

10.11.89

# LOOP

# 23

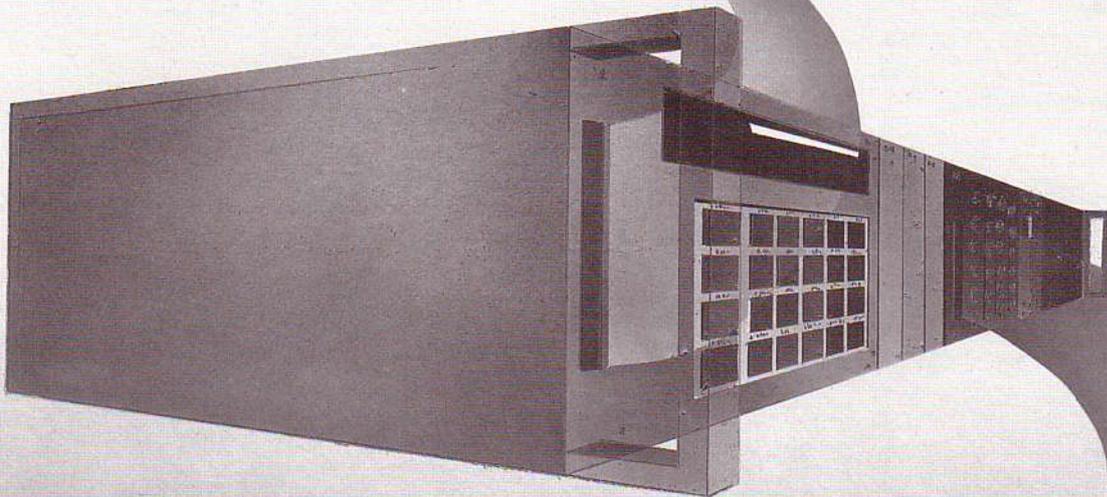
Oktober 1989

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

DM 3,50

# mic

modulare Industrie Computer

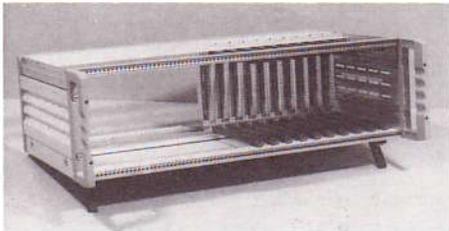


MARTEN

Leitartikel

mic - Der neue Weg in der Hardware Ausbildung

ein modulares Industrie Computersystem, das die Vorteile der NDR-Computer nutzt und ist mit dem Standard ECB-Bus realisiert



Gehäuse

23/4

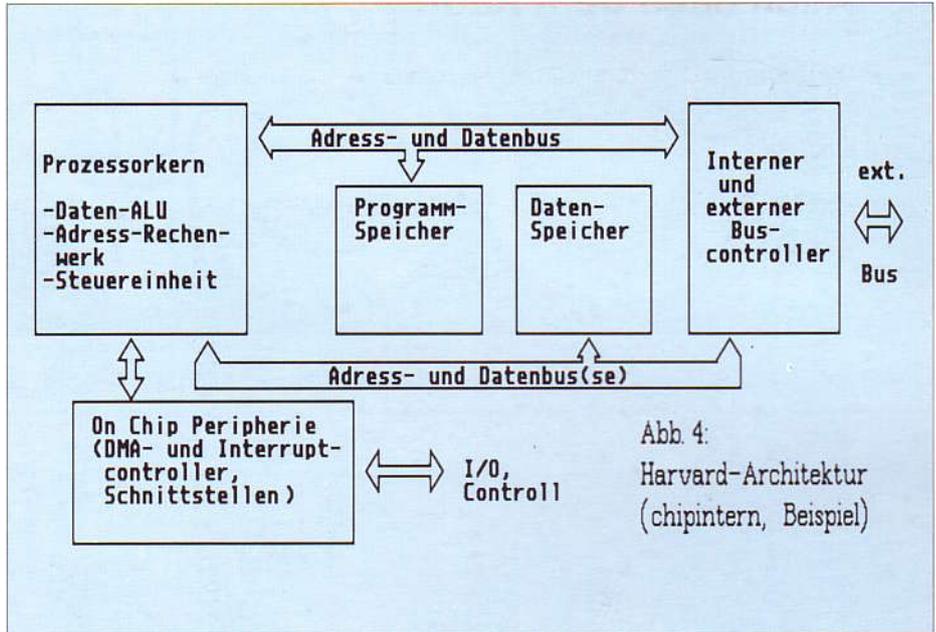


Abb. 4: Harvard-Architektur (chipintern, Beispiel)

CPU Z 80

Hexmon - Texte haben FüÙe bekommen

Der Hexmon bietet sich für Lauftexte gut an, mit einer Einschränkung: die Sieben-segmentanzeige

23/5

Ökosystem eines Interpreters

Teil 4: Geheimsprache

23/8

Grafik zu Fuß programmiert

Teil 2: die Rückkehr der GDP Vektoren

23/10

CPU 680XX

Mausgesteuerte Menüauswahl mit Grundprogramm ab v. 6.0

Das Programm ermöglicht die Anzeige von 30 Menüzeilen und 2 Statuszeilen auf dem Bildschirm

23/13

Wichtige Adressen für den NDR-Computer

23/17

New Technologies - oder wie lehre ich einem AT das Fürchten

Teil 1: Ein Einblick in die Welt der "Digitalen Signalprozessoren"

23/19

Der Jados Patchbereich

23/26

Jetzt wird gelesen

Teil 5: Lesen von Dateien und Disketten

23/24

mc-modular-AT

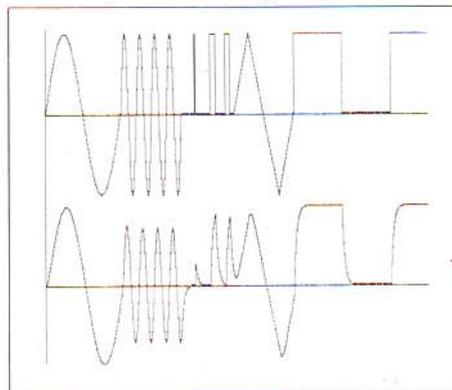
CPU8088 - MS/DOS

Teil 2: PC-Basiswissen

23/27

DSP zum "reinschnuppert" mit dem NDR-Computer

Teil 1: Digitale Filtrierung



Ergebnis einer digitalen Tiefpassfiltrierung.

Verwendung eines Erweiterungsspeichers unter MS-DOS oder PC-DOS

MS-DOS und PC-DOS bieten die Möglichkeit, einen Teil des Haupt- oder Erweiterungsspeichers als logisches Laufwerk zu verwenden.

23/35

Rubriken

Editorial

23/3

Comic - Strip

23/7

Aus der Technik

SCSI Festplatten für den NDR

23/37

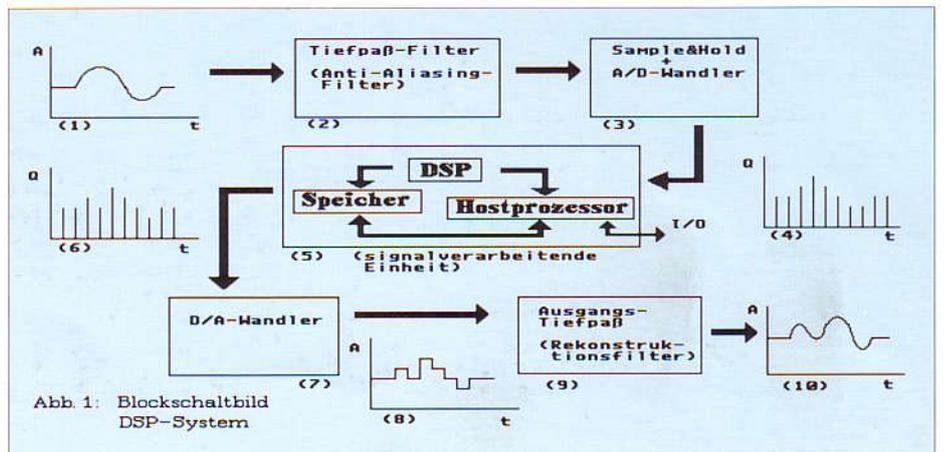


Abb. 1: Blockschaltbild DSP-System

**Korrektur und Ergänzung**  
KEY3 r2

23/38

**Für Sie gelesen:**

**Turbopascal 4.0/5.0 kompakt**  
Auperle

23/18

**Informationsbeschaffung mit dem PC**  
Mc-Graw-Hill-Verlag

23/12

**Tewi OS/2 - Einführung und Referenz**  
J. Emmet Beam

23/33

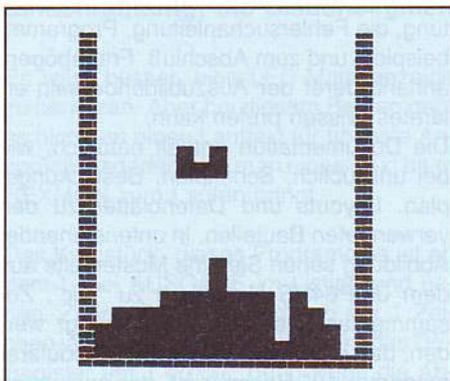
**Kleinanzeigen**

23/32

**Jetzt lieferbar**

Für die ganze Familie

23/36



**Hardcopy Tetris**

**Impressum**

**LOOP Zeitung für Computerbauer**

**Herausgeber:** Gerd Graf

**Redaktion:** Rolf Dieter Klein, Gerd Graf, Ralph Pawlowitz

**Gestaltung:** bbs Computer Systeme  
Elisabeth Mayr, Marten Heuckelbach

**Druck:** Karl-Heinz-Rieder, Kempten

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

**Anzeigenpreisliste:** 10/89

**Editorial**

**Ausbildung ist unser Metier**

Wie bereits in der letzten LOOP angekündigt, wird jetzt zur Systems 89 das neue 19" Ausbildungssystem "mic" vorgestellt. Damit beschreiten wir ein neuen Weg in der Ausbildung, bei dem wir die bekannten Vorzüge des NDR-Computers nutzen, und den mechanischen Rahmen in der 19" Technik neu konzipiert haben.

Dies allerdings macht noch nicht allein die Ausbildungsfähigkeit des "mic" Systems. Es ist vielmehr die Dokumentation, die dem Ausbilder oder dem Lernwilligen ein umfangreiches und detailliertes Hilfsmittel in die Hand gibt. Der Leitartikel verrät ihnen alles Wissenswerte zum Thema "mic".

**EGA und NDR**

In der letzten LOOP war die Rede von einer EGA-Karte auf NDR-Bus. Diese Baugruppe sollte die NDR - 8088 Graphikmisere beheben und ein für allemal, zumindest im MS-DOS Bereich für einen Standard sorgen. Leider ist das Interesse der Kunden nicht in dem Umfang vorhanden, wie wir uns dies vorgestellt haben. Viele sehen wohl den Umweg über die BUSKOPP-Karte und einer Standard-PC-EGA-Karte als den einfacheren und vielleicht sogar billigeren Weg, zumal man ja oft auch noch eine billige PC-Joystickkarte, PC-Schnittstellenkarte hat, die man am NDR-System auch noch betreiben will. So stellt sich für uns natürlich die Frage, ob wir die EGA-Karte überhaupt für den NDR-Bus machen sollen; für den ECB-Bus ist die EGA-Karte schon in der Testphase und steht auch nicht zur Diskussion, da es beim ECB-Bus auch keine BUSKOPP gibt.

Nun - um nicht lange um den Brei herumzureden - wir sind auf Ihre Meinung und Mitarbeit angewiesen.

Es ist wirtschaftlich nur dann möglich, eine solche Karte zu entwickeln, wenn mindestens 100 Baugruppen abgesetzt werden, d.h. wir werden diese Baugruppe nur dann

auf NDR-Bus zum Layout geben, wenn wir diese Menge absetzen werden können. Denn es nützt wohl auch Ihnen als Anwender nichts, wenn eine Standardkarte nur 10 mal verkauft wird. Wo bleibt da die Durchsetzung des Standards? Aber genug - Sie haben das Wort! Schreiben Sie uns Ihre Meinung!

**DSP**

Grundlagen Artikel waren in den letzten LOOP's doch recht selten. Aber mit dieser LOOP hat sich das gewaltig geändert. Zwei Grundlagenartikel zu Thema DSP (digital signal processing) lassen das Technikerherz höher schlagen. Die Artikel sind so verfaßt, daß man auch selbst gleich Versuche mit dem NDR-Computer starten kann.

Nikolaus Bischof  
Technischer Leiter

# mic - Der neue Weg in der Hardware Ausbildung

In Ausbildungsstätten und gewerblich-technischen Schulen konnte sich der NDR-Computer aber aufgrund von zwei Punkten nicht richtig durchsetzen: Zum ersten war dies der nicht genormte NDR-Bus, dessen Steckverbinder nicht verpolsicher sind und zum zweiten die mechanische Stabilität (19"-Technik). Diese Gesichtspunkte und die Tatsache, daß nur ein anderes Ausbildungssystem dieser Art existiert, hat uns dazu bewogen, ein modulares Industrie Computersystem (mic) auf den Markt zu bringen. Dieses "mic"-System nutzt die Vorteile des NDR-Computers (soft- und Hardwarekompatibel) und ist mit dem Standard ECB-Bus realisiert. Die ganze Mechanik, sowie das Gehäuse entsprechen dem 19" Standard. In den folgenden Abbildung sehen Sie das 19"-Gehäuse und eine Steckkarte mit ECB-Bus und Teilefrontplatte.

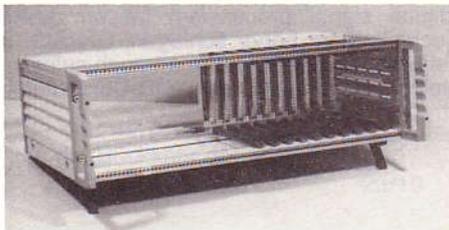


Abb. 1: Gehäuse

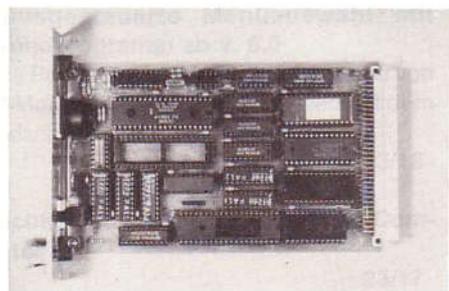


Abb. 2: Baugruppe

Da wir bereits seit einigen Jahren unsere Baugruppen mit der GES-Norm (NDR- und ECB-Bus) layouts, war der Entwicklungsaufwand für dieses neue System nicht so riesig wie man vielleicht vermutet. Es mußten eigentlich nur die CPU-Karten auf ECB-Bus umgelegt werden. Zwei CPU-Karten mit den Prozessoren 8088 und Z80 sind bereits kurz vor der Freigabe, sodaß be-

**Der NDR-Computer ist jedem von Ihnen sicher ein Begriff und wohl jeder kann wohl von sich sagen, daß er mit dem NDR-Computer sehr viel über Computer gelernt hat. Vor allem die modulare Struktur, die Offenheit des Systems, die einfach zu bedienende Software und die Möglichkeit, das Systems selbst von Grund auf zusammenzubauen verhalten dem NDR-Computer zum führenden Bausatz Computer und auch zum weit verbreiteten Ausbildungscomputer zu werden.**

reits Anfang November 1989 Z80 und 8088 System serienmäßig geliefert werden können. Zwei Z80-Systeme wurden bereits vorab an ein Berufschulzentrum und an die deutsche Entwicklungshilfe geliefert.

Dieses System soll genauso modular wie der NDR-Computer aufgebaut sein. Aus diesem Grund wollen wir auch dieselbe Palette an CPU-Karten anbieten. Dies werden zuerst die CPU Z80 und die CPU 8088 sein. In der weiteren Planung ist die CPU 68008. Überlegungen über 16 Bit und 32 Bit Bus-Lösungen sind bereits im Gange. Bis November 1989 wollen wir folgende Systeme anbieten.

1. Ein Einsteigerpaket mit HEX-Tastatur und Siebensegmentanzeige. Dieses System wir wie alle anderen Systeme im 19"-Gehäuse eingebaut sein.
2. Ein Grundsystem mit CPU Z80, ASCII-Tastatur, BAS-Monitor und Grundprogramm, SPS-Interpreter, BASIC und Einweg-Assembler-Disassembler auf EPROM
3. Ein Assembler System (Konfiguration wie unter 2.) aber mit Zwei-weg Assembler und FLOMON
4. Ein CP/M 2.2 System mit 3 1/2" Laufwerk (sonst Konfiguration wie unter 2.)
5. Ein MS-DOS System mit CPU8088 mit PC-Tastatur und wahlweise entweder GDP mit BAS-Monitor, oder EGA-Karte und NEC-Multisync II.

Um die Misere der Graphikkarte bei der CPU8088 zu umgehen, haben wir eine eigene Graphikkarte auf ECB-Bus entwickelt. Sie entspricht der EGA-Norm. Eine universelle Schnittstellenkarte (parallele und serielle Schnittstelle) ist in Planung.

Um dieses System jetzt noch ausbildungsgerecht zu gestalten, wird eine komplett auf die Ziele der Ausbildung abgerichtete Dokumentation erstellt. Die Dokumentation wird komplett in DIN A4 erstellt. Sie gliedert sich grob in erklärende Beschreibungen und Arbeitsblätter. Die erklärende

Beschreibung enthält die Beschreibung der Schaltung in drei Stufen. Die erste Stufe ist die Prinzipbeschreibung, in der die prinzipiellen Abläufe, Normen und Fachbegriffe erläutert werden. In der zweiten Stufe wird ein gegliedertes

Blockschaltbild der Baugruppe vorgestellt und die Funktion und das Zusammenwirken der Blöcke bereits erläutert. In der dritten Stufe wird die Schaltung anhand des Schaltplanes erläutert. Hier werden dann Timing-Diagramme von Logikanalysatoren und Oszilloskopen mit eingebunden, um auch komplizierte Zeitabläufe verständlich machen zu können.

Die Arbeitsblätter beinhalten die Stückliste, die Aufbauanleitung, die Testanleitung, die Fehlersuchanleitung, Programmbeispiele und zum Abschluß Fragebögen, anhand derer der Auszubildende sein erlerntes Wissen prüfen kann.

Die Dokumentation enthält natürlich, wie bei uns üblich, Schaltplan, Bestückungsplan, Layouts und Datenblätter zu den verwendeten Bauteilen. In untenstehender Abbildung sehen Sie eine Musterseite aus dem GDP64HS Handbuch zu "mic". Zusammenfassend kann wohl gesagt werden, daß das "mic"-System, ein modularer moderner 19" Rechner ist, der auch nach

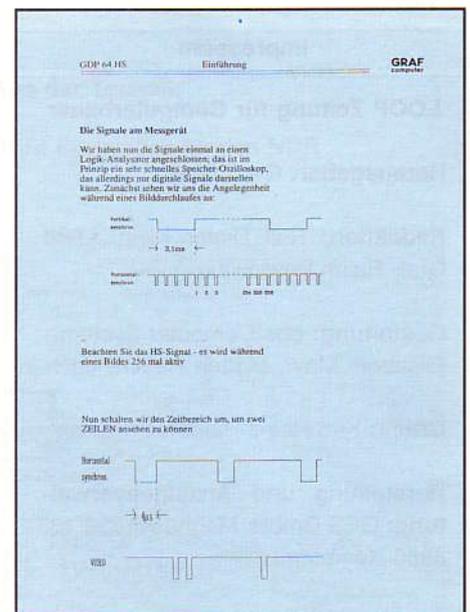


Abb. 3: Seite mic

der Ausbildung als PC verwendet werden kann. Durch seinen konsequenten 19" Aufbau und die industriegerechte Fertigung der Baugruppen und des Gehäuses ist auch ein Einsatz als Steuerrechner in der Industrie durchaus denkbar.

Nähere Informationen über das "mic" System erfahren Sie aus dem 40-seitigen neuen "mic"-Katalog, den sie direkt von uns gegen eine Schutzgebühr von 4,- DM erhalten können.

Stefan Oroz

# HEXMON-Texte haben Füße bekommen

**Überall gibt es Lauftexte (Laufschriften) z.B. in Kaufhäusern, Messehallen, Getränkeautomaten usw. und jetzt gibt es sie auch für den NDR-Computer (Einsteiger-Paket). Der Hexmon bietet sich für Lauftexte gut an, mit einer Einschränkung: die Siebensegmentanzeige.**

Es wäre besser, eine LED-Matrixanzeige zu benutzen. Aber bei diesem Beitrag geht es nicht um einen Lauftext für höchste Ansprüche, sondern wie man einen Text beim HEXMON zum Laufen bringt.

Das Kernstück dieses Programmes ist ab dem Label AUSGABE im Listing und sehen. Zuerst wird abgefragt, ob nur 1 Zeichen laufen soll, wenn ja dann wird das DE-Register um 1 erhöht, denn wenn die Abfrage nicht vorhanden wäre, würde das Programm in eine endlose Schleife laufen und man könnte nur mit der Reset-Taste das Programm beenden.

Ab dem Label LOOP1 wird die Startadresse des zulaufenden Textes ins HL geladen. Beim Label LOOP2 wird die Geschwindigkeit des zu laufenden Textes ins BC geladen. Zwischen den Labeln LAUFTEXT und AUSGABE\_E ist das eigentliche Kernstück des Programmes LAUFTEXT. Dort wird der Text mit Hilfe des Betriebssystems (Procedures PRINT und ANZEIGE) anhand von 8 Zeichen angezeigt. Die Dauer wird durch den Wert, der im Tempo steht, bestimmt.

Danach wird abgefragt, ob nun das letzte Zeichen vom Lauftext erschienen ist, wenn ja dann wiederholt sich das Ganze ab Label AUSGABE. Wenn nicht, dann wird die Startadresse um 1 erhöht und abgefragt, ob die BEF-Taste gedrückt worden ist. Wenn die BEF-Taste gedrückt wurde,

LABEL:	Erklärung:
START	Programm Anfang und Anzeige löschen
MENUE	Ist die Auswahl für DEMO, EDITOR, START_EDIT, ENDE.
ENDE	Programm LAUFTEXT verlassen.
DEMO	Ist die Voreinstellung für den Demotext der anschließend über die Anzeige laufen soll.
START_EDIT	Ist die Voreinstellung für den Lauftext der im EDITOR ab Adresse 8300H eingegeben wurde, der anschließend auch über die Anzeige laufen soll.
FAHRT	Ist eine Abfrag-Routine, wie schnell der Text über die Anzeige laufen soll.
MENUE1	Sprung in die Menue-Auswahl.
EDITOR	Voreinstellung für die Text-Eingabe (ab Adresse 8300H)
EINGABE	Texteingabe für eigene Texte, wenn man die Eingabe beendet, werden die ersten 7 Byte (von 8300 bis 8306 H) der Eingabe, ans Ende von dem Eingabe Editor (Ende der eigenen Text Eingabe)kopiert, damit es wie ein geschlossener Kreis an der Anzeige wirkt.
MENUE2	Sprung zum Label MNUE1
MINUS	Wenn ein Sprung zu diesem Label ausgeführt wird, wird die Editor-Adresse um 1 erniedrigt und man kann einen neuen Wert eingeben.

## Programm-Kurzbeschreibung

kommt man in die Menue-Auswahl, wenn nicht, dann wiederholt sich das Ganze ab Label LOOP2.

Wie im Programm der EDITOR, FAHRT usw. arbeitet, wird nun in der Programm-Kurzbeschreibung erläutert und anschließend folgt die Bedienungsanleitung mit einem Beispiel.

Im Listing ist zu sehen wie das Programm im einzelnen arbeitet. Das vorliegende Listing ist für den Betrieb mit der SBC2 ab Adresse 1000 H vorgesehen, man kann es ohne weiteres ab Adresse 8100 H eingeben. Dann muß man bei Adresse 1003 21 0A [11] den Wert [82] eingeben, z.B. 8103

21 0A 82. Wo es überall geändert werden muß ist im Listing mit eckigen Klammern markiert.

## Programm-Bedienung:

Nach dem Starten des Programmes erscheint an der Anzeige der Text MENUE. Folgende Menue-Auswahl ist möglich:

Taste D	=	DEMOLAUFTEXT
Taste E	=	EDITOR Texteingabe im Hexadezimal
Taste START	=	Starten des eingegebenen Textes
Taste BEF	=	zum Menue oder Programm beenden

**Erklärung zum Demo:**

Wenn man die D-Taste drückt erscheint an der Anzeige der Text TEMPO. Das Programm wartet auf eine zweistellige hexadezimale Eingabe für die Geschwindigkeit des Lauftextes. Es kann ein Wert von 00 - FFH eingegeben werden, wobei 00 der

langsamste und FF die schnellste Geschwindigkeit hat (mit CR-Taste bestätigen). Wenn man keinen Wert eingibt, sondern nur die CR-Taste drückt wird der langsamste Wert übernommen und der DEMOTEXT läuft. Mit der BEF-Taste kann der Lauftext unterbrochen werden und man gelangt in die Menue-Auswahl.

geht um z.B. ein falsches Zeichen zu ändern, dann muß man die restlichen Zeichen mit der CR-Taste bestätigen, sonst wird das zuletzt mit der CR-Taste bestätigte Zeichen übernommen und der Rest (7 Zeichen) überschrieben.

Anzeige:	Eingabe:	Taste drücken:
MENUE		E
EDITOR		CR
8300 00	AF (r)	CR
8301 00		
8300 AF	87 (t)	CR
8301 00	86 (E)	CR
8302 00	92 (S)	CR
8303 00	86 (t)	CR
8304 00		BEF
tempo 00	66	CR
test test		BEF
MENUE		D
tempo 00		CR
demo laufschrift u.s.w.		BEF
MENUE		START
tempo 00	AB	CR
test test		BEF
MENUE		BEF
Hallo1.1		

**Erklärung zum EDITOR:**

Wenn man die E-Taste drückt, erscheint an der Anzeige der Text EDITOR. Es muß eine Taste gedrückt werden, um zur Eingabe zu gelangen. Dann erscheint an der Anzeige: 8300 00 (8300 ist die Anfangs-Adresse des Edi dieser Speicherzelle steht). Der Editor wartet auf die Text- Eingabe in hexadezimal (näheres siehe [1]). Mit der Taste CR wird der Wert übernommen (bei falscher Eingabe kann man mit der Minustaste (-) zurückgehen, siehe im Beispiel). Mit der Taste BEF wird die Eingabe beendet, ohne den evtl. eingegebenen Wert zu übernehmen. Danach kommt die gleiche Ablauffolge wie unter: DEMO (ab Text:TEMPO) beschrieben.

**Erklärung zum Text starten:**

Die Taste START drücken, an der Anzeige erscheint TEMPO (wie bei DEMO usw.).

**Erklärung zum Programm verlassen:**

BEF-Taste drücken und man gelangt zu HALLO 1.1.

**Erklärung zum Beispiel:**

Es soll als Beispiel 'test' als TEXT eingegeben und zuvor eine falsche Eingabe gemacht werden z.B. 'r', anstatt 't' für test (nur um zu zeigen, daß bei Falscheingabe, Korrektur möglich ist.) Sollte jemand mit meiner Programm-Bedienung Schwierigkeiten haben, so empfiehlt es sich, anhand des folg. Beispielles, dieses Schritt für Schritt auszuprobieren.

