

KICK START

COMPUTERMAGAZIN

EIN GUTER FANG

AMIGA-Sommerhits

KICKSTART IM EPROM

Das eigene Betriebssystem

DAS ERBE

C64-Drucker am AMIGA



COMDEX · Sounddigitizer · DFÜ · Biochips
Coppermonitor · KICKS · Spiele · TOP 12

AMIGA



TEXTOMAT AMIGA

Egal, ob Sie Briefe oder ein ganzes Buch schreiben wollen, TEXTOMAT AMIGA einlegen und losschreiben – ohne langes Anlernen und Lesen im Handbuch. Mit Maus und Menü haben Sie den enormen Leistungsumfang von TEXTOMAT AMIGA schnell im Griff: alle Textverarbeitungsfunktionen, vielfältige Blockoperationen (Kopieren, Verschieben, etc.), hohe Geschwindigkeit bei der Ein- und Ausgabe von Texten, Direktformatierung am Bildschirm (WYSIWYG), Grafikeinbindung (IFF-Format), Kopieren und Einlesen von Bildschirmanschnitten aus anderen Programmen, automatische Silbentrennung, bis zu 30 Funktionstasten mit bis zu 160 Zeichen belegbar (als Floskelstasten, Mini-Adressdatei oder Tastaturmakros), beliebig viele Text- und Dezimaltabulatoren, Datentransfer über RS 232, Lesen von Programmen und Notepad-Dateien, Datenaustausch mit anderen Programmen über Clipboard, fertige Druckeransätze für Monitor- und Fernsehbetrieb, 2 verschiedene Zeichensätze für Monitor- und Fernsehbetrieb, ausführliches deutsches Handbuch.

TEXTOMAT AMIGA – die Textverarbeitung für alle zum nahezu konkurrenzlosen Preis.

nur DM 99,-

BECKERbase AMIGA

Der professionelle Datenbank-Manager zum Kaum-zu-glauben-Preis arbeitet nach dem Netzwerkmodell. Diese Struktur gestattet die Definition komplexer Dateiverbindungen mit schnellem Datenzugriff. Für eigene Anwendungen kann eine komfortable Benutzeroberfläche mit Pull-Down-Menüs und Window-Technik programmiert werden. Zwei leicht beherrschbare Programmiersprachen (DDL und TDL) garantieren hohe Flexibilität in der Anpassung an spezielle Benutzerwünsche. Mehrere fertige Demo-Anwendungen sind bereits installiert (Adreß-, Artikel-, Kunden-, Literaturverwaltung u. a. m.). Datenaustausch mit anderen Programmen ist möglich (ASCII-Format). Integrierter Text-Editor, Hilfe-System, einfache Datei-Definition, praktisch unbegrenzte Anzahl von Datensätzen je Datenbank, 65535 Datensätze pro Datei, praktisch unbegrenzte Anzahl von Feldern je Datei, max. Feldgröße 255 Zeichen, Paßwortschutz, ausführliches deutsches Handbuch, kurz: BECKERbase Amiga hat alles, was man für ernsthafte Datenbank-Anwendungen braucht. Minimalkonfiguration: 1 MByte RAM und Kickstart 1.2

nur DM 99,-

DATA BECKER
Merowingerstr. 30 · 4000 Düsseldorf · Tel. (0211) 31 00 10

DATAMAT AMIGA

Die Dateiverwaltung, die auch Bilder und Grafiken verwalten kann: DATAMAT AMIGA. Rundum professionelle Features: Programmsteuerung über Maus und Tastatur, frei gestaltbare Bildschirmmaske, komfortable Such- und Selektierfunktionen, Paßwortschutz, Datenaustausch mit anderen Programmen, Einbindung von Grafiken im IFF-Format – DATAMAT AMIGA kann alles, was man von einer Dateiverwaltung für den AMIGA erwartet. Weitere Pluspunkte in Kürze: Dateigröße max. 2 Milliarden Zeichen, max. 8 offene Dateien gleichzeitig, Datensatzgröße max. 64000 Zeichen, max. 2 Milliarden Datensätze, unbegrenzte Anzahl der Datenfelder, max. Feldgröße 32000 Zeichen, max. 20 Indexfelder mit wählbarer Genauigkeit (1-99) Oder-Verknüpfungen, Bildschirmmaskengenerator unterstützt Textarten und -größen, mehrzeilige Textfelder mit Wortumbruch- und Formatierungs-Möglichkeiten, integrierter Druckmasken- und Listen-Editor, ausführliches deutsches Handbuch.

nur DM 99,-

DATAMAT AMIGA

PROFIMAT AMIGA

Das komplette Programmentwicklungspaket für alle Nutzer, die ihren Amiga über die Maschinensprache voll ausreizen wollen. Mit allen Features, die der engagierte Programmierer erwartet: Integriertes Programmsystem bestehend aus Editor, Debugger, Disassembler und Reassembler, läuft unter CLI und Workbench, extrem schnell, da in Assembler geschrieben, mit umfangreicher Betriebssystembibliothek, MAKROS mit beliebig vielen Parametern unterschiedlichen Typs möglich, volle 32-Bit-Arithmetik, Fehlersuchfunktion, Cross-Referenzliste, bedingte und wiederholte Assemblierung menügesteuert, Debugger mit 68020 Single Step Emulation, erzeugt optional PC relativen und absoluten Code.

PROFIMAT AMIGA – das vielseitige Programmentwicklungspaket für den Amiga zum sensationellen Preis läuft auf jedem Amiga mit 512 KByte und Kickstart 1.2

nur DM 99,-

PROFIMAT AMIGA

COUPON

HIERMIT BESTELLE ICH FÜR MEINEN AMIGA

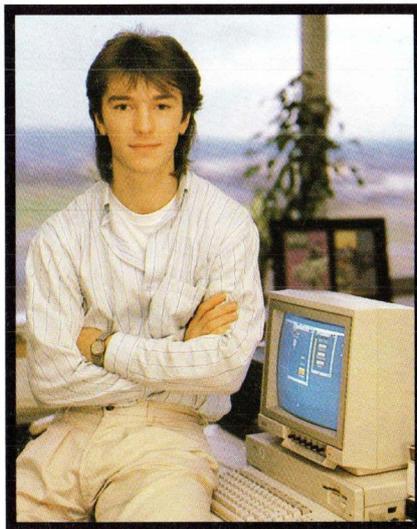
NAME, VORNAME

STRASSE, ORT

per Nachnahme Verrechnungsscheck liegt bei

HARDWARE im Selbstbau!

Ist Ihnen in den letzten Ausgaben der KICKSTART etwas aufgefallen? Sicher, immer mehr Hardware-Projekte waren im Magazin zu finden, die zum Selbstbau bzw. Nachbau geeignet waren. Bestückungspläne, Platinenlayouts, erforderliche Software und Hintergrundinformationen wurden an den Leser weitergereicht, so daß ihm das nötige Know-How zur Verfügung stand. Für solche Selbstbau-Projekte war natürlich immer ein wenig Geschick und Kenntnis im elektronischen Bereich vonnöten, aber auch den absoluten Laien ließ die Redaktion nicht alleine. In den meisten Fällen bieten wir das fertige Gerät bzw. die fertige Software zum Kauf an. Dem Käufer bringt es mitunter sogar einige Vorteile. Im Falle des Junior-Prommers ist die von der Redaktion vertriebene Software z.B. enorm aufgepöppelt worden (im Heft konnten aus Platzgründen nur die notwendigsten Routinen abgedruckt werden) und bietet eine Vielzahl von Features. Unter anderem wurde ein leistungsfähiger Monitor integriert, der seinesgleichen sucht. Eine grafische Benutzeroberfläche ist ebenfalls um die 'Kernsoftware' gelegt worden und vieles mehr. Die Software kann durchaus an professionellen Maßstäben gemessen werden.



In diesem Heft finden Sie gleich ein weiteres Hardware-Projekt, das eng mit dem Junior-Prommer zusammenhängt.

Eine Eprom-Platine, die es ermöglicht, das eigene Betriebssystem (Kickstart) im Amiga zu verewigen. Beispielsweise kann die neue Kickstart-Version 1.3 gebrannt und in den Amiga plaziert werden. Nach dem Einschalten meldet sich der Rechner daraufhin mit der Systemmeldung 1.3.

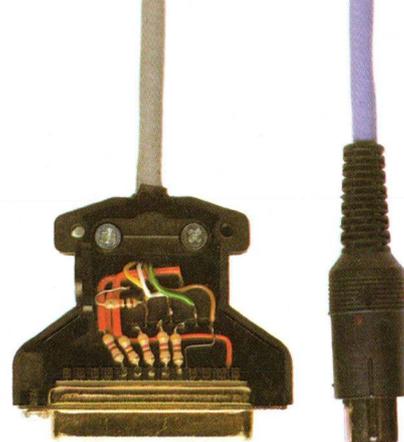
Für die Zukunft sind eine ganze Reihe von weiteren Hardwarebasteleien in Planung und Vorbereitung, die in den nächsten KICKSTART-Ausgaben zu finden sein werden.

Wenn Sie Anregungen, Verbesserungen oder eigene Hardwarebasteleien haben, melden Sie sich bei uns. Wie haben für Sie immer ein offenes Ohr.

Ihr

Andreas Krämer

INHALT



AKTUELLES

NEWS

Neuigkeiten rund um den AMIGA *Seite 6*

BIOCHIPS

Die neue Chipgeneration *Seite 16*

SPIELE-NEWS

Das Neueste aus der Spieleszene... *Seite 112*

TOP 12

Die aktuelle Spiele-Hitliste..... *Seite 115*

SOFTWARE

Druckmaster

Drucken ohne Workbench *Seite 33*

Der Druckdolmetscher

PD-Druckertreibergenerator *Seite 48*

Sicher ist Sicher

Harddiskbackupprogramme *Seite 51*

HARDWARE

Starboard II

Star oder einer von vielen? *Seite 46*

Der KICKSTART Umschalter

V 1.3 im Eprom *Seite 56*

Total digital

Sound-Sampler im Vergleich *Seite 64*

Das Erbe

Serielle Commodore-Drucker am AMIGA *Seite 67*

GRUNDLAGEN

Mit Basic ans System

Teil 5:
Auto-Requester & Gadgets..... *Seite 19*

LISP - Dialekt der Experten

Teil 2: Die Kapitel der KI..... *Seite 26*

Modula & Assembler

Maschinencode in Modula *Seite 40*

Modem Talking

Teil 2: DFÜ, wie geht das? *Seite 35*

DAS ERBE

Bei vielen Anwendern hat der Amiga den guten alten C64 in die Abstellkammer verdrängt. Mit einer kleinen Schaltung und etwas Software kann Ihr Amiga nun die verbleibende Peripherie des C64 übernehmen. Unser erster Teil zeigt die Nutzung eines seriellen Commodore-Druckers (IEC-Schnittstelle). In einer weiteren Folge werden wir auf die Verwendung der allseits beliebten 1541-Floppy eingehen.

COPPER-MONITOR

Der Copper im Amiga - ein Coprozessor mit recht geringen, aber leistungsfähigen Befehlen. Wie Sie mit drei Befehlen die Bildschirmdarstellung des Amiga auf wunderbarste Weise in ein Feuerwerk verwandeln können, zeigt dieser Betrag. Der Copper-Monitor gibt Ihnen die Möglichkeit, den Copper nach Ihren geheimsten Wünschen zu programmieren.

MESSE

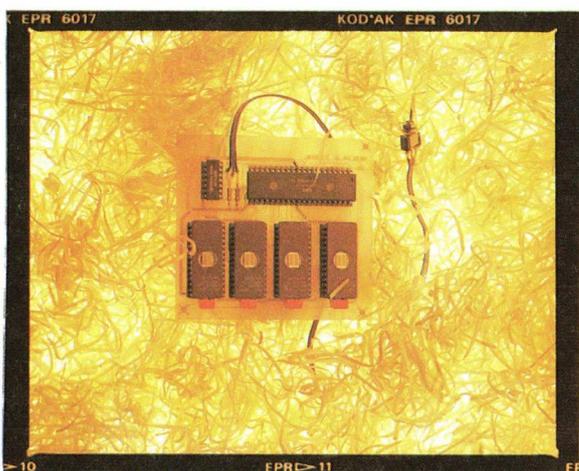
Neuheiten rund um den AMIGA sind immer gefragt. Auf der COMDEX in Atlanta hoffte man fündig zu werden. Unsere rasenden Reporter waren dort.



KICKSTART UMSCHALTER

Der neue Betriebssystem-Kern kommt! Doch die Anwender eines Amiga 500 oder Amiga 2000 haben das Nachsehen, denn Commodore liefert bis jetzt keine ROMs der Kickstart 1.3 aus.

Unser Kickstart-Umschalter macht Sie von solchen Gegebenheiten unabhängig - brennen Sie die Kickstart Ihrer Wahl auf vier Eproms, die dann auf dem Kickstart-Umschalter Platz findet.



TOTAL DIGITAL

Wir vergleichen zwei Sound-Sampler für den Amiga. Dieses Thema wird mit immer besserer Software für viele Anwender interessant.

Wie sieht es mit der Hardware aus? Was leisten solche Sampler? Was kosten sie? Antwort im Beitrag 'Total digital'.



Die inoffizielle
Lektüre der europäischen
Fußball-Nationalmannschaften

LISTING

Copper-Monitor Seite 80

KICKS FÜR INSIDER

IMAGE MAKER
Deluxe-Paint-Brushes
konvertiert Seite 88

SCHRUMPFE CLI
Das CLI im Kleinformat Seite 92

STILBRUCH
Steuercodes im CLI Seite 96

PLOT ABER FLOTT
Schnelle Punkte und Kreise Seite 99

MS-DOS

PC-Intim
Teil 5: Erinnerungsvermögen Seite 73

BATCH-Dateien
sind kein Geheimnis Seite 77

SPIELE

SPIELE-NEWS Seite 112

TOP 12
unsere monatliche
Hitliste Seite 115

Obliterator Seite 116
Reisende im Wind 1+2 Seite 118
Winter Olympics Seite 122
Wizball Seite 121

RUBRIKEN

Leserforum Seite 106
TIPS & TRICKS Seite 108
Einkaufsführer Seite 123
Inserentenverzeichnis Seite 125
Vorschau Seite 130
Impressum Seite 130

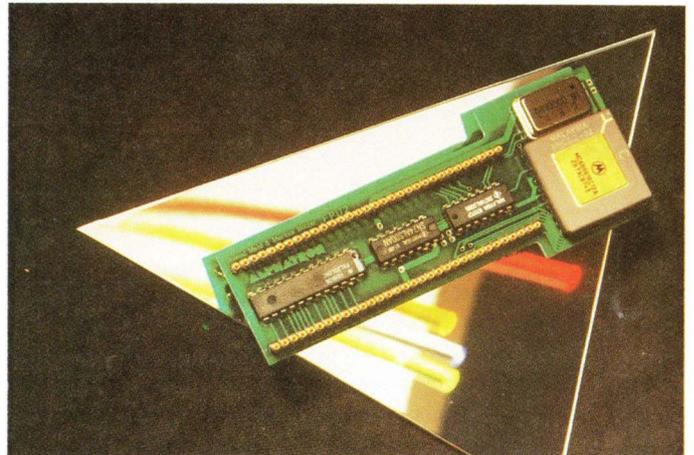
NEWS

KICKSTART 7/88

68881 IM AMIGA

Eine Zusatzplatine, auf der sich der mathematische Coprozessor 68881 befindet, bietet die Firma Alphatron Computersysteme an. Der Coprozessor wird mit einem eigenen 16 MHz-Takt versorgt, so daß unterstützende Software eine noch höhere Geschwindigkeit erlangt. Der Einbau ist denkbar einfach, hierzu muß lediglich der 68000-Prozessor aus dem vorhandenen Sockel entfernt und auf die Zu-

satzplatine gesteckt werden. Die Platine wird in den freien 68000er-Sockel gesteckt, und schon ist der Umbau abgeschlossen. Das Amiga-Betriebssystem erkennt automatisch, ob sich eine Floating-Point-Unit im System befindet, so daß keine Kompatibilitätsprobleme zu erwarten sind. Um den Coprozessor voll auszunutzen, muß die Software aber auf den 68881 zugeschnitten sein; ansonsten wird dieser nicht beachtet. Bei intensiven mathematischen Berechnungen können erhebliche Geschwindigkeitssteigerungen erreicht werden, von der mathematischen Genauigkeit ganz zu schweigen. Der Preis der Coprozessor-Platine liegt bei 798.- DM. Ein Test folgt in einer der nächsten KICKSTART-Ausgaben.



Der mathematische Coprozessor arbeitet mit einer Taktfrequenz von 16 Mhz.

Anbieter:

ALPHATRON-Computersysteme Tel. 09131-25018

Preis: 798.- DM

VIRUS ADE

In diesen Zeiten elektronischer Virusinfektionen läßt die entsprechende Software nicht auf sich warten. "Virus Finder" ist die Antibiotika für Ihren Amiga. Im Lieferumfang befinden sich gleich mehrere Programme - "Virus Finder", "Virus Kill" und "NoFastRam". Startet man das Programm "Virus Finder", so erscheint ein Menü, das vier Menüpunkte beinhaltet: Auswahl des Laufwerks, Diskette untersuchen, Virus von Diskette löschen und Ausgang. Durch Anwählen mit der Maus können die verschiedenen Punkte aufgerufen werden. "Virus Killer" ist ein Programm, das hauptsächlich für den Einsatz in der Startup-Sequence gedacht ist. Bootet man die Diskette, so zeigt das Programm einen vorhandenen Virus über einen Alert an; wenn kein Virus entdeckt werden konnte, verhält



Virus Finder gibt den Bootblock auf dem Bildschirm aus; der Anwender sieht somit sofort, ob die Diskette infiziert ist.

sich der Amiga wie gewohnt. Als Zugabe findet der Käufer noch das Programm "No Fast Ram". Das Produkt erkennt die beiden gängigen Virusarten (SCA, ByteBandit) und macht diese unschädlich. Multitasking stellt mit dem "Virus Finder" kein Problem dar; das Programm kann ständig als Hintergrund-Task laufen, um bei Bedarf genutzt zu werden.

Anbieter: PDC GmbH
Tel. 06172-24748
Preis: 49.- DM

UP-DATES

Für die Besitzer von US-Versionen einiger Programme sind jetzt von HS&Y deutsche Fassungen erstellt worden. In Absprache mit den Herstellern gibt es jetzt die deutschen Updates mit deutschen Menütexten und deutschem Handbuch. Besitzer der Originalversionen müssen dazu Diskette und Handbuch einschicken. Die Preise für die Updates im einzelnen:

- Butcher 2.0 PAL DM 49.-
 - Digi-Paint DM 39.-
 - Digi-View DM 39.-
 - Microfiche-Filer DM 60.-
 - MaxiPlan Plus DM 399.-
 - MaxiPlan 500 DM 120.-
- Heinrichson Schneider & Young
Herderstraße 94 5000 Köln 41
Tel.: 0221-43 16 87

DESK-TOP WORKSHOP

Für Desk-Top-Publishing (DTP) sind insgesamt sechs Disketten gedacht, die Motive in allen gängigen Bereichen beinhalten. Die einzelnen Motive können in alle DTP-Programme leicht eingebunden und darin weiterverarbeitet werden. Mit gängigen Zeichen-Programmen können die Motive an verschiedene Bildschirmformate angepaßt

Commodore weiter im Aufwind

Das dritte Quartal des laufenden Geschäftsjahres kann mit einer Umsatzsteigerung von 20% aufwarten. Insgesamt wurden von Januar bis März 1988 132.2 Millionen Mark umgesetzt. Der Gesamtumsatz von Commodore weltweit wird im dritten Quartal mit 200.3 Millionen US-Dollar beziffert, daraus läßt sich eine Steigerung von 18% ab-

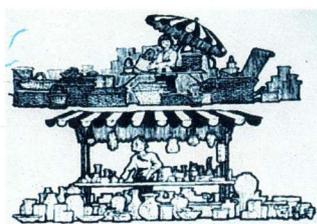
leiten. Alle drei Quartale zusammen erbrachten ein Umsatzergebnis von 699.5 Millionen US-Dollar, daraus resultiert ein Gewinn von 43.6 Millionen Dollar. Laut einer ersten Stellungnahme von Irwin Gould, Commodore Chairman und Chief Executive Officer, ist die AMIGA-Familie daran Umsatz mit 45% beteiligt.

NEUARTIGER RLL-FESTPLATTEN CONTROLLER

Die Firma EUROCOMP bietet einen neuartigen RLL-Festplattencontroller für alle PCs, XTs und ATs an und

somit auch für den Amiga 2000 mit Bridgeboard bzw. A1000 mit Sidecar an. Das Außergewöhnliche an diesem Controller ist, daß keine besonderen RLL-Festplatten mehr zum Betrieb von diesem speziellen Aufzeichnungsverfahren vonnöten sind. Daraus soll bei konventionellen Platten eine unübertroffene Speicherkapazität und Zugriffsgeschwindigkeit resultieren. Der Controller PS180 aus der Serie PERSTOR 200 verdoppelt annähernd die Kapazität von Festplatten auf elektronischem Weg. Das mitgelieferte Softwarepaket DiskExpander erlaubt noch weitere Steigerungen. Der Controller zeichnet die Daten in einem erweiterten RLL-Format auf; dadurch werden Übertragungsraten von 9 Megabit pro Sekunde erreicht.

Anbieter:
EUROCOMP Tel. 089-645415



Ein Beispiel von Objekten, die in ein DTP-Programm eingeladen werden können.

und verändert werden. Zur Zeit sind zwar nur vier Disketten erhältlich, zum 1.8.1988 sollen jedoch die restlichen zwei fertiggestellt sein. Von einzelnen Objekten bis hin zu kompletten Briefköpfen findet der Käufer zahlreiche Motive.

Anbieter:
Klaus Juris Computer-Grafik
Tel. 06081-8590
Preis: 89.- DM je Diskette

Pro-Write 2.0

In einer neuen Version 2.0 liegt das Textverarbeitungsprogramm ProWrite vor. Es besticht durch farbige Grafik und Fonteinbindung. In der neuen Version 2.0 ist ein Spell-Checker und Mail-Merge integriert. ProWrite läuft auf allen Pal-Amigas. Der Spell-Checker ist jedoch nur bedingt einsatzfähig, da das Produkt aus Amerika kommt und somit nur ein englischsprachiges Wortverzeichnis enthält.

NEUES KOPIER-PRO-GRAMM

Unter dem Namen "The Creative Vision Parameter Copier 1.03" vertreibt die Firma Future Vision ein neues Kopierprogramm, das nicht auf schnelle Kopierzeiten wert legt, sondern vielmehr auf das zuverlässige Anlegen von Sicherheitskopien. Dazu stehen dem Amiga-Anwender eine Vielzahl von Optionen zur Verfügung. Zum Kopieren können beliebige Parameter eingestellt werden, so daß laut Hersteller alle Disketten kopiert werden können. In einer englischsprachigen Anleitung (auf Wunsch auch in deutscher Sprache erhältlich) erfährt der Käufer Wissenswertes zum Kopieren von verschiedenen Disketten. Neben dem Kopierprogramm findet man desweiteren einige nützliche Zusatzprogramme -

Editor 2000

Unter der Bezeichnung Editor 2000 vertreibt die deutsche Firma Software 2000 einen leistungsfähigen Editor. Nach Starten des Programms fällt die Ähnlichkeit zu dem PD-Editor "MicroEmacs" auf. Einige Parallelen sind auch nicht zu verleugnen, jedoch ist

der Editor 2000 weitaus komplexer und komfortabler. Alle üblichen Editor-Features sind enthalten, desweiteren bietet das Produkt eine ganze Reihe von zusätzlichen Funktionen. So können beispielsweise Macros definiert und aufgerufen werden, ein neues CLI kann aufgerufen werden, ohne den Editor zu verlassen, und etliches mehr. Nützliche Hilfen zum reinen Programmieren sind ebenfalls zu finden. Mit CTRL-K können zugehörige Klammerpaare leicht ausgemacht werden, automatisches Einrücken ist selbstverständlich. Desweiteren erkennt der Editor automatisch durch die Extension des Programmnamens, ob es sich um ein C- oder ein Assembler-Programm handelt und unternimmt nötige Anpassungen.

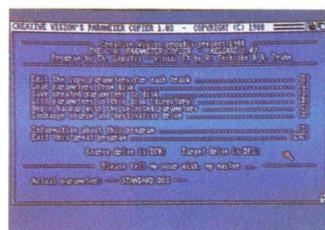
Der Editor nutzt die volle Pal-Auflösung und ist auf der Diskette in zwei Versionen, einer Fenster- und einer Screen-Version, vorhanden. Alle Funktionen können sowohl mit der Maus über Pull-Down-Menüs als auch über Tastatur-Sequenzen aufgerufen werden.

Der in deutscher Sprache gehaltene Editor ist auch mit einem deutschem Handbuch ausgestattet, das aber leider bei manchen Erklärungen etwas schwer zu verstehen ist.

Anbieter: Software 2000 Tel. 04522 - 1379 Preis:



Der Editor 2000 ähnelt etwas dem PD-Editor Emacs.



Der Copper ist verantwortlich für den farbigen Übergang des Hintergrundes.

Speedcheck, Examine und Errorchecker. Mit diesen Utilities können Sie wichtige Informationen zum Kopieren der wertvollen Originalsoftware erhalten. Um mit dem Programm richtig zu arbeiten,

ist etwas Know-How vonnöten, es bleibt aber festzustellen, daß der Parameter Copier eine Menge Features bietet, um sich eine Sicherheitskopie von kopiergeschützter Software anzulegen. Erfreulich ist auch der günstige Preis von 28.-DM.

Anbieter:
Future Vision
Tel. 06163-1278

Preis: 28.- DM

MICRO-FICHE FILER

Die bekannte Datenbank MICROFICHE FILER ist ab sofort auch in einer deutschen Version erhältlich. Dazu gehört ein 130 Seiten starkes deutsches Handbuch und die Software mit deutschsprachigen Menüs und Kommentaren. Im Gegensatz zur bisher erhältlichen Version 1.0 beinhaltet Microfiche Filer ein Utility-Programm, um ASCII-Dateien zu importieren. Außerdem unterstützt das Programm jetzt alle europäischen Zeichensätze. Ein Update auf die neue Version ist unter Zusendung der Originaldiskette und des Originalhandbuches zum Preis von 40.-DM erhältlich. Wer sich das Programm erst einmal betrachten möchte, kann gegen Zusendung von 10.-DM eine Demodiskette erhalten. Der Microfiche Filer ist für 179.-DM erhältlich.

Heinrichson Schneider & Young
Hiederstr. 94
D-5000 Köln 41
Tel. 0221-431687

Preis: 179.- DM
40.- DM UPDATE
10.- DM Demodiskette

Hair-style

Besondere Branchen benötigen besondere Lösungen. Nicht anders verhält es sich in der Friseur-Branche. Bereits auf der diesjährigen CeBIT in Hannover konnte ein bahnbrechendes System für Friseure bewundert werden. Ein ähnliches System bietet die Firma PBC - Peter Biet Computerdesign an. Der Amiga ist

Page Setter

- Integrierter Grafik- und Texteditor
 - Einlesen von Texten aus Textprogrammen (z.B. VizaWrite) und Bilder aus Zeichenprogrammen
 - Mehrseitige Dokumente
 - Verschiedene Schriften (Amiga Fonts)
 - Druckt mit allen Druckern in Preferences.
- Mit **PageSetter** erstellen Sie auf dem Amiga in einfacher Weise Publikationen wie, z.B. Vereins- oder Schülerzeitungen, Flugblätter und Prospekte.



Deutsche Version : DM 198.- (empf. VK)

GOLD DISK

Professional Page 1.1

- Desktop Publishing für professionelle Ansprüche
 - Einfache Bedienung
 - Grafik- und Texteditor
 - Trennautomatik
 - Einlesen von Texten aus Textverarbeitungsprogrammen (z.B. VizaWrite)
 - Automatisches Kerning
 - Einfügen von Bildern jeder Auflösung
 - Gestaltungsmöglichkeiten auch in Farbe
 - Ausgabe auf Postscript-Laserdrucker oder Satzmaschinen
 - Probeausdruck auf Matrixdrucker.
- Professional Page** gibt Ihnen für Ihre Drucksachen Gestaltungsmöglichkeiten auf dem Amiga, wie Sie noch bei keiner anderen Software geboten wurden. GOLD DISK hat hier ein Meisterwerk vorgelegt. Und wir bieten Ihnen als Distributor den Support, den Sie bei professioneller Software erwarten dürfen.
- Empfohlener Verkaufspreis: DM 741.-



Desktop VizaWrite 1.09

- Ein Textverarbeitungsprogramm der Superklasse
 - ▲ Vielseitiges Textbausteine-System
 - ▲ Bilder und Text mischbar
 - ▲ Automatisches Formatieren (mit Seitenumbruch)
 - ▲ Mehrzeilige Kopf- und Fusszeilen
 - ▲ Mehrere Dokumente gleichzeitig bearbeitbar
 - ▲ Serienbrieffunktion
 - ▲ Dokumentstatistik
 - ▲ Einfache Bedienung
 - ▲ Läuft mit 256 kB
 - ▲ Komplett in Deutsch.
- VizaWrite Amiga** bietet Ihnen alle nötigen Funktionen, um auf einfache Art und Weise Ihre Schreibarbeiten erledigen zu können. Ob es ein kurzer Brief, eine Dokumentation, eine Bedienungsanleitung oder die Textfassung für Professional Page ist – VizaWrite hilft Ihnen dabei.



Empfohlener Verkaufspreis: DM 198.-

Diese Anzeige wurde mit Professional Page gesetzt und auf einem Laserdrucker ausgedruckt.

Wir sind offizielle Distributoren dieser Programme und bieten Ihnen als Fachhändler und Endverbraucher folgenden Support: Hotline, Updateservice und Userinfos. Demodisketten sind kostenlos gegen Einsendung einer Leerdiskette und einem frankierten Rückumschlag erhältlich. Update PageSetter engl. auf deutsch für DM 30.- (inkl. Handbuch).



MICROTRON
COMPUTERPRODUKTE
Postfach 69 Bahnhofstr. 2
CH-2542 PIETERLEN

Poststraße 25
6200 Wiesbaden-Bierstadt
(0 61 21) 56 00 84
fax (0 61 21) 56 36 43



dank seiner hervorragenden Hardwareigenschaften Vater des Systems. Der Computer dient nur dem Wohle des Kunden, der sich individuell beraten lassen kann. Um dies zu gewährleisten, wird der Kunde digitalisiert und auf dem Amiga-Monitor dargestellt. Dort werden die alten Haare wegretuschiert, und der Kunde kann sich eine Vielzahl von Frisuren an das eigene Gesicht angleichen lassen. Der Amiga übernimmt sowohl das Ausschneiden als auch das Angleichen der neuen Frisur. Die einzelnen Frisuren können auch noch weiter manipuliert werden. Hat der Kunde eine passende Frisur gefunden, kommt es nur noch auf das Können des Friseurs an. Auf Wunsch kann der Kunde sein Bild auch auf einem Drucker ausgeben lassen. Das Hairstyle-Vario-System ist in mehreren Ausführungen bzw. Konfigurationen erhältlich. Das größte System schlägt mit ca. 16000.- DM zu Buche und beinhaltet einen Amiga 2000 mit 20 MB Festplatte, einen modifizierten Amiga 1084-Monitor, einen Echtzeitdigitizer mit s/w- bzw. Farbsoftwa-



Bild: In der s/w-Software kann man sich ebenfalls mit vielen Frisuren betrachten.

re, eine Profivideokamera mit Stativ, einen Thermotransferdrucker für Farbe und Schwarzweiß, Drucker-/Digitizerumschaltbox, Hairstyle-Software mit Frisuren und Modellkataloge sowie ein Demovideo auf Diskette für Werbezwecke und Schulung. Das System ist auch mit einem Amiga 500 als Basisgerät erhältlich und kostet dann ca. 13000.- DM. Die verschiedenen Komponenten sind zum Teil auch einzeln erhältlich. Vielleicht begegnet Ihnen beim nächsten Friseurbesuch der Amiga.

Anbieter:
PBC
Peter Biet Computerdesign
Georg-Fischer-Str. 5
6415 Petersburg 2
Tel. 0661-601263

Digitalisierer

Ebenfalls aus dem Hause PB-Computerdesign stammt ein, dem ersten Eindruck nach hervorragender Digitalisierer, dessen genaue Bezeichnung PAL-**RGB-MULTIPROZESSOR** lautet. Der Digitalisierer besteht aus mehreren Teilen, die zu einem Gerät zusammengefaßt sind. Das Multifunktionsgerät besitzt einen RGB-Farb-Splitter, einen Video-Color-Prozessor und einen RGB-Palumbwandler. Jedes Modul besitzt spezielle Eigenschaften, so daß das digitalisierte Bild ein Optimum darstellt. Die Qualität der digitalisierten Objekte ist beeindruckend.

Technische Daten:

Fernsehnorm PAL (NTSC auf Bestellung), Video-Eingang 1V FBAS, Video-Ausgang 1V 75 Ohm, externe Regelung: Helligkeit, Kontrast, Farbsättigung und -intensität. Eine Version II beinhaltet eine automatische und manuelle Umschaltung der RGB-Signale, einen eingebauten Videoverstärker und ein 10-Gang-Rasterpotentiometer.

Anbieter:

Peter Biet Computerdesign Tel. 0661-601263 Preis: 898.- DM

KURZ GETESTET

C.A.P.E. - Complete Assembler Programming Environment??!

Cape ist ein kombinierter Assembler und Editor. Man kann mit dem eingebauten Editor den Sourcecode editieren und dann mit dem Assembler Übersetzen. Somit muß der Editor nicht mehr mit RUN gestartet werden, und das ständige Abspeichern des Sourcecodes vor Aufruf des Assemblers entfällt. Mit im Paket enthalten ist auch der Blink (ein sehr schneller Linker).

Der eingebaute Assembler wird entweder per Menüleiste, mit EMACS oder Wordstarkommandos bedient. Die Geschwindigkeit des Editors ist nicht gerade berauschend, außerdem kann mit der Maus weder der Cursor positioniert noch ein Block markiert werden. Der Assembler dagegen hat sehr viele Befehle und unterstützt sogar den 68010. Die Optionen des Assemblers lassen sich entweder per Menüleiste oder im Source ein-

stellen. Leider hat der Assembler auch seine Tücken, denn er übersetzt, trotz erlaubter Adressierungsarten, nicht wie gefordert. Auch die Adressierungsart "Absolut short" scheint er nicht zu kennen. Ein weiteres Problem ergibt sich, wenn die Adressierungsart "xx(pc)" versehentlich als Zieladresse benutzt wird. Sie wird dann ohne weitere Warnungen in "absolut" umgewandelt. Dies kann sehr unangenehm sein, beispielsweise beim Schreiben eines adressunabhängigen Programms (oder wüßten Sie auf Anhieb, daß ein "tst xx(pc)" nicht erlaubt ist?).

Die PseudoOps hingegen sind gut durchdacht und sehr hilfreich. So gibt es z.B. die Möglichkeit, direkt ladbare Programme zu erzeugen. Sie müssen also nicht mehr gelinkt werden. Angenehm fällt auch auf, daß der Optimizer abschaltbar ist (so mancher hat sich nämlich schon gewundert, was der Assembler aus seinem Sourcecode gemacht hat). Obgleich der Optimizer keine Branches auf Vorwärts-Referenzen kürzen kann, sind doch einige neue Ideen in ihm verwirklicht.

Alles in allem ist CAPE ein recht nützliches Tool, mit dem sich arbeiten läßt. Trotzdem wären einige Verbesserungen für die nächsten Versionen wünschenswert.

Bezugsadresse:
GTI
Tel. 06171-73048



*Die Qualität des PAL-**RGB-Multiprocessors** ist beeindruckend.*

GENLOCK 8800 DM 998,-
PERFECT VISION Color Video Digitizer DM 498,-
POLAROID PALETTE mit Interface Imprint DM 6750,-
BILDERDIENST DiA und NEGATIV Belichtung 1 Stk. DM 4,95 ab 36 Stk. DM 3,31
THE DIRECTOR m. dt. Handbuch DM 139,-

SILVER m. dt. Handbuch DM 298,-
ANIMATE 3D m. dt. Handbuch DM 298,-
SCULPT 3D m. dt. Handbuch DM 229,-
FORMS IN FLIGHT m. dt. Handbuch DM 189,-
PHOTON PAINT m. dt. Handbuch DM 229,-
VIDEOSCAPE 3D m. dt. Handbuch DM 398,-

APPRENTICE ANIM. m. dt. Handbuch DM 598,-
PRO VIDEO CGI Fontanimation DM 398,-
UMLAUTE f. Pro Video EUROPA Font mit Umlauten DM 79,-
CREDITS Scrollprogramm m. dt. Anleitung DM 129,-
MOUSE PADS je DM 14,95 in schw., rot, bl., gr., br., lila, tan.
ECE MIDI DM 139,-

SONIX m. dt. Handbuch DM 159,-
A DRUM m. dt. Handbuch DM 119,-
STUDIO MAGIC m. dt. Anleitung DM 129,-
AUDIO MASTER m. dt. Handbuch DM 118,-
PERFECT SOUND Stereo Sound Digitizer DM 198,-
DELX. MUSIC CONSTR. SET m. dt. Handbuch DM 199,-

DIGA m. dt. Handbuch DM 189,-
ZING KEYS m. dt. Handbuch DM 129,-
AZTEC C Compiler: V. 3.6 Professional incl. dt. Handbuch DM 499,- Developer incl. dt. Handbuch DM 655,- Commercial incl. dt. Handbuch DM 998,- dt. Handbuch solo DM 128,-

weitere Info: LOFTPOST anfordern.
 video LOFT Fiedlerstr. 22-32 3500 Kassel tel.: 0561 / 87 79 26 - 87 33 99
 Für Besitzer der Originalprogramme dt. Handbücher solo DM 39,95

VIDEO LOFT
HARD SOFT
 Wulf G. Omsch

AMIGA * Public Domain Software * ab 2,75 DM

Wählen Sie selbst aus 1800 Disketten Ihre PD-Software, z. B.: Fred Fish, Panorama, TBAG, Tornado, Taifun — Auge 4000 Chiron Conceptions, Amicus, Amuse, Sacc, RW, Amigazin Kickstart, Case Mi Amiga, Juice Magazin, DBW Render 3.0, Ray Tracing V2.0, ACS, Virus Protector, Software Digest Utilities, Wieners Cycle System, ES Soft, Tools, Demos.

Einzeldisk	DM 4,85
bis 10 Stück	DM 4,80
bis 30 Stück	DM 4,70
bis 60 Stück	DM 4,60
bis 90 Stück	DM 4,50
bis 120 Stück	DM 4,40
auf 3,5"-Disketten 2DD.	

Achtung neu! Ray-Tracing-Construction-Set V2.0, siehe Amiga 1.88, S. 117. Komplettpaket 3 Programmmdisks und 2 Katalogdisks sowie ausgedruckte deutsche Anleitung für **DM 29,95** inkl. Porto.

Achtung neu! Bei Abnahme ab 30 Disketten kostenlos für den Anfänger oder Profi... CLI-Hilfe auf Diskette, lesen, kopieren, editieren, sortieren, drucken, renamen und vieles mehr, ähnlich wie CLIMATE oder ZING... DirUtil IV.12.

Nur 140,— DM für jedes Paket mit 30 PD-Disketten, inkl. Porto, Verpackung und CLI-Hilfe DirUtil, bei Vorkasse (V-Scheck oder Bar). Zum Beispiel: Paket Nr. 1c = Fred Fish Nr. 61—90
 Paket Nr. 1d = Fred Fish Nr. 91—120
 Paket Nr. 3 = Panorama Nr. 1—30
 Paket Nr. 4 = Faug Hot Mix Nr. 1—30
 Paket Nr. 7 = Kickstart Nr. 1—30
 Paket Nr. 8 = Taifun Nr. 1—30

Oder Sie stellen sich Ihr ganz persönliches Paket aus unserem Amiga PD-Katalog zusammen.

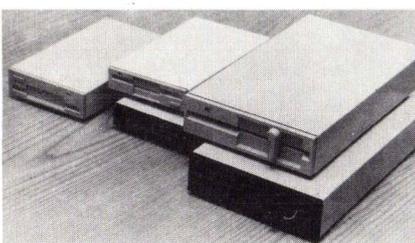
NEU *** NEU
 RPD, S.A.F.E., Ruhr, Goldfish, GET IT Magazin, UKaug Sonderaktion! 10er Paket PD-Neuheiten, inkl. Porto DM 45,—
 Taifun Nr. 41—50 .. DM 45,— S.A.F.E. Nr. 01—10 DM 45,—
 Taifun Nr. 51—60 .. DM 45,— S.A.F.E. Nr. 11—20 DM 45,—
 RPD n. Wahl, 10 St. DM 45,— Ruhr Nr. 01—10 .. DM 45,—

2 Katalog-Disketten mit Information über Inhalt der Programme für Amiga 500/1000/2000 gegen **DM 5,—** in Briefmarken/bar/V-Scheck anfordern!
 Spezial-Katalog über Original PC-SIG-Public Domain- und Shareware-Programme für den Amiga mit PC-Karte oder mit MS-DOS-Transformer gegen **DM 5,—** in Briefmarken/bar/V-Scheck anfordern!
 Am gleichen Tag des Bestelleingangs erfolgt der Versand unserer Kataloge!

Versandkosten PD-Disketten:
 Porto für Inland/Ausland ... DM 3,—
 Nachnahme für Inland ... DM 8,—
 Nachnahme für Ausland ... DM 16,—

Kombinierte Anwendung! Inhalt des Paketes: Datenbank, Textverarbeitung, Rechnerkopplung, Datenfernübertragung, Text-Editor, Disk&Harddisk-Monitor, HD-Backup, CAD, Ray-Tracing, Antivirus-Prog und vieles mehr.

Angebot des Monats: DM 45,— (V-Scheck), DM 49,— bei NN incl. Porto und Verpackung.



AMIGOS für Ihren AMIGA!

2 Jahre Garantie, 14 Tage Umtauschrecht, professionelle Leiterplatten, fast alle IC's gesockelt, Bedienungsanleitung, auf Wunsch vollständiges Manual mit allen Daten zu den Laufwerken lieferbar, 2-tägiger Liefer-Rhythmus.

Für alle Laufwerke gilt:

- voll kompatibel zur vorhandenen Soft- und Hardware,
- komplett anschlussfertig,
- amigafarbenes Metallgehäuse,
- abschaltbar (intelligente Abschaltung),
- Kapazität 880 KB,
- korrekte LED-Ansteuerung,
- erkennen Disk-Change,
- kein separates Netzteil nötig (Stromversorgung über AMIGA)
- an alle AMIGA-Modellen anschließbar

Für unsere 5.25"-Laufwerke gilt zusätzlich:

- alle umschaltbar 40/80 Tracks

Alle Laufwerke sind auch mit Busdurchführung lieferbar und sind dann mit einer automatischen Laufwerkserkennung ausgestattet, so daß beim Anschluß eines weiteren Laufwerkes an unser Laufwerk, das Fremdlaufwerk auf die nächsthöhere Laufwerksadresse als unser Laufwerk gesetzt wird.

Aufpreis: DM 25,—

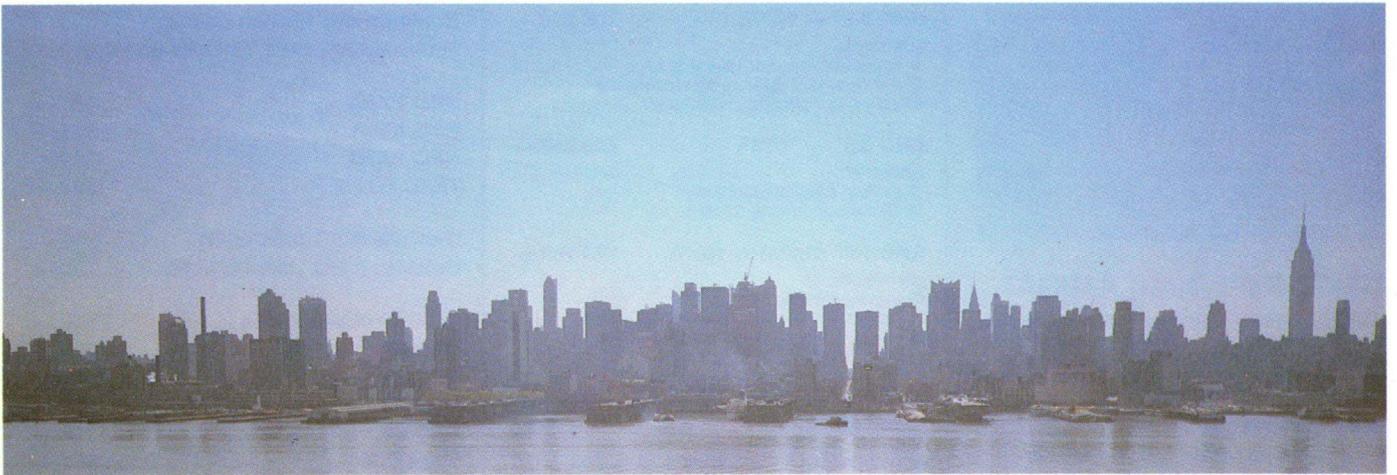
SND 3.5" — 1037A	DM 249,—
zusätzlich:	
● Superslimline, nur 25,4 mm hoch	
● nur noch 5 V Spannungsversorgung	
● sehr niedriger Stromverbrauch	
SND 3.5" — 1036A	DM 269,—
zusätzlich:	
● extrem robuste Mechanik	
● Standardbauhöhe 32 mm	
SND 3.5" Digital — 1037A	DM 289,—
zusätzlich:	
● durchgeführter Bus bis df3:	
mit automatischer Laufwerkserkennung	
● Digitale Trackanzeige mit Helligkeitsregulierung	
SDT 5.25" — TEAC FD 55 FR	DM 299,—
zusätzlich:	
● schwarze Frontblende	
● unformatiert 1 MB Kapazität	
SDN 5.25" — NEC 1157C	DM 309,—
zusätzlich:	
● helle Frontblende	
● Diskettenauswurf durch Feder	
● unformatiert 1,67 MB Kapazität	
SDN 5.25" Digital — 1157C	DM 339,—
zusätzlich:	
● durchgeführter Bus bis df3:	
mit automatischer Laufwerkserkennung	
● Digitale Trackanzeige mit Helligkeitsregulierung	
SDN 3.5" intern	DM 219,—
● für Einbau in A2000	
● komplett mit Einbauanleitung und Montagematerial	
● helle Frontblende	

Rohlaufwerke für den Bastler

(unmodifiziert, ohne Gehäuse und Kabel):

NEC 1036A	195,—
NEC 1037A	195,—
NEC 1157C	229,—
TEAC FD 55FR	229,—
Gehäuse (NEC 1036, 1037)	19,—
Gehäuse (NEC 1157, TEAC FD)	22,—
AMIGA 2000 und 1084	2590,—
XT-Karte	790,—
AT-Karte	879,—
NEC P2200	1199,—
NEC P6	588,—
Star LC 10	1499,—
Eizo Flexscan	1398,—
Mitsubishi EUM-1471A	
Festplatte 30 MB — 5.25" (für A2000 intern)	849,—
Festplatte 20 MB — 3.5" (für A2000 intern)	949,—
Festplatte 30 MB — 3.5" (für A2000 intern)	1049,—
Festplatte 30 MB (für A5000/1000 extern)	949,—
Golem 2 MB für A1000	1099,—
Profex 2MB für A500	890,—
Bootselektor	19,—
Farbband NEC P6	17,—
Farbband NEC P2200	17,—

HEISSE TAGE IN ATLANTA



Die Supershow der Computerfirmen in Atlanta, Georgia, zeigte die Leistungsbreite der Computerbranche. Dominierend wie immer: Big Blue (IBM). Aber auch der Amiga hat seinen festen Platz in der Welt der Anwendungsriesen.

Atlanta, die Hauptstadt des amerikanischen Bundes-

staates Georgia, wurde für eine Woche zum Mekka der Computerbranche. Showtime war Montag, der 2. bis Donnerstag, der 5. Juni 88. Bereits am Sonntagabend war in der Innenstadt kein Hotelzimmer mehr aufzutreiben. Mit etwas Glück und der Hilfe eines der drei Millionen Einwohner gelang es uns aber trotzdem, ein Motel in nur sechs Meilen Entfernung von der Stadt zu

finden - weniger als eine Autostunde (manchmal auch länger).

Die Comdex ist, im Gegensatz zur CeBit, nur für "Fachpublikum" geöffnet. Dies bedeutet, daß sich hier vor allem Händler und Presse über die neuesten Produkte informieren. Das ausgesuchte Publikum zahlt für dieses Vergnügen denn auch einen stattlichen Preis.

DIGI- 680?0 TI- ZER

Eine ebenfalls sehr interessante Anwendung im Bereich Graphik bietet SunRize Industries. Perfekt Vision wandelt Bilder aus der Videokamera oder vom Videorecorder in IFF-Dateien. Schwarz/weiß-Bilder werden innerhalb 1/60 sec. verarbeitet. Eine genaue Zeitangabe für Farbbilder steht noch aus. Das komplette Paket inklusive Hardware soll 250 US\$ kosten. Daneben bietet SunRize auch einen Electronic Color Splitter an. Das Programm macht die RGB-Scheibe beim Einlesen von Farbbildern via Videokamera überflüssig und arbeitet mit Perfect Vision und - mit Einschränkungen - auch mit Digi View zusammen.

GEN- LOCK

Bleiben wir noch ein bißchen bei Spielereien mit der Videokamera. AmiGen nennt Mimetics sein Genlock, das sowohl als NTSC- als auch als PAL-Version zu haben ist. Der Verkaufspreis ist mit 180 US\$ angegeben. Ein Testgerät wurde uns für Juli zugesagt. Dann wird sich zeigen, inwiefern es mit eine Alternative zu professionellen Geräten, wie dem SuperGen von Digital Creations (795 US\$), ist.

Vergleiche ganz anderer Art stellte CSA (Computer Systems Associates) in den Mittelpunkt der Produktpräsentation. Die beiden neuen 68020- und 68030-Boards bringen beachtliche Geschwindigkeitssteigerungen im Programmablauf. Zum 68020-Board wird derzeit auch noch -optional- ein 68881 angeboten.

HARD- DISK

Zeitersparnis beim Laden und ständige Zugriffsmöglichkeiten zu großen Datenmengen versprechen die Harddisks mit Speicherkapazitäten von 20 bis 250 MByte. Marktführer SUPRA hat hier jede Menge Konkurrenz bekommen. Besonders offensiv tritt dabei Allegro Tech auf. Mit einem eigenen Stand feierte das Team um Diane Nichols die auf der Comdex bekanntgegebene Exklusivvertretung für den Harddiskher-

steller PHÖNIX. Ebenfalls ins Festplattengeschäft eingestiegen ist Great Valley Productions. Daneben hat GVP, bisher hauptsächlich für Memory-Expansions bekannt, jetzt auch ein Modem entwickelt. Es unterstützt 300 bis 2400 bps. Kompatibel ist es mit Bell 103, 212A und CCITT (international) V.21, V.22 (A&B) und V.22-Standard. Dies gibt dem Benutzer die Möglichkeit, mit fast jedem System der Welt zu kommunizieren. Mehr Informationen können sich Telefonfreaks unter 001-215-889-9411 besorgen.

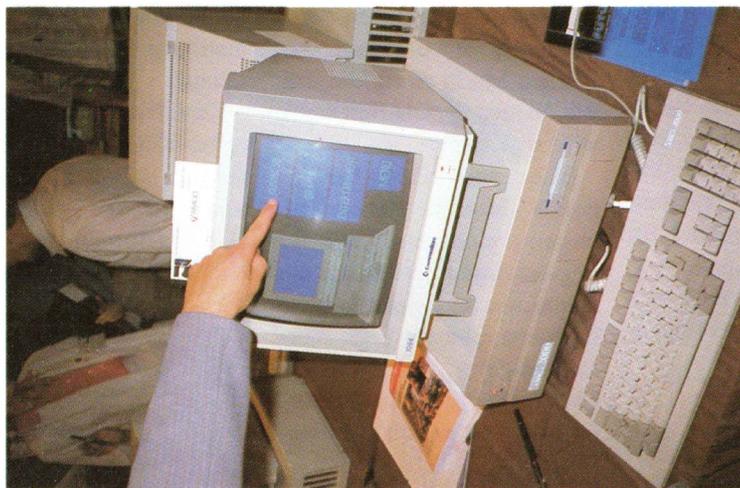
TAPE

Einen schnellen Datentransfer offeriert auch XETEC. FAST TRACK DRIVE SYSTEMS nennt sich die Serie bestehend aus SA-500 bzw. SA-1000 Amiga Host Adapter. Die Lese- bzw. Schreibgeschwindigkeiten von 300 und 200 KBYTE/SEC klingen beeindruckend. Dazu gibt es noch ein Tape Sub-System, das eine Kapazität von 51MB per Kassette verspricht.

Bevor wir das üppig bestellte Feld der Speichererweiterungen jetzt verlassen, möchte ich noch einer Bitte von ASDG entsprechen. Man hat uns gebeten, alle europäischen Kunden darauf hinzuweisen, das ASDG jetzt eine neue Adresse hat: 925 Stewart Street, Madison, WI 53713, Tel.: 001-608-273-6585. ASDG ist der Hersteller der ROCK SOLID RAM Expansion und des TWIN-X I/O-Boards.

MO- DU- LA 2

Immer stärker im Kommen ist die Programmiersprache Modula-2. Avant-Garde-Software (nicht zu verwechseln mit dem PC DITTO-Hersteller Avant-Garde-Systems) hat seinem erfolgreichen Programm 'Benchmark Modula-2' jetzt eine IFF & Image Resource Library und eine Simplified Amiga Library



Vom Touch-Screen mußten schon öfters Fingerabdrücke entfernt werden.

nachgeschoben. Beide sind Erweiterungen zu Benchmark Modula-2 und sollen die Benutzung von IFF-Dateien bzw. der ROM-Kernels unter Modula-2 vereinfachen.

UTILITY

Ebenfalls eine LIBRARY hat Word Perfect herausgebracht. Im Gegensatz zu den oben vorgestellten Programmen handelt es sich dabei aber um ein Tool, das dem "vielbeschäftigten Manager" Erleichterung verschaffen soll. Es beinhaltet einen Kalender, ein Notizbuch, einen Taschenrechner, einen File-Manager und einen Programm-Editor. Der Verkaufspreis soll bei 130 US\$ liegen.

BUCHFÜHRUNG

Bei nur 90 US\$ liegt die neue Version 3.0 von PHASAR (Antic Software), einem professionellen Buchführungsprogramm für "zu Hause". Ob es allerdings auch für den "nichtamerikanischen" Markt geeignet ist, muß sich erst noch zeigen (Test folgt).

SPIELE

Normalerweise liegt der Schwerpunkt der Neuvorstellungen bei Computermessen im Bereich Spiele-Software. Bei dieser Comdex war das Verhältnis beim Amiga zumindest ausgeglichen. Aber keine Angst, es gab auch etliche Spiele:

MICHTRON, der amerikanische Partner von Microdeal (England) präsentierte eine Umsetzung des Spielhallenautomaten COMMANDO WARRIOR. Das Amiga-Spiel führt den Namen Leatherneck und steht dem Automaten in puncto Handlung in nichts nach. Bis zu vier Spieler können sich gleichzeitig ins Gemetzel werfen.

Baudville, bekannt durch Video Vegas wird im August ein Spiel namens Dream Zone auf den Markt bringen. Bisher steht nur fest, daß es 49.95 US\$ kosten soll.

Nach Amerika importiert werden neuerdings Titel wie: City Defece, Persecutors und Spaceport. Ob sich Constellation Software allerdings mit diesen alten deutschen "Billigspielen" auf dem amerikanischen Markt einen klingenden Namen verschaffen kann, wird sich zeigen.

Einen guten Namen hat hingegen MicroIllusions. Die Neuerscheinungen Land of Legends und Encounter at the Dome machen einen hervorragenden Einruck (vgl. Vorankündigung Messebericht Heft 1/88). Ganz neu ist Turbo, eine Straßenrennsimula-

tion, bei der es dem Fahrer mehr als einmal passiert, daß er an einer geschlossenen Bahnschranke warten muß, während sich die Konkurrenz auf und davon macht. Auch Fußgänger spielen, genau wie in Firepower, wieder eine Rolle. Turbo bietet wieder eine Zwei-Spieler-Option und die Möglichkeit des Datentransfers per Modem.

MicroIllusions hat uns exklusiv ein Vorabexemplar zur Verfügung gestellt, den Test finden sie im Spieleteil unserer nächsten Ausgabe.

Weitere Vorankündigungen sind Dungeon Construction, Shrine of the Demon Soul und Craps Academy.

Dungeon Construction gibt dem Spieler die Möglichkeit sich seine eigenen Rollenspiele zu konstruieren. Es wurde z.B. bei der Entwicklung von Land of Legends benutzt. Ebenfalls ein Rollenspiel ist Shrine of the Demon Soul. Auch hier können Sie zusätzlich eigene Dungeon entwickeln und einbinden.

Nichts neues für den Amiga gibt es von Epyx zu berichten. California Games geht demnächst ins dritte Jahr der Vorankündigung und alle neuen Spiele sind nicht für den Amiga angekündigt - Konsequenz der Raubkopiererei. Übrig bleibt 4x4 Off Road Racing, man fragt sich nur wann es erscheint.

Zwei Neuvorstellungen gab es bei MICRO PROSE. Pirates soll schon in Kürze erscheinen, die Seeschlacht-Handels-Simulation ist bereits vom Atari ST und Apple

II (GS) bestens bekannt. Das andere Programm ist Gunship, der Hubschrauber-Simulator. Fred K. Schmitt hat uns aber auch gleichzeitig davor gewarnt, auf ein zu frühes Erscheinungsdatum von Gunship zu warten. Die Situation auf dem Amiga-Markt zwingt nach seiner Aussage alle Softwarefirmen zu sehr vorsichtigen Investitionen in diesem Bereich.

Ein Newcomer im Spielbereich ist Box Office. Zwar kennen die IBM-, Apple- und C64-Besitzer bereits Programme wie X-Tron, aber mit Combat Zone -Hubschrauber im Kampfeinsatz-, Viper Patrol -Shoot 'em up- und Adult Poker -nomen es omen- gibt es demnächst eine dreifache Amiga-Premiere. Box Office hat übrigens auch das erste Spiel über ALF für den IBM herausgebracht.

Mindscape setzt seine Sport-Serie mit Superstar Ice Hockey und Superstar Soccer fort. Beide Spiele bieten neben Action auch die Möglichkeit die jeweiligen Teams zu managen. Der "suggested retail price" von 49.95 US\$ ist daher sicher nicht zu hoch.

Top 12 Tip unserer letzten Ausgabe war Zoom! von Discovery Software. (siehe Spieleteil KICKSTART Juni 88). Daneben wurden von DSI noch Hybris und Sodan's Spiel mit dem Arbeitstitel SION vorgestellt. Hybris ist eine Raumschlacht mit drei Level, die jeweils mindestens acht Minuten lang sind. Die Graphik ist genau wie bei SION echte Arcade-Qualität. Alle drei Spiele werden übrigens für die Spielhalle umgesetzt.

Ähnlich stattlich war gleich die erste Aussage von Commodore, die da lautete: "...Commodore Business Machines, Inc. recently announced, that it has shipped over 600.000 Amiga computers worldwide...". Obwohl diese Zahl recht hoch klingen mag, würden ein paar Geräte mehr der AMIGA-Bewegung sicherlich gut tun.

ALT-BE-KANN-TES

Trotzdem, die Zeit bleibt nicht stehen und ständig klopfen neue Produkte an die Pforte zum Verkaufshimmel, um dort in die ewige Bestsellerliste einzugehen.

GRAFIK

Electronic Arts führt jetzt mit Deluxe Photo Lab und Deluxe Productions zwei neue Produkte in die bekannte Serie ein. Photo Lab ist ein neues Graphik-Programm und gibt dem Benutzer die Möglichkeit Bilder in jeder Auflösung zu malen und bearbeiten. Dies beinhaltet sowohl HAM als auch Extra Halfbrite-Mode. Konvertierungen zwischen den einzelnen Modi sind ebenso leicht möglich, wie die Ausgabe

TV-Meteorologen hat die Be-taversion getestet und benutzt sie jetzt zur Nachwuchsschulung. Die Konkurrenz ACCU-DATA von ACCU-Weather, Inc. wird dadurch zum Informationslieferanten für das neue System degradiert. ACCU-DATA liefert zwar nach wie vor die wesentlichen Daten, aber die graphische Umsetzung mittels Deluxe Productions ist um Klassen besser.

Deluxe Productions ist aber auch eines der Kernstücke des Softwarepakets, das zusammen mit Station Manager von ACS (Associated Computer Services) neuerdings von privaten amerikanischen Fernsehstationen benutzt wird, um jede benötigte Form von animierter Computergraphik zu erstellen und zu verwalten.

wurde es von Jim Kent, Autor des bekannten Aegis Animator. Features sind u.a. Cel, ADO und 3D delta file Animation, traveling masks und lazed compositing. Es ist kompatibel mit LIVE!, Aegis Animator, Videoscape 3D und Aegis Titler.

Zusammenarbeiten mit diesen Neuerscheinungen, aber auch mit den bewährten "Altmeistern" dieses Genres, will 3-DEMON. Mimetics hat hier ein Tool entwickelt, mit dem dreidimensionale Objekte erstellt, gedreht, gestreckt und in praktisch jeder vorstellbaren Art und Weise bearbeitet werden sollen. Die Größe der Objekte ist vom vorhandenen Speicher abhängig.

3D

Eine ganz andere Form der Dreidimensionalität präsentiert Haitex Resources, Inc. X-Specs 3D ist eine Brille mit der - speziell dafür geschriebene - Programme dem Betrachter räumlich erscheinen. Allerdings gibt es bisher kaum entsprechende Software. Das Interface für den Joystick und die Brille kosten zusammen stolze 125 US\$.

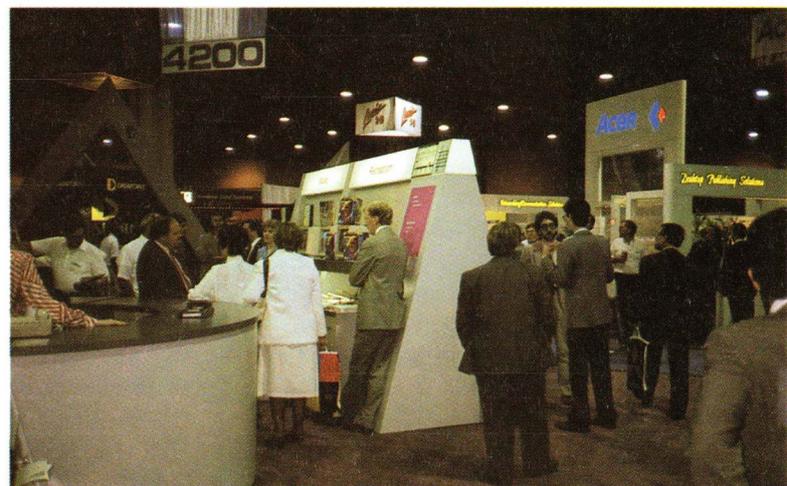
ANI-MATION

Zoetrope von Antic Software ist ebenfalls ein Animations-System. Geschrieben

Desktop Publishing:	Professional Page	Gold Disk
Video Titling:	TV Text und TV Show	Brown & Wagh
Video Production:	Station Manager	ACS
Video Capturing:	Imprint	Liquid Light
	Digi Paint & View	Newtek
Video Storyboarding:	Microfiche Filer	Software Vision
Harddiscs:	Supra Hard Drive	Supra Corporation
3D Animation	Sculpt 3D	Byte by Byte
Animation:	Photon Paint	Microllusions
CAD:	X-CAD	Haitex-Resources
Graphic-Design:	Deluxe Paint II	Electronic Arts
Graphic-Utilities:	Grabbit	Discovery Software
Word-Processing:	Word Perfect	Word Perfect Corp.
MS-DOS Compatibility:	Lotus 1-2-3	Lotus
Office Management:	Business Management	B.E.S.T..
Spreadsheet:	Maxi-Plan	Oxxi Database:
	Superbase	P.D.M.S.
	Frame Grabber	Progr. Peripherals
Education:	Math Package	Mindware
	K-12 Series	Electronic Arts
Music:	Texture	Sound Quest
Recreation:	Arkanoid	Discovery Software
	Jet	SubLOGIC

Dies also sind die derzeitigen "Klassenbesten" aus der Sicht von Commodore. Die meisten davon sind zwar schon seit mehreren Monaten auf dem Markt, aber in ihrer jeweiligen Sparte zählen diese Programme durchaus zum Besten, was es für den Amiga gibt.

von Bildern in einer Größe bis zu ca. 3x3 Metern! Deluxe Productions hingegen wird ein "Grapics-Animation-Tool". Zielgruppe sind Firmen, die animierte Computergraphiken zur Produktpräsentation benötigen. Das Lyndon State College, Amerikas größte Ausbildungsstätte für



DIE NEUE CHIP-GENERATION

Im Gespräch mit einem Entwickler

Geradezu sensationelle Gerüchte erschüttern den zur Zeit doch recht sicheren Hardware-Markt. Während in den meisten Labors und Entwicklungsbüros noch an den 4MBit-Chips auf Silizium-Basis gearbeitet wird, gelang einem kleinen Team aus Ingenieuren DIE Sensation. Eine neue Generation von Chips ist im wahrsten Sinne des Wortes "geboren" ...

An dieser Stelle werden Sie wohl eine der doch schon zu erwartenden Speichermedien, wie z.B. die optischen Speicher oder ähnliches, erwarten. Doch nichts dergleichen.

Der neue Speicher wird in einem hoch geheimen Verfahren aus biologischem Material erstellt. Einer der Entwickler des neuen Verfahrens war nicht nur zu einem Interview bereit, sondern er stellte auch gleichzeitig einen auf der neuen Chip-Art basierenden Amiga 1000 sowie einige kleinere Karten vor. Wir können leider den Namen von Herrn Langholz nicht nennen, da er verständlicherweise Angst hat, aus seiner Heimatstadt Flieden, Fliedener Hauptstr. 19a, entführt zu werden, um das Geheimnis der Chips zu erpressen. Daher nennen wir ihn nur Herrn X.

Kickstart: Herr X, Sie haben in Ihrer Firma Snemeis eine weltbewegende Entdeckung gemacht. Wie kam es eigentlich zu dieser sensationellen Entwicklung?

Herr X: Nun ja, ursprünglich waren wir 5, ich weiß nicht, wie ich es sagen soll, ah ja, Agrarökonomien. Ziel dieses Hobbys ist es ja, wie wohl jedem bekannt, die Menge des erzeugten Guts reziprok abhängig von der Intelligenz, äh, ich meine Arbeit, zu halten. Nun, wie gesagt, man versucht halt, durch ständige Kreuzung und Veredelung den Ertrag zu vergrößern, jeweils mehr oder weniger erfolgreich. Als ich eines Tages also meinen Taschenrechner im Gewächshaus repariere, weil wieder mal die Batterien leer waren, fällt mir doch so ein kleines Teil aus dem Rechner auf den Boden. Leider war der Humus doch so dunkel, daß ich das schwarze Ding nicht wiederfinde. Nun gut, dachte ich mir, was solls, der Rechner hat sowieso nur 5 Mark gekostet, und habe einige Wochen nicht

mehr daran gedacht. Doch eines Morgens betrete ich mein Gewächshaus und sehe wie immer meine Pflanzen durch. Und was sehe ich? Ein kleines Pflänzchen, ca. 10 cm hoch, mit einem silbern glänzenden Stiel und schwarzen Blättern mit weißem, schriftartigem Muster. Da fiel mir natürlich sofort wieder der Vorfall mit dem kleinen Teil aus dem Taschenrechner ein. Ich rufe also ...

Kickstart: Entschuldigen Sie, Herr X, könnten Sie den weiteren Verlauf etwas geraffter beschreiben?

Herr X: Oh, Verzeihung, natürlich. Nach dieser Entdeckung und einem Gespräch mit meinen Freunden haben wir uns natürlich besonders um dies Pflänzchen gekümmert, besonderer Dünger, etc.

Und was meinen Sie, nach 3 Monaten hat dies Pflänzchen doch tatsächlich geblüht! Wir haben die Blüten von Hand bestäubt, und nach weiteren 3 Monaten hatten wir dann unsere Ernte,

die ersten kleinen, wie heißen die doch gleich, ach ja, Schipps.

Kickstart: Chips!

