiß' ja nichts von den I/O-Vorgängen. Er unterbricht das Programm, ohne Rücksicht, präzise wie eine Uhr. Deshalb kann es nicht ausbleiben, daß die Interruptroutine manchmal durchlaufen wird, ohne daß dabei ein Zeichen ausgegeben wird; dann ist entweder der Puffer leer, oder der Drucker ist nicht 'ready'.

Die Interruptroutine wird also pro auszugebendes Zeichen im Schnitt mehr als einmal durchlaufen. Wenn man nun erreichen könnte, daß sie dafür genau einmal durchlaufen würde, so käme dabei unter dem Strich mehr verfügbare Zeit für das Hauptprogramm heraus. Diese Möglichkeit bietet sich, wenn ein Interrupt direkt von einem I/O-Baustein ausgelöst wird. Diese Auslösung erfolgt immer dann, wenn der I/O-Baustein die Notwendigkeit dazu erkennt. Da er ja mit dem entsprechenden I/O-Gerät (hier mit dem Drucker) verbunden ist, kann er es ohne CPU-Hilfe erkennen.

Ein geeigneter Zeitpunkt für die Auslösung ist der Moment, wenn der Drucker seinen Status von 'nicht ready' auf 'ready' verändert. In der Interruptroutine kann man dann sofort das nächste Zeichen ausgeben. Eine Abfrage auf 'ready' entfällt; der Interrupt wäre bei Status 'nicht ready' ja nicht erfolgt.

Voraussetzung ist hierbei, daß der Status nur dann von 'ready' auf 'nicht ready' wechselt, nachdem ein Zeichen ausgegeben wurde, und daß dieser Wechsel auch stattfindet. Der nächste Interrupt kann erst wieder ausgelöst werden, wenn ein Wechsel von 'nicht ready' nach 'ready' auftritt.

Man sieht, daß die Verwendung von I/O-Interrupts genau definierte Anforderungen an die Hardware stellt. Für die Auslösung der Interrupts ist daher ein geeignetes Interface erforderlich; das I/O-Gerät muß mit seinen Handshakesignalen daran angepaßt sein, um den Interrupt im richtigen Moment auslösen zu können.

BASIC-Lösung

In BASIC kann ein Programm unter Verwendung von I/O-Interrupts wie in Bild 4 aussehen. Bei der Betrachtung des UP 4000 lassen wir zunächst die Zeilen 4003 bis 4005 außer acht. Das UP 4000 arbeitet dann wie das UP 3000 des vorigen Beispiels. Die Interruptroutine UP 4100 gibt bei jedem Durchlauf ein Zeichen aus dem Puffer aus, falls er nicht leer ist. Sie bedient sich dazu des Hilfs-UP's 4300, das ein Zeichen aus dem Puffer entnimmt und es über den Druckertreiber UP 4200 ausgibt. Der Status des Druckers wird dabei nicht mehr abgefragt; der Interrupt wird ja gerade dann ausgelöst, wenn der Drucker 'ready' wird.

Zu einer kritischen Situation kommt es, wenn die Interruptroutine einen leeren Puffer vorfindet. Dies kann sicher irgendwann auftreten, wenn das Hauptprogramm längere Zeit nichts auszugeben hat. Die Interruptroutine läuft dann auf RETURN, ohne den Drucker anzusprechen. Aber: es wird kein nachfolgender Interrupt mehr kommen! Er kommt ja nur beim Wechsel des Status von 'nicht ready' auf 'ready'. Nun wechselt aber der Status nicht mehr. Er war zu Beginn der Interruptroutine 'ready', und er bleibt es weiterhin, da die Interruptroutine wegen des leeren Puffers kein Zeichen ausgibt. Das Programm wird also steckenbleiben.

Selbst wenn später im Hauptprogramm durch einen Aufruf des UP 4000 wieder Daten in den Puffer geschrieben werden,

so nützt dies nichts. Es führt ja nicht zu einer Statusänderung des Druckers und damit nicht zu einem Interrupt. Ein neuer Interrupt wird erst wieder nach der Ausgabe eines Zeichens kommen. Diese Ausgabe läßt den Drucker für kurze Zeit 'nicht ready' werden. Danach wird er wieder 'ready', und ein Interrupt wird ausgelöst. Nun erfolgt diese Ausgabe aber in der Interruptroutine selbst, so daß sich hier 'die Katze in den Schwanz beißt'.

In unserem Fall ist der Ausweg aus diesem Dilemma die Einführung der Variablen READY. Sie soll anzeigen, ob der Drucker gerade ein Zeichen bearbeitet (READY=0), so daß ein Interrupt erwartet werden kann, oder ob der Drucker nichts zu tun hat (READY=1). Die Interruptroutine setzt READY=1, falls sie kein Zeichen ausgibt. Bei der Ausgabe eines Zeichens wird in jedem Fall READY=0 gesetzt, zusammen mit dem Aufruf des Drucker-Treibers. Falls nun im UP4000 beim Einschreiben von Daten in den Puffer READY=1 erkannt wird, so ist kein Interrupt zu erwarten. Deshalb gibt das UP4000 in diesem Fall ausnahmsweise selbst ein Zeichen aus dem Puffer an den Drucker aus (Zeilen 4003 bis 4006).

Es bedient sich dazu des Hilfs-UP's 4300, das auch in der Interruptroutine zur Ausgabe verwendet wird. Dadurch ist sichergestellt, daß die Ausgabe nicht in zwei unterschiedlichen Programmstücken erfolgt, die voneinander nichts wissen, und die spätestens nach der ersten Änderung nicht mehr widerspruchsfrei laufen. Nach der Ausgabe dieses einen Zeichens durch das Hauptprogramm wird wieder ein Interrupt erfolgen; die Ausgabe per Interruptroutine 'kommt wieder in Fahrt'.

Der Aufruf des UP 4300 findet sich sowohl im Hauptprogramm (Zeile 4005) als auch in der Interruptroutine (Zeile 4104) wieder. Hier lauert schon die nächste Fehlermöglichkeit. Tritt nämlich ein Interrupt auf, während das Hauptprogramm gerade das UP 4300 aufgerufen hat, so wird das UP 4300 von der Interruptroutine erneut aufgerufen, bevor es im Hauptprogramm zum Ende gekommen ist. So etwas wird als

'Rekursiv-Aufruf' bezeichnet. Je nach Programmiersprache und Codierung ist das unzulässig. Auf jeden Fall ist es sehr schwierig zu überblicken, was in einem solchen Fall genau geschieht. Deshalb wurden in Bild 4 die Zeilen 4004 und 4006 eingefügt. Mit DISABLE INTR wird in der CPU eine Interruptsperre eingeschaltet, die mit ENABLE INTR wieder aufgehoben wird. Interface und Drucker wissen von der Sperre nichts, so daß auch innerhalb der Sperrzeit eine Interruptanforderung an die CPU gestellt werden kann. Die CPU gibt dieser Anforderung aber nicht statt; sie merkt sie sich jedoch und führt den Sprung zur Interruptroutine später aus: nach Aufhebung der Sperre durch ENABLE INTR. Auf diese Weise geht also kein Interrupt 'verloren'.

Das oben geschilderte Problem fehlender Interrupts bei leerem Puffer tritt auch beim Programmbeginn auf. Der Puffer ist ja zunächst leer, der Drucker 'ready'. Das Initialisierungs-UP 4400 setzt READY=1, damit löst sich das Problem ganz zwanglos von selbst.

Die Zeile 4405 des Initialisierungs-UP's entspricht der Zeile 3304 des vorigen Beispiels. Es soll ein Interrupt ausgelöst werden von dem Interface mit der Adresse 06, an das der Drucker angeschlossen ist. Dabei ist 4100 die Adresse der Interruptroutine.

Die 'Echten' in der Praxis

Wie schon aus der Beschreibung des Verfahrens zu erkennen war, ist hier noch mit erheblich mehr Fallstricken zu rechnen, als bei dem vorangehenden Verfahren. Die bei Timer-Interrupts aufgezeigten Fehlerquellen sind hier genauso vorhanden, dazu kommen aber noch weitere. Ein Timer läuft, einmal richtig initialisiert, wie eine Uhr; solange die Interruptroutine korrekt mit RETURN abgeschlossen ist, wird sie auch immer wieder durchlaufen werden. Selbst wenn einmal ein einzelner Interrupt ausfallen sollte, oder ein überzähliger kommt, so ist das nicht tragisch.

Die Auslösung eines I/O-Interrupts erfolgt dagegen im Zu-

sammenspiel von Gerät, Interface und Programm. Es kann erhebliche Mühe machen, sicherzustellen, daß die Interrupts stets kommen, wenn sie erwartet werden, und niemals kommen, wenn sie nicht erwartet werden. Das Verfahren steht und fällt mit der eben genannten Voraussetzung. Programmtechnische Sicherheitsmaßnahmen gegen fehlende oder überzählige Interrupts sind zwar grundsätzlich möglich, aber schwierig und ihrerseits sehr fehleranfällig.

Grundsätzlich muß man auch gerade bei der Verwendung von I/O-Interrupts mit 'hineinkonstruierten' Fehlern der Hardware, des Betriebssystems und gegebenenfalls der höheren Programmiersprache rechnen. Selbst die CPU's und I/O-Bausteine sind davon nicht ausgenommen. Bei externen Geräten (in unserem Beispiel beim Drucker) muß man auch mit gewissen Ereignissen rechnen, die zu beliebigen, vom Rechner aus unvorhersehbaren Zeitpunkten eintreten können: Ein/Aus-Schalten, Online/Offline-Umschalten, Papierende und ähnliches. Es gibt genügend Möglichkeiten, durch solche Ereignisse den Interruptmechanismus (Hardware und Software) zu stören.

Eine weitere Gruppe von Fehlern entsteht dadurch, daß das Hauptprogramm an Stellen unterbrochen wird, wo dies nicht zulässig ist. Dabei kann es sich um die oben erwähnten Rekursivaufrufe handeln. Es gibt aber auch andere Befehlsfolgen, die nicht unterbrochen werden dürfen. Grundsätzlich sind alle Anweisungsfolgen, die mit der Pufferverwaltung oder

mit dem Druckerstatus zu tun haben, als gefährdet anzusehen. Allein schon das Vertauschen von zwei Anweisungen, deren Reihenfolge im Hauptprogramm eigentlich gleichgültig ist, kann im Hinblick auf mögliche Unterbrechungen unzulässig sein. Selbst wenn es im ganzen Programmlauf nur eine einzige Millisekunde gibt, in der ein Interrupt stören würde: er findet sie mit Sicherheit!

Die hier gebrachten BASIC-Programme zeigen die Fehlermöglichkeiten nicht so deutlich, sie sehen sehr simpel aus. Die Schwierigkeiten beginnen aber spätestens dann, wenn einige sinnvolle Erweiterungen (zum Beispiel der Einbau einer Formatsteuerung) oder Änderungen (beispielsweise bei der Pufferverwaltung, um von der zeitaufwendigen Stringverarbeitung loszukommen) vorgenommen werden.

Centronics-Treiber für Z80

Nach so viel Theorie und 'Baugemacherei' nun aber etwas Praktisches:

Ein Treiberprogramm in Z80-Assemblersprache für die Ansteuerung eines als Centronics-Schnittstelle geschalteten Z80-PIO-Interfaces unter Benutzung von I/O-Interrupts. Das Listing ist in Bild 5 abgedruckt, die zugehörige Schaltung in Bild 6.

Die Schaltung belegt nur einen Port, und zwar Port A. Um den DATA STROBE-Impuls zu erzeugen, wird das Datenbit 7 verwendet. Es können dadurch als Daten nur die Codes \$20 bis \$7F ausgegeben wer-

```

<0000      0001 PUFETH EQU $          ;1 ZEICHEN --> PUFFER
0002
0003 ;BEI AUFRUF: A = AUSGABEZEICHEN
0004
0005 47      0005 LD B,A
0006 0001 3A0E01 0006 PUFF1 LD A,(PUFF1) ;AUFGRUNDSZÄHLER
0007 0004 00FF 0007 CP 255 ;NULL ?
0008 0006 2B0F01 0008 JR Z,PUFF1-$ ;JA --> ERST LEEREN
0009 0009 2A0F01 0009 LD HL,(PUFF1) ;EINSCHREIBEZEIGER
0010 000B 78 0010 LD HL,A ;ZEICHEN --> PUFFER
0011 000C 23 0011 INC HL ;EINSCHREIBEZEIGER +1
0012 000D 0B 0012 EX DE,HL
0013 000E 210E01 0013 LD HL,PUFF2
0014 0011 34 0014 INC HL
0015 0012 210E01 0015 LD HL,PUFF2+1
0016 0013 07 0016 OR A
0017 0016 0D02 0017 SEC ;PUFFERENDE ÜBERSCHRITTEN ?
0018 0018 2003 0018 JR NZ,PUFF12-$ ;NEIN
0019 001A 110F00 0019 LD DE,PUFF1 ;JA --> ZURÜCK ZUM ANFANG
0020 001D 0D030F01 0020 PUFF12 LD DE,(PUFF12) ;NEUER EINSCHREIBEZEIGER
0021 0021 3A0301 0021 LD A,(READY)
0022 0024 07 0022 OR A
0023 0025 F3 0023 DI ;INTERRUPT ERWARTET ?
0026 0026 420E01 0026 CALL NZ,PUFF15 ;FALLS NEIN --> ANFOLGEN

```

Fortsetzung Bild 5 nächste Seite


```

'0629 FB      0025      EI
'0629 C9      0026      RET
'0629          0027
'0629          0028 PUFHUC EQU $          ;PUFFER --> DRUCKER
'0629          0029
'0629          0030 ;BEI RUECKSPRUNG: FLAG NONZERO, FALLS PUFFER NICHT LEER WAR
'0629          0031 ;
'0629          0032
'0629          0033 LD      A,PUFBZ2          ;BELEGUNGSZAEHLER
'0629 B7      0034 OR      A          ;LEER ?
'0629          0035 JR      Z,PUFAU2-$          ;JA
'0629          0036 DEC      A
'0629          0037 LD      A,PUFBZ2          ;INCH > 1
'0629          0038 LD      HL,(PUFAU2)          ;AUSLESEZEIGER
'0629          0039 LD      A,HL          ;ALF. AUSGABEZEICHEN
'0629          0040 CALL    DTREIB          ;DRUCKER-TREIBER AUFRUFEN
'0629          0041 JR      NC          ;AUSLESEZEIGER +1
'0629          0042 EX      DE,HL
'0629          0043 LD      HL,PUFEND+1
'0629          0044 OR      A
'0629          0045 HL,DE          ;PUFFERENDE UEBERSCHRITTEN ?
'0629          0046 JR      NZ,PUFAU1-1          ;NEIN
'0629          0047 LE      DE,PUFAU2          ;JA --> ZURUECK ZUM ANFANG
'0629          0048 LD      HL,(PUFAU2),DE          ;NEUER AUSLESEZEIGER
'0629          0049 XCH      A
'0629          0050 LD      A,(READY),A          ;READY=0 NACH AUSGABE
'0629          0051 JNC      A          ;--> FLAG NONZERO
'0629          0052 PUFHUC RET
'0629          0053 INTRPT EQU $          ;INTERRUPT-KONTROLLE
'0629          0054 PUSH    AF          ;ALLE VERWENDETEN REGIST-
'0629          0055 PUSH    BC          ;TER SETZEN
'0629          0056 PUSH    DE
'0629          0057 PUSH    HL
'0629          0058 LD      A,(READY),A
'0629          0059 OR      A
'0629          0060 JR      NZ,INTRPT-1          ;JEDERZEIT NIE KOMMEN
'0629          0061 CALL    PUFHUC          ;AUFRUF AUS PUFFER
'0629          0062 JR      NZ,INTRPT-1          ;PUFFER WAR NICHT LEER !
'0629          0063 LD      A,1
'0629          0064 LD      A,(READY),A          ;PUFFER LEER --> READY=1
'0629          0065 HL,INTRPT          ;REGISTER WIEDERHERSTELLEN
'0629          0066 POP      DE
'0629          0067 POP      BC
'0629          0068 POP      AF
'0629          0069 RET
'0629          0070
'0629          0071
'0629          0072
'0629          0073
'0629          0074 DTREIB EQU $          ;DRUCKER-TREIBER
'0629          0075
'0629          0076 ;BEI AUFRUF: A = AUSGABEZEICHEN
'0629          0077
'0629          0078 CPL          ;WEIL 7404 INVERTIERT
'0629          0079 SET      7,A          ;DATEN + STROBE HIGH
'0629          0080 OUT      (12H),A          ;
'0629          0081 RES      7,A          ;DATEN + STROBE LOW
'0629          0082 OUT      (12H),A          ;
'0629          0083 SET      7,A          ;DATEN + STROBE HIGH
'0629          0084 OUT      (12H),A
'0629          0085 RET
'0629          0086
'0629          0087
'0629          0088
'0629          0089 INIT EQU $          ;INITIALISIERUNG
'0629          0090 DI
'0629          0091 XCH      A
'0629          0092 LD      A,PUFBZ2          ;PUFFER - LEER
'0629          0093 INC      A
'0629          0094 LD      A,(READY),A          ;READY=1
'0629          0095 LD      HL,PUFAU2          ;PUFFERZEIGER INITIALISIEREN
'0629          0096 LD      HL,(PUFAU2),HL
'0629          0097 LD      HL,PUFBZ2          ;
'0629          0098 LD      HL,2
'0629          0099 LD      HL,INTVEK+1          ;ADRESSE DES INTERRUPTVEKTORS
'0629          0100 RES      0,HL          ;MUSS GERADE SEIN !!!
'0629          0101 LD      A,L
'0629          0102 OUT      (14H),A          ;LOW BYTE --> PIO
'0629          0103 LD      A,H
'0629          0104 LD      A,I
'0629          0105 LD      DE,INTRPT          ;ADRESSE DER INTERRUPT-ROUTINE
'0629          0106 LD      HL,INTVEK          ;EINTRAGEN IN IN-
'0629          0107 INC      HL          ;TER-
'0629          0108 LD      HL,(HL),0          ;INTERRUPTVEKTOR
'0629          0109 LD      A,10000000B          ;STROBE HIGH
'0629          0110 OUT      (12H),A          ;
'0629          0111 LD      A,00001111B          ;PIO-WIDE=0 (OUTPUT)
'0629          0112 OUT      (14H),A          ;
'0629          0113 LD      A,10001111B          ;ENABLE PIO-INTERRUPT
'0629          0114 OUT      (14H),A
'0629          0115 EI
'0629          0116 RET
'0629          0117
'0629          0118 ;ARBEITSPEICHER:
'0629          0119
'0629          0120 PUFHUC DEFS 255          ;RINGPuffer
'0629          0121 PUFHUC EQU PUFHUC+254          ;INDRESSE DES PUFFERS
'0629          0122
'0629          0123 PUFHUC DEFS 1          ;BELEGUNGSZAEHLER
'0629          0124 PUFHUC DEFS 2          ;EINSCHREIBEZEIGER
'0629          0125 PUFHUC DEFS 2          ;AUSLESEZEIGER
'0629          0126
'0629          0127 READY DEFS 1          ;MERKER FUER DRUCKER-STATUS
'0629          0128 INTVEK DEFS 3          ;INTERRUPTVEKTOR AUF GRA-
'0629          0129 ;
'0629          0130 END

```

Bild 5. Interruptgesteuerter Centronics-Treiber

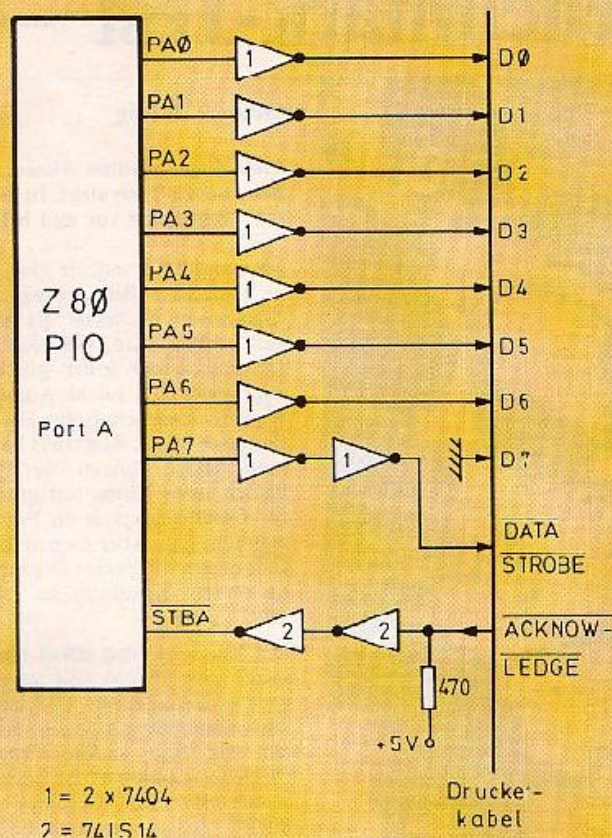


Bild 6. Centronics-Schnittstelle mit Z80-PIO

den. Die PIO wird per Software so eingestellt, daß der ACKNOWLEDGE-Impuls des Druckers den Interrupt auslöst. Die BUSY-Leitung ist bei diesem Verfahren überflüssig.

Das Assemblerprogramm in Bild 5 besteht aus 5 UP's. Es wird auch hier der schon in der ersten Folge des Beitrags beschriebene Ringpuffer verwendet. PUFHUC dient im Hauptprogramm zur Ausgabe eines Zeichens in den Ringpuffer, PUFHUC ist ein Hilfs-UP zur Entnahme eines Zeichens aus dem Puffer und dessen Ausgabe an den Drucker-Treiber DTREIB. Letzterer erzeugt den DATA STROBE-Impuls mit einer Folge von 3 Ausgabebefehlen. Die Interruptroutine INTRPT wird durch den ACKNOWLEDGE-Impuls des Druckers ausgelöst. Das UP INIT erledigt die hier sehr vielfältigen Aufgaben beim Programmstart. Dabei schaltet es

auf Interrupt-Mode 2, belegt einen Platz für den Interruptvektor im Bereich dieses Programms und lädt das I-Register entsprechend. Dies kann nur funktionieren, wenn (mit Ausnahme von NMI) keine anderen Interrupts im Rechner auftreten. Falls ein anderer Programmteil oder das Betriebssystem mit Interrupts arbeitet, so entsteht durch das UP INIT ein Fehler: entweder beim Umschalten des Interrupt-Modus oder beim Umladen des I-Registers. Wird an mehreren Stellen im Programm (Betriebssystem eingeschlossen) mit Interrupts gearbeitet, so müssen diese Aktionen koordiniert werden. Es muß überall der gleiche Interrupt-Mode verwendet werden; bei Mode 2 müssen alle Interruptvektoren in einer gemeinsamen 'Vektortabelle' liegen, auf die dann das I-Register in Verbindung mit den Lower-Byte-Interruptvektoren der Portbausteine zeigt.

Benchmark-Test — eine Computer-Meß-

Siegmar Wittig

Viele Leser wollten wissen, welche 'Philosophie' hinter den c't-Benchmark-Tests steht. In diesem Bericht stellen wir die verwendeten Programme vor und beleuchten die Hintergründe.

Niemand käme auf die Idee, eine pauschale Behauptung wie 'Beethoven ist besser als Mozart' aufzustellen, weil sich erfahrungsgemäß (oder glücklicherweise) eine solche Aussage jeglicher meßtechnischer Bestätigung entzieht. Allenfalls hätte man messen können, wer von beiden seine Noten mit größerer Geschwindigkeit zu Papier gebracht hat. Aber niemandem fällt es ein, die beiden Giganten an diesem Kriterium zu messen!

Der Vergleich mag etwas überspitzt sein, aber er soll zeigen, welche Aussagekraft Maßdaten manchmal haben können. Stellen wir eine andere Frage: 'Welcher Computer ist besser, Apple II oder Commodore 8032?' Jeder, der sich einen Computer anschaffen will, stellt sich diese oder eine ähnliche Frage. Wen wundert's, daß man nur zu gern geneigt ist, die 'Leistung' (was immer das auch heißen mag) von Computern in Zahlen (wenn möglich mit etlichen Stellen hinter dem Komma) zu 'messen'.

Der Wunsch(träum) ist so alt wie die Computer selbst. Einige der folgenden Bewertungsverfahren spielen in der Praxis eine gewisse Rolle. Werfen wir einen Blick darauf.

Was ist ein Mix?

Ein 'Mix' ist ein Bewertungsverfahren, bei dem man die Ausführungszeit für eine typische 'Mischung' (daher Mix) von Einzeloperationen bestimmt. Solche Mixes werden für bestimmte Problembereiche (zum Beispiel kommerzielle Anwendungen) eigens zusammengestellt. Hierbei erhalten die einzelnen Operationen unterschiedliche Gewichte, je nach Anwendungsgebiet, für das der Computer (genauer: die Zentraleinheit) vorgesehen ist. 'Mix I' ist zum Beispiel der Name eines solchen Mixes für kommerzielle Anwendungen. Vergleichsbefehle haben hier

ein relativ hohes Gewicht, Rechenoperationen spielen für die Bestimmung der 'Kennzahl' nur eine untergeordnete Rolle.

Der 'GIRSON-Mix' dagegen soll ein Maß für die Geschwindigkeit bei der Ausführung von mathematischen und technischen Problemstellungen liefern. Dieser Mix ist nicht einheitlich definiert, aber man kann soviel sagen, daß Rechenoperationen hier natürlich ein wesentlich höheres Gewicht haben als andere Operationen.

Für mathematisch-technische Anwendungsprogramme war auch der 'GAMM-Mix' gedacht: (GAMM ist die Abkürzung für 'Gesellschaft für Angewandte Mathematik und Mechanik'). Dieser Mix besteht aus einer Reihe typischer mathematischer Operationen. Hier zwei Beispiele: Aus 100 Zahlen soll die Zahl mit dem größten Betrag gesucht werden. Oder: Berechnung einer Quadratwurzel. Solche und andere mathematische Aufgaben hat der Computer zu lösen. Die Zeiten hierfür werden gewichtet und ergeben zusammen die gesuchte Mixkennzahl.

Gemischte Gefühle

Mixes der eben genannten Art sind und waren immer sehr umstritten. Denn beispielsweise wird die Rechengenauigkeit, eine Eigenschaft des Rechners, die die Ausführungszeiten wesentlich mitbestimmt, nicht berücksichtigt. Dürfte es doch auf der Hand liegen, daß zwei ansonsten gleiche Computer bei einer Rechengenauigkeit von sechs beziehungsweise zehn signifikanten Stellen unterschiedliche Ausführungszeiten für mathematische Operationen haben müssen!

Benchmark-Tests

Kaum ein Computer-Testheft ohne Benchmark-Test! Benchmark-Tests sind, wie die

Mixes, ein Versuch, die Leistungsfähigkeit einer Datenverarbeitungsanlage zu bewerten (wir vermeiden das Wort 'messen'). Ein Benchmark-Test wird mit einer Reihe ausgesuch-

latte?



Foto: Hanns Krebs (Bavaria)

ter Programme durchgeführt, die für den Benutzer typische Aufgaben lösen sollen. Ein Benchmark-Test ist also immer eine individuelle Angelegenheit. Der Benutzer sucht mit ei-

c't 1984, Heft 4

```
100 REM BM1
300 PRINT "S"
400 FOR K=1 TO 1000
500 NEXT K
700 PRINT"E"
800 END
```

Programm 1. Zählschleife. Die Variable K nimmt der Reihe nach die Werte von 1 bis 1000 an.

```
100 REM BM2
300 PRINT"S"
400 K=0
500 K=K+1
600 IF K<1000 THEN 500
700 PRINT"E"
800 END
```

Programm 2. Zählschleife wie in Programm 1, jedoch realisiert mit einer IF-THEN-Anweisung.

```
100 REM BM3
300 PRINT"S"
400 K=0
500 K=K+1
510 A=K/K*K+K-5
600 IF K<1000 THEN 500
700 PRINT"E"
800 END
```

Programm 3. Der Zählschleife aus Programm 2 wurden arithmetische Operationen mit der Variablen K zugefügt.

```
100 REM BM4
300 PRINT"S"
400 K=0
500 K=K+1
510 A=K/2*3+4-5
600 IF K<1000 THEN 500
700 PRINT"E"
```

Programm 4. Das gleiche Programm wie Nr. 3, jedoch werden hier die arithmetischen Operationen mit Konstanten ausgeführt. Wegen der dazu notwendigen Zahlenkonversionen wird Programm 4 langsamer als Programm 3 sein.

```
100 REM BM5
300 PRINT"S"
400 K=0
500 K=K+1
510 A=K/2*3+4-5
520 GOSUB 820
600 IF K<1000 THEN 500
700 PRINT"E"
800 END
```

nem solchen Test nach dem Computer, der seine Programme am schnellsten ausführt. Als Tom Rugg und Phil Feldman 1977 in der US-amerikanischen Zeitschrift Kilobaud zwei

820 RETURN

Programm 5. Wie Programm 4, jedoch mit zusätzlichen Sprüngen in ein 'leeres' Unterprogramm.

```
100 REM BM6
300 PRINT"S"
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"
800 END
820 RETURN
```

Programm 6. Wie Programm 5, jedoch zusätzlich mit der dimensionierten Variablen M(.) und einer kurzen geschachtelten FOR-NEXT-Schleife (Zeilen 530 bis 540).

```
100 REM BM7
300 PRINT"S"
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"
800 END
820 RETURN
```

Programm 7. Wie Programm 6, jedoch zusätzlich mit 5000 Zuweisungen zu einer Bereichsvariablen (Zeile 535).

```
100 REM BM8
300 PRINT"S"
400 K=0
500 K=K+1
530 A=K/2
540 B=LOG(K)
550 C=SIN(K)
600 IF K<1000 THEN 500
700 PRINT"E"
800 END
```

Programm 8. Dieses Programm soll die Ausführungszeit für transzendente Funktionen messen. Hier zeigen BASIC-Interpreter oft erhebliche Zeitunterschiede.

Artikel über den Vergleich der Ausführungszeiten von Mikrocomputern schrieben (Heft 6, Seite 66 bis 70, und Heft 10, Seite 20 bis 25), ahnten sie nicht, welche Konsequenzen ih-

re dort veröffentlichten 'Benchmarks' haben sollten. Ihre sieben kurzen Benchmark-Programme (siehe Kästen) sind nachgerade zu einem Standard für Mikrocomputer-Tests geworden. Dabei ist hier die Bezeichnung 'Benchmark' äußerst unglücklich, wenn nicht sogar unpassend gewählt. Bei genauem Hinsehen handelt es sich nämlich eher um Teilprogramme einer Art von Mix, denn in den kurzen Programmen werden spezielle BASIC-Anweisungen in Kombination von Rechenoperationen getestet, aber nicht problemorientierte, benutzerspezifische Anwendungen. Aber die Bezeichnung 'Benchmark' hat sich nun mal durchgesetzt, was uns einmal mehr resignierend die vielbeschworene 'normative Kraft des Faktischen' zu zitieren Anlaß gibt.

Fast alle Computertests in c't enthielten eine Messung der Ausführungszeiten der sieben 'Benchmark'-Programme. Hierzu kam noch ein achttes Programm, das von John A. Coll in der Zeitschrift Computer Education, Heft 28, Seite 18-21, veröffentlicht wurde (siehe Kästen). Auf Wunsch zahlreicher Leser drucken wir diese acht Programme hier noch einmal ab. In den zurückliegenden Heften von c't können Sie anhand der dort veröffentlichten Ausführungszeiten der acht Programme ihre eigene 'Maschine' einordnen.

Damit wir uns richtig verstehen: Auch die c't-Tester wissen, daß Mixes und Benchmark-Tests einen beschränkten Wert für die Beurteilung eines Computers haben. Wir sagten bereits, daß bei solchen Vergleichen überhaupt nicht berücksichtigt wird, mit welcher Genauigkeit der Computer Dezimalzahlen verarbeitet. Und über die Leistungsfähigkeit der Peripheriegeräte wird schon gar nichts ausgesagt, von Benutzerfreundlichkeit, Zuverlässigkeit und Qualität der Dokumentation ganz zu schweigen. Zum Messen von Ausführungszeiten braucht man halt nur eine Stoppuhr (oder nicht einmal diese, wenn der Computer eine eingebaute Uhr hat), die arden Merkmale aber sind nur mit Hilfe langer Checklisten zu bestimmen — oder gar nicht. Das ist das Geheimnis der unverdienten Beliebtheit von Mixes und Benchmarks! □

FORTH

anpassen

Teil 2: Kommunikation mit FORTH

Peter Glasmacher

Der erste Teil dieser Serie beschäftigte sich mit den Basisvereinbarungen, die nötig sind, um FORTH auf einem Tischcomputer lauffähig zu bekommen. Am Beispiel des APPLE II wurde aufgezeigt, welche Überlegungen vor Eingabe des Quellcodes notwendig sind, um allzu aufwendige Entlaufsprozeduren zu vermeiden. Im vorliegenden Teil 2 soll FORTH nun die Fähigkeit bekommen, über Ein-/Ausgabeneinheiten mit dem jeweiligen Herrn und Meister zu kommunizieren.

FORTH gehört zu den interaktiven Sprachen, deshalb sollte der Computer mit einer alphanumerischen Tastatur und einem Videodisplay ausgerüstet sein, so wie es zum Beispiel beim Apple der Fall ist. Des weiteren sollte das Betriebssystem des Rechners in der Lage sein, ein Zeichen von der Tastatur zu übernehmen und auf dem Bildschirm auszugeben. Diese beiden Grundfunktionen, Terminalein- und -ausgabe werden von FORTH mit zwei Worten, **KEY** und **EMIT** abgewickelt. **EMIT** sendet das oberste Byte auf dem FORTH-Stack zu einem Terminal und **KEY** übernimmt ein Zeichen von der Tastatur und legt es auf den Stack. Zwei weitere Worte unterstützen die Basisfunktionen:

CR sendet einen Wagenrücklauf zur Ausgabeneinheit und **?TERMINAL** fragt den Eingabekanal ab und legt ein logisches 'Wahr' oder 'Falsch' auf den Stack, abhängig davon, ob zum Zeitpunkt der Abfrage eine Taste gedrückt war oder nicht. Aus diesen Basisworten wird anschließend in bester FORTH-Manier ein ganzes Paket von Ein-/Ausgabefunktionen gebildet, eine Zusammenfassung zeigt Tabelle 1.

Nun ist es durchaus möglich, mit dem im FIG-Listing abge-

druckten **KEY** und **EMIT** Code-Fragmenten (Bild 1) die gesamte Ausgabe zu bestreiten. Mehr als Basisoperationen sind jedoch nicht möglich. Es wäre wünschenswert, vom Programm die verschiedenen Ausgabegeräte wie Drucker und 80-Zeichenkarte zu berücksichtigen und außerdem der Tatsache Rechnung zu tragen, daß dem APPLE II einige Tasten fehlen. So ist zum Beispiel die eckige Klammer (Ä) nicht auf der Tastatur zu finden.

Wenn man der Familie jedoch schon die Zeit für ein Projekt wie eine FORTH-Implementierung abgetrotzt hat, sollte man sich die spätere Arbeit auch so

bequem wie möglich machen und von vornherein die verschiedenen 'Ausgänge' mit berücksichtigen.

Programmierter Komfort

Gerade der APPLE ist das typische Beispiel einer 'Featuremaschine'. Es gibt wohl kaum einen Anwender, der seinen APPLE ohne irgendwelche Zusatzkarten betreibt. In der Regel sind zumindest eine 80-Zeichen-Karte und eine Druckerschnittstelle in einen der 8 Erweiterungsstecker (SLOT) gesteckt — womit die

Probleme beginnen. Das liegt nicht an der nackten Existenz, sondern an der Tatsache, daß jede Karte auf eine andere Weise initialisiert und über verschiedene Steuerzeichen bedient und kontrolliert wird. Die meisten 80-Zeichen-Karten lassen sich zum Beispiel nicht mit dem APPLE-Befehl 'HOME' löschen, sondern verlangen dafür meist ein CTRL-L. Printerkarten benötigen oft eine Initialisierungssequenz, die nur über die Eingabe von 'PR#' aufgerufen werden kann. Ein anderer Weg wäre, ein an einen Drucker auszugebendes Zeichen direkt in das entsprechende Register zu laden und das ebenfalls vorhandene Strohe-Register solange abzufragen, bis der Drucker wieder bereit ist. Der letzte Vorschlag ist für FORTH (fas:) ideal. Wird solch eine direkte Druckausgabe in **EMIT** eingebaut, kann das gesamte Druckergeschäft über nur ein Flag gesteuert werden.

Für die korrekte Ausführung von 'HOME' hat der Verfasser ein FORTH-Wort 'HOME' definiert und zusätzlich in **EMIT** eine Abfrage eingebaut, welche die gerade aktuelle Ausgabeneinheit feststellt und das korrekte 'HOME' ausführt. Zuletzt sollte dann noch die **CR**-Routine für alle Ausgabeneinheiten angepaßt werden (Bild 2), da **CR**

Tabelle 1: FORTH-TERMINAL I/O		
KEY	(c ---)	Ein Zeichen einlesen.
EMIT	(c ---)	Ein Zeichen ausgeben.
?TERMINAL	(---)	Wahr wenn Taste gedrückt.
CR	(---)	Sendet CR zum Terminal.

EXPECT	(addr n ---)	Liest n Zeichen und legt sie unter addr ab.
TYPE	(addr u ---)	Sendet die Zeichenkette in der Länge u (max 65k !) unter der Adresse addr.
WK	(n m ---)	Sendet die Zahl n rechtsbündig in einem m Zeichen grossen Feld.
WORD	(c ---)	Liest die direkt auf WORD folgenden Zeichen ein und stellt die Zeichenkette an die NÄCHSTE FREIE STELLE in die Bibliothek. c ist ein Trennzeichen.

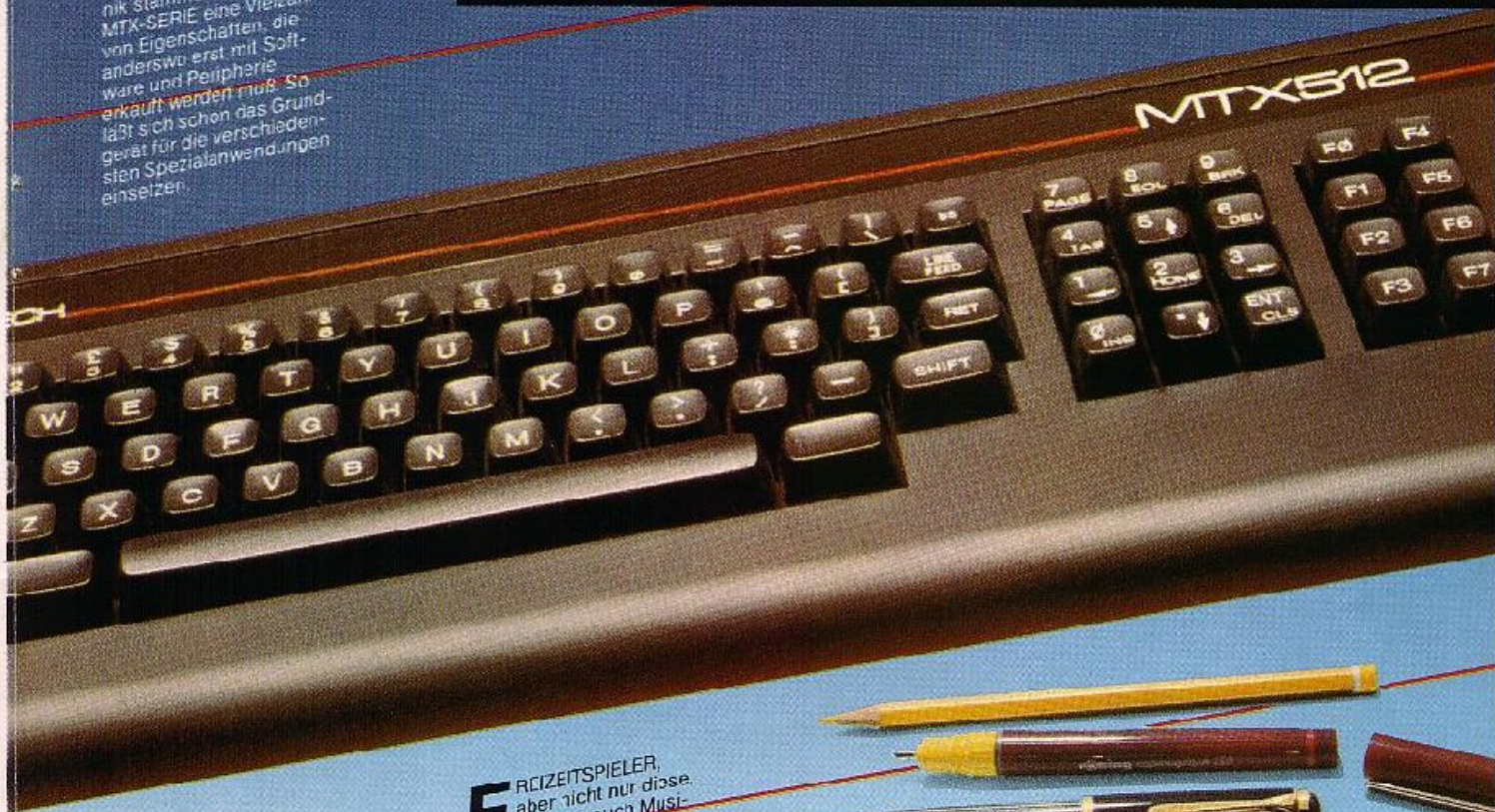
Tabelle 1. Zusammenfassung der Ein-/Ausgabefunktionen

492 *			
493 *	'KEY' STR=7, L=4		
494 *			
7511:20 0C FD	495 XKEY	JSR INCH	: ZEICHEN EINLESEN
7514:29 7F	496	AND #7F	: BIT 7 AUSBLENDEN
7516:4C DC 64	497	JMP PUSHBA	: -->STOS
498 *			
499 *	'EMIT' SCR=7, L=8		
500 *			
7519:85 00	501 XEMIT	LDA #00,X	: ZEICHEN VON STOS
751B:07 00	502	ORA #00	: BIT 7 SETZEN
751D:20 ED FD	503	JSR JUTCH	: ZEICHEN SENDEN
7520:4C EE 62	504	JMP POP	: VOM STACK ENTF.
505 *			
506 *	'CR' SCR=7, L=10		
507 *			
7523:20 BE FD	508 XCR	JSR ICR	: CR ZUM BILDSCHIRM
7526:4C 44 61	509	JMP NEXT	: UND WEITER

Bild 1. Die Basisroutinen XKEY, XEMIT und XCR. Außer Tastatur und Bildschirm wird nichts unterstützt.

MTX-512: DER PROFI ZUM HOBBYPREIS

Schon wieder ein Neuer mögen Sie klagen und am liebsten umblättern. Aber an der MTX-SERIE kommen Sie nicht ungestraft vorbei. Technologisch von der Mittlerin Datentechnik stammend, bietet die MTX-SERIE eine Vielzahl von Eigenschaften, die anderswo erst mit Software und Peripherie erkauft werden muß. So läßt sich schon das Grundgerät für die verschiedensten Spezialanwendungen einsetzen.



KAUFLEUTE werden die große Tastatur, den deutschen Zeichensatz, das numerische Tastenfeld, den Monitorausgang und den Centronics-Druckerausgang ebenso zu schätzen wissen, wie die Option, das System mit Diskettenstation unter CP/M zu fahren (wenn dann die maximale Speicherkapazität von 32 Megabyte Pseudofloppy und 20 Megabyte Winchester nicht gerügt, bitte, dann müssen wir passen).

INGENIEUREN und Technikern dagegen imponiert da schon eher die eingebaute Echtzeituhr, der parallele I/O, der zugängliche Systembus, die umfassende technische Dokumentation sowie der im ROM implementierte ASSEMBLER. DISASSEMBLER (die Option mit 512 K RAM interessiert da schon weniger).

FÜR ZEITSPIELER, aber nicht nur diese, sondern auch Musiker und die eher auf Grafik erpichten Hobbyisten werden entzückt sein über die LOGO-gesteuerte Grafik, 32 Sprites mit 7 Parametern und dem 3-Kanal-HiFi-Tonausgang unter Softwaresteuerung (16 Farben und 2 Joystick Ports helfen natürlich auch weiter).

PROGRAMMIERER – professionell oder „just for fun“ – werden die Gesamtheit aller Eigenschaften zu würdigen wissen, denn sie sind meist die Leidtragenden einer mangelhaft durchkonzipierten Maschine. Wer mit dem Befehl PANEL sofort alle Registerzustände, den Inhalt des RAMs und die Memomix des arbeitenden Programms erfährt, weiß was er hat. Wer in einer BASIC-Zelle ASS.NNN eingibt, um anschließend eine ASSEMBLER-Routine zu schreiben, um dann in der Zeile NNN+1 in BASIC fortzufahren, ist schon ein Besserwisser (daß Sie dank des getrennten 16K Bildschirm-Speichers und dem selekt. 24K ROM volle 64K als Arbeitsspeicher zur Verfügung haben, soll hier nicht weiter stören).

