

8. Jahrgang

Heft 1

Erscheint
5mal /Jahr

PC-NEWS

Lfd.Nr. 30

Jänner 1993

Preis
öS 40,-*Mitteilungen aus den Höheren Technischen Lehranstalten für aktive PC-Anwender*

PCC-TGM	Personal Computer Club - Technologisches Gewerbe-Museum
MCCA	BTX und Micro Computer Club Austria
ADIM	Arbeitsgemeinschaft für Didaktik, Informatik und Mikroelektronik

*Grafik aus dem PC-NEWS-Sonderdruck Kunst, Intelligenz und Computer von Dr. Pfliegerl.*

BTX-Angebot	⇒S.3	Kennung+Dok+Programm um S 360,-
Clubabend	⇒S.7	3. März Thema: LOTUS-Software
Preisausschreiben	⇒S.8	Programme für den Unterricht 3 Originalpakete zu gewinnen
fuzzy	⇒S.64	fuzzy wird 'schulreif' durch Schulversionen
Sprachpolizei	⇒S.13,46	Schulen Sie Ihr Sprachgefühl PASCAL (leicht), C (schwerer)
Bestellungen	⇒	Laserdrucker(S.75), 386DX(77,86), 486DX(77), Schulsoftware(75), Compaq(90), SelbstbauPC(72)

Fax - Antwort



(0222) 513 20 13

An
 Post-Generaldirektion
 BTX-Marketing
 Postgasse 8
 1011 Wien

Ja, ich bestelle.....Stück der Kommunikations-Software Decodix 2.0 und meine persönliche Zugangs-Kennung zum BTX-System um **nur öS 360,-**.

Die Kommunikations-Software *Decodix 2.0* + persönliche Zugangs-Kennung senden Sie an diese Adresse:

Für alle Fragen:
 BTX-Hotline 0660 212

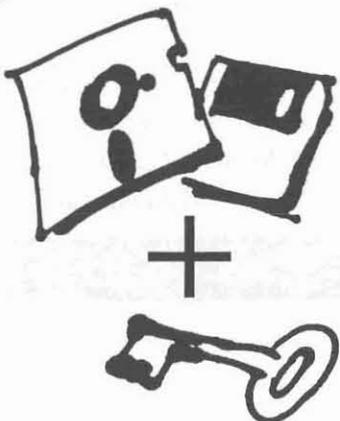
 Name, Firma

 Anschrift

 PLZ Ort

 Telefon (für allf. Rückfragen)

 Unterschrift bzw. firmenmäßige Zeichnung



Liebe Leser

Windows-Ecke

Eine ständige Einrichtung soll die Windows-Ecke werden, in der Wissenswertes über die Arbeit mit Windows-Programmen berichtet wird. In diesem Zusammenhang sammeln wir auch Beiträge für die nächste Ausgabe der PC-NEWS, mit Windows als Schwerpunkt.

fuzzy

wird eine Technik für jedermann. Konkret können Sie sich in einem Beitrag von Herrn Krebs darüber informieren aber auch gleichzeitig für den Unterricht die dazugehörige Hard- und Software bestellen. Als Vorinformation gibt's ein anschauliches Demoprogramm das zur Demonstration im Unterricht unmittelbar einsetzbar ist.

Neue Betriebssysteme

werfen ihre B-Schatten voraus: einerseits **WindowsNT**, das auch eine ganz normale DOS-Betriebsart hat, wie man aus der Liste der Kommandozeilenparameter entnehmen kann und **DOS 6.0**, für Sie getestet im Rahmen eines Referats durch Herrn Zetik, mit einem Auzug aus der Originaldokumentation, die Hinweise auf Neuerungen enthält.

Programmierung,

der wir uns als EDV-Lehrer verpflichtet fühlen, bildet auch diesmal wieder den Hauptteil unserer Ausgabe. Wenn Sie sich weniger für das Wie, sondern mehr für die praktische Seite interessieren, können Sie einfache Testprogramme für die parallele und serielle Schnittstelle und auch für das CMOSRAM aus dem Abschnitt Hardwareprogrammierung entnehmen. Prüfstecker dazu sind einfach zu fertigen. Bei Bedarf, bitte bei der Redaktion anfragen.

PCAD-Zuckerln

Mit der heutigen Ausgabe beginnt eine erfolgversprechende Zusammenarbeit von Rekirsch-Elektronik, dem Vertreter von PCAD in Österreich und der Redaktion der PC-NEWS. Alle Informationen über PCAD, die normalerweise der jeweils lizenzverwaltende Lehrer erhält, werden über die PC-NEWS auch allen interessierten Lehrern und Schülern zugänglich gemacht. Heute hat Ing. Wiesauer von Rekirsch-Elektronik Informationen zu PCAD 6.0 zusammengefaßt. Das Besondere dieses Beitrags aber ist die frei kopierbare Version von PCAD 6.0 für Schüler und Lehrer außerhalb des Schularbeitsplatzes für das Komplettpaket mit einer einzigen Einschränkung: Es können nur 35 Bauteile verwendet werden. Eine neue Vorgabe für den FTKL-Unterricht? Als Zusatzliteratur kann mit dem 'Update-Trainings-Handbuch' (LIT: 56) der Umstieg auf die neue Version vereinfacht werden.

Neue Gesetze

Neues Fernmeldegesetz und neues Urheberrechtsgesetz betreffen uns alle. Zum Ersteren finden Sie in diesem Heft eine erste Stellungnahme von kompetenter Stelle. Für eines der nächsten Hefte ist für Zweiteres ein Kommentar in Vorbereitung.

BTX-Weihnachtspost

Die nächsten Weihnachten kommen bestimmt, und damit das Schreiben der Weihnachtskarten etwas erfreulicher für Sie wird,

können Sie im Beitrag 'BTX-taugliche Adressen aus dBase-Datei' sehen, wie via BTX und Christkindl Ihre Adressaten 'die etwas andere Weihnachtskarte' bekommen können. Zeit zum Üben wie's gemacht wird gibt's genug, ein besonderes Angebot dazu von der Post eine Seite vorher. *Ran' wie nix, ans BTX.*

PC-NEWS-29 Leserumfrage

Es sind 28 Antworten eingelangt, die Ergebnisse sind wegen der besseren Lesbarkeit in Prozenten angegeben. Zuerst die allgemeine Frage über die zukünftige Gestaltung der PC-NEWS:

Bewertung (Angaben in Prozent)	ja	nein	?
PC-NEWS zu wenig anwendungsbezogen	15	81	4
Ohne Geld Teilung PC-NEWS/Club-NEWS	74	26	0
Abo-Interessent	78	19	4

Jetzt die konkrete Auswertung über die einzelnen Beiträge, wobei: 1=sehr gut, 3=brauchbar, 5=unbrauchbar, keine Antwort=kein Interesse.

Bewertung (Angaben in Prozent)	1	3	5	?
CLUBTEIL				
Liebe Clubmitglieder!	63	30	0	7
VERZEICHNISSE				
Verzeichnisse LIT,TAB,DSK	59	37	0	4
Inhalt PC-NEWS, chronologisch	33	33	19	15
PREISLISTEN				
Sonderpreisliste CompDelphin	52	37	7	4
Sonderpreisliste nds	56	33	4	7
Sonderpreisliste excon	56	33	4	7
BESTELLSCHNEI				
Antwortkarte PC-NEWS	74	19	0	7
Bestellschein ADIM	52	33	4	11
Anmeldeformular PCC-TGM	44	37	4	15
Bestellscheine PCC-TGM	67	19	0	14
SAMMELBESTELLUNGEN				
Sammelbestellung excon 386/486	41	26	15	18
Sammelbestellung CompDelphin Software	44	30	11	15
Sammelbestellung CompDelphin 386SX	30	41	11	18
BEITRAGSTEIL				
CD-ROM				
CD: Educational-Master	41	41	7	11
DFÜ				
Zwei-Draht-Modems und Nebenstellenanlagen	59	22	4	15
Modem-Aktion	48	30	7	15
Neues rund um BTX	56	30	4	10
Telesoftwarekonvertierung für PC	33	30	15	22
C E P T - Codes für B T X	30	41	7	22
VIDEO, PC, TV und Video	63	26	0	11
MIKROELEKTRONIK				
PCCARDS: PROBLEME-LÖSUNGEN	19	22	11	48
Layouts mit PCCARDS	19	19	15	49
DIGITALE SCHALTWERKE MIT EPROM	22	26	7	45
KURZFASSUNGEN				
Optionen von BORLAND-C, V 3.0	41	41	0	18
PROGRAMMIERUNG				
VGA-Bildschirm umechalten	15	48	4	33
Typematic-Rate am AT einstellen	19	59	0	22
Auf den Spuren von LapLink	33	44	0	23
TESTBERICHT				
Power-C ein C-Kompilier zum Buchpreis	41	26	0	33
Windows NT	52	37	0	11
TESTPROGRAMME				
CACHE, das wahre Wunder?	44	52	4	0
Video-Performance	41	37	7	15
UNTERRICHTSSOFTWARE				
Small is useful!	59	37	0	4
Anwendungen				
COMPUTERVIREN	74	26	0	0
MESSWERTEERFASSUNG				
Die PCL-812 LAB-Karte	22	22	4	52
Laborübung ADC	26	15	7	52
ADC-DAC-Test	22	19	7	52
Digitalspeicheroszilloskop	26	19	4	51

Hervorzuheben ist, daß der Beitrag mit dem größten positiven Echo, jener über Computerviren, aus einem Referat eines Abendschülers hervorgegangen ist.

Wir wissen jetzt zwar, was der 'harte' Kern des PCC über unsere letzte Ausgabe denkt; eine Hochrechnung auf die restlichen Mitglieder ist aber sicher nicht so ohne weiteres möglich. Wir danken fürs Mitmachen den Herren Anderwald, Beigl, Böhm, Breidler, Dorneger, Fresacher, Fröhlich, Grasel, Hajny, Hasenberger, Kanka, Ludwig, Muth, Neufingerl, Obdrzalek, Reinisch, Sauer, Schaffer, Scheiber, Schleining, Schludermann, Schrieffl, Skriwanek, Sokol, Streisselberger, Thomas, Tradinik.

EDU-CD

Die EDU-Testkopieraktion der Redaktion war ein Erfolg, der PCC-TGM war mittlerweile auch nicht untätig, Sie können eine Reihe weiterer Public-Domain-Sammlungen nach brauchbarem durchsuchen, siehe Seite 9. Beachten Sie, daß es zu der EDU-CD noch eine Kurzdokumentation gibt, die man über das Diskettenservice (DSK-269) oder über die Mailbox beziehen kann.

PC-NEWS-29 Nachtrag

Auf Grund einiger Nachfragen über die Bezugsmöglichkeiten der PC-LAB-Karte, baten wir die Lieferfirma, um eine Preisliste für diese Produkte, leider ohne Erfolg (Überlastung). Firma Hummer sprang in die Bresche. Auf Seite 75 ist ein Auszug aus einer Preisliste für diese Produkte. Die PC-LAB-Karte hat die Bestellnummer 6105. Ähnliche Eigenschaften aber etwas langsamer bei der Umsetzung ist die PC-LAB-711 S (6104).

PC-NEWS-29 Fehlerberichtigung

Bei einigen Beiträgen war irrtümlich eine falsche Diskettennummer angegeben. 234 statt richtig 269. Welche Nummer stimmt, kann man aber durch das Dateienverzeichnis auf Seite 80 entschlüsseln. Auf Seite 80 war die letzte Diskette (Titelseiten der PC-NEWS) mit TGM-DSK-271 numeriert. Richtig ist die Nummer 274.

VORSCHAU auf PC-NEWS-31

- Hardwarenahe-Programmierung Teil 5, (Direkter Speicherzugriff)
- Installation eines Points für das FIDO-Net
- Ein Menüprogramm für Laborübungen
- XILINX (3.Teil)

Es ist geplant, neben den laufenden Beiträgen folgende Schwerpunktthemen in den PC-NEWS zu besprechen:

- Windows (März 1993)
- Schülerprogramme (Mai 1993+Ergebnis Preisrätsel)
- Mikrokontroller-Programmierung (September 1993)
- Menüs und andere Gerichte (November 1993)
- BORLAND-C Application-Frameworks

Wenn Sie zu diesen und natürlich auch zu anderen Themen beitragen können, ersuchen wir um Zusendung von ASCII-Texten auf Diskette.

Franz Fiala

Liebe Mitglieder des MCCA!

Wir vom MCCA führen eine Datenbank für unsere Mitglieder und Interessenten. Wir kennen daher Ihre Interessensgebiete im BTX-System und die Geräte auf denen Sie mit BTX arbeiten.

So ist es uns möglich, gezielt Informationen zu verteilen. Für die PC-Besitzer steht unter anderem der Bezug dieser PC-NEWS zu einem ABO-Preis von ÖS 50,- für 5 Ausgaben zur Verfügung.

Die Differenz der Kosten für Druckbeitrag der PC-NEWS und dem Versand trägt der MCCA.

Um die neuen bzw. geänderten Datenbanksätze den anderen Vorstandsmitgliedern zu übermitteln, bedienen wir uns einer TICK-AREA im Fidonet, wo alle als Points eingetragen sind. Ein Update-Programm auf dem PC bei den Vorstandsmitgliedern erstellt ein vollkommenes Spiegelbild der Datenbank des Zentralen-PC's.

Um nun auch dem Leiter der MCCA-Mupidgruppe, Peter Marschat, diese Updates zu senden, wird eine .sdf-Datei erzeugt. Diese wird mit lharc 1.13 gepackt und mit diesem Programm vollautomatisch als TSW-Download für Mupid in eine BTX-Seite des MCCA eingespielt.

Unter CP/M auf Mupid mit Diskettenstation ist es dann möglich, eine dBase II Datenbank upzudaten.

Durch die Zusammenarbeit mit Johannes Sautner sind viele kleine und nützliche Deluxe-Batchfiles entstanden, die die Automatisierung der GBG-Aufnahme in, beziehungsweise Ausschluß aus der Datenbank, wesentlich erleichtern.

Logfiles ermöglichen dem Anbieter, in seiner GBG den Überblick seiner Mitglieder zu behalten. Dies ist besonders wichtig, wenn über GBG's vertrauliche Mitteilungen in geschützten Seiten oder TSW (z.B. WinWord-Dokumente) verteilt werden.

In den nächsten PC-NEWS berichte ich über weitere Deluxe-Programme und Batches für den Mitteilungsdienst (Eindeutige Abspeicherung mit Datum und Suffix, Vergleich auf Datum kleiner Tagesdatum.. und automatische Löschung aus BTX) und der automatischen Übernahme von Textteilen aus Antwortseiten in die Datenbank (Auswertung von Umfragen im BTX).

Sollten Sie Interesse an diesen Programmen und Batchfiles haben, schreiben uns auf Seite *255082# im BTX oder in der BTX.AUS ECHOAREA im Fidonet.

Helmuth Schlögl, MCCA

Liebe Mitglieder des PCC-TGM!

SOFTWARE-AKTION durch CompDelphin

Schon lange gab es bei einer Sammelbestellaktion kein solches Echo. Wir fühlten uns (fast) an den Beginn vor 7 Jahren zurückversetzt; mehr als 100 Bestellungen langten bei uns ein. Wenn Sie zu jenen zählen, die die Programme dann doch erhalten haben, herzlichen Glückwunsch, denn es kam leider auch zu Unzulänglichkeiten. Wir erhielten zahlreiche Beschwerden, daß bestellte Software nicht ausgeliefert wurde. Dafür waren drei Gründe verantwortlich:

1. Im Zeitraum Oktober haben uns einige Briefsendungen nicht erreicht. Die Ursachen dafür konnten wir jedoch, da es sich ja um "gewöhnliche" Briefsendungen gehandelt hatte, leider nicht mehr feststellen. Bestellungen (übrigens auch von Hardware) konnten wir daher nicht weiterleiten.
2. Firma COMP DELPHIN hatte auf Grund des großen Interesses Beschaffungsschwierigkeiten, so konnte (oder wollte!) z.B. Microsoft lange Zeit nicht liefern.
3. Aus **Verschulden der Firma COMP DELPHIN** ist die Auslieferung der Software bzw. die Verständigung zum Abholen nicht erfolgt. Weil sich die Beschwerden bei uns häuften, baten wir daher unsere Mitglieder, direkt mit COMP DELPHIN Kontakt aufzunehmen.

Wir bitten alle Betroffenen um Entschuldigung und haben, um in Zukunft ähnliche Mißstände zu vermeiden, folgende Regelung getroffen:

Für alle Bestellungen von Hard- und Software werden Ihnen in Hinkunft Auftragsbestätigungen von der jeweiligen Lieferfirma zugesandt, falls die Bestellung nicht sofort ausgeliefert werden kann. Sollten Sie daher spätestens zwei bis drei Wochen nach Ihrer Bestellung weder eine Verständigung zur Abholung der bestellten Waren, noch eine Auftragsbestätigung erhalten haben, ersuchen wir um telefonische oder schriftliche Verständigung des Clubs!

BÜROSTUNDEN

Unser Sekretariat ist in Hinkunft zu folgenden Zeiten erreichbar:

Dienstag von 14 Uhr bis 17 Uhr

Mittwoch von 19 Uhr bis 20 Uhr 30

Das Büro ist während der schulfreien Tage nicht besetzt!

Am Mittwoch den 20.1.1993 war das Sekretariat wegen Erkrankung leider nicht besetzt. Wir bitten um Entschuldigung!

SEMINARE

Alle Seminare im Schuljahr 1992/93 sind ausgebucht! Für den Herbst 1993 sind wieder Seminare geplant. Wenn Sie an einem Thema besonderes Interesse haben, verständigen Sie uns bitte. Wir werden versuchen, Ihre Wünsche zu erfüllen.

Robert Syrovatka

CLUBABEND

Der nächste Clubabend findet am 3. März 1993 wieder am TGM, 1200 Wien, Wexstraße 19-23 um 18:30 im Exnersaal statt. Spezialisten der Firma LOTUS stellen Software vor:

➤	AMI PRO 3.0	Top-Textverarbeitung
➤	LOTUS 123	Tabellenkalkulation
➤	LOTUS ORGANIZER	Adreß- und Terminverwaltung.

Veranstalter: *CompDelphin & LOTUS*

SOFTWARE FLOHMARKT

Suche PC-SW für den Ausdruck von Notenzeilen-Eingabe. Sollte möglichst einfach sein. Berhard Böhm, 1200 Wien, Hellwagstraße 4/2/31, Tel.: 332-01-96.

Video-Kurse zum Probieren: COREL-DRAW (60'), dBaseIV-Teil 1 Grundlagen (100'), dBaseIV-Teil 2 (70'), dBaseIV-Teil 3 Programmierung (100'), Windows 3.0 (70'), MSDOS 5.0 Grundlagen (35'), Page-Maker, Version 4.0 (180'), Ventura Publisher 3.0 für Windows (180'), Wenn-Funktion in MS-Excel (30'), Datenbankfunktion in MS-Excel 3.0, MS-Excel 3.0. Interessenten wenden sich bitte zwecks Vereinbarung eines gemeinsamen Lern-Termines an das Clubsekretariat.

HARDWARE FLOHMARKT

Verkaufe 2 Stück TI-Minidatenbank, bis zu 150 Einträge (je 6 Buchstaben, 4 Stellen Vorwahl und 8 Stellen Telefonnummer möglich), Password-Schutz, inkl. Taschenrechner, wenig gebraucht, je Stück öS 200,-. Mag. Klaus Scheiber, Gritzenweg 26, 8052 Graz.

Suche einen Gameport für einen XT, 4.88 Mhz. Suche einen **Druckertreiber für Harward Graphics 3.0** für einen Farbdrucker XEROX 4020. Andreas Schaffer Gersthoferstraße 110/2/3, 1180 Wien.

FRAGEN

Kann das **CD-ROM-Laufwerk** aus der Club-Aktion eine Kodak-Photo-CD lesen? (Wenn man alle 100 Bilder auf einmal draufspielt, dürfte es ja nur ein Inhaltsverzeichnis geben) Antworten an: Franz Streisselberger, Tel.: 408-53-29, BTX.: 912-218-709.

Wer hat ein **I²C-Interface für den COM-Port** ggf. mit Software? Wer hat einen **GAL-Programmer** mit Software? Antworten an BTX: 912-214-236 (913-110-525).

Verzeichnisse

TGM-DSK

- DSK-275: (4) CD-ROM-Verzeichnis PC-SIG, Disk 1(3)
 DSK-276: (4) CD-ROM-Verzeichnis PC-SIG, Disk 2(3)
 DSK-277: (4) CD-ROM-Verzeichnis PC-SIG, Disk 3(3)
 DSK-278: (4) CD-ROM-Verzeichnis MEGA-Win/OS/2 Disk 1(2)
 DSK-279: (4) CD-ROM-Verzeichnis MEGA-Win/OS/2 Disk 2(2)
 DSK-280: (4) CD-ROM-Verzeichnis CICA-Windows, November 1992 Disk 1(6)
 DSK-281: (4) CD-ROM-Verzeichnis CICA-Windows, November 1992 Disk 2(6)
 DSK-282: (4) CD-ROM-Verzeichnis CICA-Windows, November 1992 Disk 3(6)
 DSK-283: (4) CD-ROM-Verzeichnis CICA-Windows, November 1992 Disk 4(6)
 DSK-284: (4) CD-ROM-Verzeichnis CICA-Windows, November 1992 Disk 5(6)
 DSK-285: (4) CD-ROM-Verzeichnis CICA-Windows, November 1992 Disk 6(6)
 DSK-286: (4) CD-ROM-Verzeichnis ALDE, RBBS, SOUNDS, MS-BOOK
 DSK-287: (4) COSMIC Software Catalog, expandiert auf ca. 3MB, Verzeichnis im dBase-Format, enthält auch das Bestellformular im PCX-Format
 DSK-288: (4) Windows NT Applikationen, Verzeichnis Stand Dez.1992
 DSK-289: (4) AmiPro für Windows, Version 3, Demoversion, Disk 1(2) (LIT-60).
 DSK-290: (4) AmiPro für Windows, Version 3, Demoversion, Disk 2(2) (LIT-60).
 DSK-291: (2) PC-Professionell Utilities (Teil V) Aktueller Windows-Benchmarktest WINBENCH 2.50 aus den Ziff-Davis Labs; Neuester Landmark-Benchmark LMST-PCP und AT-Bus-Festplatten-Utility LM-IDE; Listings aus dem PC-Professionell-Assemblerkurs; Quelltext und lauffähige Version für einen neuen Bootsektor; Zeichensatzprogramm in TURBO-PASCAL aus PC-Professionell 8/92
 DSK-292: (2) PD-Diskettenverzeichnis PAB-Hard- & Software, 1050 Wien, Arbeitergasse 40, TEL:(0222)/550-608-0, FAX:(0222)/550-608-13.
 DSK-293: (2) 8051-Entwicklungspaket, Demoversion, KEIL, C51-Compiler V3.2, dScope V5.0.
 DSK-294: (2) PC-NEWS-30, Programme
 DSK-295: (4) Microsoft Journal 6/92
 WINMFC.LZH Dateien zu "Windows-Programmierung mit C++ und den Microsoft Foundation Classes (MFC)".
 WINCPP.LZH Dateien zu "Kleinere Quelltexte für Windows-Programme mit C++".
 GDI32.LZH Dateien zu "Erste Eindrücke vom 32-Bit-GDI in Windows NT".
 NETWIN.LZH Dateien zu "Windows 3.1 und NetBIOS".

- OPEN.LZH Dateien zu "Mehr Sicherheit beim Arbeiten mit MS-DOS-Dateien".Spezifikationen im DOC-Format (Microsoft Word für Windows):
 APM.LZH "Advanced Power Management 1.0".
 LSAPIL.LZH "License Service Application Programming Interface 1.0".
 MAPIL.LZH "Messaging Application Programming Interface (MAPI) 1.0".
 ODBC.LZH "Open Database Connectivity (ODBC)".
 WINSOCK.LZH "Windows Sockets. An Open Interface for Network Programming under Microsoft Windows".

DSK-296: (4) Microsoft Journal 1/93

- TOPTEN.LZH Dateien zu »Die Top Ten Programmertips für Windows«
 WINVER.LZH Dateien zu »Alles im Griff bei neuen Programmversionen und bei der Installation«
 VXD.LZH Dateien zu »Virtuelle 32-Bit-Gerätetreiber für Windows«
 CONTROL.LZH Dateien zu »Erweitern Sie die Systemsteuerung mit eigenen DLLs«
 GRAF1.LZH Dateien zu »Die große Welt der Grafikformate«
 FAC.LZH Dateien zu »Fragen und Antworten zu C«
 FADOS.LZH Dateien zu »Fragen und Antworten zu DOS«
 FAWIN.LZH Dateien zu »Fragen und Antworten zu Windows«
 ZINC.LZH Dateien zu »Zinc Interface Library«

DSK-297..DSK-300: diese Nummern werden in der nächsten Ausgabe vergeben.

- DSK-301: (4) PCAD-Version 6.0, max. 35 Bauteile Disk 1(13), (LIT-56)
 DSK-302: (4) PCAD-Version 6.0, max. 35 Bauteile Disk 2(13)
 DSK-303: (4) PCAD-Version 6.0, max. 35 Bauteile Disk 3(13)
 DSK-304: (4) PCAD-Version 6.0, max. 35 Bauteile Disk 4(13)
 DSK-305: (4) PCAD-Version 6.0, max. 35 Bauteile Disk 5(13)
 DSK-306: (4) PCAD-Version 6.0, max. 35 Bauteile Disk 6(13)
 DSK-307: (4) PCAD-Version 6.0, max. 35 Bauteile Disk 7(13)
 DSK-308: (4) PCAD-Version 6.0, max. 35 Bauteile Disk 8(13)
 DSK-309: (4) PCAD-Version 6.0, max. 35 Bauteile Disk 9(13)
 DSK-310: (4) PCAD-Version 6.0, max. 35 Bauteile Disk 10(13)
 DSK-311: (4) PCAD-Version 6.0, max. 35 Bauteile Disk 11(13)
 DSK-312: (4) PCAD-Version 6.0, max. 35 Bauteile Disk 12(13)
 DSK-313: (4) PCAD-Version 6.0, max. 35 Bauteile Disk 13(13)
 DSK-314: (4) ECAM-3.0/PCGERBER 7.0, Demoversion

TGM-LIT

- LIT-56: PCAD-6.0 Update Trainingshandbuch, 80 Seiten, englisch. Dieses Handbuch erleichtert das Kennenlernen der Veränderungen von PCAD-6.0 gegenüber der Vorgängerversion.
 LIT-57: Kurzanleitung zu PCAD, Wolfgang Scharl, Jänner 1991, 10 Seiten dt.
 LIT-58: Entwicklung elektronischer Leiterkarten mit PCAD, Rudolf König, 4/89, 102 Seiten, deutsch.
 LIT-59: fuzzy-Programmierwerkzeuge, INFORM, ergänzendes Informationsmaterial, Produktbeschreibung, 15 Seiten, dt.
 LIT-60: AMI-Pro-Demoversion, ergänzende Beschreibung, 9 Seiten, dt.

DISKETTENKOPIERDIENST

SHAREWARE hat in Österreich noch immer nicht den Stellenwert, den sie eigentlich verdienen würde. Noch immer werden diese meist kleinen Programmpakete, welche das Leben mit dem Computer erst so richtig lebenswert machen, schwer unterschätzt.

Schon vor einigen Jahren hat die Clubleitung die bekannteste Public-Domain Sammlung PC-SIG angekauft und sehr gute Erfahrungen damit gemacht. Daher wurde die neueste Version von PC-SIG und auch einige andere Programmsammlungen gekauft. Allein die PC-SIG enthält über 3000 Disketten auf CD. Dieses Volumen entzieht sich der Darstellbarkeit in unserer Zeitung, daher können Sie die genauen Verzeichnisse auf Disketten bestellen:

Verzeichnisse

DSK 275-277: PC-SIG Nummern 1 - 3018 & 3076

DSK 278,279: **Mega WinOS/2(A)** (August 92)
Diskette 1: WINDOWS.
Diskette 2: OS/2.

DSK 280-285: **CICA-Windows** (November 92)

DSK 286: **SHAREWARE GOLD, RBBS in a Box, ALDE, MS-BOOK, CREATIVE SOUND**

Alle Texte und Programme wurden noch komprimiert, um möglichst effizient auf den Disketten untergebracht werden zu können.

Mit diesem Angebot wollen wir den Mitgliedern ohne CD-ROM Laufwerk eine Übersicht geben aber auch die Möglichkeit bieten, auf die vorhandenen Programme zugreifen zu können. Für CD-ROM Besitzer ist dies eine Möglichkeit, sich vor dem Kauf einer CD von dem Inhalt zu überzeugen.

Preise

PC-SIG 5 1/4" Disketten (DS/DD) 15,--
(mit einem Programm auf Original-PC-SIG-Disketten solange der Vorrat reicht)
3 1/2" Disketten (DS/DD) 50,-- (mit zwei Programmen)

Alle anderen 5 1/4"/3-1/2" Disketten (DS/DD) 40,-/50,-

Bei Bestellung von Programmen von CD's geben Sie bitte die CD, den PFAD und den PROGRAMMNAMEN vollständig an (Bei PC-SIG bitte NUR DIE NUMMER!). Die an Sie gelieferten Disketten werden so weit wie möglich mit Daten gefüllt! Geben Sie nur an, ob sie 5 1/4" oder 3 1/2" Disketten (PC-SIG DS/DD sonst HD) bevorzugen.

Helmut Schluderbacher



Programme für den Unterricht

Gesucht werden **Programme für den Unterricht**. Die Verfasser der drei besten Einsendungen werden durch eine Jury bestimmt und erhalten ein Original-Paket LOTUS-Software. Die drei Bestplatzierten erhalten eines der folgenden Programme, wobei der Sieger zuerst, dann der Zweitplatzierte auswählt

- **Freelance Graphics**, LOTUS, Version 1.0 für Windows
- **Ami Pro**, LOTUS, Version 2.0 für Windows, Textverarbeitung
- **Lotus 1-2-3**, Version 1.1 für Windows, Tabellenkalkulation

Alle Einsender erhalten eine

- **Diskette mit True-Type-Fonts** für Windows.

Bitte senden Sie Ihr Programm mit Dokumentation auf Diskette, Fachgebiet egal, bis spätestens 12. März an die Redaktion der PC-NEWS, Siccardsburggasse 4/1/22, 1100 Wien. Legen Sie uns eine Autoren-Kurzbeschreibung bei (Beispiele siehe hinteres Deckblatt). Eine Auswahl der Einsendungen werden wir in der Juni-Ausgabe abdrucken.

Die Preise wurden uns von einem engagierten Mitglied, Herrn Schlager, Tirol zur Verfügung gestellt, wofür wir ebenso, wie für viele seiner anderen Dienste an dieser Stelle danken.

Machen Sie mit!

PASCAL - C - Vokabelheft

F. Fiala, W. Riemer, N, TGM

Die folgende Tabelle könnte zur Annahme verleiten, es ginge bei einem Sprachwechsel von PASCAL zu C nur darum, einige Begriffe zu tauschen. Eine Sprache ist nicht durch ein Vokabelheft allein erklärbar. Näheres zu den Unterschieden im Unterricht, beziehungsweise in weiteren Beiträgen zu diesem Thema in den kommenden PC-NEWS.

Ein in PASCAL geübter Programmierer wird seinen gewohnten Stil zunächst auch in C anwenden (und das ist ja ein wichtiger Grund für die anfängliche Verwendung von PASCAL). Die Handschrift solcher Programme ist auch gleich erkennbar, da neue Möglichkeiten von C zunächst unbenutzt bleiben; genauso, wie man auch routinierte BASIC-Programmierer daran erkennt, daß sie anfänglich nur zaghaf die sich bietenden Möglichkeit zu Parameterübergabe annehmen und zuerst diese wegen ihrer Nachteile eher zurückweisen.

Benutzen Sie für Ihr erstes Programm in C die folgende Tabelle!

	PASCAL	C
<u>Groß/Kleinschreibung</u>		
GROSS	Schlüsselwörter	Konstanten, Typen
KLEIN	-	Schlüsselwörter, Variablen
GR/KL	eigene Namen, "lange" Schlüsselwörter	Aktivitäten (Funktionen)
<u>Konstanten</u>		
	CONST ccc = 5;	#define CCC 5 const int CCC=5;
Hex-Konstanten	\$3A	0x3a
<u>ganzzahlige Typen</u>		
1 Byte mit Vorzeichen	ShortInt	signed char
1 Byte ohne Vorzeichen	Byte	unsigned char
1 Byte voreingestellt		char (wählbar)
2 Bytes mit Vorzeichen	Integer	signed int
2 Bytes ohne Vorzeichen	Word	unsigned int
2 Bytes voreingestellt		int (signed)
4 Bytes mit Vorzeichen	LongInt	signed long
4 Bytes ohne Vorzeichen		unsigned long
4 Bytes voreingestellt		long (signed)
<u>Spezielle ganzzahlige Typen</u>		
1 Byte boolsch	Boolean (True, False)	0 (false) !=0 (true)
1 Byte Zeichen	Char	char
2 Bytes Aufzählung	(....)	enum {....}
<u>Gleitkommatypen</u>		
4 Bytes	Single	float
6 Bytes	Real	-
8 Bytes	Double	double
10 Bytes	Extended	long double
<u>neue Typen</u>		
	TYPE Element = INTEGER; Zeichen = CHAR;	typedef int Element; typedef char Zeichen;
<u>Zusammengesetzte Datentypen</u>		
Arrays	IntArr : ARRAY[4..6] of INTEGER; Int2Arr : ARRAY[1..10,1..2] of INTEGER; int	int IntArr[3]; int Int2Arr[10][2];
Strings	string[]	char[]
Records	VAR x : RECORD END;	struct { } x;

```

TYPE
  ZusVarTyp = RECORD
    Zahl : INTEGER;
    Name : STRING[3];
  END;
...
VAR ZusVar : ZusVarTyp;
...
ZusVar.Zahl:=5;
ZusVar.Name='ABC';

WITH ZusVar DO
  Zahl:=5;
  Name='ABC';
END;
    
```

```

typedef struct
{
  int Zahl;
  char Name[3];
} ZusVarTyp;

ZusVarTyp ZusVar;

ZusVar.Zahl=5;
strcpy(ZusVar.Name, "ABC");
    
```

Varianten

```

TYPE
  VarTyp = RECORD
  CASE Integer OF
    0: (Zahl : INTEGER);
    1: (Name : STRING[3]);
  END;
...
VAR ZusVar : ZusVarTyp;
...
ZusVar.Zahl:=5;
ZusVar.Name='ABC';

WITH ZusVar DO
  Zahl:=5;
  Name='ABC';
END;
    
```

```

typedef union
{
  int Zahl;
  char Name[3];
} VarTyp;

ZusVarTyp ZusVar;

ZusVar.Zahl=5;
strcpy(ZusVar.Name, "AB");
    
```

Stringhandhabung

Strings im Speicher

```

<len><z1><z2>...
a : string[2];
b : string[2];
c : string[4];
...
a = 'AB';
b[0] = ord('C');
b[1] = 'C';
b[2] = 'D';
c = a + b;
    
```

```

<z1><z2>...0
char a[3];
char b[3];
char c[5];
...
strcpy(a, "AB");
b[0]='C';
b[1]='D';
b[2]=0;
strcpy(c,a);
strcat(c,b);
    
```

Initialisierte Variablen

```

CONST
  a : integer =5;
  b : ARRAY[0..2] of CHAR =
    { 'a', 'b', 'c' };
    
```

```

int a = 5;
char b[3] =
  { 'a', 'b', 'c' };
    
```

dynamische Variablen

Nullpointer

nil

NULL

Pointerdefiniton

```

VAR
  Zeiger = ^Integer;
..
    
```

```

int *Zeiger;
    
```

einzelne Variable

```

New(Zeiger);
Dispose(Zeiger);
Mark(Zeiger);
Release(Zeiger);
    
```

```

Zeiger =
(int *)malloc(sizeof(int));
free(Zeiger);
-
-
    
```

Speicherbereiche

```

GetMem(Zeiger, 5*SizeOf(INTEGER));
    
```

```

Zeiger =
(int *)malloc(5*sizeof(int));
    
```

globale Variablen

am Programmbeginn,
bei TURBO-PASCAL auch an
beliebigen anderen Stellen

vorzugsweise am Programmbeginn

lokale Variablen

vor dem BEGIN des Prozedur-
oder Funktionskörpers

unmittelbar nach
dem Beginn eines Blocks ({})

Typumwandlungen

in ganze Zahl	I := Integer('A');	i = (int)'A';
in Zeichen	Ch := Char(5);	ch = (char) 5;

Operatoren

Zuweisung	:=	=
Vergleich Gleichheit	=	==
Vergleich Ungleichheit	<>	!=
Vergleich größer	>	>
Vergleich kleiner	<	<
Vergleich gr/gleich	>=	>=
Vergleich kl/gleich	<=	<=
logisches UND	and	&&
logisches ODER	or	
logisches NICHT	not	!
Inkrementieren	Succ(A), Inc(A)	++a oder a++
Dekrementieren	Pred(A), Dec(A)	--a oder a--
Addition	+	+ +=
Subtraktion	-	- -=
Multiplikation	*	* *=
Division(ganzz.)	div	/ /=
Division(gleitk.)	/	/ /=
Modulo	mod	% %=
Rechtsschieben	shr	<< <<=
Linksschieben	shl	>> >>=
bitweises UND	and	& &=
bitweises ODER	or	=
bitweises EXOR	xor	^ ^=
bitweises NICHT	not	~
Adresse	Addr()	&
Zeigerzugriff	^	*
Größe	SizeOf(typ)	sizeof(typ)

sonstige reservierte Wörter

Programm	PROGRAM	-
Konstanten	CONST	#define oder const
Typen	TYPE	typedef
Variablen	VAR	-
Prozeduren	PROCEDURE	-
Funktionen	FUNCTION	-

Steuerstrukturen

Leeranweisung	;	;
Block	BEGIN...END	{...}
Verzweigung	IF bed THEN ELSE	if (bed) else
Mehrfachverzweigung	CASE aus OF :... :... ELSE ... END;	switch (aus) { case <el> : ...[break;] case <el> : ...[break;] default :[break;] }

Schleifen

Abfrage am Beginn	WHILE bed DO Anw;	while (bed) Anw;
Abfrage am Ende	REPEAT UNTIL bed	do Anw; while (bed);
Zählschleife	FOR var:=st TO en DO Anw;	for (aus1;aus2;aus3) Anw;

Novelle zum Fernmeldegesetz

Martin Weissenböck, HTL Wien 4, ADIM

Anfang Dezember 1992 war in den Tageszeitungen von einer geplanten Neufassung des Fernmeldegesetzes zu lesen, das eine Reihe von gravierenden Änderungen vor allem im Bereich der Telekommunikation mit sich bringen würde. So sollte der Verkauf und die Verwendung von nicht zugelassen Modems mit einer Strafandrohung bis 100.000 S belegt werden. Außerdem sollte das Gesetz mit 1. Jänner 1993 (Start des EWR) in Kraft treten. Der offizielle Entwurf liegt nun vor. Aus den Erläuterungen (wörtliche Zitate kursiv geschrieben):

Zur Vorbereitung der legislatischen Arbeiten hat ... das Bundesministerium für öffentliche Wirtschaft und Verkehr, Generaldirektion für die Post- und Telegraphenverwaltung, Ende 1989 der Wirtschaftsuniversität Wien einen Forschungsauftrag zur Durchführung einer internationalen rechtsvergleichenden Studie erteilt. Das Ergebnis der Studie und die erforderliche Anpassung des österreichischen Fernmelderechts an das einschlägige EG-Gedankengut (insbesondere bezüglich der Liberalisierung des Telekommunikationswesens) haben den nunmehr vorgelegten Entwurf maßgeblich geprägt.

Wesentlich erscheint die Neufassung des bisherigen Par. 26, nunmehr Par. 32 (Verwaltungsstrafbestimmungen): Wie schon bisher ist der unbefugte Betrieb einer Funkanlage strafbar. Das ist durchaus verständlich, da mit unbefugten Sendeeinrichtungen die Einsatzbereitschaft lebenswichtiger Systeme gestört werden kann. Neu ist aber Ziffer 3:

(Wer) ... nicht zugelassene und nicht entsprechend gekennzeichnete Endgeräte anbietet oder vertreibt ... macht sich einer Verwaltungsübertretung schuldig...

Wie weit die Verwendung nicht zugelassener Modems unter den Begriff "unbefugtes Betreiben einer Fernmeldeanlage" fällt, war aus dieser Stelle noch nicht herauszulesen; vielleicht fällt das aber unter den Par. 2, "Mißbräuchliche Verwendung".

Die Bewilligungspflicht ist im Par. 4 geregelt:

Die Errichtung und der Betrieb von Fernmeldeanlagen ist nur mit Bewilligung zulässig.

In den Erläuterungen dazu heißt es:

Die grundsätzliche Pflicht zur Bewilligung der Errichtung und des Betriebs von Fernmeldeanlagen ist Grundlage und Voraussetzung für einen ordnungsgemäßen und störungsfreien Fernmeldeverkehr, welcher an die Einhaltung bestimmter technischer, funktioneller und betrieblicher Bedingungen gebunden ist. Weicht eine Fernmeldeanlage von diesen Bedingungen ab, so ist es im Interesse des störungsfreien Betriebes anderer Fernmeldeanlagen erforderlich, regulierend einzugreifen.

Ein verständlicher Standpunkt; wie oft haben aber tatsächlich "nicht zugelassene Modems", die in anderen Staaten der Erde problemlos funktionieren, nachweislich Störungen im österreichischen Fernmeldenetz verursacht?

Da sich der Gesetzentwurf ausdrücklich zur Liberalisierung bekennt, wäre wohl zu erwarten, daß die Zulassung durch eine Fernmeldebehörde (z.B. in den USA) für die Zulassung in Österreich auch reicht.

Für das österreichische BTX-System wäre eine Strafbestimmung für nicht zugelassene Geräte fast ein Todesurteil, da sehr viele BTX-Betreiber auf die bisherige liberale Haltung der Post bauen: es werden bekanntlich auch Teilnehmer akzeptiert, von denen die Post nicht weiß, welche Geräte verwendet werden. Wenn nun der Betrieb von Modems strafbar wäre und die BTX-Adressen bei der Post aufliegen, erwarte ich eine gewaltige BTX-Abmeldewelle.

Ob mit einer Genehmigungspflicht für Mailboxen, wie sie in den Zeitungen auch schon diskutiert worden ist, dem Problem der Verbreitung von rechtsradikalen Programmen begegnet werden kann, darf aus der Praxis mehr als angezweifelt werden. Besser wäre sicher eine entsprechende Aufklärungsarbeit, die derartige Produkte nicht attraktiv macht.

Zusammenfassung: der Abstimmung der Schweiz entsprechend ruht der EWR, damit auch der "europakonforme" Fernmeldegesetzentwurf. Wie mir vom zuständigen Referenten in der Bundeswirtschaftskammer versichert wurde, gab es im Begutachtungsverfahren eine Reihe von sehr kritischen Stimmen zu diesem Entwurf. Eine Reaktion der Post auf den drohenden Abschluß von BTX wäre auch noch sehr interessant. Eine modifizierte Vorlage tritt kaum vor dem 1.1.1994 in Kraft; die **PC-NEWS** werden weiter über die Entwicklung berichten. □

Kommunikation total

Martin Weissenböck, HTL Wien 4, ADIM

Eine Reihe von neuen interessanten D-Netz-Telefonen sind 1993 erhältlich. Welche Wünsche können heute beim Kauf an ein Mobiltelefon gerichtet werden?

Auswahlkriterien:

- Es soll **klein, leicht und flach** sein. Achtung: in manchen Unterlagen wird der Preis **ohne** Batterie (Akku) angegeben!
- Die **Antenne** soll eingeschoben werden können: der Empfang eines Rufes muß auch bei eingeschobener Antenne möglich sein; im Sprech-Betrieb soll dann die Antenne ausgezogen werden.
- Die **Tastatur** soll gegen unbeabsichtigte Betätigung beim Tragen in einer Tasche geschützt sein: entweder über einen Deckel oder eine einfache elektronische Verriegelung. Alle Telefone verfügen über eine elektronische Sperre; wenn aber nur eine unabsichtliche Betätigung verhindert werden soll, ist diese Verriegelung zu kompliziert.
- Beleuchtung**: recht praktisch; jedoch sollte die Tastatur und nicht nur die Anzeige beleuchtet werden.
- Lautstärke des Rufes**: wenn die Lautstärke eingestellt werden kann, ist das sehr praktisch, vor allem, wenn der Mobiltelefonbesitzer gerade in einer Besprechung ist und nicht durch laute Rufsignale gestört werden will.
- Booster**: ein Verstärker für das Sendesignal, um die Sendeleistung eines Mobiltelefons von 0,6 Watt auf 3 Watt (Autogerät) zu erhöhen. Der Empfangsteil eines Mobiltelefons ist üblicherweise auch außerhalb von Städten empfindlich genug; beim Senden kann es außerhalb der Ballungsgebiete zu Problemen kommen.
- Schnittstellen**: für die totale mobile Kommunikation ist der Anschluß eines Modems - am besten mit Fax-Teil - notwendig.
- Aktive Schnittstelle**: relativ teuer; ein Faxgerät, ein Modem, ein Anrufbeantworter, ja sogar ein normales Telefon kann angeschlossen werden. Bei abgehenden Rufen kann über das Endgerät (Fax, Modem, Telfon) gewählt werden, das Mobiltelefon muß nicht betätigt werden.
- Passive Schnittstelle**: ankommende Rufe müssen am Mobiltelefon übernommen werden; abgehende Rufe (Telefonnummern) werden am Mobiltelefon eingegeben.
- Europaweites digitales Netz**: der Probetrieb beginnt bereits in kleinen Teilen des Netzes (Wien, Flughafen,...). Eine gesamtösterreichische Versorgung wird wohl noch drei bis vier Jahre benötigen. Ebenso lange wird es wohl dauern, bis Geräte in passender Größe und zu vergleichbarem Preis erhältlich sein werden.
- Betriebsdauer**: mit normalem Akku, mit Hochleistungsakku.

- Anschlußmöglichkeit für externe Antenne**: sinnvoll, wenn ein Gerät stationär eingesetzt werden soll. Magnethaftantennen können auch verwendet werden, wenn ein Mobiltelefon für ein Auto vorgesehen ist, jedoch keine Löcher gebohrt werden sollen.
- Nummernspeicher**: allgemein üblich für 100 Telefonnummern; samt alphanumerischer Anzeige und Suche.
- Notizbuchspeicher**: einspeichern einer Telefonnummer während eines Telefongesprächs.
- Anrufspeicher**: relativ neue Funktion; wenn der Mobiltelefonbesitzer nicht "abhebt", kann der Anrufende über Telefone mit Tonfrequenzwahl seine Telefonnummer für einen Rückruf hinterlassen.

Geräte:

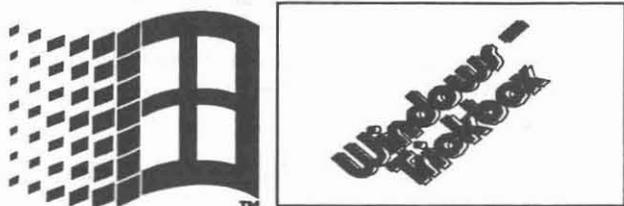
- Autotelefon**: fix in ein Auto eingebaut, meistens 3 Watt Sendeleistung. Diese Geräte sind bereits sehr preiswert erhältlich, eine gemeinsame Einkaufsaktion wäre daher auch nicht sinnvoll. Wichtig: Freisprecheinrichtung.
- Transmobile Geräte**: Geräte in der Größe einer kleinen Aktentasche; besonders geeignet, wenn ein "mobiles Büro" eingerichtet werden soll. Auch mit Datenschnittstelle erhältlich. Diese Geräte können oft auch in Autos eingebaut werden. Zu schwer zum Mittragen.
- Mobilgeräte**: derzeit bei etwa 250 g Masse. Leistung: 0,6 Watt. Betriebsbereitschaft je nach Akku bis zu 22 Stunden, Sprechzeit im allgemeinen unter einer Stunde. Mit einer Autohalterung auch in Kraftfahrzeugen zu verwenden.
- Mobilgeräte mit Booster**: auch außerhalb von Ballungszentren zu verwenden. Die Booster werden nur für den Einbau in Autos angeboten; eine Verwendung in einem passenden "Aktenkoffer" ist zu überlegen, um je nach Einsatz ein mobiles oder ein transmobiles Gerät zur Verfügung zu haben. *Wir wollen einen passenden Vorschlag für die nächsten News vorbereiten.*
- Tischladegerät**: bequeme Möglichkeit zum Aufladen der Akkus. Nickel-Cadmium-Akkus sollten regelmäßig entladen werden. Eine Möglichkeit wäre, das Gerät einmal pro Woche bis zum "Batterieende" eingeschaltet zu lassen. Moderne Ladegeräte entladen den Akku vor dem Wiederladen.

Bei Redaktionsschluß lagen bereits Informationen über einige Geräte vor, die sehr viele der gewünschten Eigenschaften kombinieren. Leider waren noch nicht alle Preise für Zusatzgeräte bekannt. Wir planen eine gemeinsame Einkaufsaktion in der nächsten Ausgabe der PC-NEWS. Wenn Sie Interesse haben, schreiben Sie bitte schon jetzt eine Postkarte an die

ADIM, Postfach 23, A-1191 Wien.

WINDOWS

Eduard Fleck, N, TGM



Windows - Feuerwerk

Man sollte meinen, die Programmierer von Microsoft sind voll ausgelastet - kein Platz für Spielereien! Dem ist aber offensichtlich doch nicht so. Für einen kleinen Scherz ist immer noch Zeit:

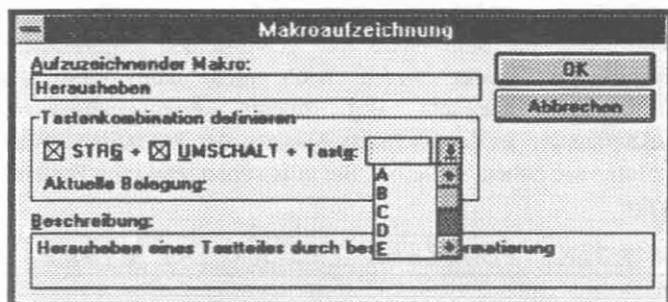
Eine kleine Animation, die mit einem Feuerwerk endet, die aber mit keiner dokumentierten Funktion zum Leben erweckt werden kann. Diese geheime Grafik kann folgendermaßen gestartet werden:

1. Im Menü *Extra* das Dialogfenster *Makro* öffnen.
2. Als Makronamen *SPIFF* eingeben.
3. Mit *Bearbeiten* das Makro- Editierfenster öffnen.
4. Löschen Sie alle Zeilen!
5. Schließen Sie das Fenster und bestätigen Sie die Frage mit ja. Das Makro wird nun gespeichert.
6. Wählen Sie jetzt im *Hilfemenü* den Punkt *Info* an.
7. Sobald Sie auf das Word für Windows - Logo klicken, wird die Animation gestartet!



Erstellen eines eigenen Makros für die Formatierung bestimmter Textteile eines Textes:

1. Wählen Sie zuerst aus dem Menüpunkt *Extras* den Punkt *Makroaufzeichnen*.
2. Im Eingabefenster müssen sie nun einen *Namen* festlegen, eventuell eine Tastenkombination bestimmen, mit der sie jederzeit einen markierten Text formatieren können. Durch Anklicken des Auswahlkästchens können sie sowohl die <STRG> - Taste als auch die <Umschalt> -

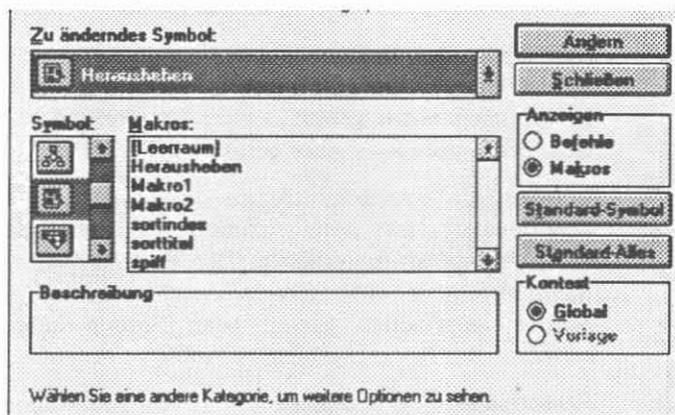


Taste deaktivieren bzw. aktivieren.

3. Durch Anklicken des OK - Buttons oder Bestätigen mit der Enter- Taste wird der Makrorekorder gestartet. Jeder angewählte Befehl wird nun aufgezeichnet, wobei Mausaktivitäten nicht mit aufgezeichnet werden!
4. Wählen sie aus der Formatierungsleiste oder durch Anwählen des Punktes *Zeichen* im Menüpunkt *Format* die Schriftart, Größe sowie weitere Formatierungen (z.B.: Fettdruck, zentriert, o.a.)
- 5: Beenden sie die Makroaufzeichnung mit *Aufzeichnen beenden* in *Extras*.

2. Einbinden in die Funktionsleiste

1. Wählen sie in *Extras* den Menüpunkt *Einstellungen* und klicken sie das Symbol für die Funktionsleiste an.
2. Wenn sie im Optionsfeld *Anzeigen* den Punkt *Befehle* anwählen, werden im Auswahlfenster *Befehle* alle Makros angezeigt, so auch das vorher erstellte *Herausheben-Makro*.
3. Durch Auswählen eines Symbols (am besten *Leerraum*) im Auswahlmenü *Zu änderndes Symbol* und Auswählen des



Makros sowie eines Symbols aus der Symbolleiste wird mit dem *Ändern-* Button ein neuer Button in die Funktionsleiste eingefügt.

4. Nach dem sie Schließen ausgewählt haben, erscheint unser neues Symbol in der Funktionsleiste.

3. Formatieren eines Textteiles:

▪ einer beliebigen Textstelle	Ziehen Sie über den zu markierenden Text. Zum Markieren eines beliebigen Textes können Sie auch die Einfügemarke an ein Ende der Textstelle setzen, die UMSCHALTTASTE gedrückt halten, und dann am anderen Ende der Textstelle klicken.
▪ eines Wortes	Doppelklicken Sie auf das Wort.
▪ einer Zeile	Klicken Sie auf die Markierungsleiste links von der Zeile.

Stufenlos zoomen:

Sie können im Menü *Ansicht Zoom* im Eingabefeld *Benutzerdefiniert* den Textausschnitt stufenlos vergrößern bzw. verkleinern.

- Eleganter und schneller ist jedoch die Variante eines eigenen Buttons in der Funktionsleiste!
- Wählen Sie dazu in *Extras Einstellungen* aus.
- Klicken Sie im Fenster Kategorie Funktionsleiste an.



- Stellen Sie das Anzeigekontrollfeld auf *Befehle* und suchen Sie im Fenster *Befehle* den Eintrag *Ansicht Zoom*
- Wählen Sie im Pull - Down Fenster *Zu änderndes Symbol* am Ende der Liste *Leerraum* aus.
- Suchen Sie im Fenster *Symbol* nach der *Lupe* und
- Klicken Sie auf *Ändern*.

Speichern im ASCII - Text - Format:

Manchmal kann es vorkommen, daß ein in *Word für Windows* erstellter und formatierter Text ohne der vorhandenen Formatierung weiterverarbeitet werden muß. In diesen Fällen kann das Word - Dokument ohne den Formatzusätzen abgespeichert werden:

1. Öffnen Sie das Eingabefenster [Datei] [Speichern unter]
2. Wählen Sie im Auswahlfenster *Dateityp DOS Text (*.txt)*
3. Geben Sie einen Dateinamen an und klicken Sie auf *OK*.

Grafik einfügen:

Es gibt insgesamt drei Möglichkeiten, Grafiken in ein Dokument einzufügen.

1. Über die Zwischenablage:

- Sie können beispielsweise eine Grafik aus Paintbrush einfügen, in dem Sie die Grafik markieren und im Menüpunkt *Bearbeiten Kopieren* auswählen oder mit den Tasten *<Strg> + <Einf>* in die Zwischenablage kopieren.

- oder
- sie kopieren ein Anwendungsfenster mit den Tasten *<Alt> + <PrintScreen>* in die Zwischenablage
- Mit *Bearbeiten Einfügen* oder *<↑> + <Einf>* kann der Inhalt der Zwischenablage in das Dokument eingefügt werden.

2. Über Einfügen Graphik:



- Öffnen Sie das Fenster *Einfügen Graphik* aus und wählen Sie die einzufügende Datei aus.
- Im rechten Teil können Sie sich vor dem Einfügen die Datei in einer Vorschau ansehen.
- Klicken Sie *OK*
- Die Datei wird nun an der Stelle eingefügt, an der der Textcursor steht.

3. Grafik mit Dokument verknüpfen:

- Wird eine Grafik mit einem Dokument verknüpft, wird das Bild nicht Bestandteil des Dokumentes. Es erfolgt lediglich ein Eintrag einer Funktion, die ein Verweis auf die Position der Grafikdatei auf einem Datenträger darstellt.
- Der Vorgang ist der gleiche wie oben beschrieben, jedoch wird im Bearbeitungsfenster *Einfügen Graphik* das Kontrollkästchen *Mit Datei verknüpfen* aktiviert.
- An die Stelle des Textcursors wird dabei die Feldfunktion eingefügt:
{IMPORT D:\\WINWORD\\WORDLOG3.BMP * FormatVerbinden}
- Mit *Ansicht Feldfunktion* oder mit *<Alt> + <F9>* können Sie nun zwischen der Darstellung der Grafik und der Anzeige der Feldfunktion umschalten.

Der Vorteil der letztgenannten Methode liegt darin, daß die Größe des Dokumentes in einem handhabbarem Maße bleibt, wohingegen eine Textdatei mit mehreren Grafikbildern sehr schnell mehrere MBytes groß sein kann!

Zu berücksichtigen ist jedoch, daß der Verweis auf die Grafikdatei statisch ist und bei einem Verlagern in ein anderes Verzeichnis der Bezug verloren geht. □

910DLP

Programm zum automatischen Einspielen von Cept-Seiten und Telesoftware

Helmuth Schlögl, MCCA

PC-NEWS-29, S.15

In der vorigen Ausgabe der **PC-NEWS** habe ich die Erstellung und das umständliche Einspielen von Telesoftware beschrieben.

Diesen Artikel nahm unser MCCA-Vorstandsmitglied, Programmator von BTX-DELUXE, Johannes Sautner zum Anlaß und erstellte ein komfortables Programm zum Einspielen von Cept-Seiten und Telesoftware.

Kurzdokumentation zu Programm 910.DLP

Zweck: Spielt vollautomatisch BTX-Seiten in das BTX-System ein.