**Beispiel**

**HINWEIS:**

Wenn man nochmal in den Editor hinein-

Literatur: [1] Mit HEXMON Programme entwickeln (ANHANG C)

```

;*****
;* HEXMON LAUFTEXT V.1.3 29.05.1989 *
;* Stefan Oros *
;* Ludwigstr. 9 *
;* 8901 Stettenhofen (bei Augsburg) *
;* Tel. (0821) 47 11 64 *
;*****
ANZEIGE EQU 00009H ;
HOLETASTE EQU 0000CH ;
PRINT EQU 00015H ;
PRTHL EQU 0001BH ;
GETC EQU 00024H ;
CLEAR EQU 0003FH ;
PORT EQU 00H ;
NEUSTART EQU 00000H ;
ANZFELD EQU 08000H ;
ZWRAM EQU 08260H ; Zwischen-Speicher für AUSGABE
TEMPO EQU 08262H ; Zwischen-Speicher für FAHRT
ZMBYTE EQU 08264H ; Zwischen-Speicher
BYTE EQU 08266H ; für Textlänge
EZWRAM EQU 08268H ; Zwischen-Speicher für EDITOR
ZWA EQU 0826AH ; Zwischen-Speicher für Akkumulator
EDIT_ADR EQU 08300H ; Anfang Adresse des Eingabe Editors

1000 CD 33 00 START:
;
1003 21 0A [11] [82] MENUE: LD HL,MENUETEXT ; lade Adresse ins HL
1006 CD 1E [10] [81] CALL TEXT_AUS ; Text an der Anzeige ausgeben
1009 CD 0C 00 CALL HOLETASTE ; warte bis eine Taste gedrückt ist
100C FE 4B CP 04BH ; ist START-Taste gedrückt ?
100E 28 27 JR Z,START_EDIT ; wenn ja, springe zum Label START_EDIT
1010 FE 47 CP 047H ; ist BEF-Taste gedrückt ?
1012 28 11 JR Z,ENDE ; wenn ja, springe zum Label ENDE
1014 FE 27 CP 027H ; ist E-Taste gedrückt ?
1016 28 48 JR Z,EDITOR ; wenn ja, springe zum Label EDITOR
1018 FE 17 CP 017H ; ist D-Taste gedrückt ?
101A 28 0C JR Z,DEMO ; wenn ja, springe zum Label DEMO
101C 18 E5 JR MENUE ; springe zum Label MENUE

MENUE_E ;
;
TEXT_AUS: CALL PRINT ; ausgeben
1021 CD 09 00 CALL ANZEIGE ; anzeigen
1024 C9 RET ; zurück ins Hauptprogramm

ENDE: ;

1025 C3 00 00 JP NEUSTART ; HEXMON neu starten

DEMO:
1028 21 12 [11] [82] LD HL,DEMOTEXT ; lade Adresse ins HL
102B 22 68 82 LD (EZWRAM),HL ; HL-Adresse zwischenspeichern
102E 11 30 00 LD DE,00030H ; lade DE mit Wert 30H
1031 ED 53 66 82 LD (BYTE),DE ; DE-Adresse zwischenspeichern
1035 18 0E JR FAHRT ; springe zum Label FAHRT

DEMO_E ;
;
START_EDIT:
1037 21 00 83 LD HL,EDIT_ADR ; lade Adresse ins HL
103A 22 68 82 LD (EZWRAM),HL ; HL-Adresse zwischenspeichern
103D ED 58 64 82 LD DE,(ZMBYTE) ; ZMBYTE-Inhalt (Adresse) ins DE
1041 ED 53 66 82 LD (BYTE),DE ; DE-Adresse zwischenspeichern

START_EDIT_E ;
;
FAHRT:
1045 21 49 [11] [82] LD HL,TEMPOTEXT ; lade Adresse ins HL
1048 CD 1E [10] [81] CALL TEXT_AUS ; Text an der Anzeige ausgeben
104B DD 21 06 80 LD IX,ANZFELD+6 ;
104F 01 00 00 LD BC,00000H ; lade BC mit FFH
1052 CD 24 00 CALL GETC ; 2 HEX-Zahlen einlesen und anzeigen
1055 79 LD A,C ; lade C-reg. in Akku
1056 2F CPL ; invertiere Akku-Inhalt
1057 4F LD C,A ; lade Akku-Inhalt ins C-Reg.
1058 ED 43 62 82 LD (TEMPO),BC ; BC-Inhalt zwischenspeichern
105C 18 78 JR AUSGABE ; springe zum Label AUSGABE

FAHRT_E ;
;
MENUE: JR MENUE ; springe zum Label MENUE

EDITOR:
1060 21 51 [11] [82] LD HL,EDITORTTEXT ; lade Adresse ins HL
1063 CD 1E [10] [81] CALL TEXT_AUS ; Text an der Anzeige ausgeben
1066 CD 0C 00 CALL HOLETASTE ; warte bis eine Taste gedrückt ist
1069 CD 33 00 CALL CLEAR ; Anzeige löschen
106C 11 00 00 LD DE,0 ; löschen DE-Inhalt
106F ED 53 66 82 LD (BYTE),DE ; DE-Adresse zwischenspeichern
1073 21 00 83 LD HL,EDIT_ADR ; lade Adresse ins HL
1076 22 68 82 LD (EZWRAM),HL ; HL Adresse zwischenspeichern

EINGABE:
1079 DD 21 00 80 LD IX,ANZFELD ;
107D CD 09 00 CALL ANZEIGE ; anzeigen
    
```

```

1080 CD 18 00      CALL PKTHL      ; HL-Adresse an der Anzeige ausgeben
1083 DD 21 06 80  LD IX,ANZFELD+6 ;
1087 2A 68 82      LD HL,(EDZWRAM); EDZWRAM-Inhalt (Adresse) ins HL
108A 4E            LD C,(HL)      ; laden HL-Inhalt ins C-Reg.
108B CD 24 00      CALL GETC      ; 2 HEX-Zahlen einlesen und anzeigen
108E 32 6A 82      LD (ZWA),A    ; Akku-Inhalt zwischenspeichern
1091 FE 5E         CP 05EH       ; ist "-" Taste gedrückt worden?
1093 28 32         JR Z,MINUS   ; wenn ja, dann springe zum Label MINUS
1095 71            LD (HL),C     ; lade C-Reg.-Inhalt ins HL
1096 23            INC HL       ; HL + 1
1097 22 68 82     LD (EDZWRAM),HL; HL-Adresse zwischenspeichern
109A ED 5B 66 82  LD DE,(BYTE)  ; BYTE-Inhalt (Adresse) ins DE
109E 13            INC DE       ; DE + 1
109F ED 53 66 82  LD (BYTE),DE  ; DE-Adresse zwischenspeichern
10A3 3A 6A 82     LD A,(ZWA)    ; ZWA-Inhalt in Akku laden
10A6 FE 47         CP 047H       ; ist BEF-Taste gedrückt worden?
10A8 20 CF         JR NZ,EINGABE; wenn nicht, springe zum Label EINGABE
10AA 1B            DEC DE       ; DE - 1
10AB ED 53 66 82  LD (BYTE),DE  ; DE-Adresse zwischenspeichern
10AF ED 53 64 82  LD (ZHWYTE),DE; DE-Adresse zwischenspeichern
10B3 06 07         LD B,7        ; lade B-Reg. mit 7
10B5 21 00 83     LD HL,EDIT_ADR; lade Adresse ins HL
10B8 ED 5B 68 82  LD DE,(EDZWRAM); EDZWRAM-Inhalt (Adresse) ins DE
10BC 1B            DEC DE       ; DE - 1
10BD 3E FF         LD A,OFFH    ; lade Akku mit FFH
10BF 12            LD (DE),A    ; lade Akku-Inhalt ins DE
10C0 13            INC DE       ; DE + 1
10C1 ED 80         LDIR        ; kopiere 7 Byte von HL-Inhalt nach DE
10C3 18 80         JR FAHRT    ; springe zum Label FAHRT

EDITOR_E
10C5 18 97        MENUE2: JR MENUE  ; springe zum Label MENUE

MINUS:
10C7 2A 68 82     LD HL,(EDZWRAM); EDZWRAM-Inhalt (Adresse) ins HL
10CA ED 5B 66 82  LD DE,(BYTE)  ; BYTE-Inhalt (Adresse) ins DE
10CE 1B            DEC DE       ; DE - 1
10CF 2B            DEC HL       ; HL - 1
10D0 22 68 82     LD (EDZWRAM),HL; HL-Adresse zwischenspeichern
10D3 ED 53 66 82  LD (BYTE),DE  ; DE-Adresse zwischenspeichern
10D7 18 A0         JR EINGABE   ; springe zum Label EINGABE

AUSGABE:
10D9 ED 5B 66 82  LD DE,(BYTE)  ; BYTE-Inhalt (Adresse) ins DE
10DD 3E 01         LD A,1        ; lade Akku mit 1
10DF BB           CP E         ; ist E-Reg. = 1?
10E0 20 01         JR NZ,LOOP1   ; wenn nein, springe zum Label LOOP1
10E2 1C            INC E         ; E-Reg. + 1

LOOP1:
10E3 2A 68 82     LD HL,(EDZWRAM); EDZWRAM-Inhalt (Adresse) ins HL
10E5 23 60 82     LD (ZWRAM),HL; HL-Adresse zwischenspeichern

LOOP2:
10E9 ED 4B 62 82  LD BC,(TEMPO) ; TEMPO-Inhalt ins BC

LAUFTEXT:
10ED 2A 60 82     LD HL,(ZWRAM) ; ZWRAM-Inhalt (Adresse) ins HL
10F0 CD 1E [0] [81] CALL TEXT_AUS ; Text an der Anzeige ausgeben
10F3 0D            DEC C         ; C-Reg. - 1
10F4 B9           CP C         ; ist CReg. = 0?
10F5 20 F6         JR NZ,LAUFTEXT; wenn nein, springe zum Label LAUFTEXT
10F7 1D            DEC E         ; E-Reg. - 1
10F8 28 DF         JR Z,AUSGABE ; wenn E-Reg. = null dann spring zur AUSGABE
10FA 2A 60 82     LD HL,(ZWRAM) ; ZWRAM-Inhalt (Adresse) ins HL
10FD 23           INC HL       ; HL + 1
10FE 22 60 82     LD (ZWRAM),HL; HL-Adresse zwischenspeichern
1101 CD 09 00     CALL ANZEIGE ;
1104 FE 47         CP 047H       ; ist BEF-Taste gedrückt worden?
1106 28 ED         JR Z,MENUE2  ; wenn ja, dann springe zum Label MENUE
1108 18 DF         JR LOOP2     ; springe zum Label LOOP2

AUSGABE_E
110A AA 86 AB C1 86 FF FF FF MENUETEXT: DEFB 0AAH,086H,0AAH,0C1H,086H,OFFH,OFFH,OFFH

1112 A1 86 AA A3 FF C7 88 C1 DEMOTEXT  DEFB 0A1H,086H,0AAH,0A3H,OFFH,0C7H,088H,0C1H
111A 8E 92 C6 89 AF CF 8E 87  DEFB 08EH,092H,0C6H,089H,0AFH,0CFH,08EH,087H
1122 FF 81 A3 AB FF 92 87 86  DEFB 0FFH,081H,0A3H,0ABH,OFFH,092H,087H,086H
112A 8E 88 AB FF A3 AF A3 B6  DEFB 08EH,088H,0AAH,OFFH,0A3H,0AFH,0A3H,0B6H
1132 FF 87 86 47 FF C0 80 A4  DEFB 0FFH,087H,086H,047H,OFFH,0C0H,080H,0A4H
113A F9 BF 9F F8 F9 82 99     DEFB 0F9H,0BFH,09FH,0F8H,0F9H,082H,099H
1142 FF A1 86 AA A3 FF C7     DEFB 0FFH,0A1H,086H,0AAH,0A3H,OFFH,0C7H

1149 87 86 AA 8C A3 FF FF FF TEMPOTEXT: DEFB 087H,086H,0AAH,08CH,0A3H,OFFH,OFFH,OFFH

1151 86 A1 CF 87 A3 AF FF FF EDITORTXT: DEFB 086H,0A1H,0CFH,087H,0A3H,0AFH,OFFH,OFFH

END
START END

```

Hexmon Listing



# Ökosystem eines Interpreters

## Teil 4: Geheimsprache

Wir lüften nun in dieser Folge das Geheimnis aller Geheimnisse. Zuerst jedoch wollen wir die Übungsaufgabe besprechen.

### 1) Übungsaufgabe

Wenn Sie, wie vorgeschlagen, den Basic-Einzeiler 10 PRINT"AAAA" durch eine weitere Programmzeile, z.B. 20 A\$ = "BBBB" ergänzt haben, so erhalten Sie den in Bild 1 dargestellten Speicherinhalt. Vergleichen Sie nun bitte diesen Speicherinhalt mit dem Bild 2 (bzw. Bild 3 für EHEBAS) aus der letzten Folge. Je nach Interpreterversion müssen Sie die zugehörige Adresse aus der Vergleichstabelle für die verschiedenen HEBAS-Versionen, die in Bild 2 nochmal aufgeführt ist, entnehmen.

```

d47b0,480f
47b0 bc 47 0a 00 95 22 41 41 41 22 00 cc 47 14 00 <G... "AAAA".L.B...
47c0 41 24 20 cd 20 22 42 42 42 22 00 00 00 00 00 00 00 00 00 00 00 00 00
47d0 06 3a 49 45 42 20 37 70 61 7e 20 20 08 08 48 45 45 45 45 45 45 45 45
47e0 42 41 53 20 4e 44 52 20 4b 4c 45 49 4e 20 43 4f 4a 4a 4a 4a 4a 4a 4a
47f0 6d 70 75 74 65 72 20 56 35 2e 31 2b 2b 43 29 20 40 40 40 40 40 40
4800 44 72 2e 20 48 45 48 4c 20 4b 61 6e 72 6b 6b 00 00 00 00 00 00 00 00
  
```

**Bild 1: Speicherinhalt mit den beiden Basic-Programmzeilen**

12 Bytes vom Anfang an ab Adresse 47B0h (Vers. 3.1+) haben sich nicht verändert. Das 13. und 14. Byte enthalten nun nicht mehr den Wert 0, sondern die neue Adresse des Schlußkennzeichens, nämlich 47CCh. Dann folgen die beiden Bytes, die die Zeilennummer darstellen, hier 14 00 für die Zeile 20. Die Bytes mit den Werten 41h und 24h stehen für A\$. Dann folgen ein Leerzeichen (20h) und das Schlüsselzeichen CDh für das Gleichheitszeichen. Die weiteren Bytes sind dann schon bekannt.

**2) Geheimniskrämerei oder ganz klar**  
Nun soll das Geheimnis gelüftet werden. Zwei Schlüsselwörter kennen Sie schon, PRINT mit dem Token 95h und das Gleichheitszeichen mit dem Token CDh. Was stellen nun diese Zahlen dar?

**a) Token-"Rezept"** allgemein würden BASIC-Befehle direkt buchstabeweise im Speicher abgelegt, dann ergebe dies zwei wesentliche Nachteile. Beim Programmablauf müßte der Interpreter alle Buchstaben

**Sollten Sie noch von dem Pseudo-Virus (BASIC-Befehl: CALL 2753 beim Interpreter HEBAS) in der Folge "Schnitzeljagd" infiziert sein, so sind Sie nun in der richtigen Stimmung. Mir hatte es besonders das ERROR #6:ÜV pWUMM (Bild Nr. 6 von Teil 3) angetan, die Bildschirmausgabe ist übrigens mit dem CP/M-Zusatzprogramm MICRO-SHELL in eine Datei umgeleitet worden. Das ging schon vor der MSDOS-Zeit.**

jedes Basic-Befehls analysieren, um den Befehl zu erkennen und dann zur dazugehörigen Maschinensprache-Routine zu springen. Außer dem würde ein Basic-Programm viel Platz im Speicher beanspruchen. Wird jedoch der BASIC-Befehl nach dem Eingeben durch eine Zahl (Token) ersetzt, so kann beim Programmaufbau sofort die Adresse der dazugehörigen Maschinenspracheroutine geholt werden. Es findet also beim Programmieren in BASIC schon eine Teilübersetzung statt.

Ein Nachteil liegt aber wieder darin, daß das im Speicher abgelegte Programm nicht mehr als Textdatei vorliegt. So gibt es beim BASIC-Befehl SAVE die Möglichkeit, mit der Angabe von SAVE"Dateiname",A das Programm tatsächlich buchstabeweise abzulegen. Nun kann man mit einem Textverarbeitungsprogramm Änderungen vornehmen. Beim LOAD-Befehl ist es egal, ob das Programm als Textdatei oder in Tokenform abgelegt wurde. Sie können dies mit dem Debugger DDT.COM überprüfen, indem Sie ein BASIC-Programm unter verschiedenen Namen in beiden Varianten abspeichern und mit DDT.COM anschauen.

Label	HEBAS 3.1+	HEBAS 04	EHEBAS 1.0	EHEBAS 1.1
SASPR01	47b0	4a39	8a1a	8a1a
TOKT84	19cf	1900	186a	186a
PHLTR11	215a	2193h	3221h	220b
TOKT85	1c80	1c0e	1e7b	1e7b
-	1cde	100b	1e99	1e99
FFFF1	1b7a	1b49	1d44	1d44
TOKT81+2	014f	014f	060a	060a
FOR1	02cc	02da	0780	0780
TOK29	0183	0183	06af	06af
RUTD1	07ee	07fc	0c6a	0c6a
TOKT82	11a5h	1179h	132e	132e
SEN	2644h	2801h	28f9	2880

**Bild 2: erweiterte Vergleichstabelle für die im Text enthaltenen Speichersadressen mit Label**

b) Auf geht's! Die Verwendung der Token wollen wir uns jetzt genauer ansehen. In Bild 3 ist ein Ausschnitt aus der Tokentabelle von HEBAS Vers. 3.1+ enthalten, die im Handbuch vollständig aufgeführt ist.

In dieser Tabelle sind Token, BASIC-Befehl und Adresse der Routine aufgeführt. Die Tabelle beginnt mit dem Befehl FOR mit dem Token 81h und endet mit TIME\$ mit FFh (TIME\$ ist aber in HEBAS nicht ausgeführt). Warum aber wurde nicht mit der Zahl 1 begonnen?

Teil 1: Aller Anfang ist schwer LOOP 20
Teil 2: Werkzeug zum Knacken LOOP 21
Teil 3: Schnitzeljagd LOOP 22
Teil 4: Geheimsprache LOOP 23
Teil 5: Ein Boarisch-BASIC LOOP 24
Teil 6: BDOS und BIOS LOOP 25
Teil 7: Innereien LOOP 26
Teil 8: Auf die Plätze, fertig, los LOOP 27
Teil 9: Patchwork mit Variablen LOOP 28
Teil 10: Hexeneinmaleins LOOP 29

Bei Zahlen ab 80h ist das Bit 7 gesetzt (80h = 1000 0000b). So können die Token von Maschinensprachebefehlen (Mnemonics) unterschieden werden. Weiterhin gibt es bei HEBAS sogenannte Zweibyte-Tokens. Das erste Byte hat immer den Wert 80h, das zweite kann



Befehle mit Zweibyte-Token enthalten, so auch Direktbefehle wie LIST, AUTO, EDIT usw.

Adresse:	alter Wert:	neuer Wert:
4A04h	D5	D4
4A0Ah	CD D7 49 FE 00 20 ED	00 00 00 00 00 00 00
4A1Ah	D4	D5

**Bild 6: Änderung in HEBAS G4 um EHEBAS 1.1-Programme zu laden**

Ausklang in der nächsten Folge werden wir diese Tabellen einmal etwas verändern, wobei dies auch ohne Quellcode,

aber halt viel mühsamer geht. Der Anfänger bekommt Tips für die Assemblierung der Quelle auf Diskette und lernt mit Werkzeugen wie M80 oder SLR180 umzugehen.

#### Nachtrag für Aufsteiger

Wer von der EPROM-Version EHEBAS zu HEBAS Version G4 unter CP/M aufsteigt, möchte eventuell seine BASIC-Programme vom Kassettenrecorder auf Diskette überspielen. Dazu enthält HEBAS G4 den BASIC-Befehl CLOAD. Dieser funktionierte bis jetzt jedoch nur mit der EHEBAS-Ver-

sion 1.0. Bei EHEBAS 1.1 werden ja die Variablen mit auf Band abgespeichert, das Programm sozusagen eingefroren. Ein kleiner Patch und HEBAS G4 lädt auch Programme von EHEBAS 1.1. Es müssen in HEBAS G4 folgende Bytes geändert werden, die in Bild 6 enthalten sind. Ab Juli ist dieser Fehler bei HEBAS G4 abgestellt.

Literatur: Röckrath, Luidgar, Microsoft-Basic: Konzepte, Algorithmen, Datenstrukturen, Franzis Computer-Praxis 1985

M. Gujber

# Graphik zu Fuß programmiert

## TEIL 2

(Die Rückkehr der GDP Vektoren)

#### Vektoren zeichnen:

Der Vorteil dieses Graphikprozessors liegt in erster Linie in den Graphikmöglichkeiten. Vor allem die Vektorgraphik ist bei dem GDP sehr gut ausgereift.

Wir unterscheiden hier zwischen drei Arten von Vektoren: Die üblichen Vektoren mit Anfangs- und Endpunkt, die Grundvektoren und die sogenannte Kurzvektoren.

#### Kurzvektoren:

Sie haben den Vorteil, daß mit nur einem Byte der Vektor voll definiert ist, daher sind sehr schnelle Darstellungen möglich.

#### -Länge des Kurzvektors:

Bei ihnen kann die Länge bequem - wenn auch nur in 4 Schritten - in einem Zug mit der Richtungsangabe über Port 70h eingegeben werden. Im Übergabebyte wird dafür Bit 3 u. 4 (X - Länge) sowie Bit 5 u. 6 (Y - Länge) benötigt.

#### -Richtung des Kurzvektors:

Dazu genügt ein Byte, das höchste Bit wird gesetzt und die drei niederwertigsten Bits werden für die Richtungsangabe hergenommen. Zur Verfügung stehen acht Richtungen, die man sich als die vier eines Fadenkreuzes und ihren Winkelhalbierenden vorstellen kann. Beispiel: Es soll ein Kurzvektor mit Schrittlänge 3 in vertikaler Richtung und mit Startpunkt in der Mitte des Bildschirms gezeichnet werden.

**Wer nach der letzten Loop fleißig Buchstaben auf seinen Schirm gebracht hat, kann jetzt sicher gut positionieren und hat ein Gefühl dafür, welche Zahl welche ungefähre Stellung ergibt. Ansonsten werden auch einige Einstellungen, wie Seitenwahl, Bildschirm löschen oder Vorbereitung des Schreibstiftes wieder auftauchen.**

#### Grundvektoren :

Grundvektoren stellen eine Zwischenform von Kurzvektoren und Normalvektoren dar. Von den Kurzvektoren übernehmen sie den Vorteil, daß die Richtung schnell festgelegt (Acht Grundrichtungen) und von den Normalvektoren den Vorteil, daß die Länge bequem eingegeben werden kann. Die Länge des Vektors wird aus den Beträgen der X- und Y-Register berechnet. Diese Achsenabschnitte werden über die Delta-Register eingegeben:

- für den Betrag der X-Koordinate auf Port 75h

- für den Betrag der Y-Koordinate auf Port 77h

#### Beispiel :

Es soll ein 100h langer waagrechter Grundvektor gezeichnet werden. Sein Startpunkt soll links in der Mitte sein. Neu ist hier, daß von dem Kurzvektorbyte nur die Richtungsinformation übernommen wird und die Länge extra in zusätzlichen Registern definiert wird.

Das "warte"-Programm ab Adresse 8900 wird vom letzten Beispiel übernommen. Natürlich kann auch hier die Richtung variiert werden, beispielsweise : Ausgabe

Adresse	Hex Code	Mnemonik	Kommentar
8800	3E 00	d a,0h	Lese und Schreibseite
8802	D3 60	out(60h),a	
8804	CD 00 89	call warte	Aufruf Unterprog.
8807	3E 06	ld a,06h	Bildschirm & Register
8809	D3 70	out(70h),a	löschen
880B	CD 00 89	call warte	Aufruf Unterprogramm
880E	3E 7F	ld a,7Fh	Y-Koordinate in der Mitte
8810	D3 7B	out(7Bh),a	
8812	CD 00 89	call warte	UP
8815	3E FF	ld a,0FFh	Länge waagrecht mit FFh
8817	D3 75	out(75h),a	Delta X-Reg
8819	CD 00 89	call warte	UP
881C	3E 00	ld a,00h	durchgezogene Schrift
881E	D3 72	out(72h),a	
8820	CD 00 89	call warte	UP
8823	3E 00	ld a,00h	Stift aktiviere
8825	D3 70	out(70h),a	
8827	CD 00 89	call warte	UP
882A	3E 18	ld a,18	waagrechter Vektor
882C	D3 70	out(70h),a	
882E	76	HALT	CPU soll anhalten

Grundvektorrichtung "nach unten" an Port 70h : 1ChTabelle :

Richtung (vgl Uhrzeiger)		Code	
3 h	18h	1 h 30 min	19h
12 h	1Ah	10 h 30 min	1Bh
6 h	1Ch	4 h 30 min	1Dh
9 h	1Eh	7 h 30 min	1FhIn

bei diesem Beispiel hat man auch die Möglichkeit, die Vektorlänge zu min 1FhIn diesem Beispiel hat man auch die Möglichkeit, die Vektorlänge (Delta - Register) oder die Strichart zu verändern. Beispielsweise sei ein gepunktet gezeichneter Vektor gewünscht :Ausgabe an CTRL 2 Register auf Port 72h : 1hWahrheitstabelle Strichart

Bit 1	Bit 0	
0	0	durchgezogen
0	1	gepunktet
1	0	gestrichelt
1	1	strichpunkt- tiert

#### Aufgabe:

Stellen sie obigen Grundvektor in gestrichelter Form dar! **Normalvektoren** : Ihre Länge kann jetzt beliebig gewählt werden und damit auch ihre Richtung. Auch die Strichart kann variiert werden, jedoch müssen drei weitere Register benutzt werden. Länge: Es wird auf Port 75h die Länge in X-Richtung und auf Port 77 h die Länge in Y-Richtung eingegeben. Zusätzlich werden aber die Vorzeichen der Achsenkoordinaten an Port 70h benötigt . X - Koord. y - Koord. Hexzahl an Port 70h

+	+	11h
-	+	13h
+	-	15h
-	-	17h

Da mehr Bildpunkte verwaltet als angezeigt werden, können auch Richtungen, festgelegt durch außerhalb des Bildschirms liegende Punkte, angenommen und bis an den Rand gezeichnet werden.- Strichart: Auf Control 2 Register kann unter vier Stricharten, wie nur gestrichelt, gepunktet oder auch beides gewählt werden (siehe auch unter Schriftform der Zeichen)

Adresse	Hex Code	Mnemonik	Kommentar
8800	3E 00	ld a,0h	Lese-und Schreibseite
8802	D3 60	out(60h),a	Null einstellen
8804	CD 00 89	call warte	Aufruf Unterprogramm
8807	3E 06	ld a,06h	Bildschirm löschen
8809	D3 70	out(70h),a	Positionsreg. löschen
880B	CD 00 89	call warte	Aufruf Unterprogramm
880E	3E FF	ld a,	FFh Mitte der X-Achse mit
8810	D3 79	out(79h),a	Reg.A an den Port für die X-Koordinate
8812	CD 00 89	call warte	Aufruf Unterprog.
8815	3E 8	ld a,80h	Mitte der Y-Achse über Port 7Bh eingeben
8817	D3 7B	out(7Bh),a	
8819	CD 00 89	call warte	Aufruf UP
881C	3E 00	ld a,00h	Stiftaktivierung über Kommandoregister
881E	D3 70	out(70h),a	
8820	CD 00 89	call warte	Aufruf Unterprogramm
8823	3E FA	ld a,FAh	Das Datenbyte für den
8825	D3 70	out(70h),a	Kurzvektor eingeben
8827	76	HALT	CPU anhalten
8900	DB 70	in a,(70h)	Unterprogramm 'warte'
8902	E6 04	and 4h	
8804	CA 00 89	jp z,warte	
8807	C9	ret	Sprung ins Hauptprog.

Adresse	Hex Code	Mnemonik	Kommentar
.Z808800	3E 00	ld a,0h	Schreib-und Leseseite
08802	D3 60	out(60h),a	über Port 60h festlege
8804	CD 00 89	call warte	Unterprogramm aufrufen
8807	3E 06	ld a,06h	Lade 6h in Akku
8809	D3 70	out(70h),a	und gib es an Port 70h
880B	CD 00 89	call warte	Aufruf Unterprogramm
880E	3E FF	lda,FFh	Lade Akku mit FFh
8810	D3 79	out(79h),a	Lege den Inhalt des Reg.A an den Port - für die X - Koordinate (70h)
8812	CD 00 89	call warte	Aufruf Unterprogramm
8815	3E 80	ld a,80h	Lade A mit 80h
8817	D3 7B	out(7Bh),a	Inhalt von A an Port 7Bh (Y LSBs) weiter leiten
8819	CD 00 89	call warte	Aufruf in das UP
881C	3E 30	ld a,30h	Delta X des Vektors festlegen
881E	D3 75	out(75h),a	Ausgabe an Port 75h
8820	CD 00 89	call warte	UP
8823	3E 40	ld a,40h	Delta Y des Vektors über
8825	D3 77	out(77h),a	Port 77h ausgeben
8827	CD 00 89	call warte	Aufruf UP
882A	3E 00	ld a,00h	Aktivierung des Schreibstiftes
882C	D3 70	out(70h),a	über Port 70h
882E	CD 00 89	call warte	UP
8831	3E 1A	ld a,1Ah	Richtung : 'nach oben'
8833	D3 70	out(70h),a	über Port 70h
8835	CD 00 89	call warte	UP
8838	3E 00	lda,00h	
883A	D3 70	out(70h),a	Schreibgerät vorbereitet
883C	CD 00 89	call warte	UP
883F	3E 19	ld a,19h	Richtung : 'schräg nach oben'
8841	D3 70	out(70h),a	an Port 70h
8843	CD 00 89	call warte	UP
8846	3E 00		ld a,00h Stift aktivieren
8848	D3 70	out(70h),a	
884A	CD 00 89	call warte	UP
884D	3E 1D	ld a,	1Dh Richtung : "schräg nach unten"
884F	D3 70	out(70h),a	
8851	CD 00 89	call warte	UP
8854	3E 00h	ld a,00h	Stift aktivieren
8856	D3 70	out(70h),a	
8858	CD 00 89	call warte	UP
885B	3E 1E	ld a,1Eh	Richtung : 'nach links'
885D	D3 70	out(70h),a	an Port 70h geben
885F	CD 00 89	call warte	UP

8862	3E 00	ld a,00h	
8864	D3 70	out(70h),a	Stift aktivieren
8866	CD 00 89	call warte	UP
8869	3E 1E	ld a,1Eh	"nochmal nach links"
886B	D3 70	out(70h),a	über Port 70h
886D	CD 00 89	call warte	UP
8870	3E 00h	ld a,00h	Stift aktivieren
8872	D3 70	out(70h),a	
8874	CD 00 89	call warte	UP
8877	3E 1C	ld a,1Ch	Richtung : 'nach unten'
8879	D3 70	out(70h),a	über Port 70h
887B	CD 00 89	call warte	UP
887E	3E 00	ld a,00h	Stift aktivieren
8880	D3 70	out(70h),a	
8882	CD 00 89	call warte	UP
8885	3E 18	ld a,18h	Richtung : 'nach
8877	D3 70	out(70h),a	rechts 'an Port 70h
8889	CD 00 89	call warte	UP
888C	3E 00	ld a,00h	Stift aktivieren
888E	D3 70	out(70h),a	
8890	CD 00 89	call warte	UP
8893	3E 18	ld a,18h	"nochmal rechts"
8895	D3 70	out(70h),a	über Port 70h
8897	CD		
00 89		call warte	UP
889A	3E 00	ld a,00h	
889C	D3 70	out(70h),a	Stift aktivieren
889E	CD 00 89	call warte	UP
88A1	3E 1A	ld a,1Ah	Richtung : 'nach oben'
88A3	D3 70	out(70h),a	an Port 70h legen
endlos:			
88A5	C3 A5 88	jp endlos	Endlosschleife
warte:			
8900	DB 70	in a,(70h)	Daten vom Port 70hein -
8902	E6 04	and 4h	lesen und prüfen ob Bit 2 gesetzt ist
8904	CA 0089	jr z,warte	wenn nicht gesetzt, zur Adresse
			"warte" zurückspringen
8907	C9	ret	nächste Adresse nach dem Aufruf an-
			springen(zurück ins Hauptprogramm )
			Symbols:
8900	warte		
88A5			endlos: Mit diesen Grundmöglichkei-
			tenkann jetzt das Programmieren
			begonnen werden.