ANFÄNGER, aber auch terringestreifte Programmierer können Memotech für das im ROM vorhandene NODDY danken. Mit nur sieben Befehlen lassen sich dialogorientierte Menüs, aber z.B. auch Bildschirmmasken ohne PRINT-Anweisung erstellen – kinderleicht (weshalb NODDY in England für den Anfangsunterricht eingesetzt werden soll).

ALLF sollten wissen, daß die Memotech MTX-SERIE in 2 Grundversionen erhältlich ist (MTX-500/32K RAM zu DM 1198,-; MTX 512/64K 3AM zu DM 1350,-), von Profisoft, dem „Software-Haus mit Service“ vertrieben wird und Sie gegen Zusendung einer Postkarte weitere technische Unterlagen erhalten können. Selbstverständlich haben wir auch für den Fachhandel Unterlagen – mit anderem Schwerpunkt!



Memotech MTX 512 mit Diskettenstation

MEMOTECH
MTX
SERIES

profisoft

Sutthausen Str. 50-52 · 4500 Osnabrück · Tel. 05 41/5 39 05


```

ES61:63 E5      648 HOME    DW ++2      ; CODE
ES63:AC CF D7   649 LDY SPNT  ; PINTER AUF SPFLAG
ES66:B1 E3      650 LDA (UP),Y ; SPFLAG LADEN
ES68:D8 06      651 BNE CR0   ; CR0=00-ZEICHEN KARTE
ES6A:28 58 FC   652 JSR $FC58  ; CALL MONITOR HOME
ES6D:4C 69 D8   653 JMP NEXT  ; END WEITER
ES70:A7 0C      654 CR0    LDA $00C   ; VIDEOTERM HOME
ES72:CA         655 DEX      ;
ES73:CA         656 DEX      ; PLATZ AUF STACK
ES74:94 0.      657 STY 1,X   ; LUECKENFUELLER
ES76:95 00      658 STA 0,Y   ; $0C-->TOP
ES78:D8 1E      659 BNE XEMIT ; IMMER
ES7A:         660 *
ES7A:         661 * *** 'KEY', SCR=7, L=4
ES7A:         662 *
ES7A:         663 XKEY   JSR INCH   ; ZEICHEN EINLESEN
ES7D:29 7F      664 AND $7F   ; BIT 7 MARKIEREN
ES7F:28 05 E5   665 JSR C$CHK ; TEST AUF SONDERZ.
ES82:4C A5 D3   666 JMP PUSH0A ; AB NACH FORTH !
ES85:         667 *
ES85:         668 * TEST AUF 'O','K','L'
ES85:         669 *
ES85:         670 C$CHECK CMP $0F   ;
ES87:D8 02      671 BNE NC1   ;
ES87:1A7 51     672 LDA $0F   ;
ES8B:C9 0C      673 NC1    CMP $0C   ;
ES8D:D8 02      674 BNE NC2   ;
ES8F:A9 5C      675 LDA $0C   ;
ES91:C7 00      676 NC2    CMP $0B   ;
ES93:D8 02      677 BNE NC3   ;
ES95:A7 5D      678 LDA $0B   ;
ES97:6A         679 NC3    RTS
ES98:         680 *
ES98:         681 * *** 'EMIT', SCR=7, L=4
ES98:         682 *
ES98:         683 XEMIT  STX XSAVE  ; PINTER RETTEN
ES9A:05 00      684 LDA 0,X   ; ZEICHEN HOLLEN
ES9C:2C 3A 0E   685 XEMIT1 BPL PFLAG ; AUSGABE AUF DRUCKER ?
ES9E:1A 07      686 BPL VOUT  ; NEIN, NUR VIDEO
ES9F:1A 07      687 JSR POUT  ; ZEICHEN --> DRUCKER
ES9F:40 0A      688 DRA VOUT  ; BIT 7-ON
ES9F:28 ED FD   689 JSR D$CHK ; UEBERGABE MONITOR
ES9F:A4 F5      690 LDY XSAVE  ; PINTER LADEN
ES9B:4D 7B D1   691 JMP POP   ; ZURUECK
ES9F:         692 *
ES9F:         693 * *** 'OCR', SCR=7, L=8
ES9F:         694 *
ES9F:         695 XCR    BIT PFLAG ; DRUCKER ?
ES9F:1A 05      696 BPL VCR   ; VIDEO
ES9F:49 00      697 LDA $0D   ; CR
ES9F:2A C0 E5   698 JSR WLOOP ; AUSGABE DRUCKER
ES9F:28 8E FD   699 VCR   JSR TCR   ; UEBERGABE MONITOR
ES9F:4C 69 D8   700 JMP NEXT  ; ZURUECK

```

Bild 2. Die geänderten Routinen reagieren nun auf eine 80-Zeichen-Karte und die etwas abweichende Apple-Tastatur. Ein Drucker wird ebenfalls unterstützt.

ebenfalls in der FIG-Originalversion nur auf den Bildschirm wirkt.

Der Vollständigkeit halber soll auch noch eine andere Art der flexiblen Terminal I/O angesprochen werden. Einige FORTH-Systeme, zum Beispiel das FORTH des AIM 65, greifen auf KEY und EMIT über eine USER-Variable (UKEY und UEMIT) zu, in die zur Kalt- oder Warmstartzeit die Code Feld Adresse (CFA) der jeweiligen Standarddefinition geladen wird. In KEY bzw. EMIT ist dann nur noch die Phrase UEMIT (UKEY) & EXECUTE enthalten. Der Inhalt der Variablen kann dann bei Bedarf durch die CFA einer anwenderspezifischen Routine ersetzt werden. Diese Art der I/O-Steuerung ist natürlich unter Umständen etwas gefährlich und sollte mit Vorsicht gehandhabt werden.

Die Änderungen ... konkret

Bild 2 zeigt eine Zusammenfas-

gleich 0 ist. In diesem Fall wird nicht die Monitorroutine des Apple angesprochen, die den Bildschirm löscht und den Cursor in die Anfangsposition bringt, sondern der Wert \$0C auf den (FORTH) Stack gelegt und nach EMIT verzweigt. \$0C entspricht aber einem Drucker-FORMFEED-Kommando oder eben einem HOME für die meisten 80-Zeichen-Karten. Kennen werden bemerken, daß mit der Anweisung STY 1,X keine Null als höherwertiges Byte auf den Stack gelegt wird. In diesem Fall ist dies jedoch ohne Bedeutung, da EMIT nur das niederwertige Byte sendet und der Stack danach gelöscht wird.

Die Erweiterung von KEY ist als eigenes Unterprogramm ausgeführt, um eventuelle spätere Erweiterungen zu erleichtern. In der Routine C\$CHECK wird nun abgeprüft, ob eine der Control-Tasten gedrückt wurde, die als Ersatz für die fehlenden Zeichen 0, _ und ^ erhalten müssen. C\$CHECK wurde vom Autor als einfache Abfrageroutine ausgeführt. Dabei wurde auf mögliche Programmiertricks verzichtet, obwohl sicher einige Bytes einzusparen wären, da die unteren Nibbles der 'fehlenden' Tasten mit denen der Control-Tasten korrespondieren, und man die Zeichen demnach durch ein 'ORA #\$80' an der richtigen Stelle bilden kann.

Die beiden Ausgaberroutinen CR und EMIT wurden ebenfalls nicht verschont und präsentieren sich erheblich verändert.

CR prüft als erstes PFLAG und sendet ein \$0D zum Drucker, falls Bit 7 in PFLAG gesetzt ist. Ist PFLAG jedoch 0, wird nur die Monitorroutine TCR aufge-

rufen. Dies ist bei einer Druckerausgabe, wie sie hier verwendet wird, nötig, wie später noch klar wird.

Um die Ausgabe auch auf einen Drucker zu lenken, wurde eine Druckausgabe in FORTH eingebunden. Die 65XX Prozessorfamilie kennt keine speziellen Ein-/Ausgabeeinheiten, sondern betrachtet einen I/O-Kanal als eine normale Speicherstelle, in die Werte gespeichert und aus der Werte geladen werden. Um nun ein Zeichen zu einem Drucker zu bringen, benötigt man ein Ausgaberegister für das auszugehende Zeichen und ein Eingaberegister, in dem der aktuelle Druckerstatus, 'bereit' oder 'nicht bereit', gespeichert wird. Eine Ausgaberroutine muß nun dieses Status oder Strcberegister solange testen, bis der Drucker 'bereit' signalisiert (WLOOP). Danach kann das Zeichen in das Ausgaberegister geschrieben und an den Drucker übergeben werden.

Die in POUT angegebenen Adressen \$C090 und \$C0C1 beziehen sich auf eine der 'Standard'-Centronics-Interfacekarten für den APPLE und müssen eventuell geändert werden. Ebenso wäre es möglich, einen Einsprung in eine Ausgaberroutine für eine serielle Schnittstelle einzubinden.

Die erste Tätigkeit von POUT ist ein weiterer Test von PFLAG. Bei gesetztem Bit 6 in PFLAG wird zum Label PTAG verzweigt, in dem hier nur ein Rücksprung programmiert ist. Was auf den ersten Blick wie ein überflüssiges Verschwendung von Maschinenzeit aussieht, hat seinen Grund. Manche Drucker scheren sich nicht um den ASCII-Code und legen verschiedene, meist oft benötigte Zeichen, auf unkonventionelle Plätze. Sollte man einen dieser Drucker besitzen, kann man unter dem Label PTAB eine eigene Umwandlungstabelle samt Laderoutine ablegen und nach Durchlaufen dieser Routine seinem Drucker den richtigen Code für ein bestimmtes Zeichen anbieten.

Sind alle diese Schritte durchlaufen, springt das Programm nach EMIT zurück und gibt das Zeichen noch auf dem Bildschirm aus; die nötigen Befehle mit einer angelegten Tabelle zeigt Bild 3.

(Wird fortgesetzt)

```

ESBE:         701 *
ESBE:         702 * *** PRINT UNTERPROGRAMM
ESBE:         703 *
ESBE:         704 POUT   BVS PTAB   ; USER DEFINED CGEN?
ESC0:2C C1 C1   705 WLOOP  BIT $01C1 ; STRAFF ?
ESC3:58 FD      706 BM: WLOOP ; NEIN, ZURUECK
ESC5:8D 90 C8   707 STA $C090 ; AUSGEBEN !!!
ESC8:68         708 RTS
ESC9:         709 *
ESC9:         710 * *** PTAB IST EIN VORBEREITETER SPRUNG
ESC9:         711 * *** UM EVENTUELL IN EINER EIGENEN ROUTINE
ESC9:         712 * *** SPEZIELLE DRUCKERZEICHEN UMZUWANDeln
ESC9:         713 *
ESC9:4C C0 E5   714 PTAB   JMP WLOOP ; NICHT IMPLEMENTIERT
1 PTAB TAX      ; ZEICHEN ALS INDEX BENUTZEN
2 LDA TAE,X     ; ALTERNATIVES ZEICHEN LADEN
3 JMP WLOOP     ; AUSGEBEN
4 *
5 108 HEX 00F1073440505 ; UND SO WEITER

```

Bild 3. Auf diese Weise kann jedes Zeichen durch ein anderes ersetzt werden, natürlich wird eine komplette Tabelle benötigt.

infos

NEU! NEU! NEU! NEU! NEU! NEU! NEU! NEU!

16/32-K-RAM- + 8-K-ROM-Supermodul für den VC-20.
Das ist die preiswerteste Lösung für mehr Speicher im VC-20. Ersetzt 3 K + 8 K + 16 K oder 8 K + 16 K + 8 K im Autostartbereich. Außerdem ist noch ein EPROM-Steckplatz für 4 oder 8 K EPROM vorhanden.
Preis mit 16 K RAM bestückt **DM 128.-**
Preis mit 32 K RAM bestückt **DM 178.-**
2 RAMs zum Umtau von 16 auf 32 K **DM 60.-**

64-K-RAM-Supermodul.
Die beste aller Speichererweiterungen für den VC-20. Ersetzt alle anderen Speichermoduln von 3 K bis 32 K. Erweitert den Speicher des VC-20 um volle 64 KByte wenn mindestens ein 0-K-Dereich im VC-20 frei ist. Wer noch mehr Speicher braucht, kann bis zu vier dieser Module gleichzeitig verwenden und erhält so über 200 KByte RAM. Für die Bereiche B-K3 und B-K5 ist softwareseitig ein Schreibschutz schaltbar.
Preis für das 64-K-Supermodul **DM 278.-**

80-Zeichen-Supermodul für VC-20.
Vollständig neues Modul mit vielen zusätzlichen Funktionen. Das kleinste Modul dieser Art. Es paßt in ein Standard-Modulgehäuse und damit auch in die Modulbox. Auf diesem Modul befindet sich außerdem noch ein komplettes TEXTPRO-GRAMM im EPROM.
Preis für das 80-Z-Supermodul **DM 278.-**

Z-80A¹-Supermodul für den VC-20.
Macht aus Ihrem VC-20 einen vollwertigen Z-80-Rechner mit 4-MHz-Z-80A-CPU, 64 KByte RAM und DMA-Interface zum VC-20 sowie wahlweise dazu einen Floppydisk Controller zum Anschluß von 5,25"-Diskettenlaufwerken. Das Diskettenformat ist TPC 80² (Double Density) kompatibel. Mit Hochleistungssystem und CPM³-BIOS im FOM. 60 K freier Speicher für CPM.
Preis für das Z-80A-Supermodul **DM 398.-**

Floppydisk-Controller für 1-3 Laufwerke. Single oder Double Side, 40/80 Track Double Density. Anschlußkabe für 3 Diskettenlaufwerke **DM 348.-**
Achtung! Das Z-80A-Modul erfordert eine 8C-Zeichenkarte für eine einwandfreie Funktion der beigelegten Software!

Achtung! An alle Bastler und Freaks. Wegen Umstellung der Produktion verkaufen wir eine größere Anzahl von Platinen unserer alten VC-20-Module, z. B. Busplatinen mit 6 Steckplätzen 30 DM, 80 Z-Karte, vollbestückt (mit kleinen Fehlern) 90.- DM, sowie diverse andere Module.
Für den kleinen Geldbeutel haben wir außerdem noch diverse Gebrauchtgeräte vorrätig: z. B. Drucker ab 400.- DM, mehrere Video-Game-Computer ab 600.- DM.

¹ Z-80A ist ein eingetragenes Warenzeichen der Firma Zilog Inc.
² TPC-80 ist ein eingetragenes Warenzeichen der Firma Tandy Corp.
³ CPM ist ein eingetragenes Warenzeichen der Firma Digital Research.

Die absolute Sensation!!!
Alle unsere VC-20-Module laufen jetzt auch am COMMODORE 64!!!

Busadapter für ein VC-20-Modul **DM 48.-**
Busadapter für drei VC-20-Module **DM 140.-**
Bus-Supermodul für den VC-20.
Busplatine mit fünf Steckplätzen für Module, 3 K-RAM-Erweiterung und einem Steckplatz für 4/8 K EPROM. Eigene Stromversorgung. Selektionen über Schalter geführt.
Preis für das Bus-Supermodul **DM 223.-**

Software für Commodore VC-20:
Super-Speed Graphic Toolkit.
Das einzige Grafikprogramm für den VC-20, mit dem Sie in Basic bewegte Grafiken erzeugen können.
Erfordert mindestens ein 24-K-Supermodul.
Preis für das Graphic Toolkit **DM 93.-**

Pseudo-Disk-Dateiprogramm.
Programmierkomfort wie nie zuvor. Laden und Abspeichern von Programmen und Daten in Bruchteilen von Sekunden. Arbeitet mit Basicbefehlen (LOAD, SAVE, PRINT+, INPUT+, GET+), die jedoch teilweise erweitert wurden (APPEND von Programmen und RANDOM-Datell). Erfordert mindestens ein 64-K-Supermodul.
Preis für Pseudo-Disk **DM 49.-**

40 Zeichen/Zeile und Grafik.
Erlaubt die Darstellung von 40 Zeichen und 26 Zeilen mit dem VC-20 ohne Extra-Hardware. Dazu kommen Grafikbefehle zur Darstellung beliebiger Grafiken **DM 49.-**

Peripheriegeräte für alle Computer

12"-Monitor BMC BW12.
12" Mon., grüne oder bernsteinfarbene Bildröhre.
Preis für BMC12 ES (grün) **DM 368.-**
Preis für BMC12 EY (ber. isten farbe) **DM 398.-**
Sonderangebot: BMC 12A **DM 288.-**

Matrixdrucker 3MC PX-80.
80 Zeichen pro Sekunde, Superschriftbild (8x13-Matrix), 40 bis 142 Zeichen/Zeile, Subscript und Superscript, Kursivschrift, Fettdruck, Unterstreichen und Grafik, Fraktionswaage und verstellbare Traktorführung.
Preis mit Centronics-Interface **DM 998.-**
Preis mit Centronics- + V-24-Interface **DM 1148.-**

Tintenstrahldrucker.
superleise, mit 1A Schriftbild, 80 Zeichen/Sekunde, Serienmäßig mit Centronics- + V-24-Interface.
Preis für diesen Superdrucker **DM 1548.-**
Preis mit VC-20-VC-64-Interface **DM 1608.-**

In allen Preisen ist die gesetzliche Mehrwertsteuer enthalten. Für alle Lieferungen gelten ausschließlich unsere allgemeinen Liefer- und Zahlungsbedingungen.

MicroComputerSysteme Ingeborg Strie

Kirchweg 5, 2831 Schwaförden, Tel. 0 42 77 / 692

PROTON intelligente Tastaturen

High Quality - Low Cost

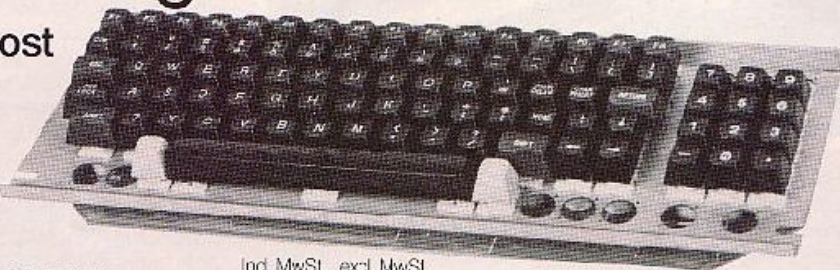
PROTON-Tastaturen sind mit Keyswitches von Futaba, weltführender Hersteller, aufgebaut. Diese Keyswitches werden auch von führenden Terminal-Herstellern wie Lear, Sigloger und Televideo eingesetzt.

PROTON-Tastaturen werden in Holland hergestellt. Neben den standardmäßigen Tastaturen sind auch kundenspezifische Tastaturen preislich sehr attraktiv, auch in kleineren Stückzahlen.

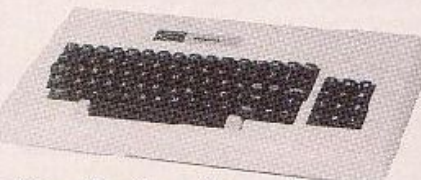
Befestigungsplatte aus Stahlblech. Auf die Platte wird keine mechanische Kraft ausgeübt. Ein zuverlässiges Funktionieren ist damit gewährleistet.

ASCII-Encoder mit wählbarer Tastenbelegung und parallelem und seriellem ASCII-Ausgang mit wählbaren Schritsteller-Daten. Größte Flexibilität: Anpassung an jeden Computer möglich.

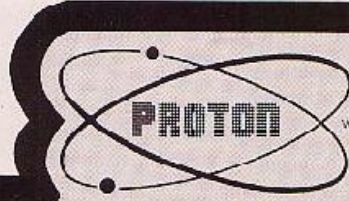
16 programmierbare Funktionstasten. Unter jeder der Funktionstasten können Sie einen String bis zu 15 Charakteren in EPROM ablegen, so daß Änderungen - auch nachträglich - problemlos sind. Von der Tastatur aus können diese Strings vorübergehend überschrieben werden (in das interne RAM). Standardmäßig sind die am häufigsten benutzten BASIC-Befehle abgelegt.



	Ind. MwSt.	excl. MwSt.
KB 2 Matrixtastatur 10 x 10	DM 239⁴⁰	210⁰⁰
KB2E-G komplett gebaute Tastatur mit Encoder und Gehäuse	DM 416¹⁰	365⁰⁰
KB2E-B wie KB2E-G, jedoch als Bausatz	DM 324⁹⁰	285⁰⁰
QWERTZ U als Bausatz auf deutsche Tastenbelegung incl. EPROM	DM 27⁸²	24⁴⁰
APPLECABLE Hochbandkabel (1 m) mit zwei 16-pol. IC-Steckern für Apple-Anschluß	DM 38⁷⁶	34⁰⁰



Bitte fordern Sie sofort das Informationsmaterial an!
OEM's fragen Sie gezielt an!



In den Preisen sind 14% MwSt. enthalten. Soweit nichts anderes vereinbart, erfolgt der Versand gegen Nachnahme. Pauschalbetrag für Versand und Verpackung 9,50 DM.

TEEPE GmbH
Vorm Tor 8 / D-5395 Weilrod
Telefon 06083/2329/553

Software-Know-how

Genie als ASCII- Fernschreiber

Diethelm Buttke

Als Anregung für erste Versuche auf dem Gebiet der Datenfernübertragung möge dieser Beitrag dienen. Zugleich bietet er eine Anleitung zum Erstellen von Programmen für den seriellen Datenverkehr in Eintonverfahren. Wer einen Genie-Computer besitzt, kann die Software unverändert abtippen. Besitzern anderer Z80-Systeme wird die Anpassung durch eine ausführliche Beschreibung erleichtert.

teil, daß er mehr Kombinationsmöglichkeiten besitzt. Es ist nicht mehr nötig, zwischen einer Buchstaben- und Zahlenebene zu unterscheiden, auch können mehr verschiedene Zeichen sowie Groß- und Kleinbuchstaben gesendet werden.

Soll ein Computer-Fernschreiber mit dem CCITT-Code arbeiten, sind mehrere Tabellen und Umschalttroutinen erforderlich, um den vorhandenen ASCII-Code in den CCITT-Code umzusetzen. Bleibt man dagegen im ASCII-Code, so gestaltet sich das Programm recht einfach.

Ausstattung

Im folgenden wird ein kleines Programm beschrieben, mit dem sich Fernschreibversuche durchführen lassen. Es sind dazu zwei Computer erforderlich, die über eine Fernleitung und ein einfaches Interface an ihren Kassettenschnittstellen verbunden sind. Für die erste Erprobung genügt es, die Zeichen auf einem Kassettenschnittstellen aufzunehmen, der dann anschließend als Sender benutzt wird. Ein Interface ist solange noch nicht nötig.

Leider läuft das Programm nur auf GENIE-Computern unverändert, bei anderen Systemen müssen die Ein- Ausgabeadressen geändert werden. Auch der sonst kompatible TRS-80 hat eine andere IN/OUT Organisation.

Wird die 'F1'-Taste am Computer gedrückt, so kann man den eingebauten Recorder verwenden. Eingriffe in das Gerät sind nicht erforderlich. Die Übertragung erfolgt im ASCII-Code nach dem Eintonverfahren. Bei 'Mark' liegen auf der Fernleitung 0 Volt, bei 'Space' ein Ton, oder besser ein 'Burst', mit einer Frequenz von 1275 Hz. Die Baudrate, das heißt die Anzahl der pro Sekunde übertragenen Bits, ist zwischen 45,45 Baud und 100 Baud wählbar. Die Übertragung ist nur wenig stör anfällig, so daß auch die akustische Einkopplung auf eine Fernleitung oder drahtlose Übermittlung möglich sind. Dabei sind die postalischen Vorschriften zu beachten!

Die gesendeten oder empfangenen Zeichen erscheinen nach jedem Tastendruck gleichzeitig auf dem Bildschirm beider Geräte, wobei auch Korrekturen mit der Linkspfeiltaste möglich sind. Beim üblichen Fernschreiben wäre das undenkbar. Wird ein eventuell vorhandener Drucker eingeschaltet, so wird der Text auch ausgedruckt.

Software

Das Programm (Tabelle 1) besteht aus zwei Teilen, einem Maschinenprogramm und einem kleinen BASIC-Programm, das die Baudrate einstellt. Nach dem Laden beider



In jedem Computer sind alle wichtigen Baugruppen einer elektronischen Fernschreibmaschine bereits enthalten. Es ist daher nicht schwierig, ein geeignetes Programm zu entwickeln, das über den meist vorhandenen Kassettenschluß das Senden und Empfangen von Fernschreibzeichen gestattet.

Beim Fernschreiben erfolgt die Datenübertragung seriell und asynchron im Start-Stopp-Betrieb. Sender und Empfänger sind nur während der Übermittlung eines Zeichens annähernd synchron. Danach wartet der Empfänger beliebig lange in Ruhestellung, bis das nächste Zeichen eintrifft.

Aus geschichtlichen Gründen werden die Zeichen beim Amateur-Fernschreiben und zum

Teil auch noch bei kommerziellen Fernschreibsendungen nach dem CCITT-Nr. 2-Code verschlüsselt, der auf Jean Maurice Baudot (1845—1903), einen französischen Telegrafentechniker, zurückgeht. Dies ist ein zweiwertiger Code (ein — aus Δ Mark — Space) mit fünf Signalelementen, der stets mit einem Startschritt (Space) beginnt und einem Stoppschritt von mindestens eineinhalb Mark-Längen endet.

Auch im Computer wird jedem Zeichen ein Code zugeordnet, der aber vom CCITT-Code abweicht. Es ist der ASCII-Code, ebenfalls zweiwertig, aber mit sieben Signalelementen, denen stets eine Null (Space) vorangestellt ist.

Gegenüber dem CCITT-Code hat der ASCII-Code den Vor-

Programmenteile erfolgt die Aufforderung, die gewünschte Baudrate einzugeben. Das BASIC-Programm berechnet nun alle Verzögerungszeiten und setzt mit 'POKE' alle Zeitschleifen auf den richtigen Wert, sofern die gewünschte Baudrate von der Grundeinstellung 45.45 Baud abweicht. Danach erscheint in der rechten unteren Bildschirmcke ein 'E'. Der Computer zeigt damit an, daß das Gerät empfangsbereit ist. Mit der NEW LINE Taste kann man auf Sendung umschalten. Dies wird durch ein 'S' in der rechten unteren Bildschirmcke signalisiert.

Betrachten wir kurz den Programmablauf, und beginnen wir mit 'Senden'.

Die Tastatur wird abgefragt und das geforderte Zeichen auf Bildschirm und gegebenenfalls Drucker ausgegeben. Mit der BREAK-Taste kann die Sendung unterbrochen werden, und das Gerät schaltet wieder auf Empfang. Wird der 'Klammeraffe' gedrückt, kehrt der Computer in das BASIC-Programm zurück, etwa um die Baudrate zu ändern.

Das Zeichen wird nun in das E-Register geladen. Dann 'testet' der Prozessor den Wert jedes einzelnen Datenbits. Nach jeder Prüfung wird der Inhalt des E-Registers um ein Bit verschoben, bis alle acht Bit erkannt sind. Danach springt das Programm zur Tastaturabfrage zurück, um ein neues Zeichen zu übernehmen.

Wurde ein Bit als Null erkannt (Space), erfolgt ein Sprung zum SPACE-Unterprogramm. Die

Datenleitungen D0 und D1 werden abwechselnd HIGH und LOW geschaltet, so daß am Ausgang von Port 'FF' eine Rechteckschwingung mit einer Frequenz von 1275 Hz und einer Dauer von 22 ms (bei 45.45 Baud) anliegt. Wird ein Bit als Eins erkannt (Mark), so geschieht im MARK-Unterprogramm 22 ms lang nichts (bei 45.45 Baud).

Das Signal wird leider einem internen Spannungsteiler auf etwa 0,4 Volt_{ss} abgeschwächt. Es liegt zwischen den Pins 2 und 5 an der Kassettenausgangsbuchse. Bevor es auf die Fernleitung gegeben wird, muß es in einem kleinen Interface wieder verstärkt werden, wenn man keinen Eingriff in den Computer vornehmen will. Einen Vorschlag für ein entsprechendes Interface findet Sie in Bild 1.

Die Empfangsroutine verläuft in wesentlichen Teilen umgekehrt. Zunächst wird das Eingangs-Flip-Flop von Port 'FF' zurückgesetzt. Hierzu wird ein kleines Unterprogramm (CALL 021Eh) aus dem ROM bemüht. Danach wartet der Prozessor in einer Schleife auf den Startschritt. Dieser ist bei ASCII genau wie bei CCITT eine Null, also ein Burst. Dieser Burst triggert das Flip-Flop, und die Schleife wird verlassen.

Nun prüft der Prozessor in der Mitte der Übertragungszeit für jedes Bit den Status des Flip-Flops. Es folgt daher eine Verzögerungsschleife, welche die weitere Abfrage auf die Schrittmittel verschiebt. Nun wird während einer kurzen Abtastzeit darauf gewartet, ob das Flip-Flop getriggert wird (Spa-

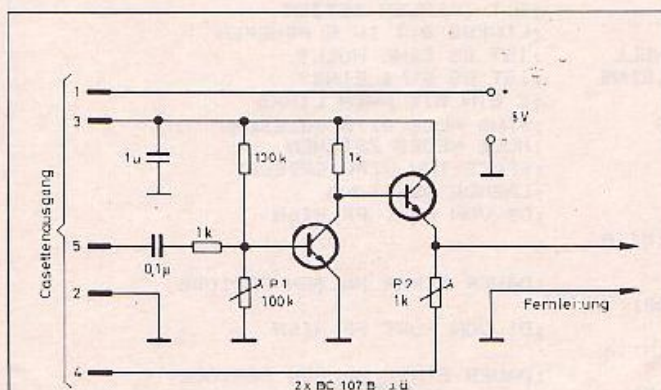


Bild 1. Interface für die Fernleitung

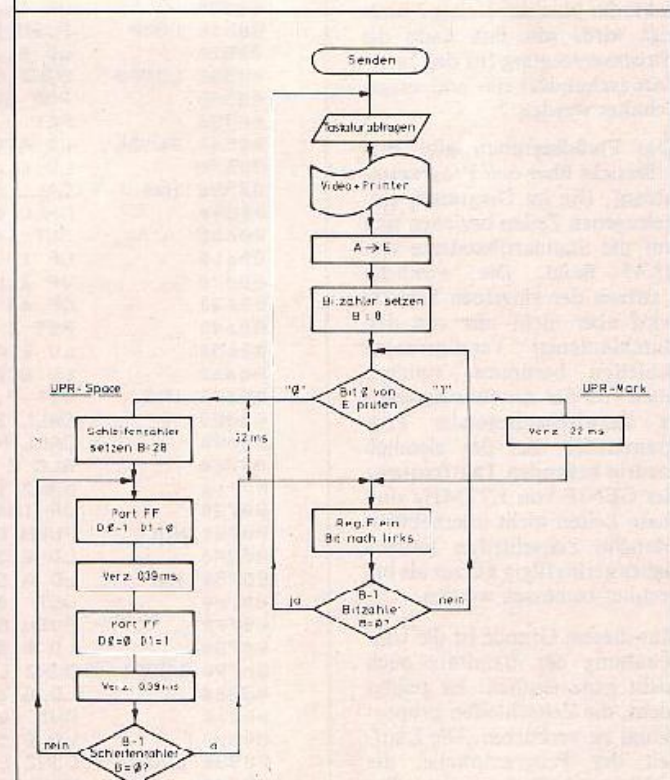
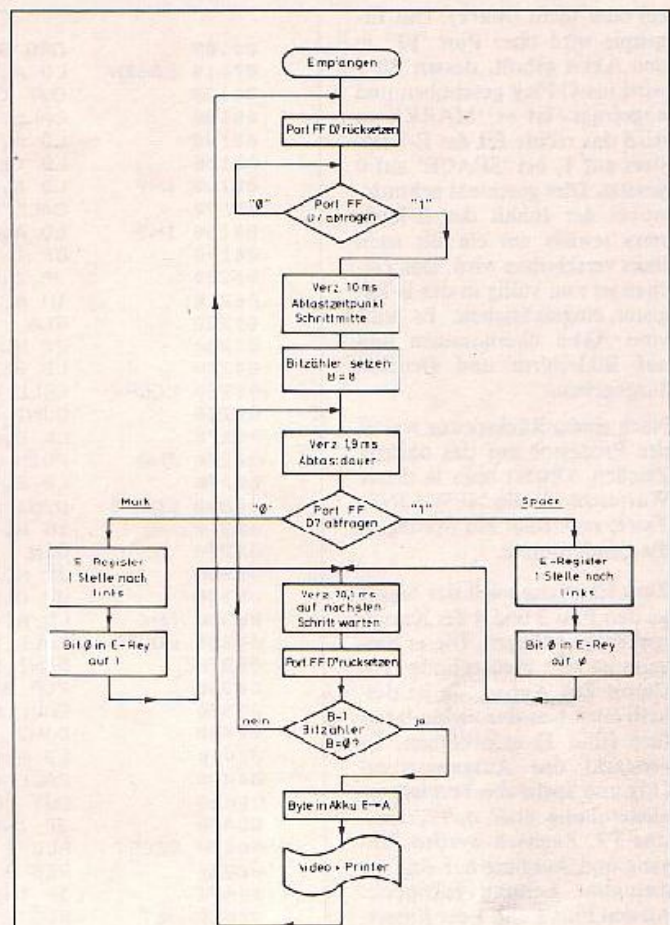


Bild 2. Flußdiagramm zu Tabelle 1

ce) oder nicht (Mark). Das Ergebnis wird über Port 'FF' in den Akku geholt, dessen Bit 7 wird ins C-Flag geschoben und abgefragt. Ist es 'MARK', so wird das rechte Bit des E-Registers auf 1, bei 'SPACE' auf 0 gesetzt. Dies geschieht achtmal, wobei der Inhalt des E-Registers jeweils um ein Bit nach links verschoben wird. Das Zeichen ist nun völlig in das E-Register eingeschrieben. Es wird vom Akku übernommen und auf Bildschirm und Drucker ausgegeben.

Nach einem Rücksprung wartet der Prozessor auf das nächste Zeichen. Drückt man in dieser Warteschleife die NEW-LINE-Taste, so erfolgt ein Sprung in die Senderoutine.

Zur Empfang muß das Signal an den Pins 2 und 4 der Kassettenbuchse anliegen. Dieser Eingang ist sehr niederohmig (100 Ohm). Zur Anpassung ist deshalb auch hier das kleine Interface (Bild 1) erforderlich. Es verstärkt das Ausgangssignal (T1) und speist die Fernleitung niederohmig über den Transistor T2. Zugleich werden Eingang und Ausgang auf eine gemeinsame Leitung gekoppelt. An den Pins 3 und 1 der Kassettenbuchse liegt ein Arbeitskontakt, der über das Relais 2 betätigt wird. Mit ihm kann die Stromversorgung für das Interface (sekundär) ein- und ausgeschaltet werden.

Das Flußdiagramm gibt eine Übersicht über den Programmablauf. Die im Diagramm eingezeichneten Zeilen beziehen sich auf die Standardbaudrate von 45,45 Baud. Die wirkliche Laufzeit der einzelnen Schritte wird aber nicht nur von den durchlaufenen Verzögerungsschleifen bestimmt, sondern auch von der Ausführungsdauer dazwischenliegender Programmteile. Bei der ziemlich niedrig liegenden Taktfrequenz des GENIE von 1,77 MHz sind diese Zeiten nicht unerheblich. Manche Zeitschleifen müssen daher geringfügig kürzer als berechnet bemessen werden.

Aus diesem Grunde ist die Umschaltung der Baudrate auch nicht ganz einfach. Es genügt nicht, die Zeitschleifen proportional zu verkürzen. Die Laufzeit der Programmteile, die nicht innerhalb dieser Schleifen liegen, bleibt ja unverändert erhalten. Daher müssen die Zähler etwas überproportional ver-

```

00100      ORG 32000
00110      LESEN  LD A,16          ;SELECT CASSETTE 2
00120      OUT (0FEH),A
00130      CALL 215H          ;CASSETTE AN
00140      LD A,69            ;E AUF BILDSCHIRM UNTEN RECHTS
00150      LD (16383),A
00160      IN9  LD E,0FFH
00170      CALL 021EH          ;EINGANGS-FLIP-FLOP RUECKSETZEN
00180      IN5  LD A,(14403)    ;PRUEFEN OB NEW LINE GEDRUECKT
00190      CP 1
00200      JP 2,SENDEN        ;WENN JA, DANN AUF SENDEN SCHALTEN
00210      IN A,(0FFH)        ;AUF STARTSCHRITT WARTEN
00220      RLA                ;BIT 7 VON AKKU IN C-FLAG
00230      JP NC,IN5
00240      LD B,25            ;ABTASTZEITPUNKT EINSTELLEN
00250      LOOP4 CALL LOOP
00260      DJNZ LOOP4
00270      LD B,8             ;BIT-ZAEHLER SETZEN
00280      IN8  PUSH BC
00290      LD B,255          ;ABTASTZEIT EINSTELLEN
00300      LOOP3 DJNZ LOOP3
00310      IN A,(0FFH)        ;BIT VON PORT FF HOLEN
00320      RLA                ;BIT 7 VON AKKU IN C-FLAG
00330      JP NC,SET          ;IST ES MARK?
00340      JP C,RESET         ;IST ES SPACE?
00350      IN4  LD B,40        ;AUF NAECHSTEN SCHRITT WARTEN
00360      LOOP5 CALL LOOP
00370      DJNZ LOOP5
00380      POP BC
00390      CALL 021EH          ;FLIP-FLOP RUECKSETZEN
00400      DJNZ IN8           ;SIND ALLE 8 BIT IN E?
00410      LD A,E             ;BYTE IN AKKU
00420      CALL 033H          ;ZEICHEN AUF BILDSCHIRM
00430      OUT (0FD4),A       ;ZEICHEN AUF DRUCKER
00440      JP IN9            ;HOLE NEUES ZEICHEN
00450      RESET RLC E        ;E EINE STELLE NACH LINKS
00460      RES 0,F           ;RECHTES BIT AUF 0
00470      JP IN6            ;HOLE NEUES BIT
00480      SET 0,E           ;E EINE STELLE NACH LINKS
00490      SET 0,E           ;RECHTES BIT AUF 1
00500      JP IN6            ;HOLE NEUES BIT
00510      LOOP  PUSH BC
00520      LD B,62           ;GEMEINSAME INNERE VERZ.SCHLEIFE
00530      LOOP4 DJNZ LOOP4
00540      POP BC
00550      RET
00560      SENDEN LD A,83      ;S AUF BILDSCHIRM UNTEN RECHTS
00570      LD (16383),A
00580      IN4  CALL 049H      ;TASTATURABFRAGE
00590      CALL 033H          ;ZEICHEN AUF BILDSCHIRM
00600      OUT (0FD4),A       ;ZEICHEN AUF DRUCKER
00610      CP 1              ;IST ES BREAK?
00620      JP 2,LESEN        ;WENN JA, DANN AUF EMPFANG
00630      CP 64             ;IST ES KLAMMERAFFE?
00640      RET 2              ;WENN JA, DANN ZURUECK IN BASIC
00650      LD E,A            ;ZEICHEN VON AKKU IN E-REGISTER
00660      LD B,8             ;BIT-ZAEHLER SETZEN
00670      IN3  BIT 7,E        ;LINKES BIT IN E PRUEFEN
00680      CALL 2,NULL        ;IST ES EINE NULL?
00690      CALL NZ,EINS       ;IST ES EINE EINS?
00700      RLC E              ;E EIN BIT NACH LINKS
00710      DJNZ IN3           ;SIND ALLE BITS GELESEN?
00720      JP IN4            ;HOLE NEUES ZEICHEN
00730      NULL  PUSH BC
00740      LD B,28           ;SPACE-TON WIRD ERZEUGT
00750      IN1  LD A,5         ;LAENGE DES TONS
00760      OUT (0FFH),A       ;D0 VON PORT FF HIGH
00770      PUSH BC
00780      LD B,50           ;DAUER EINER HALBEN PERIODE
00790      LOOP1 DJNZ LOOP1
00800      LD A,6             ;D1 VON PORT FF HIGH
00810      OUT (0FFH),A
00820      LD B,50           ;DAUER EINER HALBEN PERIODE
00830      LOOP2 DJNZ LOOP2
00840      POP BC
00850      DJNZ IN1           ;IST SPACE ZU ENDE?

```

Tabelle 1. Unterprogramme für Empfangen (Lesen) und Senden


```

00860      POP BC
00870      RET
00880 EINS  PUSH BC      ;LAENGE VON MARK ABWARTEN
00890      LD E,46      ;OHNE TON
00900 LOOP6 CALL LOOP
00910      DJNZ LOOP6
00920      POP BC
00930      RET      ;ZURUECK WENN MARK-PAUSE ZU ENDE
00940      END

```

```

13 CLS:PRINT"BITTE WARTEN!"
23 DATA 22,14,211,254,205,21,2,62,69,50,255,63,30,255,205,30
33 DATA 2,58,64,56,254,1,202,97,125,219,255,23,210,17,125,6
43 DATA 25,205,70,125,16,251,6,8,197,6,255,16,254,219,255,23
53 DATA 210,83,125,218,76,125,6,48,205,70,125,16,251,193,205,30
63 DATA 2,16,229,125,245,51,0,211,253,95,12,125,203,3,203,131
73 DATA 195,54,125,203,3,203,195,195,54,125,197,6,62,16,254,193
83 DATA 201,62,39,50,255,63,205,73,0,205,51,0,211,253,254,1
93 DATA 202,6,125,254,64,200,75,6,0,203,123,204,136,125,196,161
103 DATA 125,203,3,16,244,195,102,125,197,6,28,62,5,211,255,197
113 DATA 6,50,16,254,62,6,211,255,6,50,16,254,193,16,236,193
123 DATA 201,197,6,46,205,70,125,16,251,193,201
133 A=32000:E=32170:FOR X=A TO E:READ B:POKE X,B:NEXT
143 RESTORE:C=0:FOR X=A TO E:READ B:C=C+B:NEXT:C=C-20893
153 CLS:IF C<>0 THEN PRINT"FEHLER IN DATAZEILEN!":STOP
163 CLS:B=45.45:INPUT"BAUDRATE:";B:K=45.45/B
173 A=INT(255*K+.5):U=INT(62*K+.5):S=INT(40+(K*100-100)/(K*.62)+.5)
183 L0=INT(28*K+.5):L1=INT(46+(K*300-300)/(K*.62)+.5)
193 POKE32042,A:POKE32092,U:POKE32055,S
203 POKE32138,L0:POKE32163,L1
213 POKE16526,0:POKE16527,125:D=USR(0)
223 GOTO140

```

Tabelle 2. BASIC-Programm für GENIE als ASCII-Fernschreiber. In den Zeilen 20 bis 130 wird das Maschinenprogramm geladen. Die Prüfzeilen 135 und 136 können nach dem ersten Durchlauf wieder gelöscht werden.

```

7D00: 3E 1A D3 FE CD 15 02 3E 45 32 FF 3F 1E FF CD 1C
7D10: 02 3A 40 38 FE 01 CA 61 7D DB FF 17 D2 11 7D 06
7D20: 19 CD 5A 7D 10 FB 06 08 C5 06 FF 10 FE DB FF 17
7D30: D2 53 7D DA 4C 7D 06 2E CD 5A 7D 10 FB C1 CD 1E
7D40: 02 10 E5 7B CD 33 00 D3 FD C3 0C 7D CB 03 CB 83
7D50: C3 35 7D CB 03 CB C3 C3 36 7D C5 06 3E 10 FE C1
7D60: C9 3E 53 32 FF 3F CD 49 00 CD 33 00 D3 FD FE 01
7D70: CA 03 7D FE 40 C8 5F 06 08 CB 7B CC 88 7D C4 A1
7D80: 7D CB 03 10 F4 C3 66 7D C5 06 1C 3E 05 D3 FF C5
7D90: 06 32 10 FE 3E 06 D3 FF 06 32 10 FE C1 10 EC C1
7DA0: C9 C5 06 2E CD 5A 7D 10 FB C1 C9 00 00 00 00 00

```

Tabelle 3. HEX-DUMP zum Maschinenprogramm

```

10 CLS:B=45.45:INPUT"Welche Baudrate:";B:K=45.45/B
20 A=INT(255*K+.5):U=INT(62*K+.5):S=INT(40+(K*100-100)/(K*.62)+.5)
30 L0=INT(28*K+.5):L1=INT(46+(K*300-300)/(K*.62)+.5)
40 POKE32042,A:POKE32092,U:POKE32055,S
50 POKE32138,L0:POKE32163,L1
60 POKE16526,0:POKE16527,125:D=USR(0)
70 GOTO10

```

Tabelle 4. BASIC-Programm zum Einstellen der BAUD-Rate

ringert werden. Ein Blick auf das BASIC-Programm zeigt, daß hierzu einige zusätzliche Berechnungen erforderlich sind, um einen Korrekturfaktor zu bestimmen, der die Laufzeit der Programmelemente aus-

gleicht, die nicht von den Zeit-
schleifen erfaßt werden.