Dateien

```
910.DLP .. dieses Programm
BTS.DEF .. Konfiguration für 910.DLP
           (Kennung,Kennwort,Editierkennwort)
910.DOC .. diese Dokumentation
BEISPIEL .. Muster-Datei
```

Programmaufruf

```
*RUN 910 <Datei><Return-Taste>
```

z.B.:

```
*run 910 beispiel<Return-Taste>
```

oder aus MSDOS z.B.:

```
BTX /r910 beispiel
```

Um diesen Programmaufruf aus einer Dos-Batch aufzurufen und anschließend aus BTX z.B. das TSW-Programm zur Kontrolle wieder in Ihr TSW-Verzeichnis zu laden, ist folgender BTX-DELUXE Batch zu erstellen.

Dateiname

```
910ed.b
```

Inhalt

```
910
type Seitennummer
ANWAHL ABWAHL
SYSTEM
if not exist datei.a (tsw oder Ceptseite) goto weiter
call BTX /r910ed.b beispiel
del datei.a
:weiter
```

Die Übertragung sollte auch optisch kontrolliert werden. In der derzeitigen Programmversion ist bei Ausfüllen der Seitenparameterseite keine Echokontrolle enthalten!!!

BTS.DEF muß zur Programmausführungszeit im aktiven Verzeichnis vorhanden sein (z.B. C:\BTX)

Die Seitenparameterdatei (BEISPIEL) enthält die Beschreibung der einzelnen BTX-Seiten.

- * je Zeile eine BTX-Seite
- * Felder mit "," getrennt

```
Seitennummer, z.B.      4020a
Überarbeitungsart      (ä,ü,l,e)
GBG-Nummer, z.B.      580
TLN-Zugriff           "J" oder "N"
Seitentyp:            "I" oder "A"
Seitenpreis:          0 bis 999 z.B. "9.5"
Verzweigungen:        "F" oder "K"
Wahl-0, z.B.          "40200" oder "*"
                       (keine Verzweigung)
```

Wahl-1

Wahl-2

Wahl-3

Wahl-4

Wahl-5

Wahl-6

Wahl-7

Wahl-8

Wahl-9

Übertragungsart: "F"=Framedump, "O"=Online

Beispiele

```
25507a,ü,7a.btt,,J,I,0,f,2550,255071,255072,255075,255080,255075
3,**,2550891,*,f,
255074a,l,74a.btt,,J,I,0,f,25507,*,*,*,*,*,*,255075,f,
255071a,ü,71a.btt,,J,I,0,f,25507,*,*,*,*,*,*,255072,f,
255072a,ü,72a.btt,,J,I,0,f,25507,25507523,8800,*,*,*,*,2550891
,255075,f,
255075a,ü,75a.btt,,J,I,0,f,25507,88889701,25507523,2550754,*,*,2
550751,*,*,255080,f,
25507523a,ü,7523a.btt,,J,I,0,f,255075,255075231,88889706,*,*,*,
*,*,*,f,
2550751a,ü,751a.btt,,J,I,0,f,255075,*,*,*,*,*,*,25507510,f,
25507510a,ü,7510a.btt,,J,I,0,f,2550751,*,*,*,*,*,*,25507511,
f,
25507511a,ü,7511a.btt,,J,I,0,f,2550751,*,*,*,*,*,*,255075,f,
25507526a,ü,7526a.btt,,J,I,0,f,255075,88889707,255051616,*,*,*,
*,*,*,f,
25507527a,ü,7527a.btt,,J,I,0,f,255075,88889708,*,*,*,*,*,*,f
,
2550752a,ü,752a.btt,,J,I,0,f,25507,88889701,25507523,2550754,*,*
,2550751,*,*,255080,f,
2550754a,ü,754a.btt,,J,I,0,f,88889703,88889704,88889711,88889712
,88889713,88889714,25507526,25507527,88889702,*,f,
255032a,ü,32a.btt,,J,I,0,f,25503,*,*,*,*,*,*,255038,*,f,
```

Datei BTS.DEF

```
2550 ; Leitseite
1234567890 ; kennung
11019300 ; kennwort
912222064 ; BTX-Nummer
1234 ; Editier-Kennwort
MCCA ;Header
MCCA BTX & MICRO COMPUTER CLUB' ;Name
Traugasse 12 ;Adr1
Postfach 143 ;Adr2
A-1033 Wien ;Adr3
. ;Adr4
02225851393 ;Telefonnummer
```

Mit diesem Programm, welches lt. Aussage des Autors noch verbessert werden wird, kann man aus einer DOS Batch-Datei zeitgesteuert seine geänderten Cept-Seiten oder Telesoftware einspielen lassen. □

Anm.: Das Programm können Sie beim Autor von BTX-deluxe, Johannes Sautner, Schulerstraße 1-3/2, 1010 Wien, BTX: 912222628, Tel.: (0222)/512-57-71 beziehen.

BTX-taugliche Adressen aus dBASE-Datei

Walter Riemer, Abt. N, TGM

DSK 294:\BTXADR

Für Zwecke eines Serienbriefes oder der Christkindl-Weihnachtskarte in BTX muß die Adressendatei eine zeilenweise organisierte Textdatei sein, in der die vier Felder durch Kommas getrennt sind und kein Feld länger als 20 Zeichen ist.

Das dBASE-Programm `BTXAdr.PRG` entnimmt die Adressen aus einer `.DBF`-Datei, deren Name erfragt wird (z.B. "adressen") mit folgender Struktur:

Feld	Feldname	Typ	Länge	Beispiel:
1	NAME	Zeichen	25	NAME Max Siebenklug
2	SEX	Zeichen	1	SEX M
3	TITEL	Zeichen	20	TITEL Ing.
4	STRASSE	Zeichen	30	STRASSE Heferlgasse 314/15/9
5	PLZORT	Zeichen	20	PLZORT 1111 Wien
6	ART	Zeichen	8	ART PF
** Gesamt **			105	

Zur Erläuterung eines Adressensatzes: Im Feld `ART` können Zeichencodes oder dergleichen gespeichert werden, die ein Selektieren nach gewissen Merkmalen erlauben; Herr Siebenklug ist zum Beispiel Mitglied des PCC-TGM ("P") und ein begeisterter Fußballanhänger ("F").

Die aus dem Feld `SEX` abgeleitete Anrede (*Hrn.* oder *Frau*) wird mit dem Inhalt des Felds `TITEL` zusammengefaßt und im BTX-Adressfeld `NAME1` untergebracht. Dann folgen die Felder `NAME` (wird zu `NAME2`), `STRASSE` und `PLZORT`.

Falls der Feldinhalt in einem der sich ergebenden Felder länger als 20 Zeichen ist, wird das Feld zum Editieren angeboten. Felder, die ein Komma enthalten, werden ebenfalls zum Editieren angeboten, da das Komma in der BTX-Adressendatei als Trenner fungiert.

Mit Hilfe der Zwischendatei `BTXADR.DBF` wird die Textdatei `adressen.BTX` erzeugt (für `adressen` wird der Name der Stamm-`.DBF`-Datei eingesetzt).

Die Wortteile *straße*, *platz* und *gasse* werden automatisch auf *str.*, *pl.* und *g.* abgekürzt.

Ausländische Adressen werden nicht übernommen, da BTX-Serienbriefe nur im Inland möglich sind. Es erfolgt eine Meldung auf dem Bildschirm.

Das dBASE-Programm besteht aus einem Hauptprogramm (als `BTXAdr.PRG` angespeichert) und einem Unterprogrammmodul (als `BTXAdrU.PRG` abgespeichert). Das Programm ist ab dBASE III+ lauffähig.

```
*** BTXAdr.PRG: aus Adressen.DBF
*** BTX-taugliche Adressendatei machen
SET TALK OFF
SET SAFETY OFF
PUBLIC vName1, vName2, vStrasse, vPLZort
SET PROC TO BTXAdrU
DO BTXAdrU
SET TALK ON
SET SAFETY ON
SET PROC TO
```

```
*** BTXAdrU.PRG: Unterprogrammmodul zu BTXAdr.PRG
PROCEDURE BTXAdrU
CLEA ALL
USE BAdrStd
COPY STRU TO BTXAdr
SELE 1
USE BTXAdr ALIAS BAdr
ZAP
ACCEPT "Adressen-DBF-Datei (ohne .DBF): " TO vDBFName
? " "
SELE 2
USE &vDBFName ALIAS AAdr
GO TOP
DO WHILE .NOT. EOF()
IF .NOT. ISALPHA(PLZort)
DO CASE
CASE Sex="M"
vName1="Hrn."
DO MachName1
CASE Sex="W"
vName1="Frau"
DO MachName1
OTHERWISE
vName1=Titel
ENDCASE
DO FindKomma WITH "&vName1"
vName2=Name
DO MachFeld WITH vName2,"0"
vStrasse=Strasse
DO MachFeld WITH vStrasse,"s"
vPLZort=PLZort
DO FindKomma WITH vPLZort
SELE BAdr
APPE BLANK
REPL Name1 WITH vName1
REPL Name2 WITH vName2
REPL Strasse WITH vStrasse
REPL PLZort WITH vPLZort
SELE AAdr
ELSE
? "Nicht aufgenommen: ",RTRIM(Name)+",
"+RTRIM(PLZort)+CHR(7)+CHR(7)
ENDIF
SKIP
ENDDO
SELE BAdr
GO TOP
vBTXName=vDBFName+".BTX"
RUN DEL &vBTXName
SET ALTE TO &vBTXName
SET ALTE ON
DO WHILE .NOT. EOF()
? RTRIM(Name)+", "+RTRIM(Name2)+", "+
RTRIM(Strasse)+", "+RTRIM(PLZort)+", "
SKIP
ENDDO
? " "
SET ALTE OFF
SET ALTE TO
RETURN
PROCEDURE MachName1
vName1=vName1+" "+Titel+" "
DO WHILE LEN(RTRIM(vName1))>20
? "Feld zu lang: >|" +CHR(7)
? " "
@ ROW(),0 GET vName1
READ
? " "
DO FindKomma WITH vName1
ENDDO
RETURN
PROCEDURE MachFeld
PARAM vFeld,vSteuer
IF vSteuer="S"
DO Kuerzen WITH vFeld,"straße","str."
DO Kuerzen WITH vFeld,"platz","pl."
DO Kuerzen WITH vFeld,"gasse","g."
ENDIF
DO WHILE LEN(RTRIM(vFeld))>20
vFeld=vFeld+" "
? "Feld zu lang: >|" +CHR(7)
? " "
@ ROW(),0 GET vFeld
READ
? " "
DO FindKomma WITH vFeld
ENDDO
RETURN
PROCEDURE Kuerzen
PARAM vFeld,vAlt,vNeu
vPos=AT(vAlt,vFeld)
IF vPos<0
vRestLaenge=LEN(vFeld)-(LEN(vAlt)-LEN(vNeu))
vFeld=SUBSTR(vFeld,1,vPos-1)+
vNeu+SUBSTR(vFeld,vPos+LEN(vAlt),vRestLaenge)
ENDIF
```

```

PROCEDURE FindKomma
PARAM vFeld
DO WHILE AT(",",vFeld) <> 0
? "Komma entfernen!"
? " "
@ ROW(),0 GET vFeld
READ
? " "
ENDDO
RETURN

```

Das Ergebnis der Bearbeitung würde, wenn nur die einzige oben als Beispiel genannte Adresse besteht, so aussehen:

Hrn. Ing., Max Siebenklug, Heferlgasse 314/15/9, 1111 Wien,

""

Die vier Kommas in der letzten Zeile sind notwendig, um dem Programm eine korrekte Beendigung zu ermöglichen. Jede Zeile muß genau 4 Kommas enthalten; dies gilt auch, falls ein Feld leer bleibt.

Nun steht dem Versenden einer Mitteilung an viele Personen, deren Adressen in einer dBASE-Datei erfaßt sind, nichts mehr im Wege. Man braucht nur mehr ein geeignetes Programm (der Autor benützt dafür den VERBIS BTX-Butler).

Dieser "Diener" hat die Funktion "Serienbriefe" schon fix einprogrammiert, weil dies bei BTX-Anwendungen besonders gefragt ist.

Um allerdings Weihnachtsgrüße über Christkindl (über BTX-Seite 890003) automatisiert zu versenden, muß man ein eigenes Ablaufprotokoll für den Butler schreiben, das ihm sagt, was er tun soll. Für Personen, die ebenfalls den BTX-Butler benützen, füge ich hier noch dieses Ablaufprotokoll (Programm für den BTX-Butler) hinzu:

```

BTX-FORMULAR: ChristKi.BFM ; Anmelden des benötigten
; BTX-Formulars
DATEI 1: WeihTest.txt,LESEN ; zu sender Grubtext
; im SENDFIX-Format
DATEI 2: WeihTest.btx,LESEN ; Adressen
WAEHLEN: 890003 ; Christkindl-Seite wählen
ABWARTEN: 2000 ; 2 s warten für Bildaufbau
MARKE 1:
EINGABE: 3 ; Absender = BTX-Teilnehmer
LESEN: DATEI 2, ZEILE 1, MARKE 9 ; erste / nächste
; Adreß-Zeile lesen, Felder
; trennen und in Tabelle eintragen
SENDE FELD: 1 ; Adresse erste Zeile senden
EINGABE: # ; # (TER) senden
SENDE FELD: 2 ; Adresse zweite Zeile senden
EINGABE: # ; # (TER) senden
SENDE FELD: 3 ; Adresse dritte Zeile senden
EINGABE: # ; # (TER) senden
SENDE FELD: 4 ; Adresse vierte Zeile senden
EINGABE: # ; # (TER) senden
TEXT SENDEN: EINZELN,WeihTest.txt ; Text aus Grubtextdatei
; WeihTest.txt senden
EINGABE: # ; # (TER) senden
EINGABE: 2 ; 1 ... Senden,
; 2 im Testbetrieb ... annullieren
; # (TER) senden
EINGABE: #
SPRINGEN ZU: MARKE 1
MARKE 9:
WAEHL: 0
ENDE

```

Die Datei CHRISTKI.BFM (-.Btx-Formular) definiert die Felder auf der Christkindl-BTX-Seite, in die die Adresse bzw. der Grubtext gehört:

DEFINITION:

BTX-FELD 1 : 1, 4, 21, 6 ; Adreßfeld im Christkindl-Formular

BTX-FELD 2 : 1, 9, 40, 16 ; Textfeld

Die Datei 1 enthält den Grubtext (in BTX-tauglichem Format; er wurde mit dem zum Butler gehörenden Editor SENDFIX geschrieben), die Datei 2 enthält die auf dem vorstehend beschriebenen Weg entstandene Adressendatei (im ASCII-Format, also keine SENDFIX-Datei). □

Es gibt kaum etwas auf dieser Welt, das nicht irgend jemand ein wenig schlechter machen und etwas billiger verkaufen könnte, und die Menschen, die sich nur am Preis orientieren, werden die gerechte Beute solcher Machenschaften.

Es ist unklug, zu viel zu bezahlen, aber es ist noch schlechter, zu wenig zu bezahlen. Wenn Sie zu viel bezahlen, verlieren Sie etwas Geld, das ist alles. Wenn Sie dagegen zu wenig bezahlen, verlieren Sie manchmal alles, da der gekaufte Gegenstand die ihm zugedachte Aufgabe nicht erfüllen kann. Das Gesetz der Wirtschaft verbietet es, für wenig Geld viel Wert zu erhalten.

Nehmen Sie das niedrigste Angebot an, müssen Sie das Risiko eingehen, etwas hinzuzurechnen. Und wenn Sie das tun, dann haben Sie auch Geld, um für etwas Besseres zu bezahlen.

*John Ruskin,
engl. Sozialreformer
(1819-1900)*

Besprechung CD-Educational Master

Otto Cap, Kreisgericht Korneuburg

PC-NEWS-29

Die Einladung zum Testen der auf der EDU-CD aufgespielten Software machte mich neugierig. Wenn der leere Platz auf solchen CD's dazu benützt wird, einfach ein Kunterbunt an Programmen - wohl überwiegend Shareware - beizufügen, kann man einen für die Shareware-Szene repräsentativen Querschnitt des dort Gebotenen erwarten. Was wird da wirklich an Brauchbarem geboten? Sicherlich sind Textverarbeitungen und Desktop-Publishing-Programme Anwendungen, die in der Praxis am häufigsten benötigt werden. Sind solche aus dem anglo-amerikanischen Raum stammende und nicht für die deutsche Sprache adaptierte Programme bei uns befriedigend verwendbar? Aber auch anderes kann Interesse erwecken.

Den Empfehlungen des CD-Redakteurs folgend, habe ich zunächst die höchstbenoteten Textverarbeitungen "PC Type4"(EDU-820 bis 823) und "PC Write Standard Level 2.0"(EDU-2774 bis 2777) sowie den "Rubicon Publisher"(EDU-1238/39) getestet, weiters einen Screen-Editor (TheDraw V.3.20, EDU-881) und ein Kalkulationsprogramm (InstaCalc V.3.00, EDU-1183); weiteres soll folgen. Hier ein kurzgefaßtes Ergebnis der genannten Programme:

Alle Programme sind Shareware, d.h. die Prüfung auf Tauglichkeit ist kostenlos, ihre Verwendung in der - vor allem geschäftlichen - Praxis wäre aber an die Entrichtung einer "Registrierungsgebühr" (=Kaufpreis der Software) gebunden. Leider verbieten die Programmautoren der Textverarbeitungen ausdrücklich deren Weitergabe u.a. in Europa. Keinesfalls kann also damit gerechnet werden, daß eine den Copyright-Bestimmungen gemäßige Verwendung dieser beiden Anwendungen bei uns möglich ist. Gehen wir daher im weiteren davon aus, daß die Textverarbeitungen privat ohne Registrierung verwendet werden sollen.

PC Type V.4.0 (EDU-820...823)

Textverarbeitung in funktionaler Vollversion des US-Herstellers Buttonware Inc. vom September 1990, als Shareware jedoch nur mit nicht kontextsensitiver Basis-Online-Hilfe. Mitgeliefert wird allerdings ein ca. 400 S. umfassendes, überaus detailliertes und instruktives (englisches) Handbuch als Textdatei. (Gedrucktes) Handbuch, Support und Updates bei Registrierung. Registrierungsgebühr samt Versand rund 77 US\$. Entpackter Speicherbedarf: ca 1.3 MB.

Das Programm ist als eine Art Ergänzung zu der Datenbank PC-File und dem Kalkulationsprogramm PC-Calc+ des gleichen Herstellers gedacht. Dem entsprechend sind seine Stärken und Schwächen:

Der Import von Textdateien ist im wesentlichen auf die Formate der oben genannten "Schwesterprodukte" eingeschränkt (kann allerdings bezüglich ASCII-Files durch die GET-Funktion umgangen werden), dafür ist der Import aus dBase-kompatiblen Datendateien ebenso möglich wie von Bilddateien des GIF-Formates. Ein Export in ein anderes Textformat findet nicht statt.

Die Verwendung einer Maus ist nicht vorgesehen. Die Bedienung erfolgt durch eine hierarchisch organisierte Menüstruktur älteren Zuschnittes, welche eine flotte Arbeit (besonders beim Fehlen genügender Online-Hilfe) wohl erst nach einiger Übung zuläßt. Nach Einarbeitung dürfte allerdings der Funktionsumfang fast keinen Wunsch offenlassen: multiple Konfigurationsmöglichkeiten lassen sich in Definitionsdateien für späteren Aufruf speichern, die Standardfunktionen für das

Formatieren, Blockoperationen, Holen und Auslagern von Text(teilen), Makros etc. sind soweit überblickbar vollzählig verfügbar; dazu kommen zahlreiche für die Verarbeitung von Tabellen nützliche Funktionen wie Tabellenheader, Überschriftennumerierung, Spaltenumstellung, Parallelverarbeitung von bis zu 10 Dateien, auch mit Hilfe eines geteilten Bildschirms, Rahmungen, Erstellung und Einbindung von Geschäftsgrafiken nach dem importierten Datenmaterial und von Bildern, Kalkulationsfunktionen, Erstellen von Labels und Mailmerge, was eine besonders auf das Geschäftsleben hin orientierte Programmkonzeption erkennen läßt.

Für besonders versierte Benutzer wurde die Implementierung einer Kommandozeile nicht vergessen, mit deren Hilfe alle Befehle unmittelbar eingegeben werden können. Ein Befehlsrekorder hält diese Befehle für spätere Verwendung fest.

Druckerkonfigurationsdateien werden nur für die wichtigsten Marken und Typen mitgeliefert. Druckkommandos für spezielle Druckfunktionen (Unterstreichen, Fettdruck, Kursivdruck, Schriftgröße und -höhe etc.) müssen vor dem Setzen entsprechender Marker im Text anhand des Druckerhandbuches konfiguriert werden. Überdies besteht die Möglichkeit einer Zeichenumsetzung, wenn der Drucker bestimmte ASCII-Codes nicht erkennt. Keine PS-Features für Laserprinter implementiert.

Gesamtbeurteilung

Auf das Geschäftsleben zugeschnittene und für diese Aufgabe sehr leistungsfähige, aber im Handling eher umständliche Textverarbeitung mit besonderen Highlights bei Verarbeitung von Daten und grafischen Objekten. Moderne Entwicklungen noch nicht berücksichtigt.

PC Write Standard Level 2.01 (EDU2774...2777)

Vollversion einer Windows-3.0-fähigen Textverarbeitung des US-Herstellers Quicksoft vom Dezember 1991. Handbuch und Support sowie Updates nach Registrierung. Registrierungsgebühr samt Versand rund US\$ 150.- Entpackter Speicherbedarf: ca.1.6 MB

Die Bedienung erfolgt über Menüzeile und Pull-down-Menüs, die allerdings nicht ganz dem SAA-Standard entsprechen. Als nicht ganz geglückt empfinde ich die Anordnung der Antwortzeile bei Rückfragen am oberen Rand des Bildschirms. Die Verwendung einer Maus ist möglich. Wie bei jeder englischsprachigen Textverarbeitung fehlt ein deutsches Wörterbuch sowie ein ebensolcher Thesaurus(Synonymwörterbuch). Dessen Aufruf sowie Rechtschreibprüfung und Silbentrennung für deutsche Texte sind daher nicht verfügbar. Ansonsten sind alle Standardfunktionen wie beliebiges (auch automatisches) Formatieren des Textes, Block markieren, kopieren, versetzen und löschen, Text holen und auslagern, suchen und ersetzen in 3 einstellbaren "Perfektionsstufen" vorhanden. Auch Makros können erstellt werden. Als besondere Features werden das Bearbeiten eines Bildausschnittes sowie ein Fenstersplitting bei derselben Datei sowie das gleichzeitige Öffnen und Bearbeiten mehrerer Textdateien geboten. Nahezu alle Zeichen des

erweiterten ASCII-Codes und damit alle deutschen Sonderzeichen sowie Rahmencharaktere sind implementiert. Daher können auch Textumrahmungen und Boxen hergestellt werden. Die Möglichkeit eines Mehrspaltensatzes habe ich hingegen nicht gefunden, wohl aber ist Mailmerge möglich. An Textdateiformaten können Wordstar, Wordperfect und DCA (IBM) importiert und Texte in ASCII, Wordperfect und DCA exportiert werden.

Eine ausführliche - natürlich englische - Online-Hilfe samt Index und Hypertext-Funktion ist für das Erlernen der zahlreichen Funktionen ohne Handbuch praktisch unentbehrlich. Für den Anfang ist aber auch ein Tutorprogramm verfügbar.

Setup-Programme ermöglichen eine weitgehende Anpassung der Textverarbeitung an eigene Wünsche (z.B. auch Farb- oder Monoschirm) und eine große Auswahl handelsüblicher Drucker, insbesondere auch Laserprinter mit PS-Funktion.

Gesamtbeurteilung

Empfehlenswerte, modern konzipierte Textverarbeitung für alle üblichen Aufgaben, jedoch ohne Rechtschreibprüfung und Silbentrennung (Blocksatz daher problematisch), wenn nicht spezifische Textbaustein- oder Desktop-Publishing-Anforderungen gestellt werden müssen.

Rubicon Publisher (EDU-1238, 1239)

Ein \$ 60-Sharewareprogramm von Rubicon Computer Labs Inc. (USA) im Umfang von ca 0,9 MB zur Layout-Gestaltung einfach editierter Texte. Das Programm besteht aus zwei Teilen: 1) einer Druckersprache für die Einfügung von Layout Anweisungen und 2) einem Interpreter zur Umsetzung dieser Anweisungen in entsprechende Drucksteuerbefehle des gewählten Druckers, wobei praktisch alle im Handel befindlichen Laser-, Tintenstrahl- und Matrixdrucker inklusive PS-fähige Drucker wählbar sind. Hervorzuheben sind neben den in jeder besseren Textverarbeitung verfügbaren Formatierungsmöglichkeiten Mehrspaltensatz und zahlreiche (PS-)Fonts; es fehlen aber einige in heutigen DTP-Programmen meist vorhandene Features wie z.B. schräge Überschriften.

Hier endet der Test bereits: Das Programm kennt leider nur die ASCII-Codes bis 127 und straft deutsche Umlaute und das ß mit Stillschweigen. Ein sinnvoller Einsatz bei uns scheint mir daher unmöglich.

THEDRAW V. 3.20 (EDU-881):

\$ 15-Sharewareprogramm von Soft Programming Services (Kanada) mit leicht eingeschränktem Funktionsumfang gegenüber der registrierten Version, aber vollständigem Handbuch (ca. 90 S.) auf Diskette. Gesamtumfang ca. 419 kB.

Als Weiterentwicklung des Programmes Ansi-Draw stellt Thedraw einen komfortablen Editor für Grafik-Bildschirme in Monochrom und Color mittels ASCII-Grafikzeichen dar. Auch die Möglichkeit (einfacherer) Animation wird geboten. Die Bedienung erfolgt durch Pull-Down-Menüs und optional die Maus, kann aber auch mit insgesamt 33 Tastenkombinations-Befehlen geschehen. Alle Zeichen des erweiterten ASCII-Codes werden erkannt. Die registrierte Version bietet gegenüber der Sharewareversion eine kontextsensitive Online-Hilfe und etwas

erweiterte Funktionen im Bereich der Animation. Besondere Features liegen in der Möglichkeit, bis zu 8 "transparente" Bildschirmentwürfe übereinander zu legen und Zeichnungen darauf zu "verschieben"; ferner die Entwürfe in den Formaten ANSI(=Color), ASCII(=Text), Binär, BSave(Basic), COM(Aufruf aus DOS-Batchfiles), Assembler, Turbo-Pascal(SourceCode in V.5.0), Turbo-C und programmspezifisch (Format "TheDraw") abzuspeichern und gegebenenfalls in selbst erstellte Programme bzw. Anwendungen einzubinden.

"Schnupper-"Urteil

Nettes und preiswertes Hilfsmittel, vor allem um für eigene Programme oder Anwendungen ohne sonst nötigen größeren Zeitaufwand Titel-Logos herzustellen.

InstaCalc V.3.00 (EDU 1183)

\$ 60-Shareware der US-Firma Formalsoft aus dem Jahr 1991. Der entpackte Programmumfang beträgt ohne Handbuch rund 413 kB, das etwa 120 Seiten umfassende, auf Diskette mitgelieferte Handbuch rund 390 kB. Nach Registrierung wird die neueste Version des Programmes und ein gedrucktes Handbuch geliefert.

Das mausfähige Programm wird mittels hierarchisch gegliederter, auf dem Bildschirm in etwas eigenwilliger Weise erscheinender Menüs gut beherrscht; es kann wahlweise auch speicherresident (TSR) installiert und mittels Hotkey aus anderen Programmen heraus aufgerufen werden.

Der Spreadsheet-Umfang beträgt 4096 Zeilen zu je 256 Spalten. Jede Zelle kann bis zu 240 Zeichen enthalten. Eine dritte Dimension (mehrere Kalkulationsblätter hintereinander) ist nicht implementiert. Zur Bearbeitung stehen insgesamt 100 Funktionen für mathematische, logische, Text- und andere Operationen (z.B. Such- und Ersetzfunktion) zur Verfügung. Eine Sortierung kann nach bis zu 9 Schlüsseln erfolgen. Import- und Exportfähigkeit für die Formate Lotus 1-2-3, dBase (bis III+), DIF und ASCII. Die (natürlich englische) kontextsensitive und interaktive Online-Hilfe mit Index ist wirklich gut. Selbstverständlich ist eine Zellformatierung sowie die Erstellung und der Gebrauch von Makros vorgesehen, wofür ein kleiner Editor vorhanden ist. Und: in Textfelder und im Editor können alle Zeichen des erweiterten ASCII-Codes, somit die deutschen Sonderzeichen eingegeben werden! Ein weiteres Feature ist die Möglichkeit, anhand der gespeicherten Daten Geschäftsgrafiken zu entwerfen. Auch das Drucken von definierten Blöcken aus dem Spreadsheet macht keine Schwierigkeiten.

Gesamtbeurteilung

Dieses mit einem Preis einer Fachzeitschrift ausgezeichnete Kalkulationsprogramm wird meines Erachtens innerhalb der oben erwähnten Leistungsgrenzen wohl selbst gehobenen Ansprüchen an ein solches Produkt gerecht und ist dabei im Vergleich zu kommerziell vertriebener Software recht preiswert.

Allen, die auf Grund dieser Zeilen eines der beschriebenen Produkte benutzen wollen, wünsche ich dabei viel Spaß. Sollte mir im Rahmen der doch nur kurzen Prüfungen der eine oder andere Bug oder aber auch das eine oder andere Highlight entgangen sein, bin ich für alle Mitteilungen dankbar! □

COSMIC-Software-Katalog

DSK-287

Wir haben für sie den aktuellen Software-Katalog der NASA in die Diskettensammlung aufgenommen. Er enthält ca. 900 Programme für die verschiedensten Rechner im Source-Code und natürlich auch für den IBM-PC. Da für unsere Leser hauptsächlich der PC-Bereich von Interesse sein wird, wurden diese Titel hier abgedruckt.

Wie kommt man zu diesen Programmen?

Wenn Sie sich für eines dieser Programme interessieren, können Sie über die TGM-Diskette-287 die gesamte Datenbank als dBase-DBF-Datei erhalten. Für jedes Programm ist in einem Memo-Feld eine Kurzbeschreibung enthalten. Auf dieser TGM-Diskette (und auf der Programmdiskette zu diesem Heft, DSK-294) ist auch das Bestellformular im PCX-Format enthalten; Bestellungen können auch über Kreditkarten (VISA, EUROCARD) bezahlt werden.

Wir haben dieses Verzeichnis schon einmal mit allen Suchbegriffen in den **PC-NEWS-3/91**, S.13 abgedruckt. Da sich an der Struktur nichts geändert hat, können interessierte Leser dort die Rechnerarten und die Suchbegriffe nachlesen. Die nachfolgende Liste enthält in dieser Reihenfolge: Titel, Sprache, Programmgröße, Speichermedium, Preis. Programme, die mit einem "*" gekennzeichnet sind sind neu dazugekommen.

A FAST POLYNOMIAL TRANSFORM PROGRAM WITH A MODULARIZED STRUCTURE, C-LANGUAGE, Approximately 783 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$100.00/Documentation \$32.00

ACTOG - AUTOCAD TO GIFTS TRANSLATOR, BASIC, Approximately 454 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$300.00/Documentation \$24.00

ACTOMP - AUTOCAD TO MASS PROPERTIES, BASIC, Approximately 485 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$300.00/Documentation \$24.00

AKLSQF - LEAST SQUARES CURVE FITTING, BASIC, Approximately 364 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$300.00/Documentation \$26.00

AKPLOT- A PLOTTER ROUTINE FOR THE IBM PC, BASIC, Approximately 1461 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$400.00/Documentation \$30.00

AKSATINT - SATELLITE INTERFERENCE ANALYSIS AND SIMULATION USING PERSONAL COMPUTERS, BASIC, Approximately 207 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$400.00/Documentation \$32.00

ALGORITHM FOR SORTING GROUPED DATA, BASIC, Approximately 708 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$100.00/Documentation \$24.00

* ALPS - A LINEAR PROGRAM SOLVER, APL2, Approximately 3418 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$400.00/Documentation \$32.00

* AMPS/PC - AUTOMATIC MANUFACTURING PROGRAMMING SYSTEM, PASCAL, Approximately 1758 source statements, 5.25 Inch

IBM PC DOS Format Diskette, Price: Program \$200.00/Documentation \$46.00

* ARAM - AUTOMATED RELIABILITY/AVAILABILITY/MAINTAINABILITY, VERSION 2.0, FORTRAN 77, Approximately 5229 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$800.00/Documentation \$46.00

ASAP- ARTIFICIAL SATELLITE ANALYSIS PROGRAM, FORTRAN 77, Approximately 2231 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$600.00/Documentation \$38.00

* ASCITOVG - FORTRAN PROGRAM FOR X-RAY PHOTOELECTRON SPECTROSCOPY DATA REFORMATTING, FORTRAN 77, Approximately 738 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$100.00/Documentation \$24.00

ASSESSMENT OF ADVANCED CONCENTRATOR PHOTOVOLTAIC MODULE TECHNOLOGIES, FORTRAN 77, Approximately 2247 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$600.00/Documentation \$54.00

* ATAM - AUTOMATED TRADE ASSESSMENT MODELING, VERSION 1.0, FORTRAN 77, Approximately 4351 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$700.00/Documentation \$36.00

BINOMIAL PROBABILITY DISTRIBUTION, BASIC, Approximately 610 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$300.00/Documentation \$22.00

BLAS- BASIC LINEAR ALGEBRA SUBPROGRAMS (IBM PC VERSION), FORTRAN 77(23%),ASSEMBLER(77%), Approximately 1669 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$600.00/Documentation \$26.00

BOMTOM - BUREAU OF MINES TOMOGRAPHY, FORTRAN 77, Approximately 1610 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$200.00/Documentation \$34.00

CERTLM- A TLM AND CER METHOD OF DETERMINING CONTACT RESISTIVITY, BASIC, IBM PC, Approximately 300 source statements, Listing Available Only, Price: Documentation \$100.00 (Document Includes Program Listing)

CES- CONSTANT ELASTICITY OF SUBSTITUTION PRODUCTION FUNCTION SIMULATION, BASIC, IBM PC, Approximately 1813 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$600.00/Documentation \$64.00

CLIPS - C LANGUAGE INTEGRATED PRODUCTION SYSTEM (IBM PC VERSION WITH CLIPSITS), C-LANGUAGE, Approximately 80820 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$600.00/Documentation \$124.00

* CLIPS 5.0 - C LANGUAGE INTEGRATED PRODUCTION SYSTEM, VERSION 5.0 (IBM PC VERSION), C-LANGUAGE, Approximately 110646 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$700.00/Documentation \$280.00

CLIPSITS - CLIPS INTELLIGENT TUTORING SYSTEM, C-LANGUAGE, N/A, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$200.00

COMPsize - PRELIMINARY DESIGN METHOD FOR FIBER REINFORCED COMPOSITE STRUCTURES, BASIC, Approximately 2038 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$700.00/Documentation \$38.00

* CONFIDENCE LIMITS PROGRAM FOR THE IBM PC, BASIC, Approximately 93 source statements, Listing Available Only, Price: Documentation \$100.00 (Document Includes Program Listing)

CROSSER - CUMULATIVE BINOMIAL PROGRAMS, C-LANGUAGE, Approximately 144 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$300.00/Documentation \$30.00

CTLM- CIRCULAR TLM CONTACT RESISTIVITY PROGRAM, BASIC, IBM PC, Approximately 242 source statements, Listing Available Only, Price: Documentation \$100.00 (Document Includes Program Listing)

CUMBIN - CUMULATIVE BINOMIAL PROGRAMS, C-LANGUAGE, Approximately 89 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$300.00/Documentation \$30.00

CUMPOIS- CUMULATIVE POISSON DISTRIBUTION PROGRAM, C-LANGUAGE, Approximately 114 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$300.00/Documentation \$22.00

DISTRIBUTED PROCESSING TRADE-OFF MODEL FOR ELECTRIC UTILITY OPERATION, LOTUS 1-2-3, Approximately 450 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$100.00/Documentation \$90.00

EGRAM- ECHELLE SPECTROGRAPH DESIGN AID (IBM PC VERSION), BASIC, Approximately 794 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$400.00/Documentation \$36.00

EMA- EISENBERGER-MAIOCCO ALGORITHM FOR SPARES PROVISIONING, BASIC, Approximately 650 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$100.00/Documentation \$50.00

EXADS- EXPERT SYSTEM FOR AUTOMATED DESIGN SYNTHESIS (IBM PC VERSION), LISP, Approximately 2136 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$600.00/Documentation \$28.00

FLUID- THERMODYNAMIC AND TRANSPORT PROPERTIES OF FLUIDS (IBM PC VERSION), FORTRAN 77, Approximately 10000 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$1000.00/Documentation \$30.00

GASXFER- ISOTHERMAL GAS TRANSFER PROGRAM, LOTUS SYMPHONY, Approximately 105 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$300.00/Documentation \$24.00

GENERAL THERMAL ANALYZER (IBM PC VERSION), BASIC, Approximately 524 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$100.00/Documentation \$32.00

* IMDISP - INTERACTIVE IMAGE DISPLAY PROGRAM, C-LANGUAGE(94%),ASSEMBLER(6%), Approximately 8240 source

statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$800.00/Documentation \$42.00

IPEG- IMPROVED PRICE ESTIMATION GUIDELINES (IBM PC VERSION), PASCAL, IBM PC, Approximately 2491 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$400.00/Documentation \$114.00

KINETIC MODELER - A COMPUTER PROGRAM FOR MODELING THE KINETICS OF GENE EXPRESSION (IBM PC VERSION), BASIC, Approximately 142 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$50.00"

* LABORATORY PROCESS CONTROLLER USING NATURAL LANGUAGE COMMANDS FROM A PERSONAL COMPUTER, FORTRAN 77(74%),PASCAL(21%),ASSEMBLER(5%), Approximately 3350 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$400.00/Documentation \$24.00

* LERC-SLAM - THE NASA LEWIS RESEARCH CENTER SATELLITE LINK ATTENUATION MODEL PROGRAM (IBM PC VERSION), BASIC, Approximately 817 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$100.00/Documentation \$30.00

LMSS SERVICES FINANCIAL REPORT PROGRAM, MULTIPLAN, IBM PC, Approximately 1000 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$200.00/Documentation \$44.00

LOOP- SIMULATION OF THE AUTOMATIC FREQUENCY CONTROL SUBSYSTEM OF A DIFFERENTIAL MINIMUM SHIFT KEYING RECEIVER, FORTRAN 77, IBM PC, Approximately 120 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$100.00/Documentation \$34.00

LOP- LONG-TERM ORBIT PREDICTOR, FORTRAN 77, Approximately 3500 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$800.00/Documentation \$56.00

MAILROOM- A LOCAL AREA NETWORK ELECTRONIC MAIL PROGRAM, BASIC, Approximately 405 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$300.00/Documentation \$30.00

MATHEMATICAL ROUTINES FOR ENGINEERS AND SCIENTISTS, FORTRAN 77, Approximately 828 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$400.00/Documentation \$32.00

* METALLURGICAL PROGRAMS: CALCULATION OF MASS FROM VOLUME, DENSITY OF MIXTURES, AND CONVERSION OF ATOMIC TO WEIGHT PERCENT, BASIC, Approximately 56 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$100.00/Documentation \$20.00

MIMS - MEDICAL INFORMATION MANAGEMENT SYSTEM, FORTRAN 77, Approximately 5078 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$600.00/Documentation \$36.00

MIP- MULTIMISSION INTERACTIVE PICTURE PLANNING PROGRAM, FORTRAN 77, Approximately 8851 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$800.00/Documentation \$32.00

MMW- MISSION MANAGER'S WORKSTATION, PASCAL(64%),BASIC(19%),ASSEMBLER(17%), Approximately

15000 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$2000.00/Documentation \$134.00

MODEL- ELECTROMAGNETIC FIELDS INDUCED BY A LOOP ANTENNA, FORTRAN 77, Approximately 760 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$200.00/Documentation \$26.00

NEWTONP - CUMULATIVE BINOMIAL PROGRAMS, C-LANGUAGE, Approximately 147 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$300.00/Documentation \$30.00

NEWTPOIS- NEWTON POISSON DISTRIBUTION PROGRAM, C-LANGUAGE, Approximately 141 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$300.00/Documentation \$22.00

OAP- OFFICE AUTOMATION PILOT GRAPHICS DATABASE SYSTEM, PASCAL(99%),ASSEMBLER(1%), IBM PC, Approximately 31200 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$2000.00/Documentation \$168.00

OPTI - OPTICAL COMMUNICATIONS LINK ANALYSIS PROGRAM, FORTRAN 77, Approximately 2809 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$600.00/Documentation \$30.00

OPTIMAL NETWORK TOPOLOGY DESIGN, PASCAL, Approximately 520 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$100.00/Documentation \$34.00

PC-DADS- DATA ACQUISITION DISPLAY AND STORAGE SYSTEM, BASIC, Approximately 108 source statements, Listing Available Only, Price: Documentation \$100.00 (Document Includes Program Listing)

PC-SEAPAK - ANALYSIS OF COASTAL ZONE COLOR SCANNER AND ADVANCED VERY HIGH RESOLUTION RADIOMETER DATA, FORTRAN 77(0%),C-LANGUAGE(0%), N/A, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$900.00/Documentation \$100.00

PCACE- PERSONAL COMPUTER AIDED CABLING ENGINEERING (IBM PC VERSION), PASCAL, Approximately 1355 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$200.00/Documentation \$24.00

RESISTIVITY OF A THIN FILM DEPOSITED ON A CONDUCTIVE SUBSTRATE, FORTRAN 77, Approximately 769 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$400.00/Documentation \$30.00

ROTRAN 1 - SOLUTION OF EQUATIONS FOR ROTARY TRANSFORMERS, BASIC, Approximately 25 source statements, Listing Available Only, Price: Documentation \$100.00 (Document Includes Program Listing)

* S-CHART - SCHEDULING CHART PROGRAM, DBASE III PLUS, Approximately 1144 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$200.00/Documentation \$24.00

SAMIS- STANDARD ASSEMBLY-LINE MANUFACTURING INDUSTRY SIMULATION (IBM PC VERSION), PASCAL, Approximately 30000 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$1000.00/Documentation \$142.00

SCAT- SYSTEM COMMONALITY ANALYSIS TOOL (IBM PC VERSION), FORTRAN 77, Approximately 8427 source statements,

5.25 Inch IBM PC DOS Format Diskette, Price: Program \$1200.00/Documentation \$74.00

SIMPLIFIED CALCULATION OF SOLAR FLUX ON THE SIDE WALL OF CYLINDRICAL CAVITY SOLAR RECEIVERS, FORTRAN 77, Approximately 364 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$400.00/Documentation \$24.00

SIMRAND I- SIMULATION OF RESEARCH AND DEVELOPMENT PROJECTS, FORTRAN 77, Approximately 3146 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$600.00/Documentation \$50.00

SINW- STELLAR INERTIAL NAVIGATION WORKSTATION, BASIC(0%),PASCAL(0%), N/A, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$900.00/Documentation \$88.00

SMDOS- SHUTTLE MISSION DESIGN AND OPERATIONS SOFTWARE, FORTH, IBM PC/XT, Approximately 3675 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$800.00/Documentation \$44.00

SNIP - SINDA-NASTRAN INTERFACING PROGRAM, FORTRAN 77, Approximately 1012 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$200.00/Documentation \$36.00

SOFTCOST - DEEP SPACE NETWORK SOFTWARE COST MODEL, BASIC, Approximately 1356 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$200.00/Documentation \$52.00

SOSPAC- SOLAR SPACE POWER ANALYSIS CODE, FORTRAN IV, Approximately 640 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$200.00/Documentation \$50.00

SSINAP- SPACE STATION INTERIOR NOISE ANALYSIS PROGRAM, FORTRAN 77, Approximately 4917 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$500.00/Documentation \$40.00

STACOM- SELECTED TETHER APPLICATIONS COST MODEL, LOTUS 1-2-3, Approximately 3000 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$900.00/Documentation \$52.00

STEADY- A STEADY STATE THERMAL ANALYSIS PROGRAM FOR MICROCOMPUTERS, FORTRAN 77, Approximately 1024 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$200.00/Documentation \$26.00

TFMOD- THIN FILM MODULE DESIGN, BASIC, IBM PC, Approximately 626 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$400.00/Documentation \$24.00

* TRACER - TRACING AND CONTROL OF ENGINEERING REQUIREMENTS, CLIPPER, Approximately 20741 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$1000.00/Documentation \$70.00

* WINDY - PREDICTION OF BROADBAND NOISE FROM HORIZONTAL AXIS WIND TURBINES, BASIC, Approximately 616 source statements, 5.25 Inch IBM PC DOS Format Diskette, Price: Program \$400.00/Documentation \$28.00. □

dBase-Befehle, Version 4.1

!	Führt MS-DOS-Befehl aus und kehrt anschließend zu dBase zurück	copy sdf	kopiert Datei(teile) in SDF-Format
.	Beginn einer Kommentarzeile	copy file	kopiert komplette Dateien
?	Ausgabe von Ausdrucken	copy structure	überträgt Datenstruktur
??	Ausgabe von Ausdrucken ohne Zeilenvorschub	count	gibt Anzahl von Datensätzen einer Datei aus
@..say	koordinatengesteuerte Ausgabe eines Ausdrucks oder einer Variablen (Anzeige)	create	Neuanlegen einer Datenbank
@..get	koordinatengesteuerte Benutzerabfrage einer Variablen (Eingabe)	create from	wie create doch wird Struktur von bestehender Datei übernommen
accept	schreibt Informationen vom Benutzer in eine Zeichenkette	create label	Neuanlegen einer Labeldatei
append	Anhängen von Datensätzen an bestehende Datei	create screen	Neuanlegen einer Screendatei
append blank	Anhängen eines leeren Datensatzes an eine Datenbank	create query	Neuanlegen einer Querydatei
append from	Anhängen ausgewählter Daten aus einer anderen Datei an aktive Datei	create report	Neuanlegen einer Reportdatei
assist	bildschirmorientierte Benutzeroberfläche (Menü)	create view	Neuanlegen einer Viewdatei
average	berechnet bei einer Datei Durchschnitte von Ausdrucken	delete	Löschmarkierung von Datensätzen einschalten
browse	Bildschirmanzeige, einfaches Editieren und Erweitern einer Datenbank	delete file	löscht Datei
cancel	schaltet von Programm- in Dialogmodus um	dir	listet Disketteninhalt auf (nur .DBF-Dateien)
change	einfaches Editieren bestimmter Felder	directory display	listet Disketteninhalt auf zeigt Inhalt von Dateien an (nur einen Satz)
clear	löscht Bildschirm	display memory	listet den aktuellen Variablenspeicher auf
clear	schließt alle Dateien löscht alle Variablen und wählt Bereich 1 aus	display status	listet Arbeitsstatus auf
clear all	schließt alle Dateien löscht alle Variablen und wählt Bereich 1 aus	display structure	zeigt Datenbankstruktur an
clear gets	alle vorstehenden @.. get-Anweisungen werden deaktiviert	do	startet Befehlsdatei
close alternate	schließt aktive Protokolldateien	do case	startet Fallunterscheidungsprozedur
close databases	schließt aktive Datenbankdateien	do while	startet Befehlsblockausführung bis Bedingung falsch wird
close format	schließt aktive Formatdateien	edit	Editieren eines Datensatzes
close index	schließt aktive Indexdateien	eject	Drucker führt Seitenvorschub aus
close procedure	schließt aktive Prozedurendateien	endcase	beendet Fallunterscheidungsprozedur
continue	setzt locate-Suche nach Erfolg fort	enddo	schließt Befehlsblock nach .do while ab
copy	kopiert Dateien oder Datensätze unter festlegbaren Bedingungen in eine andere Datei	endif	schließt if-Operation ab
		endtext	beendet Ausgabe eines Textblocks
		erase	löscht den Bildschirm
		erase	löscht Datei
		exit	verläßt do while-Schleife
		end	durchsucht Indexdatei nach angegebenem Kriterium
		go	verschiebt Datensatzzeiger an bestimmte Position
		goto	wie go
		help	Aufruf der dBase-Hilfsbildschirme
		if	vergleicht Terme und führt bei bestimmten Ergebnissen bestimmte Befehlsblöcke aus

index	legt eine Indexdatei an, die eine geordnete Form der Satznummern einer Datei enthält	read	get-Informationen werden in Variablen eingelesen
input	liest Informationen vom Benutzer in numerische Variable ein	recall	Hebt Löschmarkierungen auf
insert	fügt Datensatz-sätze in Datenbank ein	reindex	Neuindizierung einer Indexdatei
insert blank	fügt leeren Datensatz in Datenbank ein	release	löscht lokale Speichervariablen
insert before	fügt Datensatz-sätze vor einer angegebenen Position in Datenbank ein	remark	gibt unbegrenzte Zeichenkette aus
join	legt neue Datei an und übernimmt bestimmte Datensätze aus existierender Datei	rename	ändert Dateinamen
label	druckt Etiketten aus	replace	ändert Inhalt von Feldern
list	zeigt Inhalt von Dateien an (alle Sätze)	report	Datenausgabe anhand von Reportdatei-Informationen
list files	listet Disketteninhaltsverzeichnis auf	restore	liest Speichervariablen aus einer .mem-Datei ein
list memory	listet Variablenspeicher auf	return	beendet Unterprogramm
list status	listet Arbeitsstatus auf	return nun	beendet Prozedur
list structure	zeigt Datenbankstruktur an	save	führt Programm aus und kehrt anschließend zu dBase zurück
locate	durchsucht Datei nach angegebenem Kriterium	seek	erzeugt .mem-Datei und speichert bestimmte Variablen
loop	verschiebt Befehlszeilenzeiger an den Beginn einer .do while.-Schleife	select	durchsucht Indexdatei nach angegebenem Kriterium
modify command	Editieren einer Befehlsdatei	select (1bis 10)	Wahl zwischen Arbeitsbereichen (1bis 10)
modify label	Editieren einer Labeldatei	set	setzt dBase-Schalter
modify query	Editieren einer Querydatei	set to	definiert Parameter
modify report	Editieren einer Reportdatei	skip	verschiebt Datensatzzeiger um die angegebene Anzahl von Datensätzen
modify screen	Editieren einer Screendatei	sort	legt aktuelle Datei in sortierter Form neu an
modify structure	Editieren einer Datenbankstruktur	store	belegt Speichervariablen
modify view	Editieren einer Viewdatei	sum	errechnet Summe eines Ausdrucks mit dem Inhalt von numerischen Feldern
note	Beginn einer Kommentarzeile	text	Textblockausgabe auf Bildschirm
pack	eliminiert zum Löschen markierte Datensätze	total	Summe der Felder einer Datei wird in Zieldatei gespeichert
parameters	legt bei Programmaufruf die Belegung von Variablen fest	update	Aktualisierung der aktiven Datei anhand einer anderen Datei
private	definiert Variablen in Unterprogrammen als lokal	use	nimmt Datenbank in Gebrauch
procedure	Beginn einer Prozedur	wait	wartet auf Tastendruck und speichert diesen nach Wahl in einer Variablen ab
public	definiert Variablen als global	zap	eliminiert alle Datensätze einer Datei □
quit	verläßt dBase und kehrt in die Betriebssystemebene zurück		

DOS-6.0, so könnte es werden

Fritz Zetik, 3AN, N, TGM

MS-DOS-Entwicklungsgeschichte, was bisher geschah

Datum	Version	Wichtigste neue Funktionen
August 1981	1.0	Erste Version für den IBM PC
Mai 1982	1.1	Unterstützung für doppelseitige Diskettenlaufwerke
März 1983	2.0	Unterstützung für hierarchische Dateisysteme und Festplatten
Oktober 1983	2.1	Einführung von PC DOS für den PCjr
August 1984	3.0	Unterstützung für 1,2-MB-Diskettenlaufwerke, AT
März 1985	3.1	Unterstützung für Microsoft Netzwerke
Dezember 1985	3.2	Unterstützung für 3,5-Zoll-Laufwerke
April 1987	3.3	Unterstützung für mehrere Partitionen, verbesserte Unterstützung für Zeichensätze in anderen Landessprachen
November 1988	4.0	Unterstützung für Festplattenpartitionen mit mehr als 32 MB, Shell, EMS-Unterstützung
Juni 1991	5.0	MS-DOS-Betriebssystemkern wird in den HMA geladen, 80386-Hauptspeicherverwaltung, neue Shell, Programm-Umschaltung, Online-Hilfe, menügesteuerter Editor. Verfügbar als Update.

Neues, kurz gefaßt

Codeseitenunterstützung für Österreich (061). Der Anwender bekommt nun häufig benötigte Standardsoftware mitgeliefert: **Central Point Antivirus:** Gutes Antivirusprogramm; von Microsoft wurde die Benutzeroberfläche auf das Nötigste beschränkt. **Central Point Undelete.** Seit MS-DOS 5 dabei, jetzt auch mit Benutzeroberfläche! Alle Programme von Central Point liegen auch in der Windows Version vor. **Peter Norton** spendierte sein **SPEED DISK & BACKUP** für DOS. Microsoft hat ein eigenes **Netzwerkprogramm** entwickelt, und bietet auf der 5. Diskette eine Benutzeroberfläche für Netzwerke. MS-DOS hat noch immer die 640K-Grenze und das FAT-System. Es ist aber jetzt voll auf Benutzung mit Windows ausgelegt, d.h. die Treiber werden für MS-WINDOWS optimiert. Ein Verzeichnis aller Befehle ist wegen des unmittelbar nach diesem Beitrag folgenden Befehlssatzes der WindowsNT-'DOS-BOX' am Ende dieses Beitrags zu finden. Hier zunächst die Dateiliste:

ANSI SYS 9029 08-07-92	ANSI-Einheitentreiber	HELP HLP 209353 08-07-92	
APPEND EXE 10774 08-07-92	Append	HELP COM 413 08-07-92	Umfangreiche Hilfe in
APPNOTES TXT 9058 08-07-92	Mitteilungen von Microsoft		QBASIC - Benutzeroberfläche
ATTRIB EXE 15796 08-07-92	Attribute ändern	HIMEM SYS 13984 08-07-92	Neuer HIMEM-Treiber
CHKDSK EXE 16200 08-07-92	Chkdsk	INTERLNK EXE 17133 08-07-92	Netzwerkprogramm Terminal
CHOICE COM 1734 08-07-92	Setzt Errorlevel	INTERSVR EXE 37266 08-07-92	Netzwerkprogramm Server
	in Batch-Dateien	KEYB COM 15360 08-07-92	
COMMAND COM 51200 08-07-92	Befehlsinterpreter	KEYBOARD SYS 34816 08-07-92	
COUNTRY SYS 17408 08-07-92	Ländertreiber mit 061	LABEL EXE 9390 08-07-92	
	für Österreich	LOADFIX COM 1131 08-07-92	
DATAMON EXE 91193 08-07-92		MEM EXE 30406 08-07-92	Zeigt jetzt auch die
DATAMON HLP 9291 08-07-92			Verteilung LOW - HIGH Mem an.
DEBUG EXE 20634 08-07-92	Daten Entfehlerungsprogramm	MIRROR COM 18201 07-29-92	bekannt von CP
	mit Benutzeroberfläche einer	MODE COM 2357 08-07-92	
	ungeschälten Gurke (hi)	MORE COM 2618 08-07-92	
	bekannt unter Speed-Disk	MSAV EXE 170127 08-07-92	bekannt auch unter
DEFRAG HLP 17574 08-07-92	von Peter Norton		Central Point Anti-Virus
		MSAVW EXE 270841 08-07-92	d.s. unter Windows
DEFRAG EXE 71168 08-07-92		MSD EXE 155538 08-07-92	Microsoft Diagnostics
DISKCOMP COM 10636 08-07-92	Diskcomp	NLSFUNC EXE 7168 08-07-92	
DISKCOPY COM 11879 08-07-92	Diskcopy	POWER EXE 7924 08-07-92	für Laptops
DISPLAY SYS 15792 08-07-92	Displaytreiber	PRINT EXE 15656 08-07-92	
DOSBACK EXE 5327 08-07-92	Backupprogramm	QBASIC EXE 255925 08-07-92	
	von Peter Norton	RAMDRIVE SYS 5873 08-07-92	
DOSHELP EXE 11481 08-07-92	Bei MS-DOS 5 ->HELP	README TXT 27460 08-07-92	
DOSHELP HLP 5743 08-07-92		REPLACE EXE 20226 08-07-92	
DOSKEY COM 5883 08-07-92	Doskey	RESTORE EXE 38294 08-07-92	
DOSHELL EXE 236394 08-07-92	Dosshell	SETVER EXE 12015 08-07-92	
DRIVER SYS 5409 08-07-92		SHARE EXE 10912 08-07-92	
EDIT HLP 17898 08-07-92		SMARTDRV EXE 43609 08-07-92	
EDIT COM 413 08-07-92		SORT EXE 6938 08-07-92	
EGA SYS 4885 08-07-92		SUBST EXE 18478 08-07-92	
EGA CPI 58873 08-07-92		SYS COM 13440 08-07-92	
EMM386 EXE 114782 08-07-92		TREE COM 6901 08-07-92	
EXPAND EXE 16896 08-07-92			
PASTOPEN EXE 12050 08-07-92	Filecompare	UNDEL EXE 132348 07-29-92	benutzerfreundliches
FC EXE 18650 08-07-92	FDISK		Undelete CP
FDISK EXE 57344 08-07-92	Suchprogramm (würg!)	UNDELETE EXE 13924 07-29-92	DOS-Undelete CP
FIND EXE 6770 08-07-92	FORMAT	UNFORMAT COM 18944 08-07-92	s.o.
FORMAT COM 33280 08-07-92		VSAFE COM 61824 08-07-92	residentes
GRAPHICS COM 19694 08-07-92			Virusschutzprogramm
GRAPHICS PRO 21232 08-07-92		WINBACK EXE 306448 08-07-92	Backup unter Windows Norton
		WNTSRMAN EXE 17296 08-07-92	TSR Manager für Windows
		WUNDEL EXE 129792 08-07-92	Undelete unter Windows
		XCOPY EXE 15820 08-07-92	XCOPY

Auszüge aus der *Microsoft Dokumentation*:

Die Zukunft von MS-DOS

MS-DOS ist inzwischen seit fast 12 Jahren auf dem Markt und wird von mehr als 100 Millionen Benutzern eingesetzt. Seit seiner Markteinführung im Jahre 1981 hat sich die Welt der Datenverarbeitung in dramatischer Weise weiterentwickelt, und trotzdem ist MS-DOS nach wie vor ein entwicklungsfähiges und erfolgreiches Betriebssystem. Auf unserem Weg in die Zukunft werden uns drei grundlegende Benutzeranforderungen als Basis für unsere Zukunftspläne für MS-DOS dienen:

MS-DOS muß eine herausragende Plattform für Windows sein

Windows ist zu einem Standard geworden. Mehr als die Hälfte aller neuen PCs werden mit Windows ausgeliefert, und diese Zahl steigt weiter an. Der Markt für Windows-Anwendungen nähert sich der Größe des Marktes für MS-DOS-Anwendungen. Angesichts des so verbreiteten Einsatzes von Windows planen wir folgende Entwicklung für MS-DOS:

- MS-DOS wird die Basistechnologie sein, die Windows benötigt, um noch leistungsfähiger zu werden.
- MS-DOS wird noch enger in Windows integriert.