Herr X: Ja, Schipps.

Kickstart: Chips! Na, egal. Doch dies war noch nicht der Weisheit letzter Schluß. Sie fingen jetzt erst richtig mit dem Experimentieren an?

Herr X: Genau, denn von nun an versuchten wir, unsere Schipp-Pflanze mit anderen zu kreuzen. Dies war lange nicht mit Erfolg gekrönt; wir haben alle möglichen Zier- und Nutzpflanzen, die auch nur im entferntesten dieser Pflanze zu ähneln schienen, damit gekreuzt, doch...

Kickstart: Doch...?

Herr X: Nichts! Gar nichts, nicht der geringste Erfolg. Doch unser Team gehört glücklicherweise nicht zu den Leuten, die gleich die Flinte ins Korn werfen. Wir also alle möglichen Pflanzen, die wir noch nicht ausprobiert hatten, herangebracht und gekreuzt.

Und endlich, vor etwa 12 Wochen, der erste Erfolg: mit einer Art Zierkartoffel bekamen wir endlich einige kleine Knöspchen. Doch leider wurde aus diesen nicht alzu viel. Jetzt hatten wir aber einen Anhaltspunkt: Kartoffeln. Langer Rede kurzer Sinn, mit einer exotischen Kartoffelart hatten wir dann Erfolg.

Kickstart: Das ist ja herrlich! Und weiter?



Herr X zeigt uns voller Begeisterung seine hochoptimierte Nährlösung, ohne die die Entwicklung nahezu unmöglich gewesen wäre.

Herr X: Nun ja, wir standen natürlich jetzt erst am Anfang der Entwicklung. Die Früchte mußten ja noch verarbeitet und veredelt werden, um sie überhaupt zu benutzen. Wir besorgten uns also kurzerhand ein Kochbuch, und dort fanden wir eine genaue Herstellungsanweisung. Nur soviel dazu, die Früchte werden geschnitten, einige Tage auf ein feuchtes Medium gelegt, und nachdem sie gekeimt haben, werden sie erhitzt, getrocknet und mit einer chemischen Substanz isoliert und veredelt, ich glaube das Zeug heißt Natriumchlorid oder so.

Kickstart: Einfach toll. Doch sagen Sie, wie stellen Sie denn nun die verschiedenen Typen her?

Herr X: Nun, das ist ganz einfach. Schon ziemlich früh bemerkten wir, daß unser Pflänzchen recht stark auf verschiedene Lichtverhältnisse reagierte.

Wir nehmen also die Masken der Schipps, die wir brauchen, und bestrahlen die Blätter 24 Stunden am Tag damit. Das führt dann zu dem gewünschten Ergebnis.

Kickstart: Bei den Fertigungsmethoden für Rechnerbauteile fällt auf, daß trotz höchster Genauigkeit bei der industriellen Fertigung doch noch recht viel Ausschuß produziert wird. Wie sieht es da bei Ihrer Produktion aus?

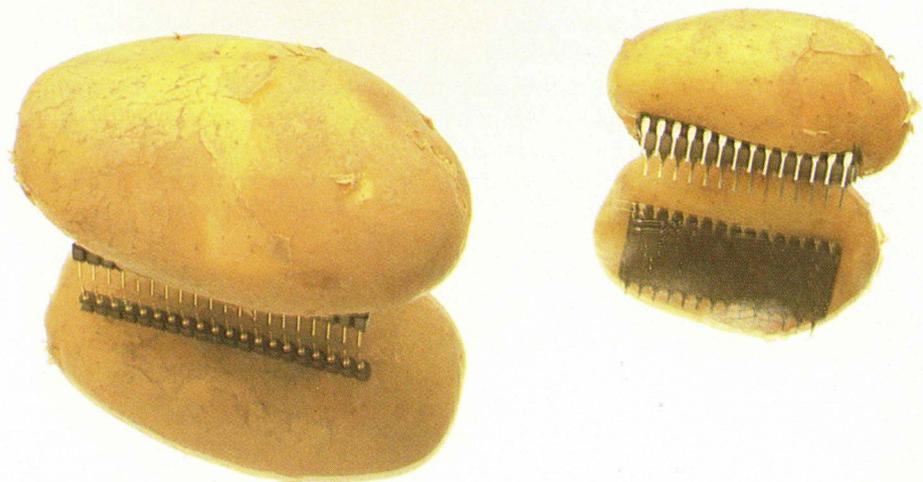
Herr X: Sehr gut. Wir haben einen Ertrag von ca. 99%, d.h. 99 von 100 bestäubten Blüten bringen Früchte von hervorragender Qualität, und das bei einem viel geringeren Mitteleinsatz.

Kickstart: Besitzen ihre Schipps, äh, Chips noch weitere Vorteile?

Herr X: Und wie! Bedingt durch die interne Struktur sind die Schipps in der Lage, bei selber Tickfrequenz ...

Kickstart: Tackt-!

Zwei Prototypen der neuen Schipp-Generation: Links eine CPU mit etwa 1 Tick, voll aufwärtszüchtbar zum 68030, rechts ein noch nicht ganz ausgereifter 57-Bit I/O - Treiber



Herr X: Bitte? Nun, wie ich schon sagte, bei selber Tickfrequenz sind sie etwa 4 bis 10mal schneller. Was das bedeutet, können Sie hier an diesem teilweise umgerüsteten Amiga 1000 sehen. Wir haben hier lediglich den Prozeß ...

Kickstart:-sor!

Herr X: Wie? Na gut. Wir haben hier also den Sor, die Sias und die Kastenschipps ...

Kickstart: CUSTOM!

Herr X:... ersetzt, dazu natürlich noch die Rahms, ...

Kickstart: Ich geb's auf!

Herr X:... und sehen Sie sich nun diese Geschwindigkeitssteigerung an: Sie kennen doch sicher den AmigaBasic-Editor?

Kickstart: Ja.

Herr X: Und sie wissen sicher auch, daß man beim Scrollen sicherheitshalber schon seinen Rasierapparat holen sollte. Nun, in diesem Bio-Amiga, wie wir ihn getauft haben, scrollt der Editor so schnell, daß man schon eine Hochgeschwindigkeitskamera braucht, um es zu verfolgen, geschweige denn eine Stelle auf Anhieb zu finden.

Kickstart: Kaum zu glauben! Doch sagen Sie, was sind jetzt Ihre weiteren Projekte?

Herr X: Nun, wie Sie sicher wissen, werden heute schon oft Ein-Schipp-Computer verwendet, um bei bestimmten Verwendungen den Platzbedarf einer Microcomputersteuerung zu verringern, so auch bei jedem Amiga, wo ein einziger Sor (siehe oben, die Red.) die Tastatur abfragt, decodiert und die Daten zum Rechner schickt. Dasselbe, bloß in größerem Umfang, haben wir vor.

Dieser neue Super-Schipp soll die Leistungsfähigkeit eines Cyper-Rechners in der Größe, nun, also wir schätzen in etwa einer Kartoffel. Sie wird dann eine Busenbreite ...

Kickstart: Bus-Breite!

Herr X: ...von 57 Bit, eine Speicherkapazität von 987654323 Byte und eine Tickfrequenz 1 Hertz haben.

Kickstart: Ist das nicht ein wenig langsam?

Herr X: Aber nein, eben nicht! Und zwar aus dem simplen Grund, daß diese interne Tickfrequenz einer normalen Rechnerfrequenz von 175.123954 Megahertz entspricht. Da gibt es nur noch ein Problem...

Kickstart: Und das ist?

Herr X: Nun, wie Sie wissen ist die Natur perfekt, somit erzeugt der Grafikeil der Kartoffel, der in der Schale angesiedelt ist, ein Farbspektrum von praktisch unendlich vielen Farben, die natürlich kein Bildschirm mehr darstellen kann. Das heißt, daß wir auch noch einen passenden Monitor züchten müssen, was erneut einige Jahre dauern dürfte. Damit wäre man allerdings im Besitz eines Computers, der alle im Moment vorhandenen und in einem Netz verbundenen Cray-Rechner schlagen würde.



Der Ingenieur und der Prototyp des Prozessors. Noch sind nicht alle Möglichkeiten ausgeschöpft, doch zeigt sich jetzt schon, daß aus klimatischen Gründen die japanische Konkurrenz diesmal das Nachsehen haben wird.

Soweit das Interview mit Herrn X von der Firma Snemeis.

Wir von der Kickstart-Redaktion wollten das natürlich genauer wissen. Wir haben uns also den im Bericht erwähnten Amiga-Prototyp ausgeliehen und uns damit auf die Bahamas, äh, Verzeihung, ich meine natürlich in die Redaktion zurückgezogen und alles, was wir so an Programmen da hatten, ausführlich auf Verträglichkeit und Geschwindigkeit getestet. Dabei kamen wir zu folgendem Ergebnis:

Der Umbau-Amiga birgt eine Leistung in sich, die alles bisher dagewesene in den Schatten stellt, darunter auch alle Turbo-Amigas, Transputer oder auch RISC-Co-Prozessoren. Das fing damit an, daß wir eine Diskette in 10 Sekunden kopieren konnten, natürlich von dem einen internen Laufwerk auf drei externe...

Desweiteren probierten wir auch eine Reihe von sowieso schon schnellen Spielen, so zum Beispiel Test Drive, daß durch seine doch recht hohe Geschwindigkeit erstaunt. Und, siehe da, man sieht kaum noch was, weil alles so schnell geht. Selbst der Lattice-Compiler mußte sich der hohen Leistung anpassen, wenn auch nur widerwillig. Immerhin konnte man nun ein Programm von 2K Source in weniger als 5 Minuten compilieren (natürlich ohne Linken...). Doch gab es auch etwas Negatives zu berichten. Eine unserer Disketten enthielt leider ein Virusprogramm. Als wir dieses später entdeckten, wollten wir es natürlich hinauswerfen. Zwar wurde der Bildschirm durch Drücken der linken Maustaste bei Reset grün, doch blieb der Virus weiter im System.

Auch Ausschalten half nichts, der Virus war, wie es auch bei uns Menschen passieren kann, unweigerlich im System, wo er sich in einige der vielen Zellen im Sor, oh, Entschuldigung, Herr X hat abgefärbt..., nun, wie gesagt hat sich der Virus in einigen der vielen Zellen des Prozessors eingenistet und ist von dort nicht mehr zu vertreiben. Soweit zu diesem Bericht. Lesen Sie auch die nächste Folge über die Bio-Schipp, wenn Sie Herrn X sagen hören wollen: "Der Sor, die Rahms, die Sias und die Kasten-Schipp sind in einem riesigen Schipp mit vielen Busen untergebracht."

(Gary Glendown)

MIT BASIC ANS SYSTEM

Teil 5: Auto-Requester & Gadgets

Auch Sie können mit Hilfe der Systemroutinen solche Requester erzeugen. Der einfachste Weg führt über folgenden Befehl:

```
flag& = Autorequest&(window,
bodyText, pos_Text, neg_Text, pos_Flag,
neg_Flag, width, height)
```

window = Zeiger auf die Window Struktur des Windows, in dem der Requester erscheinen soll.
bodyText = Zeiger auf eine IntuiText-Struktur für den Haupttext.
neg_Text = Zeiger auf eine IntuiText-Struktur für den negativen Text.
pos_Text = Zeiger auf eine IntuiText-Struktur für den positiven Text.
pos_Flag = Die IDCMP-Flags, die dem Anklicken des Ja-Feldes gleichkommen.
neg_Flag = Die IDCMP-Flags, die dem Anklicken des Nein-Feldes gleichkommen.
width = Die Breite des Requesters in Bildschirmpixel.
height = Die Höhe des Requesters.

Auto-Requester: Sicher kennen Sie die Nachricht, die Ihnen der Amiga mitteilt, wenn Sie eine andere Diskette ins Laufwerk schieben sollen. In diesem Fall wird ein spezielles Fenster in der oberen linken Ecke des Bildschirms geöffnet, und die erwähnte Nachricht erscheint. Außerdem befinden sich in diesem Fenster noch zwei Auswahlknöpfe mit einem Text, der die Nachricht bejaht und einem Text, der die Nachricht verneint. Bei dieser Art von Fenster handelt es sich um einen Requester.

Nach Beendigung dieser Routine steht in `flag&` der Wert 0, wenn das rechte, und der Wert 1, wenn das linke Auswahlfeld angeklickt wurde. Die IntuiText-Struktur und die IDCMP-Flags haben wir bereits in früheren Folgen besprochen. Wem der Autorequester zu simpel ist, der kann sich seine eigenen Requester erzeugen. Dazu ist die Kenntnis der Requester-Struktur notwendig. Diese können Sie Abb. 1 entnehmen.

Folgende Befehle stehen Ihnen für die Verwaltung von Requestern zur Verfügung:

```
window=BuildSys Request, (window,
bodyText, pos_Text,
neg_Text, flags, width,
height)
```

Die Parameter sind die gleichen wie bei der Routine `AutoRequest`, allerdings werden mit `flags` die IDCMP-Flags des Windows gesetzt.

Nach Rückkehr aus dieser Routine steht in `window` der Zeiger auf die angegebene Window-Struktur. Wird als Parameter bei `Window` der Wert 0 angegeben, so öffnet Intuition selbst ein Window.

Die Requesterstruktur :

Offset	Byte	Bedeutung
0	L	Zeiger auf zuletzt behandelten Requester (wird von Intuition benutzt)
4	W	X-Position des Requesters
6	W	Y-Position des Requesters
8	W	Breite des Requesters
10	W	Höhe des Requesters
12	W	relative X-Position zum Mauszeiger \ siehe Flags
14	W	relative Y-Position zum Mauszeiger \ Wert 1
16	L	Zeiger auf erste Gadget-Struktur
20	L	Zeiger auf Border-Struktur
24	L	Zeiger auf eine Intui-Text-Struktur
28	W	Flags (siehe Tabelle)
30	B	Hintergrundfarbe des Requesters
31	B	damit nächster Wert an gerader Adresse
32	L	Zeiger auf Layer Struktur
36	32 x B	wird von Intuition benutzt
68	L	Zeiger auf eigene Bit Map (0=keine eigene)
72	L	System Variable
76	36 x B	wird von Intuition benutzt

Flags:

Wert	Bedeutung
1	Das Requester wird relativ zur momentanen Position des Mauszeigers gezeichnet.
2	Ist zu setzen, wenn eigene Bitmaps verwendet werden.

Die nächsten Werte werden von Intuition gesetzt:

&H1000	Das Fenster indem das Requester erscheint, ist nicht aktiv
&H2000	Wenn gesetzt, wird Requester benutzt
&H4000	Requester ist ein System Requester

Abb.1

CALL InitRequest (requester)

requester = Zeiger auf die Requester-Struktur.

Diese Routine initialisiert die angegebene Requester-Struktur.

flag& = Request& (requester, window)

requester = Zeiger auf die Requester Struktur.
 window = Zeiger auf die Window-Struktur.
 flag& = 1, wenn alles geklappt hat, sonst 0.

Mit diesem Aufruf wird ein Requester auf dem Bildschirm dargestellt.

flag& = SetDMRequest& (window, requester)

Die Parameter sind dieselben wie bei Request&.

Diese Routine stellt den Requester als Double Menu Requester dar. Solche Requester werden nur dargestellt, wenn die rechte Maustaste zweimal gedrückt wurde.

flag& = ClearDMRequest& (window)

window = Zeiger auf die Window-Struktur, in der sich der DMRequester befindet.

flag& = 1, wenn der Requester gelöscht werden konnte, sonst 0.

Mit diesem Aufruf wird der Double Menu Requester gelöscht.

CALL

EndRequest (requester, window)

requester = Zeiger auf die Requesterstruktur.
 window = Zeiger auf die Windowstruktur.

Mit dieser Routine wird ein Requester gelöscht.

CALL FreeSysRequest (window)

window = Zeiger auf die Windowstruktur.

Diese Routine löscht alle mit BildSysRequest erstellten Requester, die sich in der angegebenen Window-Struktur befinden.

Gadgets

Für die Auswahlfelder werden in der Requesterstruktur die Gadgets benutzt. Ein Gadget ist ein graphisches Objekt, das auf Mausklicks und Mausbewegungen reagiert. Jedes Gadget hat einen abgegrenzten Bereich, in dem ein Mausklick wirkt. Diesen Bereich nennt man Hit-Box. Es gibt vier verschiedene Arten von Gadgets. Das erste sind die sogenannten Systemgadgets. Es handelt sich hierbei um die Symbole zum Vergrößern und Verkleinern eines Fensters oder die Symbole, mit denen man Fenster nach vorne oder hinten legen und schließen kann. Die anderen drei Gadgetarten sind selbstdefinierte Gadgets.

BOOLEAN-Gadgets

Diese Gadgets werden für eine Ja-oder-Nein-Entscheidung herangezogen. Das Gadget kann ein Border oder ein Image sein. Es lassen sich sehr schöne Effekte erzielen, wenn das Gadget aus einem Image besteht und nach dem Anklicken ein anderes Image erscheint.

Proportional Gadgets

Wenn in einem Programm Werte durch Schieberegler eingestellt werden sollen, so benutzen Sie diese Art Gadget. Ein Beispiel sind die Einsteller für die Farbwerte in dem Programm Preferences.

String- oder Integer-Gadgets

Diese Gadgets dienen zum Eingeben von Texten. Sie werden beispielsweise benutzt, wenn Sie im AmigaBasic-Menü die Funktion Open anwählen. Wie sich diese drei Gadgets in einem AmigaBasic-Programm benutzen lassen, entnehmen Sie dem Programm Gadget Demo.

Wie das beim Amiga so üblich ist, benötigen wir mal wieder eine Struktur. In diesem Fall die Gadget-Struktur (Abb. 2). Als erstes steht dort der Zeiger auf eine weitere Gadget-Struktur. Dann folgt die X/Y-Position, an der das Gadget beginnen soll. Als nächstes werden die Breite und Höhe der Hit-Box in die Struktur eingetragen. Dies ist der Bereich, in dem der Mausklick wirken kann. Durch Setzen von verschiedenen Flags (Tab. 1) können Sie bestimmen, ob die X/Y-Positionen relativ oder absolute Werte sein sollen, das Gadget ein Image oder Border ist und was nach dem Anklicken passieren soll. Die Werte für Flags entnehmen Sie der Tabelle 2. Als nächstes tragen Sie in die Gadget-Struktur den Gadget-Typ ein (Tab. 3). Nun folgt ein Zeiger auf eine Image- oder Border-Struktur. Die Image- und die Borderstruktur wurden in Teil 4 erläutert. Eine zweite Image- oder Borderstruktur ist notwendig, wenn Sie bei Eigenheiten der Gadgets den Wert 2 gesetzt haben, also nach dem Anklicken ein anderes Image oder Border erscheinen soll. Soll das Gadget beschriftet werden, so setzen Sie als nächsten Wert einen Zeiger auf eine IntuiText-Struktur. Wenn Sie ein Proportional-, ein String- oder ein Integergadget benutzen wollen, so ist nun der Zeiger auf diese Struktur einzutragen. In GadgetID tragen Sie eine Zahl ein, mit der Sie die Gadgets unterscheiden können. Dies ist wichtig bei der Abfrage der

Die Gadgetstruktur :

Offset	Byte	Bedeutung
0	L	Zeiger auf nächstes Gadget
4	W	X-Position des Gadget
6	W	Y-Position des Gadget
8	W	Breite der Hit Box
10	W	Höhe der Hitbox
12	W	Eigenheiten des Gadget (siehe Tab. 1)
14	W	Flags (siehe Tab. 2)
16	W	Gadget Typ (siehe Tab. 3)
18	L	Zeiger auf Image oder Borderstruktur
22	L	Zeiger auf zweite Image- oder Borderstruktur
26	L	Zeiger auf Textstruktur
30	L	ist auf 0 zu setzen
34	L	Zeiger auf Stringinfo oder Propinfostruktur
38	W	Gadget ID
40	L	User Data (ist auf 0 zu setzen)

Die Stringinfostruktur :

Offset	Byte	Bedeutung
0	L	Zeiger auf Textpuffer
4	L	Zeiger auf Undopuffer
8	W	Cursorposition im Text
10	W	maximale Anzahl der Zeichen
12	W	Ab welchem Zeichen Textausgabe
14	5 x W	wird von Intuition benutzt
24	2 x L	wird von Intuition benutzt
32	L	Zeiger auf Tastaturbelegung (0=Standard)

Die Propinfostruktur :

Offset	Byte	Bedeutung
0	W	Flags (siehe Tab. 4)
2	W	X-Position des Schiebers
4	W	Y-Position des Schiebers
6	W	horizontale Schrittweite
8	W	vertikale Schrittweite
10	6 x W	wird von Intuition benutzt

Abb. 2

Gadgets, damit Sie erkennen können, welches Gadget angewählt wurde. Die StringInfo-Struktur ist ebenfalls in Abb. 2 zu sehen, genau wie die Proportional-Struktur. In der StringInfo-Struktur tragen Sie den Zeiger auf den Textpuffer ein. Hier wird Ihr Text gespeichert. Wenn Sie einen Undopuffer einrichten, so wird nach Drücken der Return-Taste der Text auch in dem Undopuffer gespeichert. Während der Eingabe kann nun mit der rechten Amiga-Taste und "Q" der alte Textinhalt wieder angezeigt werden. Die rechte Amiga-Taste und "X" löschen den Eingabepuffer. Mit der Backspa-

ce-Taste werden die Buchstaben links vom Cursor, mit der Del-Taste diejenigen unter dem Cursor gelöscht. Die Cursor Tasten bewegen den Cursor, und mit Shift und den Cursor Tasten wird der Cursor auf den Anfang bzw. auf das Ende der Eingabezeile gesetzt. Bei der PropInfo-Struktur werden zuerst die Werte für Flags gesetzt. Diese sind der Tab. 4 zu entnehmen. Die X/Y-Position bestimmt den Wert des Schiebers, wenn das Gadget zum ersten Mal dargestellt wird. Außerdem können wir hinterher diesen Wert abfragen, um die Position des Schiebers zu ermitteln. Mit der horizontalen und

Wert	I	Bedeutung
0	I	Gadget wird nach dem Anklicken invertiert
1	I	Gadget wird umrandet
2	I	nach dem Anklicken erscheint ein anderes Image
3	I	keine Reaktion
0	I	Gadget ist ein Border
4	I	Gadget ist Image
0	I	Y-Pos. ist relativ zum oberen Rand
8	I	Y-Pos. ist relativ zum unteren Rand
0	I	X-Pos. ist absoluter Wert
16	I	X-Pos. ist relativer Wert
0	I	Breite der Hit-Box ist absoluter Wert
32	I	Breite der Hit-Box ist relativer Wert
0	I	Höhe der Hit-Box ist absoluter Wert
64	I	Höhe der Hit-Box ist relativer Wert

Tab.1 -Eigenheiten des Gadgets-

Wert	Bedeutung
1	Gadget wird nur aktiv, wenn linker Mausknopf über dem Gadget losgelassen wird.
2	Gadget wird sofort aktiv, wenn es angeklickt wird.
4	wenn Gadget in einem Requester liegt, so wird dieses nach dem Anklicken gelöscht.
8	meldet Mauskoordinaten bei angewählten Gadget.
16	rechts
32	links
64	oben
128	unten
256	wenn Gadget ein Border ist, dann wird die Größe bei Veränderung des Fensters angepaßt, damit es in den Rand hineinpaßt.
512	bei jedem Anklicken des Gadgets wird der Zustand umgedreht. Aus aktiviert wird nicht aktiviert u.s.w.
1024	bei Stringgadget wird der Text mittig ausgegeben. wie vor, jedoch rechtsbündig.
2048	ist keines der beiden Werte gewählt, dann erscheint der Text linksbündig. macht aus einem Stringgadget ein Integergadget

Tab.2 -Flags-

Wert	Bedeutung
1	Boolean-Gadget
4	String oder Integergadget
3	Proportional Gadget

Tab.3 -Gadget Typ-

Wert	Bedeutung
1	Erstellung eines automatischen Schiebers
2	erlaubt horizontale Verschiebung
4	erlaubt vertikale Verschiebung
8	es wird keine Umrandung gezeichnet

Tab.4 -Flags Proportional Gadget-

vertikalen Schrittweite wird angegeben, um wieviel der Schieber sich weiterbewegt, wenn Sie mit der Maus neben dem Schieber klicken. Der gesamte Weg beträgt &HFFFF. Um nun den Weg des Schiebers in 20 Schritte einzuteilen, rechnen Sie:

$$\&HFFFF/20.$$

Nun verfügen Sie über die Informationen, die Sie für die Erstellung von Gadgets benötigen. Studieren Sie auch das Programm Gadget Demo. Dort können Sie ersehen, wie Sie ermitteln, ob ein Gadget angewählt wurde und wenn ja, welches Gadget gewählt wurde. Nun folgen wieder die Erläuterungen zu den Betriebssystemroutinen:

<code>position = AddGadget&(window, gadget, position)</code>
<p>window = Zeiger auf die Window-Struktur</p> <p>gadget = Zeiger auf die Gadget-Struktur</p> <p>position = Die Position in der Gadget-Liste, an der das Gadget eingefügt werden soll.</p>

Mit diesem Aufruf wird das Gadget in die Gadget-Liste des angegebenen Fensters eingefügt.

Soll das Gadget ans Ende gesetzt werden, so ist der Wert -1 einzugeben. Sie erhalten als Wert die Position des Gadgets zurück.

<code>CALL RefreshGadget(gadget, window, requester)</code>
<p>gadget = Zeiger auf die Gadget-Struktur des ersten Gadgets, das dargestellt werden soll.</p> <p>window = Zeiger auf die Window-Struktur, in der sich das Gadget befindet.</p> <p>requester = Zeiger auf die Requester-Struktur, in der sich das Gadget befindet (ist 0, wenn sich das Gadget nicht auf einem Requester befindet).</p>

Diese Routine stellt alle Gadgets der Gadgetliste ab dem angegebenen Gadget auf dem Bildschirm dar.

```
position = RemoveGadget&(window,
    gadget)
```

window = Zeiger auf die Window-Struktur.
gadget = Zeiger auf die Gadget-Struktur.

Hiermit wird das Gadget wieder aus der Gadgetliste entfernt, allerdings nicht vom Bildschirm gelöscht. In position steht die Position, die das Gadget in der Liste hatte.

```
CALL OffGadget(gadget, window,
    requester)
```

gadget = Zeiger auf die Gadget-Struktur.
window = Zeiger auf die Window-Struktur.
requester = Zeiger auf die Requester-Struktur.

Mit diesem Aufruf wird das Gadget deaktiviert, aber nicht aus der Gadgetliste gelöscht.

```
CALL OnGadget(gadget, window,
    requester)
```

gadget = Zeiger auf die Gadget-Struktur.
window = Zeiger auf die Window-Struktur.
requester = Zeiger auf die Requester-Struktur.

Aktiviert ein mit OffGadget abgeschaltetes Gadget wieder.

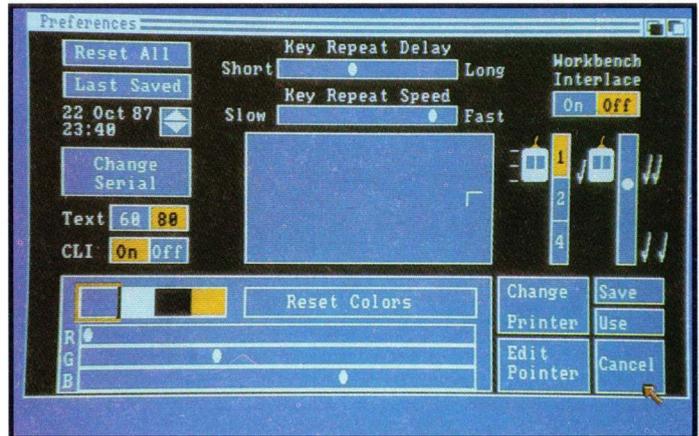
```
CALL ModifyProp(gadget, window,
    requester, flags, hor_Pot,
    vert_Pot, hor_Body, vert_Body)
```

gadget = Zeiger auf die Gadget-Struktur des Proportional-Gadgets.
window = Zeiger auf die Window-Struktur des Windows, in dem sich das Proportional-Gadget befindet.
requester = Zeiger auf eine evtl. Requester-Struktur.
flags = Die neuen Flags für das Proportional-Gadget (siehe Tab. 4).
hor_Pot = Die horizontale Schrittweite.
vert_Pot = Die vertikale Schrittweite.
hor_Body = Die horizontale Position des Schiebers nach Aufruf dieser Routine.
vert_Body = Wie zuvor für die vertikale Position.

Ein typischer Auto-Requester, der vom Betriebssystem ausgegeben wird.



Das Programm Preferences verwendet alle Gadget-Arten.



Diese Routine dient zum Verändern der Propinfo-Struktur.

Zum Schluß dieser Folge wollen wir uns noch etwas mit den Workbench-Routinen beschäftigen. In der Regel werden diese Routinen nicht von einem Programm aus benutzt, da bei der Benutzung der Workbench Intuition das meiste automatisch regelt. Aus diesem Grunde möchte ich Ihnen auch nur zwei Befehle vorstellen, mit denen Sie die Info-Datei eines Programms lesen und wieder speichern können. Diese Routinen stehen in der Icon Library.

```
pointer =
    GetDiskObject&(name)
```

name = Zeiger auf den Filenamen, der mit einem Nullbyte abgeschlossen sein muß.
pointer = Zeiger auf die geladene DiskObject-Struktur.

Mit dieser Routine laden Sie eine DiskObject-Struktur in den Speicher Ihres Computers.

```
status = PutDiskObject&(name, pointer)
name = Zeiger auf den Filenamen, mit einem Nullbyte abgeschlossen.
pointer = Der Zeiger auf die DiskObject-Struktur. In status steht der Wert 0, wenn das Speichern nicht geklappt hat.
```

Diese Routine speichert die DiskObject-Struktur wieder auf Diskette.

Nun können Sie in der DiskObject-Struktur Änderungen vornehmen und diese wieder abspeichern. Auf diese Art und Weise können Sie zum Beispiel ein Icon erstellen, das nach dem Anklicken ein anderes Aussehen annimmt. Dazu müssen Sie als Zwei-Byte-Wert in der DiskObject-Struktur+16 den Wert 6 einsetzen und vom Beginn der Struktur +26 als 4 Byte langen Wert den Zeiger auf die zweite Image-Struktur. Schreiben Sie doch einmal ein Programm, welches diese Änderungen vornimmt.

Das wär's für diese Folge. Beim nächsten und letzten Mal wollen wir uns mit den Disk-Befehlen und den mathematischen Routinen beschäftigen.

```

1:      Gadget Demo
2:      H. Kunz 1988
3:
4: DECLARE FUNCTION AllocMem&      LIBRARY `Routinen, die
5: DECLARE FUNCTION AddGadget&     LIBRARY `einen Wert
6: DECLARE FUNCTION RemoveGadget&  LIBRARY `zurueckgeben
7: DECLARE FUNCTION GetMsg&        LIBRARY `definieren
8:
9: LIBRARY "exec.library"          `Librarys oeffnen
10: LIBRARY "intuition.library"
11:
12: flag&=2^1+2^16
13: speicher&=AllocMem& (440,flag&) `Speicher reservieren
14: IF speicher&=0 THEN ERROR 7
15:
16: gadget1& = speicher&
17: gadget2& = speicher&+44         ` Zeiger der Strukturen
18: gadget3& = speicher&+88
19: stringinfo&= speicher&+132    ` festlegen
20: propinfo& = speicher&+168
21: textpuffer&= speicher&+190
22: undopuffer&= speicher&+280
23: knobpuffer&= speicher&+372
24: borderstr1&= speicher&+360
25: borderdat1&= speicher&+380
26: borderstr2&= speicher&+400
27: borderdat2&= speicher&+420
28:
29: RESTORE gadget1:  CALL struktur (gadget1&)
30: RESTORE gadget2:  CALL struktur (gadget2&)
31:                   ` Strukturen einlesen
32: RESTORE gadget3:  CALL struktur (gadget3&)
33: RESTORE stringinfo:CALL struktur (stringinfo&)
34: RESTORE propinfo: CALL struktur (propinfo&)
35: RESTORE borderstr1:CALL struktur (borderstr1&)
36: RESTORE borderdat1
37:   FOR i=0 TO 9
38:     READ wert:POKEW borderdat1&+i*2,wert
39:   NEXT
40: RESTORE borderstr2:CALL struktur (borderstr2&)
41: RESTORE borderdat2
42:   FOR i=0 TO 9
43:     READ wert:POKEW borderdat2&+i*2,wert
44:   NEXT
45:
46:
47: POKEW gadget1&+18,borderstr1&
48: POKEW gadget2&+18,borderstr2&
49: POKEW gadget2&+34,stringinfo&
50: POKEW gadget3&+34,propinfo&    ` Zeiger in die
51: POKEW gadget3&+18,knobpuffer&
52: POKEW borderstr1&+8,borderdat1& ` Strukturen
53: POKEW borderstr2&+8,borderdat2&
54: POKEW stringinfo&,textpuffer&  ` einsetzen
55: POKEW stringinfo&+4,undopuffer&
56: POKEW gadget1&,gadget2&
57: POKEW gadget2&,gadget3&
58: text$="Kickstart"              ` Text in
59: FOR i=1 TO LEN(text$)
60:   POKE textpuffer&+i-1,ASC(MID$(text$,i,1))
61:   ` Textpuffer setzen
62: NEXT
63:
64: LOCATE 10,1:PRINT "Boolean Gadget"
65: LOCATE 14,1:PRINT "String Gadget"
66: LOCATE 20,1:PRINT "Proportional Gadget"
67:
68: po&=AddGadget&(WINDOW(7),gadget1&,-1)
69:   ` Gadgets in Gadgetliste einbringen
70: po&=AddGadget&(WINDOW(7),gadget2&,-1)
71: po&=AddGadget&(WINDOW(7),gadget3&,-1)
72:
73:
74:
75: RefreshGadgets gadget1&,WINDOW(7),0 `Gadgets zeichnen
76: messkey&=PEEK(WINDOW(7)+94)
77:                   ` Zeiger auf Intui-Message
78: userport&=PEEK(WINDOW(7)+86)
79:
80: warte:
81:
82: signal&=GetMsg&(userport&)    ` auf nachricht warten
83: IF signal&<>messkey& THEN warte

```

```

84:
85: gadstr&=PEEK(WINDOW(7)+28) `Zeiger auf Gadgetstruktur
86: gadid% =PEEK(WINDOW(7)+38) ` GadgetID ermitteln
87: ReplyMsg signal& ` Nachricht an Task zurueckgeben
88:
89: IF gadid%=3 THEN drei
90: IF gadid%=1 THEN aus
91: GOTO warte
92:
93: drei:
94: knobpos$=HEX$(PEEK(WINDOW(7)+2)) `Schieberposition
95: LOCATE 20,42:PRINT knobpos$    ` ermitteln
96:
97: GOTO warte
98:
99: aus:
100: text$=""                      ` Text aus Textpuffer
101: textpos&=textpuffer&
102: WHILE PEEK(textpos&)<>0 ` in die Variable
103:   text$=text$+CHR$(PEEK(textpos&))
104:   textpos&=textpos&+1 ` text$ einlesen
105: WEND
106:
107:
108: po&=RemoveGadget&(WINDOW(7),gadget1&)
109:   ` Gadgets aus Gadgetliste
110:
111: po&=RemoveGadget&(WINDOW(7),gadget2&)
112: po&=RemoveGadget&(WINDOW(7),gadget3&) ` loeschen
113:
114: CALL FreeMem(speicher&,440) ` Speicher freigeben
115: LIBRARY CLOSE              ` Librarys schliessen
116: END
117:
118: SUB struktur (speicher&) STATIC ` Unterprogramm zum
119:   altspeicher&=speicher&      ` Anlegen Strukturen
120:   READ byte$
121:   WHILE byte$<>"e"
122:     READ wert$
123:     IF byte$="b" THEN
124:       POKEW speicher&,VAL(wert$)
125:       speicher&=speicher&+1
126:     END IF
127:     IF byte$="w" THEN
128:       POKEW speicher&,VAL(wert$)
129:       speicher&=speicher&+2
130:     END IF
131:     IF byte$="l" THEN
132:       POKEW speicher&,VAL(wert$)
133:       speicher&=speicher&+4
134:     END IF
135:     READ byte$
136:   WEND
137:   speicher&=altspeicher&
138: END SUB
139:
140: gadget1:      ` Boolean Gadget
141: DATA 1,0     ` Zeiger auf naechstes Gadget
142: DATA w,200,w,40 ` X- und Y- Position
143: DATA w,50,w,50 ` Breite und Hoehe `Hit Box`
144: DATA w,0     ` Gadgeteigenheiten
145: DATA w,1     ` Flags
146: DATA w,1     ` Gadget Typ
147: DATA 1,0,1,0 ` Zeiger auf Borderstruktur
148: DATA 1,0     ` Zeiger auf Textstruktur
149: DATA 1,0,1,0 ` Stringinfostruktur
150: DATA w,1     ` Gadget ID
151: DATA 1,0
152: DATA e
153:
154:
155:
156: gadget2:      ` String Gadget
157: DATA 1,0,w,200,w,100,w,200,w,10,w,0
158: DATA w,0,w,4,1,0,1,0,1,0,1,0,1,0
159: DATA w,2,1,0,e
160:
161: gadget3:      ` Proportional Gadget
162: DATA 1,0,w,200,w,150,w,120,w,20,w,4
163: DATA w,2,w,3,1,0,1,0,1,0,1,0,1,0
164: DATA w,3,1,0,e
165:
166: stringinfo:
167: DATA 1,0,1,0,w,0,w,20,e

```

```

168:
169: propinfo:
170: DATA w,3,w,0,w,0,w,&hfff,e
171:
172: borderstr1:
173: DATA w,0,w,0,b,2,b,0,b,0,b,5,1,0,e
174:
175: borderdat1:
176: DATA 0,0,0,50,50,50,50,0,0,0

```

```

177:
178: borderstr2:
179: DATA w,0,w,0,b,3,b,0,b,0,b,5,1,0,e
180:
181: borderdat2:
182: DATA -4,-4,-4,10,200,10,200,-4,-4,-4

```

Dieses Demoprogramm zeigt, wie man in Basic ein Gadget erzeugen kann

ENDE

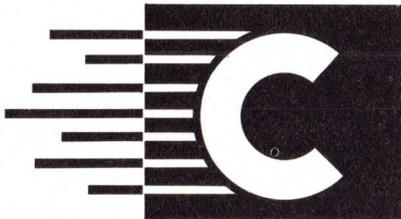
philgerma

NEWS

die neue
Version 4.0
bricht alle
Rekorde



LATTICE



COMPILER

für AMIGA mit
Blink Linker
Symbolic Debugger
Quellen Editor
Utilities
Libraries

Normalversion:
DM 448,-
Entwicklerversion:
DM 798,-

unerreicht schnell

Info-Material
anfordern!

philgerma

Barerstr. 32
8000 München 2
TEL. 089-281228

philgerma

**NEWS
pro**

die künstliche
Intelligenz
für Ihren
AMIGA



log

- schnell: 1800 Lips
- komfortabel:
DEBUG und TRACE
- leistungsfähig:
Edinburg Standard

Demoversion
anfordern!
(DM 15,-)

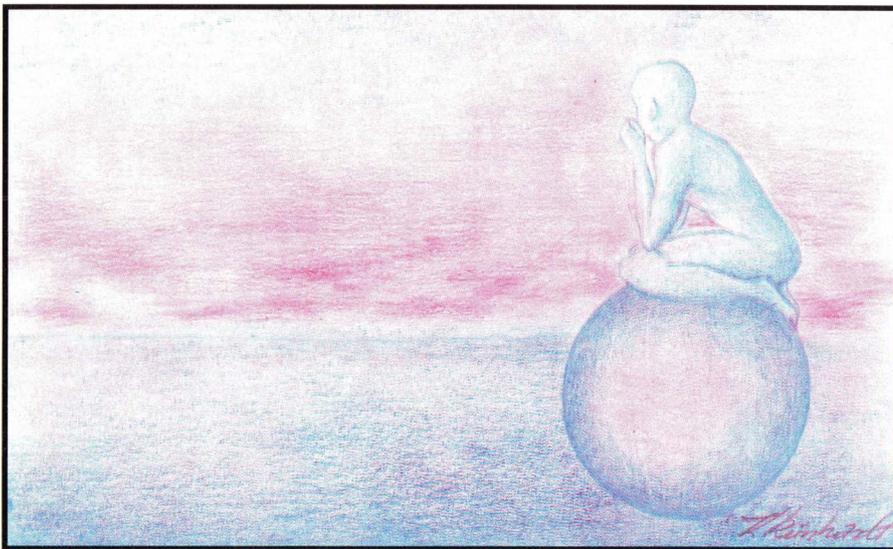
philgerma

Barerstr. 32
8000 München 2
TEL. 089-281228

LISP!

Dialekt der Experten

Teil II



Nachdem in der ersten Folge die Grundbegriffe von LISP abgehandelt wurden, geht es diesmal ans "Eingemachte". Was bisher mit LISP gezeigt wurde, hätte nämlich auch ein besserer Taschenrechner noch gepackt. Zur Erinnerung noch einmal die Aufgaben, jetzt aber mit Auflösung:

```
>(SETQ A 12)
12

>(SETQ B 14)
14

>(SETQ C 20)
20

>(EVAL (SETQ F
  '(EVAL '(SET 'X (AND (< B C) (> C A) (ATOM (* B C)))))))
T
```

Einfache Numerik ist nun aber so gar nicht das typische Anwendungsfeld für LISP. Ein Schlüsselfeld ist dagegen die Symbol- beziehungsweise Listenmanipulation.

Beispiel 1:

```
>(SETQ JUPITERMONDE '(EUROPA IO
  GANYMED KALLISTO))
(EUROPA IO GANYMED KALLISTO)
```

Beachtenswert ist wieder das QUOTE in der Zuweisung, durch dessen Anwesenheit die folgende Liste nicht ausgewertet wird. Zur Bearbeitung solcher Listen gibt es zwei Basisfunktionen in LISP:

Beispiel 2:

```
>(CAR JUPITERMONDE)
EUROPA

>(CDR JUPITERMONDE)
(IO GANYMED KALLISTO)
```

Die Namen der Funktionen entstammen beide dem ursprünglichen Implementationskonzept auf einer IBM-704-Maschine. CAR heißt eigentlich "Contents of Address Register" und liefert das erste Element einer Liste. CDR - bei echten LISPlern nur als KIDDER oder SIDDER ausgesprochen - ist die Abkürzung von "Contents of Decrement Register"; es gibt eine Liste ohne ihr erstes Element zurück. In der Regel ist der Rückgabewert von CDR also eine Liste.

Beispiel 3:

```
>(CDR '( A ))
NIL
```

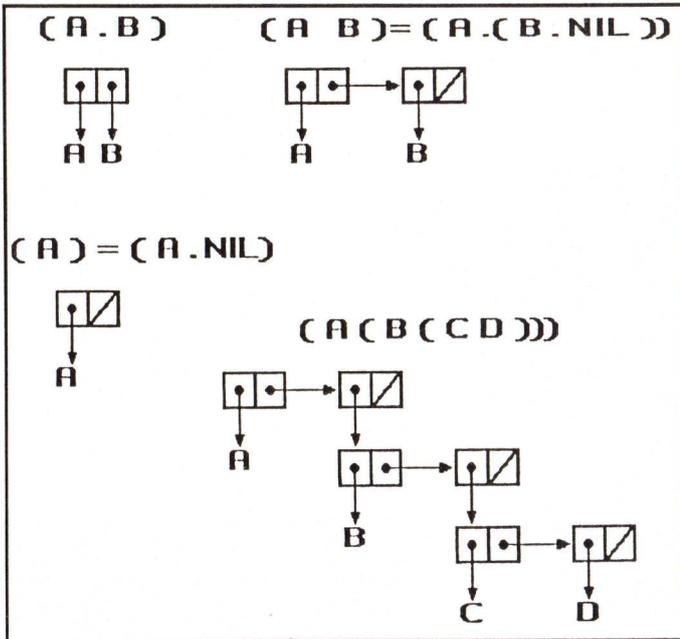


Abb 1:
Die BOX-and-
POINTER Notation hilft das Konzept zur Speicherung von Listen zu verstehen.

In diesem Beispiel scheint die Antwort der Funktion CDR keine Liste zu sein. Um zu verstehen, warum der Rückgabewert dieser Operation dennoch korrekt ist, muß man sich daran erinnern, daß NIL und () equivalent sind. Dann ist die Liste (A) ohne ihr erstes Element gleich (), der leeren Menge, welche gleich NIL ist. Übrigens weist die leere Menge eine LISP-Besonderheit auf: Sie ist gleichzeitig Atom und Liste.

Beispiel 4:

```
>(ATOM ())
T

>(LISTP ())
T

>(LISTP 'A)
NIL
```

Die Prädikatsfunktion LISTP, hier zum erstenmal erwähnt, zeigt, ob es sich bei dem ihr übergebenen Parameter um eine Liste handelt. Anhand der leeren Menge kann man auch noch einmal gut zeigen, wie bedeutungsvoll Klammern für das LISP-System sind: () und (()) etwa repräsentieren verschiedene Werte. Der erste Ausdruck hat den Wert NIL, während der zweite für die Liste mit dem Element NIL steht. Diese Tatsache darf man vor allem dann nie vergessen, wenn man schon mit anderen Programmiersprachen gearbeitet hat, bei denen die Klammern ausschließ-

lich die Bewertungsreihenfolge eines Ausdrucks anzeigen.

Noch einmal zurück zu Beispiel 1. Astronomieinteressierte haben dabei vielleicht gleich bemerkt, daß in der aufgestellten Liste nur die vier größten Jupitermonde erwähnt wurden, die schon Galileo Galilei kannte. Dieser Riesenplanet hat aber mindestens 16 Begleiter. Die Liste soll nun um einen weiteren Mond vergrößert werden:

Beispiel 5:

```
>(CONS 'AMALTEA JUPITERMONDE)
(AMALTEA EUROPA IO GANYMED CALLISTO)
```

CONS fügt einer Liste ein neues Anfangselement an. Diese Aussage ist allerdings nicht generell richtig. Ihre Korrektur führt zu einem weiteren Exkurs: Der Verwaltung von Listenstrukturen durch LISP. Der zweite Parameter von CONS muß nämlich nicht immer eine Liste sein.

Beispiel 6:

```
>(SETQ DOTTED-PAIR (CONS 'A 'B))
(A . B)

>(CAR DOTTED-PAIR)
A

>(CDR DOTTED-PAIR)
B
```

Der CONS-Funktion - deren Name übrigens die Abkürzung für CON-

Struct ist - werden in diesem Beispiel zwei Atome übergeben. Das Ergebnis ist ein Dotted Pair (gepunktetes Paar), das Grundelement der Listenspeicherung. Abbildung 1 versucht, diesen Sachverhalt grafisch zu veranschaulichen. Die verwendete Darstellungsart nennt man Box-and-Pointer (Kasten und Zeiger)-Repräsentation. Jede Box besteht aus zwei Zeigern, einem auf das jeweilige Element (dem CAR des Ausdrucks) und einem anderen, der auf die restlichen Elemente der Liste zeigt (dem CDR des Ausdrucks). Ein Dotted Pair ist demnach eine Liste mit einem Element beziehungsweise einer Box. In einer "normalen" Liste wird das Ende durch NIL repräsentiert, was in der Grafik mit einem Querstrich notiert wird. Am Beispiel von (A), was man genauso gut als (A.NIL) schreiben könnte, ist zu sehen, daß sich eigentlich jede Liste aus einem Dotted Pair zusammensetzt.

Beispiel 7:

```
>(SETQ PUNKT1 (10 . 70))
PUNKT1

>(SETQ PUNKT2 (310 . 190))
PUNKT2

>(SETQ LINIE (PUNKT1 . PUNKT2))
LINIE

(DRAW LINE)
```

Dieses Beispiel läßt sich allerdings nicht bis zum Ende fehlerfrei durchführen, da es in XLISP keine Funktion DRAW gibt. Es zeigt dafür aber, wozu sich die Dotted Pairs verwenden lassen. Man kann damit ideal Datenobjekte abbilden, die sich aus zwei Elementen zusammensetzen. Im Beispiel sind dies Punktpaare und Linien eines zweidimensionalen Koordinatensystems, wie etwa dem des Bildschirms. Die Technik der Listemverwaltung wird noch einmal am letzten Beispiel der Abbildung 1 deutlich. An (A (B (C D))) sieht man, daß jedes Listenobjekt seinen eigenen Knoten besitzt, der Zeiger auf das CAR und das CDR des Objekts enthält. Durch die Grafik kommt vielleicht auch eine weitere Eigenschaft eher zum Vorschein, die in der normalen Schreibweise nicht so deutlich zu erkennen ist. Aufgrund der Möglichkeit, Listen verschachteln zu

können, lassen sich Hierarchien beziehungsweise baumartige Strukturen abbilden. In einer zweiten Darstellung von (A (B (C D))), der Abbildung 2, ist der Baum auf Anhieb zu sehen.

Die Erstellung solcher Gebilde macht natürlich ohne die passende Möglichkeit zur Auswertung wenig Sinn. Hierfür gibt es aber das inzwischen schon häufig erwähnte CAR-CDR-Funktionspaar. Wie isoliert man damit aber zum Beispiel aus dem Ausdruck (A (B (C D))) das Symbol D?

Beispiel 8:

```
>(SETQ L '(A (B (C D))))
L
>(CAR (CDR (CAR (CDR (CAR (CDR L))))))
D
>(CADR (CADADR L))
D
```

Der untere Ausdruck macht sich eine Vereinfachung zunutze, die die schnell anwachsende Komplexität von CAR-CDR-Ausdrücken durch Übersichtlichkeit etwas reduzieren soll. Die Funktionen heißen CXXR, CXXXR und CXXXXR, wobei X jeweils für A oder D steht und dementsprechend ein CAR oder ein CDR ausführt. Trotz dieser Vereinfachung wirkt es auf den ersten Blick merkwürdig, daß, um drei Verschachtelungen zu durchdringen, sechs Operationen notwendig sind. Des Rätsels Lösung ergibt sich durch einen Blick auf Abbildung 1. Jede CAR-CDR-Operation läßt sich nämlich einer Box zuordnen. Von der ersten Box wird der Zeiger auf ihr CDR benötigt. Von der zweiten braucht man das CAR, denn das CDR zeigt ja auf NIL und immer so weiter.

Beispiel 9:

```
>(CAR L)
(B (C D))
>(CDAR L)
B (C D)
>(CDADR L) (C D)
```

Stufe für Stufe durchgeführt lassen sich die Befehle leichter nachvollziehen.

Abschließend zum CAR-CDR-CONS-Komplex noch ein paar Beispiele.

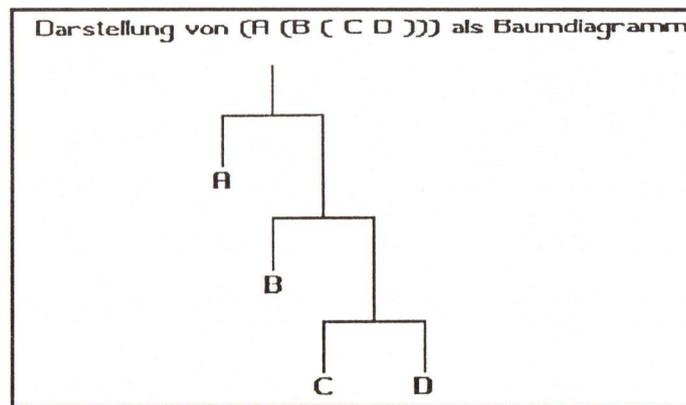


Abb 2:
Bäume wachsen auch in LISP.

Beispiel 10:

```
>(CONS 'VIELE (CDR '(ALLE KOE-
ECHE VERDERBEN DEN BREI))))
VIELE KOECHE VERDERBEN DEN
BREI
>(CAR '((A B) (C D)))
(A B)
>(CDR '((A B) (C D)))
((C D))
>(CAR (CDR '((A B) (C D))))
(C D)
>(CDR (CAR (CDR '((A B) (D))))))
(D)
>(CAR (CDR (CAR '((A B) (C D))))))
B
```

Vielleicht fragen sich inzwischen einige Leser, warum LISP eigentlich Programmiersprache heißt; bis jetzt ward noch kein Programm gesehen. Das liegt aber ausschließlich am Erzähler, der diesen Teil bis jetzt zurückhielt. Nun also zur längst überfälligen Definition von eigenen Funktionen. Eigentlich hätte man diesen Komplex schon mit der ersten Erwähnung von eingebauten Funktionen behandeln können (oder sollen?), da benutzerdefinierte Funktionen sich genauso wie eingebauten in die LISP-Umgebung einfügen. Aber es gab ja noch nichts, was diese Funktionen hätten machen können. Normalerweise schreibt man Programme, um sich stellende Probleme zu lösen. Hier wird es einmal andersherum gemacht. Es wird ein passendes Problem gesucht, um eine Funktion dafür schreiben zu können, die an das bereits Gelehrte anschließt.

Beispiel 11:

```
>(SETQ QUADRATE '(1 4 9 16 25 36))
(1 4 9 16 25 36)
>(MEMBER (* 4 4) QUADRATE)
(16 25 36)
>(MEMBER 8 QUADRATE)
NIL
```

Die eingebaute Funktion MEMBER zeigt an, ob ein Ausdruck Element einer Liste ist. Findet sich das gesuchte Element nicht in der Liste, so ist die Antwort NIL. Weist die Liste das Suchobjekt jedoch auf, so ist das Ergebnis eine Liste, die aus der Eingangsliste ab dem ersten Auftreten des gesuchten Elements besteht. Der Logik hätte es sicher mehr entsprochen, nur T zurückzugeben; man hat hier jedoch als Konvention festgelegt: Wenn schon etwas zurückkommt, was nicht NIL ist, so sollte es etwas sinnvoll Verwendbares sein. Was aber, wenn die Funktion nicht eingebaut wäre und man sie dennoch benötigte? Hier ist die Lösung: Beispiel 12:

```
(DEFUN ELEMENT (SUCHOBJEKT
LISTE)
(COND ((EQUAL '() LISTE) NIL)
((EQUAL SUCHOBJEKT (CARLISTE))
LISTE)
(T (ELEMENT SUCHOBJEKT (CDR
LISTE)))))
```

Dieser Ausdruck enthält drei neue Funktionen. DEFUN, COND und EQUAL. DEFUN ist eine Abkürzung für Define Function und stellt das Grundelement der eigentlichen Programmierwerkzeuge von LISP dar. DEFUN ist eine Funktion mit Neben-

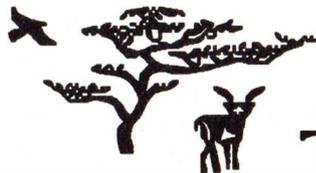
KAMPF DEN TREPPEN MIT LOGO-Printer

* Nutzen Sie alle Schriftarten in NTQ *
Erstellen Sie Ihre eigenen LOGOS

Verarbeitet werden 1 f f-Bilder, sie werden auf die 6-fache Auflösung des Druckers umgerechnet.
NEC P5, P6, P7, P2200 und kompatible Drucker

Diese Anzeige ist mit LOGO-Printer gedruckt und im Maßstab 1:1 original abgebildet.
Alle Formate von DIN A6 - DIN A2 möglich

Vertrieb:
Ing. Büro Reck
Infanterieweg 23
2900 Oldenburg



Einführungspreis
DM 98,-
Versand per Nachnahme
Tel. 0441 882847

★ ★ AUTOREN GESUCHT

Sie

- ... haben eine gute Programmidee
- ... wollen ein Buch schreiben
- ... kennen eine Menge Tips u. Tricks
- ... möchten Ihre Erfahrungen weitergeben

Wir

- ... bieten Ihnen unsere Erfahrung
- ... unterstützen Ihre Ideen
- ... sind ein leistungsstarker Verlag
- ... freuen uns von Ihnen zu hören

Buch

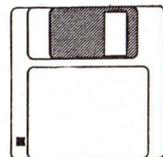
AMIGA



Heim Verlag

+

Programm



Schreiben Sie uns

Heim-Verlag
Kennwort: Autor
Heidelberger Landstr. 194
6100 Da.-Eberstadt
Tel.: 06151/56057

effekt. So wie SETQ den Nebeneffekt hat, die LISP-Umgebung um ein Symbol zu erweitern, macht es DEFUN mit einer dort beschriebenen Funktion. LISP kennt nur diese Funktionen, um Prozeßbeschreibungen identifizieren zu können. Die meisten anderen Programmiersprachen führen dagegen noch die Kategorie Programm in ihrem Sprachschatz. In LISP ist diese jedoch nicht notwendig, ja sie würde sogar die Linearität des Sprachaufbaus stören. Programme sind schließlich auch nichts anderes als Funktionen, beziehungsweise Prozeduren, nur, daß sie eben den Hauptprozeß steuern.

Beispiel 12 kann man, wie alle anderen, direkt in den LISP-Interpreter eingeben, also auch mehrzeilig. Die READ-PRINT-EVAL-Schleife wertet den gesamten Ausdruck erst dann aus, wenn alle Klammerebenen wieder geschlossen sind. Allerdings ist alles wieder hoffnungslos verloren, wenn man XLISP durch (EXIT) verläßt. Man kann LISP-Ausdrücke aber auch permanent machen, indem man sie einfach mittels eines Editors in eine Datei schreibt und diese dann mit (LOAD Dateiname) der LISP-Umgebung bekannt macht. LOAD aktiviert die READ-PRINT-EVAL Schleife, wobei das READ von der angegeben Datei anstatt wie sonst von der Konsole kommt.

Doch zurück zu DEFUN. DEFUN hat drei Argumente:

1. Funktionsname
2. Funktionsparameter
3. Funktionskörper

Der Funktionsname ist in diesem Beispiel ELEMENT, die Funktionsparameter sind SUCHOBJEKT und LISTE, und der gesamte Rest ist die eigentliche Funktion. Diese besteht selbst wiederum aus einem einzigen Funktionsaufruf, und zwar dem der Funktion COND. COND läßt sich mit dem IF-THEN-ELSE Befehl anderer Programmiersprachen vergleichen. Die Befehlsstruktur von COND wird wie folgt formalisiert:

```
(COND ([P1] [A1])
      ([P2] [A2])
      ([P3] [A3])
      :
      :
      ([Pn] [An]))
```

Die Notation 1,2,3 bis n signalisiert dabei, daß COND beliebig viele Parameter übergeben werden können. Jeder Parameter besteht seinerseits aus einem Ausdruckspaar. Der erste Teil ist das sogenannte Prädikat, ein Ausdruck, der immer zu wahr oder falsch (in LISP durch T und NIL dargestellt) ausgewertet wird. Zu dem ersten Prädikat Px, welches als wahr ausgewertet wird, führt COND den zugehörigen Ergebnisausdruck Ax aus. Der Rückgabewert dieses Ausdrucks ist auch der von COND, das mit der Beendigung der Ausführung von Ax auch selbst terminiert. COND entspricht also einer IF-THEN-ELSEIF-THEN-Kette beliebiger Länge. Der erste Teil eines jeden

Parameters ist dann das IF, während der THEN-Teil mit dem zweiten Element des Parameters gleichzusetzen ist. Sollten alle Prädikatsausdrücke zu NIL ausgewertet werden, ist das Ergebnis von COND ebenfalls NIL.

EQUAL, die dritte neue Systemfunktion, ist in COND versteckt. Es ist eine Prädikatsfunktion, die prüft, ob der Wert zweier Ausdrücke gleich ist.

Beispiel 13:

```
> (EQUAL 1 1)
T
> (EQUAL 2 4)
NIL
```

Damit sind die wichtigsten Werkzeuge bekannt, und man kann sich der Funktionsweise von ELEMENT selbst zuwenden. Schlüssel hierzu ist die COND-Anweisung. Das erste Parameterpaar ((EQUAL '() LISTE) NIL) prüft, ob die Eingangsliste überhaupt Elemente enthält. Ist LISTE gleich der leeren Liste, so wird NIL zurückgegeben und COND sowie die ganze Funktion ELEMENT augenblicklich beendet. Der nächste Parameter von COND prüft nun, ob das erste Element von LISTE, also (CAR LISTE), gleich dem gesuchten Objekt ist. In diesem Fall wird die Funktion damit verlassen, daß die gesamte LISTE zurückgegeben wird. Der Vorgang des Zurückgebens ist übrigens eine automatische Sache. Eine Funktion gibt immer den Wert des letzten von ihr ausgewerteten Ausdrucks zurück. Wird kein Ausdruck ausgewertet, wie zum Beispiel bei einer leeren Funktion, ist der Rückgabewert NIL. Unter diesen Voraussetzungen läßt sich auch eher verstehen, warum es in LISP nur Funktionen und keine Prozeduren gibt. Allerdings werden in manchen Lehrbüchern Funktionen mit Nebeneffekt wie etwa SETQ als Prozeduren bezeichnet.

So weit, so gut; diese Konstruktion könnte bis jetzt jedoch nur Listen mit einem oder gar keinem Element erfolgreich bearbeiten. Bewegung kommt erst durch den letzten COND-Parameter ins Spiel. Dabei fällt auf, daß, im Gegensatz zu den anderen Parametern, statt einem Ausdruck einfach ein T im Feld des Prädikats steht. Dieses Prädikat ist immer wahr und bildet damit eine einfache ELSE-Klausel nach.

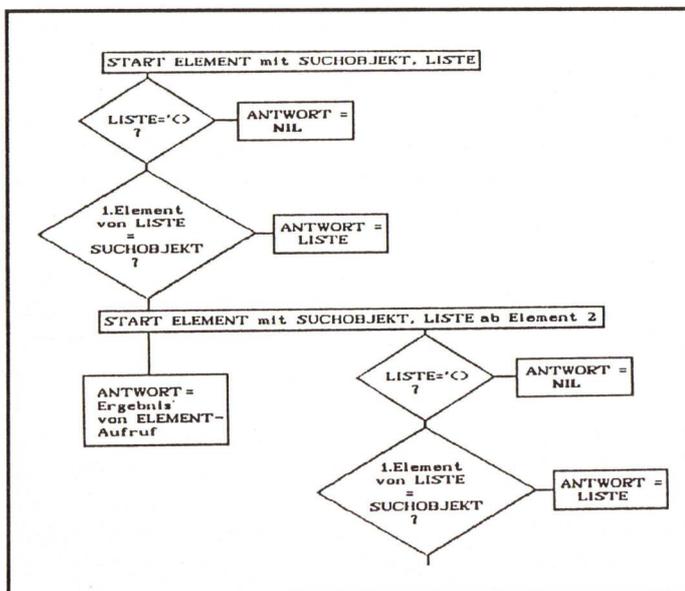


Abb 3:
Der PAP zeigt den symbolischen Ablauf einer Funktion.

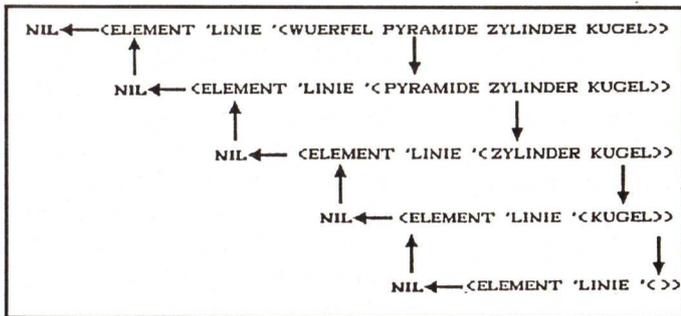


Abb 4:
Rekursion ist ein
Schrumpfungs-
und Wachstums-
prozeß.

Sind nämlich alle Prädikate der vorhergehenden Parameter falsch (NIL), so wird dieser als letzter stehende Ausdruck ausgeführt. Wichtig ist, daß diese "rest der Welt"-Klausel wirklich die letzte ist, da alle ihr folgenden Parameter unter keinen Umständen ausgeführt würden. Und was wird nun ausgeführt? Falls die ersten beiden COND-Prädikate zu NIL evaluiert wurden, wird die Funktion ELEMENT ausgeführt. Moment mal! Das würde ja heißen, daß die Funktion ELEMENT sich selbst ausführt? Genau das heißt es auch. Die Tatsache, daß die ersten beiden Klauseln falsch waren, bedeutet in Worten ausgedrückt, daß LISTE weder leer ist, noch daß ihr erstes Element dem gesuchten Objekt entspricht. Dieser Feststellung zufolge sollte die nächste Aktion logischerweise aus der Untersuchung des zweiten Elements von Liste bestehen. Der Trick dabei ist, die Liste abzüglich des ersten Elements, also (CDR LISTE), erneut durch die Funktion ELEMENT untersuchen zu lassen. Diese schaut wieder nach dem ersten Element, welches jetzt ja eigentlich das zweite der Ursprungsliste ist und ruft im Falle von Nichtübereinstimmung wieder sich selbst auf. Im Maximum ist die Anzahl der Aufrufe von ELEMENT (inklusive dem Startaufruf von "außen") gleich der Anzahl der Listenelemente. Dieses Maximum wird genau dann erreicht, wenn das fragliche Objekt nicht Element der durchsuchten Liste ist.

Puhh! Hassen Sie auch langwierige verbale Beschreibungen von Algorithmen? Das war nämlich eine, auf die jedoch leider nicht verzichtet werden konnte. Sich selbst aufrufende Funktionen (Rekursion) sind nämlich eine wichtige Technik der Programmierung in LISP, beziehungsweise der Programmierung in modernen Programmiersprachen überhaupt. Um sich auch auf andere Weise zu verdeutlichen, wie

ELEMENT funktioniert, sollte man sich einmal die Abbildungen 3 und 4 anschauen. Abbildung 3 zeigt in einer Art symbolischem Programmablaufplan (PAP), was während des ersten Aufrufs geschieht. Romben stehen in dieser Notation für Verzweigungen, während Rechtecke Anweisungen oder Funktionsaufrufe darstellen. Besonders wichtig ist bei dieser Abbildung der letzte Kasten in der Hauptfunktion ELEMENT. Er beschreibt den Rückgabewert der Funktion, ohne den die ganze Sache nicht funktionieren könnte. Abbildung 4 zeigt exemplarisch, was bei einem konkreten Aufruf hin- und zurückgegeben wird. Jedesmal, wenn das erste Element der untersuchten Liste nicht dem gesuchten Objekt entspricht, wird die Suchliste um ein Element kleiner. Der letzte Aufruf übergibt nur noch eine leere Liste. Diese Funktion endet dann mit der Antwort NIL, welches durch alle vorhergehenden Ebenen hindurch nach "oben" zurückgegeben wird.

Möglicherweise wird damit etwas deutlicher, was mit Rekursion gemeint ist. LISP-Schülern, die mit "klassischen" Programmiersprachen vorbelastet sind, fällt es oft schwer, sich an diesem Programmierstil zu gewöhnen. Nichtsdestotrotz zieht sich die Rekursion als dicker roter Faden durch die ganze LISP-Philosophie. In diesem Zusammenhang hat sich über die Jahre hinweg ein erbitterter Religionskrieg zwischen den "Iterationisten" und den "Rekursionisten" entwickelt, dessen Fronten so verhärtet sind, daß man ihn nur noch mit dem Konflikt zwischen Amiga- und ST-Fans vergleichen kann. Die Iterationisten sind die Anhänger der Schleifenprogrammierung, die in den meisten Programmiersprachen mit FOR- oder WHILE-Anweisungen bewerkstelligt wird. Mit riesigen FORs auf ihre wehenden Fahnen geschrieben, werfen sie den Rekursio-

nisten vor, langsame und speicherineffiziente Programme zu erstellen. Und dabei haben sie in der Regel sogar recht. Tatsächlich benötigt das häufige Aufrufen einer Funktion recht viel Speicher, da jedesmal mindestens die Rücksprungadresse und die lokalen Variablen auf dem Stack abgelegt werden müssen. Was die Geschwindigkeit von rekursiven im Gegensatz zu iterativen Prozessen angeht, sieht es für die Rekursion meistens auch schlecht aus, da die Schleifenprogrammierung oft direkt vom Prozessorbefehlssatz unterstützt wird. Bleiben den Rekursionisten da überhaupt noch Rückzugsargumente übrig? Sicher, aber um sie verstehen zu können muß man ein bisschen mehr über Rekursion und Iteration wissen.