Bei der Inbetriebnahme wird P1 im Interface bei angeschlossenem Computer so justiert, daß ohne Signal gerade 0 Volt auf

der Fernleitung liegen. Der Empfangspegel wird mit P2 eingestellt. Mit einem Piezosumierer, der parallel zur Fernleitung liegt, können die Signale kontrolliert werden. Auch als 'Klingel' ist er nützlich. □

Hardware für C-64

EBG 9600

Schnelles Kassettenlaufwerk für Dater- und Programmaufzeichnung, speziell Meßdatenerfassung. Übertragungsraten ca. 1 K Byte/sec. Bausatz mit Anschlußkabel (ohne Netzteil und Gehäuse)

Preis 680,- DM
incl. MWSt.

EBG 2064

Epson-Interface zum Anschluß aller Epson Drucker an den C-64

Preis 339,- DM
incl. MWSt.

EBG 0488

Paralleles IEC-Interface: ermöglicht Anschluß an C-64, an CBM 4040, CBM 8050, Drucker, etc.

Preis 298,- DM
incl. MWSt.

Rufen Sie uns an:
(0 61 51) 31 38 90

EBG

Elektronik Bauelemente GmbH
Lothar Schanuel
Heidelberger Str. 73
6100 Darmstadt
Tel. (06151) 313890
Tx. 419/160

Die Frage, wann ein Programm oder ein Betriebssystem als 'CP/M-kompatibel' gelten darf, ist von vornherein nicht einfach zu entscheiden. Das CP/M gibt es in verschiedenen, aufeinander aufbauenden Versionen, die jedoch auch nicht ganz aufwärtskompatibel sind.

CP/M-1.4 ist die Version, die bis circa 1980 überall in Betrieb war. CP/M-2.x ergab Fortschritte in Bezug auf die Disketten- und Filegröße (nun bis 8 Megabyte statt 255 Kilobyte) und die Unterstützung des Video-Backspaces sowie der Einteilung der Diskette in Benutzerbereiche ('USER' 0-15). Zu CP/M-1.4, das immer noch auf einigen wenigen Geräten das einzige Betriebssystem ist, bleibt man nur kompatibel, wenn das gleiche Diskettenformat benutzt wird, man nur unter USER 0 arbeitet, keine File-Attribute anwendet, und ein File nicht länger als 256 Kbyte lang ist.

CP/M-3.x hat gegenüber CP/M 2.x ein neues File-Attribut (Archive) und DATUM-Einträge in der Directory. Sie bringen CP/M-2.x etwas aus dem Tritt; für auszutauschende Daten sollte man darauf verzichten.

Das CP/M-Betriebssystem stellt dem Programmierer eine Reihe von Systemfunktionen zur Verfügung (Tabelle 1). Wenn man mit diesen Funktionen nicht auskommt, kann man noch die BIOS-Sprungtabelle ausnutzen (nicht mehr für CP/M-3.x). Diese Technik hat

sich eingebürgert, weil das 'alte' CP/M-1.4 keine Systemfunktion 'Input ohne Echo' besaß und weil die Umgehung des BDOS etwas schneller abläuft. Wenn der Character-Output auch über die BIOS-Sprungtabelle abgewickelt wird, kann auch das Zeichen '07' (TAB) als solches ausgegeben werden, und die Steuerzeichen '13' (^S) und '10' (^P) können eingegeben werden, ohne daß das CP/M sie interpretiert. Man darf aber nur über die Adressen der Sprungvektoren bei 0000 und 0005 zugreifen. Ein Programm, das absolute Werte für die Sprungtabelle annimmt, ist ganz klar nicht transportabel.

Sehr viele Programme nehmen an, daß die BIOS-Sprungtabelle an einer Page-Grenze beginnt. Wenn dem nicht so ist (KONTROLN) gibt es in solchen Fällen Probleme.

Wenn nur die Werte für Console- und Printer-I/O benötigt werden, sollte man sich diese im Assembler-Programm ausrechnen (Bild 1). Werden auch noch die Disk-Funktionen benötigt, kann es sich empfehlen, die ganze Sprungtabelle zu kopieren (Bild 2). Sowohl bei BDOS-Aufrufen als auch bei den Sprungvektoren des BIOS sei ausdrücklich gesagt, daß alle Register verändert werden können. Achtung für Z80-Programmierer: das gilt auch für die INDEX-Register und den zweiten Registersatz! Umgekehrt sollte derjenige, der sein BIOS selbst programmiert, die

se Register dabei nicht verändern, um nicht in Schwierigkeiten mit einigen Anwenderprogrammen zu kommen (frühe Versionen von Interchange und Mychess). Bei angeblich 'kompatiblen' Betriebssystemen sollte man genau prüfen, ob der BDOS-Return-Code, wie bei CP/M, sowohl in den Registern A, BC und auch in HI zurückgegeben wird.

Z80-Besitzer sollten beim Programmieren auch daran denken, daß eine Vielzahl von CP/M-Systemen 'nur' mit einer 8080- oder 8085-CPU betrieben wird. Wenn jemand die speziellen Z80-Befehle benutzt, sollte er auch prüfen, ob das Programm auf den genannten CPUs abläuft. Es ist inzwischen fast selbstverständlich, daß man die Versionsnummer des verwendeten CP/M prüft und entweder eine Fehlermeldung ausgibt, oder sein Programmverhalten darauf einrichtet.

In der 'guten alten Zeit' des CP/M, vor der Version 1.4, gab es nur das von der Firma IBM abgeschauten Diskettenformat (77 Spuren, je 26 Sektoren zu je 128 Bytes). Davon werden zwei Tracks für das Betriebssystem selbst gebraucht, so daß eine Diskette dann 241 Kilobyte 'netto' enthält. Auch heute ist es noch so, daß alle CP/M-Computer, die ein 8"-Laufwerk haben, dieses Standardformat lesen können. Zwar gab es für CP/M-1.4 auch schon Implementationen für Mini-Laufwerke; mit dem Erscheinen von

CP/M 2.x wurde es aber wesentlich einfacher, verschiedene Diskettenformate auf einem CP/M-Rechner zu 'fahren'. Da keine Firma schnell genug war, den 'Marktführer' zu spielen, ist es zu dem bedauerlichen Chaos der 5 1/4"-Diskettenformate gekommen. Es mag häufig auch mitgespielt haben, daß man den Kunden mit einem 'eigenen' Diskettenformat an sich binden wollte. Es gibt ja auch genug Variablen, mit denen man bei der Implementierung von CP/M-Disketten spielen kann:

Man kann 128, 256, 512 oder 1024 Bytes pro physikalischen Diskettensektor schreiben.

Durch die Wahl der Anzahl der Sektoren können zwischen 2048 (16*128) und 5632 (11*1024) Bytes pro Spur geschrieben werden.

Man kann 35 oder 40 Spuren (auf 96tpi-Laufwerken auch 70 oder 80) Spuren pro Diskettenseite nutzen. Man kommt so auf eine Diskettenkapazität, die zwischen 70 und 800 KByte pro 'Scheibe' liegt.

Für das Betriebssystem werden 1, 2, 3 oder 4 Spuren gebraucht. Sie sind nicht unbedingt in der gleichen Weise formatiert wie der Rest der Diskette.

Man sieht 32, 64, 128 oder 256 Directory Entries vor.

Der Diskettenplatz wird in 1, 2 oder 4 KByte-Blöcke eingeteilt.

Holger Petersen

Was heißt 'CP/M-kompatibel'?

CP/M-Software läuft auf jedem CP/M-Rechner. Diese Aussage gilt für fast alle kommerziell angebotenen Programme. Probleme tauchen aber dann auf, wenn man Software selbst schreibt. Dann ist es notwendig, über die verschiedenen CP/M-Versionen und die jeweils verwendeten Floppy-Formate informiert zu sein. Dieser Artikel zeigt, wann ein Programm als kompatibel gelten darf und welche Bedingungen eingehalten werden müssen.


```

init:
ld hl,(1) ; Lade Adresse des Warm-Start-Vektors
ld de,3 ; Länge eines Sprung-Befehls
add hl,de ; nun zeigt HL auf Console-Status
ld (CSTAT+1),hl ; eingesetzt
add hl,de ; zeige auf Console-Input
ld (CINP+1),hl ; einsetzen
add hl,de ; zeige auf Console-Output
ld (COUT+1),hl ; einsetzen
ret

CSTAT jp $ ; nach Ablauf der Routine 'INIT'
CINP jp $ ; stehen hier Sprünge, die direkt
COUT jp $ ; zu den BIOS-Routinen führen

```

Bild 1. Berechnung der Console-BIOS-Adressen

```

init:
ld hl,(1) ; Lade Adresse des Warm-Start-Vektors
ld de,table ; Startadresse der programm-internen Tzbl.
ld bc,16*3 ; 16 Sprünge (CP/M-2.x)
; sind zu kopieren

ld e,03h ; setz: bei Z80 das Half-Carry Flag
and ah,0fh ; macht das unterschiedliche Ver-
; halten deutlich
cp 09h ; halten deutlich
jp z,Z80 ; sonst 8080

move80:
ld d,(hl) ; hole Source-Byte
ld (de),a ; hin zur Destination
inc hl
inc de
cec bc ; 16 Bit INC/DEC setzen kein FLAG!
ld a,b
rr c ; nur NULL, wenn beide Register 0
jp nz,move80 ; nochmal ...

ret

; Z80-spezifisch LDIR
Z80:
ldir ; kopiert den Bereich
ret

TABLE:
WSTART jp $ ; hier stehen
CSTAT jp $ ; nach Ablauf
CINP jp $ ; der Routine
COUT jp $ ; 'INIT' die
LSTOUT jp $ ; BIOS-Routinen
READER jp $ ; direkt im
PUNCH jp $ ; Assembler-
HOME jp $ ; Listing als
SELDISK jp $ ; Label zur
SETRK jp $ ; Verfügung
SETSEK jp $ ; Man beachte
SETDMA jp $ ; aber die
READ jp $ ; Parameter-
WRITE jp $ ; Konventionen
LSTST jp $ ; des CP/M
SEKTRN jp $

```

Bild 2. Kopieren der Sprangleiste

Zusätzlich kann man den Sektor-Skew modifizieren oder die Sektoren invertiert auf die Diskette schreiben.

Alle Möglichkeiten vervielfachen sich, wenn man doppelseitige Laufwerke benutzt. Die Sektoren können

auf der zweiten Seite weiter-numeriert werden, oder von vorn beginnen, und so weiter.

Wer exotisch sein will, benutzt 77-Track-Laufwerke oder die neuen 5¼" Laufwerke, die in der Kapazität und Übertra-

gungsrate kompatibel zu 8"-Laufwerken sind, und kommt dann auf 1,2 Megabyte. Von hard-sektorierten und anderen, nicht das IBM-soft-sektorierte Verfahren benutzenden Aufzeichnungsformaten (Apple, Commodore, Osi bzw. Sirius), soll gar nicht erst gesprochen werden. Diese Disketten sind wohl nur auf dem jeweiligen Originalrechner wieder zu lesen. Besitzer solcher Computertypen sind für den Datenaustausch auf die Benutzung von seriellen Schnittstellen angewiesen.

Bisher hat kaum eine Firma einen neuen CP/M-Rechner mit 5¼"-Laufwerken vorgestellt, der als Boot-Format ein schon existierendes Format benutzte (und wenn, war es wohl Zaifall?). In der Tabelle 3 sind einige Vertreter von 5¼"-CP/M-Formaten vorgestellt. Erfreulich ist aber, daß immer mehr

Firmen dazu übergehen, das Lesen und oft auch Schreiben auf einigen Fremdformaten zu ermöglichen. Pionier war Adam Osborne mit vier verschiedenen Formaten auf dem Osborn-1, die automatisch beim 'Einloggen' erkannt werden. Bei einigen anderen Rechnern muß man dem zweiten Laufwerk jeweils ein neues Diskettenformat zuordnen. Gerade für den OSBORN-Rechner gibt es aber ein Programm, um noch mehr Formate lesen zu können, mit dem beziehungsreichen Titel 'OZMOSIS'. Es ist aber auch schon im OSBORN-BIOS ein Platz vorgesehen, um ein weiteres Diskettenformat automatisch zu lesen. Die Adressen sind in der Tabelle 4 aufgeführt. Diese soll nicht als Kochrezept verstanden werden, sondern als Hilfe für diejenigen, die sich in BIOS-Modifikationen auskennen.

Nummer	CP/M-1.4	CP/M-2.x	CP/M-3.x
0	Warmstart	dito	dito
1	Console Input	dito	dito
2	Console Output	dito	dito
3	Reader Input	dito	Aux. Input
4	Punch Output	dito	Aux. Output
5	List Output	dito	dito
6	Return NULL	Direct IO	dito
7	Interr. I/O-Byte	dito	Aux. Inp. Status
8	Set I/O-Byte	dito	Aux. Out. Status
9	Print String	dito	dito
10	Read Console Buffer	dito	dito
11	Console Status	dito	dito
12	Return NULL	get Version #	dito
13	Reset Disk System	dito	dito
14	Select Disk	dito	dito
15	Open File	dito	dito
16	Close File	dito	dito
17	Search First	dito	dito
18	Search Next	dito	dito
19	Delete File	dito	dito
20	Read Record	dito	dito
21	Write Record	dito	dito
22	Make File	dito	dito
23	Rename File	dito	dito
24	Return Logir Vektor	dito	dito
25	Return Current Disk	dito	dito
26	Set DMA	dito	dito
27	Get Allocation Adr.	dito	dito
28	*Write Protect Disk	dito	dito
29	*Get R/O-Vektor	dito	dito
30	*Set Directory DMA	Set File Attribute	dito
31	%	Get DPB-Adresse	dito
32	%	Set/Get User Number	dito
33	%	Read Random	dito
34	%	Write Random	dito
35	%	Compute File Size	dito
36	%	Set Random Record	dito
37	%	+ Reset Drive	dito
38	%	\$	\$
39	%	\$	\$
40	%	+ write random with Zero Fill	

* 28 — 30 unter CP/M-1.4 nicht dokumentiert
+ 37 & 40 unter CP/M-2.x nicht dokumentiert
\$ 38 & 39 nur MP/M, unter CP/M 'Return NULL'
41—50, 59—60, 98—112 & 152 nur CP/M-3.x

Tabelle 1. System-Funktionen

Nummer	CP/M-1.4	CP/M-2.x	CP/M-3.x
xx00	Coldstart	dito	dito
xx03	Warmstart	dito	dito
xx06	Console Status	dito	dito
xx09	Console Input	dito	dito
xx0C	Console Output	dito	dito
xx0F	List Output	dito	dito
xx'2	Reader Input	dito	Aux. Input
xx'5	Punch Output	dito	Aux. Output
xx'8	Home Disk Head	dito	don't use!
xx'B	select Disk	dito	don't use!
xx'E	select Track	dito	don't use!
xx21	select Sektor	dito	don't use!
xx24	Read 128 Bytes	dito	read phys. Sektor!
xx27	Write 128 Bytes	dito	write phys. Sektor!
xx30	nicht vorhanden	List Status	dito
xx33	nicht vorhanden	Sektor Translate	don't use!

Weitere Einträge nur unter CP/M-3.x definiert.

Tabelle 2. BIOS-Sprungleiste

Name	Bytes p. Sektor	Sektor p. Track	Tracks p. Disk	System-tracks	Kapazität	Sektor-Skew	Density	Block-größe	Directory-Entries
TRS-80 12	128	18	35	2	74	5	SD	1	64
Cromenco	128	18	35	2	74	4	SD	1	64
Xerox 820 SD	128	18	40	3	83	5	SD	1	12
Osborn SD	256	10	40	3	95	4	SD	2	34
Alphatronik PS	256	16	40	2	151	1	DC	1	34
IBM (CP/M-85)	512	8	40	1	156	1	DC	1	34
Siemens	512	10	40	2	90	4	DC Inv.	2	34
Osborn DD	1024	5	40	3	188	1	DC	1	34

Tabelle 3. Einige typische Diskettenformate

org CE276h
Sektor-Translate Tabelle
entspricht Skew = 1

db 1,2,3,4,5,6,7,8,9
db 10,11,12,13,14,15,16,17,18,19
db 20,21,22,23,24,25,26,27,28,29
db 30,31,32

org CE2EEh

db 4 : Drive-Code: 0000BE0D
: D = 1 für SD, 0 für DD
: BB = 00 für 128, 01 für 256,
: 10 für 512, 11 für 1024 Bytes/Sektor

nun DPB:

dw 32 : log. Sektoren/Spur
db 3 : Sektor-Shift
db 7 : Sektormask
db 0 : Extent-Mask
dw 50 : Kapazität (-1)
dw 0C00h : alloc.
dw 00h : Check-Size
dw 2 : res. Tracks

Tabelle 4. Lage der freien Disk-Parameter im OSBCRN-Bios mit Beispiel für Alphatronik P2

⊙ ⊙ DAS UMFASSENDE PROGRAMM ⊙ ⊙

INTERFACES UND PUFFERSPEICHER FÜR DRUCKER				
WIESEMANN				
COMPUTER-SCHNITTSTELLE				
DRUCKER-SCHNITTSTELLE	CEM HP-IB IEEE488	CDM04 VC20	Centronics	R3232C V24
	BUFFER 8-120K 33□□□□	Lieferant: Data Becker Düsseldorf	Interface OK (Jan. 84)	Interface OK (Nov. 83)
	Interface OK (Dez. 83)	BUFFER 8-120K (Jan. 84)	Interface OK	Interface OK (Jan. 84)
	33900	99□□□□	2900	8900
Centronics	Interfaces C-120K	Interface OK	BUFFER 8-120K	Interfaces OK-32K
	32□□□□	9200	22□□□□	82□□
R3232C V24	Interfaces J-32K	Lieferant: Data Becker Düsseldorf	Interfaces J-32K	BUFFER 8-120K
	38□□		28□□	88□□□□

ERLÄUTERUNG: Z.B.: 32□□□□ = ARTIKELNUMMER
□□□□ = BUFFERKAPAZITÄT

REINHARD WIESEMANN
WINCHENBACHSTR. 3A
5600 WUPPERTAL 2

MIKROCOMPUTERTECHNIK
POSTFACH 201605
TEL.: 0202/510444

CEPAC-65 Version A

Komplett-Bausatz o. EPROM

o. Steckerleiste DM 69,— (fertig DM 85,—)

c't-86 16-bit-Selbstbaucomputer

Platine 1: CPU-Karte mit 80386, 8 KB Monitorprogramm

Komplett-Bausatz ... DM 349,— (fertig DM 449,—)

Platine 2: I/O-Karte mit V-24-, Centronics-, Kassettenrecorder-Interface und Timer

Komplett-Bausatz ... DM 249,— (fertig DM 349,—)

Platine 3: Floppy-Controller-Karte bis zu 4 Laufwerke 5¼ oder 8 Zoll (auch gemischt), IC WD 2797

Komplett-Bausatz ... DM 498,— (fertig DM 598,—)

Platine 4: 256 KB-RAM-Karte (128 K bestückt)

Komplett-Bausatz ... DM 598,— (fertig DM 698,—)

Busplatine (96polig)

10 Steckplätze DM 49,—

fertig bestückt: DM 169,—

c't-Terminal

Version A Bausatz DM 449,— (fertig DM 549,—)

Version B Bausatz DM 498,— (fertig DM 639,—)

Versand per Nachnahme und Versandkosten.
Liste C3 anfordern.

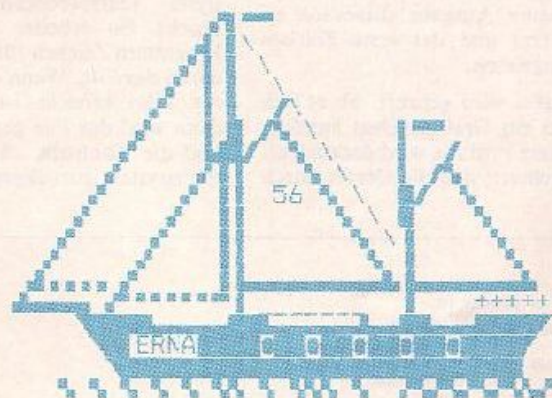
Frölje Elektronik

Gaststr. 1C ★ 2900 Oldenburg ★ Tel. (0441) 15853

Eberhard Hewicker



Der OSBORNE 1 bietet zwar einen Blockgrafik-Satz von 32 Zeichen, aber den kann man nur mit Schwierigkeiten ausnutzen. Es wird entweder ein Grafik-Editor benötigt (ist im Handel erhältlich), oder man muß umständliche Assembler-Programme schreiben, um vor jeder Cursor-Bewegung den Grafik-Mode zu verlassen und anschließend wieder einzuschalten. Und das Assembler-Programmieren ist nun mal nicht jedermanns Sache.



WordStar als Grafik-Editor

Blockgrafik mit OSBORNE 1 und Epson MX 80

Die Anschaffung eines Grafik-Editors läßt sich elegant umgehen, indem man den mitgelieferten Editor, nämlich WordStar, benutzt. Die Idee ist einfach, die Wirkung verblüffend. Nach dem Laden von WordStar startet man mit dem Kommando 'R' (Run a Program) ein kleines Assembler-Programm, das die Umschaltung in den Grafik-Modus bewirkt. Dann geht man wieder zurück ins WordStar (durch Drücken einer beliebigen Taste). Die System-Meldungen sind jetzt teilweise durch Grafik-Zeichen entsetzt, dadurch braucht man sich aber nicht beeindrucken zu lassen.

Mit dem Kommando 'N' wird ein non-document-file eröffnet. Auf dem Bildschirm werden jetzt alle Kleinbuchstaben als Grafik-Symbole dargestellt. Die vollen Editier-Möglichkeiten des WordStar stehen unein-

geschränkt zur Verfügung. Man kann also z. B. Zeilen löschen oder einfügen, Blöcke umstellen, das File speichern und wieder laden usw. Für die Darstellung von Schrift ist man jetzt allerdings auf die Großbuchstaben angewiesen, denn die Kleinbuchstaben erzeugen ja die Grafik.

Nach Herzenslust kann man nur den Bildschirm vollmalen. Die Tab-Funktion sollte man allerdings vermeiden, denn das bringt den Drucker durcheinander, der schließlich das 'Gemälde' zu Papier bringen soll. In diesem Punkt ist das Druckprogramm noch verbesserungsfähig. Eine Überraschung wird man erleben, wenn man einen Block markieren will: statt halber Helligkeit erscheinen plötzlich wieder Kleinbuchstaben anstelle der Grafik. Durch ^KH kann aber die Grafik wiederhergestellt werden.

Am Ende der Sitzung wird das Grafik-File ganz normal abgespeichert (^KD), wobei die Floppy die Zeichen natürlich nach ASCII interpretiert und speichert. Das File kann erneut geladen und editiert werden wie jedes andere WordStar-File auch. Der Grafik-Modus wird durch 'Ausstieg zum System' (^X) wieder verlassen. Ein Wiedereinschalten ist auf die oben beschriebene Art jederzeit möglich.

Nun bleibt noch das Problem, die Grafik auch mit einem MX 80 zu Papier zu bringen, der ja von Hause aus keine Block-Grafik kann.

Die Lösung ist ein Pascal-Programm, welches das Grafik-File von einer Diskette einliest, für jedes Zeichen den entsprechenden Grafik-Code erzeugt und diesen dann an den Drucker ausgibt. Zu Beginn des Hauptprogramms wird der Zei-

lenverschiebung so eingestellt, daß sich die Grafik-Blöcke nahtlos aneinanderfügen. Dies ist bei der Mischung von Grafik und Text zu beachten: Auf dem Bildschirm muß zwischen zwei Textzeilen immer eine Leerzeile eingefügt werden, sonst verfließen die Schriftzeichen beim Ausdrucken ineinander.

Da ein Drucker ohnehin nicht zu den schnellen Peripheriegeräten gehört, wird der Druckvorgang durch das Pascal-Programm in keiner Weise verzögert. Natürlich könnte solch ein Druckprogramm auch in jeder anderen Programmiersprache geschrieben werden. Pascal erschien mir jedoch schnell, kompakt und gleichzeitig leicht verständlich. Hier wird der Compiler HiSoft HD verwendet.

Mit einem ähnlichen Programm lassen sich übrigens auch griechische Buchstaben

und mathematische Sonderzeichen gemischt mit Text ausdrucken.

Der größte Teil des Programmes enthält die Definition der Grafik-Symbole, die als 'Procedures' vereinbart sind. Jede Procedure sendet 12 ASCII-Zeichen an den Drucker, deren jedes eine Spalte des Grafik-Blockes definiert. Das kurze Hauptprogramm (ein Sechsz Zeiler) erfragt zunächst den Namen des zu druckenden Files. Dann wird durch die Procedure 'Ausgabe' dieses File eröffnet und das erste Zeichen eingelesen.

Dabei wird geprüft, ob es sich um ein Grafikzeichen handelt. Diese Prüfung wird dadurch erleichtert, daß die Grafik durch

die obersten 32 ASCII-Codes repräsentiert wird (96—127). Die Abfrage ist also einfach: 'ASCII<96?'. Wenn ja, dann wird das Zeichen unverändert an den Drucker geschickt. Andernfalls wird der Drucker in den Grafik-Modus geschaltet und dann über eine Fallunterscheidung (CASE...) die entsprechende Procedure aufgerufen.

Das Zurückschalten in den Text-Modus erfolgt im Drucker automatisch nach 12 Grafik-Bytes (entsprechend einem Block). So arbeitet sich das Programm Zeichen für Zeichen durch das File. Wenn das Ende des Files erreicht ist (EOF), kann wird das File geschlossen und die Kontrolle an das Betriebssystem zurückgegeben. □

```

WRITE (CHR (15),CHR (15),CHR (15),CHR (15),CHR (15));END;
PROCEDURE IV; BEGIN
WRITE (CHR (255),CHR (255),CHR (255),CHR (255),CHR (255),CHR (255));
WRITE (CHR (255),CHR (255),CHR (255),CHR (255),CHR (255));END;
PROCEDURE IY; BEGIN
WRITE (CHR (254),CHR (254),CHR (252),CHR (252),CHR (246),CHR (246));
WRITE (CHR (224),CHR (192),CHR (192),CHR (128),CHR (128));END;
PROCEDURE IT; BEGIN
WRITE (CHR (0),CHR (0),CHR (0),CHR (0),CHR (0),CHR (0));
WRITE (CHR (240),CHR (240),CHR (240),CHR (240),CHR (240));END;
PROCEDURE IX; BEGIN
WRITE (CHR (255),CHR (255),CHR (253),CHR (253),CHR (253),CHR (15));
WRITE (CHR (15),CHR (15),CHR (15),CHR (15),CHR (15));END;
PROCEDURE IJLA; BEGIN
WRITE (CHR (60),CHR (60),CHR (60),CHR (60),CHR (60),CHR (60));
WRITE (CHR (60),CHR (60),CHR (60),CHR (60),CHR (60));END;
PROCEDURE IKA; BEGIN
WRITE (CHR (0),CHR (0),CHR (0),CHR (252),CHR (252),CHR (252));
WRITE (CHR (252),CHR (252),CHR (60),CHR (60),CHR (60));END;
PROCEDURE ILLDB; BEGIN
WRITE (CHR (60),CHR (60),CHR (60),CHR (253),CHR (252),CHR (252));
WRITE (CHR (252),CHR (252),CHR (60),CHR (60),CHR (60));END;
PROCEDURE IMAFK; BEGIN
WRITE (CHR (77),CHR (12),CHR (128));END; meldet dem X80 12 Grafikwerte;
PROCEDURE IX; BEGIN
WRITE (CHR (0),CHR (0),CHR (0),CHR (255),CHR (255),CHR (255));
WRITE (CHR (255),CHR (255),CHR (0),CHR (0),CHR (0));END;
PROCEDURE IC; BEGIN
WRITE (CHR (15),CHR (15),CHR (15),CHR (15),CHR (15),CHR (15));
WRITE (CHR (255),CHR (255),CHR (255),CHR (255),CHR (255));END;
PROCEDURE IKLZ; BEGIN
WRITE (CHR (128),CHR (192),CHR (224),CHR (240),CHR (240),CHR (124));
WRITE (CHR (15),CHR (15),CHR (77),CHR (0),CHR (0),CHR (0));END;

```

```
PROCEDURE hsbeger;
BEGIN
  RESET(Data,Name);PRIN;   (File eroeffnen, Drucker ein)
  WRITE(CHR(27),CHR(69),CHR(25)); (Zeilenbesatz=25/20 inch)
  WHILE NOT EOF(Data) DO
    BEGIN
      GET(Data);
      IF EOLN(Data) THEN WRITELN (Wenn Zeilenende, dann 1 Zeile aus-schreiben)
      ELSE BEGIN
        IF (Data<'<a') THEN WRITELN(Data);(Normalzeichen, zum Drucker)
          (Sonderzeichen, Procedure suchen)
        CASE Data OF
          'a':BEGIN GRAFIK:a END; 'b':BEGIN GRAFIK:b END;
          'c':BEGIN GRAFIK:c END; 'd':BEGIN GRAFIK:d END;
          'e':BEGIN GRAFIK:e END; 'f':BEGIN GRAFIK:f END;
          'g':BEGIN GRAFIK:g END; 'h':BEGIN GRAFIK:h END;
          'i':BEGIN GRAFIK:i END; 'j':BEGIN GRAFIK:j END;
          'k':BEGIN GRAFIK:k END; 'l':BEGIN GRAFIK:l END;
          'm':BEGIN GRAFIK:m END; 'n':BEGIN GRAFIK:n END;
          'o':BEGIN GRAFIK:o END; 'p':BEGIN GRAFIK:p END;
          'q':BEGIN GRAFIK:q END; 'r':BEGIN GRAFIK:r END;
          's':BEGIN GRAFIK:s END; 't':BEGIN GRAFIK:t END;
          'u':BEGIN GRAFIK:u END; 'v':BEGIN GRAFIK:v END;
          'w':BEGIN GRAFIK:w END; 'x':BEGIN GRAFIK:x END;
          'y':BEGIN GRAFIK:y END; 'z':BEGIN GRAFIK:z END;
          'A':BEGIN GRAFIK:A END; 'B':BEGIN GRAFIK:B END;
          'C':BEGIN GRAFIK:C END; 'D':BEGIN GRAFIK:D END;
          'E':BEGIN GRAFIK:E END; 'F':BEGIN GRAFIK:F END;
          'G':BEGIN GRAFIK:G END; 'H':BEGIN GRAFIK:H END;
          'I':BEGIN GRAFIK:I END; 'J':BEGIN GRAFIK:J END;
          'K':BEGIN GRAFIK:K END; 'L':BEGIN GRAFIK:L END;
          'M':BEGIN GRAFIK:M END; 'N':BEGIN GRAFIK:N END;
          'O':BEGIN GRAFIK:O END; 'P':BEGIN GRAFIK:P END;
          'Q':BEGIN GRAFIK:Q END; 'R':BEGIN GRAFIK:R END;
          'S':BEGIN GRAFIK:S END; 'T':BEGIN GRAFIK:T END;
          'U':BEGIN GRAFIK:U END; 'V':BEGIN GRAFIK:V END;
          'W':BEGIN GRAFIK:W END; 'X':BEGIN GRAFIK:X END;
          'Y':BEGIN GRAFIK:Y END; 'Z':BEGIN GRAFIK:Z END;
          ' ':END;
          '\n':END;
          '\t':END;
          '\f':END;
        END;
      END;
    END;
  END;
END;
```

```

BEGIN      (Beginn des Hauptprogrammes)
WRITELN   WRITE('Eingabe des Dateinamens: ');

READLN;

READ(Name); (Lies Namen des zu druckenden Files, Name muss 14 Zeichen)

Ausgabe:  (lang sein, gegebenenfalls durch SPACES auffüllen!)
          (z.B.: A:TEST .PAS oder: TEST .PAS)

END.

```

```
type B GRAF:IK.ASM
      escape equ 1FH ;escape character
      graph equ 'g' ;graphics
      sbuff equ 5 ;EDDS print buffer
      bds equ 4 ;EDDS entry

      org 0100H
      LXI D,STARGR
      CALL PRINT
      RET

PRINT: MVI C,FBUFF
      JMP RDS

STARGR: DE escape,graph,'1'

      END
```

type GRAPHIK,PAS

```
PROGRAM Graph1; {Fuer Block-Graph1 mit EPSON MX80}
      {Eberhard Hewicker, Koblenz, 1983}
VAR Name :ARRAY[1..14] OF CHAR;
    Data :FILE OF CHAR;
```

```

PROCEDURE K1: BEGIN
  WRITE(CHR(255), CHR(255), CHR(255), CHR(255), CHR(255), CHR(255), CHR(0));
  WRITE(CHR(0), CHR(0), CHR(0), CHR(0), CHR(0), CHR(0)); END;
PROCEDURE K2: BEGIN
  WRITE(CHR(15), CHR(15), CHR(15), CHR(15), CHR(15), CHR(15));
  WRITE(CHR(240), CHR(240), CHR(240), CHR(240), CHR(240), CHR(240)); END;
PROCEDURE K3: BEGIN
  WRITE(CHR(0), CHR(0), CHR(3), CHR(0), CHR(0), CHR(0), CHR(0), CHR(15));
  WRITE(CHR(15), CHR(15), CHR(15), CHR(15), CHR(15), CHR(15)); END;
PROCEDURE K4: BEGIN
  WRITE(CHR(0), CHR(0), CHR(0), CHR(0), CHR(0), CHR(0), CHR(255));
  WRITE(CHR(255), CHR(255), CHR(255), CHR(255), CHR(255)); END;
PROCEDURE K5: BEGIN
  WRITE(CHR(0), CHR(0), CHR(3), CHR(63), CHR(63), CHR(63), CHR(63), CHR(63));
  WRITE(CHR(63), CHR(63), CHR(63), CHR(63), CHR(63), CHR(63)); END;
PROCEDURE K6: BEGIN
  WRITE(CHR(240), CHR(240), CHR(240), CHR(240), CHR(240), CHR(240), CHR(255));
  WRITE(CHR(255), CHR(255), CHR(255), CHR(255), CHR(255)); END;
PROCEDURE K7: BEGIN
  WRITE(CHR(15), CHR(15), CHR(15), CHR(15), CHR(15), CHR(15), CHR(0));
  WRITE(CHR(0), CHR(0), CHR(0), CHR(0), CHR(0), CHR(0)); END;
PROCEDURE K8: BEGIN
  WRITE(CHR(1), CHR(3), CHR(3), CHR(3), CHR(1), CHR(15), CHR(15));
  WRITE(CHR(3), CHR(63), CHR(63), CHR(127), CHR(255)); END;
PROCEDURE K9: BEGIN
  WRITE(CHR(255), CHR(127), CHR(63), CHR(63), CHR(3), CHR(15), CHR(15));
  WRITE(CHR(7), CHR(3), CHR(3), CHR(3), CHR(1)); END;
PROCEDURE K10: BEGIN
  WRITE(CHR(60), CHR(60), CHR(60), CHR(255), CHR(255), CHR(255), CHR(255));
  WRITE(CHR(255), CHR(255), CHR(60), CHR(60), CHR(60), CHR(60)); END;
PROCEDURE K11: BEGIN
  WRITE(CHR(255), CHR(255), CHR(255), CHR(255), CHR(255), CHR(255), CHR(240));
  WRITE(CHR(240), CHR(240), CHR(240), CHR(240), CHR(240)); END;
PROCEDURE K12: BEGIN
  WRITE(CHR(0), CHR(0), CHR(60), CHR(126), CHR(126), CHR(126), CHR(126), CHR(126), CHR(126), CHR(126), CHR(126));
  WRITE(CHR(126), CHR(60), CHR(16), CHR(0)); END;
PROCEDURE K13: BEGIN
  WRITE(CHR(240), CHR(240), CHR(240), CHR(240), CHR(240), CHR(240), CHR(15));
  WRITE(CHR(15), CHR(15), CHR(15), CHR(15), CHR(15)); END;
PROCEDURE K14: BEGIN
  WRITE(CHR(160), CHR(160), CHR(160), CHR(63), CHR(63), CHR(63), CHR(63));
  WRITE(CHR(60), CHR(60), CHR(60), CHR(60)); END;
PROCEDURE K15: BEGIN
  WRITE(CHR(124), CHR(124), CHR(124), CHR(124), CHR(124), CHR(240), CHR(240));
  WRITE(CHR(240), CHR(252), CHR(252), CHR(254), CHR(255)); END;
PROCEDURE K16: BEGIN
  WRITE(CHR(60), CHR(60), CHR(60), CHR(63), CHR(63), CHR(63), CHR(63));
  WRITE(CHR(63), CHR(63), CHR(0), CHR(0), CHR(0)); END;
PROCEDURE K17: BEGIN
  WRITE(CHR(240), CHR(240), CHR(240), CHR(240), CHR(240), CHR(240));
  WRITE(CHR(240), CHR(240), CHR(240), CHR(240), CHR(240), CHR(240)); END;
PROCEDURE K18: BEGIN
  WRITE(CHR(240), CHR(240), CHR(240), CHR(240), CHR(240), CHR(0), CHR(0), CHR(0), CHR(0), CHR(0));
  WRITE(CHR(0), CHR(0), CHR(0), CHR(0)); END;
PROCEDURE K19: BEGIN
  WRITE(CHR(1), CHR(3), CHR(7), CHR(5), CHR(3), CHR(62), CHR(124), CHR(240));
  WRITE(CHR(240), CHR(240), CHR(192), CHR(192), CHR(192)); END;
PROCEDURE K20: BEGIN
  WRITE(CHR(15), CHR(15), CHR(15), CHR(15), CHR(15), CHR(15), CHR(15));

```

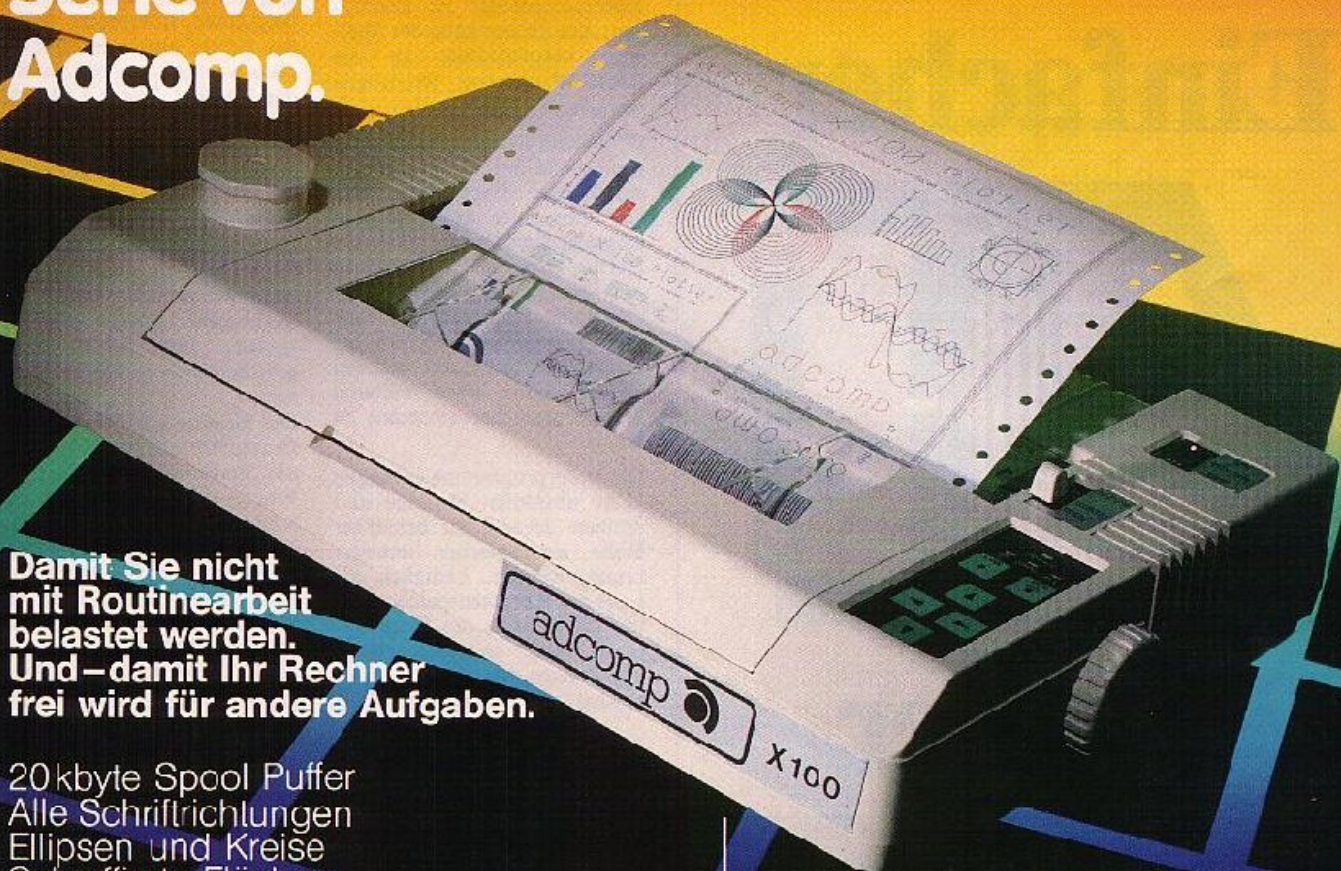

**Farbig
plotten und
drucken
mit der
X100
Serie von
Adcomp.**

**Der X100S
Superintelligenz
und Präzision**

**Damit Sie nicht
mit Routinearbeit
belastet werden.
Und – damit Ihr Rechner
frei wird für andere Aufgaben.**

20kbyte Spool Puffer
Alle Schriftrichtungen
Ellipsen und Kreise
Schraffierte Flächen
Logarithmische Achsen
Kubische Interpolation für weiche Kurven
Barcode und Hardcopy
Schnittstellen nach Wahl:
RS232C (V24) mit Prozedur
XON XOFF auch für DEC.
IEEE488 für HP und CBM
Centronics parallel

Modernste Technik:
2 Mikroprozessoren für Zeitoptimierung
Plotten mit 12cm/sek.
Drucken mit 8 Z/sek.
Format DIN A4 Endlos und Einzelblatt
4 feine Farbminen im Revolver
Präziser und schneller Farbwechsel
Plottaufösung max. 0,05mm



Plotter-Printer X100S

Die Exklusiv-Version X100S
kostet DM 4.104,- incl. MWSt.

Andere Versionen der X100 Serie sind
ab DM 2.394,- incl. MWSt. zu haben.

adcomp 

adcomp Datensysteme GmbH
Olgastraße 15, 8000 München 19
Tel. 089/1298045, Telex 52'6271

Hannover-Messe
Halle 4 · Stand 1302/I

Soll ein Computer Vorgänge aus der 'analogen Welt' erfassen, überwachen oder regeln, so müssen diese in die Digitalform übersetzt werden. In diesem Praxistip stellen wir einen einfachen und sehr preisgünstigen Analog/Digital-Wandler vor, der sich zur genauen Umsetzung von Spannungswerten eignet. Ein solcher Wandler ist vor allem für die Erfassung von Meßdaten über die verschiedensten Meßwertaufnehmer nützlich. Es lassen sich beispielsweise Temperaturfühler, Drucksensoren oder fotoelektrische Aufnehmer anschließen. Assemblerprogramme für Z 80 und 6502 ermöglichen die Übergabe der Meßdaten an den Computer.

Wolfgang Schrader

Einfacher

Wandler

CA 3162 in Z 80- und 6502-Systemen

Als Wandler-Baustein wird der CA 3162 verwendet, ein IC, das man häufig in preisgünstigen Digital-Meßgeräten vorfindet. Der Baustein ist so konzipiert, daß man mit einem nachgeschalteten Dekoder/Anzeigetreiber und wenigen externen Bauelementen ein einfaches Voltmeter aufbauen kann. Der Wandler ist zwar nicht speziell für die Zusammenarbeit mit Mikrocomputersystemen gedacht, doch lassen seine Vorzüge (nicht zuletzt der geringe Preis von rund 5 Mark) auch diese Verwendung sinnvoll erscheinen.

Mit dem CA 3162 wird eine Auflösung von 1098 Punkten (+999 mV...-99 mV) erreicht, also besser als 2^{10} Bit. Der Wandler arbeitet im 'Dual-Slope'-Verfahren und führt wahlweise vier oder 96 Messungen pro Sekunde aus.

Zum Anschluß an den Mikrocomputer werden die Steuerleitungen für die drei Ziffern (Digits) der Anzeige zweckentfremdet. Sie zeigen an, zu welchem Zeitpunkt an den vier Datenleitungen der Wert für die Hunderter-, Zehner- und Einerstelle im BCD-Code ausge-

geben wird. Die entsprechende Steuerleitung führt dann log. 0. Zum Anschluß an einen Computer werden also lediglich sieben Datenleitungen benötigt.

Der Wandler gibt den Meßwert in der Reihenfolge Hunderter, Einer, Zehner in ununterbrochener Folge aus. Die abgedruckten Assemblerprogramme für Z 80 (8080) und 6502 arbeiten beide nach demselben Prinzip: Der Computer wartet auf das Aktivieren einer Steuerleitung, um dann den Wert für die jeweilige Stelle in einen Zwischenspeicher zu übertragen. Aus diesem kann der Meßwert beispielsweise an ein BASIC-Programm übertragen werden.

