

LOOP

19

3. JAHRGANG

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

DM 3,50

Aufwind für den NDR-Computer

Betriebssystem OS-9	CPU 8088	Z 80 - Monitor FLOMON CG	680xx Spiel SPEEDY
-------------------------------	--------------------	------------------------------------	------------------------------

Leitartikel

Die CPU8088

Der NDR - Computer wird IBM - kompatibel
19/ 4

CPU Z80

FLOMONCG

Neue Version des FLOMON-Monitors
19/ 9

Turbo Modula 2

Ein brillantes Software-Entwicklungspaket
19/12

IHK Straubing:

Mikroprozessor Kurse mit dem NDR Einsteigerpaket

Beispiel: Kapazitätsmessung
19/17

Nickel-Cadmium Akku Meßgerät mit dem Z80 - Einsteigerpaket

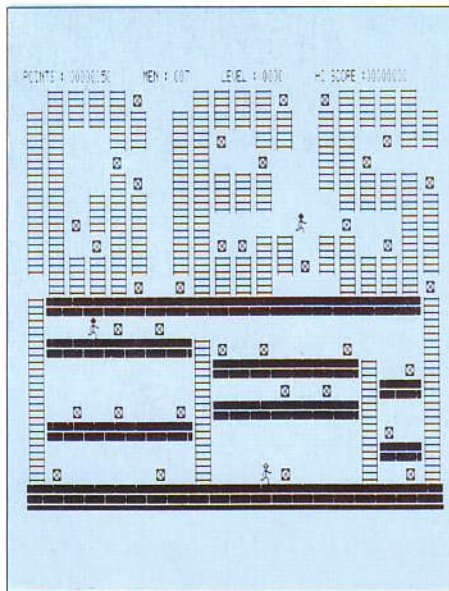
(Teil1 von 2) IST DER AKKU VOLL?
19/22

Kleiner Patch an Microsoft's MBASIC.COM

Drucker parallel zum Bildschirm schalten
19/24

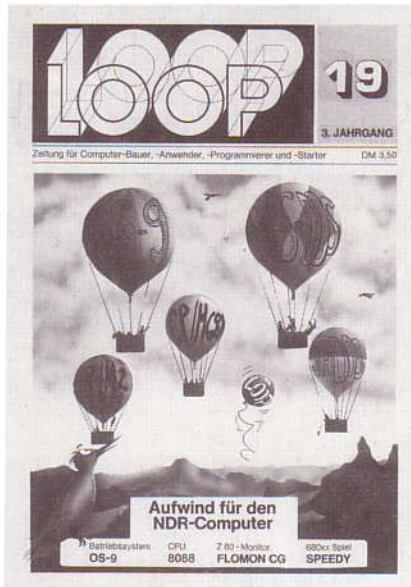
D/A - Wandler ganz stark

Ausgangsspannung auf +10V erhöhen
19/25



SPEEDY - Das neue Computerspiel für den NDR-Computer

Außergewöhnlich viel Spielsstärke ohne notwendiges Betriebssystem für alle CPU 680XX.
Ein gutes Beispiel für die Auflösung und Geschwindigkeit der GDP64HS.



LOOP19 - Titelbild

- Frischer Wind für den NDR-Computer

Entsprechend dem lang ersehnten Wunsch, den NDR-Computer unter dem Betriebssystem MS-DOS betreiben zu können, wurde die neue CPU 8088 entwickelt. Noch spektakulärer und zukunftssträchtiger dürfte das neue OS-9 für alle CPU 680xx sein ...

CPU 680xx

OS-9/6800

Eine neue Dimension für den NDR-Computer
19/ 8

Speedy

Spannendes Computerspiel für CPU 680xx
19/12

NDR - Computer hilft beim Lehrbetrieb

Das Lehrer - Schüler - System LSS
19/27

Eine gefährliche Operation

Patchwork Teil2: FATs und Hauptverzeichnis auf Diskette anlegen
19/28

COMAL-80 für NDR - Klein - Computer unter CP/M 68K

Eine Programmiersprache für Sie getestet
19/31

Korrekturen und Ergänzungen

Probleme des JADOS mit dem Grundprogramm 6.0
19/33
Probleme mit dem Booten beim neuen Grundprogramm
19/33
Sieg für SCHACH V2.9-00 auf JADOS
19/33

mc - modular AT

mc - modular AT Handbuch

Neu überarbeitet
19/ 9

VGA läßt Graphikerherzen höher schlagen

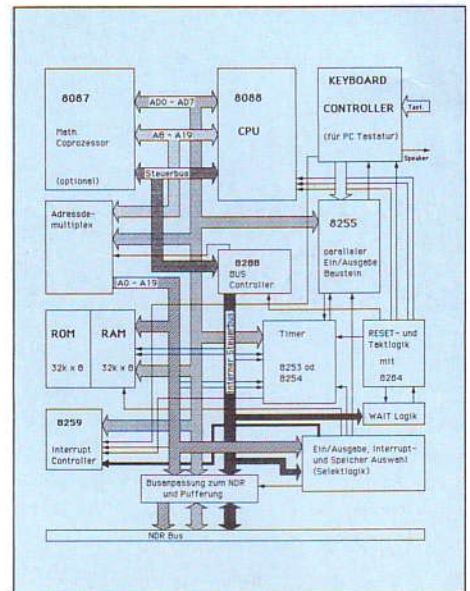
Eine hochauflösende Graphikkarte
19/34

NEC MultiSync II

Ein Monitor setzt sich durch
19/36

Der Handy Scanner

Scannen - einfach und bequem
19/37



Blockschaltbild der CPU8088

MS-DOS mit dem NDR-Computer zu betreiben, das ist der langersehnte Wunsch vieler und wird mit der neuen CPU8088 realisiert.

Alle Peripherie und Speicherarten können weiterhin verwendet werden.

Impressum:

LOOP Zeitung für Computerbauer

Herausgeber: Gerd Graf

Redaktion: Rolf Dieter Klein, Gerd Graf, Christoph Köhler

Gestaltung: Christoph Köhler

Druck: Karl-Heinz Rieder, Kempten

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

Anzeigenpreisliste 4/88

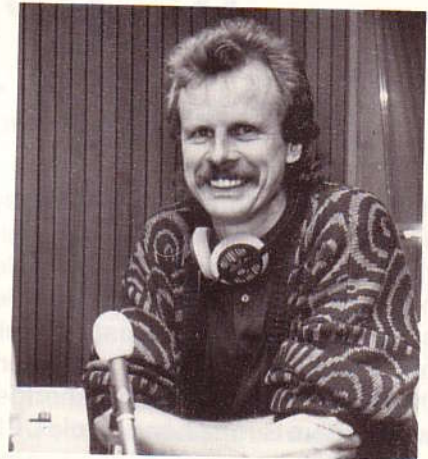
Rubriken

Editorial	19/ 3
Software-Partner stellen sich vor:	
Martin Husemann	19/ 7
Jetzt lieferbar	
OS-9/68000	19/ 8
mc - modular AT Handbuch	19/ 9
FLOMONCG	19/10
Speedy	19/12
Turbo Modula 2	19/14
Information	
Rechner modular im Fernsehen	19/11
Cherry-Tastatur	19/37
Tips und Tricks	
HEBAS: Tips und Tricks, Nr.11	19/19
HEBAS-Tips	19/19
Plottertreiber	19/26
Korrekturen und Ergänzungen	
Berichtigung zu Loop16	19/29
Probleme des JADOS mit dem Grundprogramm 6.0	19/33
Probleme mit dem Booten beim neuen Grundprogramm	19/33
Sieg für SCHACH V2.9-00 auf JADOS	19/33
Kontakte	
	19/38
Impressum	
	19/ 2



Der neue Handy Scanner

Editorial



NDR-Computer im Aufwind ...

zu Recht haben wir den Titel dieser LOOP so gestaltet - denn die Neuigkeiten zum NDR-Computer in dieser LOOP sind schon sensationell.

NDR und IBM

Wieviel ist über dieses Thema schon geschrieben, gesprochen und heiß diskutiert worden - nun ist es endlich soweit. Rolf-Dieter Klein als "Vater" des NDR-Computers hat wieder zugeschlagen und eine Baugruppe - die CPU 8088 - entwickelt, die es wirklich in sich hat.

Das faszinierende und dem modularen Konzept des NDR-Computers entsprechende daran ist: Nur diese neue CPU in den schon vorhandenen NDR-Computer eingesteckt, 256 K Speicher und Floppy bereit: und schon bootet MS-DOS! Natürlich: es ist kein 100 % kompatibler Rechner; die GDP ist langsamer als eine Hercules-Karte, die Graphik (alle IBM-Sonderzeichen werden komplett unterstützt!) ist vertikal nicht nahtlos aneinander zu führen (GDP-Hardware), bei der Verwendung der normalen ASCII-Tastatur muß man schon einige Tastenkombinationen drücken, um z.B. die IBM-Kombinationen STRG + CONTROL + ALT +... zu erhalten, (man kann aber an die CPU 8088 eine IBM-Tastatur anschliessen) aber ... es geht!

Der Anfang ist gemacht, das von RDK geschriebene BIOS liegt im Quellcode vor, ein Bus-Adapter (IBM-Baugruppen in NDR-BUS einstecken) ist in Arbeit, und wir bleiben weiter dran. Kein Grund also, den NDR-Computer - nachdem man gelernt hat, wie ein Rechner funktioniert - auf die Seite zu stellen! Stellen Sie sich unsere Begeisterung vor, als wir zum ersten Mal Turbo-Pascal 4.0 auf dem Bildschirm des NDR-Computers sahen!

NDR und OS/9

OS/9 für die 68000-Benutzer wurde ja schon in der letzten LOOP angekündigt -

hier nun der Erste einer Artikelserie von Volker Wiegand, der OS/9 angepasst hat. Das System läuft schon seit einigen Wochen im Labor - zu unserer vollsten Zufriedenheit. Die große Diskussion im PC-Bereich für und wider OS/2 - hier haben wir ein ausgefeiltes Multi-User UND Multi-Tasking System!

Ein kleiner "Nebeneffekt" von OS/9: Wir haben mittlerweile - über den oben erwähnten Bus-Adapter - eine "normale" IBM-kompatible Festplatte an das NDR-System angeschlossen. Damit ist das Thema Festplatte - zu vernünftigen Preisen - auch noch in diesem Jahr erledigt.

NDR und Z80

Auch die Z80-Seite wurde nicht vernachlässigt - Sie finden in dieser LOOP einen Artikel vom Rüdiger Nahm, dem Entwickler des Z80-Windows, über das von ihm entwickelte FLOMONC: ein "Muß" für alle Z80-Anwender. Auch große Software-Häuser geben dem Z80 gute Chancen: Das in dieser LOOP vorgestellte Turbo-Modula beweist es.

Es geht nach oben mit dem NDR-Computer, der - betrachtet man einmal die vergangenen Jahre - wohl einer der wenigen Computer ist, die diesen Zeitraum überlebt haben. Sie werden noch viel von ihm hören.

Gerd Graf

Rolf Dieter Klein / Klaus Bischof

Die CPU8088

Der NDR-Computer wird IBM-Kompatibel

Schon seit ca. 3 Jahren ist die Nachfrage nach einer IBM Kompatibilität des NDR Computers sehr groß. Jetzt gibt es eine CPU8088 für den NDR-Computer, die diesen in die Reihe der Kompatiblen einreicht. Sie denken nun sicher, daß wohl der gesamte Rechner ersetzt werden muß und nur das Netzteil und einige sonstige Kleinigkeiten weiter verwendet werden können und der Rest nicht mehr verwendet werden kann. Dem ist aber nicht so! Sie müssen nur die bisherige CPU-Karte (SBC3, CPUZ80, CPU68k usw.) aus dem Bus ziehen und die CPU8088 einstecken. Alle NDR Peripherie- und Speicherkarten können verwendet werden.

Das Geheimnis steckt im BIOS

Mit auf der CPU 8088 steckt ein 32k x 8 EPROM mit dem von Rolf-Dieter Klein entwickelten BIOS - Programm. Dieses BIOS-Programm - es gehört zum Lieferumfang - hat es in sich: es "emuliert" mit den NDR-Computer - Baugruppen KEY und GDP64k das Vorhandensein einer IBM - Tastatur und eines IBM- (Text) Bildschirms. Weiter wird das Vorhandensein einer IBM-Floppy-Steuerung - mit auf der FLO2 oder FLO3 - emuliert. Kurz gesagt: nach Austausch der CPU-Karte (und eventueller Speichererweiterung auf mindestens 256k) verhält sich der NDR-Computer wie ein IBM - PC.

Wie kompatibel ?

Das mitgelieferte BIOS unterstützt die IBM monochrom Graphik (mit Graphiksonderezichen) und das 360k IBM Standard PC Diskettenformat. Die erste uns vorliegende BIOS-Version unterstützt im Moment nur zwei 360k Laufwerke. Alle Programme, die (ordentlich) die IBM-BIOS-Calls aufrufen, sollten also funktionieren. Eine beliebte "Unsitte" beim IBM ist es, aus Geschwindigkeitsgründen direkt in den Bildwiederholpeicher des Bildschirm-Controller zu schreiben. Auch hier ist vorgesorgt: Der "Bildwiederholpeicher" befindet sich ebenfalls (als RAM-Bereich) auf der CPU 8088; das BIOS liest diesen Bereich periodisch aus und bildet ihn auf der GDP64k ab.

Wie wird es weitergehen ?

Die CPU 8088 ist der erste, wichtigste Schritt, um den NDR - Computer für die Zukunft zu rüsten. Die nächste Baugruppe wird ein NDR- IBM- Adapter sein, der es ermöglicht, IBM- Baugruppen auf den NDR- Bus zu stecken.

Der nächste Schritt ist die Anpassung einer 20 MByte Festplatte mit dem OMTI-Controller 5520 und der Buskopplung NDR - PC

Bus. Außerdem wird im Moment auch noch keine serielle und parallele Schnittstelle unterstützt. Auch dies wird bei der nächsten BIOS Version enthalten sein. Diese beiden Schnittstellen werden wohl auch über die Buskopplung betrieben, sodaß hier PC-Schnittstellenkarten eingesetzt werden können.

Weiter haben wir vorgesehen, daß über die Buskopplung auch PC-Graphikkarten angesprochen werden können, z.B. Herculeskarte, EGA- oder VGA-Karte usw. Wenn diese Karten dann angeschlossen sind, können dann auch komplexere Programme, die auf die Herculeskarte oder auf EGA bzw. VGA zugreifen, betrieben werden.

Nun aber zur CPU8088

Die Schaltung der CPU8088 ist relativ komplex und dementsprechend undurchsichtig. Aus diesem Grund ist ein übersichtliches Blockschaltbild unerlässlich. Selbst das Blockschaltbild ist noch relativ voll, aber es läßt sich leider nicht mehr abspecken,

ohne dabei ins triviale und nichtssagende zu verfallen.

Die 8088 CPU ist das Herz der Baugruppe und des Rechners. Sie ist entweder von INTEL und ist mit 8088-1 (10 MHz) bezeichnet, oder sie ist von NEC und trägt die Aufschrift V20-10MHz. Die CPU steuert sämtlich Vorgänge im Rechner (Verwaltung der Speicher und Ein/Ausgabe Einheiten, Ablauf von Programmen, Rechenarbeit...).

Der Prozessor 8088 ist ein 8 Bit Prozessor mit interner 16 Bit Architektur. Der Prozessor ist softwaremäßig kompatibel zum 8086, und kann 1 MByte direkt adressieren.

Der 8087 ist ein Mathematik Coprozessor, der normalerweise nicht verwendet wird. Deshalb ist dieser im Lieferumfang auch nicht enthalten. Bei Softwareproblemen mit vielen math. Gleichungen bringt der 8087 allerdings einen großen Zeitgewinn, da der Prozessor math. Funktionen direkt berechnet, d.h. dem Coprozessor wird nur ein Befehl übergeben und anschließend das Ergebnis abgeholt.

Der Block Adressdemultiplex macht vom Prinzip her nichts anderes, als die gemultiplexten Adress- und Datenleitung AD0 bis AD7 zu trennen. Am Ausgang sind dann nur noch die Adressleitungen A0 bis A7 und zusätzlich A8 bis A19, die zusätzlich noch gepuffert werden. Die Steuerung dieses "Adressdemultiplexers" wird von dem Signal ALE (Adress Latch Enable) gesteuert, das der Bus-Controller 8288 erzeugt.

Der BUS-Controller 8288 erzeugt aus den Prozessorsignalen S0, S1 und S2, die für den PC-Bus benötigten Signale IOR (Ein/Ausgabe lesen), IOW (Ein/Ausgabe schreiben), MEMR (Speicher lesen), MEMW (Speicher schreiben), DT/R (Daten senden oder empfangen) DEN (Daten

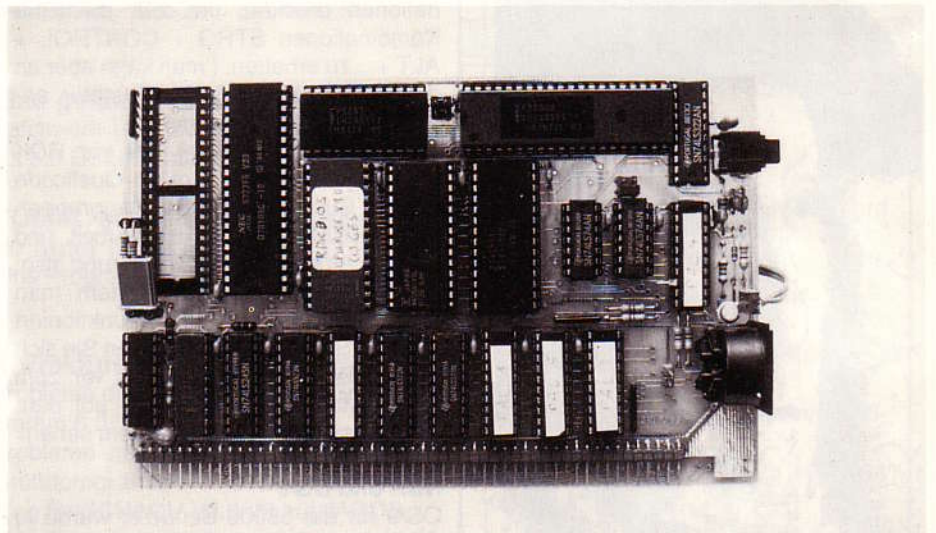


Bild 1: Foto des Prototypes der CPU 8088

Freigabe), ALE (Adressfreigabe und zwischenspeichern) und INTA (Quittieren eines Interrupts). Diese Signale werden zum Ablauf von Speicher und Ein/Ausgabe Zugriffen, sowie zur Interruptabwicklung benötigt.

An Speichern befindet sich auf der Baugruppe maximal 64kByte, die sich in 32k ROM und 32k RAM aufsplitten. Der ROM-Bereich wird in der Regel für das BIOS genutzt. Wer die Karte aber als Single Board Computer für irgendwelche Steuerungen nutzen will, kann hier natürlich auch sein eigenes EPROM stecken. Der RAM Bereich (statisches RAM 32k x 8), dient lediglich als Arbeitsbereich für das ROM. Diese beiden Speicher werden beim Booten des Betriebssystems in den Arbeitsspeicher (ROA256/1 oder RAM256) geladen und ausgeblendet.

Der Interrupt-Controller 8259 ist für die

uhr benutzt. Dabei wird ein Takt von 1,19 MHz angelegt, der Timer macht dann nichts anderes, als die Taktzyklen zu zählen und in bestimmten Zeitabständen einen Interrupt zu senden (jede Sekunde). Dabei wird dann softwaremäßig bei jedem Interrupt der Sekundenzähler um eins erhöht usw. Der zweite Timer wird zur Synchronisation der seriellen Tastaturdaten verwendet.

Reset- und Taktlogik

Die RESET und Taktlogik erzeugt einmal den Systemtakt für den Prozessor 8088 und den Coprozessor 8087. Dabei wird die Quarzfrequenz von 28,62 MHz durch drei geteilt, also 9,54 MHz. Außerdem wird über diesen Baustein auch der Takt für den Timer erzeugt. Dabei wird die Quarzfrequenz vom 8284 erst durch 3, dann durch 2 und anschließend durch zwei Flip-Flops

Wait - Logik

Dabei wären wir bereits bei der WAIT-Logik angelangt. Der Ausdruck WAIT hat bei den Computer Anwendern immer einen negativen Beigeschmack, da damit immer eine Verlangsamung des Rechners gemeint ist. Aus systemspezifischen Gründen sind aber WAITs bei bestimmten Zugriffen unerlässlich, um schnelle teure Bausteine zu sparen. Natürlich versucht man WAITs nur dort einzubauen, wo sie am wenigsten Zeit kosten. Dies ist bei den Ein/Ausgaben der Fall, da auf diese Einheiten im Vergleich zu Speicherzugriffen nur sehr selten zugegriffen wird. WAIT-States werden in unserer Schaltung bei Zugriffen auf den Timer, auf den Interrupt Baustein und auf den parallelen Ein/Ausgabe Baustein eingefügt. Außerdem natürlich auch wenn vom NDR-Bus ein WAIT-Signal anliegt.

Technische Daten

- CPU8088-1 = 8088 CPU mit 10 MHz (auch NEC V20)
- Sockel für 8087-1 vorgesehen
- 3 Timer 8253
- Interruptcontroller 8259
- Taktgenerator und RESET-Baustein 8284-1 (10 MHz)
- Buscontroller 8288
- paralleler Schnittstellen Baustein 8255
- Keyboard Controller für PC Tastatur
- Speaker Ausgang (8 Ohm Lautsprecher)
- WAIT Logik
- 32k ROM (BIOS)
- 32k RAM
- BIOS unterstützt IBM monochrom Zeichensatz (pseudographik)
- Folgende Programme wurden bereits getestet:
 - Wordstar
 - Turbo Pascal V4.0
 - DBASE II
 - Norton Utilities

F&A

Logiksimulator

- BUS: NDR-Bus
- Stromaufnahme +5V: 960 mA
- Leiterplattenformat: 100 x 160

Sämtliche Karten des NDR-Computers können verwendet werden:

- GDP64K
- KEY mit paralleler Tastatur
- ROA64, RAM64/256, ROA256/1M
- FLO3 mit TEAC-Laufwerken (FD55F..)
-sämtliche I/O Karten

Benchmarks: Nortonfaktor 2,4

Landmark Test: CPU Speed im Vergleich z. AT: 2,3 MHz

Landmark Faktor im Vergleich z. IBM PC (4,77 MHz): 1,1

Abwicklung der Interrupts verantwortlich. Dieser Baustein stellt 8 Interrupt Eingänge zur Verfügung. In unserer Schaltung sind nur 5 belegt: Zeitgeber, Tastatur, Parallel Drucker und 2 Asynchron Adapter. Wird nun ein Interrupt von einem der Geräte ausgelöst, so gelangt der Interrupt zum 8259. Dann aktiviert er den Prozessorinterrupt und wartet auf das Quittierungssignal (INTA). Ist dieses aktiv, so wird der Interrupt Vektor eingelesen und in die Interrupt Routine verzweigt. Außerdem bietet der Baustein noch einige Extras, wie Prioritätsbehandlung, Kaskadierbarkeit usw.

Der Timer

Der Timer 8253 oder 8254 (schnelle Version) beinhaltet 3 Timer, wobei bei unserer Schaltung nur zwei benutzt sind. Der erste Timer wird ausschließlich für die System-

nochmal zweimal durch 2 geteilt. Dadurch erhält man den Timertakt von 1,1925 MHz, der für die Systemuhr verwendet wird.

Zum zweiten erzeugt dieser Baustein ein RESET Signal, aus einem analog erzeugten RESET (RC-Glied für Power on Reset und RESET-Taster). Dieses analog ansteigende Signal wird an den Eingang RES (mit Schnitt-Trigger) angelegt und auf dem Ausgang RESET mit einer bestimmten Periodendauer an der Prozessor weitergeleitet. Zum dritten enthält der Baustein eine Logik zum Synchronisieren des READY-Signals. Dabei können zwei READY-Signale von außen angelegt werden, die dann mit dem Systemclock synchronisiert an den Prozessor weitergeleitet werden. Hier wird nur ein READY-Signal verwendet. Dieses READY-Signal wird von der WAIT-Logik erzeugt.

Die Tastatur - Schnittstelle

Der parallele Ein/Ausgabe Baustein 8255 dient nur zum Einlesen der Tastatur und zur Steuerung des Ablaufs des Keyboard Controllers, sowie zur Datenausgabe an den Lautsprecher.

Der Keyboard Controller auf der CPU8088 ist so ausgelegt, daß am NDR-Computer auch eine PC-Tastatur (auch MF-II Tastaturen, die auf PC oder XT Modus umschaltbar sind) angeschlossen werden kann. Dabei muß aber diese PC-Tastatur nicht verwendet werden, denn es funktioniert auch die KEY und daran die Tastaturen TAST1, TAST2 und TAST3, aber eine PC-Tastatur bietet mehr Bedienungskomfort. Vor allem die Darstellung der Sonderzeichen und Betätigen der Funktionstasten verlangt bei den bisherigen NDR-Tastaturen doch einige Klimmzüge, aber es kön-

nen auch mit TAST1, TAST2 und TAST3 alle möglichen Zeichen dargestellt werden. Nun zurück zum Tastatur-Controller. Dieser macht nichts anderes, als die seriell ankommenden Zeichen in ein paralleles Format zu wandeln. Diese Zeichen werden dann parallel an den 8255 weitergegeben und eingelesen. Der Ablauf beim Einlesen eines Zeichens wird über Interrupt gesteuert; d.h. kommt ein Zeichen von der Tastatur an, wird ein Interrupt ausgelöst, das Zeichen eingelesen und in einem Puffer abgelegt (Softwarepuffer). Sollte keine PC-Tastatur angeschlossen sein, sondern eine NDR-Tastatur, so verfügt auch diese über den Softwarepuffer, d.h. TAST1, TAST2 und TAST3 haben bei Betrieb mit MS-DOS einen Zeichenpuffer (Größe des Zeichenpuffers im DOS definiert). Die Selektlogik wählt, wie der Name schon sagt, die einzelnen Speicher oder Ein/Ausgabebausteine aus, d.h. liegt z.B. der ROM-Speicher von Adresse 0 bis 7FFFh, so wird, wenn eine Adresse in diesem Adressraum angesprochen wird dieser Baustein aktiviert (Chip Select-Signal). Ebenso verhält es sich bei Portadressen von Ein/Ausgabe Bausteinen. Diese Chip

Select Signale werden von der Selektlogik erzeugt. Die Busanpassung zum NDR-Bus besteht im wesentlichen aus zwei Bausteinen. Zum einen werden die Signale -IOR, -IOW, -MEMR, -MEMW auf die NDR-Bussignale -RD, -WR, -IORQ, -MREQ umgelegt und das RESET-Signal invertiert. Zum andern werden die Daten über einen bidirektionalen Bustreiber gepuffert und nur bei Zugriff auf externe Baugruppen (Speicher, Ein/Ausgabe) freigegeben. Damit wäre der grobe Aufbau der CPU8088 anhand des Blockschaltbildes erläutert. Warenzeichen: IBM ist ein eingetragenes Waren Zeichen der International Business Machines

Nikolaus Bischof
GES

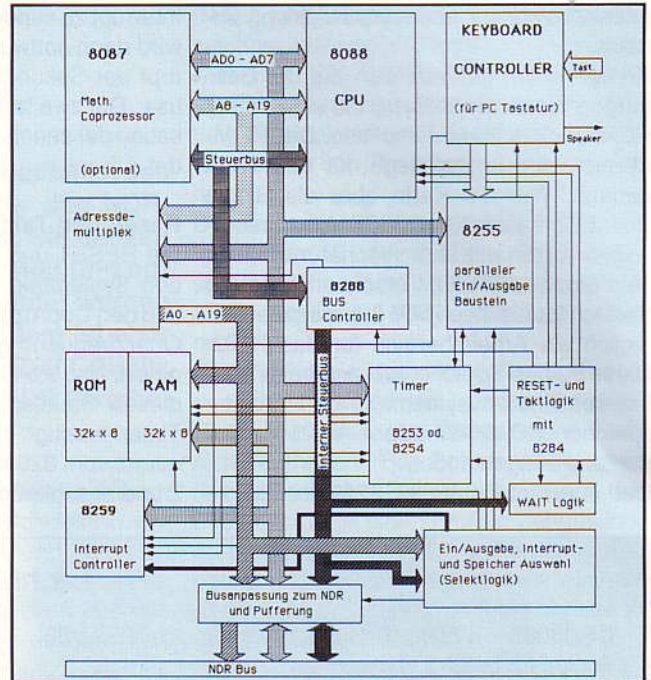


Bild 1: Blockschaltbild der CPU8088

- Neu -

Honorar für Ihre LOOP-Beiträge

Durch die Umstrukturierung bei der Erstellung der LOOP wollen wir versuchen, diese sowohl inhaltlich als auch optisch weiter auszubauen. Außerdem soll baldmöglichst wieder unsere "Erscheinungsfrequenz" von 2 Monaten erreicht werden. Um dieses Ziel zu erreichen, sind einige wichtige Punkte abzuändern: Um noch mehr Artikel von Ihnen zu erhalten, haben wir uns entschlossen, ab jetzt Honorare für Ihre Beiträge zu bezahlen. Allerdings müssen die Artikel in einer bestimmten Form bei uns vorliegen. Alle diejenigen unter Ihnen, die irgendwann durch eigene Beiträge die LOOP mitgestalten möchten, sollten diesen Artikel genau durchlesen.