**Beispiel :**

Es soll ein Häuschen in die Bildschirmmitte gezeichnet werden

Es ist klar, daß vor den Ausgaben an den verschiedenen Ports, die GDP immer erst noch auf ihre Bereitschaft abgefragt werden will.

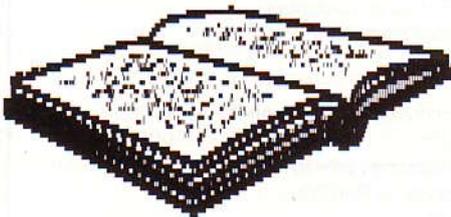
Dieser Teil sollte eine Grundroutine oder Unterprogramm darstellen. Jede Unteroutine springt durch ihren 'ret' - Befehl am Ende wieder in das Hauptprogramm und jedes Hauptprogramm springt am Ende wieder in das Grundprogramm.

Das kann sehr schnell ablaufen und um ein Bild nicht nur Bruchteile von Sekunden sehen zu können , muß man über eine Schleife verhindern, daß in das Grundprogramm (mit seinen Löschroutinen ) zurückgesprungen wird.

Noch ein paar Anregungen ,falls Sie nicht schon optische Untermalungen oder Ergänzungen alter Programme im Sinn haben:

Manche wünschen sich schon lange Bewegung auf dem Bildschirm , durch Umschalten zwischen den Seiten kein Problem.

Wer sich künstlerisch betätigen will kann ja manche Einstellungen zufällig erfolgen lassen und er erhält immer andere Versionen einer bestimmten Graphik .



Jürgen Whals :  
Informationsbeschaffung mit dem PC  
Mc-Graw-Hill-Verlag

**Informationsbeschaffung mit dem PC**

hilft bei einer Anwendung des Rechners, die noch nicht alltäglich ist: der Suche in On-line-Datenbanken.

Für Sie gelesen:

**Informationsbeschaffung mit dem PC**

Wer sich schon einmal selbst mit den ersten nervösen Fingern in eine On-line-Datenbank eingeloggt hat, den Sekundenzeiger vor Augen - mit dem Wissen, daß jede Sekunde kostet - der wird ein solches Werk zu schätzen wissen.

Das Buch enthält ausführliche Abfragebeispiele zu Forschungs- und Entwicklungsprojekten, Konjunktur und Länderinformationen sowie Firmeninformationen.

Weitere Kapitel beschäftigen sich mit Grundlagen des PC's in der Kommunika-

tion sowie öffentlichen Diensten und Netzen. Die Abfragesprache CCL ist ebenso behandelt.

Abgerundet wird das Buch mit einer Liste der europäischen Datenbankanbieter, einer Datexgebührenübersicht und weiteren Informationen.

Zusammengefaßt: ein Buch, das man sicher für die Anwender, die sich mit der zunächst schwierigen Materie des direkten Zugriffs in Datenbanken auseinandersetzen möchten, empfehlen kann.

Rüdiger Bäcker

# Mausgesteuerte Menüauswahl mit Grundprogramm ab v. 6.0

Das Programm ermöglicht die Anzeige von 30 Menüzeilen und 2 Statuszeilen auf dem Bildschirm (Bild 1). Dabei wird eine vorwählbare Zeile invers dargestellt (dunkle Schrift auf hellem Balken). Dieser Balken kann dann mit der Maus auf und ab bewegt werden. Drückt man dann nach Auswahl einer Zeile die linke Maustaste, so wird die Nummer der Zeile in D0.B zurückgeliefert.

Beim Aufruf des Programmes müssen in den Registern A0 und A1 die Adressen von zwei Tabellen übergeben werden. In D0.B steht eine Menüzeilennummer, auf die der Auswahlbalken gesetzt wird, so ist eine Vorgabe der Menüzeile möglich.

## Parameter beim Aufruf :

A0 zeigt auf Tabelle mit Adressdistanz  
A1 zeigt auf Anfang der Texttabelle  
A2 zeigt auf 1. Text für Statuszeile  
A3 zeigt auf 2. Text für Statuszeile  
D0 enthält Nr. der Menüzeile, in der das Fadenkreuz steht

Dabei enthält die erste Tabelle die Adressdistanz zwischen Anfang der Texttabelle und dem abgelegten Text :

### \*+\* Text-Adress-Tabelle

txtatab:		* In dieser Tabelle stehen die Adresse
dc.l	txt1a-txtaber	* der jeweiligen Texte als Adressdistanz
dc.l	txt2a-txtaber	* relativ zum Tabellenanfang, daher
dc.l	txt3a-txtaber	* relokativ
....		
dc.l	txt30a-txtaber	

In der zweiten Tabelle schließlich stehen die Texte :

### \*+\* Textablagegebiet

txtaber:		
txt1a:	dc.b	' Disketteninhaltsverzeichnis',0
txt2a:	dc.b	' Neues Bild von Disk laden',0
...		
txt30a:	dc.b	' Löschmodus ausschalten',0

Auf diese Art ist es möglich, den Text an einer beliebigen Speicherstelle abzulegen (relokative Programme) und die tatsächliche Textadresse mit einer kleinen Routine zu berechnen. Dies geschieht im vorliegenden Programm mit der Unterroutine HO-LEADR. Dieser Routine wird beim Aufruf in Register D0 die Nummer der gewünschten Menüzeile übergeben, als Ausgabe erhält man in A0 die reale Adresse des Textes. Die Routine selbst berechnet zunächst die Speicherstelle in der er-

Heute in der Reihe der TOOL-Programme eine Routine zur mausgesteuerten Menüauswahl. Dieses Programm, daß ursprünglich aus "nur mal so" Gründen entstanden ist, habe ich nach Verfügbarkeit des neuen Grundprogrammes zu einer kompletten Routine ausgearbeitet.

sächliche Textadresse mit einer kleinen Routine zu berechnen. Dies geschieht im vorliegenden Programm mit der Unterroutine HO-LEADR. Dieser Routine wird beim Aufruf in Register

D0 die Nummer der gewünschten Menüzeile übergeben, als Ausgabe erhält man in A0 die reale Adresse des Textes. Die Routine selbst berechnet zunächst die Speicherstelle in der er-

Disketteninhaltsverzeichnis
Neues Bild von Disk laden
Aktuelles Bild auf Disk speichern
Linienart wählen
Schriftart wählen
Schriftgröße verändern
Mausgeschwindigkeit einstellen
Halbe Bildseite vorblättern
Halbe Bildseite zurückblättern
Symbol wählen
Bildschirminhalt in Speicher ablegen
Speicherinhalt auf Bildschirm bringen
Speicherinhalt auf Drucker (klein)
Speicherinhalt auf Drucker (mittel)
Speicherinhalt auf Drucker (groß)
Speicherinhalt auf Drucker (groß & quer)
Punkt setzen
Linie zeichnen
Linienfolge zeichnen —> Untermenü
leeres Quadrat zeichnen
gefülltes Quadrat zeichnen
leeres Rechteck zeichnen
gefülltes Rechteck zeichnen
leeren Kreis zeichnen
gefüllten Kreis zeichnen
leere Ellipse zeichnen
gefüllte Ellipse zeichnen
Fläche füllen
Löschmodus einschalten
Löschmodus ausschalten
MAUSMEN (C) 88 by Rüdiger Bäcker - Postfach 4111 - 5820 Gevelsberg
Bitte mit Maus gewünschte Funktion anwählen und linke Taste drücken

## Bild 1

sten Tabelle, in der die Adressdistanz steht. Dazu wird die Menüzeilennummer mit 4 multipliziert, da die Adressdistanz mit DC.L abgelegt wird, was 4 Byte belegt. Dann wird aus dieser Speicherstelle die Adressdistanz Tabellenanfang—>Textadresse geholt. Dazu wird die Adressierungsart "Adressregister indirekt mit Index" eingesetzt (siehe auch LOOP 18, S. 23ff) eingesetzt, was sehr schnell geht. Die Adressdistanz wird in D! zwischengespeichert und einfach auf A0 aufaddiert. Da A0 ja auf den Anfang der Texttabelle zeigt, steht nun in A0 die tatsächliche Adresse des Textes.

Die Textausgabe erfolgt in den Routinen WRITETXN bzw. WRITETXI; die den Text entweder in Normalschrift (hell auf

dunkel, Stift auf Schreiben) oder in Inversschrift (dunkel auf hell, Stift auf löschen) ausgeben. Dabei wird der Text mit der Grundprogrammroutine CMDWRITE direkt an die GDP übergeben. Dies ist auch der Grund dafür, daß 30 Menüzeilen erreichbar sind. Ein Zeichen benötigt bei der GDP in der Höhe 8 Punkte (7 für Buchstaben, 1 Abstand nächste Zeile). Es sind somit 256 : 8 = 32 Zeilen darstellbar. Es bleiben also die beiden untersten Zeilen noch als Statuszeilen über. Die Wahl des Abstandes von 8 Punkten hat auch noch den Grund, das mit die Y-Position des Fadenkreuzes multipliziert mit 8 die Zeilennummer ergibt. Doch zurück zur Textausgabe. Um die inverse Schrift anzeigen zu können, wird zunächst ein Balken mit MOVETO/DRAWTO gezeichnet (Idee von H. D. Bulwien) und dann mit WRITETXI der Text ausgegeben.

Das Programm läuft nach dem Menüaufbau in einer Schleife ab. Dabei wird die aktuelle Position des Auswahlbalkens mit der alten Position verglichen. Wurde der Balken bewegt, so wird zunächst der alte Balken gelöscht (BALKENNO), der alte Text ausgegeben und dann der neu Balken gezeichnet und mit WRITETXI der neue Text ausgegeben. Weiterhin wird in der Schleife auch geprüft, ob eine Maustaste gedrückt wurde. Diese Abfrage erfolgt zweimal, um eine bessere Entprellung zu erreichen.

Wie ein Aufruf der Routine erfolgt, wird im DEMO-Programm TEST gezeigt.

Auch dieses Programm kann bei mir im JADOS-Format bezogen werden. Dazu ist wieder eine formatierte 5 1/4" oder 3 1/2" Diskette in einem wiederverwertbaren, selbstadressierten und ausreichend frankiertem Umschlag an mich einzusenden.

Alternativ kann das Programm bei mir für 10,- DM inkl. Diskette per NN bezogen werden.

Das Beispiel wurde im übrigen meinem Programm RUBAPAINTE entnommen, welches ich im Moment gerade überarbeite.

```

*+*+*+*
*+* DEMO - PROGRAMM
*+*+*+*

test:
lea txtatab(pc),a1          * Tabellenadressen in A1
lea txtaber(pc),a0         * bzw. A0
lea stattxt1(pc),a2        * Text für Statuszeile
lea stattxt2(pc),a3        * in a2 & a3
moveq #9,d0                * Vorgabewert für Auswahlbalken
bsr mausmen
rts

Der Aufbau der Tabelle wurde ja bereits erklärt. Für Menüs unter 30 Zeilen ist der Text mit DC.B 0 einzugeben :

*+* Textablagegebiet

txtaber:
txt1a:      dc.b           'Disketteninhaltsverzeichnis',0
txt2a:      dc.b           'Neues Bild von Disk laden',0
txt3a:      dc.b 0
....
txt30a:     dc.b 0
    
```

```

23.07.88-19.44 Assprint - (C) 1988 by R. Baecker - Listing des Files : 2:MAUSMEN.S10
Rolf-D.Klein 68000/08 Assembler 4.3 (C) 1984, Seite 1

020000      *   M A U S M E N U E
020000      *
020000      *   PROGRAMM ZUR MAUSGESTEUERTEN MENUEAUSWAHL.
020000      *
020000      *   COPYRIGHT (C) 1988 BY Ruediger Baecker - Postfach 4111 - 5620 Gevelsberg
020000      *
020000      *   V 0.1 - DEMOVERSION - 17.07.88
020000      *   V 0.2 - MIT BEREICHSABGRENZUNG FUER FADENKREUZ - 17.07.88
020000      *   V 0.3 - AUSWAHL DURCH INVERSEN BALKEN - 21.07.88
020000      *   V 0.4 - CODEOPTIMIERUNG - 21.07.88
020000      *   V 0.5 - MIT VORGABEMOEGELICHKEIT FUER AUSWAHLBALKEN - 22.07.88
020000      *   V 0.6 - MIT AUSGABE STATUSZEILE - 23.07.88
020000      *   V 1.0 - FREIGABE FUER L G D P - 23.07.88
020000
020000      *** GRUNDPROGRAMMFUNKTIONEN
020000
020000      WRITLF EQU 6
020000      MOVETO EQU 8
020000      DRAWTO EQU 9
020000      SETPEN EQU 37
020000      ERAPEN EQU 38
020000      CMDPRINT EQU 40
020000      HARDCOPY EQU 125
020000
020000      *** HARDCOPYFUNKTIONEN
020000
020000      MAUSABS EQU *00000001      * FADENKREUZ MIT MAUS STEUERN
020000
020000      *****
020000      *** HAUPTPROGRAMM
020000      *****
020000      *   PARAMETER BEI AUFRUF :
020000      *   A0 ZEIGT AUF TABELLE MIT ADRESSSTANZ
020000      *   A1 ZEIGT AUF ANFANG DER TABELLE
020000      *   A2 ZEIGT AUF 1. TEXT FUER STATUSZEILE
020000      *   A3 AUF 2. ZEILE DER STATUSZEILE
020000      *   D0 ENTHAELT NR. DER MENUEZEILE, IN DER DAS FADENKREUZ STEHT
020000      *
020000      *   PARAMETER NACH RUECKSPRUNG :
020000      *   IN D0 WIRD DIE ZEILENUMMER ZURUECKGEBEN, IN DER DIE LINKE
020000      *   MAUSTASTE GEDRUECKT WURDE
020000      *   ALLE ANDERE REGISTER BLEIBEN ERHALTEN
020000
020000      MAUSMEN:
020000      MOVEM.L D0-D7/A1-A6,-(A7)      * EINSPRUNG A0=ZEIGER AUF TABELLE
020000      BSR MKSTATZEI                  * REGISTER ALLE RETTEN
020004      6100 0172
020000
020008      2448      MOVEA.L A0,A2      * TABELLENSTARTADRESSE RETTEN
02000A      2649      MOVEA.L A1,A3
02000C      323C 001F      MOVE #31,D1      * Y-POSITION IST UMGEGERT ZU MENUE-
020010      9200      SUB.B D0,D1      * ZEILENUMMER !
020012      1001      MOVE.B D1,D0      * REALE NUMMER NUN IN D0
020014      C0FC 000B      MULLU #8,D0
020018      3400      MOVE D0,D2      * MENUEZEILENUMMER SETZEN
02001A      7201      MOVEQ #1,D1      * DEFAULTWERTE FUER BALKEN
02001C      48E7 6000      MOVEM.L D1-D2,-(A7)      * UND RETTEN
020020      4286      CLR.L D6
020022      4285      CLR.L D5      * HILFSREGISTER LOESCHEN
020024      741E      MOVEQ #30,D2      * 30 MENUEZEILEN
020026
020026      WDH:
020026      3005      MOVE D5,D0      * MENUE AUSGEBEN
020028      6100 0134      BSR HOLEADR      * MENUEZEILENUMMER IN D0
020028      6100 0134      BSR HOLEADR      * TEXTADRESSE HOLEN
020028
02002E      6100 00B2      BSR WRITETXN      * TEXT NORMAL AUSGEBEN
020032      5245      ADQ #1,D5      * MENUEZEILENUMMER ERHOEHEN
020034      0C45 001E      CMPI #30,D5      * ALLE 30 ZEILEN AUSGEGEBEN ?
020036      6702      BEQ.S WDH1      * JA, DANN DORT WEITER
020038      60EA      BRA.S WDH      * SONST NACHSTE ZEILE
02003C
02003C      WDH1:
02003C      4CDF 0006      MOVEM.L (A7)+,D1-D2      * DEFAULTWERTE ZURUECK
020040      48E7 6000      MOVEM.L D1-D2,-(A7)      * ABER NOCHMAL MERKEN
020044      4280      CLR.L D0      * NUN INVERSEN BALKEN AUF ZEILE 0 SETZEN
020046      6100 00D0      BSR BALKENIN      * INVERSEN BALKEN ZEICHNEN
02004A      6100 0112      BSR HOLEADR      * TEXTSTARTADRESSE HOLEN
02004E      6100 009C      BSR WRITETXI      * UND TEXT INVERS AUSGEBEN
020052      4CDF 0006      MOVEM.L (A7)+,D1-D2      * DEFAULTWERTE ZURUECK
020056
020056      MH1:
020056      6136      BSR.S FADEN      * FADENKREUZ ANZEIGEN
020058      0C00 00BF      CMPI.B #BF,D0      * TASTE GEDRUECKT ?
02005C      6728      BEQ.S ENDE      * JA, DANN RUECKSPRUNG AUF MAUSMEN
02005E      6170      BSR.S GETPOS      * SONST MENUEZEILENUMMER HOLEN
020060      BC40      CMP D0,D6      * ALTE ZEILE ?
020062      67F2      BEQ.S MH1      * JA, DANN WEITER
020064      48E7 8000      MOVEM.L D0,-(A7)      * SONST POSITION RETTEN
020068      3006      MOVE D6,D0      * ALTE TEXTZEILENUMMER HOLEN
02006A      6100 00B8      BSR BALKENNO      * NORMALEN BALKEN ZEICHNEN
02006C      6100 00EE      BSR HOLEADR      * ADRESSE DES TEXTES HOLEN
020072      616E      BSR.S WRITETXN      * TEXT AUSGEBEN
020074      4CDF 0001      MOVEM.L (A7)+,D0      * NEUE MENUEZEILENUMMER ZURUECKHOLEN
020078      6100 009E      BSR BALKENIN      * BALKEN INVERS ZEICHNEN
02007C      6100 00E0      BSR HOLEADR      * NEUE TEXTADRESSE HOLEN
020080      616A      BSR.S WRITETXI      * TEXT INVERS AUSGEBEN
    
```

```

020082 3C00 MOVE D0,D6 * UND NEUE MENUEZEILENNUMMER MERKEN
020084 80D0 BRA,S HMI * DANN WEITER
020086 ENDE:
020086 1001 MOVE.B D1,D0 * MENUEZEILENNUMMER IN D0
020088 4CDF 7EFF MOVEM.L (A7)+,D0-D7/A1-A6 * REGISTER ALLE RETTEN
02008C 4E75 RTS
02008E *** FADENKREUZ MIT MAUS STEuern - MIT BEREICHSUEBERWACHUNG
02008E * D1.W = X-POSITION (ANPASSUNG AUF BEREICH 0 - 510)
02008E * D2.W = Y-POSITION (ANPASSUNG AUF BEREICH 17 - 254)
02008E * D0.B ENTHAELT NACH RUECKSPRUNG EINE VERT FUER GEDRUECKTE TASTE *
02008E * *BF = LINKS, *7F = RECHTS, *8F = BEIDE, *9F = KEINE
02008E * * D1 & D2 BLEIBEN UNVERAENDERT
02008E FADEN: * SCHLEIFE
02008E 0C41 01FE CMI #510,D1 * X-POS. GROESSER 510 ?
020092 6F04 BLE.S FADEN1 * NEIN, DANN WEITER
020094 323C 01FE MOVE #510,D1 * JA, DANN ANPASSEN
020098 FADEN1:
020098 0C41 0000 CMI #0,D1 * X-POS. KLEINER 0 ?
02009C 6C02 BGE.S FADEN2 * NEIN, DANN WEITER
02009E 7200 MOVEQ #0,D1 * SONST ANPASSEN
0200A0 FADEN2:
0200A0 0C42 00FE CMI #254,D2 * Y-POS. GROESSER 254 ?
0200A4 6F04 BLE.S FADEN3 * NEIN, DANN WEITER
0200A6 343C 00FE MOVE #254,D2 * SONST ANPASSEN
0200AA FADEN3:
0200AA 0C42 0011 CMI #17,D2 * Y-POS. KLEINER 17 ?
0200AE 6C02 BGE.S FADEN4 * NEIN, DANN WEITER

```

Rolf-D.Klein 68000/08 Assembler 4.3 (C) 1984, Seite 3

```

0200B0 7411 MOVEQ #17,D2 * SONST ANPASSEN
0200B2 FADEN4:
0200B2 7001 MOVEQ #MAUSABS0,D0 * UND NUN AUSGEBEN
0200B4 7E7D MOVEQ #HARDCOPY,D7 * CODE FUER MAUSSTEUERUNG IN D0
0200B6 4E41 TRAP #1 * UND HARDCOPYFUNKTION AUFRUFEN
0200B8 0C40 00FF CMI #4FF,D0 * TASTE GEDRUECKT ?
0200BC 6710 BEQ.S FADEN5 * NEIN, DANN WEITER
0200BE 1600 MOVE.B D0,D3 * JA, DANN WERT IN D3 SICHERN UND NOCH-
0200C0 7001 MOVEQ #MAUSABS0,D0 * MAL HOLEN, DAMIT BESSER ENTPRELLT
0200C2 7E7D MOVEQ #HARDCOPY,D7
0200C4 4E41 TRAP #1
0200C6 B600 CMI #D0,D3 * WERTE GLEICH ?
0200C8 6704 BEQ.S FADEN5 * DANN WERT IN D0 LASSEN
0200CA 103C 00FF MOVE.B #4FF,D0 * SONST CODE FUER KEIN TASTENDRUCK IN D0
0200CE FADEN5:
0200CE 4E75 RTS
0200D0 *** POSITION DES FADENKREUZES ALS ZEILENNUMMER (0-31, 0=UNTEN) HOLEN
0200D0 * IN D0.B STEHT DANN ZEILENNUMMER ZWISCHEN 0 - 31
0200D0 * EINGANG = D2 Y-POSITION DES FADENKREUZES
0200D0 GETPOS:
0200D0 4280 CLR.L D0 * ZEILE 0-31 ERRECHNEN
0200D2 1002 MOVE.B D2,D0 * DO LOESCHEN
0200D4 80FC 0006 DIVU #8,D0 * Y-POS. MAUS NACH D0
0200D8 323C 001F MOVE #31,D1 * UND DURCH 8 TEILEN
0200DC 9200 SUB.B D0,D1 * Y-POSITION IST UMGEKEHRT ZU MENUE-
0200DE 1001 MOVE.B D1,D0 * ZEILENNUMMER !
0200E0 4E75 RTS * REALE NUMMER NUN IN D0
0200E2 *** TEXT UEBER CMD-WRITE AUSGEBEN
0200E2 * REGISTER WIE BEI WRITE, GROSSE WIRD JEDOCH IMMER AUF 11 EINGESTELLT
0200E2 * UND IN D0 STEHT MENUEZEILENNUMMER, DIE ANGEPAST WIRD AUF REALE
0200E2 * Y-POSITION
0200E2 * DABEI NORMALES SCHREIBEN ODER MIT ERAPEN MOEGLICH (INVERS AUF BALKEN)
0200E2 WRITETXN:
0200E2 48E7 E000 MOVEM.L D0-D2,-(A7) * TEXT UEBER CMD AUSGEBEN NORMAL
0200E6 7E25 MOVEQ #SETPEN,D7 * REGISTER AUF STACK
0200E8 4E41 TRAP #1 * STIFT AUF SCHREIBEN
0200EA 600C BRA,S WRITETX1 * UND AUSGEBEN
0200EC WRITETX1:
0200EC 48E7 E000 MOVEM.L D0-D2,-(A7) * TEXT UEBER CMD AUSGEBEN INVERS
0200F0 7E25 MOVEQ #ERAPEN,D7 * REGISTER AUF STACK
0200F2 4E41 TRAP #1 * STIFT AUF LOESCHEN
0200F4 6000 0002 BRA WRITETX1 * UND AUSGEBEN
0200F8 WRITETX1:
0200F8 323C 001F MOVE #31,D1 * Y-POSITION IST UMGEKEHRT ZU MENUE-
0200FC 9200 SUB.B D0,D1 * ZEILENNUMMER !
0200FE 1001 MOVE.B D1,D0 * REALE NUMMER NUN IN D0
020100 C0FC 0008 MULU #8,D0 * TATSAECHLICHE Y-POS. BERECHNEN
020104 3400 MOVE D0,D2 * UND IN D2
020106 103C 0011 MOVE.B #11,D0 * GROSSE DER ZEICHEN MUSS 11 SEIN
02010A 323C 0001 MOVE #1,D1 * X-POSITION = 1
02010E 7E28 MOVEQ #CMDPRINT,D7 * UND DANN TEXT AUSGEBEN
020110 4E41 TRAP #1
020112 4CDF 0007 MOVEM.L (A7)+,D0-D2 * REGISTER ZURUECK
020116 4E75 RTS
020118 *** BALKEN AN POSITION ZEICHEN

```

Rolf-D.Klein 68000/08 Assembler 4.3 (C) 1984, Seite 4

```

020118 BALKENIN:
020118 48E7 F000 MOVEM.L D0-D3,-(A7) * INVERSEN BALKEN AN D1,D2 ZEICHEN
02011C 7E25 MOVEQ #SETPEN,D7 * REGISTER RETTEN
02011E 4E41 TRAP #1
020120 7607 MOVEQ #7,D3 * ANZAHL LINIEN
020122 600E BRA,S BALKENNK
020124 BALKENNO:
020124 48E7 F000 MOVEM.L D0-D3,-(A7) * NORMALEN BALKEN AN D1,D2 ZEICHEN
020128 7E25 MOVEQ #ERAPEN,D7 * REGISTER RETTEN
02012A 4E41 TRAP #1
02012C 7607 MOVEQ #7,D3
02012E 6000 0002 BRA BALKENNK
020132 BALKENNK:
020132 323C 001F MOVE #31,D1 * Y-POSITION IST UMGEKEHRT ZU MENUE-
020136 9200 SUB.B D0,D1 * ZEILENNUMMER !
020138 1001 MOVE.B D1,D0 * REALE NUMMER NUN IN D0
02013A C0FC 0008 MULU #8,D0 * TATSAECHLICHE Y-POS. BERECHNEN
02013E 3400 MOVE D0,D2 * UND IN D2
020140 7201 MOVEQ #1,D1
020142 BALKENLP:
020142 7E08 MOVEQ #MOVETO,D7
020144 4E41 TRAP #1
020146 0641 01FE ADD #510,D1
02014A 7E09 MOVEQ #DRAWTO,D7
02014C 4E41 TRAP #1
02014E 5242 ADDQ #1,D2
020150 0441 01FE SUB #510,D1
020154 51CB FFEC DBRA D3,BALKENLP
020158 4CDF 000F MOVEM.L (A7)+,D0-D3
02015C 4E75 RTS
02015E *** TEXTADRESSE AUS TABELLE HOLEN
02015E * DO ENTHAELT MENUEZEILENNUMMER (0 - 31)
02015E * AD GIBT ADRESSE DES TEXTES ZURUECK
02015E HOLEADR:
02015E 48E7 C040 MOVEM.L D0-D1/A1,-(A7) * ADRESSE DES TEXTES AUS NR. ERRECHNEN
020162 204A MOVEA.L A2,A0 * REGISTER AUF STACK RETTEN
020164 224B MOVEA.L A3,A1 * TABELLENADRESSEN HOLEN
020166 4281 CLR.L D1
020168 C0FC 0004 MULU #4,D0 * HILFSREGISTER AUF 0
02016C 2231 0000 MOVE.L 0(A1),D0,D1 * TEXTNUMMER * 4 ALS INDEX ZUM ANFANG
020170 D1C1 ADDA.L D1,A0 * DER ADRESSTABELLE, DORT IST DISTANZ
020172 4CDF 0203 MOVEM.L (A7)+,D0-D1/A1 * DISTANZ AUFFADIEREN = TEXTADRESSE
020176 4E75 RTS * REGISTER ZURUECK
020178 MKSTATZE1:
020178 48E7 F0E0 MOVEM.L D0-D3/A0-A2,-(A7) * REGISTER RETTEN
02017C 760F MOVEQ #15,D3 * 16 LINIEN (DBRA)
02017E 7201 MOVEQ #1,D1 * START BEI X-/
020180 7400 MOVEQ #0,D2 * Y-POSITION 0
020182 MKSLP1:
020182 7E08 MOVEQ #MOVETO,D7
020184 4E41 TRAP #1
020186 0641 01FE ADD #510,D1
02018A 7E09 MOVEQ #DRAWTO,D7
02018C 4E41 TRAP #1