MS-DOS muß für MS-DOS-Benutzer noch besser werden

Trotz des Erfolges von Windows wird es immer Benutzer geben, die aus einer Vielzahl unterschiedlicher Gründe auch in der Zukunft nur MS-DOS einsetzen möchten. Viele Benutzer sind beispielsweise mit MS-DOS und MS-DOS-Anwendungen sehr vertraut und vermissen nichts, sie setzen Branchenlösungen ein, die noch nicht auf Windows portiert worden sind, oder sie verwenden einen Computer, der Windows nicht unterstützt. Microsoft wird MS-DOS weiter verbessern, und MS-DOS-Benutzern eine umfassendere und verbesserte Funktionalität zur Verfügung stellen und dabei immer abwärtskompatibel bleiben. Wir verpflichten uns, MS-DOS weiter zu unterstützen. MS-DOS 5, MS-DOS 6 und die parallele Entwicklung der nächsten, höheren Version beweisen unsere Verpflichtung.

MS-DOS muß sicher und bedienerfreundlich sein

Die meisten Computerbenutzer sind keine Experten, und die Anzahl der MS-DOS-Benutzer ist sehr groß. Aus diesem Grund ist es unerlässlich, daß neue Funktionen sicher sind, so daß der Benutzer keine Angst vor Datenverlust, Systemfehlern oder auch eigener Fehlbedienung haben muß. Gleichzeitig müssen neue Funktionen so bedienerfreundlich sein, daß sie nicht nur von Experten effektiv genutzt werden können.

Diese drei Ziele - Unterstützung von Windows, Verbesserung der Funktionalität von MS-DOS sowie Sicherheit und Bedienerfreundlichkeit - sind überall in MS-DOS 6 verwirklicht und werden auch in allen zukünftigen Versionen berücksichtigt werden.

Wir lernen von unseren Kunden

Vor MS-DOS 5 hat Microsoft MS-DOS ausschließlich über PC-Hersteller vertrieben. Aus diesem Grund waren MS-DOS-Verbesserungen eher Hardware-orientiert und auf die Bedürfnisse dieser Kunden ausgerichtet: Unterstützung für Diskettenlaufwerke mit hoher Dichte, Unterstützung für extrem große Festplattenpartitionen usw.

Mit der Ankündigung von MS-DOS 5 und besonders das MS-DOS 5 Update machten wir die Erfahrung, ein MS-DOS-Produkt direkt an den Endbenutzer zu verkaufen. Wir sprachen Benutzergruppen und andere Kunden an, führten Marktforschungen durch und erhielten sehr viel Resonanz auf unseren Software-Service. Auf diese Weise haben wir in den letzten anderthalb Jahren viel darüber gelernt, was Benutzer von MS-DOS erwarten, und wir haben dieses Wissen in MS-DOS 6 berücksichtigt.

Was kann der Benutzer von MS-DOS erwarten?

Aufgrund unserer Erfahrungen durch den Verkauf von MS-DOS über PC-Hersteller und durch den Verkauf vom Update wollen wir die folgenden MS-DOS-Testkriterien zur Verfügung stellen. Diese Kriterien sind aufgabenorientiert, d.h. sie basieren auf den Aufgaben, die der Benutzer normalerweise mit MS-DOS ausführen möchte. Obwohl Kompatibilität und Windows-Unterstützung keine Aufgaben in diesem Sinne sind, sind sie doch ganz deutlich wichtige Kriterien eines beliebigen PC-Betriebssystems. Wir glauben außerdem, daß die Funktionen innerhalb jedes dieser Bereiche an drei Punkten gemessen werden müssen:

- **Sicherheit:** Der Benutzer erwartet von MS-DOS höchstmögliche Verlässlichkeit und Kompatibilität
- **Bedienerfreundlichkeit:** Alle MS-DOS-Funktionen müssen einer Vielzahl von Benutzern zugänglich sein
- **Flexibilität:** Fortgeschrittene Benutzer müssen die Möglichkeit haben, MS-DOS nach ihren Wünschen konfigurieren und optimieren zu können

Kompatibilität

Der Benutzer verläßt sich darauf, Zehntausende von Anwendungsprogrammen unter MS-DOS ausführen zu können. Er erwartet, daß nicht nur Microsoft umfassende Kompatibilitätstests mit Anwendungsprodukten durchführt, sondern daß auch andere Software- und Hardwarehändler prüfen, ob ihre Produkte MS-DOS-kompatibel sind. Alles in allem ist Kompatibilität das, was MS-DOS kennzeichnet. Nur bei hundertprozentiger Kompatibilität ist es auch MS-DOS.

Windows-Unterstützung

MS-DOS muß mehr können, als nur das Ausführen von Windows zu ermöglichen. MS-DOS muß die Technologie zur Verfügung stellen, die es ermöglicht, Windows so optimal wie möglich auszuführen. MS-DOS muß in der Lage sein, sowohl Benutzern, die nur MS-DOS verwenden, als auch Benutzern die Windows einsetzen, alle Vorteile der jeweiligen Umgebung verfügbar zu machen.

Das Installationsprogramm

Das Installationsprogramm SETUP muß so bedienerfreundlich sein, daß jeder Benutzer in der Lage ist, es einzusetzen. Dabei handelt es sich nicht einfach nur um eine durchdachte Benutzerführung, sondern auch um ausgefeilte Hardware- und Software-Erkennung. Der Installationsvorgang muß darüber hinaus sicher sein, und der Benutzer muß die Möglichkeit haben, nachträglich Änderungen vorzunehmen oder nach Unfällen, wie z.B. einem Stromausfall oder einem versehentlichen Ziehen eines Steckers, neu zu installieren. Zu guter Letzt muß auch die Flexibilität gewährleistet sein.

Fortgeschrittene Benutzer müssen in der Lage sein, die Konfiguration an ihre Bedürfnisse anzupassen.

Festplattenverwaltung

Kapazität und Geschwindigkeit von Festplatten sind leistungsbeeinflussende Faktoren. MS-DOS muß sichere und einfache Lösungen bieten, die dem Benutzer erlauben, sein System so effektiv wie möglich einzusetzen. Sicherheit ist aufgrund der Bedeutung von Platteninhalten besonders wichtig. MS-DOS muß darüber hinaus Informationen bereitstellen, die es fortgeschrittenen Benutzern erlauben, ihr System zu optimieren.

Speicherverwaltung

Der Hauptspeicher ist eine weitere wertvolle Ressource, die MS-DOS optimal verwalten muß. Da die Speicherverwaltung sowohl eine risikoreiche als auch komplexe Aufgabe ist, sind Sicherheit und Automatisierung wichtige Aspekte. Trotz aufgrund der Vielzahl unterschiedlicher Konfigurierungsmöglichkeiten muß jedoch die Flexibilität erhalten bleiben.

Arbeiten in Arbeitsgruppen (Workgroups)

Der Benutzer erwartet, daß er mit seinem Betriebssystem auch Ressourcen außerhalb seines Arbeitsplatzrechners nutzen kann. MS-DOS muß nicht nur kompatibel zu Netzwerk-Betriebssystemen sein, sondern darüber hinaus auch integrierten Netzwerk-Zugriff ermöglichen.

Datenschutz

Daten können aufgrund von Fehlbedienung, eines Systemfehlers, von Computerviren oder Problemen mit Anwendungsprogrammen gelöscht werden oder verlorengehen. MS-DOS muß dem Benutzer Hilfsmittel zur Verfügung stellen, die die Unversehrtheit seiner Daten gewährleisten.

Systemstart

Das Starten des Systems ist eine der am häufigsten ausgeführten Aufgaben. Die meisten Benutzer starten ihr System mindestens einmal täglich, viele Benutzer starten es mehrmals täglich. Der Startvorgang muß daher sowohl sicher als auch flexibel sein.

Dokumentation und Hilfe

Die Dokumentation muß übersichtlich und genau sein. Online-Hilfe muß in allen Teilen des Systems verfügbar sein.

Verbesserungen in MS-DOS 6

Im nachfolgenden Abschnitt sind die wichtigsten Verbesserungen in MS-DOS 6 aufgeführt. Die Verbesserungen sind in den neun Gruppen zusammengefaßt, die unter "Was kann der Benutzer von MS-DOS erwarten?" genannt wurden.

Kompatibilität

Ein Betriebssystem kann nicht nur im Hinblick auf Funktionalität verbessert werden. Verbesserungen hinsichtlich von Testmethoden sind ebenso wichtig, da Kompatibilität für Betriebssystembenutzer ein sehr wesentlicher Faktor ist. Der MS-DOS 5 Beta-Test konnte Kompatibilität gewährleisten, und zusammen mit MS-DOS 6 haben wir neue Möglichkeiten zur Verbesserung der Qualität unserer Beta-Tests eingeführt. Dazu gehören u.a.:

- **Beta-Implementierung:** Während wir für MS-DOS 5 denselben Testern in Abständen von einigen Monaten neue Beta-Versionen

zugeschickt haben, schicken wir jetzt in Abständen von wenigen Wochen immer neue Beta-Versionen an neue Tester. So können wir ständig testen und behobene Fehler erneut testen lassen. Für MS-DOS 5 wurden drei Beta-Versionen an mehr als 7.000 Beta-Tester verschickt; für MS-DOS 6 werden mindestens sechs Versionen an weltweit mehr als 10.000 Beta-Tester gehen.

- **Testanleitungen:** Alle Beta-Tester erhalten eine Testanleitung, anhand derer sie sich mit MS-DOS 6 vertraut machen können. Die Testanleitung gewährleistet darüber hinaus, daß alle neuen Funktionen ausgetestet werden.
- **Tester aller Kenntnisstufen:** Wir bemühen uns darum, sowohl Neulinge als auch erfahrene Beta-Tester einzusetzen, um sowohl Bedienbarkeit als auch Kompatibilität testen zu können.

Das Installationsprogramm

Die Sicherheit und Bedienerfreundlichkeit des Installationsprogramms für das MS-DOS 5 Update erwies sich als wichtige Funktion, da sie Benutzern aller Kenntnisstufen ermöglichte, ihre MS-DOS-Version zu aktualisieren. Auf dieser Basis macht das Installationsprogramm für das MS-DOS 6 Update das Aktualisieren und Starten des Systems noch leichter:

- **Einfache Installation:** Ausgereifte Konfigurationserkennungsfunktionen erlauben dem Benutzer durch Drücken einiger weniger Tasten auf MS-DOS 6 zu aktualisieren, ohne das System kennen zu müssen.
- **Flexible Installation:** Erlaubt dem Benutzer zu wählen, welche der wahlfreien MS-DOS 6-Komponenten installiert werden sollen, und welche nicht.
- **Installation im Wartungsmodus:** Erlaubt dem Benutzer das Installieren der wahlfreien Komponenten zu einem späteren Zeitpunkt, ohne daß MS-DOS 6 ganz neu installiert werden muß.
- **Automatische Cache-Installation:** Erspart dem Benutzer das Installieren von SMARTDRV, dem MS-DOS-Cache-Speicher. (Weitere Informationen zu SMARTDRV finden Sie weiter unten unter "Festplattenverwaltung".)
- **Automatisches Aktualisieren von komprimierten Systemen:** Aktualisiert Systeme, die unter Verwendung von zusätzlich erworbenen Komprimierungsprogrammen, wie z.B. Stacker oder Superstor, komprimiert worden sind, auf MS-DOS 6, ohne daß der Benutzer eingreifen muß. (Hinweis: Der Benutzer kann darüber hinaus nach dem Installieren des MS-DOS 6 Updates Stacker-Systeme in DoubleSpace, das in MS-DOS 6 integrierte Komprimierungsprogramm, konvertieren. Nähere Informationen finden Sie weiter unten unter "Festplattenverwaltung".)
- **OS/2-Aktualisierung:** Wenn erkannt wird, daß OS/2 installiert ist, kann der Benutzer wählen, ob OS/2 deinstalliert oder ob MS-DOS 6 zusätzlich zu OS/2 installiert werden soll.

Festplattenverwaltung

Ziel der Festplattenverwaltung von MS-DOS 6 sind sichere und bedienerfreundliche Lösungen für die beiden häufigsten Probleme des Benutzers: volle Festplatten und zu langsame Festplatten. Die integrierte Komprimierungsfunktion arbeitet dem mangelnden Festplattenplatz entgegen. Die verbesserte Cache-Verwaltung und Defragmentierung beschleunigt Festplattenzugriff.

Integrierte Komprimierung: DoubleSpace

DoubleSpace erhöht die Kapazität der meisten Festplatten um das Doppelte. Zu den wichtigsten Merkmalen des Programms gehört u.a.:

- **Bedienerfreundliche Komprimierung:** Handhabungsprobleme, die bei nicht integrierten Komprimierungsprogrammen auftreten, wie z.B. das Tauschen von Laufwerken und das Synchronisieren von CONFIG.SYS und AUTOEXEC.BAT zwischen den Laufwerken, fallen weg, da DoubleSpace in den MS-DOS-Betriebssystemkern integriert ist.
- **Sichere Komprimierung:** Der MS-DOS-Betriebssystemkern, DoubleSpace und der SMARTDRV-Cache-Speicher sind so konzipiert, daß sie zusammen eine sichere Komprimierung gewährleisten.
- **Bedienerfreundliche Installation:** Der Benutzer kann Daten komprimieren, die bereits auf seinen Platten installiert sind, oder mit Stacker komprimierte Laufwerke in mit DoubleSpace komprimierte Laufwerke konvertieren. Dazu wird einfach nur DBLSPACE eingegeben.
- **Sichere Installation:** Der Installationsvorgang für DoubleSpace kann neu gestartet werden. Dies ist auch dann möglich, wenn während des Komprimierungsvorgangs ein Spannungsausfall aufgetreten ist.
- **DoubleSpace Manager:** Das menügesteuerte Dienstprogramm DBLSPACE erlaubt dem Benutzer das Ausführen von Verwaltungsaufgaben, wie z.B. das Erstellen, Löschen, Vergrößern und Verkleinern komprimierter Laufwerke.
- **Maximale Komprimierung:** Mit Hilfe der maximalen Komprimierung kann der Benutzer bereits komprimierte Daten um weitere 5 - 15 % komprimieren.
- **Unterstützung für Diskettenlaufwerke:** Der Benutzer kann sowohl Diskettenlaufwerke als auch andere Wechselmedien komprimieren.

Ein Komprimierungsstandard: MRCI

Das Microsoft Real-Time Compression Interface (MRCI, Microsoft Echtzeit-Komprimierungsschnittstelle) definiert einen Komprimierungsstandard für die Kommunikation zwischen Benutzern (Client) und Bereitstellern (Server) im Rahmen der Komprimierung:

- **Software-Standard:** MRCI definiert ein API (Application Programmer's Interface, Schnittstelle für Anwendungsprogrammierer), das die Komprimierungsfunktionen von DoubleSpace für sämtliche Software, wie z.B. Sicherungsprogramme oder das Microsoft Flash-Dateisystem, verfügbar macht.
- **Hardware-Standard:** MRCI definiert einen Standard für Komprimierungshardware zum Ersetzen und Erweitern der Komprimierungsfunktionen von DoubleSpace; DoubleSpace selbst prüft, ob diese Hardware vorhanden ist, und deaktiviert in einem solchen Fall die eigenen Komprimierungsroutinen.

Verbesserte Cache-Verwaltung: Smartdrv

Durch Zwischenspeichern von Plattenlese- und -schreibzugriffen im Hauptspeicher beschleunigt Smartdrv die Leistung der Festplatte:

- **Bedienerfreundlich:** Das Installationsprogramm für MS-DOS 6 installiert Smartdrv automatisch auf den Systemen aller Benutzer.

- **Zwischenspeichern zum späteren Schreiben:** Neben Lesezugriffen speichert Smartdrv nun auch Schreibzugriffe zwischen und erhöht damit erheblich die Durchsatzrate.
- **DoubleSpace-Integration:** Smartdrv speichert komprimierte Daten zwischen und erhöht damit die effektive Größe des Cache-Speichers und die Leistung, ohne mehr Speicherplatz erforderlich zu machen.
- **Smartmon:** Ein Windows-Dienstprogramm, mit dem der Benutzer die Leistung von Smartdrv überwachen und steuern kann.

Defragmentierung

Defrag, ein bedienerfreundliches Element des Programms Speedisk aus den Norton Utilities, ordnet Dateien auf Festplatten des Benutzers neu an, gibt damit Festplattenspeicherplatz frei und erhöht die Leistung:

- **DoubleSpace-Integration:** Defrag erlaubt das Defragmentieren von mit DoubleSpace komprimierten Datenträgern, sowie das Defragmentieren nicht komprimierter Datenträger.

Hauptspeicherverwaltung

Obwohl die Hauptspeicherverwaltung unter MS-DOS 5 bereits wesentlich verbessert war, fehlte ihr noch die erforderliche Bedienerfreundlichkeit. MS-DOS 6 bietet die marktweit erste "ergonomische" Hauptspeicherverwaltung, d.h. eine Hauptspeicherverwaltung, mit der jeder Benutzer umgehen kann. MS-DOS 6 unterstützt darüber hinaus einen um bis zu 104 KB größeren hohen Speicherbereich als MS-DOS 5 und bietet damit eine insgesamt vollständige Hauptspeicherverwaltungslösung:

Leichte und sichere Optimierung: Memmaker

- **Sichere Optimierung:** Memmaker ist so konzipiert, daß es ein System nie in einem instabilen Zustand läßt. Wenn Memmaker einen Speicherkonflikt hat, übernimmt es nach einem Kalt- oder Warmstart wieder die Steuerung und erlaubt dem Benutzer, mit anderen Einstellungen oder durch Verwerfen durchgeführter Änderungen fortzufahren.
- **Leichte Optimierung:** Memmaker kann im Modus "Express" ausgeführt werden. In diesem Modus müssen zum Starten des Optimierungsvorgangs nur zwei Tasten gedrückt werden; im Modus "Benutzerdefiniert" können fortgeschrittene Benutzer Standardparameter anpassen; im Modus "Batch" wird die Optimierung automatisch ausgeführt.
- **Windows-Optimierung:** Wenn mehr Speicherplatz für MS-DOS-Anwendungsprogramme erstellt werden soll, die unter Windows laufen, sorgt Memmaker beim Optimieren dafür, daß Windows den hohen Speicherbereich verwenden kann.
- **Block-Steuerung:** Memmaker nutzt die Fähigkeit von MS-DOS 6, Programme in bestimmte Blöcke des hohen Speicherbereichs laden zu können.
- **Komprimierungserkennung:** Memmaker optimiert auch Systeme, die mit zusätzlich installierten Komprimierungsprogrammen komprimiert worden sind.
- **Unterstützung von selbsthochladenden Programmen:** Memmaker optimiert Programme, wie SMARTDRV, UNDELETE und POWER, die die in MS-DOS 5 eingeführten APIs nutzen, die ihnen erlauben, sich selbst automatisch in den hohen Speicherbereich zu laden.

Mehr Speicherplatz: Erweiterter EMM386

- **Mehr Platz im hohen Speicherbereich:** EMM386 erlangt jetzt automatisch bis zu 200 KB des hohen Speicherbereichs zurück.
- **Dynamische EMS/XMS-Zuordnung:** EMM386 weist jetzt EMS- und XMS-Speicher aus einem gemeinsam genutzten Pool zu. Dies erlaubt dem Benutzer das Ausführen von Anwendungsprogrammen, die einen dieser Speicher benötigen, ohne daß er Kenntnisse über EMS und XMS haben muß.

Verbesserte Hilfsmittel zum Anzeigen der Speicherbelegung

- **MSD:** Die Microsoft System Diagnostics (Microsoft System-Diagnose) zeigt sowohl Speicherbelegungsinformationen als auch ein genaues System-Profil an.
- **Erweiterter mem-Befehl:** Die neue Option /Free zeigt genaue Informationen über nicht genutzte Speicherblöcke an; die Option /Module gibt Informationen über ein bestimmtes Programm.

Arbeiten in Arbeitsgruppen (Workgroups)

MS-DOS 6 enthält eine integrierte Netzwerk-Client-Software und eine integrierte Mail-Client-Software zum Verschicken von Nachrichten.

- **Integrierte Netzwerk-Client-Software:** PCs unter MS-DOS 6 können auf jedem Computer, auf dem Windows für Workgroups, Windows NT oder LAN Manager installiert ist, auf gemeinsam genutzte Dateien und Drucker zugreifen.
- **Einfaches Installieren:** Eine ausgereifte Netzwerkkarten-Erkennung und eine Vielzahl von Netzwerkkarten-Treibern machen das Installieren der Netzwerk-Client-Software zu einem Kinderspiel.
- **Popup-Netzwerkschnittstelle:** Netzwerkschnittstelle, die als speicherresidentes Programm geladen und später durch Drücken einer Schnell Taste auf dem Bildschirm angezeigt werden kann und dem Benutzer erlaubt, Netzwerkverbindungen herzustellen, ohne das jeweilige Anwendungsprogramm verlassen zu müssen.
- **Mail-Client-Software:** Die Mail-Client-Software bietet vollständige Funktionalität zum Verschicken von Nachrichten und kann über jedes beliebige Netzwerk mit MS Mail-Postoffice oder mit Windows für Workgroups-Postoffice kommunizieren.
- **Interlnk:** Erlaubt zwei Computern - z.B. einem Desktop-Computer und einem Laptop-Computer - den gemeinsamen Zugriff auf Daten, Anwendungsprogramme und Drucker und benötigt nur einen freien parallelen oder seriellen Anschluß und ein Kabel.

Datenschutz

MS-DOS 6 bietet dem Benutzer mehrere neue Hilfsmittel zum Schützen seiner Daten. Dies sind insbesondere die MS-DOS- und die Windows-Version der Programme Backup, Anti-Virus und Undelete.

Backup für MS-DOS und Backup für Windows

Backup für MS-DOS 6 ist ein Lizenzprogramm der Firma Symantec und basiert auf deren erfolgreichem Programm Norton Backup. Zu den Hauptvorteilen gehören:

- **Bedienerfreundlichkeit:** menügesteuerte Schnittstelle, intuitive Bedienung; sowohl die MS-DOS- als auch die Windows-Version erlauben dem Benutzer schnelles und regelmäßiges Sichern seiner Daten.
- **Verlässlichkeit:** Eine ausgereifte Fehlererkennung erlaubt das Wiederherstellen von Daten von beschädigten Platten, und die Option zum Vergleichen von Dateien überprüft die Unversehrtheit von Daten und warnt den Benutzer, wenn ungewöhnliche Änderungen an Dateien erkannt werden.
- **DoubleSpace-Komprimierung:** Unter Verwendung von DoubleSpace werden Daten in komprimierter Form gesichert. So werden Sicherungen schneller, und der Benutzer benötigt weniger Sicherungsdisketten.
- **Konfigurierbarkeit:** Neben weiteren Optionen kann der Benutzer auch wählen, welche Dateien und Verzeichnisse gesichert oder wiederhergestellt und welcher Sicherungstyp (vollständige Sicherung, Zuwachssicherung oder Differenzsicherung) durchgeführt werden soll.

Anti-Virus für MS-DOS und Anti-Virus für Windows

Anti-Virus ist ein Lizenzprogramm der Firma Central Point Software. Es basiert auf deren erfolgreichem Programm Central Point Anti-Virus. Zu den Hauptvorteilen gehören:

- **Erkennung bekannter Viren:** Anti-Virus durchsucht sowohl den Hauptspeicher als auch Platten nach bekannten Viren und kann infizierte Dateien bereinigen.
- **Schutz vor unbekanntem Viren:** Eine Prüfsummenfunktion schützt ausführbare Dateien vor Veränderungen; VSafe, ein speicherresidentes Programm, schützt ständig vor Virusbefall.
- **Bedienerfreundlichkeit:** Sowohl die menügesteuerte MS-DOS-Version als auch die Windows-Version machen das Suchen nach Viren zu einem Kinderspiel. Die Windows-Version kann darüber hinaus auch im Hintergrund ausgeführt werden.
- **Ständige Aktualisierung:** Alle MS-DOS-Benutzer erhalten zwei kostenlose Aktualisierungen für Anti-Virus und können weitere Aktualisierungen bei Central Point Software beziehen.

Undelete für MS-DOS und Undelete für Windows

Das Programm Undelete ist im Vergleich zu der in MS-DOS 5 enthaltenen Version erheblich erweitert worden. Es handelt sich um ein Lizenzprogramm der Firma Central Point Software. Zu den Hauptvorteilen von Undelete gehören:

- **Windows-Schnittstelle:** Windows-Benutzer können jetzt Dateien wiederherstellen, ohne Windows verlassen zu müssen.
- **Verbesserter Schutz:** Zusätzlich zu den beiden Schutzebenen, die bereits unter MS-DOS 5 zur Verfügung standen (Standardschutz und Löschprotokoll), unterstützt Undelete in MS-DOS 6 eine weitere Schutzebene ("Löschüberwachung"), die das Wiederherstellen gelöschter Dateien auch auf einem Netzwerk-Laufwerk gewährleistet.
- **Undelete für Verzeichnisse:** Jetzt können zusätzlich zu gelöschten Dateien auch gelöschte Verzeichnisse wiederhergestellt werden.

Systemstart

Unter MS-DOS 6 ist der Systemstart konfigurierbar und auch von unerfahrenen Benutzern leicht durchführbar.

- **Bereinigter Start:** "Das Ende der Startdiskette" - Benutzer, die Probleme beim Konfigurieren haben, können CONFIG.SYS und AUTOEXEC.BAT übergehen und ein "bereinigtes" System starten. Dazu wird während des Startens die Taste F5 gedrückt.
- **Interaktiver Start:** Benutzer, die beim Starten die F8-TASTE drücken, erhalten nach jeder Zeile in der Datei CONFIG.SYS eine [J/N]-Aufforderung, und sie können darüber hinaus wählen, ob die Datei AUTOEXEC.BAT ebenfalls übergangen werden soll.
- **Mehrfachkonfiguration:** Der Benutzer kann durch Erstellen von Menüeinträgen und der entsprechenden Menüblöcke in seiner CONFIG.SYS-Datei beim Starten ein Menü anzeigen und aus diesem Menü unter unterschiedlichen Startkonfigurationen wählen.
- **Auswahl (CHOICE):** Ein Stapelbefehl, der Tastatureingaben akzeptiert und dem Benutzer erlaubt, einfache Menüprogramme zu erzeugen.

Dokumentation und Hilfe

Neue Dokumentation und eine umfassende Online-Hilfe erleichtern den Zugriff auf Informationen zu MS-DOS 6.

- **Neue Dokumentation:** Als Reaktion auf die häufigste Beschwerde von Benutzern zu den MS-DOS 5-Handbüchern - der enorme Umfang macht den Zugriff auf Informationen sehr schwierig - haben wir die Dokumentation neu geschrieben. Damit ist sie kompakter und konzentriert sich auf die häufigsten Benutzeraufgaben und -probleme.
- **MS-DOS zum Nachschlagen - Online:** Ein menügesteuertes Hilfesystem, das Syntax, Beispiele und Anmerkungen zu allen in MS-DOS 6 enthaltenen Befehlen und Gerätetreibern enthält.
- **Universelle Online-Hilfe:** Die Online-Hilfe steht überall zur Verfügung: für die Befehlszeile, DoubleSpace, Memmaker, Backup für MS-DOS, Backup für Windows, Anti-Virus für MS-DOS, Anti-Virus für Windows, Undelete für Windows, Defrag, Editor, QBasic, MS-DOS Shell und für Smartmon.

Windows-Unterstützung und -Integration

MS-DOS 6 unterstützt eine enge Integration in Windows und bringt Windows- und MS-DOS-Benutzern dieselben Vorteile.

- **Backup-, Anti-Virus- und Undelete-Unterstützung für Windows:** Diese Hilfsmittel, die sowohl als Windows-Versionen als auch in MS-DOS-Versionen enthalten sind, können über Symbole in der Programm-Gruppe "Microsoft Hilfsmittel" aufgerufen werden. Diese Programm-Gruppe wird während des Installationsvorgangs aus einem neuen Menü, dem Menü "Hilfsmittel" im Datei-Manager oder aus der Symbolleiste im Datei-Manager von Windows für Workgroups erstellt.
- **DoubleSpace-Unterstützung für Windows:** Informationen zu komprimierten Dateien stehen auch im neuen Menü

"Hilfsmittel" des Datei-Managers zu Verfügung, und DoubleSpace installiert und arbeitet nahtlos mit permanenten Auslagerungsdateien von Windows zusammen.

- **Unterstützung für den Einsatz von Windows im Netzwerk:** Die integrierte Netzwerk-Client-Software erlaubt das Herstellen von Verbindungen zu PCs, auf denen Windows für Workgroups oder Windows NT installiert ist.
- **Speicheroptimierung für Windows:** Memmaker ist so ausgelegt, daß MS-DOS-Anwendungen, die unter Windows laufen, der größtmögliche Speicherplatz zur Verfügung gestellt wird. □

Befehle von DOS 6.0 auf einen Blick

Die mit # gekennzeichneten Befehle sind neu bei DOS-Version 6.0; vergleichen Sie auch mit dem folgenden Beitrag über die 'DOS-Box' von WindowsNT!

ANSI.SYS	FC	#NET VER
APPEND	FCBS	#NET VIEW
ATTRIB	FDISK	NLSFUNC
BREAK	FILES	PATH
BUFFERS	FIND	PAUSE
CALL	FOR	#POWER
CHCP	FORMAT	#POWER.EXE
CHDIR (CD)	GOTO	PRINT
CHKDSK	GRAPHICS	PRINTER.SYS
#CHOICE	HELP	PROMPT
CLS	HIMEM.SYS	QBASIC
COMMAND	IF	RAMDRIVE.SYS
COPY	#INSTALL	REM
COUNTRY	#INTERLNK	RENAME (REN)
CTTY	#INTERLNK.EXE	REPLACE
#DATAMON	#INTERSVR	RESTORE
DATE	KEYB	RMDIR (RD)
#DEFRAG	LABEL	SET
DEL (ERASE)	LASTDRIVE	SETVER
DEVICE	LOADFIX	SETVER.EXE
DIR	#MAGICDRV.SYS	SHELL
DEVICEHIGH	LOADHIGH (LH)	SHARE
DIR	#MAGICDRV.SYS	
DISKCOMP	MEM	SHIFT
DISKCOPY	#MEMMAKER	SMARTDRV.EXE
DISPLAY.SYS	MIRROR	SORT
DOS	MKDIR (MD)	STACKS
DOSHELP	MODE	SUBST
DOSKEY	MORE	SWITCHES
DOSSHLL	#MSAV	SYS
DRIVER.SYS	#MSBACKUP	TIME
DRIVPARM	#MSD	TREE
ECHO	#NET HELP OR /?	TYPE
EDIT	#NET LOGOFF	UNDELETE
EGA.SYS	#NET LOGON	UNFORMAT
EMM386	#NET PASSWORD	VER
EMM386.EXE	#NET PRINT	VERIFY
EXIT	#NET START	VOL
#EXPAND	#NET STOP	#VSAFE
FASTOPEN	#NET USE	XCOPY □

WINDOWS-NT-Kommandozeilenparameter

Experimentierfreudige können den Count-Down für den Start von WindowsNT durch das Testen der β -Versionen mitmachen. Wir berichteten darüber in den PC-NEWS-28, S.3 und PC-NEWS-29, S. 35. Die Oktober-Ausgabe von WindowsNT hat schon ein brauchbares DOS-Fenster, dessen Eigenschaften zwar im Wesentlichen der DOS-Version 5.0 entsprechen (PC-NEWS 25, S.37) aber doch einige signifikante Änderungen aufweist, die auf den folgenden zwei Seiten zusammengefaßt sind.

*<BEFEHL>	Befehlsumfang gegenüber DOS 5.0 geändert.
#<BEFEHL>	neuer Befehl in WindowsNT.
-<BEFEHL>	Befehl entfällt in WindowsNT
-<BEFEHL>	derzeit noch nicht verfügbar
<hr/>	
APPEND	Enables programs to open data files in specified directories as if these files were in the current directory.
-ASSIGN	
ATTRIB	Displays or changes file attributes.
BREAK	Sets or clears extended CTRL+C checking.
-BUFFERS	Windows NT and the MS-DOS subsystem do not use this command. It is accepted only for compatibility with files from MS-DOS version 5.0 or earlier.
CALL	Calls one batch program from another.
CD	Displays the name of or changes the current directory.
CHCP	Displays or sets the active code page number.
CHDIR	Displays the name of or changes the current directory.
CHKDSK	Checks a disk and displays a status report.
CLS	Clears the screen.
#CMD	Starts a new instance of the WINDOWS-NT command interpreter. CMD.EXE replaces COMMAND.COM.
COMP	Compares the contents of two files or sets of files.
#CONVERT	This new command converts a file system from FAT or HPFS to NTFS.
COPY	Copies one or more files to another location.
~COUNTRY	
~CTTY	
DATE	Displays or sets the date.
DEBUG	Starts Debug, a program that allows you to test and debug MS-DOS executable files.
*DEL	Deletes one or more files. New switches provide many more functions.
DEVICE	Loads into memory the device driver you specify.
DEVICEHIGH	Loads device drivers into the upper memory area. Loading a device driver into the upper memory area frees more bytes of conventional memory for other programs.
*DIR	Displays a list of files and subdirectories in a directory. New switches provide many more functions.

*DISKCOMP	Compares the contents of two floppy disks. Switches /1 and /8 are not supported.
*DISKCOPY	Copies the contents of one floppy disk to another. Switch /1 is not supported.
DOS	Specifies that the MS-DOS subsystem is to maintain a link to the upper memory area or is to load part of itself into the high memory area (HMA).
-DOSSHELL	Dosshell is unnecessary with the Windows interface.
DOSKEY	Edits command lines, recalls Windows NT commands, and creates macros.
-DRIVEPARM	Windows NT and the MS-DOS subsystem do not take action for this command. It is accepted only for compatibility with MS-DOS version 5.0 or earlier.
ECHO	Displays messages, or turns command echoing on or off.
#ECHOCONFIG	This new command displays messages during the processing of the MS-DOS subsystem CONFIG.SYS file.
EDIT	Starts MS-DOS Editor, which creates and changes ASCII text files.
EDLIN	Starts Edlin, a line-oriented text editor with which you can create and change ASCII files.
#EMM	EMM.SYS replaces EMM386.EXE. EMM.SYS is now a CONFIG.SYS only command.
~ENDLOCAL	
ERASE	Deletes one or more files.
EXE2BIN	Converts .EXE (executable) files to binary format. Exe2bin is included with Windows NT as a courtesy to software developers. It is not useful for general users.
EXIT	Quits the COM.EXE program (command interpreter).
-EXPAND	Expand was used only for files on the MS-DOS 5.0 distribution disks.
-FASTOPEN	Windows NT inherent caching is superior to this utility.
FC	Compares two files or sets of files, and displays the differences between them.
-FDISK	Disk Manager prepares hard disks for use with Windows NT.
FIND	Searches for a text string in a file or files.
#FINDSTR	Searches for strings in files. This new command searches for text in files using regular expressions.
FOR	Runs a specified command for each file in a set of files.
*FORMAT	Formats a disk for use with WINDOWS-NT. 2.8 MB floptical drive supported. Switches /b, /s, and /u are not supported.
GOTO	Directs WINDOWS-NT to a labelled line in a batch program.

GRAFTABL	Enables Windows NT to display the extended characters of a specified code page in graphics mode.	*RECOVER	Recovers readable information from a bad or defective disk. Now recovers files.
GRAPHICS	Loads a program into memory that allows Windows NT to print on a printer the displayed contents of the screen when you are using a color or graphics adapter.	REM	Records comments (remarks) in batch files or CONFIG.SYS.
HELP	Provides Help information for WINDOWS-NT commands.	REN	Renames a file or files.
IF	Performs conditional processing in batch programs.	RENAME	Renames a file or files.
INSTALL	Loads a memory-resident program into memory.	REPLACE	Replaces files.
-JOIN	Increased partition size and an improved file system eliminate the need to join drives.	RESTORE	Restores files that were backed up by using the BACKUP command.
*KEYB	Configures a keyboard for a specific language. KEYBOARD.SYS is no longer used	RMDIR	Removes a directory.
*LABEL	Creates, changes, or deletes the volume label of a disk. The symbols ^ and & can be used in a volume label.	SET	Displays, sets, or removes WINDOWS-NT environment variables.
LASTDRIVE	Specifies the maximum number of drives you can access.	~SETLOCAL	
~LOADFIX		SETVER	Sets the MS-DOS version number that the MS-DOS subsystem reports to a program.
LOADHIGH (LH)	Loads a program into the upper memory area. Loading a program into the upper memory area leaves more room in conventional memory for other programs.	~SHARE	
MEM	Displays the amount of used and free memory in the MS-DOS subsystem.	SHIFT	Shifts the position of replaceable parameters in batch files.
MD	Creates a directory.	SHELL	Specifies the name and location of the command interpreter you want Windows NT to use.
-MIRROR		SHIFT	Changes the position of replaceable parameters in a batch file.
MKDIR	Creates a directory.	*SORT	Sorts input. Does not require TEMP environment variable. File size is unlimited.
*MODE	Configures a system device. Extensive changes.	STACKS	Supports the dynamic use of data stacks to handle hardware interrupts.
*MORE	Displays output one screen at a time. New switches provide many more functions.	#START	Starts a separate window to run a specified program or command.
MOVE	Moves one or more files from one directory to the specified directory on the same drive.	SUBST	Associates a path with a drive letter.
NET...	Netz-kommandos aus Platzgründen hier nicht abgedruckt.	SWITCHES	Forces an enhanced keyboard to behave like a conventional keyboard.
NLSFUNC	Starts the Nlsfunc program, which loads country-specific information for national language support (NLS).	~SYS	Windows NT will not fit on a standard 1.2 MB or 1.44 MB floppy disk.
OS/2 CONFIG.SYS	Commands The content for this topic not yet available.	TIME	Displays or sets the system time.
*PATH	Displays or sets a search path for executable files. The %path% environment variable is now accepted at the command prompt.	#TITLE	This new command sets the title of command prompt window.
PAUSE	Suspends processing of a batch file and displays a message.	TREE	Graphically displays the directory structure of a drive or path.
~POPD		TYPE	Displays the contents of a text file.
*PRINT	Prints a text file. The switches /b, /u, /m, /s, /q, /t, /c, and /p are not supported.	~UNDELETE	
PROMPT	Changes the WINDOWS-NT command prompt.	~UNFORMAT	
~PUSHD		VER	Displays the WINDOWS-NT version.
QBASIC	Starts Windows NT QBasic, a program that reads instructions written in the Basic computer language and interprets them into executable computer code.	VERIFY	Tells WINDOWS-NT whether to verify that your files are written correctly to a disk.
RD	Removes a directory.	VOL	Displays a disk volume label and serial number.
		#WINVER	This new command displays Windows NT version number.
		*XCOPY	Copies files and directory trees. New switches provide many more functions.
		&&	Command following this symbol runs only if command preceding symbol is successful.
			Command following this symbol runs only if command preceding symbol fails.
		&	Separates multiple commands on the command line.
		()	Groups commands.
		^	Escape character. Allows input of command symbols as text. □

Hewlett-Packard Printer Command Language (PCL)

Walter Riemer, Abt. EN/NA, TGM

HP Seitendrucker (Laserdrucker, InkJet- und DeskJet-Drucker) werden mittels PCL kontrolliert. Die wichtigsten Prinzipien dieser Steuersprache werden nachfolgend erläutert:

Abgesehen von einigen Steuercodes aus dem ASCII-Bereich 0 bis 31 (wie etwa CR = Carriage Return, LF = Line Feed, FF = Form Feed), die nicht Gegenstand dieser Darstellung sind, arbeitet PCL mit Escape-Sequenzen.

1. Das Wichtigste über ESC-Sequenzen

Escape (ASCII 1Bh = 27d) ist kein druckbares Zeichen, sondern teilt dem Drucker mit, daß die nachfolgenden Zeichen als Steuerzeichen zu interpretieren sind.

Die Länge einer derartigen Escape-Sequenz kann zwischen zwei und etlichen Zeichen variieren.

Zwei Zeichen lange ESC-Sequenzen haben die Form

ESC a

worin a ein Zeichen zwischen 0 (30h) und ~ (7Eh) ist, meist eine Ziffer oder ein Buchstabe.

Beispiel: ESC B

Reset (Drucker auf Default-Einstellung setzen)

Längere ESC-Sequenzen heißen parametrisierte ESC-Sequenzen, weil mit ihnen verschiedene Einstellungen, meist mittels mehr oder weniger frei wählbarer Zahlen, vorgenommen werden können.

Ihre Form ist allgemein:

ESC p g # t [binärdaten]

p ist ein Zeichen zwischen ! (21h) und / (2Fh), welches anzeigt, daß die ESC-Sequenz parametrisiert ist

g ist ein Zeichen zwischen ' (27h) und ~ (7eh), welches eine Gruppe von Steuermöglichkeiten spezifiziert

ist eine ASCII-Zahl, bestehend aus ASCII-Ziffern, bei Bedarf einem Punkt als Dezimalzeichen sowie gegebenenfalls einem Plus- oder Minuszeichen

t ist entweder ein Großbuchstabe, der die ESC-Sequenz terminiert oder ein Kleinbuchstabe, der nur den Parameter terminiert, worauf ein weiterer Parameter derselben Gruppe folgt.

binärdaten kommen in erster Linie beim Grafikdrucken vor; hier wird nicht weiter darauf eingegangen. Die eckigen Klammern bedeuten, daß die Binärdaten optional sind, sind aber nicht selbst Bestandteil der ESC-Sequenz.

Beispiele:

ESC (# 6 T stellt den Schrifttyp Letter Gothic ein

ESC (# 1 2 H stellt die Zeichendichte 12 cpi (characters per inch) ein

ESC (# 6 t 1 2 H ist eine Platz sparende Kombination der beiden vorstehenden Beispiele. Da beide ESC-Sequenzen derselben Gruppe angehören, genügt es, die erste mit t (statt mit T) zu terminieren und den Parameter der zweiten einfach anzufügen. Die endgültige Terminierung muß dann mit einem Großbuchstaben (hier T) erfolgen.

Für die Schriftmerkmale besteht ein Prioritätssystem. Ein darin höher stehendes Merkmal gilt so lange, als es nicht geändert wurde.

Priorität	Merkmal	ESC-Sequenz	Werte für #
1 (höchste)	Ausrichtung	ESC & 1 # 0	0 = Hochformat, 1 = Querformat
2	Zeichensatz	ESC (# U	z.B. 1 0 für PC-8
3	Zeichenabstand	ESC (# P	0 = fest, 1 = proportional
4	Zeichendichte	ESC (# H	z.B. 1 2 für Elite-Schrift
5	Zeichengröße	ESC (# V	z.B. 1 2 für 12-Punkt-Schrift
6	Schriftlage	ESC (# S	0 = aufrecht, 1 = kursiv
7	Strichstärke	ESC (# B	0 = normal, 3 = fett
8	Schrifttyp	ESC (# T	z.B. 6 = Letter Gothic
9 (niederste)	Qualität	ESC (# Q	1 = Entwurf, 2 = Korrespondenz

Um wirksam zu werden, müssen die ESC-Sequenzen in der Reihenfolge von der höchsten zur niedrigsten Priorität gesendet werden. Nicht gesendete ESC-Sequenzen bewirken, daß das zugehörige Merkmal unverändert bleibt, ausgenommen in dem Fall, daß eine ESC-Sequenz höherer Priorität gesendet wurde, welche mit diesem Merkmal nicht vereinbar ist; in diesem Fall wird das Merkmal niedrigerer Priorität vom Drucker automatisch angepaßt. Der Vorteil dieses Konzepts liegt darin, daß man oft die gewünschte Umschaltung mit einer relativ kurzen ESC-Sequenz bewirken kann.

Beispiel:

Nomalerweise wird man während des Druckens an den Merkmalen Ausrichtung, Zeichensatz, Zeichenabstand und Qualität nichts ändern, wohl aber an der Zeichen-

dichte und der Zeichengröße sowie hinsichtlich Schriftlage und Strichstärke. Gelegentlich muß auch der Schrifttyp verändert werden, zum Beispiel wenn man nicht mit der üblicherweise eingebauten Standardschrift Courier (mit 10 cpi) drucken will, sondern etwa mit Letter Gothic (mit 12 cpi, also sogenannter Elite-Schrift).

Aus Courier in Letter Gothic schaltet man um mittels

ESC (**1 2 h 6 T** (alle anderen Merkmale bleiben unverändert)

Da aber die Wahl der Zeichendichte in der Priorität höher steht als die Wahl des Schrifttyps, genügt es, sofern der Schrifttyp Courier in einer Zeichendichte von 12 cpi nicht verfügbar ist, nur

ESC (**12 H**

zu senden; der Drucker schaltet dann auf die verfügbare Schrift mit 12 cpi um und wechselt zwangsläufig auch auf den gewünschten Schrifttyp.

Verfehlt wäre es jedoch, nur

ESC (**6 T**

zu senden, weil in diesem Fall die geltende Zeichendichte 10 cpi nicht geändert würde und der gewünschte Schrifttyp mit 10 cpi nicht zur Verfügung steht. Courier bleibt dann unverändert.

Die vorstehenden Überlegungen gelten für Schriften, die schon standardmäßig im Drucker eingebaut sind oder die mittels Schriftkassetten oder ladbaren Schriften zur Verfügung stehen.

2. Das Wechseln von Schriften mittels Primär- und Sekundärschrift

Im Prinzip können mittels der vorstehend beschriebenen ESC-Sequenzen alle gewünschten Einstellungen vorgenommen werden, jedoch oft mit recht langen ESC-Sequenzen, für die manchmal im Druckertreiber gar nicht genug Platz reserviert ist.

PCL ermöglicht zum Glück das Festlegen der einen oder anderen Schrift als Primär- bzw. Sekundärschrift. Zwischen diesen kann dann einfach mittels des Steuerzeichens Shift-In (0Fh = 15d = CTRL-O zur Auswahl der Primärschrift) und Shift-Out (0Eh = 14d = CTRL-N zur Auswahl der Sekundärschrift) gewechselt werden.

Die Schriftmerkmale für die Sekundärschrift werden mit ESC-Sequenzen eingestellt, die ein ")" anstelle eines "(" enthalten.

Angenommen, der Drucker soll als Primärschrift Letter Gothic mit 12 cpi und Punktgröße 12 verwenden. Als Sekundärschrift soll Letter Gothic mit 16.67 cpi, aber Punktgröße 9.5 verwendet werden (für Indizes, Exponenten und dergleichen).

Zu Druckbeginn werden zum Beispiel folgende ESC-Sequenzen gesendet:

ESC E	Printer Reset
ESC (12 H	12 cpi als Primärschrift wählen
ESC) 16.67 H	16.67 cpi als Sekundärschrift wählen
ESC) 9.5 V	9.5 Punkt Schriftgröße wählen; da nur ein Schrifttyp mit dieser Zeichengröße zur Verfügung steht, erübrigt sich die anschließende Festlegung des Schrifttyps
ESC (1 Q	Entwurfsqualität (um Tinte zu sparen)
CTRL-N	Auswahl der Primärschrift

Dankenswerterweise gibt sich der Drucker auch mit der Angabe einer Zeichendichte von 16 (statt 16.67) zufrieden; aufrunden auf 17 darf man jedoch nicht! Analoges gilt für 9 anstelle von 9.5. □

*"Newton said, 'If I have seen further it is because I have stood on the shoulders of great men.'
In computing we mostly stand on each other's feet."*

Richard Hamming

Hot-Key in einem nicht-residenten Assemblerprogramm

Walter Riemer, Abt. EN, TGM

DSK-294:\HOTKEY

Als Hot-Key bezeichnet man eine Taste, deren Betätigung zu einem beliebigen Zeitpunkt eine ihr zugeordnete Aktivität eines Programms auslöst. Meist werden Hot-Keys in TSR-Programmen (speicherresidenten Programmen) dazu benützt, das Programm zu aktivieren oder auch ein bestimmtes Menü einzublenden.

Es gibt verschiedene Möglichkeiten, Hot-Keys zu realisieren:

- (1) Am einfachsten ist es in einem Programm, das ohnehin ständig eine Eingabe erwartet, wie zum Beispiel ein Editor. In diesem Fall genügt es, die gedrückte Taste zu analysieren und, falls sie eine Steuertaste ist (Hot-Key ist in diesem Fall fast schon zu viel gesagt), eine entsprechende Routine auszuführen.
- (2) Wenn man damit rechnen kann, daß irgendein Programm (sei es auch DOS selbst) häufig Eingaben erwartet, kann man den BIOS-Interrupt 16h (Tastatur-Unterstützung) umlenken und den erhaltenen Tastencode analysieren. Diese Methode ermöglicht auf recht einfache Weise das Identifizieren von Tasten, deren Status in einem Tastatur-Statusbyte festgehalten wird; etwas komplizierter ist das Identifizieren von Funktionstasten. Nachteil dieser Methode ist, daß der Hot-Key nicht jederzeit wirksam werden kann, sondern eben nur, wenn eine Eingabe erwartet wird. Ein Vorteil ist, daß diese Methode relativ einfach zu realisieren ist.

In der Folge wird auf die vorstehend angeführten Methoden nicht weiter eingegangen.

- (3) Wenn gewünscht wird, daß gewissermaßen jederzeit eine Reaktion auf das Drücken eines Hot-Key möglich sein soll, muß man den noch eines Stufe tiefer liegenden Tastatur-Interrupt 9h umlenken. Diese Methode ist aus mehreren Gründen am schwierigsten zu realisieren:
- (3.1) Sobald der Tastatur-Interrupt in die eigene Interrupt-Routine umgelenkt ist, kann man mit einem Debugger, der ja naturgemäß tastaturgesteuert ist, nicht mehr testen, weil er nicht mehr auf Tastatureingaben reagiert. Es ist dadurch recht schwierig, Fehlern in der Interrupt-Routine auf die Spur zu kommen. Den Rest des Programms kann man eventuell mittels Debuggers testen, indem man unter Ausnützung der Eigenschaft des "Conditional Assembly" zum Zweck des Testens den CTRL-BREAK-Interrupt 23h für den Interrupt 9h substituiert. Dies ist durch den nachfolgenden Programmausschnitt möglich (zum Testen setzt man das Symbol TestF auf TRUE (dies bewirkt Umlenken des Interrupt 23h), für die endgültige Ausführung auf FALSE:

```

TITLE 'TastInt.ASM, Endlosschleife durch
Tastendruck abbrechen'
TRUE EQU 1
FALSE EQU 0
TestF EQU FALSE
IF TestF
TastInt EQU 23h ; CTRL-C-Interrupt für Testzwecke
ELSE
TastInt EQU 09h ; Tastatur-Interrupt für Abbruch
ENDIF
    
```

ab hier weiter im Programm

- (3.2) DOS ist nicht re-entrant geschrieben, das heißt, wenn durch den Hot-Key-Tastendruck eine zufällig gerade aktive DOS-Funktion unterbrochen wird und die darauf eingeleitete Aktivität des Programms ebenfalls eine DOS-Funktion benützt, kommt es meistens zu schwerwiegenden Störungen im Ablauf innerhalb des DOS. Re-entrant geschriebene Routinen würden es ermöglichen, auch eine zum Unterbrechungszeitpunkt gerade aktiv gewesene Routine ein weiteres Mal zu aktivieren (dies ist unumgänglich in einem Multi-Tasking-Betriebssystem; DOS ist aber kein solches).

Zum Glück gibt es eine von MicroSoft nicht dokumentierte DOS-Funktion INT 21h - 34h, welche von Insidern dazu benützt werden kann, festzustellen, ob gerade eine DOS-Funktion aktiv ist. Dies ist der Fall, wenn das In-DOS-Flag, dessen Adresse die vorgenannte Funktion liefert, einen anderen Inhalt als 00h hat; in diesem Fall sollte der Hot-Key nicht wirksam werden. Wenn aber, wie in unserem Beispielprogramm, die Aktivität, welche durch den Hot-Key unterbrochen wird, überwiegend aus DOS-Funktionsaufrufen besteht (in unserem Fall dem Ausgeben eines Zeichens auf den Bildschirm), ist die Wahrscheinlichkeit recht groß, den Hot-Key etliche Male drücken zu müssen, bis endlich einmal ein Zeitpunkt erwischt wird, in dem DOS gerade nicht aktiv ist.

- (3.3) Man muß damit rechnen, daß es auch noch andere, insbesondere speicherresidente Programme gibt, die den Interrupt 9h umlenken. Man sollte ihnen Gelegenheit geben, ihre Interruptbehandlung auch durchzuführen. Dies wird durch das sogenannte "Chaining" (Verketteten) erreicht: Die Adresse der "alten" Interrupt-Routine, welche durch die Adresse unserer eigenen Interrupt-Routine in der Tabelle der Interrupt-Vektoren ersetzt wird, wurde ja beim Umlenken in unserem Programm gespeichert. Diese Routine sollte entweder am Anfang oder am Ende unserer eigenen Interrupt-Routine mittels FAR CALLs (oder, falls eine Rückkehr in unsere eigene Interrupt-Routine nicht erforderlich ist, mittels FAR JMPs) aufgerufen werden; dabei darf auf PUSHF nicht vergessen werden, weil ja letztlich der Stack für das unvermeidliche Verlassen der aufgerufenen Interrupt-Routine mittels IRET richtig gefüllt sein muß (man simuliert aus diese Weise einen INT-Befehl).
- (3.4) Das Analysieren der gedrückten Taste erfolgt am besten direkt im Tastaturpuffer, wo ja der Interrupt 9h die Tastencodes ablegt. Heikel wird die Sache allerdings, wenn ein anderes Programm seinen eigenen Tastaturpuffer installiert

hat, der dann meist größer ist und auf einer anderen Adresse liegt. Zwar sollte auf 40h:80h ein Pointer auf den Anfang des Tastaturpuffers stehen (und auf 40h:82h ein Pointer auf sein Ende), aber wer kann schon sicher sein, daß das andere Programm wirklich sauber geschrieben ist?

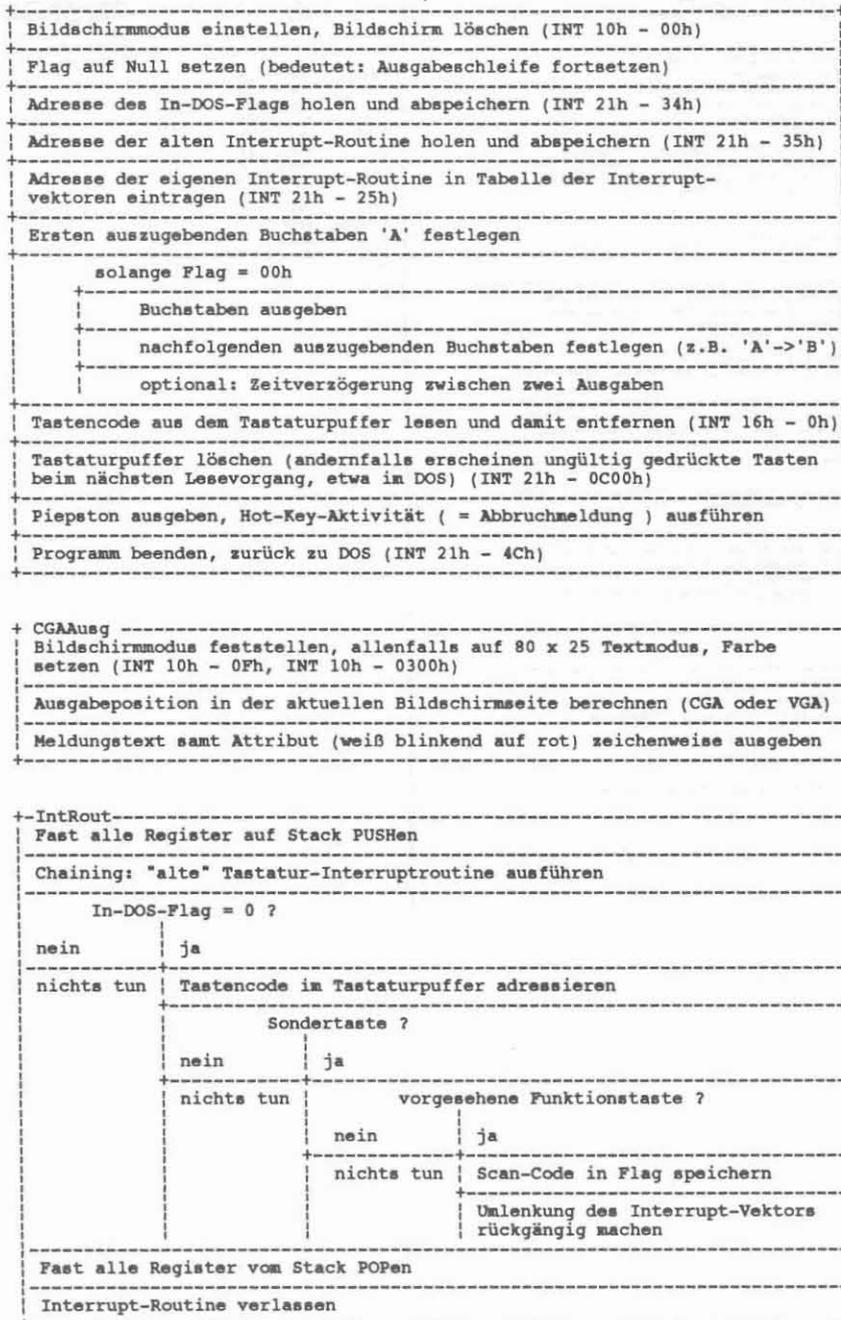
In unserem Beispielprogramm wird nur auf Funktionstasten geprüft; für andere Sondertasten wäre gegebenenfalls eine Änderung notwendig.

(3.5) Nachdem ein gültiger Hot-Key-Tastendruck festgestellt wurde, muß gegebenenfalls die Umlenkung des Tastatur-Interrupts auf Dauer (wie in unserem Beispielprogramm) oder wenigstens vorübergehend rückgängig gemacht werden. Dies kann innerhalb der Interrupt-Routine erfolgen, aber auch im Zuge des Programms außerhalb der Interrupt-Routine; die Entscheidung hängt von den Gegebenheiten im Programm ab. In unserem Fall ist beides möglich (und auch ausprobiert worden).

(3.6) Beim Testen unseres Beispielprogramms hat es sich herausgestellt, daß es in der Interrupt-Routine ratsam ist, so viele Register wie möglich auf dem Stack zu sichern, auch wenn der Sinn nicht offenkundig durchschaubar ist. So funktionierte zum Beispiel das Programm nur sehr unzuverlässig, wenn das Register ES nicht gePUSHt bzw. gePOPt wurde.

