

27.01.88

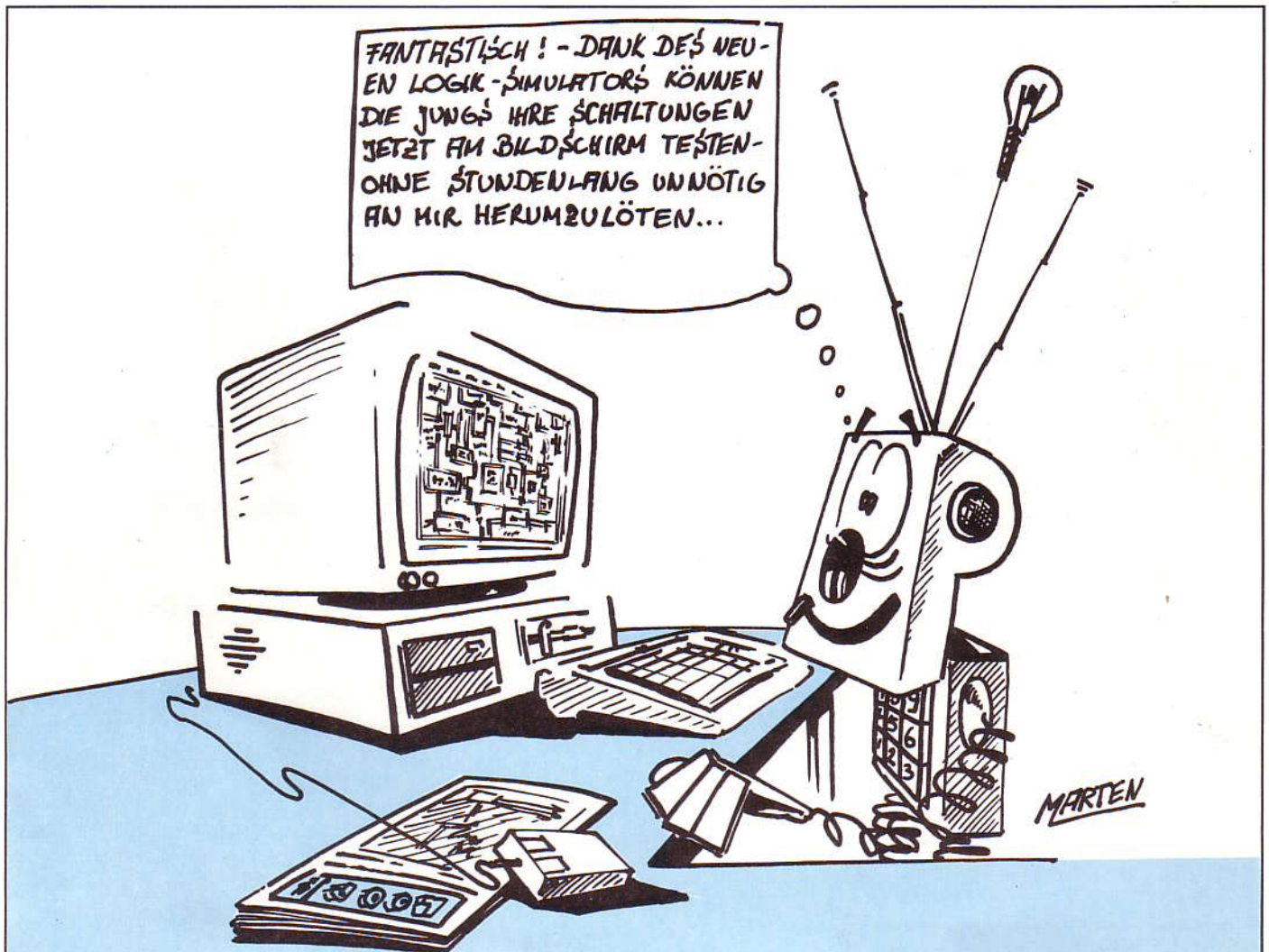
# LOOP

# 20

Dezember 1988

Zeitung für Computer-Bauer, -Anwender, -Programmierer und -Starter

DM 3,50



Aus der Software-Küche des  
Rolf-Dieter Klein:

## DER LOGIK-SIMULATOR

Leitartikel

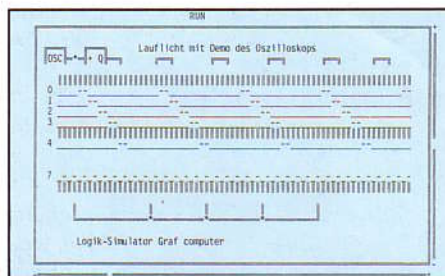
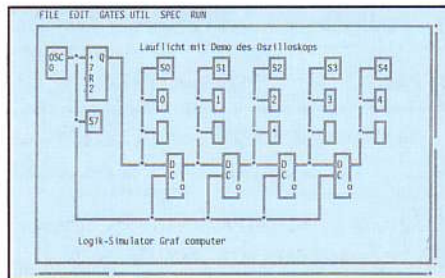
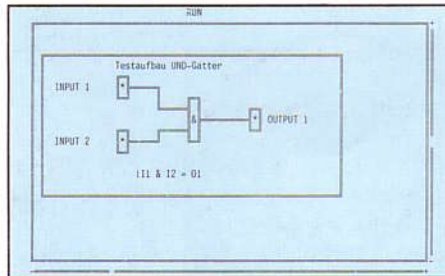
Das neue Software-Bonbon von GES:



Der LOGIKSIMULATOR

Mit diesem professionellen Entwicklungs-Programm können logische Schaltungen direkt am Bildschirm konzipiert und getestet werden. - Gute Nachrichten nicht nur für die Hardware-Bastler, die sich zu oft mit brandblasenübersäten Fingern das Lötfett aus den übernachtigten Augen wischten...

20/4



CPU Z80

**NDR-Computer als universelles Computersystem**  
Ein typischer Dialog

20/8

**Die Kapazität von Nickel-Cadmium-Akkus** mit dem Z80-Einsteigerpaket gemessen

20/12

**Tastenkodier-Programm**  
Tasten mit oft wiederholten Eingabebefehlen belegen

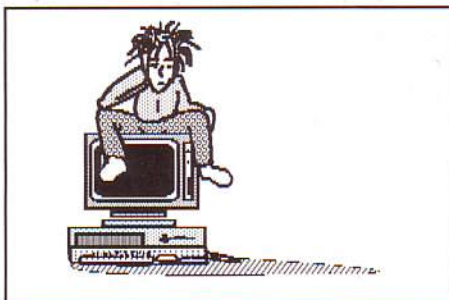
20/15

**Tips für alle Programmierer von Utilities unter FLOMONCG**

20/18

**Ökosystem eines Interpreters**  
Hinter die Kulissen geschaut

20/16



**Dialog über die Fähigkeiten des NDR-Computers** (Seite 20/8)

CPU 680xx

**Logisch-physikalisches Verwirrspiel**

(Dritter Teil)

20/19

**JADOS im Aufwind**

Vorstellung der Version 3.0

20/21

**RIP THE CHECKER**

Neue Version des Schachprogrammes unter JADOS

20/24

**Die neue Version von DATEI 2 ist da!**

Erweiterte Möglichkeiten

20/25

**Residente Treiber unter JADOS**

Erstellung und Installationstips

20/26

**OS-9/6800**

Was ist eigentlich ein Modul?

20/29

mc - modular AT

**Praktische Tip's zum MC-Modular-AT**  
Probleme bereinigt...

20/33

**Der Menübaukasten**  
MS-DOS für Anspruchsvolle

20/34

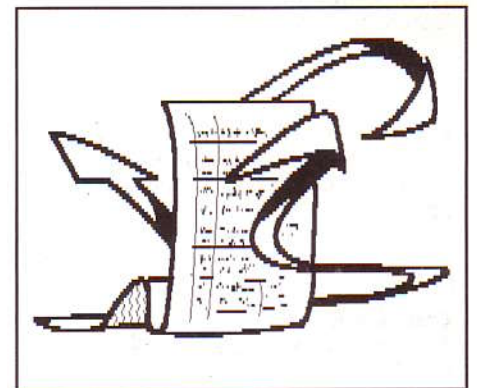
**Der Weg zur FTZ-Nummer**  
Steinige Wege bei der Produktion des MC-Modular-AT

20/36

**Hintergrundinformationen zu MS-DOS 4.0**

Die neue Version hat es in sich...

20/38



**Zeiger, Listen, Warteschlange**  
Grundlagen für Einsteiger aller Systeme

Impressum:

LOOP Zeitung für Computerbauer

Herausgeber: Gerd Graf

Redaktion: Rolf Dieter Klein, Gerd Graf, Christoph Köhler

Gestaltung: Christoph Köhler,  
MARTEN

Druck: Karl-Heinz Rieder, Kempten

Herstellung und Anzeigenverwaltung:  
GES GmbH, Magnusstraße 13,  
8960 Kempten

Anzeigenpreisliste 4/88

Rubriken

Editorial

Editorial 20/3

**In eigener Sache**

3. Software-Partner-Tagung im Allgäu 20/7

**Für Einsteiger aller Systeme**

Zeiger, Listen, Warteschlange und warum man auf keinen Fall darauf verzichten kann 20/11

**Wunschkonzert**

Softwarewünsche (damit bei den Software-Partnern keine Langeweile aufkommt) 20/13

**Neu lieferbar**

Buchhaltung für den 68K auf Diskette 20/25

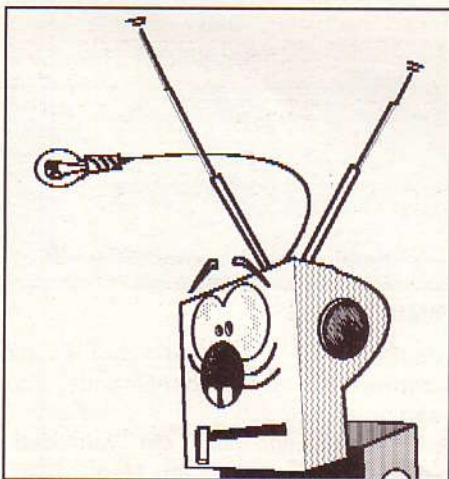
**Hardware**

Wait-Logik der ROA256/1M macht Schwierigkeiten bei schnellen Prozessoren 20/30

**Der 2988**

Start der Serie in dieser LOOP: Einleitung - Die Geschichte des 2988 20/32

Comic-Strip 20/37



Kontakte 20/37

Impressum 20/2

**Ein bewegtes Jahr...**

... innerhalb der Computerbranche nähert sich dem Ende - Zeit, etwas Rückschau zu halten.

Das Jahr war geprägt durch einen weiteren harten Wettbewerb der "Kompatiblen" - keine Zeitschrift ohne vergleichende Tests, bei denen mehr oder weniger aussagekräftige Balkendiagramme die minimalen Zeitunterschiede zum Berechnen der n-ten Primzahl darstellen.

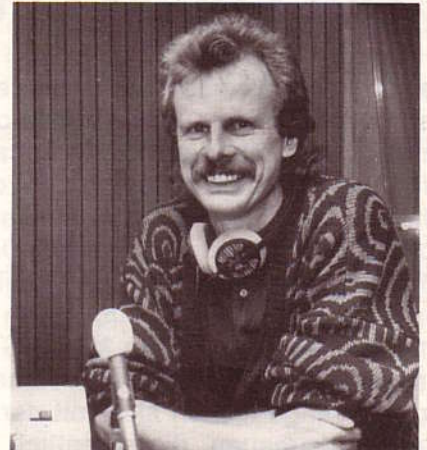
Etwas Würze in den Wettbewerb brachte die extreme Preissituation im Speichermarkt, wobei uns die Japaner wieder einmal bewiesen, wer in der Mikroelektronik das Sagen hat.

Trotz aller Gleichmacherei gibt es auch "andere" Computer, darunter den NDR-Computer. Mit einem nunmehr sehr starken Bein in der "kompatiblen" Rille erlaubt er dennoch die Kreativität und die Freiheitsgrade, die man in der kompatiblen Menge vielleicht vergeblich sucht. Dies bewies auch wieder die Kemp-tenner Software-Partner-Tagung, bei der enorme Entwicklungen auf der Hardware- wie auf der Software-Seite vorgestellt wurden.

Die **CPU8088** - oben als kompatibles Bein bezeichnet - wird seit einiger Zeit ausgeliefert und kommt gut an. Kleiner Wermutstropfen ist die langsame Ausgabe am Bildschirm mit der GDP64K. Hier haben wir fürs erste Quartal 1989 die Lösung parat: Sie ermöglicht den Anschluß einer "Standard" IBM-Graphik-Baugruppe über den neu vorzustellenden Bus-Adapter. Damit ist die Ausgabe auf Hercules, EGA oder VGA kein Problem mehr.

Einen weiteren positiven Nebeneffekt zeigt dieser Bus-Adapter: Auch der Anschluß der "normalen" Festplatte mit OMTI-Controller wird möglich sein.

Brandneu und revolutionierend innerhalb der Ausbildung ist der **Logik-Simulator**, ein hervorragendes Software-Produkt von Rolf-Dieter Klein. Digitale Schaltungen können nicht nur am Bildschirm simuliert



werden; die Ein- und Ausgänge können auch mit richtiger, realer Hardware verbunden werden. Log-Sim läuft derzeit auf IBM und Kompatiblen und auf C64 in einer eingeschränkten Version.

Der NDR-Computer wächst weiter: RDK hat noch viele Pfeile im Köcher! Anfangs 89 wird zunächst einmal das lange erwartete Betriebssystem **OS/9** ausgeliefert - zu einem, wie wir meinen, äusserst attraktiven Preis von unter DM 600.-, inclusive einem C-Compiler. Viele Software-Partner passen ihre Produkte bereits an OS/9 an.

Im neuen Jahr wird es auch mit dem NDR-System weiter nach "oben" gehen: Der erste **Transputer** läuft bereits und wird Sie in diese faszinierende Technik einführen. Aber auch Verbesserungen der "normalen" Baugruppen stehen auf dem Plan. Lassen Sie sich überraschen.

Im Frühjahr finden Sie uns auf der Didacta in Stuttgart und, wie gewohnt in Halle 6, auf der CeBit in Hannover. Besuchen Sie uns: Wir sind für jede Kritik, und sei sie auch positiv, dankbar.

Ein frohes Weihnachtsfest und ein gutes neues Jahr wünscht

Gerd Graf

Gerd Graf

# Der Logiksimulator

Die Herstellung neuer hochintegrierter Schaltkreise wäre heute ohne Logik-Simulatoren nicht mehr denkbar. Wo früher komplexe Schaltungen erst real aufgebaut werden mussten, werden diese heute lediglich über den Bildschirm eines entsprechend großen Computers eingegeben. Das Verhalten der Schaltung wird dann einfach simuliert.

Nun ist diese Form der Simulation auch für den "Hausgebrauch" möglich. Der hier beschriebene Logik-Simulator läuft auf jedem MS-DOS Rechnern und dem C64 ; der Preis liegt bei DM 250.-

Das vorgestellte Produkt kann aber noch mehr als sein großer Bruder: Reale, physikalische Eingänge und Ausgänge lassen sich mit der simulierten Schaltung verbinden. Damit sind richtig funktionierende Geräte aufzubauen, die eigentlich nur im Speicher des Computers existieren.

Heute sind "nur" die digitalen Grundfunktionen zu simulieren; in einer geplanten Erweiterung werden alle gängigen TTL-ICs verfügbar sein, um den Simulator auch in der Fertigung oder im Prüffeld einsetzen zu können.

## Das Programm wird gestartet

Die Spannung ist groß: Wie soll diese Simulation funktionieren?

Nun, wie üblich bei neuen Programmen: Erst starten und dann die Bedienungsanleitung lesen.

## Erste Überraschung:

Das Programm ist recht narrensicher. Es funktioniert (automatisch) mit allen Bildschirmkarten, von der einfachsten Monochrom-Karte über Hercules, CGA, EGA bis zu VGA. Der Logik-Simulator verwendet trotz der graphischen Bildschirmoberfläche nur den Standard-IBM-Zeichensatz. Damit sind auch Ausdrücke und Hardcopies vom Bildschirm kein Problem.

Zur Bedienung benötigt man eine Microsoft- oder kompatible- Maus. Ist eine Maus nicht vorhanden, so wird die Maussteuerung über die Cursor-Tasten sowie die Leer- und Eingabetaste simuliert. Das genügt zum Ausprobieren, zum sinnvollen Arbeiten muß man aber wirklich eine Maus haben.

## Die erste Schaltung entsteht

Nach dem Laden des Programms erscheint der in Bild 1 dargestellte Bildschirm.

Wir wollen ganz langsam anfangen und zunächst mal eine simple Schaltung aufbauen: Ein UND-Gatter.

Am oberen Teil des Bildschirms erscheint eine Menü-Zeile. Mit der Maus (oder Tastatur) positioniert man nun

einen Pfeil auf das gewählte Hauptmenü und betätigt den Maus-Knopf: Popp, das in Bild 2 gezeigte Untermenü erscheint. Deshalb nennt man eine solche Bedienungsoberfläche auch "Pop-Up-Menü". Durch Herunterziehen der Maus werden die entsprechenden Punkte des erschienenen Untermenüs ausgewählt: Wir wählen das UND-Gatter.

simulieren: Man findet sie im Hauptmenü UTIL ( für Utilities) unter "SWITCH" und "LED".

Wie schon vorher das UND-Gatter wird nun am Bildschirm positioniert:

Die Schalter zweckmäßig in der Nähe der Eingänge, die LED beim Ausgang. Eine sehr positive Eigenschaft des Programmes macht sich hier bemerkbar: Ein unsichtbares "Raster", das es

Logiksimulation auf Mikrocomputern

Digitalelektronik ganz anders

Geräte-Entwicklung leicht gemacht

Geräte am Bildschirm entwickeln

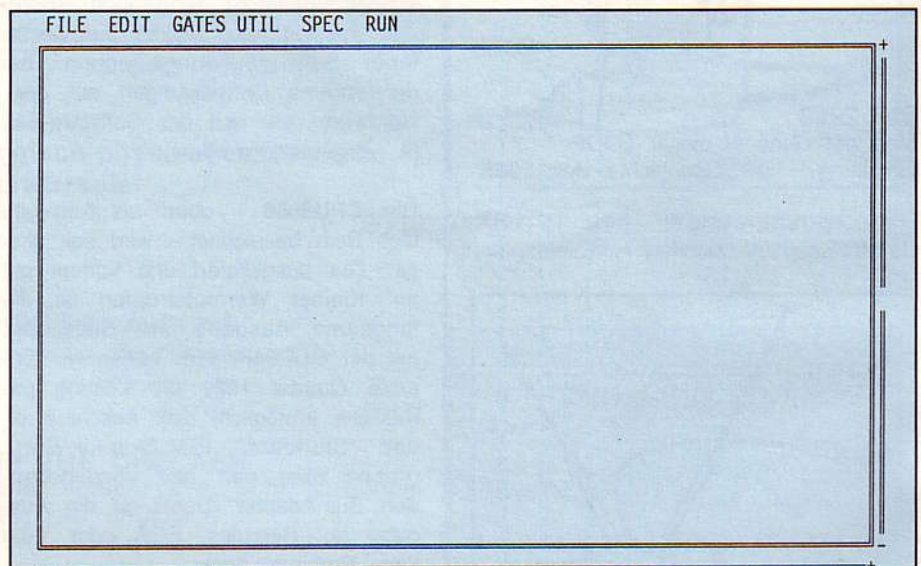


Bild 1: Eröffnungsbild des LOGIKSIMULATORS

Sofort nach dem Loslassen der Maus - Taste erscheint nun das DIN-Schaltbild eines UND-Gatters und folgt brav unseren Bewegungen. Wir positionieren es irgendwo in der Mitte und klicken noch einmal auf den Mausknopf: Das Gatter ist positioniert.

Ein UND-Gatter hat zwei Eingänge, an die zweckmäßig Schalter angeschlossen werden, und einen Ausgang für z.B. eine Leuchtdiode (LED). Natürlich lassen sich diese Elemente ebenfalls

erlaubt, die Elemente problemlos rechteckig oder nebeneinander zu positionieren.

Nun müssen nur noch die Verbindungen getätigt werden: Im Menü EDIT wird LINE aufgerufen. Ein Mausdruck kennzeichnet den Beginn einer Linie, ein zweiter das Ende: So werden die Elemente miteinander verbunden.

Das Ergebnis zeigt Bild 3; bei der Verbindung muß man darauf achten, daß wirklich genau die "Zapfen" der Baue-

lemente miteinander verbunden werden und Leitungen nicht in der Luft hängen bleiben. Nach etwas Eingewöhnung kein Problem.

### Die Simulation beginnt

Nun, bis hierher hat uns noch nichts überrascht: Jedes CAD-Programm

UND-Gatter ändert ihre Farbe - oder ihren Grauwert, je nach Video-Interface - eine logische "1" liegt an!

Die Spannung steigt: Der zweite Schalter wird betätigt, auch hier ändert sich die Farbe, das UND-Gatter schaltet, die Ausgangsleitung wird aktiv und die (simulierte) Leuchtdiode "leuchtet",

oder weniger Drähten und einem schwarzen IC eigentlich nichts.

So am Bildschirm macht das schon mehr Spaß: Man SIEHT die Schaltung und die logischen Zustände auf den Leitungen.

Schnell ein anders Gatter probiert: Das UND-Gatter wird gelöscht und einfach an diese Stelle ein EXOR-Gatter positioniert:

Testen, es funktioniert!

Nun kommt der Spieltrieb: Mal einen Oszillator holen und statt eines Schalters anschliessen: Es funktioniert; man kann sogar die Taktfrequenz des Oszillators während der Simulation einstellen!

### Das simulierte Oszilloskop

Spätestens nach Einführen des Oszillators und etwas komplexeren Schaltungen scheint den Entwicklern aufgegangen zu sein, daß die Anzeige des logischen Pegels nur auf den Leitungen nicht ausreicht: Man sollte Signale auch zeitbezogen sehen.

Was lag bei einem Logik-Simulator näher, als auch ein Oszilloskop zu simulieren? Es hat acht Kanäle, die an beliebige Punkte der Schaltung angeschlossen werden können.

Bild 5 zeigt eine etwas komplexere Schaltung, ein Schieberegister aus D-Flip-Flops aufgebaut. Die Oszilloskop-Kanäle sind am Taktgenerator und an den Ausgängen der Flip-Flops angeklemt.

Jederzeit während der Simulation kann diese gestoppt und das Oszilloskopbild eingeblendet werden, wie Bild 6 zeigt.

### 3000 Gatter, 200 facher Bildschirm

Natürlich ist die Größe einer Schaltung nicht auf einen Bildschirm beschränkt: mittels der unten und rechts angezeigten "Scroll-Bars" kann der Bildschirm über eine etwa 10 x 20 fache (Bei C 64 etwa 10 x 10) Fläche verschoben werden. Etwa 3000( ca 600 beim C64) Gatter lassen sich so zu funktionierenden Schaltungen zusammenbauen.

Das gesamte Programm ist in "C" entwickelt und optimiert worden; so gehen auch bei größeren Schaltungen die Rechenzeiten nicht ins Unermessliche, sondern steigen lediglich linear an. Der Simulator simuliert natürlich auch Schaltzeiten der Gatter, wie man aus Bild 6 erkennen kann.

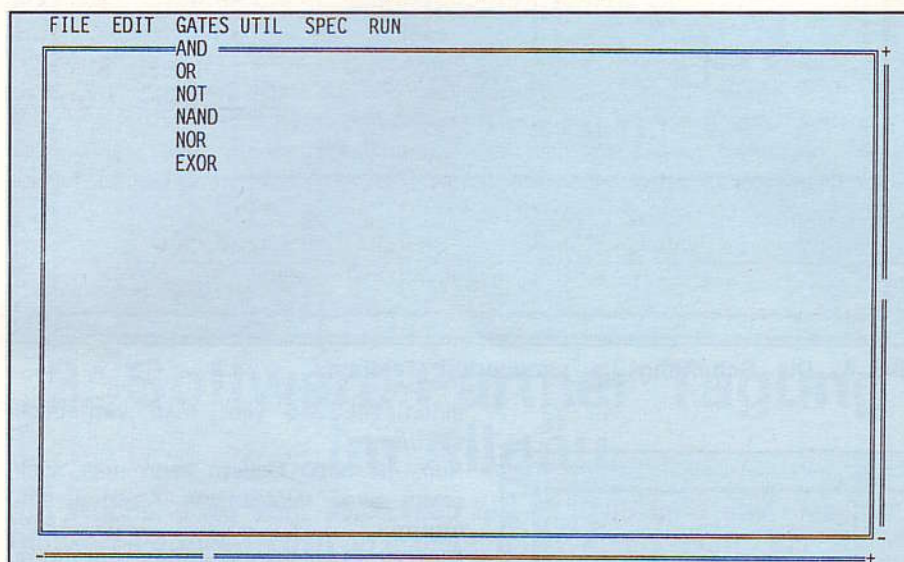


Bild 2: Komfortable Pop-Up-Menüs garantieren schnelles Handling

kann das auch.

Nun wollen wir unsere Schaltung aber ausprobieren und den Simulator zum Leben erwecken:

Im Menü RUN wählen wir den TRACE-Modus und beobachten gespannt den Bildschirm: Nichts rührt sich, lediglich die Menü-Leiste ändert sich, nur noch RUN ist sichtbar.

Kurz Überlegt: Natürlich, wir müssen die Schalter ja auch noch einschalten! Nun, ganz einfach: Pfeil mit Cursor in einen Schalter und anklicken. Aber jetzt: Die Leitung vom Schalter zum

was durch einen \* gekennzeichnet wird. (Bild 4)

### Neue Formen der Ausbildung

Wir erinnern uns noch an unsere ersten Schritte im Bereich der Digitalelektronik: Wir haben das UND-Gatter gesehen, die Wahrheitstabelle gelesen und dann geglaubt - oder auch nicht.

Vielleicht haben wir früher oder später einmal eine Schaltung aufgebaut und ausprobiert und schon ein bisschen mehr geglaubt.

Gesehen haben wir aber außer mehr

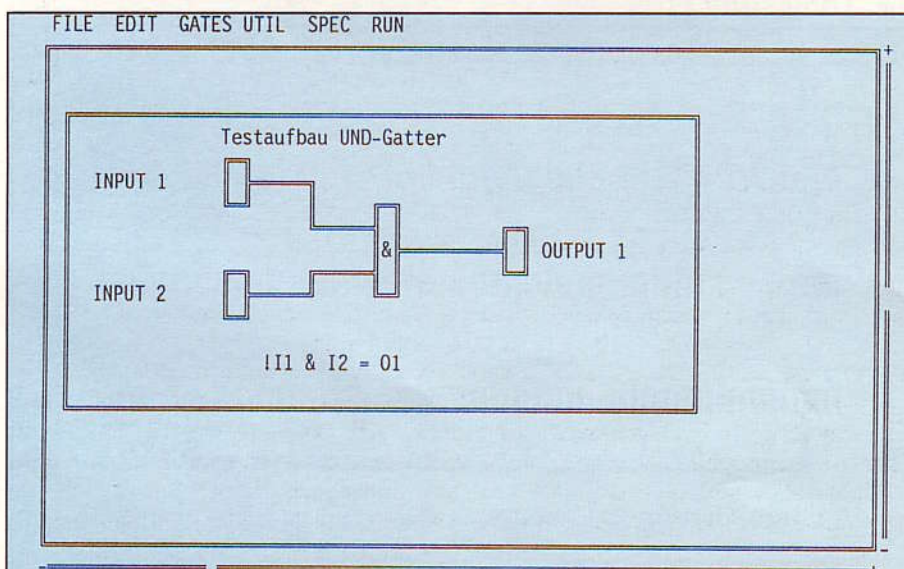


Bild 3: Testaufbau mit einem UND-Gatter

**Verbindung zur Außenwelt**

Interessant wird die Verbindung zur Außenwelt. Hier bietet der Hersteller ein relativ preiswertes Interface an, das acht digitale Ein- und Ausgänge, acht 8-bit-Analog-Eingänge und vier Motor-Ausgänge für Gleichstrommotoren bis zu 16 V, 1.5 A beinhaltet.