Wie schon in der LOOP 18 angekündigt, wird jetzt die LOOP mit dem Programm "Page Maker" auf dem Bildschirm des modular AT entworfen, dann von der Druckerei weiterverarbeitet. Daß unsere freien LOOP-Autoren nicht nur vom "guten Namen" leben müssen und außerdem zukünftig noch mehr motiviert werden, bezahlen wir ab jetzt für jeden abgedruckten Artikel folgendes Honorar:

Höhe des Honorares:
pro abgedruckte Seite Text oder Bilder: DM 50.—

pro abgedruckte Seite Listing: DM 40.—

Wie Sie sehen, lassen wir uns Ihre Ideen und Arbeit einiges kosten. Lassen Sie sich aber nicht durch die Höhe des Honorar dazu verführen, allzulange Artikel zu

schreiben. Sie sollten davon ausgehen, daß die Wahrscheinlichkeit des Abdrucks mit der Länge des Schriftstückes eher sinkt. Zu der Übergabe der Artikel gilt grundsätzlich folgendes:

Welches Textverarbeitungssystem ?

Für einen CP/M - Rechner bietet sich z.B. der bewährte WORDSTAR und für einen AT am besten das MS-WORD an. Es wäre optimal, einen automatischen Zeilenumbruch einzusetzen. (Return nur bei der Beendigung des Absatz) Ansonsten kann der Artikel mit jedem beliebigen Editor erstellt werden.

Gestaltung des Schriftstückes

Auf den Namen des Autoren sollte die Überschrift und ein zusammenfassender Vorspann folgen. Im eigentlichen Text soll-

ten Zwischen-Überschriften eingesetzt werden. Die Tabellen und Bilder sollten separat geliefert werden.

Auslieferungsform:

Alle eingesandten Artikel müssen auf Diskette sein. Außerdem sollte ein Ausdruck des Schriftstückes beigelegt werden. Die Listings müssen mit einem guten Farbband erstellt werden, da diese nicht mehr eingegeben werden (Fehlerquelle). Außerdem muß angegeben werden, welches Textverarbeitungssystem verwendet wurde.

Rechtliche Vereinbarung:

Sie müssen uns bestätigen, daß sowohl der Inhalt des Artikels von Ihnen verfasst wurde und daß Sie auch mit einer Veröffentlichung des selben einverstanden sind. Auf etwaige Bezugsquellen muß hingewiesen werden. Sollten Sie wesentliche Teile einem bereits gedruckten Schriftstück entnommen haben, können wir ggf. eine Freigabe vom Urheber erfragen. Selbstverständlich werden Ihre Artikel weiterhin grundsätzlich mit Ihrem Namen und Anschrift veröffentlicht.

Für alle von Ihnen, die glauben, wohl keine interessanten Artikel schreiben zu können, haben wir nur einen Trost:

Bis auf den EINEN im deutschsprachigen Raum, der vermutlich am wenigsten Erfahrung mit Computer hat, können ALLE anderen ihre Erfahrung an Andere weiterleiten...

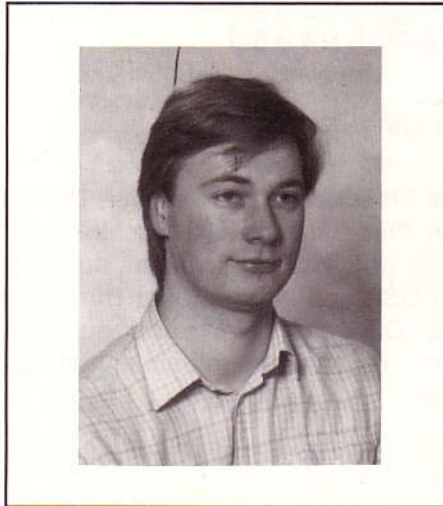
Christoph Köhler ,GES

Software-Partner stellen sich vor

Persönliche Daten

Martin Husemann
Am Ballerbrink 5
4800 Bielefeld 18

23 Jahre



Geboren wurde ich 1965, knapp zu spät für den 1. April, am 13. (das war ein Dienstag sagt mein Datumsprogramm für das Einsteigerpaket). Von 1975 bis zum Abitur 1984 besuchte ich das Gymnasium Heepen, wo ich auch den ersten Computer-Kontakt hatte. Damals, ich war im 9. Schuljahr, gab es dort noch keinen Informatik-Unterricht, es wurde lediglich eine "Informatik AG" ausserhalb der Unterrichtszeit angeboten. Aus reiner Neugier nahm ich daran teil. Zunächst lernte ich TI-59 Taschenrechner zu programmieren, dann beschäftigte ich mich kurz mit einem PDP-8 Minicomputer (12 Bit Worte, 4 k Worte Ram, Magnetkernspeicher, IO über Teletype, externes Speichermedium: Lochstreifen), der in FOCAL programmiert wurde, eine Super-Mini-Sparversion von FORTRAN. Damit konnte man immer noch nicht sehr viel machen, aber nach kurzer Zeit schaffte die Schule einen "richtigen" Computer an, einen Commodore CBM 8032 mit 31 kByte freiem Speicherplatz und BASIC-Interpreter, dazu ein Doppeldiskettenlaufwerk und einen Matrixdrucker. Damit endlich allen "Hardware" Begrenzungen entflohen, ging es richtig los und nach einem halben Jahr meldeten die ersten Programme "Memory overflow". Durch geschicktes Arbeiten auf Diskette gelang es mir und meinen Mitstreitern dennoch, ein komplettes Verwaltungsprogramm für die Gymnasiale Oberstufe zu schreiben. In der überarbeiteten Version "OSS II" (Oberstufenservice 2) ist dieses Programmpaket immer noch im Einsatz (aber natürlich inzwischen völlig veraltet). Meine BASIC-Erfahrung beschränkt sich also nicht auf 100 oder 1000 Programmzeilen. Damit trotzdem

noch etwas aus mir werden konnte, und da ich inzwischen an alle BASIC-Einschränkungen gestoßen war, lernte ich Assembler (6502) und PASCAL.

1984 begann ich, trotz großer (Selbst-) Zweifel an meinen Löffähigkeiten, einen NKC zu bauen. Die damalige Marktsituation (soweit ich sie erkannte) und insbesondere meine finanzielle Situation ließen als Alternativen nur einen C 64 oder einen Apple 2 Nachbau zu, und da war natürlich der residente PASCAL/S Compiler des 68008er NKC entscheidend. Auch die GDP 64K fand ich faszinierend (find' ich auch immer noch, nur nicht mehr 'state of the art').

Inzwischen habe ich weiter aufgerüstet, mein System besteht jetzt aus einem 68008 mit 832 kByte RAM, 2 Floppylaufwerken, Atari-Maus und NEC P7 Drucker (24 Nadeln). Ich benutzte CP/M-68-K und (gelegentlich) JADOS Programme entwickle ich ausschließlich mit dem p1 Modula-2. In dieser Konfiguration ist das System jetzt ein halbes Jahr alt, und natürlich bereits zu klein. Ich hoffe demnächst eine Festplatte zu bekommen und plane den Umstieg auf den 68020.

Seit dem Wintersemester 1985 studiere ich Informatik an der UNI-GH Paderborn und unterrichte nebenbei an der Volkshochschule Bielefeld. Ich programmiere (bisher) grundsätzlich nicht kommerziell. Alles, was aus meiner Feder stammt, habe ich irgendwann selber gebraucht. Oft ist es mir aber auch einfach zu umständlich, die Programme in den entsprechend fehlerarmen Zustand, der für eine Veröffentlichung notwendig ist, zu versetzen. Ich habe z. B. einen kleinen "Pascal nach Modula" Kon-

verter, den ich nur noch so selten benutze, daß es sich nicht lohnt, ihn endgültig fertigzustellen.

Falls jemand Interesse hat: formatierte 5 1/4" Diskette genügt...

Ich würde gerne viel mehr Programm- und Erfahrungsaustausch betreiben, möglichst nicht nur per Post - aber dem steht wohl die relativ geringe NKC Benutzer Dichte im Weg.

Anm. d. Redaktion:

Von wegen "relativ gering" !!

Hier müssen wir ganz entschieden widersprechen!

Vielleicht noch ein paar Tips für Einsteiger:

- Wer Programmieren lernen möchte, sollte mit PASCAL oder Modula-2 anfangen. Auf keinen Fall BASIC oder Assembler als erste Sprache lernen (ich glaube nicht, daß jemand nachdem er PASCAL gelernt hat noch BASIC lernen möchte), das verdirbt den Charakter (genauer: den Programmierstil). (Ausnahmen, wie ich, bestätigen natürlich nur die Regel...)

- Wer Hardware verstehen will, kommt an Assembler nicht vorbei. Schöne Möglichkeit: Einsteigerpaket und dann "drauflosbasteln". Wer Hard- und Software verstehen will, sollte erst Programmieren lernen (siehe oben) - d.h. gleich von Anfang an z.B. PASCAL/S anschaffen, das ist natürlich dann etwas teurer, meist erweitert man aber nach kurzer Zeit sowieso, warum also nicht gleich?

- Wer später einmal Informatik studieren möchte, sollte vor der UNI jeden praktischen Kontakt mit Computern meiden. (Oder umgekehrt: wer vor allem praktisch am Computer arbeiten möchte, sollte nicht gerade Informatik studieren.)

Soweit man bisheriges Computer-Leben, es ist doch etwas länger als geplant geworden.

Meine Arbeit am Einsteigerpaket läuft derzeit auf Hochtouren.

Anm. d. Redaktion:

Herr Husemann entwickelte das Programm MCOPIE und eigene kleine TOOL-Programme für das Einsteigerpaket.

Volker Wiegand

OS-9/68000

Eine neue Dimension für den NDR-Computer

Teil 1: Einführung in OS-9

Teil 2: Das Modulkonzept

Teil 3: Übersicht der Dienstprogramme

Teil 4: Wie schreibt man einen Treiber ?

“OS-9”, das ist in der letzten Zeit zu einer Art Zauberwort geworden. Dieses Betriebssystem hat sich einen festen Platz in der Welt der Motorola-Prozessoren erobert und ist mittlerweile dem Ruf entwachsen, nur im technisch-wissenschaftlichen Bereich einsetzbar zu sein. Längst findet man OS-9 in Bereichen der Bürokommunikation, aber auch Betreiber von Mailboxen haben schon die Vorzüge von OS-9 erkannt. In Verbindung mit dem NDR-Computer schickt sich OS-9 nun an, auch im Hobby- sowie im Lehrbereich seine Qualitäten unter Beweis zu stellen, ist es doch dieser Hardware geradezu auf den Leib geschnitten.

Die Geschichte von OS-9 beginnt Ende der siebziger Jahre, als Motorola den 6809 vorstellte, einen Prozessor, bei dem bereits bei der Entwicklung ein Team von Software-spezialisten mitgewirkt hat, um eine optimale Leistungsfähigkeit des Produktes zu garantieren. Dabei entstand als eine Gemeinschaftsentwicklung der Firmen Motorola und Microware, eines aufstrebenden Softwarehauses, zunächst die Programmiersprache Basic09, bei der erstmals Strategien wie “reentrant code” und “position independent code” verwirklicht wurden. Aufgrund der größeren Erfahrung im Schreiben von Systemsoftware übernahm dann Microware allein die Entwicklung des unterliegenden Betriebssystems OS-9 (Operating System for the 6809). Dieses Betriebssystem hatte Eigenschaften, die bis dahin nur Großrechner aufzuweisen hatten. Viele Ideen von OS-9 sowie die Struktur selbst stammen schließlich vom Betriebssystem UNIX ab.

Erstmals wurden auf Computern, die sogar für Hobbyanwender erschwinglich waren, Eigenschaften wie Multitasking, Multiusing oder Realtime Processing (wird unten näher erläutert) realisiert. Seitdem werden viele dieser Computer im Bereich der Prozeßsteuerung eingesetzt. Allerdings sind solche Geräte meist sehr einseitig ausgelegt und werden für die spezielle Anwendung hergestellt. Das Fehlen von Standardgeräten mit dem 6809 war auch für OS9 eine nicht zu nehmende Hürde. Der große Durchbruch kam dann, als mit den Prozessoren der Reihe 68000 eine so leistungsfähige Hardwaregrundlage gegeben war, daß nun wirklich mehrere Tasks oder Benutzer gleichzeitig unterstützt werden konnten.

Was ist nun das Besondere an OS-9 und was hat es mit Begriffen wie Multitasking, Multiusing oder Realtime auf sich? Nun, zunächst einmal: OS-9 ist nicht “kompati-

bel”. Diese profane Feststellung hat allerdings weitreichende Folgen. Ohne die Vorzüge von MS-DOS in irgendeiner Weise schmälern zu wollen (unser OS-9 kann MS-DOS Disketten lesen und schreiben...), hat OS-9 doch nicht den Ballast vieler Vorgänger mit sich zu tragen. OS-9 wurde entwickelt, ohne auf solche Vorgänger wie etwa CP/M achten zu müssen. Dasselbe gilt ja übrigens auch für den Prozessor 68000; eine bemerkenswerte Parallele.

Ein Betriebssystem hat die Aufgabe, die Ressourcen des Systems zu verwalten. Dazu gehören CPU-Zeit, Speicherplatz oder Ein-/Ausgabe-Kanäle. Man unterscheidet nach der Anzahl Prozesse oder “tasks”, die ein Computer gleichzeitig abarbeiten kann, nach Single- oder Multi-Tasking Systemen. Ein typisches Single-Tasking System ist CP/M68k. OS-9 hingegen kann bis zu 65535 Prozesse parallel verwalten. Selbstverständlich kann eine CPU zu einer gegebenen Zeit nur an einer Aufgabe arbeiten. Die Taskverwaltung, der “scheduler”, ist ein Betriebssystemteil, der die Vergabe von Prozessorzeit über eine Zeitscheibe, den “tick”, meist eingestellt auf 10 ms, regelt. Dabei können Prozesse mit unterschiedlicher Priorität ausgestattet werden. Tasks mit höherer Priorität werden dann öfter aufgerufen und scheinen schneller abzulaufen. Durch die Art der Vergabe ist jedoch sichergestellt, daß alle Tasks eine Mindestzeit erhalten.

Daneben können Tasks noch unterschiedliche Zustände annehmen. Alle aktiven Tasks etwa befinden sich in der Warteschlange oder “queue” der ablaufbereiten Prozesse und werden vom Scheduler zur CPU zugelassen. Eine Task aber z.B., die auf eine Benutzereingabe wartet, würde die CPU nur unnötig belasten. Daher wird sie in die Queue der schlafenden Prozesse eingereiht und gar nicht vom Scheduler be-

rücksichtigt. Wie wird nun ein schlafender Prozess geweckt? Hier kommt eine weitere Eigenschaft von OS-9 zum Tragen, nämlich die Echtzeitverarbeitung oder “real time processing”. Dies bedeutet, daß auf Ereignisse von außen unmittelbar reagiert werden kann. Dazu muß ein leistungsfähiges Unterbrechungssystem verfügbar sein. Fast alle geringfügigen Änderungen, die am NDR-Computer auszuführen sind, beziehen sich übrigens hierauf. Ein Interrupt also, etwa von einem Terminal, das ein Zeichen an die Zentraleinheit gesendet hat, weckt einen schlafenden Prozess. Dies geschieht über ein Signal, das der Eingabeprozess an die schlafende Task sendet.

Kann ein Computer nun schon mehrere Tasks quasi-parallel verwalten, so liegt es natürlich nahe, das auszunutzen, um mehrere Benutzer oder “user” zu unterstützen. Zu einem echten Mehrbenutzer- oder “multi user” System gehört aber doch wesentlich mehr.

Neben Datenschutz und Datensicherheit muß hier unbedingt sichergestellt sein, daß ein Benutzer nicht in der Lage ist, das ganze System lahmzulegen, etwa durch Austesten neuer Programme. Daher gibt es im OS-9 eine Trennung in System- und Benutzerbereiche - der 68000 unterstützt das ja schon hardwaremäßig,

Nach den Eigenschaften von OS-9 kommen wir nun zum Aufbau. Auch hier hält Microware einige Leckerbissen bereit, die die Beschäftigung mit OS-9 zu einem Vergnügen machen, an denen man aber auch sehr viel über durchdachten Aufbau und optimale Strukturierung lernen kann. Einer der Hauptgründe für den Erfolg von OS-9 ist sein modularer Aufbau. Durch Hinzufügen oder Wegnehmen von Modulen sogar während der Laufzeit kann das System ungemein flexibel gehandhabt werden. Was in letzter Zeit etwa durch Modula-

bekannt gemacht wurde, ist seit einem Jahrzehnt die Grundlage und das Konzept von OS-9!

Statt grauer Theorie wollen wir uns das Modulkonzept an einem Beispiel klarmachen. Wir gehen aus von einem Anwenderprogramm, sagen wir einem Texteditor. Durch den Aufruf wird dieses Programm von der Diskette geladen. Es gibt nun eine strikte Trennung von Programm- und Datenbereich. Beim Laden wird also dem Programmkopf entnommen, wieviel Datenbereich das Programm benötigt und dieser Platz wird dem Programm zur Verfügung gestellt. Das Programm erhält nur einen Zeiger auf den Anfang und wickelt alle Zugriffe indiziert ab. Wenn nun ein zweiter Benutzer unseren Editor benutzen will, so ist ja der Programmcode schon im Speicher. Also wird nur noch ein Datenbereich angelegt und der Benutzer kann sofort mit der Arbeit beginnen, ohne Ladezeiten zu haben. Wird ein Programm von niemandem mehr benutzt, so wird sein Speicherbereich dem System zurückgegeben, damit er neu vergeben werden kann. Würde zurückgegeben, wenn da nicht noch ein kleiner Trick wäre... Da man einen Editor meist mehrmals benutzt, wäre es schade, wenn er jedesmal nach Beenden und Neuaufruf wieder neu geladen würde. Also wird er zunächst im Speicher gehalten, jedoch als Schatten, als "ghost". Wird nun das Programm aufgerufen, so ist es sofort da, aber für den Fall, daß das System den Speicherplatz anderweitig braucht, wird das Programm selbständig entfernt und der Speicherplatz genutzt.

Unser Editor muß natürlich mit dem Anwender kommunizieren können. Dazu wird ihm vom System ein Pfad zugeteilt, der ihn

mit dem Terminal verbindet. Dieser Pfad wird durch einen Betriebssystemaufruf, einen "function call", geöffnet, bei dem als Parameter der Name des Terminals, sagen wir "t1", übergeben wird. Der Betriebssystemkern, der "kernel", ist ein Modul, das solche Calls entgegennimmt. Dieses Modul beinhaltet daneben noch den Scheduler, die Task-, die Queue-, sowie die Interruptverwaltung. Der Kernel also sucht ein Modul mit dem Namen "t1", welches ein "device descriptor" sein muß. Das ist eine Tabelle, die das Gerät "t1" beschreibt. Diese Beschreibung umfaßt neben der Basisadresse, Anzahl der Zeilen und Spalten sowie Zeichen für Abbruch oder Rückschritt etc. auch zwei Einträge, die den Kernel besonders interessieren. Da ist zunächst der Verweis auf den "device driver". Das ist eine Sammlung von Unterprogrammen, die die Hardware, also die serielle Schnittstelle, direkt bedienen. Ein solches Modul kann alle seriellen Schnittstellen im System bedienen, denn es ist, wie unser Editor, reentrant. Weiterhin findet sich hier ein Verweis auf ein Modul, welches die Ein-/Ausgaben des Programmes entgegennimmt und mit Hilfe des Drivers weiterleitet. Dieses Modul ist in der Lage, auch komplexe Funktionen wie Auto-Linefeed oder Baudratenumschaltung zu handhaben. Es nennt sich daher "manager", und da es für alle zeichenorientierten Geräte wie Terminals, Drucker oder Modems zuständig ist, heißt es "sequential character file manager" oder "scf". Daneben gibt es noch Manager für Disketten und Platten ("random block file manager", "rbf"), für Pipelines, das sind Warteschlangen zwischen Prozessen, ("pipe manager", "pipe") sowie für Tapestreamer und Netzwerke. Beim Öffnen eines Pfades wird vom Manager an das Programm eine Kanalnummer

übergeben. Diese Kanalnummer genügt in Zukunft, um eindeutig ein Zeichen zu einem bestimmten Gerät zu senden, was natürlich wiederum über einen Call, diesmal mit dem Zeichen und der Kanalnummer als Parameter, geschieht. Die Modularität geht hier so weit, daß das Programm sich nicht darum kümmern muß, wohin das Zeichen geht, und daß eine Ausgabeumleitung z.B. auf die Diskette ohne jede Mühe funktioniert.

Die Disketten und Platten sind übrigens mit einem hierarchischen Dateisystem ausgestattet und dadurch sehr übersichtlich. Außerdem muß man sich nicht um Partitionierung o.ä. kümmern, und die Obergrenze für ein logisches Laufwerk liegt bei ca. 4,3 GByte. Dieses Dateisystem hilft auch beim Softwareschutz, kann man doch sein Hauptverzeichnis und alle Unterverzeichnisse sehr einfach für den Rest der Welt unzugänglich machen.

Modularität heißt also das Zauberwort. Und Modularität ist genau das, was der NDR-Computer bietet. Darum ist die Verbindung von OS-9 und NDR-Computer so interessant. Und darum werden wir in Zukunft noch sehr viel über dieses Betriebssystem sagen und schreiben. Zunächst werden wir mit technischen Details und den geringfügigen Änderungen am NDR-Computer beginnen. Daneben werden wir über interessante Anwendungen berichten, in denen sich dieses Gespann schon bewährt hat. Wir versprechen, es wird interessant.

Volker Wiegand
Hauptstraße 99
D-6454 Bruchköbel

mc-modular AT Handbuch komplett überarbeitet

Nachdem der mc-modular AT doch eine gewaltige Wandlung durchgemacht hat, vor allem was das Gehäuse und die techn. Ausstattung angeht (80386, Above Board usw.), wurde eine neue Ausgabe des mc-modular AT Handbuches nötig. Vor allem was die Übersichtlichkeit betrifft, haben wir uns viel Mühe gemacht, um möglichst viel Information optimal übersichtlich darzustellen.

Das neue mc-modular AT Handbuch ist in acht Kapitel unterteilt:

1. Einführung, 2. Funktion mc-modular AT, 3. Aufbau Schritt für Schritt, 4. Software Installation, 5. Kleine DOS Einführung, 6. Diagnose, 7. Standortwechsel und 8. An-

hänge. In den Anhängen sind der Betrieb und der Anschluß von Baugruppen bzw. Zubehör wie SER, Maus, Streamer, Above Board, EGA, VGA und externes 3 1/2" Laufwerk beschrieben. Außerdem sind Steckerbelegungen, Systemspeicherbe-

legung, Interruptbelegung, DMA Kanäle, Timerbelegung usw. und Tastaturbelegung dargestellt. Ein Anhang mit 10 Seiten ist allein der 80386 Baugruppe gewidmet; hier wird die Installation und erste Funktionstests beschrieben. Ein weiterer Anhang ist für das TOWER Gehäuse reserviert. Die letzten acht Seiten des Handbuches enthalten ein umfangreiches Stichwortverzeichnis, das zum ersten das Nachschlagen von Begriffen erleichtert und zum zweiten das Suchen nach Befehlen, Baugruppen, Jumpfern, usw. die man eben nur mal schnell kurz nachschauen will, sehr vereinfacht.

Dieses neue mc-modular AT Handbuch ist, in Verbindung mit dem AT-Sonderheft, das ideale Werkzeug zum Bau eines eigenen ATs: dem mc-modular AT.

Rüdiger Nahm

FLOMONCG

Neue Version des FLOMON-Monitors

In der neuen Version des FLOMONCG sind sämtliche Fähigkeiten von RNWIN-DOWS (siehe LOOP15) enthalten. Zu Ihrer Erinnerung waren das unter anderem:

- + Geschwindigkeitssteigerung (200% beim SCROLL)
- + inverse Zeichen
- + Fensterverwaltung
- + Maus setzt Cursor (-> PULLUP-Menüs)
- + Hardcopy
- + Statuszeile am unteren Bildrand
- + EPSON-Modus
- + Uhrzeit ständig einblendbar

Die Terminalkarte

Als Terminalkarte kann wahlweise eine COL- oder eine GDP-Platine verwendet werden. Auch der Anschluß beider gleichzeitig ist möglich. (Fast) sämtliche Grafikbefehle der vorhergehenden FLOMON-Versionen sind auch bei der COL anwend-

Eine neue Version der Monitoreproms ist verfügbar, bei der auch ein Betrieb nur mit der COL256 möglich ist. Neben der Anpassung der Grafik- und Textroutinen sind noch eine ganze Reihe weiterer Verbesserungen eingeflossen.

tergrund Invers jeweils eigene Farben verwendet werden. Bei einer Hardcopy werden die Farben durch Schattierungen dargestellt.

Soll das System über ein separates Textterminal an der seriellen Schnittstelle bedient werden, wird dies am Fehlen der KEY- Baugruppe erkannt und die Ein/Ausgaben werden umgelenkt. GDP/COL bleiben für die Grafik über die Monitoreinsprünge verfügbar.

Der Selbsttest

Neu sind auch der Selbsttest und ein integrierter Monitor/Tester: Wird das System nach einem RESET hochgefahren, dann werden alle Bänke er-

Das Menü

Erst dann erscheint das gewohnte Menü, allerdings ohne die Punkte 'Go Bank 2000' und 'GO E0000'. Auf der Bank/

Boot darf jetzt nur noch der Monitor arbeiten, der in 28k-Eproms steckt. ZEAT muß auf eine andere Platine (ROA64 o.ä.) umgesetzt werden. Der andere Menüpunkt wurde durch ein allgemeineres Kommando unter 'Monitor' ersetzt.

Im Testmodus mußten Grafikbefehle bisher blind im Dunkeln getippt werden; jetzt erscheinen die eingegebenen Zeichen, soweit möglich, als Echo auf dem Schirm. Ruft man vom Menü aus 'Monitor/Tester' auf, dann erscheint ein neues Fenster und es wird auf eine Kommandoingabe gewartet. Der aufgerufene Monitor/Tester versteht ähnliche Befehle wie der Monitor des mc-CP/M Computers, ist jedoch erheblich leistungsfähiger. Es gibt zwar schon ähnliche Programme für den NDR-Computer (in ZEAT und GRUNDPRO-

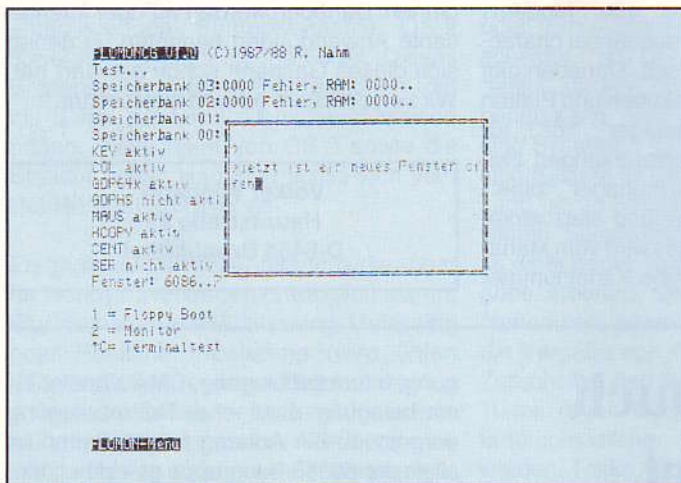


Bild 1: Speichertest und Fenster

bar, selbst die Wirkung von Registeränderungen wird simuliert. Allerdings wird die COL nur in der Auflösung 256 x 256 unterstützt; das bedeutet, daß nur 24+1 Zeilen mit je 42 Zeichen genutzt werden können. Manchen Programmen (Leider auch WS!) ist der Bildschirm dann aber zu klein. Da bei der COL die CPUZ80 komplett für den Bildaufbau verantwortlich ist, muß natürlich mit einer Geschwindigkeitseinbuße gegenüber der GDP gerechnet werden. Im Textmodus können bei der COL für Vordergrund, Hintergrund, Invers und Hin-

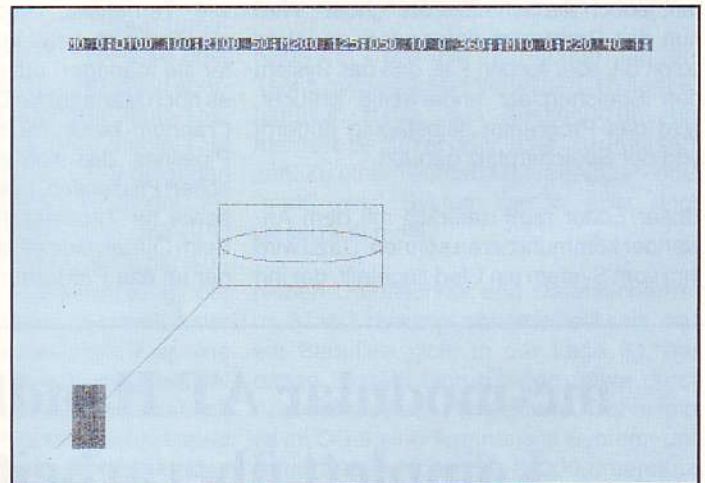


Bild 2: Testmodus mit Kommandozeile

tergrund, auf denen RAM im Bereich F000..FFFF vorhanden ist. Für die Funktionsfähigkeit genügt es, wenn auf EINER BELIEBIGEN Bank (nicht notwendigerweise Bank 0) in diesem Bereich RAM ist. Alle gefundenen Bänke werden markiert und das gesamte RAM zerstörungsfrei auf die korrekte Funktion getestet. Anschließend werden alle in den Monitor einbezogenen Baugruppen auf ihre Anwesenheit getestet und entsprechende Flags gesetzt.