Bei beiden Programmvorlägen wurde der Elcganz wegen ein kleiner Trick verwendet: Der Computer überträgt den eingelesenen Wert schon in den Puffer, bevor er die 'zuständige' Steuerleitung abfragt. So geht man sicher, das Meßwert und Steuerleitung exakt zu demselben Zeitpunkt gelesen werden, ohne einen weiteren Zwischenspeicher einsetzen zu müssen.

Die Unterprogramme enthalten keine absoluten Sprünge und können deshalb an beliebiger Stelle im Speicher untergebracht werden. Lediglich die Lage des Zwischenspeichers sowie die Adresse des Eingangsports sind rechner-spezifisch und müssen den Gegebenheiten angepaßt werden. Als Port

wird in unserem Schaltungsvorschlag (Bild 1) ein 8-Bit-Pufferbaustein (IC 1) verwendet. Wenn Ihr Computer über einen User-Port verfügt, können Sie auf den Puffer und den Adreßdeko-der IC 3 verzichten und den Wandler direkt anschließen.

Natürlich könnte man theoretisch auch ein entsprechendes BASIC-Programm zur Abfrage eines Eingangsports verwenden. Allerdings wird das bei schnell wechselnden Meßwerten gelegentlich zu Ablesfehlern führen. So kann bei einem langsamen Interpreter folgender Effekt auftreten: Die Anzeige wechselt von 600 auf 599; der Rechner liest die Hunderterstelle (6), doch noch bevor die Zehner- und Einerstellen gelesen werden, gibt der Wandler den nächsten Wert aus. Als Zehnerstelle wird nun die '9' eingelesen. So entstehen Meßfehler, die vor allem bei grafischer Ausgabe der Daten einen unschönen 'Ausreißer' darstellen. Ein Maschinenprogramm eignet sich deshalb besser.

Die Routine schreibt bei jedem Durchlauf alle drei Werte in der Reihenfolge Hunderter — Einer — Zehner in einen Zwischenspeicher. Auf diese Weise ist eine problemlose Zusammenarbeit zum Beispiel mit einem BASIC-Programm gewährleistet, das die Werte zur weiteren Bearbeitung nur noch durch PEEK auslesen muß.

Die Schaltung kann ohne

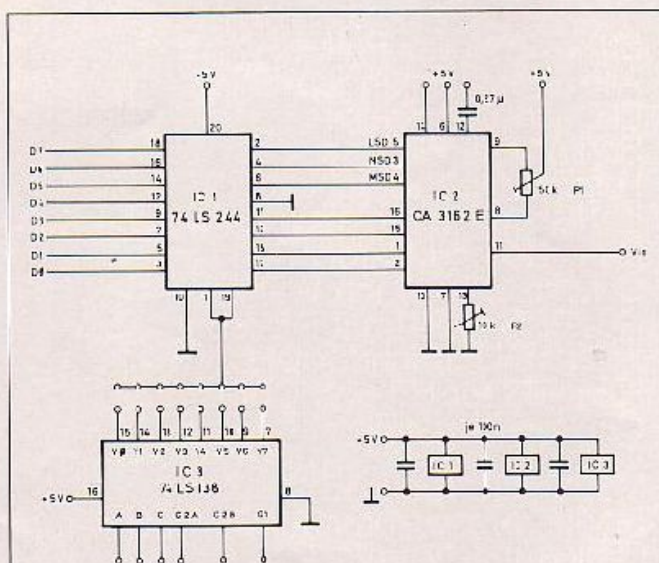


Bild 1. Pufferbaustein und Dekoder werden nur dann benötigt, wenn kein freier User-Port zum Anschluß des Wandler-ICs vorhanden ist.

Schwierigkeiten auf einer kleinen Lochrasterplatine aufgebaut werden. Der Abgleich des CA3162 erfolgt mit den beiden Trimpotis. Der Nullpunkt wird mit P1 eingestellt. Zum weiteren Abgleich legt man eine Spannung bekannter Höhe (möglichst 900 mV) an und stellt P2 so ein, daß dieser Meßwert ausgegeben wird. Der Regelbereich liegt bei ± 50 mV.

An Pin 6 des Wandlers kann die 'Conversion Rate' eingestellt werden. Legt man Pin 6 direkt an +5V, führt der Wandler 96 Messungen in der Sekunde aus; legt man den Anschluß auf Masse, sind es vier Messungen. Wird Pin 6 über einen 12-K-Widerstand mit +5V verbunden, so hält der Wandler den letzten Meßwert.

Die Eingangsimpedanz beträgt 100 MOhm. Über entsprechende Spannungsteiler können beliebige Meßbereiche eingestellt werden. Bei Überlauf in positiver Richtung ($> +999$ mV) gibt der Wandler eine '11' (0Bh) an jeder Stelle aus, bei Überschreitung des Meßbereichs in negativer Richtung (< -99 mV) eine '10' (0Ah). Ein negativer Meßwert wird außerdem durch Ausgabe einer '10' in der Hundertersstelle angezeigt.

Noch ein Wort zur Adreßdekodierung bei Verwendung des Pufferbausteins und des Adreßdekoders: Der Puffer

		;***** ;*** A/D-WANDLER SERVICE-ROUTINE ;*****	
		;Z80 ORG 100H	
0100'	F3	PUSH AF	;REGISTER RETTEN
		;A/D-WANDLER SIGNALISIERT VORHANDENSEIN DER JEWEILIGEN ;BCD-STELLE DURCH LOW-PEL DER ZUGEHÖRIGEN SIGNALLEITUNG	
0101'	D3 FF	HUND: IN A,(OFFH)	;HUNDERTERSTELLE EINLESEN
0103'	E3 2F	AND 02FH	;WENN SIGNALLEITUNG (BIT 5) ;=0, DANN IST SETZT DAS ;OBERE NIBBLE GELOESCHT
0105'	32 0124'	LD (HUN),A	;HUNDERTERSTELLE SPEICHERN
0108'	E3 20	AND 020H	;WAR BIT 5 WIRKLICH 0?
010A'	23 F5	JR NZ,HUND	;WENN NICHT, NEUER VERSUCH
		;EINER: IN A,(OFFH)	
010C'	D3 FF	EINER: IN A,(OFFH)	;CA 3162 GIBT JETZT EINER- ;STELLE AUS
010E'	E3 8F	AND 08FH	;OBERE NIBBLE LOESCHEN
0110'	32 0126'	LD (EIN),A	;EINERSTELLE SPEICHERN
0113'	E3 80	AND 080H	;WAR BIT 7 WIRKLICH 0?
0115'	23 F5	JR NZ,EINER	;WENN NICHT, NEUER VERSUCH
		;ZEHNER: IN A,(OFFH)	
0117'	D3 FF	ZEHNER: IN A,(OFFH)	;ZEHNERSTELLE EINLESEN
0119'	E3 4F	AND 04FH	;OBERE NIBBLE LOESCHEN
011B'	32 0125'	LD (ZEHN),A	;ZEHNERSTELLE SPEICHERN
011E'	E3 40	AND 040H	;WAR BIT 6 WIRKLICH 0?
0120'	23 F5	JR NZ,ZEHNER	;WENN NICHT, NEUER VERSUCH
		;POP AF	
0122'	F1	POP AF	;REGISTER ZURUECK
0123'	C3	RET	
		;HUN: DEFB 00H	
0124'	03	HUN: DEFB 00H	;SPEICHERZELLEN SETZT
0125'	03	ZEHN: DEFB 00H	;IN DURCHGANGIGER
0126'	03	EIN: DEFB 00H	;REIHENFOLGE
		;END	

Tabelle 2. Abfrageprogramm für 8080 und Z80

Eingänge					Ausgänge							
Enable	Select											
G1	C2*	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	H	H	H	H	H	L	H	H	H	H
H	L	H	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	H	L	H	H
H	L	H	H	L	H	H	H	H	H	H	L	H
H	L	H	H	H	H	H	H	H	H	H	H	L

Positive Logik
* G2 = G2A + G2B

Tabelle 1. Wahrheitstabelle des Adreßdekoders 74LS138

muß bei einem Lesezugriff mit einem Selektionssignal (log. 0) aktiviert werden. Mit dem Dekoder 74LS138 läßt sich zwar keine vollständige, aber eine für viele Anwendungsfälle ausreichende Dekodierung erreichen. Tabelle 1 ist die Wahrheitstabelle des Dekoders, an-

hand derer Sie den Anschluß vornehmen sollten. Zu beachten ist, daß der Dekoder zwei Eingänge 'G2' besitzt, die miteinander 'ODER-verknüpft' sind. Beim Einsatz in einem Z80-System sollte man hier die Signale \overline{IORQ} und \overline{RD} anlegen.

		;***** ;*** A/D-WANDLER SERVICE-ROUTINE FÜR 6502 ;*****	
		;MEM=?	
		;DREI SPEICHERSTELLEN FÜR EINGELESENE ;WERTE (HUNDERTER, ZEHNTER, EINER)	
		;FAD=?	
		;EINGANGSPORT	
		;PCGRA/MAN/MSB-ADRESSE	
		;LDY #00	
		;Y-REGISTER INITIALISIEREN	
		;HUND: LDA PAD	
		;PORTREGISTER LESEN	
		;AND #0001111	
		;WENN SIGNALLEITUNG (HUND)	
		;=0, DANN IST BIT 5 GELOESCHT	
		;STA MEM,Y	
		;HUNDERTER-WERT ABSPEICHERN	
		;AND #00010000	
		;WAR BIT 5 WIRKLICH 0?	
		;BNE HUND	
		;WENN NICHT, NEUER VERSUCH	
		;LDY #02	
		;INDEX FÜR EINERSTELLE	
		;EINER: LDA PAD	
		;CA 3162 GIBT JETZT	
		;EINERSTELLE AUS	
		;AND #0001111	
		;OBERE NIBBLE LOESCHEN	
		;STA MEM,Y	
		;EINER-WERT ABSPEICHERN	
		;AND #00010000	
		;WAR BIT 7 WIRKLICH 0?	
		;BNE EINER	
		;WENN NICHT, NEUER VERSUCH	
		;LDY	
		;INDEX FÜR ZEHNTERSTELLE	
		;ZEHNER: LDA PAD	
		;ZEHNTERSTELLE EINLESEN	
		;AND #0001111	
		;OBERE NIBBLE LOESCHEN	
		;STA MEM,Y	
		;ZEHNTERSTELLE SPEICHERN	
		;AND #00010000	
		;WAR BIT 6 WIRKLICH 0?	
		;BNE ZEHNER	
		;WENN NICHT, NEUER VERSUCH	
		;RTS	

Tabelle 3. Abfrageprogramm für 6502

Die NS16000-Familie besteht aus mehreren unterschiedlich leistungsfähigen CPUs und einer Auswahl von Slaveprozessoren sowie Peripheriebausteinen (Bild 1).

Prozessoren

Bisher sind drei verschiedene CPUs erhältlich. Das Besondere daran ist, daß alle drei eine völlig identische interne Architektur haben, das heißt jeder Prozessor hat beispielsweise interne 32-Bit-Datenwege, 32-Bit-Architektur, die gleiche Anzahl von Registern und die gleiche Speicherorganisation.

Das bedeutet, ein Programm, das für einen 'kleinen' NS16000 entwickelt wurde, läuft auch auf einem 'Großen'. Bei der schnell fortschreitenden Technologie und angesichts der hohen Softwarekosten ist das ein großer Vorteil.

Die CPUs unterscheiden sich eigentlich nur nach außen hin, nämlich in der Breite des externen Datenbus. Es existieren CPUs mit einem externen Datenbus von 8 Bit (NS08032), 16 Bit (NS16032) und 32 Bit (NS32032).

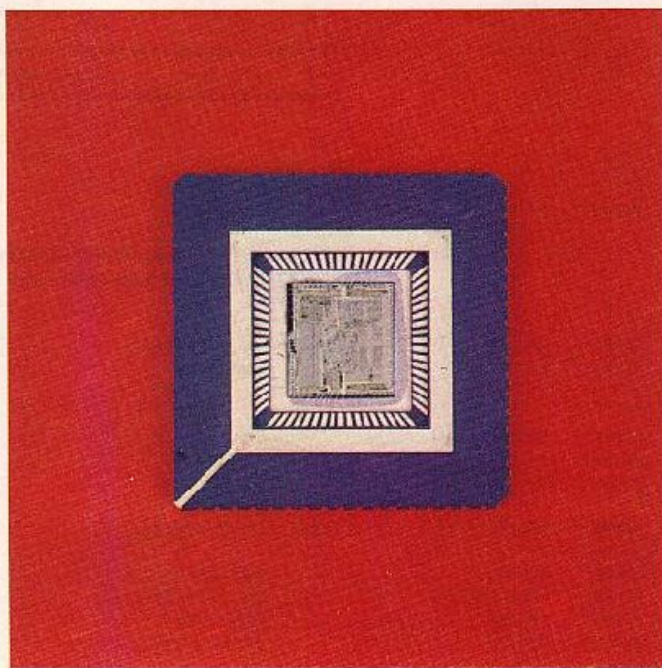
Neben den CPUs gibt es noch sogenannte Slaveprozessoren, die bei Bedarf ohne Softwareveränderung eingesetzt werden können, da sie für den Programmierer unbemerkt arbeiten.

Sie übernehmen Aufgaben wie zum Beispiel exakte und schnelle Fließkommaberechnungen und virtuelle Speicherverwaltung. Ein Begriff, der später etwas ausführlicher erklärt wird.

Mit der Floating Point Unit (FPU) NS16081 lassen sich schnelle Fließkommaberechnungen durchführen. Es kann eine Genauigkeit von 32 Bit oder 64 Bit gewählt werden.

Wird die CPU nicht mit dem Arithmetik-Slaveprozessor erweitert, können Fließkommaberechnungen auch softwaremäßig durchgeführt werden. Die Befehle zur Ausführung der Gleitkommaberechnungen bleiben gleich, deshalb muß die Anwender-Software nicht geändert werden.

Dies ist möglich mit Hilfe der Modularen Software, die ich in einem späteren Abschnitt noch genauer beschreibe.



Super-Chips

Hans-Joachim Gelke

Der NS32032 ist augenblicklich der leistungsfähigste Baustein der NS16000-Mikroprozessorfamilie von National Semiconductor. Diese Familie wurde für einen hohen Datendurchsatz entwickelt, wie er zum Beispiel in Minicomputern, CAD-Arbeitsplätzen, Prozellrechnern und für Telekommunikationsanwendungen benötigt wird. Man verfolgte strikt die Verwirklichung des Konzeptes, Eigenschaften von Großcomputern, wie Hochsprachenprogrammierung und virtuelles Speicherkonzept, für Mikroprozessoren zu übernehmen. Auf diese Weise entstand eine Mikroprozessoren-Familie einer völlig neuen Generation.

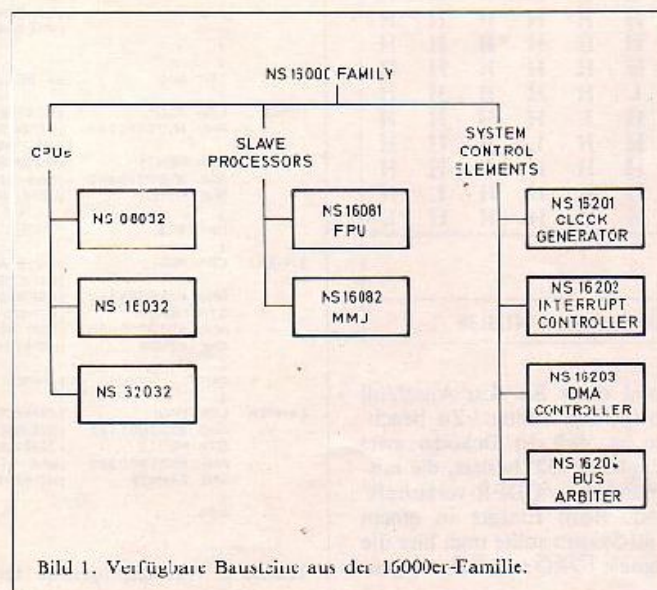


Bild 1. Verfügbare Bausteine aus der 16000er-Familie.

Peripheriebausteine

Die CPU NS32032 besitzt zwei Hardwareinterrupt-Eingänge, einen für maskierten und einen für nichtmaskierten Interrupt. Durch Anschluß einer oder mehrerer Interrupt-Steuerbausteine (Interrupt Control Unit), lassen sich diese Eingänge erweitern. Mit einer ICU sind bis zu 16 Interrupt-Eingänge verfügbar. Diese Eingänge sind mit bis zu 16 ICUs kaskadierbar. Somit erhält man bei Bedarf bis zu 256 Interrupt-Eingänge.

Das Taktsignal, bis zu 10 MHz, wird mit Hilfe eines externen Schaltkreises, genannt Timing-Control-Unit, kurz TCU, erzeugt. Mit ihm werden auch die Read- und Write-Zyklen gesteuert. Mit seiner Hilfe läßt sich der Bus-Zugriff bei Bedarf für eine vorprogrammierbare Zeit anhalten, zum Beispiel für langsame Speicher. In Entwicklung ist auch ein DMA-Controller für direkten Speicherzugriff.

Entworfen für Hochsprachen

Die Architektur beschreibt dem Anwenderkreis, der die Programme erstellt, wie der Computer aussieht. Sie beschreibt, welche und wie viele Register er hat, wie auf Speicherplätze zugegriffen werden kann, wie die Arithmetik organisiert ist, der Befehlssatz aufgebaut ist usw. Sie legt damit die Schnittstelle zwischen Software und Computer fest. Auf die interne Architektur wird etwas später genauer eingegangen.

Die Erfahrung von Programmherstellern hat gezeigt, daß sich Software für Mikroprozessoren in Hochsprachen (Pascal, Fortran, ADA) schneller erstellen und testen läßt als in Assemblersprache. Deshalb geht der Trend dahin, Programme zunächst mit Hilfe von Hochsprachen zu entwickeln und danach in die Maschinensprache des Prozessors zu übersetzen. Dieser Schritt erfolgt mit Hilfe von sogenannten Compilern.

Bei den gewöhnlichen Maschinensprachen und Prozessorarchitekturen entsteht dabei oft das Problem, daß sich ein Befehl nicht ohne weiteres übersetzen läßt. Man spricht dann, wie auch bei zwei Völkern mit unterschiedlichen Sprachen,

von einem Semantic Gap oder einer Übersetzungslücke. Um diese Lücken zu überbrücken, benötigt man meist lange und aufwendige Programmteile, die sich ungünstig auf die Rechengeschwindigkeit auswirken.

Aus diesen Gründen ist die Architektur der NS16000-Serie so ausgelegt, daß sie Hochsprachen wie zum Beispiel Pascal oder C entgegenkommt. Dadurch ergeben sich wesentlich kürzere Maschinencodes und damit schnellere Rechenzeiten. Die Compiler werden einfacher und kleiner.

Ein Programmbeispiel (Bild 2) zeigt die Verschiebung von 16 Datenbytes von einer Quelle (SOURCE) zu einem Ziel (DESTIN).

1. Im 8080-Assembler-Code
2. In der Programmiersprache Pascal
3. Im Assembler Code des NS32032

Datentypen

$A + B = B$
Operand 1 Operand 2 Ergebnis

Bei den Prozessoren der NS16000-Serie können für die Operanden unterschiedliche Bit-Längen gewählt werden. Man nennt dies Datentypen. Die beim NS32032 möglichen Datentypen zeigt Bild 3.

Bei Fließkommaberechnungen werden die Zahlen in Exponent und Mantisse aufgeteilt, zum Beispiel

$$0,0473 = 4,73 \times 10^{-2}$$

oder

$$3970,54 = 3,97054 \times 10^3$$

Adressierungsarten

Um den Inhalt einer Speicherzelle auslesen zu können, zum Beispiel um mit ihrem Inhalt und dem Inhalt einer anderen Speicherzelle eine Multiplikation durchzuführen, muß man zunächst einmal ihre Adresse bestimmen. Beim NS32032 gibt es hierfür neun Möglichkeiten. Die üblichen drei Methoden sind:

Direkte Adressierung:
Der Wert des Operanden steht direkt im Programm.

Absolute Adressierung:
Die Adresse der Speicherstelle, die als Operand dient, steht im Programm.

<pre> a) program MOVE; var SOURCE, DESTIN: array[1..16] of char; begin DESTIN := SOURCE end.</pre>		
③	TITLE MOVE	
LENGTH EQU 16	CSEG	;CODESEGMENT;
		;DEFINE LENGTH.
MOVE:	MVI HL,SOURCE	;LOAD HL WITH ADDRESS OF 'SOURCE'.
	MVI DE,DESTIN	;LOAD DE WITH ADDRESS OF 'DESTIN'.
	MVI BC,LENGTH	;LOAD BC WITH NUMBER OF BYTES.
LOOP:	MOV A,M	;MOVE BYTE FROM SOURCE TO DESTIN.
	STAX D	
	INX HL	;INCREMENT POINTER REGISTER HL.
	INX DE	;INCREMENT POINTER REGISTER DE.
	DCX BC	;DECREMENT COUNTER REGISTER BC.
	MOV A,B	;CONTINUE UNTIL BC = 0.
	ORA C	;BC = 0?
	JNZ LOOP	;NO, DO AGAIN.
	RET	;YES, END.
	DSEG	;DATA SEGMENT:
SOURCE: DS	LENGTH	;ALLOCATE 16 BYTES TO SOURCE.
DESTIN: DS	LENGTH	;ALLOCATE 16 BYTES TO DESTIN.
	END	
③	TITLE MOVE	
	STATIC	
SOURCE: BLKB 16		
DESTIN: BLFB 16		
	ENDSEG	
	MOVMB SOURCE,DESTIN,16	

Bild 2. Quellcodes für einen 16-Byte-Datentransfer im Vergleich.

- ③- Pascal-Programm
③- 8080-Assembler-Programm
③- NS16000-Assembler-Programm

Registeradressierung:

Der Operand steht in einem Register.

Daneben gibt es noch sechs weitere, speziell zur Unterstützung von Hochsprachen wichtige Adressierungsarten.

Register-relative Adressierung:

Sie entspricht im Grunde der Adressierung durch einen Registerwert, mit dem Unterschied, daß im Befehlswort noch ein Zusatzwert (oft als Offset oder Displacement bezeichnet) enthalten ist, der zum Inhalt des Registers dazugaddiert wird und so die Adresse ergibt (Bild 4).

Zeiger-relative Adressierung:

Wie register-relativ, nur werden jeweils ein Zeigerregister (Stackpointer u.ä.) und ein Zusatzwert zu einer Adresse zusammengestellt.

Speicher-relative Adressierung:

Hier ergibt sich die Operanden-

adresse aus einem Speicherinhalt und einem Zusatzwert.

'Top of Stack'-Adressierung:

Der Speicherplatz, auf dem sich der Operand befindet, wird durch den Stackpointer adressiert.

'External'-Adressierung:

Ist das Programm aus Modulen aufgebaut, kommt es vor, daß während der Abarbeitung eines Programms auf die Werte eines anderen Programms zugegriffen muß. Für diesen Zweck gibt es eine Adressierungsart, die sich 'External' nennt.

Skalierter Index:

Zusätzlich zu jeder Adressierungsart, außer zur direkten Adressierung, kann die skalierte Adressierung gewählt werden. Sie spielt bei der Verarbeitung von Bit-Feldern in der Programmiersprache Pascal eine wichtige Rolle.

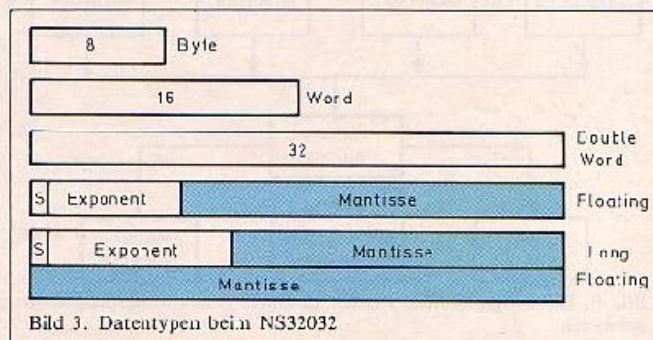


Bild 3. Datentypen beim NS32032

Symmetrie

Die Länge der Ausführungsbe-
fehle kann von einem bis zu 13
Bytes variieren. Ein Standard-
Befehlscode ist zum Beispiel wie
in Bild 5 aufgebaut.

Symmetrie ist eine Eigenschaft,
die es gestattet, daß für jeden
Operanden alle neun möglichen
Adressierungsarten gelten, zum
Beispiel

MOVB R0,A

(Schiebe den Inhalt des Re-
gisters R0 auf einen Spei-
cherplatz, der mit A defi-
niert ist. Die Länge der Ope-
randen betrage 1 Byte (B))

oder:

MOVW HANS,TOS

(Schiebe den Inhalt von
HANS in eine Speicherzelle,
die mit dem Stackpointer
adressiert ist. Die Operan-
den sollen die Länge 1 Word
(W) haben.)

Symmetrie sagt auch aus, daß
jeder Operation jeder Datentyp
zugeordnet werden kann. Der
Befehl MUL (multipliziere) ist
zum Beispiel möglich mit:

W = Word	MULW
R = Byte	MULR
D = Double Word	MULD
Q = Quadword	MULQ
F = Fließkomma	MULF
L = Fließkomma mit doppelter Genauigkeit	MULL

Arithmetik

Der Prozessor arbeitet mit
Zweioperandenarithmetik.

Operand 1 wird mit Operand 2
verknüpft, das Ergebnis wird in
Operand 2 geschrieben.

$A + B = B$ ADDW A,B
(Addiere A und B und lege das
Ergebnis in B ab.)

$R0 + R1 = R1$ ADDB R0,R1
(Addiere Register R0 mit Re-
gister R1 und lege das Ergebnis in
R1 ab.)

Einen 'Akkumulator' gibt es
nicht mehr, da jedes Register
und jeder Speicherplatz als
'Akkumulator' verwendet werden
kann. Wegen der Symme-
trie kann man für die Operan-
den 1 und 2 auch jede Adressie-
rungsart benutzen, der Umweg
über den 'Akkumulator' ent-
fällt dann ganz. Diese Eigen-
schaften sind für die Unterstüt-
zung von Hochsprachen sehr
wichtig.

NS16000-Befehlssatz

'Call External Procedure',
'Exit', 'Enter' und 'Return'
sind Befehle zum Unterpro-

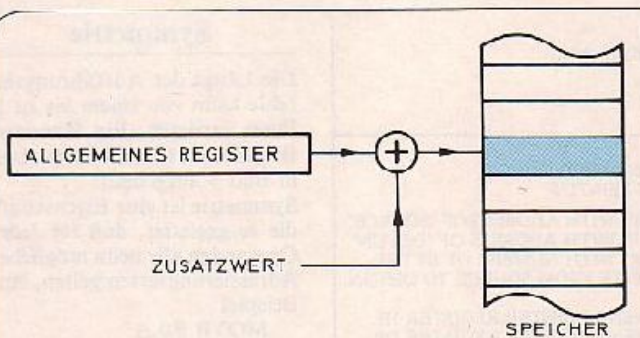


Bild 4. Register-relative Adressierung mit Zusatzwert (Offset)

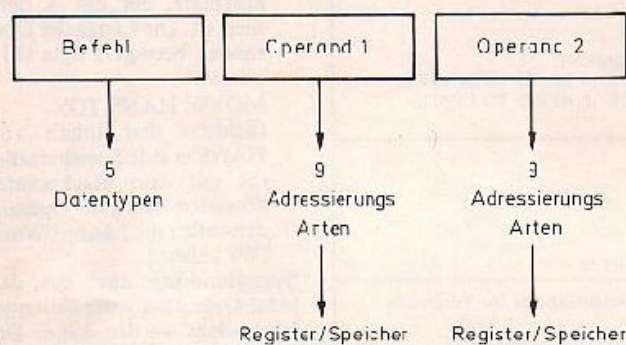


Bild 5. Befehlsstruktur im Maschinencode

grammaufruf. Automatisch retten sie die Register und manipulieren den Stackpointer. Spezielle String-Befehle gestatten die Abarbeitung von String-funktionen. Die Register R0 bis R4 übernehmen dabei besondere Aufgaben. Wichtig bei Pascal ist die Behandlung von Feldern und Arrays. Die Prozessoren der NS16000-Serie verfügen auch hierfür über einschlägige Befehle.

Moderne Hochsprachen (Pascal, C) gestatten die Aufteilung der Programme in Module. Jedes Modul kann unabhängig von allen anderen Modulen entwickelt und getestet werden. Beispielsweise kann ein Unterprogramm zur Berechnung von Quadratwurzeln unabhängig vom Hauptprogramm entwickelt und später dann im Hauptprogramm aufgerufen werden.

Der besondere Vorteil davon ist, daß man auch käufliche Pascal- oder C-Programme für seine Zwecke einbinden kann. Die Prozessoren der NS16000-Serie unterstützen als erste einen Programmaufbau in Modulen (Bild 6).

Die fertigen Programme werden von ihrer Quelle in den Be-

fehlscode des NS32032 kompiliert. Da, wie schon erwähnt, der Befehlscode an Hochsprachen angepaßt ist, entstehen dadurch kaum aufwendigere Assembler-Programme als unter Verwendung von einfachen Compilern. Quellprogramme in Assemblersprache werden assembliert.

Danach müssen die Programme gebunden (im Englischen spricht man von 'linken') werden, die einzelnen Programmenteile werden also in die gewünschten ROM-Bereiche geschrieben. Gleichzeitig werden Tabellen im ROM angelegt, die

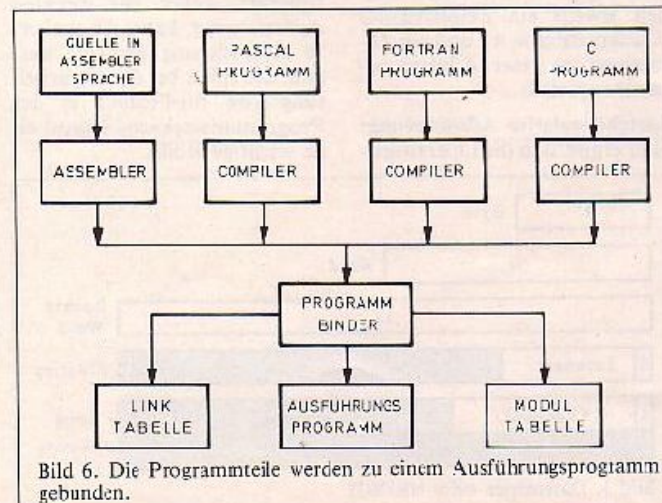


Bild 6. Die Programmteile werden zu einem Ausführungsprogramm gebunden.

Auskunft darüber geben, an welcher Stelle der Programmcode der Module beginnt, wo die globalen Variablen untergebracht sind und wie auf die Daten der anderen Programmodule zuzugreifen ist (Linkbasis). Diese Tabellen nennt man Modultabelle (Bild 7) und Linktabelle.

Adressen auch Speicherplätze, also als läge ein voller 16-M-Adressbereich vor.

In Wirklichkeit ist der Speicher unterteilt in einen RAM-Bereich, also einen direkt zugreifbaren Halbleiterspeicher, und einen Massenspeicher, also Platte oder Band. Da sowieso nie zur gleichen Zeit der gesam-

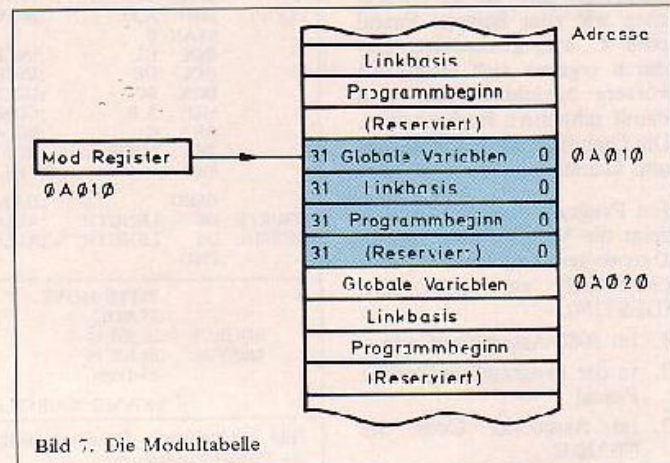


Bild 7. Die Modultabelle

Virtual Memory

24 Adressleitungen der CPU 32032 legen den Adressraum auf 16-M-Speicherzellen fest.

Der Optimalzustand wäre, diesen Adressraum mit direkt zugreifbaren RAM- oder ROM-Speichern voll auszunutzen. Dies ist mit heutigen Speichertechnologien wegen des hohen Kosten- und Platzaufwandes jedoch nicht möglich.

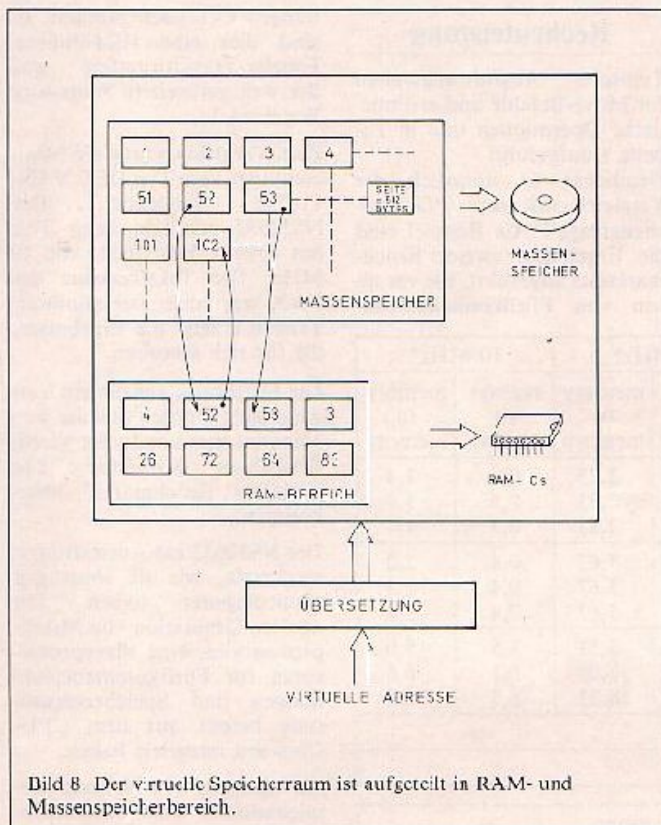
Um aber trotzdem den gesamten Bereich auszunutzen, bedient man sich der virtuellen Speicherung (Bild 8). Wie der Name schon sagt, sieht der Mikrocomputer den Speicher so, als wären alle verfügbaren

te Adressbereich benötigt wird, kann man die gerade benötigten Programnteile in einem RAM einlagern und die gerade nicht erforderlichen in einem Massenspeicher ablegen.

Der gesamte virtuelle Speicherbereich wird in Seiten (pages) unterteilt, von denen jede 512 Speicherzellen groß ist. Die gerade benötigten Seiten werden in den RAM-Bereich kopiert, dort eventuell manipuliert und nach längerem Nichtgebrauch wieder zurückgegeben oder verworfen. Benötigt der Mikrocomputer einen Programnteil, der nicht im RAM-Bereich vorhanden ist, wird der gerade ausgeführte Befehl angehalten. Der erforderliche Programnteil wird aus dem Massenspeicher in das RAM geladen. Danach wird der unterbrochene Befehl wiederholt.

Dieses Verfahren nennt man 'Demand Paging', es wird hier erstmals in einem Mikroprozessorsystem angewandt. Großcomputer, wie zum Beispiel die VAX/PDP 11 von der Firma Digital Equipment oder der IBM-370-Rechner, arbeiten mit diesem Verfahren der Speicher-verwaltung.

Eine andere Speicher-verwaltungstechnik ist die Segmentierung. Hier werden ganze Programme vom Massenspeicher einbeziehungsweise ausgelagert. Das Mikroprozessorsy-



stem MC68000 benutzt dieses Prinzip. Erfahrungen aus Großcomputern haben jedoch gezeigt, daß das Verfahren des 'Demand Paging' günstiger ist als die 'Segmentierung', weshalb sich National Semiconductor für das erstere entschied.

Die Verwaltung der Speicher übernimmt die 'Memory Management Unit', kurz MMU (NS16082). Dieser Slaveprozessor sitzt in einem separaten 48-Pin-Gehäuse und ist mit den Adreß-, Daten- und Steuerleitungen der CPU verbunden.

Der Einsatz der MMU bietet weitere Vorteile:

- Die teuren RAM-Bereiche können sehr klein dimensioniert werden. Die billigen Massenspeicher, Platte oder Band, können ein Maximum vor Daten aufnehmen. Auf diese Weise hat man die Vorteile von RAMs, nämlich schneller Speicherzugriff, mit denen des Massenspeichers (billig) vereint.
- Jeder Speicherseite kann ein gesondertes Lese- und Schreibrecht zugeordnet werden, so daß ein Schutz vor Benutzer eingriffen gegeben ist.

Der Speicherbereich läßt sich in zwei Räume aufteilen, in einen nur für den Benutzer zugänglichen, und in einen für den Systemverwalter zugreifbaren Teil. Auf diese Weise entsteht Sicherheit vor unbelegten Veränderungen von Daten.

Interne Architektur

Die CPU NS32032 ist in einem 68 Pin, 24 mm x 24 mm großen Chip-Carrier-Gehäuse untergebracht. Die Abmessungen des Chips selbst betragen 7 mm x 7 mm (siehe Foto). Zur Stromversorgung genügt eine einfache 5-V-Spannungsquelle.

Der Adreßbus ist 24 Bit und der Datenbus 32 Bit breit. Die Adreß- und Datenleitungen werden von den Anschlüssen AD0 bis AD23 gemultiplext aus der CPU geführt. Die restlichen Datenleitungen D24 bis D31 besitzen eigene Ein-/Ausgänge.

Die CPU besitzt acht 32-Bit-Register R0 bis R7. Sie können zum Halten von Operanden für jeden Befehl verwendet werden. Jedes Register kann auch als 'Akkumulator' benutzt werden. Bei 'String'-Befehlen haben die Register R0 bis R4 besondere Funktionen (Bild 9).

Acht fest zugeordnete Register der CPU speichern Adressen und Zustandsinformationen, die während des Programmablaufes entstehen (Bild 10).

Programmzähler und Statusregister sind wohl hinlänglich bekannt. Das Modulregister beinhaltet den Modulzeiger des ge-

rade ablaufenden Programmoduls (siehe auch Bild 7). Die CPUs der NS16000-Serie enthalten zwei voneinander unabhängige Stackpointer (SP0 und SP1). SP0 bezeichnet man als Interrupt-Stackpointer. Er wird benutzt, wenn die CPU im Supervisor-Mode (s.u.) ist. Der SP1 oder Anwenderstackpointer

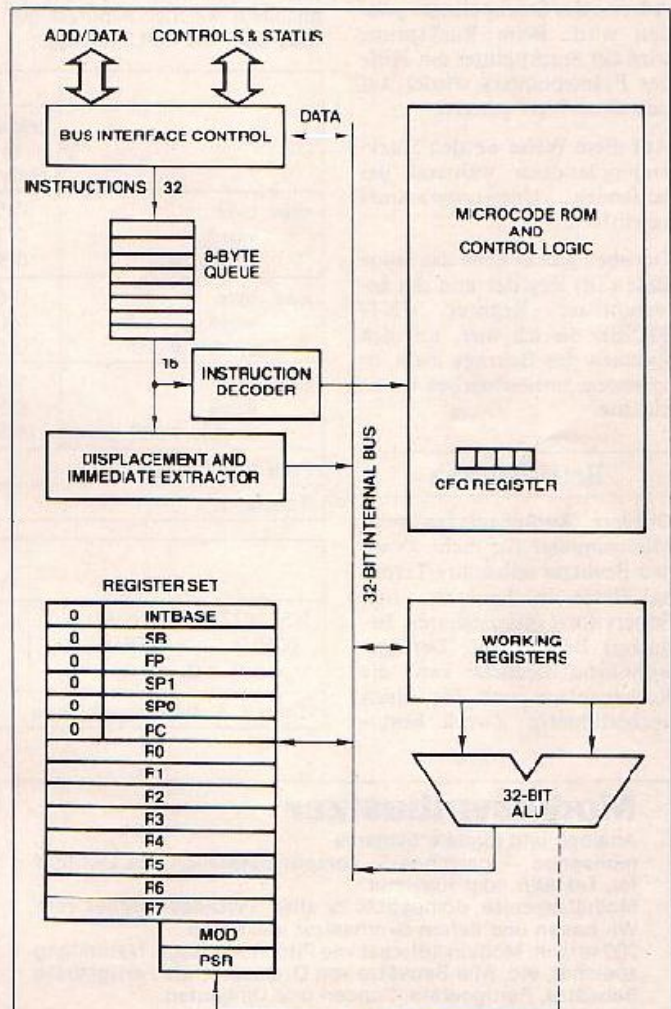


Bild 9: Interne Architektur des 32032

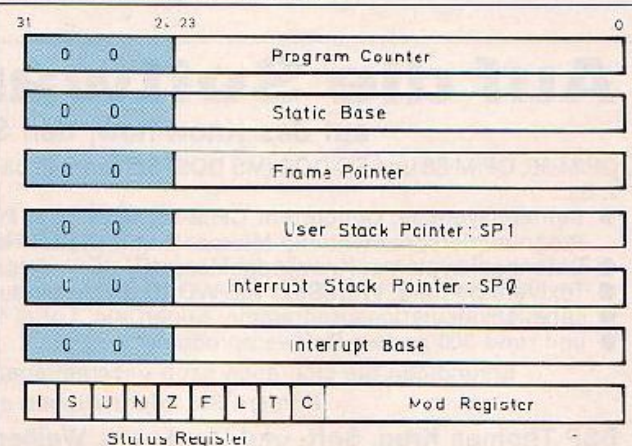


Bild 10: Die fest zugeordneten Register

ter kann vom Programmierer beispielsweise für 'Top-of-Stack'-Adressierungen verwendet werden. Ob SPC oder SPI verwendet wird, hängt davon ab, wie das I-Bit im Zustandsregister gesetzt ist.

Der Framepointer ist ein Register, das nach einem Unterprogrammaufruf mit der aktuellen Adresse des Stackpointers geladen wird. Beim Rücksprung wird der Stackpointer mit Hilfe des Framepointers wieder auf den alten Wert gesetzt.

Auf diese Weise werden Stackmanipulationen während des laufenden Unterprogrammes vereinfacht.

Daneben gibt es noch das Static Base (SB) Register und das Interruptbase Register (INTBASE), die ich hier, um den Rahmen des Beitrags nicht zu sprengen, unbeschrieben lassen möchte.

Betriebsarten

Größere Rechenanlagen oder Minicomputer für mehr als einen Benutzer teilen ihre Terminal-Plätze in Benutzer- und Supervisor- (privilegierter Benutzer) Betrieb auf. Der 'gewöhnliche' Benutzer kann die Rechenanlage nur für einen vorbestimmten Zweck benutzen.

Er kann die Programme nicht löschen oder ändern und nur die ihm zugeteilten Programme betreiben.

Der Supervisor hingegen hat die 'Macht', in alle Programme einzugreifen. Die Prozessoren der NS16000-Serie können zwischen diesen beiden Betriebsarten umgeschaltet werden, je nachdem welcher Benutzer gerade über die CPU verfügt.

	6 MHz*		10 MHz*	
	register to register	memory to memory	register to register	memory to memory
mov byte	0,50	2,33	0,3	1,4
word	0,50	2,33	0,3	1,4
double word	0,50	2,83	0,3	1,7
add byte	0,67	3,67	0,4	2,2
word	0,67	3,67	0,4	2,2
double word	0,67	5,67	0,4	3,4
mul byte	5,83	8,33	3,5	5,0
word	8,50	11,00	5,1	6,6
double word	13,83	18,33	8,3	11,0

* all times in μ s

Tabelle 1.

	Fourier-Transformation	Whetstone
NS32032 + NS16081 (CPU) (FPU)	3,5 μ s	7,8 μ s
VAX 11/750	4,0 μ s	6,0 μ s

Tabelle 2. Benchmark-Tests

Rechenleistung

Typische Ausführungszeiten für Move-Befehle und arithmetische Operationen sind in Tabelle 1 aufgeführt.

Deutlicher ist natürlich der Vergleich mit einer 'Großrechenanlage'. Als Beispiel sind die Ergebnisse zweier Benchmarktests angeführt, die vor allem von Fließkommaberechnungen Gebrauch machen. Es sind dies eine 1024-Punkte-Fourier-Transformation und der weit verbreitete Whetstone Benchmark.

Zum Vergleich wurde ein Minicomputer vom Typ DEC VAX-11/750 verwendet. Der NS32032 arbeitete beim Test mit einer Taktfrequenz von 10 MHz. Die Taktfrequenz der VAX war nicht beeinflussbar. Tabelle 2 zeigt die Ergebnisse, die für sich sprechen.

Die Ergebnisse aus diesem Vergleichstest wurden aus der von National Semiconductor veröffentlichten Broschüre 'The NS16000 Benchmarks' übernommen.

Der NS32032 hat Ausstattungsmerkmale, wie sie sonst nur Minicomputer haben. Die nächste Generation von Mikroprozessoren wird Slaveprozessoren für Fließkommaberechnungen und Speicherverwaltung bereits auf dem CPU-Chip mit integriert haben.