Beispiel 14:

```
>(DEFUN FIBONACCI (N)
  (COND ((EQUAL N 0) 0)
        ((EQUAL N 1) 1)
        (T (+ (FIBONACCI (- N 1))
              (FIBONACCI (- N 2))))))
```

Rekursion ist ein anerkanntes mathematisches Beweisverfahren. Es bezieht sich auf Objekte beziehungsweise Prozesse, die zu ihrer Definition Gebrauch von sich selbst machen (z. B. $X[t]=X[t-1]+5$). Für ein Beispiel einer angewandten Rekursion in LISP siehe übrigens auch [6]. An diesem Punkt muß man aber eine haarfeine Unterscheidung machen. Zu unterscheiden ist zwischen der Notation rekursiver Prozesse und der Notation rekursiver Prozeduren. Ein rekursiver Prozeß sind zum Beispiel die Fibonacci-Zahlen. Deren Definition lautet: $fib(n)=1$ für $n=1$ oder $n=0$; ansonsten $fib(n)=fib(n-1)+fib(n-2)$. Die zugehörige LISP-Funktion in Beispiel 14 ist ebenfalls rekursiv implementiert. Dagegen ist etwa der der Funktion ELEMENT zugrunde liegende Prozess iterativer Natur. Aber was genau ist denn nun iterativ? Im Allgemeinen läßt sich ein iterativer Prozeß durch eine bestimmte Anzahl von Zustandsvariablen, einer Regel, die besagt, wie die Zustandsvariablen im jedem Prozeßstadium zu verändern sind, und eventuell einer Abbruchbedingung beschreiben [3]. Im ELEMENT-Beispiel ist die Zustandsvariable die zu durchsuchende Liste mit ihren Initialwert. Die Regel heißt,

daß Liste nach jedem Durchlauf um das erste Element zu reduzieren ist. Und die Abbruchsbedingung (das kennen wir ja alles schon) lautet, daß die Liste entweder leer sein muß oder ihr erstes Element dem gesuchten zu entsprechen hat. Und nun noch einmal der Unterschied zur Rekursion. Beim iterativen Prozeß greifen nur die Zustandsvariablen auf ihren früheren Wert zurück (z.B. (SETQ LISTE (CAR LISTE))); bei der Rekursion ruft der Prozeß eine vollständige Kopie von sich selbst auf (z. B. FIBONACCI(4)=FIBONACCI(3)+FIBONACCI(2)). Wenn man, wie bei Element geschehen, offensichtlich einen iterativen Prozeß rekursiv implementieren kann, geht es dann nicht auch umgekehrt? In der Tat läßt sich jeder linear endrekursive Prozess (wie etwa die Funktion ELEMENT) in einen linear iterativen umsetzen. Diese Tatsache ist gleichzeitig auch die stärkste Waffe der Rekursionisten in ihrem Wissensstreit mit den Iterationisten. Das Argument lautet dann meistens: Wenn sich iterative Prozesse auf vorgegebener Hardware effizienter verhalten als rekursive, die einen sich aber in die anderen umsetzen lassen, dann ist es Sache des jeweiligen Sprachwerkzeugs (Compiler, Interpreter), austauschbare Verfahren zu erkennen und die effizienteste Lösung zu implementieren. Mithin sind nicht die rekursiven Programme ineffizient, sondern nicht-rekursive Interpreter und Compiler. In der symbolischen Datenverarbeitung wird man die rekursive der iterativen Problemabstraktion immer vorziehen, da sie sich mit den Regeln der bekannten Mathematik stets besser formalisieren läßt als iterative Abstraktion. Immerhin kommt die Rekursion ausschließlich mit dem Funktionsaufrufmechanismus der Sprache aus und benötigt keine speziellen Schleifenkonstrukte wie die Iteration. Das führt die Autoren von [3] sogar zu der Behauptung, daß Befehlsgruppen wie REPEAT UNTIL, DO WHILE und FOR für einen endrekursiven Sprachübersetzer völlig überflüssiges Sprachkonfekt (syntactic sugar) seien.

Für solche, die nun nicht ganz so weit wollen, hat LISP aber einen Trost übrig. Der DO-

Befehl läßt die direkte Umsetzung von iterativen Prozeßen zu. In Beispiel 15 ist die Fakultätsfunktion einmal iterativ mit DO und einmal rekursiv umgesetzt. Die Fakultät $n!$ ist ja bekanntlich das Produkt aller ganzen Zahlen von n bis 1, also zum Beispiel $5! = 5*4*3*2*1=120$.

Beispiel 15:

```
>(DEFUN FAKULTAET (N)
  (DO ((ERGEBNIS 1)
      (ZAEHLER N) (( < ZAEHLER
2) ERGEBNIS)
      (SETQ ERGEBNIS (* ERGEBNIS
ZAEHLER))
      (SETQ ZAEHLER (- ZAEHLER
1))))))
>(DEFUN FAKULTAET (N)
  (COND ((EQUAL N 1) 1)
        (T (* N (FAKULTAET (-
1))))))
```

Mit dieser auszehrenden, wenn auch nicht erschöpfenden Diskussion von Repititionsverfahren soll der Deckel über dieser kleinen LISP-Fibel auch langsam geschlossen werden. Leider konnten in der Kürze nur Ansätze einiger wichtiger Konzepte von LISP dargelegt werden. Unter den noch nicht einmal angeschnittenen Themen finden sich Begriffe wie: Datenabstraktion, Lambdaotation, objektgerichtete Programmierung, oder wie man LISP in LISP realisiert (wieder mal eine Rekursion). Wer LISP schon kennt und diesen Bericht trotzdem liest, wird wissen, was dieser Verlust bedeutet. Allen anderen sollte zumindest klar geworden sein, daß LISP anders als die meisten Programmiersprachen ist. Aus diesem Grund könnte allerdings auch leicht der Eindruck entstehen, LISP sei ein wenig beachteter Exot. Dem ist nun

aber gar nicht so. LISP ist noch immer die erste Wahl unter der Programmiersprachen für Anwendungen, die unter das Gebiet der künstlichen Intelligenz fallen. Einige konkrete Problembereiche, in denen es als Werkzeug dominiert, sind: Verarbeitung natürlicher Sprache, Konstruktion und Umsetzung von Expertensystemen, Mustererkennung und Verarbeitung. Im mehr theoretischen Teil des AI-Gebiets sind Lösungen in den Bereichen Suchstrategien, Entscheidungstheorie, symbolische Algebra und Compiler- beziehungsweise Interpreterbau oft in LISP erstellt. Der einzige ernst zu nehmende Konkurrent der letzten Jahre, die Programmiersprache PROLOG, ist übrigens ein Subset von LISP. Das heißt, die meisten größeren LISP-Implementationen weisen PROLOG als einfaches Programm auf.

Will man LISP im Vergleich mit anderen Programmiersprachen sehen, sollte man nicht vergessen, sich vor Augen zu halten, für welche Art von Problemlösungen das Sprachkonzept geschaffen wurde. Im allgemeinen wird bei Sprachvergleichen ja häufig vergessen zu erwähnen, für welchen Zweck die jeweilige Sprache ursprünglich entwickelt wurde. So ist zum Beispiel C zur Betriebssystementwicklung ausgelegt, PASCAL ist eine Lehr- und Lernsprache, FORTH hingegen wurde zur Steuerung von Radioteleskopen geschrieben. Sieht man LISP demzufolge im Licht seiner oben angeführten Hauptanwendungsbereiche, kann man getrost behaupten, daß es seinen Zweck gut erfüllt.

Hoffentlich ist es mit diesen Ausführungen nun gelungen, einen Überblick zu geben und hier und da ein bißchen Interesse für das Thema zu wecken. Die Konzepte von LISP gehen eben

mehr in die Tiefe denn in die Breite. Um das Arbeiten mit LISP wirklich zu erlernen, muß man deshalb auch das Studium darin etwas vertiefen. Zu diesem Zweck ist besonders das Buch von Winston und Horn [2] zu empfehlen, das jetzt auch in deutscher Übersetzung vorliegt. Da dort COMMON LISP als "Haussprache" verwendet wird, lassen sich alle Beispiele auch mit XLISP auf dem Amiga nachvollziehen.

LITERATUR: siehe Teil 1.

XLISP V1.7

KI zum Nulltarif

Um diese Folgen auch praktisch nachvollziehen zu können, empfehlen wir Ihnen XLISP. XLISP ist ein leistungsstarker LISP-Interpreter, der sich zum Einstieg in die Sprache sehr gut eignet. Die neueste Version ist auf der KICKSTART PD-Diskette 16 enthalten. Diese können Sie direkt bei der Redaktion beziehen. (siehe auch PD-Ecke am Ende dieses Heftes)

DRUCK- MASTER

Mit Fleiß und ohne Workbench

Das Programm Druckmaster von Software 2000 soll IFF-Bilder in besonders hoher Qualität ausdrucken können. Dies ist jedoch mit einem erheblichen Zeitaufwand verbunden. Die Möglichkeiten des Programms sollen deshalb näher untersucht werden.

Zuerst zum Punkt Kompatibilität. Für alle Leser, die keinen NEC P6, NEC P7 oder kompatiblen Drucker besitzen, ist dieses Programm uninteressant, weil es nur für diese Drucker geschrieben ist. Außerdem sollte man einen Amiga mit mindestens 512 KB Arbeitsspeicher besitzen, wobei es egal ist, um welchen der drei Amigas es sich handelt. Das Programm akzeptiert alle Bilder, die im IFF-Format abgespeichert sind und ist dadurch kompatibel zu den meisten Malprogrammen wie zum Beispiel zu Deluxe Paint II.

Bedienung des Programms

Das Handbuch sowie die Funktionen im Programm sind deutsch, so daß man keine Probleme mit dem Verständnis hat. Die Druckprogramme (für verschiedene Auflösungen gibt es verschiedene Programme) sind alle leicht zu verstehen und zu bedienen. Da das Programm keinen Kopierschutz be-

sitzt, kann man das Original an einem sicheren Ort aufbewahren und mit der Kopie arbeiten. Auf der Diskette sind ein paar Demobilder gespeichert, mit denen man das Programm gleich ausprobieren kann. Entscheidet man sich für ein Lores-Bild, muß man nicht mehr tun, als das Programm für Lores-Bilder zu starten. Dort muß man dann nur noch den Namen des zu druckenden Bildes und die gewünschte Größe des Ausdrucks bestimmen, und schon beginnt der Drucker mit der Arbeit.

Eine interessante Option ist, daß farbige Bilder in Schwarz-weiße umgewandelt werden. Ein farbiger Ausdruck mit den Farbversionen der Drucker (NEC CP6 und NEC CP7) ist nicht möglich. Das Drucken von Hires-Bildern ist etwas umständlicher. Hierbei muß man vor einem Ausdruck nämlich erst einmal ein Programm starten, das die Grauwerte für das jeweilige Bild berechnet. Hat man dies erledigt, läßt man das eigentliche Druckprogramm. Bei Hires-Bildern stehen mehr Formate als bei den Lores-Bildern zur Verfü-

gung, die jedoch von NEC P6-Besitzern nicht alle genutzt werden können, sondern nur von NEC P7-Besitzern, weil es außer des DIN A4 auch noch das DIN A2-Format gibt. Zudem besteht noch die Möglichkeit, bis zu 6 Hires-Bilder hintereinander oder 4 parallel auszudrucken, wodurch man eine Art Tapetendruck erhält.

Sehr zeitaufwendig

Der größte Nachteil des Programms sind die langen Ausdruckzeiten. So benötigt man zum Beispiel für einen senkrechten DIN A4-Ausdruck ungefähr 2 Stunden und für einen DIN A2-Ausdruck die unglaubliche Zeit von 8 Stunden. In diesem Zusammenhang finde ich es etwas vermessen, wenn man im Handbuch darauf hinweist, daß die Druckzeit eines Hires-Bildes durch das Ausschalten der Monitoranstellung um ca. die Hälfte reduziert werden kann. Dieser Zeitaufwand, so das Handbuch, sei durch die "bislang unerreichte Qualität" in Kauf zu nehmen.

Die Qualität der Ausdrücke

Vergleicht man ein mit den Workbenchdruckertreibern und ein mit dem Druckmaster ausgedrucktes Bild, kann man durchaus einen Qualitätsunterschied feststellen. Auf dem "Druckmaster"-Bild sind mehr Feinheiten zu erkennen. Dies kommt dadurch, daß der

Druckmaster für die verschiedenen Grautöne andere Schraffierungen errechnet als die Workbenchdrucker. Sehr groß sind die Unterschiede zwischen den Ausdrucken allerdings nicht.

Fazit

Für einen guten Ausdruck kann man sich etwas Zeit nehmen, allerdings sollte man die Geschwindigkeit nicht vollkommen vergessen. Da die Qualität der Ausdrücke aber nicht sehr viel

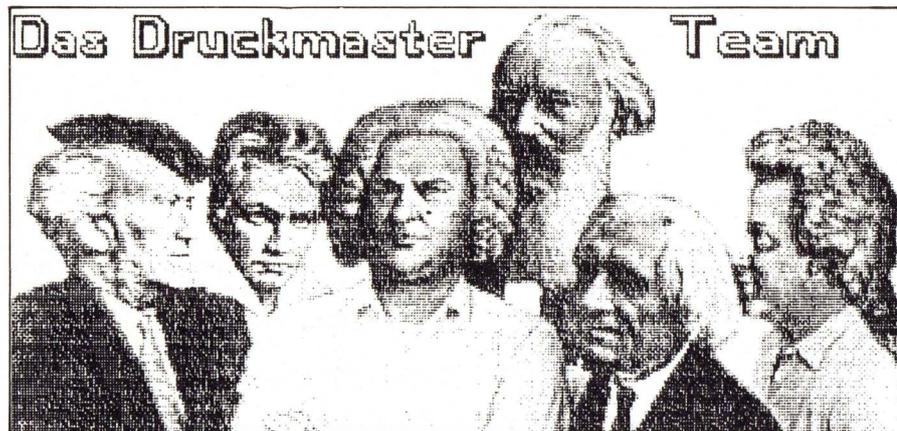
besser ist als die der Workbenchdrucker, ist der Druckertreiber eher etwas für Leute, die zu viel Geld und zu viel Zeit haben. Etwas schade finde ich es, daß nur NEC P6 bzw. NEC P7-Besitzer dieses Programm benutzen können. Daß noch nicht einmal Farbausdrücke mit den Farbdruckern NEC CP6 bzw. NEC CP7 möglich sind, spricht auch nicht gerade für das Programm. Der Preis von 89,- DM ist in Anbetracht all dieser Einschränkungen zu hoch.

Wem dennoch ein etwas besserer oder größerer Ausdruck wichtiger ist als die damit verbundene lange Wartezeit, der sollte sich einen Probeausdruck des Programms ansehen, bevor er es kauft, um zu sehen, ob er den eigenen Vorstellungen entspricht.

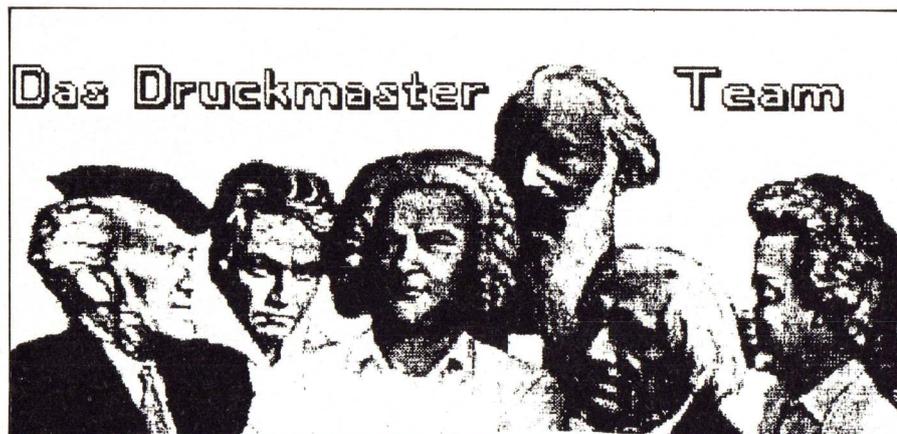
Tip

Wenn Sie sich das Programm gekauft haben, stört Sie sicher der lange Vorspann vor dem Laden der Workbench. Dies können Sie leicht ändern, indem Sie die Startup-Sequence (im Verzeichnis s) mit einem Editor bis auf drei Zeilen löschen:

```
setmap d
loadwb
endcli > nil:
```



Ein Testdruck mit Druckmaster.



Der Workbenchtreiber im Vergleich.

DRUCKMASTER

- + Bearbeitet das gängige IFF-Format
- + Alles auf deutsch
- + Druck in verschiedenen Größen
- + Konvertierung von Farbbildern in s/w

- funktioniert nur mit NEC P6/P7 und Kompatiblen
- Druckzeit viel zu lang
- druckt nur in s/w

Hersteller und Vertrieb:
Software 2000
Preis : 89,-

★ ★ AMIGA ★ ★		Jagd auf Roter Oktober	72,-
SPRACHEN/ENTWICKLUNG		Jinxter	72,-
Metacomco Assembler	159,-	King of Chicago	64,-
Metacomco Pascal	239,-	Kings Quest III	74,-
Lattice C-Compiler Vers. 4.0	399,-	Mission Elevator	57,-
SPIELE		Ports of Call	99,-
Bad Cat	54,-	Shadowgate	69,-
Barbarian (Psygnosis)	64,-	Terrorpods	64,-
Bard's Tale	79,-	Test Drive	79,-
Bureaucracy	89,-	The Guild of Thieves	67,-
Chessmaster 2000	79,-	The Pawn	72,-
Dark Castle	69,-	Ultima III	69,-
Defender of the Crown	74,-	Uninvited	74,-
Flight Simulator II	119,-	Western Games	57,-
Garrison II	62,-	DRUCKER	
Goldrunner	59,-	NEC P 2200	1049,-
Hellowoon	59,-	Epson LX-500	998,-
Impact	44,-	Epson LX-800	699,-
		Star LC 10	649,-

☛ Sofort kostenlos Preisliste bei Abteilung KS anfordern! ☛
Computer & Zubehör Versand Gerhard und Bernd Waller GbR
Kieler Str. 623, 2000 Hamburg 54, ☎ 040/570 60 07 + 570 52 75

Lange Straße 51,
2320 Plön / Holstein
Telefon 0 45 22 / 13 79

Weitere AMIGA - SOFTWARE in Vorbereitung! Händleranfragen erwünscht! Alle Programme laufen auf allen AMIGA - Modellen! Bestellungen schriftlich oder telefonisch unter 0 45 22 / 13 79. Gegen 1,30 DM in Briefmarken erhalten Sie ausführliche Produktinfo's Versand gegen Vorkasse oder per Nachnahme zuzüglich 5,- DM für Porto und Verpackung

KREUZ AS

- Sehr guter Poker-Spielautomat
- Nach dem Original-Spielhallen
- Hit „Herz As“
- Sehr gute Grafik und dig. Sound

Best.-Nr. S 02 002 88
Preis: 79,- DM

AMIGA ROULETTE

- Tolle Grafik
- bis zu 4 Spieler
- hoher Spielspaß
- alle Roulette-Regeln werden berücksichtigt
- oft Anleitung

Best.-Nr. S 01 001 88
Preis: 69,- DM

BUNDESLIGA - MANAGER

- Werden Sie zum Manager
- Ihres Fußballvereins
- Verwalten Sie Ihrem Verein zu Siegen und Geld
- Sehr gute Wirtschafts-Simulation
- hoher Spielspaß
- bis zu 4 Spieler

Best.-Nr. S 02 001 88
Preis: 69,- DM

NEUE AMIGA - SOFTWARE

MODEM TALKING

Teil 2: DFÜ, wie geht das?

Nachdem wir letztes Mal kennenlernten, was DFÜ überhaupt ist, will ich Ihnen heute die ganze Technik erklären. Bei einer seriellen Datenübertragung (Grundsatzprinzip der DFÜ), also bei der Übertragung der Bits NACHEINANDER, muß man auf einige Punkte achten. Man muß ganz einfach Vereinbarungen treffen, wie die Daten vom Sender verschlüsselt und vom Empfänger wieder entschlüsselt werden, sozusagen eine NORM festlegen.

Als erstes wäre da die Schnittstelle. Als Schnittstelle bezeichnet man im Prinzip alle Anschlüsse am Computer bzw. dem Modem, in die Sie irgendwelche Kabel hineinstecken können. Schauen Sie doch einmal nach. Die übliche serielle Schnittstelle (RS 232C, später mehr) besitzt zwei Anschlußreihen mit insgesamt 25 Pins (in der oberen Reihe 13 Pins, in der unteren Reihe 12 Pins). Irgendein internationales Gremium (genauer gesagt das CCITT -Consultatif International Telegraphique et Telephonique, kurz CCITT genannt) machte es sich dereinst zur Aufgabe, eine serielle Schnittstelle zu normen. In dessen Empfehlung "V.24" wurde unsere serielle Schnittstelle genormt (man sagt auch RS 232C).

V.24

Die CCITT V.24-Empfehlung besagt also, daß es sich um eine 25-polige Schnittstelle handelt, und was an diese 25 Pins angeschlossen werden soll, damit die Bit's rollen. In folgender Liste können Sie sich eine Übersicht darüber verschaffen, was alles in so einer kleinen Schnittstelle steckt. Eine ausgefallene Leitung, die nur durch MODEMS belegt wird, ist z.B. die Leitung "RI" (RING INDICATOR). Ein Modem kann über diese Leitung ein Telefonklingeln an den Computer melden, was zum Beispiel für den Betrieb einer Mailbox unerlässlich ist.

Die wichtigsten zwei Leitungen sind TxD und RxD. TxD heißt TRANSMIT DATA, auf gut deutsch "Sende Da-

ten". Über diese Leitung fließen die Bits unseres Computers nach draußen. Die andere Leitung heißt RxD, also RECEIVE DATA bzw. "Empfange Daten". Über diese Leitung gelangen die Daten in unseren Computer, die die Gegenseite sendet. Hier nun die versprochene Übersicht über alle Leitungen der RS 232C:

PIN-Kürzel	Bedeutung	
1	GND	FRAME GROUND
2	TxD	TRANSMIT DATA
3	RxD	RECEIVE DATA
4	RTS	REQUEST TO SEND
5	CTS	CLEAR TO SEND
6	DSR	DATA SET READY
7	GND	SYSTEM GROUND
8	DCD	DATA CARRIER DETECT
9		
10		
11		
12	S.SD	SPEED INDICATE
13	S.CTS	
14	S.TxD	
15	TxC	
16	S.RxD	
17	RxC	
18		
19	S.RTS	
20	DTR	DATA TERMINAL READY
21	SQD	
22	RI	RING INDICATOR
23	SS	
24	TXCI	
25		

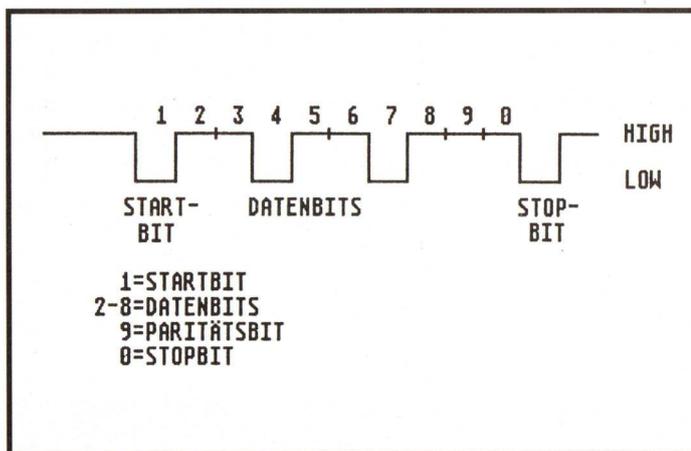
Über die oben aufgeführten Leitungen fließen also die Bits zwischen unserem Computer und dem anderen Datenendgerät, wie man in der Fachsprache einen DFÜ-Computer nennt. In all dem

Bitgeflüster muß der Computer erkennen, wann ein Zeichen beginnt und endet.

Das ist unser Problem Nummer 1:

PROBLEM NUMMER 1:

Seit Beginn der mechanischen Fernschreibertechnik befindet sich die Datenleitung im 1-Zustand, solange kein Zeichen gesendet wird (man erinnere sich: 1 = tiefer Ton). Soll nun ein Zeichen übertragen werden, so wird als erstes ein STARTBIT übertragen (also ein Mal auf 0 umschalten = höherer Ton). Danach folgen die 7 oder 8 eigentlichen Datenbits, also das Zeichen, das übertragen werden soll. Danach folgt bei Bedarf ein Prüfbit (dazu später mehr) sowie noch ein oder zwei STOPBITS (also ein oder zwei 1-Zustände, siehe Bild).



Der Computer erkennt also anhand der START- und STOP-Bits, wann ein Zeichen beginnt und wann es endet. Was passiert aber, wenn ein Bit durch eine schlechte Telefonleitung verschluckt oder gekippt wird? Das Zeichen würde in solch einem Falle beim Empfänger falsch ankommen. Ohne die Paritätsprüfung würde ein solcher Fehler nicht auffallen. Dafür gibt es die gerade und die ungerade Paritätsprüfung. Auf englisch heißt das "EVEN" bzw. "ODD PARITY CHECK". Wählt man die gerade Paritätsprüfung, so muß die Summe aller übertragenen Bits eines Zeichens gerade sein. Nach diesem Schema wird dann das Paritätsbit gewählt. Wählt man die ungerade Parität, so muß die Summe aller übertragenen Bits ungerade sein.

Beispiel:

UNGERADE PARITÄT	011100101000	FÜNF EINSEN => ODD PARITY
GERADE PARITÄT	011111111110	ZEHN EINSEN => EVEN PARITY

Wird bei einer Übertragung mit Paritätsprüfung ein Bit durch irgendwelche Einflüsse "umgelegt" (eine 1 wird zu einer 0 oder umgekehrt), so erkennt dies der Empfänger daran, daß die Summe der Einsen im übertragenen Zeichen nicht mehr mit der vorgegebenen Paritätsprüfung übereinstimmt. Nun kann der Empfänger eine Fehlermeldung ausgeben oder den Sender veranlassen, das betreffende Wort noch einmal zu senden etc. Für die "Otto-Normal-Mailboxen" gibt es übrigens eine solche Paritätsprüfung nicht. Falsch übertragene Zeichen kann man ja durch Lesen sofort am Bildschirm erkennen, und im Kopf verbessern. Mit der Zeit gewöhnt sich der

kommen also real etwa 30 Zeichen pro Sekunde auf den Bildschirm des Empfängers! Vielleicht erscheint Ihnen jetzt dieser Wert unheimlich hoch, er ist es aber nicht. Wie gesagt: An das Lesen mit 300 Baud gewöhnen Sie sich schnell, und modernere Mailboxen arbeiten ohnehin mit 1200 Baud, also mit glatt der vierfachen Geschwindigkeit! Noch ein Tip für DFÜ-Fachgespräche: BAUD wird "BOOD" ausgesprochen. Also wundern Sie sich nicht, wenn Hacker über vermeintliche Boote reden!

DUPLEX?

Die Begriffe Voll- und Halbduplex-Betrieb hatten wir ja das letzte Mal schon einmal erklärt. Kurz zur Wiederholung: Im Vollduplex-Betrieb können Sender und Empfänger quasi gleichzeitig senden und empfangen, während im Halbduplex-Betrieb jeweils nur eine Station, also Sender oder Empfänger zu einem Zeitpunkt senden bzw. empfangen kann.

BTX!

Aber wie immer gibt es eine Ausnahme: BTX (Bildschirmtext). In der Regel sendet eine Station genauso schnell, wie sie empfängt. Mailboxen arbeiten zumeist mit 300 Baud. BTX sendet aber mit 1200 Baud und empfängt mit nur 75 Baud!

Der einfache Grund liegt darin, daß Sie bei BTX ja viele Daten lesen möchten, und nur wenige Daten eingeben (z.B. Seitenzahlen etc). Um die Übertragung schneller und effizienter zu gestalten, hat sich ein kluger Mitarbeiter der Post gedacht, daß man eigentlich kaum etwas in BTX eingibt, daß also das BTX-System Eingaben deshalb auch nicht schnell verarbeiten muß. Stattdessen will der Anwender aber alle Informationen schnell lesen (Time is money .. auch bei der Post !), weshalb die Bits vom BTX-System zum Anwender mit rasenden 1200 Baud huschen.

Die Reihenfolge der übertragenen Bits (Datenbits) ist nach einem festen Schema festgelegt.

versierte DFÜ-Betreiber daran, verstümmelte Texte zu entziffern (ein Studium in Graphologie folgt!). Mit wachsender DFÜ-Tätigkeit gewöhnt man sich auch an die Übertragungsgeschwindigkeit von 300 Baud, so daß man alsbald auch alle Bücher und Zeitungen in 300 Baud lesen wird!

BAUD ?

Halt, da war es schon wieder, ein Fremdwort: BAUD. Die Zahl vor diesem Wort gibt die Übertragungsgeschwindigkeit an, mit der die einzelnen Bits über die Leitungen huschen. Baud bedeutet BITS PRO SEKUNDE. Bei 300 Baud fließen 300 Bits pro Sekunde vom Sender zum Empfänger. Bei ca. 10 Bits pro zu übertragendem Zeichen

MAILBOXEN

Gehen wir jetzt aber endlich zur Praxis über: MAILBOXEN. Wenn Sie einige Mailbox-Listen durchgehen, so werden Sie sicherlich folgende Einträge sehen:

06192-7960 CVCM 24h 300bd
fd 7/N/1

Nicht erschrecken, das ist alles halb so wild. Die erste Zahlenkombination gibt die Telefonnummer der Mailbox an. Die Buchstaben, die daraufhin folgen, geben den Namen der Mailbox an. "24h" besagt, daß die betreffende Mailbox Tag und Nacht (rund um die Uhr) erreichbar ist. Steht da z.B. "18 bis 22h", so sollten Sie im eigenen und im Interesse des Mailbox-Betreibers auch nur zwischen 18 Uhr und 22 Uhr die betreffende Nummer anwählen. "300bd" bedeutet, daß die Mailbox mit einer Übertragungsrate von 300 Baud arbeitet, das "fd" bedeutet FULLDUPLEX. Die Sieben sagt, daß die betreffende Mailbox mit 7 Datenbits arbeitet, "N" bedeutet "Keine Parität" (ein "E"

stünde für EVEN, ein "O" für ODD PARITY). Die letzte Zahl, also die Eins, gibt die Anzahl der Stopbits an. Alle diese oben aufgeführten Daten müssen Sie vor Kontaktaufnahme noch Ihrem Terminalprogramm mitteilen, damit sich Ihr Computer auch mit der Mailbox versteht. Man spricht bei diesen Daten von den Kommunikations-Parametern.

KOMMUNIKATIONS-PARAMETER

Die meisten deutschen Mailboxen arbeiten mit den oben aufgeführten Parametern. Das weitgängigste Protokoll ist 8/N/1. Anzumerken sei noch, daß die oben aufgeführte Mailbox nicht mehr existiert. Wenn Sie irgendwann einmal eine Telefonnummer anwählen, hinter der sich eine Mailbox verbergen sollte, aber ein Mensch an das Telefon geht, so legen Sie nicht gleich wieder auf. Fragen Sie die Person am anderen Ende, ob er von einer Mailbox weiß bzw. was mit selbiger los ist. Vielleicht ist die Mailbox vorübergehend außer

Betrieb, oder irgendein netter Mensch wollte seinem Nachbarn 'mal einen Streich spielen, und hat seine Nummer als Mailboxnummer veröffentlicht. Im letzten Falle verständigen Sie sofort den Betreiber der Mailbox, von der Sie die falsche Nummer haben, sonst kann der arme Mensch überhaupt nicht mehr schlafen.

Möchten Sie sich jetzt einen Akustikkoppler oder ein Modem kaufen, so sollten Sie darauf achten, daß es 1200 Baud schafft, denn bei dieser Baudrate liegt die Zukunft. Ferner sollte er/es eine FTZ-Nummer besitzen (siehe letzte Ausgabe!).

Im nächsten Heft

Im nächsten Teil geht's richtig los. Anhand einer Beispiel-Mailbox werde ich Ihnen den Umgang mit einer Mailbox zeigen. Schließlich sollten Sie schon wissen, daß Sie einen Mailboxbetreiber ärgern können, wenn Sie das System nicht ordnungsgemäß verlassen. Und schließlich sind Telefoneinheiten teuer, also spart das auch Ihr Geld, und die Post verdient nicht allzuviel!

ENDE

White Box & Hard-Disk

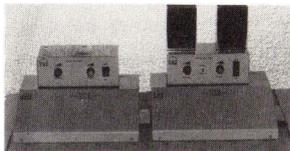
preisgünstig und bedienfreundlich

White Box für Atari und Amiga

Stereo Super-Sound (500) mit Boxen, auch für Walkman, Monosound (520), Video Control RGB, TTL, Composite

20, 30, 40, 65 MB Festplatten am Expansion Port, Expansion Bus durchgeführt, OK mit Speichererweiterung, Startup Programm auf Diskette, modernes Design im Amiga Look

einfach steckbar, Garantieverlust
Netzteil integriert, deutsche Dokumentation



Whity Box 520 f. Atari 20 MB = 1348 DM, 30 MB = 1549 DM
Whity Box 500 f. Amiga 40 MB = 1798 DM, 65 MB = 2295 DM
jeweils 198,— DM The Future is Yours! A Generation Ahead

EDV-Technik Behrendt

Am Kahnplätzchen 14 · 6301 Heuchelheim
Telefon (06 41) 6 27 45



AMIGA - SOFTWARE

Public Domain Disketten

Jetzt über 500 Public Domain Disketten vorrätig von Fish, Faug, Panorama, Chiron Conception, Software Digest, ACS, Amicus, Tornado, TBAG, SACC, AUGÉ, Spiele und Bilder.

Einzeldiskette 3,50 DM

Paketangebote:
10 Disketten Ihrer Wahl 33,00 DM
20 Disketten Ihrer Wahl 65,00 DM
Gesamtkatalog 6,00 DM

Für unsere Public Domain Software verwenden wir 2DD Qualitätsdisketten.

Aufpreis für 2DD Markendisketten 1,30 DM pro Diskette.

Zubehör:
Diskettenlaufwerk mit NEC 1037a anschlussfertig für alle Amiga-Modelle 328 DM, Speichererweiterung 512 KB für Amiga 500, abschaltbar, mit Uhr 239 DM

A. Fischer, Kirchstr. 40, Tel. 05257- 4347

4794 Hövelhof

Musik- und Grafiksoftware Shop

Wasserburger Landstr. 244 * 8000 München 82
Telefon: 089/430 62 07

Umfangreiches Musik- und Midi-Software-Angebot für alle Amiga

Midi-Editorssoftware für folgende Synthesizer:
DX7II, DX21, DX27, DX100, TF1, TX7, DX9, TX802, MT32, D50, ESQ1, ESQM, EX 80, Casio CZ 101, CZ 1000, CZ 3000, CZ 5000
Zahlreiche Sequenzerprogramme für AMIGA jetzt lieferbar!

MIDI-Interfaces für AMIGA 500/1000/2000 in verschiedenen Versionen.

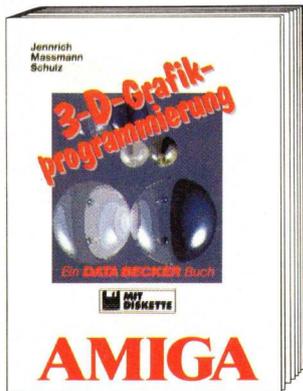
Fordern Sie unsere kostenlose AMIGA-MIDI-LISTE an. Dort finden Sie Beschreibungen und Preise zu all unseren AMIGA-MIDI-Produkten.

Über 300 Public-Domain-Disketten für AMIGA lieferbar - PRO DISKETTE DM 7,-

Alle gängigen Grafikprogramme, Soundsampler, Videodigitizer, Scanner und Grafiktableaus für Amiga auf Lager. Fordern Sie unsere kostenlose AMIGA-Preisliste an. Versand per Vorkasse oder Nachnahme.

Rufen Sie uns einfach an oder besuchen Sie uns in unserem Laden!
MO-FR 10-18.30 UHR * SA 9-13.00 UHR

RUND UM DEN



Licht und Schatten.

Dieses Buch öffnet Ihnen den Weg in eine faszinierende Welt. Anhand ausgefeilter Grafikalgorithmen erzeugen Sie phantastische Bilder: naturgetreue Spiegelungen, Licht und Schatten. Alles absolut realistisch und automatisch berechnet. In allen Auflösungen mit bis zu 4096 Farben!

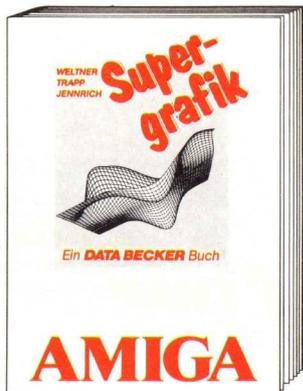
Amiga 3-D-Grafikprogrammierung
Hardcover, 283 Seiten
inkl. Diskette, DM 59,-



Da steckt Musik drin.

Zaubern Sie zarte Klänge oder heiße Rhythmen aus Ihrem Amiga – mit dem Musikbuch. Hier werden Sie zu einem Komponisten ausgebildet, der nicht nur die notwendigen Grundbegriffe der Musiktheorie beherrscht, sondern auch modernste Technik einzusetzen weiß: Musikprogramme wie Sonix, DeLuxe Construction Set oder Audio Master, Sampler, MIDI-Interface und, und, und.

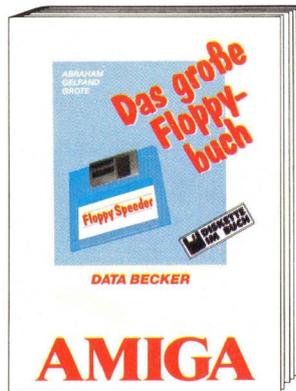
Amiga-Musikbuch
Hardcover, ca. 300 Seiten,
DM 49,-
erscheint ca. 6/88



Das Buch zum Thema Nr. 1.

Grafik auf dem Amiga – mit dem entsprechenden Know-how ist hier fast alles möglich. Dieses Buch bietet es Ihnen: Nutzung der Libraries, die Register der Grafik-Chips, Aufbau und Programmierung von Screens, Windows, Halfbrite und Interlace aus BASIC und C uvm.

Amiga Supergrafik
Hardcover, 686 Seiten
inkl. Diskette, DM 59,-



Alles zur Amiga-Floppy.

Wie umfassend die Informationen eines Floppy-Buches von DATA BECKER sind, dürfte ja wohl bekannt sein. Im Amiga Floppybuch finden Sie darüber hinaus noch eine ganze Menge mehr, z.B. ein Superkopierprogramm, einen Floppyspeeder, einen Diskmonitor...

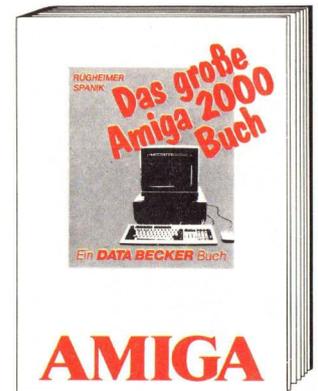
Amiga Floppybuch
Hardcover, 398 Seiten
inkl. Diskette, DM 59,-



Gleich loslegen.

Bei „500 für Einsteiger“ heißt es aufstellen, anschließen und sofort loslegen. Schnell und für jedermann verständlich zeigt Ihnen dieses Buch, was Sie mit Ihrem neuen Rechner so alles anstellen können. Workbench, AmigaBASIC, CLI und DOS – schon bald wissen Sie, worauf es ankommt.

Amiga 500 für Einsteiger
343 Seiten, DM 39,-



Ihr ständiger Begleiter.

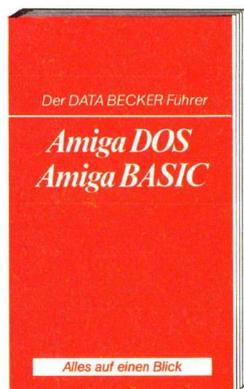
Haben Sie einmal mit dem großen Amiga-2000-Buch gearbeitet, wird es sicherlich seinen festen Platz neben Ihrem Amiga behalten. Denn dieses Buch bietet Ihnen mehr als eine detaillierte Einführung. Vom Laufwerkseinbau bis hin zum Kickstart im RAM wird hier echtes Profiwissen vermittelt.

Das große Amiga-2000-Buch
Hardcover, 684 Seiten, DM 59,-

DATA BECKER

Merowingerstr. 30 · 4000 Düsseldorf · Tel. (0211) 310010

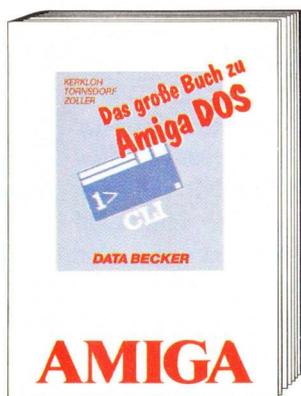
AMIGA!



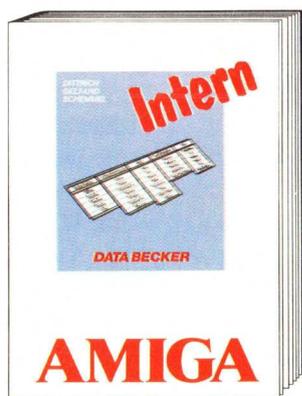
Alles auf einen Blick.
Der DATA BECKER Führer zu AmigaDOS und AmigaBASIC – das schnelle Nachschlagewerk für jeden Amiga-Anwender. Alle Befehle und Kommandos finden Sie hier auf einen Blick.
Der DATA BECKER Führer zu AmigaDOS & -BASIC
269 Seiten, DM 24,80



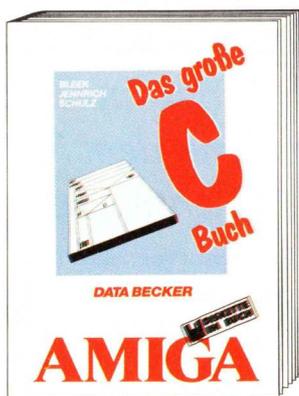
AmigaBASIC komplett.
Dieses Buch zeigt auf über 770 Seiten, worauf es beim Programmieren in AmigaBASIC ankommt. Natürlich mit jeder Menge interessanter Programmbeispiele, die auch gleich auf Diskette mitgeliefert werden. Ebenfalls im Buch: eine detaillierte Beschreibung des AC-BASIC-Compilers.
AmigaBASIC Hardcover, 775 Seiten inkl. Diskette, DM 59,-



Runter von der Workbench.
Rein ins AmigaDOS. Denn hier eröffnen sich ganz neue Möglichkeiten: Umlenken von Ein- und Ausgabe, mit RAM-Disk und CLI arbeiten, eigene CLI-Befehle programmieren, Batch-Dateien, Multitasking mit dem CLI, STARTUP-Sequenz... Das Know-how hierzu finden Sie im großen Buch zum AmigaDOS.
Das große Buch zu AmigaDOS Hardcover, 310 Seiten, DM 49,-



Know-how aus erster Hand.
Amiga Intern – der ganze Rechner komplett in einem Buch: 68000-Prozessor, CIA, Blitter, Customchips, die Strukturen von EXEC, I/O-Handhabung, Verwaltung der Ressourcen, EXEC-Base, resetfeste Programme, IFF-Format, Programmierung der EXEC- und DOS-Routinen... Eben ein typisches Intern von DATA BECKER.
Amiga Intern Hardcover, 639 Seiten, DM 69,-



Profi-Programme in C.
Wenn Sie an C Spaß gefunden haben, gibt Ihnen das große C-Buch den letzten Schliff. Denn hier erfahren Sie nicht nur, wie ein C-Compiler arbeitet und wie Sie selbst die schwierigsten Probleme in C lösen, sondern auch, wie Sie eine optimale Benutzeroberfläche entwickeln.
Das große C-Buch zum Amiga Hardcover, 682 Seiten inkl. Diskette, DM 69,-

Coupon!

HIERMIT BESTELLE ICH

NAME, VORNAME

STRASSE

ORT

MEIN COMPUTER

zzgl. DM 5,- Versandkosten unabhängig von der
bestellten Stückzahl
 per Nachnahme Verrechnungsscheck liegt bei

MODULA 2 & ASSEMBLER

*Maschinencode in Modula
oder*

Die Rechnung ohne den Wirth gemacht

Nach dem Artikel C & Assembler stellen wir dem Leser in dieser Ausgabe ein Programm vor, mit dem man Modula 2 und Assembler verbinden kann. Dieses Programm wurde in Modula 2 geschrieben und mit dem M2Amiga-Compiler von Mayer&Vogt entwickelt.

Das Programm `AssemToINLINE` führt eine an sich ganz einfache Operation durch: Es wandelt ein PC-relatives Assemblerprogramm in Modula 2-INLINE-Code um. Um zu erklären, was es mit PC-relativ auf sich hat, muß man allerdings etwas ausholen.

Sobald man in einem Assemblerprogramm eine feste (absolute) Adresse angibt, ist das Programm an eine bestimmte Stelle im Speicher gebunden. Falls der vorgesehene Adressbereich schon belegt ist, führt das zu Problemen, sprich der Rechner stürzt ab.

Damit dies nicht passiert, muß das Programm an einem anderen Ort im Speicher abgelegt werden.

Diese Art der Programmierung nennt man *relokatibel* (verschiebbar). Normalerweise sorgt der Assembler dafür,

daß ein Programm *relokatibel* ist, indem er eine Liste der absoluten Adressen abspeichert. Ein Loader sorgt dafür, daß diese Liste beim Programmstart neu errechnet wird, indem er den Startoffset zu der Tabelle addiert. Als zweite Möglichkeit bietet sich an, ein Programm *lageunabhängig* zu programmieren; in einem solchen Programm dürfen keine absoluten Adressen vorkommen. Beim 68000er benutzt man dazu die PC-relative Adressierung, wobei PC ProgrammCounter bedeutet und die Adresse meint, an der die CPU gerade einen Befehl abarbeitet. Die aktuelle Adresse berechnet sich dann aus dem Stand des PC plus einem Offset. Der Offset ist 16 Bit groß und reicht folglich von -32768 bis +32767.

Allerdings erzeugt das Modula-Programm nicht nur den INLINE-Code, sondern es kümmert sich auch um die Datenübergabe an das Assemblerprogramm, wobei nicht nur die Datenübergabe, sondern auch die Datenrückgabe von dem Programm erledigt wird. Der INLINE-Code, der vom Programm `AssemToINLINE` erzeugt wird, läßt keinen direkten Schluß auf das Assemblerprogramm zu. Man kann an der erzeugten Zahlenkolonne also nicht erkennen, was das ursprüngliche Assemblerprogramm macht. Daraus läßt sich eine wichtige Tatsache ableiten: Das Assemblerprogramm muß in Ordnung sein, bevor man es in INLINE-Code umwandelt. An einem Beispiel wird die Anwendung gezeigt. Das als Beispiel aufgeführte Assemblerprogramm führt die folgenden Aktionen durch:

1. Es kopiert nur den Inhalt der Adresse, auf die `a0` zeigt, nach `d0`. Es werden nur 8 Bit kopiert (ein Byte).
2. `d0` wird auf das Wordformat gebracht (von 8 Bit auf 16 Bit).
3. Beenden des Programmes mit einem `rts` (RETURN from SUBROUTINE).

Beispiel

```
Inhalt von d0 vor dem move.b:
1111 1111 1111 1111 1111 1111 1111 1111

Inhalt von d0 nach dem move.b:
1111 1111 1111 1111 1111 1111 0101 1010

Inhalt von d0 nach dem Extend:
1111 1111 1111 1111 0000 0000 0101 1010

Programm Byte -> Integer

move.b (a0),d0
ext.w d0
rts
end
```

Welche Nutzenanwendung hat das Assemblerprogramm? Es macht etwas, was in Modula an sich nicht erlaubt ist: Eine Datentypenkonversion über Bitgrenzen hinweg. Man kann mit diesem Programm eine BYTE-Zahl in eine INTEGER-Zahl umwandeln. Falls die Umwandlung nicht nach INTEGER, sondern nach LONGINT stattfinden soll, muß das Assemblerprogramm so aussehen:

```
Programm Byte -> LongInteger

move.b (a0),d0
ext.w d0 ; Die beiden-ext Befehle sind um 8 Zyklen schneller als andi.l
$F,d0.
ext.l d0 ; Ein move.q #0,d0 vor dem move.b
rts ; sollte man vermeiden, da es sonst zu Vorzeichenproblemen
kommt.
end ; Extend berücksichtigt dies.
```

Der einzige Unterschied zum ersten Programm ist ein weiteres Extend von Word auf Longword.

Wie oben erwähnt, müssen die Assemblerprogramme PC-relativ assembliert sein, damit sie laufen. Diese PC-Relativität kann man leicht erreichen, falls der verwendete Assembler

```
(*SE-*)
PROCEDURE ByteInt ( I{8} : ADDRESS ) : INTEGER;
BEGIN
  INLINE(004112, 018560, 020085, 000000, 000000) ;
  RETURN REG(0)
END
ByteInt;
(*SE-*)
PROCEDURE ByteLongInt ( I{8} : ADDRESS ) : LONGINT;
BEGIN
  INLINE(004112, 018560, 018624, 020085, 000000) ;
  RETURN REG(0)
END
ByteLongInt;
```

Im folgenden Programm sieht man, wie die erzeugten Prozeduren anzuwenden sind:

```
MODULE INLINETest;

FROM SYSTEM IMPORT ADR, ADDRESS, BYTE, INLINE, REG;

FROM InOut IMPORT WriteString, WriteLn, WriteInt;

VAR b : BYTE;
    i : INTEGER;

(*SE-*)
PROCEDURE ByteInt ( I{8} : ADDRESS ) : INTEGER;

BEGIN
  INLINE(004112, 018560, 020085, 000000, 000000) ;
  RETURN
  REG(0)
END ByteInt;

BEGIN
  b := 45;
  i := ByteInt(ADR(b)); (* Die Adresse übergeben an der *)
                        (* das Byte im Speicher steht *)
  WriteInt(i,6);
  WriteLn;
END INLINETest.
```

Falls man bei der Eingabe von Modula- und Assemblerprogramm keinen Fehler gemacht hat, startet man das Programm AssemToINLINE und übergibt als Argument den Namen des assemblierten, PC-relativen Programmes. Anschließend fragt das Pro-

gramm nach, ob Daten in die Prozedur übernommen werden.

Die Eingabe eines "j" führt zu der Frage, in wievielen Registern Daten übernommen werden sollen und zur Frage, in welchen Registern dies geschehen soll. Hierzu muß gesagt werden, daß die Modulnotation bei der Eingabe der Registernummern übernommen wurde, d.h., daß 0=d0, 1=d1, ... 8=a0 und 15=a7 bedeutet.

Die Eingabe wird übrigens auf ihren Sinn hin getestet. Die Frage, welcher Datentyp übergeben werden soll, muß man nach den schon angesprochenen Gesichtspunkten beantworten.

Falls man aus der PROCEDURE eine FUNCTION machen möchte, muß man auf die Frage "Sollen Daten aus der Procedure übernommen werden?" ein "j" eingeben. In diesem Fall folgt die Frage, aus welchem Register welcher Datentyp hertransportiert werden soll.

Nachdem man alle Daten eingegeben hat, erzeugt AssemToINLINE nun den Sourcecode. Der Name, unter dem der Sourcecode abgespeichert wird, ergibt sich aus dem übergebenen Namen mit einem angehängten ".modP".

Bei den Tests mit AssemToINLINE trat nur ein Problem auf: Teilweise war der Stack zu klein. Falls der Requester "Stapelüberlauf" erscheint, sollte man den Stack auf 50kB bis 60kB heraufsetzen. Der CLI-Befehl dafür lautet "STACK 50000".

ENDE

MACHEN SIE MIT

IN EIGENER SACHE

*Liebe Leser,
die KICKSTART gibt es nun schon seit
über einem Jahr. Die KICKSTART
machte sich als erste monatliche AMIGA-
Zeitschrift Deutschlands daran,
Licht in das damals noch unbekannte We-
sen AMIGA zu bringen. Viele
Geheimnisse dieses Rechners wurden ge-
lüftet und in Form von Programmen
oder Artikeln jedem zugänglich
gemacht.*

Doch nicht alle Gebiete lassen sich von uns abdecken. Der einzelne Leser ist hierbei gefordert, seine eigenen Erfahrungen und Entdeckungen zu präsentieren. Deshalb der Aufruf an alle Leser: Beteiligen Sie sich aktiv an der KICKSTART.

Wir möchten Ihnen auf diesen Seiten die einzelnen KICKSTART-Rubriken darlegen und auf einige Besonderheiten verweisen, die Sie wissen sollten, wenn Sie diesem Aufruf folgen.

TIPS & TRICKS

Hier sind, wie der Name schon sagt, alle kurzen Tricks & Kniffe gefragt, die helfen, mit dem AMIGA besser umzugehen und mehr aus ihm rauszuholen zu können. Angefangen beim Umgang mit dem CLI, der Workbench und dem Betriebssystem, bis hin zur Bedienung von PD-Software und Anwenderprogrammen. Auch kurze Programmerroutinen sind immer willkommen.

Kennwort: ``TIPS & TRICKS``

KICKS FÜR INSIDER

Diese Rubrik versuchen wir jeden Monat mit interessanten Listings, die ein wenig Licht in die Programmierkunst des Amiga bringen, zu füllen. Daß hierbei eine aktive Mitarbeit unserer Leserschaft erforderlich ist, können

Sie sich sicherlich denken.

Was sollten Sie beachten?

KICKS FÜR INSIDER enthält Programme oder Routinen zu allen Bereichen der AMIGA-Programmierung. Die Sprache überlassen wir dabei Ihnen. Wichtig ist nur, daß die Listings entweder nützlich oder lehrreich sind.

- Schicken Sie uns den Source-Code und das ausführbare Programm auf Diskette zu.
- Der Artikel sollte sich ebenfalls auf Diskette befinden. Bei vorhandenem Drucker bitten wir Sie, einen Ausdruck von Text und Listing beizulegen.
- Falls der Artikel Bildschirmfotos benötigt, so sollten sie die Bilder auf Disk abspeichern, z.B. mit Grabbit.
- Die Länge des Listings sollte dem Nutzen des Programms entsprechen, d.h., daß man auf Schnörkel, wenn diese das Programm unnötig verlängern, besser verzichtet.
- Ein KICK muß nicht unbedingt aus einem ablauffertigen Programm bestehen. Auch kurze Routinen, die der Anwender in sein eigenes Programm einbinden kann, sind gefragt. Auf diese Art und Weise lassen sich leicht leistungsstarke Bibliotheken erstellen.

Ihrer Fantasie lassen wir hierbei freien Lauf.

Kennwort: ``KICKS FÜR INSIDER``

LISTINGS

Für Listings gilt im allgemeinen das gleiche wie für KICKS. Der einzige Unterschied zu den KICKS besteht in der Programmlänge. Ein Listing kann schon mal etwas länger werden, aber sollte noch abtippbar und dementsprechend nützlich sein. Den Maßstab für die Länge des Programms können sie leicht selbst beurteilen. Wie viele Zeilen würden Sie noch abtippen?

Kennwort:

``PROGRAMMEINSENDUNG``

HARDWARE

Der Lötcolben glüht bei uns eigentlich immer. Der gute AMIGA hat darunter zu leiden, denn an ihm werden die neue Hardwareschaltungen ausprobiert. Aber solange es noch keinen AMIGA-Schutzverein gibt, werden unsere Entwickler kein Mitleid mit

ihm haben. So muß unser leidgeplagter A1000 seit mehreren Monaten mit einer Taktfrequenz von über 9Mhz seine Arbeit verrichten. Für sein Alter verkraftet er dieses Tempo aber ganz prima. Auch unser JuniorPrommer-Projekt wird seine Fortsetzung finden, beginnend mit dem KICKSTART-Umschalter in diesem Heft.

Keine Angst, es werden auch einfachere Schaltungen, die sich vor allem für Hardware-Bastler mit weniger Fachkenntnis eignen, angeboten. Doch auch hier sind sie aufgerufen, sich mit eigenen Projekten zu beteiligen.

Kennwort: ``HARDWARE``

GRUNDLAGEN

Nicht zuletzt sind wir für gute Grundlagenartikel über 'AMIGA und die Welt' jederzeit empfänglich. Themen gibt es genügend. Haben Sie sich in einen bestimmten Bereich erfolgreich hineingearbeitet und wissen Sie darüber zu berichten, so fassen Sie Ihre Erkenntnisse doch einfach in einem Artikel zusammen und schicken ihn uns zu.

Bitte beachten:

- Text auf Diskette (ASCII-Format)
- Keine Trennung und linksbündig, kein Blocksatz
- Bilder auf Diskette abspeichern

SATIRE

Wie auch in dieser Ausgabe (siehe Biochips), sind wir von satirischen Artikeln sehr angetan.

Diese Serie soll und wird auch in Zukunft weitergeführt werden.

Also, wenn eine Idee in Ihrem Hinterkopf schlummert, so lassen Sie diese nicht verkümmern - wenden Sie sich an die Redaktion.

Wir hoffen, daß Sie sich mit dem einen oder anderen Artikeln an der KICKSTART beteiligen werden. Natürlich wird Ihre Mühe auch honoriert. Bei Fragen wenden Sie sich bitte direkt an die Redaktion, die Ihnen gerne zur Verfügung steht.

Mo-Fr 11-17 Uhr

Tel. 06196/481811

Kontaktadresse:

'Merlin'-Computer GmbH
KICKSTART Redaktion

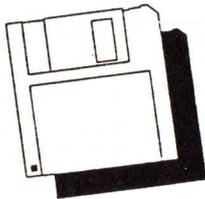
Kennwort:

Industriestr. 26
6236 Eschborn

KICKSTART DISKETTEN-SERVICE

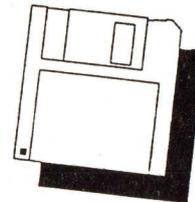
Für AMIGA-Fans, denen das Abtippen der abgedruckten Programme zu viel Mühe bedeutet, gibt es jetzt eine Alternative, denn alle in der KICKSTART abgedruckten Programm-Listings (bei Compilersprachen natürlich auch die ausführbaren Program-

me) befinden sich auf den KICKSTART-MONATSDISKETTEN. Auf jeder Diskette sind die Programme von zwei Monaten enthalten. Der Preis liegt bei nur DM 19.- pro Diskette zuzüglich DM 5.- Versandkosten.



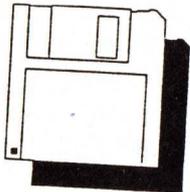
Juli/Aug. 87

Fractals
Business-Grafik
Funktions-Plotter



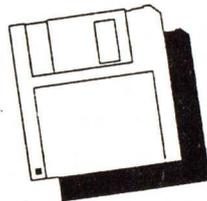
März/April 88

Röhrenschoner (Bildschirmabschalter)
Jump (IF-Befehl für's CLI)
Find (Datei-Sucher)
Fractale Landschaften (Basic)
Systemfonts in Basic
Colorhunter
DigiSound (Basic)
Viruschecker
(SCA,Byte Bandit)



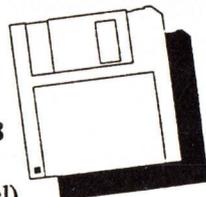
Nov./Dez. 87

MovieMaker
C-Shell
Tastenbelegung
Vokabeltrainer
Viruschecker



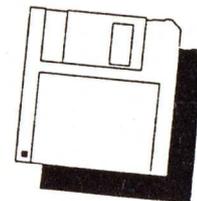
MAI / JUNI 88

- BOOTLETTER
- GURU ADE (FEHLERABFANG)
- GROSSE AUSWAHL
(BATCH MENU)
- BASIC GURU (ALERT)
- GROSS GEWÖRDEN
(DIR MIT LÄNGENANGABE)
- CURSERPOSITIONIERUNG
- AUFGERISSEN (GROSSES
CLI / FENSTER)
- IFF / FORMAT / WANDLER
- SINGENDE FLOPPY
- AUTOBOOT



Jan./Feb. 88

TRON (Spiel)
KICKS (Requester,
Scroll-Demo in Basic,
Out, Fullscreen)
Konverter (Amiga-
ASCII - PC-ASCII)
SetPCDate (Uhrzeit
für PC)



Für die Bestellung können Sie die Postkarte in der Heftmitte benutzen!

**KICK
START**

DataCom

Golem-Hardware

3,5 Display Drive	339,-
3,5 Drive	319,-
5,25 Display Drive	409,-
5,25 Drive	389,-
Golem-RAM-Box 2 Megabyte	1198,-
Golem-RAM-Box A500	1198,-
Golem Sound Stereo	179,-
Mono	129,-
Kickstart und Uhr A1000	279,-
z.B. mit Anti-Virus-Kick oder Kick 1.3 (Booten von Harddisk)	

Peripherie

Drucker	NEC P6	1186,-
	NEC P6 Color	1598,-
	Star LC-10	598,-
	NEC P 2200	998,-
Plotter DIN A3		2498,-
HP GL- und HP 7475-kompatibel, 8 Farben, RS232 und Centronics, plottet z.B. mit NEWIO		
Monitor Amiga 1084		649,-
Festplatten von NEC		
	20 MB, 40 ms	699,-
	40 MB, 40 ms	999,-
Controller von OMTI		
	5520 (MFM)	169,-
	5527 (RLL)	199,-

Amiga-Hardware

3,5 Amiga Drive	309,-
NEC 1037 Superslimline, Abschalter, Busdurchführung, Amiga-farbenes Metallgehäuse	
5,25 Amiga Drive	369,-
40/80 Track, Abschalter, Busdurchführung, Amiga-farbenes Metallgehäuse	
3,5 A2000 Drive	227,-
NEC 1036/37 inkl. Einbausatz	
3,5 NEC 1036A	207,-
3,5 NEC 1037	207,-
5,25 TEAC FD 55 FR	269,-
5,25 NEC 1157, 1,6 MB	279,-
Kick-Umschaltplatine	149,-
für A500 und A2000, bestückt mit Anti-Virus oder 1.3 Kickstart (Booten von Platte), umschaltbar auf orig. Kick	
512-K-Karte A500	89,-
komplett mit Uhr und Akku, ohne RAM (Plätze gesockelt)	

Zubehör

Druckerkabel A2000, 500	19,-
Monitorkabel, alle Amiga	21,-
Fernsehkabel (Skart)	21,-
Abdeckhaube A500	21,-
Mouse Pad (Mausunterlage)	19,-
Kabel Amiga/C64-Floppy	15,-
Bootselector, alle Amiga	19,-

Bestellen Sie einfach per

**Telefon 0231/81 8210
und 81 8219**

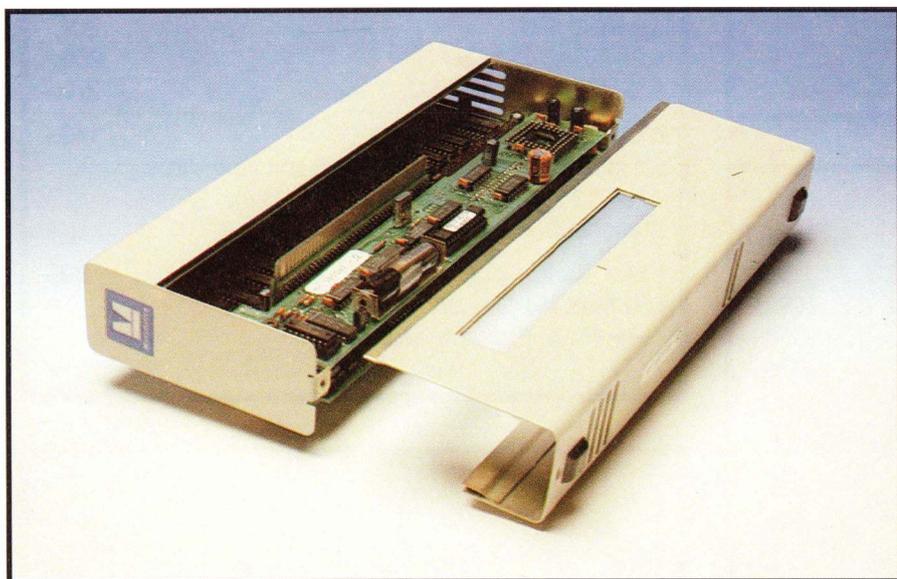


Computer Apelank 28, 4600 Dortmund

Ihr Partner im EDV-Bereich

STARBOARD II

Star oder einer von vielen?



Es gibt inzwischen eine ganze Reihe von Herstellern, die RAM-Erweiterungen für die Amiga-Serie herstellen. Eine bisher wenig bekannte Karte war die Starboard II von Microbotics, die durch ihre Vielseitigkeit Überzeugt. Kickstart fragt deshalb: Ist Starboard II ihren Preis wert?

Die RAM-Erweiterung selbst ist in verschiedenen Ausbaustufen erhältlich, von Null-MByte (die Selbstbestück-Version), über 1 MByte, bis hin zur 2 MByte-Version. Man kann also nach und nach aufstocken, gerade so, wie der Geldbeutel es erlaubt.

Weiterhin hat man noch die Möglichkeit ein Zusatzmodul aufzubauen, das die Leistung der Erweiterung noch vergrößert. Erhältlich sind zur Zeit das Multifunction-Module sowie das Star-Drive-Modul. Im Test hatten wir das Starboard mit 1 MByte und die Multifunction-Module für Amiga 1000.

Der Aufbau

Das Starboard II im Metallgehäuse ist farblich genau auf den Amiga abgestimmt, ist etwa 4cm breit, und so hoch wie der Amiga.

Der Bus ist durchgeschleift und versperrt somit keinen anderen Erweiterungen den Kontakt zum Rechner.

Einen Schalter zum Abschalten sucht man übrigens vergeblich, abschalten läßt es sich nur mittels Software, die leider nicht beiliegt. Ein solches Programm befindet sich in unserer PD-Sammlung (KICKSTART-PD 71).

Die Innereien des Starboards sind ordentlich aufgebaut. Während man bei anderen Erweiterungen desöfteren ein Kabelwirrwarr vorfindet, weil die zwei Seiten der Platine nicht mehr zum Rooten ausreichen, findet man hier gerade 2 oder 3 Brücken.

Die Aufrüstung auf mehr als 1MByte erfolgt durch eine weitere Platine, die wiederum 32 zusätzliche RAMs aufnimmt. Da man hierzu lediglich zwei Schrauben entfernen und die neue Platine aufstecken muß, kann auch jeder Elektronik-Laie diese 'Operation' durchführen.

Der Bus ist, wie es sich für eine gute Erweiterung gehört, durchgeschleift und Tests ergaben, daß das Starboard im Gegensatz zu machen anderen Erweiterungen für andere Geräte praktisch 'durchsichtig' ist.

Der Geschwindigkeitstest ergab, daß das Starboard ohne Waitcyclen läuft, also mit 'Maximalgeschwindigkeit', es brems somit den Amiga in keiner Weise.

AUF DER TESTSTRECKE...

Nach dem Anschließen an den Bus waren wir doch recht gespannt, wie sich das Starboard verhalten würde. Doch vorweg gesagt: Es gab weder positive noch negative Überraschungen, das Starboard verhält sich wie sich eine gute RAM-Erweiterung zu verhalten hat.

Von DPaint II, das nun (endlich) auch 640x256 bei 4 Bitplanes, sowie einem 2. Schirm ohne Murren über sich ergehen läßt, über ProfessionalPageSetter bis hin zu diversen Spielen, läuft alles was 'gut' programmiert ist, d.h. alles was darauf achtet, daß Grafik wirklich im Chip-RAM liegt.

Auch diverse reset-feste RAM-Erweiterungen (ASDG, Card [WB1.3]) hatten keine Schwierigkeiten mit dem Starboard; sie blieben auch weiterhin reset-fest, wodurch man von einer in die Card: kopierten Diskette booten konnte. Da stellt sich natürlich die Frage, welche schlechten Seiten überhaupt vorhanden sind? Nun, in unserem Test fanden wir eigentlich überhaupt nichts, das laufen sollte und nicht gelaufen wäre.

DAS MULTIFUNKTIONS-MODUL

Diese Karte trägt das Wort 'Multi' mit vollem Recht. Neben einer batteriegepufferten Uhr, hat man mit dieser Karte die Möglichkeit, das Fast-RAM einem Parity-Check zu unterziehen, womit der Benutzer bei einer Speicherverfälschung auf einen möglichen Fehler in den Daten hingewiesen wird. Dazu kommt die Möglichkeit, die gesamte Erweiterung zu einer resetfesten RAM-Disk zu machen, die im Falle eines Absturzes hardwaremäßig gegen falsche Daten am Bus geschützt ist. Das Non plus ultra der Karte ist jedoch etwas ganz anderes - der Steckplatz für den Mathe-Coprozessor 68881. Da dieser einfach mittels einer neuen Mathe-Library angesprochen wird, braucht man diese nur auf die betreffende Diskette zu kopieren. Da zu erwarten ist, daß der '881 demnächst im Preis ein wenig fällt, ergeben sich hier ganz neue Dimensionen...