GRAMM z.B.), jedoch fehlt dort noch die Möglichkeit des Zugriffs auf beliebige Bänke. Der neue Monitor arbeitet bankunabhängig, d.h. die aktive Bank kann beliebig durch ein Kommando gewechselt werden. Auch Verschieben von Speicherbereichen zwischen verschiedenen Bänken ist möglich. Folglich können jetzt ROMS für GRUNDPROGRAMM, GOSI, etc. (fast) beliebig an irgendwelchen Adressen im 1MByte Adressraum stehen, bei Bedarf kann der Code dann mit Hilfe eines Monitorbefehls an die richtige Stelle

gebracht werden. Eine Umwandlung ASCII-HEX-DEZIMAL-BINÄR erfolgt automatisch. Anstelle von Zahlenwerten kann jederzeit auch ein Rechenausdruck eingegeben werden; dafür stehen eine ganze Reihe von arithmetischen Operatoren zur Verfügung. Taschenrechner und ASCII-Tabelle werden somit unnötig. Mit einem weiteren Kommando kann auf ein Notizbuch zugegriffen werden, in dem Text beliebig editiert werden kann. Nach dem Starten von CP/M von der Diskette steht der Monitor immer noch zur

buch kopiert werden. Das Notizbuch kann dann im Monitor editiert und von einem beliebigen anderen Programm wieder eingelesen werden. So können Ausgaben eines Programms nachträglich noch zu Dokumentationszwecken in einen Editor (WS, Turbo) übertragen werden!!!

1 MByte adressieren

Auch die Systemfunktionen, die über die Sprungleiste zu erreichen sind, wurden um einige wichtige Punkte ergänzt, etwa

allen Bänken rundet die Systemerweiterung ab.

Nicht alle Funktionen der alten FLOMON-Versionen wurden voll übernommen. Entfallen sind z.B. die Ansteuerung der Cassettenschnittstelle. Probleme mit bestehender Software kann es allenfalls geben, wenn sich diese nicht an die Monitorsprungleiste hält. Inwieweit die Fähigkeiten des neuen Monitors auch auf nicht-original GES-BANKBOOT oder SBC3-Platinen ausgenutzt werden können, ist noch nicht genau untersucht. Im Prinzip

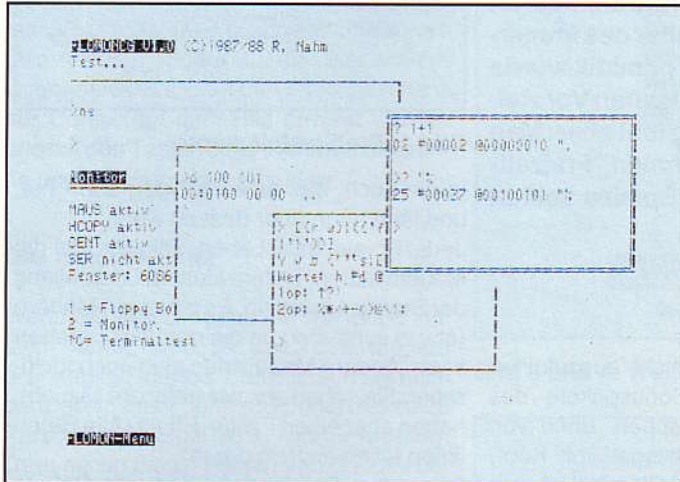


Bild 3: Einsatz des Monitors/Testers

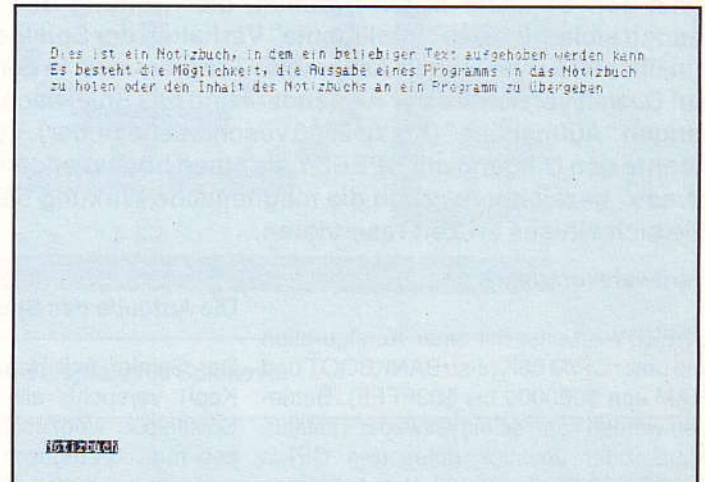


Bild 4: Notizblatt aus dem Tester

Verfügung. Die Aktivierung kann während eines jeden Programms, das CSTS oder CI aufruft, durch Drücken einer Taste erfolgen. Nach Verlassen des Monitors/Testers wird das unterbrochene Programm verlustfrei fortgesetzt. Nur durch diese Möglichkeit gewinnt der Monitor erst seinen vollen Wert. Auf Tastendruck kann jederzeit eine Bildschirmseite in das Notiz-

Mauskoordinaten lesen und Fadenkreuz setzen. Es wurde eine Möglichkeit geschaffen, Unterprogramme auf beliebigen Bänken aufzurufen (entspricht etwa FAR CALL beim 8086). Somit wäre die 64k-Barriere im Prinzip überwunden. Die Sprungleiste steht auf allen Bänken zur Verfügung. Eine Speicherverwaltung für den freien Platz auf

müßten zumindest alle Funktionen korrekt arbeiten, die sich NICHT auf irgendwelche Bänke beziehen. Den neuen Monitor FLOMONCG gibt es auf 2 Eproms, zusätzlich können bei Bedarf Quelltexte (ca. 600 kByte !!) und ein paar DEMOS und UTILITIES auf Diskette angefordert werden.

Rechner modular wieder im Fernsehen

Die Sendereihe des Norddeutschen Rundfunks zum NDR-Computer, dem eigens für diese Sendung entwickelten Rechner.

Fernsehfolgen	Länge in min	Sendetermin NDR, Berlin		Sendetermin BR Samstag
		Freitag	Montag	
Schritt für Schritt	15	02. 09., 17 ⁴⁰	05. 09., 11 ¹⁰	24. 09., 12 ¹⁵
Die Spannungsversorgung	15	09. 09., 17 ⁴⁴	12. 09., 11 ¹⁵	01. 10., 12 ¹⁵
Startlogik und Taktgenerator	15	16. 09., 17 ⁴¹	19. 09., 11 ¹⁵	08. 10., 12 ¹⁵
Die Zentraleinheit	14	23. 09., 17 ⁴⁴	26. 09., 11 ¹⁵	15. 10., 12 ¹⁵
Dem Speicher auf der Spur	15	30. 09., 17 ⁴⁴	24. 10., 11 ¹⁵	22. 10., 12 ¹⁵
HEXMON	14	11. 11., 17 ⁴³	14. 11., 11 ¹⁵	29. 10., 12 ¹⁵
Lauflichter	14	18. 11., 17 ⁴⁰	21. 11., 11 ¹⁰	05. 11., 12 ¹⁵
Roboter programmieren	15	25. 11., 17 ⁴³	28. 11., 11 ¹⁵	12. 11., 12 ¹⁵
Der springende Punkt	15	02. 12., 17 ⁴⁴	05. 12., 11 ¹⁵	19. 11., 12 ¹⁵

(insg. 26 Folgen)

SPEEDY

Spannendes Computerspiel für CPU 680XX

(C) Ralf Ulrich

In einem Punkt sind sich momentan alle einig: SPEEDY ist das beste Computerspiel, welches bisher für den NDR-Computer veröffentlicht wurde. Optisch betrachtet ist es eine hervorragende Ausnutzung der graphischen Fähigkeiten der GDP64(HS)-Baugruppe. Die Geschwindigkeit versetzt den Benutzer in ein Staunen, die Krönung des Programmes ist jedoch sicherlich das "intelligente" Verhalten der Spielgegner des immerhin über 90 KByte langen Assembler-Programmes. In der Thematik wurde auf Gewalt verzichtet. Der Anwender kann das Spiel nach eigenen Vorstellungen "Aufmachen" (bis zu 2500 verschiedene Bilder). Vorsicht aber: Man könnte den Umgang mit SPEEDY als einen höchst angenehmen "Freizeitstress" bezeichnen; durch die magnetische Wirkung des Spieles sollten Sie sich einiges an Zeit reservieren.

Hardwarevoraussetzung:

SPEEDY arbeitet mit einer Konfiguration wie unter CP/M 68K (also BANKBOOT und RAM von \$000000 bis \$03FFFF). Betrieben werden kann es mit entweder Tastatur, Maus oder Joystick unter den CPU's 68008, 68000 oder 68020. Von Anfang an wird die neue GDP64HS (siehe LOOP18) unterstützt. Es ist kein Betriebssystem notwendig, es kann einfach aus dem Grundprogramm mit "FLOOPY BOOT" gestartet werden. Das Spiel kann einfach an die jeweilig verwendete Hardware angepasst werden.

Erst mal die technischen Daten:

- CPU 680xx- mit Tastatur, Maus oder Joystick
- kein Betriebssystem notwendig
- über 90 KByte Assembler-Quelltext
- 2500 verschiedene Bilder möglich
- "High Score" Tabelle für die besten 100 Spieler
- Editor zum Erstellen und Ändern der Bilder
- Individuelle Hardwareanpassung
- bis zu 20 Mitspieler
- Übungsteil
- Zwischenspeichern möglich

Nun zum Spiel:

Die Spielelemente bestehen aus dem Spieler (siehe Bild1 oben links), dem Gegner, den verschiedenen Mauern (grabbare, nicht grabbare und welche mit Falltüre), den Bonuspaketen, den Leitern zum Klettern und Abspringen und den Stangen zum Entlanghängeln.

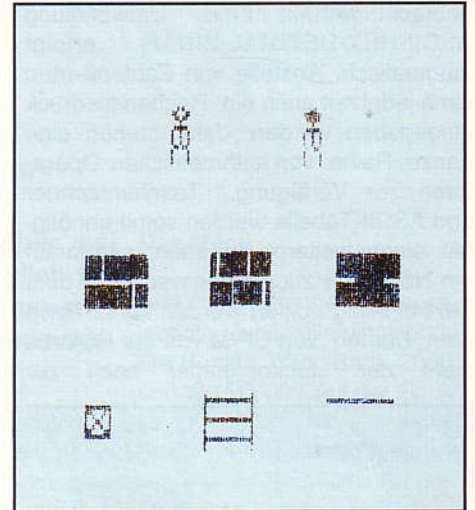


Bild 1: Die Spielelemente

in ein Loch, das er vorher gegraben hat, und läuft dann über dessen Kopf.

Jeder Spieler hat 9 Leben. Er kann über die Mauern laufen, Leitern klettern und entlang der Stangen hängeln. Es gibt auch Mauern (etwas schmaler), in die man nicht graben kann. Andere Mauerarten sind noch tückischer: Sie sehen aus wie grabbare Mauern, haben aber einen Falltür-Effekt (Alle Beteiligten fallen einfach durch).

Ist es dem Spieler gelungen, alle Pakete des Bildes (Levels) zu schnappen, ohne daß die Gegner noch welche besitzen, gelangt er in den nächsten Level.

Die Aufgabe des Spieles:

Der Spieler (mit dem nicht ausgefüllten Kopf) versucht, alle Bonuspakete des Spielfeldes wegzuschnappen, ohne von den max. 6 Gegnern (ausgefüllter Kopf) erwischt zu werden. Als Hilfsmittel ist er in der Lage, in die Mauer Löcher zu graben, in die die Gegner fallen. (Achtung, er kann auch selber reinfallen!)

Er muß dabei auch die Bonuspakete der Gegner, von denen jeder eines hat, erwischen. Dazu "lockt" der Spieler die Gegner

Selbstverständlich werden die Punkte gezählt. Die besten 100 Spieler werden in einer "High-Score" Tabelle "verewigt".

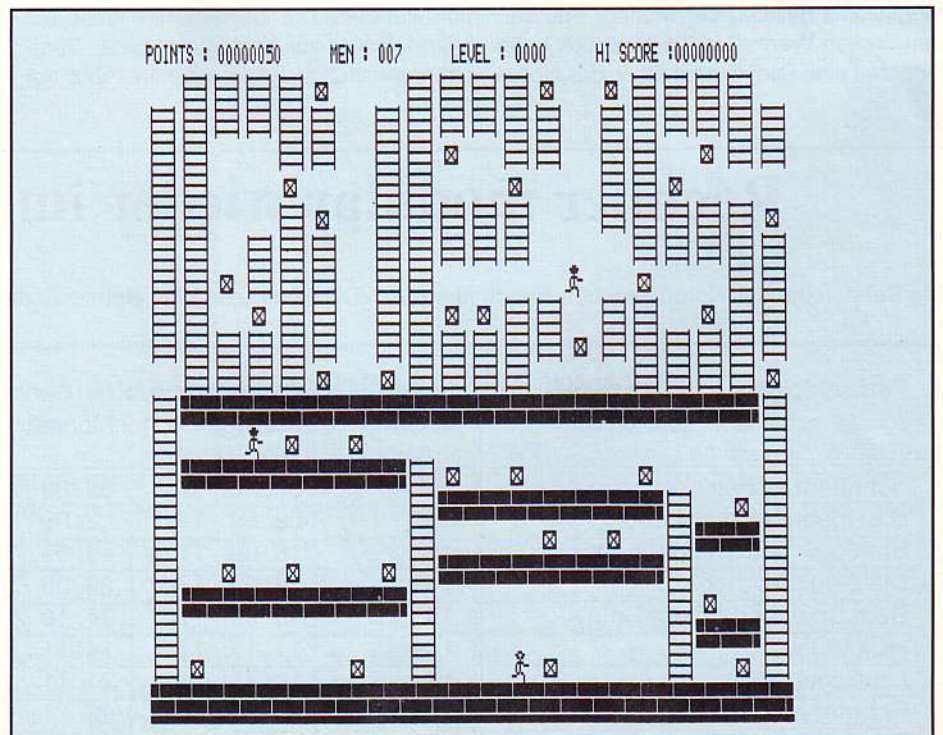


Bild 2: Der Eröffnungslevel

Das Übungsspiel

Bevor man "ernsthaft" mit dem Spielen beginnt, sollte man erst mal alles mit dem Übungsspiel ausgiebig testen. Man kann dabei entweder einen speziellen Level einstudieren, oder alle mal kurz "durchblättern" und nur die interessantesten ausprobieren.

Der Editor

Wenn Ihnen ein Level zu einfach wird, kann dieser mit dem Editor blitzschnell "aufgemacht" werden. Dies geschieht am schnellsten dadurch, indem Sie ein paar weitere Gegner einsetzen. Mit den Pfeiltasten kann der Cursor auf dem Bild bewegt werden und mit den Tasten 0 bis 8 können Bausteine und Spieler gesetzt werden.

Das normale Spiel

Wenn Sie den Umgang mit "SPEEDY" im Griff haben, wird es Zeit, daß Sie auf "Punktejagd" gehen. Das müssen Sie nicht ganz alleine tun, denn es können bis zu 20 Personen mitspielen (Turnier-Modus). Dazu geben Sie im Hauptmenü die Zahl der mitspielenden Teilnehmer und die jeweiligen Namen ein. Sind alle 9 Leben eines Spielers beendet, wird dieser in der "High-Score" Tabelle eingetragen.

Beendet man ein "richtiges" Spiel, erhält man die Gelegenheit, dieses Spiel abzuspeichern, um es später vielleicht weiterzuspielen.

Es läßt sich aber nur ein Spiel abspeichern.

Übrigens noch ein letzter Tip des Autors:

Wenn Sie am Ende Ihrer nervlichen und körperlichen Kräfte sind, wenn die Nacht schon wieder mal um 3 Stunden zu lang geworden ist und die Temperatur Ihrer Stirn der der CPU in Ihrem NDR-Computer entspricht:

AUSSCHALTEN !!!

Christoph Köhler, GES

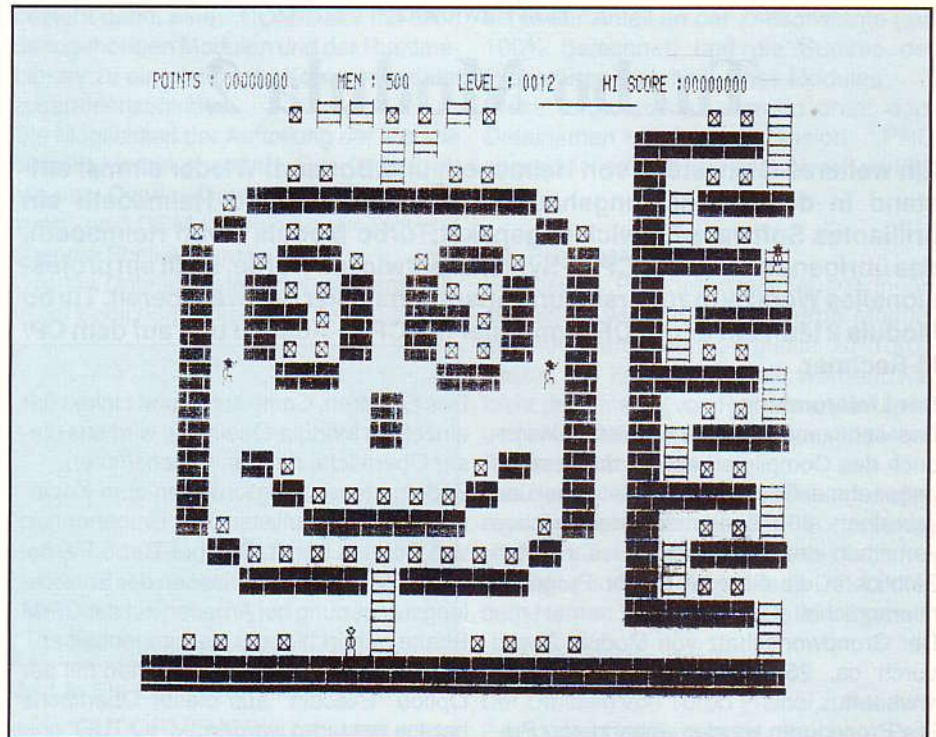
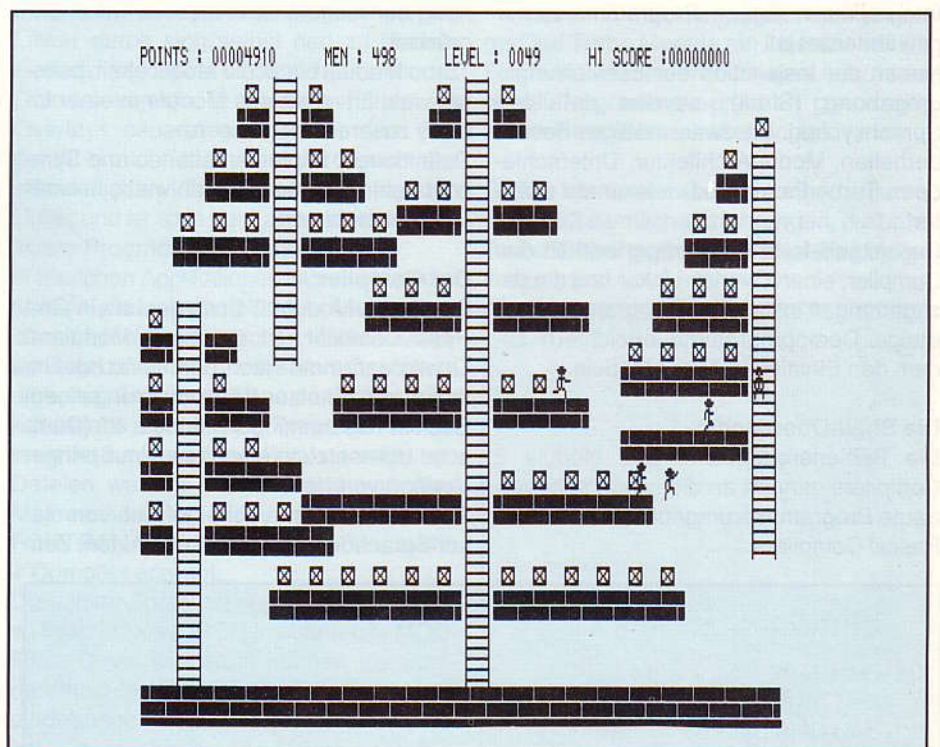


Bild 3 und 4: Weitere Beispiele für Spiellevel



!!! SONDERANGEBOT !!!

Speziell für alle LOOP-Leser

RAM 8K
Bestellnummer 10357

zum

RAM 8K
Bestellnummer 10357

Super - Preis DM 12.-

Ralf Pawlowitz

Turbo Modula 2

Ein weiteres Glanzstück von Heimsoeth und Borland! Wieder einmal entstand in den Entwicklungshäusern von Borland und Heimsoeth ein brillantes Software-Entwicklungspaket. Turbo Modula 2 von Heimsoeth, das übrigens nur für Z80 CP/M-Systeme entwickelt wurde, stellt ein professionelles Werkzeug zur Erstellung anspruchsvoller Software bereit. Turbo Modula 2 läuft auf dem NDR-Computer mit CP/M-Ausbau und auf dem CP/M-Rechner

Der Lieferumfang:

Das sehr umfangreiche englische Handbuch des Compilers besticht durch seine angenehme Gliederung. Die - sage und schreibe - 600 Seiten des Handbuchs vermitteln dem Leser einen sehr guten Einblick in die Struktur dieser Programmiersprache.

Der Grundwortschatz von Modula 2 wird durch ca. 250 zusätzliche Prozeduren erweitert.

Die Prozeduren werden anhand von Programmbeispielen und Querverweisen ausgiebig erklärt, sodaß eine Anwendung und Integration in eigene Programme sicher gewährleistet ist.

Neben der Installation der Entwicklungsumgebung (Shell), werden detailliert Sprachsyntax, hardwaremäßige Besonderheiten, Modul-Architektur, Unterschiede zu Turbo Pascal und vieles mehr erläutert.

Der mitgelieferte Datenträger enthält den Compiler, einen Overlay-Linker und die dazugehörigen Installationsprogramme.

Einige Demoprogramme erleichtern zudem den Einstieg in Turbo Modula.

Die Shell-Oberfläche:

Die Bedieneroberfläche des Modula 2 Compilers erinnert an die schon sehr vertraute Programmierumgebung des Turbo-Pascal Compilers.

Das Editieren, Compilieren und Linken der einzelnen Modula-Quelltexte wird aus dieser Oberfläche heraus vorgenommen.

Zudem stehen Möglichkeiten zum Kopieren, Löschen, Auflisten oder Umbenennen von Dateien bereit. Das bei Turbo Pascal sehr umständliche Verlassen der Entwicklungsumgebung bei Arbeiten auf der CP/M Ebene gehört hier der Vergangenheit an.

Selbst gelinkte Com-Files können mit der Option "Execute" aus dieser Oberfläche heraus gestartet werden.

Die Angst vor der Modul-Hierarchie und die damit entstehenden Probleme der Ordnung der Module ist in diesem Fall unbegründet.

Turbo Modula bietet die Möglichkeit, beliebig viele artverwandte Module in einer Library zusammenzufassen.

Definitionen, Implementationen und Symboldateien werden hier wahlweise in einer Library kombiniert.

Der Compiler:

Der Turbo Modula 2 Compiler ist ein One Pass Compiler. Er übersetzt Modula 2 Quellprogramme nach den Sprachdefinitionen von Nikolaus Wirth, die er in seinem Buch "Programming in Modula 2" (Deutsche Übersetzung erschienen im Springer Verlag) veröffentlichte.

Turbo Modula unterscheidet sich von dieser Sprachdefinition in zwei Punkten. Zum



Einen bietet es wesentliche Erleichterungen durch seine umfangreichen Libraries, zum Anderen werden I/O-Operationen durch den erweiterten Wortschatz ausgiebig unterstützt.

Der Compiler legt unterschiedliche Ausgabeformate an.

Implementationsmodule

*.MOD -> Nativ - Code *.MCD
M -Code *.MCD

Definitionsmodule

*.DEF -> Symboldateien *.SYM

Für Nicht-Eingeweihte hier kurz der Unterschied zwischen M-Code und Native-Code:

M-CODE ist ein Zwischencode des Compilers, der nur mit Hilfe der Runtime-Library des Compilers ablauffähig ist.

Der **NATIVE CODE** dagegen ist direkter Z80 Maschinencode, der zwar um den Faktor Zehn schneller ist als der M-Code (Handbuch - Angaben), jedoch das Dreifache an Speicherplatz benötigt. Außerdem ist er für eine Vielzahl von mathematischen Funktionen (floating Point) ungeeignet.

Die Softwareschalter:

Das Entwicklungssystem bietet die Möglichkeit, durch definierte Softwareschalter den Übersetzungsvorgang und das Ausgabeformat des Compilers zu steuern.

Diese Softwareschalter werden entweder direkt in das Quellprogramm integriert, oder durch entsprechende Eingaben über die Shell gesteuert.

- Bei dem Ausgabeformat kann zwischen Native- oder M-Code unterschieden werden.

- Die Ausgabe einer Overflow-Meldung kann unterdrückt oder auch durch Abfangen mit Hilfe einer Exception-Fehlerroutine abgefangen werden.

- Der Umfang jedes ARRAY-Bereiches kann bei Bedarf eingegrenzt werden. Dadurch übernimmt das compilierte Programm nur den wirklich verwendeten Speicherplatz im Arbeitsspeicher des Computers an. Programme werden dadurch kompakter.

- Der Compiler akzeptiert bei Bedarf auch die Kleinschreibung einer Variablen, obwohl sie im Deklarationsteil als Großbuch-

```

Selected drive: B
Work file:

Edit Compile Run execute
Link Options Quit Librarian

Dir Filecopy Kill rename type

>D
Directory mask: *.mod
 1: .MCD          5: HELLO .MCD      8: PRIME .MCD     11: SHELL .MCD
 2: COMPILER .MCD 6: LIBRARY .MCD    9: PROFILE .MCD  12: STRTEXTS.MCD
 3: FILTER .MCD   7: LINK .MCD     10: REL .MCD     13: TEST .MCD
 4: GENZ80 .MCD
Bytes Remaining on B: 444K
    
```

Bild 1: Die Bedieneroberfläche

stabe definiert wurde. (In der strengen Struktur von Modula 2 nicht üblich)

- Im Sprachumfang von Modula 2 nicht inbegriffene Erweiterungen bilden bei entsprechender Einstellung einen erweiterten Bestandteil des Sprachumfangs. (z.B Turbo Pascal's WRITE und READ Operation)
- Ein Listing wird bei Durchlauf des Quellprogrammes durch den Compiler am Bildschirm angezeigt und auf dem Datenträger angelegt. Dadurch kann der Compiler bei seiner momentanen Arbeit beobachtet werden.

Die Suche nach einem Fehler beim Abbruch eines Programmes wird erleichtert. Die Diagnose von Runtime-Fehlern wird durch Statustabellen des Listings verbessert.

Der Linker:

Der Linker erzeugt zwei verschiedene Ausgabeformate. Einmal *.MCD-Files, zum anderen *.COM-Files, also direkt ausführbare Kommandodateien.