```

Rolf-D.Klein 68000/08 Assembler 4.3 (C) 1984, Seite 5

```

02018E 5242 ADDQ #1,D2
020190 0441 01FE SUB #510,D1
020194 51CB FFEC DBRA D3,MKSLP1
020198 204A MOVEA.L A2,A0 * TEXTSTART IN A0
02019A 7E25 MOVEQ #ERAPEN,D7 * STIFT AUF LOESCHEN
02019C 4E41 TRAP #1
02019E 7201 MOVEQ #1,D1 * START BEI X-/
0201A0 7408 MOVEQ #8,D2 * Y-POSITION 8
0201A2 103C 0011 MOVE.B #11,D0 * GROSSE DER ZEICHEN MUSS 11 SEIN
0201A6 7E28 MOVEQ #CMDPRINT,D7 * UND DANN TEXT AUSGEBEN
0201A8 4E41 TRAP #1
0201AA 204B MOVEA.L A3,A0 * UND 2. ZEILE
0201AC 7201 MOVEQ #1,D1 * START BEI X-/
0201AE 7400 MOVEQ #0,D2 * Y-POSITION 0
0201B0 103C 0011 MOVE.B #11,D0 * GROSSE DER ZEICHEN MUSS 11 SEIN
0201B4 7E28 MOVEQ #CMDPRINT,D7
0201B6 4E41 TRAP #1
0201B8 4CDF 070F MOVEM.L (A7)+,D0-D3/A0-A2 * REGISTER ZURUECK
0201BC 4E75 RTS
0201BE *** ENDE MAUSMEN
0201BE *****
0201BE *** D E H O - P R O G R A M M
0201BE *****
0201BE TEST:
0201BE 43FA 0016 LEA T1TAB(PC),A1 * TABELLENADRESSEN IN A1
0201C2 41FA 006A LEA T1TAB(PC),A0 * BZW. A0
0201C6 45FA 0300 LEA STATTX1(PC),A2 * TEXT FUER STATUSZEILE
0201CA 47FA 0413 LEA STATTX2(PC),A3 * IN A2 & A3
0201CE 7009 MOVEQ #9,D0 * VORGABEWERT FUER AUSWAHLBALKEN
0201D0 6100 FE2E BSR MAUSMEN
0201D4 4E75 RTS
0201D6 *** TEXT-ADRESS-TABELLE

```

```

0201D6
0201D6      TXTAB:
0201D6 00000000 DC.L TX1A-TXTABER
0201DA 0000001D DC.L TX12A-TXTABER
0201DE 00000038 DC.L TX13A-TXTABER
0201E2 0000005B DC.L TX14A-TXTABER
0201E6 0000006E DC.L TX15A-TXTABER
0201EA 00000082 DC.L TX16A-TXTABER
0201EE 0000009D DC.L TX17A-TXTABER
0201F2 000000B0 DC.L TX18A-TXTABER
0201F6 000000C6 DC.L TX19A-TXTABER
0201FA 000000F0 DC.L TX110A-TXTABER
0201FE 0000010D DC.L TX111A-TXTABER
020202 00000133 DC.L TX112A-TXTABER
020206 0000015A DC.L TX113A-TXTABER
02020A 0000017E DC.L TX114A-TXTABER
02020E 000001A3 DC.L TX115A-TXTABER
020212 000001C7 DC.L TX116A-TXTABER
020216 000001F2 DC.L TX117A-TXTABER
02021A 00000200 DC.L TX118A-TXTABER
02021E 00000210 DC.L TX119A-TXTABER
020222 00000235 DC.L TX120A-TXTABER
020226 0000024E DC.L TX121A-TXTABER
02022A 00000268 DC.L TX122A-TXTABER
02022E 00000285 DC.L TX123A-TXTABER
020232 000002A3 DC.L TX124A-TXTABER
    
```

Rolf-D.Klein 68000/08 Assembler 4.3 (C) 1984, Seite 6

```

020236 000002BA DC.L TX125A-TXTABER
02023A 000002D5 DC.L TX126A-TXTABER
02023E 000002EC DC.L TX127A-TXTABER
020242 00000307 DC.L TX128A-TXTABER
020246 00000318 DC.L TX129A-TXTABER
02024A 00000331 DC.L TX130A-TXTABER
02024E
02024E      *** TEXTABLAGEGEBIET
02024E
02024E      TXTABER:
02024E 20446973686574 TX1A: DC.B ' Disketteninhaltsverzeichnis',0
020255 74656E696E6861 TX12A: DC.B ' Neues Bild von Disk laden',0
02025C 6C747376657274 TX13A: DC.B ' Aktuelles Bild auf Disk speichern',0
020263 656963686E6973 TX14A: DC.B ' Linienart waehlen',0
02026A 00 TX15A: DC.B ' Schriftart waehlen',0
02026B 20466575657320 TX16A: DC.B ' Schriftgrosse veraendern',0
020272 42696C6420766F TX17A: DC.B ' Mausgeschwindigkeit einstellen',0
020279 6E204469736B20 TX18A: DC.B ' Symbol waehlen',0
020280 6C6164656E00 TX19A: DC.B ' Halbe Bildseite vorblaettern',0
020286 2041687475656C TX20A: DC.B ' Halbe Bildseite zurueckblaettern',0
02028D 6C65732042696C TX21A: DC.B ' Speicherinhalt auf Drucker (klein)',0
020294 64206175662044 TX22A: DC.B ' Speicherinhalt auf Drucker (gross)',0
02029B 69736820737065 TX23A: DC.B ' gefuelliges Quadrat zeichnen',0
0202A2 69636865726E00 TX24A: DC.B ' gefuelliges Rechteck zeichnen',0
0202A9 204C696E69656E TX25A: DC.B ' gefuelliger Kreis zeichnen',0
0202B0 6172420776165 TX26A: DC.B ' leeres Quadrat zeichnen',0
0202B7 686C656E00 TX27A: DC.B ' leeres Rechteck zeichnen',0
0202BC 20536368726966 TX28A: DC.B ' gefuelliges Rechteck zeichnen',0
0202C3 74617274207761 TX29A: DC.B ' leeres Kreis zeichnen',0
0202CA 65686C656E00 TX30A: DC.B ' leeren Kreis zeichnen',0
0202D0 20536368726966 TX31A: DC.B ' gefuelliger Kreis zeichnen',0
0202D7 7467726F657373 TX32A: DC.B ' leere Ellipse zeichnen',0
0202DE 65207665726165 TX33A: DC.B ' gefuelliger Kreis zeichnen',0
0202E5 6E6465726E00 TX34A: DC.B ' leere Ellipse zeichnen',0
0202EB 20486175736765 TX35A: DC.B ' leere Ellipse zeichnen',0
0202F2 73636877696E64 TX36A: DC.B ' leere Ellipse zeichnen',0
0202F9 69676865697420 TX37A: DC.B ' leere Ellipse zeichnen',0
020300 65696E7374656C TX38A: DC.B ' leere Ellipse zeichnen',0
020307 6C656E00 TX39A: DC.B ' leere Ellipse zeichnen',0
02030B 2048616C626520 TX40A: DC.B ' leere Ellipse zeichnen',0
020312 42696C64736569 TX41A: DC.B ' leere Ellipse zeichnen',0
020319 746520766F7262 TX42A: DC.B ' leere Ellipse zeichnen',0
020320 6C616574746572 TX43A: DC.B ' leere Ellipse zeichnen',0
020327 6E00 TX44A: DC.B ' leere Ellipse zeichnen',0
020329 2048616C626520 TX45A: DC.B ' leere Ellipse zeichnen',0
020330 42696C64736569 TX46A: DC.B ' leere Ellipse zeichnen',0
020337 7465207A757275 TX47A: DC.B ' leere Ellipse zeichnen',0
02033E 656368626C6165 TX48A: DC.B ' leere Ellipse zeichnen',0
020345 747465726E00 TX49A: DC.B ' leere Ellipse zeichnen',0
02034B 2053796D626F6C TX50A: DC.B ' leere Ellipse zeichnen',0
020352 20776165686C65 TX51A: DC.B ' leere Ellipse zeichnen',0
020359 6E00 TX52A: DC.B ' leere Ellipse zeichnen',0
02035B 2042696C647363 TX53A: DC.B ' leere Ellipse zeichnen',0
020362 6869726D696E68 TX54A: DC.B ' leere Ellipse zeichnen',0
020369 616C7420696E20 TX55A: DC.B ' leere Ellipse zeichnen',0
020370 53706569636865 TX56A: DC.B ' leere Ellipse zeichnen',0
020377 722061626C6567 TX57A: DC.B ' leere Ellipse zeichnen',0
02037E 656E00 TX58A: DC.B ' leere Ellipse zeichnen',0
020381 20537065696368 TX59A: DC.B ' leere Ellipse zeichnen',0
020388 6572696E68616C TX60A: DC.B ' leere Ellipse zeichnen',0
02039F 74206175662042 TX61A: DC.B ' leere Ellipse zeichnen',0
020396 696C6473636869 TX62A: DC.B ' leere Ellipse zeichnen',0
02039D 726D206272696E TX63A: DC.B ' leere Ellipse zeichnen',0
0203A4 67656E00 TX64A: DC.B ' leere Ellipse zeichnen',0
    
```

Rolf-D.Klein 68000/08 Assembler 4.3 (C) 1984, Seite 7

```

0203AB 20537065696368 TX13A: DC.B ' Speicherinhalt auf Drucker (klein)',0
0203AF 6572696E68616C TX14A: DC.B ' Speicherinhalt auf Drucker (mittel)',0
0203B6 74206175662044 TX15A: DC.B ' Speicherinhalt auf Drucker (gross & quer)',0
0203BD 72756368657220 TX16A: DC.B ' Speicherinhalt auf Drucker (gross & quer)',0
0203CA 28686C65686E29 TX17A: DC.B ' Speicherinhalt auf Drucker (gross & quer)',0
0203CB 00 TX18A: DC.B ' Speicherinhalt auf Drucker (gross & quer)',0
0203CC 20537065696368 TX19A: DC.B ' Speicherinhalt auf Drucker (gross & quer)',0
0203D3 6572696E68616C TX20A: DC.B ' Speicherinhalt auf Drucker (gross & quer)',0
0203DA 74206175662044 TX21A: DC.B ' Speicherinhalt auf Drucker (gross & quer)',0
0203E1 72756368657220 TX22A: DC.B ' Speicherinhalt auf Drucker (gross & quer)',0
    
```

```

0203E8 286D697474656C TX23A: DC.B ' gefuelliges Rechteck zeichnen',0
0203EF 2900 TX24A: DC.B ' gefuelliger Kreis zeichnen',0
0203F1 20537065696368 TX25A: DC.B ' gefuelliger Kreis zeichnen',0
0203FB 6572696E68616C TX26A: DC.B ' gefuelliger Kreis zeichnen',0
0203FF 74206175662044 TX27A: DC.B ' gefuelliger Kreis zeichnen',0
020406 72756368657220 TX28A: DC.B ' gefuelliger Kreis zeichnen',0
02040D 2867726F737329 TX29A: DC.B ' gefuelliger Kreis zeichnen',0
020414 00 TX30A: DC.B ' gefuelliger Kreis zeichnen',0
020415 20537065696368 TX31A: DC.B ' gefuelliger Kreis zeichnen',0
02041C 6572696E68616C TX32A: DC.B ' gefuelliger Kreis zeichnen',0
020423 74206175662044 TX33A: DC.B ' gefuelliger Kreis zeichnen',0
02042A 72756368657220 TX34A: DC.B ' gefuelliger Kreis zeichnen',0
020431 2867726F737329 TX35A: DC.B ' gefuelliger Kreis zeichnen',0
020438 26207175657229 TX36A: DC.B ' gefuelliger Kreis zeichnen',0
02043F 00 TX37A: DC.B ' gefuelliger Kreis zeichnen',0
020440 2050756E687420 TX38A: DC.B ' gefuelliger Kreis zeichnen',0
020447 73657474656E00 TX39A: DC.B ' gefuelliger Kreis zeichnen',0
02044E 204C696E696520 TX40A: DC.B ' gefuelliger Kreis zeichnen',0
020455 74656963686E65 TX41A: DC.B ' gefuelliger Kreis zeichnen',0
02045C 6E00 TX42A: DC.B ' gefuelliger Kreis zeichnen',0
02045E 204C696E69656E TX43A: DC.B ' gefuelliger Kreis zeichnen',0
020465 666F6C6765207A TX44A: DC.B ' gefuelliger Kreis zeichnen',0
02046C 656963686E656E TX45A: DC.B ' gefuelliger Kreis zeichnen',0
020473 202D203E20556E TX46A: DC.B ' gefuelliger Kreis zeichnen',0
02047A 74657269656E75 TX47A: DC.B ' gefuelliger Kreis zeichnen',0
020481 6500 TX48A: DC.B ' gefuelliger Kreis zeichnen',0
020483 206C6565726573 TX49A: DC.B ' gefuelliger Kreis zeichnen',0
02048A 20517561647261 TX50A: DC.B ' gefuelliger Kreis zeichnen',0
020491 74207A65696368 TX51A: DC.B ' gefuelliger Kreis zeichnen',0
020498 6E656E00 TX52A: DC.B ' gefuelliger Kreis zeichnen',0
02049C 2067656875656C TX53A: DC.B ' gefuelliger Kreis zeichnen',0
0204A3 6C746573205175 TX54A: DC.B ' gefuelliger Kreis zeichnen',0
0204AA 6164726174207A TX55A: DC.B ' gefuelliger Kreis zeichnen',0
0204B1 656963686E656E TX56A: DC.B ' gefuelliger Kreis zeichnen',0
0204B8 00 TX57A: DC.B ' gefuelliger Kreis zeichnen',0
0204B9 206C6565726573 TX58A: DC.B ' gefuelliger Kreis zeichnen',0
0204C0 20526563687465 TX59A: DC.B ' gefuelliger Kreis zeichnen',0
0204C7 6368207A656963 TX60A: DC.B ' gefuelliger Kreis zeichnen',0
0204CE 686E656E00 TX61A: DC.B ' gefuelliger Kreis zeichnen',0
0204D3 2067656875656C TX62A: DC.B ' gefuelliger Kreis zeichnen',0
0204DA 6C746573205265 TX63A: DC.B ' gefuelliger Kreis zeichnen',0
0204E1 63687465636820 TX64A: DC.B ' gefuelliger Kreis zeichnen',0
0204E8 74656963686E65 TX65A: DC.B ' gefuelliger Kreis zeichnen',0
0204EF 6E00 TX66A: DC.B ' gefuelliger Kreis zeichnen',0
0204F1 206C656572656E TX67A: DC.B ' gefuelliger Kreis zeichnen',0
0204F8 20487265697320 TX68A: DC.B ' gefuelliger Kreis zeichnen',0
0204FF 74656963686E65 TX69A: DC.B ' gefuelliger Kreis zeichnen',0
020506 6E00 TX70A: DC.B ' gefuelliger Kreis zeichnen',0
020508 2067656875656C TX71A: DC.B ' gefuelliger Kreis zeichnen',0
02050F 6C74656E204872 TX72A: DC.B ' gefuelliger Kreis zeichnen',0
020516 656973207A6569 TX73A: DC.B ' gefuelliger Kreis zeichnen',0
02051D 63686E656E00 TX74A: DC.B ' gefuelliger Kreis zeichnen',0
020523 206C6565726520 TX75A: DC.B ' gefuelliger Kreis zeichnen',0
02052A 456C6970736520 TX76A: DC.B ' gefuelliger Kreis zeichnen',0
    
```

Rolf-D.Klein 68000/08 Assembler 4.3 (C) 1984, Seite 8

```

020531 74656963686E65 TX77A: DC.B ' gefuelliger Kreis zeichnen',0
020538 6E00 TX78A: DC.B ' gefuelliger Kreis zeichnen',0
02053A 2067656875656C TX79A: DC.B ' gefuelliger Kreis zeichnen',0
020541 6C746520456C69 TX80A: DC.B ' gefuelliger Kreis zeichnen',0
020548 707365207A6569 TX81A: DC.B ' gefuelliger Kreis zeichnen',0
02054F 63686E656E00 TX82A: DC.B ' gefuelliger Kreis zeichnen',0
020555 20466C61656368 TX83A: DC.B ' gefuelliger Kreis zeichnen',0
02055C 65206675656C6C TX84A: DC.B ' gefuelliger Kreis zeichnen',0
020563 656E00 TX85A: DC.B ' gefuelliger Kreis zeichnen',0
020566 204C6F65736368 TX86A: DC.B ' gefuelliger Kreis zeichnen',0
02056D 6D6F6475732065 TX87A: DC.B ' gefuelliger Kreis zeichnen',0
020574 696E736368616C TX88A: DC.B ' gefuelliger Kreis zeichnen',0
02057B 74656E00 TX89A: DC.B ' gefuelliger Kreis zeichnen',0
02057F 204C6F65736368 TX90A: DC.B ' gefuelliger Kreis zeichnen',0
020586 6D6F6475732061 TX91A: DC.B ' gefuelliger Kreis zeichnen',0
02058D 7573736368616C TX92A: DC.B ' gefuelliger Kreis zeichnen',0
020594 74656E00 TX93A: DC.B ' gefuelliger Kreis zeichnen',0
020598 TX94A: DC.B ' gefuelliger Kreis zeichnen',0
020599 STATTX1:
020599 DC.B ' MAUSMEN (C) B8 by Ruediger Baecker - Postfach 4111 - 5820 Gevelsberg',0
02059F 454E2028432920 TX95A: DC.B ' gefuelliger Kreis zeichnen',0
0205A6 38382062792052 TX96A: DC.B ' gefuelliger Kreis zeichnen',0
0205AD 75656469676572 TX97A: DC.B ' gefuelliger Kreis zeichnen',0
0205B4 20425165636865 TX98A: DC.B ' gefuelliger Kreis zeichnen',0
0205BB 72202D20506F73 TX99A: DC.B ' gefuelliger Kreis zeichnen',0
0205C2 74686163682034 TX00A: DC.B ' gefuelliger Kreis zeichnen',0
0205C9 313131202D2035 TX01A: DC.B ' gefuelliger Kreis zeichnen',0
0205D0 3832020476576 TX02A: DC.B ' gefuelliger Kreis zeichnen',0
0205D7 656C7362657267 TX03A: DC.B ' gefuelliger Kreis zeichnen',0
0205DE 00 TX04A: DC.B ' gefuelliger Kreis zeichnen',0
0205DF STATTX2:
0205DF DC.B ' Bitte mit Maus gewuenschte Funktion anwaehlen und linke Taste druecken',0
0205DF 20426974746520 TX05A: DC.B ' gefuelliger Kreis zeichnen',0
0205E6 6D6974204D6175 TX06A: DC.B ' gefuelliger Kreis zeichnen',0
0205ED 7320676577655 TX07A: DC.B ' gefuelliger Kreis zeichnen',0
0205FA 6E736368746520 TX08A: DC.B ' gefuelliger Kreis zeichnen',0
0205FB 46756E6874696F TX09A: DC.B ' gefuelliger Kreis zeichnen',0
020602 6E20616E776165 TX10A: DC.B ' gefuelliger Kreis zeichnen',0
020609 686C656E20756E TX11A: DC.B ' gefuelliger Kreis zeichnen',0
020610 64206C696E6865 TX12A: DC.B ' gefuelliger Kreis zeichnen',0
020617 20546173746520 TX13A: DC.B ' gefuelliger Kreis zeichnen',0
02061E 64727565636865 TX14A: DC.B ' gefuelliger Kreis zeichnen',0
020625 6E00 TX15A: DC.B ' gefuelliger Kreis zeichnen',0
020627 END;
088E0 Ende-Symboltabelle
    
```

# Wichtige Adressen für den NDR-Computer

Diese Aufstellung sollte gleichzeitig auch allen Einsteigern helfen, die vielleicht noch nie von einigen Baugruppen (wie z.B. der ACRTC) gehört haben. Mit diesem Standard ist auch die Gefahr ausgeschlossen, daß Hardware-Entwickler bereits belegte

Adressen für ihre neuen Karten vorsehen.

## IO-Adressen - was ist das?

Die IO-Adressen stammen noch aus der Zeit des Z80. Der konnte nur sehr wenig Speicher adressieren (64 KByte). Zusätzlich konnte er aber noch 255 Ein- und Ausgabegeräte ansprechen, die Ports. Diese hatten Nummern (Adressen) von 00 bis ff. Um die Portadressen von den Speicheradressen 00 bis ff unterscheiden zu können, erzeugt der Prozessor ein spezielles Signal bei Ein- oder Ausgabeoperationen. Auf dem Bus heißt dieses Signal IORQ (Input/Output Request, Ein/Ausgabeanforderung).

Im Gegensatz dazu können die 680xx'er Prozessoren sehr viel Speicher ansprechen - kennen aber keine Ports. In 680xx'er Systemen werden spezielle (normale Speicher-) Adressen zum Ansprechen bestimmter Ein- und Ausgabegeräte benutzt.

Da (fast) alle NDR-Baugruppen sowohl im Z80 System als auch mit dem 680xx funktionieren sollen, mußte für die 680xx'er ein Ersatz für die Ports gefunden werden. Der Trick ist einfach: sobald das erste Wort einer Adresse ffff ist, wird das IORQ-Signal erzeugt. Alle Ein- und Ausgabebaugruppen testen nur das zweite Wort aller Busadressen - und das IORQ-Signal. So wird also aus dem Z80 Port 60 die Adresse fffff60 auf dem 68008.

Leider können nun einige neuere Baugruppen, die für IBM Kompatible entworfen wurden, nicht mit dem Z80 angesteuert werden, da der keine Portadressen wie etwa 3b4 erzeugen kann (s.o.). Ausserdem kann die neue SER2 nur einmal verwendet werden.

**Durch ständig neue Baugruppen und immer wieder geänderte Angaben (z.B. bei der SOUND) ist inzwischen einige Verwirrung bezüglich der Adressen einiger Baugruppen entstanden. Deshalb wurde auf der letzten Software-Partnertagung beschlossen, hierfür einen verbindlichen Standard festzulegen. Dazu wurden nun einige kompetente Software-Partner befragt und die Ergebnisse in einer Tabelle zusammengestellt.**

ein Problem: die angesprochenen Karten sitzen beim 68000er und 68020er nur in einem Teil des Busses, genau genommen in einer Hälfte oder einem Viertel. Um dies auszugleichen, müssen alle Adressen mit zwei bzw. vier malgenommen werden.

## Der Trick mit der Adressberechnung

Durch den modularen Aufbau des NDR-Klein-Computers entsteht bei der Adressierung von Ein-/Ausgabegeräten leider

## Dazu ein Beispiel:

Auf einem 68000 ist der 16-Bit Bus aufgeteilt in zwei 8-Bit Hälften. Jede dieser BUS-Platinen ist für gerade bzw. ungerade

Adressen	Belegt durch
f890-f89f	SER2 (serielle Schnittstellen) Nr. 15 und 16
f990-f99f	SER2 (serielle Schnittstellen) Nr. 13 und 14
fa90-fa9f	SER2 (serielle Schnittstellen) Nr. 11 und 12
fb90-fb9f	SER2 (serielle Schnittstellen) Nr. 9 und 10
fc90-fc9f	SER2 (serielle Schnittstellen) Nr. 7 und 8
fd90-fd9f	SER2 (serielle Schnittstellen) Nr. 5 und 6
fe90-fe9f	SER2 (serielle Schnittstellen) Nr. 3 und 4
ff00-ff03	HEXIO (Hexadezimale Ein- und Ausgabe)
ff24-ff27	SCSI (Festplattenkontrolller)
ff40-ff41	indirekt durch alte CENT (Spiegelung)
ff48-ff49	CENT (Drucker)
ff50-ff51	SOUND (Tongenerator)
ff60	GDP64 oder GDP64HS (Grafikkarte)
ff61	GDP64HS (Grafikkarte)
ff68-ff69	KEY (Tastatur)
ff70-ff7f	GDP64 oder GDP64HS (Grafikkarte)
ff80-ff87	PROMER 2 (EPROM-Brenner)
ff88-ff8f	HCOPY/MAUS (Anschluß für Atari-Maus)
ff90-ff9f	SER2 (serielle Schnittstellen) Nr. 1 und 2
ffa4-ffa6	CLUT (Farbtabelle)
ffa8-ffa9	ACRTC (Grafikkarte)
ffac-ffaf	COL (Farbgrafikkarte)
ffb4-ffbf	Hercules (Grafikkarte/nur unter OS-9)
ffc0-ffc7	FLO 3 (Diskettenkontrolller)
ffc0-ffdf	EGA (Farbgrafikkarte)
ffc8	BANKBOOT (RAM auf Adr. 0 ausblendbar)
ffc9	RELAIS
ffca-ffcb	CAS (Cassettenrekorder Interface)
ffcc-ffcf	SASI (Festplattenkontrolller)
ffd0-ffd7	AD/DA (Wandler)
ffd8-ffdf	SPRACHE (Sprachausgabe)
ffe0-ffef	AD8x16 (Analog-Digital Wandler)
fff0-fff3	SER (Serielle Schnittstelle)
fff8-fff9	DA2x8 (Digital-Analog Wandler)
fffc-fffd	AD1x10 (Analog-Digital Wandler)
fffe-ffff	UHR (alte Baugruppe, wurde durch SmartWatch ersetzt)

IO-Adressen im 68008er

Adressen	Belegt durch
b0000-bfff	Hercules (nur unter OS-9)
c0000-cfff	EGA
ec000-efff	COL (Ab Grundprogramm 6.0)

### Spezielle Speicherbereiche:

Adressen zuständig. Die GDP64 sitzt aber nur in einer der beiden BUS-Platinen. Also kann die GDP auf der geraden Busseite nicht die Adresse ff71 belegen, dort ist die erste Adresse nach ff70 die ff72.

Alle Programme, die Ein-/Ausgabeadressen direkt ansprechen (was möglichst vermieden werden sollte), nehmen daher die 68008er Adressen mit zwei oder vier mal, wenn sie feststellen, daß sie auf einer CPU68000 oder gar einer CPU68020 laufen.

Das wiederum ist gar nicht so einfach. Im neuen Grundprogramm gibt es die Funktion @SYSTEM, die unter anderem auch den CPU-Faktor (1, 2 oder 4) liefert. Soll das Programm auch mit älteren Grundprogrammversionen laufen, muß man im

Grundprogramm-ROM nachschlagen. Dort steht bei Adresse GRUNDSTART + 414 (hex) ebenfalls der CPU-Faktor. Die Grundprogrammadresse kann man unter CP/M-68K vom BIOS erfragen (Funktion 23).

Arbeitet man unter JADOS, so muß man das Grundprogramm suchen. Auf der Adresse GRUNDSTART + 400 (hex) enthält es die Kennung 5aa58001. Andererseits bietet JADOS mit INPORT und OUTPORT Funktionen an, die unabhängig von der CPU bestimmte Ein-/Ausgabeadressen ansprechen.

Die Tabelle gibt den Standard auf dem 68008er wieder. Es wird nur die rechte Hälfte der Adresse abgedruckt, da die linke immer ffff ist. Also ist die richtige Adresse

der SCSI-Baugruppe nicht ff24, sondern fffff24.

Aus dem gleichen Grund wie bei den IO-Adressen müssen diese Speicheradressen mit zwei oder vier malgenommen werden, wenn das Programm auf einem 68000 oder 68020 läuft. Allerdings beginnen Speicheradressen nicht mit ffff. Die komplette Adresse der COL lautet also 000ec000 (auf dem 68008).

Im alten Grundprogramm Version 4.3 liegt die SOUND noch bei Adresse ff40 - das führt aber zu Problemen mit der alten CENT, deshalb wurde diese Adresse auf ff50 geändert

Hier besteht ein Problem: die EGA kann derzeit nur in 68000er oder 68020er Systemen angewendet werden. Dort steckt man sie auf die ungerade Busseite - so kann sie nicht mit den Adressen der FLO kollidieren.

Früher wurde die COL bei dc000 bis dfff eingeblenet, aber das neue Grundprogramm benötigt inzwischen 64KB ROM und liegt bei Adresse d0000. Daher mußte die COL auf ec000 weichen. Diese Änderung wird derzeit noch nicht von allen Programmen berücksichtigt.

# Welt-hunger. Ernte-dank.

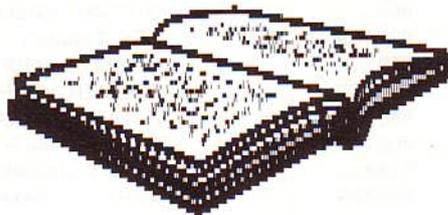
Welthungerhilfe

Die Deutsche Welthungerhilfe unterstützt Selbsthilfe-Projekte von Bauern der Dritten Welt, damit für sie Ernährung aus eigener Kraft möglich wird. Und sie hilft den Bauern, Natur und Umwelt als Lebensgrundlage zu erhalten, damit Entwicklung auch Zukunft hat.



**DEUTSCHE WELTHUNGERHILFE**  
Spendenkonto Sparkasse Bonn: 111

Adenauerallee 134 · 5300 Bonn 1 · Tel.: 02 28/22 88 0



Für Sie gelesen:

## Turbopascal 4.0/5.0

Franzis Kompakt - Auperle;  
Turbopascal 4.0/5.0 kompakt  
Franzis Verlag

Das vorliegende, tatsächlich kompakte Buch hat sich schon nach kurzer Benutzung als unentbehrlicher Helfer für die erfahrenen Anwender des Turbopascal 4.0 und 5.0 erwiesen.

Es enthält alle für den Programmieralltag benötigten Informationen, die schnell und leicht zu finden sind. Die klare Gliederung enthält alle Befehle, die mit einer umfangreichen Erklärung, einem ansprechenden Beispiel sowie Referenzen ausgefüllt sind. Damit ist ein rascher Zugriff auf die vielleicht weniger häufig eingesetzten Befehle möglich.

Das Buch wird abgerundet durch die Com-

pilerdirektiven, durch Fehlermeldungen während der Übersetzung und während der Laufzeit sowie durch die Beschreibung des integrierten Entwicklungssystems.

Zusammengefaßt: eine Anschaffung, die man nur empfehlen kann.