DAS SCSI-INTERFACE

Eine weitere Erweiterung für den freien Steckplatz ist ein ziemlich billiges Sesi-Interface für Macintosh-kompatible Festplatten. Es enthält ebenfalls eine Uhr, wofür auf Diskette diverse Programme zur Steuerung mitgeliefert werden.

STARBOARD II - DIE ERWEITERUNG

Starboard ist im Augenblick wohl das Allerbeste, was man seinem Amiga antun kann (außer einer 70Meg Hard-disk...). durch seine offene Struktur sowie der guten Qualität ist es eine Erweiterung, die allen Ansprüchen gerecht wird und jedem Amiga-Besitzer die Freudentränen in die Augen treiben kann. Damit es nicht nur den 1000er Besitzern zugute kommt, gibt es auch ein Starboard für den 500er, bei dem ein neues Gehäuse verwendet und der Bus nach oben durchgeschleift wurde. Für die 2000er Besitzer gibt es eine kleine Zusatzsteckkarte für den 'Zorro-Bus', an die die Starboard-Platine (ohne Gehäuse) gesteckt wird. Somit könnten alle Amigas zu 'ihrem' Starboard kommen.

Bezugsquelle: Soltau Electronic KG / Esplanade 39 / 2000 Hamburg 36 / Tel. 040/340445

ENDE

USPA II

- Kickstartumschaltplatine für A 500/2000
- originales und ein weiteres Betriebssystem
- einbaufertig ohne Eproms

DM 58,—

MTR 512

- statische 512 KB Ramerweiterung für A 1000
- in 64 KB Schritten aufrüstbar
- durchgeführter Bus
- kompl. aufgebaut ohne Rambausteine (16 Stück) DM 139,—

Wir liefern die gesamte Amiga Hard + Softwarepalette.
Fordern Sie unseren Amiga-Katalog an.
Händleranfragen erwünscht!

ACHTUNG — NEUE ANSCHRIFT!

TRÖPS + HIERL

Computertechnik GmbH
Jordanstraße 3 · 5040 Brühl · Telefon (0 22 32) 4 50 18



DER DRUCK- DOLMETSCHER

Druckertreiber-Generator V2.2

Vielleicht ist es Ihnen auch schon mal passiert, daß der Ausdruck Ihres Schriftstückes, das Sie mit viel Mühe mit Ihrer Textverarbeitung erstellt haben, nicht so aussieht, wie es eigentlich hätte sein müssen. Schuld daran sind weder Sie noch der Drucker, sofern dieser die gewählten Schriftarten beherrscht, sondern der Druckertreiber, der bei PREFERENCES eingestellt werden kann.

Ein Druckertreiber ist ein Programm, welches die einheitlichen Befehle des Amiga an den Drucker in eine dem Drucker geläufige "Sprache" (Kontrollsequenzen) übersetzt. Da nun fast jeder Drucker-Hersteller seinem Produkt eine eigene Sprache mit auf dem Weg gibt, gibt es natürlich auch eine Menge an Druckertreibern. Trotzdem kommt es vor, daß nicht der richtige Treiber für Ihren Drucker vorhanden ist.

Erste Hilfe

Doch man kann sich helfen. Schreiben Sie sich Ihren eigenen Druckertreiber. Das Utility "PrtDrvGen" nimmt Ihnen dabei die Hauptarbeit ab. Das ganze Programmpaket besteht aus mehreren Files, wobei PrtDrvGen und PrtDrvGen2 die wichtigsten sind. Die anderen Files werden entweder nachgeladen oder dienen als kurze Ein- und Anleitung. Bei PrtDrvGen handelt es sich um einen speziellen, aber komforta-

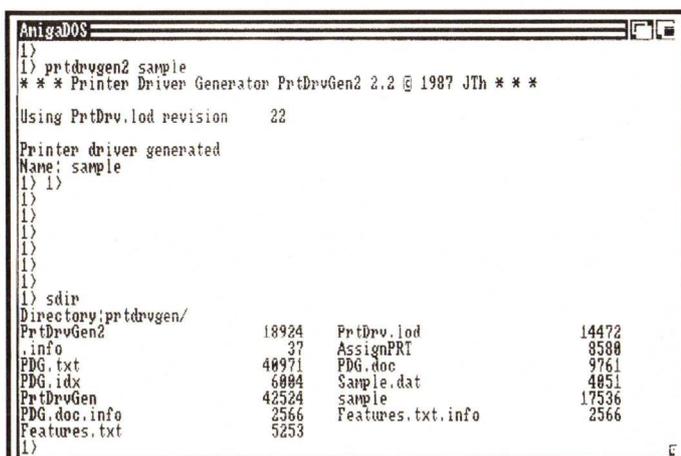
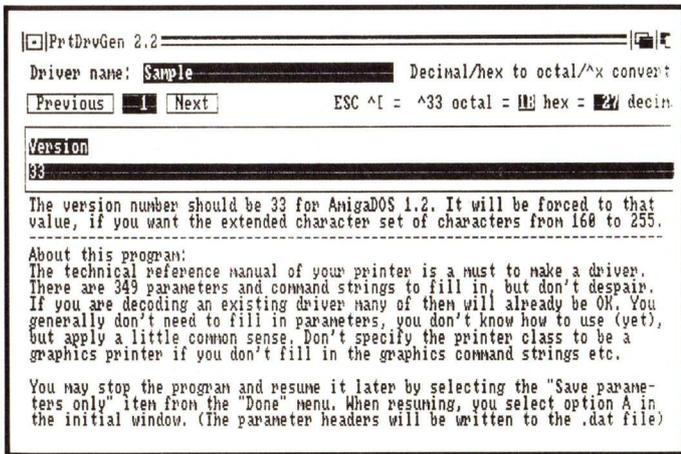
blen Editor. Dort werden die Kontrollsequenzen und andere Mitteilungen für den Treiber eingegeben. Der Editor erzeugt anschließend ein sogenanntes Datenfile, in dem alle Eingaben der Reihe nach aufgeführt sind. Aus diesem Datenfile generiert dann der Compiler PrtDrvGen2 den fertigen Druckertreiber.

Doch nun an die Arbeit. Das Utility liegt im Unterverzeichnis PrtDrvGen und kann nur vom CLI aus gestartet werden. Rufen Sie zunächst den Editor mit "PrtDrvGen" auf. Beim ersten Aufruf wird erst noch ein File Namens "PDG.idx" kreiert. Sollten Sie während dieses Vorgangs den AMIGA zurücksetzen, so müssen Sie beim nächsten Mal erst dieses File von der Diskette löschen. Andernfalls funktioniert der Editor nicht richtig. Wenn alles korrekt abgelaufen ist, müssen Sie sich entscheiden, ob Sie ein vorhandenes Datenfile benutzen bzw. verändern wollen, oder ob Sie einen existierenden Druckertreiber bearbeiten wollen.

Wählen Sie ersteres, so können Sie bei Erstbenutzung auf das File "Sample.dat" zurückgreifen, welches die richtigen Parameter für NEC-Drucker enthält. Normalerweise müssen Sie dem Programm auch den Namen des Datenfiles bzw. des Treibers verraten. Dies tun Sie, indem Sie oben neben "Driver name" das Feld anklicken und den vollständigen Name des Files mit allen Unterdirectories angeben. Bei Datenfiles müssen Sie jedoch das Kürzel ".dat" am Ende des Namens weglassen. Nun klicken Sie entweder das "A" für das Bearbeiten von Datenfiles oder das "B" für das Ändern von Druckertreibern mit der Maus an.

Die Parameter

Wenn Sie Ihren eigenen Druckertreiber schreiben wollen, dann nehmen Sie am besten den, mit dem Sie die besten Erfahrungen gemacht haben, denn dort müssen Sie am wenigsten ändern. Wenn wir nun einen fertigen Treiber oder ein Datenfile geladen haben, ändert sich das Erscheinungsbild des Editors. So erscheinen zunächst die zwei Felder "Previous" und "Next" mit einer Eins dazwischen. Die Eins zeigt an, daß es sich hier um den ersten Parameter des Treibers handelt. Durch Anklicken von "Previous" und "Next" kann man den vorherigen und den nächsten Parameter auswählen. Will man mehrere Parameter überspringen, so klickt man die Zahl an und ändert sie auf die Nummer des gewünschten Parameter. Unter dieser Zeile befindet sich jeweils eine kurze Definition über die Funktion dieses Parameters. So steht in Pa-



parameter 48: "SGR0 ^[[0m normal character set". "SGR0" ist ein Synonym des Befehls, den der Amiga an den Druckertreiber sendet. Dieser Befehl lautet ESC [0m. Daraus läßt sich schließen, daß ^[gleich ESC gleich CHR\$(27) ist.

In die nächste Zeile werden die Parameter bzw. Kontrollsequenzen eingetragen. Dies geschieht wieder durch Anklicken. Ziehen Sie hierbei Ihr Druckerhandbuch zu Rate. Auch hier muß für ein ESC ein ^[geschrieben werden. Um eine Änderung auch zu übernehmen, müssen Sie die Eingabe, wie bei allen anderen auch, mit RETURN abschließen. Alle Zahlenwerte sind im etwas ungebräuchlichem oktaalen Zahlensystem einzugeben. Jedoch steht hierfür rechts oben ein Konverter zur Verfügung, der Ihnen diese etwas lästige Arbeit abnimmt. Auf dem übrigen Platz des Bildschirms werden Erklärungen zum jeweiligen Parameter in englischer Sprache gemacht. Diese

können aber auch von denjenigen verstanden werden, die mit dieser Sprache nicht so befreundet sind. Man kann z.B. auch einen Treiber kreieren, der keine Grafik unterstützt, was sich in der Länge des Treibers niederschlägt. Dies hängt alles von den Parametern ab, für die Sie sich entscheiden. Den

```

iton$=CHR$(27)+"[3m"      :REM Parameter 49
itoff$=CHR$(27)+"[23m"   :REM Parameter 50
ulon$=CHR$(27)+"[4m"     :REM Parameter 51
uloff$=CHR$(27)+"[24m"  :REM Parameter 52
OPEN "prt:" FOR OUTPUT AS #1
PRINT #1,"Normalschrift (Pica)"
PRINT #1,ulon$;"Das ist unterstrichen."
PRINT #1,iton$;"Das ist unterstrichen und kursiv."
PRINT #1,uloff$;"Und nur noch kursiv.";itoff$
PRINT #1,"Normalschrift (Pica)"
CLOSE #1

```

Editor können Sie über das Auswahlménü verlassen. Dabei können Sie sich gleich einen fertigen Treiber erstellen lassen oder die Daten in einem Daten-

file zwischenspeichern, um sie später zu vervollständigen. Es ist allerdings ratsam, den Druckertreibernamen vor dem Abspeichern zu ändern, um nicht den ursprünglichen Treiber zu löschen.

Funktionstest

Alles in allem ist dieses Programmpaket sehr hilfreich und bequem zu bedienen, allerdings kommt man nicht umhin, sich ein kurzes Testprogramm in Basic zu schreiben, um den neuen Druckertreiber zu testen. Als Beispiel soll folgendes Programm dienen. Sie sollten es aber mit möglichst vielen Druckerbefehlen ausprobieren. Die Druckerbefehle entnehmen Sie bitte der obengenannten Spalte.

Den hier beschriebenen Druckertreiber finden Sie auf der KICKSTART PD-Diskette 71.

ENDE

AZTEC-C FÜR AMIGA VERSION 3.6

Wußten Sie, daß eines der verbreitetsten und komplexesten Betriebssysteme - UNIX - in C geschrieben ist ?

**NEU!
V.3.6**

Wußten Sie, daß auch das Betriebssystem des AMIGA größtenteils in C geschrieben wurde ?

Wußten Sie, daß C eine der wichtigsten und modernsten Programmiersprachen ist ?

Wußten Sie, daß Aztec-C einer der schnellsten und leistungsfähigsten Compiler für den AMIGA ist ?

Wußten Sie, daß jetzt Aztec-C in der Version 3.6 verfügbar ist ?

Möchten Sie mehr darüber wissen?
Dann schicken Sie uns einen ausreichend frankierten Rückumschlag und Sie erhalten ausführliche Information.

Up-Date-Service für alle MANX-Kunden auch bei uns. Fragen Sie nach!

HIERMIT BESTELLE ICH:

- AZTEC-C68K/AM-P
PROFESSIONAL SYSTEM FÜR DM 398.-
- AZTEC-C68K/AM-D
DEVELOPER SYSTEM FÜR DM 598.-
- AZTEC-SDB SOURCE
LEVEL DEBUGGER FÜR DM 149.-

Versandkosten: Inland DM 7,50 Ausland DM 10,-
Auslandbestellungen nur gegen Vorkasse
Nachnahmegebühr DM 3,70

- Vorkasse
 Nachnahme

NAME: _____

VORNAME: _____

STRASSE: _____

ORT: _____

UNTERSCHRIFT: _____



MERLIN COMPUTER GMBH
INDUSTRIESTRAÙE 26
6236 ESCHBORN
TEL. 06196/481811

SICHER IST SICHER

Zwei Backup-Programme im Vergleich

Der Alptraum eines jeden Festplattenbesitzers ist ein Fehler der Festplatte, ohne eine Sicherheitskopie dieser zu besitzen. Den Qualen und Mühen eines Backups von Hand wollen zwei Backup-Programme ein Ende setzen. Ein Vergleich dieser beiden Programme steht in diesem Bericht an.

In letzter Zeit ist die Zahl der Betreiber von Festplatten am AMIGA stark angestiegen. Sinkende Hardwarepreise und steigende Verbreitungszahlen des Bridgeboard mit der dadurch gegebenen preisgünstigen Aufrüstungsmöglichkeit haben dazu beigetragen. Hat man die Platte angeschlossen und sie arbeitet tadellos, denkt man erst einmal nicht an das Auftreten eines Festplattenfehlers. Nach und nach sammelt sich so manches auf der Festplatte an, was man selbst erarbeitet hat und nur auf der eigenen Festplatte findet. Texte, selbstgeschriebene Programme und dergleichen werden meist nur auf der Festplatte abgespeichert, an das Sichern auf Disketten denkt wohl kaum jemand. So sammeln sich leicht im Laufe der Zeit die Früchte mehrerer hundert Stunden Arbeit auf der Platte an. Falls es nun zu einem Festplattenfehler kommt, sind diese Daten womöglich alle unwiderruflich verloren. Solch ein Verlust kann schnell zur Katastrophe auswachsen, wenn sich wichtige Ge-

schäftsunterlagen auf der Platte befunden haben.

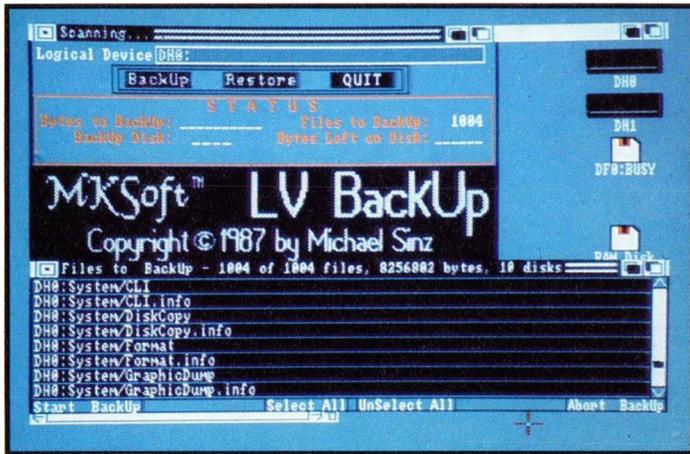
Es steht wohl außer Frage, daß man sich vor solchem Unheil schützen sollte. Als Lösung würde sich erst einmal die Sicherung der Dateien auf ganz normale Disketten mit Hilfe des Copy-Befehls anbieten. Bei Datenbeständen von bis zu 20 MByte kann das jedoch eine abendfüllende Tätigkeit werden. Außerdem ist es äußerst schwierig, die Subdirectory-Strukturen zu erhalten. Wesentlich schnelleres Arbeiten versprechen nun die beiden Backup-Programme LV Backup von MKSoft und QUARTERBACK von Central Coast Software.

LV BACKUP

Erster Kandidat ist das nicht kopiergeschützte LV Backup. Ausgeliefert wird es in einer CD-Hülle inclusive eines kleinen zehneitigen Handbuchs. Neben Installationshinweisen und der Bedienungsanleitung enthält dieses Heft noch einige Tips zu logischen

Volumes und zur allgemeinen Datensicherheit auf Disketten.

Das Programm kann sowohl vom CLI als auch von der Workbench gestartet werden. Nach dem Starten des Programms erscheint zuerst ein Fenster, das Raubkopierer auf ihre Unrechtstat hinweist. Nach dem Wegklicken dieser Meldung erscheint das eigentliche PC-Backup-Fenster. Die Auswahlmöglichkeiten sind nicht sehr umfangreich ausgefallen. Neben dem zu sichernden Device ist es nur noch möglich, die Verwendung des ARCHIVE-Bits anzuzeigen und auszuwählen, ob man 'backups' oder 'restoren' will. Als Device ist jedes logische Device zulässig. So kann man jeden beliebigen Bereich von Disketten, RAM-Disks und natürlich Festplatten sichern. Voreingestellt ist das Device "DH0:". Ein Tip am Rande: Wenn Sie Ihrer Janus-Festplatte den Namen "DH0:" geben (mit dem Menüpunkt "Rename" aus dem Workbench-Menü), brauchen Sie die Voreinstellung des Device nicht zu ändern. Außerdem arbeiten dann auch viele Programme mit der PC-Festplatte zusammen, die von Natur aus nur auf AMIGA-eigene Platten zugeschnitten sind (wie zum Beispiel CLImate, Aegis Audio Master oder Deluxe Music). Wählt man "Backup" an, so katalogisiert LVB erst einmal alle Files des Device. Dies kann bei größeren Geräten schon im Minutenbereich dauern. Daraufhin erscheint ein Fenster, in dem alle Files der Diskette angezeigt werden. Hier tritt schon ein Manko von LVB ans Tageslicht: Es können immer



So zeigt sich LV Backup dem Benutzer.

nur acht Dateinamen angezeigt werden. Dabei wird jede Datei mit dem kompletten Pfadnamen angezeigt. Eine Selektion kompletter Verzeichnisse ist nicht möglich. Es besteht nur die Möglichkeit, alle Files zu selektieren oder zu deselektieren. Will man aber ein Unterverzeichnis vom Backup ausschließen, so muß man jedes(!) File extra anklicken. Bei einem Compiler mit acht Subdirectorys und bis zu 100 Files je Unterverzeichnis ist dies ein netter Zeitvertreib. In der Fenster-Ziehleiste werden dabei ständig die zum Backup benötigten Disketten, die Anzahl der selektierten Dateien und deren Gesamtgröße angezeigt. Nach dem Starten des Backups fordert LVB den Bediener auf, die Disketten in Laufwerk DF0: einzulegen. Ein eventuell vorhandenes zweites Laufwerk wird nicht unterstützt. Die Disketten müssen nicht formatiert sein. Während des Backups wird man ständig über die verbleibende Diskettenkapazität und die Anzahl und Größe der noch zu sichernden Dateien informiert. Auf den Diskettenwechsel wird der Benutzer durch einen Requester und einen Pieps hingewiesen. Man sollte aber darauf achten, daß man auch eine Diskette ins Laufwerk schiebt, bevor man das Kommando zum Weiterarbeiten gibt, denn andernfalls rattert das Diskettenlaufwerk mit einem ohrenbetäubenden Lärm los, daß einem Angst und Bange um seine Hardware wird. Dieser Vorgang läßt sich auch nicht abbrechen. Die Backup-Disketten können nicht von AMIGA-DOS gelesen werden. LVB schreibt aber in die Diskettenkennung die Folgenummer (B001 für die erste, B002 für die zweite Diskette usw.) der Diskette. So ist es möglich, die Disketten vom AMIGA-DOS aus

zu identifizieren. Bei 20 Backup-Disketten (soviele sind für eine fast volle 20 MByte-Platte schon nötig) ist es schon von Vorteil, diese von der Workbench aus sortieren zu können, wenn sie einmal durcheinander geraten sind. Beim Restoren verhält sich LV Backup genau wie beim Backupen. Die Fenster und Menüs sind identisch. Weder beim Backup noch beim Restore traten mit LVB Probleme auf. Auch Dateien, die größer als die Disketten-Kapazität sind, werden fehlerfrei gehandhabt.

Quarterback

Zweiter Streiter um die Gunst des Käufers ist Quarterback von Central Coast Software aus Californien. Zum Lieferumfang gehören hier ein Handbuch und eine einseitige(!) Diskette. Über die Qualität von einseitigen Disketten kann man sich streiten, der Preis von QB ist aber nicht so knapp angesetzt, daß die Verwendung einer doppelseitigen Diskette die Firma in die roten Zahlen brächte.

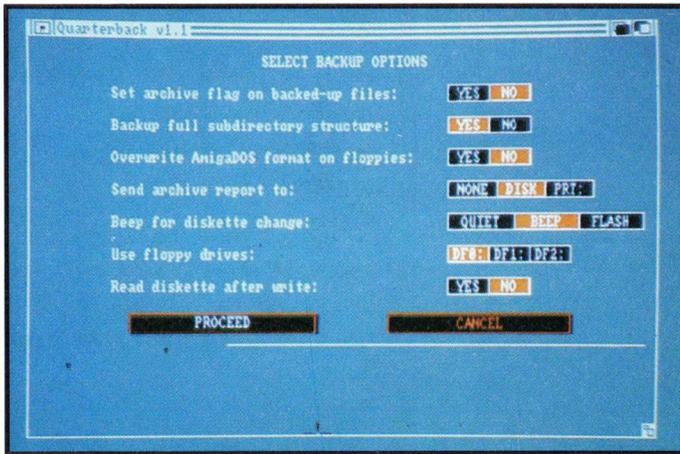
Der File-Selector von Quarterback.



Quarterback ist ebenfalls nicht kopiergeschützt, kann also leicht auf die Festplatte kopiert werden. Im Handbuch wird ausdrücklich darauf hingewiesen, eine Kopie von QB auf Diskette zu besitzen, da man im Ernstfall ja nicht mehr auf die Datei auf der Festplatte zugreifen kann. Auch im weiteren Verlauf des Handbuchs merkt man, daß der Autor der Bedienungsanleitung sich über die reine Benutzung des Programms hinaus einige Gedanken gemacht hat. So werden verschiedene Vorgehensweisen zur Archivierung dargestellt und auf eventuelle Schwierigkeiten (von anderen Programmen allozierte Dateien beim Restoren usw.) eingegangen.

Wie LVB kann auch QB von der Workbench oder vom CLI aus gestartet werden. Nach der Abfrage, ob ein Backup erwünscht oder ein Restore angesagt ist, muß man das Device angeben, das verwendet werden soll. Es können auch bei Quarterback alle Arten von Devices benutzt werden.

Bei der Selektierung der Dateien erweist sich QB als äußerst komfortabel. In einem gegenüber LVB wesentlich größeren File-Selector werden die Dateien in ihrer Directory-Struktur angezeigt. So ist es mit einem Mausklick möglich, ganze Verzeichnisse zu selektieren oder zu deselektieren. Außerdem bietet QB noch die Möglichkeit, Dateien nach verschiedenen Kriterien ein- oder auszuschließen. Neben der Selektierung über das Archiv-Bit ist es auch möglich, über das Erstellungsdatum der Dateien eine Auswahl zu treffen. Ebenfalls besteht die Option, den Dateinamen (inclusive der Verwendung von Wildcards) als Selektionskriterium heranzuziehen. Auch bei QB



Quarterback bietet reichhaltige Optionen beim Backup.

Auf die Plätze,...

Ein wesentlicher Punkt bei der Anwenderfreundlichkeit eines Backup-Programms ist die Geschwindigkeit. Nur ein schnelles Backup wird man regelmäßig betreiben. Zum Test wurde eine zu etwa 9 Mbyte gefüllte Janus-Platte gesichert. Bei Quarterback wie bei LV Backup wurden etwa 20 Minuten benötigt. Zu beachten ist dabei, daß Quarterback mit Verify sicherte. Diese Auswahl wurde getroffen, da diese Option unbedingt eingeschaltet werden sollte, um eine größtmögliche Datensicherheit des Backups zu gewährleisten. Ohne Verify liegt die benötigte Zeit etwa 30 % niedriger. Bei der benötigten Zeit kommt es unter anderem darauf an, ob hauptsächlich große oder sehr kleine Dateien auf der Platte sind. Je nach Gegebenheit kann das bis zu 100% Differenz zwischen den Zeiten führen. Beide Programme benötigten 9 Disketten, wobei Quarterback auf der letzten Diskette noch etwas mehr Freiraum hatte. Sowohl LV Backup als auch Quarterback können aber wesentlich mehr Daten aufnehmen, als dies eine DOS-Diskette könnte. Beide speichern die Dateien nicht sektororientiert, was sich bei vielen kleinen Dateien sehr platzsparend bemerkbar macht. Der Freiraum im Sektor hinter dem Dateieinde entfällt damit und wird für die nächste Datei genutzt. Im Extremfall (40 Dateien à 2 Byte) ist das Verhältnis 1 Sektor zu 85 Sektoren.

Fazit

Beide Programme erledigen ihre Arbeit korrekt und zuverlässig. Im direkten Vergleich zeigte sich QuarterBack als das eindeutig zu bevorzugende Pro-

werden die benötigten Disketten, die Anzahl der selektierten Dateien und deren Gesamtgröße angezeigt.

Hat man diesen Punkt abgeschlossen, gelangt man in das nächste Menü. Hier hat man nun diverse Optionen zur Auswahl, wie Quarterback beim Sichern zu verfahren hat. Dabei zeigen sich drei gravierende Vorteile von QB. Der erste und wichtigste Punkt ist die Möglichkeit, die Überprüfungen der geschriebenen Backups anzuordnen. Dies trägt ganz erheblich zur Sicherheit eines Backups bei. Des Weiteren ist es möglich, neben DF0: auch DF1: und DF2: zur Sicherung heranzuziehen. Im täglichen Gebrauch zeigt sich dies als sehr komfortabel. Dritter Pluspunkt ist die Option, beim Einlegen einer AMIGADOS-formatierten Diskette die Abfrage zu ermöglichen, ob diese Diskette auch wirklich beschrieben werden soll. Ein versehentliches Überschreiben wichtiger Dateien ist so fast ausgeschlossen.

Daneben stehen noch einige Punkte zur Auswahl, wie das Setzen des Archiv-Flags, die Sicherung der kompletten Verzeichnisstruktur oder die Erstellung eines Sicherungs-Kontrollblattes. Alle Einstellungen können in einer Datei gespeichert werden, damit sie beim nächsten Backup erhalten bleiben. Übrigens kann von Quarterback aus jederzeit ein CLI eröffnet werden, was unter Umständen sehr nützlich sein kann.

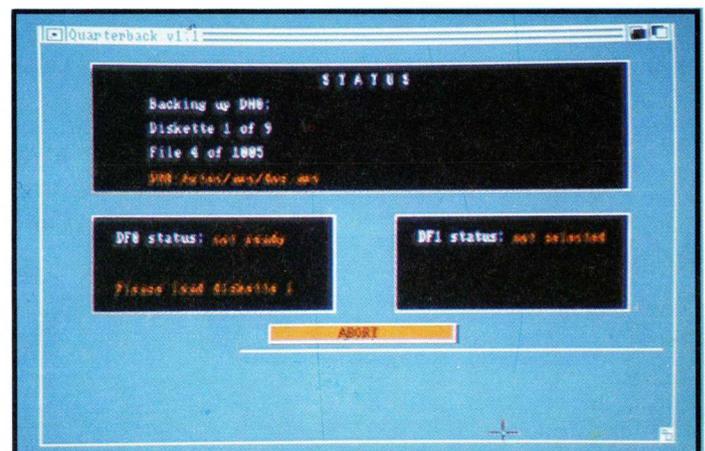
Beim eigentlichen Backup wird die Anzahl der bereits gesicherten Dateien angegeben und die Folgenummer der Diskette angezeigt. In Fenstern wird der Status der selektierten Floppystationen angezeigt. Hat man zwei Diskettenlaufwerke ausgewählt, wird automatisch zwischen beiden umgeschal-

tet. Dadurch wird das Backup beschleunigt, da die Zeit des Diskettenwechsels nicht mehr zu Buche schlägt. Die Disketten werden mit einem eigenen Format beschrieben, wodurch eine höhere Kapazität der Disketten erreicht wird. Da QB nur die Kennung "NDOS" auf die Disketten schreibt, ist es vom DOS aus nicht möglich, die Disketten genauer zu identifizieren. Für Leute, die nichts von Labels halten, ist dies sicherlich ein Manko.

Das Sichern übergroßer Dateien bereitete auch hier keine Probleme.

Das Zurückschreiben funktioniert bei QB in ähnlicher Weise wie das Sichern. Die Selektionsmöglichkeiten sind die gleichen. Daneben kann man noch auswählen, ob nur bestehende Dateien zurückgeschrieben werden sollen und ob bestehende Files überschrieben werden. Beim Restore wird nur das interne Laufwerk benutzt, was aber bei der (hoffentlich) seltenen Notwendigkeit eines Restore kein großer Nachteil ist.

Quarterback arbeitet auch mit 2 Laufwerken.



gramm. Als wichtigsten Punkt muß man hier wohl die geschwindigkeitsmäßige Überlegenheit von Quarterback aufführen. Das überaus schnelle Sichern inclusive Verify läßt den Benutzer erst regelmäßig ein Backup machen. Die fehlende Schreib-Überprüfung bei LV Backup ist meines Erachtens ein zu großes Sicherheitsrisiko. Doch nicht nur bei der Geschwindigkeit hat Quarterback die Nase vorn. Bedienungskomfort und Möglichkeiten des Programms liegen weit über dem von LV Backup gebotenen Standard. Als Vorteil von LV Backup ist zu werten, daß die Disketten auf der Workbench als Backup-Disketten zu erkennen und in ihrer Reihenfolge sortierbar sind. Im dreimonatigen Gebrauch hat sich bei mir jedoch ein eindeutiger Favorit herauskristallisiert. Und dieser heißt Quarterback.

Quarterback

- + schnelles Backup
- + sehr komfortabel
- + nicht kopiergeschützt
- + Unterstützt mehrere Diskettenlaufwerke

- Disketten nicht vom DOS erkennbar

Preis: DM 135,-

Anbieter: GTI

LV Backup

- + Disketten von DOS
- + identifizierbar
- + nicht kopiergeschützt

- keine Verifizierungsmögl.
- unkomfortable Fileselektion
- arbeitet nur mit DFÜ

Preis: DM 120,-

Anbieter: GTI

ENDE

Das Amiga-Drive

Mit dem NEC-Laufwerk FD 1037 A

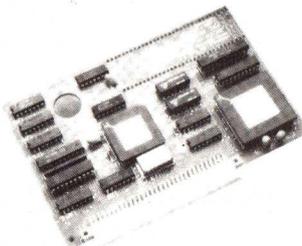


Das 3 1/2" Kompaktgerät mit dem NEC-Diskettenlaufwerk FD 1037 A. Mit Disk-Change-Erkennung und Ein-/Aus-Schalter. Abgeschirmtes 70 cm langes Anschlußkabel. Durch Linear-Steppermotor kaum noch hörbar. Lackierung im Original-Amiga-Farbton. Maße: 104 x 29 x 165 mm. Für Amiga 500/1000/2000, Sidecar und PC-1.... Sofort ab Lager.
AGS 3701, Edelstahl lackiert DM 278,-
AGS 3702, " geschliffen DM 275,-

Elektronik-Zubehör OHG · Werwolf 54
5650 Solingen 1 ☎ 02 12/13084

Versandkostenpauschale bei Lieferung durch
Nachnahme DM 7,- oder Vorkasse DM 4,-
Ladenverkauf Mo.-Fr. 9.00-18.30 Uhr

Hurricane! DM 1998,-



- Für AMIGA 500, 1000, 2000
- Amiga läuft mit hohem Systemtakt (16 od. 20 MHz)
- Superschnell: bis 1000% schneller.
- Schnellstes Turbo-Board auf dem Weltmarkt!!!
- 32 Bit-Ram (100 ns)

nur DM 2498,-



Borsigallee 18
6000 Frankfurt/M.
☎ 069-41 0071/72

Schweiz:
MICROTRON
Bahnhofstraße 2
CH-2542 Pieterlen
Tel 032 87 24 29

Aztec C Prof. V3.6 DM 299,00
2 MB-Rambox A1000/A500 DM 899,00
MCC-Shell + Toolk + Assemb. DM 239,00
Digi-View Pal/Deutsch DM 309,00
Pawn / Barbarian / Guild je DM 49,95
Source Level Debugger DM 119,00
Aztec C Dev. V3.6 DM 399,00
Archon 2 / Seven Cities je DM 29,00
PD-Disketten (Fish, Auge...) .. ab DM 3,00

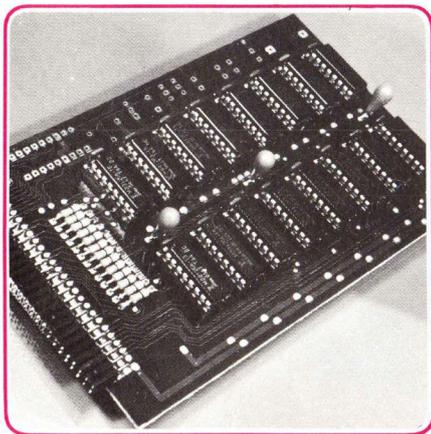


Computerversand CWTG Joachim Tiede
Bergstraße 13 · 7109 Roigheim
Tel. BTX 0 62 98/30 98 von 17 - 19 Uhr
Schriftliche Händleranfragen erwünscht!

ausgereifte Ingenieurleistung ● 14 Tage Umtauschrecht ● 2 Jahre Garantie ● fast alle IC'S gesockelt ● nur professionelle Leiterplatten ● Bauteile namhafter Hersteller ● mit Bedienungsanleitung ● Blockschaltbild ● teilweise Schaltplan



ALCOMP
COMPUTERHARDWARE



500er Speichererweiterung

Für 512k zusätzliches RAM ● alle RAM's gesockelt ● selbstkonfigurierend ● abschaltbar ● Uhrenschaltung auf Platine mit Akku- bzw. Batteriepufferung nachrüstbar

Komplett mit 512k **Preis auf Anfrage**
Superpreis mit Uhr **Preis auf Anfrage**
Bauteilesatz für Uhr ohne Akku **24.-**
Leerplatine mit Stecker ***39.-**

*mit Schaltplan und Bestückungsliste



Profilaufwerk 3,5"

Metallgehäuse ● einstellbare Laufwerknummer mit Displayanzeige ● digitale Trackanzeige ● Write Protect am Laufwerk schaltbar ● abschaltbar ● durchgeschleifter Bus
1 Jahr Garantie

SuperALCOMPPreis **349.-**

Trackanzeige

Für DF0-DF3 einstellbar ● für alle Laufwerke (3,5"/5,25") ● Laufwerkbus durchgeschleift ● mit Gehäuse

SuperALCOMPPreis **49.-**

Wir suchen ständig Hardware-Entwicklungen. Wir garantieren gute Umsatzprovisionen und ehrliche Abrechnung

3,5" Laufwerk

Für alle Amiga's ● einstellbare Gerätenummer ● abschaltbar ● Metallgehäuse ● superflach ● 1 Zoll (2,54 cm) ● durchgeschleifter Bus ● TEAC Laufwerk
1 Jahr Garantie

komplett anschlussfertig **279.-**

Laufwerk 5,25"

40/80 Track ● Laufwerkbus durchgeschleift ● abschaltbar ● einstellbare Adressen ● MS-DOS kompatibel ● mit Diskchange
1 Jahr Garantie

SuperALCOMPPreis **339.-**
HD 1,6 MB (umschaltbar) **359.-**

Gemischtes Doppel 3,5/5,25"

einzel ein-/abschaltbar ● einstellbare Laufwerknummern mit Anzeige ● durchgeschleifter Bus ● bei 5,25" 40/80 Tracks umschaltbar ● Metallgehäuse
1 Jahr Garantie

SuperALCOMPPreis **598.-**

Basislaufwerke

1 Jahr Garantie

TEAC FD 135 FN 3,5" 1MB superslimline **218.-**

TEAC FD 55 FR 5,25" 40/80 Tracks 1MB **239.-**

TEAC FD 55 FR 5,25" 40/80 Tracks 1,6MB Diskchange **259.-**

3,5" Gehäuse **25.-**

5,25" Gehäuse **25.-**

Gehäuse für "Gemischtes Doppel" **45.-**

Laufwerkanschlußkabel

Zum Anschluß von Laufwerken an alle Amiga's ● mit Ansteuerelektronik

Für 3,5" Laufwerk **39.-**

Für 5,25" Laufwerk **49.-**

Steckplatzerweiterung 3-fach für Laufwerke

Jeder Steckplatz abschaltbar und einstellbare Laufwerknummer ● Steckplatzerweiterung direkt am Amigagehäuse ● Dadurch keine Kabellängenprobleme

Anschlussfertig zum SuperALCOMPPreis **49.-**



Soundsampler

Für Amiga 1000 und 500 mit Software ● Type bei Bestellung bitte angeben ● 8-Bit Datenbreite ● Betrieb am Parallelport (Druckerport) ● Mit Vorverstärker für Micro-Anschluß (Cinch-Buchsen) ● Musik- und Sprachdigitalisierung möglich ● Arbeitet mit fast allen Digitizer-Programmen ● Formschönes Gehäuse

SuperALCOMPPreis **79.-**

MIDI-Interface

4 Kanäle einschließlich 1 Thru ● Optische Datenanzeige ● Formschönes Gehäuse

Wahnsinnspreis von nur **89.-**

Bootselector

19.90

Kickstartumschaltung

Bauen Sie die anderen Kickstart-Versionen in Ihren Amiga 500 ● Einfacher Einbau ohne Löten ● für Original-Kickstart-ROM und 2 zusätzliche Versionen auf EPROM ● EPROM-Programmierservice auf Anfrage

SuperALCOMPPreis **59.-**

Public-Domain-Disketten

Große Auswahl ● schnelle Auslieferung

Einzelstück **6.-**

Staffelpreise auf Anfrage

Bestellung und Versand

ALCOMP
A. Lanfermann
Lessing Str. 46
5012 Bedburg
Tel. 0 22 72/15 80

Nachnahmeversand NN-Spesen 7,50 DM b. Vorkasse 3.- DM. Auslandsbestellungen: Nachnahmeversand NN-Spesen 10.- DM b. Vorkasse 5.- DM. Wir liefern Ihnen auf Ihre Rechnung und Gefahr zu den Verkaufs- und Lieferbedingungen des Elektronikverwerbes. Postgroomt Köln (BLZ 370 100 50) 275 54-509

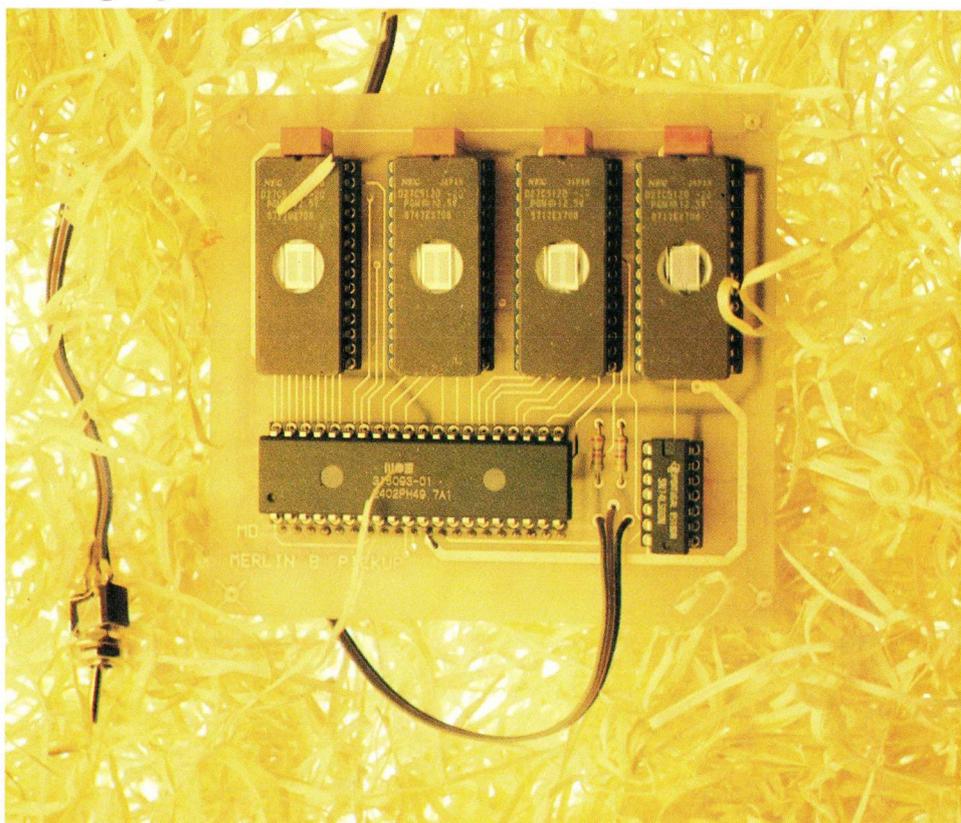
VON GERALD CARDA

DAS KICKSTART MODUL

für Amiga 500 und Amiga 2000

Sie ist endlich da! Die Kickstart und Workbench 1.3. Für Amiga 1000 Besitzer ist die Sache kein größeres Problem, man legt einfach die neue Kickstart in das interne Laufwerk und wartet genüsslich bis das neue Betriebssystem geladen ist. Die Anwender, deren Kickstart im ROM abgelegt ist, konnten bis jetzt nur lange Gesichter ziehen. Wir wollen dem hier Abhilfe schaffen. Lesen Sie selbst.

Der Betriebssystem-Kern des Amiga befindet sich bei den Modellen 500 und 2000 in einem ROM. Dies hat im täglichen Gebrauch einen entscheidenden Vorteil, denn man braucht das Be-



triebssystem nicht ständig zu laden. Was aber wenn sich die Betriebssystemroutinen ändern oder man selbst einige kleine Verbesserungen vornehmen will? Hier hatte der Anwender mit einem ROM-Kickstart bis jetzt wenig Chancen.

Die hier gezeigte Lösung basiert auf der Tatsache, daß sich der Inhalt des ROM auch auf vier EPROMs des Typs 27512 verteilen läßt. Diese EPROMs haben eine Kapazität von jeweils 64 KByte und erreichen so die Größe des ROM von 256 KBytes. Die hier gezeigte Platine geht noch einen Schritt weiter und enthält zusätzlich zu den angesprochenen vier EPROMs, auch noch das im Rechner vorhandene ROM, so daß bei Bedarf zwischen der stützen Variante umgeschaltet werden kann.

Die Platine paßt mit ihren Abmessungen sowohl in den Rechnertyp Amiga 500, als auch in einen Amiga 2000.

Das Betriebssystem-ROM der Amiga-Rechner findet in einem 40-poligen DIL (Dual In Line) Gehäuse Platz. Bild 1 zeigt die Pin-Belegung:

- 18 Adressleitungen A1 bis A18
- 16 Datenleitungen D0 bis D16
- 2 Masseleitungen
- 1 Versorgungsspannung
- 1 Chipselect
- 1 Outputenable
- 1 Pin unbelegt

Die 18 Adressleitungen überstreichen einen Bereich von 262144 Bytes und erlauben somit die Auswahl eines Adressbereichs von 256 KByte. Dem Prozessor stehen die Daten an den Datenleitungen D0 bis D15 zur Verfügung.

Die beiden Signale Chipselect und Outputenable dienen zur Ansteuerung des ROM. Liegen beide Signale auf Low, gibt der Chip die adressierten Daten auf den Bus. Die Steuerung die-

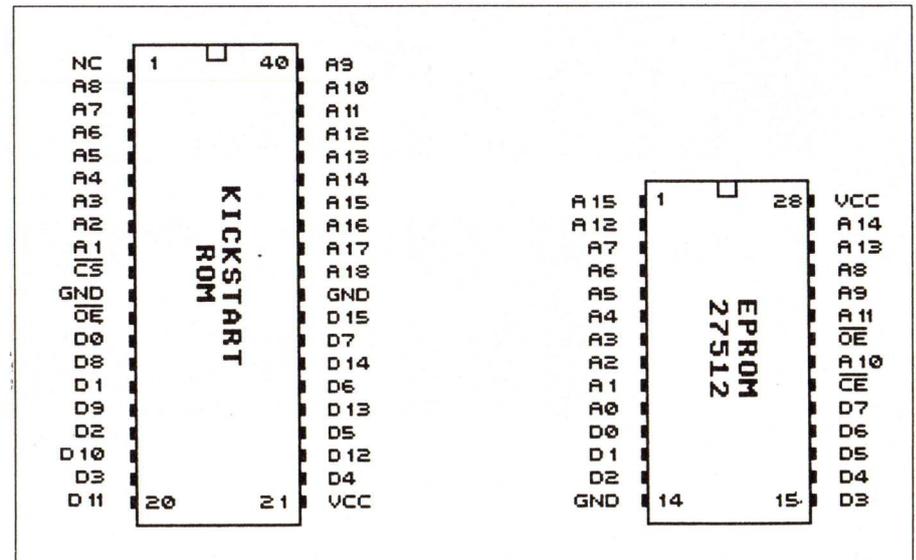


Bild 1: Das Pinout des ROM und der Eproms

Das Konzept

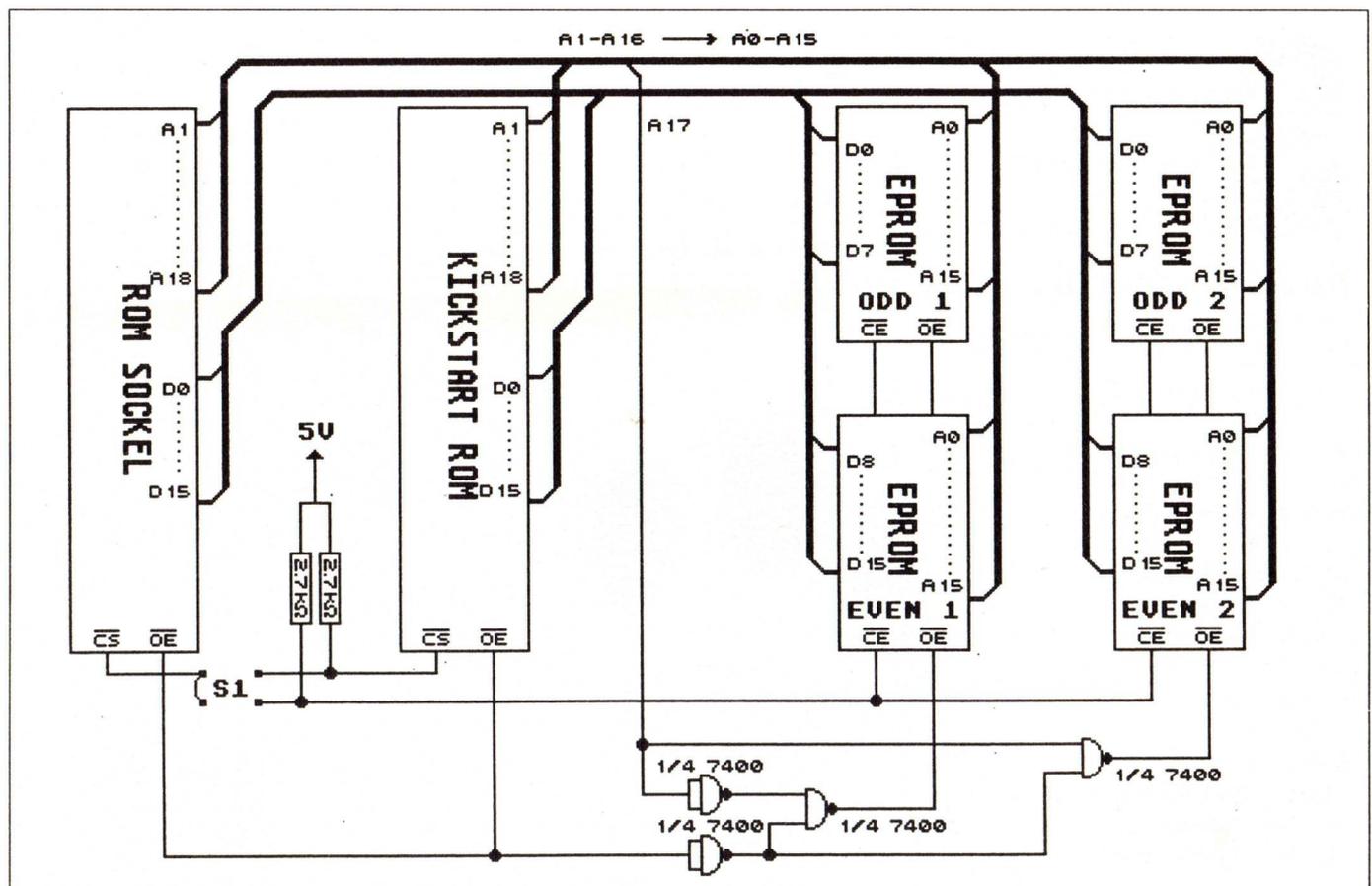


Bild 2: Das Schaltbild der Umschaltplatine

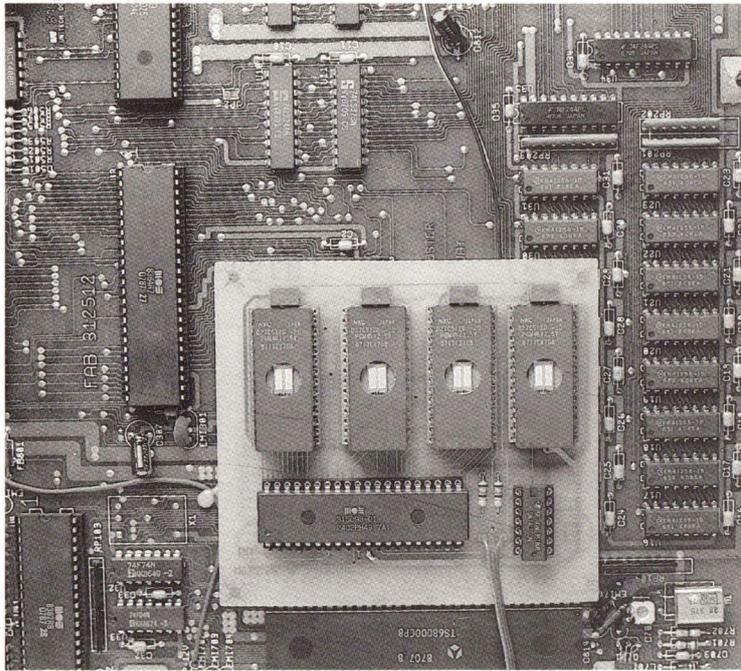


Bild 5: Die eingebaute Umschaltplatine im A500

ser Signale wird im Amiga von einem PAL vorgenommen. Das PAL hat die Aufgabe, den ROM-Baustein nur in einem bestimmten Adressbereich auf den Bus zu schalten. Da das ROM nur die Adressleitungen A1 bis A18 verwendet, würden die Daten ohne das angesprochene PAL, alle 256 KBytes im Adress-Bereich des 68000 gespiegelt sein.

Wenn nun das Betriebssystem in EPROMs gebrannt wird, muß eine entsprechende Beschaltung der EPROMs dafür sorgen, daß sich die vier EPROMs wie das zu ersetzende ROM verhalten. Hierzu muß zuerst der 8 Bit breite Datenbus der EPROMs auf 16 Bit erweitert werden. Für diesen Zweck werden immer zwei EPROMs mit jeweils 8 Bit parallel geschaltet, womit man die erforderliche Busbreite von 16 Bit erhält. Um die Speicherzellen in den EPROMs ansprechen zu können, werden jetzt noch die Adressleitungen A1 bis A16 des 40 poligen ROM-Sockels mit den entsprechenden Adressleitungen der EPROMs (A0 bis A15) verbunden.

Zum Unterscheiden der beiden EPROM-Bänke, dient die Adressleitung A17 am ROM-Sockel.

Diese Leitung wird auf ein 7400 gelegt, welches hier als einfacher Adressdekodierer geschaltet ist. Dessen Ausgänge schalten die EPROMs über deren Outputenable-Eingänge ein. Die Chipenable-Leitungen der vier EPROMs sind direkt mit der Chipselect-Leitung am ROM-Sockel verbunden.

In Bild 2 sehen Sie das eben besprochene Schaltbild.

Und Umschalten

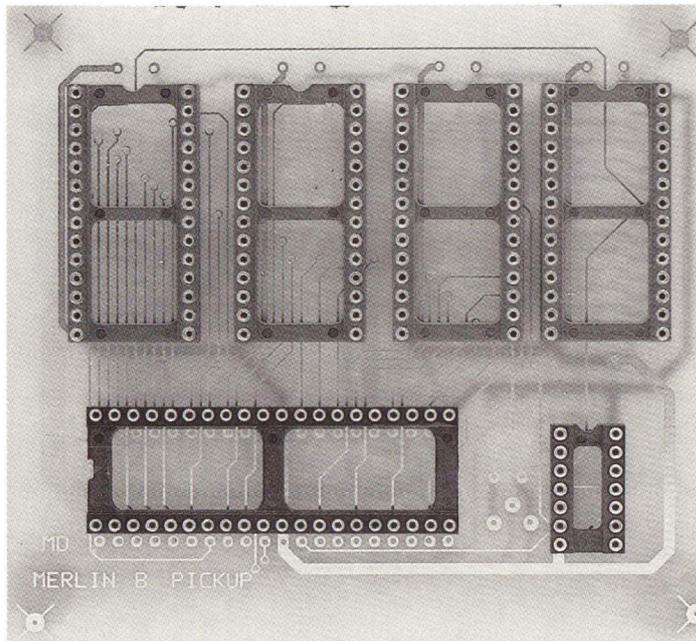


Bild 3: Die teilbestückte Platine

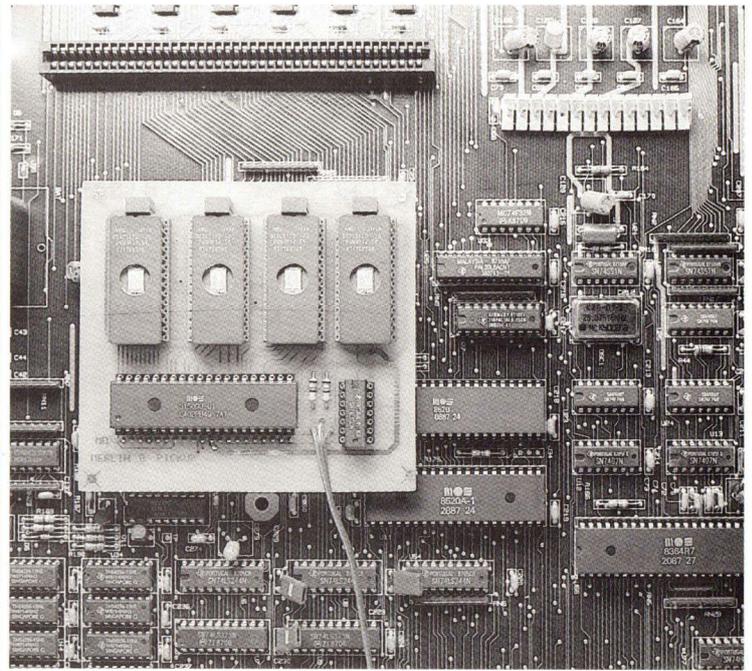
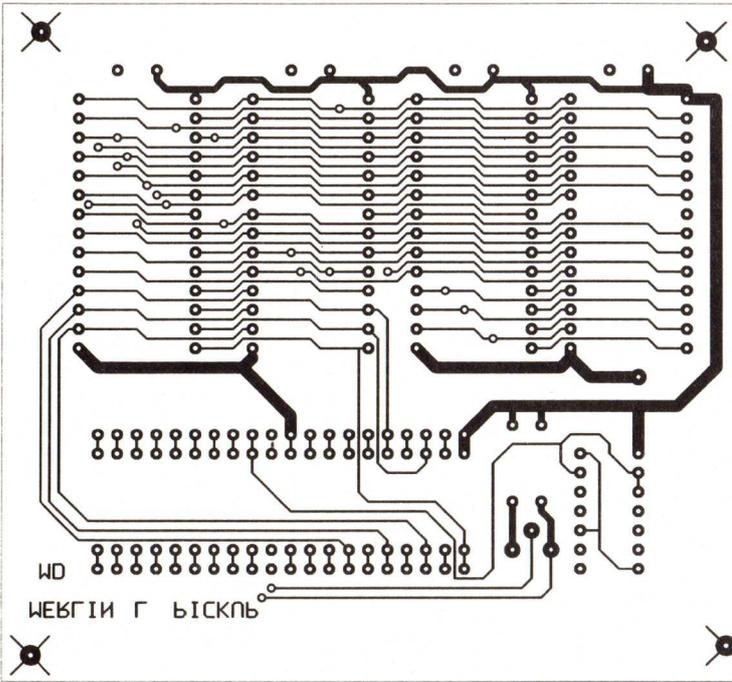


Bild 6: Auch im A2000 findet die Platine ihren Platz

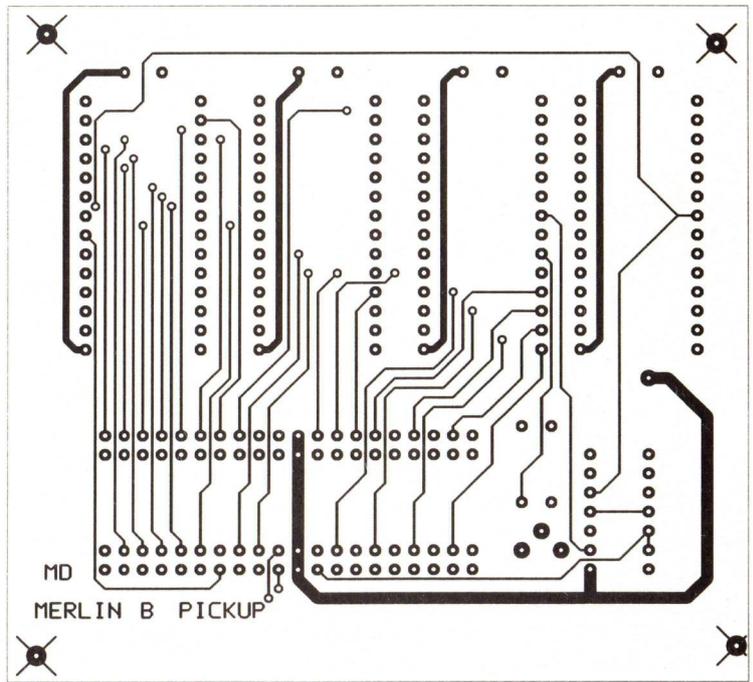
Da die Platine auch das originale ROM enthalten soll, ist eine Möglichkeit vorgesehen, zwischen der EPROM- und der ROM-Variante umzuschalten. Dies kann auf einfache Weise über die Chipselect-Leitung vorgenommen werden. Das vom ROM-Sockel kommende Chipselect-Signal wird auf einen einfachen einpoligen Umschalter geführt, der entsprechend seiner Stellung, das Signal an das ROM bzw. die EPROMs schaltet. Das jeweils andere Speichermedium wird über die Pullup-Widerstände R1 bzw. R2 abgeschaltet. Da beim Umschalten zwischen den zwei Betriebssystem-Varianten kurzzeitig weder das ROM noch die EPROMs am Bus liegen, wird dem Rechner jedoch

nichts anderes übrigbleiben als abzustürzen. Theoretisch würde sich dieses Problem mit einem einfachen Flip-Flop lösen lassen. Jedoch ist eine solche Lösung bei der Betriebssystem-Architektur des Amiga nicht praktikabel - aus folgendem Grund:

Beim Systemstart des Amiga werden sämtliche Einsprungadressen des Betriebssystems in einen bestimmten Rambereich kopiert. Wenn man nun mit dem Flip-Flop auf das andere System umschal-



Platinenlayout Lötseite



Platinenlayout Bestückungsseite

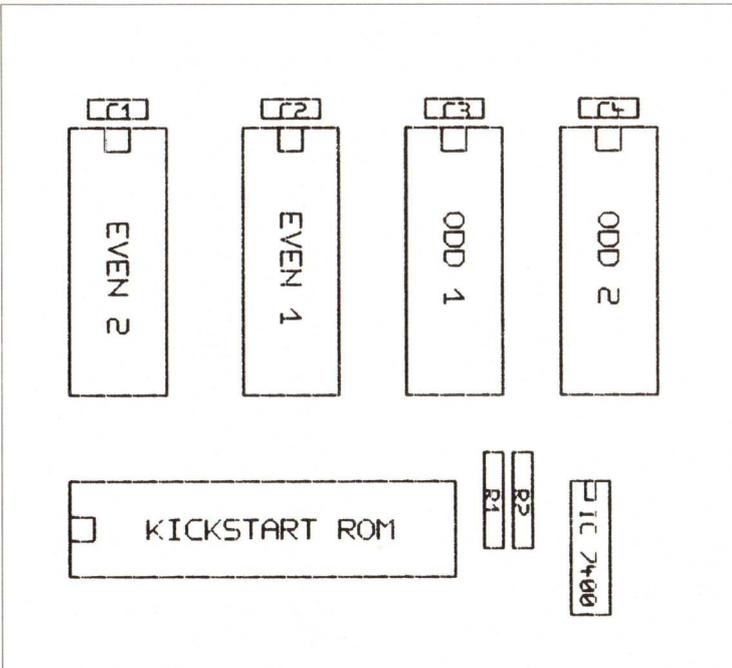


Bild 4: Der Bestückungsplan

ten, würde dies plötzlich auf die Einsprungadressen des nun ausgeschalteten Systems treffen, diesen blind vertrauen und gnadenlos ins Verderben rennen. Ein netter Absturz wäre die Folge.

Der einzige Zeitpunkt, an dem die Umschaltung erfolgen kann, ist die Zeit nach einem Reset (Ctrl-Amiga-Amiga), in der die Leuchtdiode am Rechner erlischt.

Der Aufbau

Wenn Sie die Platine selbst anfertigen wollen, können Sie die hier abgedruckten Layoutvorschläge verwenden. Die Löt- und Bauteile-Seite ist durch einen entsprechenden Buchstaben gekennzeichnet. Bei richtiger Lage der Filme müssen beide Schriftzüge lesbar sein und sich die Lötäugen der Oberseite, mit denen der Unterseite decken. Nun kann die Platine belichtet werden. Nach dem Entwickeln und Ätzen ist sie fertig zum Bohren, was mit einem 0.8mm-Bohrer und etwas Geduld, schnell erledigt ist. Bei den Durchkontaktierungen ist es jedoch besser, einen 0.6mm-Bohrer zu verwenden, da die Lötäugen dort wesentlich kleiner sind. Nach dem Durchkontaktieren können die Fassungen für EPROMs und das ROM eingelötet werden. Vergessen Sie nicht die Stützkondensatoren an den EPROMs und die beiden Wider-

(c) Merlin Computer GmbH

```

erstellt am 12.5.88 Gerald Carda
1: #include "exec/types.h"
2: #include "exec/nodes.h"
3: #include "exec/lists.h"
4: #include "exec/memory.h"
5: #include "exec/interrupts.h"
6: #include "exec/ports.h"
7: #include "exec/libraries.h"
8: #include "exec/io.h"
9: #include "exec/tasks.h"
10: #include "exec/execbase.h"
11: #include "exec/devices.h"
12: #include "devices/trackdisk.h"
13: #include "stdio.h"
14: #include "functions.h"
15: #include "libraries/dos.h"
16:

```

Listing 1: Dieses Programm erzeugt auf jeder KICKSTART die benötigten Module zum Prommen

stände. Mit der Fassung für das 7400 und der Anschlußleiste für den ROM-Sockel ist die Platine betriebsbereit. Bild 3 zeigt Ihnen noch einmal die teilbestückte Platine, Bild 4 zeigt den kompletten Bestückungsplan.

Inbetriebnahme

Die Platine sollte nun noch einmal gründlich auf Kurzschlüsse zwischen einzelnen Leitungen geprüft werden. Bei diesem Arbeitsgang ist auch gleichzeitig auf fehlende Lötverbindungen oder Durchkontaktierungen zu achten. Hat die Platine diese Kontrollen bestanden, kann ein erster Test am Objekt vorgenommen werden. Öffnen Sie hierzu bitte Ihren Rechner (Achtung, die Garantie geht verloren) und entfernen Sie eventuell vorhandene Abschirmbleche. Als nächstes ist das Kickstart-ROM in Ihrem Rechner zu lokalisieren. Es befindet sich in unmittelbarer Nähe des Prozessor und ist das einzige 40polige IC in diesem Bereich. Entfernen Sie es vorsichtig aus seiner Fassung.

Das ROM wird nun in den 40poligen Sockel der Adapterplatine gesetzt, welche wiederum in dem freigewordenen Sockel des Rechners ihren Platz findet. Da die Platine über andere Bauteile auf dem Motherboard ragt, ist noch ein 40poliger Sockel als Abstandshalter zwischen Platine und Rechner zu stecken.

Achten Sie unbedingt auf die richtige Polung der Platine und des ROM. Legen Sie nun den Schalter in die Position, die das am Sockel abgegriffene Chipselect-Signal an das eingesetzte ROM schaltet. Wenn Sie den Rechner nun einschalten, muß er sich in gewohnter Weise verhalten. Bild 5 und 6 zeigen die Position der Platine im Amiga 500 und 2000.