Das Interface ist übrigens für den C64 und den IBM das gleiche: Beim C64 wird es direkt in den Expansions-Port gesteckt, beim IBM wird eine Anpassungs-Baugruppe und ein Flachbandkabel eingesetzt. Die Anpassungsbaugruppe stellt auf der anderen Seite am Kabel den C64-Port her.

Bei der IBM-Version muß man das Interface nicht unbedingt benutzen: Hier kann man eine beliebige (digitale) I/O-Karte verwenden und im Simulator deren Port-Adresse und Bit-Nummer eingeben.

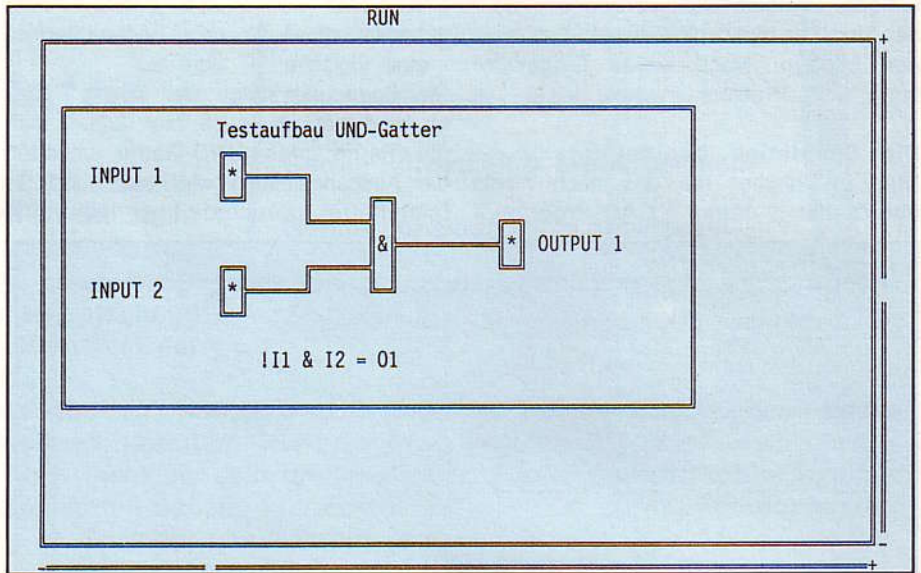


Bild 4: Die Schaltung im simulierten Testlauf

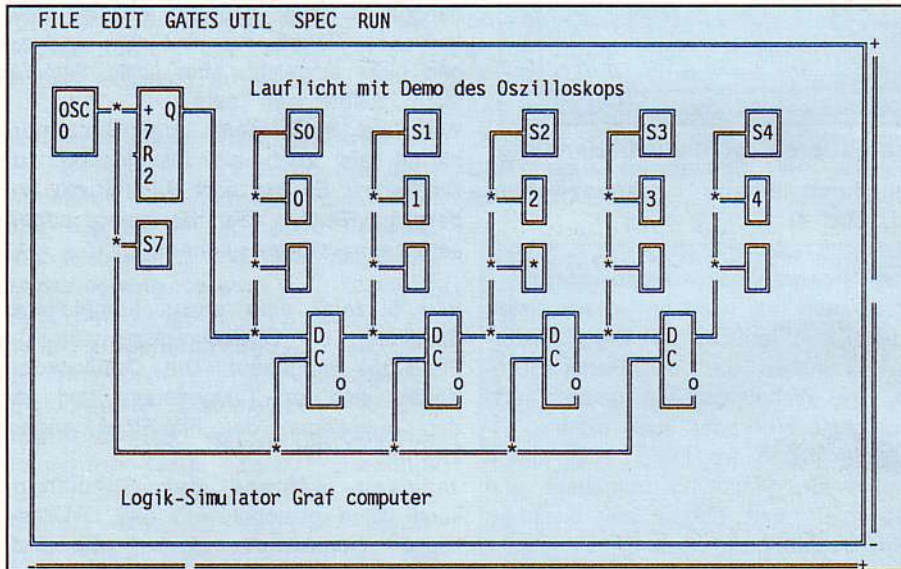


Bild 5: Komplexere Schaltung auf dem LOGIKSIMULATOR

In Verbindung mit dem Interface eröffnen sich interessante Anwendungsmöglichkeiten: So ist eine vollautomatische Zimmerblumengießanlage leicht zu realisieren. Zwei Drähte in den Blumentopf, über + 5V an den Analog-Eingang, und eine Wasserpumpe aus dem Modellbaukasten an den Motor-Ausgang angeschlossen! Den Rest erledigt eine kleine Schaltung mit dem Simulator: Triumph der Technik. Im Ernst: Selten haben wir so einfach Digitaltechnik gelernt.

Natürlich muß man sich darüber im Klaren sein, daß man keine - zumindest heute noch nicht - komplexen Schaltungen aufbauen kann. Der Simulator verwendet eben nur die Grundelemente der Digitalelektronik;

einen 74LS245 wird man vergeblich suchen.

Nun, mit 3000 Gattern kann man auch schon ganz interessante Anlagen aufbauen:

Die Ein-Adress-Maschine aus dem digitaltechnischen Praktikum kommt mit weit weniger Gattern aus.

**Ausblick**

Die Entwickler des Logik-Simulators - oder sollte er besser Logik-Emulator heißen? - haben noch viel vor.

So ist zunächst an die Einführung von "Makros" gedacht: Schaltungsteile, die getestet sind, werden in eine Box verpackt und können dann, versehen mit der entsprechenden Anzahl Ein- und Ausgängen, in die Bauelemente-Bibliothek des Logik-Simulators aufgenommen werden.

Diese und die anderen Erweiterungen

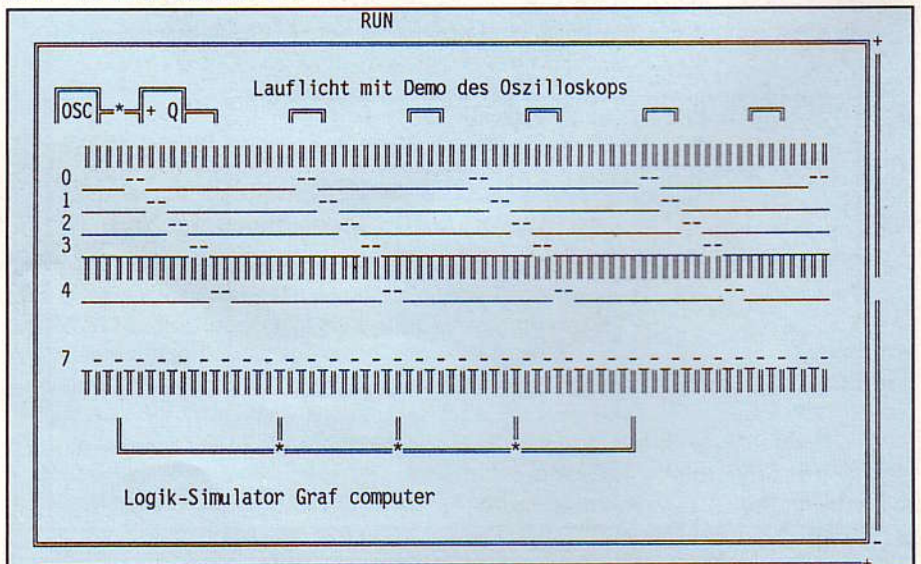


Bild 6: Simuliertes Oszilloskop zeigt Realzeit-Signale

werden jedoch nur auf der IBM-Version funktionieren, da der Speicherplatz und die Rechengeschwindigkeit des C64 bereits voll ausgeschöpft sind.

Der nächste Schritt wird noch viel weitreichender werden: Aus der Schaltung wird der Logik-Simulator ein Programm erzeugen können, das in einer Hochsprache zur Verfügung stehen wird. Das Programm kann dann - nach Übersetzung - auf einem beliebigen Zielrechner, z.B. einem Ein-Platinen-Computer ablaufen. Der Zielrechner,

versehen mit entsprechender I/O, verhält sich dann genau so wie die gezeichnete Schaltung, also das simulierte Gerät.

Neben den Hochsprachen PASCAL und BASIC sollen auch C und, besonders interessant, OCCAM erzeugt werden.

Daß dies alles keine Phantasiegebäude sind, dafür bürgt **Rolf-Dieter Klein**, der Entwickler des Systems. Er hat sich mit seinen bisherigen Entwicklungen vom zehn Jahre alten mc-CP/

M-Computer über den modularen NDR-Computer bis hin zum Parallel-Rechner-System PARWELL einen guten Namen in der deutschen Mikrocomputer-Szene gemacht.

Der Logik-Simulator wurde während der COMDEX SPRING in Atlanta, USA, vom Lieferanten, der Kemptener Firma Graf Elektronik, in der "Höhle des Löwen" vorgestellt. Nicht nur das enorme Interesse war beachtlich: Auch die Tatsache, daß diese deutsche Entwicklung bis jetzt einmalig auf dem Weltmarkt zu sein scheint.

Christoph Köhler

### 3. Software-Partner Tagung im Allgäu

Am Wochenende vom 30.09.1988 bis 02.10.88 fand das diesjährige Software-Partner-Treffen in Kempten im Allgäu statt. Bis auf die wenigen entschuldigenden SW-Partner stellten sich am Freitag alle dem vollgestopften Tagungsprogramm. Die Tatsache, daß mittlerweile die meisten Gesichter bekannt waren, lockerte die Atmosphäre schnell auf. Die Schulungsräume füllten sich schon am ersten Tag zügig mit den gestellten und den mitgebrachten, selbstgebaute NDR-Computern.



Bild 1: Teilnehmer der 3. SW-Tagung

Nach der Begrüßung durch H. Graf bekamen die SW-Partner das Wort. Dabei konnten sie Ihre neu überarbeiteten Programme bzw. neuen Routinen vorführen, was sich als hochinteressant herausstellen sollte. Im einzelnen

wurden folgende Programme gezeigt:

**Klaus Janßen** führte das neue JADOS V3.0 vor, das seinesgleichen sucht. Wie auch dem LOOP-Artikel "JADOS im Aufwind" zu entnehmen ist,

besticht die neue Version durch eine noch größere Anwenderfreundlichkeit, die Übernahme von vielen Anwenderwünschen und einigen wertvollen neuen Befehlen und Funktionen. Ein kleiner Lohn für die viele Programmierarbeit war für H. Janßen die Begeisterung der anwesenden Teilnehmer.

Daß es das neue 680XX Grundprogramm V6.0 in sich hat, bewies **Ralf Dombrowski** in seiner Vorführung. Er zeigte den SW-Partnern einige neue Kniffe im Umgang mit dem bereits den meisten bekannten Grundprogramm.

Das neue FLOMONCG wurde von dem Autor **Rüdiger Nahm** vorgeführt. Besonders die "680xx SW-Partner" staunten, wie umfangreich und vielseitig die Programme auf der Z80-Seite werden können. Wie von H. Nahm längst bekannt, sind seine Programme hervorragend dokumentiert und vorbildlich beschrieben.

Mit dem neuen "RIP THE CHECKER" (alias: Schach unter Jados) stellte **Klaus Rumrich** unter Beweis, daß sein Programm weiterhin wächst und mittlerweile schon mit Maus und Farbe betrieben werden kann. Rumrich zeigte außerdem, daß die Zugberechnung wesentlich beschleunigt und verbessert wurde.

**Kei Thomson** führte den Einsatz einer EGA-Karte mit einem NDR-680xx Computer auf seinem eigenen Rechner vor. Als Bildschirm diente der neue NEC-Multisynch II. Auf das dazu notwendige Know-How wird in einer der

nächsten LOOP näher eingegangen.

Das neue 680xx-Spiel "SPEEDY" zeigte **Ralf Ullrich** zur Freude der Anwesenden. Dieses Programm wurde von allen SW-Partnern einstimmig als das bisher beste Spiel für den NDR-Computer bewertet.

Verschiedene andere Programme(-teile) wurden von einigen weiteren SW-Partnern vorgestellt. Bis zur Vorstellung in der LOOP bzw. zur Freigabe werden auch diese Routinen noch nicht näher beschrieben.

Am Nachmittag wurden in einem Gedankenaustausch viele brisante Themen zum Werdegang des NDR-Computer besprochen. In der anschließenden Diskussion wurden einige neue Impulse an das GES-Team weitergeleitet.

Auch in den gemütlichen Abendstunden konnten die vielfältigen Ideen der SW-Partner kaum gebremst werden. Gerade in diesen Stunden wird oft so manches Thema angesprochen, welches sonst nicht im Schriftwechsel einbezogen wird.

Am Samstag startete **Rolf-Dieter Klein** mit der Vorführung der neuen Baugruppe CPU 8088 für den NDR-Computer. Zum Staunen der SW-Partner beschrieb er den Aufwand des umgeschriebenen BIOS und die Möglichkeiten, externe Baugruppen (z.B. Standard-Festplatte vom mc-modular AT) anzuschließen.

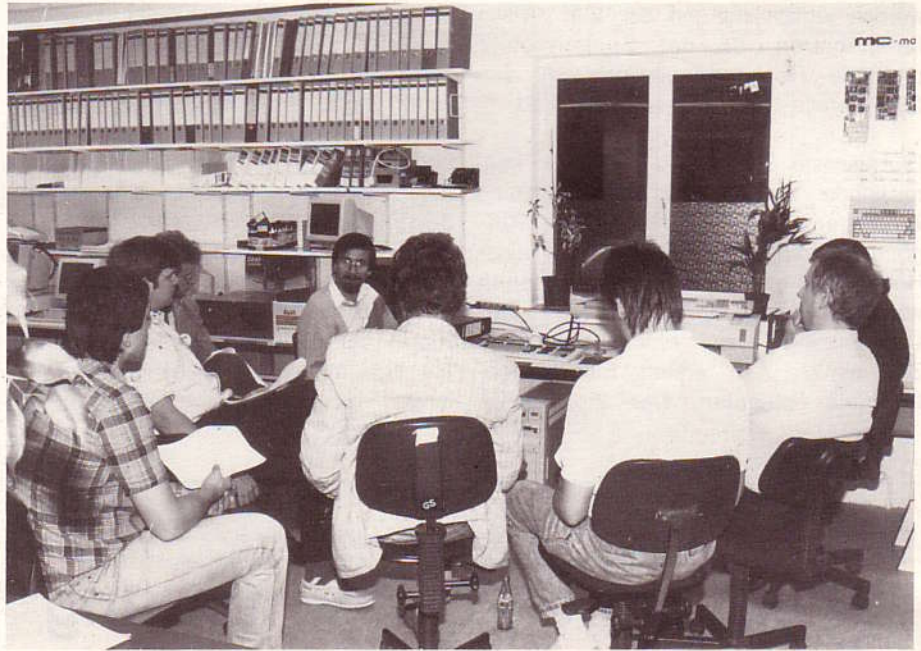


Bild 2: Arbeitsgruppe "OS-9" mit A. Wiegand (Mitte)

Anschließend gab **Volker Wiegand** grünes Licht für OS-9, das neue Betriebssystem für den NDR-680xx Computer. Seine didaktisch vorbildliche Schulung war für die Teilnehmer eine sehr interessante Einführung in das erwähnte Programm. OS-9 (siehe ab LOOP 19) ist ein Betriebssystem mit Mehrplatz- und Mehrbenutzerfähigkeit. Insgesamt hebt es sich der Meinung der Teilnehmer nach wesentlich von den bekannten Betriebssystemen wie JADOS, CP/Mxxx oder MS-DOS ab. In der von H. Graf und H. Klein geführten Diskussion wurde an fachlichen Argumenten, feurigen Beiträgen und

humorvollen Anmerkungen nicht gespart.

Die wenig verbliebene Zeit wurde noch zur Bildung von Arbeitsgruppen mit den Themen "OS-9", "JADOS", "LOOP" und "Spiele" genutzt.

In den abendlichen Stunden des Samstages wurden die Fachbücher in die Regale bzw. die Mäuse in die Schubladen gestellt.

Der Mann am Klavier spielte zum Schweinshax'n-Essen den Auftakt für einen, wie wir meinen, recht gemütlichen Abend.

## NDR-Computer als universelles Computersystem

Das modulare System des NDR-Computers, das immer gepflegt und erweitert wurde, hat sich als das Universalsystem in der Ausbildung, Selbstbildung und Hobby erwiesen.

Mit der Freigabe der CPU8088 und des Betriebssystem OS9 (Dezember 88) sind dem NDR-Computer (fast) keine Grenzen mehr gesetzt. Es gibt kaum ein vergleichbares System, das eine solche Vielfalt an CPU-Baugruppen, Peripherie-Schnittstellen und Betriebssystemen bietet.

Um etwas Klarheit in die Informationsflut zum NDR-Computer zu bringen und dem typischen Computer-Einsteiger einen besseren Überblick zu verschaffen, wurde dieser Artikel verfasst.

Das nachfolgende Zwiegespräch könnte zwischen einem Computer-Interessenten und einem technischen Berater von GES entstehen.

**Interessant:** Ich will eigentlich seit langem etwas mit einem Computer machen. Aber es gibt soviel und ich traue mir nicht zu, mich für den einen oder anderen Computer zu entscheiden. Ich habe immer das komische Gefühl, daß ich mich eventuell doch falsch entschieden habe.

**GES-Berater:** Was wollen Sie denn mit dem Computer machen?

**I:** Ja ... eigentlich so alles was man mit einem Computer machen kann. Hmm



... Spiele, Buchführung, "lernen", eigene Programme, kleine Steuerungen,...

**G:** Das ist wirklich ziemlich viel, was ihr Computer können sollte. Die von Ihnen angegebenen Anwendungsgebiete kann fast jeder Computer. Es kommt natürlich darauf an, auf was Sie die Schwerpunkte legen.

**I:** In erster Linie wollte ich eigentlich so einen Computer einmal kennenlernen, d.h. ich will im Groben ungefähr wissen wie das Ding funktioniert, aber dann müßte ich wohl erst einen Kurs belegen und ein Ausbildungssystem kaufen, und zu guter Letzt steht mir dann dieses Ausbildungssystem in der Ecke und für meine Anwendungen muß ich dann wieder ein anderes System kaufen.

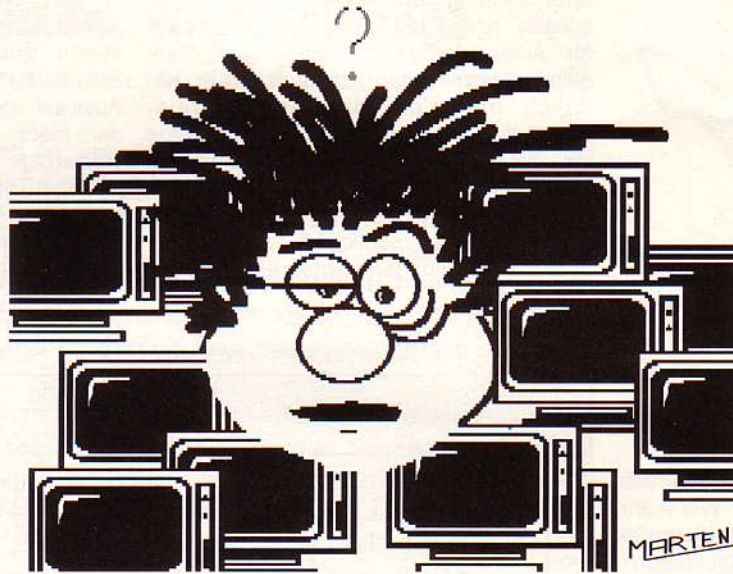
**G:** Beim NDR-Computer sind Sie durch den modularen Aufbau völlig frei in der Wahl Ihres Systems. Sie können mit dem Einsteigerpaket beginnen und damit die grundlegende Funktionsweise eines Rechners und eines Mikroprozessors ergründen und dann Ihr System je nach Ihren Vorstellungen ausbauen.

**I:** Ja, aber ich kann doch nicht mit einem Ausbildungssystem "IBM kompatibel" werden, oder professionelle Betriebssysteme fahren, oder professionelle Steuerungen oder Regelungen aufbauen. Diese Ausbildungssysteme sind doch meistens mit "uralten" Prozessoren ausgerüstet, die ja wohl für die Ausbildung noch gut sind, aber nicht für professionelle Anwendungen.

**G:** Das modulare System des NDR-Computers ermöglicht alle diese Anwendungen. Natürlich können Sie nicht mit dem Einsteigerpaket MS-DOS fahren, aber Sie können Ihr System ständig erweitern, ohne daß Sie irgendwelche Systemeinheiten wegwerfen bzw. in die Ecke stellen müssen. Anhand eines kleinen Beispiels will ich Ihnen dieses Konzept erklären:

Sie sind Computer-Neuling und möchten mit dem Einsteigerpaket beginnen, um den Computer von Grund auf kennenzulernen. Haben Sie sich dann das

## Welche Hardware ???



Grundlagenwissen angeeignet und die Scheu vor dem Computer abgelegt, können Sie die nächste Ausbaustufe anvisieren. Sie erweitern Ihre Anlage um einen Monitor, eine Tastatur, diverse zusätzliche Schnittstellen und die dazugehörigen Programme und verfügen über einen vollständigen Computer.

Dann steht einer zeitgerechten Programmentwicklung und -handhabung nichts mehr im Weg.

**I:** Also wie beim PC ?

**G:** Das können Sie festlegen. Sie können sich gleich noch ein Laufwerk einbauen und mit der CPU8088 dann wie mit einem PC arbeiten und natürlich dann auch die Programme, die auf IBM-Kompatiblen laufen, verwenden.

**I:** Dann habe ich wohl einen IBM kompatiblen NDR-Computer?

**G:** Ja. Sie können aber den NDR-Computer auch mit anderen Betriebssystemen betreiben. Sollten Sie sich für die bewährte CPU Z80 entscheiden, können Sie den Rechner zum vollständigen CP/M 2.2 - System ausbauen. Dabei bieten sich die preiswerten Programmpakete wie "Wordstar", "Multiplan" oder "dBASE II" an. Oder sie wollen den NDR-Computer zum professionellen Steuerungscomputer ausbauen? Dann empfiehlt sich der NDR-Computer mit den 68000-Prozessoren von Motorola. Hier gibt es den 68008 (8 Bit Prozessor), den 68000 (16 Bit) und den 68020 (32 Bit

Prozessor). Auf dem NDR-Computer läuft also sogar ein 32 Bit Rechner.

**I:** Warum ein 32-Bit Prozessor auf einem Ausbildungssystem?

**G:** Sie sehen selber, daß wir hier den Ausbildungsbereich verlassen haben. Das System ist jetzt ein professioneller Rechner mit vielfältigen Anwendungsmöglichkeiten geworden. Mit den zahlreichen Peripheriebaugruppen zum NDR-Computer können Sie mit dem NDR-Computer zahlreiche Aufgaben, wie Messen, Steuern, Regeln, Überwachen, verwirklichen.

Vielfach wird der NDR-Computer als Entwicklungssystem verwendet. Mit den 680xx Prozessoren können Sie sogar das Betriebssystem OS-9 betreiben; ein modernes Multitasking und Multiuser Betriebssystem. Dieses Betriebssystem wird hauptsächlich für professionelle Zwecke eingesetzt.

**I:** Aha...

**G:** Ich sehe schon, das hat Sie im Moment noch erschlagen, aber ich wollte nur kurz aufzeigen, was der NDR-Computer für Möglichkeiten bietet.

**I:** Mit welcher Ausbaustufe sollte ich als Laie eigentlich beginnen?

**G:** Wenn Sie sich wirklich fundiert in die Hardware einarbeiten wollen, empfehlen wir natürlich das Einsteigerpaket. Der weitere Ausbau hängt davon ab, was Sie später machen wollen. Ansonsten können Sie einen Rechner entsprechend Ihren Vorstellungen konfigurieren (zusammenstellen).

**I:** Ich habe gehört, daß es den NDR-Computer auch als Bausatz gibt. Spar ich damit Geld, oder was soll das bezwecken ?

**G:** Natürlich sparen Sie dabei auch Geld. Aber wichtiger ist wohl auch, daß Sie sehen, daß ein Computer kein Hexenwerk ist und problemlos von jedem gebaut werden kann. Das Einzige, was Sie dazu benötigen, ist gutes Lötwerkzeug und etwas Lötpraxis, die Sie sich aber problemlos aneignen können.

Wie Lernen ???



I: Ja, ... wie kann ich denn dann was über den Computer erfahren. Wie kann ich mich ausbilden? Oder bilden Sie mich aus mit Kursen oder Seminaren?

G: Die ideale Ausbildung umfaßt einen theoretischen und einen praktischen Teil. Sie haben sicher auch schon festgestellt, daß sie dann am schnellsten und besten lernen, wenn sie den Lernstoff nicht nur theoretisch durchkauen, sondern das Erlernete praktisch anwenden müssen. Aus diesem Grund bieten wir sowohl die theoretischen Unterlagen als auch das praktische Werkzeug (Computer) an. Dabei setzen wir auf die Fernausbildung. Sie erhalten von uns Kurse (von der Fa. Christiani) und die nötige Hardware, um sich entsprechend auszubilden.

I: Was sind das für Kurse?

G: Wir bieten Christiani Kurse für das Einsteigerpaket, SPS-Programmierung und Assembler Programmierung an. Außerdem bieten wir zahlreiche Fachbücher über Prozessoren, Betriebssysteme und Anwendersoftware an.

I: Kann man von Ihnen auch Schaltpläne oder technische Unterlagen bekommen?

G: Wir haben bei unserer Hard- und Software immer alles veröffentlicht. Zu jeder Baugruppe bekommen Sie ein umfangreiches Handbuch, in denen nicht nur der Schaltplan veröffentlicht, sondern auch beschrieben wird. Auch die Aufbau- und Testanleitung für die Selbstbauer, sowie Softwarebeispiele sind enthalten. Bei der von uns oder unseren Softwarepartnern veröffentlichten Software können Sie immer auch die Quellistings erhalten.

I: Kann ich das Ganze überhaupt verstehen?

G: Wenn Sie die Kurse bearbeitet haben werden Sie relativ schnell solche Schaltpläne oder Listings lesen können. Sicher ist jeder Anfang schwer, aber sie werden, wie unsere Erfahrung zeigt, hier relativ schnell Fuß fassen.

So oder ähnlich geht es bestimmt vielen Anwendern, oder ist es vielen Anwendern gegangen die in die "Computerei" einsteigen wollen bzw. eingestiegen sind. Das größte Problem stellt für einen Anwender sicher das mangelnde Wissen dar. Dadurch ist er natürlich den Computer-Verkaufsstrategen hilflos ausgeliefert. Deshalb wurde bei der Entwicklung des Systems auf Modularität und Offenheit geachtet. Sie können sich mit diesem System ausbilden und es schließlich zu einem brauchbaren System auszubauen.