Was uns natürlich besonders gefreut hat: das Buch wurde in Kempten gesetzt und gedruckt.

# NEW TECHNOLOGIES

## oder wie lehre ich einem AT das Fürchten!

Ein Einblick in die Welt der "Digitalen Signalprozessoren"

Der vielzitierte technische Fortschritt scheint kein Ende zu nehmen. Kaum hat sich der medienüberflutete Computer-Interessent ein mattes Bild vom 'Stand der Technik' gemacht, schon schwebt ihm eine weitere Innovation ins Haus, verwirklicht als neuer Prozessor 80x86 oder 680xx, der - natürlich aufwärtskompatibel zu den bisherigen CPUs - angeblich "Unglaubliches" zu leisten vermag. Dabei übersieht er jedoch leicht, daß der Chipmarkt zwischen Nord- und Südpol einiges mehr zu bieten hat, als die vielgelobten 'Rechenmaschinen der Nation'.

Wie verhält es sich beispielsweise mit dem Innenleben eines modernen CD-Players, eines Radarnavigationssystems, eines digitalen Musiksynthesizers oder auch eines Multieffektgerätes? Hier spielen die sog. digitalen Signalprozessoren (DSPs) eine entscheidende Rolle. Die vornehmliche Aufgabe dieses eigenständigen Chip-typus ist es, aufgrund seiner hohen arithmetischen Rechenleistungen eine große 'Signalflut' durch algorithmische Bearbeitung umzusetzen - und das in Echtzeit! Bevor wir jedoch im ersten Teil näher auf diese Signalprozessoren eingehen, erscheint es notwendig, voran die Grundlagen der digitalen Signalverarbeitung etwas genauer zu beleuchten.

Schaltungen (z.B. bei RC-Gliedern) und die Bauteiltoleranz zurückzuführen sind. Der Alterungsprozeß wirkt sich darüberhinaus negativ auf die Leistungs- und Qualitätsmerkmale des Gesamtsystems aus, wie dies z.B. jeder Hifi-Freund am abnehmenden Rauschabstand seines (alten) Verstärkers erkennen kann.

Bauteile digitaler Systeme unterliegen zwar auch einem gewissen Alterungsprozeß, allerdings bleibt letzterer i.d.R. unbemerkbar, da die Signale im Gegensatz zu analogen Systemen nicht als kontinuierliche Spannungswerte, sondern als digitale Zahlenwerte (in der sog. zeitdiskreten Signalform) vorliegen. Und damit sind wir auch gleich beim Thema.

handle sich um ein 'Audio Processing System'. An die Stelle der 'Manipulationen' mit Wechselspannungen in analogen Systemen treten hier Algorithmen, die lediglich Zahlenwerte verändern!

Das unbearbeitete analoge Signal (1) wird zunächst durch einen Tiefpaß-Filter (2) begrenzt und passiert daraufhin einen Analog-Digital-Wandler (3). Dieser 'friert' das kontinuierliche Signal periodisch in sehr kleinen Abständen ein (Sample & Hold) und tastet diese Spannungswerte mit einer vorgegebenen Abtastrate (engl. 'sampling rate') ab. Am Ausgang des Wandlers erscheinen dann lediglich die umgewandelten Spannungswerte in Form der sog. zeitdiskreten Abtastwerte (4). Sie geben damit

### Prinzip der digitalen Signalverarbeitung

'Digitale Signalverarbeitung steht als Oberbegriff für unterschiedliche Operationen und beschäftigt sich mit der Verarbeitung von Signalen mit digitalen Verfahren.' (1)

Im Gegensatz dazu bringt die übliche analoge Signalverarbeitung mit Widerständen, Kondensatoren und Halbleitern (wie Transistoren, Operationsverstärkern) einige Probleme mit sich.

Eine gute Verarbeitungsqualität ist häufig mit sehr hohem Aufwand (Anzahl und Güte der Bauelemente) verbunden und treibt so die Systemkosten in die Höhe. Als Beispiel sei hier die Komplexität eines aktiven analogen Mehrpolfilters angeführt. Des Weiteren sind den analogen Systemen immer unliebsame Nebenwirkungen wie lineare und nichtlineare Verzerrungen, Phasenverschiebungen und Eigenrauschen anhaftet, die auf gewisse Eigenarten der

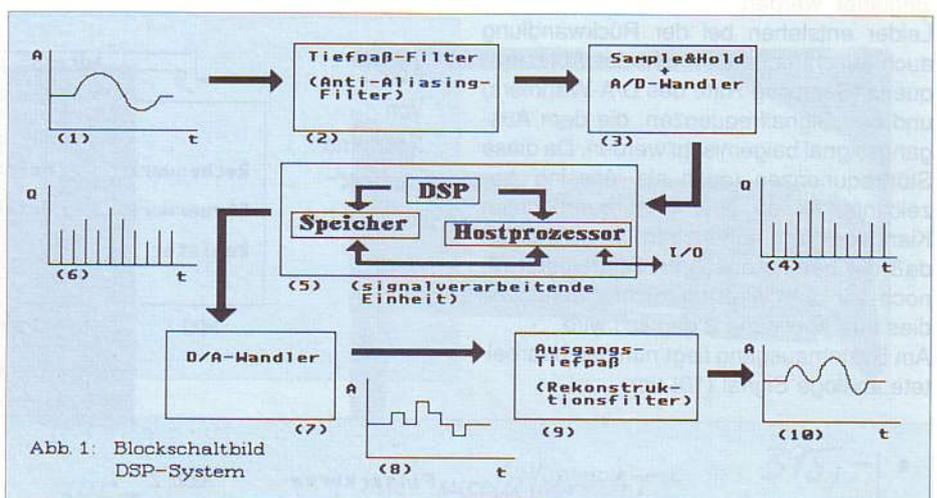


Abbildung 1 zeigt deutlich das vereinfachte Blockschaltbild eines digitalen signalverarbeitenden Systems. Es hat darüberhinaus lediglich Beispielcharakter und kann in der Praxis mitunter abweichen. Der Einfachheit halber nehmen wir an, es

die Amplitudenauslenkung des Eingangssignals (1) zu bestimmten Zeitpunkten an. Das Zeitraster (t-Achse) ist dabei von der Sampling-Rate, das Amplitudenraster (Quantisierung Q) dagegen von der Wandlerauflösung abhängig.

Der gesamten Wandlung liegt das Abtasttheorem für bandbegrenzte Signale zugrunde, welches besagt, daß die Abtastfrequenz mindestens doppelt so hoch sein muß, wie die max. Frequenz des Eingangsspektrums (1) - andernfalls entstehen digitale Verzerrungen (sog. Aliasing). Da die Eingangssignale meist keine Frequenzbegrenzung nach oben aufweisen, ist aus obigem Grund ein entsprechendes Tiefpaßfilter (2) (auch Anti-Aliasing-Filter genannt) vorzuschalten.

In der signalverarbeitenden Stufe (5) werden die diskreten Signalwerte durch die gewünschten Algorithmen im DSP weiterverarbeitet. Soll dies in Echtzeit erfolgen, so muß der DSP jeden Abtastwert in einer Abtastperiode vollständig bearbeiten. Bei einer 'Sampling-Rate' von 50kHz bleiben ihm dazu reichliche 20 µs (im Monobetrieb) und knappe 10 µs bei zwei gemultiplexten Signalen am A/D-Eingang (im Stereobetrieb). Entsprechend der Wanderauflösung (12, 16 oder 20 Bit) ist auf die korrekte Verarbeitungsbandbreite zu achten, da sonst wichtige Dynamikinformationen des Signals unterschlagen werden. Der Hostprozessor hat im wesentlichen Kontroll- und Steuerungsaufgaben und kann beispielsweise Bildschirmgrafiken ausgeben. Die bearbeiteten diskreten Werte (6) werden im Digital-Analog-Wandler (7) wieder rückgewandelt und erscheinen als Spannungstrepfen am Ausgang (8). Diese auf die Quantisierung und die Abtastrate zurückzuführende Ungenauigkeiten - die spektral gesehen zusätzliche lineare und nichtlineare Oberwellen darstellen - müssen durch ein steilflankiges Tiefpaßfilter (9) 'geglättet' werden.

Leider entstehen bei der Rückwandlung auch Mischfrequenzen aus der Abtastfrequenz (Sampling-Rate des D/A-Wandlers) und den Signalfrequenzen, die dem Ausgangssignal beigemischt werden. Da diese Störfrequenzen (auch als 'Aliasing' bezeichnet) knapp über dem bearbeiteten Klangspektrum auftauchen, ist es wichtig, daß die bereits erwähnte Filtereinheit (9) noch vor dem Aliasing 'dicht macht', wie dies aus Abbildung 2 deutlich wird. Am Systemausgang liegt nun das bearbeitete analoge Signal (10) vor.

Je nach Systemkonfiguration können einige Verarbeitungsstufen fehlen. Beim CD-Player ist beispielsweise der Eingangswandler überflüssig, da der Rechner die digitalen PCM-Signale direkt von der CD-Abtasteinheit erhält.

Nachdem diese Grundlagen bekannt sind, taucht nun die Frage auf, warum diese Aufgaben nicht von universellen Mikroprozessoren gelöst werden können.

**Weshalb sind herkömmliche Mikroprozessoren für die digitale Signalverarbeitung meist ungeeignet?**

Herkömmlichen CISC-Prozessoren (CISC steht für Complex Instruction Set Computing), die über einen reichen Befehlssatz verfügen, sind trotz Weiterentwicklung Grenzen in der Rechengeschwindigkeit gesetzt. Jeder Maschinenbefehl benötigt ein eigenständiges chipinternes Mikrocodeprogramm, das nach Empfang einer Anweisung in der sog. Dekodierzeit zunächst gefunden und dann nacheinander abgearbeitet werden muß. Da hierzu oft mehrere Taktzyklen notwendig sind, verlängern sich die einzelnen Befehlsausführungszeiten und verringern somit die Leistung des Prozessors. Durch das 'Debakel' der Aufwärtskompatibilität, wie es bei nahezu allen Prozessorfamilien üblich ist, wird das Problem durch die erweiterten Befehlssätze bei jedem neuen Prototyp verschlimmert, da sich die Dekodierzeiten für die unterschiedlichen Mikrocodes verlängern.

Abbildung 3 zeigt den vereinfachten Aufbau eines Mikrocomputers auf CISC-Basis - wie er jedem von Ihnen bekannt sein wird.

Diese sog. 'von Neumann' Architektur erweist sich jedoch für signalverarbeitende Systeme i.d.R. als zu langsam. Dies ist im wesentlichen auf folgende Tatsache zurückzuführen:

Das System besitzt lediglich einen Adress- und Datenbus (auch chipintern). Sämtliche Instruktionen müssen daher nacheinander (d.h. sequentiell) abgearbeitet werden. Sehr viele Befehle erfordern mehrere externe Speicherzugriffe, da sowohl die einzelne Anweisung wie auch das zugehörige Datum sequentiell zur CPU transportiert werden muß. Bei der Verwendung externer dynamischer Speicher sind zudem entsprechende Wartezyklen einzulegen.

Dies alles führt dazu, daß selbst ein leistungsfähiger AT mit 16 MHz Takt auf 80386-Basis noch weit über 1µs für eine simple 16 x 16 Bit Multiplikation benötigen (eingeschlossen Datentransport) - zu langsam für die Echzeitausführung eines rechenintensiven Algorithmus (wie dem Butterfly-Algorithmus der Fast Fourier Transformation) bei einer typ. Abtastperiode von 20 µs.

Die RISC-Technologie (RISC steht für Reduced Instruction Set Computing) ermöglichte es durch reduzierte Befehlssätze höhere Taktraten einzusetzen. Anstelle der chipinternen Mikrocodes traten für jeden Befehl schnelle Hardwarelogiken. Dies führte zur Verwirklichung von 'Einzyklus-Befehlen', wie sie für jedes RISC-System charakteristisch sind.

In der Tat weisen digitale Signalprozessoren einige Gemeinsamkeiten mit handels-

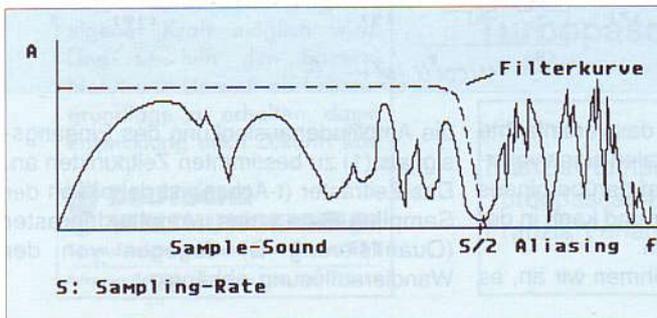
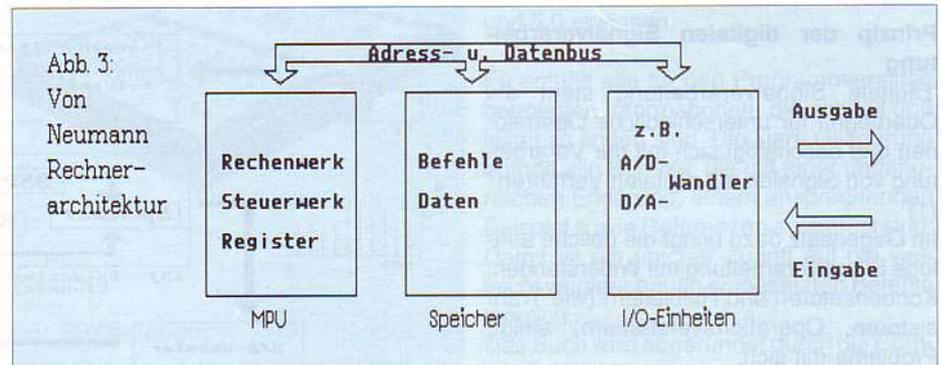


Abb. 2: Spektrum eines Samples mit Aliasing und Filterkurve

üblichen RISC-Prozessoren auf, so daß wir jetzt übergreifend auf die eigentliche signalverarbeitende Einheit eingehen werden.

**Merkmale und Architektur der DSPs**  
Zunächst wollen wir uns mit den charakteristischen Merkmalen der Signalprozessoren vertraut machen. Die folgenden Features treffen auf nahezu alle gebräuchlichen DSPs zu:

# Sonderangebote



## Textsystem

Komplett mit TA-Computer P3,  
TA-Typenrad-Drucker,  
Textprogramm "TEX-ASS"

1.950,-



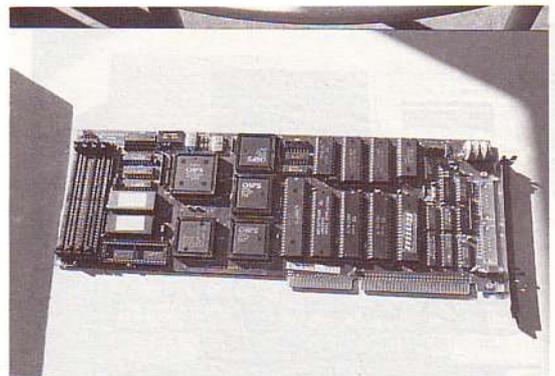
Lap-Top TL 3240, Ausstell.-Gerät  
3 1/2" 1,44MB, Platte 40 MB, Extras 5.900,-



EPSON RX 80 FIT 345,-



SANYO Farbmonitor, 40 345,-



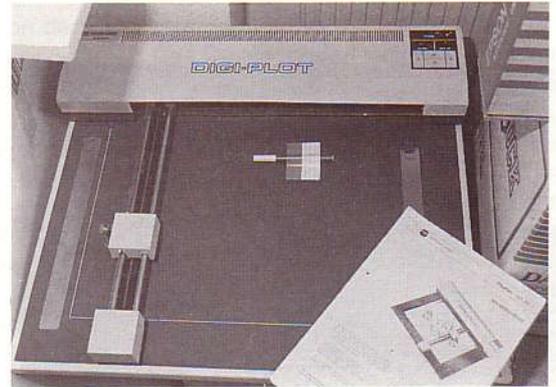
mc-modular-AT 395,-  
CPU-KARTE, SIP-Module, ohne RAM

# Sonderangebote



70289

Schreibmaschine mit PC-Anschluß  
TRIUMPH-ADLER SE1 Neu 3690,-



Wartungabe Plotter A3  
Vorführgerät, ca 4 Jahre 580,-



VICTOR-SIRIUS "Vicki": Portable PC  
(nicht MS-DOS), ca 5 Jahre 500,-



SIRIUS-Computer mit Festplatte  
und Finanzbuchhaltung 1.900,-  
(Profi-Fibu!)



70152

EPSON HX20 Handheld-Computer  
mit BASIC und Drucker,  
Vorführgerät 950,-



Triumph-ADLER P3, CPM-Rechner,  
mit 2 Laufwerken, ca 4-5 Jahre 950,-

# Sonderangebote

## Original IBM - Drucker :



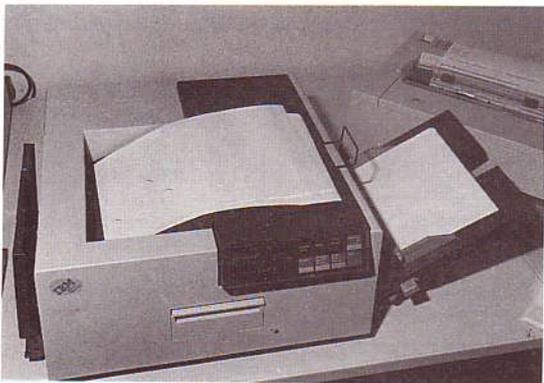
IBM Modell 4201-003 Neu  
9 Nadeln, DIN A4 1.098,-

4202-003  
9 Nadeln, breit 1.428,-



IBM Modell 4207-002  
24 Nadeln, Parkposition,  
Neuestes Modell,  
Einzelblatt, DIN A4 1.598,-

IBM Modell 4208-002  
wie oben, breit 1.998,-



IBM Laserdrucker 4216-010  
Lasersystem, 300 x 300 Punkte,  
2 MB Seitenpeicher 4.895,-

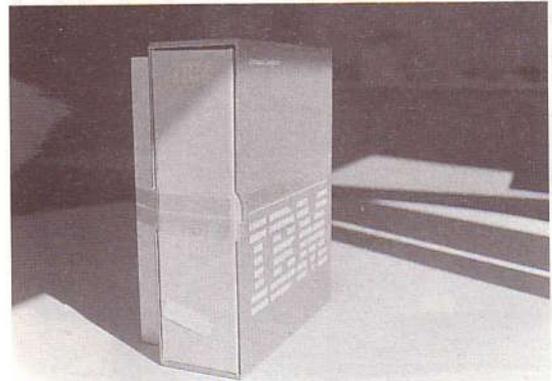
IBM PS/2  
Modell 30-002

IBM Personal-Computer  
Modell 30-002  
640k RAM, 2 x 3 1/2"-720k  
NEU, Originalverpackt! 2.298,-

# Sonderangebote



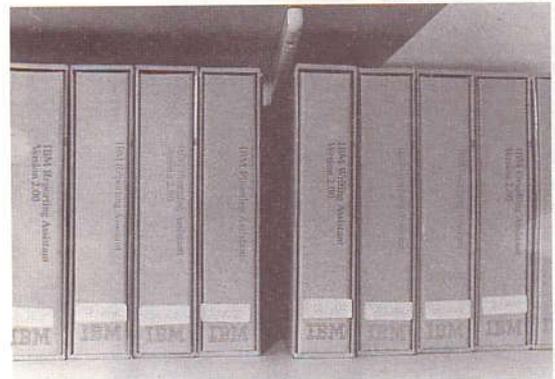
Original IBM-Manuals  
XT, AT, AT03 20,-



Original IBM PC-DOS (=MS-DOS)  
V. 2.00 und 2.01 39,-  
Endlich ein lizenziertes DOS  
komplett mit Beschreibung



IBM Finanzbuchhaltung  
Schulversion 395,-



Original IBM-Programme:  
- Reporting Assistent deutsch #70141  
- Planning Assistent alt. #70142 je 199.-  
- Writing Assistent alt. #70143  
- Graphing Assistent alt. #70144

Weitere Programme siehe Liste!

- Harvard-Architektur
- Hardware-Multiplizierer mit Überlaufregister
- Pipelining-Technik
- getrennte Daten- und Programmspeicher (RAMs) auf dem Chip
- interner ROM-Speicher, wahlweise mit Wertetabellen der Sinus- und Exponentialfunktion
- vergleichsweise geringer Befehlssatz (viele 'Ein-Zyklus-Befehle')
- größtenteils spezifische Hochgeschwindigkeitsschnittstellen (SSI, SCI, parallel) zur ext. Datenübertragung
- spezielle Hardwareeinrichtungen, die eine Implementation gebräuchlicher Algorithmen erleichtern
- geringe Zykluszeiten

Wie sich leicht erkennen läßt, handelt es sich bei Signalprozessoren letztendlich um spezifische für die Signalverarbeitung zugeschnittene RISC-Chips. Die interne Ausstattung mit Speicherbausteinen (Kapazität mehrere kByte), MPU (MicroProcessor Unit) und verschiedenen Schnittstellen macht diesen Halbleitertypus praktisch zu einem Single-Chip-Computer.

Die DSPs wurden derart konzipiert, daß möglichst viele Daten in geringsten Zeitperioden aufgenommen, verarbeitet und wieder abgegeben werden können. Externe Speicherzugriffe werden nur selten durchgeführt, da sie i.d.R. zu viel Zeit kosten.

Bei den durchzuführenden repetitiven Algorithmen - welche in vielen Fällen reine Multiplikations- und Additionsprozesse darstellen - soll weitgehend auf sequentielle Verarbeitungsmethoden zugunsten paralleler Operationen verzichtet werden.

Hierauf hat die interne parallele Busstruktur (=Harvard-Architektur) einen wesentlichen Einfluß. Durch Trennung der Bus- und

Speichersysteme für Programmbefehle und Daten (vgl. Abbildung 4) und die Anwendung des Pipelining-Prinzips (zeitliche 'Überlappung' aufeinanderfolgender Befehle) wird die Ausführung von ganzen Befehlskomplexen in häufig nur einem Maschinenzyklus realisiert!

Ein typischer derartiger Befehl könnte lauten:

- Multipliziere die Daten im X- und Y-Register (je 24 Bit) mit Berücksichtigung des Überlaufs (bis zu 8 Bit)
- Addiere das (56 Bit) Ergebnis zum Akkumulator B
- Lade die X, Y-Register mit neuen Werten

Der komplexe Multiplikations/Additionsvorgang ist letztendlich ein Verdienst des integrierten Hardware-Multiplizierers mit nachfolgender Additionsarithmetik. Die Registergröße variiert je nach DSP-Typ zwischen 16 Bit und 32 Bit. Man differenziert weiterhin zwischen Festkomma- und Gleitkommarechenwerken. Letztere bieten bei der Signalverarbeitung eine große Genauigkeit, treiben jedoch den Hardwareaufwand beträchtlich in die Höhe. Bei einer Multiplikation wird der 'Überlauf' meist durch die Implementation eines größeren Ergebnisregisters mitberücksichtigt. Dessen Format weist i.d.R. ein Vielfaches der Wandlerbandbreite auf; dies ist jedoch für repetitive komplexe Operationen unabdingbar, da sich hier die einzelnen Rundungsfehler aufaddieren und das Ergebnis beträchtlich (d.h. hör- oder erkennbar) verfälschen können.

#### Ein weiterer Grund, der einen Universal-Mikroprozessor als ungeeignet entlarvt.

Wie auch bei allen RISC-Prozessoren gilt der Grundsatz:

Je größer der chipinterne Speicher (in Stack- oder Registerform), desto seltener muß auf externe (langsamere) Bausteine zugegriffen werden, da alle notwendigen Programmteile (Anweisungen und die zu bearbeitenden Daten) dem Prozessor intern zur Verfügung stehen.

Dies erhöht wiederum die Bearbeitungsgeschwindigkeit. Die Kapazitäten handelsüblicher DSPs sind in nahezu allen Fällen auf die Implementation der gebräuchlichsten Algorithmen hin ausgelegt, so daß der Signalprozessor im Echtzeitbetrieb größtenteils ohne externe Speicherzugriffe auskommt (einmal abgesehen von manchen Systemen, die eine Zwischenspeicherung der Abtastwerte durch 'Direct Memory Access' des ext. RAMs vorsehen).

Von den meisten Herstellern werden processorinterne ROM- oder EPROM-Speicher angeboten, die entweder kundenspezifische Programme oder vollständige Sinus- und Exponentialwertetabellen enthalten; letztere erleichtern viele algorithmische Arbeitsschritte.

Die oftmals integrierten digitalen Schnittstellen wie SSI (Serial Synchronic Interface), SCI (Serial Communication Interface) und ungenormte parallele Datenausgänge erweitern die Möglichkeiten des Datendurchsatzes erheblich. Eine direkte Verbindung mit externen Bausteinen wie Hostprozessor, D/A-Wandler, weiteren DSPs sowie ein direkter Speicherzugriff über DMA (Direkt Memory Access) wird dadurch möglich.

Besonders interessant sind die in einigen DSPs implementierten Hardwareeinrichtungen (spezielle Adressrechenwerke), die es dem Programmierer erleichtern seine Signalalgorithmen umzusetzen. Dazu zählen meist gesonderte Adressierungsarten wie die 'Modulo-Adressarithmetik' (Anwendung bei der Verwaltung von Ringspeichern) oder die Einrichtung 'Bit-Reverse-Carry'; sie ermöglicht eine automatische Reihenfolgekorrektur der Ergebnisse beim 'Butterfly-Algorithmus' (genauerer dazu im Artikel "DSP - zum reinschnuppern").

Alle diese Features, wie die - mit der RISC-Architektur verbundenen - hohen Taktraten ergänzen den DSP zu einem ausgereiften Spezialprozessor für hohe bis höchste Ansprüche in der Signalverarbeitung.

Dieses alles darf aber nicht darüber hinwegtäuschen, daß ein derartiger Prozessor nur für eingeschränkte Anwendungen prädestiniert ist. Als Hauptprozessor in einem IBM-Kompatiblen wäre er genauso falsch am Platze, wie als Speichercontroller oder Single-Chip-Computer in einer Waschmaschine.

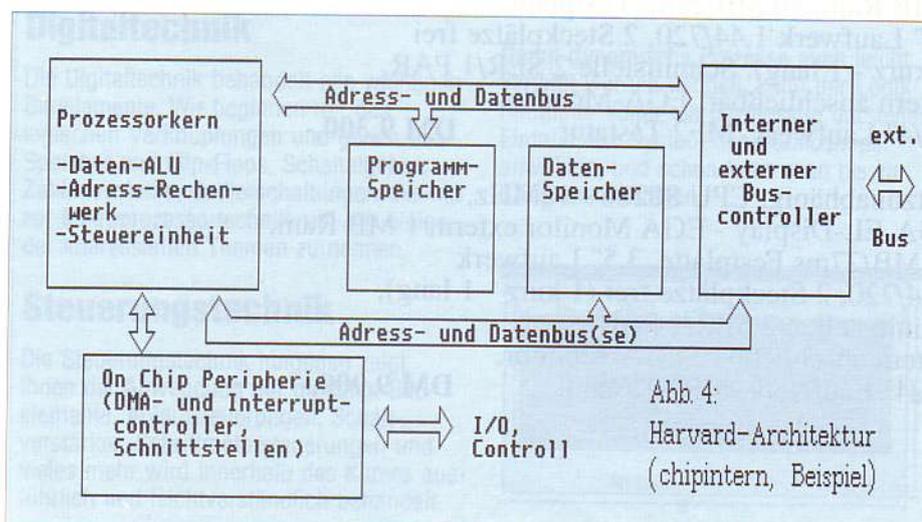


Abb. 4:  
Harvard-Architektur  
(chipintern, Beispiel)

## Achtung - Jetzt neue Preise

Unsere Systeme:

**modular AT 286-10**  
 10MHz-512K (bis 1MB)  
 20MB Platte-5 1/4" LW  
 Gehäuse-200W Netzteil  
 Cherry-Tastatur  
**Best.Nr.11445 DM 3.298,-**

**modular AT 286-12**  
 12MHz-512K (bis 1MB)  
 20MB Platte\_5 1/4" LW  
 Gehäuse-200W Netzteil  
 Cherry-Tastatur  
**Best.Nr.11446 DM 3.598,-**

**modular AT 286-12**  
 12MHz-512K (bis 1MB)  
 40MB Platte-5 1/4" LW  
 Gehäuse-200W Netzteil  
 Cherry-Tastatur  
**Best.Nr.11447 DM 3.998,-**

**modular At 286-16 NEAT**  
 16MHz NEAT CPU - 512K  
 (bis 4MB) 40MB Platte  
 5 1/4" LW-Gehäuse  
 200W Netzteil-Cherry-Tastatur  
**Best.Nr.11448 DM 4.398,-**

**modular AT 386-16**  
 16MHz-1MB (bis 2MB)  
 80MB Platte-5 1/4" LW  
 Gehäuse-200W Netzteil  
 Cherry-Tastatur  
**Best.Nr.11449 DM 7.298,-**

**Für das Auge, hier die  
 Bildschirmangebote:**  
 (Monitor und Controllerkarte)  
**Hercules-Paket DM 498,-**  
**EGA-Paket DM 1.398,-**  
**Profi VGA-Paket DM 1.998,-**  
 (mit NEC-MultiSynch)

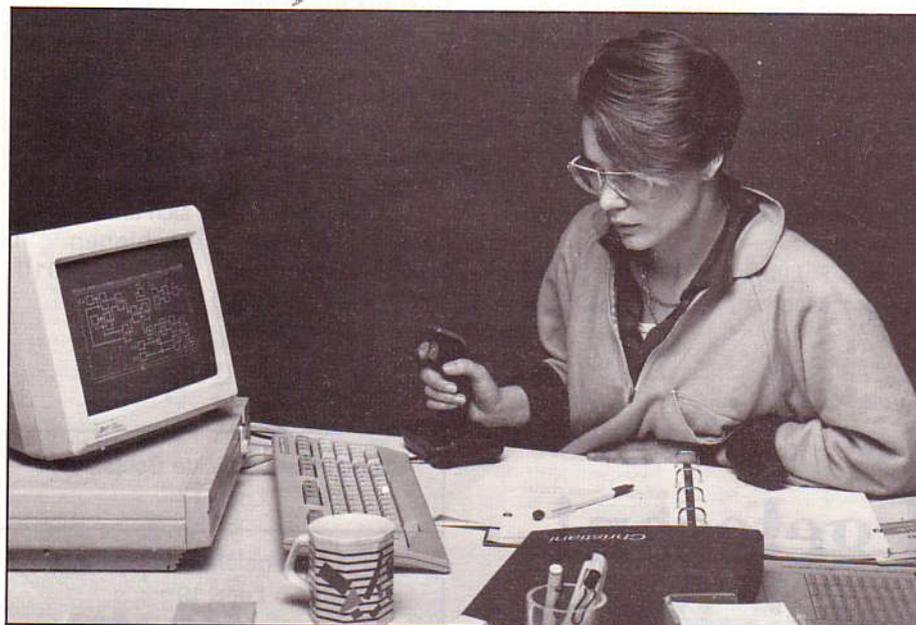
**NEU:** **CPU 80386-25MHz**  
 Best.Nr.11476 ohne Speicher **DM 2.648,-**  
 Best.Nr.11477 1MB Speicher **DM 590,-**

### Die neuen Laptops jetzt lieferbar

**TL3240N** Netzabhängig, CPU 80286N - 16MHz, NEAT  
 EGA-Plasma-Display, 16 Graustufen,  
 1 MB Ram, 40 MB/27ms Festplatte,  
 3,5" Laufwerk 1,44/720, 2 Steckplätze frei  
 (1 kurz - 1 lang), Schnittstelle 2 SER/1 PAR,  
 Extern anschließbar: EGA-Monitor,  
 5 1/4" Laufwerk, MF2 Tastatur **DM 9.300,-**

**TL3240EL40** Netzunabhängig, CPU 80286 - 12MHz,  
 CGA-EL-Display - EGA Monitor extern, 1 MB Ram,  
 40 MB/27ms Festplatte, 3,5" Laufwerk  
 1,44/720, 2 Steckplätze frei (1 kurz - 1 lang),  
 Schnittstellen 2 SER/1 PAR,  
 Extern anschließbar: EGA-Monitor,  
 5 1/4" Laufwerk, MF2 Tastatur **DM 9.900,-**

**GRAF**  
 computer



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

200 Seiten reichbebildertes Lehrmaterial mit Sammelordner, Register und Logik-Simulator für den Commodore C64 / C128 oder IBM- und compatible 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. Nahezu alle Schaltungen, die im Lehrmaterial beschrieben sind, können mit dem Logik-Simulator ausgetestet werden. Das Aufbauen von Schaltungen hat damit ein Ende. Sie werden begeistert sein.