(3.7) In der Ausgabeschleife befindet sich eine Zeitverzögerungsschleife, welche je nach Bedarf wirksam oder unwirksam gesetzt werden kann (im nachfolgenden Listing ist sie unwirksam, indem sie durch Strichpunkte in Spalte 1 als vier Kommentarzeilen gekennzeichnet ist). Wenn sie wirksam ist, erfolgt die Ausgabe auf dem Bildschirm wesentlich langsamer, dafür ist die Wahrscheinlichkeit, mit dem ersten gültigen Tastendruck das Programm zu unterbrechen, entsprechend groß.

Zur besseren Übersicht wird ein Struktogramm des Beispielprogramms `TastInt9.ASM` dargestellt:



```

TITLE 'TastInt9.ASM, Endlosschleife durch Funktions-Tastendruck abbrechen'
; Nach Tastendruck Ausgabe einer Meldung. Das Programm ist so ausgelegt, daß
; auf mehrere Funktionstasten geprüft werden könnte, hier sind jedoch nur F4
; und F6 implementiert
TastInt EQU 09h ; Tastatur-Interrupt für Abbruch
Attribut EQU 0C7h

SegBIOS SEGMENT AT 40h
ORG 1Ah
BufHead DW ? ; Zeiger auf am längsten im Puffer befindliches
; Zeichen
BufTail DW ? ; Zeiger auf nächste freie Position im Puffer
Buffer DW 16 DUP (?) ; Buffer-Bereich
SegBIOS ENDS

CodeSeg SEGMENT PARA PUBLIC 'Code'
ASSUME CS:CodeSeg,ES:CodeSeg
ASSUME DS:CodeSeg,ES:SegBIOS
ORG 100h
Begin: JMP Start
;
ScanCode DB 2 ; Anzahl zu prüfender Scancodes
DB 3Eh,40h ; 3Eh = F4-Taste, 40h = F6-Taste
DB 0 ; mindestens ein Byte hinter letztem Scan-Code
; muß 00h enthalten (nötig für Prüfung in IRPruef)
; 0 = Programm fortsetzen, Scan-Code = Hot-Key
; wurde gedrückt
Flag DB 0 ; Piepston
BellText DB 7 ; Vektor des Tastaturinterrupts: Offset, Segment
TastAdr DD 0 ; Adresse des In-DOS-Flags (Offset, Segment)
InDosAdr DD 0 ; Abbruch '
MeldText DB ' Abbruch '
;
Start: MOV AX,3 ; Bildschirmmodus auf Text 80x25, Bildschirm
INT 10h ; dabei löschen
MOV Flag,0 ; Flag auf "Schleife fortsetzen" setzen
MOV AH,34h ; undokumentierte DOS-Funktion:
INT 21h ; Adresse des "In-DOS-Flag" holen
MOV WORD PTR InDosAdr+2,ES ;
MOV WORD PTR InDosAdr,BX ; Adresse abgespeichert
AH,35h ; Interrupt-Vektor erfragen
MOV AL,TastInt ; zugehörige Interrupt-Nummer
INT 21h ; jetzt alte Interrupt-Adresse in ES:BX
MOV WORD PTR TastAdr+2,ES ;
MOV WORD PTR TastAdr,BX ; jetzt gesichert in Reihenfolge IP, CS
LEA DX,IntRout ; DX erhält Offset der Interrupt-Routine
MOV AH,25h ; Interrupt-Vektor belegen
MOV AL,TastInt ; zugehörige Interrupt-Nummer
INT 21h ;
;
Alpha: MOV CX,26 ; 26 Großbuchstaben gibt es
MOV AH,2 ; Bildschirmausgabe
MOV DL,'A' ; erstes auszugebendes Zeichen
Loop1: INT 21h ; DOS-Funktionsaufruf
CMP Flag,0 ; Schleife fortsetzen ?
JNE Ende ; nein
INC DL ; ja: nächster Großbuchstabe im Alphabet
; diese 4 Zeilen wirksam oder unwirksam machen!
; XOR CX,CX ; Zeitschleife zum Verzögern der zeichenweisen
; LOOP $ ; Ausgabe, Zähler von 65536 abwärts
; POP CX ;
MOV AH,2 ;
LOOP Loop1 ; Schleife 26 mal
JMP Alpha ; Endlosschleife
;
Ende: MOV AH,0 ; BIOS-Funktion Tastatur - Zeichen aus Tastatur-
INT 16h ; Puffer lesen, damit das Zeichen, das dem
; Abbruch-Botkey entspricht, nicht nachher
; erscheint, etwa im DOS
;
MOV AX,0C00h ; Tastaturpuffer löschen
INT 21h ; damit Tastendruck für Abbruch wirksam wird
;
PUSH CS ; hier könnte noch der in Flag zurückgegebene
POP DS ; Scan-Code identifiziert werden und je nach
; gedrückter Funktionstaste reagiert werden
MOV DL,BellText ; Piepston
MOV AH,2 ; ausgeben
INT 21h ;
CALL CGAAusg ; Bot-Key-Aktivität (hier: Meldung einblenden)
MOV AH,4Ch ; EXIT (Programm beenden)
INT 21h ;
;
CGAAusg: XOR BL,BL ; Routine, die direkt ins Video-RAM schreibt
MOV AH,15 ; Bildschirmmodus feststellen
INT 10h ; BIOS-Video-Interrupt
CMP AL,3 ; Bildschirmmodus 3 (Textmodus 80x25, Farbe) ?
JE CGAA0 ; ja
MOV AX,3 ; nein: Bildschirmmodus auf 3 setzen
INT 10h ;
CGAA0: MOV CL,4
SHL BX,CL ; 0 bis 3 mal 4096 (aktuelle Bildschirmseite)
MOV AX,0B800h ; CGA-Videopeicher Anfang
ADD AX,BX ; Anfang der aktuellen Seite
ADD AX,12*160/16 ; zwölfte Zeile, /16 weil Segmentadresse
PUSH AX
POP ES
MOV CX,9 ; Länge der Abbruch-Meldung
LEA SI,MeldText
MOV DI,36*2 ; Spalte 36 für erste Zeichenausgabe
CGAA1: MOV AL,[SI]
MOV BYTE PTR ES:[DI],AL
INC SI
INC DI
MOV BYTE PTR ES:[DI],Attribut
INC DI
LOOP CGAA1
RET
    
```

```

IntRout PROC FAR ; Interrupt-Routine für INT 9
PUSH AX
PUSH BX
PUSH CX
PUSH DX
PUSH SI
PUSH DI
PUSH BP
PUSH ES
PUSH DS
PUSHF ; "Chaining" zur alten Interrupt-Routine
CLI ; zunächst normale Tastatur-Interrupt-Routine
CALL CS:TastAdr ; ausführen, statt INT-Befehl wird PUSHF + CALL FAR
; ausgeführt, damit nicht beim nächsten Tastendruck
; wieder IntRout aktiviert wird
;
PUSH CS ;
POP DS ; Adressierbarkeit im Programm sicherstellen
LES BX,InDosAdr ; ES:BX zeigt jetzt auf In-DOS-Flag
CMP BYTE PTR ES:[BX],0 ; das Byte dort sollte 00h sein
JNE IREnd ; wenn nicht: Interrupt-Routine verlassen, weil
; DOS gerade aktiv ist
;
MOV AX,SegBIOS ; Segmentadresse
MOV ES,AX ;
MOV SI,ES:BufTail ; Adresse des letzten Zeichens im Puffer
SUB SI,2 ; Zeiger auf vorhergehendes Wort im Puffer
CMP BYTE PTR ES:[SI],0 ; Sondertaste ?
JNE IREnd ; nein: Taste ungültig
MOV AL,BYTE PTR ES:[SI+1] ; ja: Scancode aus Tastaturpuffer holen
LEA DI,ScanCode ; Byte vor erstem zu prüfenden Scan-Code
XOR CX,CX
MOV CL,ScanCode ; Anzahl zu prüfender Scan-Codes
INC CL ; damit Endbyte (mit 00h) noch mitgeprüft wird
IRPruef: INC DI ; nächster zu prüfender Scan-Code
CMP AL,[DI] ; richtige Funktionstaste ?
LOOPNE IRPruef ; nein: Taste ungültig
MOV AL,[DI] ; ja: gültiger Scan-Code oder 00h
CMP BYTE PTR [DI],00h ; 00h ?
JE IREnd ; ja
MOV Flag,AL ; nein: Scan-Code als Signal "Schleife beenden"
;
MOV AB,25h ; Interrupt-Vektor wieder auf INT 9 legen
CLI ; Interrupts sperren
MOV AL,TastInt ; zugehöriger Interrupt-Vektor
LDS DX,TastAdr ; DS:DX enthält FAR-Adresse der alten INT-9-Routine
INT 21h ; jetzt Interruptroutine wieder unwirksam
; Interrupts wieder freigeben
; Anmerkung: diese Routine könnte auch in der
; Routine Ende nach dem Lesen aus dem Tastatur-
; puffer stehen
;
IREnd: POP DS
POP ES
POP BP
POP DI
POP SI
POP DX
POP CX
POP BX
POP AX
IRET ; zurück ins unterbrochene Programm
IntRout ENDP
CodeSeg ENDS
END Begin
    
```

ANSI-Bildschirmsteuerung in C

Franz FIALA, N, TGM

DSK-294:\MYLIB

Auf der einen Seite ist C eine sehr portable Sprache. Wir benutzen Compiler, die immer konform zum ANSI-C-Standard arbeiten. Normen, wie der ANSI-Standard für die Sprache C, stellen aber in den meisten Fällen nur ein Minimum von Möglichkeiten zur Verfügung, eben jenen, der nicht von einer besonderen Hardware abhängig ist.

Eine der ersten Schülerfragen: "Wie steuere ich den Bildschirm?" haben BORLAND und MICROSOFT durch die Funktionen der nicht-ANSI-kompatiblen Headerdatei `conio.h` beantwortet, da die ANSI-Methode mit ESC-Steuresequenzen für flotte Bildsteuerung einfach zu langsam ist. Dennoch gibt es noch viele Anwendungen, etwa die gesamte Datenübertragung, die ausschließlich mit solchen Bildschirmsteuerungen auskommen muß; und auch BTX ist vereinfacht ausgedrückt nicht anderes als ein überaus komplexes Terminal.

Wie kann man nun eine elementare Bildschirmsteuerung ohne Verwendung gerätespezifischer Funktionen programmieren? Erinnern wir uns, daß praktisch alle PCs und verwandte Kleinrechner zwei Geräte in einem sind: Der eigentliche Rechner und eine Konsole (Terminal). Wenn auch beide im PC zu einem hohen Grad verschmolzen sind, können wir noch rudimentäre Züge eines Terminals erkennen: einerseits im selten geladenen Bildschirm- und Tastaturtreiber `ANSI.SYS` und andererseits in der Möglichkeit, mit dem DOS-Kommando `CTTY`, Tastatur und Bildschirm auf eine andere Konsole umlenken zu können, die dann bezüglich der Bildschirmsteuerung dieselben Eigenschaften, wie auch der PC hat. In dem Treiber `ANSI.SYS` sind nur die wichtigsten Elemente zur Bildschirm- und Tastatursteuerung enthalten. Sie sind ein kleiner Teil der Möglichkeiten des bekannten VT-100-Terminals von DEC.

Die weitgehend selbständige Installation von DOS 5.0 installiert zwar alle landesspezifischen Eigenschaften, wie `KEYB.EXE`, `COUNTRY.SYS` usw. nicht aber den ANSI-Treiber. Alle folgenden Programme sind aber nur wirksam, wenn Sie die folgende Zeile in `CONFIG.SYS` einbauen:

```
DEVICEHIGH=C:\DOS\ANSI.SYS
```

War maschinenunabhängiges Programmieren zumindest am PC-Sektor nicht erforderlich, da ja ohnehin alle Rechner PCs waren, kommt jetzt durch viel komplexere Bediensysteme (Windows, X-Windows u.a.) wieder die hardwareunabhängige Programmierung zum Zug, wie sie schon viel früher eben auf der Ebene der Terminalprogrammierung und auch beim CP/M-System üblich war.

Die ANSI-Bildschirmsteuerung am PC ist durch genau 15 Escape-Sequenzen beschrieben. Dazu kommt noch als 16. Steuersequenz eine für die Veränderung der Tastaturbelegung, die aber in diesem Beitrag nicht beschrieben wird.

Aufbau der ANSI-Steurelemente

ESC[parameter	COMMAND
ESC	das Zeichen Escape (=0x1b=27=^)
[das Zeichen '['
parameter	ein oder mehrere Zahlenwerte in ASCII-Format, die in der allgemeinen Schreibweise durch das Zeichen '*' ersetzt werden. Mehrere Parameter werden mit Strichpunkten getrennt.
COMMAND	ein oder mehrere Alphazeichen, die den Befehl kennzeichnen. Achtung Groß-/Kleinschreibung wichtig!

Kursor-Steuerung

ESC[# ; #H	Kursor-Position festlegen. Der erste Parameter ist die Zeile, der zweite ist die Spalte. Wird nichts angegeben (ESC[H oder ESC[#H), wird die erste Zeile und erste Spalte angenommen.
ESC[# ; #F	Kursor-Position festlegen. Der erste Parameter ist die Zeile, der zweite ist die Spalte. Wird nichts angegeben (ESC[F oder ESC[#F), wird die erste Zeile und erste Spalte angenommen.
ESC[#A	Kursor nach oben. # ist die Anzahl der Zeilen, Anfangswert ist 1. Wird in der ersten Bildschirmzeile ignoriert.
ESC[#B	Kursor nach unten. # ist die Anzahl der Zeilen, Anfangswert ist 1. Wird in der letzten Bildschirmzeile ignoriert.
ESC[#C	Kursor nach rechts. # ist die Anzahl der Zeilen, Anfangswert ist 1. Wird in der letzten Bildschirmspalte ignoriert.
ESC[#D	Kursor nach links. # ist die Anzahl der Zeilen, Anfangswert ist 1. Wird in der ersten Bildschirmspalte ignoriert.
ESC[6n	Kursor-Position abfragen. Unmittelbar nach dem Absetzen dieser ESC-Sequenz muß über <code>stdin</code> die Antwort erfragt werden.
ESC[# ; #R	Kursor-Positionsformat. Diese ESC-Sequenz wird nicht gesendet, sondern über <code>stdin</code> empfangen. Sie ist die Folge der Abfrage der Kursorposition mit <code>ESC[6n</code> . Der erste Parameter ist die Zeile, der zweite ist die Spalte.
ESC[s	Speichern der aktuellen Kursor-Position. Kann mit <code>ESC[u</code> wiederhergestellt werden.
ESC[u	Wiederherstellen der Kursor-Position. Diese Sequenz kann nicht verschachtelt werden.

Löschen

ESC[2J	Löscht den Bildschirm und geht in die Position 1,1.
ESC[K	Löscht die aktuelle Zeile ab und inklusive der Kursorposition bis zum rechten Bildrand.

Bildschirmsteuerung

ESC[# ; . . ; #m	Setzen der Bildschirmattribute, siehe Tabelle im Listing
ESC[= #h	Setzen eines Bildschirmmodus, siehe Tabelle im Listing.
ESC[= #07l	Rücksetzen des Zeilenumbruchs (Auto-NewLine).

So, wie sich diese Steuerelemente präsentieren, kann man zur Not bereits einfache Steuerungen durchführen, die annähernd dasselbe leisten, wie etwa die Funktionen aus `conio.h`, die allesamt nicht ANSI-kompatibel sind. Will man mehr, besorge man sich eine erweiterte Version des Treibers `ANSI.SYS` mit Dokumentation, um zusätzliche Steuerelemente zu haben.

Bildschirmsteuerung in Maschinensprache

Die Steuerung des Bildschirms erfolgt durch Senden definierter Escape-Sequenzen über die MSDOS-Funktion `0x3f` und `0x40` (oder auch `0x01`, `0x02` und `0x06`) /5/. Diese Sequenzen entnehmen wir aus /1/ für DOS-Versionen bis 3.3 und aus /2/ für DOS-Versionen ab 4.0, da sich einige Veränderungen ergeben haben.

Bildschirmsteuerung in C

Ganz einfach. Man tut so, als wolle man die Zeichenfolge an den Bildschirm senden. Der ANSI-Treiber nimmt sich der Bildschirmsteuerelemente an, indem er sie aus der Zeichenkette herausfiltert und in entsprechende Steuerungen umsetzt.

```
printf("\x1b[2J"); /* Löscht den Bildschirm */
```

Es ist wichtig, die Funktion `printf()` zu verwenden und nicht etwa `cprintf()` (von BORLAND), da nur die Funktionen aus `stdio.h` auch die Ein- und Ausgabefunktionen des Betriebssystems benutzen. Auch `puts()` kann, da es Bestandteil von `stdio.h` ist verwendet werden, nicht aber `cputs()`, welches zu `conio.h` gehört.

In dieser Form ist aber die Ausgabe nicht sehr anschaulich und ohne Kommentar nicht verständlich. Besser ist schon folgendes:

```
#define CLRSCR "\x1b[2J"
...
printf(CLRSCR);
```

In dieser Form können wir praktisch auf den Kommentar verzichten; außerdem ist klar ersichtlich, daß es sich um eine an die Standardausgabe gesendete Zeichenkette handelt.

Noch einfacher wäre es aber, (wie es auch in den Funktionen von `conio.h` geschieht) das Bildschirmlöschen einer Funktion zu übertragen:

```
void clrscr(void)
{
    printf("\x1b[2J");
}
...
clrscr();
```

Das wieder erfordert - da man jede einzelne Steuerung in eine Funktion verpackt - eine eigene Bibliothek für diese Funktionen. Außerdem wird die ohnehin langsame Bildausgabe über das Betriebssystem noch ein bißchen langsamer.

Deshalb wurde ein in /3/ beschriebener Weg über Makros beschritten. Dieselbe Funktion zum Löschen des Bildschirms schaut dann so aus:

```
#define CLRSCR() printf("\x1b[2J")
...
CLRSCR();
```

Die Großschreibweise deutet daraufhin, daß es sich um ein Makro handelt und daher von `clrscr()` aus der Standardbibliothek zu unterscheiden ist.

Der Vorteil der Makroschreibweise ist, daß man lediglich eine Headerdatei und keine Bibliothek benötigt. Die Makros expandieren bei Anwendung in entsprechende `printf`-Anweisungen.

Der Nachteil der Makroschreibweise ist, daß bei Makros mit Argumenten keine Typenprüfung erfolgt, da es sich um eine reine Textsubstitution handelt. Außerdem kann es bei der Kompilierung makrointensiven Codes zu anscheinend zusammenhanglosen Fehlermeldungen kommen, die sich aus dem expandierten Code ergeben.

In diesem einfachen Fall sei diese Vorgangsweise einmal erlaubt. [Achten Sie darauf, alle Warnungen des BORLAND-Kompilers eingeschaltet zu haben und besonders bei der Kompilierung dieser Makros auf die Warnungen zu hören.]

ANSI.H

Die Headerdatei `ANSI.H` stellt jedem C-Programm funktionsähnliche Makros und vordefinierte Konstanten zur Verfügung, die für die Bildschirmsteuerung bei geladenem ANSI-Treiber geeignet sind.

Die Headerdatei `ANSI.H` sollte für den weiteren Gebrauch ausreichend kommentiert sein. Die folgenden Hinweise sind mit der etwas später abgedruckten Datei `ANSI.H` zu lesen.

Bildschirmwahl

Die modernen EGA/VGA-Karten haben eine Vielzahl von Darstellungsarten, von denen einige über den ANSI-Treiber ansprechbar sind. Sie werden mit

```
SETSCREEN(mode)
```

eingestellt, wobei `mode` eine Zahl zwischen 0 und 19 ist. Die Bedeutung der einzelnen Werte ist der Tabelle aus `ANSI.H` zu entnehmen. Das Testprogramm für die Makros wählt jeden Bildschirmmodus an. Nicht in allen Fällen ergibt sich Übereinstimmung mit dem Handbuch. z.B. sollte Modus 2 eine Schwarz/Weiß-Darstellung erzwingen, was aber nicht geschieht. Im Zweifel Testprogramm laufen lassen. Was die Grafik-Betriebsarten in einem reinen Text-Treiber verloren haben ist unklar, da man über denselben Treiber den Bildschirm nicht pixelorientiert ansprechen kann.

Bildattribute

Eine bestimmte Bildschirmeinstellung besteht i.a. aus drei Teilen: Vordergrundfarbe, Hintergrundfarbe und ein zusätzliches Attribut, wie Blinken o.ä. das entsprechende Makro arbeitet daher mit drei Argumenten. Man kann auch nur ein Argument benutzen; unbenutzte Argumente bleiben auf 0 (normal). Das Makro `SETDISPLAY()` stellt den Bildschirm ein. z.B.:

```
SETDISPLAY(ANSI_LOW,ANSI_FORE_CYAN,ANSI_BACK_BLUE); oder
SETDISPLAY(0,0,7);
```

Die Konstanten `ANSI_...` wurden statt der nichtssagenden Nummern eingeführt. Einige der Konstanten sind nur auf Bildschirmdaten anwendbar, die dieses Attribut unterstützen, wie z.B. Schrägschrift und schnelles Blinken.

Häufig vorkommende Steuerelemente wurden eigens definiert:

VIDHIGH()	Helle Schrift
VIDBLINK()	Blinkende Zeichen
VIDNORM()	zurück zum Normalbetrieb
VIDINVISIBLE()	Zeicheneingabe unsichtbar machen

Löschfunktionen

Die Löschfunktionen `CLS()` und `EEOL()` sind unproblematisch. Leider fehlt eine Steuersequenz zum Löschen bis Bildschirmende.

Kursorbewegung

Die Kursorbewegung wurde in Anlehnung an die BORLAND-Konsolenfunktion benannt:

<code>GOTOXY()</code>	positioniert den Cursor und
<code>WHEREXY(x,y)</code>	liefert die aktuelle Cursorposition in zwei ganzzahligen Variablen.

Die letztere Abfrage der Position bereitet schon einige Schwierigkeiten, da die Abfrage mit `scanf()` ein störendes Echo am Bildschirm zur Folge hat, welches mit dem Bildschirmattribut 8 (unsichtbar) ausgeschaltet werden muß. Außerdem erfordert das korrekte Lesen der ESC-Sequenz genauere Kenntnisse der Parameter der Funktion `scanf()`. Hier ist es besonders das Unterdrücken unwichtiger Zeichen (`ESC`) und nach der Zeilenposition das Zeichen `;`. Zur Unterdrückung unerwünschter Eingabezeichen wird das Zeichen `*` gemeinsam mit einem Multiplikator verwendet. `*2c` unterdrückt 2 Zeichen des Typs `char`.

Darüberhinaus gibt es noch die Möglichkeit die aktuelle Cursorposition festzuhalten und zu einem späteren Zeitpunkt wiederherzustellen

<code>CURSPUSH()</code>	Kursorposition merken
<code>CURSPOP()</code>	Kursorposition wiederherstellen

und die Möglichkeit zu relativen Kursorbewegung:

<code>CUUP(z)</code>	Kursor um z Zeilen nach oben
<code>CUDN(z)</code>	Kursor um z Zeilen nach unten
<code>CULT(z)</code>	Kursor um z Spalten nach links
<code>CURT(z)</code>	Kursor um z Spalten nach rechts

wobei ein Parameter die Anzahl der Zeilen oder Spalten angibt.

Zeichenausgabe

Die Zeichenausgabe auf eine bestimmte Position ist zwar ohnehin mit

```
GOTOXY(10,12); printf(txt);
```

möglich. Da diese Sequenz aber häufig gebraucht wird, wurden drei Makros definiert, die die positionierte Ausgabe von Zeichen und Zeichenketten erleichtern:

<code>XYCHAR()</code>	gibt ein Einzelzeichen aus
<code>XYSTRING()</code>	gibt eine Zeichenkette aus
<code>XYZSTRING()</code>	gibt eine Zeichenkette zentriert zur Cursorposition aus.

```
#ifndef __ANSI_H
#define __ANSI_H

#ifdef __STUDIO_H
#include <stdio.h>
#include <string.h>
#endif

/* SETSCREEN() Bildschirmmodi
0=40x25 mono
1=40x25 color
2=80x25 mono
3=80x25 color
4=320x200 color
5=320x200 mono
6=640x200 mono
7=enable word wrap at end of line
14=640x200 16 color EGA/VGA, MSDOS 4.0 and later
15=640x350 2 color EGA/VGA, MSDOS 4.0 and later
16=640x350 16 color EGA/VGA, MSDOS 4.0 and later
17=640x480 2 color EGA/VGA, MSDOS 4.0 and later
18=640x480 16 color EGA/VGA, MSDOS 4.0 and later
19=320x200 256 color EGA/VGA, MSDOS 4.0 and later
*/
#define SETSCREEN(a) printf("\033[=uh",a)
/* reset screen graphics mode, Attribut wie SETSCREEN
a=7: disable word wrap */
#define RESETSCREEN(a) printf("\033[=ul",a)

#define ANSI_NORMAL 0
#define ANSI_HIGH 1
#define ANSI_LOW 2
#define ANSI_ITALIC 3
#define ANSI_UNDERLINE 4
#define ANSI_BLINK 5
#define ANSI_FASTBLINK 6
#define ANSI_REVERSE 7
#define ANSI_INVISIBLE 8

#define ANSI_FORE_BLACK 30
#define ANSI_FORE_Red 31
#define ANSI_FORE_Green 32
#define ANSI_FORE_Yellow 33
#define ANSI_FORE_Blue 34
#define ANSI_FORE_Magenta 35
#define ANSI_FORE_Cyan 36
#define ANSI_FORE_White 37

#define ANSI_Back_Black 40
#define ANSI_Back_Red 41
#define ANSI_Back_Green 42
#define ANSI_Back_Yellow 43
#define ANSI_Back_Blue 44
#define ANSI_Back_Magenta 45
#define ANSI_Back_Cyan 46
#define ANSI_Back_White 47

#define SETDISPLAY(a,b,c) printf("\033[%u;%u;%um",a,b,c)
#define VIDHIGH() SETDISPLAY(0,0,1)
#define VIDBLINK() SETDISPLAY(0,0,5)
#define VIDNORM() SETDISPLAY(0,0,0)
#define VIDINVISIBLE() SETDISPLAY(0,0,8)
#define BEEP() printf("\007") /* 800 Hz Ton, 1/4 Sekunde */
/* CLS() löscht den Bildschirm und positioniert den Cursor
in der linken oberen Bildschirmecke */
#define CLS() printf("\033[2J")
/* EEOL() Löscht die Zeile ab und
inklusive der aktuellen Cursorposition */
#define EEOL() printf("\033[K")

/* Kursorsteuerung */
#define GOTOXY(x,y) printf("\033[%u;%uH", (x),(y))
#define SETCURPOS(x,y) GOTOXY(x,y)
#define WHEREXY(x,y) {VIDINVISIBLE();\rewind(stdin);\
printf("\033[6n");\
scanf("%2c%2d%1c%2d%2c", (x),(y));\
VIDNORM();}
#define GETCURPOS(x,y) WHEREXY(x,y)

/* Kursorbewegung um eine bestimmte Anzahl von Zeilen/Spalten */
#define CUUP(x) printf("\033[%uA", (x))
#define CUDN(x) printf("\033[%uB", (x))
#define CURT(y) printf("\033[%uC", (y))
#define CULT(y) printf("\033[%uD", (y))

/* Kursorposition merken und wiederholen */
#define CURSPUSH() printf("\033[s")
#define CURSPOP() printf("\033[u")

#define DEVICESTATUS() printf("\033[6n")

/* Zeichenausgabe */
#define XYCHAR(x,y,z) printf("\033[%u;%uH%c", (x),(y),(z))
#define XYSTRING(x,y,str) printf("\033[%u;%uH%s", (x),(y),str)
#define XCTRPRINTF(x,y,str) printf("\033[%u;%uH%s", (x),
((80-(strlen(str)-1))/2),str)

#define GETKBE() {int c;\rewind(stdin);c=getchar();c++;}
#define GETKB() {VIDINVISIBLE();GETKBE();VIDNORM();}
#define GETCH() {VIDINVISIBLE();\rewind(stdin);\
c=getchar();VIDNORM();}
#define GETCHE(c) {rewind(stdin);c=getchar();}
#define GETCHHI(c) {VIDHIGH();printf("%b");\
GETCHE(c);VIDNORM();}
#endif /* ANSI_H */
```

Zeicheneingabe

Die Zeicheneingabe ist zwar nicht unmittelbare Aufgabe des ANSI-Treibers, es ist jedoch wünschenswert, auch alle Eingaben solcher Programme über `stdio.h` abzuwickeln. Hier liegt zur Zeit auch noch die grundsätzliche Schwierigkeit, daß das Lesen eines einzelnen Zeichens durch `getchar()`, siehe auch /5/, keineswegs dieselben Eigenschaften hat, wie z.B. `getch()` aus `conio.h`. Darüberhinaus lassen sich auch die Eigenschaften von `getch()`, wie z.B. Unterdrückung des Echos oder Abbrechen der Eingabefunktion nach einem Zeichen nicht vollständig nachbilden.

Die Funktion `kbhit()` aus `conio.h` konnte ich nicht nachbilden. Eine Eingabe von Einzelzeichen erfordert jedenfalls einen Abschluß in Form eines ENTER-Zeichens, um von `getchar()` verarbeitet zu werden, welches dann auch nur das erste Zeichen aus dem Puffer liest. Das Makro `GETKB()` ist ein quasi-Ersatz für `kbhit()`, wobei die Echos der Zeichen durch das Attribut 'unsichtbar' unterdrückt werden. Die zusätzliche Funktion `rewind()` erwies sich als notwendig, siehe /4/, um eventuelle Restzeichen im Tastaturpuffer zu löschen, damit wirklich nur das zuletzt eingegebene Zeichen bearbeitet wird.

Die Makros `GETCH()` bzw. `GETCHE()` sind ähnlich, nur wird das eingegebene Zeichen auch zurückgeliefert.

Das Makro `GETCHHI()` intensiviert die Bildschirmfarbe während der Eingabe.

Alle Makros, die aus mehr als nur einem Befehl bestehen, werden eingeklammert (`()`), da sie sonst (etwa als Teil einer `if`-Anweisung) falsch interpretiert werden.

Die merkwürdige Befehlsfolge im Makro `GETKBE()` speziell die lokale Variable `c`, hat die Aufgabe eine Compiler-Fehlermeldung zu unterdrücken, die sonst durch die einzelstehende Anweisung `getchar()` entsteht.

Was ist schon ein 'PC-Haus' ohne Fenster!

Die schönste Bildschirmsteuerung ist aber unzeitgemäß, wenn sie nicht auch einige Möglichkeiten zur Fenstergestaltung aufweist. Da dies aber die Möglichkeiten einer Headerdatei übersteigt, wurden in einer Datei `ANSIW.C` die benötigten Funktionen ergänzt. Werden dann in einer Anwendung auch gerahmte Textabschnitte benötigt, kann man dann diese Datei `ANSIW.C` inkludieren.

ANSIW.H

Zu `ANSIW.C` gehört auch eine Headerdatei `ANSIW.H`, in der auch alle Rahmenzeichen und die Funktionsprototypen enthalten sind.

Die Bezeichnungen für die Rahmenzeichen bedeuten: `IBM_` mit Hilfe des IBM-Zeichensatzes, `ASC_` mit Hilfe des ASCII-Zeichensatzes. `_Sxxx..single line border`, `_Dxxx..double line border`, `..ULC..upper left corner`, `..LLC..lower left corner` usw.

```
#ifndef ANSIW_H
#define ANSIW_H

#define WINDOW(a,b,c,d,e,f) draw(a,b,c,d,e,f);fill(a+1,b+1,c-1,d-1,e)

#define IBM_SULC '+'
#define IBM_SLLC '+'
#define IBM_SLRC '+'
#define IBM_SURC '+'
#define IBM_SVTL '^'
#define IBM_SHTL '-'
#define IBM_DULC '+'
#define IBM_DLLC '+'
#define IBM_DLRC '+'
#define IBM_DURC '+'
#define IBM_DVTL '^'
#define IBM_DHTL '-'

#define ASC_SULC '+'
#define ASC_SLLC '+'
#define ASC_SLRC '+'
#define ASC_SURC '+'
#define ASC_SVTL '|'
#define ASC_SHTL '-'
#define ASC_DULC '#'
#define ASC_DLLC '#'
#define ASC_DLRC '#'
#define ASC_DURC '#'
#define ASC_DVTL '*'
#define ASC_DHTL '='

#define IBM_NONE ' '
#define IBM_SINGLE 'i'
#define IBM_DOUBLE 'I'
#define IBM_LIGHT ' '
#define IBM_MEDIUM '-'
#define IBM_DARK '-'
#define IBM_BLACK '-'

#define ASC_NONE ' '
#define ASC_SINGLE 'a'
#define ASC_DOUBLE 'A'
#define ASC_LIGHT '+'
#define ASC_MEDIUM 'x'
#define ASC_DARK '*'
#define ASC_BLACK '#'

void hlin(int row, int col, int len, int c);
void vlin(int row, int col, int len, int c);
int draw(int row1, int col1, int row2, int col2, int type);
int fill(int row1, int col1, int row2, int col2, int icon);

#endif /* ANSIW_H */
```

- `hlin()` zeichnet horizontale Linien aus beliebigen Zeichen
- `vlín()` zeichnet vertikale Linien aus beliebigen Zeichen
- `draw()` zeichnet einen Rahmen, verwendet `hlin()` und `vlín()`. Rahmenbegrenzung vielfältig gestaltbar.
- `fill()` füllt eine Fläche mit wählbarem Füllzeichen, verwendet `hlin()` und `vlín()`.
- `WINDOW()` Kombination aus `draw()` und `fill()`.

Die mit `IBM_` eingeleiten Zeichen erscheinen in diesem Listing anders, da der Windows-Zeichensatz an diesen Stellen Kompromißzeichen einsetzt. Da es immer auch zu Kommunikation mit Nicht-IBM-Endgeräten kommen kann (besser Nicht-IBM-PC), ist auch als Alternative eine reine ASCII-Variante für die Rahmenzeichen angegeben (`ASC_`).

ANSIW.C

```

/* ANSIW.C Fensterfunktionen für ANSI-Bildschirm */
#include <ansiw.h>
void hlin(int row, int col, int len, int c)
{
    GOTOXY(row,col); while (len-->0) printf("%c",c);
}
void vlin(int row, int col, int len, int c)
{
    int r;
    for (r=0; r<len; r++) XYCHAR(r+row,col,c);
}
int draw(int row1, int col1, int row2, int col2, int type)
{
    int hlen, vlen; /* Breite und Höhe des Fensters */
    /* Zeichnen für die verschiedenen Rahmenpositionen */
    int hzl, /* horizontal line */
        vtl, /* vertikal line */
        ulc, /* upper left corner */
        llc, /* lower left corner */
        urc, /* upper right corner */
        lrc; /* lower right corner */
    hlen=col2-col1; vlen=row2-row1;
    switch (type)
    {
        case IBM_SINGLE: /* IBM-Zeichensatz, einfache Linie */
            ulc=IBM_SULC;llc=IBM_SLLC;lrc=IBM_SLRC;
            urc=IBM_SURC;vtl=IBM_SVTL;hzl=IBM_SHZL;
            break;
        case IBM_DOUBLE: /* IBM-Zeichensatz, doppelte Linie */
            ulc=IBM_DULC;llc=IBM_DLLC;lrc=IBM_DLRC;
            urc=IBM_DURC;vtl=IBM_DVTL;hzl=IBM_DHZL;
            break;
        case ASC_SINGLE: /* ASCII-Zeichensatz, einfache Linie */
            ulc=ASC_SULC;llc=ASC_SLLC;lrc=ASC_SLRC;
            urc=ASC_SURC;vtl=ASC_SVTL;hzl=ASC_SHZL;
            break;
        case ASC_DOUBLE: /* ASCII-Zeichensatz, doppelte Linie */
            ulc=ASC_DULC;llc=ASC_DLLC;lrc=ASC_DLRC;
            urc=ASC_DURC;vtl=ASC_DVTL;hzl=ASC_DHZL;
            break;
        default: /* jedes andere Zeichen wird als einheitliches
            Rahmenzeichen angenommen */
            ulc=llc=lrc=urc=vtl=hzl=type;
            break;
    }
    if (hlen<0 || vlen<0) /* Längenangaben waren fehlerhaft */
        return (1);
    if (hlen==0 && vlen==0) /* es gibt nur einen Punkt */
    {
        XYCHAR(row1,col1,ulc);return (0);
    }
    if (vlen==0) /* es ist eine horizontale Linie */
    {
        hlin(row1,col1,hlen,hzl);return (0);
    }
    if (hlen==0) /* es ist eine vertikale Linie */
    {
        vlin(row1,col1,vlen,vtl); return (0);
    }
    /* es ist ein Rechteck */
    hlin(row1,col1+1,hlen-1,hzl); /* obere Begrenzungslinie */
    hlin(row2,col1+1,hlen-1,hzl); /* untere Begrenzungslinie */
    vlin(row1+1,col1,vlen-1,vtl); /* linke Begrenzungslinie */
    vlin(row1+1,col2,vlen-1,vtl); /* rechte Begrenzungslinie */
    XYCHAR(row1,col1,ulc); /* linker oberer Eckpunkt */
    XYCHAR(row1,col2,urc); /* rechter oberer Eckpunkt */
    XYCHAR(row2,col2,lrc); /* rechter unterer Eckpunkt */
    XYCHAR(row2,col1,llc); /* linker unterer Eckpunkt */
    return (0);
}
int fill(int row1, int col1, int row2, int col2, int icon)
{
    int row;
    int hlen=col2-col1+1;
    int vlen=row2-row1;
    if (hlen<0 || vlen<0) /* Längenangaben waren fehlerhaft */
        return (1);
    if (icon==0) /* keine Füllung erwünscht */
        return (2);
    if (hlen==0 && vlen==0) /* es gibt nur einen Punkt */
    {
        XYCHAR(row1,col1,icon);
        return (0);
    }
    for (row=row1; row<=row2; row++)
        hlin(row,col1,hlen,icon);
    return (0);
}
/* ANSIW.C END */

```

ANSIC.C

```

/* ANSIC.C TESTPROGRAMM für ANSI-Bildschirmsteuerung in C */
#include <ansi.h> /* inkludiert auch
                  stdio.h und string.h */
#include <..\source\ansiw.c> /* Funktionen WINDOW(),
                             draw(), fill() */
#include <mem.h> /* _fmemset */
#include <dos.h> /* MK_FP */
#include <stdlib.h> /* exit() */
void ZeichneRaster(void)
{
    ...
}
void CursorTest(void)
{
    ...
}
void DeviceStatus(void)
{
    ...
}
void Bildschirmloeschen(unsigned char c,
                        unsigned char a1,
                        unsigned char a2,
                        unsigned char a3)
{
    ...
}
void Zeilenloeschen(void)
{
    ...
}
void FarbenTest(void)
{
    ...
}
void SetScreen(int mode, char *text)
{
    ...
}
void DisplayModi(void)
{
    ...
}
void ZeilenEndeTest(void)
{
    ...
}
void IntensitaetsTest(void)
{
    ...
}
void Fenster(int row1, int col1,
             int row2, int col2, char type, char *txt)
{
    ...
}
void FensterTest(void)
{
    ...
}
void EingabeTest(void)
{
    ...
}
void (*TestFunktion[])(void) =
{
    CursorTest,
    DeviceStatus,
    Zeilenloeschen,
    FarbenTest,
    DisplayModi,
    ZeilenEndeTest,
    IntensitaetsTest,
    FensterTest,
    EingabeTest
};

```

```

void main(void)
{
    unsigned char c;

    while (1)
    {
        SETSCREEN(2); /* 80x25, B/W */
        SETDISPLAY(0,0,0);
        CLS();
        printf("0. Ende\n");
        printf("1. Kursorstest\n");
        printf("2. Devicestatus\n");
        printf("3. Zeilenloeschen\n");
        printf("4. Farbentest\n");
        printf("5. Bildschirmmodi\n");
        printf("6. Zeilenendetest\n");
        printf("7. Intensitätssteuerung\n");
        printf("8. Fenstertest\n");
        printf("9. Eingabetest\n");

        do
        {
            GETCHE(c);
        }
        while ((c<'0') || (c>'9'));

        if (c=='0') exit(0);

        SETDISPLAY(0,0,0);
        CLS();
        TestFunktion[c-'1']();
        GETKB();
    }
}
/* ANSIC.C END */

```

Testprogramm ANSIC.C

Das Testprogramm ist zwar lang, wiederholt dabei aber nur eher gleichbleibende Sequenzen, daher ist es nur rudimentär abgedruckt, um auf eine Besonderheit hinzuweisen: In einer while-Schleife des Hauptprogramms main() können die einzelnen Tests unabhängig voneinander beliebig oft aufgerufen werden. Das Testprogramm ist ebenfalls in Funktionen zergliedert, allein um auch nach längerer Zeit noch einen klaren Überblick zu haben. Es werden keine Parameter übergeben, daher könnte man die Testfunktionen in einer großen switch-Anweisung unterbringen. Diese switch-Anweisung aber sucht man vergeblich, da hier ein Array von Pointern auf Funktionen

konstruiert wurde (TestFunktion) und das Hauptprogramm in Abhängigkeit vom Tastendruck auf die Funktion des jeweiligen Index verzweigt. Der Vorteil dieser Implementierung ist die übersichtliche Darstellung und die leichte Wartbarkeit bei Modifikationen. Eine weitere Verbesserung wäre die Erweiterung von TestFunktion durch weitere Parameter, die Auswahlstasten repräsentieren. Dann muß im Falle von Erweiterungen das Hauptprogramm nicht mehr angetastet werden, sondern nur mehr die Tabelle erweitert werden und die neue Testfunktion geschrieben werden.

Zusammenfassung

Der ANSI-Treiber ist ein selten angewendetes Programmelement. Wenn es aber gebraucht wird, mangelt es an Dokumentation und Unterstützung. Die hier vorgestellte C-Headerdatei ANSI.H ist eine Hilfe zur übersichtlichen Bildschirmsteuerung mit dem ANSI-Treiber. Das C-Programm ANSIV.C stellt einfache Funktionen zur Fenstergestaltung zur Verfügung. Alle Funktionen wurden mit dem Testprogramm ANSIC.C getestet. Durch die genaue Besprechung des Testprogramms sollte auf die Bedeutung des sorgfältigen Testens von Programmen hingewiesen werden.

Literatur

- /1/ DOS 3.30, Technical Reference, IBM, 1987.
- /2/ MS-DOS, Programmers Reference, Microsoft PRESS, 1991.
- /3/ Don F. Ridgway, A C Interface, BYTE, Nov. 1987, S.363.368.
- /4/ W. Riemer, Rückspulen der Tastatur in C, PC-NEWS-28 (3/1992), S.31.
- /5/ W.Riemer, F.Fiala, Elementare Ein-/Ausgabe mit XENIX-orientierten Funktionen, PC-NEWS-28 (3/1992), S.38. □
- /6/ ADIM-Band 40, Abschnitt über ANSI-Treiber

C-Sprachpolizei

War das PASCAL-Sprachbeispiel weiter vorne einfach, so können Sie mit dem Folgenden Ihrem C-Kompilier auf den Zahn fühlen: Wieviele Fehler finden Sie, wieviele findet Ihr Kompilier? CLINT findet 15 Fehler! Auflösung im nächsten Heft. Zum Ausprobieren, wie auch das PASCAL-Beispiel auf DSK-294.

```

/* How many of the 15 errors can your eyes spot
   in this code fragment
   (not counting the symbols not referenced).
   */

01 #include <stdio.h>
02 #include <stdlib.h>

03 #define FA 0x1
04 #define FB 0x2
05 #define FC 0x4
06 #define BOTH (FA | FC)

07 char name[11] = "Joe Jakeson";
08 int ounces[4] = { 1, 2, 3 };
/* how many struct's in w? no, its not 1 */
09 struct { int a[3], b; } w[] = { { 1, 2, 3 }, 2 };
10 enum color { red, blue, green } hue = blue;
11 char *as[] =
12 {
13     "Apricot Brandy"
14     "May Wine"
15     "Orange Pekoe"
16 };

```

```

17 char *report( int n, char *p )
18 {
19     int result;
20     int m = 30000 * 2;
21     void free(void *);
22     char *temp;
23     long nm;
24     int i, k, kk;

25     nm = n * m;
26     temp = p == "" ? "null" : p;
27     if( n > 40000 )
28         printf( "%s %d\n", temp, n );
29     switch(hue)
30     {
31     case red: k = 4; kk = 8; break;
32     case blue: k = 2 * k; kk = 16; break;
33     }
34     for( i = 0; i < kk; i++ )
35     {
36         if( nm + i == m ) { free(p); break; }
37     }
38     if( n & (BOTH & FB) ) m = 1;
39     if( n == 1 ) result = 0;
40     else if( n > 0 ) result = 1;
41     else if( n < 0 ) result = -1;
42     if( m == result ) return "";
43     else return p;
44 }

```

Hardwarenahes Programmieren

Teil 4: Hardwareprogrammierung

Franz FIALA, N, TGM

DSK-294:\MYLIB

Wir werden versuchen niederwertigste Aktionen (IN- und OUT-Befehle) mit höchstwertigen Softwaremethoden (Kapselung, Polymorphie, Vererbung) in C++ zu verbinden. Damit entsteht zwischen Hardware und Anwendung eine Zwischenschicht, die verhindert, daß in einem Programm von mehr als einem Programmteil auf einen Hardware-Port zugegriffen wird.

Leser, die sich mehr für die Anwendung als das 'Wie' interessieren, seien die Programme HC04PA1.EXE, HC04SE1.EXE zum Testen der parallelen und seriellen Schnittstelle und HC04CM1.EXE zum Auslesen und Verändern des CMOS-RAM empfohlen.

Gegenstand der Hardwareprogrammierung ist die Kommunikation mit den Ports der Peripheriebausteine. Wir können uns natürlich nur wichtige Beispiele ansehen; dabei wollen wir aber eine solide Basis in Form einer IO-Klasse entwickeln. Möglichst viele der Beispielprogramme sollen als wiederverwendbarer Kode in einer Bibliothek MYLIB.LIB zusammengefaßt werden. Diese Bibliothek soll mit jeder neuen Folge unserer Programmierübungen wachsen.

Eine Übersicht über die Aufteilung des IO-Bereichs kennen wir schon aus den PC-NEWS 27, S.32. Jetzt, beim Programmieren werden wir genauere Angaben über die eine oder andere Adresse benötigen. Hilfreich sind da die Beschreibungen der IO-Bausteine, siehe TGM-LIT-32.

Die Hardwareprogrammierung am Baustein hat nur in Ausnahmefällen eine Berechtigung:

- zum Lernen
- zur Entwicklung von Treibern
- zur Erschließung von Möglichkeiten, die auf höheren Ebenen nicht verfügbar sind.
- zum Testen

Mit Rücksicht auf Punkt a) werden wir auch einige Programme erproben, für die auf höheren Ebenen Ersatz vorhanden wäre.

Die direkte Kommunikation mit Hardware-Ports ist auch bei Windows-Programmierung nicht grundsätzlich unmöglich - es ist nur praktisch nicht anwendbar -, da sich Windows vorbehalten zu wissen, welchen Zustand der IO-Port oder die Hardware allgemein gerade hat.

Programmübersicht

In der folgenden Dateienübersicht sieht man die Beispielprogramme (HC04*.*). Während das erste vorgestellte Beispielprogramm HC04TA1*.* in vielen Varianten gezeigt wird, werden später die Beispielprogramme jeweils in der Art von HC04TA1G abgebildet. Auf den zugehörigen Disketten finden Sie aber oft auch eine Variante in Standard-C.¹

¹ Anmerkung: Vielen Programmierern ist zunächst die Benutzung von C++ noch fremd. Um den Umgang zu erleichtern, sind viele Beispiele in dieser und in den nächsten Folgen sowohl in einer C- als auch in einer C++-Version enthalten. Abgedruckt wird jeweils die C++-Version.

c:\mylib	
---lib	mit jeder Folge wachsende Bibliothek
---include	alle Headerdateien
---source	Quellkode
---sample	Beispielprogramme HC*.*
c:\mylib\lib\	
mylib.lib	Bibliothek
mylib.prj	Projektdatei für mylib.lib
c:\mylib\include\	
ansi.h	portable Bildschirmsteuerung in C (siehe Beitrag weiter vorne)
bit.h	Bitprozessor
cmosram.h	CMOSRAM-Klasse
io.h	IO-Klassen
iodef.doc	Dokumentation zu iodef.h
iodef.h	Definition von IO-Adressen im PC
keyscan.h	Alle SCAN-Kodes am PC
mylib.h	alle Funktionen
mytypes.h	INT, CHAR...
portable.h	erleichtert das Portieren von:
P_CONSOLE	Konsolenfunktionen
P_FILE	DOS-Funktionen
P_IO	IO-Ansprache
P_MEM	Speicherzugriffe
P_POINTER	MP_FP, LCODE, LDATA, NCODE, NDATA
P_SOUND	Tonausgabe
P_STREAMS	Streams
P_TIME	Zeitabfrage
c:\mylib\source	
cmosram.cpp	Funktionen der CMOSRAM-Klasse
file.c	filesize()
io.cpp	Funktionen aus IO-Klassen
kbd.c	kbd_read(), clear_kbdqueue()
mou.c	mouse_read()
mytypes.cpp	UCHAR, UINT, BOOL als Klasse
port_bc.c	in BORLANDC nicht enthaltene Funktionen
str.c	putstr(), convert_string()
tim.c	tim0_read(), u.a.
c:\MYLIB\SAMPLE\	
hc04.prj	kompiliert alle HC04*.*-Dateien
hc04bil.cpp	testen des Bitprozessors
hc04cm0.cpp	Beispiel für die Ansprache des CMOSRAM
hc04cm1.cpp	Lesen/Setzen des CMOS-RAM (C-Version)
hc04cm1.exe	Lauffähige Version von hc04cm1.cpp
hc04cm1a.cpp	Lesen/Setzen des CMOS-RAM (C++-Version)
hc04cm1a.exe	Lauffähige Version von hc04cm1a.cpp
hc04io0.cpp	Demo für Grundfunktionen der IO-Klasse
hc04io1.cpp	Testen der Klassen IOBYTE_IN, -OUT, -IO
hc04io2.cpp	Anzeige/Änderung von IO-Ports
hc04io2.exe	Lauffähige Version von hc04io2.cpp
hc04io3.cpp	Testen der Klassen IOBIT_IN, -OUT, -IO
hc04io4.cpp	Demo für die Bearbeitung von Bits
hc04io5.cpp	Ausschalten der Tastatur am XT
hc04mol.cpp	Polling des Maus-Ports
hc04pal.cpp	Testen der parallelen Schnittstelle
hc04pal.exe	Lauffähige Version von hc04pal.cpp
hc04sel.cpp	Testen der seriellen Schnittstelle
hc04sel.exe	Lauffähige Version von hc04sel.cpp
hc04ta1a.c	Polling der Tastatur, BORLAND-Version
hc04ta1b.c	--, ANSI-Version
hc04ta1c.c	--, MAKRO-Version
hc04ta1d.cpp	--, CPP-Version
hc04ta1e.cpp	--, Funktions-Version
hc04ta1f.cpp	--, Bibliotheks-Version
hc04ta1g.cpp	--, IO-Klassen-Version
hc04ti0.cpp	Polling des Timers

Headerdateien port*.H

Gerade im nicht-genormten, hardwarenahen Bereich gibt es große Unterschiede zwischen den Compilern. Bei einem Wechsel des Compilers müssen viele Zeilen mit `#if..#endif`-Konstruktionen angepaßt werden. Einiges davon läßt sich durch geeignete Header-Dateien beseitigen. In einem späteren Beitrag wird die Datei `portable.h` vollständig vorgestellt, hier nur ein Auszug:

IO-Zugriff in C

Jede höhere Sprache enthält eine Entsprechung für die IO-Befehle, so auch C. Bei TURBO-C(BORLAND-C) gibt es die Funktionen `inport()` und `outport()`.

Assembler	TURBO-C
IN AL,adr	unsigned char inportb(unsigned int adr);
IN AL,DX	
IN AX,adr	int inport(unsigned int adr);
IN AX,DX	
OUT adr,AL	outportb(unsigned int adr, unsigned char c);
OUT DX,AL	
OUT adr,AX	outport(unsigned int adr, int c);
OUT DX,AX	

Es gibt für jeweils 2 Assemblerbefehle einen C-Befehl. Das liegt daran, daß man im Adreßbereich 0..255 (0x00..0xff) direkt oder registerDX-indirekt adressieren kann und über der Adresse 255 nur über das DX-Register.

Bei Microsoft heißen diese Funktionen aber `inp()` und `outp()`¹. Damit wir kompilierunabhängig arbeiten können, empfiehlt es sich, selbstdefinierte Makros zu verwenden, die in der Header-Datei `portio.h` je nach Compiler entsprechend expandieren. In unserem Fall unterscheidet sich TURBOC vom Rest der C-Kompiler. TURBOC verwendet `inportb()`, die anderen Compiler `inp()`. Das Makro `IN_PORT` substituiert, je nach Compiler die richtige Funktionsbezeichnung.

```

/*-----
I/O Port Macros
-----*/
IN_PORT    read byte from I/O port
IN_PORTW   read word from I/O port
OUT_PORT    write byte to I/O port
OUT_PORTW  write word to I/O port
-----*/

#if defined(__TURBOC__)
  #ifndef __DOS_H
    #include <dos.h>
  #endif
  #define IN_PORT(port)    inportb(port)
  #define IN_PORTW(port)   inport(port)
  #define OUT_PORT(port,val) outportb(port,val)
  #define OUT_PORTW(port,val) outport(port,val)
#else
  #ifndef __CONIO_H
    #include <conio.h>
  #endif
  #define IN_PORT(port)    inp(port)
  #define IN_PORTW(port)   inpw(port)
  #define OUT_PORT(port,val) outp(port,val)
  #define OUT_PORTW(port,val) outpw(port,val)
#endif

```

¹ Die hier beschriebenen Programme wurden zunächst mit BORLAND-C 2.0 geschrieben, Mittlerweile hat man bei BORLAND auch die Funktionsbezeichnungen von Microsoft 'ins Programm' aufgenommen, sodaß in diesem Punkt die beschriebenen Makros entbehrlich geworden sind.

An dieser Stelle könnte man sich fragen, ob die Art Ports anzusprechen, wie es die 80x86-Familie tut, wünschenswert ist oder ob man nicht grundsätzlich memory-mapped IO, wie bei den Motorola-CPU's vorziehen sollte. Auch die Konstruktion von IO-Bausteinen, besonders der älteren Baureihen unterliegen keiner einheitlichen Konzeption, sodaß die nachfolgende Software viele Konstruktionsmängel der Chips ausgleichen muß, in vielen Fällen aber diese Mängel nicht mehr ausgleichen kann.

Nachteile des IO-Konzepts der 80x86 CPUs

- * Beschränkung des IO-Adreßraumes (sowohl CPU- als auch konstruktionsbedingt)
- * Eingeschränkte Zugriffsmöglichkeiten über IN und OUT
- * Daten und Programme in verschiedenen Speicherbereichen

Vorteile des IO-Konzepts der 80x86 CPUs

- * Der IO-Speicher kann mit Wartezyklen betrieben werden ohne, daß der Hauptspeicherzugriff darunter leidet.
- * Ob ein Programm eine Ein-/Ausgabe durchführt ist mnemonicisch erkennbar.
- * IO-Zugriffe können generell verhindert werden

Besonderheiten herkömmlicher Chipkonzeptionen

- * Nicht rücklesbare Ports zwingen den Programmierer eine Kopie der Einstellung anzulegen. Schwierig verwaltbar, wenn auch andere, gleichzeitig laufende Programm die Einstellung ändern wollen. Deshalb ist auch unter Windows und anderen Multitasking-Systemen der Portzugriff untersagt.
- * Bedeutungswechsel beim Zugriff auf eine Adresse (um Adreßraum zu sparen) Beispiel: Interrupt-Kontroller (Programmierung über eine Sequenz von Steuerbefehlen auf derselben Adresse) oder Timer (Ablesen und Schreiben des Zählerstandes in zwei Zugriffen auf derselben Adresse)
- * Vom Hauptspeicher abweichende Zugriffszeiten

Tastatur, Polling HC04TA1A.C

Die Tastatur stellt einen sehr komplexen Bestandteil des PC dar. Immerhin werden 2 Mikrokontroller gebraucht (einer auf der Tastatur, einer im PC (ab AT)), um Tastatur und PC zu koordinieren. Jede Tastenberührung veranlaßt den Mikrokontroller in der Tastatur einen sogenannten SCAN-Kode gemeinsam mit einem Takt an den PC abzusenden. Der empfangende Mikrokontroller 8742 liefert den Kode auf IO-Adresse 0x60 ab und aktiviert die Interruptleitung IRQ1. Der PC und sein BIOS müssen jetzt reagieren. IRQ1 ruft die Interruptserviceroutine ISR9. Diese liest den Scan-Kode von IO-Port 0x60, quittiert den Empfang an Port 0x64 und füllt ein zu dem SCAN-Kode passendes ASCII-Zeichen in die Tastaturwarteschlange. Ist diese voll, piepst! Genaueres zu diesem Vorgang und auch eine Liste dieser SCAN-Kodes siehe im Beitrag "Dem PC auf die Tasten geschaut", PC-NEWS-21(1/91), S.15..21 und im vorliegenden Heft der Beitrag "Hot-Key...".

Unter **Polling** versteht man die zyklische Abfrage einer Datenquelle, bei der man eine von außen kommende Änderung erwart

tet. Polling ist schnell - wenn man sonst nichts zu tun hat. Ist Polling allerdings in ein komplizierteres Programm eingebunden, sinkt die Abfragehäufigkeit, dann ist der Zugriff über Interrupts bedeutend schneller, obwohl bei Interrupts mit einem Overhead gerechnet werden muß.

HC04TA1A.C

Bevor wir den Vorgängen rund um die Tastatur in einer späteren Folge im Detail nachgehen, wollen wir durch Polling feststellen, wie diese SCAN-Kodes beschaffen sind. Das Programm fragt eine Änderung an Port 0x60 ab und stellt den Code am Bildschirm dar. Wir sind noch weit vom ASCII-Code entfernt, es kommt nur eine laufende Nummer pro Taste. Die Taste mit Nummer 1 bricht das Programm ab! Welche ist das?

Wir brauchen eine Hilfsvariable `oldkey`, in der der jeweils aktuelle Wert an Port 0x60 gespeichert wird. Solange sich dieser Wert nicht ändert, wird nichts ausgegeben. Die Änderung bewirkt die Ausgabe und außerdem wird `oldkey = key;` weiter geht!