Um die Universalität des NDR-Computers zu demonstrieren, haben wir Ihnen in den Bildern die im NDR-Computer einsetzbaren CPU's und Betriebssysteme aufgelistet:

Ein sehr wichtiges Hilfsmittel bei der Auswahl der richtigen Ausbaustufen ist der mehr als 200 Seiten umfassende, vierfarbige GES-Katalog. Dieser wurde von Anfang an als Nachschlagewerk konzipiert und kann gegen eine Schutzgebühr von 10 DM auch telefonisch bestellt werden. (0831-6211)

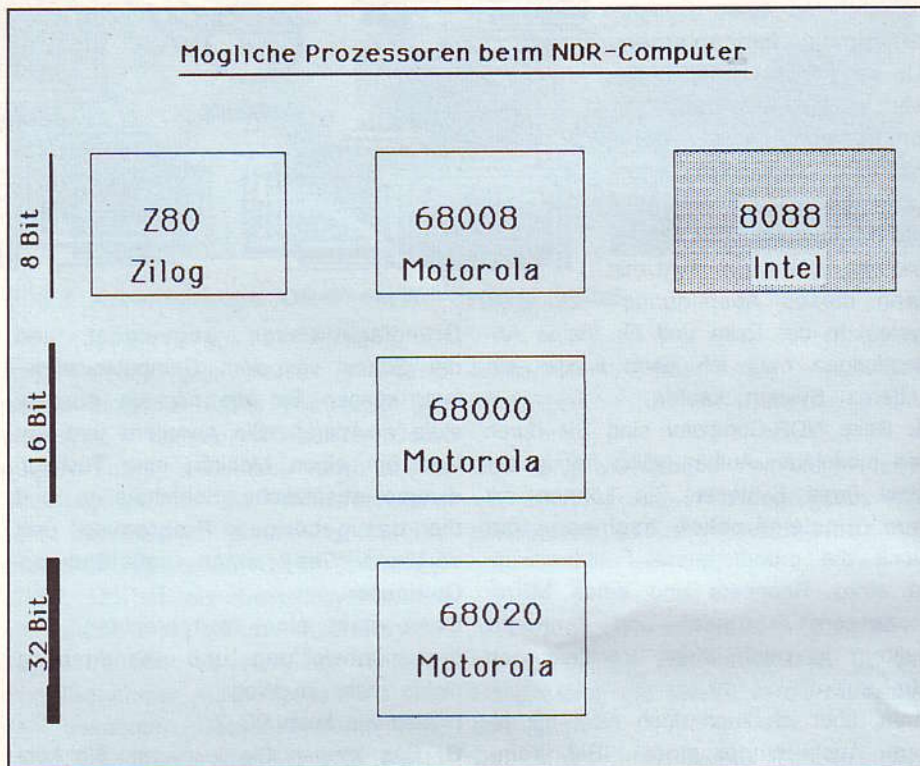


Bild 1: Welche Zentraleinheiten sind möglich ?

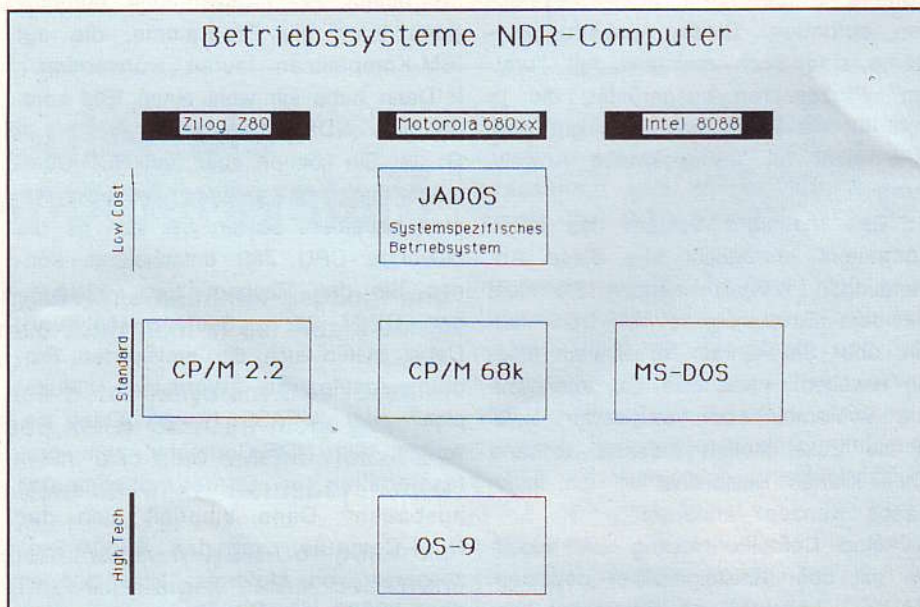


Bild 2: Welche Betriebssysteme für welche CPU ?

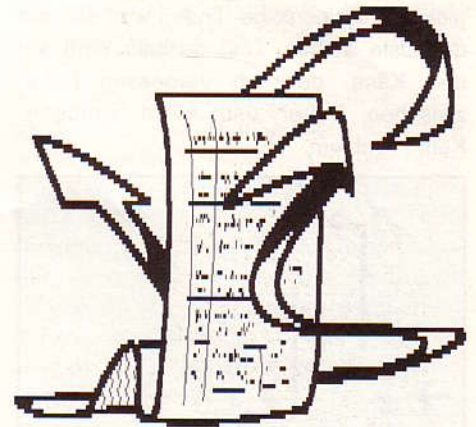
Martin Husemann

Für Einsteiger aller Systeme: Grundlagen

## Zeiger, Listen, Warteschlange

und warum man auf keinen Fall darauf verzichten kann

Sie haben sicherlich auch in den vergangenen Jahren, zur Hochzeit der C64 Welle, von 15-jährigen gehört, die mehrere Tausend Mark verdient haben, indem sie Computerspiele programmierten. Und obwohl dies ganz und gar nicht meiner Idee einer sinnvollen Freizeitgestaltung entspricht (ich schätze den Arbeitsaufwand für viele Spiele auf mindestens ein Mannjahr), wollte ich doch gerne einen Blick hinter die Kulissen werfen und wissen, wie das funktioniert.



Der Computer bewegt furchtbar viele Dinge gleichzeitig über den Bildschirm, alle von verschiedenen Programmen gesteuert, aber es steckt nur ein Mikroprozessor dahinter. Das gleiche Problem gibt es natürlich auch in anderen Bereichen, z.B. bei der Steuerung eines Chemiewerks oder eines Flugzeuges (nur bewegen sich dort Dinge nicht nur über den Bildschirm!). Nach eigenen Versuchen bei der Programmierung von Spielen habe ich inzwischen herausgefunden wie man das macht und möchte hier versuchen, es zu erklären.

Wichtige Grundlagen dazu sind zunächst Zeiger und Listen. Listen sind genau das, was Sie sich darunter vorstellen, nämlich mehrere Dinge, die unter- oder hintereinander stehen. Zum Beispiel schreibe ich gelegentlich Einkaufslisten, und GES führt eine Kundenliste (im Computer, nehme ich an). Bleiben wir bei der Einkaufsliste:

Butter, 2 Pakete  
Milch, 4 Liter  
Brot, 1500g

Da ich sehr vergesslich bin, muß ich mir auch solch kurze Listen aufschreiben, oder: im Computer speichern (den trage ich natürlich überall bei mir). Also entwerfe ich mir eine Computergerechte Datenstruktur, um die Informationen geschickt abzulegen. Ein Datensatz (RECORD) sieht so aus:

Artikel: TEXT  
Menge: ZAHL  
Einheit: TEXT

Wenn ich nun drei solche Datensätze irgendwo in den Speicher des Computers bringe, muß ich mir ihre ADRESSEN merken (die Nummer des Speicherplatzes, an dem sie beginnen).

Die einfachste Methode wäre es, sie auf ein Blatt Papier zu schreiben, aber dann könnte ich natürlich gleich alles zu Papier bringen. Also muß ich auch ihre Adressen irgendwie im Speicher des Computers festhalten. Das geht zum Beispiel, indem ich den Datensatz vergrößere:

Artikel: TEXT  
Menge: ZAHL  
Einheit: TEXT  
nächster: ADRESSE

Jetzt brauche ich mir nur noch die Adresse des ersten Postens auf der Liste zu merken. Im Feld "nächster" des ersten Datensatzes finde ich dann die Adresse des zweiten, in dessen "nächster"-Feld dann die des dritten, und dort? Irgendwie muß ich erkennen, daß hier die Liste zuende ist. Ich könnte dazu ein weiteres Feld in meinen Datensatz einfügen, das mir sagt, ob dieser Posten der Letzte auf der Liste ist (eine sogenannte Flagge). Damit würde ich aber Speicherplatz verschwenden, denn es geht auch einfacher: die Adresse 0 ist kein vernünftiger Wert für das "nächster"-Feld, denn bei 0 ist entweder kein Arbeitsspeicher, oder dieser Speicher wird vom System belegt. Also schreibe ich in das "nächster"-Feld des letzten Postens einfach eine 0. Um die Adresse 0 von der Zahl 0 zu unterscheiden, benutzt man üblicherweise das Symbol NULL oder NIL dafür, im Speicher steht aber trotzdem

die 0, denn für den Mikroprozessor sind Adressen auch nur Zahlen.

Meine Liste sieht jetzt so aus:

```
start: 124

124:
  Artikel: Butter
  Menge: 2
  Einheit: Pakete
  nächster: 150

150:
  Artikel: Milch
  Menge: 4
  Einheit: Liter
  nächster: 210

210:
  Artikel: Brot
  Menge: 1500
  Einheit: g
  nächster: 0
```

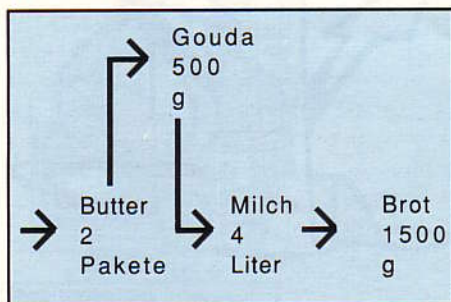
Man nennt das Feld "nächster" einen Zeiger (POINTER), denn es zeigt, bildlich gesprochen, auf den nächsten Datensatz.

Da die wirklichen Adressen der Daten nicht interessant sind, kann man die Liste auch grafisch darstellen:

```
→ Butter → Milch → Brot
   2     → 4     → 1500
   Pakete Liter g
```

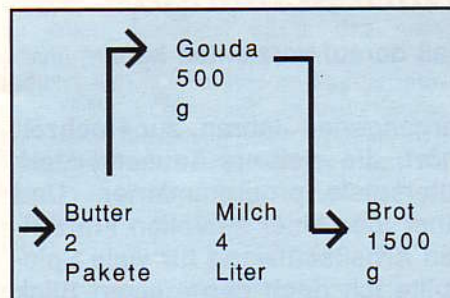
Wie bereits erwähnt bin ich sehr vergesslich, sodaß es durchaus vorkommt, das ich eine Einkaufsliste ändern muß, weil ich etwas vergessen habe. Ausserdem möchte ich die Liste so ordnen, daß ich bei meinem Gang durch den Supermarkt die Artikel in der

gleichen Reihenfolge finde, wie sie auf der Liste stehen. Und deshalb muß ich den Käse, den ich vergessen habe, zwischen Butter und Milch einfügen. Kein Problem:



Ich biege einfach den "nächster"-Zeiger des Datensatzes "Butter" um zum neuen Datensatz "Gouda". Den alten Zeiger von Butter nach Milch kopiere ich in den neuen Datensatz - und schon stimmt die Liste wieder. Aber halt, mir

fällt gerade ein, daß ich gestern schon Milch mitgebracht habe! Gut, das Streichen in einer Liste ist sehr einfach:



Milch steht zwar noch im Speicher, aber nicht mehr auf der Liste.

Soweit dieses kleine Beispiel. Zusammenfassend noch einmal das wichtigste:  
- Zeiger sind Adressen

- Zeiger sind Hilfsmittel, um Listen im Computer zu verwalten
- In Listen (so wie hier vorgestellt) kann sehr einfach (und schnell) an beliebigen Stellen eingefügt werden.
- In Listen kann sehr schnell gelöscht werden.

Es gibt noch einige interessante Abwandlungen dieser einfachen Listen, mit denen man dann noch einiges mehr anstellen kann. Dazu, und wie der Computer mit Hilfe von Listen vieles gleichzeitig tun kann, komme ich in der nächsten Folge.

**Geplante Fortsetzungen:**

- Warteschlangen, Programmbeispiel mit dem Einsteigerpaket
- Anwendung in Spielen, Pascal-S Beispiel

Ulrich Kracker

## Die Kapazität von Nickel-Cadmium Akkus mit dem Z80-Einsteigerpaket gemessen

(Teil 2)

Im ersten Teil des Beitrages (LOOP 19) wurde das Grundkonzept und die Hardware des Prototypes einer Nickel Cadmium Test Station vorgestellt. In diesem abschliessenden Artikel geht es nun zunächst um die Software für das verwendete Einsteigerpaket.

**Was muß die Software können?**

Direkt nach dem Start gelangt man in eine kleines Auswahlmenü, woraus der gewünschte Betriebsmodus (Nur Entladen, nur Aufladen oder kombinierter Ablauf) ausgewählt wird. Entsprechend der Betriebsart fragt das Programm nach den Parametern: Widerstandswerte, Zellenzahl, Nennkapazität und Spannungsteilverhältnis am AD-Wandler. Mit diesen Informationen kann das Werk nun begonnen werden: Die Hintergrundroutine zeigt lediglich wichtige Betriebsdaten an und überprüft die je nach Betriebsart gültige Aufhörbedingung. An Betriebsdaten werden im Entlademodus die bisher gemessene Akkukapazität in mAh ausgegeben und im Auflademodus die bisher vergangene Ladezeit in Minuten und Stunden. Die Aufhörbedingung beim Entladen des Akkus lautet: Ist die momentane Akkuspannung kleiner als die Entladeschlussspannung? Die Aufladung

wird abgebrochen, wenn die Ladungsmenge, die seit dem Beginn des Ladens in den Akku 'hineingepumpt' wurde, die 1.4-fache Nennkapazität übersteigt. Die eigentliche 'Musik' macht aber eine Interruptroutine, die im Entlademodus, wie auch im Lademodus einmal pro Sekunde aufgerufen wird: Zuerst wird ein AD-Wert einlesen, in die entsprechende Spannung umgerechnet und abgespeichert. Danach errechnet sich die Interruptroutine über das Ohmsche Gesetz den momentan fließenden Strom. Zusammen mit dem Messergebnis der letzten Sekunde wird nun ein mittlerer Strom errechnet ('Trapezbildung'). Da das Interruptprogramm nun annimmt, dieser mittlere Strom sei in der letzten Sekunde in dieser Stärke geflossen, wird eben dieser Rechenwert als eine kleine Kapazitätsfläche (mA \* 1s) interpretiert. Folglich kann der unveränderte Zahlenwert (in mAs) zu der bisher ermittelten Kapazität (auch in mAs) hinzugezählt werden.

Wurde beispielsweise ein mittlerer Strom von 100mA errechnet, entspricht die Aufaddierung dieses Wertes zur Gesamtkapazität einer Teilkapazität von 100mAs. Dieser Vorgang ist das diskrete Äquivalent zur Integration:

$$C = \int i dt \text{ (in As)}$$

Da intern in mV und mA, bzw. in mAs gerechnet wird, ist das 32-Bit Arithmetik-Paket von Jens Decker in LOOP 12/7 nach geringfügiger Modifikation verwendet worden (im dort abgedruckten Programmlisting befinden sich übrigens auch ein paar Übertragungsfehler!).

Als Abschluss der Interruptroutine wird noch eine Software-Uhr nachgestellt, damit das Hintergrundprogramm im Auflademodus die bereits vergangene Zeit darstellen kann.

**Ansätze zur Weiterentwicklung, Tips zum Selbermachen**

Als ersten Punkt zur Verbesserung der Zuverlässigkeit der Anlage ist die Steigerung der EMV zu nennen. Was ist die 'EMV'? EMV bedeutet 'Elektromagnetische Verträglichkeit' und drückt die Anfälligkeit eines Systems gegenüber elektrischen und magnetischen Störungen aus. Daneben wird im Allgemeinen auch noch das Ausmaß der Störungen betrachtet, die wiederum vom System ausgehen. Woher kommen nun die von aussen einwirkenden Störungen? Sie kom-

men zum einen über den Äther und zum anderen über die Netzzuleitung. Berühmte 'Verseucher' sind: Leuchtstofflampen beim Einschalten, Phasenanschnittsteuerungen (u.a. Dimmer), größere Energieverbraucher (Motoren), vor allem beim Einschalten und so weiter. Eine 'hohe' EMV ist hier des-

ends abgelaufen ist), monostabile Kippstufe, die beim Ausbleiben der periodischen Triggerungen seitens der Software einen Reset auslöst und somit das Programm von vorne starten läßt.

Dazu wäre es sinnvoll einen akkugepufferten RAM-Bereich (SBC3!) zu haben, damit auch nach einem Netzausfall alle Betriebsparameter erhalten bleiben und das Programm fortfahren könnte.

als monostabile Kippstufe beschaltet, als Interrupterzeuger dienen (siehe Bild 1).

Am Ende der Interrupt-Routine muß dann noch die monostabile Kippstufe erneut ausgelöst werden, um die nächste Verzögerungszeit sicherzustellen. Diese Schaltung wurde aus [3] entnommen und nicht selbst ausprobiert. Der genaue Abgleich des Monoflops könnte relativ einfach vorgenommen werden, da ja während des Aufladens die Uhrzeit vom Hintergrundprogramm ausgegeben wird.

AD-Wandler: Im Prinzip sind alle AD-Wandler, die mindestens mehrere Messungen pro Sekunde durchführen können geeignet, also auch die meisten integrierenden Typen, wie etwa der CA3162, der ICL 7109 oder der ICL7135. Natürlich können auch die NDR-Baugruppen AD8\*16 und ADDA verwendet werden.

Da das Programm schön gegliedert ist, muß nur die AD-Wandleroutine auf den konkreten Fall zugeschnitten werden. Die Arithmetikroutinen sind so ausgelegt, daß sie auch 16-Bit-Werte verarbeiten können. Bei mehreren AD-Kanälen können durchaus auch mehrere NC-Akkus gleichzeitig 'bearbeitet' werden, vorausgesetzt die Lade/Entladeschaltungen sind in entsprechender Zahl vorhanden und die Messintervalle liegen weit genug auseinander, damit genügend Zeit zum Rechnen verbleibt.

Dieser Beitrag ist als Anregung all den Lesern gewidmet, die (mindestens) ein NDR-Einsteigerpaket besitzen, nach einer neuen Anwendung dafür suchen und ein paar leere oder 'zweifelhafte' NC-Akkus zuhause haben. Intressenten stehe ich gerne für weitere Auskünfte bereit.

#### Literatur:

[3] Norbert Hoffmann: 'Digitale Regelung mit Mikroprozessoren' Vieweg-Verlag Braunschweig 1984

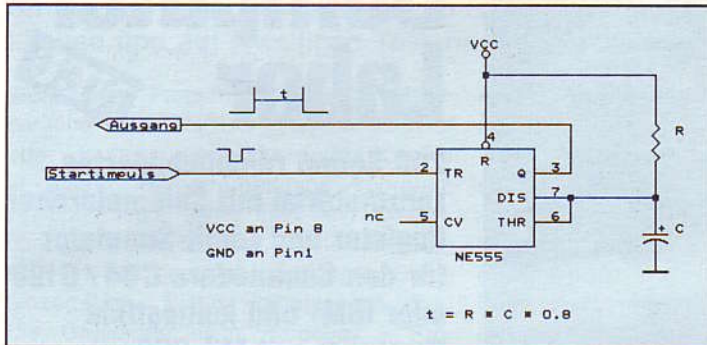


Bild 1: Ein NE555 als Interruptgenerator beschaltet

#### Alternativen über Alternativen...

halb wichtig, weil das Akku-Center schliesslich über viele Stunden korrekt arbeiten muß, ohne 'abzustürzen'. Mir ist es schon einmal passiert, daß ein Akku gerade entladen wurde, aber der Timer offensichtlich keinen Interrupt mehr erzeugt hatte. Die Folge war, daß keine Spannungsmessungen mehr durchgeführt wurden und der Akku praktisch unkontrolliert tiefentladen wurde.

Wie kann man aber ein System robuster gegenüber Störungen machen? Gegen Störungen von der Netzseite her ist ein Netzstörfilter das relativ einfachste Mittel, wobei man aber hiervon keine Wunder erwarten darf. Ein Gehäuse aus Metall schirmt noch am besten gegen Einflüsse aus der Luft. Wenn es dann trotzdem zu einem Systemabsturz kommt, müßte ein 'Watchdogtimer' den Rechner nach wenigen Sekunden wieder auf die Füße stellen können. Ein Watchdog ist bekanntlich eine retriggerbare (=erneutes Auslösen der Kippstufe ist möglich, noch bevor die laufende Verzögerung voll-

Lade/Entladeschaltung:

Beim Mustergerät werden zwei Relais mit je zwei Wechslern für den automatischen Betrieb eingesetzt. Eine einfachere Möglichkeit, die allerdings wenig Komfort bringt, wäre es, die Teilschaltungen aus Bild 3a, bzw. Bild 3b (LOOP 19) behelfsmäßig mit 'Krokodilstricken' fliegend aufzubauen und dieses Gebilde an den AD/Wandler anzuschliessen. Jedoch darf dann das Programm nicht unbeaufsichtigt ablaufen, da am Ende ja keine automatische Abschaltung mehr erfolgen kann! Man könnte sich auch eine Schaltungsvariante mit kontaktlosen (Leistungs-)Schaltern und Analogmultiplexern vorstellen. Dabei könnte dann im Entlademodus eventuell auch das kurzzeitige Zuschalten eines zweiten Belastungswiderstandes realisiert werden, um den Innenwiderstand des Akkus zu ermitteln.

Zeitgeber:

Anstatt des programmierbaren Timers 8253 könnte sicherlich auch ein analoges Timer-IC, zum Beispiel der NE555

## Wunschkonzert:

Mein Wunsch an die Softwarepartner wäre ein bescheidenes DTP-Programm unter Jados oder PC/M 68k:

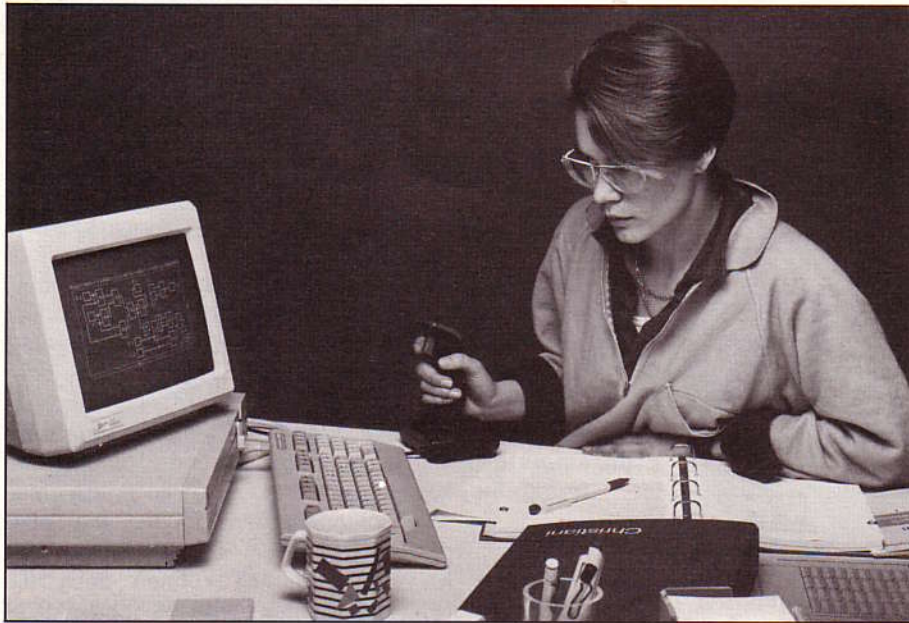
- Mit der Möglichkeit Bilder in einem verbreiteten Format abzulegen, zB. Image-Format.
- Mit der Möglichkeit, diese Bilder ohne größere Verrenkungen in einen Text einbauen zu können, zB. in einen der

860XXer Editors.

- Mit der Möglichkeit eines Grafikeditors und einer bescheidenen Symboldatei für den Anfang.
- Mit bescheidenen CAD-Funktionen (Drehen, Vergrößern) und Freihandzeichnen.
- Mit einer flexiblen Druckeranpassung und Plottertreiber
- Mit einer offenen Benutzerschnittstelle, sodaß man einige Module oder die

anderen Softwarepartner einbauen kann, z.B. die Sprachausgabe, die leider so eine Schnittstelle nicht offenlegt. Die Schnittstelle aber sollte für Hochsprachen zugänglich sein.

- Mit einem Filterprogramm, daß man Bild- oder Textdateien oder gemischte Dateien mit anderen MS-Dos Programmen weiterverarbeiten kann.
- Und zuletzt, ein preiswerter GES-Scanner !



## Kompakt-Kurs Digital- Computer- Labor **NEU**

200 Seiten reichbebildertes Lehrmaterial mit Sammelordner, Register und Logik-Simulator für den Commodore C64 / C128 oder IBM- und kompatible Computer mit MS-DOS. Gesamtpreis DM 448,-

### Lernen Sie die Bauelemente und Grundschaltungen der Digital-technik mit einem ausgezeichneten Logik-Simulator kennen.

#### Der Lehrgangsaufbau

Der Lehrstoff ist in zwei Fachgebiete aufgeteilt; die Digitaltechnik und die Steuerungstechnik. In der Digitaltechnik machen Sie zunächst Bekanntschaft mit den Grundlagen. Theorie und Praxis gehen Hand in Hand. Sie lernen einfache Schaltungen mit dem Logik-Simulator auf dem Bildschirm darzustellen und auszutesten.

Eine Zeichnung auf dem Bildschirm, die man ausprobieren kann – ein einmaliges Erlebnis sinnvoller Computeranwendung. Nahezu alle Schaltungen, die im Lehrmaterial beschrieben sind, können mit dem Logik-Simulator ausgetestet werden. Das Aufbauen von Schaltungen hat damit ein Ende. Sie werden begeistert sein.

#### Digitaltechnik

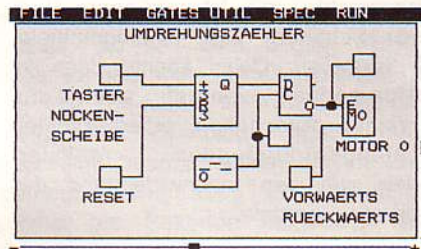
Die Digitaltechnik behandelt alle wichtigen Bauelemente. Wir beginnen mit den logischen Verknüpfungen und gehen über Speicher und Flip-Flops, Schaltalgebra, Zahlensysteme, Codierschaltungen, bis hin zur Mikroprozessortechnik, um nur einige der interessanten Themen zu nennen.

#### Steuerungstechnik

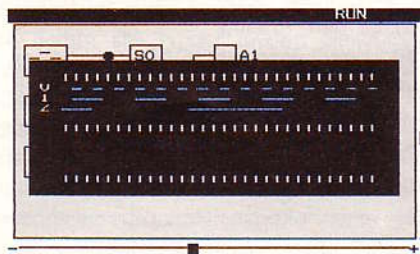
Die Steuerungstechnik hingegen zeigt Ihnen die Anwendung der digitalen Bauelemente. Ablaufsteuerungen, Schaltverstärker, Schrittmotorsteuerungen und vieles mehr wird innerhalb des Kurses ausführlich und leichtverständlich behandelt.

#### Der Logik-Simulator

Sicher wird Sie der neue leistungsstarke Logik-Simulator begeistern. Über POP-UP-Menüs können die verschiedensten Funktionen angewählt werden. Digitale Verknüpfungen, Schalter, Leuchtdioden, Motor, Lautsprecher, Flip-Flops, Zeitgeber und Zähler sind abrufbare Elemente.



Damit dynamische Prozesse auch leicht verfolgt werden können, kann der Logik-Simulator sogar ein Oszilloskop darstellen. Einfach das Symbol des Oszilloskops anwählen, und schon kann man bis zu 8 Signale verfolgen.



#### Für wen ist der Kompakt-Kurs interessant?

Für alle, die sich in das Gebiet der Digitaltechnik einarbeiten wollen. Für alle, die den Computer als Werkzeug bei der Schaltungsentwicklung kennenlernen und ausnutzen wollen. Und natürlich für jeden, der sich mit Computeranwendungen und -steuerungen auskennen will.

#### Was Sie brauchen

Sie brauchen als Vorkenntnisse lediglich Elektronik-Grundlagen und natürlich einen Commodore C64 oder C128 mit Diskettenlaufwerk und Joystick oder einen IBM- oder kompatiblen Computer mit Mouse. Die Software (Logik-Simulator) ist in der Kursgebühr enthalten.

Fangen Sie jetzt an! Die Technik der Computeranwendungen wartet nicht. Bald wird es selbstverständlich sein, Schaltungen aller Art mit dem Computer zu entwickeln. Mit dem Wissen aus diesem Kurs liegen Sie dann ganz vorn.