#### Digitaltechnik

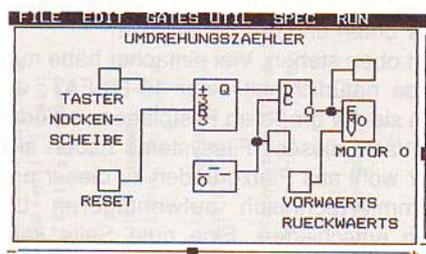
Die Digitaltechnik behandelt alle wichtigen Bauelemente. Wir beginnen mit den logischen Verknüpfungen und gehen über Speicher und Flip-Flops, Schaltalgebra, Zahlensysteme, Codierschaltungen, bis hin zur Mikroprozessortechnik, um nur einige der interessanten Themen zu nennen.

#### Steuerungstechnik

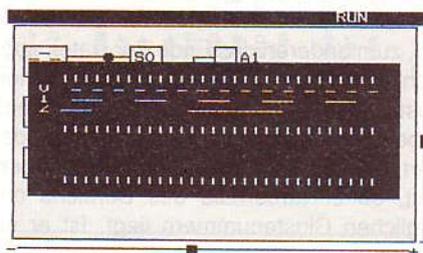
Die Steuerungstechnik hingegen zeigt Ihnen die Anwendung der digitalen Bauelemente. Ablaufsteuerungen, Schaltverstärker, Schrittmotorsteuerungen und vieles mehr wird innerhalb des Kurses ausführlich und leichtverständlich behandelt.

#### Der Logik-Simulator

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



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



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

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

#### Was Sie brauchen

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

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

# Christiani Fortbildung

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

DSPs können nicht den Luxus eines umfangreichen Befehlssatzes bieten. Dies hat relativ eingeschränkte Programmiermöglichkeiten zur Folge. Außerdem sind Abstriche in der max. Speicherverwaltung zu machen (viele DSPs können nur einige kByte extern adressieren) sowie die höheren Systemkosten mitzukalkulieren. Die Implementation von Hochsprachen auf Basis eines DSP Single-Chip-Computers steht daher außer Frage.

Im zweiten Teil gehen wir näher auf die einzelnen DSP-Gruppen und die Bedeutung des Echtzeitbetriebs ein. Schließlich werfen wir noch einen Blick auf zukünftige Entwicklungsprojekte und Anwendungsmöglichkeiten.

#### Literaturangaben:

(1) aus F.J.Leyerer: "Warum digitale Signalprozessoren", Artikel erschienen in

Design&Elektronik 12/89, S.104, Markt&Technik Verlag

weiterhin wurde verwendet:

- M.Fabig: "Analoge Welt A/D", Artikel erschienen in c't 3/89, S.198-210, Heise-Verlag

- V.Dirksen: "Alles dreht sich um den Speicher", Artikel erschienen in Design&Elektronik 7/89, S.91ff., Markt&Technik Verlag

Günter Renner

# Jetzt wird gelesen!

**Das Beschreiben von Datenträgern ist bisweilen eine gefährliche Sache. Vor allem dann, wenn man sich in fremdes Territorium - sprich Systeme - begibt. Es soll deshalb erst einmal mit einer harmloseren Operation begonnen werden, durch die allenfalls Speicherinhalte zerstört werden können: dem Lesen von Dateien von der Diskette.**

Diese Aufgabe erledigt die Menüfunktion 'load'. Zunächst wird der Benutzer zur Eingabe des Dateinamens aufgefordert. Dann geht es mit dem bekannten Testen der Disk weiter. Stimmen die Parameter nicht überein, wird mit der eigentlichen Arbeiten gar nicht erst begonnen.

Nun wird das Inhaltsverzeichnis durchsucht mittels des Unterprogramms 'lookfor'. Alle 112 Namenseinträge des Hauptinhaltsverzeichnisses werden verglichen. Wird keine Übereinstimmung mit dem Inhalt des Namenspuffers gefunden, kehrt dieses Unterprogramm mit gesetztem Minusflag zurück, und das Programm 'load' meldet das Nichtvorhandensein des gesuchten Eintrags. Ist der Eintrag jedoch gefunden, zeigt (a4) auf das entsprechende Entry. Das sehr einfache Unterprogramm 'firstclu' liest nun den Startcluster ins Register d2.w, sodaß damit gleich der erste Cluster der Datei bearbeitet werden kann. Hier begegnet einem wieder die verschiedene Anordnung der Bytes eines Wortes, die prozessorabhängig ist und durch Rotation korrigiert werden muß.

Da nun die wenigsten Dateien nur aus einem Cluster bestehen, sollte man wissen, wie es weitergeht. Auskunft darüber gibt das Unterprogramm 'nextclu'. Es verwendet den Inhalt von d2.w als Zeiger auf die aktuelle FAT-Position und liest deren Inhalt nach d3.w. Von den geladenen 16 Bits interessieren hier natürlich nur 12 davon; die vier übrigen gehören zu einem anderen Eintrag.

Die Hardware gestattet nur ein Arbeiten mit 16 Bit breiten Worten. Man muß deshalb den Wert eines Eintrags mit einer UND-Verknüpfung herausmaskieren, ehe man ihn weiterverwenden kann. Vor dieser Maskierung ist bei allen ungeraden FAT-Positionen, noch eine Rechtsverschiebung um vier bits erforderlich, damit man die richtigen 12 bits auch tatsächlich erwischt. Diese müssen stets in d3.w unten und der unbenutzte Rest oben stehen. Viel einfacher hätte man es da natürlich mit einer 16-Bit-FAT, wie man sie auf größeren Festplatten vorfindet; die Väter dieses Filesystems haben sich aber wohl aus Platzgründen zu dieser programmiertechnisch aufwendigeren Lösung entschieden. Eine gute Seite kann man dieser Sache aber immer abgewinnen: wer sich ein Buch über MS-DOS kaufen will, um in den 'Niederungen' zu programmieren, sehe erst einmal darin nach, wie verständlich oder wie verwirrend die Logik und die Handhabung der FATs darin beschrieben ist...

Zur Erinnerung: der gewonnene Wert kann zum einen den nächsten Cluster bezeichnen, zum anderen das Ende der Datei, aber auch einen reservierten oder fehlerhaften Cluster kenntlich machen. Der Einfachheit halber wird im vorgestellten Monitor der Wert des FAT-Eintrags nur darauf überprüft, ob er außerhalb des Bereichs der möglichen Clusternummern liegt. Ist er innerhalb dieser Grenzen, gibt der gefundene Wert den nächsten Cluster an, so daß man

Teil 1: Der verrückte Bootsektor Loop 17
Teil 2: Eine gefährliche Operation Loop 19
Teil 3: Log. physikalischer Verwirrspiel Loop 20
Teil 4: Sage mir, was Du hast Loop 21
<b>Teil 5: Jetzt wird gelesen Loop 22</b>
Teil 6: Ärger mit Ä Loop 23

ihn einfach nach d2.w laden und damit erneut die Routine 'cluster' aufrufen kann. Liegt er außerhalb derselben, ist das Dateiende erreicht, oder aber die FAT ist fehlerhaft.

Das Programm 'load' enthält nun eine Schleife, die fortlaufend Cluster nach Cluster einliest. Das geht solange, als 'nextclu' einen Wert liefert, der einer möglichen Clusternummer entspricht. Andernfalls wird die Schleife abgebrochen und zuletzt geprüft, ob ein Floppyfehler aufgetreten ist.

In einer weiteren Folge werden wir es wieder mit solchen Clusterketten zu tun haben. Dann, wenn Programme zum Löschen und Speichern von Dateien vorgestellt werden. Das nächste Mal ist es noch nicht so weit. Denn es gibt noch ein kleines Problem, das bei Textfiles auftreten kann: Umlaute und andere Sonderzeichen!

Günter Renner  
Schloßbühlstr. 11  
7206 Emmingen-Liptingen

## Patchwork-Listing Teil 5

```

****Menuefunktionen
filebuf equ $....
maxclu equ 345

load:
bsr getname
lea fhflag(pc),a4
clr.b (a4)
bsr testdis
bmi load2

bsr lookfor
bmi nichtda
bsr firstclu
moveq #1,d1
lea filebuf(pc),a0

load1:
bsr cluster
bsr nextclu
move d3,d2
and #ffff,d2
cmp #1,d2
ble load2
cmp #maxclu,d2
ble load1

load2:
bsr testflg
bra menue

****Unterprogramme

lookfor:
bsr getdir
lea dirbuf(pc),a4
movem.l d6-d7,-(a7)
move #112-1,d7

lookf1:
cmp.b #8,11(a4)
beq lookf3
btst.b #4,11(a4)
bne lookf3
movea.l a4,a3
lea namebuf(pc),a2
moveq #11-1,d6

* nach Bedarf...
* hoechste Nr.

*Namen erfragen
*fehlerfrei
*annehmen
*Disk testen

*Eintrag suchen

*ersten Cluster
*Befehl lesen
*Ziel im RAM

*und lesen
*steht in d3.w

*ist neuer Cl.
*nicht < 2

*Ende der Datei?
*wenn noch nicht

*Lesefehler?

*Datei suchen
*DIR einlesen
*Suchzeiger

*max. 112 entries

*Disk-Kennsatz

*Unterverzeichnis
*wird uebergangen
*a4 schonen
*Eingabereferenz
*11 Zeichen

lookf2:
move.b (a2)+,d0
move.b (a3)+,d1
cmp.b d0,d1
bne lookf3
dbra d6,lookf2
movem.l (a7)+,d6-d7

clr d0
rts

lookf3:
adda.l #$20,a4
dbra d7,lookf1
move d7,d0
movem.l (a7)+,d6-d7
rts

firstclu:
move $1a(a4),d2
rol #8,d2
and #ffff,d2
rts

nextclu:
lea fatbuf1(pc),a4
move d2,d4
mulu #3,d4
lsl #1,d4
and #3ff,d4
move.b 1(a4,d4.w),d3
lsl #8,d3
move.b 0(a4,d4.w),d3
btst #0,d2
beq nextcl1
ror #4,d3

nextcl1:
rts

nextcl1:
rts

sldung

estf1:

testdis:
lea bootsek(pc),a0

```

# DER JADOS PATCHBEREICH

## Warum patchen ?

Es gibt allerdings einige Probleme, die auch mit der größten Flexibilität nicht automatisch gelöst werden können. Ein markantes Beispiel ist die Adresse der SOUND-Karte. Bei vielen Rechnern wird zum Druckeranschluß die IOE auf Adresse \$FFFFFF48 eingesetzt. Da sie nicht vollkommen ausdekodiert ist, gibt es Konflikte, wenn die SOUND-Karte, die standardmäßig auf Adresse \$FFFFFF40 liegt, angesprochen wird. Da dieser Konflikt nur durch Ausdekodierung zufriedenstellend gelöst werden kann, benutzt JADOS zur Zeit die Adresse \$FFFFFF50 für die SOUND-Karte.

Dies wiederum verursacht einigen Usern Ärger, da sie Programme für die SOUND einsetzen, die von der Adresse \$FFFFFF40 ausgehen. Die bisherigen Versionen von JADOS benutzten an einigen wenigen Stellen die SOUND-Adresse im Programmcode. Mit Hilfe eines Diskmonitors und eventuell eines Disassemblers konnte man diese Stellen finden und dann nach seinen Wünschen ändern. Dieser Vorgang wird auch mit "patchen" bezeichnet. Der Begriff stammt wie so vieles aus der Computerei aus dem Englischen und heißt soviel wie "Flicken". Es soll aber nicht verschwiegen werden, daß diese Art des Patchens ziemlich fehleranfällig ist.

## Der Patchbereich

Ab Version 3.0 von JADOS gibt es nun einen sogenannten Patchbereich am Anfang der Datei JADOS.SYS. In diesem Bereich stehen einige wichtige Werte, die nach dem Programmstart in Variablen kopiert werden. Im weiteren Prograema blaue werden dann nur die Variablen benutzt. Zwecks Anpassungen müssen dann nur die Werte im Patchbereich geändert werden, ohne daß man das gesamte Programm untersuchen muß. Dadurch wird das Patchen wesentlich sicherer.

Zur Zeit existieren nur die Diskparameter im Patchbereich sowie die SOUND- Adresse, der Textoffset beim Assembler und die maximale Größe der Texte, die mit dem Grundprogramm bearbeitet werden können. Während man an den Diskpara-

**Das JADOS-Betriebssystem ist so aufgebaut, daß es in der Regel auf sämtlichen Konfigurationen, in denen der NDR-Computer im 68000-Ausbau vorkommen kann, gut arbeitet. Dies liegt daran, daß JADOS sehr flexibel ist und fast alle Eventualitäten berücksichtigt. Wenn man ein neues Laufwerk an den Rechner anschließt oder den RAM ausbaut, muß man z.B. CP/M68K ändern, neu übersetzen und binden, um den Ausbau überhaupt nutzen zu können. Dies ist im JADOS nicht nötig, da es so ungemein flexibel ist.**

metern tunlichst nicht "drehen" sollte, sind die anderen Werte für das Patchen "freigegeben". Ein komfortables Werkzeug zum Patchen ist z.B. mein Programm INSPEC. Die Abb. 1 zeigt den Patchbereich. Er wird eingeleitet durch den Text "PATCHBEG" und beendet durch den Text "PATCHEND". Zur Zeit gibt es fünf Parameter-Tabellen für die Laufwerke 0: bis 4: Jeder Tabelle ist der Text "DSKn" vorangestellt, wobei "n" von "0" bis "4" variiert.

Die Adresse der SOUND-Karte wird durch den Text "SOUN" angekündigt. Es folgt der 4-Byte-Wert \$FFFFFF50. Wer die SOUND auf einer anderen Adresse betreiben will, der muß den Wert \$FFFFFF50 durch \$FFFFFF40 ersetzen.

Der Text "ASST" ist dem Textoffset beim Assemblieren vorangestellt. Hier ist der Wert 00004000 eingetragen. Dies entspricht einem Offset von \$4000 = 16KByte. Da JADOS aber den Text soweit wie möglich im Speicher verschiebt, ist der Textoffset nicht mehr so wichtig. Der momentan letzte zu patchende Wert bezieht sich auf die maximale Größe eines Textes, den der Grundprogramm

bearbeiten kann. Dieser Wert wird durch "EDIT" eingeleitet und beträgt 00010000, also 64 KByte. Ab der Version 6.0 des Grundprogramms ist diese Grenze unwichtig, da sie nicht mehr existiert. Viele User arbeiten aber noch mit der Version 4.3, die diese Grenze noch hat. Diejeni-

gen User, die mit einem geänderten Grundprogramm 4.3 arbeiten, das die Editorgrenze nicht mehr aufweist, müssen den Wert nach oben korrigieren.

## Ausblicke

Der Patchbereich wird in Zukunft weiter ausgebaut werden, um den Usern mehr Möglichkeiten zur Anpassung und Konfiguration des JADOS zu bieten. Weiter ist geplant, das Patchen noch weiter zu vereinfachen. Der User kann sich dann darauf beschränken, mit einem normalen Texteditor Anweisungen in eine Dateinamens "CONFIG.SYS" einzugeben. Diese Anweisungen wird der Urlader abarbeiten und das JADOS im RAM gezielt patchen. Denkbar wären etwa folgende Anweisungen in der CONFIG.SYS:

```
SOUN=$FFFFFF40
EDIT=$00100000
```

Auch kann durch eine solche Konfigurationsdatei z.B. die Größe der Ramdisk festgelegt werden. Die Abfrage nach der Ramdisk im Urlader wäre damit überflüssig.

NDR-68K DISK-INSPEKTOR Version 1.10 (C) Copyright 1986 by Klaus Janssen

Laufwerk:	1	Spur:	4	Sektor:	1	Block:	1	Datei:	JADOS.SYS									
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	!!	
000000:	55	AA	01	80	4A	41	44	4F	53	20	20	20	00	00	00	40	!!	0123456789ABCDEF
000010:	00	00	58	9E	01	00	00	00	00	00	00	00	00	00	00	00	!!	U...JADOS ...
000020:	55	AA	01	80	4A	41	44	4F	53	20	20	57	00	00	03	5A	!!	U...JADOS W...Z
000030:	00	00	58	9E	01	00	00	00	00	00	00	00	00	00	00	00	!!	...X.....
000040:	60	00	00	8E	50	41	54	43	48	42	45	47	44	53	4B	30	!!	...PATCHBEGDSKO
000050:	00	01	00	00	05	00	00	00	00	00	00	00	00	00	00	04	!!	.....
000060:	44	53	4B	31	00	02	00	50	05	20	00	03	03	0C	02	80	!!	DSK1...P.....
000070:	00	9C	00	10	44	53	4B	32	00	02	00	50	05	20	00	03	!!	...DSK2...P...
000080:	03	0C	02	80	00	9C	00	10	44	53	4B	33	00	02	00	50	!!	...DSK3...P
000090:	05	20	00	03	03	0C	02	80	00	9C	00	10	44	53	4B	34	!!	...DSK4
0000A0:	00	02	00	50	05	20	00	03	03	0C	02	80	00	9C	00	10	!!	...P.....
0000B0:	53	4F	55	4E	FF	FF	FF	50	41	53	53	54	48	45	4E	44	!!	SOUN...PASST...
0000C0:	45	44	49	54	00	01	00	00	50	41	54	43	48	45	4E	44	!!	EDIT...PATCHEND
0000D0:	61	00	57	4C	41	FA	09	98	3B	50	00	00	3B	50	00	06	!!	a.WLA...;P...;
0000E0:	3B	50	00	0C	41	FA	09	86	2B	48	00	02	2B	48	00	08	!!	+P...A...+H...+H...
0000F0:	2B	48	00	0E	4D	FA	FF	0A	DD	FC	00	00	60	00	28	4E	!!	+H...M.....(N
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	!!	0123456789ABCDEF

Bild 1: Der Patchbereich (dargestellt mit dem Programm INSPEC)

Volker Stahl

# CPU8088 - MS/DOS

## Teil 2: PC-Basiswissen

PC ist die Abkürzung von Personal Computer, das ein Rechnersystem von IBM ist. Doch meistens meinen wir im Zusammenhang mit PC's IBM kompatible Rech-

ner, so wie es auch unser NDR-PC mit der CPU8088 ist. Die einzelnen Rechner aus der PC-Familie werden nach ihrem Prozessor bzw. nach dem Ausbau unterschieden:

Der IBM PC ist ein Rechner mit dem Prozessor 8088 (8-Bit) oder 8086 (16-Bit) und einem Minimalspeicher von 256 kB, sowie einem 5,25" Diskettenlaufwerk mit 360 kB. Der IBM PC/XT hat ebenfalls einen 8088 bzw. 8086 Prozessor, aber noch zusätzlich eine Festplatte.

Der IBM PC/AT hat einen echten 16-Bit Prozessor, den 80286, einen Minimalspeicher von 512 kB und ein AT-Laufwerk (5,25" 1,2 MB).

Zuletzt gibt es dann noch den IBM PC/AT386 mit einem 80386 Prozessor (32-Bit!). Vor kurzer Zeit hat IBM ein neues System herausgebracht, mit der Bezeichnung IBM PS/2. Hier wird wieder im PS/2 System unter den verschiedenen 30-, 50-60- und 80-Modell unterschieden; und zwar in bekannter Weise nach Prozessor und Ausbau. Außerdem ist das PS/2 - System mit 3,5" Diskettenlaufwerken ausgestattet.

Unser NDR-PC entspricht also einem etwas besserem IBM PC oder sogar einem IBM PC/XT! Das 'etwas besser' darum, weil unsere Prozessor-Karte mit 10 MHz getaktet ist und zu vergleichweisen 8088 Prozessoren mit 6 oder 8 MHz wesentlich schneller ist. Durch das Aufstecken des 8087 Coprozessors gewinnt der NDR-PC noch an weiterer Rechengeschwindigkeit (reine Geldfrage!).

Im Zusammenhang mit PC's hört man immer wieder die Begriffe MS/DOS oder PC/DOS. Den meisten 'Computer-Leuten' wird wohl dieser Begriff nicht fremd sein, ebenso wie das bei den NDR-Computer-

**In diesem Teil möchte ich einige Unklarheiten über den PC und seiner Familie beseitigen! Wer schon alles darüber und um die Bedeutung von MS/DOS bzw. PC/DOS Bescheid weiß, soll sich doch Bitte bis zur nächsten LOOP gedulden, und dann in Teil 3 weiterlesen.**

Besitzern weit umstrittene Wort Betriebssystem; dennoch soll es Leute geben, die mit diesen Begriffen nichts anzufangen wissen. Sollten Sie ein solcher Mensch sein, bitte jetzt nicht getroffen fühlen, sondern wissenshungrig weiterlesen!

MS/DOS bzw. PC/DOS sind die Standard-Betriebssysteme der PC's (der Unterschied zwischen MS/DOS und PC/DOS ist dieser, daß PC/DOS im Auftrag von IBM von Microsoft entwickelt wurde und MS/DOS direkt von Microsoft an kompatible Rechner verkauft wird), was auch schon das Kürzel DOS (Disk Operating System) verrät. Ein Betriebssystem stellt die Schnittstelle zwischen Benutzer und Computer her.

Also vergleichsweise wie das Ihnen vielleicht bekannte Betriebssystem CP/M2.2 (beim Z80), CP/M68k oder JADOS (beim 68008). Nur mit dem (gravierendem) Unterschied, daß es für dieses Betriebssystem MS/DOS wesentlich mehr und bessere Anwenderprogramme gibt! Nur nebenbei, gerade deshalb stieg ich auch auf die CPU8088 um

Soviel zur Konfiguration und zum Betriebssystem, jetzt aber etwas für den praktischen User. Haben Sie schon einmal versucht das Textscrollen, z.B. beim TYPE-Befehl, anzuhalten oder abzubrechen? - Ganz einfach: hierzu gibt es Steuerbefehle, meistens aus einer Tastenkombination mit der CTRL (bzw. STRG)-Taste oder der ALT-Taste und einem Buchstaben/Zahl.

Zum Textanhalten ist dies die Tastenkombination CTRL-S (Control-Taste zusammen mit S-Taste drücken), und zum Abbruch CTRL-C; CTRL-C hat die selbe Funktion wie die BREAK-Taste.

Hat sich Ihr Rechner schon einmal 'aufgehängt'? Nichts geht mehr, der Computer nimmt kein Zeichen mehr an und gibt keins

mehr aus, er stellt sich tot. Meistens hilft hier nur noch das Ausschalten, aber halt! Das ist nicht die feine Art - Holzhammermethode! Dazu gibt es den Reseta-

ster. Wenn Sie keinen solchen haben, oder dieser aber an einer unzugänglichen

Teil 1: NDR PC - IBM PC

Teil 2: PC-Basiswissen

Teil 3: EDLIN.COM

Teil 4: CONFIG.SYS

Teil 5: AUTOEXEC.BAT

Stelle angebracht ist, dann drücken Sie doch die Tastenkombination CTRL-ALT-DEL, und die Floppy bootet von neuem.

Mit der PRTSC-Taste (bzw. DRUCK-Taste) können Sie eine Hardcopy Ihres Textbildschirms machen.

Durch das Drücken der ALT-Taste zusammen mit einer dezimalen Zahl, können Sie das dazugehörige ASCII-Zeichen ausgeben. Geben Sie z.B. ALT 65 ein (ALT-Taste drücken, 6 und 5 eingeben, ALT-Taste loslassen), so erscheint der Buchstabe A. Natürlich ist diese Funktion zur Darstellung von Sonder- und Grafikzeichen (siehe ASCII-Tabelle von Microsoft DOS) gedacht!

In der nächsten LOOP lernen Sie in Teil 3: EDLIN.COM den Umgang mit dem Texteditor der DOS-Diskette kennen. Bis dahin viel Spaß beim 'Sondertasten-ausprobieren'.

Volker Stahl

# DSP zum 'reinschnuppern' mit dem NDR-Computer

## Teil 1: Digitale Filterung

Wie schon an anderer Stelle in dieser Ausgabe der LOOP gezeigt wurde, ist die Digitale Signalverarbeitung, kurz DSP (DSP steht für Digital Signal Processing, aber auch für Digital Signal Processor) genannt, ein leistungsfähiges Werkzeug in der Meß- bzw. Steuerungstechnik und nicht zuletzt auch in der Elektroakustik. DSP unter Echtzeiteinsatz macht jedoch leistungsfähige Hardware, vor allem in Hinblick auf RISC-Prozessor und schnelle Speicherbausteine erforderlich. Damit wird DSP schnell relativ kostspielig. Trotz alledem kommt jedoch der Wunsch auf, die faszinierenden Möglichkeiten der digitalen Signalverarbeitung mit dem NDR-Computer wenigstens in Grundzügen nachzuvollziehen.

### Mit dem NDR-Rechner kein DSP möglich?

Wenn man einmal kurz Bilanz zieht und bedenkt, welche DSP-spezifischen Baugruppen für den NDR-Rechner verfügbar sind, befindet man sich im wesentlichen auf dem Punkt Null. Dies ist im Grunde auch nicht weiter schlimm, wenn man sich (oder vielmehr dem Rechner) ein wenig Zeit läßt und die anfallenden Signalanalysen und Signalmanipulationen nicht in Echtzeit, sondern eben 'nachher' erledigt. Dabei muß man sich um die notwendigen Algorithmen voll und ganz selbst kümmern, was nebenbei äusserst lehrreich ist. Bei dieser Vorgehensweise gliedert sich ein hypothetischer Programmablauf im Wesentlichen in 3 Teile:

- Meßwerte in Block einlesen
- Manipulation der Werte im Speicher
- Ausgabe der Ergebnisse in Block

Obwohl im Allgemeinen eine Wissenschaft für sich, ergeben sich beim ersten und beim letzten Punkt die geringsten Probleme für den NDR-Anwender. Schon oft wurde in der LOOP der Einsatz der AD- und DA-Wandlerbaugruppen geschildert.