Die Dateien mit der Extension MCD bestehen aus kompilierten Modula 2 Quelltexten. Sie können nur aus der Entwicklungsumgebung heraus gestartet werden.

Dateien, die eine COM Spezifikation besitzen, wurden bereits fertig gebunden (gelinkt) und benötigen die Shell für einen anschließenden Programmablauf nicht mehr.

```

compiler options:
List (ON)  Native (OFF)  Extensions (ON)
Test (ON)  Overflow (OFF) Upper/lower (OFF)

Path to search: SYSLIB
Find run-time error
Save current selection      Quit
>
    
```

Bild 2: Compiler-Option

Jetzt besteht die Möglichkeit, die Objektdateien mit der Extension-MCD unterschiedlich zu organisieren.

Prinzipiell hat ein Linker ja die Aufgabe, mehrere Dateien in eine einzige Datei einzubinden. Nun, wie verhält es sich jetzt mit den MCD-Dateien? Diese Aufgabe wird doch eigentlich von dem Compiler erledigt. Er komprimiert doch die Vielzahl der Module zu einer einzigen Datei.

Doch in der Praxis werden oft mehrere Organisationsdateien für verschiedenartige Aufgaben benötigt. Daß diese Dateien jedoch erst von dem Datenträger gelesen werden müssen, ist ein unangenehmes Übel, das es zu beseitigen gilt.

Der Linker übergeht dieses Manko, indem er diese Dateien zu dem eigentlichen Hauptprogramm virtuell dazubindet.

Die eigentliche Hauptaufgabe des Linkers

besteht darin, eine *.COM-Datei mit allen dazugehörigen Modulen und der Runtime-Library zu einer einzigen Kommandodatei zusammenzubinden.

Die Möglichkeit der Aufteilung der Module besteht hierbei ebenfalls. Das Organisieren von Overlay-Dateien wird bei Benutzung von *.COM-Dateien angewandt und über die Shell abgewickelt.

Overlay-Dateien bieten auch effiziente Hilfe bei Speicherproblemen. Die Fehlermel-

```

Link main module: BOO:FILTER
Output File: filter.com

Include all needed modules (Y/N)? Y

Linking KONVERT
Linking TEXTS
Linking FILES
Linking KONVERT
Linking STRINGS
Linking TERMINAL
Linking COMLINE

Linked Modules:
FILES      KONVERT  TEXTS      STRINGS    TERMINAL
COMLINE

Unresolved External References: None
Missing Modules: None
Order of Initialization:
TERMINAL  TEXTS      COMLINE    KONVERT
    
```

Bild 3: Ein Linkerdurchlauf

dung "OUT OF MEMORY" hat sicher schon so manchen CP/M User das Fürchten gelehrt.

In diesem Fall besteht die Möglichkeit, den Linker durch sich selbst neu zu binden. Module, wie etwa Dateioperationen des Linkers, können in solchen Fällen als Overlays separat gelagert werden. Dadurch erreicht man das Verringern des erforderlichen Speicherplatzes für den Linker und ist so in der Lage, speicherintensivere Programme zu schreiben.

In manchen Applikationen ist es notwendig, Module, die in Assembler geschrieben wurden, mit in den eigentlichen Modula-Quelltext einzubinden. Entweder aus Geschwindigkeitsgründen oder aus der Notwendigkeit heraus, bestehende Programm-Module weiter zu verwenden. Solche relokativen Dateien werden üblicherweise von dem Makroassembler M80, dem Microsoft FORTRAN Compiler oder dem Pascal MT/ + Compiler erzeugt.

Der Linker übersetzt sogenannte relokative Programme (*.REL) in übersetzte MCD-Files. Diese wiederum können als Standardmodule in jedes Modula Programm eingebunden werden. Die Variablenübergabe wird hier in der Regel über den STACK gelöst.

Der Profiler:

User, die an zeitkritischen Programmen ihre Freude finden, sollen auch ihr Bonbon bekommen.

Auf Wunsch wird bei Probelauf des kompilierten MCD-Files ein Protokoll der Durchlaufzeiten der einzelnen Prozeduren erzeugt.

Aufgezeichnet wird die Anzahl der Durchläufe der einzelnen Prozeduren, ihr pro-

zentualer Anteil an der Zeitkonstante (auf 100% gerechnet) und die Summe der Prozeduren innerhalb eines Modules.

Diese Informationen werden unter dem Dateinamen mit der Extension *.PRO abgespeichert und sind so jederzeit verfügbar.

Ein Programmbeispiel:

Ein kleines Programmbeispiel soll Ihnen aufzeigen, wie strukturiert Turbo Modula arbeitet. Durch den Vergleich zu Turbo Pascal 4.0 kann dargestellt werden, wie leicht der Umstieg von Pascal zu Modula ist und wie effizienter das Arbeiten mit dieser Programmiersprache ist.

Das dargestellte Programm liest eine beliebige Textdatei, durchsucht sie nach bestimmten Zeichen, ersetzt diese durch frei wählbare und legt die gefilterte Datei unter dem Namen *.Fil auf dem Datenträger ab.

Modula: Ja oder Nein

Der Umstieg von Turbo Pascal zu Modula 2 wird das Gemüt von so manchem Programmierer unter Ihnen in Aufruhr bringen. Pascal-Programmierern sei gesagt, daß mit diesem Entwicklungspaket das Umsteigen auf Turbo Modula ein Leichtes ist. Die Flexibilität dieses Werkzeuges erlaubt es, beide Programmiersprachen gleichzeitig zu erlernen und zu beherrschen.

Schon bei ersten Gehversuchen erkennt man, daß sämtliche Erfahrungen, die unter Turbo Pascal gemacht wurden, hier eingeflossen sind.

```

Profile:

Module: KERNEL

Procedure Occurences Per cent
TDABOL      51      0.38%
FJGGPA     325      2.45%
V WEDL      776      5.85%
XHEPA       26      0.20%
Total:      1178      8.88%

Module: COMLINE

Procedure Occurences Per cent
FROMPT       30      0.23%
Total:       30      0.23%

Module: LOADER

Procedure Occurences Per cent
SJI42T       84      0.63%
20DBWX      446      3.36%
BRLYFX       200      1.51%
CFU5ZX       774      5.83%
R7T5ZX        65      0.49%
YENISX       387      2.92%
EGJKRN        30      0.23%
XXB5ZX       544      4.10%
E6P093        48      0.36%
WAZUOI     1308      9.86%
WH4ZX        48      0.36%
BM47KC        62      0.47%
LZDI8U        28      0.21%
SOBVX        495      3.73%
PRCMRX        150      1.13%
LOADTH        49      0.37%
CALL         450      3.39%
Total:      5168      38.95%
    
```

Bild 4: Auszug aus einer Profiler-Datei

```

*****
*) Programm sucht beliebiges Zeichen aus einer Textdatei heraus
*) und erstellt es durch ein frei wählbares
*) (C) R.Falkowitz 11.06.88
*) Turbo Modula 2 V1.0-01
*) 808 Version 1.0
*****

```

```

MODULE Konvert;
FROM Texts IMPORT TEXT,ReadLine,OpenText,EDIT,CloseText,ReadLn;
FROM Strings IMPORT String;
FROM Terminal IMPORT WriteScreen,WriteLn;
FROM Console IMPORT PromptFor;

TYPE (Laenge = $ARRAY[1..100] OF CHAR);

VAR
  DateName : $ZLaenge;
  Index,I,Counter : INTEGER;
  Datei,Datei1 : FILE;
  ZersetzendeZeichen : $CHAR;
  ErsetzendeZeichen : $CHAR;
  EntwerzendeZeichen : $INTEGER;
  DatenNameZiel : $STRING[50];
  QuellIndex : $INTEGER;
  PunktFos : $INTEGER;
  EOL : $STRING[20];
  DatenLaenge : $INTEGER;
  QuellIndex : $INTEGER;
  EOL : $STRING[20];

```

```

PROCEDURE FILTER;
  BEGIN
    PromptFor('DateiName(Quelle): ');
    PromptFor('DateiName(Ziel): ');
    OpenText(Datei,DateiNameQuelle);
    IF NOT FALSE THEN WRITELN('Datei nicht vorhanden!'); END;
    CreateText(Datei1,DateiNameZiel);
    Counter:=0;
    Index:=0;
    WHILE EOF(Datei) = FALSE DO
      ReadLn(Datei,QuellIndex);
      IF QuellIndex<CHR(10) THEN
        THEN
          QuellIndex:=INDEX(1);
          EntwerzendeZeichen:=EntwerzendeZeichen;
        ELSE
          WRITE(Datei1,QuellIndex);
        END;
      CloseFile(Datei);
      CloseFile(Datei1);
      WRITELN('EntwerzendeZeichen: ',EntwerzendeZeichen);
    EXCEPTION
      TextDatei:=DateiNameZiel;
    END;
  END;

  IF Counter=0 THEN WriteLn(Datei1,QuellIndex);
  END;
  CLOSE DATEI1;
  WRITELN('EntwerzendeZeichen: ',INDEX);
  WRITELN('EntwerzendeZeichen: ',EntwerzendeZeichen);
  END;
  BEGIN
  FILTER;
  END.

```

```

*****
*) Programm sucht beliebiges Zeichen aus einer Textdatei heraus
*) und erstellt es durch ein frei wählbares
*) (C) R.Falkowitz 11.06.88
*) Turbo Pascal 4.0 V1.0-01
*) 808 Version 1.0
*****

```

```

PROGRAMM FILTER;
USES Graph;
Type
  Char;
  Integer;
  String;

VAR
  DateName : $STRING[50];
  Index,I,Counter : INTEGER;
  Datei,Datei1 : FILE;
  ZersetzendeZeichen : $CHAR;
  ErsetzendeZeichen : $CHAR;
  EntwerzendeZeichen : $INTEGER;
  DatenNameZiel : $STRING[50];
  QuellIndex : $INTEGER;
  PunktFos : $INTEGER;
  EOL : $STRING[20];
  DatenLaenge : $INTEGER;

```

```

PROCEDURE FILTER;
  BEGIN
    DateName:=DateiNameQuelle;
    IF DateName=<> THEN EXIT;
    PunktFos:=INDEX(1);
    CreateText(Datei1,DateiNameZiel);
    DatenLaenge:=length(DateiNameQuelle)-length(DateiNameZiel);
    WriteLn('EntwerzendeZeichen: ',DateiNameQuelle);
    Assign(Datei,DateiNameQuelle);
    Assign(Datei1,DateiNameZiel);
    Reset(Datei);
    Reset(Datei1);
    Counter:=0;
    WHILE EOF(Datei) = FALSE DO BEGIN
      ReadLn(Datei,QuellIndex);
      IF QuellIndex<CHR(10) THEN
        THEN
          QuellIndex:=INDEX(1);
          EntwerzendeZeichen:=EntwerzendeZeichen;
        ELSE
          WriteLn(Datei1,QuellIndex);
        END;
      CloseFile(Datei);
      CloseFile(Datei1);
      WRITELN('EntwerzendeZeichen: ',EntwerzendeZeichen);
    EXCEPTION
      TextDatei:=DateiNameZiel;
    END;
  END;

  IF Counter=0 THEN WriteLn(Datei1,QuellIndex);
  END;
  CLOSE DATEI1;
  WRITELN('EntwerzendeZeichen: ',INDEX);
  WRITELN('EntwerzendeZeichen: ',EntwerzendeZeichen);
  END;
  BEGIN
  FILTER;
  END.

```

Bild 5: Vergleich: Turbo Modula 2 oder Turbo Pascal

IHK Straubing

Mikroprozessor Kurse mit dem NDR Einsteigerpaket

Beispiel aus dem Kurs: Kapazitätsmessung mit dem Einsteigerpaket

Fast jedes elektrische Gerät, angefangen von der Waschmaschine bis hoch zur Mikroprozessorgesteuerten vollautomatischen Drehmaschine, enthält einen Mikroprozessor. Leider ziehen die Berufsschulen dieser Entwicklung nur sehr langsam

nach. Um so wertvoller muß die Pionierarbeit, die die IHK Straubing auf Initiative von Herrn Fachstudienrat Alois Stirner, geleistet hat, eingeschätzt werden.

Der Kurs umfaßt den Aufbau eines Mini-

Das NDR Einsteigerpaket findet immer mehr Anklang bei Ausbildern und Schülern im Bereich der Mikroprozessortechnik. Die IHK Straubing arbeitet schon seit 1985 mit dem Einsteigerpaket und hat bereits einige Kurse durchgeführt. Vor allem die Auszubildenden der elektrotechnischen Berufe (Nachrichtengerätetechnik, Informationselektroniker, usw.) werden mehr und mehr mit Mikroprozessoren in ihrem Beruf gefordert.

des RC-Gliedes als der zu messende Kondensator C_x eingesetzt. Bei konstantem Widerstand ist die Länge des Monoflop-Impulses direkt proportional zur Kapazität des Kondensators.

schalter (CD4066) jeweils der günstigste Widerstand zugeschaltet. Der Ausgang des Monoflops steuert den Interrupt des Z80.

Die Software ist eigentlich recht einfach. Das Monoflop wird getriggert und der Interrupt freigegeben. Ein Softwarezähler (Register HL) wird hochgezählt, bis das Monoflop abfällt und dadurch ein Interrupt ausgelöst wird. Da zum Anfang des Programmes der Interrupt Modus 1 eingestellt wird, springt die Software eine feste Interruptadresse an, wo die Interruptroutine steht. Diese Routine hat die Aufgabe, den Zähler in einen Dezimalwert zu wandeln, zur erkennen, ob der ideale Meßbereich gewählt wurde und das Meßergebnis auf die Anzeige zu bringen. Sollte der ideale Meßbereich noch nicht eingestellt gewesen sein, wird der Analogschalter umgeschaltet und die Messung noch einmal durchgeführt. Dies wird solange wiederholt (max. vier mal) bis der ideale Meßbereich gewählt wurde.

Das Programm (Listing) ist auf einer SBC2 (Adresse 1000h) ablauffähig. Zum Betrieb auf der SBC3 (Adresse 2000h) ist der im Listing hervorgehobene Wert von 10 in 20 (Adresse 1007) abzuändern.

Achtung ! Die 16 Bit-Werte aus dem Listing müssen in verdrehter Reihenfolge eingegeben werden.

z.B. statt **C3 8800** ----> **C3 00 88**



Bild 1: Kursteilnehmer der IHK-Schulung

mal Mikroprozessorsystemes, sowie die meßtechnische Erfassung von Mikroprozessorsignalen. Außerdem umfaßt der Kurs eine Einführung in die Maschinsprache des Mikroprozessors Z80, natürlich mit dem NDR Einsteigerpaket.

In Straubing wird inzwischen ein Grund- und ein Aufbaukurs durchgeführt; die nächsten Kurse sind schon wieder im Laufen oder in Vorbereitung. Die Kursteilnehmer erhalten nach Abschluß des Kurses ein Zertifikat.

An dieser Stelle wollen wir ein Produkt dieses Kurses vorstellen: Eine Zusatzschaltung für die IOE und die dazugehörige Software zur Kapazitätsmessung von Kondensatoren.

Beschreibung der Zusatzschaltung

Die Schaltung ist vom Prinzip her ein Monoflop, das über ein RC-Glied eingestellt wird. Dabei wird der Kondensator

Um nun aber für den entsprechenden Kondensator den idealen Meßbereich einstellen zu können, wird über einen Analog-

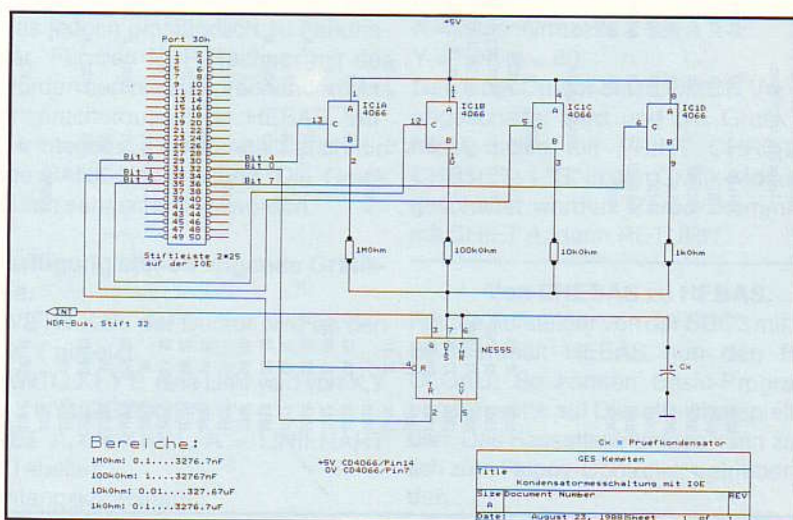


Bild 2: Schaltplan des Kondensatormeißgerätes

TITLE 'Kondensatorkapazitäten mit Einstiegpaket messen'
 ;HE555 auf einer IOE wird mit unbekanntem Cx als monostabile
 ;Kippstufe betrieben.

```

8008 EQU 800BH
0030 EQU 30H
0009 EQU 09H
0033 EQU 33H
0021 EQU 21H
0024 EQU 24H
8000 EQU 8000H
0015 EQU 15H

.PHASE 1000H

INTLOC EQU 800BH
PORT EQU 30H
ANZEIGE EQU 09H
CLEAR EQU 33H
PRIDEZ EQU 21H
ANZFELD EQU 24H
PRINT EQU 15H

.Z80
LD A,0C3H
LD (INTLOC),A
LD HL,INTSERV
LD (INTLOC+1),HL
IM 1
DI
LD A,10H
LOOP1: LD A,10H
ADD A,3
LD HL,0
OUT (PORT),A
SUB 02H
OUT (PORT),A
LD B,0
WART: DJNZ WART
ADD A,2H
OUT (PORT),A
DEC A
OUT (PORT),A
INC A
OUT (PORT),A
INC A
OUT (PORT),A
EI
LOOP2: LD C,1CH
WARTZ: C
NZ,WARTZ
INC HL
BIT 7,H
JR Z,LOOP2
AND 0F0H
RLA
JR NC,LOOP1

CALL CLEAR
LD HL,XTERR
CALL PRINT

LOOP3: CALL ANZEIGE
JR LOOP3

INTSERV:
;Interruptbearbeitungsroutine
PUSH HL
CALL CLEAR
    
```

```

104C HL
104D IX,ANZFELD + 5
1051 IX,ANZFELD
1055 CP 13H
1057 Z,BEREICH1
1059 CP 23H
105B Z,BEREICH2
105D CP 43H
105F Z,BEREICH3
1061 BEREICH4:
1061 LD A,0E3H
1063 (ANZFELD + 6),A
1066 A,BEH
1068 (ANZFELD + 7),A
1068 PRIDEZ
106E RES 7,(1Y+4)
1072 JR INTENDE

BEREICH1:
1074 LD A,0ABH
1076 (ANZFELD + 6),A
1079 A,BEH
107B (ANZFELD + 7),A
107E PRIDEZ
1081 RES 7,(1Y+4)
1085 JR INTENDE

BEREICH2:
1087 LD A,0ABH
1089 (ANZFELD + 6),A
108C A,BEH
108E (ANZFELD + 7),A
1091 JR INTENDE

BEREICH3:
1093 LD A,0E3H
1095 (ANZFELD + 6),A
1098 A,BEH
109A (ANZFELD + 7),A
109D CALL PRIDEZ
10A0 RES 7,(1Y+3)

INTENDE:
10A4 CALL ANZEIGE
10A7 JR INTENDE

XTERR:
;Bringt 'FEHLER' in die Anzeige
DEFB 8EH,86H,89H,0C7H
DEFB 86H,0AFH,0FFH,0FFH

END
    
```

No fatal error(s)

Bild 3: Programm zum Messen von Kondensatorkapazitäten

Dr. Hehl Hans

HEBAS: Tips und Tricks , Nr. 11

Wer kennt das nicht? Da läßt man seinen Computer eine einfache Rechenaufgabe lösen - schließlich heißt er ja "Rechner" - und dann bringt der Kasten bei den simpelsten Rechnungen falsche Ergebnisse! Da fragt man sich, ob man seinem Computer überhaupt noch trauen kann. Wieso das passieren kann, woran es liegt und wie man solche Fehler vermeidet, soll hier erklärt werden.

Der Leserbrief von Herrn Kesa in LOOP16 wies auf ein Problem hin, das prinzipiell bei einer binären Verarbeitung von Zahlen auftritt. Auch HEBAS und z.B. GW-BASIC für MS-DOS-Rechner zeigen den gleichen Fehler. Der Fehler läßt sich vermeiden, wenn im jeweiligen Programmschritt eine sehr kleine Zahl, z. B. 0,0000000001 (je nach Rechengenauigkeit) dazu addiert wird. So ergibt die BASIC-Anweisung PRINT INT(1.3 - 0.3 + 0.0000000001) den Wert 1.

```
10 REM Fallentest HEBAS
20 FOR I 0 -0.5 TO +0.5 STEP 0.1
30 PRINT I
30 : IF I=0 THEN PRINT "I=0"
40 NEXT I
```

Bildschirmanzeige für I bei 0:
-2.2737363544 E-13

Bild1: Fallentest für binäre Verarbeitung

Dies ist ein Hinweis darauf, daß Zahlen in binärer Darstellung bei begrenzter Stellenzahl nicht exakt darstellbar sind. Der bekannte Fallentest in Bild 1 zeigt nochmal die Problematik. Die PRINT-Anweisung in Zeile 30 wird nie ausgeführt.

```
1234,5 -->1,2345*10+3-->1.2345E+3
0,0012345 -->1,2345*10-3-->1.2345E-3
```

Bild 2: Zerlegung in Mantisse und Exponent

Zahlen (Realzahlen) werden intern durch ein Zahlwort mit einer festen Stellenzahl repräsentiert (je nach Computer verschieden). Um Zahlen mit Zahlwörtern fester Stellenlänge zu erfassen, werden sie in Mantisse und Exponent zerlegt, wie Bild 2 zeigt.

Rechnet ein Computer mit 7 Dezimalziffern für die Mantisse und 2 für den Exponenten, so ergibt sich bei einer Addition

```
a=1 -->1.00000E+00
b=0.00012345 --> 1.2345E-3
```

Für die Addition vorbereitet:

```
a = 1.000000 E+00
b = 0.001235 E+00 (Rundung!)
```

```
a + b: 1.001235 E+00
```

Bild 3: Fehler bei der Addition von kleinen Zahlen bei siebenstelliger Genauigkeit

der Zahlen a=1 und b=0.0012345 normiert auf 7 Ziffern die in Bild 3 gezeigte Situation. Damit treten durch die Verschiebung der Ziffer bei der Mantisse Stellenverluste auf, die bei immer kleineren Werten immer größer werden.

Daher funktioniert bei HEBAS (mit zwölfstelliger Arithmetik) das in Bild 4 gezeigte Programm nicht. Die Addition der Schrittweite 1E-13 läßt den Wert unverändert.

```
100 FOR X = 1 TO 1.1 STEP 1E-13
120 PRINT I
130 NEXT X
```

Bild 4: Endlosschleife bei kleiner Schrittweite

HEBAS-Tips

HEBAS Version G4 und Autostart von BASIC-Programmen:

Hebas lädt und startet beim Aufruf nun auch ein angegebenes BASIC-Programm. Der Aufruf erfolgt z.B. von CP/M aus mit 'HEBAS TEST', wobei TEST das Basic-Programm TEST.BAS ist. Damit können Autostart-Systeme mittels SUBMIT aufgebaut werden. Es ist auch z.B. 'HEBAS C:Test' möglich, wenn sich TEST.BAS auf Laufwerk C befindet. Achtung: Autostart funktioniert nur für BASIC-Programme im internen Format, d.h. der Autostart geht nicht mit Programmen, die mit SAVE "Name.bas", A als ASCII-File abgespeichert wurden.

Neu bei EHEBAS:

EHEBAS Version 1.1 speichert nun vorhandene Variablen mit Inhalt auf Band ab. So können Programmabläufe sozusagen "eingefroren" werden. Nach dem Laden zeigt HEBAS durch die Meldung "Basic verfügbar" an, daß ein Programm mit Variablen geladen wurde. Nun darf allerdings nicht mit RUN gestartet werden, da sonst die Variableninhalte wieder gelöscht werden. Man kann z.B. mit GOTO 10 starten (mit Zeile 10 beginnt das Pro-

gramm). Wird allerdings das Programm erweitert, so werden die Variablen auch wieder gelöscht.

UP-Date:

- HEBAS auf Version G4: siehe LOOP16
- EHEBAS " " E1.1 Fa. Graf

HEBAS Version G4 und Grafik-Befehle (nur NDR-Rechner):

Grafik-Befehle konnten mittels FLOMON durch ESCAPE-Sequenzen simuliert werden, was jedoch umständlich zu handhaben war. Für den NDR-Rechner mit der GDP wurden nun die entsprechenden Maschinenspracheroutinen in HEBAS Version G4 integriert, damit entfällt auch die ständige BANK-Umschaltung. Die Grafik ist dadurch sehr schnell geworden.

Zur Verfügung stehen folgende Grafik-Befehle:

- MOVETO X,Y: der Cursor wird an den Punkt X,Y gesetzt.
- DRAWTO X1,Y1: eine Linie wird von X,Y NACH X1,Y1 GEZOGEN
- LINE\$ A,X,Y,X1,Y1: A = LINIENART (siehe Tabelle): X,Y: Anfangskordinaten,

0 = durchgezogene Linie
1 = gepunktete Linie
3 = gestrichelte Linie
4 = Kombination

X1,Y1: Endkoordinaten der Linie

d) PAGE X,Y: X = Schreibseite, Y = Leseseite (jeweils 0 - 3)

e) CLR: löscht alle Bildschirmseiten

f) PSET X,Y: ein Punkt wird an X,Y gesetzt

g) LOCATE X,Y: Cursor-Adressierung im normalen Text-Modus

X = Bildschirmzeile < 25,

Y = Spalte < 80

Damit der Cursor bei FLOMON Vers.: 1.6 abgeschaltet wird und die Grafik nicht blinkt, sollte mit PRINT CHR\$(27) + CHR\$(27) + "G" in den Grafik-Modus umgeschaltet werden. Zurück kommt man mit SHIFT A, dann RETURN.

Von EHEBAS zu HEBAS:

Für die Aufsteiger von der SBC3 mit EHEBAS enthält HEBAS nun den Befehl CLOAD. So können Basic-Programme von Kassette auf Diskette überspielt werden. Das Kassetteninterface kann zusätzlich zum Floppy-Controller betrieben werden.