Als nächstes will National Semiconductor weitere Slaveprozessoren und Peripheriebausteine zur NS16000-Serie anbieten. Zum Beispiel eine Steuerung für direkten Speicherzugriff, einen Fehlerkorrekturchip und eine Plattenlaufwerksteuerung. □

Modulsynthesizer

Analoge und digitale Systeme monophon — polyphon — computergesteuert, via Lichtgriffel, Tastatur oder Klaviatur. Modulbauweise, kompatibel zu allen V/Okav-Synthesizern. Wir bauen und liefern Synthesizer nach Maß. 200 versch. Module lieferbar wie Pitch-to-Voltage, Naturklangspeicher, etc. Alle Bausätze von D. Doepfer als Fertiggeräte. Bausätze, Fertiggeräte, Sonder- und Umbauten. Info "M" anfordern.

s/w Graphic-Interface für μ P's

Komplett auf Europakarte, Auflösung 256 x 256, (adressierbar und darstellbar), 4 Bildspeicherebenen, (umschaltbar, getrennt für Display bzw. Write) High Intensity Attribut zur Hervorhebung einzelner Objekte oder Buchstaben. Bildspeicher auslesbar, Wort- und Pixelweise, einfacher Cursor-Darstellung mit passendem Adapter für alle PC's und HC's, 'Lightpen und Joy-Stick-Anschluß, BAS-Video-Ausgang (7 MHz—75 Ohm) Graphic-karte komplett mit Bechhissatz. Info "G" anfordern.

P. Meinhold, Eichenweg 4, 5900 Siegen 1 - Trupbach, Tel. 02 71/3 74 21

Auf die Software kommt es an, auf das Know-how, den Support und auch den Preis!

CP/M-80, CP/M-86 und PC-DOS (MS DOS)-Software in ca. 50 verschiedenen Diskettenformaten in der Regel ab Lager, z. B.:

- Betriebssysteme: Concurrent CP/M-86, CP/M IBM PC/XT (DM 212,99 incl. MwSt.)
- Programmiersprachen: alle Microsoft und Digital Research-Sprachen
- Datenbanksysteme: KnowledgeMan, dBASE II, Friday!
- Textverarbeitung: WordStar, MS-WORD, in Verbindung mit der MS-MOUSE
- Tabellenkalkulationsprogramme: SuperCalc, Lotus 1-2-3, Multiplan
- und rund 300 andere Softwareprodukte

Erkundigen Sie sich auch nach unserem speziellen IBM Personal Computer-Hardwareangebot!

Fordern Sie Informationen und unsere aktuelle Preisliste an:

BSP Thomas Krug, Soft- und Hardware, Weißenburgstr. 49, Postfach 110324, D-8400 Regensburg
Tel. 09 41/5 19 45 und 5 18 66, Tlx 6 52 510

Arithmetik-Unterricht für 6502 und Z80

Ulrich Schinke

Teil 2:

Multiplizieren und Dividieren

Die Prozessoren 6502 und Z80 kennen wie fast alle 8-Bit-CPU's keine Befehle zum Multiplizieren und Dividieren. Diese Grundrechenarten müssen durch Programme realisiert werden, die sich im wesentlichen auf Additions- und Subtraktionsbefehle stützen. Im zweiten Teil der 'Nachhilfestunde' für CPU's geht es um die Grundlagen für diese 'Hilfsprogramme', die an fertig programmierten Beispielen gezeigt werden.

Bei der Programmierung von Multiplikations- und Divisionsroutinen sind neben den Additions- und Subtraktionsbefehlen die Rotations- und Schiebefehle wichtig. Sie bewirken eine Verschiebung der Datenbits um eine Binärstelle. Dadurch erhält jedes Bit als neue Wertigkeit die seines 'Nachbarn' zur Rechten oder Linken. Bei endlicher Wortbreite, normalerweise 1 Byte, 'fällt' dabei an einer Seite des Wortes ein Bit hinaus, welches im Carry-Flag aufgefangen wird. Dem zufolge wird auf der anderen Seite des Wortes eine Binärstelle frei.

Die verschiedenen Befehlstypen unterscheiden sich nun lediglich durch die Behandlung des so freiwerdenden Bits:

Bei Verschiebungen erhält dieses eine 0, bei Rotationen den vorherigen Wert des Carry-Flags (Bild 1).

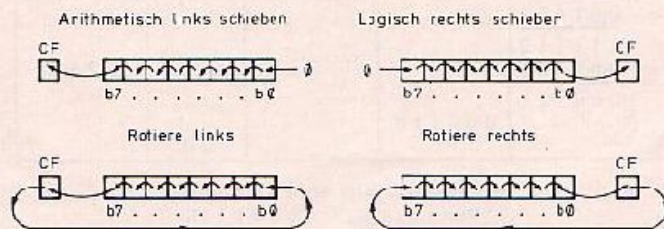


Bild 1. Rotier- und Schiebeoperationen

c't 1984, Heft 4

Somit sind Schiebeoperationen nichts anderes, als Rotationen mit gelöschtem Carry.

Für die 'zirkuläre' (8-Bit-) Rotation muß das C-Flag zuvor mit dem Wert des MSB beziehungsweise LSB des zu rotierenden Wortes geladen werden. Die einzelnen Bits dieses Wortes werden hierdurch zwar zyklisch vertauscht, nicht jedoch ihr Wert oder ihre Reihenfolge verändert. Ein 1 Byte Operand beinhaltet somit nach acht zyklischen Rotationen wieder sein ursprüngliches Bitmuster.

Die Verarbeitung von Mehrbyteoperanden muß durch eine Programmschleife realisiert werden, da sich die Rotations- und Schiebefehle sowohl beim 6502 als auch beim Z80 auf eine Wortbreite von 8 Bit (1 Byte) beziehen. Hierbei reicht das C-Flag das herausgeschobene Bit jeweils von Byte zu Byte weiter.

Fazit:

Mit Hilfe der Rotationsbefehle lassen sich alle Schiebe- und Rotieroperationen für Operanden beliebiger Wortlänge implementieren, indem zuvor das Carry-Flag mit dem entsprechenden Wert geladen wird.

Die in diesem Artikel aufgeführten Programme liefern zahlreiche Beispiele für den Einsatz von Rotationsbefehlen.

Im einfachsten Fall kann eine Multiplikation durch eine fortgesetzte Addition ersetzt werden:

$$7 \cdot 3 = 7 + 7 + 7 = 21.$$

Verständlicherweise wächst die Abarbeitungszeit derartiger Programme mit dem Wert des Multiplikators und ist, besonders für Mehrbyteoperanden, nicht mehr akzeptabel.

Einen Sonderfall stellt die Multiplikation und Division mit der Basis des gewählten Zahlensystems dar. Im Dualzahlensystem entspricht das arithmetische Linksschieben der Multiplikation mit zwei, das logische Rechtsschieben der Division durch zwei.

Leistungsfähigere Programme stützen sich auf die bei der schriftlichen Multiplikation und Division angewendeten

Methoden. Die Analyse dieser Rechenvorschriften führt auf die in den Bildern 2 und 3 abgebildeten Flußdiagramme. Die hierzu angegebenen Beispiele (Bild 4 und 5) dienen der Überprüfung und dem besseren Verständnis der Algorithmen. Sie helfen überdies, Fehler bei der Umsetzung der Flußdiagramme in die jeweilige Assemblersprache zu vermeiden.

Bei der Multiplikation (Bild 6) werden zunächst Teilprodukte gebildet und diese dann stellengerecht addiert. Im Binären entartet die Multiplikation zu einer Ja-/Nein-Entscheidung, ob der Multiplikand zum Zwischenergebnis addiert wird oder nicht.

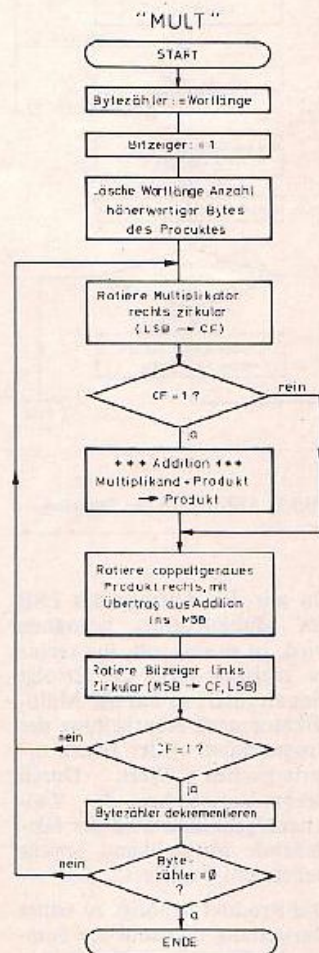


Bild 2. Ablaufplan der Multiplikation

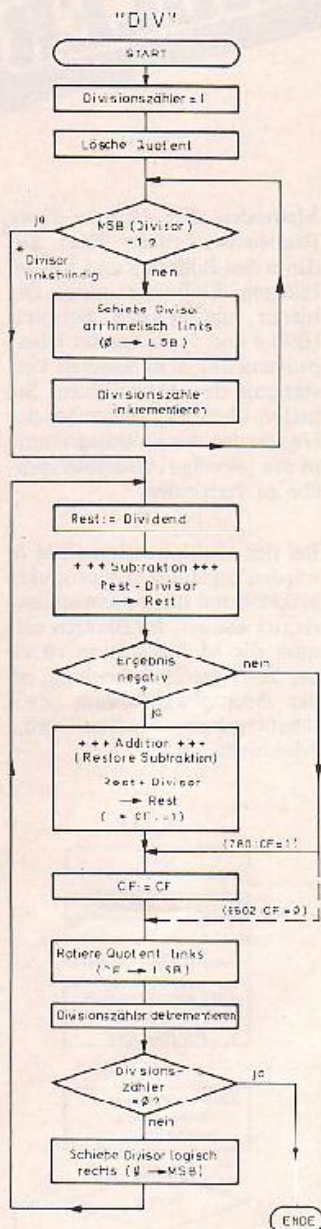


Bild 3: Ablaufplan der Division

Da mit der Abfrage des LSB des Multiplikators begonnen wird, ist es sinnvoll, ihn rechts zu rotieren (Bild 2). Erfolgt dies zirkular, so hat der Multiplikator nach Abarbeitung des Programmes wieder seinen ursprünglichen Wert. Durch Rechts-Verschieben des Zwischenergebnisses wird der feststehende Multiplikand jeweils stellengerichtet addiert.

Das Produkt benötigt zu seiner Darstellung maximal die Summe der Wortlängen der beiden Operanden. Bei zwei gleichlangen Operanden mit je 8 Bit rei-

chen somit in jedem Falle 16 Bit zur Produktdarstellung aus. 'Gerechnet' wird aber nur mit der höherwertigen Produkthälfte, die daher vor Beginn der Multiplikation gelöscht werden muß. Nachdem die Multiplikation vollständig durchlaufen wurde, steht das Produkt rechtsbündig im Speicher und hat damit den alten Inhalt der unteren Produkthälfte überschrieben.

Die etwas ungewöhnlich anmaßende Verwendung eines 'Bitzeigers' realisiert lediglich einen Zähler mit den Werten 1, 2, ..., 8. Die zirkulare Linksrotation entspricht somit einem Inkrementieren. Hierbei gilt: $8 + 1 = 1$ mit einem Übertrag ins C-Flag.

Während der 'Bytezähler' die Anzahl der bereits abgearbeiteten Multiplikatorbytes angibt, kennzeichnet der Bitzeiger das getestete Bit dieses Multiplikatorbytes.

Der Divisionsalgorithmus (Bild 7) ist dem der Multiplikation ähnlich. Der Divisor wird, beginnend beim MSB des Dividenden, jeweils stellengerichtet vom Dividenden subtrahiert. Aus diesem Grund wird der Divisor zunächst linksbündig geschoben. Die Anzahl der hierzu notwendigen Schiebeoperationen plus 1 entspricht der Anzahl auszuführender Subtraktionen. Bei gleicher Wertigkeit des Divisor- und Dividenden-MSB ist somit eine erfolgreiche Subtraktion denkbar.

War der Dividend kleiner als der Divisor, wird das Ergebnis negativ. In diesem Fall erhält der Quotient im LSB eine 0, sonst eine 1.

Während ein Mensch 'erkennt', welche zweier Zahlen die größere ist, muß der Computer diese testen. Divisionsprogramme arbeiten somit nach dem Prinzip 'Try and Error' (Bild 3). Der Divisor wird in jedem Falle vom Dividenden subtrahiert. Wird hierbei das Ergebnis negativ, so muß die Subtraktion durch eine erneute Addition des Divisors wieder storniert werden (restore method).

Die Z80-CPU (Tabellen 1,2) unterstützt die modulare Programmierung, indem sie Unterprogrammaufrufe (CALL) und Rücksprünge (RET) in Abhängigkeit der Flagzustände gestattet, ähnlich den Verzweigungs-

$$14 \cdot 9 = 126$$

C_D Multiplikand: $(14)_{10} = (1110)_2$

C_D Multiplikator: $(9)_{10} = (1001)_2$

Wortbreite := 2 Bits/Byte

Wortlänge := 2 Bytes

x := beliebiges Bit
0 oder 1

Produkt	Multiplikator	Bitzeiger	Bytezähler	
$\begin{array}{r} 0000XXXX \\ + 1110 \\ \hline 0110XXXX \\ 01110XXXX \end{array}$	$\begin{array}{r} 1001 \\ 1110C \end{array}$	01	10	1. Zyklus
$\begin{array}{r} 001110XX \\ + 1110 \\ \hline 011110XX \\ 011110XX \end{array}$	$\begin{array}{r} 0111C \\ 1101 \end{array}$	101	01	2. Zyklus
$\begin{array}{r} 0001110X \\ + 1110 \\ \hline 0111110X \\ 01111110 \end{array}$	$\begin{array}{r} 01011 \\ 11001 \end{array}$	10		3. Zyklus
$\begin{array}{r} 0001110X \\ + 1110 \\ \hline 0111110X \\ 01111110 \end{array}$	$\begin{array}{r} 11001 \\ 1101 \end{array}$	101	00	4. Zyklus

$$\rightarrow \text{Produkt: } = (126)_{10} = (01111110)_2$$

Bild 4: Ausführliches Beispiel zur Multiplikation

$$61 : 9 = 6 \text{ Rest } 7$$

→ Dividend: $(61)_{10} = (111101)_2$

→ Divisor: $(9)_{10} = (1001)_2$

Wortbreite := 2 Bits/Byte

Wortlänge := 2 Bytes

Divisor	Divzähler	
$\begin{array}{r} 001001 \\ - 010010 \\ \hline 110100 \end{array}$	01	Divisor linksbündig → Divisionszähler = 3
Rest: = Dividend		

Rest	Quotient	Divisor	Divisionszähler	
$\begin{array}{r} 111101 \\ - 100100 \\ \hline 011001 \\ 1 \end{array}$	000000	100100	11	1. Zyklus
$\begin{array}{r} 011001 \\ - 100100 \\ \hline 110101 \end{array}$	000001	→ 100100	10	
$\begin{array}{r} 011001 \\ - 100100 \\ \hline 110101 \end{array}$	000011	→ 100100	01	2. Zyklus
$\begin{array}{r} 011001 \\ - 100100 \\ \hline 110101 \\ + 100100 \\ \hline 110011 \end{array}$	000110		00	3. Zyklus

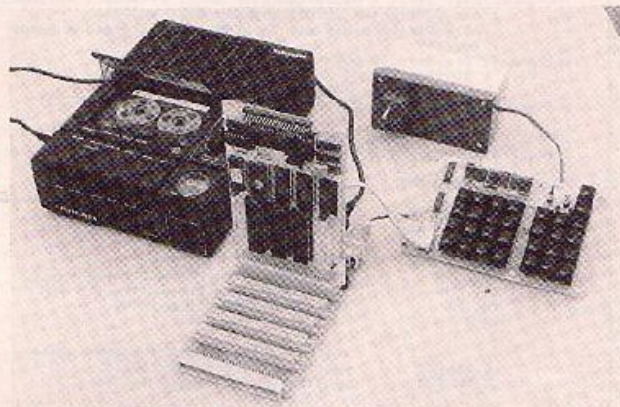
$$\rightarrow \text{Quotient: } (6)_{10} = (000110)_2$$

$$\text{Res: } (7)_{10} = (000111)_2$$

Bild 5: Ausführliches Beispiel zur Division

COBOLD

IHR Lern- und Proficomputer auf drei Platinen!
Der ideale Einstieg in die Microprozessortechnik



COBOLD — ein Computer mit zauberhaften Qualitäten dank eines neuen, raffinierten Hardware-Konzepts und eines sagenhaft komfortablen Betriebssystems. Auf drei Platinen:
— ein Maschinensprache-Computer auf Basis 6502/65C02, der auch Textverarbeitung, BASIC und FORTRAN kann.
— der sinnvollste Einstieg in die Microprozessortechnik.
— der Computer für alle — auch Ihre — Problemstellungen.
— beschrieben mit Bauanleitung in ELRAD 3, 4 + 5/83.

Lernen auch Sie zaubern wie ein Cobold — steigen Sie ein in die Microprozessortechnik mit dem neuen elrad-COBOLD-System! Fordern Sie Prospekte an!

Die Komplett-Ausstattungen:

GRUNDVERSION: (CIM 65-Prozessor, Basis- und "D"-Platine) mit CPU 6502, RIOT 6532, 2 K RAM, Monitor-EPROM. Basisplatine besückt mit 1 Federleiste.

Bausatz DM 298,—
Bausatz mit fertiger CPU-Karte DM 389,—
Fertig aufgebautes System DM 449,—

ERWEITERTE VERSION (Grundversion mit 4K RAM, 3x RIOT 6532, Basisplatine mit 5 Federleisten).

Bausatz DM 398,—
Bausatz mit fertiger CPU-Karte DM 498,—
Fertig aufgebautes System DM 519,—

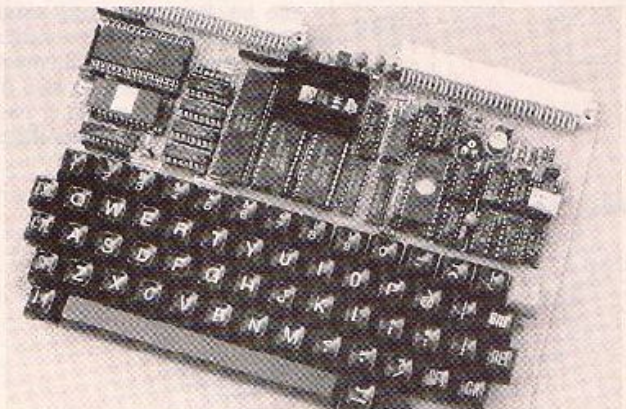
NETZTEIL für den COBOLD im Steckergehäuse DM 49,— (Bausatz) bzw. DM 86,— (fertig) (auch für c't-Terminal geeignet).

DAS HANDBUCH für den COBOLD: „6502/65C02 Maschinensprache“ von C. Persson DM 48,—.

Intelligentes Terminal
mit professionellen Attributen:

c't-Terminal

DER Terminal-Computer auf Doppel-Euro-Karte
mit oder ohne integrierter Tastatur!



- beschrieben in c't Nr. 12/83 und 1/84
- 6511-Singlechipcomputer mit 6545-Videocontroller
- 4 KB-Bildwiederholpeicher (scrollbar)
- Bildformate: 80x25 oder 64x20 (per Software umschaltbar)
- Zeichenmatrix 8x11 (bei 80x25) oder 8x13 (bei 64x20)
- max. 8 Zeichensätze (inkl. Blockgrafik)
- Invers-, Blink-Modus, Breitschrift, halbe Helligkeit
- seriell Interface (V24- oder TTL-Pegel)
- integrierte Centronics-Schnittstelle
- integrierte Spannungsregelung und -wandlung für V24
- 8-bit-parallel (ASCII) oder 8x9 Tastenmatrix Tastaturanschluß
- PREISE: Version A (ohne Tastatur)

Bausatz DM 449,— DM 549,— Fertigkarte
Platinenmaße 230x85 mm

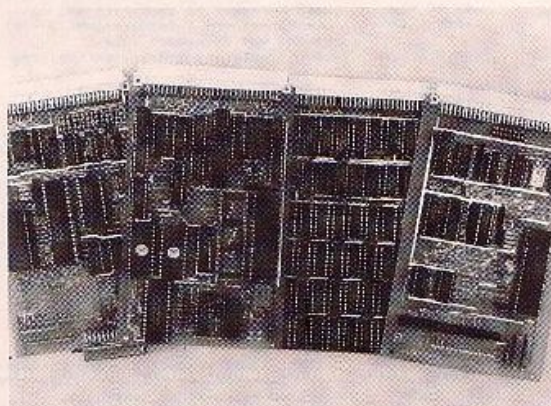
Version B (mit integrierter Tastatur)

Bausatz 498,— DM 639,— Fertigkarte
Platinenmaße 233x160 mm

Prospektmaterial auf Anforderung!

Endlich: c't-86

Das 1. echte 16-bit-Microcomputer-System
der Welt (auch) zum Selbstbau!



Ein Vier-Karten-System, basierend auf dem für 16 bit erweiterten ECB-Bus:

- echte 16-bit-Rechenleistung
- kein neuer, sondern ein weit verbreiteter Bus
- dadurch bereits existierende ECB-Peripherie-Karten einsetzbar
- Betriebssysteme CP/M-86 und MS-DOS II
- Komplettpreis unter DM 1700,— (Bausatz)
- vorgestellt in Heft 1, 2 + 3/84 von c't — dem neuen Magazin für Computertechnik

Die vier Karten:

Platine 1: **CPU-KARTE mit 8086**, optional 8087 Arithmetik-Prozessor, 8259 Interrupt-Controller, 8 KB Monitorprogramm mit CP/M-86-Urlader.

Komplett-Bausatz DM 349,—; DM 449,— Fertigkarte

Platine 2: **I/O-KARTE** mit V24-Interface für Terminal-Anschluß, Centronics-Schnittstelle, Kassettenspeicher-Interface und Timer.

Komplett-Bausatz DM 249,—; DM 349,— Fertigkarte

Platine 3: **FLOPPY-CONTROLLER-KARTE** zum Anschluß bis zu 4 Laufwerke 5 1/4 oder 8 Zoll (auch gemischt) mit dem neuen Controler IC WD 2757.

Komplett-Bausatz DM 498,—; DM 598,— Fertigkarte

Platine 4: **256-KB-RAM-KARTE** mit 128 oder 256 KB dyn. RAM (max. 3 Karten einsetzbar ± 768 KB RAM!).

Komplett-Bausatz DM 598,— (128 KB) bzw. DM 899,— (256 KB),
DM 698,— bzw. DM 999,— Fertigkarte

Geplante Ergänzungen: CPU-Karte mit 68000, SASI-Interface, Vollgrafikkarte, Z80 Subprozessor-Karte.

Bus Karte mit 10 Steckplätzen fertig DM 169,—

Leerplatinen, Floppy-Laufwerke, Netzteile und Gehäuse auf Anfrage

Komplettsystem, anschlussfähig im Gehäuse, mit Laufwerken ab DM 3985,—

Fordern Sie Prospekte an!

VERSAND: per NK (+ Versandkosten) oder per Vorauskasse (V-Scheck oder Überweisung auf Pschtkto Han 1429 28-308, keine Versandkosten). Ausland nur gegen Vorauszahlung (+ DM 15,— Versandkosten). Alle Preise inkl. MwSt. Händleranfragen erwünscht.

MARFLOW — COMPUTING GmbH

Brüderstraße 2 · 3000 Hannover 1 · Telefon 05 11/32 60 98


```

13 14 1101 1110
 52      0000
 13      1101
182      1101
          1101
          111
          10110110

```

Bild 6: Multiplikation

```

182:14 = 13      10110110:1110 = 1101
-14
 42
-42
 0

191:15 = 12 Rest 11  10111111:1111 = 1100
-15
 41
 30
 11

```

Bild 7: Division ohne und mit Rest

anderen Programme beliebig oft zugreifen, müssen jedoch die im Programmkopf aufgeführte Eingaben bereitstellen. Bei jedem Unterprogrammaufruf muß der Prozessor sich die Stelle 'merken', an die er nach Abarbeitung der aufgerufenen

den Verzweigungsbefehlen (JR, JMP). Der Vorteil dieser Programmierweise liegt in der größeren Übersichtlichkeit und sicheren Testbarkeit. Auf diese Unterprogramme können alle

Routine zurückkehren muß. Hierzu legt die CPU die Rückkehradresse auf einen Stapel (Stack). Dieser Speicher wird nach dem 'LIFO'-Prinzip (Last In, First Out) verwaltet, da das

Tabelle 1.

```

;Aufgabe: "MLT" multipliziert zwei Mehrbyte-Wörter mit der in
; : WORTL festzulegenden Genauigkeit und speichert das
; : Produkt mit doppelter Wortlänge ab PRD.
; : Eingaben werden durch das Programm nicht zerstört.
;Eingaben: Multiplikand ab MLD
; : Multiplikator ab MTR
; : Wortlänge in WORTL
;Ausgaben: Produkt ab PRD mit 2*(WORTL)
;Variablen: PRD, MLD, MTR, WORTL, BITZ, BITZ+1
;Unterprogramme: ADD, RRV
;Zerstört: AF, BC, HL, DE, IX

MULT: LD A,(WORTL) ;Wortlänge über Akku
      LD C,A ;ins C-Register laden
      LD IX,BITZ ;BITZ: nächstes abzu-
      LD (IX+0),1 ;beitendes Bit eines Multiplikatorbytes
      LD (IX+1),C ;Bytzähler (BITZ+1): noch abzuarbeitende
;Bytes des Multiplikators

      LD HL,PRD ;+++ Produkt löschen +++
      LD B,C ;Wortzeiger --> HL, Wortlänge --> B
      CLR HL ;obere Hälfte des Produktes löschen, da
      DJNZ CLR1 ;untere Hälfte ohnehin durch Schieberegister-
;tionen überschrieben wird

      LD HL,MTR ;+ Multiplikator rechts zirkular rotieren
      LD B,C ;Wortzeiger --> HL und
      PUSH HL ; auf Stack zwischenspeichern
      ADD HL,BC ;LB-Adresse (Multiplikator) =
      DEC HL ; Wortzeiger + Wortlänge - 1
      RRC (HL) ;LSB (Multiplikator)
      RL (HL) ; -- CF kopieren
      POP HL ;Wortzeiger wieder vom Stack holen
      CALL RRV ;Multiplikator 1 Bit rechts zirkular
;rotieren

      LD DE,PRD ;+++ Additionseingaben:
      LD HL,MLD ;Wortzeiger (Zwischenergebnis) --> DE
      ;Wortzeiger (Operand) --> HL

      CALL C,ADD ;Falls LSB (Multiplikand) 1 war (-> CF=1):
; Multiplikand + Produkt --> Produkt

      LD DE,HL ;+++ Produkt rechts rotieren ++
      LD HL,PRD ;Wortzeiger (Produkt) vom DE ins HL
      RRC HL ;Rotiere doppelt genau Produkt um 1 Bit
      RRC HL ;rechts, mit Übertrag aus Addition ins MSB

      RLC (IX+0) ;nächstes Bit des Multiplikatorbytes
      JR NC,MLT1 ;wiederhole bis Bit 7 abgearbeitet wurde
      DEC (IX+1) ;8 Bits abgearbeitet --> Bytzähler - 1
      JR NZ,MLT1 ;wiederhole bis alle Bytes abgearbeitet

      RET

```

Tabelle 2.

```

;Aufgabe: "DIV" teilt 2 Mehrbyte-Wörter durch einander, entspricht
; : den Quotienten ab QTT und den Rest ab DVD.
; : Alle Wörter haben die in WORTL festzulegende Genau-
; : igkeit.
;Eingaben: Dividend ab DMD
; : Divisor ab DVS
; : Wortlänge in WORTL
;Ausgaben: Quotient ab QTT
; : Rest ab DVD
;Unterprogramme: RRV, RLW, ADD, SUB
;Zerstört: Dividend, HL, BC, DE, AF, IX

DIV: LD A,(WORTL) ;Wortlänge über Akku
     LD C,A ;ins C-Register laden
     LD IX,DIVZ ;Divisionszähler (DIVZ) := 1, da
     LD (IX+0),1 ;wenigstens 1 Division möglich

     LD HL,QTT ;+++ Quotient löschen +++
     LD B,C ;Wortzeiger --> HL, Wortlänge --> B
     LD (HL),0 ;alle Bytes des Quotienten auf 03H
     INC HL ;setzen, beginnend beim H3
     DJNZ CLR2

     LD DE,DVD ;++ Addition-, Subtraktionseingaben:
     LD HL,DVS ;Wortzeiger (Rest) --> DE
     ;Wortzeiger (Divisor) --> HL

     ;+ Divisor linksbündig schieben ++

     RLC (HL) ;Kopiere MSB (Divisor) ins CF
     RR (HL) ;Falls CF=1, ist Divisor links-
     JR C,DVIS ;bündig --> weiter bei DVIS

     XOR A ;CF = 0 --> Schreibe Divisor 1 Bit
     CALL RLW ;arithmetisch links
     INC (IX+0) ;inkrementiere Divisionszähler
     JR RL1 ;Verzweige zur nächsten MSB Abfrage

DVIS: CALL SUB ;< Rest = Divisor --> Rest >
      ;Falls Ergebnis negativ (CF=1)
      ; --> Restore Subtraktion:
      ; < Rest + Divisor --> Rest >

      PUSH HL ;++ Rotiere Quotient links ++
      CCF ;Rette HL auf Stack
      LD HL,QTT ;CF = CF'
      CALL RLW ;Rotiere Quotient 1 Bit links,
      POP HL ; mit CF' ins LSB
      ;hole HL wieder vom Stack

      DEC (IX+0) ;dekrementiere Divisionszähler

      RET Z ;Fertig, falls Divisionszähler=0!

      PUSH HL ;++ Schiebe Divisor rechts logisch ++
      XOR A ;Rette HL auf Stack
      LD HL,DVS ;CF = 0 --> Schiebe Divisor 1 Bit
      CALL RRV ;rechts logisch
      POP HL ;hole HL wieder vom Stack

      JR DVIS ;Verzweige zur nächsten Division

```

Unterprogramm

```

;Aufgabe: "ADD" addiert einen Operanden zu einem Zwischenergebnis
; : mit der im BC-Register festzulegenden Genauigkeit.
;Eingaben: Wortlänge -- BC
; : Wortzeiger (Operand) -- HL
; : Wortzeiger (Zwischenergebnis) -- DE
;Ausgaben: Ergebnis ab (DE)
;Zerstört: AF

ADD: EX DE,HL ;Inhalt des DE mit dem des HL-Registers
      ADD HL,BC ;vertauschen, um mit 2 Byte Addition die
      EX DE,HL ;LB-Adressen +1 zu berechnen,
      ADD HL,BC ;zunächst vom Zwischenergebnis (DE),
      ;dann vom Operanden (HL)
      LD E,C ;Wortlänge ins E-Register kopieren

      XOR A ;(AKKU) und C-Flag löschen
      DEC DE ;nächst höheres Byte adressieren
      DEC HL ;++ Byteweise Addition ++
      LD A,(DE) ;Zwischenergebnis + Operand + mög-
      ADC A,(HL) ;licher Übertrag --> Zwischenergebnis
      LD (DE),A ;+ Wortlänge (B) - 1
      DJNZ NADD ;Wiederhole bis alle Bytes abgearbeitet

      ;Aufgabe: "SUB" subtrahiert einen Operanden von einem Zwischener-
      ; : gebnis mit der im BC-Register festzulegenden Genauigkeit
      ;Eingaben: Wortlänge -- BC
      ; : Wortzeiger (Operand) -- HL
      ; : Wortzeiger (Zwischenergebnis) -- DE
      ;Ausgaben: Ergebnis ab (DE)
      ;Zerstört: AF

SUB: EX DE,HL ;Inhalt des DE mit dem des HL-Registers
      ADD HL,BC ;vertauschen, um mit 2 Byte Addition die
      EX DE,HL ;LB-Adressen +1 zu berechnen,
      ADD HL,BC ;zunächst vom Zwischenergebnis (DE),
      ;dann vom Operanden (HL)
      LD B,C ;Wortlänge ins B-Register kopieren
      XOR A ;(AKKU) und C-Flag löschen

      DEC DE ;nächst höheres Byte adressieren
      DEC HL ;+++ Byteweise Subtraktion +++
      LD A,(DE) ;Zwischenergebnis - Operand - mög-
      SBC A,(HL) ;licher Übertrag --> Zwischenergebnis
      LD (DE),A ;+ Wortlänge (B) - 1
      DJNZ NSUB ;Wiederhole bis alle Bytes abgearbeitet

      RET

```


Unterprogramm

```

;Aufgabe: "RRW" rotiert ein Wort um 1 Bit rechts, wobei die
; : Operationsart durch (CF) bestimmt wird.
;Eingaben: Wortzeiger -- HL, Wortlänge -- C
; : CF: 0 --> Schiebe rechts logisch
; : HSB --> Schiebe rechts arithmetisch
; : LSB --> Rotiere rechts zirkular
;Ausgaben: 1 Bit rechts verschobenes Wort ab Wortzeiger (HL)
; : (HL) = Wortzeiger + Wortlänge
;Zerstoert: Flags, IL

RRW: LD B,C ;Wortlänge ins B-Reg. kopieren
RR: RR (HL) ;Wort byteweise um 1 Bit rechts rotieren, mit CF ins MSB
INC HL ;Wiederhole bis alle Bytes abgearbeitet
DJNZ RR
RET

;Aufgabe: "RLW" rotiert ein Wort um 1 Bit links, wobei die
; : Operationsart durch (CF) bestimmt wird.
;Eingaben: Wortzeiger -- HL, Wortlänge -- C
; : CF: 0 --> Schiebe links arithmetisch
; : HSB --> Rotiere links zirkular
;Ausgaben: 1 Bit links verschobenes Wort ab Wortzeiger (HL)
; : (HL) = Wortzeiger + Wortlänge
;Zerstoert: Flags

RLW: PUSH AF ;Flags auf Stack zwischenspeichern
ADD HL,BC ;Wortzeiger + Wortlänge --> HL
POP AF ;Flags wieder vom Stack holen

LD B,C ;Wortlänge ins B-Reg. kopieren
RL: RL (HL) ;Wort byteweise um 1 Bit links rotieren, mit CF ins LSB
DJNZ RL ;Wiederhole bis alle Bytes abgearbeitet
RET

END

```

zuletzt aufgestapelte Byte auch als erstes wieder heruntergenommen werden muß.

Die 'Höhe' des Stapels bestimmt der Stapelzeiger (Stackpointer), der immer die Adresse des nächsten freien Stapelplatzes enthält.

Der Stack kann auch vom Anwender zum Zwischenspeichern von Variablen sowie zum Kopieren und Austauschen von Speicher- und Registerinhalten mit Hilfe der PUSH- und POP-/PULL-Befehle genutzt werden. Es ist jedoch darauf zu achten, die von der CPU angelegte Ordnung nicht zu stören. Alle aufgestapelten Daten müssen wieder vom Stack 'heruntergenommen' worden sein, bevor die CPU durch einen 'Return'-Befehl veranlaßt wird, ins Hauptprogramm zurückzuspringen. Sie interpretiert nämlich die beiden zu oberst auf dem Stack liegenden Daten als Rücksprungadresse, unabhängig davon, ob dies eventuell dort 'vergessene' Operanden sein mögen. Solche Fehler führen im allgemeinen zum 'Absturz' des Rechners.

Die Abfrage des Divisors-MSB wird durch zwei Rotierbefehle (RRC (HL) und RR (HL)) realisiert, die das MSB in das C-Flag kopieren. Dies demonstriert eine weitere Anwendung der Rotationsoperationen.

Der 6502 (Tabellen 3,4) unterstützt gegenüber dem Z80 die Einbindung der abzuarbeitenden Sequenzen in das Hauptprogramm. Hierdurch wird meist eine kürzere Rechenzeit

erzielt, da keine Parameter an Unterprogramme übergeben werden müssen und die Programm-Module speziell angepaßt werden können.

Durch die 'indirekte Adressierung', die tatsächliche Adresse wird aus der Summe einer Basisadresse plus dem Inhalt des Distanzregisters (X) bestimmt, sind für die einzelnen Programmblöcke gegenüber den Z80-Programmen weniger Befehle notwendig.

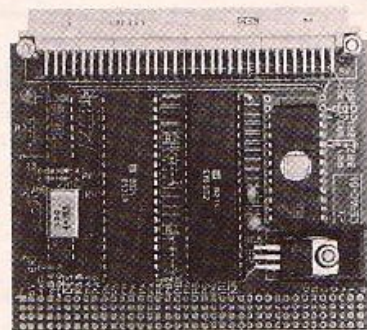
Obwohl der 6502 einen absoluten Sprungbefehl besitzt, sollten innerhalb eines überschaubaren Programmblocks die schnelleren Branchbefehle diesem vorgezogen werden, wobei die Verzweigung sich auf ein mit Sicherheit gesetztes oder gelöscht Flag stützt.

Besondere Beachtung muß die unterschiedliche Behandlung des Carry-Flags bei der Subtraktion finden:

Während beim Z80 das C-Flag bei einem negativen Ergebnis gesetzt (CF=1) und bei positivem Resultat gelöscht wird (CF=0), handhabt der 6502 dies genau umgekehrt. Im Divisionsprogramm des Z80 (Programm 2) wird die zur Stornierung notwendige Addition nur dann ausgeführt, wenn das C-Flag gesetzt ist; im 6502-Programm (Programm 4) wird sie bei gesetztem Carry übersprungen.

Wer Programmierung nicht als Selbstzweck betreibt, steht spätestens jetzt vor der Aufgabe, sich die hier entwickelten Programme 'MULT' und 'DIV'

NEU: CEPAC-65



Der preisgünstige Single-Board-Computer für Festprogramm-Anwendungen. Festsatz — Version A — DM 69,—
Version B DM 89,—

Massenweise MONITORE! Massenweise

Heath-Zenith ZVM 122 (bernstein) DM 355,—
(12 Zoll, 15 MHz) ZVM 123 (grün) DM 355,—
PRINCE (bernstein) DM 555,—
(12 Zoll, 25 MHz) (grün) DM 495,—
LOW-COST-DRUCKER
EUROPRINT 6311 (100 Z/s) DM 999,—

TANDON-FLOPPY-Laufwerke

— u.a. von IBM für gut befunden ...

TM 50-1 (ss/dd, 250 KB, 1 x 40 Track) DM 586,—
TM 50-2 (cs/dd, 500 KB, 2 x 40 Track) DM 693,—
TM 55-2 (cs/dd, 500 KB, 2 x 40 Track) DM 793,—
TM 55-4 (cs/gd, 1 MB, 2 x 80 Track) DM 943,—
TM 55-2 + 55-4 sind mikroprozessor-gesteuert (extrem genaue Kopfpositionierung)
Alle Laufwerke in Slimline-Ausführung (1/2 Bauhöhe).
P.S. Tandon hat einen Weltmarktanteil von 75 %!

Mit uns speichern Sie richtig!

5,25" 10 Stck.
SS/SD 48TPI 52,—
SS/DD 48TPI 58,—
SS/QD 96TPI 74,—
DS/DD 48TPI 79,50
DS/QD 96TPI 84,—
8"
SS/SD 58,—
SS/DD 67,—
DS/DD 79,50
Größere Mengen Preis auf Anfrage. Preise pro Stück inkl. MwSt. zzgl. Versandkostenanteil: DM 5,—.

Leerplatten für c't-Projekte:

c't-Terminal — Version A — (ohne Tastatur) DM 59,—
c't-Terminal — Version B — (mit Tastatur) DM 76,—
Universelles Netzteil DM 14,—
c't-86-System: Prozessor-Karte DM 85,—
RAM-Karte DM 88,—
I/O-Karte DM 69,—
Floppy-Karte DM 65,—
BUS-Platine (96pol.) für 10 Steckplätze DM 49,—
c't-Sprachsynthesizer DM 21,—
CEPAC-65 auf Anfrage

VERSAND: per NN (+ Versandkosten) oder per Vorauskassa (V-Check oder Überweisung auf Postkonto Han 1429 26-308, keine Versandkosten). Ausland nur gegen Vorauszahlung (+ DM 15,— Versandkosten). Alle Preise inkl. MwSt.

MARFLOW—COMPUTING GmbH

Brüderstraße 2 · 3000 Hannover 1 · Telefon 05 11/32 60 98

Tabelle 3.

```

;Aufgabe: "MULT" multipliziert zwei Mehrbytewörter mit den in
; : WORTL fest: zitierten Genauigkeit und speichert das
; : Produkt mit doppelter Wortlänge ab PRD.
; : Eingaben werden durch das Programm nicht zerstört.
;Eingaben: Multiplikand ab MLD
; : Multiplikator ab MTR
; : Wortlänge in WORTL
;Ausgaben: Produkt ab PRD mit 2*WORTL
;Variablen: PRD, MLD, MTR, WORTL, BIT2, BYTEZ
;Zerstört: alle Register

MULT: CLD                ;Binärmode !
      LDA WORTL          ;Wortlänge über Akku
      TAX                ;X=Zählregister kopieren
      DEX                ;Wortzähler (X) := Wortlänge - 1
      STA BYTEZ          ;Bytezähler (BYTEZ) := Wortlänge
      ROL                ;Wortlänge * 2 -->
      STA DWRTL          ;Doppelte Wortlänge (DWRTL)
      LDA #1             ;Bitzeiger (BIT2) := 1
      STA BIT2           ;

      LDA #0             ;+++ Produkt löschen +++
      STA PRD,X          ;Obere Hälfte des Produktes löschen,
      DEX                ;da untere Hälfte ohnehin durch Schiebe-
      BPL CLR1           ;operationen überschrieben wird
      ;Wiederhole, bis Wortlänge Bytes
      ;gelöscht sind

      LDX WORTL          ;Multiplikator rechts zirkular rotieren +
      DEX                ;Wortzähler := Wortlänge - 1
      LDA MTR,X          ;LSB (Multiplikators) über Akku
      ROR                ;ins CF kopieren
      LDX #A             ;Wortzähler (X) := 0

      LDA WORTL          ;Wortzähler (Y) := Wortlänge
      ROR MTR,X          ;Multiplikator 1 Bit rechts zirkular
      INX                ;rotieren, beginnend beim HB
      DEY                ;Wiederhole, bis alle Bytes abgearbeitet
      BNE RR1            ;worden --> Wortzähler (Y) = 0

      BCC NOADD          ;überspringe die Addition, falls
      ;MSB (Multiplikators) 0 war --> CF = 1

      DEX                ;Wortzähler (X) := Wortlänge - 1
      DEC C              ;CF := 0

      ADD: LDA PRD,X     ;(Zwischenergebnis + Operand + möglicher
      ADC MLD,X          ;Übertrag) --> Zwischenergebnis +
      STA PRD,X          ;Wortlänge - 1, beginnend beim LB
      DEX                ;Wiederhole, bis alle Bytes abgearbeitet
      BPL ADD            ;--> Wortzähler (X) < 0

      NOADD: LDX #0      ;++ Produkt rechts rotieren ++
      LDY WORTL          ;Rotiere doppelt genaues Produkt um
      ROR PRD,X          ;(DWRTL) ins Y-Register kopieren
      INX                ;1 Bit rechts, mit Übertrag aus Addition
      DEY                ;ins MSB, beginnend beim HB
      BNE RR2            ;Wiederhole, bis alle Bytes abgearbeitet
      ;--> Doppelte Wortlänge (DWRTL) = 0

      ROL BIT2           ;Rotiere Bitzeiger links
      DEC MLI            ;Wiederhole, bis Bit 7 abgearbeitet
      ROL BIT2           ;Kopiere CF ins Bit 0
      DEC BYTEZ          ;8 Bits abgearbeitet --> Bytezähler - 1
      BNE MLI            ;Wiederhole, bis alle Bytes abgearbeitet
      ;--> Bytezähler = 0

      RTS

```

Tabelle 4.

```

;Aufgabe: "DIV" teilt 2 Mehrbytewörter durch einander, speichert
; : den Quotienten ab GTT und den Rest ab DVD.
; : Alle Wörter haben die in WORTL festzulegende Genau-
; : keit.
;Eingaben: Dividend ab DVD
; : Divisor ab DVS
; : Wortlänge in WORTL
;Ausgaben: Quotient ab GTT
; : Rest ab DVD
;Variablen: DVD, DVS, GTT, DIVZ
;Zerstört: alle Register

DIV: CLD                ;Binärmode !
      LDA WORTL          ;Wortlänge über Akku ins
      TAX                ;X=Zählregister kopieren
      DEX                ;Wortzähler (X) := Wortlänge - 1
      LDA #1             ;Divisionszähler (DIVZ) := 1,
      STA DIVZ           ;da wenigstens 1 Division möglich

      LDA #0             ;+++ Quotient löschen +++
      STA GTT,X          ;alle Bytes des Quotienten löschen,
      DEX                ;beginnend beim LB
      DPL CLR2           ;Wiederhole, bis alle Bytes gelöscht

      LDIV: LDA DVS       ;++ Divisor linksbündig schieben ++
      RMI DIVS           ;Kopiere HB (Divisors) in Akku
      LDY WORTL          ;linksbündig, falls MSB = 1
      DEX                ;Wortzähler (X) := Wortlänge - 1
      CLC                ;CF := 0
      RL1: RMI DVS,X      ;Schiebe Divisor 1 Bit arithmetisch
      JEX                ;links
      RPI RPI            ;Wiederhole, bis alle Bytes verschoben
      INC DIVZ           ;inkrementiere Divisionszähler
      BCC RL1            ;Da CF immer 0 --> Verzweige unbedingt
      ;zu erneuter MSB Abfrage

      DIVS: LDX WORTL     ;
      DEX                ;Wortzähler (X) := Wortlänge - 1
      SEC                ;CF := 1

```

Unterprogramm

```

SUB1: LDA RVN,X          ;++ Byteweise Subtraktion ++
      SBC DVS,X          ;(Zwischenergebnis - Operand + möglicher
      STA RVN,X          ;Übertrag) --> Zwischenergebnis +
      DEX                ;Wortlänge - 1, beginnend beim LB
      BPL SUB1           ;Wiederhole, bis alle Bytes abgearbeitet
      ;--> Wortzähler (Y) < 0

      BCC NOADD          ;Falls Ergebnis pos (CF=1), überspringe
      ;Addition
      ;Kopiere Wortlänge - 1 in Wortzähler (X)
      LDX WORTL
      DEX                ;

      ADD2: LDA DVD,X     ;++ Byteweise Addition ++
      ADC DVS,X          ;(Zwischenergebnis + Operand + möglicher
      STA DVD,X          ;Übertrag) --> Zwischenergebnis +
      DEX                ;Wortlänge - 1, beginnend beim LB
      BPL ADD2           ;Wiederhole, bis alle Bytes abgearbeitet
      ;--> Wortzähler (Y) < 0
      DEC C              ;CF := 0

```

Unterprogramm

```

NOADD: LDX WORTL         ;Wortlänge - 1 in Wortzeiger (X)
      DEX                ;
      ROL GTT,X          ;++ Quotient links rotieren ++
      DEX                ;Rotiere Quotient 1 Bit links,
      BPL RL3            ;mit CF ins LSB
      ;Wiederhole, bis alle Bytes rotiert

      DEC DIVZ           ;Dekrementiere Divisionszähler
      BNE NEXT          ;Fertig, falls Divisionszähler = 0

      NEXT: LDY WORTL    ;++ Divisor rechts logisch schieben ++
      LDX #0            ;Wortlänge in Wortzähler (Y) kopieren
      CLC                ;Wortzähler (X) := 0
      ROR DVS,X          ;CF := 0
      INX                ;Divisor 1 Bit logisch
      DEY                ;rechts schieben
      BNE RR3            ;Wiederhole, bis alle Bytes rotiert

      BCC DIVS          ;Da CF immer 0 --> Verzweige unbedingt
      ;zu nächster Division

```

für ihren eigentlichen Zweck, der Multiplikation und Division von Mehrbyteoperanden dienbar zu machen.