**DSP ist zu einem vielbenutzten Schlagwort der modernen Rechneranwendung geworden. In diesem Artikel soll unter anderem untersucht werden, wie denn der NDR-Rechner dieser Herausforderung gegenübersteht. Desweiteren ein wenig Grundlageninformation zur digitalen Filterung und zur Spektralanalyse von Signalen. Ein ausgeführtes Anwendungsbeispiel zur Fourieranalyse soll den zweiten Teil des Artikels abrunden.**

Zu beachten ist lediglich der Einsatz der obligatorischen Antialiasingfilter mit einer Grenzfrequenz  $F_g$  unterhalb der halben Abtastfrequenz der Wandlerbaugruppe. Die Wartezeit zwischen 2 Abtastungen kann für erste Versuche (obwohl unelegant) mit Softwarewartezeiten ausgefüllt werden. Für weitergehende Versuche könnte durchaus eine Hardwaretimer-Lösung ins Auge gefaßt werden. Wie groß die Anzahl der einzulesenden Meßwerte gewählt werden kann, hängt natürlich in erster Linie vom maximal freien Speicherplatz ab. Aus den beiden Kriterien Abtastfrequenz und max. Speicherplatz resultiert dann letztendlich die gesamte Aufzeichnungsdauer (Sampletime). Zum Beispiel erhält man pro 64kByte freiem Speicherplatz bei einer Abtastrate von 8kHz (Daten aus der Fernsprechtechnik: Sprachsignale von 300Hz bis 3400Hz werden mit 8kHz abgetastet) eine Gesamtaufzeichnungsdauer von etwas über 8s. Die Abtastrate von 8kHz kann sogar noch von der langsamsten NDR-AD-Wandlerkarte erbracht werden, nämlich von der AD 8\*16.

### Wie werden digitale Eingabewerte verarbeitet?

Etwas gründlicher soll nun im Folgenden auf den Punkt b) in der obigen Auflistung eingegangen werden: Manipulation der zuvor eingelesenen Meßwertkolonnen. Hierfür werden in der Regel zwei unterschiedliche Verfahren eingesetzt: Einmal die digitale Filterung als direktes Verfahren auf der einen Seite und die Fouriertransformation als quasi indirektes Verfahren. Die Bezeichnungen direkt und indirekt wurden hier vom Autor eingeführt, um zu verdeutlichen, daß die Filterung eine Methode aus-

schließlich im Zeitbereich ist, wogegen die Transformation den Übergang in den Frequenzbereich, die Manipulation der Spektrallinien und schließlich die Rücktransformation in den Zeitbereich beinhaltet. Doch zunächst einmal langsam

und der Reihe nach:

### Grundlagen der digitalen Filterung

Zuerst wollen wir die digitale Filterung beleuchten. Bei der Erläuterung dieser Filter wird im Allgemeinen von der analogen Technik ausgegangen, so auch hier: Betrachten wir einen analogen, aktiven Integrator, wie er in Abb. 1a dargestellt ist.

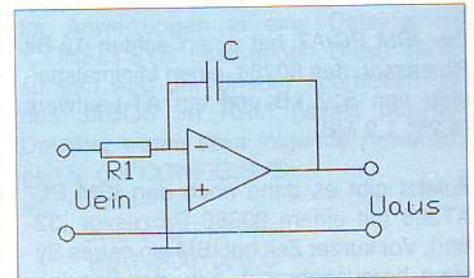


Abb. 1a: Aktiver Integrator

Die Funktionsweise liegt klar auf der Hand: Die am Eingang der Schaltung anliegende Spannung wird fortlaufend aufsummiert, das heißt positive Eingangsspannungen erhöhen die Ausgangsspannung, negative Eingangsspannungen reduzieren die ausgegebene Spannung wieder. Ist die Eingangsspannung gerade auf Null Volt angelangt, bleibt die Ausgangsspannung auf dem Niveau stehen, das sie zuvor erreicht hatte. Es erscheint offensichtlich, daß die Funktion in einfachster Weise von einem digitalen Akkumulator simuliert werden kann; einfach alle eingehenden Zahlen aufaddieren. Man erkennt aber auch, daß es hierbei schnell zu einem Überlauf des Akkumulators kommen kann, wenn nämlich die Ausgewogenheit der Eingangsdaten nicht stimmt. In so einem Fall wird auch das analoge Pendant durch die Versorgungsspannung begrenzt.

Wie können wir das bisher Erarbeitete verwenden? Ein Integrator ist schließlich noch kein Filter im eigentlichen Sinn! Durch Hinzunahme eines ohmschen Widerstandes R2 parallel zum Kondensator C wird daraus sehr wohl ein Tiefpassfilter, auch Mittelwertbildner genannt, wie in Abb. 1b gezeigt.

Es ist hier ein Filter erster Ordnung entstanden, da nur ein Energiespeicher enthalten ist. Welche Funktion erfüllt dieser Widerstand R2? Er dient sozusagen zur Verringerung des Leckstromwiderstandes von C und entlädt den Integrationskondensator C proportional zur momentanen Spannung an den Platten des Kondensators. Die Spannung am Kondensator ist jedoch exakt gleich der Ausgangsspannung der Filterschaltung, da der invertierende Eingang des Operationsverstärkers eine virtuelle Masse darstellt (der OP versucht mit allen Mitteln, die ihm zur Verfügung stehen die Spannung an beiden Eingangsklemmen auf gleiches Potential zu bringen, wobei hier der nichtinvertierende Eingang fest mit Masse verbunden ist). Die Richtung des Stromes durch den Widerstand R2 ist stets so, daß C entladen wird.

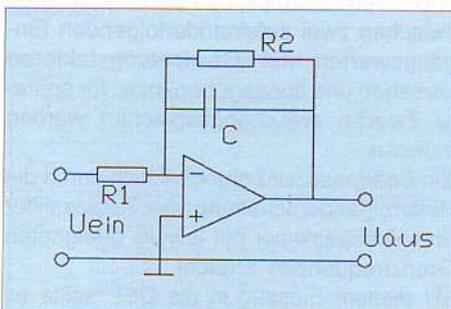


Abb. 1b: Aktiver Mittelwertbildner

Sie ahnen schon, worauf dieser Vergleich hinführen soll: Die entladende Wirkung von R2 kann auf digitaler Ebene durch Subtraktion eines proportionalen Anteils vom augenblicklichen Akkumulatorstand simuliert werden. Wie groß dieser prozentuale Anteil ist, der vom Akkumulatorinhalt abgezogen werden muß, läßt sich einfach berechnen:

Durch den Entladeeffekt von R2 hat die Spannung über C das Bestreben nach einer e-Funktion abzunehmen:

$$U_c = U_0 \cdot \exp(-t/(R2 \cdot C)) \quad (1)$$

Hierbei ist t die laufende Zeit, U<sub>0</sub> die Kondensatorspannung zum Startzeitpunkt der Betrachtung und R2·C die sogenannte Zeitkonstante des Tiefpasses. Zum Übergang vom zeitkontinuierlichen Bereich in

den zeitdiskreten Bereich (von analog nach digital) ist es besser, obige Formel etwas umzuformen:

Aus der laufenden Zeit t wird die Zeit T, die nun nur noch in Schritten der Abtastintervalle hochgezählt wird und R2·C wird ausgedrückt durch die Grenzfrequenz f<sub>g</sub> des gewünschten Filters: R2·C = 1/(2·π·f<sub>g</sub>). Damit ergibt sich für die Kondensatorspannung

$$U_c = U_0 \cdot \exp(-2 \cdot \pi \cdot f_g \cdot T) \quad (2a)$$

bzw. mit der Abtastfrequenz F<sub>A</sub>=1/T des AD-Wandlers:

$$U_c = U_0 \cdot \exp(-2 \cdot \pi \cdot f_g / F_A) \quad (2b)$$

Daraus kann man jetzt den geeigneten Prozentsatz K ermitteln, dernach jeder Abtastperiode vom digitalen Akkumulator abgezogen werden muß:

$$K = 1 - \exp(-2 \cdot \pi \cdot f_g / F_A) \quad (3)$$

Nebenbei bemerkt, jetzt sind keine absoluten Frequenzen mehr maßgebend, sondern lediglich das Verhältnis von der Grenzfrequenz f<sub>g</sub> des Filters zur Abtastfrequenz F<sub>A</sub>! Die mathematisch korrekte Formulierung des Filters würde nun folgendermaßen aussehen:

$$x(i+1) = x(i) - K \cdot x(i) + y(i+1) \quad i=1 \dots N \quad (4)$$

Der Vektor x nimmt alle Ausgabewerte auf und Vektor y beinhaltet die angesammelten Eingabewerte der Anzahl N. Jetzt sind wir unserem digitalen Filter 1. Ordnung schon recht nahe, wenn man aber ein kurzes Demoprogramm in BASIC o.ä. schreiben würde, käme eine unerwünschte Gleichanteilverstärkung in der Ausgangszahlenfolge zutage. Es würde sich herausstellen, daß dieser Effekt abhängig vom 'Leckstromfaktor' K ist (kleine Ks bringen eine hohe Gleichanteilverstärkung, große Ks dagegen weniger).

Um dieser Erscheinung entgegenzuwirken, können die eintreffenden Abtastwerte mit dem Faktor K selbst korrigiert werden. Gl.4 nimmt damit folgende Form an:

$$x(i+1) = x(i+1) - K \cdot (x(i) + y(i+1)) \quad (5)$$

Ein Basicprogrammsegment, das eine solche Tiefpassfilterung durchführt, könnte etwa so aussehen:

```
100 FOR I=1 TO N-1
110 AKKU(I+1)=AKKU(I)-K*(AKKU(I)+EINGABE(I))
```

### Von mathematischen Gleichungen weg zum Blockschaltbild

An dieser Stelle sind wir nun schon soweit, die erarbeitete Gleichung für ein digitales Tiefpassfilter in die allseits bekannte Blockschaltbildform zu übertragen. In der Gleichung(5) kommen zwei Additionen und eine Multiplikation vor. Desweiteren finden wir hier die Zwischenspeicherung des Ausgangswertes x(i) vor. Für diese drei mathematischen Operationen (in der Mathematik nennt man das Zwischenspeichern auch ein 'Verschieben auf der Zeitachse') existieren die folgenden Symbole:

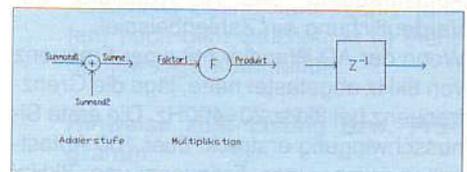


Abb. 2: Math. Rechenvorschriften in Bildsprache ausgedrückt

Damit kann man jetzt ein Übersichtsbild zusammenbauen, aus dem der Vorgang der Filterung mit einem Blick erfassbar wird:

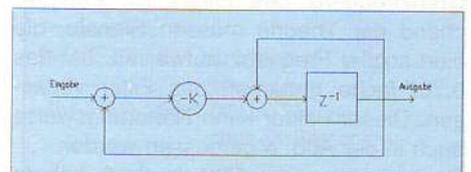


Abb. 3: Eine mögliche Darstellung eines Tiefpasses 1. Ord.

Zur Erläuterung des Blockschaltbildes folgende Bemerkung: Die Gleichung(5) für das Tiefpassfilter wird quasi von hinten her aufgearbeitet. Innerhalb des Klammersausdruckes wird der alte Zählerstand mit dem neuen Eingabewert addiert, dies wird im Blockschaltbild durch das Zurückführen des alten Ausgabewertes auf den Additionspunkt am Eingang verdeutlicht. Die entstandene Summe der beiden wird nachfolgend mit dem Faktor (-K) multipliziert und in einen weiteren Additionspunkt eingespeist. Dort wird noch einmal der alte Zählerstand von hinten herübergeleitet und eine weitere Summe gebildet.

Dieses Zwischenergebnis wird jetzt in den Verzögerungskasten hineingeführt und steht somit auch gleich wieder am Ausgang dieses Kastens an.

Jetzt kann der ermittelte Wert am Ausgang des Blockschaltbildes als neuer Ausgangswert abgenommen werden. Wenn sich wegen eines nachfolgenden Eingabewertes ein weiterer Tiefpassfilterdurchlauf anschließt, fällt dem eben so mühevoll errechneten und zwischengespeicherten Ausgangswert die Rolle des 'alten Ausgangswertes' zu. So kurzlebig ist unsere Zeit nun einmal!

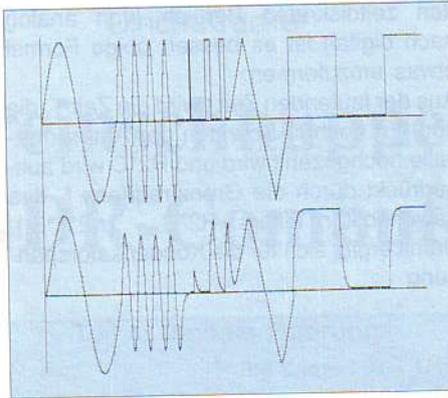


Abb. 4: Ergebnis einer digitalen Tiefpassfilterung.

Abb. 4 zeigt das Ergebnis eines Demoprogrammes zum Tiefpaßfilter, das exakt die oben aufgeführten Programmzeilen enthält. Bei der Ermittlung des Leckstromfaktors  $K$  wurde entsprechend der Gleichung (3) ein Verhältnis von  $F_g/F_A=1/20$  festgelegt, um  $K=0.269$  zu erhalten. Die Ergebnisse können folgendermaßen interpretiert werden:

Die Grenzfrequenz liegt  $1/20$  unterhalb der Abtastfrequenz des AD-Wandlers. Zur Verdeutlichung ein Zahlenbeispiel:

Wenn der AD-Wandler mit einer Frequenz von 8kHz abgetastet hätte, läge die Grenzfrequenz bei  $8\text{kHz}/20=400\text{Hz}$ . Die erste Sinusschwingung erstreckt über 100 Abtastzeiten, was einer Frequenz von  $8\text{kHz}/100=80\text{Hz}$  entspräche. Aus diesem Grund erscheint diese Sinusschwingung nahezu ungedämpft am Filterausgang. Im Vergleich dazu haben die anschließenden vier Schwingungen eine Periodendauer von nur 20 Abtastzeiten, gleichbedeutend mit einer Signalfrequenz, die gleich der Grenzfrequenz des Filters ist. Entsprechend der Theorie müssen Signale, die eine solche Frequenz aufweisen, um das 0.707-fache gedämpft das Filter verlassen. Dieser Faktor kann näherungsweise auch in der Abb. 4 gemessen werden.

Der resultierende Signalverlauf könnte ebenso durch Abtastung des Ausgangs eines hypothetischen analogen Tiefpassfilters realisiert werden, wobei die Abtastfrequenz das 20-fache der Grenzfrequenz des Tiefpassfilters betragen müßte.

Oben die Eingabewerte, unten die Ausgabewerte des Filters.

Vom Filter 1.Ordnung zu komplexeren Gebilden

Um komplexere Filter zu realisieren, muß man nur dafür sorgen, daß ein einmal errechneter Ausgangswert nicht so schnell an Aktualität verliert. Will sagen, man muß die Ausgangswerte länger, bzw. öfter zwischenspeichern, damit sie mit Gewichtungsfaktoren versehen noch mehrere nachfolgende Ausgangswertegenerationen beeinflussen können. In der Blockschaltbild-Darstellung könnte dies so aussehen:

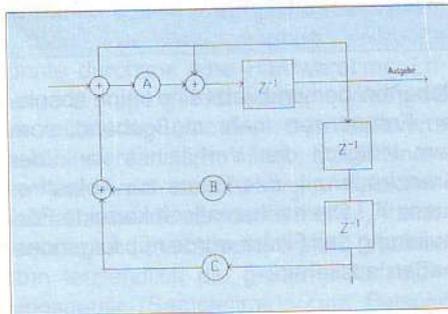


Abb. 5: Komplexere Filter auf der Basis des Tiefpasses 1. Ordnung

Der Pferdefuß bei der Angelegenheit ist der, daß mit zunehmendem Umfang des Filters natürlich auch die Anzahl der Zwischenspeicher und vor allem die Anzahl der Multiplikationen anwächst.

Auf eine Besonderheit dieser Filteranordnung sei noch hingewiesen: Angenommen auf den Eingang des Filters gelangt einmal ein Ausreißer in Form eines besonders hohen Zahlenwertes (Nadelimpuls), so sind dessen Auswirkungen auch noch nach vielen Generationen von nachfolgen-

den Ausgangswerten zu spüren. Theoretisch zieht ein solches Geschehnis aufgrund der geschlossenen Kreisstruktur des Filters unendlich lange seine Spuren hinter sich her. Deshalb nennt man diesen und ähnliche Filteraufbauten auch Infinte Impuls Response Filter, kurz IIR, was soviel bedeutet, wie unendlich langes Antworten auf ein Impulsereignis.

IIR und FIR (Finite Impuls Response als Gegensatz dazu) sind zu stehenden Begriffen in der digitalen Filtertechnik geworden.

### Hoch- und Bandpass, ähnlich aufgebaut wie der Tiefpass

Bisher haben sich alle Erklärungen nur um das Tiefpassfilter gedreht, wie sieht es aber mit den übrigen Filterarten wie Hochpass, Bandpass und so weiter aus?

Es wäre wohl etwas langatmig, die obigen Ausführungen ähnlichen Herleitungen zu wiederholen, aber als Gedankenreiz die Ansätze zum Hochpass:

Ein solches Filter wirkt im Gegensatz zum Integrator als Differenzierer. In der Digitaltechnik werden Differenzen durch Subtraktion gebildet. Angewandt auf die Filterpraxis bedeutet dies, daß die Unterschiede zwischen zwei aufeinanderfolgenden Eingangswerten mit Gewichtungsfaktoren versehen und ausgegeben, bzw. für spätere Zwecke zwischengespeichert werden müssen.

Ein Bandpass wird grundsätzlich durch die Hintereinanderschaltung von Tiefpassfilter und Hochpassfilter mit jeweils geeigneten Grenzfrequenzen erreicht.

Mit diesem Einstieg in die DSP sollte es nun durchaus möglich sein, ein entsprechendes Programm zu entwerfen. Bei Einsatz von Assembler Routinen wird am besten mit Fixkomma-Arithmetik gearbeitet, da die Filterkoeffizienten immer kleiner als 1 sind.

Im zweiten Teil des Artikels soll die Zerlegung der Signale in deren Frequenzanteile näher beleuchtet werden.

#### Literatur:

-Hal Chamberlin: Musical applications of microprocessors, Hayden Book Company INC, New Jersey

Neu

mc-modular -AT  
Neue AT-CPU-Karte:

25 Mhz AT 386 'er

aufrüstbar bis maximal 8MB on Board ( SIP-Module)

DM 2.648,- ( Ohne RAM ) 1MB RAM DM 590,- (SIP)

Stefan Oroz

# Welche Hardware ist beim NDR-PC verfügbar ?

Mit 10 Pfennig Materialkosten und einem LötKolben kann man das Problem lösen. Doch dazu später beim TIP. Es ist sehr hilfreich die Systemkonfiguration softwaremäßig zu ermitteln, damit sämtliche Möglichkeiten, die beim eigenen PC vorhanden sind richtig ausgeschöpft werden können. Nach dem Einschalten oder Reset des PC's wird die Systemkonfiguration automatisch ermittelt, jedoch nicht gleich am Bildschirm sichtbar gemacht.

Wenn man feststellen will, welche Hardware und Ram-Speichergröße vorhanden ist, kann man dies mit Hilfe von Interrupt 11H (siehe Bild 1) und 12H (siehe Bild 2) erreichen.

(Bild 1)

```

Interrupt 11h: Konfiguration feststellen
Rückgabewerte AX=Konfigurationsbyte

Diese DOS-Funktion ermittelt die Bitbelegung der
Systemkonfiguration.
  
```

Bits	Belegung
0	1, wenn im System Diskettenlaufwerk verfügbar ist
1	Keine Bedeutung (in manchen Systemen I=8087 vorhanden)
2,3	Speicherbank-Größe in Kbyte 00 = 16 KB, 01 = 32 KB, 02 = 48 KB, 11 = 64 KB
4,5	Bildschirmmodus beim Booten 10 = 40*25, 01 = 80*25 (Color), 11 = 80*25 (Mono)
6,7	Anzahl der Diskettenlaufwerke wenn Bit 0 = 1 ist 00 = 1, 10 = 2, 01 = 3, 11 = 4 Laufwerke
8	Keine Bedeutung
9,10,11	Anzahl der seriellen Schnittstellen 100 = 1, 010 = 2, 001 = 3 serielle Schnittstellen
12	1, wenn Spieladapter angeschlossen
13	Keine Bedeutung
14,15	Anzahl der Druckeranschlüsse (1-3) 10 = 1, 01 = 2, 11 = 3 Druckeranschlüsse

## Erklärung zu Bild 1:

Um die Konfiguration der Bitbelegung zu ermitteln, muß man die ROM-BIOS-FUNKTION (Interrupt 11 H) aufrufen. Das Ergebnis wird dem AX-Register in hexadezimal übergeben, z.B. 427D H, dies muß in binär umgewandelt werden = z.B. 1011 1110 0100 0010 B, damit man nach der (in Bild 1 zu sehenden) Tabelle feststellen kann, welche Hardware vorhanden ist. Diese Umwandlung (von hexadezimal in binär) wird im Programm ab dem Label DUALWERT bis Label WEITER1 ausgeführt. Gleichzeitig wird dies am Bildschirm ausgegeben. Jedes einzelne Bit, also Bit 0 bis 15 wird auch gleichzeitig im CONFIG zwischengespeichert, weil die einzelnen BIT noch für die Auswertung gebraucht werden.

## Erklärung zu Bild 2:

Beim Aufruf von Interrupt 12H wird die Speichergröße in hexadezimal dem AX-

**Das Programm NDRCHECK habe ich eigentlich deshalb geschrieben, weil ich mit dem Drucker Schwierigkeiten hatte, mal druckte er und mal nicht, bis ich drauf kam, daß der Drucker an der IOE-Karte angeschlossen sein muß, bevor man den PC einschaltet, sonst findet das Betriebssystem die Druckerkarte (IOE-Karte) nicht.**

Register übergeben, dies muß dann in dezimal umgewandelt werden. Die Proedur

(Bild 2)

```

Interrupt 12h: Speichergröße ermitteln
Rückgabewerte AX = Speichergröße

Diese DOS-Funktion ermittelt die Speichergröße des Systems in
Kbyte
  
```

HEX\_AUS aus dem Programm übernimmt die Umwandlung und gibt dies am Bildschirm aus. Es kann nur ein Gesamt-Ram-Speicher von 640 KB ermittelt werden (über Interrupt 12H). Wenn man den Erweiterungsspeicher ermitteln will, kann man dies wie folgt: AH-Register mit 88H laden und INTERRUPT 15H aufrufen. Dies ist nur beim AT sinnvoll. Denn wenn kein Erweiterungsspeicher vorhanden ist, zeigt das AX-Register den Wert 0 an. Bei einem PC jedoch wird ein Wert angezeigt, z. B. (8000 h) 32781 KB, auch wenn keine Erweiterung vorhanden ist. Anhand der 3. Abbildung ist zu sehen, wie das Programm NDRCHECK aussieht und wie es zu starten ist.

(Bild 3)

```

A:\NDRCHECK ( CR )

SYSTEMKONFIGURATION

Bitbelegung der Prüfbites: 1011 1110 0100 0010
Kein Coprozessor (8087) vorhanden
64 KB hat eine Speicherbank in Hauptspeicher
640 KB Gesamt-RAM-Speicher vorhanden
Bildschirmmodus: 80 * 25 Monochromkarte
2 Floppy-Laufwerke vorhanden
1 Serielle Schnittstelle vorhanden
Kein Spieladapter vorhanden
1 Drucker Schnittstelle vorhanden

A:\>
  
```

## Folgende Hardware ist bei mir vorhanden:

Speicher: 2 \* RAM64/256 und 2 \* ROA64  
 Bildschirnkarte : GDP64  
 Serielle Karte : SER  
 Druckerkarte : IOE Druckeranschluss nach IOE Handbuch  
 Floppy: Flo2 und zwei Laufwerke 1 MB mit 80 Spuren

## Hinweis:

Wenn der NDR-PC mit der FLO (2 oder 3) betrieben wird und nur ein Diskettenlauf-

werk vorhanden ist, zeigt das Programm trotzdem ein 2. Laufwerk an. Der Fehler ist nicht auf das Programm zurückzuführen, sondern auf die FLO. Das Programm wurde auf anderen PC's und AT's getestet und lief fehlerfrei.

## Hinweise zum Listing:

Unter der Rubrik ADR stehen die Hex-Adressen. Unter Hex-Code stehen die Maschinenbefehle (bei Debug-Eingabe, direkte Übernahme möglich). Unter Assembler steht das Assembler-Listing. Ganz rechts (unten) nach dem Strichpunkt (;) steht der jeweilige Kommentar.

## Hinweise zum Listing bzw. Programm:

### Byte Reservierung:

FLAG 1 reserviertes Byte für die Proedur  
 HEX\_AUS reservierte Bytes für die Bitbelegung (Auswertung)  
 CONFIG 16

### Proceduren:

NEUE\_ZEILE macht einen Zeilenvorschub  
 ZEICHEN\_LEER gibt ein Leerzeichen am Bildschirm aus  
 ZAHL\_AUS gibt eine Zahl am Bildschirm aus  
 TEXT\_AUS gibt einen Text am Bildschirm aus  
 HEX\_AUS wandelt die Hexzahl  
 FRAGE Abfrageroutine ob DL-Register den Wert 1 hat, wenn nein, dann den Text "Kein" ausgeben

AB den Label MARKE bis CODE wird die im CONFIG gespeicherte Konfiguration ausgewertet und ausgegeben (mit entsprechendem Text). Das Programm (vorliegendes Listing) läßt sich mit einem Public Domain Assembler (wie z. B. A86, sofern Sie einen besitzen) übersetzen. Bei

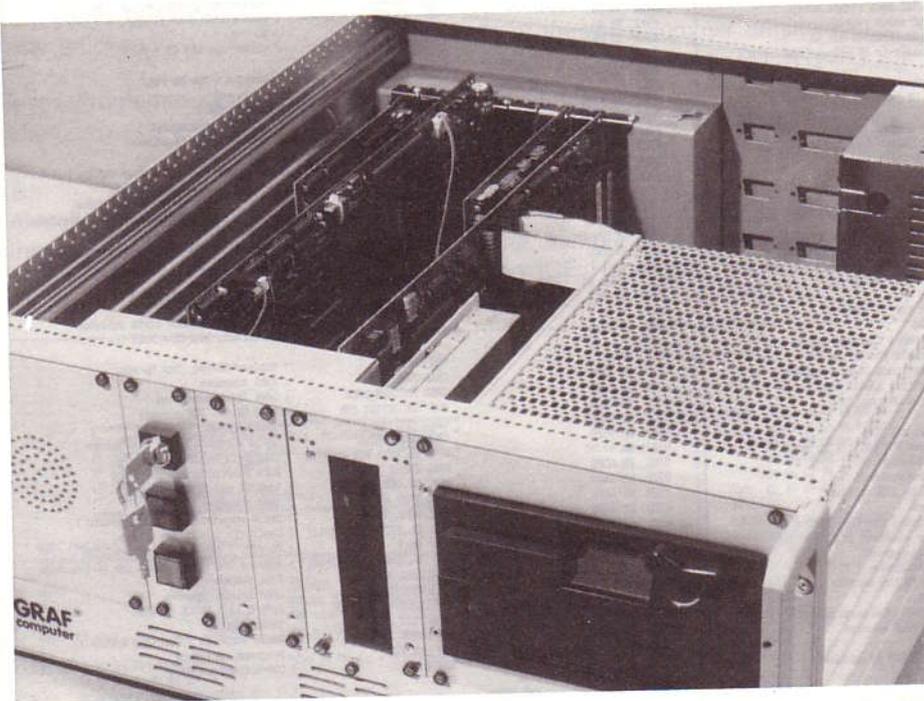




Nikolaus Bischof

# Der 19" Industrie AT

Vor fast genau zwei Jahren haben wir in Zusammenarbeit mit der Zeitschrift mc des Franzis-Verlages den ersten modularen AT, den mc-modular AT auf den Markt gebracht. Dieser modulare Computer fand in der Zwischenzeit viele Freunde. Vor allem durch den passiven Bus und die steckbare CPU-Karte fand der Rechner auch in der Industrie Einzug.



Für industrielle Anwendungen sind Rechner im herkömmlichen Gehäuse und mit den herkömmlichen Sicherheitsstandards meistens nicht ausreichend. Deshalb war der erste Gedanke, den Rechner in ein industriell geprüftes 19" Gehäuse einzubauen. Um hier dem modularen Aufbau des Rechners nachzukommen, haben wir das Gehäuse ebenfalls möglichst modular gehalten. So können z.B. die Stecker auf den Leiterplatten nach hinten zur Rückwand oder nach vorne auf die Frontplatte gelegt werden. Es bietet Platz für mindestens drei 5 1/4" Massenspeicher und für mindestens zwei 3 1/2" Laufwerke.

Das 200 Watt Netzteil ist dem Rechner angepaßt und ist als handlicher 19" Einschub 4 HE ausgeführt. Selbstverständlich sind Zubehör, wie Hard-RESET, LED's für die Spannungsversorgungen und die Festplatte, extern zugänglicher Batteriehalter mit Lithium-Batterie, Schlüsselschalter, Turbo-Taster, etc.

Aber nicht die Verpackung macht einen AT zum Industrie AT. Es gilt vielmehr die

Schwachpunkte eines AT's auszubügeln. Jeder der einen AT oder ähnlichen Rechner hat kennt die Schwachpunkte eines solchen Rechners. An erster Stelle kann hier sicher die Festplatte genannt werden. Hier treffen gleich zwei Faktoren aufeinander: die Gefahr durch Vibrationen oder starke Stöße und die Anfälligkeit der Festplatten gegen Überhitzung. Beide Punkte lassen sich natürlich nur in gewissen Schranken in den Griff bekommen, so wäre es z.B. unsinnig eine Festplatte in der Raumfahrttechnik einzusetzen. In vielen Fällen im industriellen Anwendungsspektrum liegen sowohl die thermischen als auch die kinetischen Bedingungen in dem Rahmen, um eine Festplatte einsetzen zu können. Um die Betriebssicherheit zu erhöhen, haben wir folgende Veränderungen vorgenommen:

- 1) Die Festplatte wird schockabsorbierend gelagert. Vor allem niederfrequente Schwingungen und starke Stöße können dadurch gedämpft werden.
- 2) Ein zweiter Ventilator sorgt dafür, daß die Festplatte auch bei höheren Temperaturen sicher läuft. Test haben ergeben, daß die Umgebungstemperatur des Rechners bis

zur Spezifikationsgrenze der Festplatte liegen darf, d.h. die Belüftung der Festplatte ist so gut, daß die Temperatur im Rechner nur wenig zur Umgebungstemperatur differiert.