# Christiani Fortbildung

Postfach 35 000 · 7750 Konstanz  
Telefon (0 75 31) 58 01-0

Jost-Reimer Hoof

# Tastenkodier - Programm

**Eigentlich wollte ich die Funktionstasten meiner Tastatur besser in die Programmierung einbinden. Daraus ist ein universelles Assembler-Programm geworden, mit dem 22 häufig gebrauchte Befehle oder Strings mit einer Länge bis zu 30 Buchstaben abgespeichert werden können und auf jeweiligen Tastendruck zur Verfügung stehen.**

Wenn man Programme in einer Hochsprache wie TURBO Pascal schreibt oder immer wiederkehrende Namen in einem Textverarbeitungs-Programm eintippen muß, sehnt man sich danach, über eine Taste oder Tastenkombination ein ganzes Wort oder einen String einzugeben. Nichts leichter als das! Die Grundidee beim Hardcopy - Programm wurde schon vorgestellt: die Tastatur-Abfrage - "Leitung" wird in das entsprechende Programm umgebogen. Beim Programm TAST-KOD gibt es die Möglichkeit, mit einer frei wählbaren Taste das Abrufen von Befehlen oder Strings auszulösen. Dann lassen sich 22 frei wählbare Tasten mit einem String "kodieren". Als auslösendes Zeichen schlage ich "\ (ASC-Zeichen 5Ch) vor, da ESC oder CTRL-Zeichen in vielen Programmen schon vorbelegt sind. Dann bietet es sich an, die ABC-Tasten - klein oder groß - zu nutzen. Wem z.B. das "J" nicht gefällt, verwendet vielleicht ein "Z". Das Schöne an diesem Programm ist die völlige Freiheit! Das Programm ist in Assembler mit ZEAT entwickelt, modular aufgebaut und bewußt nicht auf Programmkürze optimiert. So lassen sich Teile daraus auch für andere Programme nutzen.

## Bedienung

Folgende Menü - Punkte werden angeboten (Bild 1). Dabei bedeuten:

- **K** Hier legt man die auslösende Taste und die Tastenkodierung mit maximal 30 Buchstaben fest.
- **Z** Die ausgewählten Tastenkodierungen werden auf dem Bildschirm ausgegeben.
- **S** Die z.Z. vereinbarte Tastenkodierung wird als TAST-KOD.DAT abgespeichert. Eine schon vorhandene Datei dieses Namens wird in TAST-KOD.BAK umbenannt.
- **E** Die abgespeicherte Datei TAST-KOD.DAT wird in das Programm eingelesen und überschreibt den dort vorhandenen Eintrag. Ist keine Datei dieses Namens vorhanden, wird eine Fehlermeldung ausgegeben.

- **A** Aktivieren des Programms, siehe unten.

- **D** Deaktivieren = Abschalten der Tastenkodierung.

- **I** Hiermit wird eine Information über das Programm und seine Bedienung ausgegeben.

Beim Aktivieren wird der Datenspeicher mit den vorher aktualisierten Kodierungen und ein kleiner Programmteil nach F800h kopiert, und im BIOS bei EA81h der Sprung nach F003h in diesen Programmteil umgelenkt. Wichtig ist, daß die bei EA81h befindliche Adresse gesichert wird, um sie beim Deaktivieren zu nutzen. Außerdem wird die FLOMON-Adresse "LASTMON", die die letzte belegte Adresse beinhaltet, korrigiert. Deshalb läßt sich das Programm nicht zweimal hintereinander aktivieren; dazwischen muß immer deaktiviert werden, damit der Urzustand wieder hergestellt wird. Der im aktivierten Programm gewünschte Befehl oder String wird aufgerufen, indem zuerst die auslösende Taste und dann die entsprechende Taste gedrückt wird: z.B. sei "\ die auslösende Taste und unter B "Begin" abgespeichert. Der Aufruf "\B" ergibt ==> "Begin" auf dem Bildschirm und im System.

## Verträglichkeit

Das Programm TAST-KOD wurde erfolgreich bei ZEAT, TURBO Pascal und Wordstar 3.3 eingesetzt. Dabei lassen sich auf der Wordstar

und TURBO - Diskette verschiedene TAST-KOD.DAT - Files abspeichern.

Bei WINDOW kann es mit einer, meines Wissens noch nicht veröffentlichten Version Komplikationen geben, wenn WINDOW nach TAST-KOD aufgerufen wird, da der gleiche RAM-Bereich benutzt wird. Wählt man erst WINDOW und dann TAST-KOD, gibt es keine Probleme.

Mit dem neuen FLOMONCG konnte noch kein Test gemacht werden, in der zur Auslieferung kommenden Version wird dies berücksichtigt und dokumentiert.

## Bezugsbedingung

Da das Listing 11 DIN-A 4 - Seiten umfaßt, eignet es sich nicht zum Abdruck in der Loop.

Das Programm TAST-KOD (als COM-File 4 K und 20 K als Assembler-Quelle) wird auf der geplanten CPM-Tool-Diskette sein. Da eine Auslieferung bei GES noch nicht möglich ist, kann das Programm von mir bezogen werden: Bitte eine formatierte Diskette 3 1/2" oder 5 1/4" und einen Unkostenbetrag von 10,- DM in Briefmarken oder Verrechnungsscheck senden an:

Jost-Reimer Hoof  
Wilhelm-Ivens-Weg 87  
2305 Heikendorf b. Kiel  
Tel (0431-) 24 20 70

Ich kopiere zusätzlich zu dem Programm TAST-KOD noch folgende Programme als Quelle und COM-Files:

<b>COMPARE</b>	zum Vergleichen zweier Files
<b>FIND-ASC</b>	zum Suchen von ASC-Zeichen in unbekanntenen Programmen
<b>STATUS</b>	zum Feststellen, welche Ports im System sind

Diese Programme werden in der nächsten Loop besprochen.

```
C>TAST-KOD.COM
```

```
Tastenkodierungs-Programm von JoHo - Software
```

```
===== (1988)
```

```
A = Aktivieren des Kodierungsprogramms
D = Deaktivieren = Normalisieren
E = Einlesen der Tastenkodierungs-Datei
I = Information
K = Kodieren der einzelnen Tasten
S = Sichern der Tastenkodierung
Z = Zeigen der Tastenkodierung
Ende= jede Taste außer A, D, E, I, K, S, Z
```

Bild 1: Hauptmenü

Neue Serie

Dr. Hans Hehl

# Ökosystem eines Interpreters

Wer sich heute einen Computer wie den NDR-Rechner selber baut, möchte diesen nicht nur benutzen, sondern sozusagen hinter die Kulissen schauen. Dies gilt für Hard- und Software. Leider wird dies mit den modernen IBM-kompatiblen Rechnern immer schwieriger, was nicht nur für den Anfänger gilt.

## Zielsetzung

In dieser mehrteiligen Serie wird nun das Grundwissen vermittelt, mit dem Anfänger und Fortgeschrittene hinter die Kulissen des HEBAS-Basic-Interpreters für den NDR-Rechner mit dem Betriebssystem CP/M schauen.

Vom Autor wurde das Konzept dieser Serie in mehreren Wahlkursen über Elektronische Datenverarbeitung am Gymnasium Markt Schwaben erprobt.

## Speisenplan

Was wird in dieser Serie alles geboten? Eine der Höhepunkte wird eine "neue Programmiersprache" sein, nämlich ein BASIC-Interpreter mit deutschen (wer will, auch mit bayerischen) Befehlswörtern völlig ohne Englisch. Wie so etwas gemacht wird, ist ein Teil des Kurses. Dazu werden Utilities zum Disassemblieren selber programmiert und auch kommerzielle Debugging-Hilfen verwendet.

Bei der Besprechung des BASIC-Interpreters lernt der Anfänger Teile des CP/M-Betriebssystems wie BDOS und BIOS ebenso kennen wie elementare Grundlagen der Z80-Maschinensprache. Der fortgeschrittene Z80-Programmierer wird eigene BASIC-Befehle in den Interpreter einbauen können, da der Quellcode zur Verfügung steht.

## Trostpflaster

Bei der Besprechung des Interpreters wird immer das praktische Arbeiten mit dem Computer im Vordergrund stehen, also kein Trockenkurs! Wer ohne Diskettenbetriebssystem mit dem NDR-Rechner arbeitet, kann mittels BASIC-Interpreter im EPROM, mit Grundprogramm oder mit dem neuen FLOMONCG-Monitor ebenfalls mitmachen. Auch Besitzer des legendären mc-Computers sind nicht ausgeschlossen...

## Was brauchen wir für den Anfang?

Beim NDR-Rechner sind entsprechend dem modularen Gedanken verschiedene Ausbaustufen vorhanden. In der Serie werden besonders berücksichtigt:

### a) die Ausbaustufe ohne Diskettenbetriebssystem:

- Z80-CPU-Karte: da kann das Grundprogramm EGRU2000 im EPROM auf der Bank-Boot-Karte als Debugger Verwendung finden.
- die SBC-3-Baugruppe: ebenfalls mit dem Grundprogramm

Bei beiden Ausbaustufen kann der Interpreter EHEBAS in zwei EPROM's auf einer zusätzlichen Speicherkarte verwendet werden (Bank E000h, JMP2: Brücke nur bei A16). Das Grundprogramm wird als Monitor zum Analysieren verwendet. Wer den neuen FLOMONCG von R. Nahm verwendet, kann direkt auf andere Speicherkarten (Banks) zugreifen und benötigt das Grundprogramm nicht.

### b) das CP/M-Betriebssystem:

Umfangreiche Debugging-Software und HEBAS, auch im Quellcode, stehen auf Diskette zur Verfügung.

### Nicht ganz zur Serie passend:

Wer mal kurz nur mit der SBC-2 gearbeitet hat, wagt vielleicht jetzt den Sprung zur nächsten Ausbaustufe. Wer allerdings gleich mit der CPU 8088 und MS-DOS begonnen hat, schade .... Vielleicht wird HEBAS für MS-DOS einmal umgeschrieben.

### Jetzt gehts los...

Das verhexte Dezimalsystem: Oft benötigt man die binäre oder sedezimale Darstellung von Zeichen, auch mit gesetztem Bit 7, die z.B. in der Schlüssel-tabelle des Interpreters als Endzeichen verwendet werden.

Teil 1: Aller Anfang ist schwer  
LOOP 20

Teil 2: Werkzeug zum Knacken  
LOOP 21

Teil 3: Schnitzeljagd  
LOOP 22

Teil 4: Geheimsprache  
LOOP 23

Teil 5: Ein Boarisch-BASIC  
LOOP 24

Teil 6: BDOS und BIOS  
LOOP 25

Teil 7: Innereien  
LOOP 26

Teil 8: Auf die Plätze, fertig, los  
LOOP 27

Teil 9: Patchwork mit Variablen  
LOOP 28

Teil 10: Hexeneinmaleins  
LOOP 29

Das Programm DEZHEXBI.BAS in Bild 1 erstellt eine Umwandlungstabelle mit vier Spalten, bestehend aus Dezimalzahl, Sedezimalzahl, Binärmuster und soweit möglich mit dem ASCII-Zeichen. Anfangs- und Endwerte der Tabelle können im Bereich von 1 bis 255 eingegeben werden. Die Ausgabe erfolgt wahlweise am Bildschirm oder mit dem Drucker. Zahlen größer 255 können so nicht dargestellt werden. Die Variable W\$ ergänzt Dezimalzahlen auf drei Stellen. Es gibt aber auch Taschenrechner, die die Umwandlung in verschiedene Zahlensysteme ermöglichen. Aber selber machen ist schöner.

Auch hier gilt Murphy's Gesetz:

Ist ein Programm brauchbar, wird es sofort durch ein anderes ersetzt.

Grundsätzlich stellen die Programme nur Grundversionen dar, die sicher verbessert werden können. So müßte man z.B. falsche Eingaben durch entsprechende Programmzeilen abfangen.

### Fortsetzung folgt

Wie geht es weiter im nächsten Teil? Mit dem Grundprogramm, mit Monitor



```

100 REM dezhexbi.bas Dezimal-Hex-Binär-Liste
110 REM Algorithmus aus mc 12/82 Seite 44
115 REM angepaßt an HEBAS.COM
120 ZZ=0:
130 INPUT"von Dezimalzahl";VD:PRINT
140 INPUT"bis Dezimalzahl";BD:PRINT
150 : IF BD>255 THEN PRINT"zu große Zahl!":GOTO 140
160 INPUT"Bildschirm = 0, Drucker = 2";DV
170 PRINT:PRINT
180 FOR K = VD TO BD
190 : W$="": IF K<100 THEN W$=" " : Formatierung
200 : IF K<10 THEN W$=" "
210 : PRINT#DV," " ;W$;K;" " ;HEX$(K);" " ;
220 :
230 : FOR I = 7 TO 0 STEP -1
240 : PRINT#DV,SGN(K AND 2^I);
250 : NEXT I
260 :
270 : PRINT#DV," " ;CHR$(K):ZZ=ZZ+1
280 : IF ZZ/60=INT(ZZ/60) THEN FOR T=1 TO 12:PRINT#DV:NEXT T
290 NEXT K
300 END

```

Bild Nr. 1: Programm Dezhexbi.bas:

oder Debugger DTT.COM wird der Speicher des Rechners untersucht, wo sozusagen "der Interpret haust" und daher auch die Überschrift.

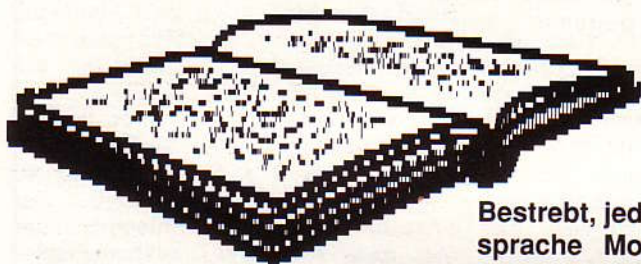
**Literatur:**

Klein, Rolf-Dieter, Rechner modular, Franzis-Verlag  
Murphy's Law, Arthur Bloch, Los Angeles

33	0021	0 0 1 0 0 0 0 1	!
34	0022	0 0 1 0 0 0 1 0	"
35	0023	0 0 1 0 0 0 1 1	#
36	0024	0 0 1 0 0 1 0 0	\$
37	0025	0 0 1 0 0 1 0 1	%
. . . . .			
104	0068	0 1 1 0 1 0 0 0	h
105	0069	0 1 1 0 1 0 0 1	i
106	006A	0 1 1 0 1 0 1 0	j

Bild Nr. 2: Liste des Programms

Für Sie gelesen!



von Herbert Schildt

## Modula-2 Einführungskurs

Bestrebt, jedem Anfänger die Grundkenntnisse der Programmiersprache Modula 2 beizubringen, führt der Autor Herbert Schildt den Leser durch den Inhalt des Buches.

Der erste Teil des Buches weist in die Hintergründe von Modula2, spricht deren Philosophie und Konventionen. Vergleiche werden zwischen anderen Sprachen wie Turbo Pascal und Basic gezogen. Erläutert wird der Vorteil eines Compilers gegenüber vielen Interpretern. Einfache Programmbeispiele werden durch Struktogramme aufgebaut.

Der zweite Teil führt wie gewohnt durch die Datentypen wie Konstanten, Dekla-

rationen, Operanten und Ausdrücke. Die Konstruktionen der verschiedenen Modulhierarchien und deren Einsatz als lokale Module bilden den dritten Teil. Hier wird zum ersten mal der entscheidende Unterschied dieser Programmiersprache aufbereitet. Fachbegriffe wie Coroutinen, Prioritäten, Portabilität oder Effizienz eines Programmes werden ebenso detailliert und einfach dargestellt wie der Einsatz von scheinbar gleichzeitigen Prozessen unter Modula2.

Die Programmcode-Beispiele in diesem Leitfaden können in der Regel mit jedem herkömmlichen Modula2 Compiler durchgearbeitet werden. Das Buch ist also durchaus für unter dem NDR Computer laufende Modula 2 Compiler geeignet.

Modula 2, McGraw-Hill Book Company GmbH, Hamburg, 1988, ISBN3-89028-109-5 397 Seiten.

## Verschenken Sie Freude !

Mit einem LOOP-Abo für das Jahr 1989 können Sie Ihren Freunden und Bekannten ein Jahr Freude für DM 25.-- bereiten.

(Tel.: 0831 / 6211)

Rüdiger Nahm:

# Tips für alle Programmierer von Utilities unter FLOMONCG

Gegeben sei der folgende Fall:

Es soll ein Programm konstruiert werden, das alle Kleinbuchstaben, die an den Bildschirm ausgegeben werden sollen, in Großbuchstaben umwandelt.

## Bisherige Technik:

Das Programmstück wird mit einem kleinen Ladeprogramm versehen; die Adressen werden für einen Bereich oberhalb des Monitors ausgelegt, in dem hoffentlich (!) noch Platz ist. Ärger gibt es, falls eine zweite Utility auch den gleichen Bereich benutzen möchte.

Beim Laden des Programmstücks wird hoffentlich (!) der im Monitor dafür vorgesehene Zeiger aktualisiert (FREEMEM...), anschließend wird per Hand der richtige Einsprung z.B. 0F00F verbogen.

## In Zukunft:

1. Der Programmierer darf nicht mehr selbst im Monitorbereich Änderungen vornehmen. Damit wird eine gewisse Einheitlichkeit gewährleistet; nur so kann die Bankunabhängigkeit auch nach dem Laden von Erweiterungen erhalten bleiben.

2. Der Programmierer braucht sich nicht um die endgültige Adresse seiner Prozedur zu kümmern; beim Aufruf der entsprechenden Monitorprozedur 'Einsprung umleiten' werden die notwendigen Adressen automatisch korrigiert.

Für seine eigenen Entwicklungen kann sich der Programmierer am folgenden Beispiel orientieren, das dem Handbuch zu FLOMONCG entnommen ist. (Bild 1)

Im Beispiel wurden einige Makros verwendet, um häufig wiederkehrende Sequenzen abzukürzen. Die Makrodefinition steht weiter unten.

Aus dem kurzen Programmstück wird nach dem Assemblieren und Linken eine CP/M Kommandodatei, nach deren Aufruf alle Kleinbuchstaben in Großbuchstaben umgewandelt werden.

Die benutzten Makros befinden sich auch auf der Quellediskette zu FLOMONCG.(Bild 2)

```
INCLUDE RSX.MAC
BEGIN      0f009h ; Adresse des
            neuen Einsprungs
REL_BEGIN
ld      a,c
cp      'a'
REL      <jp   c,next>
cp      'z'+1
REL      <jp   nc,next>
sub     32
ld      c,a
REL      <jp   next>
REL_END
end      start
```

Bild 1: Beispiellisting

```
BEGIN      macro      jump
set_jp     equ 0f06ah      ; Umleitung einsetzen
reset_jp   equ 0f06dh      ; Umleitung entfernen
bdos      equ 00005h
start:
        ld      hl,relbeg
        ld      bc,relend-relbeg
        ld      de,jump
        exx
        ld      hl,reltbl
        ld      bc,reltbl_len
        ld      de,relbeg
        exx
        call   set_jp
        ret    nc
        ld      c,9
        ld      de,msg
        jp      bdos
msg: db   'RSX: nicht mehr genuegend Platz',10,13,$
        endm
REL_BEGIN macro
dseg
reltbl:          ; Anfang der Tabelle mit Korrekturadressen
cseg
relbeg:
next: jp 0000 ; dummy-Eintrag
      dw 0 ; intern verwendet
      endm
REL_END macro
dseg
reltbl_len equ ($-reltbl)/2
cseg
relend:
      endm
REL macro      m1
local x
m1
x equ $-2
dseg
dw x
cseg
      endm
```

Bild 2: Listing der Makros

Günter Renner

# Logisch-physikalisches Verwirrspiel

Im letzten Beitrag ist sicher vieles offengeblieben. Da sind zunächst einmal die Daten enthaltenden Gebilde, die Cluster heißen. Und vor allem ist es die Zählweise der Sektoren, die bei eingehender Betrachtung ganz anders ist als im Listing des Bootsektors. Eigentlich gibt es nur eine Art von Sektoren - es sind die, die auf der Diskette stehen. Trotzdem gibt es Unterschiede, denn wie so oft, wird ein und dasselbe verschieden benannt.

Das verwendete MS-DOS Format hat insgesamt  $40 \times 2 \times 9 = 720$  Sektoren auf der Disk. Es wäre sicher logisch, diese nun ganz einfach fortlaufend durchzuzählern. Das tut man tatsächlich auch. Und weil Computer bei null zu zählen anfangen, gibt es die logischen Sektoren 0 bis 719.

Auf der Diskette selbst sieht das ganz anders aus. Dort hat jeder Sektor ein Kennfeld, das unter anderem Spur, Seite und Sektornummer enthält. Letztere darf nicht mit der logischen Sektornummer verwechselt werden. Vielmehr ist es eine Zahl, die auf jeder Spur und Seite bei eins beginnend erneut hochgezählt wird - in unserem Fall bis neun. Diese Informationen bezeichnet man gemeinhin als physikalische Sektorkenndaten. Will man etwa den Floppytreiber des Grundprogramms aufrufen, so muß man stets genau diese Werte benennen, damit der richtige Sektor gefunden und bearbeitet werden kann. Die logischen Sektornummern müssen dagegen erst in die physikalischen Werte umgerechnet werden.

Diese Aufgabe erledigt in unserem Filehandler das Unterprogramm 'sektor'. Ruft man es auf, so muß in Register d2.w die logische Sektornummer stehen. Dies wird in die physikalische Spur- und Sektornummer umgerechnet; außerdem wird die Seite bestimmt und das entsprechende Bit im Laufwerkscode beeinflusst.

Und weil man nun schon einmal bei der Arbeit ist, wird gleich der Floppytreiber aufgerufen. Je nach dem Inhalt von d1.w wird gelesen oder geschrieben. Quelle oder Ziel steht in a0. Dieser Zeiger ist nach der Ausführung bereits auf die nächste RAM-Stelle positioniert. Auf diese Weise kann man sehr bequem logische Sektoren bearbeiten.

Leider ist es damit noch nicht genug. Denn das Inhaltsverzeichnis und die FAT geben keine Auskunft über die Sektoren, aus der eine gesuchte Datei

besteht. Woher soll man also die logischen Sektornummern nehmen? Es bleiben nur die Clusternummern übrig, die man offenbar

an Hand eines Dateinamens bestimmen kann. Weil das so ist, empfiehlt es sich, noch ein Programm zu haben, mit dem man gleich ganze Cluster bearbeiten kann - es ist das Unterprogramm 'cluster'.

An sich ist die Aufgabe desselben ziemlich einfach. Denn so ein Cluster besteht aus nichts anderem als aus zwei aufeinanderfolgenden Sektoren, die entsprechend ihrer logischen Nummer nacheinander bearbeitet werden müssen. Etwas Rechnerei muß sein, weil die Zählung wieder einmal anders ist. Eigentlich läge es nahe, die Clusternummer einfach zu verdoppeln, um den ersten zugehörigen Sektor zu erhalten. Es gibt aber seitens des Betriebssystems MS-DOS Konventionen der Zählweise, die berücksichtigt werden müssen.

Bei dem verwendeten Format beginnt der nutzbare Datenbereich unmittelbar nach dem Hauptinhaltsverzeichnis. Das entspricht dem logischen Sektor 12 oder \$c. Die Cluster werden nun aber ab zwei durchnummeriert, sodaß sich folgende Zuordnung ergibt:

Cluster	log. S.	Spur	Seite	phys. S.
2	12,13	0	1	4,5
3	14,15	0	1	6,7
4	16,17	0	1	8,9
5	18,19	1	0	1,2

Diese Tabelle läßt sich leicht erweitern. Die Relation zwischen Clustern und Sektoren läßt sich in gewissen Grenzen frei vorgeben; sie muß stets in den BIOS-Parameterblocks spezifiziert sein. Der vorgestellte Handler macht allerdings von diesen Einträgen nur zu Prüfzwecken Gebrauch und ist fest auf die oben genannten Verhältnisse eingestellt.

Trotzdem kann man die BIOS-

Teil1: Der verrückte Bootsektor  
Loop 17

Teil 2: Eine gefährliche Operation  
Loop 19

Teil 3: Log.-physikalisches Verwirrspiel

Teil 4: Sage mir, was Du hast  
Loop 21

Teil 5: Jetzt wird gelesen  
Loop 22

Teil 6: Ärger mit Ä  
Loop 23

Parameter nicht unbeachtet lassen. Denn sie bieten die Gelegenheit, überhaupt einmal zu überprüfen, ob die eingelegte Disk mit dem Handler bearbeitet werden kann. Diese Prüfung kann man mit der im Listing angegebenen Menüfunktion 'testdisk' vornehmen.

Zunächst wird das Unterprogramm 'testdis' aufgerufen. Der logische Sektor 0 wird eingelesen und die BIOS-Parameter mit einer Referenzvorgabe verglichen. Stimmen sie nicht überein, erfolgt eine Meldung. Der Menüfunktion wird der Ausgang der Untersuchung mit \$f in d0 und gesetztem Minusflag rückgemeldet. Im Fall der Übereinstimmung geht der Test weiter. Im Unterprogramm 'testfat' werden die beiden FATs auf Übereinstimmung überprüft. Im negativen Fall gibt es auch dafür eine Fehlermeldung und dieselbe Rückmeldung für die aktuelle Menüfunktion.

Diese Prüffunktionen sind sehr wichtig. Beim Lesen von Dateien könnte allenfalls Unsinn im RAM stehen, wenn man sich an einer falschen Diskette versucht. Ganz und gar unabsehbar sind aber die Folgen bei allen Operationen, die auf die Disk schreiben, wie etwa das Abspeichern oder Löschen von Dateien. Um Sicherheit vor diesem Fehlerfall zu haben, rufen deshalb viele später noch vorgestellten Menüfunktionen erst einmal 'testdisk' auf.

Erweist sich die eingelegte Disk als geeignet, wird von der Menüfunktion 'testdisk' noch das Inhaltsverzeichnis eingelesen und nach einem Kennsatz durchsucht, der - falls vorhanden - angezeigt wird. Dann gehts zurück ins Menue.

Bis jetzt war der Stoff vielleicht etwas trocken. Dafür wird es nächstes Mal interessanter. U. a. wird die Menüfunktion 'Inhaltsverzeichnis anzeigen' vorgestellt. Gewisse Ähnlichkeiten mit MS-DOS werden mithin immer deutlicher...