GRAF
computer

Sonderpreisliste

GRAF
computer

Hardware	DM		DM
10515 RFLO Bausatz	95.-	10463 CRT3 Bausatz	150.-
10624 HEXIO2 Platine 2	25.-	10781 RFLO 512K Fertiggerät	398.-
10179 CPU68K Bausatz	129.-	10319 Lochrasterplatine klein	5.-
10339 POW26/22 Bausatz V2	55.-	10318 Lochrasterplatine groß	10.-
10473 FLO1 Platine	10.-	10077 AMPEL Platine	5.-
10527 SYS1 Platine	20.-	10180 CPU68K Fertiggerät	100.-
10412 TAST2	220.-	10516 RFLO Fertiggerät	200.-
10176 CPU68020 Fertiggerät	700.-	10487 HCOPYMAUS Platine mc	20.-
10189 D/A Platine	20.-	10472 FLO1 Fertiggerät	200.-
10367 REL Platine	25.-	60947 IBMKOPP Platine	25.-
10070 AD10/1 Platine	30.-	60484 2x25pol. Stiftleiste gerade	2.-
60898 LOG16 Platine	50.-	10714 CLUT Bausatz	248.-
10514 RAM16 Platine	10.-	40633 IBM Matixdrucker A3 24Nadel 1.	560.-
60312 Bopla-Gehäuse 718	35.-	10715 CLUT Platine	40.-
20015 Graphik Subsystem s/w	700.-	10172 CPU68000 Bausatz 12MHz	200.-
10369 ROA16 Bausatz	35.-	10186 CPUZ80 Platine	20.-
10371 ROA16 Platine	15.-	10543 TERM1 Platine-4	35.-
11030 ROA256/1M+Option Bausatz	128.-	40093 Speicherkarte AT 128KB	300.-
10183 CPU Z80 Bausatz	49.-	10506 PROM Platine	10.-
10407 Stiftleiste 2x20pol.	2.-	10792 ECB NDR Adapter Platine	10.-
10190 D/A Bausatz	80.-	10169 CPU64180 Platine	35.-
10650 Eprom-Floppy Platine	30.-	10175 CPU68000 Platine	30.-
10981 Gehäuse MC4	200.-	10378 SASINDR Bausatz	50.-
40529 IBM Graphikdr. II DIN A4 9Nadel	850.-	10499 OUT1 Bausatz	80.-
10484 HCOPYMAUS Bausatz mc	148.-	10423 Gehäuse zu Tastatur Cherry	29.-
10464 CRT3 Fertiggerät	200.-	10478 FLOSASI Komplettfertiggerät	250.-
10359 RAM64/256 Bausatz	70.-	10334 Musik Platine	5.-
10284 HEXIO Bausatz	70.-	61038 C64-Steuerung Platine	50.-
10477 FLOSASI Komplettbausatz	200.-	10713 CLUT Fertiggerät	250.-
10361 RAM64/256 Platine-5	25.-	10941 ECBKOPP Fertiggerät	99.-
11243 Math. Co-Prozessor 80287	600.-	11202 BUS4A Fertiggerät	200.-
10104 CENTRONICS Einbaubuchse	7.-	30327 Drucker FX 1050 Epson	1.400.-
10316 KEY Platine	15.-	10485 HCOPYMAUS Fertiggerät mc	150.-
10111 BUS2 Platine	20.-	10898 LOG16 Platine + Software 5 1/4", 80 Sp.	148.-
10094 Roboteranpassung Bausatz	45.-	10462 CRT2 Platine	30.-
10538 TERM Farberweiterung Platine	45.-	10167 CPU64180 Fertiggerät	195.-
10332 Musik Bausatz	25.-	10466 CRT3 Platine	20.-
10943 ACRT-Basis Platine 3	150.-	11011 Lautsprecher 8 Ohm, 0,2 Watt	6.-
10342 POW26/22 Platine	5.-	11145 ROA256/1M mc Fertigg.m.Option	148.-
10942 CPU68020 Platine-3	198.-	10979 20MB Festplatte XEBEC	400.-
10123 C64ADAPT Platine	20.-	10382 SBC2 Bausatz	45.-
10261 GDP64K Bausatz	148.-	10512 RAM16 Fertiggerät	90.-
10645 FLO3 Bausatz	198.-	10735 AD/DA Platine-3	60.-
10783 ROB2 Platine-4	30.-	10329 Farbmonitor Sony CPD 1301e	1.299.-
10381 SASINDR Platine	15.-	50041 EINAUS Platine	50.-
10458 CRT1 Platine	20.-	10519 Kleines Schrofengehäuse	65.-
10364 REL-Bausatz	148.-	50008 CRT4A Platine	20.-
10518 RFLO Platine	20.-	10277 HCOPYMAUS Fertiggerät NDR	99.-
10090 BANKSEL Platine	10.-	10659 2x18pol. Buchsenleiste	3.-
10625 IOE2 Fertiggerät	99.-	10644 FLO3 Fertiggerät	198.-
11111 RFLO 512 Platine-3	40.-	10649 Eprom-Floppy Bausatz	130.-
10902 LOG16 Bausatz 5 1/4 80 Sp.	348.-	10379 SASINDR Fertiggerät	80.-
40446 IBM Graphikdr. II A3 9Nadel	1.398.-	10324 Modul f. 5 1/4" Laufwerk	49.-
10074 Ampel-Bausatz	20.-	10127 CAS Platine	15.-
10847 BUS10ECB Platine-1	38.-	10684 SOUND Fertiggerät	80.-
10780 RFLO512K Bausatz	298.-	10475 FLO2MC Bausatz nur FLO-Teil	100.-
10627 IOE Platine-5	25.-	10734 AD/DA Bausatz	498.-
10376 ROB Platine	10.-	10905 LOG16 Fertiggerät 3 1/2" 80Sp	298.-

GRAF
computer

GRAF
computer

	DM		DM
10266 Gehäuse NDR	90.-	10399 Sonderheft NDR Teil 1	10.-
10546 8" Laufwerk, Slimline DD/DS	350.-	40142 DOS 2.1 Handbuch	50.-
60606 Tip 120 Transistor	1.-	61044 Ringmappe leer	5.-
60026 CPU68020 Baustein	350.-	10060 Dokumentation f. CP/M68K	135.-
40404 80MB Festplatte	2.200.-	10380 SASINDR-Handbuch	5.-
10614 Shugart SA851, DS/DD	400.-	10581 Buch Einführung in LOGO	30.-
10673 Gehäuse3 NDR	198.-	40379 IBM PC Bedienerhandbuch	90.-
60176 Quarz 4.000 MHz	2.-	10452 Manual zum CP/M Betriebssystem	40.-
20000 Video-Verteiler	350.-	11009 LOOP-Ausgaben 2 bis 18	40.-
10470 Floppy-Laufwerk 3"	150.-	11112 Buch Kl. Hardware Leitfaden	10.-
10422 TEAC 3 1/2" Laufwerk	298.-	10079 M68000-Familie, Teil 2	50.-
40374 Video-Verteiler für IBM	400.-		
60023 MC68000 12MHz Baustein	120.-	Software	
40071 Speicher 43128 AT01-02	20.-	10249 ZEAT Betriebssystem	140.-
10323 Modul f. 3 1/2" Laufwerk	20.-	11150 WordStar V 3.1 IBM-PC	228.-
10335 Netzteil NE1	148.-	11154 Microsoft Word V 2.0	248.-
10870 Einsteigerpaket	598.-	40392 dBasell Plus deutsch	999.-
40047 IBM Monochrom Bildschirm	360.-	40133 IBM Basic-Compiler	999.-
60587 Teileplatte f. NE2	10.-	10337 Modula-Compiler CP/M68K	400.-
10424 Kabel zu Taster Cherry	12.-	10725 CAD 3 1/2", 80 Sp.	300.-
10536 TERM Farberweiterung	200.-	11151 dBasell, V 2.43 IBM	248.-
10426 Teileplatte f. 5 1/4" Laufw.	5.-	11157 Turbo C deutsch IBM	299.-
40584 Autom. Einzelblatteinzug f. IBM		11065 CP/M Z80 Emulator 3 1/2"	248.-
Matrixdrucker A4, 24 Nadeln	90.-	40414 IBM PC DOS 3.2 deutsch	150.-
40484 IBM Netzteil 63W für PC	98.-	10059 Pascal Listing	5.-
11141 UDC-X Karte	30.-	11224 SOUND f. JADOS 5 1/4" 80 Sp.	45.-
40653 Math. Co-Proz. 80387 16MHz	1.350.-	10200 EBasic V 1.5	20.-
50022 EF9365p	45.-	10216 GOSI-Compiler f. 68000	30.-
10101 Spannungsvers. f. ext. Gerät	3.-	10197 EASS0-3 Grundprogramm	70.-
11038 Steckernetzgerät 5V-/700mA	49.-	10232 Pascal für 68000	30.-
10100 Diodenbuchse 8pol.	1.-	10331 Multiplan 5 1/4" 80 Sp.	99.-
50003 Steckerleiste 3x32polig	5.-	10193 dBasell Datenbank 5 1/4" 80 Sp.	99.-
40728 IRWIN 64MB Streamer	1.999.-	40408 Word deutsch V 4.0	1.200.-
40622 EGA-Wonder	500.-	10873 Schach 3 1/2", 80 Sp. f. JADOS	69.-
10785 Speicher 6116	12.-	10668 WordStar 3 1/2", 80 Sp.	99.-
10298 Flachbandkabel Centronics	19.-	11160 NORTON Utilities IBM	199.-
10392 Reset Taster	3.-	10552 CP/M68K 68020 3 1/2" 80 Sp.	349.-
		60025 CPU68008 Baustein	30.-
		11152 dSoftII, Ergänzung	399.-
Zubehör		10178 CP/M68K f. 68020, 8"	398.-
10438 Video VCC Teil 1 + 2	99.-	11153 MS Multiplan f. MS-DOS, 5 1/4", 80 Sp.	198.-
11185 EINDHEX TOOL Handbuch	20.-	10569 CP/M68K f. 68008 m. 1 Eprom, 8"	348.-
10444 Z80 Aufbauhandbuch	9.-	40189 Lotus 1-2-3 deutsch	800.-
10939 NDR-IBMKOPP Handbuch	10.-	10568 CP/M68K f.08 m. 1 Eprom 31/2"	348.-
10185 CPUZ80 Handbuch	5.-	11248 Modula2 Compiler f. CP/M 2.2	199.-
10122 C64ADAPT Handbuch	5.-	11033 Upd. ERLBASIC 68000	50.-
10270 Gehäuse NDR Handbuch	5.-	10560 CP/M68K m. 2 Eproms, 8",f.68000	349.-
10155 COL256 Handbuch	20.-	10680 RLASIC 3 1/2", 80 Sp.	99.-
10436 Video BETA Teil 1 + 2	100.-		
10513 RAM16 Handbuch	5.-	Komplettsysteme geprüft:	
10396 SER Handbuch	5.-	10997 ACRT Komplettsystem	3.000.-
10647 FLO3 Handbuch	5.-	10871 LOG16 Komplettsystem	2.200.-
10520 Sonderheft CP/M	10.-	11088* 68008 Fertigsystem SN 4F050	2.500.-
60195 Sockel 64polig	3.-	11089* 68008 Fertigsystem SN 4F055	1.900.-
10642 HCOPIY GES-Norm Handbuch	5.-	* Näheres auf Anfrage	
10102 Canon Einbaubuchse 25pol.	2.-		
10311 KEY2 Handbuch	10.-		
10181 CPU68K Handbuch	5.-		
10286 HEXIO Handbuch	10.-		

Die Preise verstehen sich incl. der gesetzl. Mehrwertsteuer.
Angebot freibleibend, solange der Vorrat reicht.

Ulrich Kracker

Nickel-Cadmium Akku Meßgerät mit dem Z80-Einsteigerpaket

(Teil 1 von 2) IST DER AKKU VOLL?

Hierbei handelt es sich nicht etwa um eine neue Abfrage des in Computerkreisen so bekannten Prozessorregisters auf seinen Inhalt hin, sondern es wird die Kapazität eines Nickel-Cadmium Akkus mit dem Einsteigerpaket und etwas Peripherie gemessen. Nickel-Cadmium Akkus (abgekürzt NC-Akkus) findet man heutzutage in fast jedem Haushalt mehr oder weniger häufig an. Sie ersetzen auf preiswerte Art Trockenbatterien für Taschenlampen, Walkman, Cassettenrecorder und so weiter. Ganz besonders häufig werden sie zum Beispiel auch im Modellbau-Hobby eingesetzt.

Vor allem im letztgenannten Anwendungsgebiet ist es interessant zu wissen, wie weit man den Sollkapazitätswerten, die auf den Akkuzellen angegebenen werden, (noch) Glauben schenken kann. Damit NC-Akkus auch auf längere Zeit ihren vollen Nutzen entfalten können, bedürfen sie außerdem einer gewissen Pflege (keine Tiefentladungen und keine Überladungen). Um sich also Klarheit über den Zustand seiner Akkus zu verschaffen, muß eine Meßvorrichtung her!

Wie wurde es bisher gemacht ?

Es sind zum Thema 'Aufladen von Akkus und Kapazitätsprüfung' schon viele bautechnische Vorschläge gemacht worden [1],[2], zum überwiegenden Teil handelt es sich hierbei um Analogschaltungen. Fast alle bisherigen (anspruchsvolleren) Schaltungen zum Aufladen der NC-Akkus arbeiten mit einem konstanten Ladestrom, der eine vorbestimmte Zeitdauer lang eingeschaltet wird (typischer Wert: 1/10 der Kapazitätgröße fließt 14 Std. lang). Daraus resultiert das Produkt:

$$\text{Strom} \times \text{Zeit} = \text{Kapazität (in Ah)}$$

Bei obigen Beispielswerten ergibt sich dann eine Ladungsmenge, die das 1.4-

fache der Nennkapazität ausmacht.

Eine andere Art der Akkuaufladung ist das Laden ohne konstanten Strom auf unbestimmte Zeitdauer. Als Kriterium zum Beenden der Ladephase wird die sich am Akku einstellende Spannung herangezogen. Dabei wird ihre Tendenz überwacht, d.h. steigt die Spannung an, oder fällt sie ab. Wenn sie während des Ladens abfällt, ist das ein Zeichen dafür, daß der Akku voll ist. Diese Technik wird aber vorwiegend bei der sog. Schnellladung angewandt, da hier der Effekt des Spannungsabsinkens am besten bemerkbar ist.

Die bisherige Lösung zum Messen der Akkukapazität beruht darauf, die Zeitdauer zu messen, in der der Prüfling einen voreingestellten Konstantstrom abzugeben in der Lage ist. Die Genauigkeit dieser Verfahren steht und fällt mit der Genauigkeit, bzw. Langzeitkonstanz der eingesetzten Stromquellen.

Im Gegensatz dazu: Das 'NDR-AKKU-Fitness-Center'

Bei meinem Entschluß, eine komfortable, computergesteuerte und vor allem universell anwendbare Test- und Ladestation zu konzipieren, hat mich aber der zusätzliche

Hardwareaufwand für diverse einstellbare Stromquellen und Stromsenken gestört, zumal viele unterschiedliche Akkugrößen (Nennkapazitäten von 40...4000mAh und 1..12 Zellen) unter einen Hut gebracht werden sollten. Daneben forderte ich eine gewisse Langzeitkonstanz. Die Devise lautete also: Möglichst wenig Hardware, dafür mehr Software!

Kein Konstantenstrom, dafür Trapezregel

Das angewandte Kernprinzip beruht darauf, daß sowohl zum Aufladen, als auch beim Messen der Kapazität auf einen voreingestellten Konstantstrom verzichtet wird, dafür aber laufend der fließende Strom gemessen und aufintegriert wird. Zunächst jedoch folgt die Hardware des Prototypes:

Wie sieht die Hardware aus ?

In Abb. 1 ist das Blockschaltbild der Konfiguration zu erkennen: Man benötigt eine AD/Wandlerkarte und einen Zeitgeber auf dem NDR-Bus. Eine Zusatzplatine, die an einer IOE angeschlossen wird, trägt ein einfaches Netzteil und zwei Umschaltrelais, damit der automatische Ablauf gewährt werden kann. Das Netzteil wird von einem LM317 repräsentiert und ist einstellbar von 10...30V, bei einem maximalen Strom von ca. 1A.

Der AD - Wandler

Im Prototyp wird als AD-Wandler die NDR-Baugruppe AD 10*1 eingesetzt. Sie bietet einen Eingangsspannungsbereich von +-10V, der in 1024 Schritte unterteilt wird. Das bedeutet, die Auflösung liegt theoretisch bei 19.5mV pro Digit. Auf dem freien Lochrasterfeld der AD10*1 ist ein Eingangsspannungsteiler aufgebaut, damit auch größere Akkubatterien anschliessbar sind.

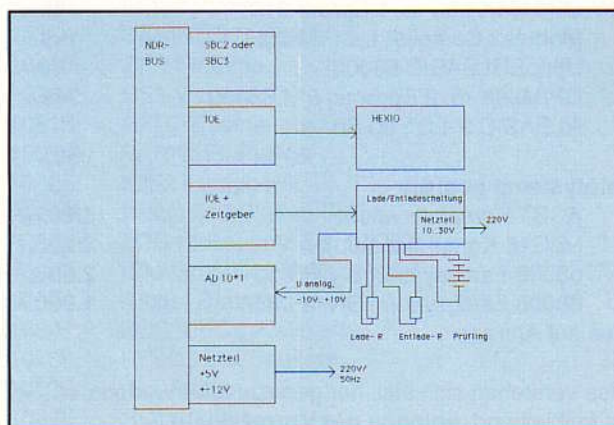


Bild 1: Blockschaltbild der Schaltung

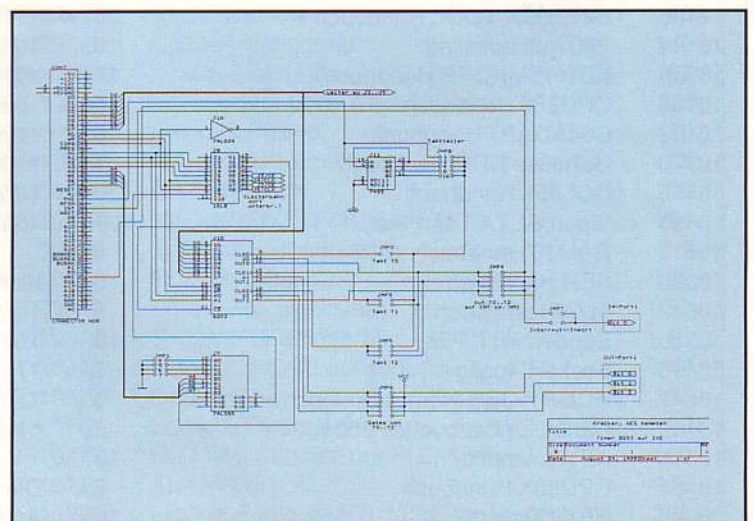


Bild 2: Schaltbild des Timers

Interrupt - Timer

Der Zeittaktgeber (siehe Bild 2) ist ein auf einer IOE beschalteter Timerbaustein 8253 von Intel. Er bietet 3 * 16 Bit-Timer bei einem maximalen Eingangstakt von 2.6MHz, das heißt, der Systemtakt der SBC2 oder der SBC3 von 4MHz muß zuvor mindestens durch Zwei geteilt werden, damit sich der 8253er nicht daran 'verschluckt'.

Der 4MHz-Systemtakt wird mittels eines Binärteilers 7493 durch Zwei, Vier und Acht geteilt, damit stehen drei verschiedene Eingangstaktfrequenzen zur Auswahl. In dieser Anwendung wird dem Timer 8253 ein 1MHz-Takt (1:4) zugeführt.

Die Anschlüsse der drei Einzeltimer (je ein Takteingang, ein Zählerausgang und ein Gate-Eingang) können über ein kleines Jumperfeld individuell verschaltet werden. Die Jumper sind hier so gesteckt, daß der aufbereitete Systemtakt (1MHz) an den Zähleringang des Timers 0 heranführt. Timer 0 und 1 ergänzen sich dann durch eine weitere JumperEinstellung zu einem 32 Bit-Timer (Ausgang von Timer 0 führt an den Eingang von Zähler 1). Damit ist es möglich, eine Verzögerungszeit von einer Sekunde zu programmieren.

Der Ausgang des Timers 1 ist auf die Interruptleitung INT gelegt, sowie an den IN-Port1/Bit0, um das Ende einer Interruptanforderung per Software abwarten zu können. Die I/O-Adressen unter denen die

Auflade/Entladeschaltung

In Bild 3a ist das Ersatzbild des Entladungsteiles aus Bild 3c hervorgehoben. Man erkennt, der Akku wird lediglich über einen (Hochlast-) Widerstand entladen, dessen Wert so bemessen sein muß, daß der Entladestrom auch gegen Ende

nierung des Belastungswiderstandes: Da er zugleich als Messwiderstand fungiert, ist es wichtig, daß seine maximale Verlustleistung um einiges größer (zwei-bis dreimal) sein sollte, als die tatsächlich Auftretende. Der Grund dafür ist, den Effekt der Temperaturdrift möglichst gering zu halten. Um

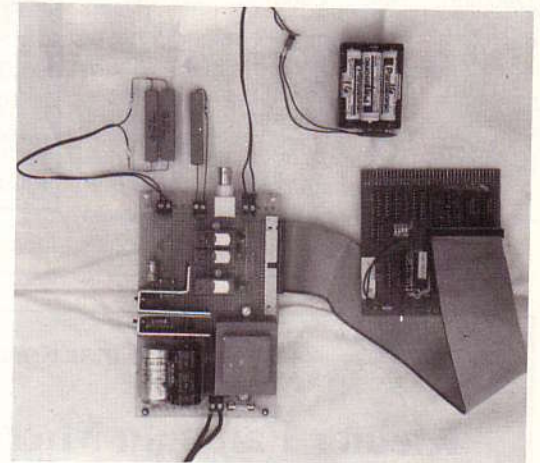
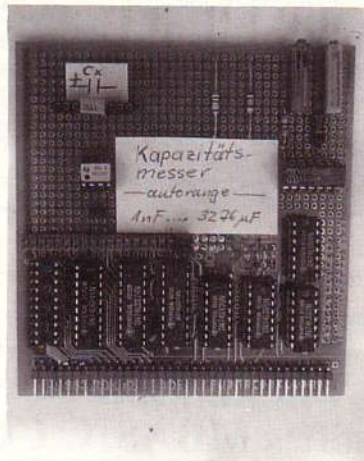


Bild 4: Foto des Prototyps

der Entladung (0.9V pro Akkuzelle) in der gewünschten Größenordnung liegt.

Der Verzicht auf eine Konstantstromsenke hat einerseits den Nachteil, daß sich die Angaben der Hersteller eben schon auf eine Entladung mit konstantem Strom beziehen. Somit ist ein direkter Wertevergleich nicht unbedingt mehr möglich.

in der Anpassung an unterschiedliche Akkus flexibel zu sein, wurde u.a. ein Schraubklemmenpaar zum Anschluß verschiedener Entladewiderstände an den Rand der Zusatzplatine gesetzt (siehe Bild).

Abb. 3b zeigt die resultierende Schaltung aus Abb. 3c, wenn auf Aufladung umgeschaltet wird. Auch hier erfüllt der Strombegrenzungswiderstand zugleich die Aufgabe des Messwiderstandes. Aus diesem Grund gilt bezüglich der Temperaturdrift das Gleiche, wie es für den Entladewiderstand bereits gesagt wurde. Um auch hier flexibel in der Anpassung an verschiedene Akkugrößen zu sein, ist wieder ein Schraubklemmenpaar zur Aufnahme des Ladestrombegrenzers am Platinenrand platziert worden.

Vor der Einstellung der Spannung am Netzteil sollte folgende Überlegung stehen: Zu Beginn der Aufladung fällt am NC-Akku eine geringere Spannung ab, als gegen Ende der Ladung. Am Begrenzungswiderstand sind die Verhältnisse gerade umgekehrt. Das bedeutet, daß die zu messende Spannung am Begrenzungswiderstand zu Beginn der Aufladung den maximalen Eingangsspannungsbereich des verwendeten AD/Wandlers nicht übersteigen darf. Man kann überschlagsmäßig rechnen:

$$\text{Einzustellende Netzteilspannung} = 0.9V \cdot \text{Anzahl der Zellen} + \text{max. Eingangsspg des AD/Wandlers} - 10\% \text{ Sicherheitsabstand.}$$

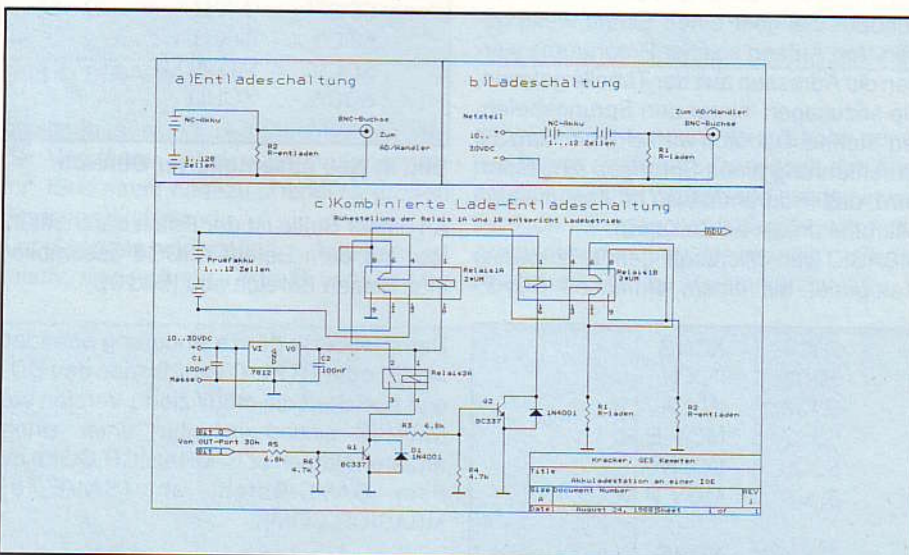


Bild 3: Schaltbild der Auf- u. Entladeschaltung

Register des Bausteines 8253 erreichbar sind, wurden auf (Basisadr. der IOE + 4) ... (Basisadr. der IOE + 7) festgelegt. Da die ursprüngliche Adressdecodierung auf der IOE nur unvollständig ist, wurde dieser Umstand mit einem PAL-Baustein behoben (Vgl. Bild 2).

Andererseits entspricht diese Entladungsart eher den realen Gegebenheiten im Alltag; und um die Alterung oder den 'Memoryeffekt' eines speziellen Akkus nachzuweisen ist es ja nur wichtig stets die gleiche Messmethode anzuwenden. Noch ein Wort zur Auswahl und Dimensio-

Nachdem weiterhin die am Widerstand abfallende Spannung gegen Ende der Ladung geschätzt wurde (= Netzteilspannung - Anzahl der Zellen * 1.5) kann aus diesem Ergebnis die Größe des Begrenzungswiderstandes in Ohm ermittelt werden. Aus den obigen Überlegungen resultieren zum Beispiel die Standardwerte:

Akkugr. in mAh	Netzteilsp.	Ladewid.	Entlade- wid.
4/500	13V	47Ohm	11Ohm
5/1200	14V	47Ohm	50Ohm
6/500	16V	56Ohm	11Ohm
7/1200	21V	47Ohm	50Ohm
8/1200	21V	47Ohm	50Ohm

Literatur:

[1] Zeitschrift 'Flug-und Modelltechnik', Verlag für Technik und Handwerk, Baden-Baden:

- R.W. Nessel: Elektronik-Schnellader, Hefte 1/2/3/81
- W. Papendiek: Akku-Kapazitäts-Prüfer, Heft 6/81, Seite 523
- H. Günther Schlicker: Und noch ein Ladegerät..., Heft 1/82, Seite 59. Nachtrag im Heft 3/82, Seite 206.
- Hans Luz: Programmierbares Lade- und Entladegerät für NiCd-Akkumulatoren, Heft 8/83, Seite 708
- K.-P. Egler: Akku-Kapazitäts-Test, Heft 12/83, Seite 1092

[2] Zeitschrift Modell, Neckar-Verlag, Villingen-Schwenningen:

- K. Zagel: Mikroprozessorgesteuertes Ladegerät, Heft 2/83, Seite 112.

Peter Kesa

Kleiner Patch an Microsoft's MBASIC.COM

Wer hat sich das nicht schon oft überlegt: Jetzt, wenn der Drucker zum Bildschirm parallel läge Was könnte man da nicht alles machen? Es ist soweit. Wer seine CP/M-Programme kennt und keine Angst davor hat, Microsoft's MBASIC.COM-Datei zu verändern, kann sich jetzt diesen Wunsch erfüllen.

CP/M-Programme, welche die Monitor-Ausgabe über die BDOS-Funktion 2 bewerkstelligen, haben den Vorteil, daß durch eine kleine Änderung in der BIOS-Einsprungtabelle zur Monitorausgabe der Drucker parallelgeschaltet werden kann.

	.	
	.	
JMP	.	CONIN
CALL	.	CONOUT
JMP	.	LISTE
	.	
	.	

Bild 1: BIOS-Einsprungtabelle

Betrachtet man sich diese Einsprungtabelle mal näher, so erkennt man, daß nach dem Sprung zur Monitorausgabe direkt der Sprung zur Druckerausgabe folgt. Ändert man nun den Sprungbefehle zum Monitor in einen Call-Befehl um, so wird erst das Zeichen zum Monitor ausgegeben und danach erfolgt die Ausgabe desselben Zeichens auch auf dem Drucker. Die BIOS-Einsprungtabelle bekommt also folgende Form (Bild1).

Wird der Call-Befehl wieder durch einen Sprungbefehl ersetzt, so ist die Ausgabe

auf den Drucker wieder rückgängig gemacht worden.

Es gibt aber auch CP/M-Software, die nicht die Einsprungtabelle des BIOS benutzen, sondern vielmehr einen Schritt weitergehen. Am Anfang solcher Programme werden die Adressen aus der Tabelle gelesen, die sozusagen 'hinter den Sprungsbefehlen' stehen. Dadurch wird erreicht, daß die Durchführung eines Sprunges eingespart wird, das Programm also im Ablauf einige Mikrosekunden an Zeit spart. MBASIC von Microsoft gehört zu dieser Kategorie. Mit einem einfachen POKE-

5DC4	XCHG
5DC5	INX H
5DC6	INX H
5DC7	MOV E,M
5DC8	INX H
5DC9	MOV D,M
5DCA	XCHG
5DCB	SHLD 41B8

Bild 2: Speicherinhalt

Befehl kann man den Drucker also nicht zum Monitor parallelschalten.

Führt man aber einen kleinen Patch des Programmes durch, so ist diese Möglichkeit auch unter MBASIC möglich. Im Fol-

genden soll diese Programm-Modifikation beschrieben werden.

Mit Hilfe des Debuggers DDT wird das MBASIC.COM in den Speicher geladen. Mit dem Befehl L5DC4 läßt man sich den folgenden Speicherinhalt (Bild 2) auflisten bzw. disassemblieren.