Wieviele Codes werden pro Tastenbetätigung geschickt? Wie unterscheiden sich diese? Gibt es auch Tasten, die mehr als einen Code senden? Wie verhalten sich die Tasten SHIFT, CTRL, ALT? Sendet jede Taste einen Code? Haben die Codes eine Systematik? Wie sieht die Kodefolge beim Liegenlassen der Taste aus?

Programmierstil HC04TA1B.C..1F.CPP

So, wie das vorige Programm sehen etwa Schülerprogramme aus. Es werden BORLAND-spezifische Funktionen hier `cprintf()` und `clrscr()` verwendet, daher ist das Programm weniger portierbar. Wir wollen das Programm durch einige Maßnahmen etwas höherwertiger gestalten.

HC04TA1B.C

Dieses und das folgende Beispiel zeigen, wie man die Portierbarkeit auf andere Rechner und Compiler erhöhen kann. Eine Maßnahme zur Verbesserung der Portierbarkeit ist die Verwendung von ANSI-Funktionen, `printf()` statt `cprintf()` und `CLS()` statt `clrscr()`. `CLS()` ist ein Makro, das unter Verwendung von `printf()` den ANSI-ESC-Kode für die Bildschirmlöschung an `stdout` absendet, vorausgesetzt, der Treiber `ANSI.SYS` ist geladen.

<u>BORLANDC</u>	<u>ANSI-C</u>	
<code>cprintf()</code>	<code>printf()</code>	
<code>conio.h</code>	<code>stdio.h</code>	
<code>clrscr()</code>	<code>CLS()</code>	<code>/* printf("\033[2J") */</code>

HC04TA1C.C

Ersetzt man die konkreten Zahlenwerte im Programm durch Makros, kann der Leser besser erkennen, was die Zahlenwerte bedeuten. Ebenfalls der Zahlenwert für den SCAN-Kode, der das Programm bestimmt. Die an und für sich nicht portierbare Funktion `inportb()` existiert nur bei BORLAND und wird daher durch ein Makro ersetzt, das für jeden neuen Compiler einfach anpaßbar ist. Makros erhöhen die Portierbarkeit!

HC04TA1D.CPP

Die so definierten Adressen und Codes werden am besten in eigenen Header-Dateien (`iodef.h`, `keyscan.h`) zusammengefaßt, damit sie später bei Bedarf wieder zur Verfügung stehen. Damit bekommen alle Programme zunehmend einheitliches Aussehen.

Auch wollen wir die neuen Möglichkeiten von C++ nutzen, daher wird die Dateieindung auf `CPP` geändert. Wir verwenden zwar noch keine selbstdefinierten Klassen (das kommt erst in der Version `HC04TA1G.CPP` dieses Programms) aber wir benutzen aus der Stream-Bibliothek die Objekte `cout` und `cin`, sowie die überladenen Operatoren `<<` und `>>`.

<u>C</u>	<u>C++</u>
<code>printf</code>	<code>cout <<</code>
<code>scanf</code>	<code>cin >></code>
<code>printf("%02x\n")</code>	<code>cout << setfill('0') << setw(2) << hex << endl</code>

HC04TA1E.CPP

Unser Ziel war wiederverwertbarer Code. Darunter versteht man die Aufspaltung eines 'Bandwurm'-programms in Funktionen mit allgemeiner Verwendungsmöglichkeit. Damit wird das Programm immer etwas länger und langsamer aber auch lesbarer (, da anstelle eines 'Bandwurm'-teils eine verständliche Funktionsbezeichnung tritt) und eben wiederverwendbarer (, da die Funktion gleich etwas allgemeiner abgefaßt wird, um auch in anderen Fällen zur Stelle zu sein).

In unserem Beispiel bieten sich dazu zwei Dinge an: Löscher des Tastaturbuffers (`clear_kbdqueue()`) und Lesen des nächster Scankodes (`read_kbd()`). Diese Funktionen werden in diesem Schritt aus dem Hauptprogramm herausgelöst und selbständig dokumentiert ans Programmende gestellt. An den Programmstart kommen die Prototypen dieser Funktionen. Die Funktion `read_kbd()` wird dabei gleich etwas bereichert, es gibt einen Parameter, ähnlich, wie bei `bioskey()`, wobei 0 'Warten auf Änderung' bedeutet und 1 'Lesen ohne Warten'.

HC04TA1F.CPP

Wenn man bei jedem Programm diese Trennung in verwendbare Funktionen durchführt, müßte man beim Wiederverwenden den jeweiligen Funktionstext in das nächste Programm kopieren. Dabei entstünden im Laufe der Zeit so viele Programmversionen als Programmkopien angefertigt wurden, da man immer mit Änderungen rechnen muß. Es gibt zwei Möglichkeiten, das zu verhindern: Man ordnet Funktionscode und Hauptprogramm in getrennten Dateien an und (1) inkludiert immer den Sourcecode des Funktionsteils in den Modulen, in denen dieser benötigt wird oder (2) bildet eine eigene Bibliothek aus diesen Funktionen, hier `mylib.lib`, und beschreibt diese durch eine Headerdatei `mylib.h`, wie das auch der Compiler mit der C-Bibliothek macht. Mit jedem neu geschriebenen Programm wächst die eigene Bibliothek, mit jeder Wiederverwendung werden die Module perfekter, da immer mehr zunächst unerkannte Fehler ausgemerzt werden. Die beiden Funktionen `kbd_read()` und `clear_kbdqueue()` werden gemeinsam mit vielen anderen Funktionen zur Bibliothek `MYLIB.LIB` zusammengefaßt. Damit die Programme auch ohne die Bibliothek ablaufen und bei Änderungen der Module nicht immer die gesamte Bibliothek neu gebildet werden muß, wird in diesem Programm der Quellcode `kbd.c` bedingt eingebunden. Wird in der Projektdatei für die Bibliothek dann das Makro `MYLIB` definiert, entfällt die Einbindung des Source-Code, das Programm muß nicht umgeschrieben werden. Diese Maßnahme werden Sie bei allen folgenden Programmen finden.

```

/* HC04TA1A.C */
/*
 * Tastatur-kodes prüfen, BORLAND-Version
 * =====
 */
#include <dos.h> /* inportb() */
#include <bios.h> /* bioskey() */
#include <conio.h> /* cprintf() */

void main(void)
{
  unsigned char key, oldkey;
  clrscr();
  cprintf("Tastatur-SCAN-Codes, Ende mit ESC\n\r");
  oldkey=inportb(0x60);
  do
  {
    key=inportb(0x60);
    if (key!=oldkey)
    {
      cprintf("%2x ",key);
      oldkey=key;
    }
    // Abholen und ignorieren des Tastenkode
    while (bioskey(1)) bioskey(0);
  }
  while (key!=1);
  clrscr();
}

```

```

/* HC04TA1B.C */
/*
 * Tastatur-kodes prüfen, ANSI-Version
 * =====
 */
#include <dos.h> /* inportb() */
#include <bios.h> /* bioskey() */
#include <stdio.h> /* printf() */
#include <stdlib.h> /* exit() */
#include <ansi.h> /* CLS() */
/* ansi.h: #define CLS() printf("\033[2J") */

void main(void)
{
  unsigned char key, oldkey;
  CLS();
  printf("\n\nTastatur-SCAN-Codes, Ende mit ESC\n");
  oldkey=inportb(0x60);
  do
  {
    key=inportb(0x60);
    if (key!=oldkey)
    {
      printf("%2x ",key);
      oldkey=key;
    }
  }
  while (key!=1);
  CLS();
}

```

```

/* HC04TA1C.C */
/*
 * Tastatur-kodes prüfen, mit Makros
 * =====
 */
#include <bios.h>
#include <conio.h>
#include <mytypes.h>

#define PC_KBD_DATA 0x60
#define SCAN_ESCAPE 1
#define IN_PORT inportb

VOID main(VOID)
{
  UCHAR key, oldkey;
  clrscr();
  cprintf("Tastatur-SCAN-Codes, Ende mit ESC\n\r");
  oldkey=IN_PORT(PC_KBD_DATA);
  do
  {
    key=IN_PORT(PC_KBD_DATA);
    if (key!=oldkey)
    {
      cprintf("%2x ",key);
      oldkey=key;
    }
    while (bioskey(1)) bioskey(0);
  }
  while (key!=SCAN_ESCAPE);
  cprintf("\n\r");
  clrscr();
}

```

```

/* HC04TA1D.CPP */
/*
 * Tastatur-kodes prüfen, CPP-Version
 * =====
 */
#include <bios.h> // bioskey()
#include <portable.h> // cout, setw(), setfill()
// IN_PORT
#include <mytypes.h> // VOID, UCHAR...
#include <iodef.h> // PC_KBD_DATA
#include <keyscan.h> // SCAN_ESCAPE

VOID main(VOID)
{
  UCHAR key, oldkey;
  cout << "Tastatur-SCAN-Codes, Ende mit ESC" << endl;
  oldkey=IN_PORT(PC_KBD_DATA);
  do
  {
    key=IN_PORT(PC_KBD_DATA);
    if (key!=oldkey)
    {
      cout << setfill('0') << setw(2) << hex
        << (UINT)key << ' ';
      oldkey=key;
    }
    while (bioskey(1)) bioskey(0);
  }
  while (key != SCAN_ESCAPE);
  cout << endl << endl;
}

```

```

/* HC04TA1E.CPP */
/*
 * Tastatur-kodes prüfen, wiederverwendbarer Kode
 * =====
 */
#include <bios.h> // bioskey()
#include <iodef.h> // PC_KBD_DATA
#include <keyscan.h> // SCAN_ESCAPE
#include <mytypes.h> // INT, UCHAR...
#include <portable.h> // IN_PORT
// cout, setw(), setfill()

INT cdecl kbd_read(INT mode);
VOID cdecl clear_kbdqueue(VOID);

VOID main(VOID)
{
  UCHAR key;
  cout << "Tastatur-SCAN-Codes, Ende mit ESC" << endl;
  do
  {
    key=kbd_read(0);
    if (key!=SCAN_ESCAPE)
      cout << setfill('0') << setw(2) << hex
        << (UINT)key << ' ';
    clear_kbdqueue();
  }
  while (key!=SCAN_ESCAPE);
  cout << endl << endl;
}

/*
 * UCHAR kbd_read(INT mode)
 * =====
 * ENTRY: mode=0 ... wait for change on kbd-port
 * mode=1 ... immediate return
 * EXIT: returns SCAN-code
 */

INT cdecl kbd_read(INT mode)
{
  UCHAR key;
  static UCHAR oldkey=0;

  switch (mode)
  {
    case 0:
      do
      {
        key=IN_PORT(PC_KBD_DATA);
        while (key==oldkey);
        break;
      }
      case 1:
        key=IN_PORT(PC_KBD_DATA);
        break;
  }
  oldkey=key;
  return key;
}

/*
 * VOID clear_kbdqueue(VOID)
 * =====
 * clears keyboard queue
 */

VOID cdecl clear_kbdqueue(VOID)
{
  while (bioskey(1)) bioskey(0);
}

```

Maus, Polling HC04MO1.CPP

Diese schrittweise Verallgemeinerung des Codes wurde nur zu Beginn gezeigt. Jetzt wollen wir versuchen, die letzte Version auf einen anderen Port zu übertragen.

Jeder Port, an dem sich ändernde Daten anliegen, kann ähnlich abgefragt werden. Das folgende Programm liest die Daten von der Maus. Es setzt voraus, daß die Maus an COM-Port-1 angeschlossen ist. Den exakten Wert für andere Ports entnehmen Sie in Analogie zu 0x3f8 aus der Belegungstabelle der IO-Ports in `iodef.h`. Die Funktion `mouse_read()` entspricht der Funktion `kbd_read()`, hat aber einen zusätzlichen Parameter bekommen, der die Portnummer bestimmt.

```

/* HC04MO1.CPP */
/*
 * Maus-Kodes prüfen
 * =====
 */
#include <portable.h> // cout
#include <keyscan.h> // SCAN_ESCAPE
#include <mylib.h> // mouse_read(), clear_kbdqueue()
#ifndef MYLIB
#include "\mylib\source\mou.c"
#include "\mylib\source\kbd.c"
#endif

#define COMPORT 1
#define MOUSE_SCAN 1
#define MOUSE_WAIT 0

VOID main(VOID)
{
    UCHAR mouse, oldmouse=0;
    constream con;
    con.clrscr();
    con << "Maus-Daten, Ende mit Taste" << endl << endl;
    do
    {
        mouse=mouse_read(COMPORT,MOUSE_SCAN);
        if (mouse!=oldmouse)
        {
            con << setw(2) << setfill('0') << hex
              << (UINT)mouse << ' ';
        }
        oldmouse=mouse;
    }
    while (kbhit());
    clear_kbdqueue();
    clrscr();
}

```

Die Funktion `mouse_read()` ist in der Datei `mou.c` zu finden, der Prototyp in `mylib.h`, der übersetzte Funktionscode in `mylib.lib`.

Wenn Sie dieses Programm testen, können Sie versuchen herauszufinden, welche Codes die Maus in bestimmten Situationen sendet und welchen Code die verschiedenen Maustasten haben!

Timer, Polling HC04TI0.CPP

	Adresse	Schreiben	Lesen
Channel 0	0x40	Anfangszählerstand	Zählerstand
Channel 1	0x41	Anfangszählerstand	Zählerstand
Channel 2	0x42	Anfangszählerstand	Zählerstand
Control	0x43	Steuerwort	-

Die Programmierung der Timer sehen wir uns später genauer an, jetzt wollen wir nur das Auslesen der Timer versuchen. Wir haben 3 Kanäle mit je 16 bit Zählbreite zur Verfügung. Die Timer werden mit derselben Frequenz von 1,19 MHz gespeist, haben aber verschiedene Anfangszählerstände. Der Timer, beginnt beim programmierbaren Anfangszählerstand, nach abwärts, Richtung 0 zu zählen und beginnt danach wieder von vorne.

Das Programm liest zweimal hintereinander dieselbe Adresse, in unserem Programmbeispiel 0x40 (=Kanal 0). Wollten wir die beiden anderen Kanäle auslesen, müßten wir die Adressen 0x41 (Kanal 1) oder 0x42 (Kanal 2) einsetzen. Der Timer ist so programmiert, daß

beim ersten Lesevorgang das niederwertige, danach das höherwertige Byte gelesen wird.

Wären die vorangegangenen Beispiele nur von der Benutzerreaktion abhängig (die Reaktion erfolgte mit unmerklicher Verzögerung), haben wir es hier mit der Beobachtung eines raschen Zählvorganges durch ein Programm zu tun. Die Periodendauer des Zählvorganges ist etwas kleiner als 1 µs. Aus dem Abstand zweier abgelesener Werte können Sie sich ein Bild von der Geschwindigkeit von C machen: Der Programmabbruch erfolgt nicht durch die Funktion `bioskey()`, sondern nach 300 Abfragen. Je nachdem, wie man die Bildschirmausgabe anordnet (innerhalb oder außerhalb der Polling-Schleife, im Listing durch Anmerkungen `///
//` in der for-Schleife unwirksam gemacht) ist der Abfrage-Abstand um Größenordnungen verschieden. 386SX-Notebook, 20MHz:

Bildausgabe außerhalb: 15 µs

Bildausgabe innerhalb: 2700 µs

Stellen Sie fest, welchen Zählzyklus die drei Timer haben (Anfangswerte)! Welcher Timer zählt nicht? (Programm `HC04TI05.CPP` schaltet ihn ein, Objekt `tim2gat` genügt.)

```

/* HC04TI0.CPP */
/*
 * Lesen des Timers
 * =====
 */
#include <portable.h>
#include <mylib.h>
#ifndef MYLIB
#include "\mylib\source\tim.c"
#endif

#define READINGS 300
LONG timer_readings[READINGS];

VOID main(VOID)
{
    constream con;
    con.clrscr();

    for (INT i=0; i<READINGS; i++)
    {
        timer_readings[i]=tim0_read();
        // cout << setw(4) << setfill('0') << hex
        // << timer_readings[i] << ' ';
    }

    ULONG differenz=0;
    INT n=0;

    for (i=0; i<READINGS; i++)
    {
        if (i)
        {
            if (timer_readings[i]<timer_readings[i-1])
            {
                n++;
                differenz+=
                    (timer_readings[i-1]-timer_readings[i]);
            }
        }
        cout << setw(4) << setfill('0') << hex
          << timer_readings[i] << ' ';
    }
    cout << endl
      << n << " Differenzen gemittelt" << endl;
    cout << "Zählerstandsdifferenz : "
      << dec << (UINT)(differenz/n) << endl;
    cout << "Schleifendurchlaufzeit : "
      << (differenz/n)/(1.19) << " us" << endl;
    cin.get();
}

```

Aus den Beobachtungen können wir mehrere Schlüsse ziehen: Das Ablesen der Zählerstände ist zwar möglich, ein bestimmter Zählerstand kann aber nicht abgefragt werden. Am besten ist noch eine Abfrage auf kleiner gleich möglich. Es ist daher wesentlich präziser mit dem Timer-Interrupt zu arbeiten, als sich auf eine Abfrage einzulassen; das wollen wir später unternehmen. Jetzt können wir uns merken, daß die Interruptroutine sparsam zu programmieren sein wird, will man dem Vordergrundprogramm nicht allzuviel Zeit wegnehmen, dh. z.B. keine Bildausgaben.

PC-Adressen auf einen Blick

Während es in der vorigen Folge nur darum ging, die Portadressen zu nennen, müssen wir heute dem C-Programm mitteilen, wie die Ports heißen sollen und welchen Wert sie haben. Aus einigen Quellen zusammengesucht und im ständigen Wachsen begriffen ist die folgende Headerdatei IODEF.H, die schon bei den vorangegangenen Beispielen angewendet wurde:

```
#ifndef __IODEF_H
#define __IODEF_H

// Bitadressen B... 0xrb
// r...relative Adresse des IO-Bausteins
// b...Bitnummer

#define PC_DMA 0x000
#define PC_PIC 0x020
#define AT_CHIPS_ADR 0x022 /* OUT */
#define AT_CHIPS_DAT 0x023 /* IO */
#define PC_TIM 0x040
#define PC_KBD_DATA 0x060
#define XT_PPI 0x060
#define XT_PPI_A 0x060
#define XT_PPI_B 0x061
#define XT_PPI_C 0x062
#define XT_PPI_CMD 0x063
#define AT_KBD_DATA 0x060
#define AT_KBD_CMD 0x064
#define AT_CMOS_ADR 0x070
#define AT_CMOS_DAT 0x071
#define AT_NMI_MASK 0x070
#define AT_DIAG 0x080
#define PC_DMAPG_0 0x080
#define PC_DMAPG_1 0x081
#define PC_DMAPG_2 0x082
#define PC_DMAPG_3 0x083
#define PC_NMI_MASK 0x0a0
#define AT_PIC 0x0a0
#define AT_DMA 0x0c0
#define AT_287_LATCH_CLEAR 0x0f0
#define AT_287_RESET 0x0f1
#define AT_287_DATA 0x0f8
#define AT_HD1 0x1f0
#define PC_GAME 0x200
#define XT_EXPANSION 0x210
#define PC_LPT3 0x278
#define PC_EGA2 0x2c0
#define PC_COM4 0x2e8
#define PC_COM2 0x2f8
#define PC_PROTOTYPE 0x300
#define XT_HD 0x320
#define PC_FD2 0x370
#define PC_LPT2 0x378
#define PC_SDL1 0x380
#define PC_MDA1 0x3b0
#define PC_LPT1 0x3bc
#define PC_EGA1 0x3c0
#define PC_CGA 0x3d0
#define PC_COM3 0x3e8
#define PC_FD1 0x3f0
#define PC_COM1 0x3f8

/* LPT */
#define LPT_DATA 0
#define LPT_STATUS 1
#define LPT_CTRL 2
#define B_LPT_ERROR 0x14
#define B_LPT_PAPER 0x15
#define B_LPT_ACK 0x16
#define B_LPT_BUSY 0x17
#define B_LPT_STROBE 0x20
#define B_LPT_AUTOFEED 0x21
#define B_LPT_INIT 0x22
#define B_LPT_SELECT 0x23
```

```
/* COM */
#define COM_DATA 0 /* DLAB=0 */
#define COM_IEN 1 /* DLAB=0 */
#define COM_DIV_LSB 0 /* DLAB=1 */
#define COM_DIV_MSB 1 /* DLAB=1 */
#define COM_IID 2
#define COM_CONTROL 3
#define COM_MODEM 4
#define COM_LINE_STATUS 5
#define COM_MODEM_STATUS 6
#define B_COM_LINE_DCD 0x67
#define B_COM_LINE_RI 0x66
#define B_COM_LINE_DSR 0x65
#define B_COM_LINE_CTS 0x64
#define B_COM_LINE_DDCD 0x63
#define B_COM_LINE_DRI 0x62
#define B_COM_LINE_DDSD 0x61
#define B_COM_LINE_DCTS 0x60
#define B_COM_LINE_TX 0x56
#define B_COM_LINE_TH 0x55
#define B_COM_LINE_BREAK 0x54
#define B_COM_LINE_FE 0x53
#define B_COM_LINE_PE 0x52
#define B_COM_LINE_OR 0x51
#define B_COM_LINE_RX 0x50
#define B_COM_MODEM_LOOP 0x44
#define B_COM_MODEM_OUT2 0x43
#define B_COM_MODEM_OUT1 0x42
#define B_COM_MODEM_RTS 0x41
#define B_COM_MODEM_DTR 0x40
#define B_COM_CONTROL_DLAB 0x37
#define B_COM_CONTROL_BREAK 0x36
#define B_COM_IID 0x20 /* 0=Interrupt */
#define B_COM_IEN_MODEM 0x13 /* DLAB=0 */
#define B_COM_IEN_LINE 0x12 /* DLAB=0 */
#define B_COM_IEN_TX 0x11 /* DLAB=0 */
#define B_COM_IEN_RX 0x10 /* DLAB=0 */

/* XT-EXPANSION */
#define XT_EXPANSION_CONTROL 0
#define XT_EXPANSION_ENABLE 3
#define XT_EXPANSION_DATA 5
#define XT_EXPANSION_ADDRESS_HIGH 5
#define XT_EXPANSION_ADDRESS_LOW 6

/* TIMER */
#define TIM_CH0 0
#define TIM_CH1 1
#define TIM_CH2 2
#define TIM_CTL 3

/* PPI 8055 */
#define PPI_A 0
#define PPI_B 1
#define PPI_C 2
#define PPI_CMD 3
/* A */ /* IN */
/* Keyboard-Data */

/* B */
#define B_PPI_KBDENABLE 0x17
#define B_PPI_KBDCLOCKLOW 0x16
#define B_PPI_IOCTLCHECKENABLE 0x15
#define B_PPI_RAMCHECKENABLE 0x14
#define B_PPI_DIPSWITCH 0x13
#define B_PPI_TIM2DATA 0x11
#define B_PPI_TIM2GATE 0x10

/* C */
#define B_PPI_RAMCHK 0x27 /* IN */
#define B_PPI_IOCTL 0x26 /* IN */
#define B_PPI_TIM2OUT 0x25 /* IN */
#define B_PPI_REFRESH 0x24 /* IN */
#define B_PPI_DIP3 0x23 /* IO */
#define B_PPI_DIP2 0x22 /* IO */
#define B_PPI_DIP1 0x21 /* IO */
#define B_PPI_DIP0 0x20 /* IO */

#endif // IODEF_H
```

Adressen, die bei allen PC-Varianten gleich sind, werden mit PC_ eingeleitet, sonst wird XT_ und AT_ zur Kennzeichnung verwendet.

Bei allen Portbausteinen, die sich im Adreßbereich wiederholen, wurde lediglich die Basisadresse im Verzeichnis aufgenommen. Dazu kommen dann die relativen Adressen der einzelnen Register. Beispiel PC_LPT3=0x278. Will man den Statusport dieser Schnittstelle ansprechen, schreibt man PC_LPT3+LPT_STATUS. Ähnliches gilt auch für die serielle Schnittstelle.

Eigens zusammengesetzt sind Bitadressen und beginnen mit B_. Da bei vielen Ports jedes Bit eine andere Bedeutung hat, wurde die Schreibweise BITADR=0xpb gewählt, wobei p die relative Portnummer ist und b die Bitnummer des Anschlusses. Ein Beispiel für die Anwendung der selbstdefinierten Bitadressen finden Sie in HC04IO5.CPP.

Diese Headerdatei ist keineswegs vollständig, sie kann jeweils nach Bedarf erweitert werden.

Allgemeine Ein- Ausgabe, IO-Klassen IO.HPP, IO.CPP

Haben Sie es auch bemerkt? Wir haben zwar erfolgreich ein für die Tastatur, für die Maus und auch für den Timer anwendbares Unterprogramm geschrieben. Diese Programme unterscheiden sich aber nur in der Adressen, die sie verwenden. Man kann sich daher eine allgemeinere Ansprache der IO-Ports vorstellen, als es bisher geschah. C++ bietet uns dabei Hilfen von bisher unbekannter Qualität an.

Während Hauptspeicheradressen bei Belegung durch eine Variable vielfältig bearbeitbar sind (+,-,*,++,>>,<<..) muß man beim Handhaben von IO-Ports im allgemeinen eine Kopie des Portinhalts im Hauptspeicher bearbeiten und dann wieder an der Port übergeben. Bearbeitungsreihenfolge:

Lesen	inport()
Bearbeiten	+*++>>...
Verändern	outport()

Beim Lesen und Verändern muß immer auch darauf geachtet werden, die richtige Adresse als Operand zu übergeben. Außerdem gibt es IO-Ports, die nur zum Lesen geeignet sind, andere nur zum Schreiben und wieder andere schreib- und lesbar sind, wobei hier wieder unterschieden werden muß zwischen Ports, die rücklesbar sind (parallele Schnittstelle) und Ports, die in Schreib- und Leserichtung verschiedene Funktion haben (serielle Schnittstelle). Dazu kommen auch komplexere IO-Bedienungsfunktionen, wie sequentiell mehrere Bytes mit verschiedener Funktion nacheinander zu schreiben. Besonders die nicht-rücklesbaren Ports erfordern vom Programmierer die Verwaltung einer Variablen im Hauptspeicher, die den jeweils aktuellen Wert enthält. Es gibt also beim Umgang mit IO-Ports Vieles zu berücksichtigen.

Bei der Bemühung um Verallgemeinerung der Portansprache wurde eine C++-Klasse `io` mit daraus abgeleiteten Klassen `IOBYTE`, `IOBYTE_IN`, `IOBYTE_OUT`, `IOBYTE_IO`, `IOBYTE_LOOP`, `IOBIT_IN`, `IOBIT_OUT`, `IOBIT_IO`, `IOBIT_LOOP` gebildet. Diese Klassen haben folgenden Zusammenhang:

<code>IO</code>	Basisklasse
<code>-IOBYTE</code>	Simulation ohne Portzugriff
<code>-IOBYTE_IN</code>	Byte-Input-Port
<code>-IOBYTE_OUT</code>	Byte-Output-Port
<code>-IOBYTE_IO</code>	Byte-Rücklesbarer Port
<code>-IOBIT_IN</code>	Bit-Input-Port
<code>-IOBIT_OUT</code>	Bit-Output-Port
<code>-IOBIT_IO</code>	Bit-Rücklesbarer Port
<code>IOBIT_LOOP</code>	Testschleife zweier Bits

Alle Klassen sind in der Datei `IO.HPP` definiert. Die Namen sind größtenteils selbsterklärend. Die meisten Funktionen sind `inline`-Funktionen, erfordern daher keinen Funktionsaufruf bei der Ausführung. Der Code ist in `IO.CPP` enthalten. Besonders eindrucksvoll sind dabei die überladenen Operatoren. Was leisten nun diese Klassen im Detail?

* Die Klassen unterscheiden zwischen bitweiser und byteweiser Ansprache.

- * Ports, die als Inputs definiert werden, können nicht beschrieben werden und umgekehrt, können Outputports nicht gelesen werden, es fehlen einfach die entsprechenden Zugriffsfunktionen, lediglich die Klasse `..io` umfaßt alle Möglichkeiten.
- * Der letzte Wert in einem In- oder Outputport kann jederzeit, auch bei nicht rücklesbaren Ports, geholt werden (Funktion `val()`); dasselbe gilt für den Wert davor (Funktion `old()`).
- * Für Outputports können einfache Operatorenrechnungen angewendet werden, wie `+=, -=, *=, /=, %=, ++, --, &=, |=, -=, ^=, <<, >>`.
- * Bei Input-Ports kann auf eine Änderung gewartet werden
- * Die Gültigkeit des Adreßbereiches wird überprüft.
- * Bitein- und ausgänge werden ähnlich wie Byte-Ports behandelt.
- * Bildschirmausgabe der Portinhalte wird durch Überladen des Operators `<<` implementiert.
- * Ein definierter Port kann nur einmal verwendet werden, versehentliche Doppelbelegungen erzeugen einen runtime-Fehler, d.h. die Bearbeitung von Hardware über die IO-Klassen bindet eine 'Isolierschicht' zwischen Ports und Anwendung.

Damit diese speziellen Schutzmechanismen der IO-Klasse (Verhinderung von Doppelbenutzung von Ports) wirksam werden können, muß man alle Ein- und Ausgaben über die IO-Klasse abwickeln.

Was fehlt?

- * Abgeleitete Klassen für komplexere Portansprache, und
- * Abgeleitete Klassen (oder eine Erweiterung der Basisklasse `IO`) zu Ansprache mehrerer zusammengehöriger Bits innerhalb eines Ports.

```
#ifndef __IO_HPP
#define __IO_HPP

#include <mytypes.h>
#define P_IO
#include <portable.h>
#include <mylib.h>
#include <bit.h>

#define GETPORT() intoff();Old=IN_PORT(Adr)
// #define GETPORT() intoff();Old=Val;Val=IN_PORT(Adr)
#define SETPORT() OUT_PORT(Adr,Val);inton()

#define PC_MAXPORTS 0x400

#define TRENN_OUT ':'
#define TRENN_IO '='
#define TRENN_BIT '#'

#define DISP_IN "->"
#define DISP_OUT "<-"
#define DISP_IO "<>"

extern BOOL first_IO;

enum DIRECTION
{
    IN,
    OUT,
    IN_OUT
};

enum PORT
{
    BYTE,
    BIT
};

enum IO_ERR
{
    IO_NOERROR = 0,
    IO_ERR_ADDR,
    IO_ERRS_IOTYP,
    IO_ERRS_LEN,
    IO_ERRS_ADDR,
    IO_ERRS_BITN,
    IO_ERRS_DAT,
    IO_ERRBIT_NR,
    IO_ERR1,
    IO_ERR0
};

enum IO_DISP
{
    IO_DISP_NONE=0,
    IO_DISP_BIT=1,
    IO_DISP_BYTE=2,
    IO_DISP_BOTH=3
};

struct PORTUSAGE
{
    UCHAR in;
    UCHAR out;
};

class IO
{
public:
    UCHAR val()
    { return (Port==BYTE)?Val:BITTST(Val,Nr); }
    UCHAR old()
    { return (Port==BYTE)?Old:BITTST(Old,Nr); }
    UCHAR byte() { return Val; }
    UCHAR oldbyte() { return Old; }
    UCHAR bit() { return BITTST(Val,Nr); }
    INT adr() { return Adr; }
    DIRECTION io() { return Io; }
    CHAR nr() { return Nr; }
    PORT port() { return Port; }

    UCHAR Disp;

protected:
    IO(INT a, DIRECTION dir, BOOL check);
    IO(UCHAR n, INT a, DIRECTION dir, BOOL check);
    IO(PORT p, IO_DISP d)
    { Port=p; Disp = d; } // nur für IOBYTE
    ~IO();

    INT Adr;
    DIRECTION Io;
    UCHAR Val;
    UCHAR Old;
    UCHAR Nr;

private:
    PORT Port;
    static PORTUSAGE portusage[PC_MAXPORTS];
    BOOL Check;
};
```

```
ostream& operator << (ostream& o, IO& iob);

IO_ERR chk (CHAR *s,
            INT& adr, DIRECTION& io, UCHAR& val, BOOL disp);
IO_ERR chkb(CHAR *s, CHAR& nr,
            INT& adr, DIRECTION& io, UCHAR& val, BOOL disp);

class IOBYTE : public IO
{
public:
    IOBYTE() : IO(BYTE, IO_DISP_BYTE) { Io=IN_OUT; }
    VOID setaddr(INT a) { Adr=a; }
    VOID setdata(UCHAR d) { Val=d; }
};

class IOBYTE_IN : public IO
{
public:
    IOBYTE_IN(INT a,BOOL c=TRUE) : IO(a,IN,c)
    { Old=Val=IN_PORT(Adr); }

    UCHAR operator () ()
    { Old=Val; Val=IN_PORT(Adr); return Val; }
    UCHAR get_c();
};

class IOBYTE_OUT : public IO
{
public:
    IOBYTE_OUT(INT a, UCHAR v,BOOL c=TRUE) : IO(a,OUT,c)
    { OUT_PORT(Adr,v); Old=Val=v; }
    IOBYTE_OUT(INT a,BOOL c=TRUE) : IO(a,OUT,c) { }

    VOID operator = (UCHAR v)
    { Old=Val; Val=v; OUT_PORT(Adr,v); }

    UCHAR operator - ()
    { *this = -Val; return Val; }
    UCHAR operator ++ ()
    { *this = (Val+1); return Val; }
    UCHAR operator -- ()
    { *this = (Val-1); return Val; }
    VOID operator += (UCHAR v) { *this = Val+v; }
    VOID operator -= (UCHAR v) { *this = Val-v; }
    VOID operator *= (UCHAR v) { *this = Val*v; }
    VOID operator /= (UCHAR v) { *this = Val/v; }
    VOID operator %= (UCHAR v) { *this = Val%v; }
    VOID operator <<= (UCHAR v) { *this = Val<<v; }
    VOID operator >>= (UCHAR v) { *this = Val>>v; }
    VOID operator ^= (UCHAR v) { *this = Val^v; }
    VOID operator &= (UCHAR v) { *this = Val&v; }
    VOID operator |= (UCHAR v) { *this = Val|v; }
};

class IOBYTE_IO : public IO
{
private:
    BOOL hide;
    VOID inton() { if (hide) enable(); }
    VOID intoff() { if (hide) disable(); }
public:
    IOBYTE_IO(INT a, UCHAR v, BOOL h=TRUE,BOOL c=TRUE)
    : IO(a,IN_OUT,c)
    { Old=Val=v; hide=h; }
    IOBYTE_IO(INT a, BOOL h=TRUE, BOOL c=TRUE)
    : IO(a,IN_OUT,c)
    { hide=h; }

    UCHAR operator () ()
    { Old=Val; Val=IN_PORT(Adr); return Val; }
    VOID operator = (UCHAR v)
    { GETPORT(); Val=v; SETPORT(); }

    UCHAR operator ++ ()
    { GETPORT(); Val=Old+1; SETPORT(); return Val; }
    UCHAR operator -- ()
    { GETPORT(); Val=Old-1; SETPORT(); return Val; }
    UCHAR operator - ()
    { GETPORT(); Val=-Old; SETPORT(); return Val; }
    VOID operator += (UCHAR v)
    { GETPORT(); Val=Old+v; SETPORT(); }
    VOID operator -= (UCHAR v)
    { GETPORT(); Val=Old-v; SETPORT(); }
    VOID operator *= (UCHAR v)
    { GETPORT(); Val=Old*v; SETPORT(); }
    VOID operator /= (UCHAR v)
    { GETPORT(); Val=Old/v; SETPORT(); }
    VOID operator %= (UCHAR v)
    { GETPORT(); Val=Old%v; SETPORT(); }
    VOID operator <<= (UCHAR v)
    { GETPORT(); Val=Old<<v; SETPORT(); }
    VOID operator >>= (UCHAR v)
    { GETPORT(); Val=Old>>v; SETPORT(); }
    VOID operator ^= (UCHAR v)
    { GETPORT(); Val=Old^v; SETPORT(); }
    VOID operator &= (UCHAR v)
    { GETPORT(); Val=Old&v; SETPORT(); }
    VOID operator |= (UCHAR v)
    { GETPORT(); Val=Old|v; SETPORT(); }

    IO_ERR tst(ostream& o, BOOL t);
};
```

```

class IOBIT_IN : public IO
{
public:
    IOBIT_IN(CHAR n, INT a, BOOL c=TRUE) : IO(n,a,IN,c)
    { Old=Val=IN_PORT(Adr); }

    UCHAR operator () ()
    { Old=Val; Val=IN_PORT(Adr);
      return BITTST(Val,Nr);
    }
    UCHAR get_c();
};

class IOBIT_OUT : public IO
{
public:
    IOBIT_OUT(CHAR n,INT a,UCHAR v,BOOL c=TRUE)
    : IO(n,a,OUT,c)
    { Val=Old=BITMAK(v,n); OUT_PORT(Adr,Val); }
    IOBIT_OUT(CHAR n,INT a,BOOL c=TRUE)
    : IO(n,a,OUT,c)
    { }

    VOID operator = (UCHAR v)
    { Old=Val; Val=BITINS(Val,Nr,v);
      OUT_PORT(Adr,Val);
    }
    VOID set(UCHAR v)
    { Old=Val; Val=v; OUT_PORT(Adr,v); }

    UCHAR operator - ()
    { Old=Val; Val = BITINV(Val,Nr);
      OUT_PORT(Adr,Val); return BITTST(Val,Nr); }
    UCHAR operator ++ ()
    { return (*this).operator - (); }
    UCHAR operator -- ()
    { return (*this).operator - (); }
    VOID operator ^= (UCHAR v)
    { Old=Val; Val = BITXOR(Val,Nr,v);
      OUT_PORT(Adr,Val);
    }
    VOID operator &= (UCHAR v)
    { Old=Val; Val = BITAND(Val,Nr,v);
      OUT_PORT(Adr,Val);
    }
    VOID operator |= (UCHAR v)
    { Old=Val; Val = BITOR(Val,Nr,v);
      OUT_PORT(Adr,Val);
    }
};

class IOBIT_IO : public IO
{
private:
    BOOL hide;
    VOID inton() { if (hide) enable(); }
    VOID intoff() { if (hide) disable(); }
public:
    IOBIT_IO(CHAR n, INT a, UCHAR v,
             BOOL h=TRUE,BOOL c=TRUE)
    : IO(n,a,IN_OUT,c)
    { hide=h; Val=IN_PORT(Adr); Val=BITMAK(Val,n);
      Old=Val; OUT_PORT(Adr,Val); }
    IOBIT_IO(CHAR n, INT a,
             BOOL h=TRUE,BOOL c=TRUE)
    : IO(n,a,IN_OUT,c)
    { hide=h; Old=Val=IN_PORT(Adr); }

    VOID operator = (UCHAR v)
    { GETPORT(); Val = BITINS(Old,Nr,v); SETPORT(); }
    UCHAR operator () ()
    { Old=Val; Val=IN_PORT(Adr);
      return BITTST(Val,Nr);
    }
    UCHAR operator - ()
    { GETPORT(); Val = BITINV(Old,Nr); SETPORT();
      return BITTST(Val,Nr);
    }
    UCHAR operator ++ ()
    { return (*this).operator - (); }
    UCHAR operator -- ()
    { return (*this).operator - (); }
    VOID operator ^= (UCHAR v)
    { GETPORT(); Val=BITXOR(Old,Nr,v); SETPORT(); }
    VOID operator &= (UCHAR v)
    { GETPORT(); Val = BITAND(Old,Nr,v); SETPORT(); }
    VOID operator |= (UCHAR v)
    { GETPORT(); Val = BITOR(Old,Nr,v); SETPORT(); }
    IO_ERR tst(ostream& o, BOOL t);
    INT looperror();
};

```

```

class IOBIT_LOOP
{
public:
    IOBIT_LOOP(INT txa, CHAR txb, INT rxa, CHAR rxb,
              BOOL pol)
    { tx = new IOBIT_OUT(txb,txa);
      rx = new IOBIT_IN(rxb,rxr);
      polarity=pol; make = TRUE; }
    IOBIT_LOOP(IOBIT_OUT *out, IOBIT_IN *in, BOOL pol)
    { tx = out; rx = in; polarity=pol; make = FALSE; }
    IOBIT_LOOP(IOBIT_OUT& out, IOBIT_IN& in, BOOL pol)
    { tx = &out; rx = &in; polarity=pol; make = FALSE; }
    ~IOBIT_LOOP()
    { if (!make) return; delete tx; delete rx; }

    VOID operator = (UCHAR v) { tx->operator = (v); }
    UCHAR operator()() { return rx->operator()(); }

    INT tst(ostream& o);

private:
    IOBIT_OUT *tx;
    IOBIT_IN *rx;
    BOOL polarity;
    BOOL make;
};

#endif // IO_HPP

```

Ein IO-Objekt wird durch die geschützten Variablen `val` (der zuletzt gelesene oder der zuletzt geschriebene Wert), `old` (der davor gelesene Wert), `adr` (Adresse), `io` (Eingangs-, Ausgangs- oder bidirektionaler Port), `nr` (Bitnummer, nur bei Bit-Ports), `port` (Bit- oder Byte-Port) und `check` beschrieben. Die Variablen haben jeweils große Anfangsbuchstaben, während die sie zurücklesenden, öffentlichen Funktionen `val()`, `adr()`, `io()`, `nr()` und `port()` kleine Anfangsbuchstaben haben. Die Variablen sind gegen Zugriff von Programmen außerhalb der Klasse geschützt, abgeleitete Klassen können aber mit ihnen arbeiten. `check` ist für die Überprüfung von Doppelbenutzung von Ports vorgesehen, der Destruktor benötigt diese Variable, um die Belegung wieder frei geben zu können.

Ein IO-Objekt merkt sich in der Variablen `val` den zuletzt gelesenen oder den zuletzt geschriebenen Wert. Die Funktion `val()` liefert den zuletzt gelesenen Wert. Die Funktion `bit()` leistet dasselbe für ein Bit-Objekt.

Allgemeine Vereinbarungen

`IN` und `OUT` sind Makros, mit denen bei den rücklesbaren Ports der Port zuerst gelesen und dann gesetzt wird. Dieser Vorgang kann wahlweise mit ein- oder ausgeschalteten Interrupts erfolgen. Je nachdem, wie der alte Wert `old` gebildet wird, kann auch eine andere Lesereihenfolge angewendet werden (durch // ausgeschaltet). Das Schalten der Interrupts erfolgt nicht unmittelbar über `enable()`, `disable()`, sondern über inline-Funktionen `inton()` und `intoff()` und eine bei der Konstruktion übergebene Variable `hide` (Anfangswert: `TRUE`) bestimmt, ob die Interrupts geschaltet werden oder nicht.

`PC_MAXPORTS` legt die größte erlaubte IO-Adresse fest. Alle größeren Adressen werden auf diesen Maximalwert abgeschnitten.

Die Zeichen `TRENN_OUT`, `TRENN_IO` und `TRENN_BIT` dienen zur Trennung der Eingabedaten, wenn ein Port oder ein Portbit als Zeichenfolge eingegeben wird (siehe Beispiel `HC04IO2.CPP`). Die Strings `DISP_IN`, `DISP_OUT` und `DISP_IO` trennen Adreß- und Datenbestandteil bei der Ausgabe und geben gleichzeitig die Übertragungsrichtung an.

Die Variable `first_io` ist beim Programmstart `TRUE` und bewirkt das Initialisieren der statischen Variablen `portusage[]`.

Jeder Bit- oder Byteport kann als Eingang, Ausgang oder bidirektionaler Port betrieben werden. Der Aufzählungstyp `DIRECTION` dient zur Beschreibung dieser Porteigenschaft.

Ein IO-Objekt kann ein `BIT`- oder `BYTE`-Objekt sein. Die Basisklasse ist in beiden Fällen die Klasse `IO`. Die Unterscheidung der Portart bei der Konstruktion eines Objekts übernimmt der Aufzählungstyp `PORT`.

Bei der Konstruktion eines IO-Objekts mit einem String werden viele Fehler durch den Fehlertyp `IO_ERR` beschrieben.

`IO_DISP` legt fest, ob eine Anzeige erfolgen soll, und ob die Anzeige des Portwertes oder auch zusätzlich die Anzeige des Bits erfolgen soll.

Der Typ `PORTUSAGE` legt fest, ob ein Port (oder ein Bit) bereits benutzt ist (`in` oder `out` sind 1) und ob es sich um einen Eingang (`in=1, out=1`) oder Ausgang (`in=0, out=1`) oder bidirektionalen Port (beide 1) handelt. Wird im Rahmen eines Programms ein Port-Objekt mit derselben Adresse an mehreren Stellen konstruiert, wird das Programm abgebrochen. Diese Überprüfung ist während der Konstruktion eines IO-Objekts abschaltbar. Der Anfangswert ist mit Überprüfung.

Basisklasse IO

Die Klasse `IO` beschreibt entweder einen Bit- oder einen Byte-Port und unterscheidet diese via Konstruktor und durch die private Variable `port`. Die geschützten aber durch abgeleitete Klassen erreichbaren Variablen `adr`, `io`, `val` `old`, `nr` werden durch die gleichnamigen Zugriffsfunktionen, die aber mit Kleinbuchstaben beginnen, gelesen: `adr()`, `io()`, `val()`, `old()`, `nr()`.

Die Klasse `IO` dient als Basisklasse für alle IO-Klassen. Es gibt drei, allerdings geschützte Konstruktoren, daher kann von dieser Klasse kein Objekt gebildet werden.

```
IO(INT a, DIRECTION dir, BOOL check=TRUE);
```

Konstruiert einen Byte-Port mit Adresse `a`, Richtung `dir` und ausschaltbarer Adreßüberprüfung `check`.

```
IO(UCHAR n, INT a, DIRECTION dir, BOOL check=TRUE);
```

Konstruiert einen Bit-Port mit Adresse `a`, Bitnummer `n`, Richtung `dir` und ausschaltbarer Adreßüberprüfung `check`.

```
IO(PORT p, IO_DISP d) { Port=p; Disp = d; }
```

Konstruiert eine IO-Objekt ohne Adreßangabe. (Wird nur von der Klasse `IOBYTE` benötigt). Der Destruktor `-IO` sorgt dafür, daß die Adreßüberprüfung für diesen Port ausgeschaltet wird.

```
UCHAR Disp;
```

Die öffentliche Variable `Disp` regelt die Ausgabe über den überladenen Operator `<<`. Da ihr Wert nicht besonders kritisch ist, bleibt sie öffentlich, Zugriffsfunktionen entfallen, sie kann die Werte des Aufzählungstyps `IO_DISP` (weiter oben) annehmen.

```
static PORTUSAGE portusage[PC_MAXPORTS];
```

`portusage[]` ist als statische Variable der Klasse `IO` auch allen Objekten des IO-Typs gemeinsam. Jedes generierte Objekt setzt ein seiner Adresse entsprechendes Element und auch die Schreib/Leserichtung. Bei einer versehentlichen Doppelbenutzung einer IO-Adresse oder eines bestimmten Bits in verschiedenen Programmteilen wird das Programm mit einer entsprechenden Fehlermeldung abgebrochen. Diese Eigenschaft ist über den Konstruktor abschaltbar.

```
ostream& operator << (ostream& o, IO& iob);
```

Der Operator `<<` ist für Objekte der Klasse `IO` überladen und gibt den Portwert im Format: `n:adr##w` aus, wobei `n` die Bitnummer (entfällt bei Byte-Ports), `adr` die Adresse, `##` ein Symbol, das die Richtung des Ports angibt (`<`, `>`, `<->`) und `w` der Zahlenwert ist. Alle Angaben erfolgen hexadezimal.

```
IO_ERR chk(CHAR *s,
            INT& adr, DIRECTION& io, UCHAR& val, BOOL disp);
IO_ERR chkb(CHAR *s, CHAR& nr,
            INT& adr, DIRECTION& io, UCHAR& val, BOOL disp);
```

Ein IO-Objekt kann auch durch einen String konstruiert werden, etwa durch eine Benutzereingabe (siehe Beispiel `HC04IO2.CPP`). Die Funktionen `chk()` und `chkb()` überprüfen das Eingabeformat und geben eventuelle Formatfehler bekannt.

Die Klasse `IOBYTE` ist eine behelfsmäßige Klasse, die es erlaubt, ein IO-Objekt zu bilden, dessen Adresse und Daten veränderbar sind. Diese Klasse wird nur benötigt, um die Ausgabefunktion mit dem überladenen Operator `<<` testen zu können ohne wirklich einen IO-Port ansprechen zu müssen. Außerdem wird diese Klasse als Hilfsklasse bei der Klasse `CMOSRAM` benutzt.

Die Klasse `IOBYTE_IN` enthält neben dem Konstruktor für eine Adresse `a` die Funktionen `operator()()` und `get_c()` mit der aus dem Port gelesen wird. Der Funktionsoperator liest aus dem Port und die Funktion `get_c()` wartet auf eine Änderung des Portwertes. Sie werden wie folgt angewendet:

```
IOBYTE_IN user(0x300);
...
UCHAR last = user();
```

oder

```
if (user()==5) ...
```

Die Klasse `IOBYTE_OUT` enthält neben zwei Konstruktoren (mit und ohne Initialisierung des Ports) für eine Adresse `a` die Operatorfunktionen für `=`, `++`, `--`, `-`, `<<`, `>>`, `+=`, `-=`, `*=`, `/=`, `%=`, `&=`, `|=` und `^=`, die es ermöglichen, mit dem IO-Objekt genauso, wie mit einer Variablen des Typs `UCHAR` zu arbeiten. Sie werden wie folgt angewendet:

```
IOBYTE_OUT out(0x300,0);
```

```
out=1;
++out; --out; -out; ...
```

oder

```
out+=5; out&=0xf0;
```

Der Operator `-` wirkt auf den Port, obwohl er in dieser Form auf eine Variable nicht anwendbar wäre (man müßte schreiben `out--out`). Die Klasse `IOBYTE_IO` vereinigt die Möglichkeiten von

`IOBYTE_IN` und `IOBYTE_OUT` für einen rücklesbaren Port, wie es etwa der Datenport der parallelen Schnittstelle darstellt und ergänzt diese Möglichkeiten durch die Testfunktion `test()`, die die korrekte Funktion des Ports durch alternierendes Anlegen verschiedener Bitmuster testet. Die Portansprache wurde hier noch etwas erweitert: Wenn mehrere Programmteile einen Port bedienen, kann zwischen dem zuletzt gelesenen Wert und dem aktuellen Wert ein Unterschied bestehen, da inzwischen ein anderes interruptgesteuertes Programm diesen verändert haben könnte. Um diesen Effekt zu verhindern, wurde in die Klassen `IOBYTE_IO` und `IOBIT_IO` die Möglichkeit eingebaut, die Interrupts vor Zugriff aus- und nachher wieder einzuschalten.

Die Klassen `IOBIT_IN`, `IOBIT_OUT` und `IOBIT_IO` entsprechen den `BYTE`-Klassen arbeiten aber jeweils nur mit einem Bit eines Ports und lassen alle anderen Bits unverändert. Die Anzahl der überladenen Operatoren ist entsprechend geringer, `++`, `--` und `-` sind identisch. Eine zusätzliche Schwierigkeit bei den `BIT`-Klassen ist, daß man zur Veränderung eines Bits jedenfalls den ganzen Port lesen und schreiben muß, die nicht beeinflussten Bits aber unverändert läßt.

Die Klasse `IOBIT_LOOP` arbeitet mit zwei `IOBIT`-Objekten `tx` und `rx`, die einer Drahtverbindung zwischen zwei beliebigen Portbits entsprechen. Diese Klasse wird beim Testen der seriellen und parallelen Schnittstelle verwendet.

'Bitprozessor' BIT.H, HC04BI1.CPP

Bei vielen IO-Ports hat jedes einzelne Bit eine andere Bedeutung. Es besteht dabei der Bedarf diese Bits unabhängig voneinander bearbeiten zu können. Im allgemeinen werden dafür arithmetische oder logische Operationen verwendet. Einfache Makros in `BIT.H` vereinheitlichen die Bearbeitung einzelner Bits und reduzieren die Fehleranfälligkeit individueller Erfindungen. Um ein Bit in einem Byte zu Setzen, zu Löschen oder zu Invertieren wendet man eine ODER, UND oder EXKLUSIV-Oder Verknüpfung mit einer geeigneten Maske an, die das eine Bit beeinflusst, die anderen aber unverändert läßt. Die Maske kann man als Konstante, als dimensionierte Variable oder als das Ergebnis einer Schiebeoperation gewinnen. Es wurde die Schiebeoperation gewählt, da in diesem Fall die Makros gleichermaßen auf 8- oder 16- oder 32-bit-breite Variablen anwendbar sind.

```
#ifndef BIT_H
#define BIT_H

#define SETMSK(n) (1<<(n))
#define CLRMSK(n) (~(1<<(n)))

#define BITSET(v,n) ((v)|SETMSK(n))
#define BITCLR(v,n) ((v)&CLRMSK(n))
#define BITINV(v,n) ((v)^SETMSK(n))
#define BITTST(v,n) (((v)&SETMSK(n))!=0)
#define BITMAK(b,n) (((b)==0)?0:SETMSK(n))
#define BITINS(v,n,b)
  (((b)==0)?BITCLR((v),(n)):BITSET((v),(n)))
#define BITOR(v,n,b) (((b)==0)?(v):BITSET((v),(n)))
#define BITAND(v,n,b) (((b)==1)?(v):BITCLR((v),(n)))
#define BITXOR(v,n,b) (((b)==0)?(v):BITINV((v),(n)))

#endif
```

Die MAKROS bestehen aus zwei Gruppen: zunächst jene, die in einer Variablen `v` das Bit `n` beeinflussen `BITSET`, `BITCLR` und `BITINV` sowie `BITTST`, das prüft, ob ein Bit gesetzt ist; dann die zweite Gruppe, die mit einer Variablen `b` arbeiten: `BITMAK`, generiert in Abhängigkeit von `b` an der Stelle `n` ein Bit, `BITSET`, setzt in `v` an der Stelle `n` ein Bit. Die Makros `BITOR`, `BITAND` und `BITXOR` wenden die entsprechenden booleschen Operation auf ein Bit an. Die Anwendung dieser Makros zeigt am besten das Testprogramm:

```
/* HC04BI1.CPP */
/*
 * TESTEN des 'BITPROZESSORS'
 * =====
 */
#include <portable.h>
#include <mytypes.h>
#include <bit.h>

#define F hex << setw(4)

VOID main(VOID)
{
  cout << "Testing unary operations" << endl;
  for (INT i=0; i<8; i++)
  {
    UCHAR v;
    cout << "BIT" << i << endl;
    for (INT val=0; val<2; val++)
    {
      if (val==0) v=0; else v=0xff;
      cout <<F<< (INT)v << ":SET:" << i << ":" << endl;
      cout <<F<< (INT)BITSET(v,i) << i << ":" << endl;
      cout <<F<< (INT)BITCLR(v,i) << i << ":" << endl;
      cout <<F<< (INT)BITINV(v,i) << i << ":" << endl;
      cout <<F<< (INT)BITMAK(v,i) << i << ":" << endl;
      cout <<F<< (INT)BITTST(v,i) << i << ":" << endl;
      cout << endl;
    }
  }
  cin.get();

  cout << "Testing binary operations" << endl;
  for (i=0; i<8; i++)
  {
    UCHAR v;
    cout << "BIT" << i << endl;
    for (INT val=0; val<2; val++)
    {
      if (val==0) v=0; else v=0xff;
      for (INT j=0; j<2; j++)
      {
        cout <<F<< (INT)v << ":&:" << j << ":" << endl;
        cout <<F<< (INT)BITAND(v,i,j) << i << ":" << endl;
        cout <<F<< (INT)BITOR(v,i,j) << i << ":" << endl;
        cout <<F<< (INT)v << ":^:" << j << ":" << endl;
        cout <<F<< (INT)BITXOR(v,i,j) << i << ":" << endl;
      }
      cout << endl;
    }
  }
  cin.get();
}
```

Demonstration der BYTE-Klassen HC04IO0.CPP, HC04TA1G.CPP

Das folgende Programm demonstriert die Verwendung der rücklesbaren IOBYTE-Klasse am Beispiel der parallelen Schnittstelle. Zum Vergleich wurde als Kommentar die Portansprache in Standard-C angeschrieben.

```

/* HC04IO0.CPP */
/*
 * Beispiel für die Grundfunktionen der IO-Klasse
 * =====
 */
#include <io.hpp>
#include <iodef.h>
#ifdef MYLIB
#include "\mylib\source\io.cpp"
#endif

VOID main(VOID)
{
    // IO-Port zum Lesen und Schreiben definieren
    IOBYTE_IO lpt(PC_LPT1);

    lpt = 0xf0;          // outport(0x3bc,0xf0);
    lpt();              // UCHAR a=inport(0x3f8);
    cout << lpt << endl; // cout << (int)a << endl;

    lpt ^= 0x55;        // a=inport(0x3bc); a^=0x55;
    lpt();              // outport(0x3bc,a);
    cout << lpt << endl; // cout << (int)a << endl;
}
    
```

Beim Lesen eines Ports kann entweder eine Zuweisung erfolgen: `a=lpt()` oder, wie im Beispiel, der Funktionsaufruf allein, dann bleibt der Wert in der privaten Variablen `val` erhalten und kann entweder durch `lpt.val()` oder, wie im Beispiel durch die Ausgabe mit dem überladenen Operator `<<` angewendet werden.

Keihen wir jetzt zu unserem Tastaturprogramm zurück:

```

/* HC04TA1G.C */
/*
 * Tastatur-kodes prüfen, mit Klasse IO
 * =====
 */
#include <portable.h> // cout
#include <keyscan.h> // SCAN_ESCAPE
#include <mylib.h> // clear_kbdqueue()
#include <iodef.h> // PC_KBD_DATA
#include <io.hpp>
#ifdef MYLIB
#include "\mylib\source\io.cpp" // IOBYTE_IN
#include "\mylib\source\kbd.c" // clear_kbdqueue()
#endif

VOID main(VOID)
{
    IOBYTE_IN key(PC_KBD_DATA);

    constream con;
    con.clrscr();
    cout << "Tastatur-SCAN-Codes, Ende mit ESC" << endl;
    do
    {
        if (key.get_c() != SCAN_ESCAPE)
            con << setfill('0') << setw(2) << hex
                << (UINT)key.val() << ' ' << endl;
        clear_kbdqueue();
    }
    while (key.val() != SCAN_ESCAPE);
    con.clrscr();
}
    
```

Neue Elemente:

IOBYTE_IN key(PC_KBD)	Objekt key ist ein Leseport
key()	Lesen des SCAN-Kode
key.get_c()	--, Änderung abwarten
key.val()	liefert zuletzt gelesenen Wert ohne den Port erneut anzusprechen

Obwohl rein äußerlich kein wesentlicher Unterschied zu der vorigen Version `HC04TA1F.CPP` besteht, beachten Sie, daß die Klassen `IOBYTE_IN` (oder `_OUT` oder `_IO`) auf jeden beliebigen Port anwendbar sind; auch der Versuch, durch Parameterübergabe eine gleichwertige Funktionalität zu erzielen wäre nicht genug, die Leistungen der Klasse in C auszudrücken!