## Patchwork Listing Teil 3

```

*s33 Patchwork-Listing Teil 3
(C) Guenter Renner

****Menuefunktionen

testdisk:
  bsr crlfkon
  lea fhflag(pc),a4
  clr.b (a4)
  bsr testdis
  bmi testd3

  bsr getdir
  lea dirbuf(pc),a4
  move #112-1,d7

testd0:
  cmp.b #8,11(a4)
  beq testd1
  adda.l #20,a4
  dbra d7,testd0
  bra testd3

testd1:
  move #11-1,d1

testd2:
  move.b (a4)+,d0
  move #!co2,d7
  bsr traprts
  dbra d1,testd2
  bsr crlfkon

testd3:
  bsr testfig
  bra menue

****Unterprogramme

cluster:
  movem.l d2,-(a7)
  subq #2,d2
  lsl #1,d2
  add #3c,d2
  bsr sektor
  addq #1,d2
  bsr sektor
  movem.l (a7)+,d2
  rts

sektor:
  *log. Nr. in d2

  *neue Zeile
  *fehlerfrei
  *annehmen
  *Format/FATS

  *Inhaltsverz.
  *einlesen
  *max. Eintraege

  *kennsatz?

  *falls vorhanden
  *ausgeben

  *11 Zeichen
  *neue Zeile

  *Floppy-Fehler?
  *zurueck

  *zwei Sektoren
  *ist log. Nr.
  *umrechnen:
  *verdoppeln
  *offset dazu

  *alles retten
  *Spur 0 annehmen

  *0 - 8 fuer die
  *Sektoren 1 - 9

  *ie Spur/Seite
  *Spur inkrem.

  *phys. Sektor
  *lwd, Seite 0
  *ist phys. Spur
  *gerade?
  *fuer Seite 1

  *Fehler annehmen
  *fuer max. Spur
  *mehr geht nicht

  *gut gegangen?

  *Fehlerflag
  *updates

  *neue RAM-Adr.
  *Rest zurueck

  *Bootsektor
  *lesen

  *Parameter
  *Referenz
  *Laenge

  *vergleichen

  *dann ists ok

testdi2:
  lea meld4(pc),a0
  bsr meldung
  clr.l d0
  subq #1,d0
  rts

testfat:
  bsr getfat1
  bsr getfat2
  lea fatbuf1(pc),a3
  lea fatbuf2(pc),a4
  move #1024-1,d7

testfal:
  move.b (a3)+,d0
  move.b (a4)+,d1
  cmp.b d0,d1
  bne testfa2
  dbra d7,testfal
  clr.l d0
  rts

testfa2:
  lea meld3(pc),a0
  bsr meldung
  clr.l d0
  subq #1,d0
  rts

****Defaults

biosref:
  dc.b 0,2
  dc.b 2
  dc.b 1,0
  dc.b 2
  dc.b #70,0
  dc.b #d0,2
  dc.b #fd
  dc.b 2,0
  dc.b 9,0
  dc.b 2,0
  dc.b 0,0
  dc.b 0,0
  dc.b #a

  *Referenzvorgabe
  *512 Bytes/Sektor
  *Sektoren/Cluster
  *res. Sektoren
  *Anzahl FATS
  *fuer 112 DIR-Entries
  *fuer 720 Sektoren
  *Kennung
  *Sektoren/FAT
  *Sektoren/Spur
  *zweiseitig
  *versteckte Sektoren
  *Laenge der biospara

'A:\>

```

# Farbige Weihnachten!

Wir bieten Ihnen drei Super-Paketangebote, mit denen Sie Farbe in Ihren PC oder AT bringen:

**11305 EGA-Paket DM 1.348.-**

**11308 VGA-Paket DM 1.899.-**

**11309 Profi-VGA-Paket 2.598.-**

## Das EGA-Paket

Kurze EGA-Karte 640 x 480 autoswitch, Printer Port Anschlußmöglichkeit für BUS-Mouse Treiber & Handbuch

14" EGA-Monitor, Typ 14CH117 Text-Schalter: bernstein/grün  
Frequenzgang: horizontal 15,75 und 21,85 kHz (auto), vertikal 60 Hz dreh- und schwenkbarer Kippfuß max. Auflösung 640x350

## Das VGA-Paket

MIRA VGA-Karte. Cirrus Chip Set, kurze Karte Grafik: 640x480, Text: 720x400

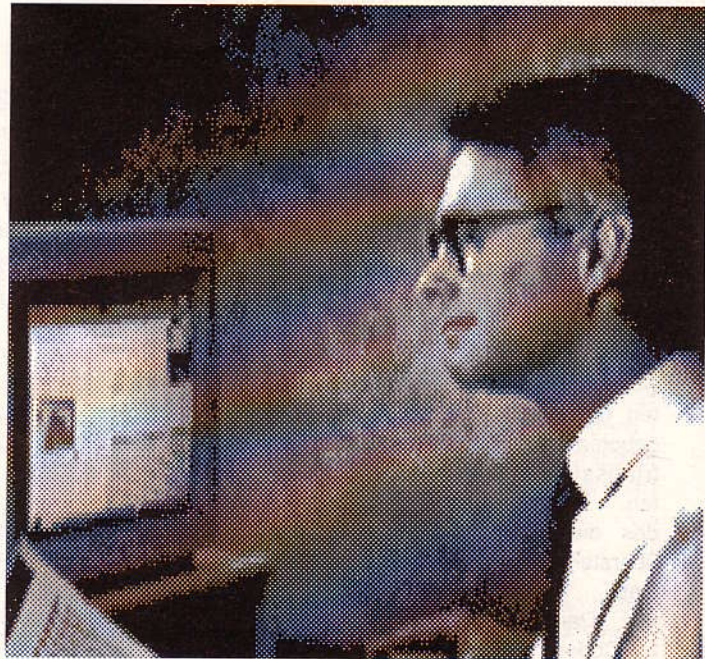
256 kB, 2 ports: ttl und analog Bildschirmunterstützung:

MCGA, VGA, EGA, CGA, MDA, Hercules, Treiber & Handbuch

Farbmonitor: 12" analog/color Frequenzgang: horizontal 31,5 kHz (fix)

vertikal 45-75 Hz Text-Schalter: bernstein/grün/weiß auf blau dreh- und schwenkbarer Kippfuß

Farb-Bildschirm zum Anschluß an VGA-Grafikkarten oder zum direkten Anschluß an PS/2-Systeme. Dieser Monitor verdankt sein besonders klares Bild einer Lochmaske von nur 0,28 mm. Grafiken können in 256 unterschiedlichen Farbstufen dargestellt werden. Sie können mit einem seitlich angebrachten Text-Schalter zwischen den Darstellungen farbig, bernstein, grün und weiß auf blau wählen. Der Bildschirm arbeitet mit einer festen Zeilenfrequenz von 31,5 kHz; er eignet sich sowohl für die Software, die für den neuen VGA- und MCGA-Standard geschrieben wurde, als auch für alle Programme unter MDA, CGA, EGA und Hercules. Der MD-12 erreicht im Grafikmodus eine Auflösung von 640x480, im Textmodus sind 720x400 Punkte darstellbar.



## Das Profi-VGA-Paket

Video7 Vega VGA: Kurze VGA-Karte

max. 800x600 Auflösung in Verbindung mit NEC MultiSync II 256kB, ein ttl port und ein analog port. Bildschirmunterstützung:

MCGA, VGA, EGA, CGA, MDA, Hercules

Unterstützt PS/2 analog Monitor, MultiSysnc, EGA-Monitor, IBM Farb- und IBM Monochrom Bildschirm.

max. Textdarstellung 60 Zeilen, 80 Spalten oder 43 Zeilen, 132 Spalten. Treiber und Handbuch.

11120 NEC-MultiSync II

14" multiscan

Frequenzgang: horizontal 15,5-35 kHz (auto)

vertikal 50-80 Hz

Text-Schalter: bernstein/grün/schwarz auf weiß dreh- und schwenkbarer Kippfuß

Der Klassiker unter den Multi-Funktions-Bildschirmen zum Anschluß an PC, XT und AT mit handelsüblichen Grafikkarten mit hoher Bildwiederholfrequenz oder zum direkten Anschluß an PS/2-Systeme oder an Apple Macintosh II. Der Monitor erreicht eine maximale Auflösung von 800x560. Sie können mit einem an der Frontseite angebrachten Text-Schalter zwischen den Darstellungen bernstein, grün und weiß wählen.

Alle Angebote verstehen sich freibleibend, solange der Vorrat reicht. Alle Preise gelten inclusive der ges. Mehrwertsteuer ab Kempten.

**GRAF ELEKTRONIK SYSTEME GMBH**

**GRAF**<sup>®</sup>  
computer

Klaus Janßen

# JADOS IM AUFWIND

## Vorstellung der neuen Version 3.0

**JADOS, das Standardbetriebssystem für die 68000er-Linie des NDR-Klein-Computers liegt nun in der Version 3.0 vor. Die hervorstechenden Merkmale der neuen Version sind vor allem die RAM-Disk, der komfortable Kommandozeileneditor mit Historybuffer, die Verwaltung des Datums im Directory, der Soft-Schreibschutz für Dateien und die Verbesserung der Befehle "COPY" und "ERA", die nun die Jokerzeichen "\*" und "?" verstehen und auch in einem per Option einstellbaren Interaktivmodus arbeiten.**

### Ramdisk

Nachdem JADOS seit bald 1 1/2 Jahren unverändert in der Version 2.1 angeboten wird, wurde es nötig, das Betriebssystem kräftig zu überarbeiten. Da der Umfang des Programmcodes die Kapazität der Systemspuren übersteigt, liegt JADOS als normale Datei mit dem Namen "JADOS.SYS" vor. Der Umfang liegt jetzt bei 22 KByte Code, was einer Steigerung von etwa 50% entspricht. Auch der Variablenbereich mußte vergrößert werden, so daß das komplette JADOS-System inklusive Systemstack nun genau 32 KByte umfaßt.

Auf den Systemspuren befindet sich jetzt ein erweiterter Urlader, der ein Mini-DOS enthält, das in der Lage ist, JADOS-Dateien zu laden. Der Urlader bietet die Möglichkeit, eine RAM-Disk einzurichten, falls sich keine im Speicher befindet, d.h. beim ersten Bootvorgang. Dabei kann die Größe der RAM-Disk in weiten Grenzen frei gewählt werden. Bei den eventuell nach einem RESET folgenden Bootvorgängen bleibt die RAM-Disk unzerstört.

Jeder Sektor der Disk wird mit einer Prüfsumme abgesichert, so daß die Daten konsistent bleiben. Die RAM-Disk wird im JADOS dann als Laufwerk 0: angesprochen. Mindestvoraussetzung sind allerdings wenigstens 128 KByte Speicher.

### Uhrenbaugruppen

Es werden nun auch diverse Uhrenbaugruppen unterstützt. Beim Start des JADOS prüft dieses zunächst, ob die Uhrenkarte mit dem IC E050 im System ist. Wenn nicht, dann sucht JADOS den Speicher nach der Smartwatch ab. Diese kann also unter einem beliebigen statischen 8 KByte RAM liegen. Ist auch keine Smartwatch vorhanden, so muß der Benutzer das

Datum von Hand eingeben. Dieser Wert bleibt bis zum Ausschalten des Rechners erhalten und wird bei folgenden Bootvorgängen nicht noch einmal abgefragt.

Da das Datum nun im Inhaltsverzeichnis der Dateien gespeichert wird, lohnt es sich, den Rechner mit einer der angegebenen Uhren nachzurüsten. Dabei wird jeweils der letzte Schreibzugriff auf eine Datei gewertet.

### Soft-Schreibschutz

Bisher konnten nur komplette Disketten durch Anbringen der Schreibschutzmarke gegen Löschen und Überschreiben geschützt werden. Ab Version 3.0 können nun auch einzelne Dateien mit einem Softwareschutz versehen werden. So geschützte Dateien können weder von der Kommandoebene noch durch die JADOS-Traps verändert werden. Lediglich mit Routinen, die unterhalb der Betriebssystemebene arbeiten, wie z.B. FLOPPY, können die Dateien verändert werden. Der Schreibschutz wird durch den neuen Befehl "CHMOD" (= change mode) gesetzt oder aufgehoben. Die Befehlsyntax lautet:

```
chmod r:w filespec [i]
```

Mit dem Parameter "r" wird der Schutz gesetzt ( r = readonly ) und mit "w" aufgehoben ( w = write enable ). Der Parameter "filespec" enthält den Dateinamen, bzw. ein Suchmuster für mehrere Dateien unter Verwendung der Jokerzeichen "\*" und "?"; ähnlich dem Befehl "DIR". Alle auf das Muster passenden Dateien werden angezeigt und erhalten den gewünschten Schutzstatus.

Der dritte Parameter ist optional. Wird er eingegeben ( i = interactive ), so wird jede auf das Muster passende Datei zunächst nur angezeigt. Der

Benutzer kann nun wählen, ob der Schreibschutzstatus für die angezeigte Datei übernommen werden soll oder nicht.

### Kopieren und Löschen

Auch die Befehle "COPY" und "ERA" bieten nun erweiterte Möglichkeiten. Die Syntax zu "COPY" lautet:

```
copy sourcefilespec [destination] [i]
```

Der erste Parameter gibt die Quelldatei an, die kopiert werden kann. Mit den Jokerzeichen "\*" und "?" können mehrere Dateien selektiert werden. Die Zielangabe ist optional. Wird sie weggelassen, so wird das aktuelle Laufwerk als Ziel angenommen. Ansonsten wird als Ziel eine Laufwerksnummer gefolgt von einem Doppelpunkt erwartet. Wenn der erste Parameter genau eine Datei angibt, so kann als Ziel auch ein einzelner Dateiname gewählt werden. Jede selektierte Datei wird angezeigt und auf das Ziel kopiert. Der dritte Parameter ist optional und schaltet den bereits oben erklärten Interaktivmodus ein. In Bild 1 ist eine solche Interaktivsitzung zu sehen.

```
1>copy *.68k 0: i
1:FORMAT.68K (J/N)? kopiert
1:BANNER.68K (J/N)? kopiert
1:ROMSTART.68K (J/N)? NEIN
1:1COPY.68K (J/N)? NEIN
1:SYS.68K (J/N)? NEIN
1:VERS.68K (J/N)? NEIN
1:DISKCOPY.68K (J/N)? kopiert
1:MORE.68K (J/N)? kopiert
1:BRDRUCK.68K (J/N)? kopiert

1>
```

**Bild 1: COPY-Befehl im Interaktivmodus**

Die Syntax zu "ERA" lautet jetzt:

```
era filespec [exec]
```

Der erste Parameter selektiert die zu löschenden Dateien. Da der Löschbefehl zur Kategorie der gefährlichen Befehle zu zählen ist, arbeitet er grundsätzlich im Interaktivmodus. Jede selektierte Datei wird also zuerst angezeigt.

Der Benutzer muß dann wählen, ob er die Datei löschen will oder nicht. Nur wenn als zweiter Parameter "exec" (= execute) eingegeben wird, dann unterbleiben die Sicherheitsabfragen.

## Directory

Da im Directory der Disketten nun das Datum und der Schreibschutzstatus verzeichnet sind, der "DIR"-Befehl aber nicht verändert werden durfte, stehen jetzt neue Befehle zur Directoryanzeige zur Verfügung. Wenn nur ein schneller Überblick gewünscht wird, dann ist der neue Befehl "SDIR" genau richtig (= short directory). Er zeigt in fünf Spalten nur die Dateinamen an. Die Befehlssyntax lautet:

```
sdir filespec [l:p]
```

Mit dem optionalen zweiten Parameter wird das Directory auf den Drucker umgelenkt. Bild 2 zeigt ein Beispiel für das Ergebnis des Befehls.

```
1>sdir *.*
FORMAT.ASM  MORE.ASM      GP.ASM      DSAVE.ASM  1COPY.ASM
DISKCOPY.ASM ROMSTART.ASM SYS.ASM     BRDRUCK.ASM BANNER.ASM
FORMAT.68K   BANNER.68K  ROMSTART.68K DSAVE.COM  1COPY.68K
SYS.68K      GP.COM      VERS.ASM    VERS.68K   DISKCOPY.68K
MORE.68K     JADOS.SYS  BRDRUCK.68K UPDATE.DOC

1>
```

Bild 2: Inhaltsverzeichnis in Kurzform

Wer genauere Angaben zu den Dateien wünscht, muß den neuen Befehl "LDIR" benutzen (= long directory). Außer dem Namen, wird die Länge in KByte, das Datum und der Schreibschutzstatus ausgegeben. Die Ausgabe in zweispaltig. Bild 3 zeigt ein Beispiel.

```
1>ldir *.*
FORMAT.ASM  15  13.09.88  RW  MORE.ASM  13  17.09.88  RW
GP.ASM      11  13.09.88  RW  DSAVE.ASM  4  11.09.88  RW
1COPY.ASM   6  11.09.88  RW  DISKCOPY.ASM 26 16.09.88  RW
ROMSTART.ASM 12 11.09.88  RW  SYS.ASM    9  11.09.88  RW
BRDRUCK.ASM 60 18.09.88  RW  BANNER.ASM 14 13.09.88  RW
FORMAT.68K  3  13.09.88  RW  BANNER.68K 2  13.09.88  RW
ROMSTART.68K 2  13.09.88  RW  DSAVE.COM  1  13.09.88  RW
1COPY.68K   1  13.09.88  RW  SYS.68K    2  16.09.88  RW
GP.COM      2  13.09.88  RW  VERS.ASM   3  16.09.88  RW
VERS.68K    2  16.09.88  RW  DISKCOPY.68K 3 16.09.88  RW
MORE.68K    2  17.09.88  RW  JADOS.SYS 22 17.09.88  R-
BRDRUCK.68K 6  18.09.88  RW  UPDATE.DOC 40 18.09.88  RW

1>
```

Bild 3: Inhaltsverzeichnis in ausführlicher Form

## Neue und verbesserte Befehle

Es gibt noch eine Reihe weiterer neuer Befehle, deren genauere Erläuterung aber den Rahmen des Artikels sprengen würde. Der Vollständigkeit halber seien sie kurz aufgelistet:

- **COLD** Kaltstart oder Boot von Disk
- **DATE** Datum setzen oder anzeigen
- **DI** (= disk info) Aktueller Belegungsstatus eines Laufwerks
- **DPATH** (= drive path) Definiert einen Suchpfad für Laufwerke. Eingetippte Kommandos werden auf allen im Suchpfad eingetragenen Laufwerken gesucht.
- **EXIT** Rückkehr in das Grundprogramm

## Kommandoeditor mit Historybuffer

Besonderes Augenmerk wurde in der neuen Version auf den Eingabekomfort gerichtet. Die Kommandos werden jetzt mit einem richtigen Zeileneditor eingegeben. Dieser erlaubt Cursorsteuerung wie Zeichen links, Zeichen rechts, Wort links und Wort rechts. Es können einzelne Zeichen gelöscht werden oder der Rest der Zeile hinter dem Cursor. Man kann im Einfügemodus oder im Überschreibmodus arbeiten. Die Eingabe kann abgeschlossen oder abgebrochen werden. Alle abgeschlossenen Eingaben werden in einen Historybuffer übernommen. Dieser speichert die letzten zwanzig Eingaben. Mit den Cursorstasten für AUF und AB kann man durch den Historybuffer "wandern" und somit blitzschnell alte Eingaben wieder hervorholen. Dies erspart enorm viel Tipperei.

## Hilfsprogramme

Auch die externen Kommandos haben Verbesserungen und Erweiterungen erfahren.

- **BANNER** Erzeugung von Großschrift
- **DISKCOPY** Duplizierprogramm für Disketten. Ab Version 3.0 wird ein Verify angeboten.
- **FORMAT** Formatierer für Disketten. Ab Version 3.0 wartet das Programm nach der Formatierung auf einen Tastendruck
- **MORE** Schnelle Dateianzeige. Ab Version 3.0 können nun auch Strings gesucht werden.
- **VERS** Ermittelt die Version des JADOS auf Diskette.

Für Programmierer werden weit über zwanzig neue Traps angeboten, darunter ein sehr komfortabler zur Auswertung der Directories und einige zur Behandlung von Datum und Uhrzeit.

Allen Anwendern des JADOS wünsche ich mit der neuen Version ein angenehmes und erfolgreiches "Arbeiten".

## Geprüfte Komplettsysteme :

**GRAF**  
computer

10997	ACRT Komplettsystem	3.000.-
10871	LOG16 Kompletcomputer	2.200.-
11088*	68008 Fertigsystem SN 4F050	2.500.-
11089*	68008 Fertigsystem SN 4F055	1.900.-

\* Näheres auf Anfrage

Sonderpreisliste - gültig bis zum 31.12.88

GRAF ELEKTRONIK SYSTEME, Magnusstraße 13, 8960 Kempten, Tel.: 0831 / 6211

Klaus Rumrich

# RIP the CHECKER

alias SCHACH 3.4

Die neue Version 3.4 des Schachprogrammes für alle 680XX CPUs unter JADOS ist nun unter dem Namen RIP the CHECKER verfügbar. Große Teile des Programmes wurden völlig überarbeitet und neue Routinen eingebaut.

## COL und Maus

Die auffälligste Neuerung besteht in der Unterstützung der Farbgrafikbaugruppe COL 256 und der Maus. Die Adresse des COL-Bildspeichers ist dabei frei wählbar. Selbstverständlich können die verwendeten Farben beliebig eingestellt werden. Der "Denkprozess" des Programmes läßt sich nun auf Wunsch verfolgen. Dabei wird momentane Rechentiefe, der bis jetzt gefundene beste Zug und der Bruchteil der bereits untersuchten Züge angezeigt.

Auch ohne Hardcopy-Baugruppe oder neue auslesbare GDP ist jetzt ein Diagrammausdruck möglich. Die Schnittstelle zum Drucker ist dabei in das Overlay-File PLOT.OVL gelegt worden. Dies ermöglicht die Anpassung an jeden grafikfähigen Drucker (Der Quelltext PLOT.ASM für EPSON-Kompatible Drucker mit 9 Nadeln wird auf der Diskette mitgeliefert).

## Verbesserte Zugsberechnung

Viel Arbeit wurde in die Verbesserung der Zugsberechnung investiert. Bei Spielstufe 3.4 ergibt sich gegenüber der alten Version 2.9 eine etwa fünffache Geschwindigkeitssteigerung. Das wurde vor allem durch eine nochmals verbesserte Ausnutzung des sogenannten Alpha-Beta-Prinzips erreicht. Bekanntlich hängt der Nutzen von Alpha-Beta empfindlich von der Reihenfolge der Zugversuche ab (siehe dazu auch den Artikel in LOOP 15/3). Deshalb werden die vom Zuggenerator erzeugten Züge nicht mehr in der erzeugten Reihenfolge durchprobiert, sondern zunächst in einer Liste zusammen mit einer Plausibilität für jeden Zug abgelegt. Anschließend werden die Züge in der Liste in der Reihenfolge ihrer Plausibilität probeweise gezogen. Bei der Bestimmung der Plausibilität werden insbesondere Züge, die eine im vorherigen Halbzug bewegte Figur geschlagen und die sogenannten "Killerzüge" besonders hoch bewertet. Da-

durch wird die Chance, daß Alpha-Beta bereits bei einem der ersten Gegenzüge greift, beträchtlich erhöht.

Ein Gegenzug, der einmal zu einem Alpha-Beta-Abschneiden geführt hat, wird mit großer Wahrscheinlichkeit auch nach einem anderen Zug zum Abschneiden führen. Einen solchen Gegenzug merkt sich deshalb das Programm als "Killerzug".

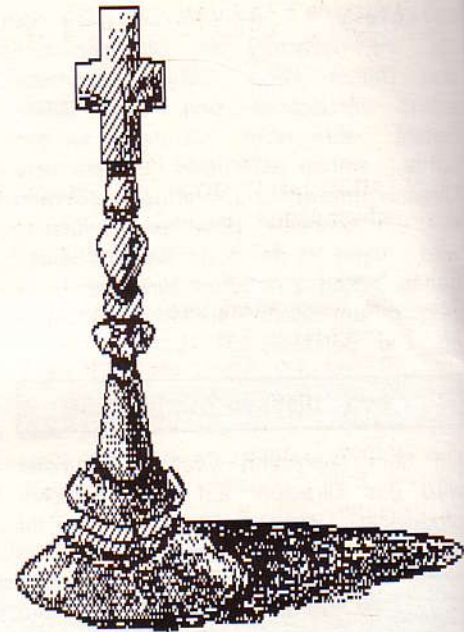
Findet der Zuggenerator, daß der Killerzug auch nach einem anderen Zug ein zulässiger Zug ist, so erhält er maximale Plausibilität, um von der nachgeschalteten Zugausführung als erstes verwendet zu werden.

Eine andere Neuerung erhöht in bestimmten Stellungen die Spielstärke, ohne die Rechenzeit übermäßig zu erhöhen.

Dazu muß man wissen, daß mit der Angabe der Spielstufe die Rechentiefe für Normalzüge und anschließende Schlagzüge festliegt. Bei RIP the CHECKER kann nun optional über mehrere Parameter eingestellt werden, in welchen Situationen die Rechentiefe eines einzelnen Teilbaumes erhöht werden soll. Auslöser für eine Erhöhung können Schlagzüge, Schachgebote oder Schlagzüge mit Schachgebot während der Normalzugphase sein. Dies bringt ein klein wenig Selektivität in das Programm.

## Automatische Spielstufeneinstellung

Eine Schwäche der früheren Programmversionen war die sehr unterschiedliche Rechenzeit für verschiedene Stellungen bei gleicher Spielstufe. Bei Turnierspielen, die das Programm bestritt, mußte daher die Spielstufe gelegentlich manuell an die vorhandene Bedenkzeit angepaßt werden. Ab jetzt kann RIP the CHECKER seine Bedenkzeit selbst verwalten (dies ist allerdings nur bei Verwendung einer Uhrenkarte oder der Interrupt-gesteuerten Uhr möglich). Dabei wird keine feste Bedenkzeit pro Zug vorgege-



ben, sondern die Bedenkzeit für eine ganze Anzahl von Zügen oder gar für die gesamte Partie, genau wie es bei Turnierpartien üblich ist.

Diese "variable Spielstufe" wird als 'V.m' eingestellt, wobei 'V' auf variabel hindeutet; die Ziffer 'm' gibt nach wie vor an, wie weit Schlagzüge weiterverfolgt werden (Für Schachpartien wird mindestens m=6 empfohlen).

## Interrupt-Uhr

Zur Verwendung der variablen Spielstufen wird unbedingt eine Uhr benötigt. In der alten Version 2.9 wurden nur die Uhrenkarte und die Smartwatch über eigene Routinen verwendet; die Grundprogrammrountinen waren zu langsam. Dies verhinderte allerdings den Einsatz anderer Uhrenbausteine in Verbindung mit einer Änderung des GETUHR-Unterprogrammes.

Um möglichst unabhängig vom Einsatz einer bestimmten Uhr zu werden, wird ab dieser Version auch die Grundprogrammuhre unterstützt, sofern sie nicht zu langsam arbeitet (Auslesen muß schneller als 20 msec sein). Zusätzlich ist die Möglichkeit gegeben, eine interruptgesteuerte Softwareuhr zu verwenden. Dazu muß auf der GDP der IRQ-Anschluß des Grafikprozessors mit der INT-Leitung auf dem Bus verbunden werden bzw. auf der GDP 64 HS die Änderung für Rev. 10 durchgeführt und die Steckbrücke JMP 5 gesetzt werden. Dadurch wird alle 20 Millisekunden ein Interrupt erzeugt, der die Uhr hochzählen läßt.

Klaus Rumrich



# Die neue Version von DATEI II ist da!

DATEI II gibt es als JA-DOS-, JOGI-DOS- oder SYSTEM-P.D.-DOS-Version und kostet ab Version 2.0 DM 136,00.

**Das Datenbank- und Textverarbeitungsprogramm wurde erweitert. Jetzt sind Modem- und Rechenfunktionen implementiert.**

## Modemfunktionen

Der Modemteil nutzt die Fähigkeiten der SER-Karte. Damit ist es möglich, einen Akustik-Koppler einzusetzen und z.B. mit dem GES-Datentelefon zu kommunizieren. Die Daten werden in DATEI II-Textblocks abgelegt. Komplette Textblocks können gesendet werden.

DATEI II-Daten können natürlich auch zwischen Computern ausgetauscht werden.

Wahlweise können nun Drucker mit paralleler (Centronics) oder serieller Schnittstelle angesteuert werden.

Da schon bisher beliebige Druckertreiber auf einfache Weise mit DATEI II erstellt werden konnten, gibt es durch

die Schnittstellen-Umschaltung zukünftig keine Probleme mit der Druckeranpassung.

## Rechenfunktionen

DATEI II beherrscht die vier Grundrechenarten. Rechengvorgänge werden mittels Anweisungen ausgeführt, mit denen auch das Ausgabeformat (Kommastellen, Rundung, 1000er Punkte) vorgegeben wird. Die Anweisungen können an beliebigen Stellen im Block stehen. (Beispiel Bild 1)

Die gesamte Berechnung nach der Formel in Zeile 6 wird mit den Anweisungen in Zeilen 9 und 10 auf "Knopfdruck" erledigt. Werden z.B. der Betrag in Zeile 12 oder die Konstanten verändert, so erscheinen nach dem Rechenlauf sofort die neuen Ergebnisse in den Zeilen 13 und 14.