5DC4	XCHG
5DC5	INX H
5DC6	MOV E,L
5DC7	INX H
5DC8	INX H
5DC9	MOV D,H
5DCA	XCHG

Bild 4: Neu assemblierter Bereich

An dieser Stelle ist der Patch durchzuführen. Mit dem Befehl A5DC4 assembliert man diesen Bereich neu (Bild 3).

Damit ist diese ganze Änderung beendet. Mit G0 oder CTRL-C verläßt man den DDT und speichert die modifizierte Version von MBASIC sicherheitshalber unter einem anderen Namen (z.B. MBASICR.COM) mit dem SAVE-Befehl ab (SAVE 95 MBASICR.COM).

Zum Testen ruft man MBASICR auf und gibt den Befehl (Drucker einschalten nicht vergessen) POKE &HEA0C,&HCD ein. Danach müssten alle Ausgaben auf Monitor auch auf dem Drucker erscheinen (vorausgesetzt, er ist funktionsbereit). Mit dem Befehl POKE &HEA0C,&HC3 wird der Drucker wieder von der Ausgabe ausgeschlossen.

Stephan Glatthaar

D/A - WANDLER

GANZ STARK

Die Tatsache, daß der D/A - Wandler nur eine maximale Ausgangsspannung von 2,5V liefert, hat uns angeregt, uns diesbezüglich etwas einfallen zu lassen. Um größere Ausgangsspannungen zu erhalten, muß der Ausgang des Umsetzers verstärkt werden. Mit der folgenden einfachen Schaltung kann der Spannungsbereich bis +10V erhöht werden. Außerdem lassen sich auch negative Spannungen abgreifen.

Das Eingangssignal der Schaltung (vgl. unter anderem auch als Spannungsfolger bezeichnet. Der zweite OP arbeitet als invertierender Verstärker. Die Eingangsspannung wird

eines OP sehr viel größer als das Verhältnis R_0/R_1 , dann gilt (Vgl. Bild 2):

$$U_a/U_e = -R_0/R_1$$

Um das invertierte Signal wieder auf positive Werte zu bringen, wird es mit OP3 ein zweites Mal invertiert. Auch hier kann wieder verstärkt werden, wie es schon oben beschrieben wurde. Zusätzlich kann am invertierenden Eingang, der praktisch Massepotential besitzt, über einen Widerstand eine weitere Spannung angelegt werden. Die Schaltung eignet sich also zum Addieren von Spannungen (Bild 3) und wird auch 'Umkehraddierer' genannt. Mit der Knotenpunktregel läßt sich die Ausgangsspannung berechnen:

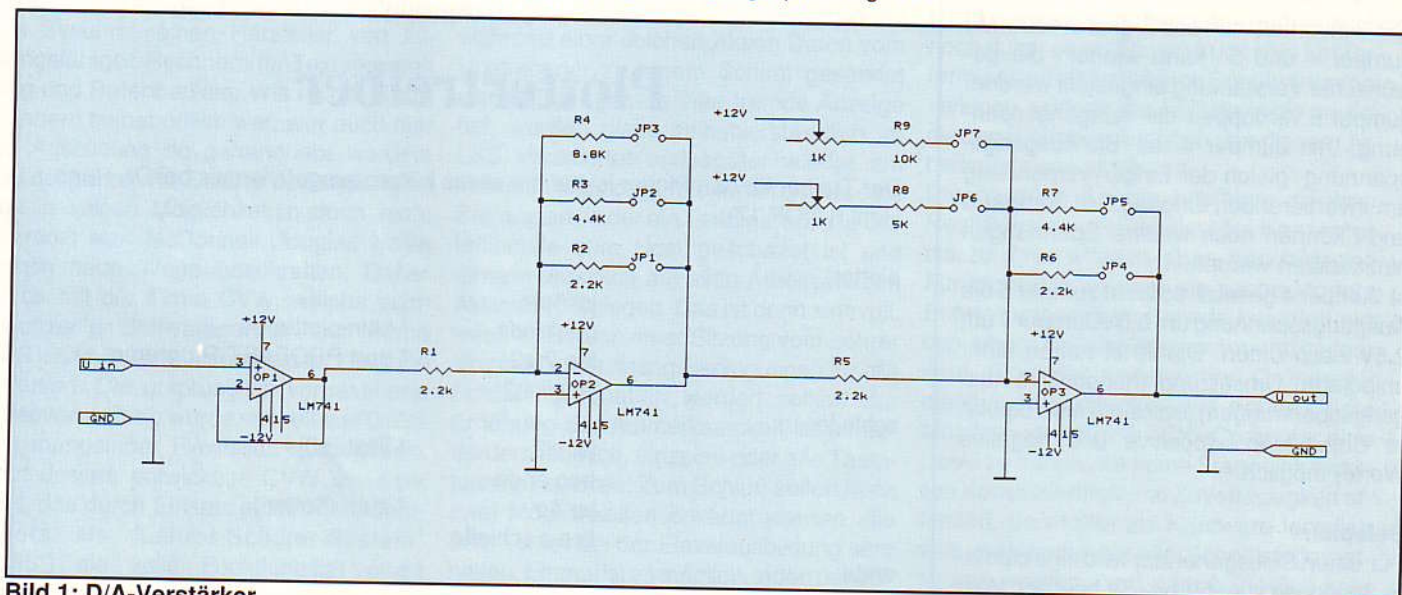


Bild 1: D/A-Verstärker

besitzt in diesem Fall als nichtinvertierender Verstärker mit dem Verstärkungsfaktor Eins einen nahezu unendlich großen Eingangswiderstand. Der Ausgangswiderstand dagegen ist relativ niederohmig. Diese Schaltung wird

dabei verstärkt und würde auf hohe negative Werte ansteigen. Da jedoch der Ausgang über einen Widerstand auf den invertierenden Eingang zurückwirkt, wird die Ausgangsspannung gleichzeitig wieder verkleinert. Ist der Verstärkungsfaktor V

$$0 = U_a/R_0 + U_{e1}/R_1 + U_{e2}/R_2 + \dots + U_{en}/R_n$$

$$\rightarrow -U_a = (R_0/R_1)U_{e1} + (R_0/R_2)U_{e2} + \dots + (R_0/R_n)U_{en}$$

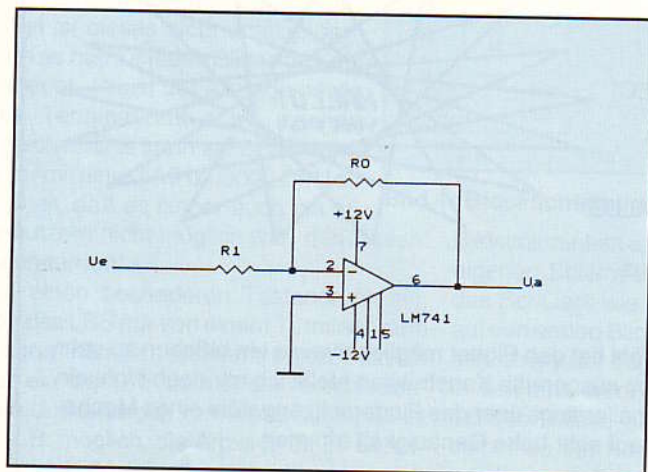


Bild 2: Invertierender Verstärker

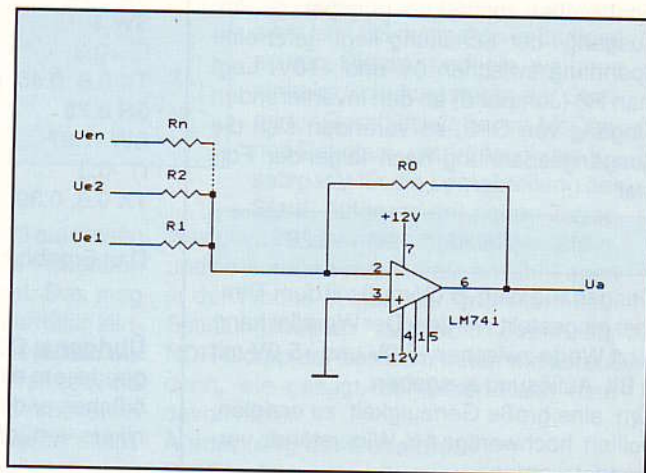


Bild 3: Invertierender Addierer

Der Eingang in unserer Schaltung greift das analoge Ausgangssignal von Kanal 1 oder Kanal 2 ab. Da der DA/-Wandler nur mit maximal 4uA belastet werden darf, mußte der Impedanzwandler eingesetzt werden, der eine Stromstärke von 20 mA erlaubt.

An OP2 kann mittels Jumper die Verstärkung eingestellt werden. Jumper 1 verbindet den Ausgang von OP2 über den Widerstand R2 mit dem invertierenden Eingang. Da $R2 = R1$ (Vorwiderstand), ist die Ausgangsspannung gleich der Eingangsspannung. Mit Jumper 2 kann die Ausgangsspannung verdoppelt werden (0..5V), da auch R3 doppelt so groß ist wie R1. Jumper 3 vervierfacht die Ausgangsspannung (0..10V).

OP 3 invertiert nun das Signal erneut. Mit Jumper 4 und 5 kann wieder die gewünschte Verstärkung eingestellt werden. Jumper 5 verdoppelt die Ausgangsspannung, mit Jumper 4 ist die Ausgangsspannung gleich der Eingangsspannung am invertierenden Eingang. Mit Jumper 6 und 7 können noch weitere Spannungen hinzuaddiert werden. |

st Jumper 4 gesetzt, so zieht Jumper 6 die Ausgangsspannung um 5.0V Jumper 7 um 2.5V nach unten. Damit ist neben einer unipolaren Umsetzung (nur positive Ausgangsspannungen) jetzt auch eine bipolare Umsetzung (positive und negative Werte) möglich.

Beispiel:

Für einen Sinusgenerator wird eine bipolare Spannung von -5V bis +5V benötigt. Wird also zB. an OP2 das vom D/A-Wandler kommende Signal verdoppelt (Jumper 2), so liegt der Ausgang von OP2 auf -5V. Vervierfacht man an OP2 das vom Wandler gelieferte Signal (Jumper3), so besitzt der Ausgang von OP2 ein Potential von -10V..0V, je nach Aussteuerung des Wandlers. OP3 invertiert das Signal, soll jedoch nicht weiter verstärken (Jumper 4). Am Ausgang der Schaltung liegt jetzt eine Spannung zwischen 0V und +10V. Legt man R8 (Jumper 6) an den invertierenden Eingang von OP3, so verändert sich die Ausgangsspannung nach folgender Formel:

$$-U_a = (R_6/R_8) * 12V = -5.28V$$

Der genaue Wert -5.0V muß mit dem Trimmer eingestellt werden. Der Wandler kann jetzt Werte zwischen -5.0V und +5.0V mit 8 Bit Auflösung ausgeben.

Um eine große Genauigkeit zu erzielen, sollten hochwertige 1% Widerstände verwendet werden.

Aussteuerungsbereich:

0...+2.5V
0...+5.0V
0...+10V
-2.5...+2.5V
-5.0...+5.0V
-10...+10V

gesetzte Jumper:

1;4
2;4
3;0
2;4;7
3;4;6
3;5;6

Bild 4: Aussteuerbereiche

Tip

von Curt Michaelis

Plottertreiber

Der Treiber für den Plotter ist bei mir etwas kürzer ausgefallen als bei Dr. Hans Hehl (LOOP 17):

plotter:

jsr 1sts
geq ende
lea ?,a0
jsr ciinit2

* kein plotter da
* dort PROFAST-Programm

schleife:

jsr ci2
beq ende
jsr 1o
bra.s schleife

* liest (a0)+
* zum Plotter

ende:

RTS

An der aktuellen Startadresse muß dazu das PROFAST-Programm (der sehr einfache Treiber erlaubt keine Variablen und keine Kommentare; typischerweise wird man sich allerdings wohl für die Bemaßung in Millimetern entscheiden):

ZO 0.6
AC 0.6, 0.40, 0.15, 0.11, 90, 2610, 148.235
SS 1.25
SW 1
IT +0.3
TX 0.6, 0.40, 0, M, HALLO!
SS 0.75
SW -1.67
IT -0.3
TX 0.6, 0.39, 180, M, HALLO!



Das Ergebnis sieht so aus:

Übrigens: Der Doktor Hehl hat den Plotter möglicherweise ein bißchen zu sehr gepriesen; eine technisch ausgereifte Konstruktion stelle ich mir doch wohl ein bißchen anders vor. Wer allerdings über das Fingerspitzengefühl eines Mechanikers verfügt, kann ihn auf sehr hohe Genauigkeit trimmen.

Volker Wiegand

NDR-Computer hilft beim Lehrbetrieb

Als die Landeshauptstadt Stuttgart für einige ihrer Ämter Computer für Textverarbeitung anschaffte, war von Anfang an klar, daß eine gute Ausbildung der Fachkräfte von entscheidender Bedeutung für die Auswahl des Herstellers sowie für die spätere Systembenutzung war. Also wurde Ausschau nach einem System gehalten, das ein gutes Ausbildungskonzept beinhaltete. Die Wahl fiel auf die Firma McDonnell Douglas Information Systems, einen Hersteller von leistungsfähigen Rechnern für Textverarbeitung und Datenbanken. Wie es bei Großrechnern bisher üblich war, war auch hier die Ausbildung so gehandhabt worden, daß der Lehrer von Schüler zu Schüler ging und in seinen Möglichkeiten doch recht begrenzt war. McDonnell Douglas wollte jedoch neue Wege beschreiten. Daher wurde mit der Firma CVW, welche auch gleichzeitig Softwarepartner der Firma GES ist, über eine Lösung dieser Aufgabe diskutiert. Die ursprünglich vorgesehene

Im Folgenden wollen wir über eine Anwendung des NDR-Computers berichten, bei der dieser in entscheidendem Maße dazu beiträgt, die Ausbildung von Stenotypistinnen an einem Großcomputer effizient zu gestalten. Durch eine Art "Sprachlabor für Computer" kann der NDR-Computer dem Lehrer wirksame Hilfen im Lehrbetrieb geben.

deren Tastatur zu sperren, um etwa einem schwachen Schüler gezielt von einem stärkeren helfen zu lassen. Man kann auch eine Anzeige auf alle Schirme geben, etwa wenn der Lehrer einen Ablauf erklärt. Wenn während einer solchen Aktion Daten vom Hostrechner zu einem Schirm gesendet werden, der gerade eine fremde Anzeige hat, werden diese selbstverständlich im LSS verarbeitet und später wieder am Terminal angezeigt. Weiterhin gibt es eine Betriebsart, in der ein Terminal auf alle Datenkanäle zum Host geschaltet ist und diesem praktisch auf allen Arbeitsplätzen Aktivität vorspiegelt. Das ist dann sinnvoll, wenn zu Beginn einer Sitzung vom Lehrer die gleichen Ausgangsbedingungen für alle Schüler geschaffen werden sollen. Zur

aber wohl, was wir "Direktverbindung" nennen. Dabei wird der Lehrer auf einen Schülerarbeitsplatz "aufgeschaltet". Der Lehrer hat also wie beim Beobachten das Bild des Schülers vor Augen. Aber nun wirkt seine Tastatur ebenso

wie die des Schülers auf den Schülerkanal. Der Lehrer kann also dem Schüler direkt und ohne den "Mindestabstand" zu verletzen, Hilfen geben oder ihn korrigieren. Nicht genug damit, kann der Lehrer auch noch durch einen Tastendruck sein Lehrterminal auf ein beliebiges Schülerterminal verlegen, wonach das LSS dann die privilegierten Eingaben nur noch von diesem Arbeitsplatz aus akzeptiert.

Die gesamte Anlage benötigte zu ihrer Realisierung vom Erstellen des Konzeptes bis zu ihrer jetzigen oben geschilderten Funktionalität weniger als sechs Monate Entwicklungszeit und wurde komplett nur von einem Entwickler geplant und aufgebaut. Einer der wesentlichen Gründe für diese Schlagkräftigkeit war sicherlich die

den mangelnder Flexibilität aufgegeben. Statt dessen entwickelte CVW ein Konzept, das durch Einsatz eines NDR-Computers als "Lehrer-Schüler-System" ("LSS") die volle Funktionalität eines Sprachlabors wie in der Schule oder Universität erreicht.

Das Prinzip wird aus der Abbildung ersichtlich. Der NDR-Computer wird in die Datenleitungen zu den Terminals eingeschleift und verhält sich zunächst neutral, d.h. transparent. Die Terminals "merken" also nichts vom Vorhandensein des LSS. Allerdings ist dieses nicht untätig, sondern es hält für jeden Bildschirm ein getreues Abbild der Anzeige sowie des Terminal-Status im eigenen Speicher. Das kann ein NDR-Computer mit einer 8 MHz 68000 CPU so schnell, daß es bisher auch mit 16 Benutzern nicht möglich war, den "Spion" zu enttarnen!

Auf einen besonderen Tastendruck hin, den das LSS nur von einem Terminal, dem "Lehrer" nämlich, annimmt, erhält letzterer nun ein Menü, in dem er die verschiedensten Umstellungen vornehmen kann. So ist es z.B. möglich, die Anzeige eines Benutzers auf die eines anderen oder auch mehrerer zu schalten und dessen oder

Berdem möglich, einzelne oder alle Tastaturen zu sperren. Zum Schluß sollen noch zwei Möglichkeiten erwähnt werden, die dem Lehrer bei der Einzelausbildung sehr helfen. Einmal ist es möglich, jeden beliebigen Schüler zu beobachten, ohne daß dieser es merkt. Der Lehrer holt also den

Möglichkeit, in der NDR-Computer eine Basis zu haben, die keine Fragen hinsichtlich Kompliziertheit und Zuverlässigkeit offenließ. So mußte als Hardware lediglich eine mehrfache serielle Schnittstelle entwickelt werden - auf dem NDR-Bus kein Thema (diese Schnittstelle mit 4 Kanälen sowie OS-9 Treiber werden noch in diesem

Jahr allgemein zur Verfügung stehen). Damit war die Hardware bis auf eine Verstärkung des Netzteil für -12 Volt fertig. Es konnte also von Anfang an mit Software-Entwicklung begonnen werden.

Seit über einem halben Jahr versteht ein LSS nun schon in Stuttgart seinen Dienst, und ein weiteres steht im Schulungsraum der Stuttgarter Geschäftsstelle von McDonnell Douglas - ursprünglich als Ersatzgerät für die Lernabteilung der Stadt. Aufgrund der hohen Zuver-

lässigkeit ist das aber längst überflüssig, und so werden beide Geräte an fünf Tagen in der Woche eingesetzt. Und wenn kein Schulungsbetrieb stattfindet, kann der NDR-Computer trotzdem weiter mitlaufen, denn, wie gesagt, bemerken kann man davon nichts...

Anmerkung der Redaktion:

Der Entwickler des LSS ist Volker Wiegand
- Ehre, wem Ehre gebührt -

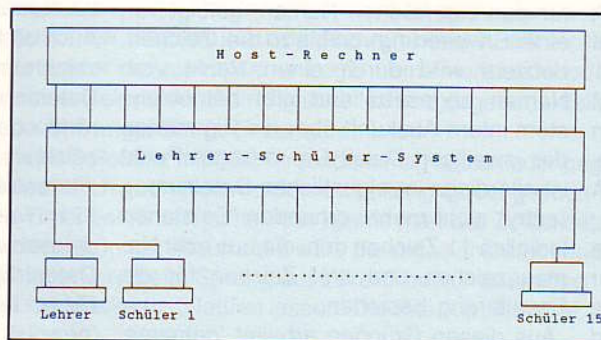


Bild 1: Blockschaubild des LSS

Bildschirminhalt eines Schülers auf seinen eigenen Schirm und sieht alle Ausgaben des Schülers wie dieser selbst. Das mag auf den ersten Blick etwas hinterhältig klingen, aber jeder frage sich einmal selbst, wie er sich fühlt, wenn ihm der Lehrer über die Schulter blickt; also ich bekomme dann immer so ein Kribbeln und werde etwas nervös...

Eine der interessantesten Betriebsarten ist

Günter Renner

Eine gefährliche Operation

Jetzt wird es auch interessant für alle Leser, die nur PC-Benutzer sind. Was Sie schon immer über MS-DOS wissen wollten... vielleicht steht es hier. Denn wenigstens ein kurzer Blick in die Tiefen dieses Systems muß sein. Nach dem der Bootsektor nun erstellt ist, müssen noch die legendären FATs und das Hauptinhaltsverzeichnis auf der Diskette angelegt werden.

Im letzten Beitrag wurde die Anordnung dieser Elemente auf der Diskette beschrieben. Weil die Einträge aber im RAM angelegt werden, ehe sie auf die Disk geschrieben werden können, sollte man sich auch einmal auf eine Belegung des Speichers einigen. Die Lage des Handlers ergibt sich bereits aus dem Bootsektor. Es fehlen nur noch die FATs und das Inhaltsverzeichnis. Für ein System, dessen Grundprogramm auf \$0 liegt, könnte die Belegung folgendermaßen sein:

Bootsektor:	\$9800
Handler:	\$9a00
FAT1:	\$a600
FAT2:	\$aa00
Inhaltsverz.:	\$ae00
	bis \$bbff

Zu diesen Adressen muß ggf. die Basis des Grundprogramms dazuaddiert werden. Dieser RAM wird später nur vom Handler benutzt und darf deshalb keine anderen Daten enthalten. Wie man sieht, ist man mit zwei 6264-Chips auf der ROA hinter dem Grundprogramm erst einmal dabei. Für den Stack bleibt dann genau 1 kB übrig. Hinzu kommt dann noch RAM für den Filepuffer, der 64 kB groß sein sollte und beliebig placiert werden kann. Selbstverständlich darf es auch die TPA von CP/M sein...

Für das Erstellen einer neuen Diskette steht im Handler eine Menüfunktion zur Verfügung. Sie heißt 'newdisk' und ist aus dem Listing ersichtlich. Selbstredend müssen Bootsektor und Handler bereits im RAM geladen sein. Dann sind noch die Einträge in den FATs und im Hauptinhaltsverzeichnis anzulegen. Zum Schluß wird alles einfach auf die Disk geschrieben. Damit ist der Datenträger in PCs gebrauchsfertig.

Weil es sich dabei um eine sehr gefährliche Operation handelt, mit der man sehr leicht Diskinhalte unwiderbringbar zerstören kann, muß man zunächst einmal bestätigen, daß man es ernst meint und 'J'

oder 'j' auf der Tastatur drücken. Spätestens hier aber sollte die richtige Disk im Laufwerk sein!

Eine Chance hat man dennoch. Denn MS-DOS kennt sogenannte Datenträgerkennsätze. Das ist nichts anderes als ein Name für die ganze Diskette als solche, der im Hauptinhaltsverzeichnis eingetragen wird. Zur Eingabe eines solchen Namens fordert das Unterprogramm 'getname' auf. Sollte der Benutzer zu früh 'ja' gesagt haben, hilft jetzt allerdings nur noch der Resetknopf. Alles andere führt über kurz oder lang zum Neubelegen der Diskette.

'Getname' erlaubt die Eingabe von bis zu 14 Zeichen. Es werden jedoch nie mehr als die ersten 11 davon verwendet. Wer der Disk keinen Namen geben will, drückt einfach nur 'CR', womit die Eingabe, wie üblich, abgeschlossen wird.

Etwas schwieriger wird es später bei den Dateinamen. Auch diese werden mittels 'getname' eingegeben. In MS-DOS bestehen diese aus maximal acht Zeichen für den eigentlichen Namen, gefolgt von einer Erweiterung von bis zu drei Zeichen. Letztere wird durch einen Punkt vom Namen abgesetzt und gibt betriebssystem intern Auskunft über die Eigenarten der jeweiligen Datei. Der besagte Punkt ist allerdings im eigentlichen Dateieintrag (entry) nicht mehr vorhanden. Es stehen lediglich 11 Zeichen drin, die aus acht Namenszeichen und drei Zeichen für die Erweiterung bestehen.

Aus diesen Gründen arbeitet 'getname' mit zwei Puffern. In den einen wird die Eingabe zunächst einfach einmal eingelesen. Nachdem sie mit 'CR' abgeschlossen ist, wird die Zeichenfolge in den eigentlichen Namenspuffer übertragen, ohne den Punkt, versteht sich. Außerdem wird der Namensteil ggf. mit Leerzeichen aufgefüllt, die drei Zeichen der Erweiterung sind stets die letzten in der Zeichenfolge. Endlich steht der Name so im Puffer, wie er für den Eintrag übernommen werden kann.

Teil1: Der verrückte Bootsektor
Loop 17

Teil 2: Eine gefährliche Operation
Loop 19

Teil 3: Log.-physikalisches Verwirrspiel
Loop 20

Teil 4: Sage mir, was Du hast
Loop 21

Teil 5: Jetzt wird gelesen
Loop 22

Teil 6: Ärger mit Ä
Loop 23

Jeder Eintrag besteht insgesamt aus 32 Bytes. Das gilt für den Kennsatz genauso wie für jeden Dateieintrag. Er beginnt mit dem Namen und ggf. der Erweiterung. Darüber hinaus sind weitere Elemente vorgesehen. Für unseren Handler spielen vor allem die Folgenden eine Rolle (bezogen auf den Anfang des Eintrags):

\$0 - \$7	Name
\$8 - \$a	Erweiterung
\$b	Attribut
\$1a - \$1b	Startcluster
\$1c - \$1f	Länge/Bytes

Das erste Zeichen des Namens (\$0) hat eine Schlüsselfunktion für den gesamten Eintrag. Als Beispiele seien hier nur erwähnt:

\$00	unbenutzter Eintrag
\$e5	gelöschter Eintrag

Auch das Attributbyte (\$b) dient der Kennzeichnung des Eintrags. Damit können Dateien versteckt, als zum System gehörend oder als Unterverzeichnis gekennzeichnet werden und vieles mehr. Uns interessieren zunächst einmal die Werte \$21, was bedeutet, daß die Datei nur gelesen werden kann, sowie \$08, was den Datenträgerkennsatz bezeichnet.

Diese beiden Attributwerte werden von 'newdisk' für das Anlegen des Kennsatzes und des Eintrags für den Handler verwendet. Gewöhnliche Dateien haben \$20 als Attribut. Weitere Einträge, die hier nicht näher erwähnt werden sollen, enthalten Datum und Uhrzeit.

Ein Dateieintrag wäre nun aber vollkommen nutzlos, wenn er nicht irgendwie dorthin zeigen würde, wo die gesuchten Daten tatsächlich auf der Disk stehen. Deshalb ist auch dafür ein Feld vorgesehen. Es enthält den sogenannten Startclu-

ster. Auf eine Weise, die später noch eingehender erläutert werden muß, zeigt dieser auf den Beginn der Datei. Der Eintrag enthält dann noch die Angabe der Länge der Datei in Bytes. Es drängt sich nun aber die Frage auf, wie man vom Dateianfang beginnend durchgehend alle Sektoren der Diskette erhält, die die gesuchte Datei enthalten. Diese Frage führt hin zu einem Gebilde, das vielen PC-Nutzern gar unheimlich ist, auch wenn sie es nicht so recht zugeben wollen: der FAT.

Sieht man die Dinge grundsätzlich, so hat der Startcluster zwei Bedeutungen. Zum einen bestimmt er die beiden ersten Sektoren (= 1kB) der Datei. Zum anderen zeigt er aber auch auf eine ganz bestimmte Position in der FAT. Unter dieser Position läßt sich aus der FAT ein Wert auslesen, der über alles Weitere Auskunft gibt. Ist der Wert z. B. größer als \$ff7, war der erste Cluster auch schon der letzte: das Ende der Datei ist damit gekennzeichnet. Ist der Wert kleiner, aber ungleich Null, so geht es bei genau diesem Cluster weiter. Derselbe Wert zeigt natürlich auch wieder auf die nächste Position in der FAT, bis man zuletzt doch auf die Endemarkierung (> \$ff7)

stößt. Alle unbenutzten FAT-Positionen enthalten den Wert \$000.

Die Werte der FAT-Einträge sind übrigens nicht willkürlich dreiziffrig angegeben. Tatsächlich besteht ein Wert aus zwölf Bits. Und das macht die Sache so unverständlich. Denn ansonsten hat man es doch immer mit Bytes, Worten oder Langworten zu tun. Hinzu kommt, daß Worte bei den Prozessoren der 86er Serie byteweise anders abgelegt werden. Ein Beispiel:

Wert:	86er:	68er:
\$1234	34 12	12 34
\$12345678	78 56 34 12	12 34 56 78

Diese Besonderheit erschwert nicht nur die manuelle Lesbarkeit der FAT. Sie gilt z. B. auch für die Werte des Startclusters und der Dateilänge.

Weil nun 'newdisk' für den Handler einen Eintrag im Hauptinhaltsverzeichnis anlegt, muß auch ein entsprechender Eintrag in den FATs vorhanden sein. Er besteht aus acht Bytes, die die Werte \$fd \$ff \$ff \$03 \$40 \$00 \$ff \$0f haben. Dieser Eintrag entstand nicht am Schreibtisch. Er wurde auf einem PC durch

das Laden einer knapp 3 kB großen Datei erzeugt.