Die Praxis

Sicher ist es wenig sinnvoll die Platine nur mit dem originalen ROM zu betreiben. Es muß also eine Möglichkeit ge-

```

17: #define TD_READ CMD_READ
18: #define BLOCKSIZE TD_SECTOR
19: #define MAXSECTOR 512
20: #define KICKSIZE 0x40200L
21:
22: SHORT error;
23: struct MsgPort *diskport;
24: struct IOExtTD *diskreq;
25: char *diskdata;
26: SHORT testval;
27: char *kickdata;
28: LONG openererror;
29: struct DosBase *DosBase;
30: extern struct MsgPort *CreatePort();
31: extern struct IORequest *CreateExtIO();
32:
33: ULONG diskChangeCount, *ModData;
34:
35:
36: ReadCylSec(cyl, sec, hd)
37:
38: /* diese Funktion liest den durch cyl, sec und hd *
39:  * bestimmten Sektor in den Lesepuffer ein */
40:
41: SHORT cyl, sec, hd;
42: {
43:     LONG offset;
44:
45:     diskreq->iotd_Req.io_Length = BLOCKSIZE;
46:     diskreq->iotd_Req.io_Data = (APTR)diskdata;
47:     diskreq->iotd_Req.io_Command = ETD_READ;
48:     diskreq->iotd_Count = diskChangeCount;
49:     offset=TD_SECTOR*(sec+NUMSECS*hd+
                    NUMSECS*NUMHEADS*cyl);
50:     diskreq->iotd_Req.io_Offset = offset;
51:     DoIO(diskreq);
52:     return(0);
53: }
54:
55:
56: WriteCylSec(cyl, sec, hd)
57:
58: /* diese Funktion schreibt den durch cyl, sec und hd *
59:  * bestimmten Sektor auf Diskette. ACHTUNG der *
60:  * zuletzt geschriebene Sektor wird erst bei einem *
61:  * Spurwechsel auf Diskette übertragen. */
62:
63: SHORT cyl, sec, hd;
64: {
65:     LONG offset;
66:
67:     diskreq->iotd_Req.io_Length = BLOCKSIZE;
68:     diskreq->iotd_Req.io_Data = (APTR)diskdata;
69:     diskreq->iotd_Req.io_Command = ETD_WRITE;
70:     diskreq->iotd_Count = diskChangeCount;
71:     offset=TD_SECTOR*(sec+NUMSECS*hd+
                    NUMSECS*NUMHEADS*cyl);
72:     diskreq->iotd_Req.io_Offset = offset;
73:     DoIO(diskreq);
74:     return(0);
75: }
76:
77: MotorOn()
78:
79: /* diese Funktion schaltet den Motor des im *
80:  * diskreq angegebenen Laufwerks ein */
81:
82: {
83:     diskreq->iotd_Req.io_Length = 1;
84:     diskreq->iotd_Req.io_Command = TD_MOTOR;
85:     DoIO(diskreq);
86:     return(0);
87: }
88:
89: MotorOff()
90:
91: /* dies Funktion schaltet den Motor aus */
92:
93: {
94:     diskreq->iotd_Req.io_Length = 0;
95:     diskreq->iotd_Req.io_Command = TD_MOTOR;
96:     DoIO(diskreq);
97:     return(0);

```

```

98: }
99:
100: OpenStuff()
101:
102: /* alle benötigten Systemressourcen werden *
103: * hier geöffnet bzw. reserviert */
104:
105: {
106:     if(!(DosBase=(struct DosBase *)
OpenLibrary((UBYTE *)"dos.library", 0L)))CloseStuff();
107:     if(!(diskdata=(char *)
AllocMem(BLOCKSIZE, MEMF_CHIP|MEMF_CLEAR)))CloseStuff();
108:     if(!(kickdata=(char *)
AllocMem(KICKSIZE, MEMF_PUBLIC)))CloseStuff();
109:     if(!(diskport=CreatePort(0L, 0L)))CloseStuff();
110:     if(!(diskreq=(struct IOExtTD *)
CreateExtIO(diskport, (LONG)sizeof(struct IOExtTD)))
CloseStuff());
111:     if(openerror=OpenDevice(TD_NAME, 0L, diskreq, 0L))
CloseStuff();
112:     diskreq->iotd_Req.io_Command = TD_CHANGENUM;
113:     DoIO(diskreq);
114:     diskChangeCount = diskreq->iotd_Req.io_Actual;
115:     return(0);
116: }
117:
118: CloseStuff()
119:
120: /* dies Funktion beendet das Programm *
121: * ordnungsgemäß */
122:
123: {
124:     if(!openerror)
125:     {
126:         MotorOff();
127:         CloseDevice(diskreq);
128:     }
129:     if(DosBase)CloseLibrary(DosBase);
130:     if(diskdata)FreeMem(diskdata, BLOCKSIZE);
131:     if(kickdata)FreeMem(kickdata, KICKSIZE);
132:     if(diskport>DeletePort(diskport);
133:     if(diskreq)
DeleteExtIO(diskreq, (LONG)sizeof(struct IOExtTD));
134:     printf("\n Bye Bye\n\n");
135:     exit();
136: }
137:
138: StartUp()
139:
140: /* Ausgabe der Einschaltmeldung */
141:
142: {
143:     printf("\n Legen Sie eine Kickstart Diskette
in das\n");
144:     printf(" interne Laufwerk df0: und betätigen
Sie\n");
145:     printf(" <RETURN>\n");
146:     getchar();
147:     return(0);
148: }
149:
150: Schreibe()
151:
152: /* die Funktion Schreibe entfernt die KICK *
153: * Kennung am Anfang der gelesenen Daten *
154: * und teilt die 256 KByte in zwei 128 *
155: * KByte große Teile, die nach geraden und *
156: * ungeraden Adressen getrennt in seperaten*
157: * Dateien abgelegt werden */
158:
159: {
160:     char *schreibepuffer, *konvertpuffer;
161:     struct FileHandle *datei;
162:     LONG error, i;
163:
164:     konvertpuffer=kickdata;
165:     if(!(schreibepuffer=AllocMem(0x10000L, MEMF_PUBLIC)))
166:     {
167:         puts("puffer fehler");
168:         return();
169:     }
170:     konvertpuffer+=512L; /* sector 0 ausblenden */

```

Stückliste Kickstart-Umschalter

R1 2.7 kΩ
R2 2.7 kΩ
C1 0.22 μF
C2 0.22 μF
C3 0.22 μF
C4 0.22 μF

TTL
74LS00

Eproms
4x 27512-20

Sockel gedreht
1x 14 polig
4x 28 polig
2x 40 polig

Sonstiges
2 Kontaktstreifen
20 polig zum
Anschlu_ im Rechner

1 Schalter 1xUm
1 Platine
doppeltseitig
fotopositiv
Grö_e siehe
Layout

Kosten der Bauteile: ca. DM120.-

funden werden, eine beliebige Kickstart-Diskette in eine EPROM-nutzbare Form zu bringen. Listing 1 zeigt ein in Aztec-C geschriebenes Programm, das eine beliebige Kickstart-Diskette in Laufwerk df0: in den Speicher lädt und daraus vier EPROM-Dateien erzeugt. Die Dateien haben die Namen KICK1.EVEN, KICK1.ODD, KICK2.EVEN und KICK2.ODD und entsprechen den im Bestückungsplan angegebenen EPROM-Positionen. Die Dateien können nun mit einem EPROM-Programmiergerät, z.B. dem JuniorPrommer (siehe KICKSTART 5/88), in die EPROMs gebrannt werden. Das vorgestellte Listing verarbeitet alle Kickstart-Versionen, die für den Amiga zur Verfügung stehen. Sie können so wahlweise geänderte Versionen z.B. mit Virus-Überprüfung oder geänderter Speichererkennung, sowie die Versionen 1.1 oder 1.3 in EPROMs ablegen. Nachdem Sie die EPROMs in die entsprechenden Positionen eingesetzt haben, lassen sie sich durch den Selektierungsschalter aktivieren. Der Rechner meldet sich bei ordnungsgemäßer Verdrahtung der Platine und korrekter Bestückung, mit dem neuen Betriebssystem.

```

171:         for(i=0;i<0x10000;i++)
172:             {
173:                 *(schreibepuffer+i)=*(konvertpuffer+(i<<1));
174:             }
175:         datei=Open((char *)"KICK1.EVEN",MODE_NEWFILE);
176:         error=Write(datei,schreibepuffer,0x10000L);
177:         if(error==-1L)puts("Fehler Schreiben");
178:         Close(datei);
179:         for(i=0;i<0x10000;i++)
180:             {
181:                 *(schreibepuffer+i)=*(konvertpuffer+1+(i<<1));
182:             }
183:         datei=Open((char *)"KICK1.ODD",MODE_NEWFILE);
184:         error=Write(datei,schreibepuffer,0x10000L);
185:         if(error==-1L)puts("Fehler Schreiben");
186:         Close(datei);
187:         konvertpuffer+=0x20000; /* zweite bank */
188:         for(i=0;i<0x10000;i++)
189:             {
190:                 *(schreibepuffer+i)=*(konvertpuffer+(i<<1));
191:             }
192:         datei=Open((char *)"KICK2.EVEN",MODE_NEWFILE);
193:         error=Write(datei,schreibepuffer,0x10000L);
194:         if(error==-1L)puts("Fehler Schreiben");
195:         Close(datei);
196:         for(i=0;i<0x10000;i++)
197:             {
198:                 *(schreibepuffer+i)=*(konvertpuffer+1+(i<<1));
199:             }
200:         datei=Open((char *)"KICK2.ODD",MODE_NEWFILE);
201:         error=Write(datei,schreibepuffer,0x10000L);
202:         if(error==-1L)puts("Fehler Schreiben");
203:         Close(datei);
204:         FreeMem(schreibepuffer,0x10000L);
205:         return();
206:     }
207:
208:     main()
209:
210:     /* die Funktion main überprüft das Vorhanden *
211:     * sein einer Kickstart Diskette und liest *
212:     * diese ein. Bei Erfolg wird die Funktion *
213:     * Schreibe aufgerufen */
214:
215:     {
216:         int track,head,sektor,i;
217:         char *lesepointer;
218:
219:         OpenStuff();
220:         lesepointer=kickdata;
221:         StartUp();
222:         MotorOn();
223:         Delay(60L);
224:
225:         ReadCylSec((SHORT)0,(SHORT)0,(SHORT)0);
226:         if(strcmp(diskdata,(char *)"KICK")!=0)
227:             {
228:                 puts("keine Kickstart Diskette !\n");
229:                 CloseStuff();
230:             }
231:         for(track=0;track<80;track++){
232:             for(head=0;head<2;head++){
233:                 for(sektor=0;sektor<11;sektor++){
234:                     ReadCylSec(track,sektor,head);
235:                     if(diskreq->iotd_Req.io_Error != 0)
236:                         {
237:                             printf("Fehler bei Track %d Sektor %d Seite %d\n",
238:                                     track,sektor,head);
239:                             CloseStuff();
240:                         }
241:                     for(i=0;i<512;i+=1)
242:                         {
243:                             *(lesepointer++)=(diskdata+i);
244:                         }
245:                     if((lesepointer-kickdata) >= KICKSIZE)
246:                         {
247:                             Schreibe();
248:                             CloseStuff();
249:                         }
250:                 }
251:             }
252:         }

```

```

253:         CloseStuff();
254:         return(0);
255:     }
256:
257:
258:
259: /*
260:
261:  Älter Compilerversionen benötigen dies beiden
262:  Funktionen, da sie noch nicht in den Libraries
263:  verfügbar waren.
264:
265:  struct IORequest *CreateExtIO(ioReplyPort, size)
266:      struct MsgPort *ioReplyPort;
267:      LONG size;
268:  {
269:      struct IORequest *ioReq;
270:
271:      if (ioReplyPort == 0)
272:          return ((struct IORequest *) 0);
273:      ioReq = (struct IORequest *)
274:      AllocMem (size, MEMF_CLEAR | MEMF_PUBLIC);
275:      if (ioReq == 0)
276:          return ((struct IORequest *) 0);
277:
278:      ioReq -> io_Message.mn_Node.ln_Type = NT_MESSAGE;
279:      ioReq -> io_Message.mn_Node.ln_Pri = 0;
280:      ioReq -> io_Message.mn_ReplyPort = ioReplyPort;
281:
282:      return (ioReq);
283:  }
284:
285: DeleteExtIO(ioExt, size)
286:     struct IORequest *ioExt;
287:     LONG size;
288:  {
289:     ioExt -> io_Message.mn_Node.ln_Type = 0xff;
290:     ioExt -> io_Device = (struct Device *) -1;
291:     ioExt -> io_Unit = (struct Unit *) -1;
292:
293:     FreeMem (ioExt, size);
294:     return(0);
295:  }
296:
297: */

```

AMIGA-BUREAU

ein Programm nach Ihren Wünschen



EDOTRONIK®

D-8000 München 80, St.-Veit-Straße 70, ☎ 0 89/40 40 93

TRANSFILE

Verbinden Sie Ihren SHARP Pocket-computer mit Commodore AMIGA.

Übertragen von Daten und Programmen des SHARP-Rechners in beide Richtungen, Editieren und Drucken auf allen AMIGA Modellen möglich. Alle Daten und Programme können schnell und sicher auf Diskette abgespeichert werden. TRANSFILE AMIGA ist vollständig mit der Maus zu steuern und ohne Kopierschutz! Es ist für die SHARP-Pockets PC 1260/61/62/80, PC 1350/60, PC 1401/02/03/21/25/30/45/50/60/75 geeignet. Weitere SHARP-Rechnertypen sind in Vorbereitung!

TRANSFILE AMIGA anschlussfertig und komplett mit Interface, Diskette und Anleitung nur DM 129,00

* Bei Bestellung unbedingt Rechnertypen angeben *

Händleranfragen erwünscht!

Als TRANSFILE 64 auch für C-64 bzw. C-128 erhältlich!

Ausführliche Info gegen adressierten Freiumschlag anfordern. Versand ins Ausland nur gegen Vorkasse!

YELLOW-COMPUTING

Wolfram Herzog & Joachim Kieser

Im Weingarten 21 D-7101 Hardthausen 3
Telefon 07139 / 8355

C.V.S.-VERSAND

Zubehör für AMIGA:

Laufwerke extern	
Chinon drive 3,5" abschaltbar	300,00 DM
Chinon drive 5,25" 40/80 TR	350,00 DM
Speichererweiterungen	
A 500 intern 500 KByte	249,00 DM
Drucker/Scanner	
NEC 2200 Pinwriter (24 Nadel)	930,00 DM
Präsident PRINTER	
Amiga Zeichensatz, Centronics Schnittstelle	428,00 DM
Handy-Scanner mit 16 Graustufen incl. Grafikpaket, von Cameron	858,00 DM
Monitore	
s/w Monitor	
anschlussfertig für den Amiga, 20 MHz	230,00 DM
Software	
AMIGA Tools neue Utility-Disk mit Copy,	
Viruskiller, RAM-Deleter etc.	49,95 DM
PD (sehr große Auswahl)	ab 3,80 DM
Katalogdisk (3 Disk)	9,00 DM

C.V.S.

Rauher Berg 1 · 2306 Schönberg
Bestellannahme: ☎ (04 31) 55 15 15

Aktuelle Preisliste a. Anforderung
Preisänderungen vorbehalten!

TOTAL DIGITAL

Zwei neue Soundsampler im Test

Die Zahl der Hardwareerweiterungen für die Amigamodelle steigt ständig. Besonders auf dem Gebiet der digitalen Musikverarbeitung kommen gerade in letzter Zeit sehr viele Neuerungen auf den Markt, so auch die beiden Soundsampler der Firmen Kupke und Alcomp, die wir uns einmal näher angesehen haben.

Beide Geräte werden in Deutschland entwickelt und hergestellt; die Preise liegen bei 79.- DM (Alcomp-Sampler) bzw. bei 139.- (189.-) DM (Golem-Sound). Der Sampler der Firma Alcomp ist für alle Amigamodelle lieferbar (siehe Bild 1). Das Testgerät, welches uns zur Verfügung stand, war ein Modell für den Amiga 500. Zum Lieferumfang gehören außer der Hardware noch eine Diskette mit einem Samplingprogramm und eine sehr dürftige, nicht ganz einseitige Bedienungsanleitung. Auch das mitgelieferte Programm kann man eher zum Ausprobieren als zum wirklichen Arbeiten verwenden (zum Glück gibt es ja einige sehr ausgereifte Samplingprogramme auf dem Markt). Der Sampler selbst hat die Größe einer Zigarettenschachtel und wird direkt auf den Parallelport gesteckt. Mittels eines Kabels mit einem 9poligen Stecker bezieht der Sampler

seine nötige Arbeitsspannung von 5V aus einem der beiden Joystickports des Rechners (dies ist allerdings nur bei den Modellen für Amiga 500 und 2000 der Fall, weil beim Amiga 1000 die Arbeitsspannung direkt über den Parallelport bezogen werden kann). Auf der Rückseite befindet sich eine Cinchbuchse, mit der man die entsprechende Signalquelle wie Mikrofon, CD, Plattenspieler etc. leider nur in Mono mit dem Sampler verbinden kann. Es wurde vom Hersteller auch darauf verzichtet, noch eine DIN-Buchse als Alternative zum Cinch-Anschluß anzubringen, was Besitzern von deutschem HIFI-Equipment den Ärger mit aufwendigen Adaptern erspart hätte.

Im Test erwies sich das Aussteuern der Samples als eine knifflige Aufgabe, denn der Aussteuerungsregler (in diesem Fall ein Trimpoti) befindet sich zwar auf Geräteoberseite, ist aber ca. 1 cm versenkt angebracht. Das führt

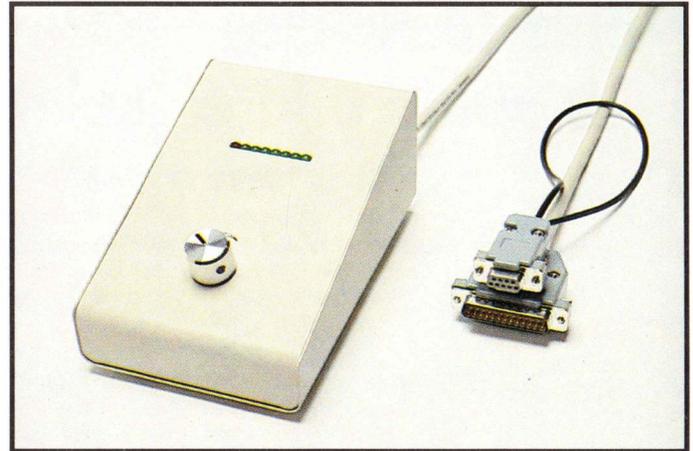
dazu, daß man bei jeder Aussteuerungskorrektur einen sehr feinen Schraubenzieher zu zücken hat, probieren muß, das Poti zu treffen (was sich als gar nicht so einfach erweist, denn der Sampler steckt ja auf der Rückseite des Rechners) und gleichzeitig versuchen muß, die richtige Aussteuerung einzustellen. Dieser Umstand macht es fast unmöglich, Musikpassagen mit dem Alcomp-Sampler aufzunehmen, weil gerade hier der Pegel öfter nachjustiert werden muß. Hier muß ich leider sagen, daß der Hersteller nicht mitgedacht hat und so dem Anwender unnötig den Umgang mit dem Gerät erschwert.

Ansonsten versah der Sampler brav seinen Dienst und hatte auch mit Samplingraten bis zu 20 kHz keine Schwierigkeiten. Empfehlen würde ich den Alcomp-Sampler wegen seines niedrigen Preises (79.- DM) den Einsteigern oder Neugierigen, die 'mal kurz in die Materie schnuppern wollen; für ernsthafte Anwendung ist er meiner Meinung nach aber nicht geeignet.

Unser zweites Testgerät wird von der Firma Kupke unter dem Namen Golem-Sound angeboten. Es ist in zwei Versionen erhältlich, nämlich als Mono- oder Stereogerät. Der Preis für den Monosampler beträgt 139.- DM, der Stereosampler ist für 189.- DM zu haben. Golem-Sound ist auch für alle Amigamodelle lieferbar. Der Lieferumfang besteht aus der eigentlichen Hardware und einer kurzen Anleitung.



Alcomp-Sampler



Golem-Sound

Die Firma Kupke bietet zusätzlich eine stereofähige Steuersoftware für knapp 30.- DM an.

Das Gerät besteht aus einem keilförmigen, ca. 20 cm langen und sehr sauber verarbeiteten Metallkasten (siehe Bild 2). Der Sampler wird mittels zweier Kabel am Parallelport und am Joystickport (zwecks Stromversorgung) angeschlossen. Dies gilt wie beim Alcomp-Sampler nur für die A500/A2000-Modelle, weil beim guten alten Amiga 1000 die nötige Spannung direkt am Parallelport anliegt. Positiv ist hier zu erwähnen, daß die Anschlußkabel ausreichend lang sind und es so ermöglichen, den Sampler fast frei auf dem Arbeitstisch zu positionieren. Auf der Rückseite des Gehäuses befindet sich ein Ein- und Ausschalter, zwei Cinchbuchsen (wir hatten die Stereo-version im Test) und eine 5-polige DIN-Buchse. Durch diese Kontaktfreudigkeit wird ermöglicht, daß fast jede Signalquelle ohne problematische Adapterverbindungen direkt angeschlossen werden kann. Die Oberseite von Golemsound wird von einer 8stelligigen LED-Anzeige und einem wunderbar großen, leichtgängigen Aussteuerungsregler verziert. Die LED-Anzeige erfüllt hier gleich zwei Aufgaben: erstens als Einschaltkontrolle und zweitens als Pegelanzeige, vergleichbar mit LED-Displays bei Tapedecks. Im Test erwies sich der Umgang mit der Golemsoundbox als ausgezeichnet. Die hohe Taktfrequenz des Samplers (1 MHz!) machte das Samplen zu einem wahren Vergnügen, und die Aussteuerung mittels der LED-Anzeige ermöglichte eine wirklich exakte Einpegelung von allen Signalquellen. Auch Stereosampling (wir verwendete-

ten einige stereofähige Programme wie z.B. Aegis Audiomaster und Perfectsound) wurde problemlos bis weit über den 20Khz-Samplingbereich zu vollsten Zufriedenheit vom Sampler erledigt.

Golemsound zählt nach diesem Test für mich zu den besten Samplern für den Amiga überhaupt. Ob es seine wirklich hervorragende Verarbeitung oder seine bemerkenswerten Soundergebnisse sind: In allen Details zeigt sich, daß vom Hersteller mitgedacht und gezeigt wurde, daß Spitzenhardware nicht teuer sein muß. Wer professionell (zumindestens für Amiga-Verhältnisse) samplen will, der sollte den Golem-Sound-Sampler, wenn möglich, einmal ausprobieren, es lohnt sich auf jeden Fall.

Alcomp-Sampler

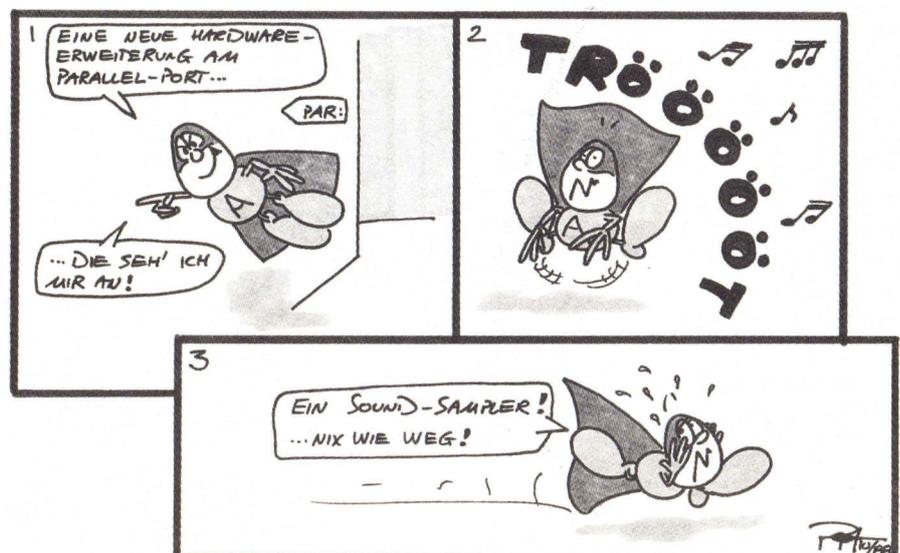
- + niedriger Preis (79.- DM)
- + mitgelieferte Software

- Sampler nicht frei beweglich
- Aussteuerung problematisch
- Anleitung sehr dürftig
- Verarbeitung könnte besser sein

Golem-Sound

- + Hervorragende Verarbeitung
- + optische Aussteuerung durch LED's
- + frei beweglich durch ausreichend lange Kabel
- + Stereo

- Steuersoftware nicht im Lieferumfang inbegriffen



© 1988 INSEL/CALIBRE GMBH

Die Retter kommen – Eine todsichere Sache ... Ab heute gibt es keine VIREN mehr!



1. Virus Finder

nur 49,90 DM

Mit dem Virus Finder können Sie schnell und problemlos alle Ihre Disketten nach einem Virus durchsuchen und „heilen“. Z. B. SCA und Byte Bandit Virus gibt es danach **nicht** mehr! Er erkennt und löscht auch Viren, im Speicher! Bedienungsanleitung in deutsch auf Diskette (Virus Finder Doc). Virus Finder läßt sich auch auf der Workbench installieren!

2. Second Chance

nur 49,90 DM

Speichert, auch von kopiergeschützten Programmen, den Boot-Block auf eine separate Diskette ab (auf einer Diskette können über 800 Boot-Blöcke gespeichert werden). Wenn der Boot-Block auf dem Original durch Viren zerstört wurde, kann der zuvor gesicherte Boot-Block auf das Original zurückgeschrieben werden. Die Software ist wieder einsatzfähig.

3. Int Switch

nur 29,90 DM

Dieses Programm löst alle Ihre Probleme mit Software, die mit Speichererweiterungen nicht laufen. Int Switch schaltet beim Amiga 2000 XT/AT-Karte und Speichererweiterungen ab. Int Switch schaltet beim Amiga 1000 Sidecar und Speichererweiterungen ab. Int Switch schaltet beim Amiga 500 Expansionskarten und Speichererweiterungen ab.

Nachnahme 6,— DM

Vorkasse 4,— DM

Ausland: nur gegen Vorkasse 10,— DM

PDC GmbH, Louisenstraße 115, 6380 Bad Homburg
Telefon (0 61 72) 2 47 48 oder 2 07 99

LAUFWERKE FÜR AMIGA

3 1/2" Extern, Metallgehäuse, helle Front, durchgef. Port m. Schraubverr., abschaltbar 299,—
3 1/2" Intern, komplett m. Einbausatz u. Anleitung 229,—
5 1/4" Extern, Metallgehäuse, helle Front, 40/80 Spur, durchgef. Port mit Verr., abschaltbar. 349,—

Rainbow Data

SPEICHERERWEITERUNG FÜR AMIGA 500

512 KB Ram, akkugep. Uhr und Abschaltung a. A. 948,—
2 MB Extern, (Profex), durchg. Port, abschaltbar

COMPUTERLEITUNGEN

Druckerkabel für alle Amiga 23,—
Monitorleitung Amiga/Scart 25,—
Emulatorkabel 19,90
Bootselctor DF 0/DF 1/2 od. 3 19,—
Weitere Angebote auf Anfrage.
Preisänderungen vorbehalten.

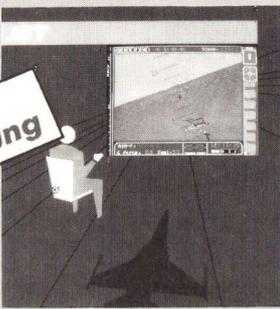
NEU IN UNSEREM ANGEBOT

Mouse-Pad, antistat., rutschfest 16,90
DISKETTEN
3 1/2" NO NAME 2DD 23,90
5 1/4" NO NAME 2DD 96TPI 8,90
5 1/4" NO NAME 2DD 96TPI 13,50
5 1/4" PRECISION 14,50
3 1/2" Markendisketten Preis a. A.
3 1/2 3MDS 39,90

Versand per Nachnahmen: Rainbow Data, Am Kalkofen 32, 5603 Wülfrath, Telefon (0 20 58) 13 66 — ACHTUNG: NEUE HAUSNUMMER

Jet ist da!

mit deutscher Anleitung



- Fliegen Sie einen F16-Falcon oder F18-Harrier!
- Superschnelle Grafik!
- Vollinstrumentiertes Jet-Cockpit.
- Luft-Luft/Luft-Boden-Kampf wählbar. Spieleroption (z.B. via Modem)!
- Scenery Disk kompatibel zum Flugsimulator II
- Zögern Sie nicht, reservieren Sie sich Ihren persönlichen Jet-Simulator!



Borsigallee 18
6000 Frankfurt/M.
☎ 069-410071/72

Schweiz:
MICROTRON CH-2542 Pieterlen
Bahnhofstraße 2 Tel. 032 87 24 29

AMIGA aktuell – Dieter Hieske · Tel. 06 21 / 67 31 05

Ladenlokal Schillerstraße 36 · 6700 Ludwigshafen-Oggersheim

Öffnungszeiten: Mo – Fr. 9.30 – 12 Uhr, 14 – 18 Uhr. Mi. 9.30 – 12 Uhr. 14 – 16 Uhr
Sa 9 – 13 Uhr. Danach Anrufbeantworter Bestellservice

AMIGA SPIELE SOFTWARE

Bubble Double	52,50	Battle Ships	52,50
Clever & Smart	52,50	Destroyer	65,90
Giana Sisters	51,95	Jet	96,50
Obliterator	68,95	Pink Panther	52,50
Ports of Call	84,50	Strike Force Ha.	65,90
Ferrari Form. 1	76,95	Garrison II+I je	49,95
Golden Path	49,95	Indoor Sports	59,95
Jinks	49,95	Jump Jet	46,95
King o. Chicago	65,90	In 80Tg u.d. Welt	52,90
Kickstart II	27,95	Rallye Master	27,95
Roadwars	59,95	Sky Blaster	49,95
Analen der Römer	72,90	Armagedon Man	64,90
Winter Olympiad	54,00	Winter Challenge	49,95
Qiuwi	44,95	Terrorpods	59,95
Wizzball	68,90	Jinxter	59,95
Sargotafaser	Preis?	Leatherneck	62,90

Täglich Neuheiten. Preislisten Kostenlos.
Versand per NN + DM 8,00 Portoselbstkosten
Vorauskasse ohne Portokosten

PUBLIC DOMAIN SERVICE:

Über 700 Disketten lieferbar auch auf 5,25".

PUBLIC DOMAIN PREISE

Kopierpreise,
wenn wir die Disketten stellen:
1 – 9 4,00 / 10 – 19 6,50
20 – 29 6,00 / 30 – 49 5,50
50 – 99 2,00 / 100 – 199 4,50
ab 200 4,00

Kopierpreise, wenn Sie uns Ihre Disketten per Einschreiben schicken:
1 – 9 3,00 / 10 – 19 2,70
20 – 29 2,40 / 30 – 49 2,10
50 – 99 1,80 / 100 – 199 1,50
ab 200 1,20

Kopierpreise auf 5,25" Disk von uns
1 – 9 3,00 / 10 – 19 2,70
20 – 29 2,40 / 30 – 49 2,10
50 – 99 1,80 / 100 – 199 1,50
ab 200 1,20

ALLE KOPIEN MIT VERIFY.
PREISE PER KOPIE

DAS ERBE

Serielle Commodore-Drucker am AMIGA

Viele Amiga-Besitzer sind "ihrer Marke" treu geblieben und haben vielleicht noch einen C64 oder PC128. Schön wäre es, wenn man das einst teuer bezahlte Zubehör, beispielsweise den Drucker, am neuen Rechner weiterverwenden könnte. Leider lassen sich die Drucker mit seriellen Commodore-Anschluß nicht direkt am AMIGA betreiben, den dieser hat (endlich) den üblichen Centronics-Standard. Die Lösung des Problems bietet das Programm CPrint zusammen mit dem zugehörigen Kabel. Die als ASCII-Textdatei abgespeicherten Texte werden von CPrint gelesen und dann im seriellen Commodore-Datenformat zum Drucker gesendet. Das Kabel ist mit etwas Löterfahrung schnell zusammengelötet und kostet nicht mehr als 20.- DM.

Die Bedienung des Programms ist recht einfach: Nachdem bei abgeschaltetem Drucker und Rechner die Verbindung hergestellt ist, wird das Programm CPrint gestartet. Die zu druckende Datei läßt sich auf zwei Arten angeben:

- Datei selektieren (Anklicken des Icons) und das Programm CPrint durch Doppelklick mit gedrückter SHIFT-Taste starten. (So lassen sich auch mehrere Dateien automatisch

hintereinander ausdrucken, indem mehrere Texte selektiert werden und das Programm wie beschrieben gestartet wird.)

-Nach Starten des Programms Dateiname, evtl. auch die Laufwerks- und Directory-Bezeichnung mit abschließenden <RETURN> eingeben.

In beiden Fällen wird dann sofort gedruckt, es sei denn, der Drucker ist nicht eingeschaltet oder seine Standardsekundäradresse ist nicht auf 4 eingestellt. Ein <RETURN> ohne Da-

teiname und das übliche Schließen des Windows beenden das Programm. Natürlich kann man es auch im Hintergrund auf den nächsten Druckvorgang warten lassen. Leider ist das von Commodore einst eingeführte IEC-Datenformat so zeitkritisch, daß während des Druckvorgangs das Multitasking des AMIGA abgeschaltet werden muß. In "Notfällen" kann das Programm aber durch Drücken der <Ctrl>-Taste gestoppt werden.

Zum Programm-Listing

Das Modula-2-Programm zeigt, daß es auch in einer Hochsprache möglich ist, schnelle und der Hardware sehr nahe Programme zu schreiben. Insbesondere die komfortable Deklaration von Mengen (SET) erlaubt es hier, die fünf benutzten Portleitungen der CIA einfach und selbstredend zu setzen oder abzufragen.

Das Programm besteht im wesentlichen aus der Sende-Routine (PROCEDURE sende), die die entsprechende ROM-Routine des C64-Betriebssystem nachahmt. Die Vorgänge der seriellen IEC-Übertragung sind recht komplex und in der Fachliteratur oft nur ungenau wiedergegeben (siehe Sonderteil).

Da Commodore sich bei den "alten Rechnern" nicht an den ASCII-Standard gehalten hat, muß in einer Konvertierungsroutine (PROCEDURE Konvert) der AMIGA-Zeichencode in den alten Commodore-Code umgewandelt werden. Hier lassen sich noch die spezifischen Druckereigenarten (Sonderzeichen und SteuerCodes) be-

rücksichtigen. Auch wäre es leicht möglich, die Umlaute im Grafikmodus des MPS 801/MPS 802 über eine eigene Grafikroutine auszudrucken. Der Phantasie sind hier keine Grenzen gesetzt. In der vorliegenden Version werden die Sonderzeichen von Modula-2 für den MPS 801 und die Umlaute für den Typenrad-Drucker DPS 1120 umgesetzt.

Interessant ist vielleicht noch, wie einfach es in Modula ist, die vom Betriebssystem kommende Meldung über selektierte Dateien auszuwerten (PROCEDURE dateiselektieren/ GetArg). Das Programm wurde mit dem Modula 2-Entwicklungssystem von A.+L. Meier-Vogt erstellt. Sicherlich läßt es sich auch auf andere Modula-2 Compiler übertragen und sogar in C umschreiben.

Zum Kabel:

Das Kabel stellt die Verbindung zwischen dem Parallel-Port des AMIGACIA und dem seriellen Eingang des Druckers her. Die integrierte Schaltung puffert die Ausgabe- und Eingabe-Ports, ohne die digitalen Pegel zu verändern, und dient zum Schutz des teuren Computerbausteins.

Die Widerstände R1 - R5 stellen sicher, daß bei einem Verdrahtungsfehler kein Schaden angerichtet wird. Sie sollten deshalb zunächst direkt an die Kontakte des Steckers gelötet werden. Mit etwas Geschick kann man die Schaltung im Gehäuse des D-Sub-Steckers unterbringen. Das Kabel arbeitet übrigens auch mit dem bekannten C64-Emulator.

Bauteilliste:

- 1 Diodenstecker 6-pol
- 1 D-Sub-Stecker, Stifte, 25-pol
- 1 Kabel, abgeschirmt, 4-pol, ca. 2 m
- 1 IC SN74LS007
- 1 Widerstand 1 kOhm, 1/8 Watt
- 5 Widerstände 470 Ohm, 1/8 Watt

Schaltbild: siehe Artikel Ende.

```

1:  MODULE CPrint;
2:
3:  (* ----- *)
4:  (* AMIGA-Drucktreiber fuer Commodore-Drucker mit seriellem *)
5:  (* Eingang *)
6:  (* (z.B. MSP801/2/3 u. komp.) und Anschluss ueber *)
7:  (* CPrint-Kabel *)
8:  (* ----- *)
9:  (* Aufruf der ASCII-Files durch Eingabe oder erweiterte *)
10: (* Auswahl *)
11: (* ----- *)
12: (* Erstellt in Modula-2 mit M2 AMIGA von A.+ L. Meier-Vogt *)
13: (*      Christoph Münzer *)
14: (*      Ferd.-Wallbrecht-Str.20 ,   3000 Hannover 1 *)
15: (* *)
16: (*      (c) MERLIN-Computer           KICKSTART 6/88 *)
17: (* *)
18: (* ----- *)
19:
20: FROM InOut      IMPORT WriteLn,WriteString,ReadString;
21: FROM SYSTEM    IMPORT SHIFT, ADR, BYTE;
22: FROM FileSystem IMPORT Lookup,File,Response,ReadChar,Close;
23: FROM Strings   IMPORT Length;
24: FROM Arguments IMPORT NumArgs,GetArg;
25: FROM Arts      IMPORT Error;
26: FROM Exec      IMPORT Enable,Disable;
27:
28: CONST version  ="CPrint [1.1], 16.4.88 ";
29:
30: TYPE Bits=SET OF (b0,b1,b2,ATNout,CLKout,DATAout,CLKin,DATAin);
31: (*Parallel-Port Bit: 0 1 2 3   4   5   6   7 *)
32: (*      Pin 2 3 4 5   6   7   8   9 *)
33: (* Ein gesetztes Element/Bit (z.B. ATNout) bedeutet, daß der *)
34: (* Port elektrisch auf HIGH-Pegel liegt und der logische *)
35: (* Wert der IEC-Leitung FALSE ist. *)
36:
37: CONST EOI      = TRUE;
38:              (* IEC Statusmeldung: End Of Information *)
39: listen        = 020H;
40:              (* IEC Befehlscode: Gerät soll empfangen *)
41: unlisten      = 030H+00FH;
42:              (* IEC Befehlscode: Ende des Empfang *)
43: io            = 060H;
44:              (* IEC Befehlscode: weitere Infos *)
45: CR            = 00AH;
46:
47: VAR  Port [00BFE101H] : Bits;
48:      ok              : BOOLEAN;
49:      Datei           : File;
50:      Argzahl         : INTEGER;
51:
52: PROCEDURE interrupt(zulassen:BOOLEAN);
53: BEGIN
54:   IF zulassen THEN Enable();
55:   ELSE Disable();
56: END;
57: END interrupt;
58:
59: PROCEDURE initPort(flag:BOOLEAN);
60: VAR  DRPort[00BFE301H] : Bits;
61: BEGIN
62:   IF flag
63:   THEN DRPort := Bits{ATNout,CLKout,DATAout};
64:        (*Ausgang: D3..D5 (Pin 05..07)*)
65:   Port := Bits{ATNout,CLKout,DATAout};
66:        (*ATN,CLK,DATA-Leitung:=HIGH *)
67:   ELSE DRPort := Bits{};
68:        (*Standardeinstellung *)
69: END;
70: END initPort;
71:
72: PROCEDURE fehler(text: ARRAY OF CHAR);
73: BEGIN
74:   interrupt(TRUE);

```

```

75:   initPort(FALSE);
76:   Error(ADR("Uebertragungsfehler"),ADR(text));
77: END fehler;
78:
79: PROCEDURE Atn(flag:BOOLEAN);
80: BEGIN
81:   IF flag THEN Port:=Port-Bits{ATNout};
82:   ELSE Port:=Port+Bits{ATNout};
83: END;
84: END Atn;
85:
86: PROCEDURE devicetest;
87: VAR t : INTEGER;
88: BEGIN
89:   interrupt(FALSE);
90:
91:   Port := Port - Bits{CLKout};
92:
93:   t:=0;
94: REPEAT
95:   INC(t)
96: UNTIL (NOT (DATAin IN Port)) OR (t= 100) ;
97: IF t=100 THEN fehler("Device not present"); END;
98: END devicetest;
99:
100: PROCEDURE sende(byte:INTEGER; eoi:BOOLEAN);
101: CONST Ctrl = CHAR(039H);
102:           (*Ctrl-Tastencode*)
103: VAR t,bit : INTEGER;
104:     Taste[00BFEC01H] : CHAR;
105:           (*CIA-Adresse des Tastaturschiebe-*)
106:           (*Registers *)
107: PROCEDURE sendeEOI;
108: BEGIN
109:   FOR t:=0 TO 10 DO END; (*mit Einsprung 250 ys Wartezeit*)
110:   t:=0;
111: REPEAT
112:   INC(t)
113: UNTIL (NOT (DATAin IN Port)) OR (t=100) ;
114: IF t=100 THEN fehler("EOI-Fehler 1"); END;
115:
116: FOR t:=0 TO 3 DO END; (*40 ys Wartezei*)
117: t:=0;
118: REPEAT
119:   INC(t)
120: UNTIL (DATAin IN Port) OR (t=100);
121: IF t=100 THEN fehler("EOI-Fehler 2"); END
122: END sendeEOI;
123:
124: BEGIN (* sende *)
125:   interrupt(FALSE);
126:
127:   Port := Port + Bits{DATAout,CLKout};
128: REPEAT
129:   IF Taste=Ctrl THEN fehler("Benutzer-Abbruch") END;
130: UNTIL DATAin IN Port ;
131:
132: IF eoi THEN sendeEOI; END;
133: FOR t:=0 TO 5 DO END; (*60 ys Wartezeit*)
134: Port:=Port - Bits{CLKout};
135: FOR bit := 0 TO 7 DO
136:   IF NOT ODD(byte)
137:     THEN Port := Port - Bits{DATAout};
138:     ELSE Port := Port + Bits{DATAout};
139:   END;
140:
141: FOR t:=0 TO 5 DO END; (*70 ys Wartezeit*)
142: Port:=Port + Bits{CLKout};
143:
144: FOR t:=0 TO 2 DO END; (*30 ys Wartezeit*)
145: Port:=Port - Bits{CLKout};
146: FOR t:=0 TO 1 DO END; (*20 ys Wartezeit*)
147: Port:=Port + Bits{DATAout};
148:

```

Der serielle Commodore IEC-Bus

Commodore hat einst bei der Entwicklung des legendären Rechners VC20 ein serielles Datenbussystem entwickelt, das bis heute beim C64 und seinen Peripheriegeräten im Einsatz ist.

An einem Port mit nur drei für die Übertragung relevanten Kontakten lassen sich gleichzeitig mehrere Laufwerke und Drucker ansprechen und Daten in beide Richtungen übertragen. Damit dies möglich ist, hat Commodore zwei Ideen verwirklicht: Die serielle Datenübertragung und die Trennung von Aktions- und Datenmodus. Doch bevor wir in den Ablauf der Datenübertragung einsteigen, etwas über die Logik-Pegel auf der Bus-Leitung:

Der Datenbus

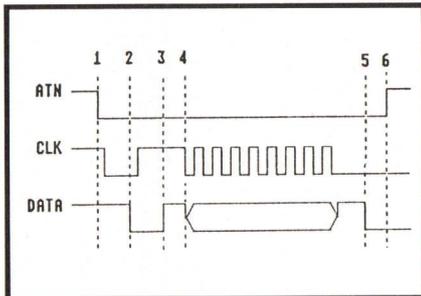
Der serielle Port hat insgesamt 6 Pins mit folgender Bedeutung:

PIN	SIGNAL	(Bedeutung)
1	SQR	SERVICE REQUEST (nicht verwendet)
2	GND	GROUND (Signalmasse)
3	ATN	ATTENTION (Daten oder Kommados folgen)
4	CLK	CLOCK (Takt der Daten- übertragung)
5	DATA	DATA (Serielle Daten (In/Out))
6	RES	RESET (Gerät wird zurück- gesetzt)

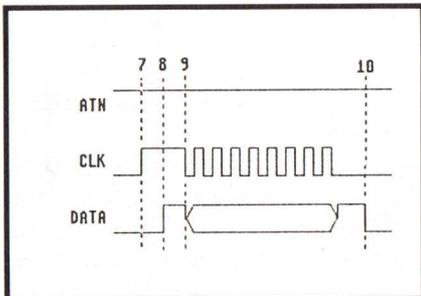
Die Leitungen werden in negativer Logik betrieben: Logisch FALSCH (FALSE) entspricht ca. +5 V (HIGH), logisch WAHR (TRUE) entspricht ca. 0 V (LOW).

Warum dies? Sonst ist es in der Digitaltechnik genau anders herum. Der Grund liegt darin, daß an eine Leitung mehrere Geräte angeschlossen sind. Die Portausgänge sind deshalb als "Open Collector-Treiber" ausgelegt und werden im passiven Zustand eines Ausgangs durch "Pullup-Widerstände" auf dem HIGH-Pegel (+5 V) ge-

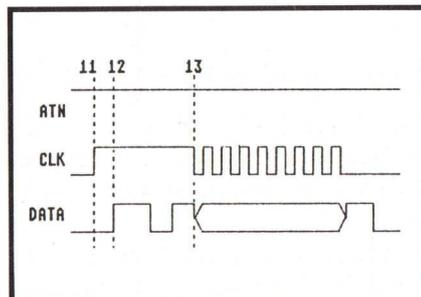
halten. Die Leitung hat dann den logischen Zustand FALSCH. Jetzt kann jedes der Geräte am Bus die Leitung auf den LOW-Pegel (0 V) bringen und den Zustand TRUE übertragen. Natürlich muß das Gerät die Leitung erst wieder freigeben, bevor ein anderes Gerät darauf zugreifen kann. Das bedeutet: LOW wird eine Leitung, wenn mindestens ein Gerät LOW sendet, HIGH wird eine Leitung erst dann, wenn alle Geräte HIGH senden.



Die Adressierung



Datentransfer



letztes Datenbyte

Die Datenübertragung

Das gesamte Geschehen auf dem Bus wird vom Controller, also dem Computer, überwacht und gesteuert. Dazu bedient er jeweils die drei Leitungen DATA, CLK und ATN. Seine Port-Ausgänge DATAout, CLKout und ATNout können die Leitungen jeweils auf LOW setzen. Gleichzeitig überwacht der Controller den Zustand der Leitungen DATA und CLK durch die Eingänge DATAin und CLKin.

```

149:     byte:=SHIFT(byte,-1); (*Rechts-Schieben des Bitmusters*)
150:     END; (* bit *)
151:
152:     t:=0;
153:     REPEAT
154:         INC(t);
155:     UNTIL (NOT (DATAin IN Port)) OR (t =31000) ;
156:     IF t=31000 THEN fehler("ACK-Fehler "); END;
157:
158:     interrupt(TRUE);
159: END sende;
160:
161: PROCEDURE DruckerAnsprechen;
162: BEGIN
163:     initPort(TRUE);
164:     Atn(TRUE);
165:     devicetest;
166:     sende(listen+4,NOT EOI);           (* Geraeteadresse: 4 *)
167:     sende(io+7,NOT EOI);             (* Sekundaeradresse: 7 *)
168:     Atn(FALSE);
169: END DruckerAnsprechen;
170:
171: PROCEDURE DruckerAbhaengen;
172: BEGIN;
173:     sende(CR,EOI);
174:     Atn(TRUE);
175:     sende(unlisten,NOT EOI);
176:     initPort(FALSE);
177: END DruckerAbhaengen;
178:
179: PROCEDURE Dateiwaehlen(VAR ok: BOOLEAN);
180: CONST buffer      = 10000; (*DiskBuffer*)
181: VAR Dateiname     : ARRAY[0..40] OF CHAR;
182: BEGIN
183:     LOOP
184:         WriteString("in> ");
185:         ReadString(Dateiname);
186:         ok:= (Length(Dateiname) > 0);
187:         IF NOT ok THEN EXIT END;
188:         Lookup(Datei,Dateiname,buffer,FALSE);
189:         IF Datei.res = done THEN EXIT END;
190:         WriteString(" -> Datei nicht vorhanden ! "); WriteLn;
191:     END;
192: END Dateiwaehlen;
193:
194: PROCEDURE Dateiselektieren(VAR Argzahl: INTEGER;
195:                             VAR ok: BOOLEAN);
196: CONST buffer      = 10000; (*DiskBuffer*)
197: VAR Dateiname     : ARRAY[0..40] OF CHAR;
198:     Arglaenge     : INTEGER;
199: BEGIN
200:     GetArg(Argzahl,Dateiname,Arglaenge);
201:     (* Uebernahme der Dateinamen bei *)
202:     WriteString("in> ");
203:     (* erweiterter Auswahl *)
204:     WriteString(Dateiname);
205:     WriteLn;
206:     Lookup(Datei,Dateiname,buffer,FALSE);
207:     ok:= (Datei.res = done);
208:     IF NOT ok THEN WriteString(" -> Datei nicht lesbar ! ");
209:     WriteLn;
210:     END;
211: END Dateiselektieren;
212:
213: PROCEDURE konvert(zeichen:CHAR): INTEGER;
214: (* Konvertiert den AMIGA-ASCII_Code C64-Drucker_Code *)
215: (* Hier druckerspezifische Aenderungen vornehmen *)
216: VAR byte : INTEGER;
217: BEGIN
218:     byte:=INTEGER(zeichen);
219:     IF byte = 00AH THEN RETURN 00DH (*LF->CR*);
220:     ELSIF byte < 041H THEN RETURN byte;
221:     ELSIF byte < 05BH THEN RETURN byte+080H;
222:     ELSIF byte < 061H THEN RETURN byte;

```

```

222:   ELSIF byte < 07BH THEN RETURN byte-020H
223:   ELSE
224:     CASE byte OF
225:     | INTEGER("ä") : RETURN 0BBH
226:                   (* Umlaute des DPS 1120 *)
227:     | INTEGER("ö") : RETURN 0BCH
228:     | INTEGER("ü") : RETURN 0BDH
229:     | INTEGER("ß") : RETURN 0BEH
230:     | INTEGER("Ä") : RETURN 0DDH
231:     | INTEGER("Ö") : RETURN 0DCH
232:     | INTEGER("Ü") : RETURN 0DDH
233:     | INTEGER("|") : RETURN 0DDH
234:                   (* MPS 801 *)
235:     | INTEGER("{") : RETURN 0F3H
236:                   (* geschweifte Klammer auf *)
237:     | INTEGER("}") : RETURN 0EBH
238:                   (* geschweifte Klammer zu *)
239:     ELSE
240:       RETURN byte
241:     END
242:   END konvert;
243:
244:   PROCEDURE Dateidrucken;
245:   VAR zeichen : CHAR;
246:   BEGIN
247:     ReadChar(Datei, zeichen);
248:     WHILE NOT Datei.eof DO
249:       sende(konvert(zeichen), NOT EOI);
250:       ReadChar(Datei, zeichen);
251:     END;
252:     Close(Datei);
253:   END Dateidrucken;
254:
255:   PROCEDURE Druck;
256:   BEGIN
257:     DruckerAnsprechen;
258:     WriteString("Kein Multitasking - STOP mit <Ctrl> ");
259:     WriteLn;
260:     Dateidrucken;
261:     DruckerAbhaengen;
262:   END Druck;
263:
264:   BEGIN (* CPrint *)
265:     WriteString("Amiga IEC-Drucktreiber, ");
266:     WriteString(version);
267:     WriteString(", (c) MERLIN-Computer"); WriteLn;
268:     Argzahl:=NumArgs();
269:     IF Argzahl>0
270:     THEN REPEAT
271:       Dateiselektieren(Argzahl,ok);
272:       IF ok THEN Druck END;
273:       DEC(Argzahl);
274:     UNTIL Argzahl=0;
275:     ELSE REPEAT
276:       Dateiwaehlen(ok);
277:       IF ok THEN Druck END;
278:     UNTIL NOT ok;
279:   END;
280:   WriteString(" -> Programmende");
281:   WriteLn;
282: END CPrint.mod

```

CPRINT, der Prinzenkuß für einen seriellen Commodore-Drucker.

Jetzt geht's los

Um die Übertragung von Daten nachzuvollziehen, wollen wir folgendes Problem lösen:

Ein Drucker am Bus soll die Zahl "1" ausdrucken!

1. Der Controller setzt ATNout auf LOW, um anzuzeigen, daß ein Befehl folgt, und setzt dann CKLout auch auf LOW.
2. Als Reaktion muß der Drucker innerhalb einer Millisekunde die Leitung DATA auf LOW legen. Geschieht dies nicht, gibt der Controller die bekannte Fehlermeldung "Device not present" aus. Liest der Controller aber rechtzeitig ein LOW auf DATAin, so signalisiert er mit CLKout = HIGH, daß ein Byte übertragen werden soll.

3. Solange der Drucker noch nicht bereit ist, Daten aufzunehmen, hält er die DATA-Leitung weiter auf LOW. Ist er soweit, wird DATA auf HIGH gesetzt.

Jetzt beginnt der Controller mit der Übertragung eines Aktionsbytes. Das zu sendene Byte hat den Wert hexadezimal \$24 = \$20 +\$04. Darin ist die Aktion LISTEN = \$20 (Drucker auf Empfang schalten) und die Geräteadresse des Drucker \$04 enthalten.

Handshaking

Die Übertragung erfolgt nach dem "Handshake"-Prinzip, das auch im Datenmodus verwendet wird:

4. Zunächst legt der Controller die CLK-Leitung auf LOW. Das Byte wird (in unserem Beispiel ist dies dual 0110110) nun bitweise von rechts nach links übertragen. Ist das jeweilige Bit nicht gesetzt, so bleibt DATAout auf HIGH, andernfalls wird es LOW. Die Gültigkeit des DATA-Wertes wird dem Drucker signalisiert, indem die CLK-Leitung für min. 20 Mikrosekunden auf HIGH und dann wieder auf LOW gelegt wird. Dann wird DATAout auf HIGH gesetzt und das jeweils nächste Bit bis hin zum achten übertragen.

5. Hat die Übertragung geklappt, d.h., hat der Drucker die Information verstanden, setzt er innerhalb einer Millisekunde die DATA-Leitung auf LOW. Der Controller überwacht dies und gibt gegebenenfalls eine Fehlermeldung aus.

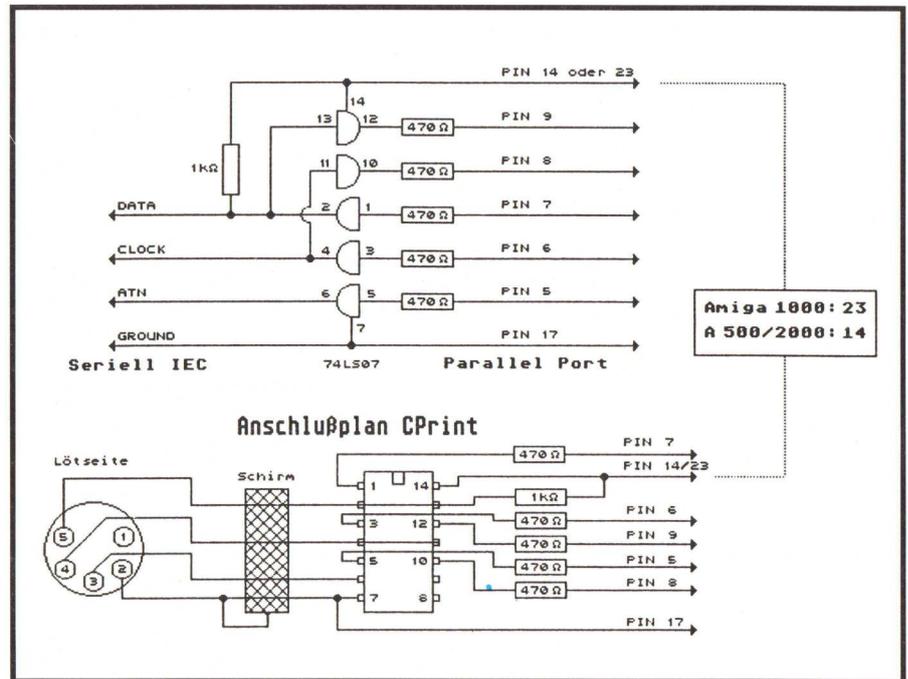
6. Wenn keine weitere Information übertragen werden soll, ist der Kommandomodus beendet, die ATN-Leitung wird wieder auf HIGH gelegt.

7. Der Datentransfer beginnt damit, daß der Controller CLKout auf HIGH setzt. Damit zeigt er an, daß ein Byte zur Übertragung bereitsteht.

8. Der Drucker meldet seine Empfangsbereitschaft durch DATA = HIGH.

9. Innerhalb von 250 Mikrosekunden setzt der Controller die CLK-Leitung auf LOW und schickt dann mit jedem folgenden HIGH-Impuls auf der CLK-Leitung die Bits über die Leitung DATA.

10. Das komplette Byte wird vom Drucker mit DATA = LOW quittiert. Es war unsere Zahl "1", im COM-MODORE-ASCII-Code das Byte \$31.



Schalt-, und Anschlußplan des Converters

Das letzte Byte

Nun bekommt die CLK-Leitung eine weitere Aufgabe: Dem Drucker soll signalisiert werden, daß nun das letzte Byte der Transfers folgt (EOI = End Of Information).

11./12. Nach dem bekannten CLK = HIGH des Controllers und der Bestätigung DATA = HIGH des Druckers läßt der Controller die CLK-Leitung zunächst unverändert. Der

Drucker erkennt dies als EOI-Kommando und quittiert diese Erkenntnis durch die Folge DATA = LOW und DATA = HIGH.

13. Erst jetzt folgt die Controllerbestätigung CLK = LOW und die bekannte Übertragung des Codes \$0D (CR = Wagenrücklauf).

Dies bringt den Drucker endlich dazu, seine Zahl auf das Papier zu hämmern.

ENDE

Personal Computer
GENTELE

Poststr.41 7110 Öhringen Ruf 07941-62673

Festplatte für Amiga 500/1000
21 MB komplett Anschlußfertig
nur 998.-DM

PROBLEME MIT ENGLISCHER ANLEITUNG?

Übersetzen von englischen Bedienungsanleitungen, Computerhandbüchern und Programmen für AMIGA, Atari, Macintosh, MS-DOS, Unix, Xenix, Periferiegeräte und andere durch *Spezialisten*.

Sonderangebote deutscher Anleitungen für AMIGA

Championship Golf	30,-	DBW Render	20,-
Flight Simulator II	35,-	Digal doc Kurzanleitung	15,-
Starglider	20,-	ASDG-RRD Ram Disk	15,-
Digi-View V 2.0 NTSC/PAL	30,-	NEWZAP V 3.0 Diskettenmonitor	15,-
Sound Scape Sampler	20,-	PrDrvGen Druckertreibergenerator	15,-
The Music Studio	40,-	Expert Cartridge	15,-

Bei Versand im Inland berechnen wir DM 5,- für Porto Verpackung/NN. Versand ins Ausland nur mit Vorkasse + DM 4,- für Porto/Verpackung

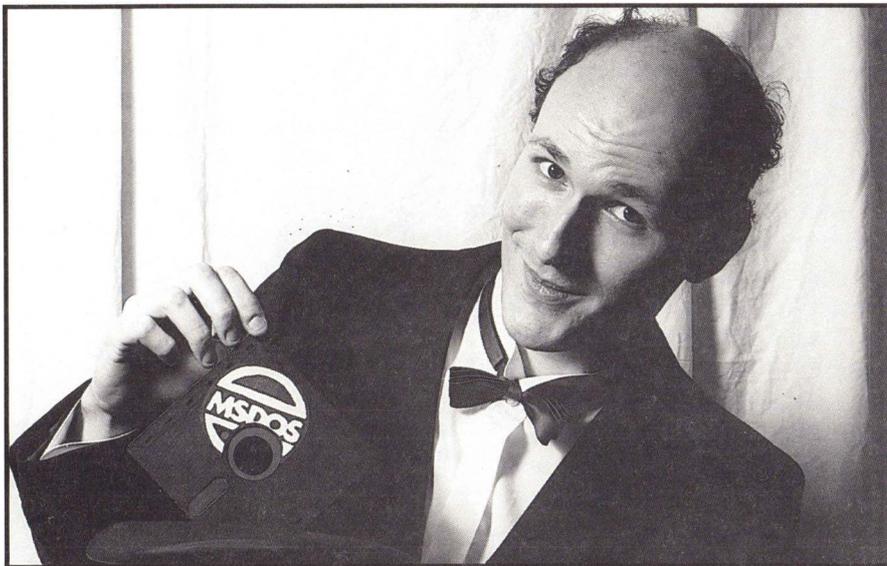


0 83 74 - 98 73

T. Sonnenmoser · Hauptstraße 26 · D-8961 Haldenwang

PC-INTIM

Teil 5: Erinnerungsvermögen



Heutiges Thema von PC-Intim soll der Zugriff auf Massenspeicher unter MS-DOS sein. Dieses Thema ist aber auch für AMIGA-Programmierer interessant, sofern eine Janusfestplatte im System vorhanden ist.

Das Betriebssystem des IBM ist so ausgelegt, daß alle wichtigen Geräte wie Tastatur, Bildschirm oder die Schnittstellen über einen Softwareinterrupt angesprochen werden. Das gilt auch für die Massenspeicher wie Festplatte, Diskettenlaufwerke oder den Kassettenrecorder. Der Kassettenrecorder wird über den Interrupt 15h angesprochen. Da man bei Sidecar und Bridgeboard auf die Implementierung dieses (heutzutage) unsinnigen Gerätes verzichtet hat, gibt ein aufgerufenes INT15h nur den Wert 86h im AH-Register zurück, was

einen "Illegal Command Error" anzeigt. In diesem Punkt gibt es also eine gewisse Inkompatibilität zum Original. Beim Interrupt 13h, der für die restlichen Massenspeicher zuständig ist, herrscht wieder eitel Sonnenschein. Hier ist alles so wie beim großen blauen Bruder.

Im Gegensatz zum AMIGA werden Festplatten und Diskettenlaufwerke über einen Treiber angesprochen. Im Grundzustand (ohne Festplatte) ist der PC nur in der Lage, Diskettenlaufwerke zu bedienen. Die Routinen hierzu befinden sich im eingebauten ROM.

Die Erweiterungen zum Betrieb einer Festplatte befinden sich in einem EPROM, das seinen Platz auf dem Festplattencontroller hat. Ist ein Festplattencontroller in einen Slot eingesteckt, wird diese Bioserweiterung beim Booten des Rechners erkannt. Dazu steht in den ersten zwei Bytes des Eproms die Kennung "55 AA". Beim Hochfahren des Rechners wird eine Routine ausgeführt, die im Bereich C0000h bis F0000h in Abständen von 2 kBytes nachsieht, ob diese Kennung vorhanden ist. Wird die Kennung entdeckt, liest die Routine das dritte Byte aus, das die Anzahl der 512-Byte-Blocks angibt. Über diesen Bereich wird eine Prüfsumme errechnet und, falls diese korrekt ist, die Initialisierungsroutine des ROMs ausgeführt. Der Anfang dieser Routine ist das vierte Byte des ROMs.

Kräftig Biegen

Diese Routine biegt nun den Interruptvektor des INT13h auf das BIOS des Festplattencontrollers um. Wird nun ein Interrupt abgearbeitet, muß dieser den Festplattencontroller passieren, was diesem die Möglichkeit gibt, die an ihn gerichteten Kommandos abzufangen. Ansonsten wird der Interrupt an das PC-BIOS weitergereicht. Dies geschieht über den Interrupt 40h, der vom Harddisk-BIOS auf die Adresse des Standardeinsprungs im PC-BIOS F000:EC59 gerichtet wird. Im Normalfall lädt das MS-DOS aber noch eine Routine für den INT13h nach, wodurch dieser noch einmal umgelenkt wird und zuerst in das Segment 70h zeigt.

Durchnummeriert

Systemintern werden die physikalischen Laufwerke mit Zahlen gekennzeichnet. Dabei tragen die Diskettenlaufwerke die Nummern 0 bis 3, die Festplatten die Nummern 80h und 81h. Der PC kann hardwareseitig bis zu 4 Floppydrives und 2 Harddisks ansprechen. Die Kennzeichnung der Festplatten mit 80h und 81h erfolgte aus dem einfachen Grund, über das MSB (Most Significant Bit, das höchstwertige Bit eines Bytes) eine leichte Unterscheidung zwischen Festplatten- und Diskettenzugriff treffen zu können.

Der Interrupt 13h ist, wie alle Interrupts, in verschiedene Unterfunktionen gegliedert. Diese werden über einen Wert im AH-Register ausgewählt. Die Parameterübergabe ist für alle Funktionen standardisiert. Dabei gelten folgende Vereinbarungen:

AH: Funktionsnummer.

AL: Anzahl der zu bearbeitenden Sektoren. Bei Diskettenzugriffen müssen diese Sektoren alle auf die angegebene Spur passen, ansonsten wird der Fehlercode 04h zurückgegeben. Wird bei einer Festplattenoperation das Spurende erreicht, so wird auf dem ersten Sektor des nächsten Kopfes weitergearbeitet. Wird das Ende des letzten Kopfes erreicht, wird auf der nächsten Spur mit dem ersten Sektor des ersten Kopfes fortgefahren.

CL: Sektornummer. Diese darf bei Diskettenzugriffen zwischen 1 und 8 bzw. 9 liegen (je nach Diskettentyp), bei Festplattenzugriffen zwischen 1 und 17. Bei Festplattenoperationen werden die Bits 6 und 7 als Bits 8 und 9 der Spurnummer interpretiert.

CH: Spurnummer (0 bis 39 bzw. 79 bei Disketten, 0 bis 1023 bei Festplatten). Da mit einem Byte nur bis 255 nummeriert werden kann, werden bei Festplattenzugriffen die beiden oberen Bits von CL zu CH genommen. Die Spurnummer berechnet sich dann wie folgt:

```
(CL AND C0h) * 4 + CH = Spurnummer;  
CH = (Spurnummer AND FFh);  
CL = (Spurnummer AND 300h) / 4 + Sektornummer;
```

DL: Laufwerksnummer. Bei Diskettenlaufwerken kann diese 0 bis 3 lauten, bei Festplatten 80h und 81h.

DH: Kopfnummer (Disketten 0 oder 1, Festplatten 0 bis 7).

ES: Segmentadresse des Datenpuffers.

BX: Offsetadresse des Datenpuffers. Nicht alle Funktionen benötigen alle Parameter.

Die einzelnen Funktionen gliedern sich wie folgt auf:

AH=00h

Der PC setzt das Diskettensystem in einen definierten Zustand zurück. Außerdem wird ein Reset-Impuls an den Disk-Controller gesendet. Wird im DL-Register eine Festplatte angegeben, wird auch ein Reset des Festplattensystems durchgeführt. Diese Funktion sollte immer aufgerufen werden, wenn ein Fehler bei einer Operation des INT 13h aufgetreten ist.

Benötigte Parameter: DL.

AH=01h

Es wird der Fehlerstatus der letzten Diskoperation angezeigt. Dieser Status entspricht dem, der im AH-Register von der Funktion zurückgegeben wurde. Wird im DL-Register eine Diskettenkennung abgelegt, wird der Status der letzten Diskettenoperation abgefragt, auch wenn diese auf ein anderes Laufwerk bezogen war. Der Status-Zwischenspeicher wird dabei nicht verändert. Im Gegensatz dazu verändert eine Statusabfrage der Festplatten den Status, wodurch ein mehrfacher Aufruf zu Fehlinterpretationen führen kann.

Benötigte Parameter: DL.

AH=02h

Lesezugriff auf das angegebene Laufwerk.

Benötigte Parameter: AL,CL,CH,DL, DH, ES, BX.

AH=03h

Schreibzugriff auf das angegebene Laufwerk.

Benötigte Parameter: AL,CL,CH,DL, DH, ES, BX.

AH=04h

Verifizieren der angegebenen Sektoren. Da nur eine Überprüfung über eine Prüfsumme controllerintern stattfindet, muß kein Datenpuffer angegeben werden.

Benötigte Parameter: AL,CL,CH, DL, DH.

AH=05h

Formatieren einer Spur. Bei dieser Operation wird in ES:BX ein Pointer übergeben, der auf einen Puffer zeigt, in dem die einzelnen Sektor-IDs stehen. Für jeden Sektor sind 4 Bytes anzugeben:

1. Spurnummer
2. Seitennummer
3. Sektornummer
4. Anzahl der Bytes im Sektor:
0 = 128 Bytes
1 = 256 Bytes
2 = 512 Bytes
4 = 1024 Bytes

Weitere Parameter, die zum Formatieren notwendig sind, werden aus der Disk-Parameter-Tabelle übernommen, die über INT 1Eh erreichbar ist [1].

Bei der Formatierung der Festplatte wird eine andere Parameterübergabe gefordert, die pro Sektor nur zwei Bytes belegt. Im ersten Byte wird der Status der Formatierung zurückgegeben (00 = Sektor formatiert, 80 = Sektor nicht formatiert), im zweiten Byte wird die logische Sektornummer des physikalischen Sektors angegeben (hierdurch wird der Interleave festgelegt). Alle weiteren Funktionen sind bei einem PC nur auf die Festplatte anwendbar. Beim AT gibt es noch einige Diskettenoperationen, die nur der Vollständigkeit halber hier kurz aufgeführt sind:

AH=15h

Ermittelt den Typ des Laufwerks.

AH=16h

Fragt den Diskettenwechsel ab (AT-Laufwerke können diesen anzeigen).

AH=17h

Bestimmt das Diskettenformat beim Formatieren.

Die Festplattenfunktionen lauten wie folgt:

AH=06h

Kennzeichnet defekte Spuren für das System mit dem Bad-Track-Flag.

Benötigte Parameter: CH,DL,DH.

AH=07h

Formatiert die Festplatte ab der in CH angegebenen Spur.

Benötigte Parameter: CH,DL,DH.

AH=08h

Ermittelt die Laufwerksdaten des in DL angegebenen Festplattenlauf-

werks. Die Daten werden in folgendem Format zurückgegeben:

DL: Anzahl der angeschlossenen Festplatten.

DH: Kopfanzahl(beginnend mit 0).

CH,CL: Sektoranzahl und Spuranzahl, codiert wie oben spezifiziert.

Anzumerken ist, daß diese Daten nicht aus der Tabelle des INT 41h (Festplattenparameterblock) übernommen werden. Bei Controllern mit dynamischer Konfigurationsmöglichkeit sind die Werte des INT 41h unter Umständen nicht korrekt, da dieser Vektor auf die per Jumper definierte Standardtabelle zeigt.

AH=09h

Initialisiert die Tabelle des INT41h (1.Festplatte) und INT46h (2.Festplatte).

Benötigte Parameter: DL.

AH=0Ah

Liest Sektor wie Funktion 02h, übergibt aber zusätzlich die vier ECC-Bytes an das DOS.

Benötigte Parameter: wie Funktion 02h.

AH=0Bh

Schreibt Sektor wie Funktion 03h, benötigt aber hinter dem Sektorpuffer die Angabe der vier ECC-Bytes, die vom Anwender berechnet werden müssen.

Benötigte Parameter: wie Funktion 03h.