Im Datenbank-Modus können sowohl Zahlen innerhalb der Datensätze als auch beliebige Ergebnisse aus allen ausgewählten Datensätzen in einem Durchgang errechnet und weiterverarbeitet werden. Anwendungen hierfür sind z.B. die Erstellung von Umlage-Abrechnungen aufgrund einer Mieter-Datei oder die Ermittlung des Gesamt- und Zeitwertes des Lagerbestandes aufgrund einer Artikel-Datei oder Mahnungen an säumige Beitragszahler aufgrund einer Mitglieder-Datei mit Außenstands-Ermittlung.

Natürlich wurde bei DATEI II weiteres "Sichtbares" und "Unsichtbares" geändert und ergänzt. Besonders dank der Hinweise und Anregungen aus dem Benutzerkreis sind Bugs entfernt und Funktionen ergänzt worden.

```

1   EINK2
2   Errechnung der Einkommensteuer nach Paragraph 32 EkStG
3   für zu versteuernde Einkommen ab 18036 DM bis 80027 DM
4   bei Ledigen (Jahr 1987).
5
6   Formel: (((2,10y - 56,02)y + 600)y + 2200)y + 2962
7   y = (zu versteuerndes Einkommen - 18000) / 10000
8
9   @#x/112/k2/s0/+12/-16/:17/=15/
10  @#x/+18/*15/-19/*15/+20/*15/+21/*15/+22/=13/*23/:12/=14/
11
12  Zu versteuerndes Einkommen: 48.000,00      Eingabe
13  Einkommensteuer:           13.619,56      Ergebnis
14  Einkommensteuer in Prozent: 28,37         Ergebnis
15  Faktor y:                   3,00
16  1. Konstante:               18000,00
17  2. Konstante (Teiler)       10000,00
18  3. Konstante:               2,10
19  4. Konstante:               56,02
20  5. Konstante:               600,00
21  6. Konstante:               2200,00
22  7. Konstante:               2962,00
23  Prozent:                    100,00
24

```

**Bild 1: Ausschnitt aus einem Text-Block zur Errechnung der Einkommenssteuer**

Neu lieferbar:

## Buchhaltung für den 68K auf Diskette

BUCHHALTUNG II baut auf dem bewährten Finanzbuchhaltungsprogramm EFIBU auf. Es ist auf Diskette im JOGI-DOS oder JADOS-Format erhältlich und kann auf allen 68K-CPU's eingesetzt werden.

Ein 39-seitiges Handbuch wird mitgeliefert.

Grundkenntnisse der Buchführung reichen völlig für die Arbeit mit BUCHHALTUNG II aus. Die Buchungssätze werden - wie man es von der manuellen Buchhaltung her kennt - in Journal-

form eingegeben.

Eine Journalzeile besteht aus Datum (Tag/Monat), Soll- und Habenkonto, Buchungstext und Betrag (bis zu 7 Stellen vor dem Komma).

Der Kontenrahmen kann nach eigenen Bedürfnissen festgelegt und jederzeit erweitert werden. Bis zu 9998 Konten können angelegt werden.

Den Abschluß erledigt das Programm jederzeit auf Knopfdruck. Die fertigungsbuchten Konten können angesehen oder ausgedruckt werden.

Eröffnungsbuchungen für eine neue Buchungsperiode werden auf Knopf-

druck automatisch durchgeführt.

Journale aus mehreren gleichzeitig geführten Buchhaltungen oder aus aufeinanderfolgenden Buchungsperioden können kombiniert werden.

Journalzeilen und Konten können gelöscht und korrigiert werden. Journalzeilen können per Sortierlauf auf die zeitlich richtige Position (Buchungsdatum) eingeordnet werden.

Auch Journal und Kontenliste können jederzeit ausgedruckt werden. Zur Anpassung des Druckers und zur Steuerung des Ausgabeformates können beliebige Initialisierungsbefehle vorgegeben werden.

Uwe Koch

# RESIDENTE TREIBER unter JADOS

Möglichkeiten für den Einsatz zeigen die bereits existierenden Treiber wie z.B. FASTFLO, HARDCOPY oder BILD\_AUS. FASTFLO von R. Bäcker ersetzt die Floppy-Routine des EGRUND durch eine eigene mit schnellerer Steprate. HARDCOPY hängt sich in den CI-Aufruf und ermöglicht eine formatgetreue Grafik-Hardcopy.

BILD\_AUS hängt sich ebenfalls in die CI-Routine und schaltet den Bildschirm dunkel, wenn die CI-Routine länger als etwa 5 Minuten auf eine Eingabe wartet.

Wie müssen aber nun solche nachträglich eingebundenen Programme aussehen, damit sie einerseits korrekt funktionieren, sich aber andererseits auch nicht gegenseitig beeinflussen? Am Beispiel eines Mini-Treibers, der lediglich die LO-Routine zur SO-Routine umleitet und damit alle Druckausgaben auf die serielle Schnittstelle ausgibt, möchte ich den Aufbau eines solchen Treibers vorstellen.

Treiber-Programme sollen ständig im Speicher bleiben und sich in Betriebssystem-Aufrufe einhängen, um diese durch eigene Routinen zu ersetzen. Dazu gibt es in JADOS seit Version 2.1 die Möglichkeit, Programme resident zu installieren. Mit dem Befehl:

## INST filename

werden die Programme in den Arbeitsspeicher geladen und dort am oberen Ende des freien Bereiches abgelegt. Werden diese Programme mit ihrem Namen aufgerufen, so gibt es zwei Möglichkeiten. Programme mit der Endung **.68K** werden zunächst auf die Programm-Startadresse \$400 (bei Systemen mit BOOTKARTE) kopiert und dort gestartet.

Programme mit der Endung **.COM** werden direkt an der Stelle gestartet, an der sie abgespeichert wurden. Diese Möglichkeit ist ideal für Treiberprogramme, denn man kann den Treiber mit der Endung **.COM resident** (dauerhaft) in den oberen Bereich des Speichers laden und ihn dort aktivieren. Voraussetzung für einen solchen Einsatz ist allerdings, daß das Programm hundertprozentig relokatable geschrieben ist. Wie dies geschieht, hat Folkert Hallenga in LOOP 19 ausführlich be-

**EGRUND und JADOS enthalten fast 200 Unterprogramme und Funktionen, die man in eigenen Programmen verwenden kann.**

**Manchmal wäre es aber angenehm, wenn man einzelne dieser Funktionen erweitern oder ändern könnte, ohne jedesmal neue EPROMs brennen zu müssen, bzw. JADOS zu patchen. Hierfür bietet sich die Möglichkeit an, residente Treiber unter JADOS zu installieren.**

schrieben, weshalb ich hier nicht nochmals darauf eingehen möchte.

Ist ein Treiber im Speicher installiert, so muß er aktiviert werden. Dafür gibt es nur die Möglichkeit des normalen Programm-Aufrufes. Der Treiber muß sich dann selbst in die Betriebssystem-Aufrufe einhängen und initialisieren. Danach soll die Initialisierungsroutine des Treibers wieder ins Betriebssystem zurückspringen. Das Einhängen in Betriebssystem-Aufrufe funktioniert am einfachsten und sichersten, wenn der Treiber den jeweiligen Trap (#1 für EGRUND oder #6 für JADOS) auf sich selbst umleitet, dort die zu ändernde Funktion herausucht und den Trap zu seiner ursprünglichen Adresse weiterleitet. Dies ist natürlich nur möglich, wenn die Trap-Vektoren im RAM-Bereich liegen, also eine BANKBOOT-Karte eingesetzt wird und EGRUND nicht auf \$00000000 liegt. Der Treiber sollte so konstruiert sein, daß er den Trap nach der Bearbeitung nicht fest auf die Einsprungadresse von EGRUND oder JADOS umleitet, son-

dern an die Adresse, die vor der Initialisierung in dem jeweiligen Trap-Vektor stand.

Damit ist sichergestellt, daß man mehrere solcher Treiberprogramme hintereinander schalten kann, ohne daß sie sich gegenseitig stören. Das erfordert aber große Vorsicht bei der Installationsroutine, denn der Treiber muß erkennen können, ob er schon initialisiert ist. Wird er nämlich aufgerufen obwohl er schon initialisiert war, so würde die Trap-Umleitung beim zweiten Mal wieder auf den Treiber selbst umgeleitet und somit kurzgeschlossen, was zum sicheren Aufhängen des Rech-

ners führt.

Außerdem sollte der Treiber eine Möglichkeit bieten, mit der er sich selbst abschalten läßt. Dies ist sehr wichtig, denn bei einem JADOS-Kaltstart oder beim Entfernen des Treibers mit REMOV aus der Tabelle der residenten Programme ist das Programm dem JADOS zwar nicht mehr bekannt, aber solange dieser Speicherbereich nicht durch ein anderes Programm belegt wird, ist der Treiber immer noch aktiv. Wird der Bereich durch andere Programme oder Daten belegt, so führt dies ebenfalls zum sicheren Absturz des Rechners, da der Trap-Vektor nun in unpassende Bereiche zeigt. Die Trap-Vektoren bleiben ja vom REMOV unberührt. Da im Initialisierungsteil des Treibers schon erkannt wird, ob er bereits installiert wurde, kann man hier auch noch eine zusätzliche Routine unterbringen, die bei Aufruf mit dem Parameter AUS alle Vektor-Veränderungen rückgängig macht und damit den Treiber deaktiviert. Dabei ist wieder zu beachten, daß ein bereits inaktiver Treiber nicht nochmals abgeschaltet werden darf, da sonst, ähnlich wie beim zweifachen Installieren, die Vektoren nicht mehr stimmen und der Rechner abstürzt. Alle bisher aufgeführten Forderungen an einen residenten Treiber unter JADOS ab Version 2.1 sind in dem folgenden Beispiel realisiert. Es handelt

1>EDIT SERPRINT.ASM	Eingeben des Textes
1>ASS SERPRINT.ASM	Assemblieren
1>REN SERPRINT.68K SERPRINT.COM	Umbenennen in COM

Bild 1: Treiber SERPRINT erstellen

sich um einen kleinen Treiber, der die LO-Routine (Druckerausgabe auf CENT) zur seriellen Schnittstelle SER umleitet. Natürlich funktioniert dies nur bei Programmen, die ihre Druckausgabe über den EGRUND-Aufruf LO leiten (DRUCKE.68K). Ich mußte leider feststellen, daß weder JADOS (PRINT), noch RL-BASIC (LPRINT, LLIST) sich an diese Konvention halten. Daher funktioniert die Umleitung bei diesen Programmen leider nicht. Der Treiber soll aber hauptsächlich die Funktionswei-

se eines residenten Hintergrundprogrammes verdeutlichen.

Der Treiber besteht aus mehreren Einzelteilen. Die Zeilen 1 bis 13 enthalten den Kommentar, in 14 bis 22 folgt die Definition der Konstanten. In Zeile 26 beginnt das Initialisierungsprogramm. Zuerst wird ein eventueller Parameter geholt, in Großbuchstaben gewandelt und dann die Länge bestimmt. Existiert kein Parameter, so wird zur Routine STARTEIN verzweigt. Andernfalls wird geprüft, ob der Parameter AUS heißt, dann wird zu STARTAUS ge-

gehört. Hier (Zeile 64 bis 68) wird die serielle Schnittstelle initialisiert. Im Beispiel werden die Werte 9600 Baud, 8 Datenbit, keine Parität eingestellt. Hier lassen sich, wie im EGRUND-Handbuch beschrieben, auch alle anderen Konfigurationen einstellen.

Zum Schluß wird noch eine kurze Meldung des Programmes (Zeile 88 bis 107) ausgegeben und dann ins JADOS zurückgesprungen (Zeile 83, 84). Die Routine STARTAUS vergleicht den Trap#1-Vektor mit der Adresse der Trap-Routine in SERPRINT (Zeile 52 bis 55). Sind diese ungleich, so ist

rufen.

In SO könnten natürlich noch weitere Funktionen implementiert werden. So z.B. eine Zeichensatz-Umdefinition für nicht Epson-kompatible serielle Drucker (z.B. Typenrad-Schreibmaschinen). Hier ist eben nur die Minimalfunktion des Treibers vorgestellt worden.

Die Erstellung von SERPRINT ist in Bild 1 dargestellt.

Die Installation und der Aufruf von SERPRINT können entweder von Hand oder in der AUTOEXEC.BAT geschehen (Bild 2).

```

1>INST SERPRINT.COM      Residente Installation im
                          Speicher
1>SERPRINT               Aufruf mit Initialisierung
1>...                    Weitere Arbeiten

1>SERPRINT AUS          Abschalten der Umleitung

Eventuell bei JADOS 3.0 :

1>REMOV SEPRINT.COM     Entfernen aus dem Speicher
    
```

**Bild 2: Installieren von SERPRINT**

sprungen.

In STARTEIN wird zunächst geprüft, ob der Dummy-Eintrag bei SPRUNG noch existiert. Wenn ja, dann ist SERPRINT noch nicht initialisiert und die Trap-Vektoren müssen noch umgeleitet werden, was dann in Zeile 45 bis 49 geschieht. Dazu wird die Adresse von TRAP in die Vektor-Tabelle an Adresse \$00000084 für den Trap#1 eingetragen und der Wert, der vorher hier stand wird in den Adressteil des JMP-Befehls bei SPRUNG eingetragen. Dieser selbstmodifizierende Teil des Programmes stellt sicher, daß mehrere residente Treiber hintereinandergeschaltet werden können.

War SERPRINT schon initialisiert, so wird über MELDINST eine entsprechende Meldung ausgegeben und das Programm beendet. Nach der Initialisierung der Trap-Vektoren geht es bei INIT weiter, einem Programm-Teil, der schon zum eigentlichen Treiber

steht, wieder in den Trap#1 -Vektor eingetragen und ein Dummy wird bei SPRUNG eingesetzt. Danach wird eine Meldung (Zeile 109 bis 115) ausgegeben und das Programm beendet.

Der eigentliche Treiber befindet sich in den Zeilen ab 133. Dort steht zunächst der neue Trap#1-Handler TRAP. Hier wird kontrolliert, ob in D7 die Funktion LO angefordert wird. Wenn ja, so wird in die eigene SO-Routine verzweigt. Andernfalls wird geprüft, ob LSTS angefordert wird. Auch hier wird dann in die eigene Routine SOSTS gesprungen. Waren beide Abfragen negativ, so wird in SPRUNG auf die, vorher in STARTEIN eingetragene Adresse des ursprünglichen Trap-Handlers gesprungen.

Waren die Abfragen positiv, so wird in SO bzw. SOSTS einfach die Anforderung in D7 für die serielle Schnittstelle geändert und danach ebenfalls über SPRUNG der alte Trap-Handler aufge-

SERPRINT  
noch nicht aktiv, was entsprechend gemeldet wird (Zeile 124 bis 129). Ist SERPRINT aber schon aktiv, so wird der ursprüngliche Vektor, der jetzt unter SPRUNG

Zum Schluß möchte ich noch eine wichtige Bitte an alle NKC-Programmierer weitergeben. Bitte schreibt Eure Programme so, daß sie möglichst alle Hardware-Zugriffe über EGRUND oder JADOS (CP/M-68K) vornehmen.

Es ist zwar verständlich, daß einige Programme (PET, EDITEX) eigene CO-Routinen haben, um die GDP-Ausgabe zu beschleunigen, aber bei allen anderen Routinen des EGRUND lassen sich höchstens minimale Geschwindigkeitssteigerungen erzielen. Der große Nachteil bei eigenen Hardware-zugriffen ist aber, daß solche Treiber wie SERPRINT, HARDCOPY, BILD\_AUS usw. nicht mehr wirken. Außerdem gibt es bei Hardware-Änderungen oder Erweiterungen sicher bald ein neues EGRUND, aber wer kann und will schon alle Programme ändern. Wir sollten mit dem NDR-Klein-Computer nicht in so chaotische Verhältnisse abrutschen, wie beim IBM-PC, wo ja kaum ein Programm die MS-DOS-Routinen verwendet. Dort müssen ja alle Nachbauten 100%-ig identisch sein, damit alle Programme laufen. Der große Vorteil des NKC und von JADOS ist ja, daß hier unterschiedliche (Eigenbau-)Hardware mit den gleichen Programmen läuft. Also nochmals :

**Bitte verwendet ausschließlich EGRUND- oder JADOS-Traps für Eure Hardware-Zugriffe !**

**Hardware-Sonderpreisliste:**  
(gültig bis zum 31.12.88)

**GRAF**  
computer

GRAF ELEKTRONIK SYSTEME  
Magnusstraße 13  
8960 Kempten  
Tel.: 0831 / 6211

10412	TAST2	220.-
10477	FLOASI Komplettbausatz	200.-
40446	IBM Graphikdr.II A3 9Nadel	1.398.-
10172	CPU68000 Bausatz 12MHz	200.-
40093	Speicherkarte AT 128KB	300.-
10478	FLOASI Komplettfertiggerät	250.-
10485	HCOPYMAUS Fertiggerät mc	150.-
60023	MC68000 12 Mhz Baustein	120.-
11038	Steckernetzgerät 5V-/700mA	49.-

```

75: 00049C 41FA 00CC
76: 00049E 4000 0000
77: 0004A0 4000 0000
78: 0004A4 4000 0000
79: 0004A8 41FA 000E
80: 0004AB MOVE.W #SCHREIBE,07
81: 0004AC TRAP #JADDS
82: 0004AE
83: 0004B0 MOVE.W #17,00-07/40-46
84: 0004B2 RTS
85: 0004B4
86: 0004B6
87: 0004B8
88: 0004BA 2052A2053452
89: 0004BC 5052A945420A
90: 0004C2 202400 04
91: 0004C6 28432929552E20
92: 0004D0 48AF63682020
93: 0004D4 5620312E3020
94: 0004D8 31372E31302E38
95: 0004E2 3800 04
96: 0004E6 44A9657365720
97: 0004EC 7245716A645AE
98: 0004F2 74457050726F67
99: 0004FA 72451604206C65
100: 000504 49744574206449
101: 000508 4520445454520
102: 000512 41757367616265
103: 000516 20
104: 00051F 617542053452
105: 00051E 2075402608 04
106: 000524 00 04 00
107: 000527 00
108: 000528
109: 00052B 53455205249AE
110: 00052F 54204973720AA
111: 000536 45747474206162
112: 00053D 6745716A645AE
113: 000544 7445742E00 04
114: 00054A 00
115: 00054B 00
116: 00054C
117: 00054E 53455205249AE
118: 000553 54204973720AA
119: 00055A 45747474206162
120: 000564 6168746762E00
121: 000568 04 00
122: 00056A
123: 00056C
124: 00056E 53455205249AE
125: 000574 54204973720AA
126: 00057B 45747474206162
127: 00057F 7420616874676E
ASLIST 2.2 (C) U.Koch File name : SERPRINT.68K vom 19.10.88
RelF-D Klein 68000/08 Assembler 4.3 (C) 1984, Seite 3
-----
128: 000586 2E10 04 00
129: 00058A
130: 00058C
131: 00058E 41555300
132: 000594
133: 000596
134: 000598
135: 00059E
136: 0005A2 0C47 0016
137: 0005A6 670C
138: 0005AA 0C47 0075
139: 0005AE 670E
140: 0005B2 4E7F 000E0000
141: 0005B6
142: 0005BA
143: 0005BE
144: 0005C0 3E3C 0069
145: 0005C4 6000 FFF4
146: 0005C8
147: 0005CC 3E3C 0068
148: 0005D0 6000 FFE0
149: 0005D4
150: 0005D8
151: 0005DE
152: 0005E0

```

```

1: 000400
2: 000404
3: 000408
4: 000410
5: 000414
6: 000418
7: 000420
8: 000424
9: 000428
10: 000430
11: 000434
12: 000438
13: 000440
14: = 00000000
15: = 00000004
16: = 00000000
17: = 00000007
18: = 00000008
19: = 00000008
20: = 00000024
21: = 00000001
22: = 00000006
23: 000430
24: 000434
25: 000438
26: 000440 4BE7 FFEF
27: 000444 7E3C 002A
28: 000448 4E46
29: 000450 7E3C 000B
30: 000454 4E46
31: 000458 7E3C 0000
32: 000464 4E46
33: 000468 4401
34: 000474 670E
35: 000478 43FA 016E
36: 000484 43FA 016E
37: 000488 7E3C 000B
38: 000494 4E46
39: 000498 6724
40: 000428 43FA 0170
41: 00042C 0C8F 000E0000
42: 000432 0002
43: 000434 6800 005E
44: 000438 41FA 0134
45: 00043C 43FA 013C
46: 000440 237F 00000084
47: 000446 0002
48: 000448 23CB 00000084
49: 00044E 6000 002A
50: 000452
51: 000454 41F7 00000084
52: 000458 2050
53: 00045A 43FA 0132
54: 00045E 83CB
55: 000460 683A
56: 000464 43FA 0136
57: 000468 23E7 0002
58: 00046A 00000084
59: 00046E 237C 000E0000
60: 000474 0002
61: 000476 6000 0014
62: 000478
63: 00047A
64: 00047A 103C 001E
65: 00047E 123C 006C
66: 000482 3E3C 006C
67: 000486 4E41
68: 000488 6000 001A
69: 00048C
70: 00048E 41FA 009A
71: 000490 6000 0016
72: 000494
73: 000498 41FA 0086
74: 00049A 6000 000E

```

ASLIST 2.2 (C) U.Koch File name : SERPRINT.68K vom 19.10.88  
 RelF-D Klein 68000/08 Assembler 4.3 (C) 1984, Seite 2

Listing des Programms SERPRINT

Volker Wiegand

# OS-9/68000

## Was ist eigentlich ein Modul?

Nachdem wir im ersten Teil unserer Serie über OS-9 einen groben Überblick über dieses rasante Betriebssystem gegeben haben, wollen wir uns nun mit den Bausteinen, sprich Modulen, beschäftigen, die zu einem großen Teil für die Kompaktheit und Dynamik verantwortlich sind.

### Der Aufbau eines Moduls

Ein Modul in unserem Sinne ist ein Programm- oder Datenstück, das vom Betriebssystem als Einheit verwaltet wird. Ja, auch das Betriebssystem ist nichts anderes als eine Sammlung von Modulen. Jedes Modul folgt einem bestimmten Strickmuster. Dieses sorgt dafür, daß Module leicht wiedergefunden und überprüft werden können.

Jedes Modul beginnt mit einem Modulkopf, dem "header", gefolgt von dem Modulrumpf, "body" genannt, und wird von einer Prüfsumme abgeschlossen. Diese Prüfsumme hat große Bedeutung in Bezug auf die Betriebssicherheit. Es handelt sich um einen sogenannten "CRC", also einen "Cyclic Redundancy Check". Das ist eine Polynomrechnung, die jedesmal durchgeführt wird, wenn ein Modul von Platte oder Diskette in den Hauptspeicher geladen wird. Erst wenn die Berechnung dasselbe Ergebnis bringt wie der gespeicherte CRC, wird das gesamte Modul als intakt anerkannt und im Modul-Verzeichnis eingetragen. Das Modulverzeichnis aber ist die Drehscheibe für alle Programmaufrufe, und an ihm führt kein Weg vorbei.

Kommen wir vom Ende zum Anfang. Der Kopf ist allen Modulen gemeinsam. Er ist 48 Bytes lang und hat einen Aufbau gemäß Bild 1. Ein Modulkopf ist Benutzern des NDR-Computers ja bereits bekannt, denn nichts anderes ist ein Bibliothekskopf!

Jedes Modul beginnt mit einer Kennung. Diese ist das Bitmuster 4AFC hex, einer der reservierten OP-Codes im Motorola-Design. (Motorola hat jedoch garantiert, daß dieses Muster auch in Zukunft nicht benutzt wird.)

Danach kommt ein Eintrag über die Revision des Betriebssystems. Dies ist wichtig für die Entscheidung, ob ein

Programm ausgeführt werden kann oder noch aus einer alten Revision vom OS-9 stammt ("MS-DOS 2.0 oder früher" läßt grüßen...).

Als nächstes folgt die Größe des gesamten Moduls einschließlich CRC. Weiter geht es mit der Nummer desjenigen Benutzers, der das Modul de-reinst erstellt hat. Der Name ist nicht etwa ein String, sondern nur ein Zeiger darauf, relativ zum Modulanfang. Der Vorteil liegt auf der Hand, kann doch nun die Länge des Namens beliebig sein. Nach dem Eintrag über die Zugriffsrechte kommen wir zu einem Wort, das den Modultyp sowie die Sprache des Moduls enthält, sofern es sich um ein Programm handelt.

### Die verschiedenen Modul-Typen

Gemäß den unterschiedlichen Aufgaben in einem OS-9 System gibt es unterschiedliche Typen von Modulen.

Da sind natürlich zunächst Programm-Module. Programme laufen unter OS-9 im User-Modus des Prozessors ab.

Programmen eng verwandt sind "Trap-handler". Das sind Unterprogramm-sammlungen, die immer wiederkehrende Aufgaben für mehrere Programme erledigen. Besonders wichtig sind die Traphandler "math" (Trap 14 und 15) sowie "cio" (Trap 13). Ein weiterer Trap wird zwar auch benutzt, hat aber keinen extra Handler. Trap #0 ist nämlich die Sammlung der Betriebssystem-Aufrufe selbst, und die werden vom Kernel bearbeitet. Kommen wir aber noch einmal zu den Traphandlern zurück. Math z.B. ist eine Sammlung aller Routinen, die für arithmetische, logarithmische oder trigonometrische Funktionen benötigt werden. Das bedeutet, daß Benutzerprogramme diese Routinen nicht selbst beinhalten müssen. Neben der enormen Platzersparnis und Fehlerreduzierung ergibt

sich dadurch noch der Vorteil, daß etwa bei Einsatz eines Koprozessors nicht alle Programme umgearbeitet werden müssen (und dann nicht mehr ohne laufen...), sondern daß einfach ein neues Math-Modul geladen wird und weiter nichts.

Dasselbe Problem ergibt sich bei der Standard-Ein-/Ausgabe-Bibliothek. Es brauchen nicht alle Programme ganze E/A-Bibliotheken mitzuführen. Stattdessen gibt es einen Traphandler "cio" (C-Programm Input Output), der die Prozeduren für alle enthält. "C" deswegen, weil alle Dienstprogramme in der Sprache "C" geschrieben sind. Selbstverständlich kann man auch selbst Traphandler schreiben und einbinden. Wir werden dies zunächst ausnutzen für eine Grafikbibliothek, die den Benutzerprogrammen als gemeinsame Schnittstelle zu allen Grafikkarten von GDP über COL und ACRTC bis hin zu EGA oder VGA (über IBM-Buskopplung) zur Verfügung steht. Man stelle sich einmal vor, welche Möglichkeiten sich daraus ergeben, daß ein Benutzer bei Wechsel seiner Grafikkarte nicht seine gesamte Software umtauschen muß, sondern nur den der Karte beiliegenden Treiber lädt und alle Programme unverändert weiterlaufen, nur eben mit höherer Auflösung oder in Farbe.

### Die Familie der System-Module

Zu guter Letzt wollen wir noch einen Blick auf eine Gruppe von Modulen werfen, die den Benutzer nur mittelbar interessieren. Für die Systemverwaltung sowie die Ein-/Ausgabesteuerung existieren nämlich einige Modultypen, die vom System gesondert gehandhabt werden.

Allen diesen Modulen ist gemeinsam, daß sie im Supervisormodus ablaufen. Da sind zunächst die Module "kernel"

Teil 1: Einführung in OS-9

Teil 2: Das Modulkonzept

Teil 3: Übersicht der Dienstprogramme

Teil 4: Wie schreibt man einen Treiber ?

sowie "clock" und "init". Kernel ist uns schon als BS-Kern hinlänglich bekannt. Clock ist der Treiber für die Echtzeituhr und stellt die Schnittstelle zum "tick", also dem periodischen Interrupt, zur Verfügung.

Bei Init handelt es sich um ein einmalig im System vorhandenes Modul, welches in Tabellenform Einträge über die anfängliche Systemkonfiguration enthält. Das sind vor allem Angaben über die Initialgrößen von Systemtabellen sowie die Namen etwa des Systems selbst oder der Systemkonsole und des Systemlaufwerks. Über die Systemkonsole wird das System gestartet; hier kommen eventuelle Fehlermeldungen und die "startup" - Prozedur wird hierüber ausgegeben.