Vor der Assemblierung der Programme muß der Anwender den symbolischen Variablen tatsächliche, physikalische Adressen zuweisen. Dies kann durch manuelles Ersetzen der Symbole durch ihre Adressen im Quelltext oder mit Hilfe von Pseudoassembleranweisungen geschehen, zum Beispiel MLD EQU \$2000, die diesen Vorgang automatisch während der Assemblierung durchführen.

Alle Routinen können als Subroutinen von einem übergeordneten Programm aufgerufen werden. Sie verarbeiten alle im Programmkopf als Eingaben deklarierten Operanden, die, wie in Bild 8 dargestellt, zuvor im Speicher abgelegt sein müssen. Es ist darauf zu achten, daß der Stapelzeiger einen genügend großen Speicherbereich als Stack verwalten kann. Vereinbarungsgemäß adressiert er diesen vom höchstwertigsten Byte aus abwärts.

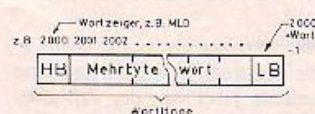


Bild 8: Datenformat

Zum Schluß noch einige Bemerkungen, inwieweit diese Divisions- und Multiplikationsprogramme mit der Zahlencarstellung im Zweierkomplement vereinbar sind: Bei einer im Zweierkomplement dargestellten Zahl repräsentiert das MSB vereinbarungsgemäß das Vorzeichen (0 --> positiv, 1 --> negativ), wobei alle dazustellenden Zahlen die gleiche Wortlänge haben müssen (c't 3/84).

'MULT' ordnet jedoch dem Produkt die doppelte Wortlänge der Operanden zu. Trotzdem ist diese Routine für im Zweierkomplement dargestellte Zahlen nutzbar, wobei dann nur die untere Produkt Hälfte das Ergebnis repräsentiert.

Demgegenüber setzt 'DIV' positive Integerzahlen voraus, da der Algorithmus das C-Flag als Überlaufanzeige verwendet. Diese Aufgabe übernimmt aber bei der Zweierkomplement-Darstellung das V-Flag! Dies bedeutet nun nicht zwangsläufig eine Umänderung des Divisionsprogrammes, sondern läßt sich durch betragsmäßige Verarbeitung der Zahlen umgehen. Hierbei muß der Benutzer entscheiden, ob Quotient oder Rest negativ ist, und diese dann gegebenenfalls komplementieren.

SYSTECH GMBH

OLYMPIA-COMPACT: Qualität mit Rechneranschluß



Typenrad Schreibmaschine mit Universal-Interface (c't-Februar-Heft), kann parallele oder serielle Daten verarbeiten (nur DIL-Schalter umschalten und Stecker austauschen!). Die Halbleiterschaltung erlaubt die Darstellung mathematischer Formeln (Matrizen, Potenzen). Typenräder mit verschiedenen Schriften und Formelzeichen sind verfügbar.

Technische Daten:

- parallele Schnittstelle (kann z. B. anstelle des MX-80 angeschlossen werden)
- serielle Schnittstelle mit 1200 Baud (paßt z. B. an den Osborne- oder Zenith-Rechner) mit DTR-Handshake (DTR-Pegel wählbar)
- deutscher Zeichensatz, Datenpuffer mit 44 Zeichen
- Schreibgeschwindigkeit max. 14 Zeichen/s, 9 Zeichen/s inkl. Werenrücklauf

NEU: Interface-Version 3.2 mit wesentlichen Verbesserungen

Baugruppe mit paralleler Schnittstelle (jetzt auch Acknowledge-Signal verfügbar), RS-232-Schnittstelle mit DTR-Handshake und wählbarem CBM-Zeichensatz (z. B. C-64, VC-20). Wahlweise RS-232-Stecker bzw. Stecker für parallele Schnittstelle, SYCOM 3.2B DM 343,—
wie vor, anstelle des CBM-Zeichensatzes XON-XOFF-Protokoll für die RS-232-Schnittstelle, SYCOM 4.0B DM 343,—
CBM-Bausatz (2000-, 3000-, 5000er-Serie):
wie vor, für den CBM-IEC-BUS mit CBM-Zeichensatz SYCOM 1.0B DM 393,—
wie vor, jedoch als Bausatz SYCOM 3.2BS, SYCOM 4.0BS DM 293,—
wie vor, jedoch nur Platine & CPU SYCOM 3.2F, SYCOM 4.1P DM 193,—
Anschlußleitungen für C-64, VC-20 mit aktivem Interface DM 63,—
Anschlußleitungen für Zenith Z89/90, Osborne, Acorn DM 49,—
Es liegen allen Baugruppen und Bausätzen Einbauhinweise, eine Bedienungsanleitung und der Zeichensatz bei.
OLYMPIA-COMPACT Typenrad Schreibmaschine, komplett mit Interface nach Wahl (SYCOM 1.0BS, SYCOM 4.0BS) DM 1398,—
mit Interface für CBM-IEC-BUS, SYCOM 1.0 DM 1448,—

c't-Sprachsynthesizer:



Sprachausgabe aus jedem Rechner mit einer parallelen Schnittstelle. Freier Textaufbau durch Phonemsynthese, notwendige Datenrate ca. 10 Byte/s. Programmierbar über BASIC, PASCAL, C oder Assembler. Die Sprachausgabe nach dem c't-Projekt aus dem Februar-Heft.

Sonderausführungen für RS-232 und CBM-IEC-Bus sind lieferbar.

Fertigkarte (parallele Eingabe), Anschluß an VC-20, C-64, CBM 2001, ..., (über den USER-Port) und jede parallele Drucker-schnittstelle möglich, entsprechende Beschreibung wird mitgeliefert: DM 298,—

Wie vor, Bausatz: DM 248,—

Wie vor, jedoch nur Sprachsynthesizer IC und Platine: DM 172,—

Komplettsystem im Gehäuse mit Netzrafo, anschlußfertig: DM 448,—

Wie vor, für CBM-IEC-BUS (z. B. CBM 2001, ...) oder RS-232-Schnittstelle, anschlußfertig: DM 548,—

Einkartenrechner mit Z80-CPU und Monitor

Für Meß- und Regelsysteme, Steuerungen.

- ZE-2: 2 Z80 PIO's, Z80-Interrupt (IM2), bis 8K-EPROM, 3K-RAM, 2,5 MHz Takt: DM 298,—
- ZE-64: wie ZE-2, zus. Z80A CTO, bis 16K-EPROM, 64K-RAM, 4 MHz Takt: DM 598,—
- AD 12B: passender 8-Kanal-12-Bit-Umsetzer (50 µs/Kanal): DM 998,—
- AT 2: Anzeige und 24 Tasten passend zu ZE 2 & 64: DM 168,—
- ZE-2-SER: Serielle & parallele Schnittstelle für ZE-2 & ZE-64: DM 98,—
- SYS-264: Monitorprogramm für ZE-2/64, unterstützt ADU, Schnittstelle, Anzeige und Fehlersuche. Mit ZE-264 Hardwarebeschreibung, Monitorbeschreibung, Listing und Monitor im 2716-EPROM: DM 98,—

Ab April lieferbar:

Datenerfassung mit dem VC-20/C-64

Grundeinheit mit Eingangsbuffer für maximal 8 A/D-Umsetzer, Datenerfassungsprogramm auf Kassette. Die Grundeinheit wird an den USER-Port angeschlossen. DM 98,—

Analog-Digitalumsetzer für den VC-20/C-64

Eingänge für Temperaturfühler (LM 390), $\pm 1V$, $\pm 1mA$ sind vorhanden, Auflösung 2000 Schritte, eine Messung pro Sekunde, maximale Entfernung vom Rechner 50 m, 3adrige Zuleitung DM 98,—



Referenzspannungsquelle

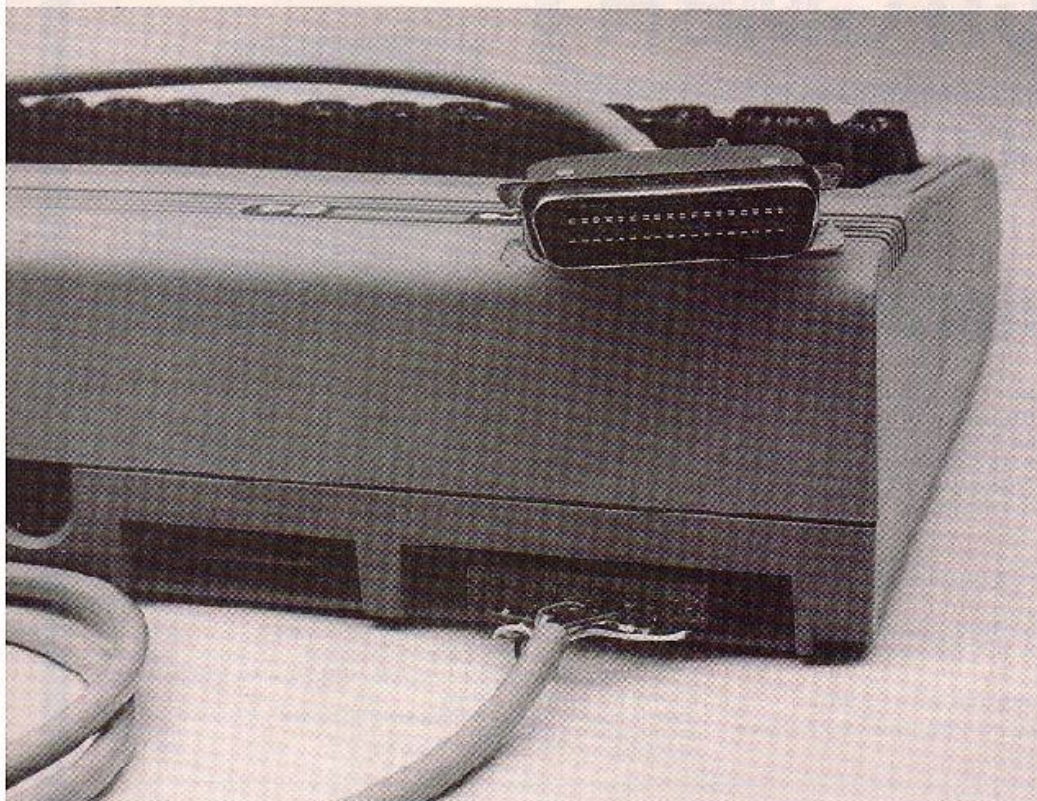
Digitale einstellbare Gleichspannungsquelle 10 V in 10000 Schritten zu 1 mV, kurzschlußfest. Festwertigkeit: unter 100 Mikrovolt.

Ideal zum Abgleich und zur Kontrolle von Analog-Digital-Umsetzern und als preiswerte Referenzspannungsquelle am Arbeitsplatz DM 398,—

Auftragserwicklungen: analoge und digitale Schaltungen, Mikroprozessoren: Z80, 8085, MCS-48 und 6801/68701, Speicherprogrammierte Steuerungen.

Alle Preise einschließlich Mehrwertsteuer, Versand per Nachnahme oder Vorkasse. Lieferzeiten 2...4 Wochen.

SYSTECH GmbH, Gillesmaroder Str. 26, 3300 Braunschweig, Tel.: 0531/34 56 41 & 37 23 91



weitere Steuerleitungen vorhanden. Sie dienen dazu, den angeschlossenen Drucker zu initialisieren und weitergehende

Pin	Signal	Pin	Signal
1	Strobe	19	GND
2	Data 0	20	GND
3	Data 1	21	GND
4	Data 2	22	GND
5	Data 3	23	GND
6	Data 4	24	GND
7	Data 5	25	GND
8	Data 6	26	GND
9	Data 7	27	GND
10	Ack	28	GND
11	Busy	29	GND
12	Paper out	30	GND
13	NC	31	Initial
14	GND	32	Error
15	GND	33	GND
16	GND	34	CLK
17	Chassis	35	Test
18	+5 V	36	+5 V

Bild 2. Pinbelegung einer Centronics-Schnittstelle

Centronics-Schnittstelle für VC-20 und C64

Nur das Kabel löten

Georg Werner

Alle hardwareseitigen Voraussetzungen für den Anschluß eines handelsüblichen Druckers mit Centronics-Schnittstelle sind sowohl beim VC-20 als auch beim C64 erfüllt. Warum Commodore die Software für diese Schnittstelle nicht mitliefert, wird wohl für immer ein Rätsel bleiben. Die paar Bytes, die nötig sind, um die Schnittstelle zu treiben, wären gewiß noch frei gewesen. Vielleicht wollte man die Kompatibilität zu den Commodore-Druckern nicht gefährden ...? — Welche Motive auch immer hinter dieser Produktpolitik stecken mögen, Schwamm drüber! Zum Anschluß eines Druckers Ihrer Wahl brauchen Sie nur ein Verbindungskabel und die hier vorgestellte Software. Auch der c't-Sprachsynthesizer läßt sich an dieser Schnittstelle betreiben.

Zur Erinnerung: Die Centronics-Schnittstelle ist eine Parallel-Schnittstelle mit acht Daten- und mehreren Handshake-Lei-

tungen. Letztere übernehmen die Steuerung des Datenverkehrs ('Handshake': wörtlich 'Händeschütteln'). Der Sender signalisiert mit einem Strobe-Impuls (STB), daß gültige Daten vorhanden sind und vom Empfänger übernommen werden sollen. Dieser quittiert den Empfang mit einem Acknowledge-Impuls (ACK). Daneben gibt es eine weitere Handshake-Leitung, nämlich BUSY. Der Empfänger setzt diese Leitung auf log. 1, wenn er ein Datenbyte empfangen hat. Ist er bereit, das nächste Byte zu übernehmen, legt er den Anschluß wieder auf 0-Pegel. Der sendende Computer kann diese Leitung abfragen, um zu erfahren, ob er ein weiteres Byte übertragen kann.

Einen genauen Überblick über die zeitliche Abfolge gibt Bild 1. Wer es noch genauer wissen will, sollte sich den Grundlagen-Artikel 'Parallel-Schnittstellen' in der c't-Ausgabe 1/84 (Seite 74) noch einmal durchlesen.

Bild 2 zeigt die Anschlußbelegung einer Centronics-Schnittstelle. Außer der Daten- und Handshake-Leitung sind noch

Informationen über Fehlerzustände an den Computer zu übertragen. Für den Datentransport werden sie nicht benötigt, denn jeder Fehlerzustand wird vom Drucker zusätzlich durch das Busy-Signal angezeigt. Die zusätzlichen Leitungen werden deshalb in unserem Vorschlag nicht berücksichtigt.

VC-20 und C64 stellen an ihren User-Ports jeweils 24 Leitungen zur Verfügung, von denen

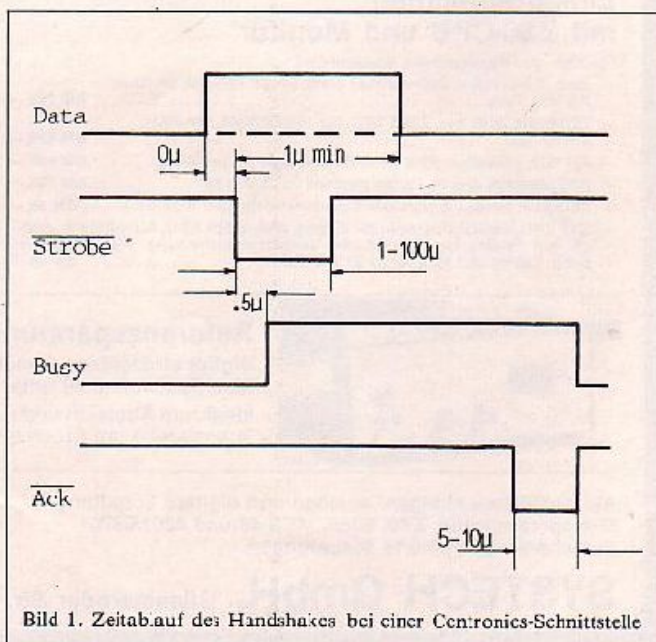


Bild 1. Zeitablauf des Handshakes bei einer Centronics-Schnittstelle

zehn für die Druckerschnittstelle verwendbar sind. Einige dieser Leitungen werden übrigens auch für die RS-232-Karte benötigt. Man muß sich also entscheiden: Die Datenausgabe über den User-Port kann entweder nur seriell (RS-232) oder parallel (Centronics) erfolgen.

Beide Commodore-Computer ähneln sich weitgehend, so daß die folgende Beschreibung im Prinzip auf VC-20 und C 64 anwendbar ist. Wo es Unterschiede gibt, werden diese vermerkt, oder die für den C 64 geltenden Angaben werden in Klammern gesetzt.

Als Datenleitungen werden die Anschlüsse PB0...PB7 verwendet (siehe Bild 3). Beim VC-20 kommen diese direkt aus dem Interfacebaustein VIA 6522 mit der Basisadresse 9110h. Beim C 64 gehören diese Portleitungen zu dem Interface-Chip CIA 5526 mit der Basisadresse DD00h.

Die beiden übrigen Leitungen dienen dazu, den Handshake auszuführen: Nach einer Schreib- oder Leseoperation an Port B liefert CB2 (PC2) einen 0-Impuls, der bei der Centronics-Schnittstelle als Strobe-Signal Verwendung findet. Der Anschluß CB1 (FLAG2) reagiert auf eine negative Flanke des Eingangssignals. Der Interfacebaustein setzt dann ein Flag in seinem Interruptregister. Hier wird das Busy-Signal des Druckers angeschlossen.

Der gesamte Hardware-Aufwand für den Einbau der Centronics-Schnittstelle beschränkt sich also auf die Herstellung eines Verbindungskabels. Bild 3 zeigt, wie der Amphenolstecker der Centronics-Schnittstelle mit dem Stecker für den User-Port zu verbinden ist. Das zwölfadrige Kabel sollte allerdings nicht länger sein als 1,5 Meter, denn die Interface-Bausteine können eine längere Leitung nicht störungsfrei treiben.

Drucken in BASIC

Ein einfaches BASIC-Programm genügt, um einen Drucker, den c't-Sprachsynthesizer oder andere Peripheriegeräte zu steuern. Die Struktur eines solchen Programms ist aus dem Beispiel (siehe Tabelle 1) leicht ersichtlich. Zunächst muß die Schnittstelle initialisiert werden. Dafür sorgt das

Amphenolstecker	Userport
1	M (VC-20) 8 (C 64)
2	C
3	D
4	E
5	F
6	H
7	J
8	K
9	L
11	B
19...29 verbinden	A,N

Bild 3. Verbindung des Amphenolsteckers der Centronics-Schnittstelle mit dem Userport.

Unterprogramm ab Zeile 20000. In Zeile 20010 werden die Portleitungen des Interface-Bausteins als Ausgänge programmiert. In Zeile 20020 wird der richtige Handshake-Modus vereinbart. Der C 64 kennt nur eine Art von Handshake, deshalb kann für diesen Rechner Zeile 20020 entfallen. In Zeile 20030 wird als erstes Zeichen eine '00' (ASCII: NUL) übertragen. Dieses Zeichen dient dazu, den Drucker zu initialisieren. Die Übertragung ist auch dann notwendig, wenn das Peripheriegerät diesen Befehl nicht versteht, denn es antwortet auf jeden Fall mit einem ACK-Impuls. Das ist nötig, damit der Rechner nicht später vor der Ausgabe des ersten Zeichens 'hängenbleibt' (Zeile 10050).

Der zweite Teil ist die eigentliche Druckroutine (10000). Hier wird der String in einzelne Zeichen zerlegt. Diese werden in den ASCII-Code umgesetzt und dann über den Port B ausgegeben. Dabei wartet das Programm stets in Zeile 10050, bis der Drucker den Empfang des zuletzt gesendeten Zeichens mit Acknowledge bestätigt und damit die Bereitschaft zum Empfang des nächsten gemeldet hat.

VC 20 und C 64 können mit verschiedenen Zeichensätzen arbeiten, nämlich mit Großbuchstaben/Grafik und mit Groß- und Kleinschreibung. Im zweiten Zeichensatz wird die ASCII-Norm teilweise nicht eingehalten. Damit dies bei der Ausgabe auf einem Standard-Drucker nicht zu unverständlichem Kauderwelsch führt, wird der Code im Programmsegment ab Zeile 15000 entspre-

```

10 gosub 20000 :rem interface initialisieren
20 p$="Drucken mit Centronics"
25 p$ = p$ + chr$(13)
30 gosub 10000 :rem drucken
40 input p$
50 if p$="ende" then end
55 p$ = p$ + chr$(13)
60 gosub 10000 : goto 40
70 :
10000 rem ausgabe des strings
10010 for i = 1 to len(p$)
10020 p = asc(p$)
10030 gosub 15000
10040 p$ = mid$(p$,2)
10050 wait 37136+13,16 :rem c64: 56576+13,16
10060 poke 37136+0,p :rem c64: 56576+1,p
10070 next
10080 return
10090 :
15000 rem veraendern des Zeichensatzes
15010 if (peek(36864+5) and 2) = 0 then return
15015 rem c64: if <peek(53248+24) and 2> = 0 then return
15020 if p < 64 then return
15030 if p > 191 then p = p and 127 :return
15040 p = p or 32 :return
15050 :
20000 rem initialisieren
20010 poke 37136+2,255 :rem c64: 56576+3,255
20020 poke 37136+12,175 :rem c64: entfällt
20030 poke 37136+0,0 :rem c64: 56576+1,0
20040 return

ready.

```

Tabelle 1. BASIC-Programm zur Ansteuerung der Centronics-Schnittstelle

chend umgesetzt. Welche der Betriebsarten der Benutzer ausgewählt hat, erkennt das Programm in Zeile 15010 an der Adresse des Zeichengenerators im Videocontroller. Eine genaue Beschreibung der Codewandlung entzieht der Bericht 'Seriell drucken' in c't 3/84.

Schneller drucken

Für die Textverarbeitung mit einem Schönschift-Drucker oder für die Sprachausgabe mit dem c't-Sprachsynthesizer ist die Geschwindigkeit des BASIC-Programms ausreichend. Wer jedoch einen Matrixdrucker oder einen Plotter betreiben will, wird ein schnelles Maschinenprogramm vorziehen.

Das hier vorgestellte Maschinenprogramm besteht aus drei Teilen:

Die Subroutine CHROUT wird vom Betriebssystem über den Vektor 0325h aufgerufen, wenn ein Zeichen ausgegeben werden soll. Der Computer prüft in diesem Programmsegment, ob die Ausgabe über die Centronics-Schnittstelle erfolgen soll. Ist das nicht der Fall, erfolgt ein Rücksprung in das Betriebssystem. Im anderen Fall wird eine Routine ausge-

führt, die ein Zeichen über den Userport ausgibt. Dieser Teil entspricht in seiner Struktur exakt dem BASIC-Programm ab Zeile 10000.

Der zweite Teil ist das Unterprogramm CHKOUT. Dieses Unterprogramm wird vom Betriebssystem über den Vektor 0320h aufgerufen, um festzustellen, auf welchem Gerät eine Ausgabe erfolgen soll. Dabei wird aus der Speicherzelle BA die Gerätenummer geholt und mit der Nummer der Centronics-Schnittstelle (5) verglichen. Bei Übereinstimmung wird die Nummer in der Speicherzelle 9Ah als die des aktuellen Ausgabegeräts abgelegt. Andernfalls erfolgt wiederum ein Rücksprung in das Betriebssystem.

Der dritte Teil ist die Routine INI. Sie übernimmt die Funktion des BASIC-Programms ab Zeile 20000 und zusätzlich das Setzen der Vektoren 0320h und 0326h. Außerdem wird in diesem Programmteil der Zeiger 'End of RAM' heruntergesetzt, um ein Überschreiben des Maschinenprogramms durch BASIC-Variablen zu verhindern.

Die Routine INI muß vor der ersten Benutzung der Centronics-Schnittstelle mit 'SYS

Assembler "CENTRONICS"

Achtung für C64 Werte in Klammern
verwenden!!!

```

1000 43 PHA ,BEGIN CHROUT
1001 A5 5A LDA GERÄT
1003 C3 C5 CMP #CENTRON
1005 F0 C3 BEQ AUS
1007 4C 7D F2 JMP RÜCK
      (4C CD F1)
AUS 100A A0 1D 91 LDA FLAG
      (A0 CD DD)
100E 29 10 AND #10
100F F0 F9 BEQ AUS ;ENTSPRICHT WAIT
1011 A0 C5 90 LDA VIDEO ;ZEICHEN ÄNDERN
      (A0 18 DD)
1014 29 02 AND #02
1016 F0 10 BEQ OK1
1018 68 PLA
1019 C9 40 CMP #40
101B 90 0A BCC OK2
101D C9 C0 CMP #C0
101F 90 04 BCC CN3
1021 29 7F AND #7F
1023 D0 02 BNE OK2
1025 09 20 ORA #20
OK2 1027 48 PHA ;ENDE ÄNDERN
OK1 1028 68 PLA
1029 8D 10 91 STA PORT ;AUSGABE
      (8D 01 DD)
102C 18 CLC
102D 60 RTS
102E EA NOP ;BEGIN CHKOUT
102F 20 CF F3 JSR FILENR
      (20 0F F3)
1032 F0 03 BEQ 1D37
1034 4C 84 77 JMP FILE NOT OPEN
      (4C 01 F7)
1037 20 DF F3 JSR SETPARA
      (20 1F F3)
103A A5 5A LDA GERÄTNR
103C C9 05 CMP #CENTRON
103E F0 05 BEQ CEN
1040 A5 5A LDA GERÄTNR
1042 4C 16 F3 JMP RÜCK1
      (4C 5D F2)
CEN 1045 4C 2E F3 JMP RÜCK2
      (4C 75 F2)
1048 00 BRK
1049 00 BRK
104A 00 BRK
104B 00 BRK
104C A9 FF LDA #FF ;BEGIN INI
104E 8D 12 91 STA DDRB
      (8D 03 DD)
1051 A9 AF LDA #A7
      (EA EA)
1053 8D 1C 91 STA INTER
      (EA EA EA)
1056 A9 00 LDA #00
1058 8D 10 91 STA PORT
      (8D 01 DD)
105B 8D 26 03 STA CHRVEKL
105E A9 ** LDA #START
1060 8D 27 03 STA CHRVEKH
1063 8D 21 03 STA CHKVEKH
1066 85 38 STA END BASIC
1068 A9 2F LDA #STRICH
106A 8D 20 03 STA CHRVEKL
106D 60 RTS

```

Tabelle 2. ASSEMBLER-Programm 'Centronics'

(INI)' aufgerufen werden. Das-
selbe gilt nach Betätigen der
'Restore'-Taste, denn dabei

werden vom Betriebssystem
sämtliche Vektoren zurückge-
setzt und die Portbausteine neu

	START	Anfangs- adresse
VC-20 8K	3F	3F00
VC-20 16K	5F	5F00
C64	9F	9F00

Tabelle 3. Anfangsadressen und
Werte der Variablen START bei
verschiedenem Speicherausbau des
Rechners

initialisiert. Verfügt man diesen
Befehl, erscheint bei einem
Druckversuch ein 'device not
present error'.

Das Maschinenprogramm ist so
geschrieben, daß es an jeder
Stelle im Speicher des VC-20
und des C64 arbeiten kann.
Dadurch ergibt sich eine einfache
Anpassung an die jeweilige
Rechnerkonfiguration. Nur ein
einziges Byte muß in Abhängig-
keit vom Startpunkt eingesetzt
werden. Dieses Byte ist mit '*'
gekennzeichnet und wird im
folgenden mit 'START' be-
zeichnet.

Bei VC-20 mit maximal 3
KByte Speichererweiterung
kann man das Maschinenpro-
gramm am einfachsten im
RAM direkt unterhalb des
Bildschirmspeichers ablegen.
START erhält dann den Wert
1Dh. Beim C64 und beim
VC-20 mit 8- oder 16-K-Erwei-
terung sollte das Programm die
oberen RAM-Adressen be-
legen. Die entsprechenden Werte

```

SYS (INI)
OPEN 1, 5
PRINT #1, "DRUCKEN MIT CENTRONICS"
CLOSE 1

```

oder

```

SYS (INI)
OPEN 1, 5
CMD 1
PRINT "DIESES IST DER DRUCKER"
PRINT #1
PRINT "DIESES IST DER BILDSCHIRM"
CLOSE 1

```

Tabelle 4. Beispielprogramme, die die Schnittstelle ansprechen

Für START und die jeweiligen
Anfangsadressen des Pro-
gramms finden Sie in Tabelle 3.
Die Adresse 'INI' ergibt sich
immer aus $\text{Start} * 256 + 76$.
Für den nicht erweiterten
VC-20 also: $29 * 256 + 76 =$
7500.

Mit diesem Maschinenpro-
gramm läßt sich die Centro-

nics-Schnittstelle wie der einge-
baute Commodore-Druckeran-
schluß bedienen. Die Befehls-
folgen in Tabelle 4 zeigen ent-
sprechende Beispiele.

Retten auf Datasette

Nachdem Sie das Maschinen-
programm nun mit viel Mühe
in den Speicher 'gepopt' haben,
hier noch ein Tip für das Retten
auf die Datasette. Dazu ver-
wenden Sie bitte die Befehls-
sequenz in Tabelle 5. Die Befehls-
folge fügt zwei Nullbytes vor
dem Maschinenprogramm ein
und speichert es auf der Data-
sette ab. Die beiden Nullbytes
sind notwendig, damit der Zei-
lenlinker das Programm beim
Laden nicht 'umgräbt'.

Geladen wird das Programm
mit:

```

LOAD "C", 1, 1
SYS (INI)
NEW

```

Um das Ganze mit dem
c't-Sprachsynthesizer auszu-
probieren, könnten Sie einmal
den folgenden Text ausgeben
lassen:

```

CÖwwFMCjUHÖ
CCCCCCCjÜAMjkuMi
YÜCiÜjCYXHÜÜBCC

```

dabei bedeuten:

```

Ö = Commodoretast + '—'
Ü = Shift + '—'

```

Damit die Sprachplatine diesen
Text verarbeiten kann, muß der
Rechner in den Kleinschrei-
bungsmodus geschaltet werden.

```

POKE 43,254
POKE 44,START - 1
POKE 45,128
POKE 46,START
POKE (256*START)-2, 0
POKE (256*START)-1, 0
SAVE "CENTRONICS"

```

Tabelle 5. Befehlsfolge, um das
Programm abzuspeichern.

NASCOM-C

CP/M ist ein eingetragenes Warenzeichen der Firma Digital Research



Jumbo-Komfort für Ihren Tischrechner

bietet Ihnen das neue
CP/M 3.0 Betriebssystem!

NASCOM-C ist maßgeschneidert für CP/M3

Kompatibilität und Systemkomfort.

NASCOM-C ist ein neues deutsches System zur bestmöglichen und komfortabelsten Ausnutzung der modernsten CP/M-Software-Produkte.



Ergonomie für jedermann erschwinglich.

Ein System, das einiges auf dem Kasten hat und trotzdem viel preiswerter ist als so mancher Hcmccomputer, vor allem voll ausgebaut. Es entspricht den deutschen Vorstellungen von Benutzerfreundlichkeit, Service und Ergonomie. Ob Sekretärin, Handwerker, Student, Tüftler, Techniker oder Manager: Jeder findet in NASCOM-C einen zuverlässigen Freund, der allzeit bereit zur helfenden Routinearbeit übernimmt und seinen menschlichen Partner versteht anstatt ihn zu verappeln.

Ein System, das mitdenkt.

NASCOM-C versteht Sie, auch wenn Sie bisher nichts über Computer wissen, hilft Ihnen aber auch, Computer zu verstehen, hat keine Geheimnisse. Das geht so weit, daß Sie ihn auch selbst zusammenbauen und so von Grund auf kennenlernen können. Wer seinem NASCOM-C ein bißchen Zeit widmet, kann so viel Geld sparen und noch mehr lernen.



Gute Software braucht bessere Hardware!

NASCOM-C bedient sich des neuen CP/M-Plus-Betriebssystems, das bisher auf Mikros nicht bekannten Komfort mit der Kompatibilität zu seinen Vorläufern verbindet, für welche das größte Angebot an hochwertiger Software existiert. Aber was nützt die schönste Software, wenn die vorhandene Hardware deren Möglichkeiten nicht nutzen kann? Faule Kompromisse auf der Hardwareseite sind bei CP/M+ nämlich schwerlich möglich. Bessere Betriebssysteme brauchen nun mal modernere Computer. NASCOM-C ist ein für CP/M+ maßgeschneidertes Rechner, der allen alle Möglichkeiten von CP/M+ bietet, ohne Nerven, Geldbeutel oder Spezialkenntnisse zu überfordern.



Wer bietet mehr auf einer Karte?

- Z8C A/B Zentralsinheit mit 4/6 MHz Takt
- Speicherverwaltungseinheit (MMU) und DMA
- 128 Kilobyte Arbeitsspeicher mit Paritätsprüfung, erweiterbar auf 1 Megabyte
- Floppy-Disk Controller für alle 5 1/4- und 8-Zoll Laufwerke



- Festplattenschnittstelle vorhanden
- Ein DEC VT52, -HEATH H19- und ANSI-Aufwärtskompatibles Terminal mit Grafik in 8 Farben und lesbarem Zeichengenerator
- Zweifach V.24 und eine Centronics Schnittstelle zum Anschluß von Druckern, Plottern, Modems und Festrechnern
- Über RS-422 Schnittstellen zum Netzwerk erweiterbar
- Der 77-polige NASBUS macht NASCOM-C kompatibel zu vieler Erweiterungskarten (wie Farbgraphik mit 792*256 Punkten und 255 KB Speichererweiterungen)
- Farbgraphik kompatibel zu den Normen Tektronix 10XX, Plot-10 und GKS, umfangreiche Objektcode-Bibliotheken
- Jede wichtige Programmiersprache verfügbar
- Branchenlösungen, Text- und Datenbanksysteme in kaum überschaubarer Vielfalt
- Von der Leerplatte mit Dokumentation und Firmware bis zur Fertiggerätlieferung
- Der Grundbausatz einschließlich Dokumentation und Firmware kostet unter DM 1.000,-, als Leerplatte sogar nur DM 298,-!!

Wer lieber gleich den richtigen Computer kaufen will oder den Frust mit seinem jetzigen satt hat, bekommt für 2 DM sein NASCOM-C INFO-Paket direkt von

LAMPSON & ZERBE GmbH

Mikrocomputervertrieb

Odenwaldstraße 21-23

6087 Büttelborn · Tel. 06 52/5 67 30

Centronics-Interface für ZX 81

Wer die Ausdrucke des Sinclair-Computers nicht mehr auf dem vornehmen Silberpapier, sondern schlicht schwarz auf weiß, dafür aber in DIN-A4-Format haben möchte, stößt auf ungeahnte Schwierigkeiten. Vor den Anschluß eines handelsüblichen Druckers hat Sinclair einige Hindernisse gestellt. Soll also ein Drucker mit Centronics-Schnittstelle vom Sinclair gesteuert werden, muß ein Interface Vermittlerdienste leisten.

Der ZX81 bereitet alle nur denkbaren Schwierigkeiten bei der Realisierung dieser Schnittstelle. Die I/O-Ports können nicht ohne weiteres genutzt werden, da die vom Rechner genutzten Portadressen nicht sauber dekodiert sind. Weil der ZX81 den ASCII-Code nicht kennt, muß die Treiber-Software den Sinclair-Code übersetzen. Da die ZX-Software zur Druckersteuerung unerreichbar im ROM liegt, können Befehle wie 'LPRINT' nicht zur Steuerung der Centronics-Schnittstelle verwendet werden. Es ergeben sich also einige Probleme, die es zu lösen gilt.

Vier ICs führen zum Drucker

Die eigentliche Schnittstelle zwischen dem ZX81 und dem Drucker bildet ein PIO-Baustein (Parallel Input/Output). Dieser stellt drei 8-Bit-Ports bereit, wovon die Schnittstelle zwei belegt. Soll an die PIO kein Drucker angeschlossen werden, können die Ports natürlich frei verwendet werden. Aber auch, wenn ein Drucker betrieben wird, kann man über einen Port frei verfügen.

Die drei anderen ICs dekodieren die Portadresse der PIO. Aus dem eingangs erwähnten Problem ergibt sich, daß die PIO nicht als normaler I/O-Baustein angesprochen werden kann. Als Ausweg bietet sich an, den Port 'Memory-Mapped' zu betreiben. Das bedeutet, daß die PIO wie normaler Speicher angesprochen wird. Dies bringt zusätzlich den Vorteil, daß die Befehle 'PEEK' und 'POKE' zur Steuerung der PIO verwendet werden können. Als Nachteil ergibt sich, daß die Adresse, auf der die PIO liegt, nun nicht

mehr als Speicherplatz benutzt werden kann. Daß der Port Memory Mapped arbeitet, erreicht man dadurch, daß der Pegel der Leitung MREQ in die Adreßcodierung mit eingeht. Ist der Pegel dieser Leitung 'Low', zeigt die CPU an, daß sie einen Schreib-/Lesezugriff auf den Speicher vornehmen will.

Die Dekodierung der Portadresse geschieht also nach folgendem Schema:

MREQ
A15
A14
A13
A12
A11
A10
A9
A8
A7
A6
A5
A4
A3
A2
A1
A0
0001010011100002XX

Das entspricht den Adressen 29C0h...29C3h (10688d...10691d). Die Adressen A8 und A1 werden von der PIO selbst dekodiert, so daß sie in diese Auswertung nicht mit eingehen.

Um zu verhindern, daß auf dieser Adresse das ROM des ZX als 'Spiegelbild' erscheint (bedingt durch eine unvollständige Adreßdekodierung im Computer), muß immer, wenn die Portadresse selektiert ist, das ROM des Sinclair abgeschaltet werden. Das ermöglicht die Leitung ROMCS des Computers. Liegt hier ein Signal mit dem logischen Pegel 1 an, ist das ROM abgeschaltet. Für diese Abschaltung ist aber etwas Leistung nötig, da der Sinclair intern eine merkwürdige Schaltung hat: Der Ausgang des Steuerbausteins ULA liegt über einen 680-Ohm-Widerstand an dem CS-Eingang des ROMs. Selektiert die ULA das ROM durch ein Signal mit dem logischen Pegel 'Low' und soll das ROM gleichzeitig mit einem High-Signal an der Kontaktleiste (Pin ROMCS) 'deselektiert' werden, so fließt über den Widerstand ein Strom von

mindestens 3,5 mA. Diesen Strom bringt bei der Interface-Karte der Transistor T1 auf.

Nicht nur drucken

In diesem Artikel geht es zwar um die Frage, wie mit der PIO-Karte eine Centronics Schnittstelle realisiert werden kann; es ergeben sich aber daneben noch eine ganze Reihe nützlicher Anwendungen für die Platine.

Der Taster 'RESET' bewirkt einen Reset des Computers. Damit befindet sich der Rechner in dem Zustand, in dem er nach dem Einschalten der Stromversorgung ist.

An die Kontaktleiste an der Platine können alle üblichen Erweiterungen des ZX81 gesteckt werden, da die Belegung dieser Leiste der des Sinclairs entspricht.

Um die Ports der PIO für freie Anwendungen zu nutzen, muß man einiges über die Programmierung der PIO wissen. Hier

also ein Kurzlehrgang in Sachen PIO:

Die PIO 8255 verfügt über drei 8-Bit-Ports, die unabhängig voneinander als Ein- oder Ausgangsport definiert werden können. Dazu wird in das Steuerregister der PIO ein Steuerwort geschrieben, das die gewünschte Portdefinition ergibt (siehe Tabelle 1). Ist ein Drucker angeschlossen, sollte man darauf achten, daß Port A als Ausgang, Port C, Bit 0...3 als Eingang und Port C, Bit 4...7 als Ausgang definiert bleiben.

Steuerwort (dez.)	Port A	Port B	Port C Bit 0...3	Port C Bit 4...7
128	out	out	out	out
129	out	out	in	out
130	out	in	out	out
131	out	in	in	out
136	out	out	out	in
137	out	out	in	in
138	out	in	out	in
139	out	in	in	in
144	in	out	out	out
145	in	out	in	out
146	in	in	out	out
147	in	in	in	out
152	in	out	out	in
153	in	out	in	in
154	in	in	out	in
155	in	in	in	in

Tabelle 1. Steuerworte der PIO

	A	B
1	PA4	PA3
2	PA5	PA2
3	PA6	PA1
4	PA7	PA0
5	GND	PC7
6	+5V	PC6
7	PB3	PC5
8	PB4	PC4
9	PB5	PC0
10	PB6	PC1
11	PB7	PC2
12	PB2	PC3
13	PB1	PB0

Tabelle 2. Belegung der Port-Steckerleiste X1

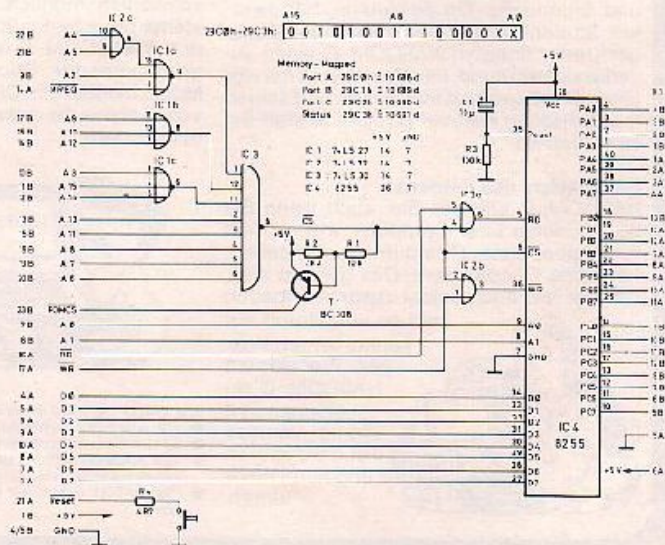


Bild 1. Die Werte der Bauteile können dem Schaltbild entnommen werden.

CP/M 86 f. PC/XT

205 DM

TEAC-Floppy-Disk-Laufwerke: Ganz neu im Programm, 3"-Laufwerke anschließbar wie 5"-Laufwerke, vollständig kompatibel.