Das erste Byte enthält eine Kennung des verwendeten Formats, die auch im BIOS-Parameterblock zu finden ist. Und daß im Inhaltsverzeichnis unter 'Startcluster' des Handlers der Wert zwei steht, läßt die Vermutung aufkommen, daß zwei weitere Ziffern im FAT-Eintrag zwei weitere Cluster bezeichnen könnten, sodaß alles zusammen drei kB ergibt, also soviel, wie unser Handler auf der Disk belegt. Und die 'f's sind verdächtig, das Ende davon zu markieren. Mit etwas Glück könnten also die Cluster zwei bis vier tatsächlich unseren Handler beherbergen.

Für diesen ist der FAT-Eintrag übrigens lebenswichtig. Er würde sonst demnächst von MS-DOS überschrieben werden, und zwar ohne Rücksicht darauf, daß er im Inhaltsverzeichnis steht!

Der nächste Teil wird sicher etwas kürzer. In ihm ist u. a. noch etwas zur Zählweise der Sektoren und Cluster zu sagen. Daneben werden weitere Programme des Handlers vorgestellt, die zum Schluß in ein Bedienmenü einbezogen werden. (Listing - siehe Rückseite)

Berichtigung zu LOOP16:

"Jump auf Interupt"

von

Curt Michaelis, Jahnkeweg 25, 2000 Hamburg

Nachdem meine Schreibmaschine wieder heil ist, muß ich erst einmal einen fatalen Fehler berichtigen: Ich habe in LOOP 16 behauptet, daß der 68000 im Supervisor-Modus die Interrupt-Verarbeitung zurückstelle. Dies ist leider nicht so; Priorität haben nur Reset, Busfehler, Adressfehler und Trace-Modul aber nicht Supervisor-Betrieb! Dies hat leider zur Folge, daß man den unechten Interrupt nicht aussperren kann, der dann auftritt, wenn während der Bearbeitung der Interrupt-Routine zu Level 5 auch noch Level 2 angefordert wird. (Im umgekehrten Fall gibt's überhaupt keine Probleme). In diesem Falle kommt es zu einem rekursiven Aufruf der gleichen Routine jetzt auf Level 7, der sicherlich Verwirrung stiften dürfte. Eine absolut zuverlässige Abhilfe bietet leider nur eine externe Logik - ein Prioritätsdekoder oder gar ein Hilfsprozessor. Man kann aber auch durch die Software den rekursiven Aufruf feststellen und gegebenenfalls mit dem Befehl RTR in die Routine zurückspringen, die dann unter Level 7 ungestört zu Ende laufen kann.

Dazu wird man am Beginn der Level 5/7-Routine mit BSET irgendein Flag setzen oder gegebenenfalls verzweigen: nach Ablauf der Routine muß dieses Flag wie natürlich auch die anfordernde Leitung zurückgesetzt werden, wozu man, wenn man ganz streng sein wollte, eigentlich einen Timer bemühen müßte. Nun könnte man das Zurücksetzen des Flags unnötig machen, wenn man Bit 11 im Statusregister markieren könnte; abzufragen wäre dann das entsprechende Bit auf dem Stack. Leider läßt sich Bit 11 SR nicht markieren - Bit 10 aber wird beim Sprung auf Level 5 oder 7 automatisch gesetzt! Um das zu nutzen, dürften wir die Programme allerdings nur mit Maske 0...3 laufen lassen! Die Interrupt-Routine zu Level 5 und 7 bekäme dann einen Vorspann:

Einsprung:		Routine:	* Hauptroutine Int. Level 5 und 7
btst #2, (a7)	* Letzter Level auf dem Stack	..	
beq routine	* kein rekursiver Aufruf	..	
bchg #2,1 (a7)	* Z-Flag wieder auf den	..	* externe Leitung zurücksetzen
bchg #2,1 (a7)	* alten Wert bringen und	rts	
RTR	* fortfahren auf Level 7		

Falls die externe Leitung doch noch vor rts aktiv wird, richtet ein rekursiver Aufruf keinen Schaden mehr an; daß im ärgsten Fall der Stack vollläuft, ist wenig wahrscheinlich. Es bleibt der Zeitbedarf für die unechten Interrupts, durch den zeitkritische Routinen aus dem Tritt geraten könnten.

```

Patchwork-Listing Teil 2
(C) Guenter Renner -
fatin1 equ $fdfff03 *FAT-Eintraege
fatin2 equ $4000f0f *f. Handler
****Menuefunktionen
newdisk:
lea maldn(pc),a0
bsr meldung
move #1ci,d7
bsr traprts
and #5f,d0
cmp #7,j,d0
bne menue
bsr getname
lea fatbuf1(pc),a4
move #(11*512)-1,d4
newclear:
clr.b (a4)+
dbra d4,newclear
lea fatbuf1(pc),a4
move.l #fatin1,(a4)+
move.l #fatin2,(a4)
lea fatbuf2(pc),a4
move.l #fatin1,(a4)+
move.l #fatin2,(a4)
lea dirbuf(pc),a4
lea namebuf(pc),a3
cmp.b #'.',(a3)
ble newdi2
move #11-1,d4
newdi1:
move.b (a3)+(a4)+
dbra d4,newdi1
move.b #8,(a4)
lea dirbuf+32(pc),a4
*Attribut dazu
*naechster entry
newdi2:
lea entryha(pc),a3
move #32-1,d4
newdi3:
move.b (a3)+(a4)+
dbra d4,newdi3
lea fhflag(pc),a4
clr.b (a4)
lea bootsek(pc),a0
clr d2
moveq #2,d1
bsr sektor
bsr putfat1
bsr putfat2
bsr putdir
lea handler(pc),a0
move #12,d2
*Punkt entfernen
*an den Anfang
*fuer 11 Zeichen
*dann ists Ende
*Name uebertragen
*max. 8 Zeichen
*Rest des Namens
*bis 8. Zeichen
*auffuelien
*Punkt uebergehen
*in jedem Fall
*noch 3 Zeichen
*Punkt uebergehen
****Defaults
entryha:
dc.b 'HANDLER 68K'
dc.b 21 *Attribut 'nur lesen'
dc.b 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
dc.b $21,0,0,0 *waere Datum
dc.b 2,0 *Startcluster
dc.b 0,12,0,0 *Groesse/Bytes
einbuf: ds.b 16
namebuf: ds.b 12
fhflag: dc.b 0

```

Bild 1: Patchwork-Listing Teil 2

Rolf Lobreyer 7/88

COMAL-80

für NDR-Klein-Computer unter CP/M 68K

COMAL steht für **COM**mon **AL**gorithmic **L**anguage und wurde 1973 von **Borge R. Christensen** und **Benedict Loefstedt** in Dänemark entwickelt, fand aber, wie viele andere Programmiersprachen, zunächst kein großes Echo. COMAL wurde ursprünglich für den Informatikunterricht an Schulen entwickelt, aber auch die Industrie setzt, laut Handbuch, COMAL wegen der kurzen Programmentwicklungszeiten ein. **COMAL-80 für CP/M 68K-Systeme** entstand als Nebenprodukt der Entwicklung von COMAL für den AMIGA. Diese COMAL Version wird bereits auf dem C64 und dem IBM PC/XT/AT angewendet.

Das vorliegende Programm ist frei von Lizenzgebühren und steht für Anwender des mc68000 Computers zur freien Verfügung. Geliefert wurde COMAL auf einer Diskette für CP/M 68K, die auf den mc68000 Rechner abgestimmt war, sowie eine englische COMAL - Kurzreferenz in Form einiger kopierter Seiten. Frei von Lizenzgebühren heißt nur daß für das Programm selbst keine Lizenzabgabe zu entrichten ist, für Kosten die dem Vertrieber durch Kopieren, Disketten und Druck der Kurzreferenz entstehen, aber eine geringe Gebühr zugebilligt wird. Was ist COMAL überhaupt ?

- COMAL stellt eine relativ neue Entwicklung dar, die Konzepte der Programmiersprachen BASIC, PASCAL und MODULA II in sich vereinigen soll.
- Von BASIC wurde die Einfachheit der Sprache, die schnelle Erlernbarkeit durch Ähnlichkeit der Schlüsselwörter und das Interpretierkonzept übernommen.
- Von Pascal wurden höhere Schleifenkonstrukte wie REPEAT..UNTIL, WHILE..ENDWHILE sowie ein Funktions- und Prozedurkonzept entlehnt.
- Über PASCAL und BASIC hinaus, in Richtung MODULA II, wurde ein Modulkonzept, die sogenannten Packages, eingeführt.

Diese können als eigenständige Module in ein COMAL-Programm eingebunden werden, wobei sowohl COMAL-Programme als auch Assemblermodule zum Einsatz kommen können. Hierdurch läßt sich COMAL um zusätzliche Funktionen und Prozeduren, ähnlich den Compilersprachen, erweitern. COMAL-80 umfaßt einen Grundwortschatz von 138 Schlüsselwörtern, was bereits eine recht hohe Anzahl an Funktionen und Prozeduren darstellt, die im Grund-Comal integriert ist und frei von jeder direkten Hardwareabhängigkeit ist. Die Routinen stehen letztendlich jedem CP/M 68k Benutzer zur Verfügung. Eine Reihe von Funktionen und Prozeduren wird dem

Grund-Comal beige packt, um es grafikfähig zu machen, wobei es eine Art Schildkrötengrafik gibt, die der Grundprogrammgrafik sehr ähnlich ist und auf LOGO zurückgeht.

Die zum Test zur Verfügung stehende Version war leider nicht auf den NDR-Klein-Computer sondern auf den mc68000-Computer der Zeitschrift "mc" angepasst. Startet man COMAL.REL von der Diskette, erfolgt nach Laden des Programms die Meldung Exception \$08 at user address \$00ff7424 und das Programm wird abgebrochen. Allerdings muß zur Ehrenrettung gesagt werden, daß man eine neue COMAL-Version mit dem ebenfalls auf der Diskette befindlichen Installationsprogramm generieren kann, die dann keinerlei mc68000 Eigenheiten aufweist. Das Erzeugen einer angepassten Version für den NDR-Klein-Computer unter CP/M 68k kann der Kurzreferenz auf Seite 19 entnommen werden.

Zum Test habe ich dann ein COMAL generiert, das für den NKC geeignet ist und mit NEWCOMAL.REL aufgerufen wird. Bei der Installation können Tastatur- und Bild-

schirmcodes definiert werden. Hier habe ich die gebräuchlichen Wordstarcodes und die Terminalemulation des Grundprogrammes angegeben. Eine genaue Zuordnung entnehme man Tabelle 1.

Arbeiten mit COMAL-80:

Ruft man NEWCOMAL.REL auf, wird der Bildschirm, nach Laden des Programms, gelöscht und eine Statuszeile COMAL-68K Ver 1.1 Free xxxxxx.xx KB oben ausgegeben. Beenden kann man das Programm mit BYE. Laut Handbuch werden einige Beispiel-Programme mitgeliefert, die aber fast ausschließlich nur mit dem mc68000-Computer ablauffähig sind, da sie Funktionen und Prozeduren aus hardwareabhängigen Packages verwenden. Ein nahezu Hardware-unabhängiges Programm lag mit sorttest.sav vor. Es konnte, nach Änderung der Anweisung über Verwendung des Timermoduls mit der Entfernung der Funktionen aus dem Programm, gestartet werden.

Im Übrigen ist COMAL, wie vielleicht von einigen BASIC-Dialekten her bekannt, bildschirmorientiert. D.h. man kann auf dem Bildschirm, wie mit einem Editor, das Programm eingeben und editieren. Die Programmzeilen werden dabei mit RETURN übernommen, in ein internes kompaktes Format abgelegt und ein erster SYNTAX-Test vorgenommen. Eine fehlerhafte Programmzeile wird solange nicht akzeptiert, bis sie syntaktisch korrekt ist. Eventuelle Fehlermeldungen erscheinen in der Statuszeile rechts oben. Die syntaktische Überprüfung bietet bei der Programmerstellung eine große Hilfe, da nun nicht erst beim Ablauf oder bei der Übersetzung, durch einen Compiler, vorhandene syntaktische Fehler erkannt werden, sondern bereits bei der Eingabe. Weit reicht diese Überprüfung allerdings nicht, daeben nur Syntaxfehler, nicht aber semantische Feh-

Clear to end of line	CTRL-T	14
Clear to end of screen	-	
Right tabulate	CTRL-I	09
Left tabulate	-	
Clear screen	CTRL-Z	1A
Start of	CTRL-L	0A
End of line	CTRL-Q	11
Top of screen (Home)	CTRL-R	12
Bottom of screen	CTRL-C	03
Delete character	DEL	7F
Insert character	CTRL-V	16
Cursor right	CTRL-D	04
Cursor left	CTRL-S	13
Cursor up	CTRL-E	05

Tabelle 1: (Zuordnung Tastatur Steuer- und Funktionstasten)

ler erkannt werden können. Diese werden erst beim Programmablauf erkannt, bzw. bei einem PRE-Run, was einem semantischen Test vor der eigentlichen Programmausführung entspricht, der nach Eingabe von RUN automatisch erfolgt.

Dieser PRE-Run wird bei nahezu allen heutigen Interpretierern durchgeführt, um z.B. zu testen, ob ein IF mit ENDIF abgeschlossen ist oder eine Sprungmarke vorhanden ist. Der Programmablauf kann durch Drücken der ESC-Taste jederzeit unterbrochen werden. Die Geschwindigkeit, mit der COMAL-Programme ausgeführt werden, kann mit der moderner BASIC-Interpreter schritthalten. Ein Vergleich zu Compilersprachen wäre hier unfair, da Interpretierer immer einen gewissen Overhead bei der Abarbeitung von Programmen aufweisen, die prinzipbedingt sind.

Fazit:

Wie eingangs schon erwähnt, bietet COMAL die Möglichkeit der strukturierten Programmierung mit Prozeduren, Funktionen, gegenüber den ursprünglichen BASIC-Versionen erweiterten Schleifen und Kontrollstrukturen und ist somit insbesondere für den Ausbildungssektor interessant, wo BASIC eventuell bereits bekannt, PASCAL aber schon zu kompliziert erscheint, um schnell programmieren zu lernen.

Für die Industrie erscheint mir COMAL nicht sehr geeignet zu sein, obwohl es sicherlich keine schlecht entworfene Sprache ist, mit der sich auch größere Projekte aufgrund der Modulstruktur verwirklichen lassen könnten.

Ein entscheidender Nachteil von COMAL, BASIC, LOGO ... ist aber, daß das Strukturierungskonzept wie die RECORDs in PASCAL oder die , Strukturen von C ... fehlen, sowie, daß ein Typkonzept nur im Ansatz vorhanden ist, d. h., es sind nur einige Grundtypen vorhanden, die mit den Mitteln der Sprache nicht erweiterbar sind.

Möglicherweise noch gewichtiger erscheint mir, die geringe Durchdringung des Marktes mit COMAL zu sein. Vorherrschend sind hier Sprachen wie BASIC, C, PASCAL oder neuerdings auch MODULA II. Als Ersatz von BASIC könnte ich mir COMAL sehr gut vorstellen, da ein Umstieg von BASIC nach COMAL keinen allzu großen Lernaufwand voraussetzt.

Allerdings muß eine Vielzahl vorhandener BASIC-Programme dann

an COMAL angepasst werden, was nicht übermäßig schwer fallen dürfte, da die beiden Sprachen von den verwendeten Schlüsselwörtern her sehr ähnlich sind. Die bei COMAL gewürdigte Möglichkeit der strukturierten Programmierung wird neuerdings auch in den moderneren BASIC-Interpretieren, wie sie im übrigen von Microsoft und Borland für den IBM PC/XT/AT vertrieben werden, angeboten.

Als gelungen empfinde ich die Möglichkeit, eigene Module aus COMAL, Assembler oder C-Programmen zu schreiben, deren Prozeduren und Funktionen aus einem beliebigen COMAL-Programm heraus Verwendung finden können. Bei BASIC-Dialekten vermisst man diese Tatsache schmerzhaft. Man ist dort auf den gebotenen Sprachschatz angewiesen, eine Erweiterung entfällt, oder wird nur sehr schwerfällig unterstützt.

Toll finde ich, daß ein recht arbeitsintensives Produkt wie eine Programmiersprache umsonst abgegeben wird. Vergleichbare BASIC-Interpreter kosten so um die 100DM aufwärts.

Allerdings werden dort dann auch ordentliche Handbücher mitgeliefert, die alle Befehle erläutern. Die den Testunterlagen mitgelieferte Kurzreferenz kann wirklich nur als solche dienen, da die Wirkungsweise der Befehle nicht erklärt wird. Man ist damit aufgefordert, ein COMAL-Buch zu erstellen, ohne das eine effektive Nutzung aller Funktionen nicht möglich erscheint.

Dennoch ist COMAL als günstig anzusehen, da es ja frei kopiert werden kann, somit vielleicht der eine oder andere damit herumspielt und so auf den Geschmack kommt und sich ein COMAL-Buch kauft. In Zeiten des Schwarzkopierens eine legale Sache, gute Software zu erstellen.

Schade fand ich, daß das Modul für die Turtlegrafik nicht funktionierte, da es für den mc68000-Computer geschrieben ist. Die Assemblerquellen liegen aber der Diskette bei, so daß sich vielleicht jemand findet, der eine Anpassung an den NDR-Computer vornimmt. Da durch das Package-Konzept eine Erweiterung der Prozeduren und Funktionen unterstützt wird, sollte sich jemand dazu bereit erklären, eine eingehendere Anpassung an den NDR-

Klein-Computer vorzunehmen, wobei auch die COL256, die AD-DA Wandlerkarten, die UHR ... unterstützt werden. Dann bliebe COMAL nicht nur auf reine textuelle Eingabe beschränkt. Diese Pakete können dann im übrigen durchaus etwas kosten. Erweiterungspakete für den IBM PC/XT/AT werden mit ca. 150 DM pro Paket gehandelt!!! So bleibt als Quintessenz, daß COMAL aufgrund des günstigen Preises eigentlich jedem empfohlen werden kann, dem ein BASIC-Interpreter zu teuer, eine Compilersprache wie C oder PASCAL zu umständlich und zu schwierig zu erlernen scheint. D.h. allen Einsteigern. In der jetzigen Version wird der Spaß allerdings bald ein Ende finden, wenn auch weitergehende Fähigkeiten des Rechners genutzt werden wollen, da noch keine entsprechend angepassten Module für den NKC vorhanden sind, ein Einsteiger sich ein solches aber schwerlich selbst erstellen kann. Um eine Zukunft auf dem NDR-Klein-Computer zu haben, müssen entsprechende Module angeboten werden, um nicht letztendlich in der Ecke zu landen.

Anmerkung der Redaktion:

Aufruf an LOOP - Leser:

Wer möchte Module dazu erstellen? Wir stellen Ihnen das COMAL 80 kostenlos zur Verfügung.

```

0010 // Sortierungstest med quicksort
0020
0030 // USE TIMER FROM "TIMER.PCK"
0040
0050 PROC quicksort(REF t$( ),start,slut)
0060 DIM temp$ OF 10, x$ OF 10
0070 a:=start
0080 z:=slut
0090 x$:=t$((start+slut)/2)
0100 REPEAT
0110 WHILE t$(a)<x$ DO a:=+1
0120 WHILE x$<t$(z) DO z:= -1
0130 IF a<=z THEN
0140 temp$:=t$(a); t$(a):=t$(z); t$(z):=temp$
0150 a:=+1
0160 z:= -1
0170 ENDIF
0180 UNTIL a>z
0190 IF start<z THEN quicksort(t$( ),start,z)
0200 IF a<slut THEN quicksort(t$( ),a,slut)
0210 ENDPROC quicksort
0220
0230 n:=1000 // Tekst mängde
0240 DIM t$(n) OF 10
0250 FOR i:=1 TO n DO
0260 FOR j:=1 TO 10 t$(i):+chr$(rnd(65,90))
0280 ENDFOR j
0290 ENDFOR i
0300
0310 // stoptimer
0320 // resettimer
0330 // starttimer
0340 quicksort(t$( ),1,n)
0350 // stoptimer
0360 // PRINT USING "Sorteringstid: ##.##": timer

```

Bild 1: Beispielprogramm in COMAL (von Testdiskette)

Korrekturen und Ergänzungen

von
Ralf Dombrowski, Große Deichstr. 33, 2208 GLÜCKSTADT

Probleme des JADOS mit dem Grundprogramm 6.0

Falls Sie Besitzer des neuen Grundprogramms sind, werden Sie es sicher schon bemerkt haben. Bei mehr als 32 Einträgen auf einer JADOS-Diskette werden die ersten Einträge vier Mal wiederholt. Dies liegt daran, daß das neue Grundprogramm mehr Floppy-Befehle versteht und im JADOS ein kleiner Fehler vorliegt. JADOS liest den ersten Sektor des Inhaltsverzeichnis richtig, dann aber werden nur noch die Sektorinformationen gelesen, die nur ein paar Bytes lang sind. Dadurch werden die alten Daten nicht überschrieben und noch drei Mal ausgegeben. Bei der Version 2.10-01 kann der Fehler leicht behoben werden.

Für die Ausgabe des Directories existiert dort folgende Schleife.

```
move #1,d1
lea $a8(a6),a0
clr d5
schleife:
    einen Sektor lesen
    Inhaltsverzeichnis ausgeben
    Wiederholen, bis alle Sektoren gelesen
```

In der Schleife wird aber d1 zerstört. Dadurch ist die FLOPPY-Routine des Grundprogramms nicht mehr auf Sektor-Lesen geschaltet. Es kommt zu dem Fehler.

Nun funktioniert JADOS wieder fehlerfrei.

Abhilfe schafft folgendes Programm:

```
lea $a8(a6),a0
clr d5
schleife:
    move #1,d1
    einen Sektor lesen
    Inhaltsverzeichnis ausgeben
    Wiederholen, bis alle Sektoren gelesen
```

Die Änderung ist im JADOS-System folgendermaßen durchzuführen:

- 1.) Sektor 4 der zweiten Spur laden
- 2.) Adresse \$134 suchen
- 3.) Wert-Folge \$323c 0001 41ee 00a8 4245 in \$41ee 00a8 4245 323c 0001 ändern
- 4.) Adresse \$1f6 suchen
- 5.) Wert \$ff48 in \$ff44 ändern
- 6.) Sektor wieder abspeichern

Probleme mit dem Booten beim neuen Grundprogramm

Leider gibt es beim Grundprogramm V. 6.0 ein Problem mit der Routine GETFLOP beim FDC des Elektronik-Laden. Dieses Problem ergibt sich durch die abweichende Schaltung der Karte. Es tritt dann auf, wenn ein Betriebssystem gebootet werden soll. Die GETFLOP-Routine V. 6.0 erkennt nämlich, ob sich eine Diskette im Laufwerk befindet oder nicht. Die alte Routine von R.-D. Klein hat ewig gewartet, während die Neue nach einiger Zeit abbricht und einen lästigen Reset vermeidet. Leider funktioniert dieses nicht beim FDC. Deshalb sollten alle Besitzer der Diskettenversion das Programm GETFLOP neu eingeben und das Grundprogramm neu übersetzen. Die EPROM-Besitzer kön-

nen sich nur durch Neubrennen der Eproms behelfen. Das Problem kann eventuell umgangen werden, in dem vor jedem Booten der Motor des Diskettenlaufwerks angeschaltet wird.

Beispiel:
start:
move.b #\$21,\$ffffc4*cpu (cpu= 1,2,4)
rts

Dieses Programm aufrufen, dann normal booten. Die GETFLOP-Routine hat jetzt wieder das ursprüngliche Aussehen, da ich das Problem nicht in den Griff bekommen habe. Falls jemand eine gute Lösung weiß, möge er sie mir bitte zuschicken.

```
getflop: * Floppyformat feststellen,
          d4 = Laufwerksnummer
move.b #$d0,flo0.w * Controller
                    rücksetzen
tst.b flo0.w * Interrupts löschen
and #$8f,d4 * Nur Seitenbit und
                    Laufwerkscodierung
                    lassen
bsr wawa
or.b #%00110000,d4
bsr.s get1flop * Mini SD
bcc flook
and.b #%10101111,d4
bsr.s get1flop * Mini DD
bcc flook eori.b #%00110000,d4
bsr.s get1flop * Maxi SD
bcc flook
and.b #%10001111,d4 * Maxi DD
get1flop:
move.b d4,flo4.w * Laufwerkscode
                    übertragen
move.b #$f,flo0.w * Restore mit Ve-
                    rify und Head-
                    load
```

```
get2flop:
btst.b #6,flo4.w
beq.s get2flop * Warten bis Ready
move.b flo0.w,d0 * Status einlesen
                    und Interrupts lö-
                    schen
move.b d0,d1 * Merken für Analyse
and.b #%10011000,d0
bne floerr
bra flook
```

Sieg für SCHACH V 2.9-00 auf JADOS

In der Jugendvereinsmeisterschaft des Schachklubs Maintal trat Klaus Rumrich mit seinem selbstgeschriebenen Schachprogramm außer Konkurrenz an und erreichte den ersten Platz mit dem Mitsieger Lars Jessen. Den einzigen Punkt (Remis) gab das Programm durch einen "Chipinfarktes" ab, bei dem es 3 mal den gleichen Zug ausführte. Übrigens hat es zum Erstaunen der Zuschauer eine Partie mit einem dreizügigen Matt mit Damenopfer gewonnen. Der Kommentar von H. Rumrich zu seinem Erfolg: Ich werde das Programm noch schneller und besser machen und im nächsten Jahr in der offenen Meisterschaft antreten.

VGA läßt Graphikerherzen höher schlagen

Die VEGA VGA-Karte kann folgende Graphik-Adapter emulieren:

- Monochrom-Adapter
- Hercules-Graphikadapter
- Color-Graphikadapter
- EGA-Bildschirmadapter

Damit läßt sich auch alte Software weiterhin mit dieser Karte einsetzen.

Insgesamt kennt die VGA-Karte 17 verschiedene Graphik- und Textmodi. Hierbei sind 7 Modi zu den Modi 0 bis 6 der CGA-Farbgraphikkarte kompatibel, ein Modus entspricht dem Modus 7 des Monochromadapters, weitere 4 Modi sind zu den EGA-Modi 13 bis 16 kompatibel und neu gegenüber dem EGA-Standard sind je 3 Graphik- und Textmodi.

Die neuen Betriebsarten bieten Auflösungen mit 320 x 200 Bildpunkten, bei gleichzeitiger Darstellung von 256 Farben, und 640 x 480 Bildpunkten bei 16 Farben. Diese beiden Modi stellen den VGA-Standard dar. Daß, bei maximaler Farbanzahl von 256 Farben, nur eine Auflösung von 320 x 200 Bildpunkten möglich ist, liegt an dem nach wie vor auf 256 KByte begrenzten Speicherplatz der VEGA VGA-Karte (für jeden Bildpunkt muß ein Byte im Speicher reserviert werden).

Ebenfalls im niedrigauflösenden Modus, aber nur mit 16 Farben, wird die maximale Anzahl von 8 Graphikseiten der VEGA VGA unterstützt. Aufgrund der Double-Scan-Technik, es wird eine Bildzeile zweimal hintereinander auf dem Bildschirm dargestellt, ist trotz der niedrigen Auflösung die Qualität der Bilder überraschend gut, sodaß dieser Mangel keinen wesentlichen Nachteil für die VEGA VGA darstellt.

Die maximale Auflösung dieser Karte wird erst bei 800 x 600 Bildpunkten erreicht. Damit übertrifft sie den VGA-Standard bei weitem. Bei dieser Betriebsart können Sie 16 Farben gleichzeitig darstellen.

Weitere Vorteile weist die VEGA VGA bei Betrieb der Textmodi auf. Im Vergleich zur EGA-Karte, welche in den Textmodi mit einer Matrix von 8 x 14 Bildpunkten pro Zeichen arbeitet, kennt die VGA-Karte zwei Varianten der Zeichendarstellung. Hierbei wird entweder mit einer Matrix von 9 x 16 oder 8 x 8 Bildpunkten pro Zeichen bei 16 Farben verwendet. Bei Benutzung der 9 x

Was ist ein PC ohne hochauflösende Graphik? Viele LOW-COST Systeme bieten die EGA-Graphik als Standard in Ihrem System an. Mit der VEGA VGA - Graphikkarte für Ihren mc - modular AT übertreffen Sie diese bei weitem. Die maximale Auflösung von 800 x 600 Bildpunkten, sowie die gleichzeitige Darstellung von 256 Farben sind Werte, die für sich sprechen.

Mit der Einführung des neuen PS/2-Systems durch IBM hat sich ein noch besserer Farbgraphik-Standard gebildet. Dies ist die VGA-Farbgraphik (Video Graphic Array).