Testen der BYTE-Klassen HC04IO1.CPP

Jetzt zu einem Testprogramm, das versucht möglichst alle Operatoren der IOBYTE-Klassen zu benutzen. Vergleichen Sie die Leistung einzelner Aktionen mit der Kürze der Darstellung, etwa gibt `cout << kbd` das Format `xxx->hh` aus oder `++par1` liest den aktuellen Portwert ein, erhöht um eins und gibt ihn wieder aus. (Diese Operation könnte auf `kbd` nicht angewendet werden, da `par1` ein IO-Port und `kbd` nur ein IN-Port ist.)

Der Test besteht aus zwei Teilen: zuerst wird die parallele Schnittstelle aus unabhängigen Schreib- und Leseports `po` und `pi` zusammengesetzt, dann wird ein rücklesbares Objekt `pio` gebildet. Die Funktion `out()` vereinheitlicht die Ausgabe, die Makros `OUT` und `OUTIO` übergeben jeweils den richtigen Zahlenwert.

Bei diesem Testprogramm geht es darum, möglichst viele Eigenschaften der Elementfunktionen zu testen. Dazu wird ein rücklesbarer Port (parallele Schnittstelle) verwendet. Der Test besteht aus zwei Teilen, der erste zum Testen von `IOBYTE_IN` und `IOBYTE_OUT`, der zweite zum Testen von `IOBYTE_IO`. Da beide Male derselbe Port (`0x3bc`) verwendet wird, gibt es innerhalb der Klassengemeinschaft `IO*` wegen der gemeinsamen Variablen `portusage[]` eine Doppelbelegung und ohne weitere Maßnahme würde das Programm in der Mitte mit einer Fehlermeldung abbrechen. Es gibt zwei Möglichkeiten das zu verhindern: (a) die beiden Programmteile werden als getrennte Blöcke behandelt, wie im Beispiel, denn dann wird automatisch am Ende der ersten Blockes der Destruktor von `pi` und `po` gerufen, der die Belegung in `portusage[]` zurücknimmt, oder (b) der Konstruktor von `pio` wird mit zusätzlichen Parametern gerufen `pio(PC_LPT1|LPT_DATA,TRUE,FALSE)`, wobei `TRUE` die Interruptabschaltung aktiviert und `FALSE` die Überprüfung der Doppelbelegungen ausschaltet. Hier wurde (a) angewendet, da es gar nicht notwendig ist, daß alle drei Objekte `pi`, `po` und `pio` gleichzeitig aktiv sind, `pi` und `po` haben ihre Schuldigkeit getan und werden nicht mehr benötigt.

```

/* HC04IO1.CPP */
/*
 * Testen der Klassen IOBYTE_IN, IOBYTE_OUT, IOBYTE_IO
 * =====
 */
#include <portable.h>
#include <iodef.h>
#include <keyscan.h>
#include <mylib.h>
#include <io.hpp>
#ifdef MYLIB
#include "\mylib\source\io.cpp"
#endif

#define F hex<<setw(2)<<setfill('0')
#define OUT(o,a,n) out(o,a.old(),n())
#define OUTIO(o,io) out(o,io.old(),io.val())

VOID out(CHAR* operation, UCHAR alt, UCHAR neu)
{
    cout << operation <<F<< ' ' << endl;
    cout << "alt:" <<F<< (INT)alt << ' ' << endl;
    cout << "neu:" <<F<< (INT)neu << ' ' << endl;
}

INT main(VOID)
{
    cout << endl
        << "IOBYTE_OUT/IOBYTE_IN: Operatoren-Test,\n";
    {
        IOBYTE_IN pi(PC_LPT1|LPT_DATA);
        IOBYTE_OUT po(PC_LPT1|LPT_DATA);
        po=0x13; OUT("po=0x13:",po,pi);
        ++po; OUT("++po:",po,pi);
        --po; OUT("--po:",po,pi);
        po += 5; OUT("po+=5:",po,pi);
        po -= 5; OUT("po-=5:",po,pi);
        po *= 5; OUT("po*=5:",po,pi);
        po /= 5; OUT("po/=5:",po,pi);
        po = 13; OUT("po=13:",po,pi);
        po ^= 5; OUT("po^=5:",po,pi);
        ~po; OUT("~po:",po,pi);
        po &= 0xf0; OUT("po&=0xf0:",po,pi);
        po |= 0xf0; OUT("po|=0xf0:",po,pi);
        po ^= 0xff; OUT("po^=0xff:",po,pi);
        po <<= 2; OUT("po<=2:",po,pi);
        po >>= 2; OUT("po>=2:",po,pi);
    }
    cout << endl
        << "IOBYTE_IO: Operatoren-Test, \n";
    cin.get();
    {
        IOBYTE_IO pio(PC_LPT1|LPT_DATA);
        // aus Platzgründen fehlt hier ein Block, ähnlich,
        // wie oben mit den Klassen in und out
    }
    cout << "Erfolgreich beendet" << endl;
    return EXIT_SUCCESS;
}
    
```


Jetzt ein paar nützliche Anwendungen:

Anzeige und Änderung des Zustandes von IO-Ports HC04IO2.CPP

Zu Testzwecken ist es wünschenswert, rasch ein Programm zur Hand zu haben, mit dem man eine oder mehrere Portadressen zyklisch ausliest und andere einstellt. Das folgende Programm erlaubt die Ausgabe und Einstellung des Inhalts von Portadressen durch Angabe der Adresse in der Kommandozeile.

Programmaufruf:

```
HC04IO2           Zeigt einen Hilfetext an
HC04IO2 adr...   Anzeige des Inhalts einer IO-Adresse
HC04IO2 adr:hh... Setzen einer IO-Adresse
HC04IO2 adr=hh... Setzen und Anzeigen einer IO-Adresse
```

Es können beliebig viele Adressen angezeigt werden.

Das Pointerarray port[] enthält Zeiger auf Objekte des Typs IOBYTE_IO. Die Funktion chk() prüft für jede Eingabe in der Kommandozeile das Eingabeformat und bricht im Fehlerfall mit einer Meldung das Programm ab. Ohne Fehler wird weiters geprüft, ob der Port bereits im Portarray enthalten ist, man kann also nicht mehrmals dieselbe Adresse angeben.

Die eigentliche Anzeigeroutine läuft, bis eine Taste gedrückt wird. Da der Bildschirmaufbau die meiste Zeit in Anspruch nimmt, wird der Portwert nur bei Änderungen auf den Bildschirm geschrieben.

Bei diesem Programm zeigt sich auch eine Schwäche der IO-Klassen: Es kann kein gemeinsamer Pointer gebildet werden, der sowohl Funktionen eines _IN-Objekts, eines _OUT-Objekts und eines _IO-Objekts aufruft, d.h. der Pointer kann schon gebildet werden aber die Funktionen sind nicht aufrufbar, da sie die Basisklasse nicht enthält. Auch eine rein virtuelle Basisklasse mit virtuellen Elementfunktionen ist keine Lösung, da es abgeleitete Klassen gibt, die nur einen Teil der Elementfunktionen enthalten. Kompliziertere Konstruktionen sind aber aus Geschwindigkeitsgründen nicht ratsam, sodaß einfach die universelle IOBYTE_IO-Klasse verwendet wird.

```
/* HC04IO2.CPP */
/*
 * Anzeige und Änderung des Zustandes von IO-Ports
 * =====
 */
#include <string.h>
#include <stdlib.h>
#include <mylib.h>
#include <portable.h>
#include <iodef.h>
#include <keyscan.h>
#include <io.hpp>
#ifdef MYLIB
#include "mylib\source\io.cpp"
#endif

const INT MAXPORT=255;
```

```
INT main(INT argc, CHAR * argv[])
{
    IOBYTE_IO *port[MAXPORT];

    if (argc==1) // Hilfe-Anzeigen
    {
        cout << "\nAnzeige und Änderung von IO-Ports\n";
        cout << "=====\n";
        cout << "Aufruf: " << FILE
            << "{adr|adr" << TRENN_OUT << "hh|adr"
            << TRENN_IO << "hh" << endl;
        cout << "adr ... drei HEX-Stellen \n";
        cout << "hh ... zwei HEX-Stellen \n";
        cout << "adr ... IN: zeigt Portwert adr an ";
        cout << DISP_IN << endl;
        cout << "adr" << TRENN_OUT
            << "hh ... OUT: ändert Portwert adr auf hh "
            << DISP_OUT << endl;
        cout << "adr" << TRENN_IO
            << "hh ... IN OUT: ändert Portwert adr auf hh "
            << "und zeigt Portwert an "
            << DISP_IO << endl;
        return EXIT_FAILURE;
    }
    for (INT i=1; i<argc; i++)
    {
        INT adr_t;
        DIRECTION io_t;
        UCHAR val_t;

        if (chk(argv[i],adr_t,io_t,val_t,TRUE)==IO_NOERROR)
        {
            port[i-1] = new IOBYTE_IO(adr_t,val_t);
            port[i]=NULL;
            continue;
        }
        else
            exit(EXIT_FAILURE);
    }

    // testing multiple assignments
    // =====

    BOOL exist[PC_MAXPORTS][2];

    for (INT i=0; i<PC_MAXPORTS; i++)
        for (INT j=0; j<2; j++)
            exist[i][j]=FALSE;

    IOBYTE_IO **iop=port;
    while (*iop)
    {
        INT adr=(*iop)->adr();
        DIRECTION io=(*iop)->io();
        if (exist[adr][io])
        {
            cout << "\nDoppelbelegung bei: "
                << hex << (*iop)->adr();
            return EXIT_FAILURE;
        }
        else
            exist[(*iop)->adr()][(*iop)->io()]=TRUE;
        iop++;
    }

    constream con;
    con.clrscr();
    BOOL start=TRUE;

    for (;;)
    {
        INT displayposition=0;
        IOBYTE_IO **iop=port;

        while (*iop)
        {
            if (kbhit())
            {
                UINT c = getch();
                if (c==0x1b) return EXIT_SUCCESS;
            }

            UCHAR old_value = (**iop).old();

            // Bildschirm nur bei Änderungen
            // und zu Beginn auffrischen
            if ((old_value!=(**iop)()) || (start))
            {
                INT char_position, row, column;
                char_position=displayposition*8;
                column = char_position % 80;
                row = char_position / 80;
                con << setxy(column+1,row+1) << (**iop);
            }
            iop++;
            displayposition++;
        }
        start = FALSE;
    }
}
```

Testen der parallelen Schnittstelle HC04PA1.CPP

Ein weiteres Programm, das die Bitklasse benutzt, testet den parallelen Port mit einem Prüfstecker. Die Belegung entnehme man dem Hilfetext am Programmanfang, den man erhält, wenn man das Programm mit einem beliebigen Parameter aufruft. Der Hilfetext erklärt die Steckerbelegung, die Polarität der Ein- und Ausgabe, die Position des Interruptpins und die Belegung des Prüfsteckers.

Parallele Schnittstelle

Parallele Schnittstelle								
PORT+0								
Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
Pin	9	8	7	6	5	4	3	2
Logik	+	+	+	+	+	+	+	+
E/A	IO	IO	IO	IO	IO	IO	IO	IO
PORT+1								
Bit	7	6	5	4	3	2	1	0
Name	BUSY	ACK	PEND	SELE	ERROR			
Pin	11	10	12	13	15			
Logik	-	+	+	+	-			
E/A	I	I	(IRQ)I	I	I			
PORT+2								
Bit	7	6	5	4	3	2	1	0
Name				IRQ	SI	INIT	AF	STROB
Pin					17	16	14	1
Logik E				+	+	+	-	-
Logik A				+	-	+	-	-
E/A				RW	IO	IO	IO	IO

Masse: Pins 18-24

if PORT+2,bit 4 == 1 --> INT=ON,
Hi-to-Lo-Transition on Pin 10 causes IRQ

Parallel Testadapter connects
pins 1-13,2-15,10-16,11-17,12-14

Zuerst wird für alle drei erlaubten Portadressen überprüft, ob ein Port existiert, das erfolgt auf Grund der Rücklesbarkeit des Datenports durch `lpt.tst()` (Testfunktion aus `IOBYTE_IO`).

Vorhandene Ports werden auf Grund der Verbindungen des Prüfsteckers mit `IOBIT_LOOP`-Objekten getestet. Jedes `IOBIT_LOOP`-Objekt 11,12,13 und 14 entspricht einer Drahtbrücke am Teststecker. Als Argument wird jeweils die Adresse und die Bitnummer angegeben. Der letzte Parameter (`TRUE` oder `FALSE`) gibt die Polarität der Verbindung an. Die Testfunktion `tst()` (hier jene der Klasse `IOBIT_LOOP`), stellt die Korrektheit der Verbindung fest.

Abschließend wird geprüft, ob der Interrupt für den betreffenden Port ein- oder ausgeschaltet ist.

Die Testqualität hängt wesentlich von der Funktion `tst()` in der Klasse `IOBIT_IO` ab.

```

/* HC04PA1.CPP */
/*
 * Prüfen des parallelen Ports mit Teststecker
 * =====
 */
#include <iodef.h>
#include <portable.h>
#include <mytypes.h>
#include <io.hpp>
#ifdef MYLIB
#include "\mylib\source\io.cpp"
#endif

const INT NUMPORTS=3;

UINT ports[NUMPORTS] = { PC_LPT1, PC_LPT2, PC_LPT3 };
BOOL exist[NUMPORTS] = { FALSE, FALSE, FALSE };

VOID main(INT argc)
{
    cout << endl;
    cout << "PARALLEL-PORTTEST\n";
    cout << "=====\n";

    if (argc>1)
    {
        cout << "Portadressen: ";
        for (INT port=0; port<NUMPORTS; port++)
        {
            cout << hex << setw(3) << ports[port] << ", ";
        }
        cout << endl;
        cout << // siehe nebenstehenden Kasten
        return;
    }

    for (INT port=0; port<NUMPORTS; port++)
    {
        cout << "Port-Nr: " << port;
        cout << "\nPortadresse: "
            << hex << ports[port] << " : ";
        IOBYTE_IO lpt(ports[port]);
        cout << "Port " << port << endl;
        if (lpt.tst(cout,TRUE)==0)
        {
            exist[port]=TRUE;
        }
        else
        {
            exist[port]=FALSE;
            cout << " not";
        }
        cout << " found" << endl;
    }

    for (port=0; port<NUMPORTS; port++)
    {
        if (exist[port]==FALSE) continue;

        INT error=0;

        cout << "Testing port " << port
            << " at " << hex << ports[port] << endl;
        cout << "Insert test plug in port " << port
            << " press a key" << endl;
        cin.get();
        IOBIT_LOOP 11(ports[port]+2,0,ports[port]+1,4,FALSE);
        if (11.tst(cout)) error++;
        IOBIT_LOOP 12(ports[port],0,ports[port]+1,3,TRUE);
        if (12.tst(cout)) error++;
        IOBIT_LOOP 13(ports[port]+2,2,ports[port]+1,6,TRUE);
        if (13.tst(cout)) error++;
        IOBIT_LOOP 14(ports[port]+2,1,ports[port]+1,5,FALSE);
        if (14.tst(cout)) error++;
        cout << error << " errors" << endl;

        IOBIT_IO portbit(4,ports[port]+2);
        cout << "Interrupt: "
            << ((portbit())?"ON":"OFF") << endl;
        cout << "Port ";
        if (error)
            cout << "failed";
        else
            cout << "passed";
        cout << endl;
    }
}

```

Testen der seriellen Schnittstelle HC04SE1.CPP

Ein ähnlicher Test ist natürlich auch für die serielle Schnittstelle durchführbar:

```

Serielle Schnittstelle
=====
PORT-Adressen: 1:3f8 2:2f8 3:3e8 4:2e8
PORT+0 3f8 (DLAB=1) Baudrate low Byte
PORT+1 3f9 (DLAB=1) Baudrate high Byte
PORT+1 PORT+0
 f e d c b a 9 8 7 6 5 4 3 2 1 0
0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 50
0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 300
0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 1200
0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 2400
0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 4800
0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 9600
0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 19200
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 38400
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 56000
PORT+0 3f8 (DLAB=0) DATERN-RX/TX
PORT+1 3f9 (DLAB=0) INTERRUPT-Enable
 7 6 5 4 3 2 1 0
 0 0 0 0 0 CTS OR TX RX
 DTR PE
 RI FE
 DCD BI
 1 2 3 4
Priorität:
PORT+2 3fa INTERRUPT-Status
 7 6 5 4 3 2 1 0
 0 0 0 0 0 0 x x 1 NO INT
 1 1 0 RX-Line
 1 0 0 RX-Data
 0 1 0 TX-Data
 0 0 0 0 Modemst.
PORT+3 3fb LINE-Control
 7 6 5 4 3 2 1 0
 <---PARITY-----> STOPB DATABITS
DLAB SP 0 ODD 0->1 0 0 5 Bits
 1=TXOFF 1 EVEN 1->2 1 0 6 Bits
 0=TXON 0 NO 0 1 7 Bits
 1 YES 1 1 8 Bits
PORT+4 3fc MODEM-Control
 7 6 5 4 3 2 1 0
 0 0 0 0 LOOP OUT2 OUT1 RTS DTR
PORT+5 3fd LINE-Status
 7 6 5 4 3 2 1 0
 0 TS TX BI FE PE OR RX
PORT+6 3fe Modem-Status
 7 6 5 4 3 2 1 0
 DCD RI DSR CTS DDCD DRI DDSR DCTS
Pinbelegung
 TX RX DTR DSR RTS CTS DCD RI GND
24-pol: 2 3 20 6 4 5 8 22 1,7
9-pol: 3 2 4 6 7 8 1 9 5
Serieller Testadapter verbindet:
24-pol: 1-7, 2-3, 4-5-8, 6-20, 11-22, 15-17-23, 18-25
9-pol: 2-3, 1-7-8, 4-6
    
```

Leider gibt es bei der seriellen Schnittstelle mehrere Teststecker und daher auch verschiedenartige Testprogramme. Die Steckerbelegung im Kasten entspricht dem alten IBM-Prüfstecker, das weitverbreitete Programm CHECKIT verwendet aber andere Belegungen. Hier ein Vergleich:

Verbindungen beim Testen der seriellen Schnittstelle

	IBM	CHECKIT 2.0	CHECKIT 3.0
9-pol.	2-3 1-7-9 4-6	2-3 1-4-9 6-7-8	2-3 1-4-6-9 7-8
25-pol.	2-3 4-5-8 6-20 11-22 15-17-23 18-25	2-3 4-5-6 8-20-22	2-3 4-5 6-8-20-22

```

/* HC04SE1.CPP */
/*
 * Testen der seriellen Schnittstelle
 * =====
 */
#include <iodef.h>
#include <portable.h>
#include <ntypes.h>
#include <io.hpp>
#ifdef MYLIB
#include "mylib\source\io.cpp"
#endif

const INT NUMPORTS=4;

UINT ports[NUMPORTS] =
{ PC_COM1, PC_COM2, PC_COM3, PC_COM4 };
BOOL exist[NUMPORTS] =
{ FALSE, FALSE, FALSE, FALSE };

INT error=0;

VOID main(INT argc)
{
  cout << endl;
  cout << "SERIAL-PORTTEST\n";
  cout << "=====\n";

  if (argc>1)
  {
    cout << // siehe nebenstehender Kasten
    exit(EXIT_FAILURE);
  }
  // hier fehlt Kode zur Portidentifikation
  for (port=0; port<NUMPORTS; port++)
  {
    if (exist[port]==FALSE) continue;

    cout << "Insert Test-Plug in port " << port
    << " and press a key!";
    cin.get();

    IOBYTE IO data(ports[port]);
    IOBIT_OUT dtr(0,ports[port]+4);
    IOBIT_OUT rts(1,ports[port]+4);
    IOBIT_IN dcd(7,ports[port]+6);
    IOBIT_IN ri(6,ports[port]+6);
    IOBIT_IN dsr(5,ports[port]+6);
    IOBIT_IN cts(4,ports[port]+6);
    IOBIT_LOOP 11(dtr,dsr,TRUE);
    IOBIT_LOOP 12(rts,cts,TRUE);

    INT error=0;

    cout << "Testing port:" << port
    << " at " << hex << ports[port]
    << endl;
    data = 0xff;
    delay(100);
    if (data()!=0xff)
    {
      error++;
      cout << "OUT:0xff "
      << "IN:" << hex << (INT)data.val()
      << endl;
    }
    data = 0x00;
    delay(100);
    if (data()!=0x00)
    {
      error++;
      cout << "OUT:0x00 "
      << "IN:" << hex << (INT)data.val()
      << endl;
    }
    if (11.tst(cout))
      error++;
    if (12.tst(cout))
      error++;
    cout << error << " errors" << endl;
    cout << "Port " << port;

    if (error)
    {
      cout << " failed";
    }
    else
    {
      cout << " passed";
    }
    cout << endl;
  }
  cout << "Press a key" << endl;
  cin.get();
}
    
```

Anwendung: CMOS-RAM, HC04CM*.*

Wir beschließen unseren heutigen, etwas länger geratenen Abschnitt mit einem Programm zur Kontrolle von IO-Ports, welche auf höherer Ebene tatsächlich nicht unterstützt werden: die Ports des CMOS-RAM (Es sei denn, man versteht das Setup-Programm zu Beginn als eine Art Unterstützung; hier ist der programmgesteuerte Zugriff gemeint). Diese Anwendung kann, unvorsichtig angewendet, Ihren AT oder 386 ganz schön durcheinanderbringen. Bevor wir das Programmprojekt angehen, müssen wir genaue Kenntnis über die einzelnen Speicherbereiche des CMOS-RAM haben. Diese entnehmen wir TGM-DSK-140.

In jedem AT und allen Nachfolgern (386, 386SX und 486) wurde ein kleines 64/128-Byte großes batteriegepuffertes CMOS-RAM eingebaut, welches aber nicht direkt in die IO-Adressen gemappt ist (es wäre ja kein Platz mehr dafür gewesen), sondern über einen eigenen Mechanismus über die Adressen 0x70 (enthält CMOS-RAM-Adresse), 0x71 (enthält CMOS-RAM-Daten) schreib- und lesbar ist. Neben der eigentlichen Uhr, die die CMOS-RAM-Adressen 0 bis 0xf belegt, sind die darauffolgenden Bytes 0x10 bis 0x2d für die Rechnerkonfiguration reserviert. Der darauf folgende Speicherbereich ist im Prinzip frei es werden jedoch, je nach BIOS, dort Festplattentypen gespeichert, die im BIOS nicht von vornherein enthalten sind.

Bedeutung der CMOS-Adressen

00H-0dH	used by real-time clock	
0eH	POST diagnostics status byte	
0fH	shutdown status byte	
10H	diskette drive type	-----+
11H	reserved	
12H	hard disk drive type	
13H	reserved	
14H	equipment byte	
15H-16H	Base memory size	
17H-18H	extended memory above 1M	
19H	hard disk 1 type (if > 15)	
1aH	hard disk 2 type (if > 15)	
1bH-2dH	reserved	-----+
2eH-2fH	storage for checksum of CMOS addresses	
	10H through 2dH	=> checksum
30H-31H	extended memory above 1M	(10H-2dH)
32H	current century in BCD (eg, 19H)	
33H	miscellaneous info.	
34H-3fH	reserved	

Die gewünschte CMOSRAM-Adresse wird in den IO-Port 0x70 geschrieben, die Daten können an Port 0x71 gelesen werden. Eine Ausgabe an Port 0x71 beschreibt diese CMOSRAM-Adresse.

Wir wollen das beschriebene Prinzip ausprobieren und das Jahrhundert-Byte auf CMOS-Adresse 32H auslesen, welches wir ziemlich sicher kennen.

```

/* HC04CM0.CPP */
/*
 * CMOS-RAM: Auslesen des Jahrhundert-Byte
 * =====
 */
#include <stdio.h>
#include <dos.h>
#include <mytypes.h>
#include <portable.h>

#define PC_CMOS_ADD 0x70
#define PC_CMOS_DAT 0x71

VOID main (VOID)
{
    // Adresse des Jahrhundertbyte setzen
    OUT_PORT(PC_CMOS_ADD,0x32);
    INT century = IN_PORT(PC_CMOS_DAT);
    cout << endl;
    cout << "Wir haben jetzt das " << hex << century;
    cout << "-te Jahrhundert, stimmts?" << endl;
}

```

Wenn wir Daten nicht nur lesen, sondern auch schreiben wollen, müssen wir den Sicherheitsmechanismus beachten, der die Daten schützt. Die Adressen 0x2e, 0x2f enthalten eine Prüfsumme für die Daten 0x10-0x2d, die bei einem Batterieausfall oder einer zufälligen Zerstörung der Daten falsch sein wird und daher den weiteren Betrieb

des Rechners bis zur Behebung des Schadens verhindert. Im Normalfall ist das Konfigurationsprogramm im ROM für die Veränderung der Bytes ausreichend. Wer allerdings mit gezielten Veränderungen experimentieren will oder den Speicherbereich bis zur CMOS-RAM-Adresse 0x7f nutzen will, der kann das folgende Patch-Programm als Ausgangspunkt benutzen.

Das hier nicht dargestellte Programm HC04CM1.CPP benutzt nur die Makros IN_PORT und OUT_PORT zur Kommunikation mit dem CMOSRAM. Die Funktionen CMOSRAM_read(), CMOSRAM_write(), CMOSRAM_chksum() und CMOSRAM_printf() dienen zu Kommunikation mit dem CMOS-RAM. Die vier CMOSRAM-Funktionen ergänzen die entstehende Bibliothek MYLIB.LIB.

HC04CM1A.CPP

Da die Ansprache des CMOS-RAM durchaus von allgemeiner Bedeutung sein kann, werden diese Funktionen nunmehr zu einer eigenen Klasse CMOSRAM zusammengefaßt, von der nur ein Objekt, hier cmosram gebildet werden kann. Das Programm zeigt nur mehr die Anwendung der Klasse, nicht aber den Code der Elementfunktionen.

```

/* HC04CM1A.CPP */
/*
 * Schreiben und Lesen des CMOS-RAM, mit Klasse CMOSRAM
 * =====
 */
#include <cmosram.hpp>
#ifndef MYLIB
#include "\mylib\source\cmosram.cpp"
#endif

VOID main(VOID)
{
    INT adr, checksum, value;
    CMOSRAM cmosram;

    constream con;

    for (;;)
    {
        con.clrscr();
        cmosram.print(con);
        checksum=cmosram.chksum();
        con << setxy(1,18);
        con << "Checksum of adr 0x10..0x2d:";
        con << setw(4) << hex << checksum;

        do /* Read address to change */
        {
            con << setxy(1,19) << clrscr();
            con << "Alter byte (adr=0 ends) adr: ";
            cin >> hex >> adr;
            if (adr==0) return;
        }
        while ((adr<0x10) || (adr>0x7f));

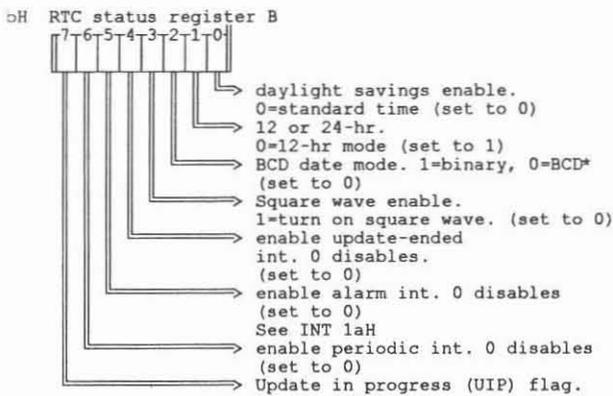
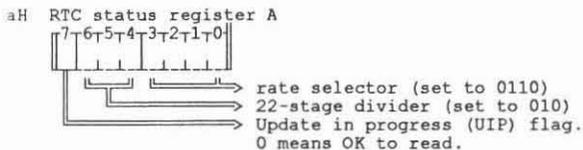
        con << clrscr();
        con << "value: ";
        cin >> hex >> value;
        cmosram.write(adr,value);
        con << clrscr();
        // Update CHECKSUM if necessary
        checksum=cmosram.chksum();
        if (checksum != cmosram.chksum_read())
            cmosram.chksum_write(checksum);
    }
}

```

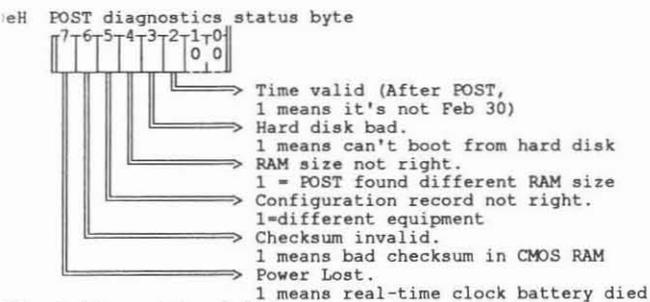
Um auch für den Letztverbraucher anwendbar zu sein, ist das Programm noch ein bißchen spartanisch, benötigt man doch zu seiner Bedienung die genaue Tabelle mit der Bedeutung der einzelnen Bytes (siehe nächste Seite). Daher: Schreiben Sie eine erweiterte Programmversion, welche die Inhalte genauer analysiert. Benutzen Sie dabei die folgende erweiterte Tabelle. Noch ein Vorschlag: Speichern Sie die aktuellen Inhalte des CMOS-RAM in eine Datei und ermöglichen Sie den Vergleich mit dem aktuellen Zustand, etwa als Virusschutz! □

CMOS-RAM im Detail

- idr Description
- 0 current second for real-time clock
- 1 alarm second
- 2 current minute for real-time clock
- 3 alarm minute
- 4 current minute for real-time clock
- 5 alarm hour
- 6 current day of week for real-time clock
- 7 current date of month for real-time clock
- 8 current month for real-time clock
- 9 current year for real-time clock



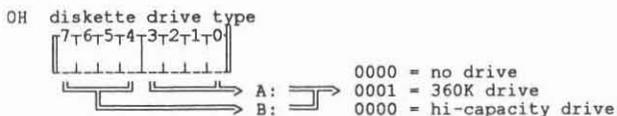
- cH RTC status register C. Read-only interrupt status bits.
- dH RTC status register D.
 Bit 7=1 when a CMOS-RAM is receiving power
 =0 to indicate a dead battery.



fH shutdown status byte

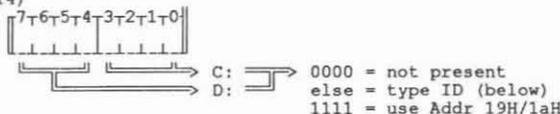
This byte is read upon startup after processor reset in order to determine if the reset was used as a way to get out of 80286 protected mode.

- 0 = soft reset (Ctrl-Alt-Del) or unexpected shutdown
- 1 = shutdown after memory size is determined
- 2 = shutdown after memory test is performed
- 3 = shutdown after memory error (parity check 1 or 2)
- 4 = shutdown with bootstrap loader request
- 5 = shutdown with FAR JMP (restart int controller and jmp to 0:[0467H])
- 6,7,8 = shutdown after passing a protected mode test
- 9 = shutdown after performing block move. See INT 15H SubFn 87H
- 0aH = shutdown with FAR JMP (immediate jmp to address at 0:[0467H])



1H reserved

12H hard disk drive type (for drives C: and D:, when between 1 and 14)

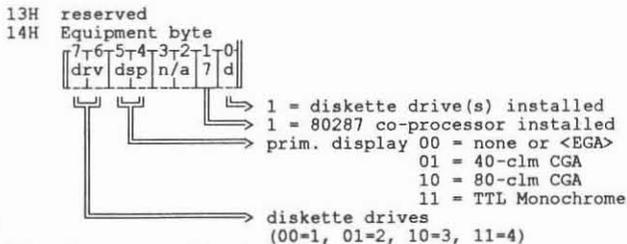


These drive types are pre-defined by the ROM-BIOS. The vectors for INT 41H and INT 46H are initially set to a ROM table containing the information for the types of drive(s) installed. See Hard Disk Parameter Table

Type	Cyls	Hds	Write PreComp	Land Zone	Size
1	306	4	128	305	10M
2	615	4	300	615	21M
3	615	4	300	615	32M
4	940	8	512	940	65M
5	940	6	512	940	49M
6	615	4	0ffffh	733	21M
7	462	8	256	511	32M
8	733	5	0ffffh	733	31M
9	900	15	0ffffh	901	117M
10	820	3	0ffffh	820	21M
11	855	5	0ffffh	855	37M
12	855	7	0ffffh	855	52M
13	306	8	128	319	21M
14	733	7	0ffffh	733	44M
15	Use contents of byte 19 or 1A				
16	612	4	0	663	21M
17	977	5	300	977	42M
18	977	7	0ffffh	977	59M
19	1024	7	512	1023	62M
20	733	5	300	732	31M
21	733	7	300	732	44M
22	733	5	300	733	31M
23	306	4	0	336	10M
24-47	reserved				

If the disk-type nibble is 0fH (15), then the disk type is stored in CMOS address 19H (drive C:) or address 1AH (drive D:)

- Notes:
- Type 1 is the original XT hard disk
 - Type 2 is the standard 20M AT hard disk (and a good first guess)
 - Types 16-23 were added to ROM-BIOS dated 11/15/85



- 15H Base memory (low byte)
- 16H Base memory (high byte)
 0100H=256K, 0200H=512K, 0280H=640K
- 17H extended memory above 1M (low byte)
- 18H extended memory (high byte)
 (in K bytes. 0-3c00H) See INT 15H SubFn 88H
- 19H disk 0 (drive C:) hard disk type
 if (CMOS addr 12H & 0fH) is 0fH
- 1aH disk 1 (drive D:) hard disk type
 if (CMOS addr 12H & f0H) is f0H
- 1bH-2dH reserved
- 2eH checksum of CMOS addresses 10H through 2dH (high byte)
- 2fH (low byte)
- 30H extended memory above 1M (low byte)
- 31H extended memory (high byte)
 (in K bytes. 0-3c00H) See INT 15H SubFn 88H
- 32H century in BCD (eg, 19H)
- 33H miscellaneous info.
 Bit 7=IBM 128K memory option installed
 shadow-RAM is available
 Bit 6=used by "Setup" utility
 first bootup after running setup routine
- 34H-3fH reserved.
 Put your name here for everlasting amusement. ☐

FUZZY LOGIC TECHNOLOGIE

eine praxisorientierte
Entwicklungsmöglichkeit

Arnulf O. Krebs, fuzzy logic

LIT-59, DSK-294:FUZZ

Genauigkeit ist nicht Wahrheit (Henri Matisse, 1947)

Dieser Ausspruch läßt sich sehr gut auf die Fuzzy Logic-Philosophie anwenden: Beschränkung auf das, was nötig ist. So bleiben auch komplexe Aufgaben einfach und übersichtlich lösbar. Entwürfe und Entwicklungen sind früher verfügbar und auf wechselnde Marktverhältnisse läßt sich schneller reagieren.

Trotz des unglaublichen Fuzzy-Booms in Europa, ausgelöst 1990 durch einen Artikel der deutschen Zeitschrift High Tech, gibt es nur wenige professionelle Entwicklungswerkzeuge. **fuzzyTECH** ist ein Werkzeug, welches alle Ansprüche erfüllt. In vielen Tests renommierter Fachmagazine wurden neben der Benutzerfreundlichkeit vor allem die Integration neuester wissenschaftlicher Erkenntnisse hervorgehoben. Als einziges Fuzzy-Entwicklungswerkzeug beinhaltet **fuzzyTECH** eine Option der automatischen Fuzzy-Regelgenerierung mit neuronalen Techniken.

fuzzyTECH wird in Österreich - sowohl in der Industrie als auch in der Forschung - bereits erfolgreich angewandt.

Hochleistung durch Konzentration auf das Wesentliche

Fuzzy Logic ist eine innovative Technologie zur Lösung regelungstechnischer Problemstellungen. Obwohl die Fuzzy Logic vor über 25 Jahren in den USA erfunden und in Europa weiterentwickelt wurde, befinden sich ihre erfolgreichen Anwendungen überwiegend in Japan.

Dort wird diese Technologie bereits breitbandig eingesetzt: Die Anwendungen reichen von Konsumgütern der Unterhaltungselektronik, Hausgeräten oder der Automobilelektronik bis zur Automatisierung technischer Prozesse.

Auch wenn zunächst hauptsächlich Massenprodukte mit Fuzzy Logic aus Japan auf den internationalen Märkten erscheinen, liegen die mittelfristig größten Potentiale in der industriellen Prozeßautomatisierung. Hier wird die Fuzzy Logic zur Erstellung unaufwendiger Lösungen eingesetzt.

Erweiterte Funktionalität und steigende Qualitätsanforderungen, welche zu immer komplexeren Systemen führen, erfordern in der Zukunft innovative Entwurfsmethoden. Fuzzy Logic verzichtet dabei auf "Genauigkeit um jeden Preis": Über das mit Fuzzy formulierte Modellverständnis können komplexe Zusammenhänge formuliert werden, die mit konventionellen mathematischen Methoden nicht mehr zu erfassen sind.

Anders ausgedrückt: Ein sinnvoller Einsatz von Fuzzy ist immer dann gegeben, wenn entweder die Komplexität keine Anwendung von konventionellen mathematischen/statistischen Methoden erlaubt oder die Problemstellung umgangssprachlich einfacher zu beschreiben ist.

Fuzzy Logic ist jedoch nicht ausschließlich auf Anwendungen der Regelungstechnik (als Fuzzy Control) beschränkt - auch in 'reinen' Expertensystemen ermöglicht der Einsatz der "unscharfen Logic" einen oftmals reduzierten Entwicklungsaufwand. Der Autor dieser Zeilen spricht hier aus Erfahrung: Der Einsatz von **fuzzyTECH** bei der Entwicklung eines großen Expertensystems für den Umweltschutz zeigte klar den Vorteil dieser Technologie. Die Wissensakquisition - der 'Flaschenhals' bei jeder XPS-Entwicklung - gestaltete sich im Vergleich zur konventionellen Technologie (hybrides objektorientiertes XPS) nicht nur wesentlich einfacher und zeitsparender, sondern bestimmte

Problemstellungen wären mit anderen Methoden absolut unlösbar geblieben!

Eine der größten Fuzzy Logic-Expertensysteme befindet sich übrigens am Institut für Medizinische Computerwissenschaften der Universität Wien (Prof. Adlassnig). Das seit einiger Zeit im Krankenhausbetrieb eingesetzte System namens CADIAG2 enthält ca. 40.000 Fuzzy-Regeln.

Einkaufsvorteile für PCC-TGM-Mitglieder

Die anschließenden Produktübersichten sind Originale Produktbeschreibungen der Firma INFORM aus Aachen. **fuzzyTECH** ist eine vollständige grafische Entwicklungsumgebung auf dem PC für Fuzzy-Methoden und baut auf der Weiterentwicklung des professionellen Fuzzy Werkzeugs FIT (Fuzzy Intelligence Tool) der Technischen Hochschule Aachen (RWTH) auf.

INFORM hat sich auf die schlüsselfertige Einführung "intelligenter" Steuerungssysteme spezialisiert. Seit 15 Jahren werden die Fuzzy-Technologien in INFORM-Produkten eingesetzt. Auch das erste professionelle Fuzzy-Expertensystem in Europa wurde vor 5 Jahren von INFORM vorgestellt.

Der Autor machte die erste Bekanntschaft mit den Mitarbeitern von INFORM und der RWTH auf der Hannover Messe Industrie '91. Die Fuzzy-Leute führten dort ihr Fuzzy Logic-gesteuertes Modellauto vor, indem sie es mit bis zu 80 km/h durch die Halle rasen ließen, ohne dass es je zu Kollisionen kam! Die Entwicklung des Modellautos zog sich übrigens über mehrere Monate, während der Aufbau der Fuzzy-Wissensbasis und die Implementierung in knapp 2 Stunden erledigt waren. Unterlagen darüber sowie eine Demodiskette (mit Simulation des Autos und der Möglichkeit die zugehörigen Fuzzy-Regeln On-Line zu ändern und diese Auswirkungen zu studieren) sind beim Autor gegen einen Unkostenbeitrag erhältlich.

Für Mitglieder des PCC-TGM gibt es auf alle Software-Produkte einen Rabatt von 60 Prozent. Bitte beachten Sie, daß diese Schullizenzen keinesfalls für die Entwicklung industrieller Produkte und auch nicht zur Drittmittelforschung eingesetzt werden dürfen. Bei Bedarf können Schullizenzen gegen Vollizenzen in Zahlung gegeben werden.

Mehrfachlizenzen (nur Software) werden grundsätzlich nur für ein abgeschlossenes und im gleichen Gebäude ansässige Abteilungsmitglied vergeben. Bei Hochschulen und ähnlichen Institutionen erstreckt sich die Mehrfachlizenz nur maximal auf einen Lehrstuhl. Die Bestellung einer Mehrfachlizenz muß für die gewünschte Anzahl der Anwender auf einmal erfolgen. Folgender Multiplikator für den Einzelpreis wird zugrundegelegt: 1 Anwender = 1,0 mal; 2 = 1,79; 3 = 2,56; 4 = 3,28; 5 = 3,98; 6 = 4,64; 7 = 5,27; 8 = 5,87; 9 = 6,43; 10 = 6,96. Jede weitere Lizenz kostet zusätzlich 50% des Einzelpreises.

Zur Bestellung verwenden Sie bitte den umseitig abgedruckten vorgedruckten Bestellzettel. Bitte geben Sie beim Kauf eine Schullizenz unbedingt den Zusammenhang zu einer der genannten Institutionen mit näherer Bezeichnung an (z.B. Lehrer oder Schüler an PCC-TGM, Assistent am Inst., PCC-TGM-Mitglied zum Zwecke der eigenen Ausbildung etc.). □

Anm: Die Produktübersichten sind in TGM-LIT-59 zusammengefaßt und können bei Bedarf beim PCC-TGM oder auch bei Herrn Krebs angefordert werden.

Fax: 0222 - 36 34 80

Preisliste Stand Januar 1993

BESTELLUNG

fuzzy logic
entwicklung & vertrieb

dipl.-ing. arnulf o. krebs
 rodlergasse 5/17

a-1190 wien

Bitte kreuzen Sie nachfolgend Ihre Bestellung und/oder Nachfragen an und senden Sie uns die Seite per Post oder per Fax.

FAX 0222-36 34 80 ☎ 0663-82 88 71

Name, Titel, Funktion	Zugehörigkeit (nur bei Erwerb von Schullizenzen)
Institut/Firma/Organisation	Adresse
Telefon Telefax	Unterschrift
Ort, Datum	

(Ich bestätige durch meine Unterschrift, daß ich Schullizenzen weder für industrielle Projekte noch zur Drittmittelforschung einsetzen werde.)

Ich bestelle nachfolgend angekreuzte Produkte (Mehrfachlizenzen reduziert):

Einzelprodukte	Anzahl	Schullizenz	Vollizenz
<i>fT</i> -Shell Version 2.1		öS 8.520,-	öS 21.300,-
<i>fT</i> -C-Precompiler Version 2.1		öS 8.520,-	öS 21.300,-
<i>fT</i> -Debugger Version 2.1		öS 6.480,-	öS 16.200,-
<i>fT</i> -NeuroFuzzy-Modul		öS 6.120,-	öS 15.300,-
<i>fT</i> -On-Line-Modul		öS 29.560,-	öS 73.900,-
<i>fT</i> -Precompiler für FUZZY-166		öS 8.520,-	öS 21.300,-
Fuzzy-Hardware	Anzahl	Schullizenz	Vollizenz
FUZZY-166 Prozessor		öS 1.790,-	öS 1.790,-
FUZZY-166-Evaluation Board		öS 15.750,-	öS 15.750,-
Pakete	Anzahl	Schullizenz	Vollizenz
<i>fT</i> -C-Precompiler-Paket		öS 15.960,-	öS 39.900,-
"On-Line"-Paket		öS 39.800,-	öS 99.500,-
FUZZY-Hardware-Paket		öS 28.820,-	öS 49.400,-

Schicken Sie mir bitte nachfolgend angekreuzte Unterlagen zu:

Software-Wartungsverträge: 12% jährlich vom Softwarepreis	
Quick Reference Guide zur FUZZY-166 Hardware/Software-Produktpalette	
Workshop "Fuzzy-Entwurfsmethodik: Von der Aufgabenstellung bis zur Implementation" Termine (2-tägig): Januar bis Juni 1993 einmal pro Monat, Ort: Aachen, Preis: DM 1.420,-	
Seminar "Industrielle Anwendungen der Fuzzy Logic Control", 25.1. Essen, 3.2. Wuppertal, 8.3. und 3.5. Aachen, 29.3. Wien, 5.5. Nürnberg, 7.6.1993 Frankfurt am Main	
Symposium "3. Aachener Fuzzy-Symposium", 10. Mai 1993, Aachen, Preis: DM 870,-	

Alle Preise verstehen sich zuzüglich 20% USt. und gelten ab 1.1.1993

PCAD 6.0

Kurt Wiesauer, Rekirsch Elektronik

LIT-56, DSK-301-313, 314

Weitere 90 PCAD-6.0-Plätze für HTLs

Somit sind an Österreichs Höheren Technischen Bundeslehranstalten schon mehr als 200 PCAD System installiert, was PCAD zur absoluten Nummer 1 sowohl in der Ausbildung als auch in der Industrie macht. Da die neue PCAD-Version 6.0 neben vielen neuen Möglichkeiten auch eine komplett neue Benutzeroberfläche hat, ist die Bedienung jetzt noch einfacher geworden. Dies bringt für alle PCAD-Anwender einen noch größeren Bedienungskomfort. So werden jetzt alle Projekte von einem eigenen Dateimanager verwaltet, der einen größeren Überblick gewährleistet und schnelle Projektwechsel durchführen kann. Auch die Systemeinstellungen sind einfacher geworden, und so gehören die Vorbereitungen eines neuen Projekts fast der Vergangenheit an. Schon bei den ersten Einschulungen haben die HTL-Lehrer mit großer Begeisterung PCAD-6.0 aufgenommen. Durch die große Anzahl von PCAD-ausgebildeten Schulabgängern wird die Position von PCAD als Nummer 1 weiter ausgebaut. Die Lehrerschulung ist wichtig, um diese Begeisterung auch an die Schüler weitergeben zu können.

Die oft gängige Meinung, daß es wesentlich einfachere Programme auf dem Gebiet der Leiterplattenentflechtung als PCAD gibt, ist richtig, allerdings ist es für die Schüler wichtig, schon in der Schule mit dem CAD-Programmen konfrontiert zu werden, welches sie dann später in der Praxis einsetzen werden. Was AutoCAD für allgemeine CAD ist, stellt PCAD für PCB-Design dar.

Einige der Neuheiten bei PCAD-6.0 sind unter anderen:

- * neue Oberfläche bei allen Modulen
- * 32-Bit-Datenbank für größere Auflösung
- * 386er Code mit PharLap DOS-Extender
- * Autoplazierer im Layout integriert
- * automatische Bemaßung im Layout
- * Übersichtliche Picklisten beim Arbeiten
- * Expertmode mit PullUp-Menüs
- * einige Erweiterungen im Layoutmodul
- * Weiterentwicklung des Autorouters
- * Verbesserte Multilayermöglichkeiten
- * und vieles mehr.

Als besondere Erleichterung des Unterrichts für alle Schüler, die auch über einen Heim-PC verfügen, darf die frei kopierbare Version angesehen werden, die mit der einzigen Einschränkung von 35 Bauteilen ansonst den vollen Leitungsumfang bietet.

Anm.: Sie können die frei kopierbare Version von PCAD 6.0 über den TGM-Diskettenservice (DSK-301-313) beziehen, ebenso die Demoversion von ECAM (DSK-314). Die zu diesem Beitrag vorgesehenen Bilder konnten aus Platzmangel nicht abgedruckt werden. Fordern Sie weitere Information bei Rekirsch-Elektronik (0222)/25-36-26/31. Weitere Details zu PCAD-6.0 entnehmen Sie den beiden folgenden Seiten.

PCGERBER 7.0/ECAM 3.0 von CSI

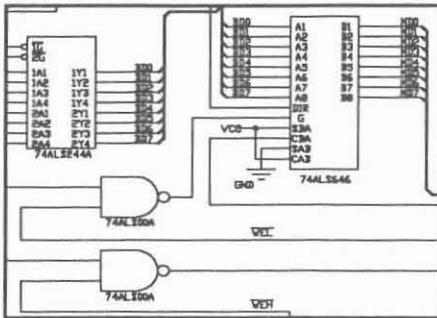
Jetzt kann der PCB Layouter die Gerber- oder HPGL-Dateien direkt am Computer testen und ausbessern, bevor er sein Projekt zur Fertigung geben kann. Der Designer spart durch rechtzeitiges Erkennen von Fertigungsfehlern viel Zeit und auch so viel Geld. Ebenfalls entfallen die teuren "Probephotos", da alle Designdaten schon am Computer überprüft werden. Es können bis zu 32 Gerberfiles zur selben Zeit geladen werden, und so hat der Anwender die totale Kontrolle über sein Projekt. Es können auch wichtige Fertigungs- und Produktionsschritte vorgenommen werden, die sonst von der Lieferfirma um teures Geld gemacht werden. Aber PC-GERBER kann noch mehr:

- * Einfaches Userinterface(Pulldown)
- * Multilayer bis 32 Lagen
- * 32-Bit-Datenbank mit 1/1000 mil Inch
- * View & Edit
- * Mehrfachnutzengenerierung
- * Ausgabe auf Plotter und Laserdrucker
- * Abschirmflächengenerierung
- * eigene Codeerstellung
- * Teardrop Generierung
- * Clear für Silkscreen
- * Online Design Rule Check
- * Testpunktgenerierung
- * Bohrdatengenerierung für NC-Maschine
- * Kontrolle Ihrer Designparameter
- * Meßfunktionen
- * für MSDOS und UNIX

Specetra von CCT jetzt auch für MSDOS

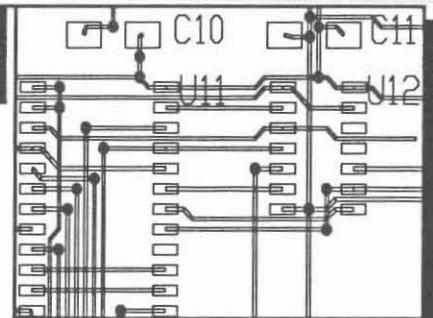
Der beste HighTech Autorouter der Welt!! Dieses Prädikat verleihen sich viele Autorouter selbst doch nur Specetra hat es sich wirklich verdient. Durch einen neuen Algorithmus kann Specetra neben **Maze**, **Ripup&Retry**, **Optimierung** und **Push&Shove** auch die ganz neue "Shape Based"-Technologie, welche bei sehr dichten Leiterplatten zu überraschend guten Ergebnissen führt. Der Specetra-Autorouter ist für **Fast Circuit** genauso optimal einzusetzen wie bei **Hybriden** oder bei **analog-digital**-Platinen. Benchmarks bei einigen Anwendern haben verblüffend gute Ergebnisse gebracht.

Das Geheimnis seines Erfolges ist ein gridloses Routen, welches ein optimales Ausnützen des Platzes sichert. Die Einstellung ist kinderleicht, was Sie nur angeben, sind Ihre spezifischen Routingparameter, wie minimale und maximale Werte von Breite, Länge, Abstände, Masseflächen, Fanout, Abschirmungen etc. Specetra kann sowohl interaktive (Pin to Pin) als auch vollautomatisch arbeiten. Die Einstellungen entsprechen dem, was Sie bei händischem Routen selbst berücksichtigen. **Hardware:** MSDOS, UNIX. **Interface:** alle gängigen PCB-CAD-Systeme. □



p-cad

**USER
Information
Nr. 12/92
Dezember 1992**



1/2

Infos zur Version PCAD 6.0:

Zu Beginn zwei Entschuldigungen:

* Durch die gleichzeitige Auslieferung und Installation von weiteren 90 PCAD Arbeitsplätzen bei 15 österreichischen HTLs und die Einschulung der zuständigen Lehrer, ist das persönliche Ausliefern der Updates PCAD 6.0 bei den meisten Industriekunden entfallen.

* Die ersten Updates sind sehr spät eingetroffen, da es in den USA Probleme mit den Hardwarelocks gegeben hat. Einige Updates fehlen leider immer noch, die aber in Kürze von CADAM geliefert werden.

Tips zur Installation:

Verwenden Sie die PCAD Set Variablen zur Einstellung von PCAD:

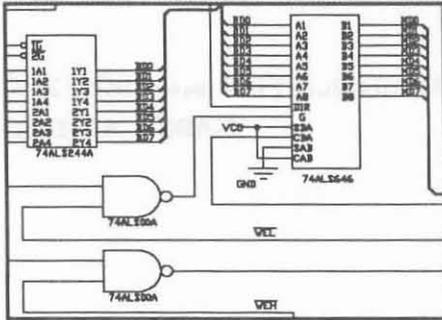
- * set PCADDRV= gibt den Pfad von pcaddrv.sys an (z.B.: x:\pcad)
- * set PCAD_PATH= gibt den Pfad der *.cfg file (bisher append) und der *.exe Files an
(z.B.:x:\pcad\xfg;x:\pcad\exe)
- * set PCADDOSX=-swapdir gibt den Pfad für die swap files an (z.B.: x:\ Ramdisk)
- * set PCAD_HOME= gibt das Startverzeichnis für PCAD60 an, egal wo Sie es starten.

Wenn Sie bisher mit Unterverzeichnissen für ein Projekt gearbeitet haben (z.B.: SCH,PCB, SYM, PRT,etc.), so verwenden Sie bitte mit PCAD 6.0 nur noch ein Verzeichnis pro Projekt, da es sonst vom Shell nicht verwaltet werden kann. Im Shell werden alle angezeigten Schaltpläne automatisch "zusammengelinkt", also bei verschiedenen Revisionen Ihres Projektes aufpassen. Im Cross-Reference File wird das \$path,library,etc. nicht mehr unterstützt, da die Informationen schon in Library "Set Path.." eingegeben worden sind.

Hinweis zur Mausbedienung:

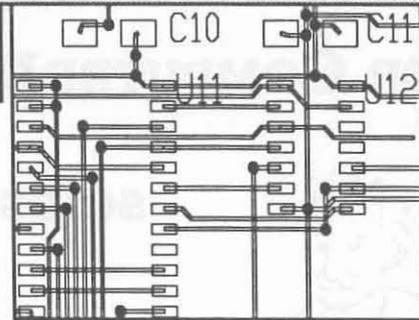
- * PCAD verwendet jetzt den DOS Maustreiber als Basis für den Treiber im pcaddrv.sys
- * in pcaddrv.sys wird mit "MOUSE 1 1" das Mausport auf com1: gesetzt und die Maustasten getauscht. Mit MOUSE 1 0 wird die gewohnte Maustastenbelegung erreicht.
- * Bei einer 3- Tasten Maus (Logitech) ist die mittlere Taste der Aufruf des Menüs (tauschbar).
- * Bei einer 2- Tasten Maus (Microsoft) ist das Menüaufruf mit "ctrl+linke Taste" (tauschbar).
- * Mit "shift+linke Taste" führen Sie den PAN Befehl aus.
- * Bisher wurde eine Koordinateneingabe mit dem Drücken der Maustaste gemacht, jetzt zählt erst der Punkt, wenn Sie die Maustaste wieder auslassen. Wenn Sie also die Maustaste drücken, die Maus bewegen und dann wieder auslassen, gilt der Punkt des Auslassens als Koordinateneingabe. Bei der Space Taste verhält es sich wie bisher.





p-cad

USER
Information
Nr. 12/92
Dezember 1992



2/2

Hinweis bei Verwendung von hochauflösenden Grafikkarten (ELSA) :

- * Stellen Sie im pcaddrv.sys die "Fonts" auf klein (z.B.: font8x8.fnt), um eine größere Zeichenfläche zu erhalten. Wenn Sie zusätzlich das rechte Menü ausblenden, gewinnen Sie noch mehr Arbeitsfläche.
- * Zur Elsa Grafikkarte: Es gibt einen Standardtreiber ohne spezielle Elsa Features, der mit den Updates mitgeliefert worden ist. Es gibt darüber hinaus auch einen Displaylisttreiber mit den gewohnten Elsa Features, der gerade fertig gestellt wurde. Da Elsa für die Entwicklung dieses Treibers etwa 2 Monate investiert hat und es für Elsa Grafikkarten keinen Wartungsvertrag gibt, wird ein Unkostenbeitrag für die Auslieferung verrechnet. Der genaue Preis steht noch nicht fest, soll aber bei etwa ÖS 4.000,- liegen. Sonderpreise für Mehrfachinstallation gibt es natürlich. Der Schutz des Treibers erfolgt durch ein Pal auf der Karte. Ich bitte alle Elsa Benutzer, bis 10.01.1993 Ihr Interesse an diesem Treiber mittels Fax zu bestätigen, da bei einer großen Sammelbestellung der Preis sicher noch geringer werden kann.

Neue VGA Grafikkarte von ELSA: bis 1024x768 Auflösung und AutoCAD+ Win3.1 Treiber um 6.300,-

PCAD 6.0 auf 386er PC's ohne Coprozessor:

Es gibt hierbei einen Fehler beim Pharlap Dosextender: Sie dürfen keinen Speichermanager (Himem, Qemm, 386max, etc.) verwenden. Bei 386ern mit Coprozessor und bei 486er gibt es diesen Fehler natürlich nicht.

Vertragsabkommen zwischen PCAD und Spectra für DOS:

Manche von Ihnen haben sich sicher schon von der ungeheuren Leistungsfähigkeit des Spectra Autorouters von CCT überzeugen können. Jetzt gibt es ein Vertriebsabkommen zwischen CCT und CADAM und auch ein erstklassiges Interface zwischen den beiden Systemen. Genauere Infos zu dieser Partnerschaft wird es in Kürze geben.

Interface PCAD 6.0 zu Maxroute:

Dies ist schon fertig und über Mailbox bei Massteck USA oder per Post bei WR anzufordern.

PCAD 6.0 ohne Hardwarekey:

Seit kurzem gibt es von PCAD 6.0 eine Version ohne Hardwareschutz, die auf 35 Bauteile im Schaltplan und im Layout beschränkt ist. Sie können jedoch sonst alle **PCAD Features** der Version 6.0 ausnützen. Wenn Sie also jemanden innerhalb oder außerhalb Ihrer Firma kennen, der sich für PCAD interessiert, so kann dieser eine Evaluationsoftware PCAD 6.0 mit Kurzhandbuch um etwa ÖS 5.000,- bekommen. Bei einem möglichen Kauf **bekommt der Käufer** dann den Preis gutgeschrieben, und Sie bekommen den selben Betrag als **Provision** auf Ihr Bankkonto.