3"-Laufwerke	FD 30 A	40 Spuren einseitig	550 DM
5"-Laufwerke	FD 55 A		640 DM
(alle in SLIMLINE-Ausführung)	FD 55 E		750 DM
	FD 55 F		740 DM
	FD 55 G	1,66 Mio. Byte	880 DM
			1040 DM

Genie 16, IBM-kompatibel 8085, 256 K, 2x360 KB Disk, HyRes-Graphik, flache freibewegliche Tastatur 5900 DM

Markendisketten ab DM 5,47 (10er Preis) mit Verstärkungsring. Wir liefern auch Steckverbinder, Bauteile, Drucker zu interessanten Preisen. Alle Preise incl. MwSt., Versand ab Straßenhaus. Fordern Sie unsere Infos an!!!

Dr. Aumann GmbH Computersysteme, Schulstraße 12, 5451 Straßenhaus
Tel. 02634/4081, Bürozeiten: montags — freitags 9—12/14—17 Uhr

Ihr Computer versteht auch Spaß Testen Sie es!!!

mit SUPERSPIELN für

ZX SPECTRUM

Mined Out	48 k(C) DM 29,—	Space Intruders	48 k(C) DM 29,—
The Hobbit	48 k(C) DM 78,—	Masterfile	48 k(C) DM 46,—
Penetrator	48 k(C) DM 37,—	Display	16/48 k(C) DM 20,—
Terror Daktil	48 k(C) DM 37,—	Matcalc	16/48 k(C) DM 39,—
Time Gato	48 k(C) DM 37,—	Monitor	16/48 k(C) DM 30,—
Jet Pac	16 k(C) DM 31,—	EdiAas	16/48 k(C) DM 35,—
RAISIN COMPIL FR von Rohrer			48 k(C) DM 69,—

Weitere Spiele für ZX 81, VC 20, VC 64, APPLE, IBM PC

HARDWARE

ZX Spectrum mit 48 k und deutschem Handbuch	DM 535,—
Centronics Interface für ZX Spectrum	DM 170,—
Zusatztastatur (Spectrum wird eingebaut)	DM 100,—
COMPETITION-PFO für Atar 400/800VC 20VC 64	DM 68,—
SUPERJOYSTICK mit Interface für ZX SPECTRUM	DM 129,—
ERGOTILT! für ergonomische Monitor-Unterstützung nach Ihren Monitor dreht u. kippt!!!	DM 98,—

Preise inkl. 14 % MwSt. Versand per Nachnahme

STEDE Spezialversand Postfach 12 66, 3542 Willingen

CEPAC-80

CMOS-Computer (wie in c't 1/84 beschrieben) mit Z80-kompatibler CPU, 3 Timern und 48 I/O-Leitungen.

Fertiggerät auf Europaplattform mit NSC 810 und 27C64 268,— DM

Flatline 100 x 86 mm	19,— DM	NSC 800	48,— DM
Flatline 100 x 160 mm	59,— DM	NSC 810	68,— DM
Eautellesatz		8155	15,80 DM
(Mit CPU [1 MHz] ohne Flatline, Speicher, Riot)	99,— DM	6116 LP-3	15,80 DM
Fertig aufgebaut,		2764	19,80 DM
Aufpreis	40,— DM	27C16	22,50 DM
		NE 590	11,80 DM

Alle Preise inkl. MwSt. Info gratis.

CONITEC Christian Lotter KG

Postfach 11 06 22, Schuchardstraße 4
6100 Darmstadt, Telefon (0 61 51) 2 60 13

Wir haben die Lösung

Floppy-Laufwerk

Halbspur und Disketten werden 40 Track
jetzt nur noch 699,— DM

Floppy-Controller

vorbereitet für bis zu 2x80 Track DS-Betrieb
jetzt für 199,— DM

APPLE®-kompatibel-Rechner

Profi-Version

großes Netzteil, großes Gehäuse, Pich AK 88 Tastat., Groß- und Kleinschreibung, deutsche Umlaute, jetzt nur 1499,— DM

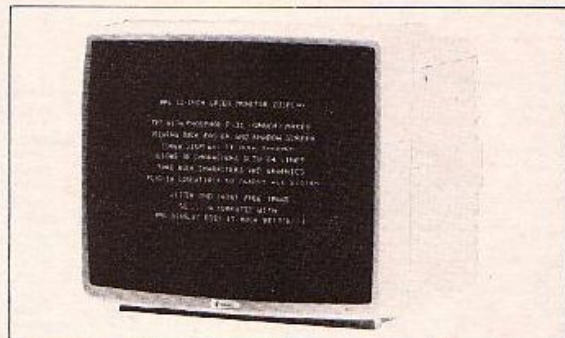
Teac-Floppy FD5E A 40 Track 639,— DM
Disketten für APPLE, 10 St. 62,— DM

Liste kostenlos!

UEDING ELECTRONICS
5750 Menden 1 · Holtewiese 2 · Telefon (0 23 73) 6 31 59

BMC Neue Produkte ab 1984:

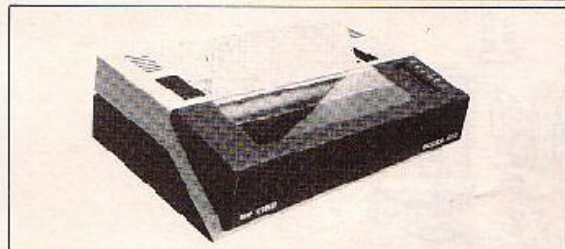
BM 8181	Farbmonitor RGB 640 x 240 Pkt. für IBM	DM 1584,60
BM 12 EN	hohe Bandbreite v. 20 MHz grün, entspiegelt	DM 498,—
BM 12 EY	Bandbreite > 18 MHz, bernstein	DM 438,—
BM 12 A	Bandbreite > 15 MHz, grün	DM 298,—



BMC-Monitor, einzigartig in Qualität und Leistung, mit Filterscheibe, > 18 MHz, grüner Hölzer, im formschönen Datenmonitor-Gehäuse

DM 12 ES = 398,— DM inkl. MwSt.

(349,12 DM netto) Dazu passend ERGOTILT 89,— DM inkl. MwSt.



Typenrad-Drucker mit 40 Z/s

Friktion + Trektor,
380 mm Papierbreite,
Acapionsmöglichkeit
für Sheet-Feeder,
Standard-8-bit-Schnittstelle
Optional RS 232 lieferbar

SD 1152:

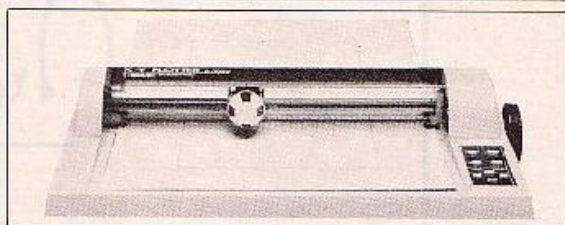
4390,— DM inkl. MwSt.
(netto 3850,88 DM)



Low-Cost-Typenrad-Drucker

16 Cps, Schreibbreite 335 mm, 96-7-Typenrad, Friktionsführung, mit Einzelblatteinzug, 8-bit-Schnittstelle 2-KB-Puffer

TD 16 = 2490,— DM inkl. MwSt.
(netto 2184,21 DM)



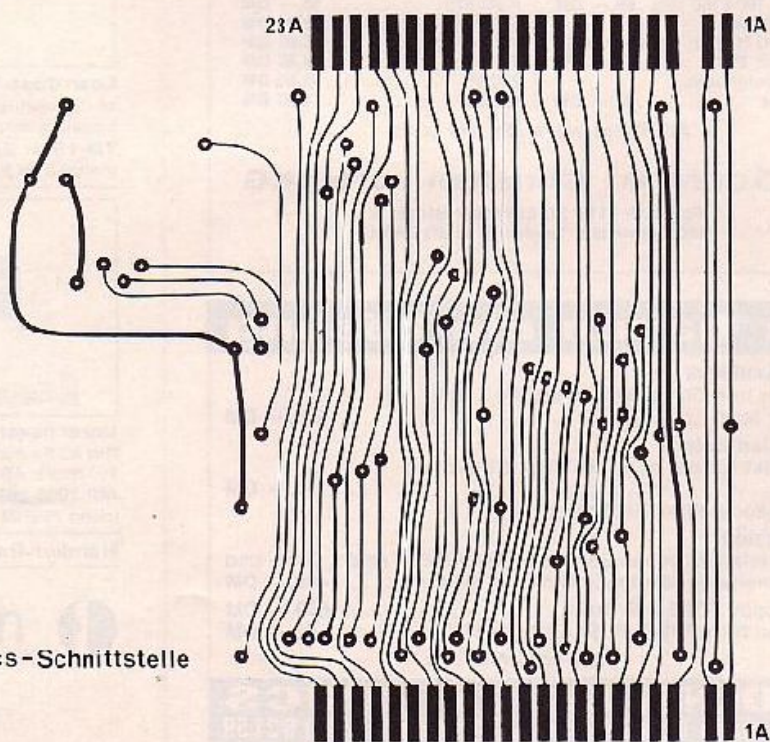
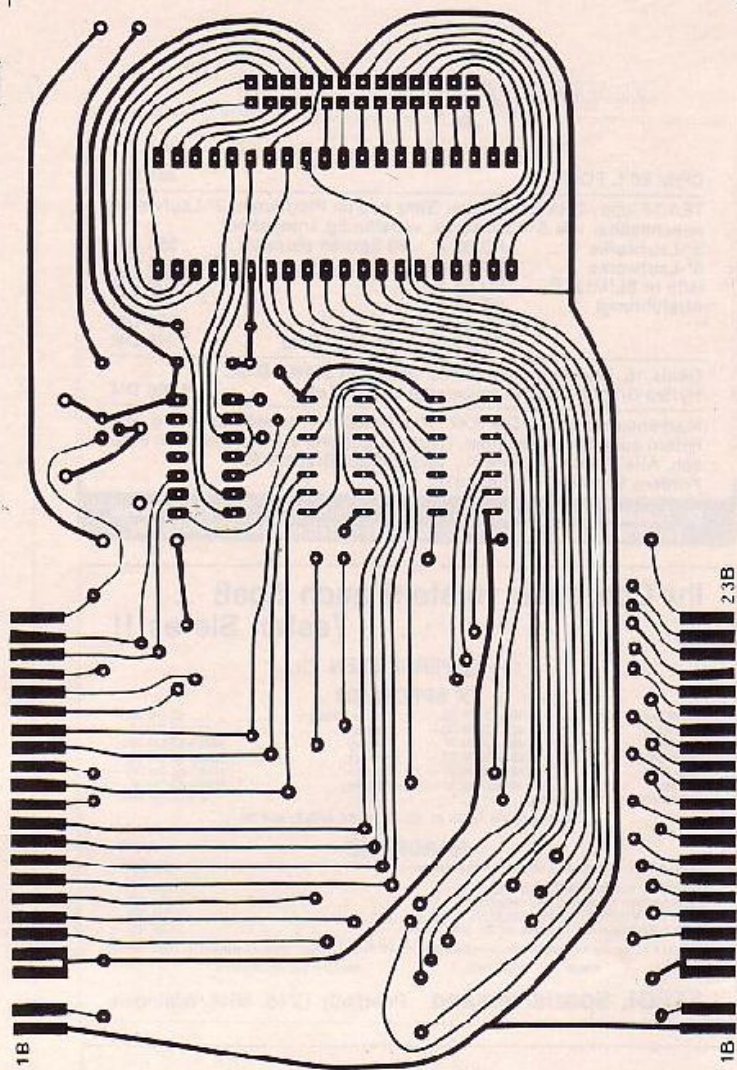
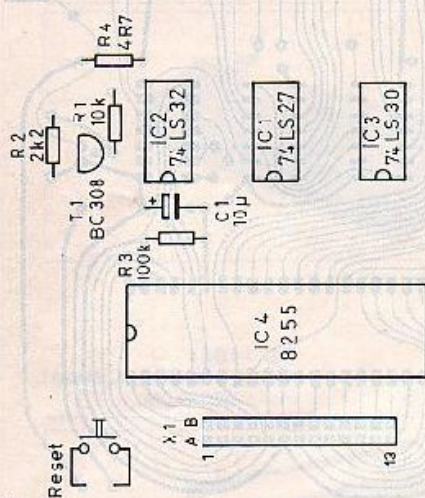
Unser neuer unschlagbarer 4-Farben-Plotter

DIN A3-Format, 0,1 mm-Genauigkeit, Schreibgeschwindigkeit 100 mm/s, ASCII-Zeichensatz u. Kreisfunktion! Optional Graph-ROM!
MP 1003 2690,— DM inkl. MwSt.
(netto 2353,64 DM)

Händler-Rabatte auf alle Produkte ab dem 1. Stück!

mirwald electronic **BMC**

Frauenstraße 8b, 8025 Unterschleißheim/München,
Telefon (0 89) 6 11 12 24, FS 5 213 476
Büro Frankfurt: Adalbertstr. 16
Telefon (06 11) 70 35 38



c't
PIO-Centronics-Schnittstelle

PEANUT-PROFI DS-64

Doppelprozessorsystem
mit 6502 und Z80 A

12 k B ROM, 64 k B RAM
für CF/M und Applesoftware
Tastatur mit eigenem Prozessor
10 Funktionstasten

PAL und FGB-Ausgang

Komplett-System mit 2 integr.
Slimline-Floppies DM 2948 00
mit 1 Laufwerk DM 2598 00

IEC-Bus Interface
(auch für Apple) DM 698,00

Sämtliche IF-Karten für Apple
z.B. 80-Zeichen DM 249,00
10 Kanal A/D + D/A-Wandler
DM 448,00

Sortierprogramm für VISICALC
SORTITEC DM 410,40

Original CP/M Software
zu Sonderpreisen

Info gegen 130 Porto
Händlerkonditionen auf Anfrage

IMETEC EDV

6200 Wiesbaden

Wasserrolle 2

☎ 06121/24622

FUTTER für den C64

Ihr Computer ist ohne Pro-
gramme wie ein Auto ohne
Benzin.

Gute und preiswerte Pro-
gramme für Ihren C64 bieten
wir mit dem SYNTAX-Pro-
gramm-Kassetten-Magazin.

Jeden Monat erscheint eine
Kassette mit 6 neuen, viel-
seitigen Programmen für In-
ren C64.

SYNTAX-Programme auf
Kassetten und Disketten
sind auch für die Commo-
dore CBM und VC 20 erhält-
lich. Nutzen Sie Ihr Gerät
verstärkt durch neue Ideen.

Fordern Sie gleich heute
noch unter Angabe Ihres
Gerätetyps kostenlose In-
formationen von

SYNTAX

Soft- u. Hardware GmbH

P.B. 18 68, 7550 Rastatt
Telefon (07222) 7 23 15

RITEMAN



Matrixdrucker 120 cps
parallel-Centronics-Schnittstelle,
bidirektional-Druck, graphikfähig
360 dpi. DM 1150,-

Drucker von C. ITOH.

8510 A Matrixdrucker 120 cps
parallel mit dt. Handbuch für Endlos
und Einzelblatt DM 1400,-

8510 S Matrixdrucker 180 cps
freiladbarer Zeichensatz und 5 zu-
sätzliche Zeichensätze DM 1999,-

Super-Graphik-Interface
für alle Itoh-Geräte in parallel, 32sei-
tiges Interface-Handbuch für Apple-
Computer II kompatibel DM 199,-

BASE-48

„Apple-kompatibler“ Computer 48 K
mit Groß- und Kleinschrift, original
Gehäuse – Handbuch. DM 999,-

Elektronik- Vertrieb KÖLLER

COMPUTER UND ZUBEHÖR
Lotte, Niesetalstraße 4
4938 Schiede-Schwalenberg
Telefon 052 31 77 50

Preise inkl. MwSt.
Versand per NN plus Porto und Verpackung

ZX 81 und ZX Spectrum Zubehör von Logitek

Druckerinterface für ZX Spectrum
Anschlußfertig für fast alle anhaltende
Drucker wie EPSON, STAR, CP80 usw.
LPRINT, LIST und COPY sind sofort verfü-
bar. Viele zusätzliche Druckfunktionen sind
in BASIC ansprechbar, wie wählbare Schrift-
größe und Schriftart, direkte ASCII-Ausgabe
über LPRINT, Erweiterung des Spectum-
zeichensatzes mit deutschen Zeichen, in Ver-
bindung mit der 80 K Speichererweiterung ist
während des Drucks zugleich ein Weiter-
arbeiten am Rechner möglich.
Komplett mit Kabel DM 275,-

LOGITEK Spectrumgehäuse
Dieses formstabile schwarz eloxierte Alumi-
niumgehäuse nimmt den Spectrum mit Netz-
teil und Busplatine für 5 Erweiterungen auf.
Mit EIN/AUS-Schalter.
Gehäuse komplett mit Bus
DM 238,-
Busplatine für 5 Karten DM 118,-

Speichereverweiterung von 16 K auf 80 K zum
Einstecken in den Spectrum. DM 228,-
(Bitte bei Bestellung Issue 2 oder 3 angeben!)

32 Bit Portmodul
Über die 32 Leitungen lassen sich elektroni-
sche Steuer-, Regel- und Meßschaltungen an-
schließen, die digitale Ein- und Ausgänge
haben.
Für ZX-Spectrum und ZX-81 DM 138,-
Spectrumstecker DM 14,-
Gegenteil dazu DM 7,-

64 K RAM Modul für ZX-81
schwarz elox. Gehäuse, nach an den ZX-81
ansteckbar, Port durchgeführt DM 238,-
ZX-81 Stecker DM 12,-
Gegenteil dazu DM 6,-

Räumliche Grafik für ZX Spectrum
Mit diesem Programm lassen sich mathema-
tische Funktionen zum ich darstellen und
anschaulich erklären. DM 38,-

Weitere Programmkassetten auf Anfrage.
Datenlochkassette C30 DM 2,-

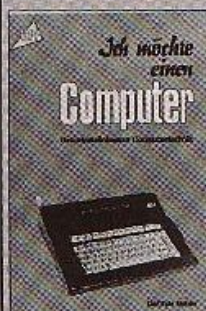
Deutsche Beschreibungen werden mitgelie-
fert. Preise incl. MwSt. Versand per NN zzgl.
6,50 DM. Porto und Verpackung ab Lager
Berlin.

LOGITEK

Hoff und Lesser GbR
Pankstraße 49, 1000 Berlin 65
Telefon (030) 4 61 64 92

TOPP

aktuell



Best.-Nr. 354
D. Böhm
Ich möchte einen
Computer
DM 10,80



Best.-Nr. 355
D. Böhm
Computergesteuerte
Meßtechnik
DM 25,80



Best.-Nr. 428
J. Kwiatkowski
FORTRAN
in 8 Lektionen für Anfänger
DM 29,80



Best.-Nr. 455
C. Abeldt
BASIC –
Grundlagen und Beispiele
DM 9,-



Best.-Nr. 401
Rowley
Atari BASIC
DM 10,80



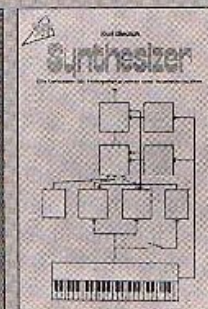
Best.-Nr. 496
Libos/Wahl
Personal Computer
Handbuch
DM 19,80



Best.-Nr. 475
M. Reinhard
Mikroprozessor SCMP II
DM 19,80



Best.-Nr. 420
L. Schüssler
Fernsteuern –
Sonder- u. Zusatzfunktionen
DM 19,80



Best.-Nr. 424
K. Dierich
Synthesizer
DM 14,80

Diese informative Buchreihe von TOPP
wird ständig durch weitere Titel ergänzt,
die zum Verständnis unserer Tage beitragen.
Prospekte über die Gebiete Elektronik und Amateurfunk
stehen kostenlos zur Verfügung – Bitte anfordern.

frech-verlag

Turbinenstr. 7, 7000 Stuttgart 31

Telefon (0711) *83 20 61, Telex 7 252 156 fr d

Die Software für den ZX81

Port B wird für die Druckersteuerung nicht gebraucht, steht also dem Benutzer für andere Zwecke zur Verfügung. Es kommen also die Steuerwörter 129 und 131 in Betracht, wobei wir für das vorliegende Programm 129 (81h in Zeile 9) ausgewählt haben. Bit 0 von Port C soll als BUSY und Bit 7 als DATA STROBE angeschlossen werden.

Zum Betreiben der Centronics-Schnittstelle können die Druckerbefehle des ZX81 leider nicht verwendet werden. Diese arbeiten im Grafik-Mode und gehen genau wie bei der Bildschirmansicht Pixel für Pixel aus. Dabei ist die Druckzeile auf 32 Zeichen beschränkt.

Das Programm PRCENT erzeugt aus einem ZX81-String einen ASCII-String. Alle ASCII-Zeichen können ohne CHR\$-Befehl aus dem ZX81-Zeichensatz generiert werden (Tabelle 3). Als Besonderheit enthält der ZX81-Zeichensatz nur das Pfund-Symbol — es handelt sich ja schließlich um ein britisches Gerät — und die Blockgrafik-Zeichen. Diesen Symbolen sind in der Umform-

tabelle andere ASCII-Zeichen zugeordnet. Die Inversbuchstaben des ZX werden als Kleinbuchstaben interpretiert. Der Rest wird auf Sonder- oder Steuerzeichen abgebildet. Für den D-Drucker sind dabei das inverse \$ als 'Carriage Return' und das Blockgrafikzeichen über graphic shift F als 'Linefeed' von Bedeutung. Diese Zeichen müssen also als Abschluß einer Printzeile in dem Ausgabestring auftauchen. Für die Ansteuerung des c't-Sprachsynthesizers sind diese Steuerzeichen natürlich nicht nötig.

Die Stringeingeabe erfolgt mit

```
PRINT CHR$USR
31488 = "....",
```

wobei der String auch in einer Variablen enthalten sein kann, zum Beispiel:

```
PRINT CHR$USR
31488 = X$
```

Für die Eingabe des Maschinenprogramms empfiehlt es sich, den Mikroassembler aus dem SuperTape-Artikel im gleichen Heft zu verwenden. In Spalte drei des Assembler-Listings befindet sich dann die Zeilennummer der REM-Statements, und Spalte zwei enthält der Hexcode. Als Startadresse ist 7B00 für die Ausgaberroutine PRCENT gewählt. Vor der Eingabe und vor jedem Laden von der Kassette sollte dieser Bereich mit

```
POKE 16389,123
NEW
```

geschützt werden.

Das Programm hat vier Aufgaben zu erfüllen: String-Startadresse und -länge zu bestimmen, den String abzuwerten, das jeweilige Zeichen in ASCII umzuwandeln und schließlich an die Schnittstelle auszugeben. Die ASCII-Zeichen werden bei Buchstaben und Steuerzeichen berechnet; bei Sonderzeichen dient eine Tabelle zur Umwandlung.

Die Druckerausgabe folgt im wesentlichen dem Flußdiagramm in dem Artikel über parallele Schnittstellen in c't 1/84. Auf die PAPER OUT-Abfrage wurde dabei verzichtet, dafür aber eine BREAK-Abfrage mit einbezogen.

	0	1	2	3	8	9	A	B
0	SP	<	4	K	NUL	DLE	'	X
1	a	>	5	L	SOH	DC1	a	I
2	o	<	6	M	STX	DC2	o	M
3	>	>	7	N	CTX	DC3	o	N
4	a	+	8	O	EOT	DC4	B	C
5	+	+	9	P	ENQ	NAK	DEL	P
6	1	-	A	Q	ACK	SYN	a	q
7	h	*	B	R	BEL	ETB	b	r
8	z	/	C	S	BS	CAN	c	s
9	;	:	D	T	HT	EM	d	t
A	\$,	E	U	LF	SUB	e	u
B	"	.	F	V	VT	ESC	f	v
C	!&	0	G	W	FF	FS	g	w
D	\$	1	H	X	CR	GS	h	x
E	!	2	I	Y	SO	RS	i	y
F	?	3	J	Z	SI	US	j	z

Tabelle 3. Umwandlung ZX-Code in ASCII (deutscher Zeichensatz)

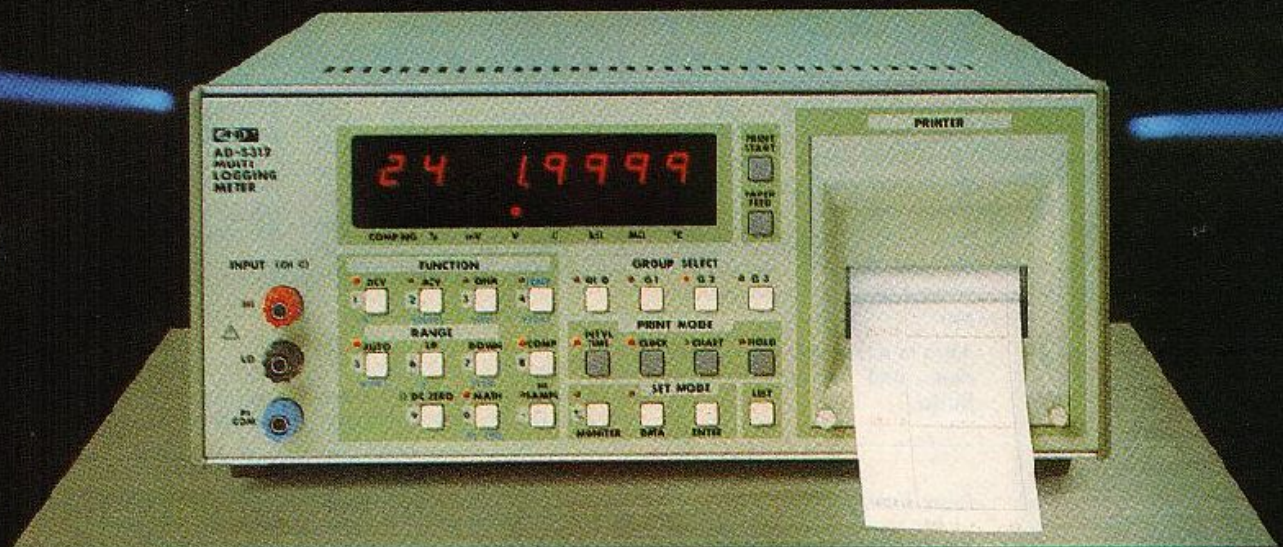
(7B00)	0000	0002	ORG	7B00H	
	0003	0004	PRCENT	PRCENT	
7B00	E7	0050F	RET	20H	ÜBERSPRINGT " "
7B01	CD550F	0007	CALL	0F55H	TEST, DE-STRINGADP,
7B04	CD5513	0008	CALL	1F55H	BC-LÄNGE
7B07	3E81	0009	LC	A,81H	SETZT PORTKONFIG
7B09	32C329	0010	LE	(2F02H),A	
7B0C	3D	0011	DEC	A	SETZT STROBE-FLAG
7B0D	32C229	0012	LE	(2F02H),A	
7B10	1A	0013	LOOP	A,(DE)	CHAR, VOM STRING
7B11	CD1F7B	0014	CALL	ZWASCI	ASCII-WANDLUNG
7B14	CD787B	0015	CALL	LPRCHR	PORTAUSGABE
7B17	13	0016	INC	DE	INKR. STRINGADR
7B18	0B	0017	DEC	DE	DEKR. STRINGADR
7B19	7B	0018	LC	A,B	
7B1A	B1	0019	OR	C	
7B1B	CD787B	0020	CALL	LPRCHR	
7B1D	CF	0021	RET	0EH	SPRUNG, FALLS <0
7B1E	FF	0022	DEFB	0FFH	ENDE DES AUFRUFS MIT
		0023			ERR-CODE = 0
		0024			
		0025	PRCENT	ZWASCI	xxxxx
7B1F	F003	0026	ZWASCI	CF	
7B21	2B26	0027	JR Z,	4FH	TEST AUF
7B23	CB87	0028	RES	6A	ANFÜHRUNGSZEICHEN
7B25	FE00	0029	CF	8CH	TEST AUF NORMAL ODER
7B27	3B0C	0030	JR C,	NORMAL	INVERS
7B29	D645	0031	CF	50H	
7B2B	FE41	0032	CF	61H	RETURN FALLS KLEIN-
7B2D	00	0033	RET NC		BUCHSTABE
7B2E	D63B	0034	SUB	3EH	
7B30	FE20	0035	CF	20H	RETURN FALLS
7B32	00	0036	RET C		Sonderzeichen
7B33	1B07	0037	JR	SONDER	SONST: SONDERZEICHEN
7B35	C61B	0038	NORMAL	4FH	
7B37	FE41	0039	CF	41H	RETURN FALLS
7B39	D0	0040	RET NC		BUCHSTABE
7B3A	D61B	0041	SUB	11H	
7B3C	E5	0042	SONDER	PUSH	
7B3D	CA	0043	PUSH	HL	SONDERZEICHEN
7B3E	21407B	0044	LD	HL,7AB	
7B41	0A00	0045	LD	B,09H	TEST, DES ASCI-
7B43	09	0046	ADD	HL,BC	ZEICHENS AUS DER
7B45	7E	0047	LE	(HL),C	TABELLE
7B46	01	0048	POP	BC	
7B47	E1	0049	POP	HL	
7B48	C9	0050	RET		
		0051			
7B49	0C32	0052	AND	A,22H	ANFÜHRUNGSZEICHEN
7B4B	C9	0053	RET		
		0054			
7B4C	20B5C0D	0055	TAB	DEFB	20H,5BH,5CH,5DH
7B4D	5E5F2123	0056	DEFB	5EH,5FH,21H,22H	
7B4E	20D74022	0057	DEFB	20H,27H,40H,22H	
7B4F	26243A3F	0058	DEFB	24H,24H,3AH,3FH	
7B50	2A2A2A2A	0059	DEFB	2AH,2AH,2AH,2AH	
7B51	302B202A	0060	DEFB	30H,2EH,20H,2AH	
7B52	2F30C0C1	0061	DEFB	2FH,3EH,2CH,2EH	
7B53	29114C44	0062	DEFB	29H,21H,20H,20H	
7B54	34303437	0063	DEFB	34H,34H,34H,37H	
7B55	3B3B417B	0064	DEFB	3BH,3BH,41H,7BH	
7B56	7C707070	0065	DEFB	7CH,7CH,7EH,7EH	
		0066			
		0067			
		0068			
		0069			
		0070			
		0071			
		0072	PRCENT	LPRCHR	xxxxx
7B7B	E5	0073	LPRCHR	PUSH	HL
7B7D	F0	0074	PUSH	AF	
7B7E	21C229	0075	LD	A,7FH	HLADR, VOM PORT C
7B7F	3E7F	0076	LD	A,7FH	TEST AUF BREAK
7B81	1F	0077	RRA		
7B82	300F	0078	JR NC,	BREAK	
7B84	7E	0079	LD	A,(HL)	
7B85	1F	0080	RRA		TEST AUF BUSY
7B86	3A0F	0081	JR C,	BUSY	BIT 0 VON PORT C
7B88	F1	0082	POP	AF	AUSGABE VON A
7B89	20B5C0D	0083	LD	(2F02H),A	AN PORT A
7B8C	AF	0084	XOR	A	STROBE AUF NULL
7B8D	77	0085	LD	(HL),C	
7B8E	3E60	0086	LD	A,60H	NULL AUF EINS
7B90	77	0087	LD	(HL),C	
7B91	F1	0088	POP	HL	
7B92	C9	0089	RET		
		0090			
7B93	CF	0091	BREAK	RST	0H
7B94	0C	0092	DEFB	0CH	BREAK MIT
		0093			ERR-CODE 10H

Bild 3. Druckerausgabe eines ZX-String in ASCII

Bild 2. ZX81 druckt den Anfang des Druckprogramms

Ein neues Konzept der Labordaten- Messung und Aufzeichnung

AD-5312 MULTILOGGER



- 24 Kanäle, ACV, DCV, OHM, Thermoelemente und Pt-100
- programmierter Scanner und Drucker/Plotter für Datenregistrierung
- integrierte Intelligenz, 5 Mathematik-Funktionen und Komparator

Siber Kikai K.K.
Zweigmeterfassung, Deutschland
Resseltstr. 2 6030 Frankfurt/M. 1
Telefon: 06 11/7171 21
Telex: 411 380 shf. d



Zu besichtigen auf der Hannover-Messe, Halle 13, Stand 911

Geppert, Kuhn, Marcus, Ney, Sickert

Automatische Spracheingabe und Sprachausgabe

Analyse, Synthese und Erkennung menschlicher Sprache mit digitalen Systemen.

432 Seiten, Hardcover, 275 Abbildungen, DM 73,—
Verlag MARKT & TECHNIK
— Geschäftsbereich Fachliteratur —
8613 Haar bei München.
Erscheinungsjahr 1983.
ISBN 3 922120-31-8

Das Buch behandelt die Vorgänge beim Sprechen und Hören, die Grundlagen der digitalen Signalverarbeitung und die für die Analyse und Synthese von Sprachsignalen üblichen Methoden. Dieses neue und umfangreiche Gebiet wird zum ersten Mal geschlossen in deutscher Sprache dargestellt.



Das Buch setzt allerdings einige mathematische Grundkenntnisse voraus. Als Leserkreis dürften deshalb vor allem Ingenieure und Studenten in Betracht kommen. Wegen der umfassenden Darstellung eignet sich das Buch als Nachschlagewerk ebenso wie zum Selbststudium. Durch die zahlreichen Literaturhinweise können sich alle Interessierten über spezielle Fragen weitergehend informieren.

Die Grundbegriffe der digitalen Signalverarbeitung werden, vom Abtasttheorem bis zur Z-Transformation, knapp aber gründlich behandelt.

Die akustischen und linguistischen Grundlagen, insbesondere die Mechanismen des Hörens und Sprechens, stellen die Autoren in anschaulicher Form dar.

Die bei der Sprachanalyse und Sprachsynthese vorgestellten Methoden sind auf dem neuesten technischen Stand. Computergesteuerte Sprachsynthesizer sind seit einiger Zeit im Einsatz. Die automatische Spracherkennung dagegen ist wegen des komplexen Charakters der Sprache noch in den Anfängen, aber bereits bei weitem das umfangreichere Gebiet. Die Verfahren der Mustererkennung, die Sprach- und Sprechererkennung nehmen daher einen breiten Raum ein.

So bietet das Buch die Verknüpfung der akustischen Grundlagen mit der digitalen Signalverarbeitung. Es ist in einer klaren und treffenden Ausdrucksweise geschrieben. Die diversen Abbildungen und Diagramme verdeutlichen den Text und liefern einen wesentlichen Beitrag zum besseren Verständnis.

Detaillierte Produktübersichten und ca. 400 Literaturhinweise ergänzen die Darstellungen.

Alles in allem also ein erfreuliches und empfehlenswertes Buch. AT

Peter Kahlig

Programmieren von Mikrocomputern Assembler-Programmierung von Mikroprozessoren (8080, 8085, Z 80) mit dem ZX 81

Braunschweig: Vieweg 1983
186 S., 100 Programmbeispiele
Kart. DM 38,—
ISBN 3-528-04238-9

Anwender von Mikrocomputern, die in der BASIC-Programmierung bereits erfahren sind, haben oftmals den berechtigten Wunsch, maschinennah, nämlich auf Assembler-Ebene, zu programmieren.

Das Buch ist, auch wenn der Titel es manchmal flüchtigen Betrachter suggerieren mag, kein Assembler-Lehrgang für ZX81-Besitzer, die erste Gehversuche in Maschinensprache erwägen. Es ist vielmehr eine Sammlung einfacher 8-Bit-Arithmetikroutinen, also von Programmen für die vier Grundrechenarten (einschließlich der Kehrwertbildung) für



Dr. Peter Albrecht

Das Datenbanksystem dBase II

Haar bei München, 1983
Verlag Markt und Technik
284 Seiten, kart. DM 68,—
ISBN 3-922120-36-9

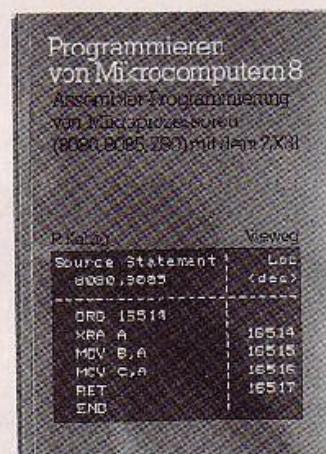
Ein erfolgreiches Softwarepaket für die Datenverarbeitung wird in diesem Buch von einem Fachmann vorgestellt, der sich nicht nur mit Datenbanken auskennt, sondern auch in der Lage ist, sein Wissen an der Leser weiterzugeben. Nach einer knappen und präzisen Ein-

führung in die Befehle von dBase II kommt der Autor schnell zum Entwickeln von dBase II-Programmen.

In kurzer, verständlicher Beispielerklärung erklärt er Schleifen, Prozeduren und spezielle Programmiertricks wie zum Beispiel den Schutz eines Anwenderprogrammes durch Passwörter und dialoggesteuertes Erzeugen von Programmen. Darüber hinaus zeigt er die Stärken (seltener die Schwächen) von dBase auf, besonders, wenn es um die Verknüpfung von Mikrocomputern mit großen Datenbanksystemen geht. Er beschreibt unterstützende Software und bietet eine knapp und gerade deshalb gut zu lesende Befehls-, Funktionsübersicht und einen wohl sortierten Index. Ein Buch, das sowohl als Einführung für den Newcomer als auch als Nachschlagewerk für den Datenverarbeiter sehr gut geeignet ist.

AB

natürliche Zahlen kleiner 256. Alle Assembler-Routinen werden parallel für die Mikroprozessoren 8080/8085 und Z 80 erstellt. Verkürzte Programmversionen für den Z 80, der den leistungsfähigeren Befehlssatz hat, sind, wo dies möglich ist, aufgeführt.



Besitzern des Kleincomputers ZX 81 ermöglichen die geistigsten BASIC-Programme, alle Arithmetikroutinen hexadezimal einzugeben, abarbeiten zu lassen und sich die Ergebnisse auf dem Bildschirm anzusehen.

Betrachtet man Programmierung als Umsetzung verschiedener Rechenvorschriften (Algorithmen) in eine für den Com-

puter verständliche Form, so muß das Fehlen jeglicher Ablaufpläne bemängelt werden. Sind die den Arithmetikprogrammen zugrunde liegenden Algorithmen dem Leser nicht bekannt, so muß er sie sich mühsam aus den Assembler-Listings erarbeiten. Obwohl dies durch eine gute Kommentierung der Programme unterstützt wird, sollte der Leser zumindest die Grundrechenarten im Dualzahlensystem beherrschen. Zudem muß er die wichtigsten Assemblerbefehle und ihre Wirkungsweise kennen.

Um die Verschiedenartigkeit und Vielfältigkeit von Programmierlösungen aufzuzeigen, werden zahlreiche Beispiele aus der Sekundärliteratur modifiziert. Kahlig arbeitet jedoch die einzelnen Kriterien nicht in einer allgemein verständlichen Form heraus. Dem schon reichlich vorhandenen Programmangebot werden lediglich die geänderten Listings angefügt. Der schlechte Programmierstil einiger Programme, zum Beispiel die Verwendung spezieller Befehle entgegen ihrer eigentlichen Funktion, wird sogar als trickreich hervorgehoben.

Empfehlenswert nur für ZX 81-Besitzer mit beträchtlichen Vorkenntnissen. US

**Auch wir
sind in**



Hannover

ct magazin für
computer
technik

**Besuchen Sie c't
auf der
Hannover-Messe
4. — 11. April 1984
Halle 12, II. OG,
Stand 3200**

Drucker

gibt es viele
Typenrad- u. Nadeldrucker
von: Epson, Olivetti, Nec,
Qume, Seikosha, Star,
Brother, Juki, Mannes-
mann, Phillips ...

Doch

den Richtigen für SIE und
Ihr Problem gibt es bei
uns. Über Superpreise
reden wir nicht. Wir
haben Sie!

Computer

große und kleine, be-
kannte, weniger bekannte,
ob Televideo, Epson,
Genie, Kaypro, Olivetti,
ob unser ASS 'Axio', ...
Fragen SIE nach Ihrem
Computer, er wartet
darauf, Ihnen sein
Können zu demonstrieren.
Geballte Leistung zu
unglaublichen Preisen.

Software

darüber reden viele,
wir haben die, die
Ihre Probleme löst.

Zubehör

Disketten-Kassetten-Bän-
der, Papier-Boxen-Farb-
bänder, traumhafte Preise

Wirklich nicht zu glauben,
daraus:

**Anwender — Freaks —
Händler**

für jeden das RICHTIGE
zu Preisen??

Drucker Computer Zubehör



S.T.A. Data Control Corp.
Türmergasse 25 · 6900 Heidelberg
Telefon 0 62 21 - 78 05 55

STA



**Computer~
Versand**
Kolf W. Neumann

Alles für den Home-Computer

**NEWMAN
BERATUNGS-
KATALOG**

Gleich anfordern!

Kostenlos und unverbindlich erhalten
Sie den großen Home-Computer
Beratungs-Katalog mit vielen tollen
Angeboten. Einfach Coupon einsenden.

GRATIS

Wir haben nur Qualitäts-Markenartikel zu Preisen, die uns so
leicht keiner nachmacht.

Wir sind ein Versandhaus nur für Home-Computer, Hardware,
Programme und Zubehör.

Wir beraten Sie neutral und unverbindlich. Am besten rufen Sie
uns an. Wir helfen Ihnen weiter.

Wir liefern in der Regel innerhalb von 8 Tagen. Sofort-Liefer-
Erschließung bei telefonischer Anfrage.

Und außerdem erhalten Sie die Original-Hersteller-Garantie auf
alle Artikel.

Teilzahlung ab sofort auch möglich.

Commodore 64

Newman liefert alles sofort, wer kennt Ihnen das sonst noch
bieten! Die gesamte Peripherie original von COMMODORE
sofort ab Lager lieferbar. Und das zu Preisen, die uns so leicht
keiner nachmacht. Fragen Sie unbedingt nach unseren Paket-
Angeboten. Sie werden staunen.

Neu! Commodore SX 64

tragbarer "C 64" 64 K, 170 KB, - Floppy und
Farb-Monitor zum Sonder-Preis von **DM 2.948,-**
(kein Druckfehler)

Sharp MZ - 721

64 K, integrierter Cassette-Recorder,
10 Spielprogramme gratis **nur DM 939,-**
(da nur begrenzte Stückzahl sofort lieferbar, bitte noch heute
bestellen).

Sharp MZ - 731

wie MZ - 721, zusätzlich mit
integriertem Vierfarben-Drucker, **nur DM 1.239,-**
20 Programme am Lager.

Sinclair ZX Spectrum, 394,90
16 KB, RAM **DM 394,90**

Sinclair Spectrum,
48 KB, RAM **nur noch DM 539,-**

Dragon 32
32 K-RAM Super-Graphic **DM 795,-**

Seikosha GP 100 A,
Matrix-Drucker, 50 Zeichen/sec. **DM 675,-**

Sanyo, Later-Monitor, 2112,
restschon scharf, grüne Anzeige **DM 299,-**

Anzeige in orange-farben, 2112 **DM 309,-**

Spectravideo, SV 318 **DM 888,-**

Spectravideo, SV 328 **DM 1.248,-**
(die gesamte Peripherie ist auch lieferbar).

Epson-Drucker

besonders preiswert, z.B. RX 80, nur **DM 1.148,-**
Außerdem lieferbar: Texas Instruments, Brothers und jede
Menge Fachbücher, Spiel- und Lern-Programme, Drucker, Lauf-
werke, Monitore und und und ...

Am besten gleich nachfragen: **040/830 26 27**
040/830 28 29

Ausschneiden auf Postkarte kleben (60 R Porto)

Ja, ☐ bitte senden Sie mir sofort kostenlos und
unverbindlich Ihren Beratungs-Katalog.

Für Ihre Bestellung bitte hier eintragen. Alle Preise incl. MwSt. zuz. Versand- kosten. Lieferung per Nachnahme, Teilzahlung ab sofort möglich!			Name/Vorname	
Artikel			Straße	
Stück			PLZ/Ort	
Preis			Vorwahl/Telefon-Nr.	
			Unterschrift	
			Alter	

NEWMAN Computer-Versand
Kolf W. Neumann, Postfach 5712 61, 2000 Schenefeld.

Commodore CBM-64

FORTH 64

mit Grafik & Sound
Modul 198,- DM

mod 64®

das Grafikmodul 179,- DM

IEEE - 488

CBM Schnittstelle 228,- DM

A - CALC

Electronic Spreadsheet
Disc (Dt. Anl.) 146,- DM

Light Pen 85,-

ROM Board 98,-

5-fach Modul Steckplatz 198,-

BÜCHER, SPIELE

Info 1/84 gegen 2,- Briefmarken

bst *compulronic*

Burgstraße 126a
6000 Frankfurt 60
Tel. 0611-453857



OPERATOR

OPERATOR, die neue universell einsetzbare Eingabe-einheit, ist speziell konzipiert worden, um die Flexibilität moderner Microsysteme zu unterstützen.

1. Das Haupttastenfeld: Tastenanzahl: 64 Tasten. „ASCII“ - „deutsch“, Zeichensatz, 4 Cursor-tasten, Auto-repeat auf allen Tasten, durch Repeat-taste beschleunigbar, Alphablock-Taste.