16 Bildpunktmatrix bedeutet dies eine ca. 30 Prozent höhere Auflösung.

Sämtliche Farbangaben haben sich auf eine Auswahl aus der 262144-Farbpalette bezogen. Unter Beachtung, daß die drei Grundfarben ROT, GRÜN und BLAU, jeweils 64 Varianten im Bereich von hell bis dunkel besitzen, ergibt sich rechnerisch die Palette von 262144 Farbtönungen. Dadurch ist es dem Benutzer möglich, sehr farbenfrohe Bilder mit extrem weichen Farbübergängen zu erzeugen.

Um die gleichzeitige Darstellung von 256 Farben zu ermöglichen, mußte das analoge Bildsignal eingeführt werden. Bei Betrieb mit digitalen Signalen können nur 16 Farben aus einer 64er-Farbpalette gleichzeitig verwendet werden.

Aus diesem Grunde weist die VEGA VGA zwei Signalausgänge auf. Zum einen den 9-poligen Sub-D-Stecker für die digitalen Bildsignale und zum anderen einen 15-poligen Steckkontakt für das Analogsignal. Daß die VEGA VGA die Möglichkeit bietet, analoge Signale zu verwenden, ersieht man auch daran, daß einmal die Anzahl der Quarze von zwei auf drei erhöht wurde, und zum anderen ein Digital-Analog-Wandler von INMOS auf der Karte Verwendung findet.

Um die Analogsignale verarbeiten zu können, benötigt der Anwender den richtigen Monitor (beachten Sie hierzu den Beitrag über den NEC MultiSync II Farbmonitor in dieser LOOP-Ausgabe). Dies sollte ein hochauflösender Farbmonitor sein, der in der Lage ist, analoge und digitale Bildsignale zu verarbeiten. Selbstverständlich können auch herkömmliche Monitore angeschlossen werden. Dabei verlieren Sie jedoch die Vorteile der VGA-Karte gegenüber der EGA-, CGA-, Hercules- und Monochrom-Karte, da eine analoge Signalverarbeitung nicht möglich ist. Positiv bleibt zu erwähnen, daß alle Betriebsarten nicht über die Software emuliert

werden, sondern eine hardwaremäßige Umschaltung erfolgt. D.h., die Karte erkennt den jeweils zugehörigen Bildschirmmodus selbstständig und schaltet dann automatisch um. Auch die Art des Monitors kann die VEGA VGA erkennen und sich automatisch darauf einstellen. Selbstverständlich kann diese Automatik blok-

kiert werden, um momentan unerwünschte Betriebsarten verhindern zu können.

Auch in Bezug auf die Verarbeitungsgeschwindigkeit ist die VEGA VGA der EGA-Karte bei weitem überlegen. Die Begründung hierfür liegt zum einen in der schnelleren Hardware und zum anderen in der optimierten Systemsoftware im ROM der VGA-Karte.

Die Geschwindigkeit des Bildaufbaus ist im wesentlichen von der Breite des Video-Daten-Bus abhängig. Die VGA-Karte benötigt aus diesem Grunde einen 32-Bit Bus zwischen dem VGA-Chipsatz und dem Bildwiederholungspeicher und der Datenbus zwischen VGA-Chipsatz und Video-RAM wurde mit 16-Bit Breite ausgelegt. Beides trägt zu einer schnelleren Verarbeitung bei. Lediglich die PC-Bus-Schnittstelle besitzt nur 8-Bit Datenbreite, da leider immer noch der Video-Controller 6845 verwendet wird, der nur einen Zugriff mit 8-Bit Breite auf seine Ports zuläßt.

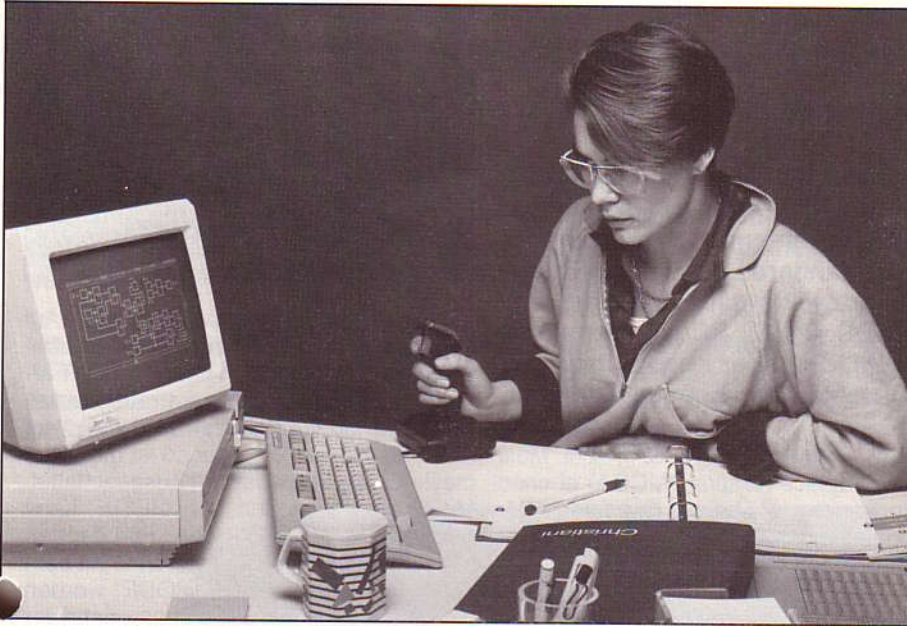
Wenn man sich die Vorteile der VGA-Karte noch einmal vor Augen führt:

- Emulierung sämtlicher bisherigen Graphikkarten
- Hardwaremäßige Umschaltung der Betriebsarten,
- Selbstständiges Erkennen des Monitors und der Konfiguration,
- Schnellerer Bildaufbau,
- Auflösung von 800 x 600 Bildpunkten,
- Gleichzeitige Darstellung von 256 Farben,
- Farbpalette von 262144 verschiedenen

so läßt sich abschließend sagen, daß mit der VEGA VGA, trotz ihrer Mängel:

- nur 256 KByte Arbeitsspeicher
- Handbuch nur in englischer Sprache
- Verwendung des Video-Controllers 6845

ein hervorragender Graphikadapter von Video Seven entwickelt wurde.



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

200 Seiten reichbebildertes Lehrmaterial mit Sammelordner, Register und Logik-Simulator für den Commodore C 64 / C 128 oder IBM- und kompatible Computer mit MS-DOS. Gesamtpreis DM 448,-

Lernen Sie die Bauelemente und Grundschaltungen der Digitaltechnik 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. Hierzu 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

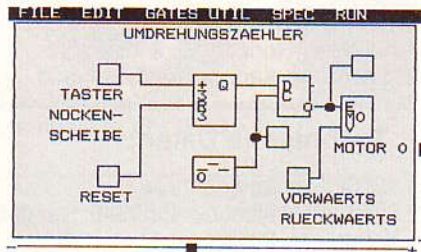
Die Digitaltechnik behandelt alle wichtigen Bauelemente. Wir beginnen mit den logischen Verknüpfungen und gehen über Speicher und Flip-Flops, Schaltalgebra, Zahlensysteme, Codierschaltungen, bis hin zur Mikroprozessortechnik, um nur einige der interessanten Themen zu nennen.

Steuerungstechnik

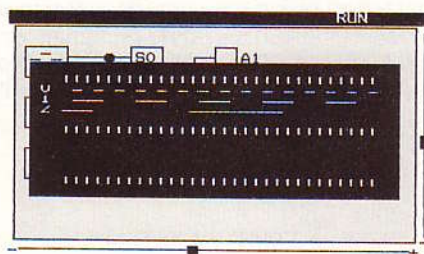
Die Steuerungstechnik hingegen zeigt Ihnen die Anwendung der digitalen Bauelemente. Ablaufsteuerungen, Schaltverstärker, Schrittmotorsteuerungen und vieles mehr wird innerhalb des Kurses ausführlich und leichtverständlich behandelt.

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 C 64 oder C 128 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.

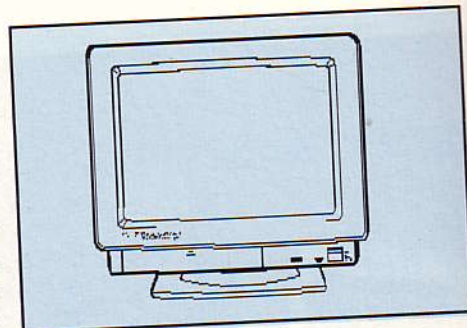
Christiani Fortbildung

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

Andrea Kullak

MultiSync II

Ein Monitor setzt sich durch



Die Ansprüche der Anwender an die Monitore und die Grafikstandards der führenden Hersteller werden immer höher. MultiSync II, der intelligente Monitor von NEC, ist ein hochauflösender Farbmonitor, der auch den höchsten Anwenderansprüchen und Grafikstandards genügt. Der MultiSync II garantiert eine gestochene scharfe Farbwiedergabe und lebendige Farbgrafiken in Verbindung mit einem IBM PC, PC/XT, PC/AT oder einem kompatiblen System, ebenso mit einem IBM PS/2 oder Kompatiblen. Er kann auch mit anderen IBM-kompatiblen Grafikkarten benutzt werden.

Bei der Auswahl eines Monitors sind verschiedene Punkte von Bedeutung:

Die Horizontalfrequenzen des Monitors und der verwendeten Grafikkarte müssen zueinander passen.

Eine Grafikkarte mit einem Analog-Ausgang braucht einen Monitor mit einem ebensolchem Eingang und umgekehrt braucht eine Grafikkarte mit einem digitalen Ausgang einen Monitor mit entsprechendem Eingang.

Wichtig ist natürlich auch die Auflösung des Monitors. Je höher die Auflösung ist, desto besser kann die Wiedergabe auf dem Bildschirm werden. Im Zusammenhang damit ist auch der Abstand der Bildpunkte voneinander interessant. Je enger die Bildpunkte (Üblich: 0,31mm Punktabstand = dot pitch) beieinander liegen, umso höher kann die Auflösung und die Schärfe des Bildes sein. Schließlich und endlich interessiert auch die Größe des Bildschirms, die als Länge der Diagonalen in Zoll angegeben wird. Ein großer Bildschirm ist jedoch nur dann sinnvoll, wenn auch die Auflösung entsprechend hoch ist.

Der MultiSync II von NEC ist ein sogenannter Multifrequenzmonitor. Das heißt, er

kann auf verschiedene Horizontal- und Bildfrequenzen eingestellt werden. Beim NEC MultiSync II müssen diese Frequenzen nicht einmal von Hand eingestellt werden. Die Zeilen- oder Horizontalfrequenz stellt sich automatisch auf den richtigen Wert zwischen 15,5 und 35 kHz ein und die Bildfrequenz darf zwischen 50 und 80 Hz liegen.

Der MultiSync II kann an einen IBM PC, PC/XT, PC/AT, PS/2 oder einen kompatiblen Computer angeschlossen werden und ist für den IBM Professional Graphics Controller (PCG), IBM Enhanced Graphics Adapter (EGA), IBM Color Graphics Adapter (CGA), IBM

VIDEO Graphics Array (VGA), IBM Multi Color Graphics Adapter (MCGA) ebenso zu verwenden, wie bei allen anderen IBM-kompatiblen Grafikkarten. Dank dieser großen Kompatibilität können Grafikkarten oder Software aufgewertet werden, ohne daß hierfür ein neuer Monitor benötigt wird. Da der MultiSync II sowohl über einen Analog- als auch über einen TTL-Eingang verfügt, besteht hinsichtlich des Ausgangs des Grafikkartens keine Beschränkung. Im Analogmodus kann der MultiSync II - abhängig von der verwendeten Software und der Grafikkarte - eine unbegrenzte Farbpalette darstellen. Im TTL-Modus kann auf monochrome Darstellung umgeschaltet werden. Dabei stehen zur Wiedergabe monochromer Textverarbeitungs- und Spreadsheet-Programme, Datenbanken oder anderer Software drei Farben (paperwhite, bernstein und grün) zur Verfügung. Alle Zeichen werden gestochen scharf auf einer High Contrast Bildröhre (14 Zoll, sichtbar Bereich 13 Zoll diagonal) dargestellt.

Obwohl der MultiSync II sich automatisch auf die richtigen Horizontal- und Bildfrequenzen einstellt, kann man diese und

auch noch anderes manuell einstellen. Auf der Rückseite befinden sich drei Schiebeshalter, mit denen man per Hand die Farbauswahl einstellen kann, vorausgesetzt, daß der erste der Schalter "MANUAL" auf "ON", also links steht. Nur so ist die Automatik für die Farbauswahl ausgeschaltet. Mit dem zweiten Schalter "MODE" kann Monochrom- ("GRAY") oder Farbdarstellung ("COLOR") gewählt werden. Mit dem dritten Schalter "COLOR MODE" werden die darstellbaren Farben entsprechend der Grafikkarte eingestellt, wenn der Schalter "MODE" auf "COLOR" steht.

Die Einstellung der vier Möglichkeiten ist notwendig, um mit nicht IBM-kompatiblen Grafikkarten arbeiten zu können. Dabei ist es wichtig, daß die eingestellten Werte denen des Grafikkartens entsprechen. Auf der Vorderseite kann man Kontrast ("CONT") und Helligkeit ("BRIGHT") mittels zwei Rädchen an der Unterseite einstellen. Wenn die Klappe auf der linken Seite mit einem leichten Druck geöffnet wird, erscheinen drei Drehregler und drei Schiebeshalter. Mit den Reglern "V POST" und "H POST" kann das Bild vertikal und horizontal positioniert werden. Der Regler "V SIZE" und der Schalter "H SIZE" sind für die Einstellung der Bildgröße in vertikaler und horizontaler Richtung zuständig. Steht der Schalter "TEXT" auf "ON", so kann mit dem

letzten Schalter "TEXT COLOR" die Farbe der Zeichen in monochromer Darstellung (paperwhite, bernstein und grün) gewählt werden. Allerdings ist die Umschaltung auf monochrome Darstellung nur möglich, wenn der MultiSync II mit TTL-Signalen angesteuert wird. Steht der Schalter "TEXT" auf "OFF" werden die Zeichen in den Farben des benutzten Programmes dargestellt.

Das Einstellen auf nicht IBM-kompatible Computer oder Monitore ist etwas zeitaufwendiger, da alles manuell eingestellt werden muß, ist jedoch anhand der im Handbuch mitgelieferten Pinbelegungen ebenfalls möglich.

Technische Daten:

Bildröhre	13" Bildschirmdiagonale 90 Grad Ablenkung, 0,31mm Trio dot pitch
Eingangssignal	Video TTL positiv ANALOG 0,7 Vss positiv/75Q Sync: getrennt TTL-Pegel horizontal positiv/negativ vertikal positiv/negativ composite TTL-Pegel positiv/negativ composite auf grünem Videosignal 0,3 Vss negativ (Video 0,7 Vss positiv)
Darstellbare Farben	TTL-Eingang: 8/16/64 Farben ANALOG-Eingang: unbegrenztes Farbenspektrum
Synchronisation	Horizontal: 15,5 bis 35kHz (automatisch) Vertikal: 50 bis 80 Hz (automatisch)
Auflösung	Horizontal: 800 Punkte (Die Bildgröße ist abhängig vom Video-Timing) Vertikal: 560 Linien
Videobandbreite	30MHz
Nutzbare Bildfläche	250mm(H) x 180mm(B)
Konvergenzfehler	kleiner als 0,6mm
Stromversorgung	220/240V, 50/60Hz
Leistungsaufnahme	85W
Abmessungen	360(B) x 372(H) x 400(T) mm
Gewicht	16kg

F. Wolfart

DER HANDY SCANNER

für den mc - modular - AT

Der Scanner bietet uns eine neue Art der Eingabe von Grafik und Text auf den Bildschirm Ihres Computers. Die Bedienung des Scanners ist sehr einfach: Der Scanner wird manuell über die zu rasternde Fläche gerollt und schon erscheint die Vorlage auf dem Bildschirm.

Der Handy Scanner läuft auf allen IBM PC/XT/AT's und allen kompatiblen PC's. Das Paket besteht aus dem Scanner, einer Interfacekarte und der dazugehörigen Software.

Von außen, wenn man den Handy Scanner genauer betrachtet, hat dieser eine ausgesprochene Ähnlichkeit mit einer zu groß geratenen Mouse.



Bild 1: Eines der beliebtesten Gesichter in der BRD

Die Einsatzgebiete des Handy Scanner sind:

- Bilder, Grafiken, Skizzen, Zeichnungen, Entwürfe
- Handschriften, Sonder-Schrifttypen und beliebige Schriften
- Buch- und Zeitungstexte, Manuskripte
- Unterschriften und Fotos

Die Abbildung der jeweils eingescannten Vorlage erscheint in weniger als 3 Sekunden auf dem Bildschirm.

Sie müssen nur die Interfacekarte in Ihren Computer einstecken, dann die Scanner-Software laden und schon kann es los gehen.

Die mitgelieferte Software enthält ein Setup-Programm zur Einstellung Ihrer verwendeten Konfiguration und einen Scanner-Treiber als Betriebssystemerweiterung unter MS-DOS, die danach durch Drücken zweier Tasten aus sämtlichen Grafikprogrammen (wie z.B. Dr. Halo, PC-Paintbrush, Star-Writer) aufgerufen werden kann. Das eigentliche Scanner-Paket ist ein eigenständiges Programm, welches auf den Scanner-Treiber zugreift, jedoch RAM-orientiert ist. Dadurch ergeben sich folgende Möglichkeiten:

- Abrastern, Speichern, Drucken, Laden von Flächen die größer sind, als der Bildschirmanzeigebereich
- Zusammensetzen mehrerer abgerasterter Flächen
- Diskettenzugriff im PCX-Bildschirmspeicherformat
- Hardcopy von beliebigen Bildschirmsegmenten

Mindestkonfiguration: 128 kByte RAM
Betriebssysteme: ab DOS V 2.0
Scanner Maße : 90 x 122 x 28 mm
Gewicht: 150 g

Kurzinfo

Cherry G80 - 1600

Für den Einsatz von Barcode-Lesestiften, Laser-Scanner-Pistolen und Durchzug-Leser bietet sich für den mc-modular-AT die neue Cherry G 80 - 1600 Tastatur.

Die Verwendung des erwähnten Eingabegerätes ist vielfältig und bietet rationelle Möglichkeiten. Beispielgebend dafür stehen die Betriebsdaten-Erfassung, Inventur-Eingabe, Arzneimittelverwaltung, Produktidentifizierung bei Warenein- und Ausgang oder etwa die Datenerfassung im Produktionsablauf. Die lesbaren Codearten sind: Code 39, Standard oder Full ASCII, Code 2 aus 5, Standard oder Interleaved, Code 128, Codabar und UPC/EAN/JAN-Codes.

Lieferbar: Ende 1988



NKC - fertig aufgebaut

Wegen Systemwechsel **zu verkaufen:** CP/M, IBM-PC-Gehäuse, Tast2, Philips Monitor, 2 TEAC-LW (zu 800KB), RAM 64256, AD 8/16, DA, SER, IOE, PROMER. Zusätzlich: ROA 64 voll bestückt, ROA 64 mit EGRUND und 1k*8 RAM. Turbo Pascal, M80 Assembler, Tool-Disk, Disk Doc. Vollständige Dokumentation (incl. alle Loop Ausgaben). Preis Sfr 950,—
M. Schwager, Tüffenwies 19/31, 8064 Zürich, Tel. P: ab 19:00, 01/62'45'63, G: 01/215'62'64

Suche ZEAT und ZEAT-Disk und DFÜ-Software angepaßt an NKC (CP/M80). Adresse: Harald Dietz, Windhukstr. 54A, 56 Wuppertal 22, Tel. 0202/664893.

Suche CP/M DF-Programm angepaßt an NDR-Computer 8" Standard Format. Harald Dietz, Windhukstr., 54A, 56 Wuppertal 22, Tel. 0202/664839

Verkaufe preisgünstig ACRTC-Baugruppen.
Willi Sick, Jägerweg 31, 4423 Gescher, Tel. 02542/6114

Verkaufe: IBM-Geh. m. BUS2 + BUS3, NETZ1, PROMMER, POW22-26, PASST-LÜF. 150,— /SRAM 512 m. 8*32KB 150,— /DRAM 256KB 100,— /KEY2 m. Puffer 90,— /SER 40,— /CENT2B 80,— /TEAC FD55FVC-13-U 180,— /FLO2B100,— /GDP 100,— /CPU68K 90,— /HCOPY2B 130,— /ATARIMAUS 50,— / IOE M. LAUTSPR. 25,— /CPUZ80 6MHz 30,— /RA 64K 50,— /BANKBOOT 35,— und sonst. VB-Preise bis 16.30 Tel. 0531-39007171 o. ab 18:00 05341-47394 Schaffhausen.

Verkaufe wegen Zeitmangel:

NDR-Computer im 19" Gehäuse, Z80 & 68008, 160KB RAM, GDP (auch mit Hardcopy+Scroll-Erweiterung), CENT, FLO, PROMER, UHR, 2x FD 35F-N, 1x FD 55 F-V, große Tastatur, Monitor (bernstein) Netzteil 5V 6A, 12V 2A, -5V, -12V, jede Menge Software und Literatur, auch Einzelverkauf. T. Krähmer, Kolpingweg 9, 7344 Gingen/Fils, Tel. 07162/3998

Verkaufe meinen NDR-Klein-Computer an Leute mit sehr wenig Geld! Komplettes CP/M2.2-System mit viel Software und über 20 Baugruppen. Hermann Obermeyer, Silvanerstr. 34, 6500 Mainz 42, Tel. 06131-507634

Verkaufe: 1 Gehäuse f. ML-CP/M-Computer 130,—, Netzgerät NE1 160,—, Leiterpl. Prommer 20,—, Leiterpl. POW 22/26 10,—, BUS1 30,—, SBC2 30,—, ROA64 o. Speicher 30,—, POW 5V 20,—, HEXIO 40,—, KEY 30,—, GDP 64K 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. 0711/752

Biete an: Original-SW mit HB für NDR-Comp. (Z80) 5 1/4" NDR-Formatz. T. mit 8" Disk.

Turbopascal 3.0 DM 150.—; Micro - Shell (CP/M-Zusatz-Betr. System (Unix)) DM 200.—; CAD58 DM 200.— mit TTL-Bibl.; Texthammer DM 50.—; Big-Zap-Diskedit (Schäfer) DM 100.—; Super-Zap-Diskedit (tesco) DM 100.—; Cher-Edit-Zeichensatz für Terminal-Eprom DM 100.—; WS-Tuner DM 125.—

Selbstabholung oder per NN.

Dr. Hans Hehl, Lindenstr. 20, 8059 Wartenberg, Tel.: 08762/3070

Verkaufe: CP/M 2.2 incl. deut. BH; FLO-MON; Grund Z80; Basic 4000H 1.5V; CPU Z80; CAS-Schnittstelle incl. Kabel; Promer; POW 21/24 V1; Tast 1 incl. Gehäuse und Kabel; RAM 256 voll bestückt; IOE; Floppy 5 1/4" Teac FD-55; VB DM 740.—
Birger Dunsche, Am Westheck 162, 4600 Dortmund 12, Tel.: 0231/252415

Verkaufe aus Zeitmangel: NDR-Klein-Computer CPU68K, funktionsfähig im gehäuse mit grünem Bildschirm, 2 x ROA64 mit Grundprogramm, Tast Kl, FLO3, Disklaufwerk 5 1/4", GDP64K, NEC3, SBC2, POW5, CPU Z80, versch. Disketten (JADOS, -TOOL, SCHACH), Literatur und Eproms. Einmalige Gelegenheit! Alles nur zusammen DM 1500.—

Dieter Lindinger, Gutenbergstr. 46 8502 Zirndorf, Tel.: 0911/607606

Verkaufe: NDR/CP/M Portable, alles im Gehäuse, betriebsbereit. Bestehend aus CPU Z80, CPU69008 zuschaltbar, 96K RAM, GDP64, Tastatur und Monitor im Gehäuse, Hardcopy, Atari Maus, Bankboot, 3 serielle und 2 parallele Schnittstellen, Cassettenbaugruppe, FLO3, 3 1/2" Floppy, 5 1/4" Floppy mit viel Software. Alles zusammen VB DM 2000.—
Bernhard Lechleitner, A-6644 Elmen, Tel.: 0043 - 05635202

Verkaufe: Netz 85.—; RDK Buch 68000/8 25.—; Key + Hb 20.—; Loop 2-18 5.—; Ralf Rigger, Frauengasse 8, 7407 Rottenburg 1, Tel.: 07472/23928

Verkaufe: Fertigplatinen - geprüft - neu. 3 x SBC3 m. Speicher a` 200.—; 8 x RAM 256K stat. 100ns a` 200.—; 1 x DRAM 256K 180.—; 1 x ROA 64K stat. 150ns 100.—; 2 x SER a` 100.—; 2 x KEY a` 80.—; 5 x I/O a` 50.—; 2 x FLO a` 200.—; 2 x GDP64K a` 180.—; 1 x Tastatur kompl. neu 150.—
U. Weber, Tel.: 04101/65946 ab 19.00 Uhr

Verkaufe NDR-Computer mit CPU 68000/12MHz, 512k RAM, 1 x Floppy 5 1/4", Centronics-Schnittstelle, BUS3 + BUS1, CP/M 68K, RL-BASIC, ...komplett aufgebaut.

A. Spring, Friedr.-v.-Teck-Str. 3, 8884 Höchstädt, Tel.: 09074/5213

Verkaufe Schach, Dame, Reversi, Mühle, 4-gewinnt, Gomoku u.a. für JADOS-Systeme. Demodisk (5 1/4") mit allen Spielen für DM 10.— (Schein o. Briefm.) oder Info bei U. Steinhaus-Peers, Schauenburgerstr. 52, 2300 Kiel, Tel.: 043/566795

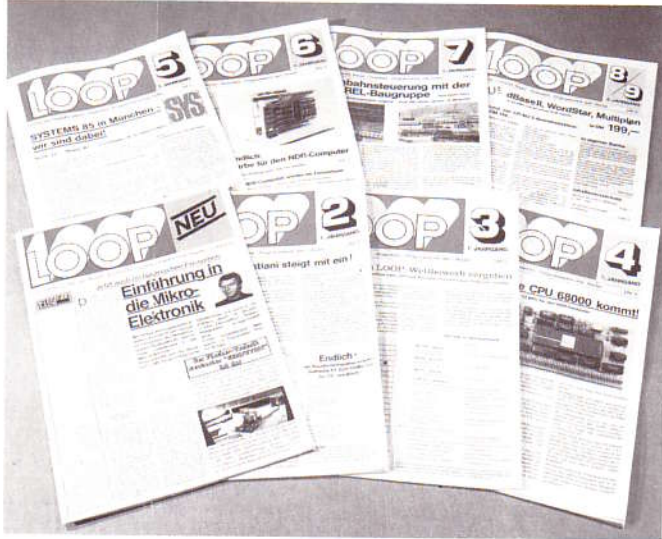
NKC-INFO-DISK 3 IST DA !

Software, Hardware, Tips und Tricks für den NDR-Klein-Computer mit einer 680xx-CPU auf einer Diskette, u. a. Programme für grund 6.0 und die neue GDP64HS! Disk 5 1/4" o. 3 1/2" für JADOS 10,00/12,80 DM zzgl. Porto & NN Info 0,80 DM bei:
Rüdiger Bäcker, Postfach 4111, 5820 Gevelsberg 11

Auch nach dem Kauf:

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

Jahres-Abo DM 20,-, 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
1	10 834	GES-Katalog	10,-
	10 061	LOOP-Abo	20,-

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	20,-

Adresse (umseitig) nicht vergessen!

Datum

Unterschrift

Bei Minderjährigen die des gesetzl. Vertreters

Neue Produkte - Neue Preise!

CPU 8088

Best.-Nr.	Bezeichnung	Preis DM
11266	Platine + Eprom + Pal	148,-
11264	Bausatz	298,-
11265	Fertiggerät	368,-

SPEEDY

Best.-Nr.	Bezeichnung	Preis DM
11222	Diskette, 5 1/4", 80 Spuren	49,-
11223	Diskette, 3 1/2", 80 Spuren	49,-

Logik-Simulator

Best.-Nr.	Bezeichnung	Preis DM
11187	LOGSIM C64	198,-
11246	LOGSIM IBM	248,-

TURBO MODULA

Best.-Nr.	Beschreibung	Preis DM
11248	Diskette, 5 1/4", 80 Spuren	255,-
11249	Diskette, 3 1/2", 80 Spuren	255,-

mc-modular AT

Best.-Nr.	Bezeichnung	Preis DM
11074	Neues Handbuch	30,-

Das neue Grundprogramm für 680xx-Systeme

Best.-Nr.	Beschreibung	Preis DM
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

Best.-Nr.	Beschreibung	Preis DM
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,-

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 _____