Last but not least möchte ich allen PCAD Anwendern frohe Weihnachten und ein gutes neues Jahr wünschen und freue mich auf ein Wiedersehen im neuen Jahr,

Mit besten Grüßen

Ihr

Kurt Wiesauer



DER COMPUTERDOKTOR

18.1.93 absolute Fix-Preise inkl. 20% MwSt.!

KASSA-ABHOLPREISE**selbst Schuld !**... wer nicht **31 020 31** anruft!

Umbau - Erweiterung - Service - Wartung
Montagematerial - Rahmen - Kabel - Zubehör
auch Nachnahmeversand

386DX40 Mainboard - große Leistung mit 128K Cache und AMI Bios	2.670,-
486SX33 Mainboard - 256K Cache, AMI Bios, 2x OPTI Local Bus	5.970,-
170MB Festplatte - Conner AT-Bus-Festplatte - 15ms, 64K Cache	4.470,-
212MB Festplatte - Maxtor AT-Bus-Festplatte - 15ms, 64K Cache	5.670,-

alle Festplatten + Mainboards inklusive **ATLast!** - Software für Speicherverwaltung.

4x 1, 1x 4MB SIMM - neuer Preis für Speicher-Chips 70ns	1.800,-
HP DeskJet 550C - der neue Farb-Tintenstrahldrucker	9.480,-
ET4000 True Color - VGA-Karte - 16,7Mio Farben bei 640x480, VESA 70/72Hz	1.290,-
FaxSwitch plus - automat. Fax-Weiche - für Fax, Telefon, Anrufbeantworter	1.470,-
A R M O U R - Testaktion für unsere neue Anti-Virus-Software	GRATIS

SYWA EDV-Handelsges.mbH

Gentzgasse 9 vis a vis vom WIFI
 1180 Wien U6 - Station Volksoper
 Tel: **31 020 31**, Fax: **31 020 32**

viele Stammkunden

individuelle Beratung, Produktkenntnis

Betriebszeiten: Mo - Fr: 9.30 - 18.30 Uhr
 Samstag: 10.00 - 13.00 Uhr
 Einkaufssamstage: 10.00 - 18.00 Uhr

faire Preise - große Auswahl

Verlangen Sie unsere Fax-Preisliste - aktuell jede Woche!

DER COMPUTERDOKTOR

18.1.93 absolute Fix-Preise inkl. 20% MwSt.!

KASSA-ABHOLPREISE

Unsere Notebooks

für aktuelle Preise bitte Anrufen!

**1 Jahr Garantie - Service binnen 24 Stunden -
Erweiterungen, Umbau und Service erfolgen bei uns**

PC-Book 286^{enhanced} Notebook mit 16MHz, 2MB RAM, 22 x 16 x 3cm

unter 1Kg mit Batterien, DR DOS 6.0, Termin-Software, Handbuch
Transferkabel, LCD-Bildschirm 640x400, Netzteil (= Ladegerät)
5 Standard-Akkus für bis 4 Stunden Betrieb, deutsche Tastatur
1 serieller u. 1 paralleler Norm-Anschluß, PCMCIA 2.0 Kontakt

mit 60 MB Festplatte 12.990,-

Tragtasche für PC-Book	480,-
PCMCIA SRam IC-Karte - 1MB	3.990,-
externes Floppylaufw. 1,44MB	1.980,-
zusätzlicher Akku-Satz (5 Stk.)	480,-
PCMCIA Intel Modem IC-Karte	3.990,-
PCMCIA Intel Modem+Fax-Karte	4.980,-

Jetbook 386SX Notebook mit 25MHz, 2MB RAM (aufrüstbar bis 16MB!)

Mit internem 1,44MB Diskettenlaufwerk, hintergrundbeleuchteter
VGA LCD-Schirm mit 32 Graustufen + Festplatte automat. abschaltbar
Akku für 2 Stunden Betrieb, Netzteil (=Ladegerät), deutsche Tastatur
Anschluß für ext. VGA-Monitor + Tastatur, Handbuch, Tragtasche

mit 85 MB Festplatte 14.970,-

Aufzahlung auf 4MB Hauptspeicher	960,-
Modem 2400Bd + Fax 9600Bd	3.600,-
Externes Floppylaufw. 1,2MB	2.820,-
Bus-Box für externen Busanschluß	3.600,-
zusätzlicher Akku	960,-
Zusatz-Netzteil/Akku-Ladegerät	2.400,-

SLIM 486SLC Notebook mit 25MHz, 4MB RAM, 28 x 22 x 4,3cm, 2,8KG

Integrierter Trackball, 1,44MB Diskettenlaufwerk, PCMCIA 2.0 Kontakt
hintergrundbel. 10" VGA LCD-Schirm - 64 Graustufen für Simultanbetrieb
Suspend Modus für Schirm u. Festplatte, Akku für 2 1/2 Stunden Betrieb
Netzteil 90-260V (= Ladegerät), deutsche Tastatur, Handbuch, Tragtasche
Anschluß für VGA-Monitor (bis 1024x768), Tastatur & PS/2 Maus

mit 85 MB Festplatte 24.990,-

Aufzahlung 120MB Festplatte	3.000,-
- " - auf 8MB Hauptspeicher	4.400,-
Externes Floppylaufw. 1,2MB	2.820,-
PCMCIA Intel Modem + Fax-Karte	4.980,-
Tisch-Station mit PC-Anschlüssen	4.400,-
zusätzlicher Akku	960,-
Zusatz-Netzteil/Akku-Ladegerät	2.400,-

Jetbook 386DX, 486DX Notebook mit 33MHz, 32K Cache, 4MB RAM

Farbversion kommt im Februar 93

3,4Kg inkl. Akku, Abmessungen: 285 x 230 x 50mm
Tauschbares CPU-Modul 386/486DX, 1,44MB Diskettenlaufwerk,
10" VGA LCD-Schirm - 64 Graustufen, Suspend-Modus
für Schirm und Festplatte, Akku für bis 3 Stunden Betrieb
Netzteil (Ladegerät), deutsche Tastatur, Handbuch, Tragtasche
Anschluß für externen VGA-Monitor, Tastatur oder PS/2 Maus

386DX mit 85 MB Festplatte 24.990,-

Aufzahlung auf 486DX-Version	6.000,-
- " - 120MB Festplatte	3.000,-
- " - 8MB Hauptspeicher	4.400,-
Externes Floppylaufwerk 1,2MB	2.820,-
Modem 2400Bd + Fax 9600Bd	4.000,-
Tisch-Station mit PC-Anschlüssen	4.400,-
zusätzlicher Akku	960,-
Zusatz-Netzteil/Akku-Ladegerät	2.400,-

Fax - Modems - Fax-Umschalter (für Puls- und Tonwahl)

Twincom Modems - von Image Communications U.S.A.

14.400 D/Fi	internes Fax-Modem MNP 2-5, V.32bis, V.42, V.42bis, V.22bis, -neu Fax V.17! + Grup III	5.280,-
14.400 D/Fe	externes Fax-Modem MNP 2-5, V.32bis, V.42, V.42bis, V.22bis, -neu Fax V.17! + Gruppe III	5.490,-

VIVA Modems - von CPI, U.S.A. Pocket Fax-Modem komplett inkl. Software - mit MNP5 und V.42bis

2.970,-

Intel

SatisFAXion 200 - Faxmodem-Karte - mit MNP1-5 + V.42bis 2400 Baud Modem!	5.980,-
SatisFAXino 400 - intern wie Modell 200 aber mit 14.400 Baud Modem	7.490,-

FaxSwitch Plus	automatischer Faxumschalter für Fax, Telefon und Anrufbeantworter	1.470,-
FaxSwitch 333	automatischer Faxumschalter für Fax, Telefon und Anrufbeantworter (programmierbar!)	2.940,-
FaxSwitch 555	automatischer Faxumsch. für Fax, Telefon, Anrufbeantworter u. Modem (programmierbar!)	3.540,-

DER COMPUTERDOKTOR

18.1.93 absolute Fix-Preise inkl. 20% MwSt.!

KASSA-ABHOLPREISE

DO IT YOURSELF - Computersysteme

für aktuelle Preise bitte anrufen!

Desk 386 - 80386DX AMD-CPU - 40MHz

- 1MB RAM, 128K Cache
- 6x 16-bit + 2x 8-bit Slots
- Sockel für 80387-Coprozessor
- Tischgehäuse, 200Watt Netzteil
- 105MB Festplatte - 15ms
- 1,44MB Diskettenlaufwerk
- 2 serielle + 1 parallele Schnittstelle, Game Port
- große AT-Tastatur mit Klick
- Anleitung für den Zusammenbau in deutsch

Preis: **8.970,-**

Desk 486 - 80486SX Intel-CPU - 33MHz

- 1MB RAM, 256K Cache
- 6x 16-bit + 1x 8-bit Slots
- Cache für Video-Bios schaltbar
- Tischgehäuse, 200Watt Netzteil
- 105MB Festplatte - 15ms
- 1,44MB Diskettenlaufwerk
- 2 serielle + 1 parallele Schnittstelle, Game Port
- große AT-Tastatur mit Klick
- Anleitung für den Zusammenbau in deutsch

Preis: **10.980,-**

Local Bus + 990,-

Für beide Systeme: Zusammenbau und Test durch uns + 900,-

Videokarte und Monitor extra laut nachfolgender Liste

Aufzahlungen - mögliche Erweiterungen - für beide Systeme:

Diskettenlaufwerk 1,2MB	780,-	120MB Festplatte 15ms statt 105MB	990,-
Speichererweiterung SIMM 70ns - pro 1MB	450,-	210MB Festplatte 15ms statt 105MB	2.400,-
Coprozessor für 386-40 von Cyrix	1.590,-	486DX-33 Prozessor statt SX-33	1.980,-
Jet Infra-Maus drahtlose Infrarot-Maus	570,-	486DX-50/DX2-66 Prozessor statt SX-33	6.000,-
MS-DOS 5.0 (zum System) - deutsch	780,-	EISA-Bus (32bit) statt ISA-Bus (16bit)	1.800,-
MS Windows 3.1 (zum System) - deutsch	1.200,-	iceFAN kühlt die CPU !	990,-
250MB FDD-Streamer- zum Anschluß am Floppy-Controller (auch als 3. Laufwerk!) inkl. Software			3.180,-
High Speed Interface - für 2,88MB Diskettenlaufwerk + 1MB Datentransfer für obigen Streamer			1.290,-
Sound Galaxy NX - Sound-Karte zum Spitzenpreis mit CD-Anschluß für Multimedia			990,-
Video-Special - VGA-Monitor + Pico VGA-Karte - unsere günstigste Kombination			3.990,-
Tiger-Video-Special - VGA-Monitor strahlungsarm/non interlaced + Tiger-10 Grafik-Karte			7.980,-
		14" 0,28mm 640x480, 800x600, 1024 x 768	
Video- & Grafikkarten alle inklusive Treiber für Windows 3.1			
Pico VGA plus 16-bit VGA-Karte mit 256K bis 800x600			570,-
ET4000 True Color Tseng ET4000 1MB - bis 1280x1024, 1024x768 NI, 70Hz, True Color bei 640x480			1.290,-
Prism Accelerator Imaging Systems, Coprozessor für DOS u. Windows,- bis 1024x768 NI, 70Hz			3.990,-
Enhanced VGA Local Bus OPTI - ET4000 Chip - 1MB, bis 1024x768 NI, 70Hz, bis 33MHz			2.280,-
VL 811-ET4000 Local Bus VESA - ET4000 Chip - 1MB, bis 1024x768 NI, 70Hz, bis 50MHz			3.470,-
VL 911-S3 Local Bus VESA - S3-Chip - 1MB, bis 1024x768 NI, 70Hz, bis 33MHz CPU			5.970,-
STEALTH VRAM Diamond Systems USA, die Referenz-Grafikkarte mit S3-Chip, 1MB			4.470,-
TIGER-10 TTGA Grafikkarte mit TT34010-60MHz, 1MB VRAM, 512KB DRAM			3.990,-
GRAPHICS ULTRA PRO neuer Mach 32 Coprozessor von ATI + 2MB VRAM			7.980,-

Monitore

Farbmonitore mit einer Auflösung bis zumindest 1024x768 Bildpunkte

GA-S	14" 0,28mm 640x480, 800x600, 1024 x 768	3.480,-
GA-NI/LR	14" 0,28mm 640x480, 800x600, 1024 x 768	4.470,-
AG MX148*	14" 0,28mm 30-64KHz, 50-120Hz, Auflösung bis 1280 x 1024	7.980,-
AG MX144*	14" 0,28mm 30-64KHz, 50-120Hz, Auflösung bis 1280 x 1024	8.980,-
AG MX15HL	15" 0,28mm 30-64KHz, 50-120Hz, Auflösung bis 1280 x 1024	8.980,-
OKIA 449A	15" 0,28mm strahlungsarm nach TCO'91, 30-62KHz, 48-100Hz, bis 1024x768	12.990,-
M 1710D	17" 0,28mm Flat Screen entspiegelt, getönt, bis 1280 x 1024, hor: 30-65KHz, ver: 40-120Hz	11.970,-
AG MX17HL	17" 0,28mm multiasync, strahlungsarm, elektron. geregelt - bis zu einer Auflösung von 1280 x 1024	18.980,-

*) mit Sony Trinitron Bildröhre

DER COMPUTERDOKTOR

18.1.93 absolute Fix-Preise inkl. 20% MwSt.

KASSA-ABHOLPREISE!

Mainboards - mit CPU, ohne Ram lagernd!

80386DX - 40MHz /128K Cache	2.670,-
80486SX - 33MHz /256K Cache	4.980,-
80486DX - 33MHz /256K Cache	6.990,-
80486DX - 50MHz /256K Cache, VESA Local Bus	10.980,-
80486DX2- 66MHz /256K Cache, VESA/OPTI Local Bus	10.980,-
OPTI Local Bus für 486SX+DX33-Mainboards	+ 990,-
EISA Aufzahlung auf obige 486-DX Mainboards	+ 1.800,-

RAM Systemspeicher 1MB SIMM 70ns	450,-
4MB SIMM 70ns	1.800,-

Festplatten / Laufwerke

Diskettenlaufwerk (japan.) 3 1/2" = 1,44MB	690,-
Diskettenlaufwerk (japan.) 3 1/2" = 2,88MB	1.980,-
Diskettenlaufwerk (japan.) 5 1/4" = 1,2MB	780,-
Kombi-Diskettenlaufwerk (japan.) 5 1/4" + 3 1/2"	1.980,-
105MB ST3120 Seagate IDE Festplatte 15ms	3.570,-
130MB ST3144 Seagate - " - 16ms	3.990,-
170MB CP30174 Conner - " - 15ms	4.470,-
212MB EN7213A Maxtor - " - 15ms 64K Cache	5.670,-
240MB LPS240A Quantum - " - 12ms/2J. Garantie	6.990,-
425MB PD425A Quantum - " -	13.980,-

Syquest SQ 5110 88MB Fest-Wechselplatten-System	4.980,-
SQ 800 88MB Wechselplatte	1.590,-

Gehäuse

Baby AT-Gehäuse mit 200W Netzteil	870,-
Baby Tower mit 220W Netzteil	1.170,-
Middle Tower mit 220W Netzteil	1.470,-
Big Tower mit 220W Netzteil	1.680,-
SCSI-Gehäuse extern mit Kabel	
5 1/4" HH - Netzteil 43 Watt	1.980,-
5 1/4" FH - Netzteil 65 Watt	2.970,-
5 1/4" 2x FH - Netzteil 200 Watt	3.990,-

Netzteile - U.S.V.

Papst-Ventilatoren = Leise-Lüfter in versch. Größen auf Anfrage	
PS/2 Netzteil - 220 Watt, TÜV, mit Schalter	960,-
Tower Netzteil - 220 Watt, TÜV, mit Schalter	1.200,-
PS/2 Netzteil 250 W.+ Notstromvers. (Batterie 5 1/4")	3.990,-
Tower Netzteil 300 W.+ Notstromvers. (Batterie 5 1/4")	3.990,-

IDE - FDC - SCSI

IDE (AT-Bus) Interface + Floppycontroller, ser.+ par. I/O	240,-
IDE Interface wie oben auch für 2,88MB Floppy	1.470,-
IDE Interface wie oben mit 2,88FDC - ohne I/O	1.290,-
IDE Interface + Floppycontroller (mit Bios für alte AT's)	990,-
IDE Cache-Spiegel-Interface + Floppycontroller	3.390,-
IDE Interface + SCSI Hostadapter + Floppycontroller	1.470,-
SCSI-Druckerport-Interface - für externes SCSI-System	1.470,-
Rancho SCSI Host-Adapter 8-bit (z.B.: für Syquest)	990,-
Adaptec 1542b SCSI Host-Adapter mit FDD-Controller	3.480,-
Floppy-Controller 360KB - 2.88MB für 4 Floppylaufwerke	990,-
I/O-Karte: 2 seriell + 1 parallel + Game Port	180,-

ICECAP von VELOX ... kühlt die CPU	1.980,-
iceFan - CPU-Kühlung	990,-

Tastatur - Maus - Trackball

AT-Tastatur mit Klick, deutsch	480,-
Cherry AT-Tastatur (B.R.D.), deutsch	990,-
Jet Infrarot-Maus, drahtlos!, - " -	570,-
SuperTrack Notebook-Trackball	1.470,-
Logitech Mouse Man - PS/2 + seriell	780,-
Logitech Track Man - Trackball	1.200,-
Microsoft Mouse - die Originale!	890,-

Telecom - PANASONIC postgenehmigt!

Schnurlostelefon KX-T9010BS	5.970,-
Faxgerät UF-121	9.990,-
Faxgerät UF-128M mit Telefon + Faxweiche	15.990,-
Tintenstrahl-Fax UF-311 Normalpapier, mit Weiche	20.970,-

Drucker

Canon BJ10ex Bubble Jet Tintenstrahldrucker	3.870,-
Einzelblatteinzug für BJ10ex	1.230,-
HP DeskJet 500 Tintenstrahldrucker von HP	6.480,-
HP DeskJet 550C neuer Farb-Tintenstrahldrucker von HP	9.480,-
HP LaserJet IIP+ 4 Seit/Min., 512KB Ram	11.970,-
HP LaserJet IIIP 4 Seit/Min., 1MB Ram	15.990,-
HP LaserJet 4 neu: RISC-Prozessor, 600dpi, 2MB Ram	22.980,-

Speichererweiterung für HP LaserJet Drucker lagernd!

CITEL Überspannungsschutz	2.940,-
Auto-Switch - 2 PC - 1 Drucker/1Drucker - 2PC's	760,-
Auto-Switch - 4 PC - 1 Drucker/1 Drucker - 4 PC's	1.200,-
Printer-Netz Sende-Modul am PC (3 Drucker wählbar)	870,-
Printer-Netz Empfangs-Modul je (max. 3) Drucker	990,-

CTF-Computer
 1180 Wien, Schulgasse 63
 Tel. 4085255 Fax 4089978

Preisliste gültig ab 1.1.1993
 Preise incl. MWST

Änderungen und Druckfehler vorbehalten

Komplettset's inkl. Monitor und Maus

386 SX/33 MHz, Desktopgehäuse,
 2 MB Speicher, aufrüstbar bis 32 MB,
 3 1/2 Zoll Floppylaufwerk, VGA Karte mit
 1024x768 Auflösung, 105 MB Harddisk
 IDE-AT-Buscontroller 2 ser. 1 par.
 Schnittstelle - 1 Gameport, Cache 128k
 erweiterte Tastatur Deutsch 12990,00

386 DX/40 MHz

Konfiguration wie 386 SX/33
 jedoch 4 Mb 15990,00

486 DX/33 MHz, Cache 256
 mit Intel Prozessor

Konfiguration wie 386 SX/33
 jedoch 4 Mb 21990,00

486 DX/50 MHz, Cache 256k
 mit Intel Prozessor

Konfiguration wie 386 SX/33
 jedoch 4 Mb 24990,00

486 DX2/ intern 66 MHz, Cache 256k
 mit Intel Overdrive Prozessor

Konfiguration wie 386 SX/33
 jedoch 4 Mb 26990,00

Aufpreise für Komplettset's

Minitower mit 3 Einschüben 330,00

Bigtower mit 6 Einschüben 900,00

Servertower, 2 Lüfter, Räder 2000,00

170 MB AT-Bus Festplatte 1500,00

210 MB AT-Bus Festplatte 2500,00

5 1/4 Zoll Floppylaufwerk 800,00

VGA 16 Mill. Farben 900,00

Windows Beschl. VGA Karte Diamond 24X 2500,00

Cherry Tastatur 1190,00

Speichererweiterung 1 MB*9 500,00

Speichererweiterung 4 MB*9 2050,00

"VGA Monitor 14" 72 Hz MNP2" 5990,00

"VGA Monitor 17" MNP2" Panasonic 13990,00

Math-CO-Prozessoren

287 XL bis 20 MHz

387 SX von 16 bis 25 MHz 1290,00

387 DX von 16 bis 33 MHz a. A

387 DX 40 MHz 1490,00

VGA Karten

ET 4000, 256 Farben 1290,00

ET 4000 Hicolor 1390,00

ET 4000 16 Mill. Farben 1990,00

Trident 512k, 800x600x256 790,00

Trident 1Mb, 1024x768x256 1190,00

OAK 512k, 800x600x256 790,00

OAK 1Mb, 1024x768x256 1290,00

Herculer Mono Graphic a.A

Diamond Speedstar 24x ,

Windows-Beschleunigerkarte 3690,00

Disketten-Laufwerke

"1,2MB 5 1/4" Floppy" 890,00

"1,44MB 3 1/2" Floppy" 850,00

1,2MB + 1,44MB Doppelfloppy 2190,00

Einbaurahmen für 3 1/2 auf 5 1/4 100,00

FD-HD Controller

I/O Karte für 2 HD, 2 FD,

2 ser., 1 par., 1 Gameport 490,00

Multi I/O - Karte für

2 ser. + 1 par. Schnittstelle 290,00

AT-Bus Cache-Controller bis 16MB

bestückbar, 4 HD, 2 FD,

2 ser. + 1 par. Schnittstelle 2490,00

Festplatten AT-Bus

80 MB Seagate 16ms a.A

105 MB Seagate 15ms 3490,00

120 MB Seagate 15ms a.A

130 MB Seagate 15ms 3990,00

170 MB Conner 14ms 4490,00

210 MB Conner 16ms 6390,00

426 MB Seagate 13900,00

SCSI Festplatten

240 MB Seagate 14ms a.A

426 MB Seagate 15ms 13990,00

520 MB Fujitsu 11ms 16990,00

Wegen der momentan starken Preisschwankungen sind größere Speicherkapazitäten auf Anfrage !
 Sämtliche Geräte werden auch gerne auf Ihre individuellen Wünsche konfiguriert

Neu !

Service -u. Reparatur Center

schnell - zuverlässig - kostengünstig

Wir reparieren ab sofort in unserer eigenen Werkstatt

***- Monitore**

(z.b. Samtron, AOC, Targ, Datas, ADI, Nec, u.a...)

***- Personalcomputer**

(IBM-kompatibel)

***- Schaltnetzteile**

***- Peripherie**

(z.b. Laser- u. Nadeldrucker, UPS, usw.)



Preise für Reparaturen bzw.Service:

- | | |
|--|----------|
| *- Reparaturpauschale für MONITORE inkl. Ersatzteile
(ausgenommen Zeilentrafo, Bildröhre und Gehäuse) | ÖS 960,- |
| *- Reparaturpauschale für PC-Netzteile inkl. Ersatzteile | ÖS 420,- |
| *- Reparaturkosten für Personalcomputer und Peripherie
Peripherie excl. Ersatzteile per Stunde | ÖS 840,- |

Zahlung: Bar/Scheck bei Warenübernahme, keine weiteren Rabatte auf Dienstleistung möglich !
Preise inkl.20% MWSt, excl.Abholung und Zustellung der zu reparierenden Geräte.

excon Warenvertriebsges.mBH,
TEL: (0222) 310 99 74-0

Röbergasse 6-8, A-1090 Wien
FAX: (0222) 310 99 74-14

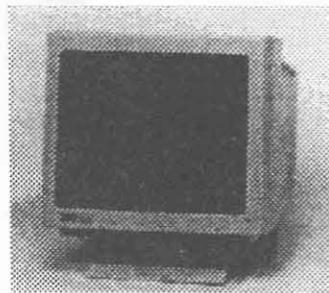
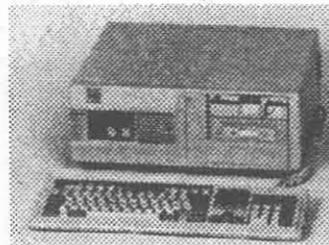
Sonderangebot für Mitglieder des PCC-TGM

(Gültig bis 26.02.93, solange der Vorrat reicht!)

Personalcomputer 386DX/40Mhz 128k Cache, 4MB RAM, 105MB/16ms, DOS 5.0 + 14" Super VGA Monitor

AT-Tisch Gehäuse + 200W Netzteil, 4 MB RAM, erweiterbar auf 32MB, 1.44MB/3½" Floppy (Teac) 105MB/16ms Festplatte (Seagate), 16Bit VGA Karte max 1024*768, 512k (Paradise kompatibel), 2 Seriell, 1 Parallel, Game Port, Tastatur deutsch (Datacomp) MS-DOS 5.0, 14" VGA Monitor (max 1024*768 interlaced, 0.28mm)

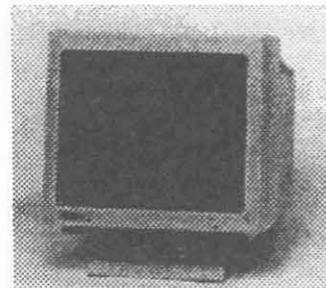
ÖS 16.980,-



Personalcomputer 486DX/33Mhz 256k Cache, 4MB RAM, 170MB/16ms, DOS 5.0 + 14" VGA Monitor non-interlaced

Baby Tower Gehäuse+ 200W Netzteil, 4 MB RAM, erweiterbar auf 32MB, 1.44MB/3½" Floppy (Teac) 170MB/16ms Festplatte (Quantum), VGA Karte max 1280*1024 / 1MB original Tsenglabs ET4000, 2 Seriell, 1 Parallel, 1 Game Port, Tastatur deutsch (Datacomp), MS-DOS 5.0, 14" VGA Monitor max. 1024*768/70Hz non-interlaced (AOC CMLB-337) strahlungsarm nach MPR II

ÖS 25.980,-



Zahlung: Bar/Scheck bei Warenübernahme!, gültig solange der Vorrat reicht
Lieferung: ab Lager Wien, Preise inkl. 20% MWSt, Garantie: 12 Monate.

excon Warenvertriebsges.mBH,
TEL: (0222) 310 99 74-0

Röergasse 6-8, A-1090 Wien
FAX: (0222) 310 99 74-14

Sonderpreisliste excon für PCC-TGM

Zahlungskonditionen:	Bei Warenübernahme Bar oder Scheck
Preise:	1/93, in öS incl. 20% MWSt.
Lieferung:	ab Lager Wien, so lange der Vorrat reicht
Mindestbestellwert:	ÖS 1.000,-
Garantie:	12 Monate auf Komplett-Geräte 6 Monate auf Einzel- und Ersatzteile und Monitore
Tel.:	0222/310-99-74-0
Fax.:	0222/310-99-74-14
Anschrift:	EXCON Warenvertriebs GesmbH, Rögergasse 6-8, 1090 Wien

Mit dieser Preisliste sind alle vorangegangenen Preislisten ebenso ungültig, wie eventuell in Zusammenhang mit diesen Listen gemachte Sonderkonditionen. Irrtümer und Änderungen jederzeit vorbehalten. Im übrigen gelten die allgemeinen Geschäftsbedingungen der Elektroindustrie Österreichs. **Sonderangebote siehe voriges Blatt.**

ALLE COMPUTER WERDEN SPEZIELL NACH IHREN WÜNSCHEN KONFIGURIERT!!!

Reklamationsabwicklung: Das Einsenden defekter Ware hat frei Haus zu erfolgen. Retouren zur Gutschrift, sowie Leihstellungen werden nur in Originalverpackung mit Manual (Disketten, Stecker, Kabel etc. sofern im Lieferumfang enthalten) zurückgenommen. Eine Rechnungskopie, sowie eine detaillierte Fehlerbeschreibung ist der defekten Ware beizulegen. Überprüft werden nur die in Ihrer Fehlerbeschreibung angegebenen Mängel. Liegt kein Fehler vor, wird der entstandene Arbeitsaufwand verrechnet.

Beliebige Sonder-Konfigurationen auf Anfrage !!!!
Alle Computer Systeme inkl. MS-DOS 5.0
(auf Wunsch ohne Betriebssystem abzüglich ÖS 900,-)

Personalcomputer 386SX

A386SX25	386SX LowCost /25Mhz BASIS-SYSTEM.....	13.644,00
	AT-Tischgehäuse mit LED Speed Anzeige + 200 W Netzteil, CPU 80386SX-25, HEADLAND-CHIP-SET, AMI-BIOS mit Passwort, 4MB RAM, erweiterbar auf 8/10/16/32MB mit 1MB/4MB SIMM, 3x5¼" und 1x3½" Slimline Einbauplätze für Floppy und Harddisks, 1.44MB/3½" oder 1.2MB/5¼" Floppy Disk Laufwerk (TEAC), 106MB/16ms Festplattenlaufwerk IDE-AT-BUS (Seagate ST3120A), FD/HDD Controller IDE-AT-BUS, Interleave 1:1, 2 seriell, 1 parallel, 1 Game Port, 16Bit Super VGA Karte 1024x768/512kB (Paradise kompatibel), erweiterte Tastatur - 102 Tasten deutsch oder US MF2-kompatibel	
A386DX40	386DX DeLuxe /40Mhz /128kB CACHE BASIS-SYSTEM.....	14.964,00
	CPU 80386DX-40, OPTI CHIP-SET, AMI-BIOS mit Passwort, Alle anderen Daten wie A386SX25 (386SX/25)	

AUFPREISE für 386SX, 386 BASIS-SYSTEME

Festplatte AT-BUS 127MB/17ms (QUANTUM).....	504,00
Festplatte AT-BUS 125MB/15ms (SEAGATE).....	624,00
Festplatte AT-BUS 170MB/16ms (QUANTUM).....	1.440,00
Festplatte AT-BUS 240MB/15ms (QUANTUM).....	3.504,00
Festplatte AT-BUS 240MB/12ms (SEAGATE).....	3.504,00
Festplatte AT-BUS 425MB/15ms (QUANTUM).....	9.864,00
Festplatte AT-BUS 540MB/15ms (CONNER).....	13.224,00
2. Floppy Disk Laufwerk 1.2MB/5¼" oder 1.44MB/3½".....	864,00
VGA 1024x768/1MB Palm Windows Accelerator - 16Mio - True Color.....	456,00
VGA 1024x768/1MB TSENG ET-4000 original.....	792,00
VGA 1024x768/1MB TSENG ET-4000 original - 16Mio. True-Color.....	1.128,00
VGA 1024x768/1MB-VRAM S3 Windows Accelerator - 32k Color.....	1.716,00
Graphikkarten speziell für CAD auf Anfrage	
Speichereverweiterung 4MB -> 8MB.....	2.040,00
Speichereverweiterung 4MB -> 16MB.....	6.120,00
Speichereverweiterung 4MB -> 32MB.....	14.280,00
Aufpreis SLIM-LINE Gehäuse.....	0,00
Aufpreis BABY-TOWER Gehäuse.....	480,00

Aufpreis BIG-TOWER Gehäuse.....	960,00
Alle Computer Systeme inkl. MS-DOS 5.0 (auf Wunsch ohne Betriebssystem abzüglich ÖS 900,-)	

Personalcomputer 486

A486DX33	486DX DeLuxe /33Mhz /256kB CACHE BASIS-SYSTEM.....	20.580,00
	CPU 80486DX-33, OPT CHIP SET, alle anderen Daten wie A386SX25 (386SX/25)	
A486D250	486DX2 DeLuxe /50Mhz /256kB CACHE BASIS-SYSTEM.....	22.284,00
	CPU 80486DX2-50, OPT CHIP SET, alle anderen Daten wie A386SX25 (386SX/25)	
A486DX50	486DX DeLuxe /50Mhz /256kB CACHE BASIS-SYSTEM.....	24.084,00
	CPU 80486DX-50, OPTI CHIP SET, alle anderen Daten wie A386SX25 (386SX/25)	
A486D266	486DX2 DeLuxe /66Mhz /256kB CACHE BASIS-SYSTEM.....	24.564,00
	CPU 80486DX2-66, OPTI CHIP SET, alle anderen Daten wie A386SX25 (386SX/25)	

AUFPREISE für 486 BASIS-SYSTEME

Festplatte AT-BUS 127MB/17ms (QUANTUM).....	504,00
Festplatte AT-BUS 125MB/15ms (SEAGATE).....	624,00
Festplatte AT-BUS 170MB/16ms (QUANTUM).....	1.440,00
Festplatte AT-BUS 240MB/15ms (QUANTUM).....	3.504,00
Festplatte AT-BUS 240MB/12ms (SEAGATE).....	3.504,00
Festplatte AT-BUS 425MB/15ms (QUANTUM).....	9.864,00
Festplatte AT-BUS 540MB/15ms (CONNER).....	13.224,00
SCSI Festplatten auf Anfrage	
2. Floppy Disk Laufwerk 1.2MB/5¼" oder 1.44MB/3½".....	864,00
VGA 1024x768/1MB Palm Windows Accelerator - 16Mio - True Color.....	456,00
VGA 1024x768/1MB TSENG ET-4000 original.....	792,00
VGA 1024x768/1MB TSENG ET-4000 original - 16Mio. True-Color.....	1.128,00
VGA 1024x768/1MB-VRAM S3 Windows Accelerator - 32k Color.....	1.716,00
Graphikkarten speziell für CAD auf Anfrage	
Speichereverweiterung 4MB -> 8MB.....	2.040,00
Speichereverweiterung 4MB -> 16MB.....	6.120,00

Personalcomputer 486 EISA-BUS

E486DX33	EISA 486DX /33Mhz /256k CACHE BASIS-SYSTEM ..29.520,00 AT-Tischgehäuse mit LED Speed Anzeige + 200 W Netzteil, CPU 80486DX-33Mhz, OPTI-CHIP-SET, AMI-BIOS mit Passwort, 4MB RAM, erweiterbar auf 128MB mit 256/1MB/4MB/16MB SIM, 3x5¼" und 1x3½" Slimline Einbauplätze für Floppy und Harddisks, 1.44MB/3½" oder 1.2MB/5¼" Floppy Disk Laufwerk (TEAC), 170MB/16ms Festplattenlaufwerk IDE-AT-BUS (QUANTUM), EISA- CACHE FD/HDD Controller IDE-AT-BUS (DC620), BUSMASTER, 2MB, CACHE, erweiterbar auf 24MB, für alle Betriebssysteme ohne zusätzl. Treiber, 16Bit Super VGA Karte 1024x768/512kB (Paradise kompatibel), erweiterte Tastatur - 102 Tasten deutsch oder US MF2-kompatibel
E486D250	EISA 486DX2 /50Mhz /256k CACHE BASIS-SYSTEM.31.176,00 CPU 80486DX2-50, alle anderen Daten wie E486DX33 (486/33 EISA)
E486DX50	EISA 486DX/50Mhz /256k CACHE BASIS-SYSTEM ...32.880,00 CPU 80486DX-50, alle anderen Daten wie E486DX33 (486/33 EISA)
E486D266	EISA 486DX2/66Mhz /256k CACHE BASIS-SYSTEM..33.360,00 CPU 80486DX2-66, alle anderen Daten wie E486DX33 (486/33 EISA)

AUFPREISE für 486 EISA BASIS-SYSTEME

Festplatte AT-BUS 240MB/15ms (QUANTUM)	2.064,00
Festplatte AT-BUS 240MB/12ms (SEAGATE)	2.064,00
Festplatte AT-BUS 425MB/15ms (QUANTUM)	8.424,00
Festplatte AT-BUS 540MB/15ms (CONNER)	11.784,00
SCSI Festplatten + SCSI EISA CACHE Controller auf Anfrage	
2. Floppy Disk Laufwerk 1.2MB/5¼" oder 1.44MB/3½"	864,00
VGA 1024x768/1MB Palm Windows Acceleratur - 16Mio - True Color	456,00
VGA 1024x768/1MB TSENG ET-4000 original	792,00
VGA 1024x768/1MB TSENG ET-4000 original - 16Mio. True-Color	1.128,00
VGA 1024x768/1MB-VRAM S3 Windows Accelerator - 32k Color	1.716,00
EISA VGA 1024x768/1MB-VRAM S3 Windows Accelerator - 32k Color	2.676,00
EISA VGA 1280x1024/1MB-VRAM ELSA Winner 1000 Accelerator - 16Mio Color	5.256,00
EISA Graphikkarten speziell für CAD auf Anfrage	
Speichererweiterung 4MB -> 8MB	2.040,00
Speichererweiterung 4MB -> 16MB	6.120,00
Speichererweiterung 4MB -> 32MB	14.280,00
Aufpreis SLIM-LINE Gehäuse	0,00
Aufpreis BABY-TOWER Gehäuse	480,00
Aufpreis BIG-TOWER Gehäuse	960,00

NOTEBOOK - PERSONALCOMPUTER

NB560623	CHICONY NB-5600/25Mhz/60MB	23988,00
	CPU 80386SX-25Mhz, Sockel für 387SX Coproz., 1MB RAM, erweiterbar auf 5MB, 1.44MB/3½" Diskettenlaufwerk, 60MB/19ms Festplatte, VGA Karte, LCD-Display, (640x480) 32 Graustufen, 1 Seriell, 1 Parallel Interface, Anschluß für externes Keyboard und VGA Monitor, Anschluß für externes Floppy 1.2MB/5¼", Keyboard mit 82 Tasten, externer Zahlenblock optional, AKKU für ca. 3 Stunden Betrieb, Ladegerät, Tragtasche	
NB560823	CHICONY NB-5600/25Mhz/80MB	25800,00
NB561223	CHICONY NB-5600/25Mhz/120MB	27000,00
NB2RAM23	Speichererweiterung auf 2MB für Chicony NB5600	1380,00
NB5RAM23	Speichererweiterung auf 5MB für Chicony NB5600	4740,00
DC312033	CHICONY DC-3000/386SXL-25Mhz/120MB COLOR- NOTEBOOK	43920,00
	CPU 80386SX-25Mhz, 4MB RAM, erweiterbar auf 8MB, 1.44MB/3½" Floppy Laufwerk, 120MB/23ms Festplatte, VGA Karte, LCD-Color Display 640x480, 1 Ser., 1 Parallel IF, Sockel f.387SX, Anschluß f. ext. Keyboard und VGA Monitor, Tastatur mit 82 Tasten, externer Zahlenblock optional, AKKU f.ca.3 Stunden Betrieb, Ladegerät u. Tasche	
DC412033	CHICONY DC-4000/486DX2-50Mhz/120MB COLOR-NOTEBOOK	60960,00
	CPU 80486DX-33Mhz, 5MB RAM, erweiterbar auf 16MB, sonstige Daten wie D312033 (Chicony DC-3000)	

DC3RAM40	4MB Memory Card für DC-3000, DC-4000	3108,00
NB5FDD23	FDD Kit für externes 5¼" Laufwerk (ohne Floppy)	1404,00
NB5AUT23	Autoadapter 12V-Stromversorgung für Chicony Notebook	2028,00
NB5FAX23	FAX-MODEM Adapter (nur für den EXPORT)	3960,00
	9600 Baud FAX, 2400 Baud Modem, nicht postgenehmigt II	
ETHPRO26	Pocket Ethernet LAN-Adapter	2940,00
	zum Anschluß an die parallele Schnittstelle	
HBDLC025	SunRace HyperBook 2300DLC/486-25Mhz, 120MB	29880,00
	CPU 486DLCX-25Mhz, Sockel für 387 Coproz., 4MB RAM, erweiterbar auf 8MB, 1.44MB/3½" Diskettenlaufwerk, 120MB/23ms Festplatte, VGA Karte, LCD-Display, (640x480) 64 Graustufen, 1 Seriell, 1 Parallel Interface, Anschluß für PS/2 Mouse, SCSI Interface (25polig), Anschluß für externes Keyboard und VGA Monitor, Tastatur mit 80Tasten, AKKU für ca.3 Stunden Betrieb, Ladegerät und Tragtasche	
HBDLC033	SunRace Hyper Book 2300DLC/486-33Mhz, 120MB	41880,00
	CPU 486DLC-33Mhz, 128k Cache, 4MB RAM, erweiterbar auf 8MB, Sockel f.80387, sonstige Daten wie HBDLC025 (Hyperbook 2300DLC/486-25)	
HBDX2050	SunRace Hyper Book 2300DX2/486-50Mhz, 120MB	53880,00
	CPU 80486DX2-50Mhz, 256k Cache, 4MB RAM, erweiterbar auf 8MB, sonstige Daten wie HBDLC025 (Hyperbook 2300DLC/486-25)	
HB4MBDXE	4MB RAM-Erweiterung	4320,00

MONOCHROM - MONITORE

4001MW24	14" Monochrom - Monitor (Hercules)	1740,00
4300MV24	9" Monochrom VGA Monitor weiss (CARRY)	2520,00
4301MV24	14" Monochrom VGA Monitor 640x480 weiss (AOC)	1740,00

COLOR - MONITORE

4305MM24	14" VGA M-55	3828,00
	1024x768 interlaced, 0.28mm, 15.5-38kHz	
4305AO24	14" VGA AOC CMLB-335 strahlungsarm	4308,00
	1024x768 interlaced, 0.28mm, 15.5-38kHz	
4306MV24	14" VGA TARGA TM1496 strahlungsarm	5100,00
	1024x768 interlaced, 0.28mm, 15.5-38kHz	
4306AO24	14" VGA AOC CMLB-337 strahlungsarm	5340,00
	1024x768/70Hz non-interlaced, 0.28mm, 30-60kHz	
4307AO24	15" VGA AOC CMLB-536 strahlungsarm	6900,00
	1280x1024 non-interlaced/60Hz, 0.28mm, 31,5-65kHz	
4303EZ24	15" EIZO Flexscan F340i strahlungsarm	15588,00
	1024x768 non-interlaced/75Hz, 0.28mm, 27-61kHz	
4303FG24	15" NEC-Multisync 3FG strahlungsarm	12780,00
	1024x768 interlaced, 0.28mm, 15.5-38kHz	
4304FG24	15" NEC-Multisync 4FG strahlungsarm	16380,00
	1024x768/70Hz non-interl. 0.28mm, 27-57kHz	
4517AO24	17" AOC-Multisync CMLB-735 strahlungsarm	12540,00
	max 1280x1024 non-interlaced, 1024x768/70Hz, 0.26mm, 30-64kHz	
4518AO24	17" AOC-Multisync CMLB-736 strahlungsarm	13440,00
	max 1280x1024/70Hz non-interlaced, 0.26mm, 30-76kHz	
4522EZ24	17" EIZO-FLEXSCAN F550 i strahlungsarm	21960,00
	1024x768/70Hz non-interl. 0.28mm, 30-65kHz, digitale Steuerung	
4527MM24	17" EIZO-FLEXSCAN T560 i strahlungsarm	33540,00
	1280x1024/70Hz non-interl.0.26mm,30-78kHz, TRINITRON, digitale Steuerung	
4527SO24	17" SONY Multisync 1704S strahlungsarm	22680,00
	1280x1024/70Hz non-interl.0.26mm,30-57kHz, TRINITRON	
4519SO24	19" SONY Multisync GDM1934 strahlungsarm	52680,00
	1280x1024/75Hz non-interlaced, 0.25mm, 30-80kHz, 50- 120Hz, TRINITRON	
4512EZ24	21" EIZO-FLEXSCAN F750 i strahlungsarm	41976,00
	1280x1024/60Hz non-interlaced 30-65kHz, digitale Steuerung	
4513EZ24	20" EIZO-FLEXSCAN T660 i strahlungsarm	52680,00
	1280x1024/75Hz non-interlaced 30-80kHz, TRINITRON, digitale Steuerung	

DTP - MONITORE

4518DT24	15" SIGMA PAGEVIEW A4 Monitor, 1024x768, inkl. Graphikkarte.....	15120,00
4519DT24	19" SIGMA LASERVIEW A3 Monitor, 1664x1200, inkl. Graphikkarte.....	29400,00
4520DT24	19" SIGMA LASERVIEW A3 Monitor, wie 4519DT24, 4 Graustufen.....	30600,00

MATRIX - DRUCKER

5006P024	Citizen LSP 120-D+, 9 Nadel.....	2628,00
5000P024	Citizen Swift 200, 24 Nadel/A4, 360x360dpi, 216cps.....	4200,00
5002P024	Citizen Swift 240C, 24 Nadel/A4, 360x360dpi, 240cps, COLOR.....	5640,00
5010P024	Citizen Swift 24X, 24 Nadel/A3 360x360dpi.....	6960,00
5010CL24	Sheetfeeder für Citizen Swift A4.....	1500,00
5014CL24	Sheetfeeder für Citizen Swift A3.....	2940,00
5009IF24	Seriell Interface f. Citizen 120D+.....	1170,00
5010FB24	Farbband für Swift 24 A4 und 120D+.....	90,00
5011FB24	Farbband für Swift 24 A3.....	192,00

TINTENSTRAHL - DRUCKER

BJ10E024	CANON Bubble Jet BJ10ex.....	4440,00
BJ10BA24	Batterie für BJ10ex (für ca. 30 Blatt).....	1020,00
5015CL24	Sheetfeeder für Canon BJ10ex (f.30 Blatt).....	1176,00
BJ10DR24	Druckkopf - Tintenpatrone für BJ10ex.....	432,00
BJ200E24	CANON Bubble Jet BJ200 /A4, automat. Einzelblatteinzug.....	6984,00
BJ300024	CANON Bubble Jet BJ300 /A4.....	8160,00
BJ30EZ24	Automatischer Einzelblatteinzug für BJ300 / 1.Schacht.....	1458,00
BJ31EZ24	Automatischer Einzelblatteinzug für BJ300 / 2.Schacht.....	972,00
BJ330024	CANON Bubble Jet BJ330 /A3.....	9960,00
BJ33EZ24	Automatischer Einzelblatteinzug für BJ330 / 1.Schacht.....	1800,00
BJ34EZ24	Automatischer Einzelblatteinzug für BJ330 / 2.Schacht.....	1260,00
BJC80024	CANON Bubble Jet BJC800 /A4 COLOR.....	31080,00
BJ30DR24	Tintenpatrone für BJ300/330.....	312,00
BJC8TPSW	Tintenpatrone schwarz für BJC800.....	312,00
BJC8TPCO	Tintenpatrone color für BJC800.....	468,00
5024P024	HP DeskJet 500 A4/aut. Einzelblatteinzug.....	7080,00
5024PC24	HP DeskJet 500 COLOR A4/aut. Einzelblatteinzug.....	9360,00
5025PC24	HP DeskJet 550 COLOR A4/aut. Einzelblatteinzug.....	11712,00
5024TP024	Tintenpatrone für HP DeskJet 500, 550 schwarz.....	348,00
5025TP024	Tintenpatrone für HP DeskJet 500C, 550C color.....	504,00

LASER - DRUCKER

CLBP4024	CANON Laserdrucker LBP 4+ /512kB.....	13788,00
CLBP4P24	CANON Laserdrucker LBP 4+ /1,5MB.....	15108,00
CLBP8024	CANON Laserdrucker LBP 8 III+ /1,5MB.....	23880,00
CLBP8R24	CANON Laserdrucker LBP 8 IIIR /1,5MB/Duplex.....	41760,00
CLBP4124	1 MB Zusatzspeicher von 512k auf 1.5MB für LBP 4+.....	2580,00
CLBP4424	1MB Zusatzspeicher von 1.5MB auf 2.5MB für LBP 4+.....	2580,00
CLBP4824	1MB Zusatzspeicher für CANON LBP 8.....	2880,00
CLPS4024	POSTSCRIPT Erweiterung für CANON.....	10080,00
CLBP4TON	TONER für CANON CLBP4.....	1428,00
HPLJ3TON	TONER für HPLJ III, IIIP, CANON CLBL8.....	1296,00
HPLJ4TON	TONER für HPLJ 4,4M.....	1968,00
5013HP24	HP Laserjet II P /512kB.....	14208,00
5013P024	HP Laserjet III P /1MB.....	16560,00
5016P024	HP Laserjet III D /1MB.....	47880,00
HPLJ3S10	HP Laserjet III Si /2MB.....	62880,00
HPLJ4024	HP Laserjet 4 /600dpi /2MB.....	26820,00
HPLJ4M24	HP Laserjet 4M /600dpi /6MB /Postscript /Apple Talk.....	35760,00
5020P024	ORIGINAL ADOBE POSTSCRIPT plus CARTRIDGE.....	9588,00
5020PS24	PACIFIC POSTSCRIPT CARTRIDGE, abschaltbar.....	7440,00
5021PI24	1MB Speichererweiterung f.HPLJ IIP,III,IIIP,IIID (INTEL)..... erweiterbar auf 4MB (8x414256 je 1MB)	1380,00
HPRAM1MB	1MB Speichererweiterung f.HPLJ IISi, 4.....	1890,00
HPRAM2MB	2MB Speichererweiterung f.HPLJ IISi, 4.....	2376,00
HPRAM4MB	4MB Speichererweiterung f.HPLJ IISi, 4.....	5700,00

Weitere Drucker sowie Farbbänder auf Anfrage!

GEHÄUSE - STROMVERSORGUNG

3202C027	AT-GEHÄUSE + 200W Netzteil (TÜV).....	1620,00
	3x5¼", 1x3½" Slim Einschubplätze, LED Display	
3204C027	BABY-TOWER GEHÄUSE + 200W Netzteil (TÜV).....	2100,00
	4x5¼", 1x3½" Slim Einschubplätze, LED Display	
3205C027	BIG-TOWER GEHÄUSE + 220W Netzteil (TÜV).....	2580,00
	6 Slim Einschubplätze 5¼", LED Display	
3206C027	SLIMLINE GEHÄUSE + 200W Netzteil (TÜV).....	1620,00
	3x16Bit,2x8Bit SLOT, 2x5¼", 1x3½" Slim Einbauplätze	
3201C027	SUPER SLIM LINE GEHÄUSE + 60W Netzteil.....	1788,00
	4x16Bit Slots, 1x3½" Slim Einbauplatz	
3203C0EC	FILE-SERVER GEHÄUSE + 300W Netzteil (TÜV).....	9000,00
	13 Slim Einschubplätze 5¼", LED Display	
1100S027	Thermo-Lüftersteuerung für PC-Netzteile.....	600,00
1200S027	200W Netzteil f. Baby-AT Gehäuse.....	1176,00
1201S027	200W Netzteil f. Baby-Tower, Slim Line Gehäuse.....	1176,00
1202S027	220W Netzteil f. Big-Tower.....	1320,00

MOTHERBOARDS ISA-BUS

202AM025	386SX/25Mhz MOTHERBOARD (2/3-Size).....	1800,00
	HEADLAND-CHIP-SET, 2/3 Size, AMI-BIOS mit Passwort, 6x16Bit, 2x8Bit Slot, SOCKEL für 80387 od. Weitek, 0kB RAM, erweiterbar wie folgt: 1/2 MB : 4/8*SIMM Modul 256k, 2/4/8MB: 2/4/8*SIMM Modul 1MB, 16/32MB: 4/8*SIMM Modul 4MB, auch gemischte Bestückung teilweise möglich	
201AM025	386/40Mhz/128k Cache MOTHERBOARD.....	3120,00
	OPTI-CHIP-SET,AMI-BIOS mit Passwort, 6x16Bit, 2x8Bit Slot, Sockel für 80387 od. Weitek, 0kB RAM, erweiterbar wie 190AM025 (386/25)	
198AM025	486/33Mhz/256k Cache MOTHERBOARD.....	8736,00
	OPTI-CHIP-SET,AMI-BIOS mit Passwort, 6x16Bit, 2x8Bit Slot, Sockel für Weitek 4167, 0kB RAM, erweiterbar wie 190AM025 (386/25)	
199AM025	486/DX2-50Mhz/256k Cache MOTHERBOARD.....	10440,00
	OPTI-CHIP-SET,AMI-BIOS mit Passwort, 6x16Bit, 2x8Bit Slot, Sockel für Weitek 4167, 0kB RAM, erweiterbar wie 190AM025 (386/25)	

MOTHERBOARDS ISA-BUS

203AM025	486/DX-50Mhz/256k Cache MOTHERBOARD.....	12240,00
	OPTI-CHIP-SET,AMI-BIOS mit Passwort, 6x16Bit, 2x8Bit Slot, Sockel für Weitek 4167, 0kB RAM, erweiterbar wie 190AM025 (386/25)	
200AM025	486/DX2-66Mhz/256k Cache MOTHERBOARD.....	12720,00
	OPTI-CHIP-SET,AMI-BIOS mit Passwort, 6x16Bit, 2x8Bit Slot, Sockel für Weitek 4167, 0kB RAM, erweiterbar wie 190AM025 (386/25)	

MOTHERBOARDS EISA-BUS

198EM025	486/DX-33Mhz/256k Cache MOTHERBOARD EISA.....	12240,00
	OPTI-CHIP-SET, AMI BIOS mit Passwort, 8x32 Bit EISA Slot, Sockel, für Weitek 4167, 0kB RAM, erweiterbar wie folgt: 1/2MB: 4/8*SIMM Modul 256k, 4/8MB: 4/8*SIMM 1MB, 16/32MB: 4/8*SIMM Modul 4MB, 64/128MB: 4/8* SIMM Modul 16MB	
203EM025	486/DX2-50Mhz/256k Cache MOTHERBOARD EISA.....	13896,00
	sonst wie 198EM025 (486DX-33Mhz)	
204EM025	486/DX-50Mhz/256k Cache MOTHERBOARD EISA.....	15600,00
	sonst wie 198EM025 (486DX-33Mhz)	
205EM025	486/DX2-66Mhz/256k Cache MOTHERBOARD EISA.....	16080,00
	sonst wie 198EM025 (486DX-33Mhz)	

RAM

DR414256	Dyn.RAM 414256-07 (256kx4).....	72,00
DR411000	Dyn.RAM 411000-07 (1024kx1).....	72,00
SR032KX8	SRAM 32kx8, 20ns Cache RAM.....	144,00
SIM25608	SIMM MODULE 70nS (256kx9).....	240,00
SIM1MB08	SIMM MODULE 70nS (1024kx9).....	510,00
SIM4MB08	SIMM MODULE 70ns (4096kx9).....	2040,00
SIP1MB08	SIP MODULE 70nS (1024kx9).....	570,00

FLOPPY/HARDDISK - CONTROLLER

436F/H26	SUPER I/O Controller (FD/IDE-HDC,2Ser/1PAr/Game)	300,00
450F0026	Floppy-Disk-Controller CI-1288 (4*FDC 720k/1.2/1.4/2.8MB)	828,00
431F/H26	MFM Floppy-Harddisk Controller	948,00
448F/H26	IDE-AT-BUS Controller 4-fach (CI-1010)	1140,00
	2*intern/2*externe IDE-AT-Bus Harddisk, mit BIOS, kann zusätzlich zu bereits vorhandenen HD Controllern eingebaut werden, nur für MS-DOS geeignet !!	
449F/H26	IDE-AT-BUS + SCSI FD/HD Contr (CI-2000)	960,00
	2*Floppy, 2*IDE-AT-BUS Harddisk, SCSI Controller (Future Domain komp) + Kabel	
435F/H26	ADAPTEC 1542B SCSI Floppy-Harddisk Controller	3660,00
433F/H26	ADAPTEC-1542B KIT	4320,00
	incl. Treibersoftware für OS/2 und NOVELL, CD-ROM und ASPI für DOS	
439F/H26	DC-226 Intelligent SCSI Floppy/Harddisk Controller	2280,00
	inkl. ASPI-Treiber für DOS, für alle Betriebssysteme ohne zusätzl. Treiber geeignet	
443F/H26	Parallel/SCSI Host Adapter (SCSI)	2700,00
	Adapter zum Anschluß an die parallele Schnittstelle, inkl. Treiber Software	

CACHE FLOPPY/HARDDISK CONTROLLER

447F/H26	DC-600 IDE CACHE Controller (256kb-16MB)	1968,00
	2x Floppy/4x IDE-AT-BUS Harddisk, BIOS-Setup mit HOTFIX, 0kB RAM, erweiterbar auf 1/2/3/4/8/12/16MB mit 256k/1MB/4MB SIMM, für alle Betriebssysteme ohne zusätzl. Treiber geeignet, ca. 1350kB/sec, 0.2ms Zugriffszeit	

CACHE FLOPPY/HARDDISK CONTROLLER EISA

447FEH26	DC-620 EISA IDE CACHE Controller (512kB-24MB)	2976,00
	2x Floppy/4x IDE-AT-BUS Harddisk, BIOS-Setup mit HOTFIX, 0kB RAM, erweiterbar auf 1/2/3/4/8/12/16/24MB mit 256k/1MB/4MB SIMM, für alle Betriebssysteme ohne zusätzl. Treiber geeignet, ca. 1350kB/sec, 0.2ms Zugriffszeit	
446F/H26	DC-820 EISA SCSI CACHE Controller (1MB-16MB)	6720,00
	2x Floppy / SCSI Harddisk Anschluß, 0kB RAM, erweiterbar auf 1/4/16MB Cache, Adaptec 154x kompatibel, Treiber für DOS, Novell 3.1X, SCO u. Interactive UNIX	

SCHNITTSTELLEN - KARTEN

630C0026	Multi I/O (2 Seriell/1 Parallel/Game)	300,00
360D0026	Parallel Printer Karte	180,00
361D0026	Parallel Printer Karte 2-Port	264,00

GRAPHIK - KARTEN

310D0026	Mono/Graphic/Printer - Karte (Hercules)	240,00
337WDB26	16Bit VGA PARADISE (1024x768/512kB)	864,00
	VESA STANDARD, erweiterbar auf 1MB	
338CIRRU	16Bit VGA PALM-WINDOWS-ACCELERATOR, 16Mio COLOR	1320,00
	Cirrus Logics AVGA3 GUI, max 1280x1024 interlaced, 1MB, VESA STANDARD	
335D/B26	16-BIT VGA MegaVGA1024/4 (ET4000)	1656,00
	ORIGINAL TSENGLABS, 1 MB ,max 1280x1024 interlaced, VESA STANDARD	
335D2B26	16-BIT VGA MegaVGA 1024/2 (ET4000/16Mio.Farben) ..	1992,00
	ORIGINAL TSENGLABS, 1 MB RAM,max 1280x1024 interlaced, VESA STANDARD (70/72Hz), 16Mio Color/640x480, 64k Color/800x600	
340D/B26	Programmers Reference Manual für Tseng ET4000	468,00
335S3026	16-BIT VGA S3 Windows Accelerator	2580,00
	VESA STANDARD, 1MB VRAM, max 1280x1024 interlaced, 32k Color/640x480	
335SS026	16-BIT VGA DIAMOND Stealth (S3) Windows Accelerator	4620,00
	VESA STANDARD, 1MB VRAM, max 1280x1024 interlaced, 32k Color/640x480	
336W1000	ELSA WINNER 1000 Windows Accelerator ISA/EISA	6120,00
	VESA STANDARD, 1MB VRAM, max 1280x1024/60Hz non-interlaced, 16Mio Color, erweiterbar auf 2MB VRAM,	