2. Das Zehnerfeld: Tastenanzahl: 8 Tasten
Nummern 1 bis 9 sowie 0 00 . % - + : RETURN

3. Das Funktionstastenfeld: Tastenanzahl: 16 Tasten
ungeschliffen, 15 Tasten geschliffen

Tasteneinhalt: standardmäßig vorprogrammiert für bekannte Rechnersysteme oder auf Kundenwunsch, z. B. für Personalcomputer, Industriesteuerung, Textverarbeitung, (WORDSTAR etc.) usw.

Definieren Sie sich Ihre eigene Tastatur

Alle Tasten des Zehnerfeldes, die Funktionstasten u. die Cursor-tasten sind mit einem zusätzlichen EPROM 2, 4 od. 8 K. selber umzudefinieren. Ausf. 1 Interlagen inkl. Schnittstellen: parallel + seriell, V24 od. TTL, Handshake serienmäßig

AFC computer GmbH, Salmstr. 20, 5000 Köln 91,
Tel. 02 21 33 80 00, Tx. 8 873 254

ASC-COMPUTER-SHOP

HIRSCHGRABEN 9 11 5100 AACHEN

02 41/2 52 26

ASC-Comp. 48 K, APPLE-komp.	1150,-
dito, mit 10er-Tastatur	1298,-
dito, 64 K, Z80-CPU - 6502	1390,-
dito, mit 10er-Tastatur	1548,-
MULTIHEBQUAD 48 K, bestückt	730,-
NETZTEIL 5 A	198,-
16-K-Karte, geprüft	145,-
LANGUAGE-Karte, geprüft	155,-
Z80-CPU-Karte, geprüft	155,-
80-Zeichen-Karte, geprüft	228,-
PAL-Karte, geprüft	198,-
PRINTER-Karte m. Anschluß	198,-
CONTROLLER I DISK-DRIVE	168,-
RS-232-SCHNITTSTELLE	268,-
HF-MODULATOR	39.90
DISK DRIVE	798,-
JOY-STICKS, 4 Tasten	47.50
Leertasten, 16 K usw.	39.90
ALPHATRONIC PC	1495,-
CP80-Nadelldr., EPSON-komp.	998,-
JUKI-6100-Typenradrunder	2350,-
MONITOR, 12", grün, entsp.	278,-
MONITOR, 12", orange, entsp.	298,-
HCB-MONITOR, 12"	998,-

FORDERN SIE UNSERE PREISLISTE AN!

c't-Einzelheft-Bestellung

c't können Sie ab Ausgabe 12/83 direkt beim Verlag nachbestellen. Den Betrag von DM 6,- (zuzügl. DM 2,- für Porto und Verpackung) überweisen Sie bitte auf eines unserer Konten* oder fügen Sie Ihrer Bestellung einen Vorkontierungsscheck bei.

Bei Bestellungen bis zu 4 Heften erhöhen sich die Kosten für Porto und Verpackung auf DM 3,-.

* s. Seite 16



Der **INTERFACE AGE**

Musik-Synthesizer

für den Commodore 64

EXTENDED SYNTHESIZER SYSTEM ist ein professionelles Musik-Synthesizer-System, das es Ihnen erlaubt, alle Sound-Möglichkeiten des Commodore 64 in vollem Umfang auszunutzen. Sie können fertige Musikstücke abspielen oder neue Kompositionen entwerfen. Die Noten samt aller Zusatzzeichen werden in grafisch hervoragender Weise in allen Details mit der Eingabe oder

dem Spielablauf auf dem Bildschirm angezeigt. Alle Möglichkeiten, die es in der Musik gibt, bietet Ihnen EXTENDED SYNTHESIZER SYSTEM in drei Stimmen. Preis: DM 98,- III

Auslieferung EXTENDED SYNTHESIZER SYSTEM für Commodore 64 und Floppy 1541 inklusive ausführlichem deutschen Handbuch mit ca. 50 Seiten.

BASIC-COMPILER

- Macht Ihre Programme wesentlich schneller.
- Kompatibel zu EXBASIC LEVEL II, SIMON'S BASIC und SOFTMODULEN.
- Ausführliche 40-seitige Dokumentation.
- Für Commodore 64/1541. DM 298,- inkl. MwSt.
- Für CBM 8032 mit 3050/8250. DM 698,- inkl. MwSt.

INTERFACE AGE Verlag GmbH

Vollburger Str. 1, 8000 München 21
Tel. (089) 5 80 67 02, Telex 5 213 489 iavmd

Ausführliche Gratisinformationen auf Anfrage

Hannover Messe
Halle 3 CEBIT
US-Stand
47/48

- 10 Vorführgeräte BASF 6106
(3 Monate Garantie) Stück **320,- DM**
- 3 Doppelgehäuse mit Netzteil
für BASF 6106 (Vorführgeräte
mit leichten Gehäusefehlern) Stück **125,- DM**

EPROM-Programmiergerät

Programmiert: 2708, 2716, 2732/2532, 2764, 27128 u. ä.,
mit Zusatz auch 8748, 8749, 87555 usw.

- Bausatz **175,- DM**
Zusatz **90,- DM**
Fertiggerät **240,- DM**
Zusatz **110,- DM**

Für Apple und kompatible Rechner

Floppy-Controller 5 1/4" für Industrielaufwerke,
z. B. BASF 6106, Shugart usw. und Originallaufwerke

- Bausatz **195,- DM**
Fertiggerät **280,- DM**

Preh-Commander-Keyboard

AK 87 mit Gehäuse, Anschlußkabel
und separatem 10er Block,

- deutscher Tastensatz **350,- DM**
Festplatzstation 10 MB brutto,
mit sämtlicher Hard- und Software
für Anschluß an Apple **6000,- DM**

KÜHN ELEKTRONIK

2909 Bösel, Postfach 67, Telefon 044 94/15 64

Für TRS 80 ● Video-Genie ● Apple

Eprom-Programmiergerät SE 40

für 2716/2732/2532/2758, kompl. anschlussfertig,
Software auf Kass. oder Disk Preis 269,— DM

Eprom-Löschgerät SE 50

für max. 5 Eproms, Löschdauer ca. 10 Min.
Preis 125,— DM

afu electronic vertriebs gmbh

Steinstraße 9, 5778 Meschede, Telefon 0291/75 85

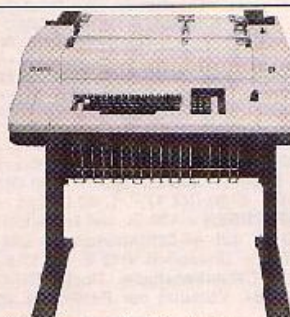
DIE FORTH-QUELLE

- CP/M 86 FORTH für C'T 86 Computer 376,20 DM
- Update für CP/M 86 FORTH auf 32 Bit Adressierung 564,30 DM
- Floatingpoint Unterstützung 376,20 DM
- Textformatter + Editor für CP/M 86 FORTH 376,20 DM
- Alle Pakete auch für Sirius erhältlich
- VC-20 FORTH Modul mit ScreenEditor, Disk, Utilities 188,10 DM
- Starting FORTH v. Leo Brodie 55,60 DM
- FORTH Encyclopedia, Beschreibung von FigFORTH 88,20 DM
- Source Listing für 1802, 6502, 6800 (6301), 8080, 8086
- 690C, APPLE, 6809 NOVA, PDP 11, 68000, Z80, Z 8000,
- Installationsmanual je 40,00 DM

Wir versenden per Nachnahme (+ 7 DM) oder Vorauskasse (+ 3 DM).
Dies ist nur ein Teil unseres Programms. Infos bei:

DIE FORTH-QUELLE Angelika Flesch

Schützenstr. 3 · D-7820 Titisee-Neustadt · Tel. 8 76 51/16 65



SCANELEC-Sonderposten-Verkauf Olympia Datensystem 6610

Buchhaltungscomputer mit Drucker (ohne Programm)

zum Umbauen — Ausschachten — neu Programmieren

Vollelektronische Tastatur mit getrenntem 10er-Block
und Sonderfunktionstasten. CPU-Platine mit 8080 A,
8255, 8257, 5101, div. EPROMs, 90% auf Sockel!
Superstromversorgung.

Fingehalter 160 Z/sac. Matrizdrucker (9 x 7 Matrix)
mit 53 cm Druckbreite!! DN 60C,—

Dazu passend: Endlos-Formulatortraktor mit eigenem Antrieb, 2 getrennte Formulantriebe (getrennt
steuerbar!) auch für andere Geräte verwendbar DN 30C,—

Dazu passend: Tischgestell mit Papierablage, sehr stabil DN 20C,—

Paketpreis, frei nächster Bahnhof DN 99C,—

Alle Preise incl. MwSt., 1 Lieferung per Nachnahme oder Verrechnungsscheck.

SCANELEC · Husumer Str. 1 · 2396 Sterup · 0 46 37/10 11

SHARP

computertechnik

G. WEBER

EULENSPIEGELSTR. 56 · 8000 MÜNCHEN 83 · TEL: 089/6012554

!!! absolute NIEDRIGPREISE !!!

PC's aller Serien, Peripherie, Softwaremodule
für PC 150C sowie 8, 16, 24 KB-Erweiterungen

MZ 3500 Serie

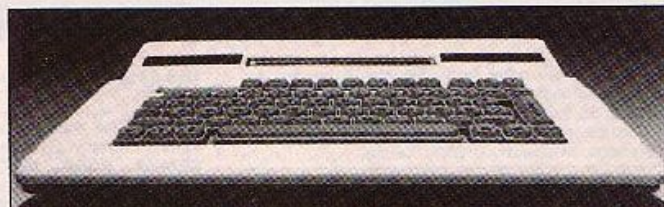
MZ 700 Serie

*Supergrafik

*Floppys

*anschlussfertige EPSON-Drucker

*EPSON**Hewlett Packard*
*APPLE*COMMODORE*SEIKOSHA
*BINDER*etc.**ZUBEHÖR**



MAX 1



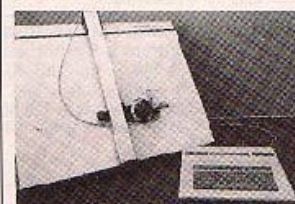
- Programmieren
- Editieren
- Schreiben



- Steuern
- Regeln
- System-Training



- Prüfen
- Testen
- Messen



- Vermessen
- Digitalisieren
- Positionieren

in erweitertem
BASIC
mit

Dietmar Böhm · Elektronik
Schömberger Straße 4—6
7542 Schömberg-3
Telefon 07084/7600

MAX 1

FORTH — die neue und einfache Sprache: Assemblierungs- und alle wichtigen Mikroprozessoren DM 35, FORTH Installat. on Manual DM 35, Informationsmaterial kostenlos. C. Schmidt — Software, Ringstr. 8, 3500 Kassel.

SUCHE preiswerter ZX81-Drucker. Tel. 0658/7707.

CBM-64 immer neueste Top-Prgrm. ★ über 500 Prgrm. auch Beschr. Verkauf/Tausch. Tel. 06193/64456.

Fernschreiber mit Interf. für TRS 80/Vicco-G., SC/MP-Mikrocomputer, je 180 LM, 089/8343864.

16Bit-CPU und Peripherie 8086A 5MHz à 58,35, 8234 Clk-Gen à 14,30, 8238 BUSCTR à 29,50, 8159 PIC à 14,30, 86545-1 à 39,50 DM. Alle Preise inkl. MwSt. Nachnahme Versand. DGM ELECTRONIC, Gudrunstr. 16, 8000 München 19.

TRS80-Color-Computer Hard- u. Softw. 06175/1510.

CT86 und IBM kompatibel!!! Anwender für Programm- und Erfahrungsaustausch gesucht. Hermes-Gehrke, Tel. 0221/724636 oder 5993515.

CBM und VC-20: Spitzensoftware aus allen Bereichen in gr. Auswahl! Günstige Preise. Z.D. Address/Textv. je 15 DM. Maxiliste 1 DM bei T. Hanne, Hertastr. 24, 4650 Gelsenkirchen.

Verkaufe BBC-Computer/B + Software + Interfaccschaltungen. Tel. 0517/6284.

Schaltbild für Oric 1 Spectrum 0203/13345.

Modulkopierer für VC-20 und C64, kopiert jedes Rom-Modul! Info 0,80 Rückporto. G. Genech, Derner Str. 363, 4600 Dortmund 14.

BÜRO-COMPUTER auf TRS-80/II-Basis (48k, 2 Disks MPI, Percom Doubler II) alles, incl. CPL + Interf. in Untertisch-Stahlgewächse. flache Tastatur und grüner Monitor (Epsonbeige). Kleinschr. + Umlauf auf Tast. u. Bildschirm. Dazu ca. 50 Disks voll Software mit Dokumentationen + zusätzl. Literatur weg. Systemumstellung zu verk. DM 3500,— VB. Tel. 040/79675-06(07) ab 19 Uhr.

CBM 64: 10 ausgesuchte Spitzensoftware 35 DM, einwandfrei auf Disk. o. Cass. Info gratis von: A. Nübel, Graefestr. 40, 1000 Berlin 61.

COMMODORE 64, alle Programme. Tel. 0201/602708.

64-FORTH Nun gibt es diese schnelle strukturierte Sprache auch f. d. C-64. FORTH ist bel. erweiterbar u. damit für Graphik u. Soundprogr. gut geeignet. FORTH a. Cass. DM 49,—, Handbuch CM 25,—. INFO D. LUDA, Staudingerstr. 65, 8030 München 83 ★ 64-FORTH ★ 64-FORTH ★ 64-FORTH

FORTH FÜR APPLE!! und Verwandte mit Assembler und Editor auf Cassette LO-Graphik implementiert, HI-Graphik auf Source-Screens. Für DM 90,— bei J. Kaiser, Heusmattenweg 5, 7301 Oberriet.

VERKAUFE Neu VC-20 + DATA SETTE + Programmierbuch, VB 380 DM. W. ZHANG, Zi 47, Tel. 0228/656961.

NEU! MICRONIC-Textsystem für APPLE 2 nur 48 DM auf Minidisk. M. SCHWEIGER, PF 111069, 6300 Gießen.

TRS 80 MOD I Lev. II verk. W. Gutsch, Tel. 0228/613929.

CBM-64 Soft- und Hardware **ZX-SPECTRUM** Originalprogramme zu günstigen Preisen! Z.B.: für **CBM-64: KONG, MOON-BUGGY, BLUE-MOON, NEO, CLYPS** u.v.a. ab 29,— (Superhit **ASSEMBLER-64: 34,—**) für **SPECTRUM: FORTH 79,—, PENGY, QUINCY, PANIC, ARCADIA** ab 19,30, Tastatur (Metallgehäuse) 169,—. Info: Rporto-Triebner, Bussardweg 39, 858 Bayreuth.

Cbm64-Cbm64-Cbm64: Achtung! Supersoftware aus allen Bereichen (Spiel-User-Daten- und Textverarbeitungsprogramme sowie Musik-Basic- und Grafikweiterungen). Auch die neuesten Spiele, z.B. Congo Bongo, Pt. Siop, Night Mission u.a. Gratis-Info sofort anfordern bei: Postlager-Nr. 012108D, 1000 Berlin 12.

COMPUTER-DÜCHER für Einsteiger, Fortgeschrittene, Profis! Große kommentierte Liste gratis. Versandbuchhandlung DLTSCHKE, Brünigheide 97/OT, D-44 Münster, Tel. 0251/214459.

ACHTUNG APPLE-ANWENDER, Einführungsangebot: 2 TEAC-FLOPPYS FD55F mit Controller, Kabel, Gehäuse inkl. MwSt. und Montage DM 1895,—. Reparatur-Service aller gängigen Floppys, Drucker usw. Tel. 02107/60263.

CBM-64 90 Top-Programm auf Cass./Disk. DM 50,—. VS/Schein an M. Günsche, Odenwaldstr. 13, 6980 Werheim 2, INFO gegen DM 2,— in Briefmarken. Lieferung innerhalb 3 Tagen.

SOFT/HARDWAREAUTOREN gesucht. Bieten gute Konditionen. Fordern Sie unsere Info-Mappa an. Oder senden Sie uns Ihr Programm mit kurzer Beschreibung. Computertyp egal. Antwort folgt sofort! BIMA-ELEKTRONIK, Heisterweg 6, 2382 Kropp.

Suche Computerfans in Berlin die sich auch ihren CT-86 bauen, zum Erfahrungsaustausch. Günter Wegener, 1/30, Tel. 030/2163719.

NEC7220: Grafik-Software in Z80 Assembler. Listing (1700 Z.) oder 8" SSSD. 150,—. Tel. 0761/580795.

Verkaufe: AIM 35, 4K+32K DRAM + Netzteil — Gehäuse, Fernschreiber (mit Interface + Softw.), Terminal IT 3210, zus. IBM 2741 (Kugelschreiber), auch einzeln. Rüdiger, Tel. 0226/465271.

68000 FORCE PROFIKT 1 mit 128 KB RAM und Schaltplan. FP 500,—, 8"-USDD Laufwerke (2) unbenutzt. Fabr. MFE à 850,— zus. 1600,—. A. Pflughaupt, Nachtigallweg, 2351 Treppenkamp.

BASIS 108 kPzt. m. 2 SIEMENS DRIVES u. div. Software, DM 3350,—. SIEMENS DRIVE YCD 100-5 neu, DM 465,—. SCHNITZER, Tel. 02922/7185.

KAYPRC II 8bit Rechner, 64KB RAM Betr. Syst. CPM, 2 Floppy à 190KB 9" Monitor, IC Programm-disk. Origin verpackt, 4700,— DM, Tel. 07475/1446.

Neu auf 16-bit: **MULTIBASE**. Verbinden Sie dBASE II, MULTIPLAN u. Wordstar (bzw. andere TV-Programme) zu einem integrierten System. Infos über Knögel, Düsseldorf, Str. 7, 8000 München 40 oder Tel. 07940/2970.

Drucken mit Spectrum + VC-20/64! Grafikfähige Centronics-Schnittstelle (Software im Fpmm) nur DM 148,— kpl. mit 80-Zeichen-Grafik-Drucker ab DM 675,—. Info „CT-1“ anfordern. **DORSCH-electronic**, Hauptstr. 23, 8501 Eckental 2, Tel. 09126/7419. Händleranfr. willkommen.

KOMTEK I SUPER 32K zu verk. VB 700,— DM. W. Gutsch, Tel. 0228/613929.

Modul-Kopierer für VC-20 oder C-64, kopiert jedes ROM-Modul problemlos. Netzunabhängig! Nur 98,—. Info gegen Rückporto. G. Genech, Derner Str. 363, C-4600 Dortmund 14.

EPROM-FROGRAMMIER-SERVICE 27 62/58 + 2732. Kopieren, löschen / Apple komp. Ursoft / Info 80 Pf., Klaus Becker, 6072 Dreieich, Eisenbahnstr. 83.

CP/M CROSS ASSEMBLER. Ideal für Einplatinencomputer. Entwicklung 6809, 6800, 65XX-C02-II je DM 249,—, 8 Zoll ssd o. Apple CP/M format. U. Corson, Röntgenstr. 3, 59 Siegen, Te. 0271/88815.

MATRIX-DRUCKER (Centronics 730) gebraucht, für DM 389,— von Privat. Tel. 040/6322453.

SPECTRUM, SUPERSOFTWARE, SPECTRUM. Info 80 Pf., Dipl.-Ing. G. Versé 4650 Gelsenkirchen, Grüner Weg 45.

COMMODORE 64-BESITZER! Für Software und Erfahrungsaustausch meldet Euch bei EPNS ZEDI, Dellsbergerallee 30, CH-4053 BASEL.

UMSTÄNDEHALBER: alphanetic-Pc (4 Wochen alt), incl. techn. Unterl. (Schaltpläne, HW-ADR.), VB: DM 1400,—. Tel. 07231/79734.

Software C 64 zu kaufen und tauschen. Tel. 030/3346158.

STECKER für Computer liefern wir ab Lager sowie sämtliche Kabelverbindungen nach Ihren Angaben. Preisliste gegen 2,— DM in Briefmarken. COM-PLERSYSTEME NIEDERGESALSS, GOEBENSTR. 26, 6200 WIESBADEN. Tel. 06121/45919.

An dieser Stelle könnte Ihre private oder gewerbliche Kleinanzeige stehen. Exakt im gleichen Format: 8 Zeilen à 45 Anschläge einschl. Satzzeichen und Wortzwischenräumen. Als priv. Hobby-Elektroniker müßten Sie dann zwar 31,92 DM, als Gewerbebetreibender 52,90 DM, Anzeigenkosten bezahlen, doch da'ür würde Ihr Angebot auch garantiert beachtet. Wie Sie sehen.

APPLE comp. I Idl-Liste, Tagestiefpreise, Rückgaberecht 10 T. GENERALIMPORTEUR STREIL, Mommsenstr. 3, 4306 Erkrath 2, Tel. 02104/43079.

ZX Spektrum — ZX81 — VC20 Hardware, Zubehör im Selbstbau zu tollen Preisen. Sofort Info gegen Rückumschlag anfordern: BALTES, Nordring 30, 6620 Völklingen.

COMPUTER-CASSETTEN — 10er Pack, BASF-Band LHD mit Boxen, Etiketten und Einlegern, C 10 DM 15,—, C 20 DM 16,—, C 30 DM 17,—, C 40 DM 18,—. **CASSETTEN-AUFKLEBER** — 100 St. auf Lochstreifen 5,— DM, 120 St. auf A4-Druckbögen 7,— DM. **VIDEO-CASSETTEN** — Domiphon VHS E-18C 17/80 DM, Christomenia-Cassettenstudio, Postf. 3584, Zwesten, 05626/281. Versand per Rechnung ab 20,— DM.

PC-1500 FORTH: Modularer Programmiersprache. CE155 erforderlich. Cassette + Handbuch DM 129,—. INFO anfordern. A. Ulrich, Kcpemikstr. 11, 8000 München 80.

SPITZENSOFTWARE f. TI 95/4A. Nicht nur Inverterspielchen, sondern gute Software. Gratisliste TI 99 besteller!! G. Mahlig, Auf dem Heidenberg 19, 5000 Köln 60, Tel. 0221/712358 PS. Suche TI Hardware.

**Kurz + bündig.
Präzise + schnell.
Informativ + preiswert.**

Wenn Sie Bauteile suchen, Fachliteratur anbieten oder Geräte tauschen, verkaufen oder kaufen wollen — mit wenigen Worten erreichen Sie durch c't schnell und preisgünstig jede Menge mögliche Interessenten.

Probieren Sie's aus! Die Bestellkarte für Ihre Kleinanzeige finden Sie am Schluß dieses Heftes.

Übrigens: Eine Zelle (= 45! Anschläge) kostet nur 3,99 DM. Inklusive Mehrwertsteuer!



Software: Branchenpaket „Reisebüro“ — Textverarbeitung „Fibu“ — u. a.
Hardware: Die professionelle Konfiguration zu Ihrem Problem
Service: Auslieferung/Wartung max. 24 Std. bundesweit
Preis: Ob Händler oder Enduser, rufen Sie uns an
Türmergasse 25 · 6900 Heidelberg · 0 62 21/78 05 55

STA

ct magazin für
computer
technik

**6/84 —
Anzeigen-
schluß
am 11. April
1984**

SIDNEY II

hib
HIB GMBH
MICROCOMPUTER
Wunderburggasse 2
8300 Nürnberg 1
0311 22 13 83

SIDNEY — die Computer
für den professionellen Anwender!

ausgestattet mit 2 Prozessoren
SIDNEY II: 6502 und 2-30
SIDNEY 88: 6502 und 8048
64 K RAM aufrüstbar bis zu 512 K
incl. PROFITastatur ab DM 3640,00

ALLES für den APPLE II

PROFITastatur, einzeln DM 598,00
Hard Disk 10 MD, kompl.
anschluß, im Gehäuse DM 5998,00
5" Flachlaufwerk
anschlußfertig DM 589,00
Plot-II Graphics Table DM 269,00
SUPER JOYSTICK DM 118,00
Minidisks, "CON"ROL
DATA", 55/80 10 Stk ab DM 55,00
Diskettenbox für 80 Mini-
Disk, mit Schloß DM 55,00

Alle Preise incl. Mehrwertsteuer
Gratispreisliste anfordern!
Wiederverkäufer-Preise nur
auf schriftliche Anfrage.

TRACE80 —ICE

**TRACE 80 — kompatibler
Slave-Emulator
für CP/M-Computer**

- Anschluß für V24
- Echtzeit-Emulation
bis 6 MHz ohne Wait
- Beliebige viele Halte-
punkte, auch im ROM
- 64-K-Emulations-RAM
- Über 50 Befehle
- Symbolisches Debugging
mit M 80, RMAC
oder MS-Compilern
- Strukturiertes Debugging
- Ablaufsteuerformeln
- Erweiterbar durch
Trigger-/Trace-Logik

Lothar Lauterbach
Ing.-Büro für Datentechnik
Sommerstraße 10
8012 Riemerling
Telefon 0 69 / 60 61 73

c't-Platinen

c't-Platinen bestehen aus Epoxid-Glasartgewebe, sind fertig gebohrt und mit Lötstopplack versehen bzw. verlötet. Die Bestellnummer bezieht sich auf den Beleg, in dem das betreffende c't-Projekt vorgestellt wurde. Sie setzt sich zusammen aus Jahrgang, Heftnummer und Seitennummer. Die zusätzlichen Buchstaben bedeuten: 'd' — doppelseitig, 'B' — Bestückungsdruck, 'E' — elektronisch geprüft.

Nr.	Projekt	Format	Preis
831241dBE	Terminal A (ohne Tastatur)	ca. 84 x 234 mm	59 DM
831242dBE	Terminal B (mit Tastatur)	Doppel-Europa	75 DM
831262	Universelles Netzteil	Europa	14 DM
840147dBE	c't 86, CPU-Karte	Europa	85 DM
840148dBE	c't 86, RAM-Karte 256 KByte	Europa	89 DM
840149dBE	c't 86, I/O-Karte	Europa	69 DM
840200dBE	c't 86, Floppy Interface	Europa	65 DM
8401501	Busplatine (96pol., 10 Steckplätze)	84 x 208 mm	49 DM
8401641	CEPAC-80 mit Wrap-Feld	Europa	69 DM
8401671	CEPAC-80 ohne Wrap-Feld	ca. 86 x 100 mm	49 DM
840242B	Centronics/V24-Interface für Olympus COMPACT	80 x 136 mm	15 DM
840252B	c't-Sprachsynthesizer	105 x 111 mm	21 DM
840352dBE	CEPAC-85, Version A	80 x 100 mm	27 DM
840354dBE	CEPAC-85, Version C	Europa	52 DM

So können Sie bestellen: Um unnötige Kosten zu vermeiden, liefern wir nur gegen Vorauskasse. Legen Sie Ihrer Bestellung einen Verrechnungsscheck oder einen von Ihrer Bank quittierten Einzahlungsbetrag über die Bestellsomme zuzüglich 3 DM (für Porto und Verpackung) bei. Bei Bestellung aus dem Ausland muß stets eine Überweisung in DM erfolgen. Die Überweisung und Ihre Bestellung richten Sie bitte an:

c't-Versand, Verlag Heinz Heise GmbH, Bissendorfer Straße 8, 3000 Hannover 61, Konto-Nr. 9305-300, Postscheckamt Hannover

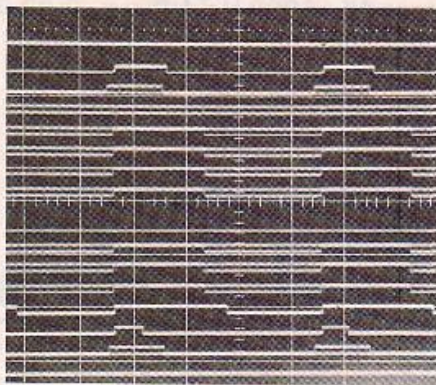
Firmenverzeichnis zum Anzeigenteil

adcomp Datensysteme, München .. 77	Elektronikladen, Detmold .. 18	Linde, Schwäbisch-Hall .. 40	SCANELED, Sierup .. 105
AFC Computer, Köln .. 104	epe, Großrinderfeld .. 3	LOGITPA, Berlin .. 39	Siber Kikal K.K., Frankfurt .. 101
afu, Meschede .. 105	EVG Electronic, Langenhagen .. 9		SYNELEC, München .. 13
arcon Computertechnik, Berlin .. 14			Syntax, Rastatt .. 99
ASC, Aachen .. 104	Forth Quelle, Flesch, Titisee .. 105	Magnetonic, München .. 18	Syscom, Winkler, Heidelberg .. 20
Almann, Strassenscheid .. 97	Frech-Verlag, Stuttgart .. 99	MEINHOFF, D. Leiste .. 18	Systech, Braunschweig .. 91
	Frölje, Oldendorf .. 74		
Behn, Düsseldorf .. 18	Glisch, Leonberg .. 18	MARFLOW, Harnover .. 87, 39	Seharuel, Darmstadt .. 71
BICC VERO, Bremen .. 33	HIB, Nürnberg .. 107	MOGEVVIC, D. Leiste .. 18	GTA, Heidelberg .. 103, 107
Böhm, Schörmberg .. 105	Hall Informatic, Fl.-Varuz .. 35	Meinhold, Siegen .. 54	Siede, Willingen .. 97
BSP Kug, Regensburg .. 84	Hösch, Düsseldorf .. 40	Meyer + Harelik, Hannover .. 17	Strie, Schwaförden .. 87
bst computer, Frankfurt .. 104		mirwald, Urterhaching .. 97	
		Münzenloher, Holzkirchen .. 35	
CE Computer, Schwerte .. 14	IMETEC EDV, Wiesbaden .. 99	NEC-Hauptändler .. 22	
COMPISSHACK, Neuwied .. 19	Interface AGE-Verlag, München .. 104	Neuhäus, Hamburg .. 41	Teep, Wellrod .. 67
COMPUTER ACCESSORIES INTL.		NEWMAN Compute-Versand,	Time Soft, Stuttgart .. 35
Ölthmann .. 10, 11	Köller, Schieder-Schwalenberg .. 99	Schenefeld .. 103	
Conitec, Darmstadt .. 97	Kühr, Bessel .. 104		Üding, Menden .. 97
Conrad, Hirschau .. 18		Piel, Bad Neustadt .. 112	Unitronic, Düsseldorf .. 20
Cromemco, Eschborn .. 15	LAMPSON & ZERBE, Böttelbörn .. 95	profiSoft, Osnabrück .. 85	
	Lauterbach, Riemerling .. 107		Weber, München .. 105
Data Becker, Düsseldorf .. 5	LECH-TECHNICS, Kerpen-Törnich .. 41	RATFV, Ratingen .. 18	Wiesemann, Wuppertal .. 74
		Röckrah, Aachen .. 18	Wilke, Aachen .. 41

unter anderem

Fehlersuche in Computer-Systemen

Professionelle Testgeräte für Mikrocomputer-Systeme sind für viele Selbstbauer unerschwinglich. c't zeigt, wie man sich behelfen kann, wenn lediglich ein einfaches Oszilloskop zur Verfügung steht. Das Projekt des Monats paßt in diesen Zusammenhang: Ein Vorsatzgerät, das es ermöglicht, 16 Kanäle gleichzeitig auf dem Oszilloskop-Bildschirm darzustellen. Der 'digitale Scope-Extender' verarbeitet TTL-Signale bis zu 20 MHz und kostet nicht die Welt (Bauteile und Platine: etwa 60 Mark).



Umgang mit MACRO-80

Microsoft hat mit dem MACRO-80 einen Assembler geschaffen, der (kaum) Wünsche offenläßt. Aber gerade die vielen Möglichkeiten, die das Programm bietet, machen dem Einsteiger das Leben schwer. Der c't-Bericht soll nicht das Handbuch ersetzen, sondern anhand von Demonstrationsprogrammen die wichtigsten Features verständlich machen und so ein 'Gefühl' für den Umgang mit diesem ausgezeichneten Werkzeug vermitteln.

SuperTape für 6502-Computer

Schnelle und sichere Datenaufzeichnung auf der Kassette, Kommunikation zwischen Rechnern verschiedener Fabrikats — all das ermöglicht das in dieser c't-Ausgabe vorgestellte SuperTape-Verfahren. Im nächsten Monat veröffentlichen wir die SuperTape-Routinen für 6502-Systeme am Beispiel der beiden populären Commodore-Homecomputer VC-20 und C64.

Prüfstand:

Tandy Colour Computer 2

Erweiterte Möglichkeiten, mehr 'Benutzerfreundlichkeit' verspricht Tandy mit dem Nachfolger des Colour 1. Wird sich der Neue auf dem heiß umkämpften Markt der kleinen Homecomputer durchsetzen können?



Außerdem:

Mehr BASIC intern
VIA 6522-Applikation
SuperTape-Bedienungsprogramm für ZX81
u.v.a.m.

Centronics-Interface für ZX Spectrum

Nach den ZX81-Besitzern erhalten nun auch die Spectrum-Anwender eine preisgünstige Möglichkeit, 'richtige' Drucker zu betreiben. Die c't-Schaltung läßt sich — wie beim ZX81 — nicht nur als Drucker-Interface, sondern auch als universelle Schnittstelle für die Ein-/Ausgabe einsetzen.

ORIC-ROM geknackt

Mit Informationen über Internis ist das ORIC-Handbuch sehr zurückhaltend. Viele Anwender beklagen, daß sie die Leistungsfähigkeit ihres Computers nicht voll nutzen können. c't beendet die Geheimniskrämerei: In der nächsten Ausgabe finden Sie alle wichtigen ROM-Adressen und weitere wertvolle Praxistipps für den Umgang mit dem ORIC-1.

Heft 5/84 (April/Mai) erscheint am 12. April 1984

Änderungen vorbehalten

Das bringt **elrad**

elrad 3/84 — jetzt am Kiosk

- Die große Bauanleitung: ElMix — Mischpult für Bühne und Studio; mit einem 72 cm () langer Platinen-Layout im Maßstab 1:1
- Bauanleitung Spezial-Hifi-Netzteil
- Bauanleitung 'Heizungsregelung für Minimalverbrauch'
- Die elrad-Laborblätter: Analog-Schalter in CMOS; über 20 Schaltungsbeispiele
- Computing Today: Bauanleitung ZX81-Soundboard
- Sprachkurs 'Going FORTH' (Schluß)

elrad 4/84 — ab 27. 3. 1984 am Kiosk

- Bauanleitung Bühne/Studio: PA-Box
- Bauanleitung 'Elektronische Sicherung'
- Die elrad-Laborblätter: Der Operationsverstärker in Meß- und Prüfschaltungen, zahlreiche Schaltungsbeispiele
- elrad-Report: Musikmesse Frankfurt — die neuen Techniken
- Computing Today: FORTH-Simulator in ZX-BASIC

Impressum:

c't

Magazin für Computertechnik

Verlag Heinz Heise GmbH

Bissendorfer Straße 8, 3000 Hannover 61

Postanschrift: Postfach 2746

3000 Hannover 1

Ruf (0511) 535 20

technische Anfragen nur freitags 9.00—15.00 Uhr

Postcheckkonto Hannover, Konto-Nr. 93 05-308
Kreissparkasse Hannover, Konto-Nr. 000-019968
(BLZ 250 502 99)

Herausgeber: Christian Hese

Redaktion:

Christian Persson (Chefredakteur)

Andreas Burgwitz (stellvertretender Chefredakteur)

Dipl.-Ing. Detlef Grell

Ständige Mitarbeiter:

Dipl.-Ing. Rolf Keller

Dipl.-Ing. Eberhard Meyer

Dipl.-Chem. Holger Petersen

Dipl.-Ing. Eckart Stefens

Dipl.-Ing. Kurt Werner

Technische Assistenz: Hans-Jürgen Bernöt

Abonnementsverwaltung, Bestellwesen:

Doris Imken, Anita Kreutzer

Anzeigen:

Wolfgang Pensler (Anzeigenleiter)

Gerinde Dörner (Disposition)

Es gilt die Anzeigenpreisliste 1 vom 1. 10. 1983

Redaktion, Anzeigenverwaltung,

Abonnementsverwaltung:

Verlag Heinz Heise GmbH

Postfach 2746

3000 Hannover 1

Ruf (0511) 535 20

Herstellung: Wolfgang Ulber

Grafische Gestaltung:

Wolfgang Ulber, Dirk Wolzschläger

Satz und Druck:

Hahn-Druckerei, Im Moore 17, 3000 Hannover 1

Ruf (0511) 7083 70

c't erscheint monatlich.

Einzelpreis DM 5,—, öS 52,—, sfr 6,—, hfl 6,80,

Jahresabonnement Inland DM 58,— inkl. MwSt. und
Versandkosten, Schweiz sfr 58,— inkl. Versandkosten,
Österreich öS 480,— inkl. Versandkosten, Niederlande
hfl 58,— inkl. Versandkosten, Sonstige Länder 65,—
DM inkl. Versandkosten.

Vertrieb (auch für Österreich, Niederlande, Luxemburg
und Schweiz):

Verlagsunion Zeitschriften-Vertrieb

Postfach 570

D-6200 Wiesbaden

Ruf (06121) 266-0

Verantwortlich:

Textteil: Christian Persson

Anzeigenenteil: Wolfgang Pensler

beide: Hannover, Bissendorfer Straße 8,

3000 Hannover 61

Eine Haftung für die Richtigkeit der Veröffentlichungen
kann trotz sorgfältiger Prüfung durch die Redaktion vom
Herausgeber nicht übernommen werden. Die geltenden
gesetzlichen und postaleschen Bestimmungen bei Erwerb,
Errichtung und Inbetriebnahme von Sende- und Empfangs-
einrichtungen sind zu beachten.

Die gewerbliche Nutzung, insbesondere der Schaltpläne
und gedruckten Schaltungen, ist nur mit schriftlicher Ge-
nehmigung des Herausgebers zulässig. Die Zustimmung
kann an Bedingungen geknüpft sein.

Honorare: Arbeiten gehen in das Verfügungsrecht des
Verlages über. Nachdruck nur mit Genehmigung des Ver-
lages. Mit Übergabe der Manuskripte und Bilder an die
Redaktion erteilt der Verfasser dem Verlag das Exklusiv-
recht zur Veröffentlichung. Für unverlangt eingesandte
Manuskripte kann keine Haftung übernommen werden.
Sämtliche Veröffentlichungen in c't erfolgen ohne Be-
rückichtigung eines eventuellen Patentschutzes. Waren-
namen werden ohne Gewährleistung einer freier Verwen-
dung benannt.

Printed in Germany

© Copyright 1984 by Verlag Heinz Heise GmbH

ISSN 0724-8679

Titelidee: c't

Titelfoto:

Foto: J. L. H. Heise, Manfred Zimmermann

Auftragskarte

Chiffregebühr DM 5,70 inkl. MwSt.

c't-Kontaktkarte

● **Platinen, Folien, Bücher, Software, bereits erschienene Hefte** beim Verlag Heinz Heise GmbH, c't-Versand, Postfach 2746, 3000 Hannover 1. **ordern.**

c't-Platinen-Folien-Abonnement

Abrufkarte

Einzelbestellungen siehe Anzeigenteil.

c't - Kleinanzeige

Auftragskarte

Bitte veröffentlichen Sie in der nächsten erreichbaren Ausgabe nachstehenden Text:

[illegible]

Pro Zeile bitte jeweils 45 Buchstaben einschl. Satzzeichen und Wortzwischenräume. Wörter, die **fettgedruckt** erscheinen sollen, unterstreichen Sie bitte. Den genauen Preis inklusive Mehrwertsteuer können Sie so selbst ablesen. Soll die Anzeige unter einer Chiffre-Nummer laufen, so erhöht sich der Endpreis um DM 5,70 Chiffre-Gebühr inkl. MwSt.

Bitte umstehend Absender nicht vergessen!

c't - magazin für computer technik Kontaktkarte

Ich beziehe mich auf die in c't _____/8__, Seite _____ erschienene

- ☐ und bitte Sie, mir weitere **Informationen** über Ihr Produkt _____

[illegible]

Absender nicht vergessen!

Datum, Unterschrift (für Jugendliche unter 18 Jahren der Erziehungsberechtigte)

c't-Platinen-Folien-Abonnement

Abrufkarte

Ja, übersenden Sie mir für 1 Jahr die c't-Platinen-Folie ab

Monat

1983

Das Platinen-Folien-Abonnement gilt nur für 12 Monate und muß im voraus bezahlt werden.
Es kostet DM 30.— inkl. Versandkosten und MwSt.

- ☐ Postsparkasse Hannover, Konto-Nr. 93 05-308;
☐ Kreissparkasse Hannover, Konto-Nr. 000-0 199 68.

Bitte geben Sie unbedingt auf dem Überweisungsbeleg „Folien-Abonnement c't“ an.

Absender und Lieferanschrift

Bitte in jedes Feld nur einen Druckbuchstaben (ä = ae, ö = oe, ü = ue)

Vername/Zuname

Strasse/Nr.

[illegible]

P17

[illegible]

Datum/Unterschrift

Ich bestätige ausdrücklich, vom Recht des schriftlichen Widerrufs innerhalb von 10 Tagen nach Abschluß beim Verlag Heinz Heise GmbH, Postfach 27 46, 3000 Hannover 1, Kenntnis genommen zu haben.

Unterschrift

Bitte beachten Sie, daß diese Bestellung nur dann bearbeitet werden kann, wenn beide Unterschriften eingetragen sind.

Absender (Bitte deutlich schreiben!)

Vorname/Nachname

Beruf

Straße/Nr.

PLZ Ort

Bitte veröffentlichen Sie den umstehenden Text von _____ Zeilen zum Gesamtpreis von _____ DM in der nächst erreichbaren Ausgabe von c't. Den Betrag habe ich auf Ihr Konto

Postscheck Hannover,
Konto-Nr. 93 05-308;
Kreissparkasse Hannover,
Konto-Nr. 000-019968

überwiesen/Scheck liegt bei.

Veröffentlichungen nur gegen Voraus-
kasse.

Datum Unterschrift (für Jugendliche unter 18 Jahren der
Erziehungsberechtigte)

Antwort

Bitte mit der
jeweils gültigen
Postkartengebühr
freimachen

ct magazin für
computer
technik

Anzeigenabteilung
Verlag Heinz Heise GmbH
Postfach 2746

3000 Hannover 1

c't - Private Kleinanzeige

Auftragskarte

Nutzen Sie diese Karte, wenn Sie
etwas suchen oder anzubieten ha-
ben!

Abgesandt am

198

Bemerkungen

c't-Kontaktkarte

Anschrift der Firma, bei
der Sie bestellen bzw. von der
Sie Informationen erhalten wollen.

Absender

(Bitte deutlich schreiben)

Vorname/Nachname

Beruf

Straße/Nr.

PLZ Ort

Telefon Vorwahl/Rufnummer

Postkarte

Bitte mit der
jeweils gültigen
Postkartengebühr
freimachen

Firma

Straße/Postfach

PLZ Ort

c't-Kontaktkarte

Abgesandt am

198

an Firma

Bestellt/angefordert

c't-Leser-Service

Antwort

Bitte mit der
jeweils gültigen
Postkartengebühr
freimachen

ct magazin für
computer
technik

Verlag Heinz Heise GmbH
Postfach 2746

3000 Hannover 1

c't-Platinen-Folien-
Abonnement

Abrufkarte

Abgesandt am

198

zur Lieferung ab

Heft 198

Jahresbezug DM 30,—
inkl. Versandkosten und MwSt.

Abbuchungen sind aus organisatori-
schen Gründen nicht möglich.

Ausführlich
und
umfangreich
wie kein
Lautsprecher-
Selbstbau-
Heft
zuvor.



Versand- bedingungen:

Die Lieferung des Heftes erfolgt per Nachnahme (+ DM 4,50 Versandkosten) oder gegen Verrechnungsscheck (i. DM 2,00 Versandkosten).

elrad EXTRA

Spitzenqualität im Selbstbau! Von der Mini-Box bis zum Lautsprecher- schrank

Laser in der
HiFi-Technik

Elektronik für Lautsprecherboxen
Passive Lautsprecherweichen
Lautsprecherweichen
Elektronische Frequenzweiche
Lautsprechererschaltung

Bauanleitungen

Transmissionline-Lautsprecher
+ Nachlese
Corner Speaker
KEF CS5

E80-Magrat
Wharfedale E90
Transmissionline

Harbeth 250
+ Nachlese

+ Ergänzungen
FCCAL 250DB
Ton-Pyramiden

Donar —
der Subwoofer
Podszus-Baßhorn
Dynaudio Pyramide

Für die Bühne:
elrad-Jumbo

14,80

Sie erhalten das Selbstbau-Heft direkt beim
Verlag Heinz Heise GmbH, Postfach 27 46, 3000 Hannover 1

YOUR COMMAND BRIDGE. PREH COMMANDER ELECTRONIC.

Die Keyboards mit dem besonderen Touch

- Programmierbares, intelligentes Keyboard
- mit Mikroprozessor
- für hohe Ansprüche



- Schnittstellen-Variationen
- Gehäuse im Preh-Design
- Überall dort, wo die Brücke zwischen Mensch und Maschine hergestellt werden muß, löst Preh die daraus resultierenden Aufgaben – und darin haben wir seit über 60 Jahren Erfahrung.



Preh Commander mit Encoder-Elektronik

Hannover-Messe '84
Halle 12, 2. OG. Stand Nr. 2262



Preh

Preh Elektrofeinmechanische Werke
Postfach 1740
D-3740 Bad Neustadt
Telex: 672503 · Tel. (097 71) 921