AH=0Ch

Setzt Kopf auf die in CH und CL kodierte Spur.

Benötigte Parameter: CL,CH,DL.

AH=0Dh

Funktion wie 00h, setzt aber die Diskettenlaufwerke nicht zurück. Mit dieser Funktion kann auch auf Festplatten mit höherer Laufwerksnummer als 81h zugegriffen werden.

Benötigte Parameter: DL.

AH=0Eh

Liest den Sektorpuffer aus. Diese Funktion kann bei auftretenden Lesefehlern nützlich sein, da in diesem Fall keine Daten ans DOS übergeben werden. Der Datenpuffer muß in ES:BX angegeben werden.

Benötigte Parameter: DL,ES,BX.

AH=0Fh

Schreibt in den Sektorpuffer des Diskcontrollers.

Benötigte Parameter: DL,ES,BX.

AH=10h

Prüft Laufwerk auf Bereitschaft. Speziell zur Abfrage geeignet, ob die letzte Operation abgeschlossen ist.

Benötigte Parameter: DL.

AH=11h

Rekalibriert das spezifizierte Laufwerk, d.h., fährt die Köpfe auf Spur 0.

Benötigte Parameter: DL.

AH=12h

Führt einen Test des Zwischenspeichers durch.

Benötigte Parameter: DL.

AH=13h

Überprüft die Lesbarkeit von Sektor 0 jeder Spur.

Benötigte Parameter: DL.

AH=14h

Ein vollständiger Selbsttest des Festplattencontrollers wird durchgeführt.

Bei all diesen Funktionen können natürlich auch Fehler auftreten. Sie werden durch das Carry-Flag indiziert. In diesem Falle wird dann in AH ein Fehlercode übergeben. Die Kennzahlen lauten:

```
AH = 0 : Kein Fehler aufgetreten.
AH = 1 : Ungültiger Funktionsaufruf.
AH = 2 : Adressmarke des Sektors ungültig.
AH = 3 : Schreibschutz ist aktiv (nur Disk).
AH = 4 : Sektor nicht gefunden.
AH = 5 : Reset des Plattencontrollers fehlerhaft.
AH = 6 : Diskettenwechsel (nur AT).
AH = 7 : Fehler bei Initialisierung des Controllers (nur HD).
AH = 8 : Timingfehler des DMA-Controllers (nur Disk).
AH = 9 : Adressüberschreitung des DMA-Controllers.
AH = B : Zugriff auf defekt gekennzeichnete Spur (nur HD).
AH = 10 : Lesefehler.
AH = 11 : Lesefehler, korrigiert über Prüfbits (nur HD).
AH = 20 : Controller fehlerhaft.
AH = 40 : Spur nicht gefunden.
AH = 80 : Laufwerk antwortet nicht.
AH = BB : BIOS-Fehler.
AH = FF : Nicht identifizierbarer Fehler.
```

An der grünen Grenze

Wer nun eine Festplatte am PC angeschlossen und auf dieser eine AMIGA-Partition eingerichtet hat, kommt nun in einen besonderen Genuß des Software-Interrupts 13h. Er kann nämlich auch vom AMIGA ausgelöst werden. Dazu wird der Interruptvektor 13h auf eine Routine umgebogen, die im Pseu-

do-ROM im Segment E000h liegt. Pseudo-ROM deshalb, weil die Speicherzellen zwar RAMs sind, der Inhalt aber als BIOS-ROM-Erweiterung identifiziert wird. Dies geschieht durch die Kennung "55 AA", die den Rechner veranlaßt, wie oben schon aufgeführt, den Speicher als BIOS-Erweiterung zu initialisieren. Diese Routine wird vom Amiga aus in diesen Speicherbereich geschrieben (das RAM liegt im Dual Ported RAM), was den Vorteil hat, die Routine jederzeit ändern zu können, falls es neue Treibersoftware für die Janus-Platte gibt (in der Tat existieren zwei Treiberversionen, V2.13 und V2.20, die nicht kompatibel sind). Ein Nachteil sei nicht verschwiegen: Gerät der PC außer Kontrolle, ist es möglich, daß in den Pseudo-ROM-Bereich geschrieben wird, was der PC bei einem Neustart mit einem "Optional ROM bad Checksum" honoriert. Festplattenzugriffe des AMIGA führen dann zu einem Absturz desselben, ein kompletter Neustart ist also angesagt. Dies tritt aber gottlob nur äußerst selten ein.

Wird nun ein INT13h ausgelöst, wird in diese Routine verzweigt und festgestellt, ob der Interrupt vom AMIGA ausgelöst wurde. Ist dies der Fall, werden die Parameter, die der AMIGA übergeben hat, vom PC übernommen und der Interrupt ausgeführt. Zu diesem Zweck stellt die Janus-Library die Datenstruktur DskAbsReq.0 zur Verfügung:

```

STRUCTURE DskAbsReq.0
UWORD  dar_FktCode  ;BIOS-Funktionscode
UWORD  dar_Count   ;Sektorzähler
UWORD  dar_Track   ;Zylindernr.
UWORD  dar_Sector  ;Sektornr.
UWORD  dar_Drive   ;Laufwerksnr.
UWORD  dar_Head    ;Kopfnr.
UWORD  dar_Offset  ;Offset im Datenpuffer
UWORD  dar_Status  ;Zurückgegebener Funktionsstatus
LABEL  DskAbsReq_SIZEOF

```

Diese Parameter werden direkt an den Interrupt 13h übergeben. Der zurückgegebene Status entspricht den Fehlercodes, die der PC selbst zurückgibt.

ENDE

WANTED! PD-SOFTWARE GESUCHT!

Belohnung: 5 PD-Disketten

Wer aufmerksam das PD-Angebot studiert, wird schnell feststellen, daß fast alle Programme und auch die meisten Grafik- und Sounddemos von amerikanischen und englischen Programmierern erstellt werden. Dies hat unter anderem zur Folge, daß fast alle Kommentare in den Programmen und vor allem die Anleitungen in englisch verfaßt sind. Dies macht es manchem deutschen AMIGA-Einsteiger nicht leicht, den AMIGA und dessen Programmie-

rung kennenzulernen.

Wir wollen deshalb noch einmal alle Programmierer im deutschsprachigen Raum aufrufen, ihre selbstgeschriebenen Programme, die sonst in einer dunklen Schublade verschwinden, in die PD-Sammlung zu geben.

Als kleinen Anreiz erhält jeder Programmierer, dessen Programm in den KICKSTART PD SERVICE aufgenommen wird, fünf PD-Disketten seiner Wahl.

BATCH-DATEIEN

SIND KEIN GEHEIMNIS

Auf die Frage, ob man mit MS-DOS auch programmieren kann, antworten die meisten PC-User mit einem klaren und nachdrücklichen Nein. Natürlich ist MS-DOS nicht mit einer Programmiersprache zu vergleichen, doch einfache Programme können tatsächlich erstellt werden.

Das Zauberwort für die Programmierung mit MS-DOS ist "Batch-Datei". Im folgenden Artikel werden wir uns mit der bekanntesten Batch-Datei, der "Autoexec.bat", beschäftigen. Die Autoexec-Datei enthält Anweisungen, die direkt nach dem Einschalten des Computers ausgeführt werden. Sie ist vergleichbar mit der "Startup-Sequence" auf dem Amiga.

Allgemeines zu Batch-Dateien

Batch-Dateien sind Stapeldateien. Sie steuern den automatischen Ablauf von Programmen, den sogenannten Stapelbetrieb eines Computers. Um eine Stapeldatei einzugeben, benötigen Sie einen Texteditor beziehungsweise eine Textverarbeitung, mit der sich reiner ASCII-Code speichern läßt. In einer Batch-Datei finden alle DOS-Befehle Verwendung, die im Direkt-Modus über die Tastatur eingegeben werden. Darüber hinaus stehen einige spezielle Befehle zur Verfügung.

Die speziellen Befehle für Batch-Dateien dienen der Kontrolle des Ablaufs einer Stapeldatei. Es handelt sich hierbei um drei komplexe Befehlsgruppen: FOR, IF und GOTO (Elemente der Befehle siehe Tabelle 1). Als vierte Ergänzung des Befehlssatzes für Batch-Dateien gibt es den Befehl REM. Er kennzeichnet Kommentarzeilen.

Bevor wir diese drei Befehlsgruppen genauer betrachten, müssen wir noch auf die Variablen eingehen. Variablen, die in Batch-Dateien eingelesen werden sollen (bis zu zehn sind gleichzeitig erlaubt), sind durch das %-Zeichen und die Ziffern 1 bis 9 gekennzeichnet (%0, %1, %2,...,%9). Die Variable des FOR-Befehls ist für das Betriebssystem eine interne Variable, darf nicht die Ziffern (0 bis 9) verwenden und wird durch zwei Prozentzeichen eingeleitet (Beispiel: %%a). Verschiedene Arbeiten mit MS-DOS sind sehr umständlich zu handhaben. Sollen zum Beispiel mehrere Dateien mit unterschiedlichen Namen und Ergänzungen gelöscht werden, bleibt nichts anderes

übrig, als immer wieder den Befehl, gefolgt von den entsprechenden Namen, einzugeben. Abhilfe schafft hier der FOR-Befehl. Er überträgt die Namen in eine Liste und führt dann den entsprechenden Befehl, in unserem Beispiel ERASE, aus.

Die Befehlszeile

```
FOR %%A IN (%0 %1 %2 %3 %4) DO
ERASE %%A
```

liest fünf Variablen in eine Liste ein und löscht sie anschließend.

Mit der IF-Anweisung werden Befehle bedingt ausgeführt. Das heißt, sie werden nur ausgeführt, wenn die angegebene Bedingung erfüllt ist. Sollen mehrere Befehle bedingt ausgeführt werden, so muß die Batch-Datei mit einem GOTO-Befehl zu einem Label springen. Die Struktur der IF-Anweisung lautet vereinfacht: Wenn Bedingung erfüllt, dann Anweisung.

Als Ergänzung für die IF-Abfrage können "NOT" als Negation, "EXIST", "STRING 1 = = STRING2" und "ERRORLEVEL < = > ZAHL" verwendet werden. Will man zum Beispiel, soweit vorhanden, alle Document-Dateien eines Laufwerkes auf ein anderes überspielen, lautet die IF-Zeile wie folgt:

```
IF EXIST A:*.DOC COPY *.DOC B:
```

Will man sich melden lassen, wenn Document-Dateien nicht existieren, lautet die Anweisung:

```
IF NOT EXIST A:*.*DOC ECHO Keine
DOC-Datei vorhanden
```

Der Stringvergleich kommt bei Eingabe von Variablen in Betracht. So wird eine einfache Passwortabfrage mit

```
IF %1 == PASSWORT ANWEISUNG
```

realisiert. Last but not least wird mit ERRORLEVEL eine Fehlerbedingung abgefragt:

```
IF ERRORLEVEL ZAHL ECHO Fehler
XX ist aufgetreten
```

In all diesen IF-Bedingungen kann mittels der GOTO-Anweisung in einen anderen Teil der Batch-Datei gesprungen werden. Die Syntax lautet:

```
GOTO LABEL
```

Die Bezeichnung Label ist ein frei wählbarer Name. Das Label selbst,

also die Adresse, zu der verzweigt wird, wird mit einem Doppelpunkt gekennzeichnet. Beispiel:

```
IF ERRORLEVEL 2 GOTO LABEL1
.
.
.
:LABEL1
ECHO Es ist der Fehler Nummer 2
ECHO aufgetreten. Sie haben etwas
ECHO falsch gemacht!
```

Die AUTOEXEC.BAT

Um dem Benutzer eines PC beim Start des Computers immer wiederkehrende Eingaben zu ersparen, wurde die Datei Autoexec.bat als sogenannte Startdatei definiert. Das heißt, findet der Computer nach dem Bootvorgang eine Datei mit diesem Namen, werden alle in ihr enthaltenen Anweisungen ausgeführt,

und erst dann die Kontrolle an den Benutzer weitergegeben. Ein Beispiel für eine einfache Autoexec-Datei finden Sie in Bild 1.

Die Echo-Befehle in einer Stapeldatei geben den Text, der hinter ihnen steht, auf dem Bildschirm aus. Mit "Keybgr" wird die deutsche Tastaturbelegung in das System eingebunden. Die Bezeichnung anderer Tastaturbelegungen (KEYxxx) entnehmen Sie bitte Ihrem Handbuch. Der Path-Befehl gehört zu den wichtigsten Anweisungen für Festplattenbenutzer. Mit ihm wird die Verzeichnisstruktur der Festplatte angegeben. Das bedeutet in der Praxis, daß Sie in jedem beliebigen Verzeichnis ein Programm aufrufen können, das in einem anderen Verzeichnis auf der Festplatte steht.

Mit einer Laufwerksangabe wird das aktuelle Verzeichnis neu definiert. Sinnvoll ist dies zum Beispiel, wenn sich in Laufwerk A: die Systemdiskette, mit der gestartet wird, befindet, und automatisch auf die Datendiskette in Laufwerk B: umgeschaltet werden soll. Schon die "kleine" Autoexec-Datei (Bild 1) erspart dem Anwender viel Tipparbeit beim Systemstart, und reicht wohl in den meisten Fällen schon aus. Bild 2 zeigt eine ausführlichere Startdatei. Gegenüber dem ersten Beispiel haben sich in den ersten Zeilen nur die Echo-Befehle geändert. Um den Bildschirm nicht mit Anzeigen zu überfüllen, wurden sie mit REM in Kommentarzeilen umgewandelt und dienen nur der Dokumentation. Neu hinzugekommen ist die Prompt-Anweisung. Mit ihr wird die "Bereit"-Anzeige in der Befehlszeile verändert. In unserem Beispiel bedeuten \$P die Anzeige des aktuellen Verzeichnisses und Laufwerks, \$G fügt hinter der ersten Anzeige das Zeichen ">" an. Dabei muß man sich allerdings nicht unbedingt an die Standards halten. Auch eine Eingabe wie "PROMPT Was willst Du?" ist zugelassen.

Die IF-Abfrage kontrolliert das Vorhandensein der Benutzeroberfläche "1dirplus". Ist sie nicht im angegebenen Laufwerk zu finden, wird zum ersten Label (Weiter1) verzweigt, und eine entsprechende Meldung ausgegeben. Danach wird die Verarbeitung der Stapeldatei beendet. Ist die Benutzeroberfläche vorhanden, wird das Programm aufgerufen und erst danach

```
D:\AUTO1.BAT
-EDIT-
ECHO Autoexec.bat zum Vorfuehren
KEYBGR
ECHO Der Befehl Keybgr stellt den deutschen Zeichensatz ein
PATH=d:\d:\system
ECHO Der Befehl Path definiert Zugriffspfade
D:
ECHO D: schaltet auf das Laufwerk D: um

<F1> <F2> <F3> <F4> <F5> <F6> <F7> <F8>
View Display Search FindNext Block New File Print Exit
View the File
```

Ein einfaches Beispiel einer Autoexec.bat-Datei.

```
D:\AUTOEXEC.BAT
-EDIT-
REM Autoexec.bat zum Vorfuehren
KEYBGR
REM Der Befehl Keybgr stellt den deutschen Zeichensatz ein
PATH=d:\d:\system
REM Der Befehl Path definiert Zugriffspfade
D:
REM D: schaltet auf das Laufwerk D: um
PROMPT $P$G
REM In der Eingabezeile (Prompt-Zeichen) wird das aktuelle
REM Verzeichnis und das >-Zeichen angezeigt
IF NOT EXIST 1DIRPLUS.EXE GOTO WEITER1
1dirplus
GOTO WEITER2
:WEITER1
ECHO Menuprogramm 1dirplus nicht gefunden
:WEITER2
REM ENDE

<F1> <F2> <F3> <F4> <F5> <F6> <F7> <F8>
View Display Search FindNext Block New File Print Exit
View the File
```

Eigenen Ideen bei der Gestaltung der Autoexec-Datei sind kaum Grenzen gesetzt.

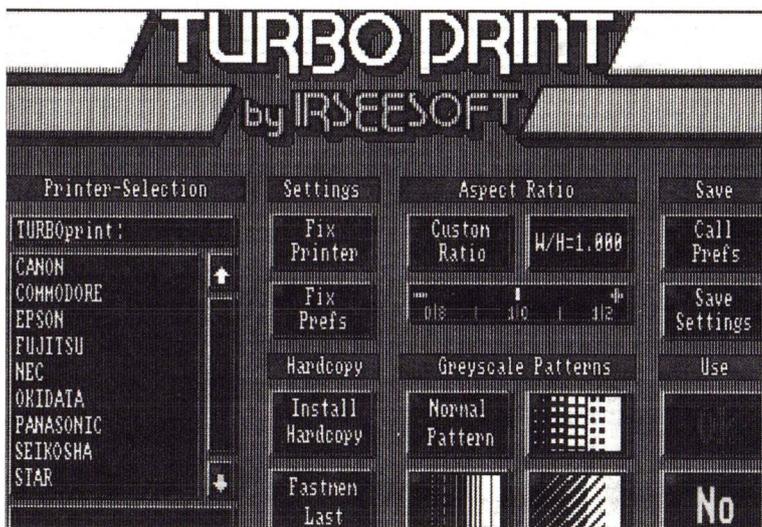
zum Ende der Autoexec-Datei gesprungen. Bei einem Programmaufruf muß nur der Programmname eingegeben werden, die Namensergänzung (im Beispiel Bild 2 ".exe") entfällt.

Anhand der zwei kleinen Beispiele können Sie ersehen, wie individuell eigene Startdateien erstellt werden können. Gerade beim Ausprobieren eigener Ideen entstehen die besten Variationen. Aber wie bei jeder Veränderung an systemspezifischen Dateien heißt die Devise: Erst eine Sicherungskopie des Originals anlegen, dann ausprobieren.

ECHO:	Ausgabe eines Textes auf dem Bildschirm
REM:	Kommentarzeile
:	Definition eines Labels
GOTO	Sprung zu einem Label
FOR	Schleifenbefehl
IF	Abfrage einer Bedingung
NOT	Negierung einer Bedingung
= =	Stringvergleich in einer Abfrage
EXIST	Bedingung einer Abfrage
ERRORLEVEL	Fehlerbedingung einer Abfrage

Spezielle Befehle für die Stapelverarbeitung unter MS-DOS

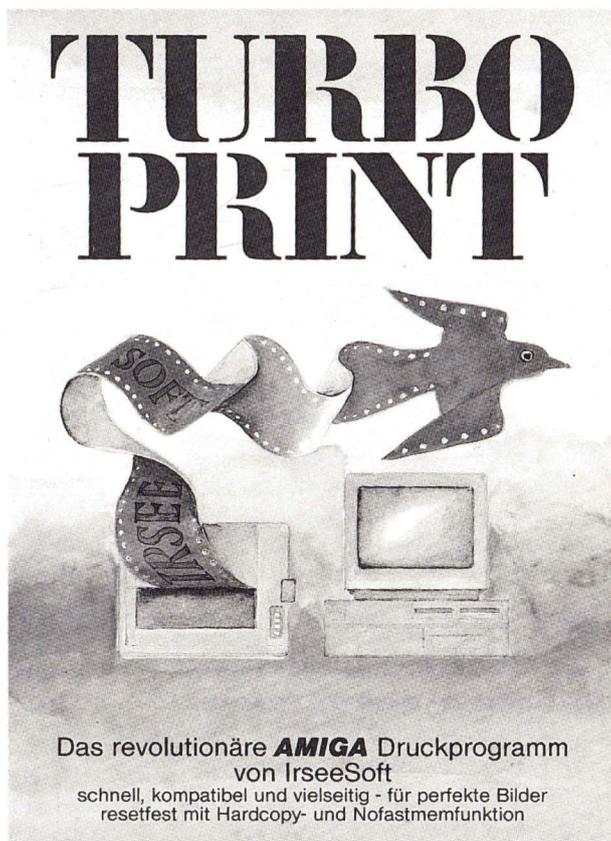
ENDE



TURBOprint – Das komfortable Druckerspeder-Paket für Ihren AMIGA

- bis zu 10x schnellerer Grafikdruck durch 100% 68000-Assembler-Programmierung
- unterstützt alle Bildschirmauflösungen des Amiga
- bessere kontrastreiche Bilder durch vier wählbare Farbsetzungen
- druckt 4096 Farben im HAM-Modus jetzt auch ausschnittsweise und gedreht
- spezielle Turbotreiber für alle Druckdichten Ihres Matrix-, Tintenstrahl-, Thermo- oder Laserdruckers
- resetzeste Hardcopyfunktion druckt jeden Bildschirm aus
- keine Probleme mehr mit Speichererweiterungen durch verbessertes, resetzfestes Nofastmem
- ohne Umstellung wie gewohnt weiterarbeiten
- vollkompatibel zur Amiga-Software
- wird resetzfest im Speicher installiert
- arbeitet unbemerkt im Hintergrund
- kein Umkopieren auf Ihre Software notwendig
- ausführliches deutsches Handbuch
- läuft auf Amiga 500, 1000 und 2000

DM 89,-



Das revolutionäre **AMIGA** Druckprogramm
von IrseeSoft

schnell, kompatibel und vielseitig - für perfekte Bilder
resetzfest mit Hardcopy- und Nofastmemfunktion

Vertrieb: **IrseeSoft SPCS Heinz Donhauser**
Grüntestraße 6
8951 Irsee
Telefon (0 83 41) 7 43 27

PDC GmbH
Louisenstraße 115
6300 Bad Homburg
Telefon (0 61 72) 2 47 48 / 2 07 99

DER COPPER-MONITOR

Dem Custom-Chip in die Karten geschaut

Wozu einen Monitor, werden Sie sich denken, ich habe doch schon einen? Doch der Begriff "Monitor" bezeichnet nicht nur Ihr würfelförmiges Datensichtgerät; auch ein Programm, mit dem Sie bestimmte Daten des Systems betrachten können (quasi die Innereien), wird als "Monitor" bezeichnet.

Der hier vorgestellte Copper-Monitor sucht die aktuelle Copperliste im Speicher und stellt diese in einem Fenster dar, wobei die Liste sogar direkt verändert werden kann. Jene, denen der Begriff "Copper" und seine Funktionsweise vertraut ist, können den nun folgenden Teil ruhig überschlagen.

Der Copper

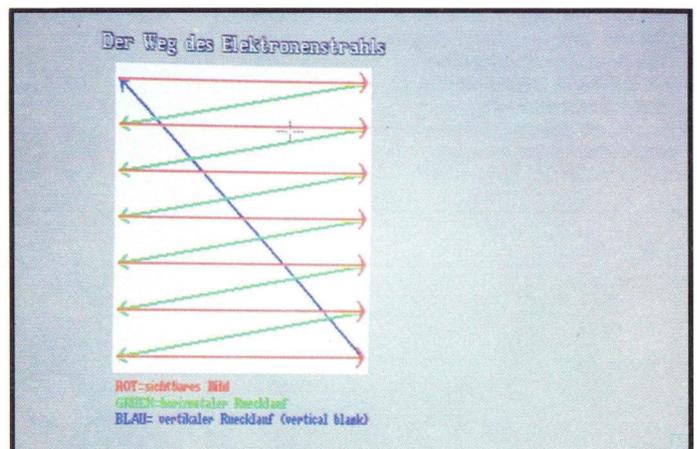
Der COPPER, das unbekannte Wesen? Der Copper ist ein echter Coprozessor in Ihrem Amiga. Das heißt, er verarbeitet in seiner eigenen "Programmiersprache" (die jedoch nur drei Befehle umfaßt) eine bestimmte Aufgabe, ohne direkt Prozessorzeit vom Hauptprozessor in Anspruch zu nehmen. Seine Hauptaufgabe ist es, bestimmte Register der Customchips (diese sorgen z.B. für Grafik und Sound) zu steuern, er organisiert also z.B. den Bildschirm-aufbau. Wie Sie sicher wissen, kann man mit der Maus verschiedene Bildschirme herauf- bzw. herunterziehen, so daß verschiedene Auflösungen und Farben gleichzeitig sichtbar sind. Dieser Effekt wird mit dem Copper er-

zeugt. Der Copper schafft das, indem er

1. auf eine bestimmte Position des Elektronenstrahls wartet. Dieser Strahl wandert 50mal in der Sekunde in der in Bild 1 beschriebenen Weise über den Bildschirm und erzeugt dabei das Bild auf dem Monitor, das Sie sehen.
2. Falls nun der Elektronenstrahl eine bestimmte Position erreicht hat, geht es weiter in der Copper-Liste, so daß der Copper einen anderen Befehl abarbeiten kann. Dies wird

meistens ein Befehl sein, der einen bestimmten Wert in ein Hardware-Register schreibt (wie z.B. Farb- oder Auflösungsänderungen o.ä.). Danach geht's weiter in der Copperliste (das "Programm" des Coppers), so daß weitere Register geändert werden oder erst wieder auf eine bestimmte Position gewartet wird, bevor Hardware-Register geändert werden. Irgendwann ist der Elektronenstrahl ganz unten rechts an einem Punkt angelangt, wo es nicht weitergeht. Dann wandert er automatisch wieder in die linke, obere Ecke, und die Copperliste wird von neuem wieder gestartet. Die Liste wird also (wenn sie zu lang ist) mitten im Lauf abgebrochen. Dieser Vorgang geschieht 50mal in der Sekunde und kann nicht verhindert werden. Am Ende der Copper-Liste reicht also eine unendliche Warteschleife, denn irgendwann geht es ja sowieso wieder von Anfang an los.

Der Elektronenstrahl des Monitors durchläuft stark vereinfacht den hier abgebildeten Weg.



Move	Wait	Skip
HM1 HM2	HM3 HM2	HM1 HM2
13	0 DM15	VP7 BFD
14	0 DM14	VP6 VM6
15	0 DM13	VP5 VM5
16	0 DM12	VP4 VM4
17	0 DM11	VP3 VM3
18	0 DM10	VP2 VM2
19	0 DM9	VP1 VM1
20	RAG DM8	VP8 VM8
21	RA7 DM7	HP8 HM8
22	RA6 DM6	HP7 HM7
23	RA5 DM5	HP6 HM6
24	RA4 DM4	HP5 HM5
25	RA3 DM3	HP4 HM4
26	RA2 DM2	HP3 HM3
27	RAL DM1	HP2 HM2
28	0 DM0	1 0 1 1

R=Register/Offset
 D=Datensort
 Vp/Hp=Vertikale/Horizontale Position
 Vn/Hn=Vertikale/Horizontale Maske
 BFD=Blitter Finish Disable

Alle Register des Customs-Chips können leicht mit dem Copper-Monitor erreicht und geändert werden, aber Vorsicht ist angebracht: Ein falscher Wert im falschen Register, und der Amiga verabschiedet sich.

per-Befehl. Das Programm gibt bei MOVE-Befehlen das Datenwort sowie die Adresse des Hardware-Registers aus (nicht den Offset, sondern die volle Adresse). Nach der Adresse sehen Sie einen Namen. Dieser Name ist der symbolische Name der jeweiligen Adresse (z.B. regelt das Hardware-Register \$DFF180 die Farbe des Hintergrundes, der symbolische Name ist "COLOR00"). Die im Programm benutzten Namen entsprechen (bis auf eine Ausnahme) den offiziellen Symbolen, die auch in Fachbüchern Verwendung finden. Die einzigen Ausnahmen sind jene Namen, die länger sind als 7 Buchstaben. Das sind jedoch nur die Datenregister der Sprites wie "SP0DATA" oder "SP3DATB", die im Programm in "SP0DATA" bzw. "SP3DATB" umbenannt wurden. Falls Sie die Namen nicht dargestellt haben wollen, können Sie sie ein- und ausschalten, indem Sie das Gadget in der unteren rechten Ecke anklicken (da, wo "N" steht, für NAMEN). Nun verschwindet das "N". Falls Sie die Namen aber doch sehen wollen, klicken Sie das Gadget (übrigens ein BOOL-Gadget) noch einmal an. Über diesem Gadget finden Sie ein Proportional-Gadget, wie Sie es z.B. von der Workbench her kennen. Damit können Sie durch die Copperliste scrollen (im Fenster werden nur 16 Befehle dargestellt, die meisten Copper-Listen sind jedoch länger). Wenn Sie das Gadget anklicken, können Sie es vertikal bewegen, wobei der Inhalt des Fensters ständig aktualisiert wird (auch wenn Sie die Maustaste nicht loslassen). So kann man bequem die Copper-Liste ganz überblicken. Wenn Sie das Gadget ganz nach unten ziehen, kommen Sie an das Ende der Copperliste. Das Ende wird automatisch vom Programm erkannt, indem ein WAIT auf eine unmögliche X-Position als Endlos-WAIT erkannt wird, welches meist das Ende der Copperliste bedeutet (INTUITION macht es z.B. so). Falls Sie in das Gadget über oder unter dem Schieber klicken, scrollt die Liste im Fenster um einen Copper-Befehl auf oder ab. Dies waren die Funktionen, um die Copperliste zu betrachten. Man kann die Copperliste jedoch mit dem Copper-Monitor auch ändern: Sie können jeden Copper-Befehl, der im Fenster sichtbar ist, ändern.

Wir haben nun schon zwei Copper-Befehle von dreien kennengelernt, dazu noch die Meistgebrauchten (der dritte ist recht kompliziert, man braucht ihn aber auch kaum): Der erste Befehl heißt MOVE, was nicht mit dem gleichnamigen Assemblerbefehl verwechselt werden darf. Er bedeutet "Bewege" und schreibt einen bestimmten Wert in ein bestimmtes Hardware-Register. Die Syntax lautet

```
MOVE #nn, #xxx
```

#nn ist hierbei das Datenwort, das geschrieben werden soll, #xxx ist der Offset für das Hardware-Register. Ein kurzes Wort zum Offset: da NUR in Hardware-Register (und da auch nicht in alle) geschrieben werden darf, und da alle diese Register ab \$DFF000 liegen, läßt man bei diesem Offset (deutsch: Ausgleich) das \$DFF000 weg, also wird aus \$DFF180 (Register für Hintergrundfarbe) der Offset \$180. Ein gültiger MOVE-Befehl wäre

```
MOVE #$f00, $180 (setzt Hintergrund auf Rot)
```

Der zweite Befehl heißt WAIT (Sie haben es erraten, er "wartet" tatsächlich) und stoppt die Abarbeitung der Copperliste, bis eine bestimmte Bildposition erreicht oder überschritten wird. Die Bildposition wird durch eine X- und eine Y-Koordinate charakterisiert. Die Y-Koordinate kann von 0 bis 312 reichen, da aber nur jeweils ein Byte die Koordinaten bestimmt, kann man nur Werte von 0 bis 255 angeben. Will man z.B. auf Zeile 260 warten, wartet man erst auf die Zeile 255 (X-Position sollte \$dd sein) und dann noch mal 4 Zeilen, also

1. WAIT \$ff, \$dd
2. WAIT \$04, \$01

Es sei noch angemerkt, daß es 112 mögliche X-Positionen gibt, also läßt sich die horizontale Auflösung nur in 4-Pixel-Schritten (Lores) angeben.

Dies war eine kurze Einführung in die Welt des Coppers, Interessierte finden mehr Informationen in Fachbüchern oder im Artikel in der KICKSTART 12/87 über den Copper.

Der Copper-Monitor dient in erster Linie dazu, die interne Copperliste auf dem Bildschirm darzustellen. So können Sie Grundsätzliches über den Copper lernen (wie verwaltet der Copper mehrere Bildschirme, wie wartet der Copper auf Y-Koordinaten größer als 255), oder auch fremde Programme auf Ihre Copperlisten hin überprüfen (starten Sie doch 'mal das PD-Programm "RainBench" auf der Fred-Fish Disk #90 mit RUN und setzen Sie den Copper-Monitor darauf an!). Man kann auf Wunsch auch die symbolischen Namen der Hardware-Register einblenden, und last not least die einzelnen Befehle der Copper-Liste ändern.

Bedienung des Programms:

Nach dem Aufruf des Copper-Monitors wird auf dem Workbench-Screen ein Window geöffnet. Sie können es wie gewohnt verschieben, in den Vorder- oder Hintergrund bringen und schließen. Einzig die Größe kann nicht verändert werden. Wollen Sie das Programm verlassen, klicken Sie einfach das Schließgadget an. Im Fenster sehen Sie sechzehn Zeilen mit je einem Cop-

Eine Warnung vorweg:

Sie sollten wirklich wissen, was Sie machen, wenn Sie die Copperliste ändern, da der Copper z.B. durch einen unkontrollierten Start des Blitters alle Speicherbereiche durcheinandermixen kann, was nicht nur einen effektvollen Absturz (Sound- and Light-Show), sondern auch eine kaputte Diskette zur Folge haben kann, also VORSICHT!! Wenn Sie jedoch zum Kreis der Eingeweihten gehören, die Herr über Copper und Blitter sind, können Sie recht einfach die schon existierenden Befehle ändern. Leider ist es nicht möglich, die Copper-Liste zu verlängern oder zu verkürzen (obwohl es für letzteres einen Tip gibt), weil die Liste eine zentrale Stellung im System einnimmt und da somit sehr viele Zeiger verändert werden müßten! Sie können aber sehr wohl schon existierende Befehle verändern: Jeder Copper-Befehl besteht aus zwei Worten, also aus einem Langwort. Wenn Sie nun einfach den Befehl, den Sie ändern wollen, im Fenster

anklicken, sehen Sie einen sogenannten Requester. Dieser liefert Ihnen erst einmal Informationen über Start, Ende und Länge der Copper-Liste. Sie können den Requester jederzeit verlassen, wenn Sie "OK" anklicken. Falls Sie aber ein Befehlswort geändert haben (einfach anklicken und ändern, es sind zwei String-Gadgets), kann es sein, daß Sie eine falsche Eingabe gemacht haben, da die Eingabe nur in hexadezimaler Form erfolgen darf. Dabei dürfen Sie jedoch führende Nullen weglassen, und Groß-/Kleinschrift wird nicht unterschieden. Bevor das Programm den Requester schließt, wird ein "Syntax-Check" durchgeführt. Wenn sich also bei "OK" nichts tut, ist Ihre Eingabe falsch (nur Ziffern von "0" bis "9" sowie Buchstaben von "a" bis "f" verwenden). Wenn alles in Ordnung ist, wird die Änderung direkt in die Copper-Liste geschrieben und das Programm kehrt in den List-Modus zurück. Beachten Sie, daß die veränderbaren Copper-Befehle im Roh-Format vorliegen (wie tatsächlich im Speicher), so daß Sie einen MOVE

#\$0f00,COLOR00 erst in \$0180 und \$0f00 umwandeln müssen (da Sie aber sowieso wissen sollten, was Sie da ändern ...). Es dürfte jetzt ein leichtes sein, die Copperliste zu ändern. Falls sie einen Copper-Befehl löschen wollen, hier ein Tip: Ersetzen Sie ihn durch einen Copper-Befehl, der nichts bewirkt wie z.B.

```
MOVE #$096,DMACON
```

Tippen Sie hierfür beim Ändern "0096" als Befehlswort 1, "0000" als Befehlswort 2 (DMACON hat den Offset 0), somit ist der Befehl unwirksam gemacht worden.

Zur Eingabe:

Der Copper-Monitor entstand auf einem A500 mit dem Seka-Assembler. Tippen Sie das Listing ohne die Zeilennummern ab, assemblieren Sie es und speichern Sie es ab. Starten Sie es vom CLI aus durch die Eingabe von CopperMonitor (eventuell mit RUN). Viel Spaß beim Experimentieren mit dem Copper!

```
1: * Copper Monitor (C) KICKSTART 1988 *
2: * Autor: Stefan Kemeny *
3:
4: ExecBase = 4
5: GetMsg = -30 -342
6: OldOpenLibrary = -30 -378
7: CloseLibrary = -30 -384
8: CloseWindow = -30 -42
9: EndRequest = -30 -90
10: OpenWindow = -30 -174
11: RefreshGadgets = -30 -192
12: Request = -30 -210
13: Text = -30 -30
14: SetAPen = -30 -312
15: SetBPen = -30 -318
16: Move = -30 -210
17: SPFix = -30 -0
18: SPFlt = -30 -6
19: SPSub = -30 -42
20: SPMul = -30 -48
21: SPDiv = -30 -54
22:
23: Start:
24: bsr.L OeffneLibraries
25: bsr.L InitFenster
26: bsr.L GadgetsRefresh
27: HauptSchleife:
28: bsr.L InitAusgabe
29: bsr.L ListCopper
30: bsr.L GetIDCMP
31: tst.l d0
32: beq.S HauptSchleife
33: move.l d0,a0
34: move.l 20(a0),d0
35: cmpi.l #$200,d0
36: beq.S Endel
37: cmpi.l #8,d0
38: bne.S NamensToggle
39: move.w 24(a0),d0
40: cmpi.w #$0068,d0
```

```
41: bne.S HauptSchleife
42: clr.l d0
43: move.w 34(a0),d0
44: subi.w #10,d0
45: divu #9,d0
46: andi.l #$ffff,d0
47: cmpi.b #15,d0
48: bhi.S HauptSchleife
49: add.l SchieberAktuell,d0
50: mulu #4,d0
51: move.l CListStart,a0
52: add.l d0,a0
53: move.l a0,EditierteAdresse
54: bsr.S EditiereCopper
55: bra.S HauptSchleife
56: NamensToggle:
57: move.l 28(a0),d0
58: cmpi.l #BoolGadget1,d0
59: bne.S HauptSchleife
60: move.l NamenFlag,d0
61: bchg #0,d0
62: move.l d0,NamenFlag
63: bra.L HauptSchleife
64: Endel:
65: bsr.L SchliesseWindow
66: Ende2:
67: bsr.L SchliesseLibraries
68: rts
69:
70: EditiereCopper:
71: move.l CListStart,d0
72: move.l #$100000,d1
73: move.l #16,d2
74: move.l #1,d3
75: lea In2Data,a0
76: bsr.L ConvertAlles
77: move.l CListEnde,d0
78: move.l #$100000,d1
79: move.l #16,d2
80: move.l #1,d3
```

```

81: lea In3Data,a0
82: bsr.L ConvertAlles
83: move.l CListLaenge,d0
84: move.l #1000,d1
85: move.l #10,d2
86: move.l #2,d3
87: lea In4Data,a0
88: bsr.L ConvertAlles
89: move.l EditierteAdresse,a0
90: clr.l d0
91: move.w (a0)+,d0
92: move.l #$1000,d1
93: move.l #16,d2
94: move.l #1,d3
95: move.l a0,-(sp)
96: lea St1Puff,a0
97: bsr.L ConvertAlles
98: move.l St1Puff,Un1Puff
99: move.l (sp)+,a0
100: move.w (a0)+,d0
101: move.l #$1000,d1
102: move.l #16,d2
103: move.l #1,d3
104: lea St2Puff,a0
105: bsr.L ConvertAlles
106: move.l St2Puff,Un2Puff
107: move.l IntBase,a6
108: lea.l EditRequester1,a0
109: move.l Window,a1
110: jsr Request(a6)
111: OKPress:
112: bsr.L GetIDCMP
113: tst.l d0
114: beq.S OKPress
115: move.l d0,a0
116: move.l 20(a0),d0
117: cmpi.w #$20,d0
118: bne.S OKPress
119: move.l 28(a0),d0
120: cmpi.l #BoolGadget2,d0
121: bne.S OKPress
122: RequesterEnde:
123: lea.l St1Puff,a0
124: bsr.S PruefeHex
125: cmpi.l #$f0000000,d0
126: beq.S OKPress
127: move.l d0,d7
128: lea.l St2Puff,a0
129: bsr.S PruefeHex
130: cmpi.l #$f0000000,d0
131: beq.S OKPress
132: move.l d0,d1
133: move.l d7,d0
134: lsl.l #8,d0
135: lsl.l #8,d0
136: add.l d1,d0
137: move.l EditierteAdresse,a0
138: move.l d0,(a0)
139: move.l IntBase,a6
140: lea.l EditRequester1,a0
141: move.l Window,a1
142: jsr EndRequest(a6)
143: rts
144:
145: PruefeHex:
146: move.l (a0),d0
147: move.l #3,d1
148: PruefeRoll1:
149: cmpi.b #32,d0
150: bne.S PruefeRoll13
151: clr.b d0
152: PruefeRoll3:
153: tst.b d0
154: bne.S PruefeRoll12
155: lsr.l #8,d0
156: addi.l #$30000000,d0
157: PruefeRoll2:
158: dbra d1,PruefeRoll1
159: move.l d0,(a0)
160: move.l a0,-(sp)
161: move.l #3,d0
162: PruefeKlein1:

```

```

163: move.b (a0),d1
164: cmpi.b #64,d1
165: bls.S PruefeKlein2
166: cmpi.b #70,d1
167: bhi.S PruefeKlein2
168: addi.b #32,d1
169: PruefeKlein2:
170: move.b d1,(a0)+
171: dbra d0,PruefeKlein1
172: move.l (sp)+,a0
173: move.l a0,-(sp)
174: move.l #3,d0
175: PruefeErlaubt:
176: move.b (a0)+,d1
177: cmpi.b #47,d1
178: bls.S PruefeFehler
179: cmpi.b #102,d1
180: bhi.S PruefeFehler
181: cmpi.b #57,d1
182: bls.S PruefeGueltig
183: cmpi.b #96,d1
184: bls.S PruefeFehler
185: PruefeGueltig:
186: dbra d0,PruefeErlaubt
187: move.l (sp)+,a0
188: move.l #3,d0
189: move.l #$1000,d1
190: clr.l d3
191: PruefeWandel:
192: clr.l d2
193: move.b (a0)+,d2
194: subi.b #48,d2
195: cmpi.b #9,d2
196: bls.S PruefeWandel2
197: subi.b #39,d2
198: PruefeWandel2:
199: mulu d1,d2
200: add.l d2,d3
201: divu #16,d1
202: dbra d0,PruefeWandel
203: move.l d3,d0
204: rts
205: PruefeFehler:
206: move.l (sp)+,a0
207: move.l #$f0000000,d0
208: rts
209:
210:
211: ListCopper:
212: clr.l d0
213: clr.l d1
214: move.w SchieberPos,d1
215: move.w SchieberSchritt,d0
216: divs d0,d1
217: andi.l #ffff,d1
218: move.l d1,SchieberAktuell
219: mulu #4,d1
220: move.l CListStart,a0
221: add.l d1,a0
222: move.l ZeilenZahl,d3
223: clr.l d0
224: clr.l d1
225: ListCopper1:
226: clr.l d0
227: clr.l d1
228: move.w (a0)+,d0
229: move.w (a0)+,d1
230: move.l a0,-(sp)
231: btst #0,d0
232: beq.L ListMove
233: move.l #"WAIT",ColName
234: btst #0,d1
235: beq.S ListWait
236: move.l #"SKIP",ColName
237: ListWait:
238: move.l d0,d2
239: andi.w #fff,d2
240: andi.w #fff0,d0
241: divu #$100,d0
242: movem.l d0-d3/a0,-(sp)
243: move.l d0,d1
244: move.l #100,d2

```

```

245: lea.l VerText, a0
246: bsr.l DezConvert
247: movem.l (sp)+, d0-d3/a0
248: movem.l d0-d3/a0, -(sp)
249: move.l d2, d1
250: move.l #100, d2
251: lea.l HorText, a0
252: bsr.l DezConvert
253: movem.l (sp)+, d0-d3/a0
254: move.l d1, d0
255: move.l d1, d2
256: andi.w #$7f00, d0
257: divu #$100, d0
258: andi.w #$ff, d2
259: move.b #"1", BFDText
260: btst #15, d1
261: bne.S BFDSets
262: move.b #"0", BFDText
263: BFDSets:
264: movem.l d0-d3/a0, -(sp)
265: move.l d0, d1
266: move.l #100, d2
267: lea.l VerMask, a0
268: bsr.l DezConvert
269: movem.l (sp)+, d0-d3/a0
270: movem.l d0-d3/a0, -(sp)
271: move.l d2, d1
272: move.l #100, d2
273: lea.l HorMask, a0
274: bsr.l DezConvert
275: movem.l (sp)+, d0-d3/a0
276: move.l d3, -(sp)
277: move.l GfxBase, a6
278: move.l #0, d0
279: move.l RastPort, a1
280: jsr SetAPen(a6)
281: move.l #3, d0
282: move.l RastPort, a1
283: jsr SetBPen(a6)
284: move.l (sp)+, d3
285: lea.l ColText, a0
286: move.l #ColEnd-ColText, d0
287: bra.l ListIt
288: ListMove:
289: movem.l d0-d3, -(sp)
290: move.l #$100, d1
291: move.l #16, d2
292: move.l #1, d3
293: lea.l RegText, a0
294: bsr.l ConvertAlles
295: movem.l (sp)+, d0-d3
296: movem.l d0-d3, -(sp)
297: move.l d1, d0
298: move.l #$1000, d1
299: move.l #16, d2
300: move.l #00, d3
301: lea.l DatText, a0
302: bsr.l ConvertAlles
303: movem.l (sp)+, d0-d3
304: lea.l NulName, a0
305: cmpi.l #1, NamenFlag
306: bne.S NamePrint
307: subi.l #$40, d0
308: lsl #2, d0
309: lea.l RegName, a0
310: add.l d0, a0
311: NamePrint:
312: lea.l Re2Text, a1
313: move.l (a0)+, (a1)+
314: move.l (a0)+, (a1)+
315: move.l d3, -(sp)
316: move.l GfxBase, a6
317: move.l #2, d0
318: move.l RastPort, a1
319: jsr SetAPen(a6)
320: move.l #1, d0
321: move.l RastPort, a1
322: jsr SetBPen(a6)
323: move.l (sp)+, d3
324: lea.l Co2Text, a0
325: move.l #Co2End-Co2Text, d0
326: ListIt:

```

```

327: move.l d3, -(sp)
328: movem.l a0/d0, -(sp)
329: move.l d3, d0
330: move.l ZeilenZahl, d1
331: sub.l d0, d1
332: mulu #9, d1
333: addi.l #17, d1
334: move.l #4, d0
335: move.l RastPort, a1
336: jsr Move(a6)
337: movem.l (sp)+, a0/d0
338: move.l RastPort, a1
339: jsr Text(a6)
340: move.l (sp)+, d3
341: move.l (sp)+, a0
342: dbra d3, ListCopper1
343: rts
344:
345: GadgetsRefresh:
346: move.l IntBase, a6
347: move.l Window, a1
348: move.l #0, a2
349: move.l 62(a1), a0
350: jsr RefreshGadgets(a6)
351: rts
352:
353: GetIDCMP:
354: move.l ExecBase, a6
355: move.l UserPort, a0
356: jsr GetMsg(a6)
357: rts
358:
359: DezConvert:
360: move.l a0, -(sp)
361: DezConvert2:
362: divu d2, d1
363: addi.w #48, d1
364: move.b d1, (a0)+
365: clr.w d1
366: swap d1
367: cmp.l #1, d2
368: beq.S DezConvertEnd1
369: divu #10, d2
370: bra.S DezConvert2
371: DezConvertEnd1:
372: move.l (sp)+, a0
373: DezConvertLoop:
374: cmpi.b #48, (a0)
375: bne.S DezConvertEnd2
376: move.b #32, (a0)+
377: bra.S DezConvertLoop
378: DezConvertEnd2:
379: rts
380:
381: ConvertAlles:
382: movem.l d0-d5, -(sp)
383: move.l MatBase, a6
384: ConvertLoop1:
385: move.l d0, d4
386: move.l d1, d5
387: bsr.S ConvertDivide
388: move.l d5, d0
389: tst.b d4
390: bne.S ConvertNoZeros1
391: btst #1, d3
392: beq.S ConvertNoZeros2
393: move.b #" ", d4
394: bra.S ConvertNoHex
395: ConvertNoZeros1:
396: bclr #1, d3
397: ConvertNoZeros2:
398: addi.b #48, d4
399: cmpi.b #57, d4
400: bls.S ConvertNoHex
401: addi.b #7, d4
402: btst #0, d3
403: bne.S ConvertNoHex
404: addi.b #32, d4
405: ConvertNoHex:
406: move.b d4, (a0)+
407: cmpi.l #1, d1
408: beq.S ConvertEnd

```

```

409: move.l d1,d4
410: move.l d2,d5
411: bsr.S ConvertDivide
412: move.l d4,d1
413: bra.S ConvertLoop1
414: ConvertEnd:
415: movem.l (sp)+,d0-d5
416: rts
417:
418: ConvertDivide:
419: movem.l d0-d3,-(sp)
420: move.l d4,d0
421: move.l d5,-(sp)
422: jsr SPFlt(a6)
423: move.l (sp)+,d1
424: move.l d0,-(sp)
425: move.l d1,d0
426: jsr SPFlt(a6)
427: move.l d0,d1
428: move.l (sp)+,d0
429: move.l d0,-(sp)
430: jsr SPDiv(a6)
431: move.l d1,-(sp)
432: jsr SPFix(a6)
433: jsr SPFlt(a6)
434: move.l (sp)+,d1
435: move.l d0,d2
436: jsr SPMul(a6)
437: move.l d0,d1
438: move.l (sp)+,d0
439: jsr SPSub(a6)
440: jsr SPFix(a6)
441: move.l d0,-(sp)
442: move.l d2,d0
443: jsr SPFix(a6)
444: move.l (sp)+,d5
445: move.l d0,d4
446: movem.l (sp)+,d0-d3
447: rts
448:
449: InitAusgabe:
450: move.l GfxBase,a6
451: move.l 50(a6),CListStart
452: move.l CListStart,a5
453: InitAusgabel:
454: move.l (a5)+,d0
455: btst #16,d0
456: beq.S InitAusgabel
457: swap d0
458: andi.l #$ff,d0
459: cmpi.b #$e4,d0
460: bls.S InitAusgabel
461: subq.l #4,a5
462: move.l a5,CListEnde
463: move.l CListStart,a0
464: sub.l a0,a5
465: move.l a5,d0
466: divu #4,d0
467: addi.l #1,d0
468: move.l d0,CListLaenge
469: subi.l #1,d0
470: subi.l ZeilenZahl,d0
471: move.l #65535,d1
472: divu d0,d1
473: andi.l #$ffff,d1
474: move.w d1,SchieberSchritt
475: rts
476:
477: InitFenster:
478: move.l IntBase,a6
479: lea.l NewWindow,a0
480: jsr OpenWindow(a6)
481: tst.l d0
482: beq.S FensterFehler
483: move.l d0,Window
484: move.l d0,a0
485: move.l 50(a0),RastPort
486: move.l 86(a0),UserPort
487: rts
488: FensterFehler:
489: move.l (sp)+,a0
490: bra.L Ende2
491:

```

```

492: SchliesseWindow:
493: move.l IntBase,a6
494: move.l Window,a0
495: jsr CloseWindow(a6)
496: rts
497:
498: OeffneLibraries:
499: move.l ExecBase,a6
500: lea.l IntName,a1
501: jsr OldOpenLibrary(a6)
502: move.l d0,IntBase
503: lea.l GfxName,a1
504: jsr OldOpenLibrary(a6)
505: move.l d0,GfxBase
506: lea.l MatName,a1
507: jsr OldOpenLibrary(a6)
508: move.l d0,MatBase
509: rts
510:
511: SchliesseLibraries:
512: move.l ExecBase,a6
513: move.l IntBase,a1
514: jsr CloseLibrary(a6)
515: move.l GfxBase,a1
516: jsr CloseLibrary(a6)
517: move.l MatBase,a1
518: jsr CloseLibrary(a6)
519: rts
520:
521: align 4
522: Window: dc.l 0
523: IntBase: dc.l 0
524: MatBase: dc.l 0
525: GfxBase: dc.l 0
526: RastPort: dc.l 0
527: UserPort: dc.l 0
528: CListStart: dc.l 0
529: CListEnde: dc.l 0
530: CListLaenge: dc.l 40
531: NamenFlag: dc.l 1
532: ZeilenZahl: dc.l 15
533: SchieberAktuell: dc.l 0
534: EditierteAdresse: dc.l 0
535:
536: IntName: dc.b 'intuition.library',0
537: MatName: dc.b 'mathffp.library',0
538: GfxName: dc.b 'graphics.library',0
539: WinName: dc.b 'The Copper Monitor (C) 1988 by KICK-
START",0
540: In1Text: dc.b 'Copper Information Service",0
541: In2Text: dc.b 'Anfangsadresse der Copperliste: $"
542: In2Data: dc.b '000000",0
543: In3Text: dc.b 'Endadresse der Copperliste: $"
544: In3Data: dc.b '000000",0
545: In4Text: dc.b 'Anzahl der Befehle in der Copperliste:
"
546: In4Data: dc.b '0000",0
547: In5Text: dc.b 'Befehlswort 1",0
548: In6Text: dc.b 'Befehlswort 2",0
549: Bo2Text: dc.b 'O.K.",0
550:
551: even
552: St1Puff: dc.b '0000",0,0
553: Un1Puff: dc.b '0000",0,0
554: St2Puff: dc.b '0000",0,0
555: Un2Puff: dc.b '0000",0,0,0
556: ColText: dc.b 32
557: ColName: dc.b 'WAIT Y='
558: VerText: dc.b ' 00, X='
559: HorText: dc.b ' 00, BFD='
560: BFDText: dc.b '1, VMask='
561: VerMask: dc.b ' 00, HMask='
562: HorMask: dc.b ' 00"
563: ColEnd:
564:
565: even
566: Co2Text: dc.b ' MOVE # $"
567: DatText: dc.b '0000 into $DFE"
568: RegText: dc.b '000",32,32,32,32,32,32
569: Re2Text: blk.b 16,32
570: Co2End:
571:
572: even

```

```

573: RegName:
574: dc.b " BLTCON0 BLTCON1 BLTAFWM BLTALWM BLTCPTH BLTCPTL BLTBPTH"
575: dc.b " BLTBPTL BLTAPATH BLTAPTL BLTDPATH BLTDPTL BLTSIZE UNUSED"
576: dc.b " UNUSED UNUSED BLTCMOD BLTBMOD BLTAMOD BLTDMOD UNUSED"
577: dc.b " UNUSED UNUSED UNUSED BLTCDAT BLTBDAT BLTADAT UNUSED"
578: dc.b " UNUSED UNUSED UNSEDED DSKSYNC COP1LCH COP1LCL COP2LCH"
579: dc.b " COP2LCL COPJMP1 COPJMP2 COPINS DIWSTRT DIWSTOP DDFSTRT"
580: dc.b " DDFSTOP DMACON CLXCON INTENA INTREQ ADKCON"
581: dc.b " AUD0LCH AUD0LCL AUD0LEN AUD0PER AUD0VOL AUD0DAT UNUSED"
582: dc.b " UNUSED AUD1LCH AUD1LCL AUD1LEN AUD1PER AUD1VOL AUD1DAT"
583: dc.b " UNUSED AUD2LCH AUD2LCL AUD2LEN AUD2PER AUD2VOL"
584: dc.b " AUD2DAT UNSEDED AUD3LCH AUD3LCL AUD3LEN AUD3PER"
585: dc.b " AUD3VOL AUD3DAT UNSEDED UNSEDED BPL1PTH BPL1PTL BPL2PTH"
586: dc.b " BPL2PTL BPL3PTH BPL3PTL BPL4PTH BPL4PTL BPL5PTH BPL5PTL"
587: dc.b " BPL6PTH BPL6PTL UNSEDED UNSEDED UNSEDED UNSEDED BPLCON0"
588: dc.b " BPLCON1 BPLCON2 UNSEDED BPL1MOD BPL2MOD UNSEDED UNSEDED"
589: dc.b " BPL1DAT BPL2DAT BPL3DAT BPL4DAT BPL5DAT BPL6DAT UNSEDED"
590: dc.b " UNSEDED SPR0PTH SPR0PTL SPR1PTH SPR1PTL SPR2PTH SPR2PTL"
591: dc.b " SPR3PTH SPR3PTL SPR4PTH SPR4PTL SPR5PTH SPR5PTL SPR6PTH"
592: dc.b " SPR6PTL SPR7PTH SPR7PTL SPR0POS SPR0CTL SP0DATA SP0DATB"
593: dc.b " SPR1POS SPR1CTL SP1DATA SP1DATB SPR2POS SPR2CTL SP2DATA"
594: dc.b " SP2DATB SPR3POS SPR3CTL SP3DATA SP3DATB SPR4POS SPR4CTL"
595: dc.b " SP4DATA SP4DATB SPR5POS SPR5CTL SP5DATA SP5DATB SPR6POS"
596: dc.b " SPR6CTL SP6DATA SP6DATB SPR7POS SPR7CTL SP7DATA SP7DATB"
597: dc.b " COLOR00 COLOR01 COLOR02 COLOR03 COLOR04 COLOR05 COLOR06"
598: dc.b " COLOR07 COLOR08 COLOR09 COLOR10 COLOR11 COLOR12 COLOR13"
599: dc.b " COLOR14 COLOR15 COLOR16 COLOR17 COLOR18 COLOR19 COLOR20"
600: dc.b " COLOR21 COLOR22 COLOR23 COLOR24 COLOR25 COLOR26 COLOR27"
601: dc.b " COLOR28 COLOR29 COLOR30 COLOR31"
602: NullName: blk.b 8,32
603:
604: NewWindow:
605: dc.w 3,11,408,156,$0102
606: dc.l $228,$1100e,SchieberGadget,0,WinName,0,0,0,0
607: dc.w 1
608: SchieberGadget:
609: dc.l BoolGadget1
610: dc.w -16,10,17,-21,87,2,3
611: dc.l SchieberImage,0,0,0,SchieberSpecial,$1000
612: dc.w 0
613: SchieberImage:
614: dc.w 0,0,9,16,1
615: dc.l SchieberData,$02000000
616: dc.w 0
617: SchieberData:
618: dc.w $ff80
619: blk.w 14,$c180
620: dc.w $ff80
621: SchieberSpecial:
622: dc.w 4,0
623: SchieberPos:
624: dc.w 0,0
625: SchieberSchritt:
626: dc.w $666,0,0,0,0,0,0
627: BoolGadget1:
628: dc.w 0,0,-16,-10,16,10,158,$102,1
629: dc.l BoolGadget1Image1,BoolGadget1Image2,0,0,0,$20000
630: dc.w 0
631: BoolGadget1Image1:
632: dc.w 0,0,15,10,1
633: dc.l Bool1Image1Data,$01020000
634: dc.w 0
635: Bool1Image1Data:
636: dc.w $7fff
637: blk.w 9,$6003
638: BoolGadget1Image2:
639: dc.w 0,0,15,10,1
640: dc.l Bool1Image2Data,$01030000

```

```

641: dc.w 0
642: Bool1Image2Data:
643: dc.w $7fff
644: dc.w $6003,$6633,$6733,$67b3,$66f3,$6673,
    $6633,$6003
645: BoolGadget2:
646: dc.w 0,0,324,128,32,8,0,2,1
647: dc.l Bool2Border,0,Bool2Text,0,0,$50000
648: Bool2Border:
649: dc.l 0,$02020005,Bool2BorderData,0
650: Bool2BorderData:
651: dc.w -2,-2,34,-2,34,9,-2,9,-2,-2
652: Bool2Text:
653: dc.l $02030100,0,0,Bo2Text,0
654: EditRequester1:
655: dc.w 0,0,4,11,384,144,0,0
656: dc.l StringGadget1,0,InfoText1,
    $00000100,0,0,0
657: InfoText1:
658: dc.b 2,0,0,0
659: dc.w 94,6
660: dc.l 0,In1Text,InfoText2
661: InfoText2:
662: dc.b 2,0,0,0
663: dc.w 24,26
664: dc.l 0,In2Text,InfoText3
665: InfoText3:
666: dc.b 2,0,0,0
667: dc.w 24,37
668: dc.l 0,In3Text,InfoText4
669: InfoText4:
670: dc.b 2,0,0,0
671: dc.w 17,58
672: dc.l 0,In4Text,InfoText5
673: InfoText5:
674: dc.b 3,2,1,0
675: dc.w 56,88
676: dc.l 0,In5Text,InfoText6
677: InfoText6:
678: dc.b 3,2,1,0
679: dc.w 224,88
680: dc.l 0,In6Text,0
681: StringGadget1:
682: dc.l StringGadget2
683: dc.w 84,102,48,8,0,$202,4
684: dc.l
BorderGadget1,0,0,0,StringSpecial1,$30000
685: dc.w 0
686: BorderGadget1:
687: dc.l 0,$03030005,Border1Data,0
688: Border1Data:
689: dc.w -2,-2,50,-2,50,9,-2,9,-2,-2
690: StringSpecial1:
691: dc.l St1Puff,Un1Puff,5,0,0,0,0,0,0
692: StringGadget2:
693: dc.l BoolGadget2
694: dc.w 252,102,48,8,0,$202,4
695: dc.l
BorderGadget2,0,0,0,StringSpecial2,$40000
696: dc.w 0
697: BorderGadget2:
698: dc.l 0,$03030005,Border2Data,0
699: Border2Data:
700: dc.w -2,-2,50,-2,50,9,-2,9,-2,-2
701: StringSpecial2:
702: dc.l St2Puff,Un2Puff,5,0,0,0,0,0,0

```

Der Coppermonitor in voller Pracht

ENDE



KICKS FÜR INSIDER

Als der nebenstehende Herr vom Programm IMAGE-MAKER hörte, versuchte er es gleich in die Tat umzusetzen. Doch wie Sie sehen können, nahm er es zu persönlich und bezog es lediglich auf sein eigenes IMAGE. Wir hingegen kümmern uns um das des AMIGA, obwohl er dies eigentlich nicht nötig haben sollte. Lesen Sie dies und anderes in den diesmonatlichen KICKS FÜR INSIDER.

INHALT

DER IMAGE- MAKER

DeLuxePaint-Brushes in Images
und Sprites konvertieren

SCHRUMPFE CLI

Das CLI im Kleinformat

STILBRUCH

Steuercodes im CLI

PLOTT ABER FLOTT

Schnelle Punkte in Assembler

IMAGE MAKER

*Deluxe - Paint - Brushes
in Images und Sprites konvertieren*



Als C-Programmierer wünscht man sich oft, seine Programme grafisch zu verschönern. Dazu stehen vielfältige Grafikroutinen aus der "graphics.library" des Betriebssystems zur Verfügung, mit denen sich auch schon so einiges bewerkstelligen läßt. Für detailreiche Grafiken, wie beispielsweise bunte, verschnörkelte Gadgets, Grafiken in Menüs oder Grafik für Spiele ist diese Art der Grafikerzeugung jedoch weniger geeignet.

Hier wünscht man sich eher kompakte Grafikobjekte, die einfach zu handhaben sind. Darüber hinaus verlangen viele Strukturen wie z.B. für Gadgets und Menüs nach sogenannten Images. Diese werden über einen Pointer mit der betreffenden Struktur verbunden oder ganz einfach mit dem Befehl DrawImage() auf den Bildschirm gebracht, womit sich sogar einfache Animationseffekte erreichen lassen.

Doch woher bekommt man die Daten für ein Image? Anbieten würden sich hier Bilder beliebiger Größe, die mit z.B. DeLuxe-Paint gemalt und als Brush gespeichert werden. Diese Grafiken könnte man von seinem Programm aus einlesen und auf dem Bildschirm darstellen oder sonstwie weiterverwenden. Dabei wird man jedoch auf das Problem stoßen, daß diese Brushes in gepackter, also komprimierter Form abgelegt werden. Auch ist es nicht der eleganteste Weg, z.B. ein paar Dutzend Grafiken bei jedem Programmstart neu einzulesen. Ein in der Testphase befindliches Programm würde den Programmierer vermutlich langsam in den Wahnsinn treiben, wenn jedesmal erst einige Bilder geladen werden müssen. Im Sinne kompakter, idealerweise einteiliger Programme sollte man daher nicht allzulange Datensequenzen in sein Programm einbinden, z.B. als Grafikdaten-Array im Source-Code oder als Datenobjekt für den Linker, was wertvolle Compilerzeit spart. Um nun an die gewünschten Daten zu gelangen, bietet sich das nebenstehende Programm an, das Brushes in C-Images und wahlweise sogar in Sprite-Strukturen konvertiert und auf Diskette ablegt. Das mit dem Aztec-Compiler V3.6 erstellte Programm wird durch folgende Zeilen kompiliert und gelinkt:

```
cc ImageMaker.c
ln ImageMaker.o -lc
```

Will man einen Brush namens "Test-Brush", der sich auf einer Diskette im internen Laufwerk befindet, in ein Image konvertieren, so lautet der Programmaufruf:

```
ImageMaker df0:TestBrush
```

Daraufhin wird auf der Diskette ein Sourcefile mit dem Namen "TestBrush.c" erzeugt, das zur weiteren Verwendung zur Verfügung steht. Das Datenfile kann mit dem Befehl TYPE betrachtet werden und hat folgenden Inhalt:

```
1: /*****\
2: * >>> IMAGEMAKER V1.2 <<< *
3: * - - - - - *
4: * (C) 1988 by KICKSTART *
5: * written by Holger Schemel *
6: \*****/
7:
8: #include <stdio.h>
9: #include <exec/types.h>
10: #include <exec/memory.h>
11: extern void *AllocMem();
12:
13: struct BMHD
14: {
15:     UWORD Width,Height;
16:     WORD LeftEdge,TopEdge;
17:     UBYTE Depth;
18:     UBYTE Mask;
19:     UBYTE Compression;
20:     UBYTE pad1;
21:     UWORD transparentColor;
22:     UBYTE xAspect,yAspect;
23:     WORD pageWidth,pageHeight;
24: } BMHeader;
25:
26: struct Image
27: {
28:     SHORT LeftEdge,TopEdge;
29:     SHORT Width,Height,Depth;
30:     SHORT *ImageData;
31:     UBYTE PlanePick,PlaneOnOff;
32:     struct Image *NextImage;
33: } Brush;
34:
35: BYTE imagename[40],imagefile[40];
36: UBYTE *imagedata=0;
37: UWORD *data;
38: LONG i,j;
39: ULONG mem;
40: FILE *fp;
41:
42: main(argc,argv)
43: SHORT argc;
44: BYTE *argv[];
45: {
46:     SHORT x,y,z;
47:     BOOL sprite = FALSE;
48:
49:     if (argc<2 || argc>3 || *argv[1]!='?')
50:         exit(puts("Usage: ImageMaker <brushname>
51:                 [-s (for sprite; default: image)]"));
52:     if (argc>2) for(i=1;i<argc;i++) if (!strcmp(argv[i],"-s"))
53:         sprite=TRUE;
54:     if (!strcmp(argv[1],"-s")) argv[1]=argv[2];
55:     for(i=strlen(argv[1])-1;i>=0;i--)
56:         if (*(argv[1]+i)=='.' || *(argv[1]+i)=='/') break;
57:     sprintf(imagename,"%s",argv[1]+i+1);
58:
59:     printf("Loading Brush %s...",argv[1]);
60:     LoadBrush(argv[1]); /* Brush laden */
61:     Brush.ImageData = (SHORT *)imagedata;
62:     Brush.Width = BMHeader.Width;
63:     Brush.Height = BMHeader.Height;
64:     Brush.Depth = BMHeader.Depth;
65:     Brush.PlanePick = (1<<BMHeader.Depth)-1;
66:
67:     data = (UWORD *)imagedata;
68:     x = Brush.Width/16+(Brush.Width%16 > 0);
69:     y = Brush.Height;
70:     z = Brush.Depth;
71:
72:     if (sprite && (Brush.Depth!=2 || Brush.Width!=16))
73:     {
74:         puts("\nNot a sprite-sized brush! (Width<16 or Depth<2!)");
75:         FreeMem(imagedata,mem);
76:         exit();
77:     }
78:
79:     printf("\nConverting to file %s.c ",&imagename[0]);
80:     if (sprite) puts("(sprite-structure)");
```

```

81:     else        puts("(image-structure)");
82:
83:
84:     sprintf(imagefile,"%s.c",argv[1]);
85:     if(! (fp = fopen(imagefile,"w")))
86:         exit (printf("Can't open file '%s' for write.\n",argv[2]));
87:
88:     if (sprite)
89:     {
90:         fprintf(fp,"struct SpriteImage *%s_data =\n{\n 0,0,\n"
91:             ,imagename);
92:         for(j=0;j<y;j++) fprintf(fp," 0x%04x, 0x%04x,\n",
93:             data[j],data[y+j]);
94:         fprintf(fp," 0,0\n};\n\n");
95:         fprintf(fp,"struct SimpleSprite %s =\n{\n",imagename);
96:         fprintf(fp,"  %s_data,\n  %d,\n  0,0,\n",imagename,y);
97:         fprintf(fp,"  2\n};\n\n");
98:     }
99:     else
100:    {
101:        fprintf(fp,"USHORT %s_data[] =\n{\n",imagename);
102:        for(j=0;j<z;j++)
103:        {
104:            for(i=0;i<x*y;i++)
105:            {
106:                fprintf(fp," 0x%04x,",data[i+j*x*y]);
107:                if (i+1<x*y) fprintf(fp," 0x%04x,",data[++i+j*x*y]);
108:                fprintf(fp,"\n");
109:            }
110:            if (j+1<z) fprintf(fp,"\n");
111:        }
112:        fprintf(fp,"};\n\n");
113:        fprintf(fp,"struct Image %s =\n{\n",imagename);
114:        fprintf(fp,"  0,0,\n");
115:        fprintf(fp,"  %d,%d,%d,\n",Brush.Width,Brush.Height,
116:            Brush.Depth);
117:        fprintf(fp,"  %s_data[0],\n",imagename);
118:        fprintf(fp,"  %d,0,\n  NULL\n};\n",Brush.PlanePick);
119:    }
120:    fclose(fp);
121:    FreeMem(imagedata,mem); /* Freigeben des res. Speichers */
122: }
123:
124: LoadBrush(name) /* IFF-Laderoutine */
125: char *name;
126: {
127:     BYTE c2;
128:     UBYTE *ptr;
129:     SHORT c,ct,ct2;
130:     UWORD BWidth,BHeight,BDepth;
131:
132:     if (!(fp = fopen(name,"r")))
133:         exit(printf("Can't read '%s'.\n",name));
134:
135:     if (!SearchHeader(fp,"ILBMMHD")) /* IFF-File? */
136:         exit(fclose(fp),printf("Not an IFF-File.\n"));
137:
138:     ptr = (UBYTE *)&BMHeader;
139:     for(i=0;i<20;i++) *(ptr++)=getc(fp);
140:     BWidth = ((BMHeader.Width+15)/16)*2;
141:
142:     if (!(imagedata = AllocMem(mem=(ULONG)
143:         /* Speicher reservieren */
144:         (BWidth*BMHeader.Height*BMHeader.Depth),MEMF_CHIP)))
145:         exit(fclose(fp),printf("Not enough CHIP-RAM!\n"));
146:
147:     if (!SearchHeader(fp,"BODY"))
148:         /* Grafikdatenbeginn suchen */
149:         exit(fclose(fp),
150:             FreeMem(imagedata,mem),
151:             printf("Not an IFF-File.\n"));
152:
153:     for(BHeight=0;BHeight<BMHeader.Height;BHeight++) /* Zeilen*/
154:     {
155:         for(BDepth=0;BDepth<BMHeader.Depth;BDepth++) /* Bitplanes */
156:         {
157:             ptr = imagedata+BDepth*BWidth*BMHeader.Height+BHeight*BWidth;
158:             ct2=0;
159:             if (!BMHeader.Compression) /* Brushdaten gepackt? */
160:                 while(ct2++<BWidth) *ptr++=getc(fp);
161:             else /* wenn ja: Entpacken */
162:                 {

```

```

SHORT TestBrush_data[] =
{
  0x0000, 0x0000,
  ...
};

struct Image TestBrush =
{
  0,0,
  <Breite>,<Höhe>,<Tiefe>,
  &TestBrush_data[0],
  <PlanePick>,<PlaneOnOff>,
  NULL
};

```

Ist hingegen die Umwandlung des Brushes in ein Sprite gewünscht, so wird die Option "-s" hinzugefügt. Der Aufruf würde lauten:

```
ImageMaker -s TestBrush
```

Hier erhält man folgende Strukturen:

```

struct
SpriteImage *TestBrush_data =
{
  0,0,
  0x0000,0x0000,
  ...
};

struct SimpleSprite TestBrush =
{
  TestBrush_data,
  <Height>,
  0,0,
  2
};

```

Dieses File kann einfach durch eine Include-Anweisung, z.B.

```
#include "TestBrush.c"
```

in den Quelltext eingebunden werden oder durch den Editor-Befehl <ESC+IF>"TestBrush.c"> (gilt nur für "Ed") an beliebiger Stelle eingefügt werden. Da auch eine Image-Struktur erstellt wird, genügt der Aufruf

```
DrawImage(rastport, &TestBrush,
xPosition,yPosition);
```

um das Image (bei geöffneter "intuition.library") z.B. auf einen Custom-Screen zu bringen. Interessant ist die Routine "LoadBrush(brushname)", die einen beliebigen IFF-Brush lädt, bei Bedarf entpackt und in einem durch Alloc-

Mem() bereitgestellten Speicherbereich zur Verfügung stellt. Die Daten können nun ausgelesen und im Image- bzw. Sprite-Format gespeichert werden. Danach wird der Speicherbereich mit FreeMem() wieder freigegeben. Die Routine eignet sich recht gut zur Verwendung in eigenen Programmen, die IFF-Grafiken nachladen. Dazu muß auch die zu Beginn des Programms definierte Struktur "BMHD" übernommen werden, die den Bitmap-Header mit Informationen wie Breite, Höhe, Tiefe und anderen Daten aufnimmt. Die Routine SearchHeader() untersucht das IFF-File nach Kennungen wie "ILBM" oder "BODY" für den Datenbeginn. Mit diesem Programm sollte es eigentlich jedem möglich sein, Programme durch Grafiken optisch aufzubessern, ohne Unmengen von Daten nachladen zu müssen.