Andere Schwachstellen von AT's liegen in dem Komplex SETUP und Uhr. Wer hat sich nicht schon aufgeregt, wenn nach dem Einschalten eine Fehlermeldung betreffs SETUP erscheint. Die Ursache eines solchen Defekts liegt entweder an der zur Neige gehenden Batterie oder Akku, oder an einem unsauberen RESET des Rechners, der dann seinen eigenen SETUP nicht mehr findet oder überschreibt. Diese beiden Effekte wurden durch eine externe Lithium-Batterie, die von außen gewechselt werden kann, und durch eine RESET-Schaltung, die die Spannungsversorgung kontrolliert, beseitigt.

Ein weiterer wichtiger Punkt ist die Auswahl der einzelnen Baugruppen. Als autorisierter IBM-Händler und als Vertreter des mc-modular AT verfügen wir über die entsprechende Erfahrung, welche Karten hier einsetzbar sind. Der Floppy- und Harddisk-Controller sowie die EGA Karte sind z.B. 90% SMD bestückt.

Ein wichtiger Gesichtspunkt ist auch die Stabilität der Steckverbinder. Sämtliche Kabel werden in der eigenen Fertigung mit AMP-Verbindern gefertigt und geprüft. Alle Verbindungen sind entweder durch Verschraubung oder durch Haltebügel (Festplatte) gesichert. Um ein Ausklinken der Steckkarten zu verhindern, werden die Karten durch Abstandshalter von oben gesichert.

Ebenfalls eine nicht unwesentliche Rolle spielt die Störanfälligkeit durch elektromagnetische Wellen oder durch hochfrequente Störungen, die sich über das Netz in das System einschleusen. Die Netzstörungen ließen sich durch einen Netzfilter stark reduzieren. Die Anfälligkeit des Systems gegen Einstrahlung von Wellen lassen sich nur durch konsequente Abschirmmaßnahmen reduzieren. Vor allem die Einstrahlung durch die Verbindungen zu den Peripheriegeräten ließ sich durch geeignete Maßnahmen stark reduzieren. Im Bezug auf die HF-Dichtigkeit des System müssen Kompromisse mit der thermischen Belastung des Rechners gemacht werden.

Wir sind sicher, daß wir mit dem 19" Industrie AT einen zuverlässigen und leistungsfähigen modularen Rechner kreiert haben, der vielen industriellen Anwendungen gewachsen ist. Das System wird von uns an die Wünsche der Kunden angepaßt.

G Klotsche

# Verwendung eines Erweiterungsspeichers unter MS-DOS oder PC-DOS

## A: RAM-Laufwerk:

### RAM-Laufwerk allgemein:

Da der DOS-Hauptspeicher mit 640 KB ohnehin schon für manche Anwenderprogramme knapp bemessen ist, kann man davon nur in sehr begrenzten Maße Speicherplatz für ein RAM-Laufwerk abzweigen (etwa bis 100 KB). Daher ist es zweckmäßig den Erweiterungsspeicher zum Einrichten eines RAM-Laufwerks zu verwenden. Grund dafür ist, daß nur sehr wenige Programme auf diesen Speicherbereich zugreifen können, und somit meist der ganze Erweiterungsspeicher als RAM-Laufwerk benutzt werden kann.

### Warum RAM-Laufwerk und wofür?

Die meisten Speichermedien, wie z.B. Festplatte oder Diskette, haben den Nachteil der mehr oder weniger langen Zugriffszeit. Ein RAM-Laufwerk hat dagegen fast die gleiche Zugriffszeit wie der Hauptspeicher. Daraus ergeben sich dann Vorteile z.B. beim Compilieren von Programmen. So kann man fast auf das Compilieren des Programms in den Hauptspeicher verzichten und gleich eine Datei vom Typ .EXE oder .COM auf dem RAM-Laufwerk erzeugen. Ebenso kann das RAM-Laufwerk als Zwischenspeicher beim Kopieren von Dateien z.B. auf verschiedene Diskettenformate dienen. Natürlich hat ein RAM-Laufwerk nicht nur Vorteile. So gehen alle auf dem RAM-Laufwerk gespeicherten Daten beim Neustarten oder Stromausfall des Rechners verloren. Wichtige Dateien daher immer rechtzeitig auf eine Diskette oder Festplatte kopieren!

### Verwendung der Datei RAMDRIVE.SYS oder VDISK.SYS in der CONFIG.SYS:

Genauere Deklaration:

```
device=Ramdrive.sys [Diskettengröße] [Sektorengröße]
                               [Einträge]/[d]
```

oder

```
device=Vdisk.sys [Diskettengröße] [Sektorengröße] [Einträge] /[d]
```

**Bedeutung der einzelnen Parameter:**  
Diskettengröße:

**MS-DOS oder PC-DOS bietet die Möglichkeit, einen Teil des Haupt- oder Erweiterungsspeichers als logisches Laufwerk zu verwenden.**

**Die Einrichtung eines RAM-Laufwerks erfolgt durch die Verwendung von RAMDRIVE.SYS (MS-DOS) oder VDISK.SYS (PC-DOS) in der CONFIG.SYS.**

Gibt die Größe des RAM-Laufwerks in k-Byte Minimum 16 k-Byte; ohne Angabe 64 KByte

Sektorengröße:

Legt die Sektorengröße in Byte fest. Möglich Werte sind 128,256,512,1024 Byte; ohne Angabe 128 Byte

Einträge:

Legt die Anzahl der Einträge im Stammverzeichnis fest. Es wird auf Vielfache von 16 aufgerundet. Minimum 4; ohne Angabe 64

/[d]: Auswahl des Speicherbereichs:

Ohne Angabe wird der Speicherplatz für das RAM-Laufwerk vom Hauptspeicher abgezogen

/E Wählt für das RAM-Laufwerk den Extensiv-Verwendung eines Erweiterungsspeichers unter MS-DOS oder PC-DOS

/A Wählt für das RAM-Laufwerk den Extensiv-Speicherbereich.

### Anwendungsbeispiele:

Die Beispiele, anhand von RAMDRIVE.SYS gezeigt, sind aber auf VDISK.SYS völlig übertragbar.

#### 1. device=Ramdrive.sys

Diese Befehlszeile in der CONFIG.SYS richtet ein RAM-Laufwerk mit 64 KByte im DOS-Hauptspeicher ein. Die Sektorengröße beträgt 128 Byte. Es sind 64 Einträge im Stammverzeichnis möglich.

#### 2. device=Ramdrive.sys /A

Gleicher Umfang des RAM-Laufwerks wie unter 1., nur wird das Laufwerk nicht im Hauptspeicher, sondern im Expanded-Speicher eingerichtet

#### 3. device=Ramdrive.sys 384 256 80/E

Mit dieser Befehlszeile wird ein RAM-Lauf-

werk mit 384 k-Byte im Extended-Speicher eingerichtet. Die Sektorengröße beträgt 256 Byte. Es sind 80 Einträge im Stammverzeichnis möglich.

## 4. Verwendung des Erweiterungsspeichers als Festplattenpuffer:

### Gründe zur Verwendung eines Festplattenpuffers:

Umfangreiche Anwenderprogramme müssen oft Datensätze von Festplatte nachladen, um überhaupt mit 640 KByte DOS-Hauptspeicher auszukommen. Mit Cache-Speicher-Programmen wird versucht diese zeitintensiven Schreib- und Lesevorgänge zu umgehen. Dazu werden die von Anwenderprogrammen häufig verwendeten Datensätze im Erweiterungsspeicher abgelegt.

Eines dieser Programme ist SMARTDRV.SYS, das die oben beschriebenen Eigenschaften besonders in Verbindung mit MS-Windows voll ausschöpft. Um aber einen Datenverlust beim Ausschalten oder Neustarten des Rechners zu umgehen, kopiert SMARTDRV.SYS die modifizierten Datensätze sowohl auf die Festplatte als auch in den Festplattenpuffer.

### Voraussetzungen und Verwendung:

Voraussetzung zur Verwendung von SMARTDRV.SYS ist ein IBM/AT oder ein kompatibler Rechner mit Expanded- und oder Extended-Speicher. Weiter sollten keine anderen Festplattenpuffer oder virtuelle Laufwerke installiert sein.

Verwendung in der CONFIG.SYS:

```
device=Smartdrv.sys [Größe] [/A]
```

### Parameterbedeutung:

Größe :

Mit diesem Parameter wird die Speicherkapazität in KByte festgelegt. Ohne Angabe werden 256 k-Byte eingerichtet.

/A : Wird angegeben, wenn der Expanded-Speicher verwendet werden soll. Bei einer Erweiterungskarte mit Expanded- und Extended-Speicher, wird der Expanded-Speicher gewählt.

**Anwendungsbeispiele:**

1: device=Smartdrv.sys

Richtet einen Festplattenpuffer mit 256 KByte im Extended-Speicher ein.

2: device=Smartdrv.sys 384/A

Richtet einen Festplattenpuffer mit 384 KByte im Expanded-Speicher ein.

**C. Direkter Zugriff von DOS-Programmen auf den Erweiterungsspeicher**

Der Gerätetreiber HIMEM.SYS ermöglicht es, daß DOS-Programme direkt auf den Erweiterungsspeicher zuzugreifen. Dieser Treiber entspricht "XMS" (eXtended Memory Specification). HIMEM.SYS richtet, für DOS-Programme, zusätzlich einen 64 KByte großen Speicherbereich zur Code

und Datenausnutzung oberhalb der 1 MByte Grenze ein.

Im Erweiterungsspeicher werden z.B. Speicherresistente Programme ausgelagert, die sonst Hauptspeicher belegen (640 KByte).

Andere Erweiterungsspeicher-Programmen, die zum Verwalten des Erweiterungsspeichers nicht die XMS-Schnittstelle benutzen, sollten so konfiguriert werden, daß mindesten 64 KByte Erweiterungsspeicher frei bleiben.

**Voraussetzungen zur Verwendung von HIMEM.SYS :**

Voraussetzung ist ein IBM/AT oder kompatibler Rechner mit CPU 80286 oder 80386. Betriebssystem MS-DOS oder PC-DOS 4.0 oder höher.

Aufrufen von HIMEM.SYS in der CONFIG.SYS:

```
device=Himem.sys [/hmain=h]
[/ numhandles=n]
```

**Bedeutung der Parameter:**

/hmain=h :  
Diese Option legt den minimalen Speicherplatz fest den ein speicherresistentes Programm im HMA verwendet. (High-Memory -Bereich) Minimalwert für h = 0 KByte, Maximalwert Byte. Der Standardwert 0 ermöglicht einen "first come, first serve" Zugriff auf den Erweiterungsspeicher.

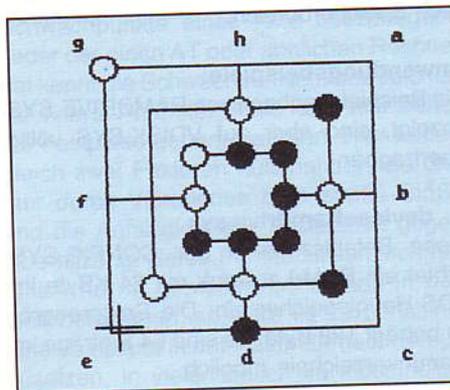
/numhandles :  
Diese Option wird auf die maximale Anzahl der Erweiterungsspeicherblock-Behandlungsroutinen festgelegt, die zu irgendeiner Zeit verwendet werden können. Minimum ist 1 KByte, Maximalwert ist 128 KByte, Standardwert ist 32 KByte.

Klaus Rumrich

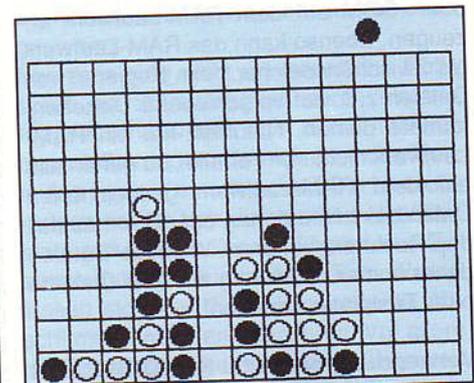
# Für die ganze Familie!

Wurde der Computer bisher nur von einem oder zwei Familienmitgliedern benutzt, so soll sich dieser Umstand jetzt gewaltig ändern. Klaus Rumrich, der Altmeister der Programmierung von Strategiespielen entwickelte wieder drei sehr populäre Spiele für den NDR Computer. Wer bereits in den Spielgenuß von seinem Schachspiel gekommen ist, erahnt sicherlich schon die Qualität der nun an dieser Stelle vorzustellenden Spiele.

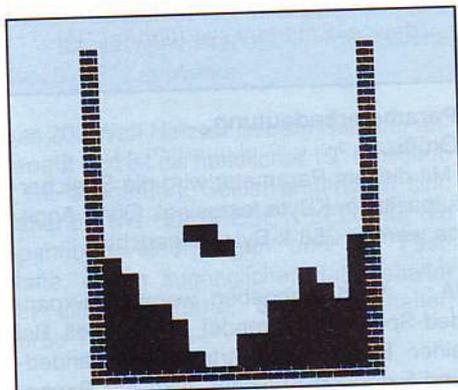
Bei zwei dieser Spiele handelt es sich um sehr bekannte Brettspiele, dem Spiel MÜHLE bzw. dem Denkspiel 4GEWINNT. Das dritte Spiel TETRIS ist ein Geschicklichkeitsspiel, das einigen vielleicht schon von anderen Computern her bekannt ist. Die Programme sind auf allen CPU's der 68000er Serie unter dem Betriebssystem JADOS lauffähig. Der zu benötigende Speicher begrenzt sich auf 32Kbyte. Die Programme benötigen keine BANKBOOT



Hardcopy Muehle 1.2



Hardcopy 4 Gewinnt



Hardcopy Tetris

Baugruppe. Die Programme sind alle mit der Tastatur zu bedienen. Folgende Baugruppen werden zusätzlich unterstützt.

**Unterstützte Baugruppen**

Prog.	COL256	MAUS	SOUND
TETRIS	X		X
4GEWINNT		X	X
MÜHLE		X	X

Um Ihnen einen ungefähren Überblick über den Verlauf der Spiele zu verschaffen, nun eine Kurzerklärung.

**MÜHLE:**

Wer 3 Steine nebeneinander auf die 3 Punkte einer Linie setzen kann, hat eine Mühle geschlossen und darf einen gegnerischen Stein schlagen. Jeder Spieler versucht, dem Gegner die Steine wegzunehmen oder sie so einzuschließen, daß sie nicht mehr ziehen können.

**TETRIS:**

Bei diesem Spiel wird ein Behälter auf dem Bildschirm angezeigt, in den geometrische Figuren fallen. Die Aufgabe des Spielers ist es, möglichst viele dieser Figuren in den Topf unterzubringen und eine waagrechte Reihe im Topf durchgehend zu füllen. In Falle eines Erfolges wird die Zeile gelöscht

und es entsteht wieder Platz für eine neue.

**4GEWINNT:**

Auf dem Bildschirm erscheint ein gerastertes Rechteck mit unterschiedlichen Spalten und Zeilen. Der am Zug befindliche Spieler muß nun einen seiner Spielsteine in eine beliebige Stelle des Feldes fallen las-

sen. Wer zuerst mindestens vier Steine der eigenen Farbe in einer Reihe zusammen hat, ist der Gewinner. Erlaubt sind waagerechte, senkrechte oder diagonale Reihen.

Natürlich ist bei jedem Spiel die Spielstärke wählbar. Dadurch ist es möglich sein Können dem eigenen Spiel anzupassen.

R.Pawlowitz

# SCSI Festplatten für den NDR

**Was heißt SCSI**

Die Abkürzung SCSI steht für 'Small Computer Systems Interface'. Diese Computer-Schnittstelle wurde allgemein als peripherie Schnittstelle definiert. Sie dient zum Beispiel als Kommunikationspunkt für Drucker, Festplatten, Scanner oder sogar einem weiteren Computersystem.

**SCSI Selbstständigkeit**

Ein SCSI Interface ist nicht nur ein blinder Befehlsempfänger, sondern es ist mit einer eigenen Intelligenz ausgerüstet. Ein eigener Prozessor sorgt in Verbindung mit einem Monitorprogramm und dem dazu ebenfalls erforderlichen RAM dafür, daß in der eigentlichen Peripherie Vorgänge ablaufen können, die den Hauptrechner nicht zu interessieren brauchen. Man könnte fast sagen, das sich hier ein eigener Rechner in dem SCSI Interface breit macht, der für sich selbständig Abläufe erledigt, die erheblich zu Entlastung des Steuerrechners beitragen. Dadurch ist auch die hohe Übertragungsrates zu erklären. Theoretisch ist durch diese Lösung eine Übertragungsrates von 2.5 Mbyte pro Sekunde zu erreichen.

Würde der Hauptrechner beispielsweise einen Sektor auf der Festplatte lesen wollen, so stellt er einfach über den SCSI Anschluß eine Anforderung dazu, das komplette Handling und deren Ansteuerung würde hierbei dann über die SCSI Steuerungseinheit laufen.

Der Befehlssatz des Interfaces ist genormt, daraus resultiert eine hohe Kompatibilität zwischen den verschiedenen Peripheriegeräten. Im Bezug auf Festplatten ist so also ein reibungsloser Austausch zwischen den einzelnen Typen gewährleistet. Genauso wie beim Befehlssatz sind auch die elektrischen Signale bzw. die Belegung der Anschluß Pins zueinander kompatibel.

Während die Z80iger sich noch mit Eprom-Programmen vergnügen, rufen die 68000'er Freak's nach mehr Massenspeicher für ihre Systeme. Ganz zu schweigen von den MS/DOS'lern die ohne Festspeicher schnell zum Disk-Jockey werden. Aus diesem Grund war eine Festplattenlösung notwendig, die allen Anwendern gerecht wird und zudem auch noch auf allen Systemen des NDR Computers lauffähig ist. Wir entschieden uns für die SCSI Schnittstelle

Der SCSI-Anschluß verfügt über einen 8 Bit breiten Parallelbus mit Paritätsprüfung. Er umfaßt insgesamt 19 Signale, die wie folgt zusammengesetzt sind.

Signale des SCSI Buses:

- 8 Datenbits
- 1 Paritätsbit
- 9 Kontrollsignale
- 1 Masse

Die Verbindung zwischen dem eigentlichen Hauptrechner und dem SCSI Interface wird über eine 2\*25 polige Steckerleiste abgewickelt. Dieser Anschluß ist genormt.

**Pinbelegung des SCSI Anschlusses**

Physikalisch ist der SCSI Bus nicht etwa

Signal	Pin Num.	Signal	Pin Num.
-DB(0)	2	Ground	28
-DB(1)	4	Ground	30
-DB(2)	6	-ATN	32
-DB(3)	8	Ground	34
-DB(4)	10	-BSY	36
-DB(5)	12	-ACK	38
-DB(6)	14	-RST	40
-DB(7)	16	-MSG	42
-DB(P)	18	-SEL	44
Ground	20	-C/D	46
Ground	22	-REQ	48
Ground	24	-I/O	50
Terminator Power	26		

eine weitere Baugruppe, sondern ist bereits fester Bestandteile der Festplatte. Eine SCSI Festplatte, unterscheidet sich aus diesem Grund von den mechanischen Aufbau Maßen nicht von einer herkömmlichen

festplatte mit ST506 Anschluß.

**SCSI am NDR Computer**

Die eben genannten Vorteile des SCSI Interfaces waren für uns Grund genug, um diese Möglichkeit auch am NDR Computer durch-

zusetzen. Von der Hardware-Seite waren bereits alle Voraussetzungen gegeben. Die SASI Schnittstelle, die im Grunde nichts anderes als der Vorgänger der SCSI Schnittstelle war, existierte ja bereits. Sie dient als direkte Verbindung zwischen dem NDR Bus und dem SCSI Bus. Nun mußte eine Software Lösung her.

**Wie stehts mit der Software?**

Der Vorreiter war Volker Wiegand, er war der erste, der eine SCSI Festplatte unter OS/9 an den NDR Computer anpaßte.

Nach ihm folgte Ralph Dombrowski, der den Anschluß in der nächsten Version des 68000'er Grundprogrammes integriert. Auch Klaus Janßen steht kurz vor der Fertigstellung seines Betriebssystems JADOS, daß bald auch mit Festplatte werkeln wird. Die CPU8088 bleibt ebenfalls nicht von dieser Anpassung unberührt. Rolf Dieter Klein hat, nachdem der OMTI Controller nur schwer zu beschaffen war, sein BIOS demnach ebenfalls hingehend geändert. Zu guter Letzt werden auch die Z80 Anwender den Spaß an einer SCSI Festplatte nicht missen müssen. Herr Ehrenberger hat sich die Aufgabe gemacht, das BIOS auf diese Konfiguration anzupassen.

**Wann wird es machbar sein?**

Die SCSI Festplattenlösung ist seit Anfang des Jahres unter OS/9 bereits lieferbar. Auch die CPU8088 Anwender können jetzt mit der neuen Version 1.9-01 des RDK BIOS Programmes SCSI Festplatten anschließen. Benutzer anderer Konfigurationen müssen sich noch etwas gedulden.

Nikolaus Bischof

# Änderungen KEY3 r2

## 1. Das READY-Signal

Das READY-Signal auf ASCII-Tastaturen wird dazu benutzt um festzustellen, ob der Rechner bereit ist von der Tastatur ein Zeichen zu empfangen. Da dieses Signal bei der KEY nicht verwendet wird (die KEY ist immer bereit) wird dieses Signal fest auf Masse gelegt. Bei älteren Tastaturen liegt dieses Signal oft schon intern auf Masse. In einem solchen Fall muß die KEY3 nicht geändert werden. Sollte dies aber nicht der Fall sein, muß Pin 12 von ST1 auf Masse gelegt werden.

## 2. Umschaltung positiver und negativer Strobe

Beim NDR-Computer wurden von Anfang an Tastaturen mit positivem STROBE-Signal eingesetzt. Sollten Sie aber eine Tastatur mit negativem STROBE haben, müssen Sie den in der Beschreibung erwähnten JMP1 umlegen. Dies ist allerdings bei der neuen KEY3 nicht möglich, da das OPEN COLLEKTOR Gatter J3C ständig ein LOW-Signal ausgibt, und damit ständig ein Zeichen erkannt wird. Um dies zu umgehen, müssen Sie den Ausgang des Gatters (J3C) vom INT des Prozessors trennen. Am einfachsten ist dies zu erreichen, indem man Pin 6 des J3 hochbiegt. (J3 aus der Fassung nehmen, Pin 6 hochbiegen, J3 wieder einstecken). Damit funktionieren auch ASCII-Tastaturen mit negativem STROBE an der KEY3.

Wenn Sie später allerdings eine PC-Tastatur oder eine ASCII-Tastatur mit negativem STROBE anschließen wollen, dürfen Sie nicht vergessen, diesen Pin wieder in die Fassung zu stecken.

## 3. Verwendung von 32k x 8 Speicher (RAMs und EPROMs)

Die KEY3 wurde so konzipiert, daß problemlos 8k x 8 organisierte Speicher verwendet werden können als auch 32k x 8. Doch bei der Umsetzung der Theorie in die Praxis schliefen sich hier zwei Fehler ein. Der an den Speichern umschaltbare Pin, der auf Jumper geführt wurde, ist nicht der mittlere Pin dieses Jumpers, sondern der rechte. Damit ist die Einstellung für 8k x 8 (Default-Einstellung) problemlos möglich, aber die Einstellung von 32k x 8 EPROMs erfordert eine Drahtbrücke (siehe Skizze).

**Die erste Serie der KEY3 Baugruppe weist noch kleine Probleme im Bezug auf die ASCII-Tastatur auf. Auch im Handbuch haben sich kleine Fehler eingeschlichen, die sich wohl bei einer Null-Serie nie ganz vermeiden lassen. Diese kleinen Fehler können aber problemlos behoben werden.**

## 4. Prellen der Tasten bei der Maus

Die KEY3 beinhaltet auch die Maus-Schnittstelle. Sind die Tasten der Maus nicht gedrückt, hängen die Eingänge für die Tasten an der KEY3 in der Luft. Bedingt durch das doch relativ lange Mauskabel und einer eventuell HF-verseuchten Umgebung kann es passieren, daß sich das hier LOW-Signale auftreten, obwohl keine Taste gedrückt wurde. Dieser Effekt kann durch zwei Pull-Up-Widerstände von 1k gegen +5V beseitigt werden (siehe Skizze).

## 5. Verlängerung des Reset-Impulses

Beim Betrieb der KEY3-Baugruppe ist ab und zu ein kleines Problem aufgetaucht. Beim Einschalten oder bei einem Reset 'kommt die Tastatur manchmal nicht hoch'. Dies liegt daran, daß der Reset-Impuls von der jeweiligen CPU für die KEY3 zu kurz ist. Falls bei Ihnen dieses Problem auch auftritt, können sie dies folgendermaßen beheben: Sie müssen nur die Zeitkonstante des R-C-Glieds, das den Reset auslöst, verlängern.

Dies erreichen Sie am besten, indem Sie auf der CPU-Karte, parallel zu dem Kondensator im Reset-RC-Glied, nochmal den gleichen Kondensator einlöten.

### Bei der CPU68000

Parallel zu C1 (10µF) noch einen Kondensator mit 10µF einlöten.  
Oder statt R7 (100 kOhm) einen 200 kOhm Widerstand einlöten.

### Bei der CPU68K

Parallel zu C1 (10µF) noch einen Kondensator mit 10 µF einlöten.  
Oder statt R8 (100 kOhm) einen 200 kOhm Widerstand einlöten.

### Bei der CPU Z80.

Parallel zu C2 (10µF) noch einen Kondensator mit 10 µF einlöten.  
Oder statt R6 (10 kOhm) einen 100 kOhm Widerstand einlöten.

### Bei der SBC3

Parallel zu C1 (10µF) noch einen Kondensator mit 10 µF einlöten. Oder statt R10 (10 kOhm) einen 100 kOhm Widerstand einlöten.

## 6. Probleme mit dem Anschwingen des Quarzes

Wie bei Karten mit eigener CLK-Erzeugung schon be-

kannt, kann es passieren, daß der Quarz nicht richtig anschwingt. Vor allem im Zusammenhang mit nicht getakteten Netzteilen war dieser Fehler nachvollziehbar. In der Regel hilft hier ein Austausch des 7404 gegen einen 74LS04 (J13).

## 7. Verbesserungen zum Handbuch

Im Handbuch haben sich bezüglich der Jumperstellungen und der Datenblätter einige Fehler eingeschlichen:

- Auf Seite 24 wird die Einstellung von JMP1 beschrieben. Dabei ist die Stellung 1 für ASCII-Tastaturen mit negativem STROBE-Signal. In Stellung 2 können ASCII-Tastaturen mit positivem STROBE-Signal angeschlossen werden (TAST1, TAST2 und TAST3). Für eine PC-Tastatur wird wie beschrieben Stellung 3 verwendet.

- Auf Seite 25 wird die Einstellung von JMP2 und JMP3 beschrieben. Auch hier haben sich Fehler eingeschlichen. Die richtige Jumperstellung ist wie folgt. Die Voreinstellung ist für die ausgelieferte Version (8k x 8 EPROM und RAM) richtig. Auf Seite 24 wurde im Text dem JMP3 die RAM- und JMP2 die EPROM-Konfiguration zugeordnet. Richtig ist aber JMP2 für RAM und JMP3 für EPROM.

- Auf Seite 38 fehlt ein Stück Text zum Punkt 9.6. Der fehlende Text wird hier abgedruckt:

### "9.6 ASCII-Tastatur

Bei Betrieb mit der ASCII-Tastatur muß beim Einschalten des Rechners die SPACE-Taste gedrückt werden, sonst wird die ASCII Tastatur nicht erkannt. Ein Tastaturpuffer für die ASCII-Tastatur ist noch nicht vorhanden."

- Auf Seite 51 bis 59 wurde das falsche Datenblatt abgedruckt. Richtig müßte hier natürlich das Datenblatt der Z80 CPU stehen. Bei den ausgelieferten Fertigergeräten wurden diese Punkte natürlich berücksichtigt, ebenso natürlich bei den nach dem 15.09.1989 ausgelieferten Bausätze.



# Neue Produkte - Neue Preise

Best-Nr.	Preis DM	Best-Nr.	Preis DM
11358	BIOSUPDATE8088 BIOS Update f. CPU8088 Jetzt mit SCSI-Anpassung	11473	SPIELEJA58 Spiele Disk f. JADOS 51/4
11474	BIOS8088Q58 BIOS Quelle Disk 51/4	11472	SPIELEJA38 Spiele Disk f. JADOS 31/2
11475	BIOS8088Q38 BIOS Quelle Disk 31/2		
11363	UTIL8088DISK58 Utility Disk f. CPU8088 51/4 Jetzt mit SCSI Formatierer		
11364	UTIL8088DISK38 gleich wie 11363, nur 31/2		

Bitte  
Porto  
nicht  
vergessen

**ANTWORT**

**GRAF  
computer**

Graf Elektronik Systeme GmbH  
Postfach 1610

**8960 Kempten**

Bitte  
Porto  
nicht  
vergessen

**ANTWORT**

**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 \_\_\_\_\_