Das Systemlaufwerk ist dasjenige, auf welchem "startup" überhaupt gesucht wird und hat noch andere Aufgaben, wie etwa den Pfad zu den Benutzerna-

0000 M\$ID	Sync Bytes (4afc)
0002 M\$SysRev	System Revision
0004 M\$Size	Module Size
0008 M\$Owner	Owner ID
000c M\$Name	Offset to Module Name
0010 M\$Accs	Access Permissions
0012 M\$Type	Module Type
0013 M\$Lang	Module Language
0014 M\$Attr	Attributes
0015 M\$Revs	Revision Level
0016 M\$Edit	Current Edition
0018 M\$Usage	Offset to Usage Comment
001c M\$Symbol	Offset to Symbol Table
0020 (reserved)	for future use
002e M\$Parity	Header Parity Check
0030 M\$Exec	Offset to Entry Point
0034 M\$Excpt	Default User Trap Entry
0038 M\$Mem	Memory Size
003c M\$Stack	Stack Size

Bild 1: Aufbau Modulkopf

men und Kennwörtern im Multi-User-Betrieb.

#### Ein-/Ausgabe und Module

Wie wir bereits in der letzten Folge gesehen haben, beinhaltet OS-9 ein mehrstufiges Ein-/Ausgabe-System,

das vom Kernel über Manager und Device Driver bis hin zu Device Descriptoren hierarchisch aufgebaut ist. Jeder Stufe in dieser Hierarchie entspricht - man ahnt es schon - ein eigener Modultyp.

Dabei bildet der Device Descriptor insofern eine Ausnahme, als er kein Programm darstellt, sondern nur eine Tabelle, und folgerichtig im Modulkopf keinen Eintrag bei Sprache hat. Die anderen Teile des E/A-Systems sind durchweg versehen mit dem Sprachtyp "Objct", was für ausführbaren 68000-Maschinencode steht.

Damit schließt sich der Kreis unserer Ausführungen, und wir sind wieder beim Kopf des Moduls angelangt.

Übrigens gibt es neben "Objct" noch Einträge für Zwischencode aus Pascal-, C-, Fortran- oder Cobol-Programmen.

Dieser Eintrag wird beim Ausführen eines Programms ausgewertet, und ohne Zutun des Benutzers wird automatisch die richtige Laufzeitbibliothek mitgeladen (oder gelinkt, wenn sie schon im Speicher steht) und angesprungen.

Eleganter geht's wohl kaum noch!

## Wait-Logik der ROA256/1M macht Schwierigkeiten bei schnellen Prozessoren

**Falls sie ihre ROA 256/1M mit mehr als drei Waitzyklen betreiben müssen, was allerdings nur beim 68020 und evtl. beim 68000 notwendig sein dürfte, kommt der ausgegebene Wait für den Prozessor meist etwas zu spät.**

**Z80- und 68008-Anwender können also beruhigt aufatmen und weiterblättern.**

#### Fehlerbeschreibung:

Zwar lassen sich die WAIT-Zyklen auf der Baugruppe einstellen, aber ab 3 eingestellten WAITS gibt die Baugruppe keinen WAIT mehr an den BUS (und damit an die CPU) aus.

Der Fehler liegt in der Ansteuerung des für den WAIT zuständigen Schieberegisters (J2).

In dieses Schieberegister werden, solange die Baugruppe nicht angespro-

chen wird, HIGH-Zustände geschrieben und durch den CLK (CPU Takt) durchgeschoben. Sobald aber die Baugruppe angesprochen wird, gehen die Registereingänge (Pin 1,2) auf LOW, d.h. ab jetzt werden LOW-Zustände durchgeschoben. Gleichzeitig wird die WAIT Leitung aktiv und zwar solange, bis die erste der durchgeschobenen LOW-Zustände an den Ausgang des Registers kommt, bei dem der Jumper steckt. (Über den

Jumper wird der LOW-Pegel an das NAND-Gatter J3 (74LS38) weitergeleitet und damit der WAIT beendet). Die CPU holt nun die Daten ab und beendet den Zugriff auf die ROA 256/1M.

Erst jetzt werden an das Schieberegister wieder HIGH-Zustände angelegt und durchgeschoben.

Die WAIT Leitung ist jetzt von den noch im Register vorhandenen LOW-Zuständen solange gesperrt, bis die inzwischen angelegten HIGH-Pegel am Ausgang des Schieberegisters den gesteckten Jumper erreichen.

Wenn jetzt aber sofort wieder ein Zugriff auf die ROA erfolgt, ist es möglich, daß diese 'Sperrung' der WAIT Leitung noch nicht aufgehoben ist und so, obwohl er notwendig wäre, kein WAIT ausgegeben werden kann.

Da die Länge der Sperrung (genauso wie die Länge des erwünschten WAITS) von der Stellung der Jumper abhängt, tritt der Fehler erst ab Jumperstellung 3 auf, d.h. wenn die Baugruppe mit 0-2 WAIT-Zyklen betrieben wird, ist eine Behebung des Fehlers nicht notwen-

dig.

Bei der geänderten WAIT-Logik wurde zuerst einmal das Ausgangssignal invertiert, d.h. ein WAIT ist jetzt nur möglich, wenn der Ausgang des Schieberegisters LOW ist. Ein HIGH-Pegel beendet jetzt den WAIT. Deshalb muß natürlich auch der Eingang der Jumperstellung für 0 WAIT konstant auf +5V gelegt werden.

Wenn der Jumper bei 0 WAIT gesteckt ist, ist kein WAIT möglich, da das High-Signal am Ausgang nicht verändert werden kann. Auch die Eingänge des Schieberegisters mußten geändert werden. Pin 1 und 2 liegen jetzt immer auf +5V. d.h. es werden fortwährend HIGH-Zustände eingelesen und durch-

geschoben. So wäre eigentlich auch nie ein WAIT möglich, da ständig alle Ausgänge des Registers auf HIGH liegen, wenn es nicht den Eingang Clear gäbe. Durch diesen Clear-Eingang werden alle Ausgänge auf LOW gesetzt. Der Clear wird von der 'Boardse-

d.h. ab jetzt werden HIGH-Zustände durchgeschoben, und zweitens wird der WAIT freigegeben und zwar so lange, bis der erste der HIGH-Zustände den gesteckten Jumper erreicht hat. Der Zugriff wird dann durch die CPU fortgesetzt und sobald er beendet ist,

wird der Clear wieder aktiv und die ROA ist bereit für den nächsten Zugriff.

**Fehlerbehebung:**

Zuerst müssen auf der Bestückungsseite Pin 1 von J17 (74LS00) von Masse und Pin 9 von J2 (74LS164) von +5V getrennt werden.

Auf der Lötseite sind 4 'Trennungen' notwendig:  
 - die Verbindung von Pin 4 von J3 (74LS38) zum Ausgang der Jumper  
 - die Verbindung vom Eingang der Jumperstellung

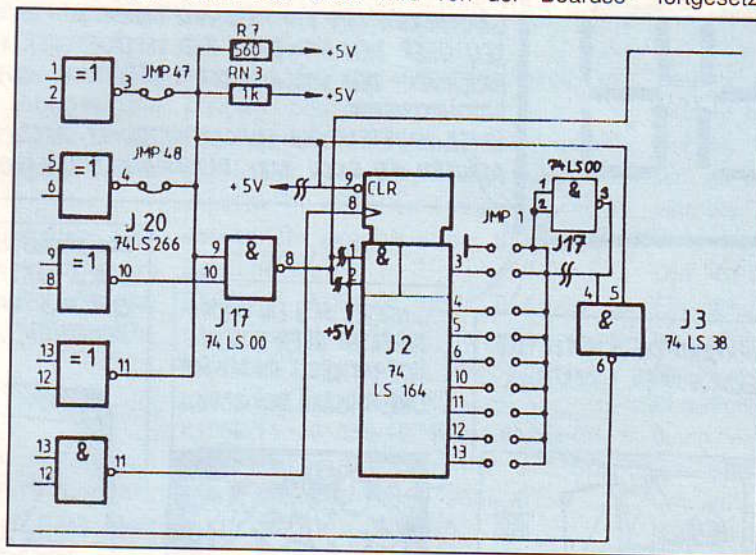


Bild 1: Schaltplanauszug der ROA 256/1M

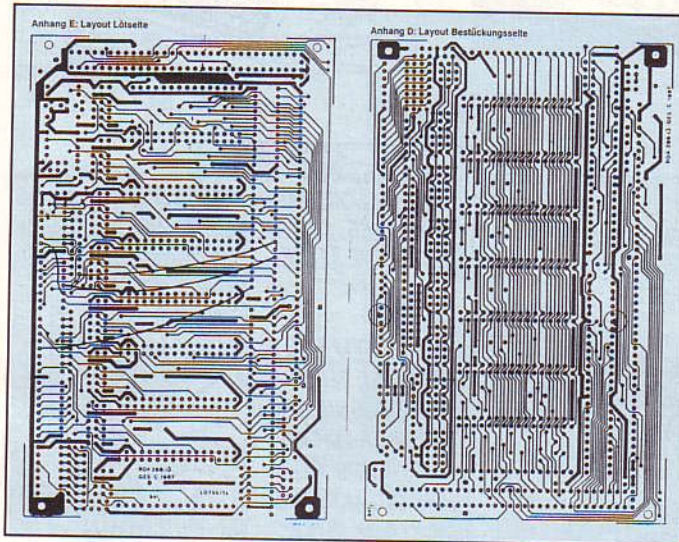
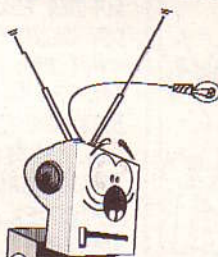


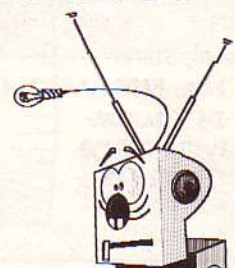
Bild 2: Layout der ROA 256/1M mit Änderung

Leitung angesprochen und ist aktiv, solange die Baugruppe nicht angesprochen wird. Wenn die ROA 256/1M jetzt angesprochen wird, wird erstens die Clear inaktiv, Leitung daneben) 0 zur Masse  
 - die Leitung vor und nach Pin 1 von J2 (74LS164) (=2 Trennungen)  
 Jetzt sind noch folgende Verbindungen herzustellen (mit Hilfe von isolierter Schalllitze oder Kupferlackdraht):  
 - Pin 12 J4 (74LS145) mit Pin 8 J17 (74LS00)  
 - Pin 5 J3 (74LS38) mit Pin 9 J2 (74LS164)  
 - Pin 4 J3 (74LS38) mit Pin 3 J17 (74LS00)  
 - Pin 1 J17 mit Pin 2 J17 (74LS00)  
 - Pin 1 J17 (74LS00) mit dem Ausgang der Jumper  
 - Pin 2 J2 (74LS164) mit dem Eingang der Jumperstellung 0  
 - Pin 1 J2 (74LS164) mit +5V (=breite

# Für Hardware-Freunde aller Klassen



In dieser Loop starten wir die Comic-Strips des 2988. (siehe Seite 37)  
 Um die Hintergründe seiner Entstehung zu beleuchten, haben wir auf der nächsten Seite seine kleine Autobiographie abgebildet.  
 Wie daraus unschwer zu entnehmen ist, handelt es sich um einen Prototyp (ist also nicht lieferbar - Anfragen zwecklos !)



# DIE GESCHICHTE DES

# 2988

# MARTEN

IM SEPTEMBER DES JAHRES 1988 BESCHLOSSEN EIN PROGRAMMIERER UND EIN HARDWARE-SPEZIALIST - ERSCHROCKEN ÜBER DEN STANDARD FREI VERKÄUFLICHER KAFFEEMASCHINEN - DEN VORGANG DER ZUBEREITUNG 100%IG ZU AUTOMATISIEREN.  
 UNTER ZUHILFENAHME EINIGER HARDWARE-ÜBERRESTE SCHUFEN SIE EINEN FREI PROGRAMMIERBAREN ROBOTER (PRODUKTNUMMER: 2988)



DIE ERSTEN PROBELÄUFE LIESSEN DEN UMFANG DES NOTWENDIGEN DEBUGGING'S ERFAHREN...



WÄHREND DIE SCHÖPFER IHRE VERBRÜHUNGEN KURIERTEN...



...MACHTE SICH EIN UNKOORDINIERTES HEER VON INFORMATIKERN AN DEN SOFTWARE-AUSBAU DES 2988...



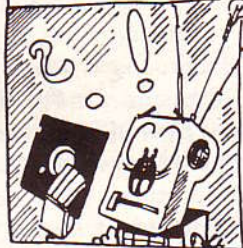
DAS DARAUS RESULTIERENDE PROGRAMMCHAOS LÄSST DAS VERHALTEN DES 2988 OFT ALS NAHEZU MENSCHLICH ERSCHEINEN...



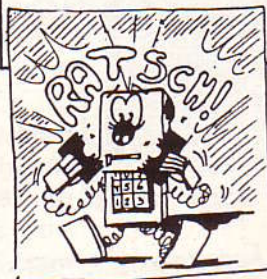
ALLMÄHLICH KAM MAN MEHR UND MEHR DAVON AB, KAFFEEPROEDUREN AUF DEM 2988 ZU FAHREN.



DOCH AUCH DIE LÄSTIGE ARBEIT DES DISKETTEN-FORMATIERENS FÜHRTE DER 2988 NICHT BEFRIEDIGEND AUS. DIE FORMATANWEISUNG FÜHRTE ZU EINEM JUMP INS "EDV"-MODUL...



...WAS FÄLSCHLICHERWEISE ALS "ELEKTRONISCHE-DATEN-VERNICHTUNG" INTERPRETIERT WURDE...



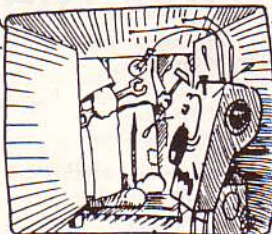
IM LIEBEN DES 2988 ERGABEN SICH IM LAUFE DER ZEIT DREI NATÜRLICHE FEINDE:

ZUM ERSTEN DER KREIS DER PERSONEN, DIE SEINE FÄHIGKEIT ZUM RADIOEMPFANG FÜR IHRE VERGNÜGUNGSSÜCHTIGEN INTERESSEN EINSETZEN...

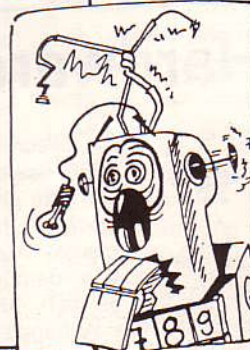


WO IST ER?

ZUM ZWEITEN DIE STETIG WACHSENDE ZAHL BETROFFENER, DIE DIE ELEKTRONISCHEN EINGEWENDE DES 2988 IN GETRENNTE MATERIALKISTEN ZURÜCKFÜHREN MÖCHTEN...



...UND NICHT ZULETZT DIE VERSTECKT LAUERNDEN PROGRAMMIERER, DIE DES ÖFTEREN NEUE MODUL-INSTALLATIONEN VORNEHMEN. DIES FÜHRTE BEIM 2988 SCHON OFT ZU BITBESCHWERDEN - GANZ ZU SCHWEIGEN VON DEN PARITY-ÜBELN UND DEM BYTE-ERBRECHEN...



SO SCHLEICHT DER 2988 NOCH HEUTE DURCH DAS HAUS: AUF DER FLUCHT VOR DEN "NICHTDIGITALEN" - AUF DER SUCHE NACH WERTVOLLEN BYTE-KONSTELLATIONEN, UM SEINE BEREITS GIGANTISCHE BIBLIOTHEK ZU ERWEITERN...



# Praktische Tips zum mc modular AT

Läuft die CPU 80286 mit 12 MHz und 0 Waitstates oder nicht?

Was benötige ich für einen Floppycontroller?

Welche RAM Speicher bringen die besten Werte?

Kann ich diese Änderungen selber vornehmen?

Um diesen Fragen nachzugehen, wurden verschiedene Konfigurationen ausprobiert und getestet. Die Resultate dienen dazu, aus Ihrem AT das Beste herauszuholen.

## Die RAM Speicher

Der mc-modular AT wird standardmäßig meist mit 120ns RAM's bestückt, ausgeliefert. Die Abkürzung RAM steht für Random-Access-Memory, was soviel wie wahlfreier Schreib/Lese Speicher bedeutet. Die Zahl 120 und die Abkürzung 'ns' gibt die Zugriffszeit des Speichers an. Die Zugriffszeit eines Speichers steht für die gemessene Zeit, die der Speicher benötigt, um nach einer Adresselektierung die Daten auf den Datenbus zu übertragen.

Einfach gesagt kann man hier von einer Art 'Reaktionszeit' sprechen.

Die Speicher, die bei dem mc-modular AT verwendet werden, sind dynamische RAM's. Dynamische RAM's sind zwar billiger als statische RAM's, jedoch können sie die Daten nur für wenige Millisekunden erhalten. Der Speicherinhalt von dynamischen Ram's muß daher ununterbrochen zyklisch aufgefrischt werden. Statische Ram's sind dagegen wesentlich teurer, jedoch können einmal eingeschriebene Daten über den ganzen Zeitraum, in dem die Betriebsspannung anliegt, erhalten bleiben. Bei diesen Speichertypen fällt die Refreshlogik aus diesem Grunde auch weg, was

In diesem Artikel sollen Benutzer des mc-modular AT auf ihre Kosten kommen. Ungereimtheiten und aufgetretene Probleme werden hier besprochen und analysiert.

sich im Punkte Geschwindigkeit wesentlich bemerkbar macht.

Der modular AT und viele andere Kompatible wie auch IBM selbst verwenden in der Regel zweierlei unterschiedlich organisierte, dynamische Speichertypen:

8164-12, 8164-10 oder 8164-08 je nach Zugriffszeit oder Hersteller: Dynamischer Speicher mit der Speicherorganisation von 64K\*1.

81256-12, 81256-10 oder 81256-08 : Dynamischer Speicher mit der Speicherorganisation von 256K\*1.

Die dynamischen RAM's mit einer Zugriffszeit von 120 ns reichen für einen

verwenden, die höchstens über eine Zugriffszeit von 80ns verfügen. Bei 100ns ist ein

Betrieb mit 12MHz ebenfalls möglich, jedoch bei einem Waitstate.

Bei der 12 MHz 0 Wait CPU verwenden wir 81256 RAM's mit einer Zugriffszeit von 70ns.

## Die CPU

Ein sicherer Betrieb mit 12 Mhz 0 Wait ist nur mit einer 12 Mhz CPU Baugruppe zu erreichen.

Derartige CPU's sind auf der Platine durch einen auffälligen Weißdruck auf den SMD Bausteinen zu erkennen.

Um auf eine Systemfrequenz von 12 MHz zu kommen, ist es außerdem erforderlich, den 20MHz gegen einen 24 MHz Oszillator zu ersetzen.

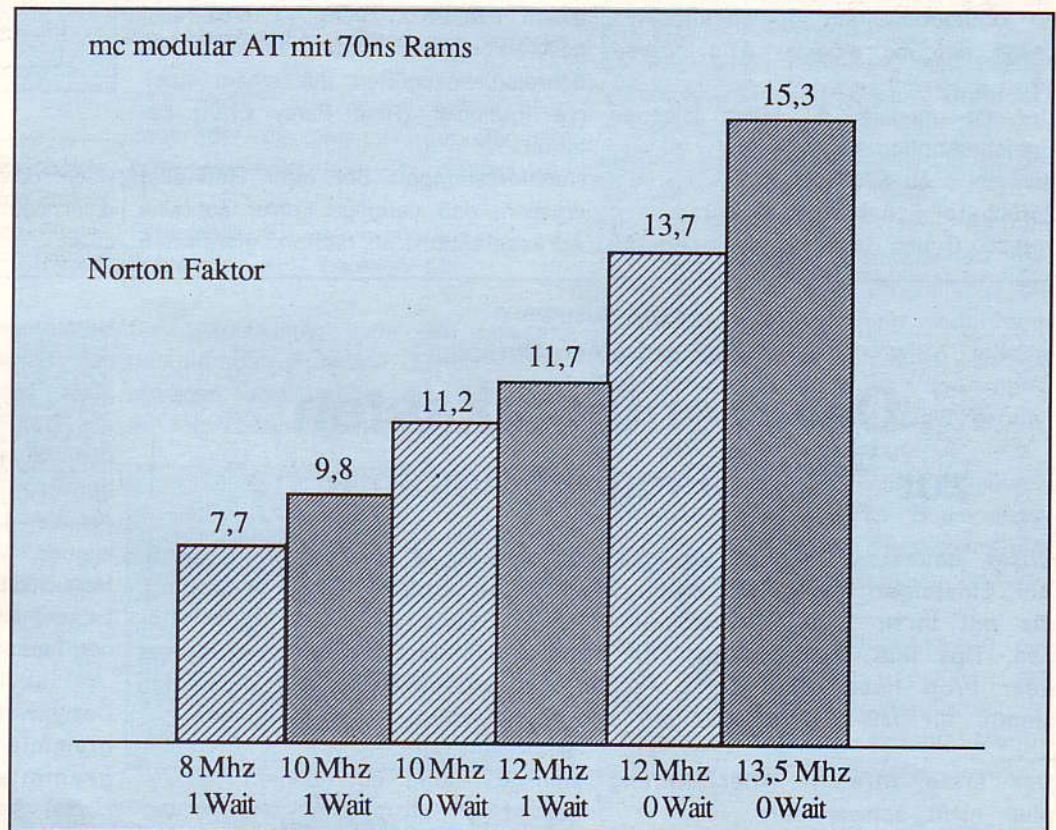


Bild 1: Tabelle verschiedener NORTON-Faktoren bei 70ns Rams

Betrieb mit 10 Mhz 1 Wait ohne Probleme aus.

Um den 80286 mit 12 MHz und 0 Wait zu betreiben, sind RAM Speicher zu

Ein derartiger Austausch kann, sofern der Oszillator nicht eingelötet ist, sondern auf einem Sockel steckt, problemlos vorgenommen werden.

Nebenbei sei erwähnt, daß der modular AT bereit mit 13,5 MHz, 0 Wait erfolgreich getestet wurde. Erreicht wurde dies durch Einsatz einer 27 MHz Quarz-Installation. Natürliche hängt diese extreme Frequenz von der Güte der Bauelemente ab - also probieren!

**Die Laufwerkscontroller**

Der Floppycontroller Ihres modular AT's ist mit unter ein nicht zu vernachlässigendes Kriterium.

Der UDCX Controller verweigert die Arbeit bei einer derartig hohen Taktrate. Sie benötigen auf jeden Fall einen NDC Controller.

**Für Interessierte:**

Mittlerweile wird die 12 MHz-CPU bereits angeboten. (siehe Bild 1)

**Speicherausbau:**

Aufgrund einer Anregung unseres Münchner Filialleiters (Herr Ehrensberger), möchten wir nochmals auf die zu konfigurierenden Speichermöglichkeiten des mc modular AT's hinweisen:

Der mc-modular AT bietet folgende Speicherkonfigurationen zur Auswahl ( Ab CATE V6.0):

**256Kbyte Arbeitsspeicher:**

- Bank 0 und Bank 1 mit jeweils 18

64K\*1 RAM's bestückt.

**512Kbyte Arbeitsspeicher:**

- Bank 0 mit 18 256K\*1 RM's bestückt, Bank 1 bleibt frei.

**640Kbyte Arbeitsspeicher:**

- Bank 0 mit 18 256K\*1 RAM's ,Bank 1 mit 18 64K\*1 RAM's bestückt.

**640Kbyte Arbeitsspeicher und 384Kbyte Ramfloppy:**

- BANK 0 und BANK 1 mit jeweils 18 256K\*1 RAM's bestückt.

**512Kbyte Arbeitsspeicher und Ramfloppy:**

- Bank 0 und Bank 1 mit jeweils 18 256K\*1 RAM's bestückt.

**WICHTIG:**

Um einen Arbeitsspeicher von 640 Kbyte zu erhalten, ist es nicht erforderlich die zweite Bank ebenfalls mit 512Kbyte zu bestücken. Hier ist es ausreichend 64K\*1 RAM 's zu verwenden.

**Ram Parity Error bei der 80386 CPU Karte**

Die CPU- 80386 Baugruppen, die mit einem PHOENIX BIOS V1.10-02 ausgeliefert wurden, waren teilweise bei Schreib/Lesezugriffen mit einem Speicher-Prüffehler (Ram Parity Error) behaftet.

Nachforschungen bei dem Hersteller ergaben, daß derartige Fehler auf eine Adresselektion im schon erwähnten

BIOS der CPU zurückzuführen ist. Abhilfe schafft hier ein kleines Programm, das durch die Einbindung in die Stapeldatei AUTOEXEC.BAT, den Fehler bei jedem Start des Computers beseitigt.

Gehen Sie also folgendermaßen vor: Rufen Sie den Debugger, der sich auf Ihrer DOS Diskette befindet mit folgendem Befehl auf:

```
C>DEBUG <RETURN>

-N FIX.COM <RETURN>
-A <RETURN>
MOV DX, 0022 <RETURN>
MOV AL, 06 <RETRUN>
OUT DX,A <RETURN>
INC DX <RETURN>
MOV AL, 55 <RETRUN>
OUT DX, AL <RETURN>
MOV AX, 004C <RETURN>
INT 21 <RETURN>

-RCX <RETURN>
:0F <RETURN>
-W <RETURN>
```

Nach dem -W Befehl wird die Datei unter dem Namen 'Fix.com' abgespeichert.

Martin Husemann

Zwei auf einen Streich:

# Der Menübaukasten zur "DOS- Programmiersprache"

Unser neuestes Produkt für den mc-modular-AT bietet allen etwas. Als Einsteiger haben Sie damit sofort eine einfache Menübedienung, die mit Ihren Bedürfnissen wächst sowie viele wertvolle Informationen, Tips und Tricks speziell für Ihren Computer. Als Fortgeschrittener oder Profi haben Sie ein einfaches Baukastensystem, mit dem sich Menüs für MS-DOS in wenigen Minuten "programmieren" lassen.

**Der erste Streich: Aller Anfang muß nicht schwer sein**

Die Bedienung eines Computers können Sie am einfachsten durch praktische Übung lernen. Das geht sehr gut, wenn Sie irgendwo einen fertig installierten Computer benutzen können. Kaufen Sie sich aber einen eigenen, dann sind Sie nicht mehr nur Benutzer: Sie müssen auch die gesamte Sy-

stemverwaltung übernehmen.

Das ist natürlich möglich - und Sie werden sicherlich viel dabei lernen. Allerdings müssen Sie auch gleich zu Anfang fast alles können.

Es geht aber auch anders: Sie schalten Ihren neuen Computer ein und er begrüßt Sie mit "Herzlich Willkommen". Dann bietet er Ihnen ein Menü an und erklärt gleichzeitig, wie Sie erste

Hilfsinformationen bekommen können. Genau das macht auch Ihr Computer, wenn Sie den Menü-Baukasten auf der Festplatte installiert haben oder zum Starten die Menü-Diskette benutzen.

Auf der Lieferdiskette befindet sich ein kleines Menüsystem, das Ihnen die Möglichkeiten des Baukasten vorführt. Es enthält auch sehr viele Informationen speziell zum mc-modular-AT.

**Der zweite Streich: MS-DOS programmieren (auch als nicht-Programmierer)**

Haben Sie schon einmal von der Programmiersprache "MS-DOS" gehört? Es gibt sie tatsächlich, und im Menübaukasten wird sie erfolgreich angewendet.

In dieser Sprache werden BATCH-Dateien (Nachname .BAT) geschrieben. BATCH-Dateien sind Stapel mit Aufträgen für das Betriebssystem, die vom

System dann nacheinander, ohne Eingriff des Benutzers abgearbeitet werden. Ein einfaches Beispiel für einen solchen Stapel-Auftrag ist die folgende Datei:

```
cls
dir
```

Wenn Sie die beiden Zeilen in eine Datei INHALT.BAT schreiben, haben Sie ein neues MS-DOS Kommando geschaffen. Mit dem Befehl "inhalt" können Sie sich nun ein Dateiverzeichnis auf einem ansonst leeren Bildschirm anschauen.