ENDE

```

159:         while(ct2<BWidth)
160:         {
161:             c=getc(fp);
162:             if (c<=128)
163:             {
164:                 ct=c+1;
165:                 for(i=0;i<ct;i++) { *ptr++=getc(fp); ct2++; }
166:             }
167:             else
168:             {
169:                 ct=256-c+1;
170:                 c2=getc(fp);
171:                 for(i=0;i<ct;i++) { *ptr++=c2; ct2++; }
172:             }
173:         }
174:     }
175: }
176: }
177: fclose(fp);
178: }
179: }
180: SearchHeader(fp,iffheader) /* Nach IFF-Header suchen */
181: FILE *fp;
182: UBYTE *iffheader;
183: {
184:     SHORT c;
185:     UBYTE *cp=iffheader;
186:
187:     while((c=getc(fp))!=EOF)
188:     {
189:         if (c==*cp) { cp++; if (!*cp) break; }
190:         else cp=iffheader;
191:     }
192:     if (c==EOF) return(0);
193:     else { for(i=0;i<4;i++) getc(fp); return(1); }
194: }

```

KICKS FÜR INSIDER

Liebe Leser, KICKS für INSIDER ist eine Rubrik, die für interessante und lehrreiche Programme geschaffen wurde, denn was nutzt das schönste Programm, wenn man es nur abtipft, aber nicht verstehen kann.

Um jeden Monat neue KICKS zu veröffentlichen, sind wir natürlich auch von unseren Lesern abhängig, denn irgendwann einmal sind die Einfälle jedes Programmierers erschöpft. Also, nehmen Sie an dieser Rubrik aktiv teil, ergänzen Sie sie mit Ihren eigenen Programmen. Werden Sie Autor in der KICKSTART.

Dabei sind alle auf dem AMIGA existierenden Programmiersprachen gewünscht. Egal, ob Basic, C, Assembler, Modula 2, Pascal oder andere.

WIE DAS GEHT?

Ganz einfach. Schicken Sie uns das Programm, samt dem zugehörigen Artikel, eventuellen Bildern und Hardcopies (nach Möglichkeit "gegrabbt") auf Diskette zu. Wir werden uns dann bei Ihnen melden. Ein interessantes Honorar wartet bei Veröffentlichung natürlich auch auf Sie.

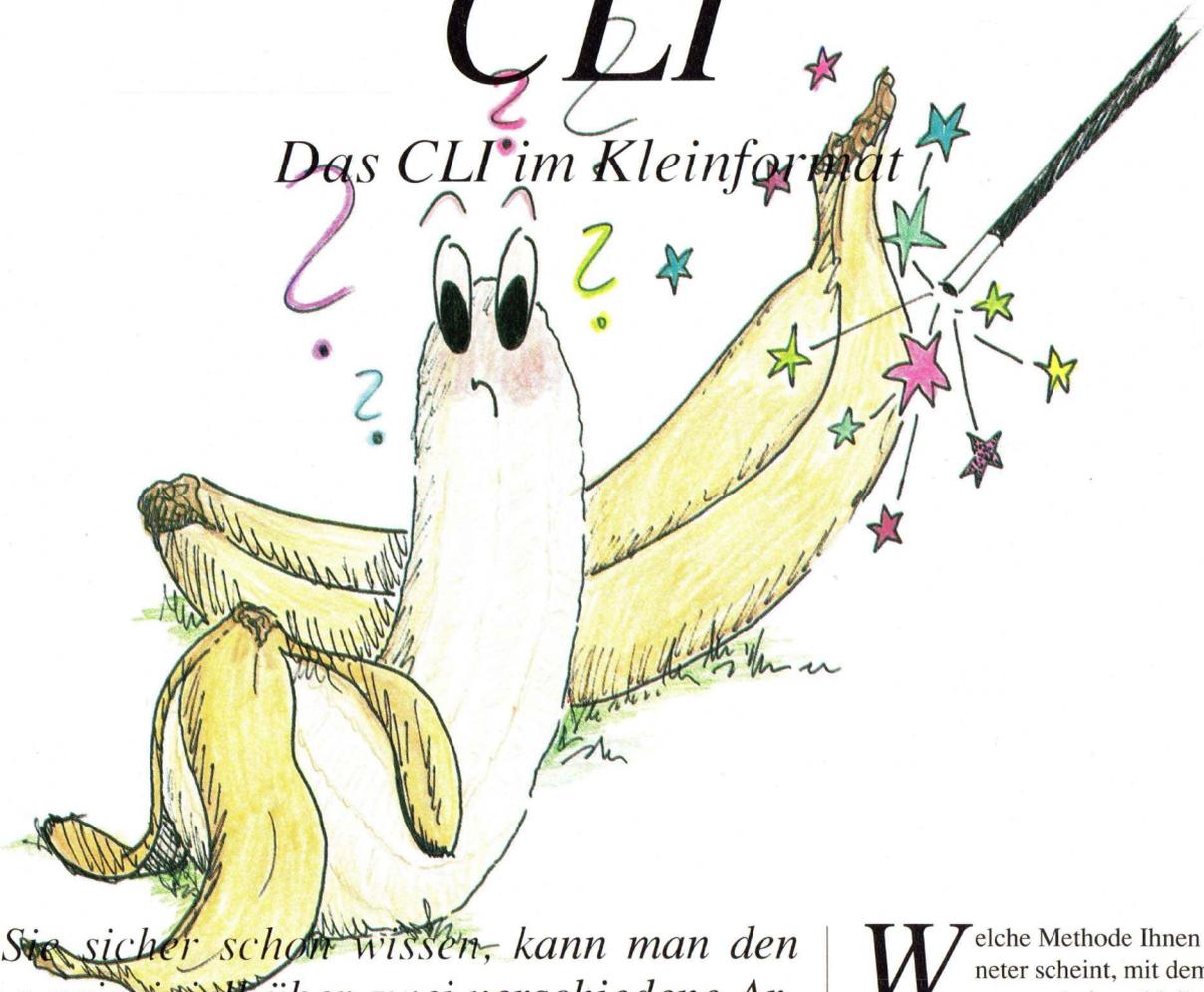
Wenn Sie Fragen haben, rufen Sie doch einfach bei uns an.

Tel. 06196/481811
Mo-Fr. 11 bis 17 Uhr

Ihre KICKSTART-Redaktion

SCHRUMPFFE CLI

Das CLI im Kleinformat



Wie Sie sicher schon wissen, kann man den Amiga prinzipiell über zwei verschiedene Arten bedienen: Mit der Workbench und über das CLI. Die Workbench hat den Vorteil der schnelleren Erlernbarkeit, ist relativ sicher gegenüber versehentlichen Datenlöschungen und sieht auch noch besser aus. Das CLI dagegen ermöglicht tieferen Zugang zum System, ist schneller und flexibler. Der Nachteil ist das nicht einfache hierarchische Dateisystem und die Möglichkeit, mit einem unbedachten Befehl ganze Disketten zu löschen.

Welche Methode Ihnen geeigneter scheint, mit dem Amiga zu arbeiten, bleibt Ihnen überlassen, jedoch scheint eine Kombination beider am besten zu sein. Dieser Kombination steht eigentlich nur die "Startup-Sequence" Ihrer Workbench-Disk im Wege: Ursprünglich beinhaltet diese nämlich den Befehl EndCLI. Um die Funktion der Startup-Sequence zu verstehen, muß man wissen, daß der Amiga normalerweise nach dem Booten einer Diskette im CLI arbeitet. Die Workbench ist ein Programm, welches mit dem CLI-Befehl "LoadWB" geladen wird. Dies geschieht auch in der Startup-Sequence der Original-Workbench. Die Amiga-Programmierer jedoch trauten dem User scheinbar nicht zu, einfach das CLI weiterzubnutzen;

das ursprüngliche CLI-Fenster wird mit dem letzten Befehl der Startup-Sequence "EndCLI" geschlossen, also "Workbench oder Nichts". Läßt man nun (Vorsicht, Trick!) diesen EndCLI-Befehl weg, bleibt das CLI erhalten, aber es liegt gemeinerweise mitten über unserer Workbench, so daß diese erst nach etwas Maus-Akrobatik sichtbar wird. Um Ihnen diese armmuskelermüdende typische Amiga-User-Bewegung zu ersparen, entstand dieses Programm. SchrumpfCLI wird statt EndCLI in die Startup-Sequence eingebaut, so daß beim erneuten Start das CLI-Fenster verkleinert (sogar kleiner, als das mit der Maus möglich ist) und in die linke untere Ecke der Workbench befördert wird, statt geschlossen zu werden. Somit kann die Workbench in ihrer ganzen Pracht benutzt werden, das CLI bleibt Ihnen jedoch für spezielle Operationen (Befehle mit CLI-Parametern) erhalten.

Doch noch mehr...

Das Programm ist jedoch eigentlich nur das "Drumherum" einer Routine, die dem Programmierer vielfältigste Manipulationsmöglichkeiten eröffnet, wie z.B. Menüs und Gadgets im CLI-Fenster und ähnliche Gags, dazu jedoch später. Die Syntax von SchrumpfCLI ist sehr einfach, nämlich

```
SchrumpfCLI [<WindowName>]
```

Der (optionale) Parameter <Window-Name> ist quasi ein Bonus, damit kann man nämlich jedes Fenster, dessen Name korrekt eingegeben wird, genau wie das CLI-Fenster verändern. Wichtig ist jedoch die richtige Schreibweise des Namens; Groß- und Kleinschreibung muß beachtet werden. Damit man nicht jedesmal den Namen des CLI-Fensters eingeben muß (AmigaDOS), ist dieser im Programm voreingestellt, so daß ein einfaches "SchrumpfCLI" das CLI schrumpfen läßt (logisch, oder?).

Zum Programm

Nun aber zum Aufbau des Programms: Es wurde ganz in Maschinencode mit dem Seka-Assembler geschrieben. Tippen Sie das Listing ein, assemblieren Sie es und speichern Sie es mit "WO SchrumpfCLI" ab.

```

1: PWID 136
2: PLEN 90
3: *   SchrumpfCLI V2.0   *
4: *   (C) KICKSTART 1988 *
5: *   Autor: Stefan Kemeny *
6:
7: ExecBase      =      4
8: OldOpenLib    = -30 -378
9: CloseLib      = -30 -384
10:
11: OpenScreen   = -30 -168
12: CloseScreen  = -30 - 36
13: SizeWindow   = -30 -258
14: WindowLimits = -30 -288
15: MoveWindow   = -30 -138
16:
17: Start:
18:  cmpi.l #1,d0           ;Falls Parameter übergeben,
19:  bne.S NameGegeben     ;als Fensternamen übernehmen
20:  move.l #WinName,a0    ;sonst "AmigaDOS" als Vorein-
21:  move.l #9,d0          ;stellung setzen
22: NameGegeben:
23:  subq.l #1,d0
24:  move.l a0,WinPointer  ;Startadresse des Parameters
25:  move.b #0,(a0,d0)    ;Name muss mit \NULL enden
26:  addq.l #1,d0
27:  add.l a0,d0
28:  move.l d0,WinPoinEnd  ;Endadresse des Parameters
29:  bsr.L OpenInt         ;INTUITION muss her...
30:  bsr.L FindeScreenHandle ;Screen-Struktur suchen
31:  bsr.L FindeFensterHandle ;Suche nach Fenster-Struktur
32: Module:                ;Die Struktur ist gefunden !!!
33:  bsr.S FensterGrenzen  ;einige Elemente, gänzlich
34:  bsr.S FensterKleiner  ;ersetzbar, weglassbar und
35:  bsr.S FensterBewegen  ;veränderbar
36:  bsr.L CloseInt        ;Bye, INTUITION
37:  move.l #0,d0          ;Schon Ende, zurück ins DOS
38:  rts
39:
40: FensterGrenzen:        ;Window-Handle ist in
41:  move.l IntBase,a6     ;"WindowHandle:" abgelegt,
42:  move.l WindowHandle,a0 ;dieses Modul kann verändert,
43:  move.l #124,d0        ;werden, da es keinen Wert
44:  move.l #29,d1         ;zurückgibt und weil die Aktion
45:  move.l #640,d2        ;nicht zwingend nötig ist für
46:  move.l #72,d3         ;weitere Module
47:  jsr WindowLimits(a6) ;VERÄNDERT MIN/MAX Größe
48:  rts
49:
50: FensterKleiner:        ;Man sollte bei der
51:  move.l IntBase,a6     ;Programmierung dieser Module
52:  move.l WindowHandle,a0 ;darauf achten, daß kein absoluter
53:  clr.l d0              ;Wert vorausgesetzt wird (z.B. eine
54:  move.w 8(a0),d0       ;bestimme Höhe, man sollte diese
55:  clr.l d1              ;Daten eher aus der Window-Struktur
56:  move.w 10(a0),d1      ;herauslesen, dann geht alles
57:  move.w 16(a0),d2      ;klar
58:  move.w 18(a0),d3
59:  sub.l d2,d0           ;Hier wird auf die Minimalgröße
60:  sub.l d3,d1           ;verkleinert, die aus der
61:  neg.l d0              ;Struktur gelesen wurde
62:  neg.l d1
63:  jsr SizeWindow(a6)
64:  rts
65:
66: FensterBewegen:        ;Dieses Modul bewegt das Fenster
67:  move.l IntBase,a6     ;in die linke untere Ecke.
68:  move.l WindowHandle,a0
69:  clr.l d0              ;Dabei wird die Höhe des Fensters
70:  clr.l d1              ;mit einberechnet, damit es nicht
71:  move.w 4(a0),d0       ;aus dem Screen ragt (und dabei
72:  move.w 6(a0),d1       ;eifrig zum Meditieren anregt...)
73:  move.w #$e3,d2
74:  sub.l d2,d1
75:  neg.l d0              ;Fügen Sie ruhig eigene Module
76:  neg.l d1              ;ein, vielleicht ein ZOOM-Gadget ??
77:  jsr MoveWindow(a6)
78:  rts
79:
80: FindeScreenHandle:     ;Hauptroutine #01, ermittelt
81:  move.l IntBase,a6     ;die Handle des Workbench-Screens aus
82:  lea DummyScreen,a0   ;einem Dummy-Screen, der so gut wie
83:  jsr OpenScreen(a6)   ;keinen Speicher braucht und eh

```

```

84:  move.l d0,a0          ;sofort geschlossen wird...
85:  move.l (a0),-(sp)
86:  jsr CloseScreen(a6)
87:  move.l (sp)+,a0
88:  NextScreen:          ;Falls mehrere Screens existieren,
89:  cmpi.l #0,a0         ;eine kleine Sicherung:
90:  bne.S ScreenVorhanden ;Jeder Screen wird mit Namen
91:  move.l (sp)+,a0      ;überprüft,
92:  bsr.l CloseInt       ;falls der Workbench-Screen
93:  move.l #60,d0        ;nicht gefunden wird,
94:  rts                  ;springt das Programm mit
95:  ScreenVorhanden:    ;Error #60 zurück
96:  move.l $16(a0),a1    ;(Irgendwie paradox, wo kam denn
97:  lea.l ScrName,a2    ; dann der CLI-Befehl
98:  ScreenLoop1:        ; SchrumpfCLI her ???)
99:  move.l #ScrEnd,d1
100: cmp.l d1,a2
101: beq.S ScreenFound   ;Augenmerk auf die VERGLEICH-
102: move.b (a1)+,d1     ;Routine richten, es muß
103: move.b (a2)+,d2     ;byteweise verglichen werden,
104: cmp.b d1,d2         ;da eine ungerade Adresse
105: beq.S ScreenLoop1   ;den Guru holt...
106: move.l $0(a0),a0
107: bra.S NextScreen
108: ScreenFound:
109: move.l a0,ScreenHandle
110: rts
111:
112: FindeFensterHandle:
113: move.l ScreenHandle,a0 ;Siehe Oben
114: move.l $4(a0),a0      ;(Dies ist prinzipiell die
115: NextWindow:          ; selbe Routine wie für
116: cmpi.l #0,a0         ; Screens, nur wird diesmal
117: bne.S WindowVorhanden ; von der ermittelten
118: move.l (sp)+,a0      ; Screen-Handle ausgegangen,
119: bsr.S CloseInt       ; nicht von der eines
120: move.l #80,d0        ; DUMMY-Screens !)
121: rts
122: WindowVorhanden:
123: move.l $20(a0),a1
124: move.l WinPointer,a2
125: WindowLoop1:
126: move.l WinPoinEnd,d1
127: cmp.l d1,a2
128: beq.S WindowFound
129: move.b (a1)+,d1
130: move.b (a2)+,d2
131: cmp.b d1,d2
132: beq.S WindowLoop1
133: move.l $0(a0),a0
134: bra.S NextWindow
135: WindowFound:
136: move.l a0,WindowHandle
137: rts
138:
139: OpenInt:             ;Ohne INTUITION läuft
140: move.l ExecBase,a6   ;hier schon gar nichts
141: lea.l IntName,a1
142: jsr OldOpenLib(a6)
143: move.l d0,IntBase
144: tst.l d0
145: beq.S KeinINTUITION
146: rts
147: KeinINTUITION:      ;Nun, schade .
148: move.l (sp)+,d0     ;Zurück ins DOS
149: rts
150:
151: CloseInt:           ;INTUITION ade
152: move.l ExecBase,a6
153: move.l IntBase,a1
154: jsr CloseLib(a6)
155: rts
156:
157: align 4              ;Es folgen Daten
158: IntBase:             dc.l 0
159: ScreenHandle:       dc.l 0
160: WindowHandle:       dc.l 0
161: WinPointer:         dc.l 0
162: WinPoinEnd:         dc.l 0
163: IntName:             dc.b 'intuition.library',0
164: ScrName:            dc.b 'Workbench Screen',0
165: ScrEnd:             ;Hier der Name des Screens, wo gesucht werden soll
166: WinName:            dc.b 'AmigaDOS',0

```

Zur Funktion: Das Kernstück des Programms ist eine Routine, welche die Window-Struktur eines beliebigen, nur durch einen Namen bekannten Windows liefert. Diese Struktur wird von INTUITION auf jeden Fall benötigt, falls das Fenster irgendwie verändert werden soll (Menüs, Gadgets, Position,...).

Normalerweise erhält man diese Struktur beim Öffnen eines eigenen Fensters im Register d0 zurück, aber hier ist das Fenster schon offen. Sicher ist diese Adresse irgendwo im Speicher, aber bei 524288 Bytes oder mehr suchen Sie schon eine Weile, vor allem wenn Sie nicht wissen, wonach... Die Adresse muß also mit einem Trick ermittelt werden: Wir müssen erst die Screen-Struktur der Workbench finden. Dazu öffnet man einen kleinen unsichtbaren Screen und liest das 1. Langwort aus. Dies zeigt auf die nächste Screen-Struktur. Nun wird mit dem Offset \$16 der Name des Screens gelesen und mit "Workbench Screen" verglichen. Falls alles stimmt, haben wir die Workbench-Screen-Struktur. Genau nach diesem Prinzip arbeitet die zweite Routine, die nach einem Window mit einem bestimmten Namen sucht, nun aber in der Workbench-Screen-Struktur. Wird dieser Name nicht gefunden, springt das Programm ins CLI mit Error 80 zurück. Wenn das Fenster gefunden wurde, erhält man die Struktur wieder, die man sucht! Hat man diese gefunden, kann man einige Operationen am Fenster durchführen. Kenner von INTUITION werden im Programm ab dem Label "Module:" drei Unterprogramme sehen, die

1. die WindowLimits neu setzen
2. das Fenster auf die Minimum-Maße verkleinern und
3. das Fenster soweit wie möglich nach links unten bewegen.

Man kann hier aber alle möglichen eigenen Operationen einbauen, wie z.B. das CLI auf PAL-Größe bringen, oder z.B. ein Schließgadget einbauen... Man sollte aber einiges von INTUITION verstehen, bevor man hier etwas verändert.

```

167: WinEnd: ;Voreingestellter Wert, kann geändert werden
168:
169: even ;Bei Namen ein NULL-Byte am Ende
170: DummyScreen: ;ULTRA-MINI, man
171: dc.w 0,1024,10,10,1 ;braucht ihn ja nur

```

```

172: dc.b 0,1 ;für Bruchteile einer
173: dc.l 15,0,0,0,0 ;Sekunde

```

Listing 1: Das Programm Schrumpf macht seinem Namen alle Ehre

```

10 `df1:schrumpfli Lader (Zeilennummern nicht notwendig)
11 Filename$= "df1:schrumpfli_1"
12 OPEN "O",1,Filename$
13 WHILE Wert<>-1
14 C=C+1
15 READ Wert
16 IF Wert=-1 THEN
17 END
18 END IF
19 IF (C MOD 11)=0 THEN
20 IF Wert<>(Summe AND 255) THEN
21 PRINT"Fehler in Datazeile "C/11
22 END
23 END IF
24 Summe=0
25 LOCATE 5,5: PRINT"Zeile: "C/11
26 ELSE
27 PRINT #1, CHR$(Wert);
28 Summe=Summe+(Wert+C MOD 11)^2
29 END IF
30 WEND
31 CLOSE(1)
100 DATA 0,0,3,243,0,0,0,0,0,0,0,221
101 DATA 0,2,0,0,0,0,0,0,0,1,162
102 DATA 0,0,0,129,0,0,0,1,0,0,155
103 DATA 3,233,0,0,0,129,12,128,0,0,138
104 DATA 0,1,102,12,32,124,0,0,1,215,14
105 DATA 32,60,0,0,0,9,83,128,35,200,16
106 DATA 0,0,1,172,17,188,0,0,8,0,243
107 DATA 82,128,208,136,35,192,0,0,1,176,195
108 DATA 97,0,1,62,97,0,0,152,97,0,221
109 DATA 0,238,97,16,97,56,97,100,97,0,97
110 DATA 1,76,32,60,0,0,0,0,78,117,209
111 DATA 44,121,0,0,1,160,32,121,0,0,138
112 DATA 1,168,32,60,0,0,0,0,124,34,60,140
113 DATA 0,0,0,29,36,60,0,0,2,128,50
114 DATA 38,60,0,0,0,72,78,174,254,194,73
115 DATA 78,117,44,121,0,0,1,160,32,121,179
116 DATA 0,0,1,168,66,128,48,40,0,8,224
117 DATA 66,129,50,40,0,10,52,40,0,16,194
118 DATA 54,40,0,18,144,130,146,131,68,128,18
119 DATA 68,129,78,174,254,224,78,117,44,121,248
120 DATA 0,0,1,160,32,121,0,0,1,168,24
121 DATA 66,128,66,129,48,40,0,4,50,40,154
122 DATA 0,6,52,60,0,227,146,130,68,128,158
123 DATA 68,129,78,174,255,88,78,117,44,121,223

```

```

124 DATA 0,0,1,160,65,249,0,0,1,224,3
125 DATA 78,174,255,58,32,64,47,16,78,174,91
126 DATA 255,190,32,95,177,252,0,0,0,0,160
127 DATA 102,14,32,95,97,0,0,156,32,60,65
128 DATA 0,0,0,60,78,117,34,104,0,22,242
129 DATA 69,249,0,0,1,198,34,60,0,0,96
130 DATA 1,215,181,193,103,14,18,25,20,26,249
131 DATA 180,1,103,238,32,104,0,0,96,200,13
132 DATA 35,200,0,0,1,164,78,117,32,121,113
133 DATA 0,0,1,164,32,104,0,4,177,252,251
134 DATA 0,0,0,102,12,32,95,97,82,113
135 DATA 32,60,0,0,0,80,78,117,34,104,154
136 DATA 0,32,36,121,0,0,1,172,34,57,78
137 DATA 0,0,1,176,181,193,103,14,18,25,68
138 DATA 20,26,180,1,103,238,32,104,0,0,121
139 DATA 96,202,35,200,0,0,1,168,78,117,148
140 DATA 44,121,0,0,0,4,67,249,0,0,178
141 DATA 1,180,78,174,254,104,35,192,0,0,243
142 DATA 1,160,74,128,103,2,78,117,32,31,245
143 DATA 78,117,44,121,0,0,0,4,34,121,156
144 DATA 0,0,1,160,78,174,254,98,78,117,45
145 DATA 255,255,0,0,0,0,0,0,0,125
146 DATA 0,0,0,0,0,0,0,0,0,129
147 DATA 0,0,105,110,116,117,105,116,105,111,168
148 DATA 110,46,108,105,98,114,97,114,121,0,40
149 DATA 87,111,114,107,98,101,110,99,104,32,220
150 DATA 83,99,114,101,101,110,0,65,109,105,118
151 DATA 103,97,68,79,83,0,0,0,4,0,5
152 DATA 0,10,0,10,0,1,0,1,0,0,223
153 DATA 0,15,0,0,0,0,0,0,0,158
154 DATA 0,0,0,0,0,0,0,0,0,129
155 DATA 0,0,0,0,3,236,0,0,0,21,165
156 DATA 0,0,0,0,0,0,0,10,0,0,133
157 DATA 0,24,0,0,0,40,0,0,0,76,193
158 DATA 0,0,0,82,0,0,0,118,0,0,25
159 DATA 0,124,0,0,0,164,0,0,0,170,109
160 DATA 0,0,0,204,0,0,0,210,0,0,213
161 DATA 1,0,0,0,1,6,0,0,1,30,234
162 DATA 0,0,1,38,0,0,1,72,0,0,43
163 DATA 1,78,0,0,1,102,0,0,1,116,170
164 DATA 0,0,1,126,0,0,1,148,0,0,91
165 DATA 0,0,0,0,3,242,0,0,3,235,24
166 DATA 0,0,0,1,0,0,3,242,236
9999 Data -1

```

Listing 2: Der Basic-Lader hilft, das Programm auch ohne Assembler einzutippen.

ENDE

GET IT

AIT-USER-GROUP

Über 1000 Amiga-Disketten im Pool. Eine der größten Public-Domain-Sammlungen Europas.
Fish, Faug, Amicus, Panorama, Auge 4000, AIT Special (Tornado), Taifun, Casa, UKaug, AMIGAZine, Amiga-Juice, Chiron Conceptions, AIT, ACS, RPD (Kanada), BCS, Tiger, SACC, Demos, Slideshows, Entertain, Tutorials, Ray-tracer, DBW-Render 2.0, SCA-Virus-Protector, Virus-Beschreibung, Utilities und...

Jede Diskette 5,- DM, ab 10 Stück portofrei! Nachnahme + 3,50 DM.

Beschreibung der Disketten auf 4 Info-Disks = 15,- DM, lauffähig auf allen Amigas.

Berechtigt zum Tausch von 4 zu 1, d. h. 1 Disk von Ihnen gegen 4 von unseren. Wir kopieren auch 5,25-Disks, dabei 1,50 DM günstiger als die 3,5"-Disk.

Neu das PD-Magazin auf Diskette: »GET IT«.

Jeden Monat neu, randvoll für 10,- DM inkl. Porto.

Mit Tips, Kursen, Infos, News, Helps über PD-Disks und Preisausschreiben.

Übrigens: »GET IT«-Abonnenten zahlen nur 4,- DM statt 5,- DM für 1 Disk aus unserem Pool.

AIT M. Rönn, Tel. (050 41) 82 29

AIT

DI DI DIGITIZE

S/W Röhrenkamera 1/2 Zoll C-Objektivanschluss 528,-

Color CCD Kamera 2/3 Zoll C-Objektivanschluss 2398,-

C-Objektive von 3-350mm Macro, Repro- u.a. Adapter

Color-Fotos von Diskette Dias, Vergrößerungen ab 2,50

OPTIVISION

Heckenerstr. 16 D-54169 Winchagen
Tel. 02645/4424

VON THOMAS BECKER

STILBRUCH

Steuercodes im CLI



Change ist ein kleines Utility-Programm für den Amiga, und zwar für den Command-Line-Interpreter. Es ermöglicht dem Anwender, auf komfortable Weise eine Vielzahl von Änderungen im CLI vorzunehmen.

Das Amiga-System besitzt, wie Sie vielleicht wissen, eine ganze Reihe von Optionen, um das Aussehen bzw. das Schriftbild des CLIs zu verändern. Alle Optionen stellt bereits das Betriebssystem zur Verfügung, so daß sie weder eingeladen noch programmiert werden

müssen. Neben den erwähnten Änderungen stehen dem Anwender noch weitere Optionen zur Verfügung, beispielsweise kann der Text hoch- bzw. heruntergescrollt werden.

Das kurze Programm erlaubt dem Anwender, schnell und komfortabel verschiedene Einstellungen zu modifizieren. "Change" bietet folgende Möglichkeiten:

0	= normale Schrift
1	= fette Schrift
2	= kursive Schrift
3	= Unterstreichen der Schrift
4	= inverse Darstellung der Schrift
5	= Hochscrollen des Textes um eine Zeile
6	= Runterscrollen des Textes um eine Zeile
7 8 9 10	= Farbänderung des Vordergrundes
11 12 13 14	= Farbänderung des Hintergrundes

Das Ändern geht denkbar einfach von-statten, hierzu muß lediglich die gewünschte Änderungsnummer nach dem Programmnamen eingegeben werden. Doch zunächst zur Syntax von "Change":

```
change [[Par1] [Par2] ...]
```

Das Programm verlangt nicht nach einem Parameter; wird keiner angegeben, so wird ein Hilfstext angezeigt, der alle möglichen Parameter auflistet und die richtige Syntax. Beachten Sie, daß ein Leerzeichen (Space) zwischen den Parametern stehen muß, da es ansonsten zu fehlerhaften Ausführung kommen kann.

Beispiel:

```
change 2
```

"2" steht für kursive Schrift, nach Drücken der Return-Taste werden Texte kursiv ausgegeben.

"Change" erlaubt des weiteren, auch mehrere Optionen zu verarbeiten; beispielsweise bewirkt der nachstehende Ausdruck

```
change 2 3 4
```

daß der Text kursiv, unterstrichen und invers dargestellt wird. Eine Sonder-

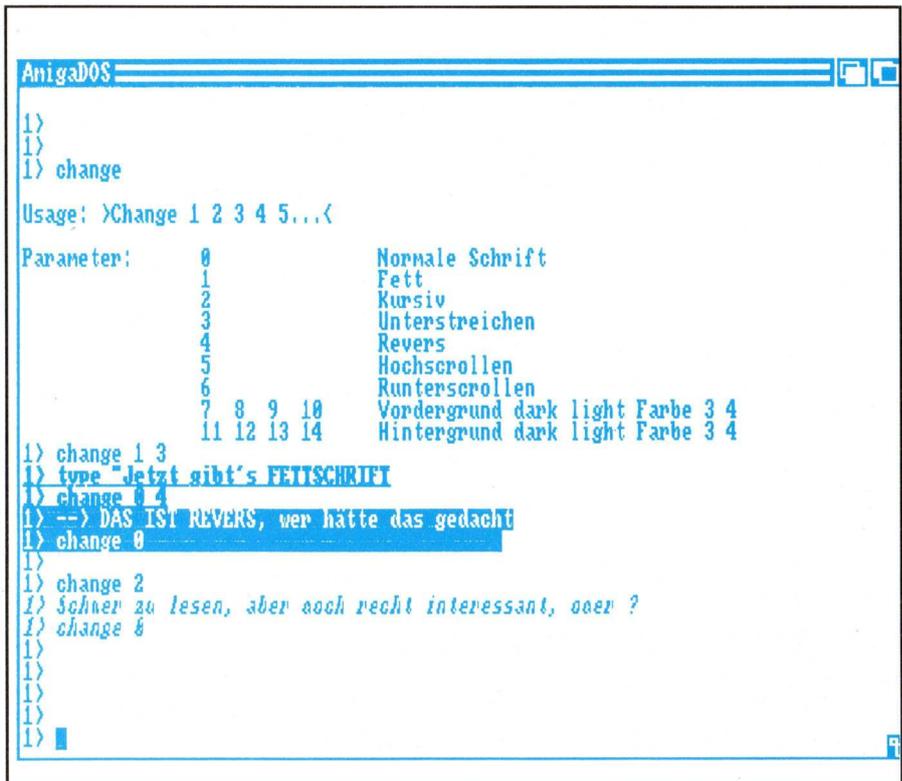
stellung nehmen die Parameter 7 bis 14 ein. Mit ihnen kann die Farbe der Schrift bzw. des Hintergrundes festgelegt werden. Das Programm verwendet hierbei die eingestellten Systemfarben, die bei der Workbench auf vier begrenzt sind.

Selbstverständlich können gleiche Parameter mehrmals hintereinander oder durch andere Parameter getrennt eingegeben werden.

```
change 6 6 6
```

bewirkt ein Herunterscrollen des Textes.

Das Programm kann beispielsweise in der Startup-Sequence Ihrer Boot-Diskette Verwendung finden, um einen eigenen Titelbildschirm zu kreieren, aber auch das einfache Aufrufen vom CLI ist möglich.



Ein kleines Beispiel zeigt, was man mit dem Programm alles anstellen kann.

Die Programmierung ist denkbar einfach; man bedient sich einfach der festgelegten Escape- und ASCII-Sequenzen. Diese werden mit den Nummern der einzelnen Parameter gleichgesetzt, so daß die möglichen Sequenzen einfach aufgerufen werden können.

Das Programm ist in der Programmiersprache C geschrieben und kann sowohl mit dem Lattice-Compiler als auch mit dem Aztec-Compiler kompiliert werden.

Die Compileroptionen für den Aztec C-Compiler:

```
cc change.c ; zum Compilieren
ln change.o -lc ; zum Linken
```

Die Compileroptionen für den Lattice C-Compiler:

```
lc -L change.c
```

Die Option "-L" veranlaßt das anschließende Linken des Programms.

Beim Lattice-Compiler sollte noch ein "return(0);" vor der schließenden main-Klammer eingefügt werden, so daß die Warnung "Function return value mismatch" nicht mehr ausgegeben wird.

```

1:  /* (c) KICKSTART 1988   *
2:  * Autor: Thomas Becker */
3:
4:
5:  #define ANZAHL 15
6:
7:  char *codename[ANZAHL] =
8:  {
9:      "0m", "1m", "3m", "4m", "7m", "1S", "1T",
10:     "30m", "31m", "32m", "33m", "40m", "41m",
11:     "42m", "43m"
12: };
13:  /* beliebig erweiterbar */
14:
15:
16: main(argc, argv)
17: int argc;
18: char *argv[];
19: {
20:     int i, n;
21:     char string[10];
22:     void hilfe();
23:
24:     for(i=1; i<argc; i++)
25:     {
26:         if((n = atoi(argv[i])) >= 0 && n<ANZAHL)
27:         {
28:             *string = '\0';
29:             strcpy(string, "\033[");
30:             strcat(string, codename[n]);
31:             printf("%s", string);
32:         }
33:         else hilfe();
34:     }
35:     if (argc == 1) hilfe();
36:  /* Lattice-Besitzer nächste Zeile miteingeben */
37:  /* return(0); */
38: } /* Ende main */
39:
40: void hilfe()
41: {
42:     printf
43:     ("\nUsage: >Change 1 2 3 4 5...<\n\n");
44:     printf
45:     ("Parameter:\t0\t\tNormale Schrift\n\t\t1\t\tFett\n");
46:     printf
47:     ("\t\t2\t\tKursiv\n\t\t3\t\tUnterstreichen\n");
48:     printf
49:     ("\t\t4\t\tRevers\n\t\t5\t\tHochscrollen\n");
50:     printf
51:     ("\t\t6\t\tRunterscrollen\n");
52:     printf
53:     ("\t\t7  8  9  10\tVordergrund dark light Farbe 3 4\n");
54:     printf
55:     ("\t\t11 12 13 14\tHintergrund dark light Farbe 3 4\n");
56: } /* Ende des Programms */

```



ABO



ABO

Absender
(Bitte deutlich schreiben)

Vorname/Name

Straße/Nr.

PLZ/Ort

Postkarte

Bitte
mit
60 Pf.
frankieren

Heim Verlag

Heidelberger Landstr. 194

6100 Darmstadt-Eberstadt

Telefon 0 61 51 / 5 60 57



Einzelheft- u. Monatsdisketten Bestellung



Einzelheft- u. Disketten Service

Absender
(Bitte deutlich schreiben)

Vorname/Name

Straße/Nr.

PLZ/Ort

Postkarte

Bitte
mit
60 Pf.
frankieren

Heim Verlag

Heidelberger Landstr. 194

6100 Darmstadt-Eberstadt

Telefon 0 61 51 / 5 60 57



Kontaktkarte



Kontaktkarte

Bitte Adresse der Firma, bei der Sie Informationen, oder etwas bestellen möchten, auf der rechten Seite eintragen - - - - -

Absender
(Bitte deutlich schreiben)

Vorname/Name

Beruf

Straße/Nr.

PLZ/Ort

Telefon Vorwahl/Rufnummer

Postkarte

Bitte
freimachen

Firma

Straße/Postfach

PLZ Ort

Ja, bitte senden Sie mir die Amiga-Computer Fachzeitschrift ab _____ für mindestens 1 Jahr (11 Hefte) zum ermäßigten Preis von jährlich DM 70,- frei Haus. (Ausland: Nur gegen Scheck-Voreinsendung DM 90,- Normalpost.) Der Bezugszeitraum verlängert sich nur dann um ein Jahr, wenn nicht 6 Wochen vor Ablauf des Abonnements gekündigt wird.

Gewünschte Zahlungsweise bitte ankreuzen

Bequem und bargeldlos durch Bankeinzug

Name _____

Konto-Nr. _____ BLZ _____

Vorname _____

Institut _____ Ort _____

Straße/Nr. _____

Ein Verrechnungsscheck über DM _____ liegt bei.

PLZ _____ Ort _____

Garantie:
Diese Bestellung kann ich schriftlich innerhalb einer Woche (rechtzeitige Absendung genügt) widerrufen. Dies bestätige ich durch meine 2. Unterschrift.

Datum _____ Unterschrift _____

Datum _____ Unterschrift _____



Einzelheft- u. Monatsdisketten

Bestellung



KICKSTART können Sie direkt beim HEIM-VERLAG zum Einzelheft-Preis von DM 7,- (zuzüglich Gebühr für Porto und Verpackung) nachbestellen. Bearbeitung nur gegen beigefügten Scheck über den entsprechenden Betrag (keine Überweisung).

Jan.	Febr.	März	April	Mai	Juni	Juli/Aug.	Sept.	Okt.	Nov.	Dez.
<input type="checkbox"/>										

1987 = DM

1988 = DM

+ Gebühr für Porto u. Verp. _____

= DM

Scheck in Höhe _____

zus. DM _____

liegt bei

Disketten Service

Alle Programme, die in KICKSTART veröffentlicht wurden, sind auf Disketten erhältlich. Die Disketten enthalten die Programme von jeweils 2 KICKSTART-Ausgaben. Bestellen Sie durch ankreuzen die gewünschten Disketten

Preis je Diskette 19,- DM	Juli/Aug.	Sept./Okt.	Nov./Dez.	Jan./Febr.	März/Apr.
	87	87	87	88	88

Lieferung: gegen beigefügten Scheck zuzügl. 5,- DM Versandkosten

Einzelheft- u. Monatsdisketten Bestellung



Ich bitte um weitere Informationen

Ich gebe folgende Bestellung auf

in Bezug auf Ihre Anzeige in Kickstart Heft _____ Seite _____



Abgesandt am:



Firma: _____

Menge	Produkt/Bestellnummer	DM	gesamt DM

Bemerkungen: _____



'TOP 12'
Mein Lieblingsspiel



'TOP 12'
Mein Lieblingsspiel

Wettbewerbsbedingungen siehe
TOP 12 in diesem Heft

Bitte
frei-
machen

'MERLIN'- Computer GmbH
Redaktion KICKSTART
'TOP 12'
Industriestraße 26

D-6236 Eschborn



Kurzmitteilung



Kurzmitteilung

Absender
(Bitte deutlich schreiben)

Vorname/Name

Straße/Nr.

PLZ/Ort

Telefon

Bitte
freimachen

Postkarte

MERLIN Computer GmbH
KICKSTART Redaktion
Industriestraße 26

6236 Eschborn



PD Bestellung



PD Bestellung

Absender
(Bitte deutlich schreiben)

Vorname/Name

Straße/Nr.

PLZ/Ort

Bitte
freimachen

Postkarte

MERLIN Computer GmbH
KICKSTART Redaktion
Industriestraße 26

6236 Eschborn

'TOP 12'
Mein Lieblingsspiel



Meine Meinung

Zu dem Artikel _____ in Heft _____, Seite _____
hätte ich folgendes zu bemerken:

- Ich möchte Ihnen folgendes Programm zur Veröffentlichung anbieten: (Kurzbeschreibung, Sprache, Länge in Druckerseiten)
- Ich kann über folgendes Thema berichten: (Tips & Tricks am AMIGA, Hardware, Software, etc.)
- Ich möchte gerne Autor in der KICKSTART werden. Meine Fachgebiete: (z.B. LISP, Modula-2, Assembler...)
- Ich möchte, daß folgendes Public-Domain Programm in Ihre Sammlung aufgenommen wird.
- Sonstiges

Bei weiteren Angaben oder Fragen wenden Sie sich bitte schriftlich oder telefonisch an die Redaktion. Tel. 0 61 96/48 18 11



Kurzmitteilung



PUBLIC DOMAIN SERVICE

Ich bestelle folgende PD-Disketten:
(Siehe PD Service in dieser Ausgabe)

Zahlung erfolgt:

- per Scheck
- per Nachnahme

Je Diskette fügen Sie bitte einen Betrag von DM 10,- bei,
für Porto und Verpackung je Sendung DM 5,- (Ausland DM 10,-)

Datum

Unterschrift



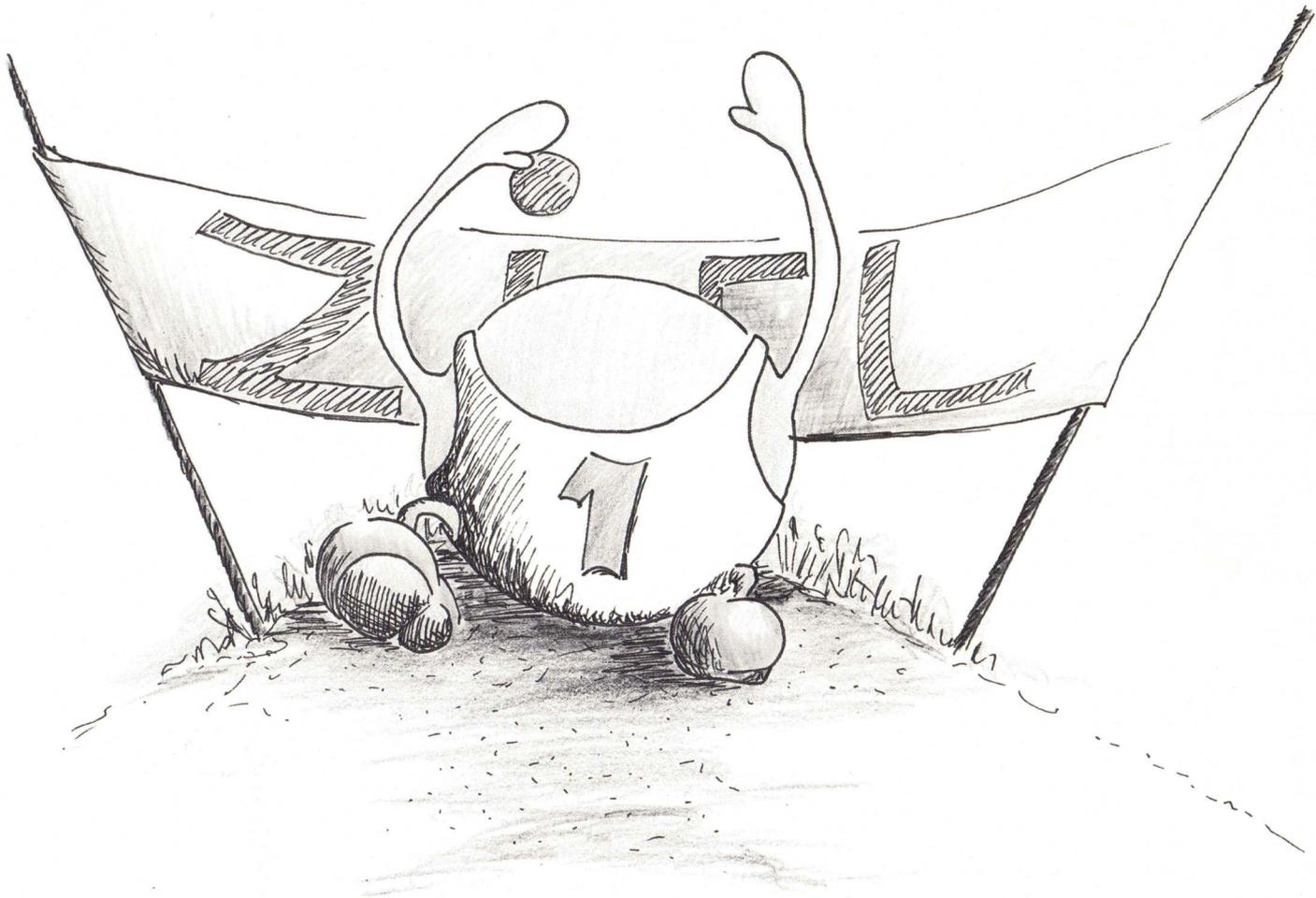
PD Bestellung

Datum

VON FRANK NEUMANN

PLOT ABER FLOTT

Schnelle Punkte im Assembler



Der Amiga ist ja von Haus aus bereits mit einer Menge von Grafikfunktionen versehen. Mit diesen kann man Kreise, Linien, Rechtecke, ja sogar Polygone zeichnen und diese Körper auch ausfüllen lassen.

Nun gibt es da aber eine Sache, bei der der Amiga ruhig noch etwas schneller sein könnte - einzelne Punkte. Moment, werden Sie sagen, dafür gibt es doch `WritePixel()` in der "graphics.library". Das stimmt, aber die Geschwindigkeit dieser Routine ist noch nicht das Opti-

mum, das man aus der Maschine heraus holen kann. So entstand die Idee zu diesem Programm, oder besser diesen Programmunterroutinen.

EINFACHE IDEE

Die Idee ist ganz einfach: Wenn der Amiga einen Punkt setzen soll, muß er jedesmal die genaue Position dieses Punktes innerhalb einer großen Bitmap herausfinden - eine Reihe von 8-Bit-Divisionen, "Shiftungen" und logischen Verknüpfungen. Das dauert natürlich seine Zeit. Anstatt aber all diese Operationen bei jedem neuen Aufruf wieder von vorne durchführen zu lassen, kann man sich einige "digitale Eselsbrücken" bauen, und zwar mittels Tabellen. Sobald ein Screen geöffnet wird, steht fest, wo sich seine Bitplanes befinden. An diesem Punkt "merkt" man sich die Adressen der Zeilenanfänge jeder Grafikzeile und jeden möglichen Abstand der x-Koordinate eines Punktes vom linken Rand, in ganzen Bytes gerechnet. So kann die eigentliche Routine zum Setzen eines Punktes auf ganze 10 Assembler-Befehle reduziert werden. Dementsprechend steigt auch die Ausführungsgeschwindigkeit.

ZUM PROGRAMM:

Das Listing besteht aus mehreren Unter-routinen, die von einem Hauptprogramm aus aufgerufen werden.

- OpenStuff:

Diese Routine öffnet die "intuition.library" sowie einen 320*256-Pixel-Screen mit einer Bitplane und baut die Tabellen auf. Sollte das Öffnen des Screens mißlungen sein (z. B. durch starke Verwüstungen im ChipRam), enthält d0 eine Null. Der User sollte dann das Programm auf geeignete Weise abbrechen.

- FastPlot:

Diese Funktion setzt einen einzelnen Grafikpunkt. Als Parameter werden die X-Koordinate in d0 und Y-Koordinate in d1 erwartet (es werden maximal Wordgrößen akzeptiert). Sollte die Koordinate außerhalb des Bildschirmbereichs liegen, so wird "geclippt", d.

```

1: ;FastPlot and FastCircle - another way to set
2: ;dots and circles quickly
3: ;Written by Frank Neumann, 18.03.1988
4: ; (c) KICKSTART 1988
5: ;Labels von exec.library
6:
7:   execbase = 4
8:   OpenLib = -408
9:   CloseLib = -414
10: ;Labels von intuition.library
11:   OpenScreen = -198
12:   CloseScreen = -66
13:
14: ;----- Code Section
15: run:
16:   jsr   OpenStuff
17:   cmp.w #0,d0
18:   bne   exit
19: ;if screen couldn't be opened, exit immediately
20: ;your own routines should be inserted here...
21: ;example: draw 100 circles with radius
22: ;from 1 to 100, centered in the middle
23: ; of the screen
24:   move.l #1,d2 ;radius...
25:
26:   h0:
27:   move.l #160,d0
28:   move.l #128,d1
29:   jsr   FastCircle
30:   addq.l #1,d2
31:   cmp.l #100,d2
32:   bne   h0
33:
34:   waitleftmouse: ;wait until left mousebutton
35:   move.b $bfe001,d0
36:   and.b #64,d0
37:   bne   waitleftmouse
38:   jsr   CloseStuff
39:
40:   exit:
41:   rts
42:
43: ;--- Subroutines
44:   OpenStuff:
45:   move.l execbase,a6
46:   lea   intname,a1
47:   move.l #0,d0
48:   jsr   OpenLib(a6)
49:   move.l d0,intbase ; OpenLibrary
                          ("intuition.library")
50:   move.l intbase,a6
51:   lea   newscreen,a0
52:   jsr   OpenScreen(a6)
53:   cmp.l #0,d0
54:   bne   set2
55:
56:   err0: ;screen couldn't be opened;
          ; do error-handling
57:   jsr   CloseStuff2 ;close opened libraries
58:   move.l #255,d0
59:   rts
60:
61:   set2:
62:   move.l d0,screen
63:   move.l screen,a0
64:   move.l 192(a0),a0
65:   move.l a0,bpnt ;save bitplane pointer
66:   move.l #10240,d0 ;amount of bytes that have
                          ;to be cleared
67:
68:   c11:
69:   move.b #0,(a0)+
70:   dbra  d0,c11 ;clear bitplane (320 * 256)
71:
72: ;move.l screen,a0
73: ;add.l #192,a0
74: ;move.l 192(a0),d0
75: ;move.l d0,bpnt
76: ;pointer to the (yet only one) bitplane
77: ;and copy it to our pointer
78:

```

```

79: ;now we set up some tables for speed-up
80: lea  adrs,a0
81: move.l #255,d0
82: move.l bpnt,d1
83:
84: l1:
85: move.l d1,(a0)+
86: add.l #40,d1
87: dbra d0,l1 ;table for starting
88: ;addresses of each line
89: move.w #0,d0
90: lea  bytes,a0
91:
92: l2:
93: move.w d0,d1
94: ror.w #3,d1
95: move.b d1,(a0)+
96: add.w #1,d0
97: cmp.w #320,d0
98: bne l2 ;this was the table for whole
99: ;byte counts at each position
100: move.w #0,d0 ;flag: d0=0 -> OpenStuff
; successful
101: rts
102:
103: CloseStuff:
104: move.l intbase,a6
105: move.l screen,a0
106: jsr  CloseScreen(a6)
107:
108: CloseStuff2:
109: move.l execbase,a6
110: move.l intbase,a1
111: jsr  CloseLib(a6)
112: rts
113:
114: ;- Now, here comes the real fast plot!
115: ;- expects:x-coord. in d0, y-coord. in d1!
116:
117: FastPlot:
118: movem.l d0/d1/a0/a1,-(sp)
119: cmp.w #0,d0
120: bge t2
121: jmp  end1 ;boundary check: x low
122:
123: t2:
124: cmp.w #0,d1
125: bge t3
126: jmp  end1 ;y low
127:
128: t3:
129: cmp.w #320,d0
130: blt t4
131: jmp  end1 ;x high
132:
133: t4:
134: cmp.w #256,d1
135: blt t5
136: jmp  end1 ;y high
137:
138: t5:
139: lea  adrs,a0
140: rol.w #2,d1
141: move.l 0(a0,d1),a1
142: lea  bytes,a0
143: clr.w d1
144: move.b 0(a0,d0),d1
145: add.l d1,a1
146: and.w #7,d0
147: eor.b #7,d0
148: bset d0,(a1)
149:
150: end1:
151: movem.l (sp)+,d0/d1/a0/a1
152: rts
153:
154: ;the FastReset subroutine is almost identical
155: ;to FastPlot with the only exeption
156: ;of the BCLR command.
157: FastReset:

```

h., der Punkt wird einfach nicht gezeichnet. Man kann natürlich die Clipping-Test-Befehle herausnehmen, wenn man weiß, daß die Punkte niemals außerhalb des sichtbaren Bereichs liegen werden, und so nochmals einige Zyklen CPU-Zeit einsparen. Aber wenn dann doch einmal ein Punkt danebenflutscht - gnade Dir GURU!

- FastReset:

Wie FastPlot, nur wird der Punkt an der Position x,y gelöscht.

- FastCircle:

Ein weiterer kleiner Zeitbrecher: FastCircle zeichnet Kreise sehr schnell, mit Hilfe eines Algorithmus, der nur mit Additionen und Subtraktionen auskommt, also auch wunderbar in Assembler übersetzbar ist. Als Parameter werden erwartet: Mittelpunkt-x in d0, Mittelpunkt-y in d1, und der Radius in d2. Diese Routine malt also nur runde Kreise, keine Ellipsen! Auf Randüberschreitungen braucht man nicht zu achten, da FastCircle wiederum FastPlot benutzt und dort diese Kontrolle durchgeführt wird.

- CloseStuff:

Schließt den Bildschirm und verschließt die "intuition.library" wieder luftdicht bis zum nächsten Öffnen.

- ToAbs:

Diese Routine ist eigentlich eher ein Nebenprodukt, das von FastCircle benutzt wird. ToAbs erhält als Argument eine Zahl in Word-Größe in d0 und liefert deren Absolutwert zurück.

Die Programmzeilen ab dem Label "run" können für Ihren Programmteil benutzt werden. Als Beispiel ist eine Schleife abgedruckt, die 100 konzentrische Kreise zeichnet. Dieses Programm wurde zeitmäßig erfaßt - etwa 1.78 Sekunden - und ist damit etwa 4 bis 5mal so schnell wie das Vergleichsprogramm in Aztec C. Wenn Sie sich das Listing näher anschauen, werden Sie sich vielleicht wundern, warum das Löschen des Screens nicht mit beispielsweise ClearScreen() aus der "graphics.library" erledigt wird. In der Tat habe ich dies zuerst gemacht, aber entweder habe ich dabei einen großen Fehler gemacht und sollte das Kick-

start in meinem Kopf auswechseln, oder es besteht - was ich für wahrscheinlicher halte - eine Inkompatibilität zwischen den beiden Libraries (Intuition und Graphics). Ein anderer Punkt, der mich auch bewegt, so wenig wie möglich von Systemroutinen Gebrauch zu machen, war die Tatsache, daß einige von ihnen die Registerinhalte verändern. Davor brauchen Sie sich bei FastPlot, FastReset und FastCircle nicht zu fürchten!

Diese erste Version des Programms ist aber, trotz aller Vorzüge, noch sehr starr. Die Auflösung ist voreingestellt, und wollte man diese ändern, so müßten auch die zugehörigen Tabellen anders aufgebaut werden. Dann ist der FastPlot - und dies ist wohl am enttäuschendsten - bisher auf nur eine Bitplane, also 2 Farben ausgelegt, und das bei der Farbenvielfalt des Amiga. Warum aber sollten Sie das Programm nicht erweitern? Wie wäre es z. B. mit sehr schnellen waagerechten und senkrechten Linien (vielleicht kann man hier noch den Blitter übertrumpfen) und frei wählbaren Auflösungen? Möglicherweise wird dadurch eine neue Art der Vektorgrafik ins Leben gerufen - nicht durchgezogene, sondern unterbrochene Linien (für Insider: ich denke hierbei an einige Spiele von Lucasfilm Ltd., die auf 8-Bit-Rechnern für Furore sorgten, vor allem Rescue on Fractalus). Auch Verbesserungsvorschläge werden gerne angenommen. Ich hoffe, daß so ein Ehrgeiz bei Ihnen geweckt wird, aus dem mächtigen Computer, den Sie besitzen, das Letzte herauszuholen! Einsendungen dieser oder ähnlicher Art wird die Redaktion mit großen Interesse entgegensehen.

```

158: movem.l d0/d1/a0/a1,-(sp)
159:
160: cmp.w #0,d0
161: bge rt2
162: jmp endr1 ;(cf. FastPlot)
163:
164: rt2:
165: cmp.w #0,d1
166: bge rt3
167: jmp endr1 ;y low
168:
169: rt3:
170: cmp.w #320,d0
171: blt rt4
172: jmp endr1 ;x high
173:
174: rt4:
175: cmp.w #256,d1
176: blt rt5
177: jmp endr1 ;y high
178:
179: rt5:
180: lea adrs,a0
181: rol.w #2,d1
182: move.l 0(a0,d1),a1
183: lea bytes,a0
184: clr.w d1
185: move.b 0(a0,d0),d1
186: add.l d1,a1
187: and.w #7,d0
188: eor.b #7,d0
189: bclr d0,(a1)
190:
191: endr1:
192: movem.l (sp)+,d0/d1/a0/a1
193: rts
194:
195: ;subroutine to calculate and plot a circle;
; it expects:
196: ;x-coord. in d0, y-coord. in d1, radius in d2
197: FastCircle:
198: movem.l d0/d1/d2/d3,-(sp)
199: move.w d0,x
200: move.w d1,y
201: move.w #0,phi
202: move.w d2,x1
203: move.w #0,y1
204:
205: circ11:
206: move.w phi,d3
207: add.w y1,d3
208: add.w y1,d3
209: addq.w #1,d3
210: move.w d3,phiy
211: sub.w x1,d3
212: sub.w x1,d3
213: addq.w #1,d3
214: move.w d3,phixy
215: move.w x,d0
216: add.w x1,d0
217: move.w y,d1
218: add.w y1,d1
219: jsr FastPlot ;x+x1,y+y1
220: move.w x,d0
221: sub.w x1,d0
222: move.w y,d1
223: add.w y1,d1
224: jsr FastPlot ;x-x1,y+y1
225: move.w x,d0
226: add.w x1,d0
227: move.w y,d1
228: sub.w y1,d1
229: jsr FastPlot ;x+x1,y-y1
230: move.w x,d0
231: sub.w x1,d0
232: move.w y,d1
233: sub.w y1,d1
234: jsr FastPlot ;x-x1,y-y1
235: move.w x,d0
236: add.w y1,d0
237: move.w y,d1

```

```

238: add.w  x1,d1
239: jsr   FastPlot  ;x+y1,y+x1
240: move.w x,d0
241: sub.w  y1,d0
242: move.w y,d1
243: add.w  x1,d1
244: jsr   FastPlot  ;x-y1,y+x1
245: move.w x,d0
246: add.w  y1,d0
247: move.w y,d1
248: sub.w  x1,d1
249: jsr   FastPlot  ;x+y1,y-x1
250: move.w x,d0
251: sub.w  y1,d0
252: move.w y,d1
253: sub.w  x1,d1
254: jsr   FastPlot  ;x-y1,y-x1
255: move.w phiy,d0
256: move.w d0,phi
257: addq.w #1,y1
258: move.w phixy,d0
259: jsr   ToAbs
260: move.w d0,d1
261: move.w phiy,d0
262: jsr   ToAbs
263: cmp.w  d0,d1
264: bge   circ13
265: move.w phixy,d0
266: move.w d0,phi
267: subq.w #1,x1
268:
269: circ13:
270: move.w y1,d0
271: cmp.w  x1,d0
272: ble   circ11
273: movem.l (sp)+,d0/d1/d2/d3
274: rts
275:
276: ;subroutine that returns the absolute value
277: ;of a word argument

```

```

278: ToAbs:
279: cmp.w  #0,d0
280: bmi   ToA1
281: rts
282: ToA1:
283: neg.w  d0
284: rts
285:
286: ;----- Data Section
287: intname: dc.b "intuition.library",0,0
288: even
289: intbase: blk.l 1
290:
291: newscreen:
292: dc.w 0 ;LeftEdge
293: dc.w 0 ;TopEdge
294: dc.w 320 ;Width
295: dc.w 256 ;Height
296: dc.w 1 ;Depth
297: dc.b 0 ;DetailPen
298: dc.b 1 ;BlockPen
299: dc.w 0 ;ViewModes...LORES
300: dc.w 15 ;Type :here we use CUSTOMSCREEN
301: dc.l 0 ;Font
302: dc.l 0 ;Title
303: dc.l 0 ;Gadgets
304: dc.l 0 ;BitMap
305:
306: screen: blk.l 1 ;pointer to screen structure
307: bpnt: blk.l 1 ;pointer to the bitplane
308: adrs: blk.l 256 ;auxiliary arrays
309: bytes: blk.b 320 ; - " -
310: phi: blk.w 1 ;reg. for circle calculations
311: phiy: blk.w 1 ; - " -
312: phixy: blk.w 1 ; - " -
313: x1: blk.w 1 ; - " -
314: y1: blk.w 1 ; - " -
315: x: blk.w 1 ; - " -
316: y: blk.w 1 ; - " -

```

Listing 1: Schnelle Plot- und Circle Routinen

ENDE

SSS — SIGGIS SOFTWARE SHOP

KNÜLLERPREISE:

Artic Fox.....	DM 54,50	Asterix.....	DM 62,50
Emetic Skimmer.....	DM 38,50	Blueberry.....	DM 62,50
Fam. Feuerstein.....	DM 54,50	BMX Simulator.....	DM 42,50
Flight Simulat.....	DM 99,50	Bureacracy.....	DM 84,50
Gridstart.....	DM 26,50	Defender o. Crown.....	DM 66,50
Giana Sisters.....	DM 52,50	Eagles Nest.....	DM 54,50
Iridon.....	DM 42,50	Ferrari Form. 1.....	DM 72,50
King of Chikago.....	DM 68,50	Jinxter.....	DM 62,50
Las Vegas.....	DM 26,50	Leaderboard Golf.....	DM 62,50
Lurkins Horror.....	DM 74,50	Moebius.....	DM 62,50
Planetfall.....	DM 74,50	Mewillo.....	DM 62,50
Slaygon.....	DM 52,50	Starglider.....	DM 62,50
Tetris.....	DM 49,50	Star Wars.....	DM 48,50
Wizball.....	DM 54,50	Winterolymp. 88.....	DM 52,50
Jet.....	DM 99,50	Interceptor.....	DM 68,50

PD Fred Fish 1-120 / Faug 1-50.....	Stück DM 4,50
Juki 5520 Parallel/Seriell.....	DM 1198,00 / 1328,00
Juki 7100.....	DM 2768,00
Golem Drive 3.5" / 5.25" extern.....	DM 328,00 / 398,00
Golem Ram Box 1000/500 2 MB LED.....	DM 1089,00
Golem Sound mono/stereo LED.....	DM 136,00 / 186,00
Vorkasse Software.....	+ 4,50 DM / NN + 7,50 DM
Vorkasse Hardware.....	+ 10,50 DM / NN + 13,50 DM

S. GEBAUER

Parkstraße 7a • 5880 Lüdenscheid • Telefon (023 51) 245 02

DONAU-SOFT

Ihr Public Domain-Partner
mit weit über 550 PD-Disk im Archiv

► ab DM 3,— ◀

Einzelstück.....	DM 4,50	Tornado, Auge, Fish,
ab 10 Stück.....	je DM 4,00	Chiron, Panorama,
ab 50 Stück.....	je DM 3,50	Amicus, ACS, RW,
ab 100 Stück.....	je DM 3,30	Kickstart, Faug,
ab 200 Stück.....	je DM 3,00	Taifun, TBAG, u. a.