338PG126	PECAD PGA 8/1280; 2MB VRAM (TI34020)	26520,00
	VESA STANDARD, 2MB VRAM, max 1280x1024/80Hz non-interlaced, nur für 2 Monitor Betrieb	
338PG1VM	PECAD PGA 8/1280VM; 3MB VRAM+4MB Displaymem. (TI34020)	33924,00
	Anschluß an die VGA Karte über Feature Connector, sonst wie 8/1280	
338PG226	PECAD PGA 8/1600M, 2MB VRAM + 4MB Displaymem. (TI34020)	35940,00
	VESA STANDARD, 2MB VRAM + 4MB Display List Memory, max 1600x1280/80Hz, non-interlaced, 16Mio.Color, nur für 2 Monitor Betrieb	
338PG2VM	PECAD PGA 8/1600VM, 3MB VRAM + 4MB Displaymem (TI34020)	39000,00
	Anschluß an die VGA Karte über Feature Connector, sonst wie 8/1600M	

GRAPHIK - KARTEN EISA-BUS

335ES326	16-BIT VGA S3 Windows Accelerator	3540,00
	VESA STANDARD, 1MB VRAM, max 1280x1024 interlaced, 32k Color/640x480	
336W1000	ELSA WINNER 1000 Windows Accelerator ISA/EISA	6120,00
	VESA STANDARD, 1MB VRAM, max 1280x1024/60Hz non-interlaced, 16Mio Color, erweiterbar auf 2MB VRAM,	
338ES126	PECAD EISA 8/1280; 2MB VRAM (TI34020)	30840,00
	VESA STANDARD, 2MB VRAM, max 1280x1024/80Hz non-interlaced, nur für 2 Monitor Betrieb	
338ES1VM	PECAD EISA 8/1280VM; 3MB VRAM+4MB Displaymem. (TI34020)	38040,00
	Anschluß an die VGA Karte über Feature Connector, sonst wie 8/1280	
338ES226	PECAD EISA 8/1600M, 2MB VRAM + 4MB Displaymem. (TI34020)	39000,00
	VESA STANDARD, 2MB VRAM + 4MB Display List Memory, max 1600x1280/80Hz non-interlaced, 16Mio.Color, nur für 2 Monitor Betrieb	
338ES2VM	PECAD EISA 8/1600VM, 3MB VRAM +4MB Displaymem (TI34020)	40680,00
	Anschluß an die VGA Karte über Feature Connector, sonst wie 8/1600M	

DISKETTEN-LAUFWERKE

810F/J27	1,2MB/5/4" FLOPPY TEAC	864,00
812F/027	1,44MB/3/2" FLOPPY TEAC ohne Rahmen	732,00
814F/027	1,44MB/3/2" + 1.2MB/5/4" DOPPEL-FLOPPY	2496,00
820F/J027	5/4" Einbaakit für 3/2" Floppy	132,00

FESTPLATTEN IDE-AT-BUS

9040AT27	40 MB IDE/28 SEAGATE ST351AX, 3/2"	2640,00
9080AT27	80MB IDE/17ms SEAGATE ST3096A, 3/2"	3720,00
9106AT27	106MB IDE/15ms SEAGATE ST3120A,3/2"	3816,00
9127ATQU	127MB IDE/17ms QUANTUM ELS120AT,3/2"	4320,00
9124AT27	125MB IDE/15ms SEAGATE ST3144A,3/2"	4440,00
9170ATQU	170MB IDE/16ms QUANTUM ELS170AT,3/2"	5256,00
9240ATQU	240MB IDE/15ms QUANTUM 240AT,3/2"	7320,00
9240AT27	240MB IDE/12ms SEAGATE ST3283A, 3/2"	7320,00
9400ATQU	425MB IDE/15ms QUANTUM LP425AT, 3/2"	13680,00
9450AT27	450MB IDE/12ms SEAGATE ST3550A, 3/2"	13920,00
9525ATQU	525MB IDE/11ms QUANTUM LP525AT, 3/2"	18000,00
9540ATCO	540MB IDE/12ms CONNER CP3554, 3/2"	17040,00

FESTPLATTEN SCSI

9240SCQU	240MB SCSI/15ms QUANTUM LP240S, 3/2"	7920,00
9240SC27	240MB SCSI/12ms SEAGATE ST3283N, 3/2"	8820,00
9320SC27	320MB SCSI/10.7ms SEAGATE ST4385N,5/4"FH	17640,00
9425SCQU	425MB SCSI/14ms QUANTUM LP425S, 3/2"	14160,00
9525SC27	525MB SCSI/10,6ms SEAGATE ST3600N, 3/2"	20376,00
9540SCCO	540MB SCSI/12ms CONNER CP3540, 3/2"	18720,00
9600SC27	660MB SCSI/15.5ms SEAGATE ST4766NV,5/4"FH	19200,00
9000SC27	1 GB SCSI/15ms SEAGATE ST41200N,5/4"FH	27300,00
91GBSCQU	1,0 GB SCSI/11ms QUANTUM, 3/2" PD 1,0	25200,00

9000SCQU	1,2 GB SCSI/11ms QUANTUM, 3 1/2" PD 1,2	28200,00
9001SC27	1 GB SCSI/12ms SEAGATE ST11200N, 3 1/2"	32280,00
92GBSCFU	2 GB SCSI/11ms FUJITSU, 5 1/4" FH	55080,00
92GBSC27	2 GB SCSI/11ms SEAGATE ST42400N, 5 1/4" FH	48360,00

FESTPLATTEN MONTAGE-KIT

823F/J027	Universal Adapterkit für 3 1/2" Festplatten	108,00
	Bestehend aus Rahmen, Schrauben, Frontblende	
HDRIDE24	Wechselrahmen für IDE-AT-BUS Harddisk 3 1/2"	1080,00

WECHSELFESTPLATTEN und MO-DRIVES

SYQ55527	SYQUEST Wechselfestplatte SQ 555/44MB/25ms SCSI	5160,00
SYQ40027	SYQUEST SQ400 44MB Cartridge	1260,00
SYQ51127	SYQUEST Wechselfestplatte SQ 5110/88MB/25ms SCSI	6708,00
SYQ80027	SYQUEST SQ800 88MB Cartridge	1728,00
SYQ01027	SCSI-Host-Adapter für SYQUEST intern/extern Rancho RT1000-3	1260,00
SYEXT027	GEHÄUSE für SYQUEST extern mit Netzteil	1872,00
PSYQ8827	SYQUEST 88MB extern / Parallel Interface	13800,00
	externes Gehäuse mit Netzteil, inkl. 1 Cartridge, Kabel und Treiber Software	
RICO3010E	RICOH 127MB/45ms SCSI Magneto Optical Drive	20760,00
	SCSI-2 Interface, schreib/Lese kompatibel zu ISO Standard, ohne Cartridge	
RICO5031E	RICOH 600MB/37ms SCSI Magneto Optical Drive	42720,00
	SCSI-2 Interface, schreib/Lese kompatibel zu SONY, ISO Standard, ohne Cartridge	
RICO3010F	127MB Magneto Optical Disk für RICOH	950,40
RICO5031E	600MB Magneto Optical Disk für RICOH	3072,00

FESTPLATTEN extern - Parallel Interface

PHD08527	85MB Harddisk extern/Parallel Interface	11976,00
PHD12027	120MB Harddisk extern/Parallel Interface	14820,00
PHD20027	200MB Harddisk extern/Parallel Interface	20016,00

TASTATUREN

2100K027	TASTATUR 102 KEYS XT/AT (Datacomp) GR	600,00
2101K027	TASTATUR 102 KEYS XT/AT (Datacomp) US	600,00
2102K027	TASTATUR 102 KEYS (CHERRY) GR	864,00

MATH-CO-PROZESSOREN

701CP087	80287XL (INTEL) für 286 bis 20Mhz	1296,00
702CP087	80387SX-25Mhz (INTEL) 16 bis 25Mhz	1380,00
703CP087	80387DX 16-33Mhz (INTEL)	1440,00
708CP087	83D87-40MHz (Cyrix) f.386	1920,00

UNTERBRECHUNGSFREIE STROMVERSORGUNGEN

UPS55027	UPS 550 VA (Stand-by)	4800,00
UPS10127	UPS 1000 VA (Stand-by)	8400,00
UPS60027	PC-CARE UPS 600VA/400W (Stand-by)	7800,00
	incl. Novell Interface, incl. AVR	
UPS10027	PC-CARE UPS 1000VA/700W (Stand-by)	15300,00
	incl. Novell Interface, incl. AVR	
UPSMON27	UPS-MONITORBOARD LTC-LAN III	2160,00
	für Novell Server 2.xx, 3.xx	
UPSINT27	SERENA UPS-210 intern (300VA)	2940,00
	bestehend aus 5 1/4" Slim-Line Steuereinheit + Batterie mit Montage-Käfig, nicht für alle Gehäusetypen geeignet.	

STREAMER TAPES

1908S027	STREAMER MAYNARD 155 MB intern	15588,00
	incl. Controller Karte und Software	
1909S027	STREAMER CASSETTE 60 MB	540,00
1913S027	STREAMER CASSETTE 155 MB	780,00
1919S027	IRWIN STREAMER 40/120MB intern	4260,00

	inkl. Software (DOS, Windows3.1 engl.), inkl. Kabel zum Anschluß als 3.Floppy	
1914S027	IRWIN STREAMER 80/250MB intern	5220,00
	inkl. Software (DOS, Windows3.1 engl.), inkl. Kabel zum Anschluß als 3.Floppy	
1915S027	IRWIN STREAMER 40/120MB extern	5880,00
	benötigt IRWIN 4251 oder 4100 Controller, inkl Software (DOS, Windows 3.1 engl.)	
1916S027	IRWIN STREAMER 80/250MB extern	6864,00
	benötigt IRWIN 4251 oder 4100 Controller, inkl Software (DOS, Windows 3.1 engl.)	
1917S027	IRWIN INTERFACE 4251 (f.ext.Streamer)	1500,00
1918S027	IRWIN CONTROLLER 4100AT (f.ext.Streamer/ double Speed)	2520,00
191WEZ27	EZTAPE Streamer-Software V.2.2 dt	570,00
19UWMA27	MAP Assist Novell Utility	1428,00
191WEW27	EZTAPE Streamer Software f. Windows 3.1	1020,00
1920S027	IRWIN DC2000-40 Cartridge 40MB	384,00
1921S027	IRWIN DC2000-80 Cartridge 80MB	432,00
1922S027	IRWIN DC2000XL-60 Cartridge 60MB	576,00
1923S027	IRWIN DC2000XL-120 Cartridge 120MB	576,00

STREAMER TAPES

WT515027	WANGTEK 250MB Streamer intern / SCSI	9588,00
	ohne Controller, ohne Software, ohne Cartridge	
WT552527	WANGTEK 525MB Streamer intern / SCSI	13440,00
	ohne Controller, ohne Software, ohne Cartridge	
WT510027	WANGTEK 1GB DAT Streamer intern / SCSI	23040,00
	ohne Controller, ohne Software, ohne Cartridge	
WT620027	WANGTEK 2GB DAT Streamer intern / SCSI	27480,00
	ohne Controller, ohne Software, ohne Cartridge	
WTC25027	250MB Cartridge für Wangtek	504,00
WTC52527	525MB Cartridge für Wangtek	600,00
WTC1GB27	1GB Cartridge für Wangtek	264,00
WTC2GB27	2GB Cartridge für Wangtek	384,00
ARCSEUDO	ARC SERVE Solofür DOS und NOVELL-Workstations deutsch	2640,00
	für Wangtek SCSI Streamer	
ARCSE05	ARC SERVE Software für NOVELL-SERVER V.3.11 (5User)	5400,00
	für Wangtek SCSI Streamer	
ARCSE20	ARC SERVE Software für NOVELL-SERVER V.3.11 (20User)	16200,00
	für Wangtek SCSI Streamer	

STREAMER TAPES extern Parallel-Interface

PST25027	250MB Streamer extern / Parallel Interface	21480,00
	ext. Gehäuse mit Netzteil, Streamer Software (SYTOS+), Treiber, Cartridge u. Kabel	

MOUSE - TRACBALL

5103AS28	MICROSOFT SERIELLE MOUSE	1428,00
5103A028	MICROSOFT BUS MOUSE	1428,00
5103LP28	LOGITECH PILOT MOUSE seriell	552,00
5103LM28	LOGITECH MouseMan Bus- oder seriell	936,00
5102A028	SUPER-MOUSE II seriell	288,00
5109A028	GENIUS MOUSE ONE seriell / 2Tasten / anthrazit	288,00
5110A028	GENIUS F-302 serielle Mouse	636,00
5101A028	GENIUS TRACBALL GTK-320 seriell	1080,00
5100A028	GENIUS HiPoint TRACBALL seriell	1080,00
5104A028	GENIUS GM W 220 kabellose MOUSE seriell	996,00
5111A028	PS/2 Adapterstecker für GENIUS F-302	126,00
5105A028	MOUSE PAD	54,00

SCANNER - GRAPHIK TABLET

5150A028	GENIUS Handy Scanner B105	2640,00
	100-400 DPI, 256 Graustufen, Dr. Genius, Scan Edit, OCR Software	
5150AC28	GENIUS Handy Scanner GSC-105 COLOR	6708,00
	100-400 DPI, 256 Graustufen, 4096 Farben, Dr. Genius, Scan Edit, OCR Software	

5149A028	GENIUS OCR Software	1236,00
5160A028	GENIUS HiSketch 1212B GRAPHIKTABLET	5400,00
	12"x12", inkl. Treiber für ACAD 10,11, AustoSketch 2.0, Windows 3.1	
5159A028	GENIUS GT-S01 Stylus Pen für GT1212B + Stylus Pen ..	780,00
5020S024	HP-SCANJET plus schwarz/weiß Tischscanner	20376,00
5020SC24	HP-SCANJET II COLOR	34560,00
5020SS24	RECOGNITA Plus Texterkennungs-Software	15216,00

MULTIMEDIA - CD-ROM - FAXMODEM

MMSB2000	SOUND BLASTER V.2.0	1908,00
MMSBPRO0	SOUND BLASTER PRO	2700,00
MMARTMK1	ARTMEDIA MULTI MEDIA KIT	6864,00
	Artmedia Sound Karte (Sound Blaster OEM f. SONY) + SONY CD-ROM CDU31A	
MMSBMMK0	SOUND BLASTER MULTIMEDIA KIT Basic Version	9360,00
	Sound Blaster Pro, Sound Blaster CD-ROM, Windows 3.1 Treiber, 3Stk CD	
MMSBMMK1	SOUND BLASTER MULTIMEDIA KIT Business Version	9900,00
	Sound Blaster Pro, Sound Blaster CD-ROM, Windows 3.1 Treiber, 7Stk CD	
MMSBMIDI	MIDI Kabel + MIDI Software für SOUNDBLASTER PRO	1380,00
MMSBCDRI	CD-ROM für SOUNDBLASTER PRO intern	6120,00
MMSBCDRE	CD-ROM für SOUNDBLASTER PRO extern	7080,00
MMSBLAUT	SCREENBEAT SPEAKERS (für Montage am Monitor)	1020,00
MMVIDE0B	VIDEO BLASTER	6720,00
	Videodigitizer, Mischpult von verschiedenen Video & Audioquellen, Farbkontrolle	
MMPCVIDE	PC-VIDEO Karte	5880,00
	Videodigitizer, Mischpult von verschiedenen Video & Audioquellen, Farbkontrolle	
MMSOUNDM	SOUND COMMANDER fx mono	2100,00
	Adlib/Soundblaster kompatibel, MIDI-Anschluß, inkl. Lautsprecher	
MMSOUNDS	SOUND COMMANDER fx stereo	2100,00
	Adlib/Soundblaster kompatibel, MIDI-Anschluß, inkl. Lautsprecher	
MMMIDIKA	MIDI Anschlußkabel für Sound Commander fx	390,00
MMCD3727	CD-ROM NEC CDR-37 extern SCSI / Portable	7500,00
MMCD7427	CD-ROM NEC CDR-74 extern SCSI	9588,00
MMCD8427	CD-ROM NEC CDR-84 intern SCSI	8580,00

FAX-MODEM

MMINTEL1	Intel SatisFAXtion 100 Fax-Modem Karte (nur für den EXPORT!)	2280,00
	9600bps FAX G3, 2400bps Modem, inkl Crosstalk und Intel FAX Software	
MMINTEL2	Intel SatisFAXtion 200 Fax-Modem Karte (nur für den EXPORT!)	5760,00
	2400bps Modem V42bis, MNP1-5, sonst wie SatisFAXtion 100	
MMINTEL4	Intel SatisFAXtion 400 Fax-Modem Karte (nur für den EXPORT!)	7680,00
	14400bps Modem V32bis, V42bis, MNP1-5, sonst wie SatisFAXtion 100	
MMINTFSW	Intel Net SatisFAXtion	17592,00
	Software für FAX senden/empfangen in Novell Netzwerken, ein dedicated Faxserver wird empfohlen	
MMZYXELI	ZyXEL Fax/Modem U-1496B 14400bps INTERN (nur f.EXPORT!)	7440,00
	inkl. FAX Software für DOS/Windows, V.32,V.32bis, MNP4/5, nicht postgenehmigt	
MMZYXELE	ZyXEL Fax/Modem U-1496E 16800bps EXTERN (nur f.EXPORT!)	8160,00
	inkl. FAX Software für DOS/Windows, V.32,V.32bis, MNP4/5, nicht postgenehmigt	

MMLINKT1	MICROLINK 2240T Modem extern /Postzulassung	5940,00
	2400bps Modem, 9600bps FAX, kein FAX Empfang, V22bis, V42bis, MNP4/5, inkl Transfax für DOS und T-LIX	
MMLINKTR	MICROLINK 2240TR Modem extern /Postzulassung	8340,00
	wie 2240T, inkl FAX empfangen, V23 f. BTX Ausland	
MMFMKAB1	Telefon Anschlußkabel für FAX-MODEM Karte	390,00
MMWINFAX	Winfax - Software	1188,00

DISKETTEN

5701A028	SONY - DISKETTEN 5¼" DS/HD	18,00
5723A028	HOST - DISKETTEN 5¼" DS/HD	10,80
5720A028	SONY - DISKETTEN 3½" DS/HD	27,60
5722A028	HOST - DISKETTEN 3½" DS/HD	14,40

KABEL

5300A028	DRUCKERKABEL PARALLEL 1.8 m	90,00
5305A028	DRUCKERKABEL PARALLEL 5 m	168,00
5306A028	DRUCKERKABEL PARALLEL 7 m	216,00
5207A028	DRUCKERKABEL PARALLEL 10 m	270,00
5301A028	CENTRONICS/CENTRONICS 3 m	120,00
5308A028	CENTRONICS/CENTRONICS 7 m	480,00
5303A028	TASTATURKABEL 2 m (Verlängerung)	90,00
5304A028	MONITORKABEL 2 m RGB-TTL (Verlängerung)	144,00
5309A028	MONITORKABEL 2 m VGA (Verlängerung)	144,00
5004Z028	VGA Adpater 15M/9F	108,00
5003Z028	RS232 Adapter 25M/9F oder 25F/9M	108,00
5403A028	FLOPPY KABEL	60,00
5402A028	HARDDISK-KABELSET (MFM/ESDI)	240,00
5405A028	HARDDISK-KABELSET (SCSI)	120,00
5404A028	HARDDISK-KABELSET (IDE-AT-BUS)	120,00
5406A028	HARDDISK-KABEL (IDE-AT-BUS)	90,00
5310A028	Netzkabel	90,00
5311A028	Netzkabel zum Anschluß des Monitors am PC-Netzteil	108,00
5313A128	Y-Power Kabel	168,00

SONSTIGES ZUBEHÖR

5515FL28	Floppy Schloß für 5¼" Laufwerke	480,00
5516FL28	Floppy Schloß für 3½" Laufwerke	480,00
5514MF28	MONITOR-GLAS-FILTER 14" UNUS	420,00
5513MF28	MONITOR GLAS-FILTER 12-14" SPACE	2100,00
5621A028	TASTATURLADE UNTERBAU	660,00
5502A028	CPU Ständer für PC-Tischgehäuse	180,00
5500A028	DRUCKERSTÄNDER A4	228,00
DS101027	DATA SWITCH CENTRONICS 2 fach (AB)	348,00
DS101127	DATA SWITCH CENTRONICS 4 fach (ABCD)	528,00
DS101227	DATA SWITCH CENTRONICS AB-BA	816,00
5107A028	DISKETTENBOX 3½" für 100Stk	108,00
5108A028	DISKETTENBOX 3½" für 5x10Stk Rainbow-Box	144,00
5152A028	QUICK SHOT QS-113 JOYSTICK f. PC	198,00
5153A028	QUICK SHOT QS-123 JOYSTICK f. PC	276,00
5154A028	QUICK-SHOT QS 163 Game Card 2-Port	312,00
PAP00024	DRUCKERPAPIER 12"x240/60g 2000 Blatt	288,00

NETZWERK - KARTEN und ZUBEHÖR

ETH01026	ETHERNET CARD, 8-Bit kompatibel	1080,00
	NE-1000 kompatibel incl.BOOT-ROM	
ETH01126	ETHERNET CARD, 16-Bit kompatibel	1296,00
	NE-2000 kompatibel incl.BOOT-ROM	
ETHBUS26	ETHERNET CARD, 16-Bit BUS MASTER kompatibel	1548,00
	NE-2100 kompatibel incl.BOOT-ROM	
ETH01226	ETHERNET CARD 32 Bit EISA	4200,00
	inkl. Treiber für Novell 2.xx, 3.xx	
ETHINT26	INTEL ETHEREXPRESS 16Bit Ethernet Karte	2376,00
	selbstkonfigurierend, incl. Treiber für Novell	
ETHEIS26	INTEL ETHEREXPRESS 32 Bit EISA	10020,00
	selbstkonfigurierend, inkl. Treiber für Novell	
ETHINB26	INTEL BOOT ROM für Novell Netware	540,00
ETH10026	NOVELL NE1000, 8-Bit Ethernet Karte	1770,00
ETH20026	NOVELL NE2000, 16-Bit Ethernet Karte	2640,00

ETHROM26	BOOTROM FÜR NE-1000/2000	780,00
ETHPRO26	POCKET ETHERNET Adapter Parallel/BNC	2940,00
ETHXIR26	XIRCOM POCKET ETHERNET Adapter Parallel/BNC ...	5880,00
	Zum Anschluß an die parallele Schnittstelle	
ETHK5826	ETHERNET Kabel (Preis/Meter)	13,20
ETHBNC26	ETHERNET BNC-Connector	43,20
ETHTER26	ETHERNET Terminator	90,00
ETHUP026	ETHERNET UNTERPUTZ BNC DOSE	588,00
ETHAP026	ETHERNET AUFPUTZ BNC DOSE	780,00

NETZWERK - KARTEN und ZUBEHÖR

ETHAK226	ETHERNET Anschlußkabel f. DOSE 2m	600,00
ETHAK326	ETHERNET Anschlußkabel f. DOSE 3m	648,00
ETHAK526	ETHERNET Anschlußkabel f. DOSE 5m	744,00
0KONF026	KONFEKTIONIEREN per Kabel	300,00

NOVELL NETZWERK-SOFTWARE

NOV30026	NETWARE 386 (3.11) 5-USER	10560,00
NOV30126	NETWARE 386 (3.11) 10-USER	23760,00
NOV30226	NETWARE 386 (3.11) 20-USER	34680,00
NOV31026	NETWARE 386 (3.11) 100-USER	68280,00
NOVLIT26	NETWARE LITE V.1.1 + DR DOS 6.0 (Lizenz je Netzwerkstation)	1020,00

Upgrades auf Anfrage !

NOVELL-NETZWERK-INSTALLATION

0INSTS20	Installation Server Netware 2.2/3.11	9000,00
0INSTT30	Installation pro Workstation - " -	600,00
0INSTL30	Installation Netware Lite je Station	1800,00

MICROSOFT NETZWERK-SOFTWARE

WFW00099	WINDOWS für WORKGROUPS (inkl. Windows 3.1)	3780,00
	bestehend aus Windows 3.1, Netzwerkfunktion, Mail, Schedule +	
WFWZ0099	WINDOWS für WORKGROUPS Zusatzlizenz	3000,00
WFWADDON	WINDOWS für WORKGROUPS Add-on	1440,00
	bestehend aus Netzwerkfunktion, Mail, Schedule +, benötigt Windows 3.1	
WFWADDZU	WINDOWS für WORKGROUPS Add-on Zusatzlizenz	1176,00
WFWSTART	WINDOWS für WORKGROUPS Starter-Kit	13080,00
	bestehend aus 2* Windows f. Workgroups, 2* Netzwerkkarte, Kabel	
WFWUSERK	WINDOWS für WORKGROUPS User-Kit	6540,00
	bestehend aus 1* Windows f. Workgroups, 1* Netzwerkkarte, Kabel	
WFWSTADD	WINDOWS für WORKGROUPS Add-on Starter-Kit	8460,00
	bestehend aus 2* Windows f. Workgroups add-on, 2* Netzwerkkarte, Kabel	
WFWUSADD	WINDOWS für WORKGROUPS Add-on User-Kit	4260,00
	bestehend aus 1* Windows f. Workgroups add-on, 1* Netzwerkkarte, Kabel	

BETRIEBSSYSTEME

7004D031	MS-DOS 5.0 (deutsch) Umsteigerpaket	792,00
7005D031	MS-DOS 5.0 (deutsch) Bootversion Microsoft	1308,00

7005DD31	Nur mit PC gemeinsam MS-DOS 5.0 (deutsch) Bootversion OEM	900,00
7003OS31	Nur mit PC gemeinsam OS /2 V.2.0 (deutsch)	2340,00

STANDARD-SOFTWARE (deutsch)

ADOBE099	ADOBE TYPE MANAGER 2.0	1650,00
WINAMI30	AMI PRO V.3.0 für Windows (LOTUS)	7548,00
WINAMI20	AMI PRO V.2.0 für Windows (LOTUS) - ABVERKAUF ...	1788,00
WINCO3CD	COREL DRAW 3.0 für Windows (auf CD)	9480,00
WINCO3CX	COREL DRAW 3.0 für Windows (auf CD + Disketten) ...	10680,00
WINLOT99	LOTUS 1-2-3 für Windows	9000,00
WINACC99	MS ACCESS 1.0 Datenbank für Windows	10080,00
WINEXL99	MS EXCEL 4.0 für Windows	9480,00
WINOFF30	MS OFFICE 3.0 (Winword 2.0, Excel 4.0, Powerpoint 3.0, Mail Liz.)	15576,00
WINPRO30	MS PROJECT 3.0 für Windows	13560,00
WINPU099	MS PUBLISHER für Windows	3780,00
WIN31099	MS-WINDOWS 3.1	2028,00
WIN31Z99	MS-WINDOWS 3.1 Zusatzlizenz	1680,00
WIN31MSS	MS-WINDOWS 3.1 + Microsoft MOUSE seriell	2376,00
WINWORKS	MS WORKS für Windows	3480,00
WORKS299	MS WORKS für DOS	2820,00
WINWOR99	MS WORD 2.0 für Windows	9480,00
WINWOZ99	MS WORD 2.0 für Windows Zusatzlizenz	7560,00
WNCD3099	Norton Commander V.3.0 für DOS	2340,00
WNU60199	Norton Utilities V.6.01 für DOS	2580,00
WINPMS99	PAGEMAKER 4.0 für Windows	16200,00
PCT71099	PC TOOLS 8.0 für DOS	2520,00
QEMM0099	QEMM 386 V.60	2268,00
WSBAS299	SUPERBASE 2 für Windows	570,00
WSBAS499	SUPERBASE 4 für Windows, dbase kompatibel	13080,00
STACK099	STACKER Software (zur Steigerung der HD-Kapazität) ...	2010,00
TURBCD99	TURBO C++ 3.0 für DOS	2700,00
TURBCW99	TURBO C++ 3.0 für Windows	3000,00
TURBPD99	TURBO PASCAL 7.0 für DOS	3000,00
TURBPW99	TURBO PASCAL 1.0 für Windows	4500,00
WINVPN99	VENTURA PUBLISHER 4.0 für Windows	20340,00
WINWP100	WORDPERFECT 5.2 für Windows	9060,00
WINWP102	WORDPERFECT 5.2 für Windows Zusatzlizenz	5760,00
WP51DS99	WORDPERFECT 5.1 für DOS	9060,00
WP51DT99	WORDPERFECT 5.1 für DOS Zusatzlizenz	384,00
WPO30S99	WORDPERFECT OFFICE 3.0 für DOS (nicht netzwerkfähig)	576,00
WPO30N99	WORDPERFECT OFFICE LAN 3.0 für DOS / 5 USER	432,00

Weitere Software sowie Upgrades auf Anfrage !!

DIENSTLEISTUNG

0TECHN20	Techniker - Stunde	840,00
0ASSEM20	Assembling (Montage) u. Test	840,00
0FAHRT20	Fahrtkostenpauschale (nur innerhalb von Wien)	600,00
0ZUSTE20	Zustellpauschale (gilt innerhalb von Wien)	480,00

Softwareaktion für PCC - TGM

(gültig nur für Sammelbestellung bis 31.3.93)

Produkt	für das Betriebssystem	empfohlener VK-Preis	Bemerkung	inkl. MwSt Aktions-Schulopreis
BORLAND:				
Paradox	WIN		Datenbank	2.490,-
Quattro Pro	WIN		Tabellenkalkulation	2.490,-
Borland Pascal 7.0	DOS&WIN	8.990,-	Programmiersprache	2.790,-
LOTUS:				
AMI Pro 3.0	WIN	10.040,-	Textverarbeitung	2.690,-
1-2-3 WIN 1.1	WIN	10.040,-	Tabellenkalkulation	2.690,-
Freelance	WIN	10.040,-	Grafikprogramm	2.690,-
Smart Suite	WIN	15.920,-	Softwarepaket	4.690,-
enthält AMIPro, 1-2-3 WIN, Freelance WIN, CC:Mail				
Lotus Organiser	WIN	2.990,-	Adreß- & Terminmanagement	1.430
Microsoft:				
Windows 3.1	WIN	2.980,-	Benutzeroberfläche	1.390,-
Windows f. WG	WIN	4.990,-	Benutzeroberfläche mit Netzwerkeigenschaften	2.390,-
WinWord 2.0	WIN	13.800,-	Textverarbeitung	3.490,-
Excel 4.0	WIN	13.800,-	Tabellenkalkulation	3.490,-
Word 5.5	DOS	12.690,-	Textverarbeitung	2.990,-
DOS 5.0 Update	DOS	1.740,-	Betriebssystem	990,-
Access 1.0	WIN	7.490,-	Datenbank	3.990,-
Utilities:				
PC-Tools 8.0	DOS		Utilities	2.090,-
Norton Desktop 2.0	WIN		Utilities	1.050,-
Norton Utilities	DOS		Utilities	1.190,-

Lieferung ab Lager Wien. Preise verstehen sich inkl. Mwst. 1 Jahr Garantie. Gültig solange Vorrat reicht!
Comp Delphin, Berggasse 5, 1090 WIEN, Tel.: 0222/3105356/18

✕

An den
 PCCTGM
 Postfach 59
 1202 Wien

Ich bestelle lt. Sammelbestellung EDV-SHOP, PC-NEWS-1/93

Ich bin Schüler der _____ (Abt./Jgg.) und nehme das Angebot der kostenlosen Probemitgliedschaft gemäß den Clubstatuten des PCC-TGM für ein Jahr an. Ich bin Mitglied des PCCTGM, MNr.: _____. Ich wünsche telefonisch/ per Postkarte verständigt zu werden (Zutreffendes ankreuzen).

Name _____ Plz _____ Ort _____

Straße/Nr. _____ Telefon _____

Datum _____ Unterschrift _____

(bei Minderjährigen: Unterschrift des Erziehungsberechtigten)

Sonderangebot für PCC – TGM

(gültig nur für Sammelbestellung bis 31.3.93)

CMP 840D 130 MB (Einsteigermodell) 17.590,- öS

Desktop Gehäuse / 200 Watt Netzteil
 Prozessor: 80386DX-40, 40 MHz, 128 KB Cache Memory
 Hauptspeicher: 4 MB erweiterbar auf 8/16/32 MB
 Diskettenlaufwerk: 3,5 Zoll, 1,44 MByte
 Festplatte 130 MB, Maxtor, 15ms
 2 serielle / 1 parallele / 1 Game Schnittstelle
 VGA-Bildschirmkarte TSENG ET4000, 1MB, VESA
 DIN-Tastatur mit 102 Tasten

RIC CX 1464 LR

14 Zoll Multisync – Farbmonitor
 max. 1024x786 Bildpunkte, non interleaced, 70 Hz
 0,28mm Lochmaske, strahlungsarm
 Regel für: hor.& ver. Lage & Größe, Parabel & Trapez-Effekt

Erweiterungen für CMP 840D:

Aufpreis auf CMP 933D (i80486DX-33, 256 KB Cache)	5.620,- öS
Aufpreis auf CMP 950D (i80486DX-50, 256 KB Cache)	8.190,- öS
Aufpreis auf CMP 966D (i80486DX2-66, 256 KB Cache)	10.690,- öS
Speichererweiterung auf 8 MByte	1.950,- öS
Speichererweiterung auf 16 MByte	6.480,- öS
Aufpreis auf Festplatte 213 MByte, Maxtor, 15ms	2.280,- öS
Floppylaufwerk 5,25", 1,2 MByte	900,- öS
Aufpreis auf Mini-Tower Gehäuse	190,- öS
Aufpreis auf Big-Tower Gehäuse	690,- öS

Mitsumi CD-ROM Drive 3.490,- öS

internes CD-ROM Laufwerk 5,25"
 553/635 MB, <375ms
 inkl. Interfacekarte, Kabel, Treiber

Canon BJ300 + ASF 8.350,- öS

64-Düsen Bubble-Jet Drucker; Auflösung: 360x360 dpi
 vollautomatischer Einzelblatteinzug 100 Blatt A4; Endlos A4
 manueller Einzug für A3; 300 Z/sek. Druckgeschwindigkeit
 Canon-, IBM- und Epson-Druckersprache; Handbuch, Druckerkabel

Canon LBP-4 Plus 14.990,- öS

A4-Laserdrucker, 300x300 dpi, 1,5 MByte Ram
 4 Seiten/Min., Capsl Druckersprache
 Handbuch, Druckerkabel

I/O Maus 3500 (MS kompatibel, 3 Tasten, 400 dpi) 249,- öS

Lieferung ab Lager Wien. Preise verstehen sich inkl. Mwst. 1 Jahr Garantie. Gültig solange Vorrat reicht! **Comp Delphin**, Berggasse 5, 1090 WIEN, Tel.: 0222/3105356/18

Bestellschein

Bitte kopieren Sie dieses Blatt bei Bedarf.

An die Arbeitsgemeinschaft für
Didaktik, Informatik und Mikroelektronik
(ADIM)

Postfach 23
A-1191 Wien

Tel. 0222-369 88 59-8

Nr= Bestellung von **fertigen Skripten**,
(Nr)= Vormerkung für **geplante Skripten** (bitte beachten Sie die BTX-Seite ***56458#** oder rufen Sie uns an):

Band Nr.	Bezeichnung Produkt (Hersteller)	Version	ISBN 3-85071-		Auf- lage	Datum	nur Band		nur Disk		Band+Disk		Gesamt- preis: öS
			ohne Disk	mit Disk			öS	Stk.	öS	Stk.	öS	Stk.	
36	LOGO (IBM).....	1.0	002-5	003-3	2.	Nov88	45		50		85		
38	Turbo-Pascal (Borland).....	3.01	006-8	007-6	5.	Sep89	90		50		130		
39	RUN/C Classic (Age of Reason Co.)...	2.03	000-9	001-7	1.	Jul87	40		50		80		
40	Turbo-C (Borland).....6226*	2.0	024-6	025-4	5.	Aug91	100		50		140		
41	Turbo/Power Basic6451*	1/2	030-0	031-9	4.	Sep92	95		50		135		
(42)	C-Bibliothek (Turbo-C, Quick-C...)..				1.		-130		50		-170		*****
(43)	MS-DOS.....				1.		-100		50		-140		*****
(44)	Modula-2 (Jensen & Partner) V3.0				1.		-110		50		-150		*****
(45)	ADA.....				1.		-120		50		-160		*****
(46)	Word (MicroSoft).....	5.5			1.		-100		50		-140		*****
47	Turbo-Pascal (Borland).....6225*	6.0	020-3	021-1	5.	Jul91	100		50		140		
(48)	Quick-C (MicroSoft).....	2.5			2.		-100		50		-140		*****
49	Quick-Basic (MicroSoft).....	4.5	018-1	019-X	2.	Okt91	95		50		135		
50	C++ (Borland)6450*	3.1	028-9	029-7	2.	Aug92	120		50		160		
98	Peter Pfenicher: Turbo Pascal Anweisungssammlung mit Beispielen...	6.0	ISBN 3- 900985-00-6		4.	Okt91	130		■ ■ ■ ■	■ ■ ■ ■	■ ■ ■ ■	■ ■ ■ ■	
Gesamtanzahl der bestellten Skripten/Disketten:.....									■ ■ ■ ■		■ ■ ■ ■		■ ■ ■ ■ ■ ■ ■ ■
Gewünschte Freixemplare (für je 20 lieferbare Bände ein beliebiger Band), bitte Bandnummern angeben.....												0	
Versandkostenanteil pro SKRIPTEN-Sendung (entfällt ab 1.000 S Bestellwert).....												30	
Endsumme (inklusive 10% bzw. 20% Umsatzsteuer).....													

Die Hardwarepreise ändern sich in kurzer Zeit so rasch, daß der Bestellschein rasch ungültig würde. Wir organisieren Sammelbestellungen für Disketten, Modems und ähnliches Zubehör. Bitte fragen Sie in der telefonischen Sprechstunde der ADIM Wien oder fordern Sie eine aktuelle Liste (telefonisch oder schriftlich) an!

Änderungen und Irrtum vorbehalten!

* Fachbuchnummer (für Höhere technische Lehranstalten auch über die Schulbuchaktion zu beziehen) Verlag nr. 970 Wien, Stand: 31. August 1992 (PCC-TGM)

Bitte beachten Sie:

Die Disketten enthalten die Programmbeispiele des jeweiligen Bandes. Alle Disketten werden im Format 5,25" (360 KByte) geliefert. Lösungsprogramme zu den Übungsaufgaben sind aus pädagogischen Gründen nicht erhältlich. Werden nur Beispieldisketten bestellt, wird kein Versandkostenanteil berechnet. (Gilt nicht für Leerdisketten!) Bände ohne Datum (Nummer in Klammern) sind zwar geplant; da die Fertigstellung vor allem vom Zeiteinsatz der ADIM-Mitarbeiter in deren Freizeit abhängt, kann ein exakter Termin nicht angegeben werden. Aus aktuellen Gründen können einzelne Bände auch vorgezogen werden.

"~" bedeutet: geschätzter Preis für geplante Bände.
Die Umsatzsteuer ist in den Preisen enthalten: ADIM-Bände und ADIM-Bände+Disketten: 10%, Disketten allein und sonstiges: 20 %.

Auslandsbestellungen - nur gegen Vorauszahlung oder Verrechnung per Kreditkarte:

Postgiroamt München (BLZ 700 100 80), Konto 1209 14-800. Postcheckamt Chur, Konto 70-40051-3. Volksbank Brixen, Konto 37283. Der Rechnungsbetrag verringert sich um die Mehrwertsteuer und das Versandkostenpauschale. Die Portospesen werden in ihrer tatsächlichen Höhe verrechnet. Wir bitten um Vorauszahlung oder Verrechnung per Kreditkarte: der Rechnungsbetrag wird Ihnen vor der Auslieferung mitgeteilt. Die Skripten werden sofort nach Zahlungseingang versandt. Größere Bestellposten werden geteilt, da Buchsendungen nur bis 5 kg zugelassen sind.

Zahlungstermine im Inland: Wir versenden üblicherweise die Skripten als Brief oder Paket und bitten um Überweisung binnen 14 Tagen bzw. ab 10 Stück) binnen 3 Wochen. Bei Zahlungsverzug können wir allerdings weitere Bestellungen nur gegen Vorauszahlung und einen Inkostenbeitrag von S 30.00 oder gegen Nachnahme und Ersatz der Nachnahmespesen ausführen. Wir bitten um pünktliche Überweisung. **Hardware:** Wir bemühen uns, die günstigsten Angebote ausfindig zu machen. Da sich alle Preise oft sehr rasch ändern, werden die aktuellen Preise auf Wunsch mitgeteilt. Wir organisieren auch immer wieder gemeinsame Bestellaktionen. Alles weitere in der telefonischen Sprechstunde der ADIM Wien. Unter dieser Nummer sind auch die Termine der Sprechstunden zu hören.

Ich wurde auf die Aktivitäten der ADIM aufmerksam durch:

Verwendung der Skripten:

ein Seminar des PI Wien	Kollegin/Kollegen:	als Lehrer/in
ein Seminar des PI Graz		als Schüler/in
ein anderes Seminar		als Student/in
ein Mitteilungsblatt des PCC - TGM	sonstige Informationen:	für mein Hobby
eine Mitteilung in einer Zeitschrift		freiberuflich

Abonnement: Neue Skripten können im Abonnement bestellt werden. Das Abo kann jederzeit gekündigt werden.

Ich bestelle neue Skripten bzw. neue und überarbeitete Auflagen im Abonnement (Ja/Nein).....	
--	--

Adressen (bitte alle Angaben in BLOCKBUCHSTABEN):

Lieferung an (Vorname, FAMILIENNAME, Adresse)	Rechnung (falls verschieden) an: (Name, Adresse):
Tel-Nr:	Tel-Nr:
BTX-Nr:	BTX-Nr:

Ein Service, vor allem für unsere Interessenten aus dem Ausland:

Skripten + Disketten können bei der ADIM-Wien auch mit folgenden Kreditkarten bezahlt werden (bitte ankreuzen):

<input type="checkbox"/> Visa-Card	Kartennummer:	
<input type="checkbox"/> Euro/Master-Card	lautend auf:	
<input type="checkbox"/> American Express	gültig bis:	

Unterschrift des Bestellers (falls der Besteller noch nicht bei der Adresse angegeben ist, geben Sie bitte den Namen hier in BLOCKBUCHSTABEN an.) Bestellung von ganzen Klassen werden gerne bearbeitet. Angaben wie z.B. "3B" reichen aber nicht aus, der Name eines verantwortlichen Bestellers muß angegeben werden.

Ort, Datum:	Unterschrift:
-------------	---------------

Weitere Bestellmöglichkeiten:

Telefonisch: ADIM-Wien, Tel. 0222-369 88 59-8. Wenn der Anrufbeantworter eingeschaltet ist, sprechen Sie bitte langsam und deutlich und geben Sie auch Ihre Telefonnummer für Rückfragen an.

Schriftlich: auch bei der ADIM-Graz,
Postfach 37, A-8028 Graz.
Über BTX (siehe auch BTX-Seite *56458#):
ADIM-Wien, BTX-Nr.: 912 218 106 oder
ADIM-Graz, BTX-Nr.: 913 110 525
Über Telefax: ADIM-Wien: 0222-369 88 59 7
Über Telex: ADIM-Wien: 75 210 388 weim a
oder ADIM-Graz: 75 210 859 sber a

HTL - INFO

Die Computer Hot Line (CHL) ist eine Abteilung der REKIRSCH Elektronischen Geräte GesmbH & CO KG, 1220 Wien, Obachgasse 28, Sitz: Wien-HRB21.465-Handelsgericht Wien

SPITZENPREISE BEI COMPAQ

COMPAQ PROLINEA 3/25 zs Mod.84/W

- Intel 386sx, 25 MHz
- Sockel für Coprozessor
- 2 MB RAM
- 2 freie ISA Steckplätze
- 1,44 MB 3,5" Diskettenlaufwerk
- 2 serielle-, 1 parallele-, 1 Mausschnittstelle
- Windows 3.1 & Compaq Dos 5.0 vorinstalliert
- 1024*768 Super VGA
- 84 MB Festplatte
- Philips 14" Monitor

um öS 20.500.-
(öS 17.080.- excl.MWSt)

COMPAQ PROLINEA 4/50 Mod.120/W

- Intel 486DX2, 50 MHz
- Coprozessor Intel 387 (integriert)
- 4 MB RAM
- 3 freie ISA Steckplätze
- 1,44 MB 3,5" Diskettenlaufwerk
- 2 serielle-, 1 parallele-, 1 Mausschnittstelle
- Windows 3.1 & Compaq Dos 5.0 vorinstalliert
- 1024*768 Super VGA
- 120 MB Festplatte
- Philips 14" Monitor

um öS 36.255.-
(öS 30.210.- excl.MWSt)

COMPAQ Farb-Notebook CONTURA 3/25c Mod.84/W

- Intel 386sl, 25 MHz
- Sockel für Coprozessor
- 4 MB RAM
- 64 KB Cache
- 1,44 MB 3,5" Diskettenlaufwerk
- 1 serielle-, 1 parallele-, 1 Mausschnittstelle
- Windows 3.1 & Compaq Dos 5.0 vorinstalliert
- VGA-LCD Farbe, 9"
- 84 MB Festplatte
- Compaq Trackball

um öS 36.000.-
(öS 30.000.- excl.MWSt)

COMPAQ PROLINEA 4/33 Mod.120/W

- Intel 486DX, 33 MHz
- Sockel für Overdrive-Prozessor
- 4 MB RAM, 8 KB Cache
- 3 freie ISA Steckplätze
- 1,44 MB 3,5" Diskettenlaufwerk
- 2 serielle-, 1 parallele-, 1 Mausschnittstelle
- Windows 3.1 & Compaq Dos 5.0 vorinstalliert
- 1024*768 Super VGA
- 120 MB Festplatte
- Philips 14" Monitor

um öS 34.050.-
(öS 28.370.- excl.MWSt)

COMPAQ Notebook CONTURA 3/25 Mod.120/W

- Intel 386sl, 25 MHz
- Sockel für Coprozessor
- 4 MB RAM
- 64 KB Cache
- 1,44 MB 3,5" Diskettenlaufwerk
- 1 serielle-, 1 parallele-, 1 Mausschnittstelle
- Windows 3.1 & Compaq Dos 5.0 vorinstalliert
- VGA-LCD Monochrom, 9,5"
- 120 MB Festplatte
- Compaq Trackball

um öS 33.350.-
(öS 27.790.- excl.MWSt)

COMPAQ Farb-Notebook CONTURA 3/25c Mod.120/W

- Intel 386sl, 25 MHz
- Sockel für Coprozessor
- 4 MB RAM
- 64 KB Cache
- 1,44 MB 3,5" Diskettenlaufwerk
- 1 serielle-, 1 parallele-, 1 Mausschnittstelle
- Windows 3.1 & Compaq Dos 5.0 vorinstalliert
- VGA-LCD Farbe, 9"
- 120 MB Festplatte
- Compaq Trackball

um öS 39.600.-
(öS 33.000.- excl.MWSt)

Bei Bestellungen von HTL's gewährt Compaq Sonderrabatte. Rufen Sie uns an! (0222) 259 1256

Autosketch 3.0 für Windows 3.1 um öS 3.600.- incl.MWSt.

AutocAD 12.0 DXU kompatibel

(öS 3.000.- excl.MWSt)

Sofort bestellen
Fax: (0222) 259 4391 Tel: (0222) 259 1256

Offenlegung gemäß Mediengesetz 1981

Medieninhaber:	PCC-TGM
Grundlegende Richtung:	Auf Anwendungen im Unterricht bezogene Informationen über Personal-Computer-Systeme, Berichte über Veranstaltungen des Vereins, Beratung der Mitglieder gemäß der Statuten des PCC-TGM.

Impressum

Medieninhaber:	PCC-TGM Wexstraße 21, Postfach 59, 1202 Wien.
Herausgeber, Redaktion:	Franz FIALA Siccardsburggasse 4/1/22, 1100 Wien BOX: 604-50-70-1 (2400-N81), FAX: 604-50-70-2 TELEBOX: FRANZ-FIALA, BTX: 912-218-242
Druck:	WUV, Wiener Universitätsverlag Beriggasse 5, 1090 Wien, (0222)-310-53-56/25
Versand:	CONCEPT Baumgasse 52, 1030 Wien, (0222)-713-59-41

PCC-TGM Personal-Computer-Club-Technologisches Gewerbemuseum

Telefon/Anrufbeantworter:	(0222)/33 22 398
Bürozeiten:	Mi: 19.00-20.30, Fr: 9.00-12.00h (Frau Jelinek, Herr Leeb, Herr Schluderbacher)
MAILBOX:	(0222)/815-48-71 (8-N-1), 14400..1200 Bit/s
BTX:	Leitseite *5645#, 912222584
DVR-Nr.:	0596299
Einschreib-/Jahresbeitrag:	Schüler: 0/150,- Student: 0/300,- sonst: 300,-/300,-
Information:	'Ein Club stellt sich vor' (=TGM-LIT-050)
Vorstand:	KÖNIG (Obmann), SYROVATKA (-stv), THUMFARTH (Schriftführer), WEISSENBOCK (-stv), ZEHETNER (Kassier), PAY (-stv), OSTERMAIER (Beirat), NAVRATIL (-), REITER (-), BERTHOLD (Rechnungsprüfer), NITSCHKE (-)

Preise-PCC-TGM, Stand Jänner 1993

Literatur, Sonderdruck pro Seite	öS	-80
Tabellen A5/A4, verschweißt	öS	15,-/25,-
Kop. Disketten 360k/1.2M/720k/1.44M	öS	40,-/50,-/50,-/60,-
Leerdisketten 360k/1.2M/720k/1.44M	öS	10,-/20,-/20,-/30,-
PC-SIG-Disketten 360k/720k (PC-SIG)	öS	15,-/50,-
3-fach Verteiler für PC-Netzteil	öS	100,-
Bausatz µPROFI-51, incl. Handbuch	öS	950,-
EPROM für µPROFI-51 (PC-NEWS-2/92)	öS	55,-
Scannerdienst erste Seite (+Disk)	öS	10,-
Scannerdienst Folgesette, pro Format	öS	2,-
Verpackungskostenanteil	öS	23,-/29,-
Diskettenabo pro Jahr 360k/720k	öS	110,-/160,-

ADIM Arbeitsgemeinschaft für Didaktik, Informatik und Mikroelektronik

Anschrift:	ADIM, Postfach 23, 1191 Wien, (0222)-369-88-59
Telefon Sprechstunde ADIM:	(0222)-369-88-59/8 Montag ab 20:00 Themen: Technik, Skripten, Unterricht.

MCCA BTX und Micro Computer Club Austria

Anschrift:	MCCA, Postfach 143, 1033 Wien BTX: *2550#, 912-222-064, FAX: (0222)/712-01-06 TEL: (0222)/585-13-93
------------	---

Inserenten

ADIM	87,88
CompDelphin	85,86
CTF-Computer	74
excon	76..84
fuzzy-logic	66
Hummer	75
nds	75
OPTV	3,4
PCC-TGM	89
Reklisch-Elektronik	90
SYWA Computer Doktor	70..73

PC-NEWS

Bezugsbedingungen:	Clubleistung für Mitglieder des PCC-TGM. Einzelbezugspreis \$ 40,- Jahresabo (5 Hefte) \$ 150,-																		
Bezahlte Werbung:	\$ 2000,-/Seite																		
Auflage dieser Ausgabe:	1800 Stück.																		
Beifaßkennzeichnung:	Name, Firma, [TGM-DSK(TGM-LIT)] Nicht namentlich gekennzeichnete Beiträge stammen von der Redaktion.																		
Kopien:	Mit Quellenangabe gerne gestattet. Zwei Belegexemplare erbeten.																		
PC-NEWS-Vorschau 1993	<table border="1"> <thead> <tr> <th>Ausgabe</th> <th>Redaktionschnitt</th> <th>Erscheint</th> </tr> </thead> <tbody> <tr> <td>PC-NEWS-1</td> <td>Mitte Jänner</td> <td>Mitte Februar</td> </tr> <tr> <td>PC-NEWS-2</td> <td>Mitte März</td> <td>Mitte April</td> </tr> <tr> <td>PC-NEWS-3</td> <td>Mitte Mai</td> <td>Mitte Juni</td> </tr> <tr> <td>PC-NEWS-4</td> <td>Anfang August</td> <td>Anfang September</td> </tr> <tr> <td>PC-NEWS-5</td> <td>Ende Oktober</td> <td>Ende November</td> </tr> </tbody> </table>	Ausgabe	Redaktionschnitt	Erscheint	PC-NEWS-1	Mitte Jänner	Mitte Februar	PC-NEWS-2	Mitte März	Mitte April	PC-NEWS-3	Mitte Mai	Mitte Juni	PC-NEWS-4	Anfang August	Anfang September	PC-NEWS-5	Ende Oktober	Ende November
Ausgabe	Redaktionschnitt	Erscheint																	
PC-NEWS-1	Mitte Jänner	Mitte Februar																	
PC-NEWS-2	Mitte März	Mitte April																	
PC-NEWS-3	Mitte Mai	Mitte Juni																	
PC-NEWS-4	Anfang August	Anfang September																	
PC-NEWS-5	Ende Oktober	Ende November																	
Layout und Satz:	WinWord 2.0																		
Erscheinungsort:	Wien, Jänner 1993																		
Programme:	Diskette TGM-DSK-294																		
Programme:	BTX-Seite *5645#, TGMBOX\DATEIEN\PCCTGM																		

Die Autoren dieser Ausgabe

Cap, Otto, Dr.,
Präsident des Kreisgerichtes Korneuburg, u.a. befaßt mit EDV-Arbeit in der Justiz und -schulung von Justizpersonal. PC-Einführungen für Rechtsberufe.
TEL: (0222) 290-23-18, BTX 912-214-392.

Fiala, Franz, Dipl.-Ing.,
Lehrer an der höheren Abteilung für Nachrichtentechnik und Elektronik im TGM, Redakteur der PC-NEWS.
BTX: 912-218-242.

Fleck, Eduard, Dipl.-Ing., Lehrer an der höheren Abteilung für Nachrichtentechnik und Elektronik im TGM. TEL.: (0222)-33-1-26.

Krebs, Arnulf O., Dipl.-Ing. war nach Absolvierung des Studiums der Kulturtechnik und Wasserwirtschaft zunächst als Assistent am Institut für Wasserwirtschaft der Univ. für Bodenkultur in Wien tätig. Danach sammelte er in der Wirtschaft praktische Erfahrung als Mitarbeiter in einer Unternehmensberatung. Seit 1989 ist er als Unternehmer aktiv, wobei er sich mit der Anwendung der Informatik im Umweltschutz (= Umweltinformatik) auseinandersetzt. Zu diesem Zweck gründete er 2 Unternehmen und entwickelte als EUREKA-Projekt ein Expertensystem für den Umweltschutz bis zur Marktreife. Seit 1991 beschäftigt er sich mit Fuzzy Logic und vertreibt seit dem u.a. die Produkte der Firma INFORM.

Riemer, Walter, Dipl.-Ing., Lehrer an der höheren Abteilung für Nachrichtentechnik und Elektronik im TGM, Leiter des Rechenzentrums der Abteilung, Autor mehrerer Lehrbücher für den Unterricht, Ingenieurkonsulent für Elektrotechnik.
BTX: 912-216-422.

Schlögl, Helmuth, Jahrgang 1940, Obmann des Btx- u. Microcomputer Clubs seit 1983, beschäftigt sich seit 1981 mit BTX (Pilotversuch 300 Teilnehmer), Beruf Bankbeamter tätig seit 1960 in der GiroCredit Bank (vormals Girozentrale) davon 25 Jahre EDV, bis 1991 mit BTX.

Weissenböck, Martin, Dipl.-Ing. Mag. Dr., Direktor der HTL Wien IV, Leiter der ADIM und Autor von ADIM-Skripten.
Vorstandsmitglied des PCC-TGM. BTX: 912 213 458.

Wiesauer, Kurt, Ing., Layout-Experte der Firma REKIRSCH-Elektronik, TEL: (0222)-25-36-26/31.

Zetik, Fritz, Schüler der 3AN, Abteilung Nachrichtentechnik und Elektronik am TGM.

Absender:

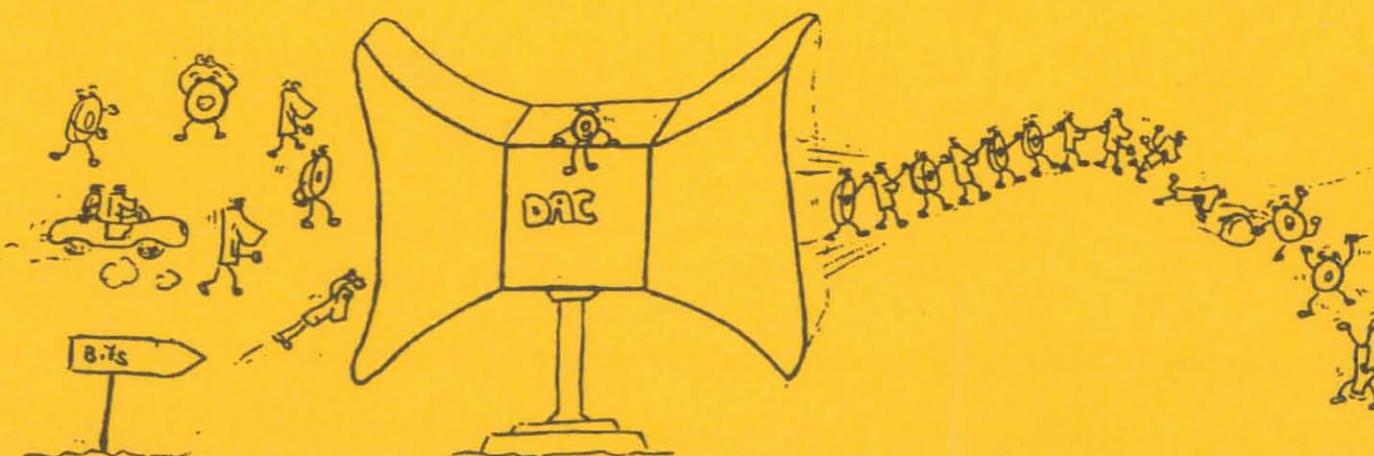
PCC-TGM
Wexstraße 21
Postfach 59
1202 Wien

P.b.b.
Verlagspostamt
1200 Wien

An:

Wenn unzustellbar, bitte zurück an Absender.

DA - WANDLER



Leitmannslehner, 5CN, TGM, Dez.1982