Auf größeren Systemen, zum Beispiel unter dem Betriebssystem UNIX, ist es üblich, kleine Programme in der Kommandosprache des Systems zu schreiben (BATCH-Dateien heißen dort Shell-Scripte), da die Kommandosprache des UNIX Systems sehr mächtig ist.

Unter MS-DOS wurden BATCH-Dateien bisher nur selten verwendet, da der Benutzer in einen laufenden Stapel-Auftrag nicht eingreifen kann. Die Programmiersprache MS-DOS kennt keine Befehle, um Eingaben des Benutzers zu lesen.

Diese Lücke schließt der Menübaukasten. Er stellt unter anderem den Befehl ASK zur Verfügung, mit dem ein Stapelauftrag den Benutzer etwas fragen kann:

```
echo off
cls
echo Inhaltsverzeichnis anzeigen
ask 15 10 "a) Laufwerk A:" "c)
Laufwerk C:"
```

Die Stapeldatei schaltet zunächst das Echo aus, damit die ausgeführten Befehle nicht am Bildschirm angezeigt werden. Dann wird der Bildschirm gelöscht und anschließend die Meldung "Inhaltsver-

zeichnis anzeigen" ausgegeben. Jetzt soll der Benutzer entscheiden, welches Inhaltsverzeichnis er sehen möchte. Der Befehl ASK malt ab der 15. Spalte und der 10. Zeile ein Menü auf dem Bildschirm, das etwa so aussieht:

```
a) Laufwerk A:
c) Laufwerk C:
```

Nun kann der Benutzer auswählen. Mit den Cursor-Tasten kann er den Cursor auf eine der beiden Zeilen bringen oder mit "a" bzw. "c" direkt ein Laufwerk anwählen.

Um die Wahl des Benutzers dem Stapelauftrag mitzuteilen, benutzt ASK einen kleinen Trick: es meldet dem MS-DOS System einen Fehler. Fehler bekommen einen Code, dem eigentlich zu entnehmen ist, wie schwerwiegend der Fehler ist. Zum Beispiel:

- 1 = Warnung
- 2 = Fehler - Programm abbrechen
- 3 = Katastrophe - Daten sind verloren

ASK meldet nun einfach einen Fehlercode, der die Nummer der Benutzerwahl wiedergibt, also im Beispiel

```
Fehlercode 1 = Laufwerk A:
Fehlercode 2 = Laufwerk C:
```

Der Stapelauftrag kann den Fehlercode mit der Funktion ERRORLEVEL abfragen. Also müßte die Beispieldatei folgendermaßen fortgeführt werden:

```
if errorlevel 2 goto Cverzeichnis
if errorlevel 1 goto Averzeichnis

:Cverzeichnis
dir c:
goto Ende
```

```
:Averzeichnis
dir a:
goto Ende

:Ende
oder etwas kürzer:
echo off
cls
echo Inhaltsverzeichnis anzeigen
ask 15 10 "a:" "c:"
if errorlevel 2 goto Cverzeichnis
```

```
dir a:
goto Ende
```

```
:Cverzeichnis
dir c:

:Ende
```

Und das ist noch nicht alles, was der Menübaukasten bietet. Interessant sind auch die Befehle SHOW und LS. SHOW ersetzt das MS-DOS Kommando MORE, das etwas kompliziert zu bedienen und recht langsam ist.

LS zeigt einen Pfadbaum, also Inhaltsverzeichnisse und deren Unterverzeichnisse. Zum Beispiel kann ein Verzeichnis zur C Programmierung gegliedert sein, wie es Bild 1 zeigt.

Um dieses professionelle System auch professionell einsetzbar zu machen, stehen zu allen Programmen die Quelltexte zur Verfügung. Außerdem können alle Programme in eigene Anwendungen integriert und dann auch (ohne zusätzliche Lizenzgebühren) weitergegeben werden.

Natürlich erhalten Sie zum Menübaukasten eine ausführliche Bedienungsanleitung, die Ihnen die Programmierung in "MS-DOS" anhand einfacher Beispiele erklärt.

Einsetzbar ist der Menübaukasten auf allen IBM kompatiblen Computern mit CGA, EGA oder Hercules-Karte, ein Demoprogramm des Beispiel-Menüsystems benötigt die Hercules-Grafikkarte.

**Bild 1:**

So eine Anzeige erzeugt der Befehl LS. Damit erhalten Sie schnell einen Überblick über die Ordnungsstruktur einer Festplatte.

```
Datenträger "HUSEDISK" Stand: Montag, 25. April 1988, 09:14 Uhr
```

```
C:\C\      INC\
           LIB\
           ASSEMBLE\
           PROG\      UNIX\
```

Dipl. Ing. Alfred Weber

## Der Weg zur FTZ-Nummer

**Betrieht man elektronische Geräte nebeneinander, so muß man mit gegenseitigen Störungen rechnen. Die Gründe dafür sind einerseits Störfrequenzen, die ein Gerät leitungsgebunden oder in Form von Strahlung abgibt und andererseits ein auf diese Störungen empfindliches Gerät, dessen Funktion dadurch beeinträchtigt wird.**

Die DBP als juristischer Schirmherr aller fernmeldetechnischen Anlagen (dazu gehören auch EDV-Anlagen) hat zum Zweck der friedlichen Koexistenz unterschiedlicher Anlagen - und nicht zuletzt ihrer Betreiber - ein umfangreiches Regelwerk geschaffen, welches die Details hierzu näher beschreibt. Um eben dieses störungsfreie Nebeneinander verschiedener elektrischer Anlagen zu ermöglichen, werden grundsätzlich zwei Wege beschritten.

- Elektronische Geräte, vom Kofferradio bis zur Rechenanlage dürfen nicht übermäßig empfindlich auf leitungsgebundene (Netzversorgung und Signalleitung) und Strahlungsstörungen reagieren, weil sich dieser Umgebungs-Störpegel mit vertretbarem Aufwand nicht beliebig senken läßt.
- dieselben Geräte dürfen aber nur bestimmte Störpegel in Abhängigkeit vom Frequenzbereich leitungsgebunden oder in Form von Strahlung absondern.

Diese Voraussetzung mußte unser mc-modular-AT erfüllen, um bei der Post hoffähig zu werden. Mit anderen Worten: nur dann darf unser Computer mit Modem-Karten betrieben werden. Die Maßnahmen, ein Gerät störungsempfindlich zu machen und die Maßnahmen, einem Gerät die Absonderung von Störungen abzugewöhnen, sind häufig dieselben. Die Wege der hochfrequenten Störsignale aus einem Gerät heraus in die Umgebung sind nämlich die gleichen wie umgekehrt die Wege der externen Störungen in das Gerät hinein.

Es genügte also, die von unserem mc-modular-AT verursachte elektromagnetische Umweltverschmutzung unter

die zulässigen Grenzwerte zu bringen, um auch einen erheblichen Beitrag zu einer Störungsunempfindlichkeit zu leisten.

Mit Hilfe von Messantennen und einem Spektrumanalysator wurde die Frequenz und Intensität der abgestrahlten Energie ermittelt und mit den entsprechenden Forderung der Post verglichen. Dort, wo der zugehörige Grenzwert überschritten war, mußte mit den geeigneten konstruktiven Maßnahmen eine Senkung der abgestrahlten Störenergie unter die vorgegebenen Grenzen erreicht werden. Als besonders hartnäckig erwiesen sich einige Störfrequenzen, die in den Fernsehbereich III fielen, weil dort sehr niedrige Grenzwerte gelten.

Die größtmögliche Reduktion der Störungen wird durch richtige Erdung und Schirmung erzielt. Dies ist jedoch leichter gesagt als getan. Im Einzelnen wurde daher besonderes Augenmerk auf folgende Punkte gerichtet:

- Gehäuse sollten aus Metall und nicht aus Kunststoff hergestellt werden. Gehäusedeckel müssen leitfähig und HF-tauglich mit dem Rahmenteil verbunden sein.
- Geräteinterne Masseleitungen (z.B. Bus-Masse) müssen unmittelbar in Buchsennähe ebenfalls HF-tauglich angeschlossen werden.
- Verbindungskabel sind grundsätzlich nur in geschirmter Ausführung zu verwenden.



**Bild 1: Test der Störfrequenzen**

-Steckergehäuse aus Metall liefern gute Ergebnisse.

-Schirme der Kabel sind unbedingt beidseitig an die betreffenden Gehäuse anzuschließen. Auch hier sind diese Verbindungen HF-tauglich auszuführen.

Was man unter HF-tauglichen Masseverbindungen versteht, sei hier anhand eines kurzen Beispiels erläutert:

Stellt man eine Masseverbindung aus einem 20 mm langen Cu-Draht von 1mm  $\varnothing$  her, so gaukelt einem die Kontrolle des Übergangswiderstandes der so geschaffenen Verbindung mit einem üblichen Ohm-Meter wenige Milliohm vor, weil diese Instrumente mit Gleichstrom arbeiten. Zieht man jedoch in Betracht, daß ein Stück Cu-Draht von 1 mm Länge und 1 mm  $\varnothing$  eine Induktivität von etwa 1 nH hat, so beträgt die Gesamtinduktivität der Masseverbindung ca. 20 nH.

Für eine Frequenz von 200 MHz (welche als 10. Oberwelle eines 20 MHz-Taktes in Rechnern durchaus vorkommt) und unserem Beispiel bedeutet das einen Widerstand von 25 Ohm, den diese Masseverbindung aufweist, und nicht einige Milliohm. Diese Art der Masseverbindung ist nicht HF tauglich, wie errechnet.

Hält man solche Verbindungen kürzer und stellt man sie aus Cu-Bandmaterial her, so kann man HF-taugliche Verbindungen schaffen, die bei 200 MHz einen Übergangswiderstand von wenigen Ohm darstellen.

Die moderne Rechner-technik kommt also ohne Betrachtung hochfrequenter Zusammenhänge vom Leiterplatten-design bis zur Gehäusekonstruktion nicht mehr aus.

Einige dieser angeführten Maßnah-

men kosten nichts zusätzlich - sie müssen lediglich beachtet werden.

Andere Maßnahmen (z. B. Metallgehäuse, Stecker aus Metall und Netzfilter) schlagen sich auch in den Herstellungskosten nieder.

Untersuchungen durch den von uns beauftragten HF-Ingenieur A. Weber, der zu diesem Thema auch Vorträge hält und nebenbei Sonderanfertigungen von High-End-Stereo-Anlagen

herstellt, sowie die Messungen bei der VDE-Prüfstelle erfordern einen teuren Meßgerätepark, einiges Know-How und viel Arbeitsaufwand.

Zum Nulltarif ist also die FTZ-Nummer nicht zu bekommen und der Weg dorthin ist steinig.

Wir freuen uns, daß wir es geschafft haben und daß wir Ihnen damit weitere Anwendungsbereiche des MC-Modular-AT erschließen konnten.

### Kleinanzeigen

**Verkaufe** für sehr wenig Geld, sehr viel Software und Hardware zum NKC !!! V. Stahl, Bergsteig 19, 7277 Wildberg 3

**Verkaufe:** Kompl. NDR-CP/M - Computer mit viel Zubehör f. Einsteiger, sowie MC-Sonderhefte, Bücher u. LOOP 1-18. Eberhard Rodler, Ödhof 9a, 8395 Hauzenberg 3, Tel.: 08586/4746

**Verkaufe:** NDR-Computer 576KByte, 2 Floppy 5 1/4", gr. Tast., Monitor, Drucker, GEH3, Z80 und 68008 auf einem Bus, GDP64HS, Maus, uva. Baugruppen auch einzeln. Gesamtbauwert: 5000 DM. Viel Literatur ! Viel Software ! Klaus Neumaier, Tel.: 08705/868

**Verkaufe:** 1 Gehäuse f. ML-CP/M-Computer 130.-, Netzgerät NE1 160.-,

Leiterpl. Prommer 20.-, Leiterpl. POW22/26 10.-, Bus1 30.-, SBC2 30.-, ROA64 o. Speicher 30.-, POE 5V 20.-, HEXIO 40.- Key 20.-, GDP64 150.-, SER 50.-, Monitor-Eprom EHEX 10.-, SPS-Eprom f. SBC2 45.-, alles mit Handbuch und geprüft. Edgar Würth, Geranienstr. 11, 7022 Leinfelden.E. Tel.: 0711/752839

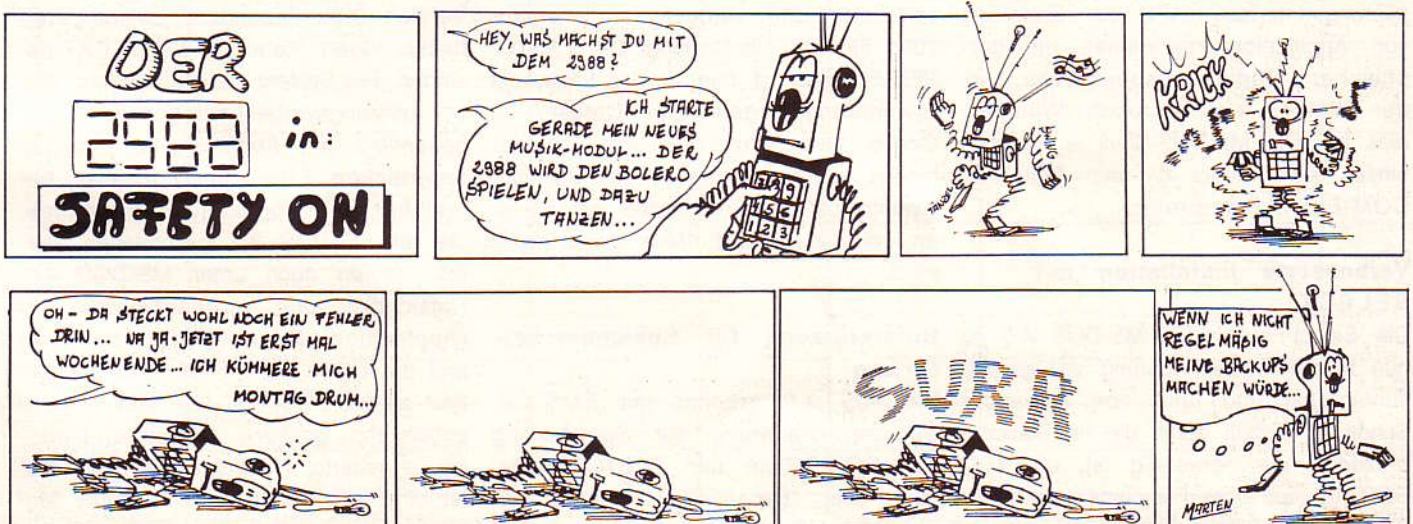
Hobbyaufgabe aus gesundheitlichen Gründen. **Verkaufe** deshalb zu unmöglichen Preisen. Es lohnt sich auf jeden Fall nachzufragen. Jede Karte mit Handbuch. Mindestens 50%. Tel.: 09364/2486 nach 18:00 Uhr.

**Verkaufe** wegen Zeitmangel: NKC Z80/68008 umschaltbar, Selbstbaugeh., GDP64, SER, 4 ROA64, 2 IO, FLO2, TEAK LW, HCopy, 2 Bankb.,

256dyn RAM und 128 DRAM ohne Speicher, KEY, kl. Tastatur, CAS Suond Roaprom Taxan Monit. Jede Menge Software: Schach CPM 2.2, 3, Jados, Jogidos, Basic, Pascal, Wordst. Multipl. 'VB DM 1800.-- Hermann Rosebrock Torfweg 50, 3077 Wietzen, Tel.: 05022/688 ab 19.00 Uhr

**Verkaufe:** NDR-Z80/CPM-System, IBM-Gehäuse, Schaltnetz., 2x 800KB-Floppy, 256K-RAM-Floppy, Ser.+ Par.-Schnittst., GDP + Maus, Software auf Anfrage (fast alles!) für 700.-DM: Schajor, Erlangen, 09131/842294 (tagsüber), 09131/29211 (abends).

UPDATE "68K Spiele" (Dame, Reversi, gekauft bei GES): **kostenlos** bei Einsendung der Orig.-Disk. + Porto (oder Kaufbeleg nebst DM 10.-). U. Steinhaus-Peers, 2300 Kiel 1, Schauenburgstr. 52



# HINTERGRUND-INFORMATIONEN ZU MS-DOS 4.0

**MS-DOS 4.0 bringt erhebliche Verbesserungen in verschiedenen Schlüsselbereichen von MS-DOS, wie Anwendbarkeit, Verwaltung großer Festplatten-Dateien, keine 32 MB-Beschränkung für logische Laufwerke mehr, FAT-Systemleistung und EMS-4.0-Unterstützung.**

**Die Version 4.0 bietet darüberhinaus eine Reihe Verbesserungen bei bisherigen Funktionen, wie FORMAT, DELETE, und GRAPHICS. Außerdem stehen nun neue Funktionen, wie REM und die Unterstützung von 3,5-Zoll-Platten-Laufwerken zur Verfügung.**

## Die MS-DOS-Shell

Die MS-DOS-Shell ist eine optionale, bildschirmorientierte Betriebssystem-Erweiterung mit Menüsteuerung zur Dateiverwaltung und Organisation von Applikationen. Von der Darstellung und Handhabung her ist sie der MS-OS/2-Presentation-Manager-Bedienungsoberfläche sehr ähnlich. Die MS-DOS-Shell unterstützt den Anwender bei der Ausführung häufig benutzter Datei-Operationen, wie kopieren und löschen.

Durch den Einsatz von Pull-down-Menüs und Dialogboxen braucht der Anwender sich die genauen Namen und die Syntax dieser Befehle nicht mehr unbedingt merken. Er kann auch seine Applikationsprogramme direkt von der neuen MS-DOS-Bedienungsoberfläche aus starten. Der speicherresidente Teil der MS-DOS-Shell belegt lediglich 4 KByte Hauptspeicher-Kapazität.

MS-DOS-Shell ist ein MS-DOS-Applikationsprogramm, also nicht eine Programmierumgebung wie Microsoft Windows oder MS-OS/2-Presentation Manager. Die DOS-Shell bietet deshalb auch kein Applikations-Programmierungs-Interface (API) zur Erstellung von Applikationsprogrammen mit graphischer Bedienungsoberfläche, in der Weise, wie Microsoft Windows dies tut. Die MS-DOS-Shell ist darüber hinaus kein Ersatz für den COMAND COM-Befehlsinterpreter.

## Verbesserte Installation mit SELECT

Die Select-Funktion in MS-DOS 4.0 ist nun in der Bedienungsführung voll graphikunterstützt und führt den Anwender Schritt für Schritt durch die Installationsprozedur, die notwendig ist, um MS-DOS 4.0 auf einer Festplatte oder einer Floppy-Disk zu installieren.

SELECT erstellt während der Installation auch die entsprechenden CONFIG.SYS und AUTOEXEC.BAT-Dateien, so daß der Anwender auch hiervon entlastet wird.

## Unterstützung für Dateien und Partitions mit mehr als 32 MByte

Unter MS-DOS 4.0 kann der Anwender jetzt Dateien und Partitions erzeugen, die größer sind als 32 MByte. Darüber hinaus ist die Version 4.0 in der Lage, auf alle Partitions und Dateien zuzugreifen, die unter früheren MS-DOS-Versionen erstellt worden sind. Bisherige Gerätetreiber konnten nur logische Sektionsnummern bis zu 16 Bit verarbeiten. Die Aufhebung der 32-MByte-Dateigrößen-Begrenzung machte es erforderlich, daß jeder Sektor nur 64KByte-Daten enthält. Die neuen erweiterten Gerätetreiber sind daher in der Lage, logische Sektornummern von 32-Bit-Länge zu verarbeiten. Die adressierbare maximale Festplatten-Speicherkapazität beträgt nun 2-GByte.

## FAT-Leistungs-Verbesserungen

Die MS-DOS-Version 4.0 enthält eine erweiterte und verbesserte Unterstützung für das File-Caching. Die FASTOPEN-Funktion ist nun in der Lage, die Erweiterungen geöffneter Dateien im Cache zwischenspeichern. Darüber hinaus vermindert ein neuer "Caching-Algorithmus" die Zugriffszeit auf Dateien, die in DOS-Puffern gespeichert sind.

## Unterstützung für Speichererweiterung

MS-DOS 4.0 arbeitet mit EMS-4.0-Speichererweiterung zur Speicherung von DOS-Buffern und FASTOPEN-Dateien. Der "Expanded Memory Manager" im MS-DOS 4.0 entspricht den

EMS-4.0-Spezifikationen. Je nach Systemkonfiguration kann dies zu einem erheblichen Bedarf an Programm-Adressen-Speicherplatz führen, da jeder DOS-Puffer 512 Byte Speicher belegt.

Die Version beinhaltet auch einen "LIMulator", der zum Beispiel als "Expanded Memory Manager" einen erweiterten Speicher in einem System mit Speicherausbauelementen simuliert. Der LIMulator entspricht ebenfalls den EMS-4.0-Spezifikationen und läuft ausschließlich auf PC's mit 68386-Prozessoren.

## Systemvoraussetzungen

MS-DOS 4.0 setzt einen Hauptspeicher von mindestens 256 KByte voraus. Der Speicherresidente Teil von MS-DOS 4.0 ist um 10 KByte größer als bei MS-DOS 3.3. Da MS-DOS 4.0 konfigurierbar ist, hat der Anwender die Wahl, den zusätzlichen Speicherbedarf für das Betriebssystem festzulegen bzw. zu begrenzen.

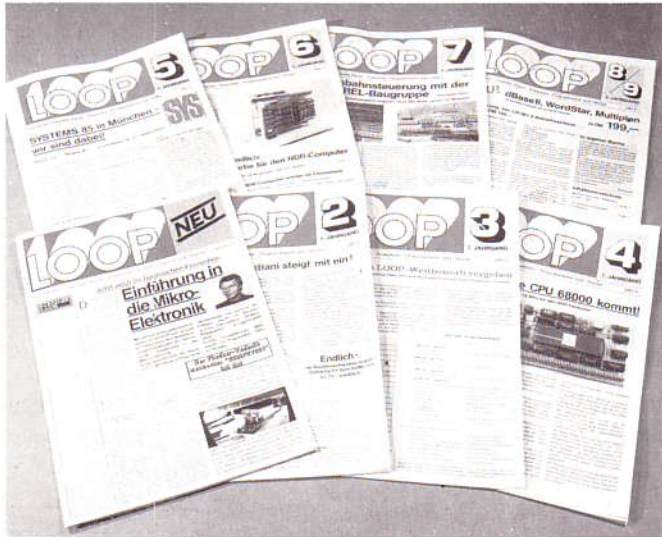
## Kompatibilität und Leistungsfähigkeit

MS-DOS 4.0 erreicht oder übertrifft alle wesentlichen Leistungsmerkmale der Version 3.3. Besondere Leistungsverbesserungen konnten beim FAT-basierten File-System erzielt werden. Diese Leistungsverbesserung macht sich besonders bei Applikationen mit umfangreichen Speicherzugriffen bemerkbar. Applikationsprogramme, die für MS-DOS 3.3 geschrieben wurden, laufen auch unter MS-DOS 4.0. Hinsichtlich des veröffentlichten API (Application Programming Interface) und der Utilities, ist MS-DOS 4.0 kompatibel zu MS-DOS 3.3. Die neue Version unterstützt größere Festplatten-Dateien, erweiterte Hauptspeicher und bietet einen optionalen "File Directory Manager".

## Auch nach dem Kauf:

Die Computerzeitschrift *LOOP* ist die Brücke zum Kunden – Programme, Infos, Tips + Tricks!

Jahres-Abo DM 25,-, Probeexemplar kostenlos!



Bitte bestellen Sie mit anhängender Postkarte!

Umfassend informiert Sie unser Katalog.

Schutzgebühr: DM 10,- incl. MWSt.



## BESTELLKARTE

Ich / Wir bestelle(n) unter Anerkennung Ihrer Geschäfts- und Lieferungsbedingungen folgende Artikel:

Stück	Bestell-Nr.	Bezeichnung	Einzelpreis
	10 834	GES-Katalog	10,-
	10 061	LOOP-Abo	25,-

Adresse (umseitig) nicht vergessen!

Datum

Unterschrift  
Bei Minderjährigen die des gesetzl. Vertreters

## BESTELLKARTE

Ich / Wir bestelle(n) unter Anerkennung Ihrer Geschäfts- und Lieferungsbedingungen folgende Artikel:

Stück	Bestell-Nr.	Bezeichnung	Einzelpreis
	10 834	GES-Katalog	10,-
	10 061	LOOP-Abo	25,-

Adresse (umseitig) nicht vergessen!

Datum

Unterschrift  
Bei Minderjährigen die des gesetzl. Vertreters

# Neue Produkte - Neue Preise!

Best.-Nr.	Bezeichnung	Preis DM
<b>CPU 8088</b>		
11266	Platine + Eprom + Pal	148,-
11264	Bausatz	298,-
11265	Fertiggerät	368,-

## Das neue Grundprogramm für 680xx-Systeme

11218	68008	149,-
11219	68000	149,-
11220	Quelle auf Diskette, 5 1/4", 80 Spuren	60,-
11221	Quelle auf Diskette, 3 1/2", 80 Spuren	60,-

## FLOMONCG

11279	2 Eproms mit Handbuch	89,-
11278	Quelle auf Diskette, 5 1/4", 80 Spuren	59,-
11277	Quelle auf Diskette, 3 1/2", 80 Spuren	59,-

## Logik-Simulator

11187	LOGSIM C64	198,-
11246	LOGSIM IBM	248,-

Best.-Nr.	Bezeichnung	Preis DM
<b>Betriebssystem JADOS</b>		
10690	JADOS 38	149,-
10292	JADOS 58	149,-
10993	UPJADOS 38	69,-
10896	UPJADOS 58	69,-

## DATEI II

11061	DATEI II 38	136,-
11062	DATEI II 58	136,-

## Rip the checker

11165	UPSCHACH 38	30,-
11164	UPSCHACH 58	30,-
10873	SCHACH 38	69,-
10874	SCHACH 58	69,-

## Menue-Baukasten

11321	Baukasten 58	29,-
11320	Baukasten 38	29,-

## Finanzbuchhaltung

11319	BUCHHALT 58	90,-
11318	BUCHHALT 38	90,-

Bitte  
Porto  
nicht  
vergessen

**BESTELL-  
POSTKARTE**

**GRAF  
computer**

Graf Elektronik Systeme GmbH  
Postfach 1610

**8960 Kempten**

Bitte  
Porto  
nicht  
vergessen

**BESTELL-  
POSTKARTE**

**GRAF  
computer**

Graf Elektronik Systeme GmbH  
Postfach 1610

**8960 Kempten**

Anschrift:

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

Lieferform:  Nachnahme  Vorkasse  
 Bankeinzug

Bankeinzug: Hiermit ermächtige ich die Firma GES GmbH, den Rechnungsbetrag für die auf dieser Karte angegebenen Bestellungen von meinem Konto:

BLZ \_\_\_\_\_ Konto-Nr. \_\_\_\_\_

Bank: \_\_\_\_\_  
abzubuchen. Falls mein Konto die erforderliche Deckung nicht aufweist, besteht seitens des kontoführenden Kreditinstitutes keine Verpflichtung zur Einlösung.

Datum \_\_\_\_\_ Unterschrift \_\_\_\_\_

Anschrift:

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

Lieferform:  Nachnahme  Vorkasse  
 Bankeinzug

Bankeinzug: Hiermit ermächtige ich die Firma GES GmbH, den Rechnungsbetrag für die auf dieser Karte angegebenen Bestellungen von meinem Konto:

BLZ \_\_\_\_\_ Konto-Nr. \_\_\_\_\_

Bank: \_\_\_\_\_  
abzubuchen. Falls mein Konto die erforderliche Deckung nicht aufweist, besteht seitens des kontoführenden Kreditinstitutes keine Verpflichtung zur Einlösung.

Datum \_\_\_\_\_ Unterschrift \_\_\_\_\_