Preise incl. 2DD 3,5"-Disk

► **Mit Qualitätsgarantie** ◀
— Alle Disk sind etikettiert —

2 ausführl. Katalogdisketten gegen DM 6,—
(V-Scheck oder Briefmarken anfordern!)

Disketten 2DD — ab DM 2,20

Versandkosten: Vorkasse DM 4,—
Nachnahme DM 6,—, Ausland Vorkasse DM 8,—

Maik Hauer

Postfach 14 01 • 8858 Neuburg/Do • Tel. (084 31) 49798



Peter Biet
Computerdesign
Georg-Fischer-Str. 5
6415 Petersberg
Tel.: 06 61/60 12 63



PAL-RGB-MULTIPROZESSOR

das kleine Gerät mit den großen
Möglichkeiten
- RGB-Farbsplittung des Farbvideo-
signals für alle Digitizer
- Digitalisierung mit Farbkamera oder
vom Videorecorder möglich
- Veränderung des Videosignals
(zur Überspiegelung, Genlocking
usw.) mit Hilfe von Helligkeits-,
Kontrast-, Farbsättigungs- und
RGB-Intensitätsreglern
- erzeugt aus dem Computer
RGB-Signal ein PAL-Video signal
(ersetzt damit herkömmliche
PAL-Videoarten)
Version 2 (Profiausführ.) 886,— DM
siehe auch Aktuell-Meldung in AMIGA 5/88



DVS 2000

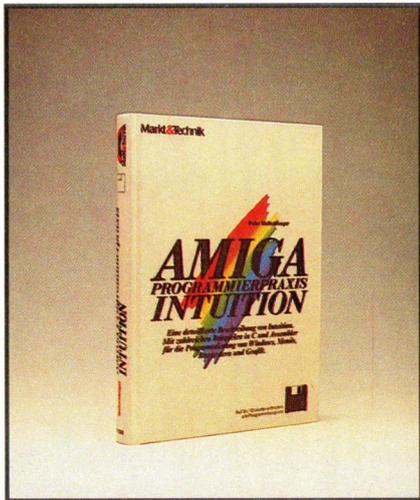
Das bewährte Realtimefixing Digitizersystem in neuer überarbeiteter
PAL-Version.
Lieferumfang:

In Verbind. m. unserem PAL-RGB-MULTIPROZESSOR 2298,— DM
Sonstige Hard- und Software auf Anfrage.

Wir digitalisieren auf Ihre Grafiken.

Nähere Infos und Produktliste gegen frankierten Rückumschlag!

Händleranfragen erwünscht!



AMIGA Programmierpraxis Intuition

Peter Wollschlaeger
330 Seiten
Markt & Technik Verlag AG
1. Auflage 1988
Preis: 69.- DM

Der Programmierung der grafikorientierten Benutzeroberfläche oder auch der Programmierung der Library Intuition ist das vorliegende Buch gewidmet. Der Autor greift zu diesem Zweck

hauptsächlich auf die Programmiersprache C zurück, aber auch Assemblerfreunde finden einige Listings, die erfreulicherweise gut dokumentiert sind, was von den C-Listings nicht behauptet werden kann.

Peter Wollschlaeger behandelt in 12 Kapiteln die verschiedenen Intuition-Komponenten und erwähnt auch ein paar Schnittstellen zu Intuition. Die typischen Intuition-Komponenten wie Fenster, Screens, Menüs, Gadgets, Requester, Alerts usw. werden ausführlich erklärt; außerdem findet der Leser einige interessante Zugaben, kleinere Abstecher in andere Libraries oder nützliche Programmiertips, beispielsweise wird erklärt, wie Grafikdaten in das Chip-Memory gelegt werden.

Der Stil des Autors ist locker und nicht zu kompliziert. Die Erklärungen sind jedoch zum Teil zu einfach gehalten und bleiben aus diesem Grund zu oberflächlich, so daß manchmal Verständnisprobleme auftreten können.

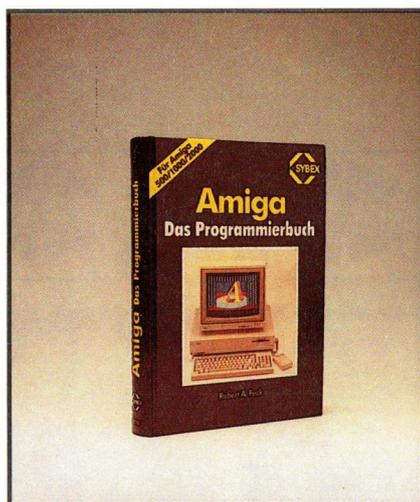
Alle Kapitel sind mit zahlreichen Listings versehen, die zum Teil aufeinander aufbauen, so daß Teile von anderen Listings eingebunden werden müssen.

Im Kaufpreis ist eine Diskette inbegriffen, auf der sich alle Beispielsprogramme finden. Der Leser kann sich die Tipparbeit sparen und erhält somit eine gewisse Garantie, daß die abgedruckten Programme auch lauffähig sind.

Fazit

Als abendliche Bettlektüre ist das Werk nicht geeignet, sondern eher als Nachschlagewerk beim Programmieren. Der Käufer erhält für sein Geld eine ausführliche Erklärung der Programmierung der grafikorientierten Oberfläche. Die Spezifizierung des Buches ermöglichte dem Autor, die einzelnen Kapitel detailreich abzuhandeln, obwohl manchmal nur die Oberfläche eines Themas berührt wird. Der Käufer sollte zumindest ein wenig C bzw. Assembler beherrschen, ansonsten muß unterstützende Literatur der Programmiersprachen herangezogen werden. Zwar ist ein Kapitel den Sprachen gewidmet, dieses kann aber keinen Anspruch auf Vollständigkeit erheben.

(AK)



Amiga Das Programmierbuch

Robert A. Peck
456 Seiten
Sybex-Verlag GmbH
1. Auflage 1988
Preis: 49.- DM

Für alle Amiga-Modelle vertreibt der Sybex-Verlag ein Programmierhand-

buch, welches global die komplette Programmierung des Amiga erfaßt.

Der Autor greift bei seinem Werk auf die Programmiersprache C zurück, so daß der Leser dieser Sprache zumindest ein wenig mächtig sein sollte, denn ein stures Abtippen der zahlreichen Listings bringt dem Leser recht wenig, er sollte schon wissen, was gerade programmiert wird.

Neun Kapitel findet der Leser in dem Buch, wobei das erste Kapitel auf grundlegende Dinge der Programmierung auf dem Amiga eingeht. Das zweite Kapitel greift die Programmierung des AmigaDOS auf, weiter geht es mit dem Betriebssystem-Kern, der Exec, der Grafik, der Programmierung von Intuition, von Devices, von Animation, Sound und des Multitaskings. Als Anhang findet man des weiteren einen Abschnitt über den System-Editor und über Amiga-C-Compiler.

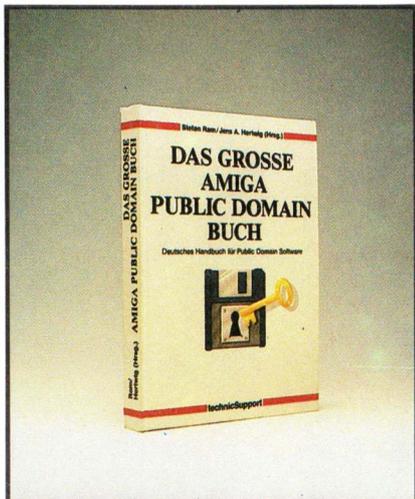
Die Stil des Autors ist sachlich, und ohne große Umschweife wird auf das Wesentliche eingegangen. Anfänger werden sich bisweilen etwas schwer

tun, da der Autor etwas Grundwissen voraussetzt. Die zahlreichen Amiga-Funktionen sind nicht der Reihe nach abgehandelt, sondern werden im fortlaufenden Text erklärt. In zahlreichen Programmbeispielen werden viele Funktionen dem Leser näher gebracht. Hin und wieder unterstützen Grafiken das Erklärte.

Fazit

“Amiga - Das Programmierbuch” bietet einen globalen Überblick der Programmierung des Amiga. Es behandelt fast alle spezifischen Amiga-Gebiete, wobei einige jedoch etwas zu kurz kommen. Das Niveau des Buches ist leicht angehoben und für den C-Laien nicht ratsam. Als Nachschlagewerk kann das Buch durchaus herangezogen werden, wobei die einzelnen Funktionen in keiner übersichtlichen Reihenfolge angeordnet sind. Die im Buch abgedruckten Listings sind leider nicht auf Diskette erhältlich.

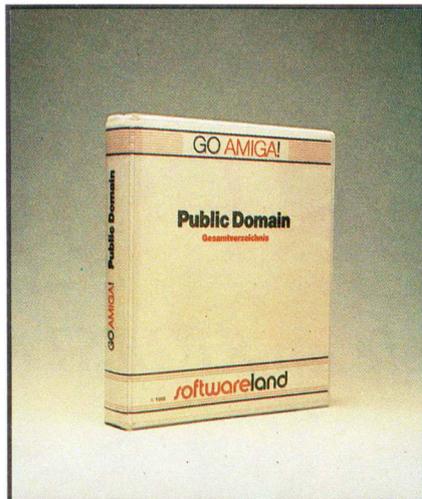
(AK)



DAS GROSSE AMIGA PD-BUCH

Ram/Hertwig (Hrsg.)
TechnicSupport
350 Seiten
DM 49.-

Gleich zwei Bücher versuchen sich daran, den rasend wachsenden Public Domain-Markt unter Kontrolle zu bringen. Grundsätzlich unterscheiden sich beide Bücher in ihrem Konzept. Das PD-GESAMTVERZEICHNIS von Softwareland ist als Ringordner ausgelegt. Dadurch ist es leicht möglich, dieses Werk ständig zu aktualisieren. Dies geschieht alle 3 Monate, denn dann erscheinen jeweils 60 weitere Seiten zu einem Preis von 24.- DM. Beschrieben werden die Serien FAUG (-51), FISH (-73) und GALIB (-10). Jeder Diskette sind dabei 1-3 Seiten gewidmet, auf denen die Programme kurz vorgestellt werden. Im Anhang findet man außerdem einen Index, der alle Programme, sortiert nach Programmnamen, aufführt, so daß man schnell die gewünschte Diskette herausfinden kann. Grundsätzlich ist der Gedanke, das Verzeichnis in Ringbuchformat anzulegen und nach und nach zu erweitern, nicht schlecht, allerdings ist durch die 3-monatige Erscheinungsweise zu befürchten, daß der aktuelle Stand der PD-Software nicht so schnell erreicht werden wird. Der Preis von 24.- DM pro Lieferung (100.- DM im Jahr) ist zudem recht happig.



PD GESAMTVERZEICHNIS

Hans-Ulrich Hugi
Softwareland
ca. 200 Seiten
DM 49.-

Das PD GESAMTVERZEICHNIS ist vor allem denjenigen zu empfehlen, die an Programmen der FISH- oder FAUG-Serie interessiert sind. Die Programme werden kurz vorgestellt, was gerade langt, um einen ersten Eindruck zu bekommen. Das GROSSE PD-BUCH des TechnicSupport ist nur der erste Teil der geplanten Buchserie zum Thema PD-Software. Das Werk beginnt mit einer knapp 40seitigen Einleitung, in der der Umgang mit PD-Disketten erklärt wird. Danach folgt eine ausführliche Beschreibung von rund 40 Programmen der Gebiete Anwendungsprogramme, Grafiksoftware, Arbeits- und Programmierhilfen. Den Abschluß des Buches bilden die Listen, in denen die Programme alphabetisch, nach Themenbereichen und nach PD-Sammlungen sortiert, aufgeführt sind. Grundlage des Buchs sind die Sammlungen von FAUG (-39), FISH (-127), PANORAMA (-17b). Das GROSSE AMIGA PD-BUCH ist besonders wegen der aktuellen und gut angelegten Listen eine große Hilfe beim Auffinden von Programmen. Die Auswahl der Serien ist relativ gut, denn neben der FISH-Reihe gibt es fast keine Sammlung, die nicht vorwiegend

auf diese aufbaut. Die Übersicht ist zudem recht aktuell, trotzdem wird es für Interessierte unerlässlich sein, auch die folgenden Werke zu erstellen. Die Beschreibung der einzelnen Programme ist ausführlich und gut, allerdings kann man sich über die Auswahl streiten (18 Seiten über die Demo-Version einer Textverarbeitung, bei der man nicht DRUCKEN kann?!). Außerdem wird die Auswahl von 40 Programmen aus vorsichtig geschätzt 2000 Programmen dem Angebot nicht annähernd gerecht.

Einen Überblick über die PD-Szene mit mehr als 800 Disketten geben die Bücher, wegen der Einschränkung auf FISH, PANORAM und FAUG, nicht. Das GROSSE AMIGA PD-BUCH bietet zwar umfangreiche Listen, aber es fehlen wichtige Angaben über Versionsnummern der Programme und Überschneidungen der Sammlungen. Das PD GESAMTVERZEICHNIS stellt die einzelnen Programme der Serien kurz, aber meist ausreichend, um sie einzuordnen, dar. Leider ist dieses Werk nicht aktuell und teuer. Eine Symbiose beider Bücher wäre sicherlich die beste Lösung, aber die ist noch nicht in Sicht.

(mn)

ENDE

LESERFORUM

Briefe an KICKSTART

RLL-FESTPLATTEN AM AMIGA

Entgegen dem Artikel von Christian Keller aus Kickstart 4/88, möchte ich Euch darauf hinweisen, daß der Betrieb einer Festplatte mit RLL-Controller in Verbindung mit dem Sidecar sehr wohl möglich ist.

Ich betreibe seit einigen Tagen eine 32MB-Goldcard mit OMTI 5527-Controller im Sidecar zum Amiga 1000, und sie läuft ohne Probleme. Ich habe 12 MB für das Sidecar und 19 MB für den Amiga partitioniert. Der DPFOR-MAT-Befehl hat die Amigapartition korrekt mit 26 Sektoren pro Track formatiert.

Die von mir benutzten Befehle haben folgende Länge:

DPFORMAT - 3280 Bytes
DJMOUNT - 2904 Bytes
JANUS.LIB - 3456 Bytes

Des weiteren möchte ich darauf hinweisen, daß es mittels des neuen Format-Befehls der Workbench 1.3 (Version 34.4) nicht möglich ist, die Amigapartition einer PC-Festplatte mit dem neuen Fast-File-System für Festplatten zu formatieren.

Es ist wohl ein eigener Format-Befehl erforderlich, so wie es bisher auch schon war (hoffentlich läßt uns Commodore nicht im Stich).

Michael Holin / Zellerfeld

*Sehr geehrter Herr Holin,
Die Aussage in meinem Artikel über die Verwendbarkeit einer RLL-Platte als Janusdisk, stützt sich auf die Erfahrungen mit einer 30 MByte Eurocomp-FileCard und darauffolgenden Telefonaten mit Commodore. Mittlerweile*

finden neue Versuche mit einer 50 Mbyte-Mountain-Filecard statt, die auch nach dem RLL-Prinzip arbeitet. Dabei zeigte sich eine langsamere Übertragung der Daten zum AMIGA im Vergleich zu MFM-Platten. Anscheinend handelt es sich um Timingprobleme bei der Benutzung einer RLL-Platte. Die Aussage im Artikel aus Heft 4/88 muß dahingehend abgeändert werden, daß ein Funktionieren einer RLL-Platte im Janus-Betrieb nicht garantiert werden kann, respektive fragwürdig ist. Als Käufer sollte man also eine gewisse Vorsicht walten lassen.

Sehr zu begrüßen ist es, daß Leser ihre Erfahrungen diesbezüglich an andere AMIGA-Eigner weitergeben. Hier also die Aufforderung an andere Janusdisk-Betreiber, ihre Erfahrungen wie Herr Holin der KICKSTART-Redaktion zu melden, damit sie an die Leserschaft weitergereicht werden können.

Der Format-Befehl des FF-Systems funktioniert in der Tat nur mit der AMIGA-seitigen Commodore-Harddisk. Der Hoffnung des Herrn Holin bezüglich der Unterstützung seitens Commodore kann ich mich nur anschließen.

Christian Keller

KICKSTART-EPROM BRENNEN

Sehr geehrte Kickstart Redaktion, ich habe den Eprombrenner aus den Ausgaben 5-6/88 nachgebaut und auch die Software abgetippt. Ich wollte mir eine Kickstart-Version auf Eproms brennen und mittels Umschaltplatine in meinen A2000 einbauen. Leider komme ich damit nicht klar. Da ich bisher keine Erfahrung mit dem Bren-

nen von Eproms habe, weiß nicht wie ich die 256 KByte große Kickstart so aufteilen muß, daß sie in vier 64 KByte Eproms paßt und funktioniert.

In der Anleitung zu meiner Umschaltplatine steht nur:

Die zusätzlichen Kickstartversionen (2) müssen auf jeweils vier 64K Eproms gebrannt werden. Um das Kickstartprogramm in 27512 Eproms unterzubringen, muß das Programm in entsprechende Blöcke aufgeteilt werden (aber wie?). Der 128 K große Adreßbereich muß in zwei Bänke aufgeteilt werden (Wie ?). So erhält man vier Eproms für die Adreßbereiche \$fa000-\$fbfff und \$fc000-\$fffff. Da der Amiga einen 16 Bit Datenbus hat, muß in ein Eprom das Low-Byte und in das andere das High-Byte gebrannt werden.

Mag sein, daß ein alter Hase im Eprombrennen da gleich Bescheid weiß, aber ich verstehe nur Bahnhof. Vielleicht können Sie einmal verständlich erklären, was zu tun ist, um die Kickstart von einer Diskette in die Eproms zu bekommen. Vielleicht gibt es ja auch ein kleines Programm, das die Kickstart-Version in brennfertige Files umwandelt. Das wäre echt super!

Jürgen Hilbich / Simmerath

Ihr Problem ist mit Erscheinen dieser Ausgabe bereits gelöst. In diesem Heft finden Sie ein Programm, das eine bestehende Kickstart-Version von Diskette einliest und daraus vier epromgerechte Dateien erzeugt.

Die Red.

KICKSTART HEFT 6/88

Sehr geehrte Damen und Herren, nachdem die ersten Hefte noch recht amateurhaft angelegt und inhaltlich und drucktechnisch fehlerhaft waren, hat sich inzwischen manches sehr positiv entwickelt - wie ich auch gehofft hatte.

Was mich allerdings in letzter Zeit stört, ist folgendes:

- Die Hardware-Basteleien nehmen zuviel Raum ein. Ich kann mir kaum vorstellen, daß der Amiga ein ähnliches Bastel-Objekte wie der C64 wird (mit welchem ich selbst auch viel 'gebastelt' habe!).

- Im neuesten Heft 6/88 werden 15 Seiten für die Eprommer-Software und 12 Seiten für 'Bootletter' verschwendet. Damit bin ich ganz und gar nicht einverstanden. Und: Ich hasse Data-Wüsten. Etwas Stupideres als das Abtippen solcher Listen gibt es wohl kaum.

Vorschlag:

Solche Mammut-Listings nur auf Service-Diskette. Dafür im Heft noch mehr von den kleineren Lösungen und Kursen zu Spezialthemen. Auch detaillierte Hinweise zur sinnvollen Bedienung von Software, für die nur englischsprachige Handbücher geliefert werden, wäre wünschenswert.

- Ich habe bisher alle Service-Disketten zu den Heften bezogen. Es fällt auf, daß manchmal Listings aus den Heften fehlen. Und warum werden die Listings in den Kursen nicht auf Diskette gebracht?

Hermann Dörries / Wildeshausen

Zu den Punkten 1 und 2: In nächsten Heft befindet sich eine Leser-Umfrage, mit deren Hilfe wir die inhaltliche Gestaltung der Zeitschrift an den Lesersprüchen messen wollen. Daß die Ausgabe 6/88 mit zwei langen Listings gefüllt war, soll keineswegs zum Regelfall werden. Bootletter ist unserer Meinung nach ein Programm, welches nicht nur interessant in der Anwendung ist, sondern ebenso wertvoll für

Leser, die den ein oder anderen Programmteil in eigenen Programmen benutzen wollen.

Zu Punkt 3: Auf den Monatsdisketten, die sowohl beim Verlag, als auch direkt bei der Redaktion erhältlich sind, sind alle Listings sowohl als Source, als auch in kompilierter Form enthalten, mit Ausnahme kurzer Beispielprogramme (aus Kursen oder Grundlagentartikeln), die in der Regel schneller eingetippt sind, als auf der Diskette gefunden. Weiterhin wollen wir natürlich gerade bei Kursen den Lerneffekt erhalten. Das Eintippen kurzer Programme und einzelner Zeilen gehört dazu.

Die Red.

FESTPLATTE AM AMIGA 2000

Sehr geehrte Damen und Herren, ich besitze einen Amiga 2000 mit PC-Karte und möchte mir nun eine Festplatte zulegen. Da Amiga-Festplatten sehr teuer sind, ziehe ich Festplatten für den PC in Betracht, habe hierzu aber einige Fragen.

Kann ich eine normale PC-Festplatte mit normalem Controller (oder einer Filecard) an meinen Amiga anschließen und gegebenenfalls im Gehäuse unterbringen?

Ist es möglich, solch eine Festplatte in einen Amiga- bzw. PC-Teil zu teilen, und wenn dies möglich ist, was muß ich tun?

Sind beide Teile bootfähig?

Uwe Henrici / Neu-Anspach

Grundsätzlich ist der Betrieb einer 'ganz normalen PC-Festplatte' in Verbindung mit der XT-Karte ohne Probleme möglich. Schwierigkeiten kann es jedoch unter Umständen mit einer RLL-Festplatte geben, was jedoch nicht unbedingt zutreffen muß (siehe auch weiter oben). Die einfachste Lösung bietet in diesem Fall eine Filecard, da hier auch gleichzeitig das Problem der Befestigung im Rechner

gelöst ist. Aber auch Festplatten mit getrenntem Controller lassen sich im Amiga betreiben. Die Platte findet dann an der Stelle des zweiten 3.25"-Diskettenlaufwerks seinen Platz, (wenn Sie eine 3.25" Festplatte verwenden und nur ein internes Laufwerk im Amiga besitzen) bzw. muß in einem geeigneten Gehäuse, extern neben dem Rechner untergebracht werden.

Die Festplatte kann mit dem im Lieferumfang des MS-DOS der PC-Seite enthaltenen Programm FDISK in beliebigen Verhältnissen für den PC und den Amiga aufgeteilt werden. Bootfähig ist in dieser Hardwarekonstellation jedoch nur die PC-Seite, da von der Amigaseite aus gesehen, erst die Janus-Library von Diskette geladen werden muß, bevor der 68000 auf die Amiga-Partition der Festplatte zugreifen kann.

Die Red.

DFÜ

Endlich wird einmal etwas über DFÜ berichtet! Hoffentlich kommen auch mal einige Tests über Modems. Außerdem würde mich persönlich interessieren, ob es für dem Amiga 500 schon etwas gibt, womit man BTX betreiben kann.

Peter Maiwald / Starnberg

Ein Test über verschiedene Modems ist bis jetzt noch nicht geplant, für die nähere Zukunft jedoch vorgesehen. BTX mit einem Amiga ist derzeit nur mit dem 2000 und der XT-Karte verfügbar. Bei dieser Lösung kann jedoch für den Amiga-Anwender nur von einer Krückstock-Methode gesprochen werden. Die BTX-Karte wird vom PC-Teil betrieben, und der Amiga übernimmt via PC Window die Darstellung. Es kursiert jedoch bei Commodore ein Gerücht von einem softwaremäßigen BTX-Dekoder für die Amiga-Modelle. Wie weit ein solches Programm schon existiert, konnten wir leider nicht in Erfahrung bringen.

Die Red.



TRUE BASIC GANZ GROß!

Beim Programm TRUE BASIC läßt sich das Eingabefenster mit folgendem Patch auf volle PAL-Größe bringen:

1. Mit einem Filemonitor (z.B. FILE-ZAP oder NEWZAP) nach \$028000C8 suchen!
2. In Sektor 28 und 29 die gefundenen Stellen in \$02800100 ändern und die Sektoren abspeichern.

Nun öffnet TRUE BASIC beim Starten sein Fenster auf der vollen Größe (dies gilt für die 143048 Byte lange Version).

(Michael Holin, Clausthal-Zellerfeld)

FENSTER SCHLIEßEN!

Wenn in der Startup-Sequence Programme mit RUN gestartet werden, läßt sich das CLI-Fenster nicht mehr schließen. Dies läßt sich umgehen, indem man die Programme folgendermaßen startet:

```
RUN >NIL: Programmname
```

JANUS ENTTARNEN...

Besitzer von Sidecar oder Bridgeboard und einer Festplatte mit einer AMIGA-Partition werden bemerkt haben, daß die Platte unter dem Namen JH0: von kaum einem Programm unterstützt wird. Die Softwareentwickler scheinen nur an die Besitzer der reinen AMIGA-Harddisk zu denken. Doch kein Problem, man muß einfach in die Startup-Sequence der BOOT-Diskette folgenden Befehl einfügen:

```
ASSIGN DH0: JH0:
```

und schon kann man auch unter dem Gerätenamen DH0: auf die Festplatte zugreifen.

(Michael Holin, Clausthal-Zellerfeld)

PREFERENCES FOR EVER

Wer kennt das Problem nicht? Man startet ein Spiel oder Anwendungsprogramm, das einen Autostart besitzt und dann kommt die Überraschung: das Bild ist nicht in der Mitte des Schirms, oder man hat zu wenig Platz auf dem Tisch, weil die Mauseinstellungen nicht passend eingestellt ist. Um dies zu ändern, müsste man entweder auf jeder Diskette die PREFERENCES aufrufen und die entsprechenden

SCHNELLE LIBRARIES IN BASIC

Jeder, der in BASIC programmiert, weiß, wie hilfreich die Verwendung der Libraries (Systembibliotheken) ist. Wenn man aber zu Testzwecken ein Programm immer wieder starten muß, dann wird das ständige Nachladen der Libraries zu einem Geduldsspiel. Abhilfe schafft man, indem diese Teile auf die RAM-Disk kopiert werden. Dies macht man entweder vom CLI aus oder in der Startup-Sequence:

```
copy #?.bmap to ram: quiet
```

Danach müssen nur noch im Basic-Programm die Pfadnamen der LIBRARIES geändert werden, damit diese auf der RAM-Disk gefunden werden:

```
LIBRARY "ram:graphics.library"
```

(Dietmar Augustin, Frechen)

te zu booten, weil das Programm PREFERENCES immer auf die Bootdiskette zugreift. Nach dem Reset muß sofort <Ctrl> d gedrückt werden, um die Startsequenz abzubrechen, denn sonst kann es sein, daß das Programm die Tastatur nicht mehr abfragt und sich deshalb nicht mehr stoppen läßt. Nun legt man die eigene Workbench-Diskette ins Laufwerk df0: und gibt 'df0:preferences' ein. Nach kurzer Zeit erscheint das Menü mit den bekannten Einstellungen, die man prinzipiell noch ändern könnte, aber wir gehen ja davon aus, daß dies die bevorzugten Werte

sind, die man beibehalten möchte. Deshalb wird der Punkt 'SAVE' ausgewählt, woraufhin Ihr Rechner die Bootdiskette verlangen wird. Wenn Sie diese mit geöffnetem Schreibschutz einlegen wird die Einstellung der PREFERENCES darauf kopiert - fertig! Dieses Verfahren wird mit den meisten Programmdisketten funktionieren, obwohl es vorkommt, daß Programme - z.B. einige Spiele - ihre Einstellungen nicht aus der SYSTEM-CONFIGURATION übernehmen, sondern auf interne Werte zugreifen.

Laufwerken sehr einfach ablaufen. Einer neu formatierten Diskette wird der Name "fonts" gegeben. Auf diese Diskette müssen dann die neuen Zeichensätze kopiert werden. Bitte beachten Sie, daß sich kein Unterverzeichnis mit dem Namen "fonts" auf der Diskette befinden darf. Starten Sie den Rechner nun wie folgt: Programmdiskette in df0:, "fonts"-Diskette in df1:. Die Zeichensätze, die sich auf der "fonts"-Diskette befinden, werden automatisch eingebunden. Die Zeichensätze der Programmdiskette sind nun allerdings nicht mehr erreichbar. Außerdem sollten Sie beachten, daß manche Programme nur eine begrenzte Anzahl von Zeichensätzen verwalten können. Sollten sich auf der "fonts"-Diskette zu viele befinden, stürzen die Programme kommentarlos ab (z.B. Notepad).

(Herbert Penke, Bad Meinberg 1)

CURSOR-POSITIONIERUNG

Die Positionierung des Cursors ist in C eine einfache Sache, wenn man die ESCAPE-Sequenzen verwendet. Diese können direkt in der printf()-Anweisung angegeben werden:

```
printf("\33[12H")
```

oder

```
printf("\x9b12H")
```

setzt den Cursor in die 12. Zeile;

```
printf("\033[12;5H")
```

oder

```
printf("\x9b12;5H")
```

setzt den Cursor in die 5. Spalte der 12. Zeile.

Selbstverständlich kann diese Positionierung auch variabel erfolgen. Der Ausdruck lautet dann:

```
printf("\33[%d;%dH", YPos, XPos")
```

(Herbert Penke, Bad Meinberg 1)

FONTS FÜR NOTEPAD

Das Einbinden weiterer Zeichensätze in Programme wie Notepad oder DPAINT kann für Besitzer von zwei

Bei uns werben bringt

GEWINN

Sprechen Sie mit uns.

Heim Verlag ☎ 06151/56057

Jet mit deutscher Anleitung

ist da!

Nur DM 99,90

Deutsche Anleitung einzeln nur DM 15,90

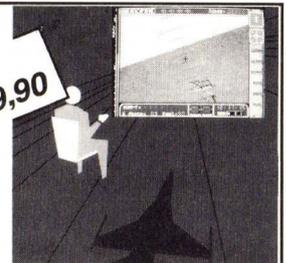
Zögern Sie nicht, kaufen Sie sich Ihren persönlichen Jet-Simulator.

Luft-Luft/Luft-Boden/Luftbodenkampf wählbar.

Nachnahme DM 6,-, Vorkasse DM 4,-, Ausland nur gegen Vorkasse DM 10,-

PDC-GmbH

Louisenstraße 115 · 6380 Bad Homburg · Tel. (06172) 24748 oder 20799



IM ÜBERBLICK

Amiga-Clubs

Des öfteren erreichen uns Anfragen von Lesern, die gerne in einen Computer-Club eintreten möchten, aber keine Bezugsadressen besitzen. Wie haben für Sie deshalb alle uns bekannten Adressen zusammengestellt und hoffen, daß Sie in Ihrem Bereich fündig werden. In der Aufstellung finden Sie kleinere Informationen über den betreffenden Club. Sie beinhalten die Kontakt-Adresse und die allgemeinen Interessen des Clubs, ob der Club eine Clubzeitschrift besitzt und wie hoch ein eventuell zu entrichteter Beitrags liegt. Für nähere Informationen können Sie sich mit den Clubs in Verbindung setzen. Jedoch sollten Sie bei schriftlicher Kontaktaufnahme Rückporto beilegen, da dies die einzelnen Clubkassen entlastet.

Die Clubs sind nach Postleitzahlen der Städte geordnet, dadurch können Sie sich den Club in Ihrer Nähe schnell heraussuchen.

Postleitzahl 1000

Das darf doch wohl nicht wahr sein. Berliner Clubs bitte sofort melden)

Postleitzahl 2000

Commodorus Computerclub

c/o Christian Hartmann
Marderstraat 72
2000 Hamburg 65
Allgemeine Clubinfo:
Hilfen für Anfänger, Erfahrungsaustausch, Public Domain Pool, spezieller Video Service.

Clubzeitschrift vorhanden
Beitrag:
Inland 10.- DM jährlich
Ausland 20.- DM jährlich

ACD

Amiga Club Deutschland
Nachtigallenweg 50
2070 Ahrensburg
Allgemeine Clubinfo:
Erfahrungsaustausch, Hilfe für Ein- und Umsteiger, Zeitschriftenbibliothek, verbilligter Hard- und Software-Einkauf durch

Sammelbestellungen.
Clubzeitung: Vierteljährlich
Beitrag: 40.- DM jährlich

GAUG
German Amiga User Club
z.Hd. Herrn Krumrey
Hansestr. 124
2400 Lübeck 1

Allgemeine Clubinfo:
Alles über und auf dem Amiga.
Clubzeitung: "Artistline", Zweimonatlich.
Beitrag: Keinen.

SCORA

Computerclub

Postfach 1724
2960 Aurich
Allgemeine Clubinfo:
Fachbibliothek, Hilfen bei Hardwareproblemen, Informationsaustausch
Clubzeitung: unbekannt
Beitrag: unbekannt

Postleitzahl 3000

(Da sag noch einer was über die Berliner)

Postleitzahl 4000

A.U.G.E. 4000

c/o U.Tempelmann
Lochnerstr. 24
4030 Ratingen
Tel. 02102/23371
Allgemeine Clubinfo:
Messebesuche, Public Domain Pool, Hardwarebasteleien, abendliche Feierlaune.
Clubzeitung: Momentan keine.
Beitrag: Unbekannt

PANORAMA Computer Club (PCC)

Am Thelenkusch 123
4100 Duisburg 26
Tel. 0203 763920
Allgemeine Clubinfo:
Verschiedene Compu-

IM ÜBERBLICK

tertypen, Erfahrungsaustausch, Hilfen für Einsteiger, Aufbau einer PD-Bibliothek, demnächst Clubeigener Amiga vorhanden. Clubzeitung vorhanden
Beitrag: 36.- DM pro Jahr

AEC

Allgemeiner Essener Computerclub
Gladbeckerstr.4
4300 Essen 1
Tel. 0201/36846
Allgemeine Clubinfo:
Wöchentliches Treffen, Erfahrungsaustausch, Hilfen für Ein- und Umsteiger, Messebesuche, andere Computertypen.
Clubzeitung: RAM
Beitrag: 36.- DM jährlich

IEC

Information Exchange Club eV
Peter Rüschoff-Nadermann
Stodtbrockweg 13
4400 Münster-Roxel
Allgemeine Clubinfo:
Erfahrungsaustausch, andere Computertypen, eigene Mailbox mit den üblichen Übertragungsgeschwindigkeiten und Parameter 8N1, Tel. 0251 - 25422
Clubzeitschrift: unbekannt
Beitrag: unbekannt

CCQ

Computerclub Quakenbrück
Christian Leyer
Kiwittstr. 10
4570 Quakenbrück 01

Allgemeine Clubinfo:
Erfahrungsaustausch, Kurse in BASIC, 6502 & 68000-Assembler, C,Pascal und Hardware, Hardwarewerkstatt, eigene Reparaturen, Klönabende, auch andere Computertypen.

Clubzeitschrift: CCQ-News, erscheint alle zwei Monate
Beitrag: 36.- DM jährlich

AIT

c/o Magnus Rönn
Erlenkamp 13
4650 Gelsenkirchen
Allgemeine Clubinfo:
Informationsaustausch, Hilfen für Ein- und Umsteiger, eigenes PD-Disketten-Magazin GETIT.
Clubzeitung: GETIT
Beitrag: unbekannt

Carsten Rose

Leostr. 35
4790 Paderborn
Tel. 05251-27323
Allgemeine Clubinfo:
lockere Amiga-User, Meinungs-austausch, gemeinsames Programmieren
Clubzeitschrift: keine
Beitrag: keinen

Postleitzahl 5000

IHSUG

International Hard- und Software
User Group
Stichwort: Mitglieder
Gertenstr. 4
5547 Hallschlag/Eifel
Tel. 06567-520
Allgemeine Clubinfo:
Untergruppen in Los Angeles, New York, München, Tokio, Köln, Vereinssitz in West-

Berlin, andere Computertypen, Programmieren in allen gängigen Programmiersprachen.
Clubzeitschrift: keine
Beitrag: unbekannt

CCH

Computer Club Hallingen

Uwe Krumscheid
Ardresweg 4
5750 Menden 1
Allgemeine Clubinfo:
Eigene Software, Erfahrungsaustausch, Public Domain Pool, Problemfragenservice, alle Amiga-Modelle.
Clubzeitung vorhanden
Beitrag: 2.- DM monatlich

Public Projekt

Dechant-Röper Straße 32
5750 Menden 1
Tel. 02373/10225
(Samstag von 16 - 18 Uhr)
Allgemeine Clubinfo:
Software und Erfahrungsaustausch, gemeinsamer Vertrieb selbsterstellter Software, alles für Commodore Amiga und Schneider CPC.
Clubzeitung vorhanden.

Postleitzahl 6000

YING-YANG Club

Nick Dollhausen
Lenastr. 38
6000 Frankfurt 1
Tel. 069/554037
Allgemeine Clubinfo:
Alles über und auf dem

AMIGA.

Clubzeitung erscheint monatlich.

Beitrag: 2.50 DM pro Monat

Rüsselsheimer Computer Verein 1986 e.V.

Helmut u. Angelika Koch
Holbeinstr.3
6090 Rüsselsheim
Tel. 06142/562449
Allgemeine Clubinfo:
Programmieren in BASIC und C, Erfahrungsaustausch, regelmäßiges Treffen, Wettbewerbe.
Clubzeitung vorhanden.
Beitrag:
5.- DM Erwachsene
3.- DM Jugendliche

16 Bit Computer Club Steinau/Schlüchtern

c/o Dieter Heilinger
Pfingsbornstr. 8
6497 Steinau 1
Tel. 06663 - 5865
Allgemeine Clubinfo:
Zweiwöchentliche Treffen, Erfahrungsaustausch, gemeinsames Programmieren, Public Domain Bibliothek
Clubzeitschrift: keine
Beitrag:
50.- DM Jugendliche
100.- DM Erwachsene

Postleitzahl 7000

IGMC Ermstal
Interessengemeinschaft Microcomputer
Kernerstr.2
7432 Bad Urach
Tel. 07125/8238 (Markus Besch)
Allgemeine Clubinfo:
Seminare, gemeinsamer Besuch von Ver-

staltungen, Clubtreffen 1 mal pro Woche, verschiedene Computertypen.

Clubzeitung alle 2 Monate
Beitrag: 2.- monatlich

Postleitzahl 8000

Michael Obennuir
K-Steinhausen-Str. 4
8081 Althegnenberg
Allgemeine Clubinfo:
Programme schreiben, billiger Großverkauf, eigene Olympiade, keine Raubkopien.
Wir gehen davon aus, daß andere Clubs diesem Beispiel folgen. (Anm. der Redaktion)
Clubzeitung in Planung
Beitrag:
Eintritt 5.- DM, sonst monatlich 1.- DM

TKC

Tom Seidel
Herrenstr. 24
8360 Deggendorf
Tel. 0991/23693
Allgemeine Clubinfo:
Software und Erfahrungsaustausch, Treffen, Messebesuche, Hardwarebasteleien, Hilfe für Ein- und Umsteiger, Zeitschriften und Bücherbibliothek, Problemlösungen, gemeinsamer Computereinkauf.
Clubzeitung: in Vorbereitung
Beitrag: keinen

NEUESTE

vom Spielmarkt

RELIN

Unter diesem Label sind bereits einige bekannte Spiele für den Amiga und andere Rechner erschienen, in naher Zukunft sind weitere Veröffentlichungen geplant, unter anderem: Window Willy. Willy ist Fensterputzer und muß sich hocharbeiten. Daß dies nicht ohne Hindernisse von statten geht, versteht sich von selbst. Beispielsweise muß Willy Blumentöpfen oder Aktenkoffern ausweichen. Auch die Tauben machen ihm das Leben schwer, denn sie hinterlassen leider allzu hartnäckigen Dreck auf den Fensterscheiben. Schließlich schwört sich auch das

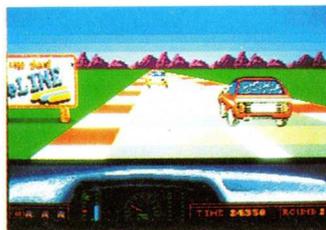


Willy beim alltäglichen Fensterputzen.

Wetter gegen Willy. Erste Testversionen lassen einiges erwarten.

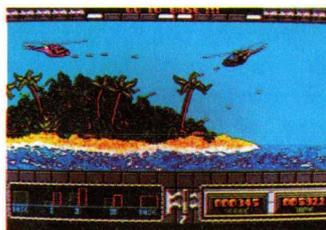
Track On heißt eine Rennsimulation, die in enger Zusammenarbeit mit RELINE Software und einem Softwareteam in Israel entsteht. Auf Realitätsnähe wird besonde-

ren Wert gelegt. Man darf gespannt sein.



Erste Bilder von der Rennsimulation Track On.

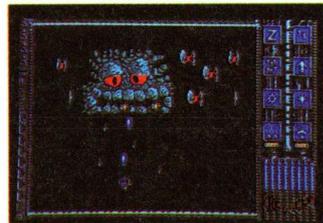
In die Karibik versetzt Dyter-07 den Spieler. Dyter-07 ist hierbei die Typenbezeichnung eines neuartigen Hubschraubersystems. Auf fünf Inseln spielt sich das Geschehen ab und ist eine Kombination von Action- und Strategieelementen. Gesplitteter Bildschirm, viele Sprites, gute Soundeffekte sollen Dyter-07 auszeichnen.



In die sonnige Karibik wird der Spieler mit Dyter-07 versetzt.

BETTER DEAD THAN ALIEN

Unter diesem makaberen Titel vertreibt ein englisches Softwarehaus eine weiteres Ballerspiel. In 25 verschiedenen Levels muß wieder einmal die Zivilisation von den bösen Aliens bewahrt werden



Ballern was das Zeug hält ist auch bei BETTER DEAD THAN ALIEN erforderlich.

(gäh!). Dem Spieler steht zur Verteidigung ein Raumgleiter zur Verfügung, der mit verschiedenen Zusatzoptionen ausgerüstet werden kann. Die Grafik bietet nichts Besonderes, ist aber nicht als schlecht zu bezeichnen. Baller-Freunde können sich auf die nächste Shoot'em Up-Variante vorbereiten.

SARCO-PHASER

Aus dem Hause Rainbow Arts stammt Sarcophaser. Es handelt sich um ein abwechslungsreiches Ballerspiel. Das Spielfeld scrollt von links nach rechts, wobei das eigene Raumschiff frei über den Bildschirm bewegt werden kann. Die verschiedenen Level sind mit unterschiedlichen Hintergrundgrafiken versehen, was die ganze Sache nicht so schell langweilig werden läßt. Vielfältige Gegner gilt es zu überwinden, auch geschicktes Steuern des Raumgleiters ist in manchem Level unumgänglich. Konnte eine Angriffsformation abgewehrt werden, erscheinen verschieden farbige Punkte



Abwechslungsreiche Grafik läßt Sarophaser nicht monoton wirken.

auf dem Bildschirm, die eingesammelt werden sollten. Der Fighter erhält dadurch

besondere Optionen, die das Weiterkommen enorm erleichtern können. Die Grafik von Sarcophaser ist recht gut, obwohl hin und wieder 'langweilige' Sequenzen auftauchen. Der Sound ist dem Ami-

ga gerecht, bietet aber nichts weltbewegendes. Sarcophaser hebt sich aus der Masse von Ballerspielen ab. Fans dieses Genres sind mit dem Spiel gut bedient.

AAARGH...

...nein, kein entnervter Schrei des Redakteurs, sondern vielmehr stellt AAARGH den Titel des neuen Spiels von Melbourne House dar. AAARGH soll wohl den Kampfschrei eines Monsters ausdrücken. Der Spieler kann wahlweise einen Zyklopen oder einen Drachen steuern. Aufgabe des Spielers ist es mit seinem riesigen, feuerspuckenden Monster alle Gebäude (auf dem Bildschirm) dem Erdboden gleich zu machen, um an das gesuchte ROC EI zu gelangen. Desweiteren findet das

Monster in den zerstörten Gebäuden allerlei Essbares, oder neue Energie um Feuerblitze abzuschießen. Leider sind die Bewohner der zahlreichen Städte nicht gewillt, sich ihre Häuser so mir nichts dir nichts zerstören zu lassen und erwehren sich mit Hilfe von Katapulten und ähnlichem, wilde Tiere eilen den Bewohnern zu Hilfe. Ist das ersehnte ROC EI gefunden, muß es schleunigst aufgesammelt und in die Schatzhöhle gebracht werden; Neider gibt es aber dort ebenfalls.

STARKE SPRACHAUSGABE, SCHWACHES SPIEL

Töten oder getötet werden. So lautet wieder einmal das Motto zum neuesten Ballerspiel von RED RAT mit dem Titel SCREAMING WINGS. Wie kann es anders sein, der Feuerknopf ist des Spielers wichtigstes Mittel, um seiner vielen Feinde habhaft zu werden. Diesmal findet das Gemetzel aber nicht im Weltraum statt. Mutter Erde ist Austragungsort des Kampfes. Der Spieler fliegt mit einem Flugzeug bei horizontalem Scrolling über eine Landschaft und ballert dabei auf entgegenkommende Jets, was der Joystick hergibt.

Manchmal besteht die Möglichkeit, Bonussymbole einzusammeln, die dem Flugzeug angenehme Extras wie zusätzliche Feuerkraft oder ähnliches bieten. Während des Geschehens kann man mit Krämpfen in den Fingergelenken rechnen (der Daumen ist besonders gefährdet!). Wenn das ganze wenigstens noch Spaß machen würde...

Eine langweilige Hintergrundgrafik in Dunkelgrün, vermischt mit einem Hauch Grün und ein paar Pfützen Blau, sorgt dafür, daß der hartgesottenste Reißaus



Bei AAARGH muß alles kurz und klein gehauen werden, um an das ersehnte ROC EI zu gelangen.

All zu ernst darf man AAARGH nicht nehmen, moralische Zweifel an den Aktionen der Monster können bei dem einen oder anderen auftreten, auch ist 'das Ende' eines Monsters recht makaber. Die Grafik ist gut, der Sound ist dem Spiel sehr gut angeglichen, er stimmt den Spieler in das zerstörerische Geschehen ein. Zumindest stellt AAARGH eine auf dem Amiga neue Spielvariante dar und wird sicher seine Freunde finden.

nimmt. Auf eine gute Hintergrundgrafik könnte man zur Not noch verzichten, wenn das Gameplay einigermaßen gut gelöst wäre. Denkste! Solange sich nicht mehr als vier Flugzeuge auf dem Bildschirm befinden, bewegt sich die Grafik in einer akzeptablen Geschwindigkeit. Kommen jedoch mehr als vier auf die Bildfläche, ruckelt die Grafik, als hätte man es mit einem Erdbeben zu tun. Die Schüsse des Flugzeuges hätte man auch besser darstellen können. Weiße Striche, die auf Knopfdruck aus der Spitze des Flugzeuges schnellen, sind nicht sehr überzeugend. Positiv fiel beim Test die digitalisierte Sprachausgabe auf. Ansonsten handelt es sich bei Red Rat's neuester Missetat um einen absoluten Flop. Screaming Wings ist ein Spiel, das keine Anforderungen an den Verstand stellt, sondern nur an einen durchtrainierten Daumen. Ein Spiel für geistige Tiefflüge.

PARADOX

Ein neues Softwarehaus macht von sich Reden: PARADOX. Gleich vier Spiele wurden zum Test vorgelegt. Alle ließen sich ohne Schwierigkeiten starten und kamen sogar ohne zu murren mit Speichererweiterungen zu recht...

PROTEKTOR...

...besitzt entfernte Ähnlichkeit mit Defender (die ältere Computergeneration wird sich vielleicht noch erinnern?). Schon beim Betrachten der Titelgrafik kommt man nicht umhin, nostalgische Gefühle zu entwickeln. Man glaubt vor einem C64 oder gar VC20 zu sitzen. Der Sound harmoniert übrigens ganz hervorragend mit der Grafik. Haben die Programmierer vielleicht sogar unter Ausnutzung aller Hardwarefähigkeiten des Amiga den Originalsound des C64 gesampelt? Auch das Scrolling ist so, wie wir es aus den Anfangsjahren der Homecomputer gewohnt sind. Es scheint sich hier um zeichenweise verschobene Blockgrafik zu handeln. Wer aufgrund angespannter Finanzlage auf den 64er-Emulator verzichten mußte, wird mit diesem Programm für nostalgische Zwecke sicherlich gut bedient sein.

SPACE STATION...

... erinnert an die altbekannten Hüpf- und Springspiele. In bezug auf die Qualität gilt uneingeschränkt das schon über PROTECTOR Gesagte. Zu erwähnen wäre noch, das hier auch noch ein Hauch vom guten alten ZX-81 mit-schwingt.

FIRE-BLA-STER...

...ist eine stark abgemagerte Phalanx-Variante (siehe Test Mai 1987). Auch hier waren wir versucht, es als C-64-Emulation mit aufgepfropftem Programm zu bezeichnen. Das wäre aber eine Beleidigung der Fähigkeiten des

Comodore-64. Im Vergleich mit REVENGE OF DOOM von Cachet kostet es das Dreifache, ist dafür aber weniger als halb so gut. WARZONE ist die Karikatur des alten C-128 Spiels THE LAST V-8. Die Konvertierungsnote müßte allerdings '5-' lauten.

Nachdem die Tests unter Aufbietung aller verfügbaren Kräfte hoffentlich ohne bleibende Schäden für den Tester überstanden waren, wurde auch der tiefere Sinn des Firmennamens klar. Es mutet wirklich paradox an, wenn jemand für derartig schlecht gemachte Programme auch noch 9.99 englische Pfund (ca. 32,- DM) verlangt. Zum Glück werden diese Spiele zur Zeit nur in Großbritannien vertrieben.

Was uns erstaunt hat, ist aber besonders die Tatsache, daß PARADOX diese Spiele überhaupt auf den Markt bringt. Derek, unser Kollege aus England hat mittlerweile die Firma besucht und festgestellt, daß im Herbst einige recht ansehnliche Produkte von PARADOX erscheinen

werden. Die Gefahr, daß sich bis dahin ein sehr negatives Image aufgebaut hat und demzufolge niemand mehr etwas von dieser Firma wissen will, ist bei einer Markteinführung mit solchen Produkten recht groß.

SO-DAN, DER UNBESIEGBARE

Discovery Software arbeitet gerade an einem Programm, das enorm große Bobs verwendet - der Arbeitstitel SODAN - DER UNBESIEGBARE. Bis zur Endgültigen Fertigstellung wird der Amiga-Spielefreak noch ca. ein Monat warten müßen, bis er das Actiongames ins Laufwerk schieben kann. Spielbare Vorabversionen stehen uns

allerdings bereits jetzt zur Verfügung. Das Spiel, SODAN - DER UNBESIEGBARE, besticht durch ausgesprochen große Bobs, die fast den ganzen Bildschirm einnehmen. Die Animation dieser großen Bobs ist sehr gut gelungen, auch sind die Hintergrundgrafiken vom Feinsten. Der Sound ist digitalisiert und ist den verschiedenen Levels angepaßt. Einige Levels sind bereits fertiggestellt und sind spielbar. Auf jeden Fall darf man Gespannt sein



In dunklen Grabkammern gilt es gefährliche Greise sich vom Leib zu halten.

Prg. für alle AMIGA-Modelle - Exzellent in Struktur, Grafik, Sound - alle Prg. in Deutsch -

ASTROL. KOSMOGRAMM
- Nach Eingabe von Namen, Geburtsort (geografischer Lage) und Geburtszeit werden errechnet: Sternzeit, Aszendent, Medium Coeli, Gestirne im Tierkreis, Häuser nach Dr. Koch/Schäck (Horoskop-Daten mit Ephemeriden. Außer dem Bildschirmdisplay kann Ausdruck auf 2 DIN A4-Seiten erfolgen; davon 1/2 Seite allgemeines Persönlichkeitsbild mit Partnerschaftskriterien und 1/2 Seite Tierkreisdiagramm (Horoskop). Alle Planeten mit Sonne und Mond. Für alle Berufs- und Hobby-Astrologen eine unentbehrliche Arbeitserleichterung. **78,-**

BACKGAMMON **68,-**

BIOKURVEN
Zur Trendbestimmung der Bio-rhythmen und des seelisch-/geistig-/körperlichen Gleichgewichts mit Druck des Kurvendiagramms von oben nach unten in beliebiger Länge. In der rechten Blatthälfte das Diagramm, links eine Auswertung des Gesamtpotentials für jeden Tag. Werte für bestimmte Tage auch auf dem Bildschirm. Ausführliche Beschreibung der wissenschaftlichen Grundlagen. Ideal für Partnervergleiche. **58,-**

GESCHÄFT
- Bestellung, Auftragsbestätigung, Rechnung, Lieferschein, Mahnung, 6 Briefrahmen mit Firmendaten zur ständigen Verfügung (Anschrift, Konten usw.,

Menge/Preis, Rabatt/Aufschlag, MwSt., Skonto, Verpackung, Versandweg usw.) Mit Einbindung von abgespeicherten Adressen und Artikeln. **198,-**

GELD
- Man wählt mit der Maus unter 25 Rechenroutinen in den Bereichen: Anlage - Kapital - Vermögensbildung - Rentensparen - Rendite - Lasten - Zinsen/Zinseszinsen - Kredit - Hypotheken - Laufzeit - Amortisation - Ratenzahlung - Wertverlust - Nominal- und Effektivzinsen - Ausdruck vollständiger Tilgungsraten - Diskontierung - Devisen/Sorten - Konvertierung **98,-**



I. Dinkler
Am Schneiderhaus 17 · D-5760 Arnsberg 1
Tel. 0 29 32/3 29 47

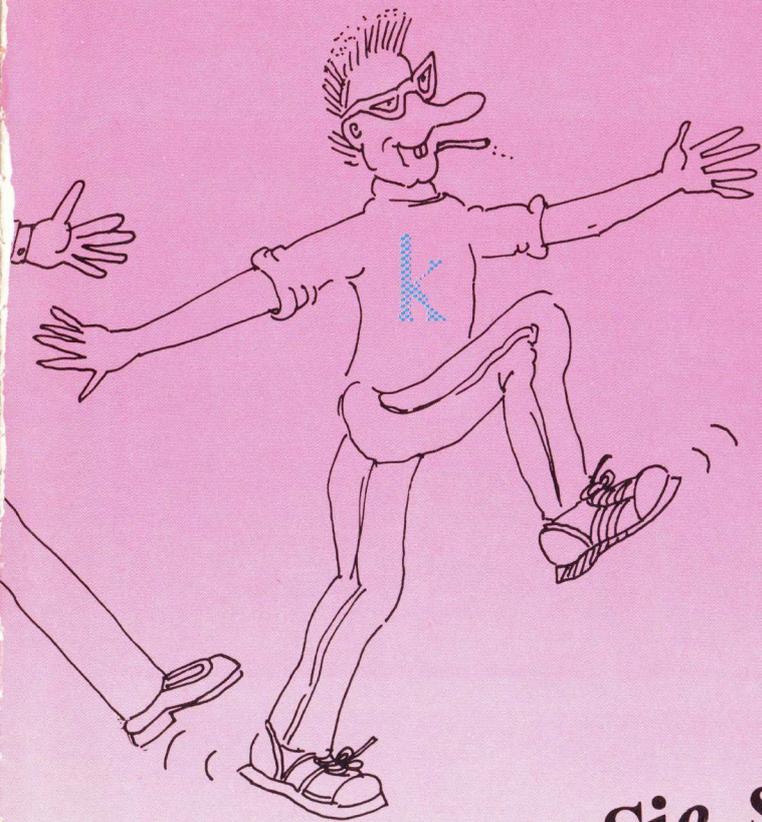
KALORIEN-POLIZEI - Nach Eingabe von Größe, Gewicht, Geschlecht, Arbeitsleistung erfolgt Bedarfsrechnung und Vergleich m. d. tatsächlichen Ernährung (Fett, Eiweiß, Kohlehydrate), Idealgewicht, Vitalstoffe, auf Wunsch Ausdruck. **58,-**

Inventur, Fibu-gerecht 118,-
Provisionsabrechnung 118,-

Dateiverwaltungen:

Adressen	69,-
Bibliothek	88,-
Diskothek	78,-
Dokumente	98,-
Exponate	118,-
Galerie	118,-
Lagerartikel	88,-
Museum	118,-
Personal	88,-
Videothek	78,-

usw. usw. - Fordern Sie mit Freiumschlag unsere Liste an!
Im Computer-Center oder bei uns zu obigen unverbindlich empfohlenen Preisen + DM 3,- bei Vorkasse oder DM 4,70 bei Nachnahme



AMIGA

K. Schneider / O. Steinmeier

GRUNDLEHRGANG

Der richtige Einstieg

Heim Verlag

Holen Sie sich auch
den neuen

AMIGA-GRUNDLEHRGANG

DM 59,-

Buch und Diskette

unverbindlich empfohlener Verkaufspreis

WICHTIGE MERKMALE:

★ Das Buch für den richtigen Einstieg mit dem Commodore AMIGA ★ Auf über 400 Seiten werden dem Leser leicht verständlich die Grundlagen der Computertechnik und der Umgang mit Hardware erklärt ★ Ein ausführlicher Hauptteil ist dem Einsatz der grafischen Benutzeroberfläche des Betriebssystems gewidmet. Hier erläutert das Buch Fenster, Pull-down-Menüs und die vielen anderen Teile der Workbench ★ Wer die Maus nicht mag, der kann aus dem Kapitel über den **Command Line Interpreter (CLI)** entnehmen, wie man den AMIGA auch ohne Maus einsetzen kann ★ Ein weiterer Bereich des Buches ist die Einführung in die Programmiersprache **BASIC**. Eine umfangreiche Befehlsübersicht sowie einige interessante Programme dienen der Erlernung und dem guten Training von BASIC ★ Anhänge wie z. B. ein **Index** und eine **Sachworterklärung** bieten das schnelle Nachschlagen und Auffinden wichtiger Punkte ★ Mit dem Buch erhalten Sie eine **Programmdiskette** mit allen abgedruckten Listings. Damit können die Beispielprogramme ohne die Mühe und Arbeit des Eintippens auf dem Computer nachvollzogen werden.

AUS DEM INHALT:

1. Die Hardware des AMIGA
 ★ die versch. AMIGA-Modelle ★ die Diskettenstation ★ Anschluß eines Druckers ★ Monitore am AMIGA ★ Erweiterung des AMIGA-Systems ★ Einstieg in die MS-DOS Welt mit dem AMIGA ★ Die „Innereien“ des AMIGA (RAM, ROM u. Prozessoren)
2. Das Betriebssystem des AMIGA
 ★ Betriebssysteme und ihre Bedeutung ★ Die Benutzeroberfläche des AMIGA ★ Steuerung der Workbench ★ Arbeiten mit Maus, Fenstern und Pull-down-Menüs ★ Verwendung von Disketten, Dateien, Directory Die Programme der Workbench Diskette im Einzelnen ★ Der CLI und seine Bedienung ★ Kopieren, Löschen und Batch-Bearbeitung im CLI
3. Programmieren in Amiga-Basic
 ★ Die Bedienung des Basic-Interpreters ★ Variable in Basic ★ Schleifenstrukturen ★ Die IF-Abfrage ★ Prozeduren zur Programmstrukturierung ★ Graphik-Programmierung in AMIGA-BASIC ★ Dateiverwaltung ★ ausführliche Befehlsübersicht mit detaillierten Erklärungen
4. Zum Training
 ★ Programm-Diskette mit allen abgedruckten Listings ★ Sachworterklärung (Fachwörter-Lexikon) ★ Ausführlicher Index (Stichwortverzeichnis mit entspr. Verweisen)

BESTELL-COUPON

an Heim-Verlag
Heidelberger Landstraße 194
6100 Darmstadt-Eberstadt

Ich bestelle _____

zzgl. DM 5,- Versandkosten (unabhängig von der bestellten Stückzahl)

per Nachnahme Verrechnungsscheck liegt bei

Name, Vorname _____

Straße, Hausnummer _____

PLZ, Ort _____

Benutzen Sie auch die in KICKSTART vorhandene Bestellkarte

Heim Verlag

Heidelberger Landstraße 194
6100 Darmstadt-Eberstadt
Telefon 0 61 51 - 5 60 57

TOP 12

PORTS OF CALL entwickelt sich zum Dauerhit, doch der Zweitplatzierte THE GREAT GIANNA SISTERS könnte ihm in Kürze Konkurrenz machen, immerhin hat sich dieses Spiel um sechs Plätze verbessert. Auf den Plätzen 3 bis 5 sind alte Bekannte zu finden - ARKANOID, THE BARD'S TALE und SHANGHAI. Auf Platz 6 machte sich der Aufsteiger der Woche breit: JET. Im letzten Monat noch nicht platziert, gelang ihm ein beachtlicher Sprung nach oben. Vielleicht ist dieser neue Flugsimulator der Grund dafür, daß der Veteran FLIGHTSIMULATOR II plötzlich auf Sturzflug geht. Auch EMERALD MINE geht es nicht besser, wogegen sich PINK PANTHER langsam nach oben bewegt.

Neu in der TOP 12 sind OBLITERATOR (siehe Vorstellung in dieser Ausgabe) und TETRIS, das interessante Reaktionsspiel. Das Ende der Hitliste bildet wiederum ein alter Bekannter, der mittlerweile starke Einbußen einstecken mußte.

Auch diesmal sollten Sie bei der TOP 12 teilnehmen, denn es gibt wieder 12 mal den TOP 12-TIP zu gewinnen. Dies ist in der Regel ein brandneues Spiel, das bei der Redaktion besonderen Gefallen gefunden hat. Diesmal verlosen wir das Spiel WIZBALL, das besonders durch seine neuartige Spielidee aufgefallen ist. Lassen Sie sich überraschen, welches Spiel im nächsten Monat zu gewinnen ist - und vergessen Sie nicht, Ihre Postkarte rechtzeitig einzuwerfen.

Mitmachen kann jeder, der die TOP 12-Postkarte (bitte keine andere!) ausfüllt und uns zurückschickt. Der Rechtsweg ist dabei ausgeschlossen. Einsendeschluß ist der 30. Juni 1988. Karten, die nach diesem Termin eintreffen, werden im nächsten Monat berücksichtigt.

Wir gratulieren den Gewinnern dieses Monats, die jeweils das Spiel WIZBALL erhalten:

Falco Kirschbein, Bad Krozingen/ Christian Gaul, Stadtlauringen/ Thomas Massmeyer, Karlsruhe/ Adam Moszczynski, Berlin/ Anne Beschl, Herford/ Ulrich Bauer, Pfarrkirchen/ Edgar Spalt, Heilbronn/ Gitte Heise, Osterode/ Joachim Merz, Zeilsheim/ Roland Altschöfle, Nürnberg/ Heinz Banowski, Pöding/ Kai Jost, Homburg

1. (1) Ports of Call



2. (8) The Great Gianna Sisters

3. (2) Arkanoid

4. (5) The Bard's Tale

5. (7) Shanghai

6. (-) JET

7. (4) Emerald Mine

8. (9) Pink Panther

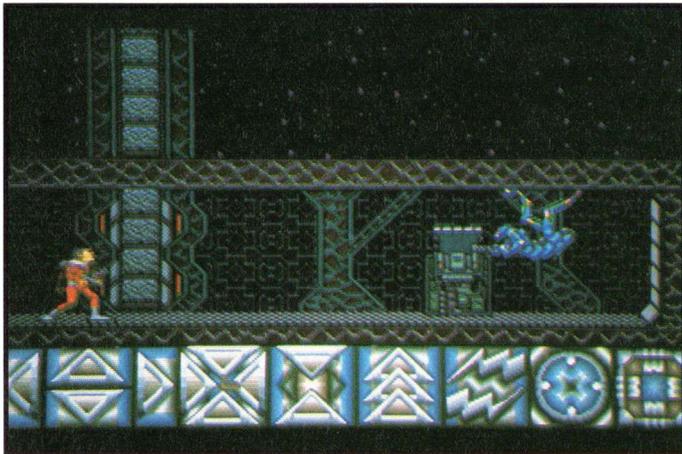
9. (3) Flightsimulator II

10. (-) Obliterator

11. (-) Tetris

12. (6) Barbarian

OBLITERATOR



Obliterator besticht durch sehr gute Grafik.

Psygnosis ist sicher für viele schon ein Begriff; bekannt geworden ist das englische Sopsftwarehaus durch das legendäre Spiel Barbarian und das neuere Terrorpods. Mit Obliterator bringt Psygnosis nach diesen beiden Spielen ein neues Produkt auf den Markt. Der Käufer findet neben einen englischsprachigen Handbuch mit Hintergrundgeschichte, Erklärung der Steuerung und der Aufgabenstellung wieder einmal ein Poster, das eine Aktion-Einlage zeigt.

Nach Einlegen der Diskette erscheint, wie bei Barbarian und Terrorpods, zunächst das Firmenlogo und daraufhin eine sehr gut animierte Sequenz. Der Held (der Obliterator) schmunzelt und grinst den Spielefreund an, bevor er genüßlich das futuristische Gewehr auf ihn richtet und hemmungslos abdrückt. Etwas makaber nach meinen Geschmack. Danach erscheint ein Introscreen mit einem sehr gut gezeichneten Bild, das doch sehr bekannt wirkt - Barbarian läßt grüßen.

BARBARIAN LÄßt GRÜßEN

Nach einem Tastendruck findet man sich in einem feindlichen Sternenkreuzer wieder. Wie bei Barbarian ist das untere Drittel des Bildschirms als Steuerkonsole für den Obliterator vorgesehen, und bis auf zwei Ausnahmen be-

sitzt diese Konsole dieselben Funktionen wie die von Barbarian. Die Unterschiede: der Held ist nicht mehr in der Lage, langsam zu laufen, demzufolge fehlt dieses Feld; des weiteren kann er dazu bewegt werden, eine Bodenrolle zu vollführen. Mit Space schaltet man wie gewohnt in ein weiteres Menü um, hier erfährt der Spieler alles über seinen Score, den Energie- und Munitionsvorart sowie die Waffen, die er besitzt. Vier verschiedene Waffen kann der Held in den vielen Räumen des Schiffs finden; diese müssen natürlich gezielt eingesetzt werden, denn einige Monster sind gegen so manches Projektil gefeit. Apropos Monster, der Spieler stößt im Verlauf des Spiels auf eine ganze Reihe der unterschiedlichsten Bösewichter, einige sitzen in einer Art Raumgleiter und machen ihm das Leben schwer, andere gebären erst die bösen Unholde.

Obliterator kann auch abgespeichert werden, dazu ist es jedoch notwendig, daß unser Held in einem Schott verschwindet. Die Schotts sind aber nicht allzu zahlreich, so daß der Spieler die wenigen, auf die er stößt, ausnutzen sollte.

Die Grafik von Obliterator gehört zum den Feinsten, was auf dem Amiga bisher zu sehen war, lediglich die Sprites könnten ein wenig besser animiert sein, denn des öfteren ruckeln sie etwas über den Bildschirm. Der Sound hat

sich gegenüber Barbarian gebessert. Schon während des Ladevorgangs ertönt der Amiga-Lautsprecher. Beim Spiel ist leider nur eine gute Hintergrundmusik zu vernehmen, der Held bleibt hingegen stumm, ebenso seine Waffen.

FAZIT

Obliterator ist sehr stark an Barbarian angelehnt. Die Steuerung ist beispielsweise fast identisch mit dem Vorgänger, ebenso wie die Spielhandlung. Die grafische Gestaltung des Spiels ist aber wieder vom Feinsten. Der Sound kann abgeschaltet werden, was gut ist, da er nach einer gewissen Zeit die Nerven strapaziert. Wer gerne Barbarian gespielt hat, der wird mit Obliterator mehr als zufrieden sein. Wem Barbarian nicht gelegen hat, der kann auch mit dem neuen Psygnosis-Programm nichts anfangen.

(AK)

Preis: ca 80.- DM



REISENDE IM WIND



Dieses Abenteuerspiel entstand nach dem gleichnamigen fantastischen Comic von François Bourgon. Die sehr schönen und detaillierten Bilder des Comics, der mit rund drei Millionen verkauften Exemplaren ein Weltbestseller ist, wurden mit größter Sorgfalt auf den Computer übertragen und übertreffen damit alle anderen Produkte.

“Es gibt Abenteuer, die kann man nicht erzählen - man muß sie erleben.” Nach diesem Motto sind die Aktionen des Spiels gestaltet. Es ist nicht nötig (und sogar unmöglich), etwas auf der Tastatur einzugeben. Der Handlungsablauf entsteht vielmehr dadurch, daß zwischen bestimmten Personen ein Gespräch hergestellt wird. Die handelnden Figuren werden dazu aus dem Szenenbild geklickt und erscheinen daraufhin im Porträtfenster. Zwischen den Personen kann beliebig hin- und hergeschaltet werden, so daß es zu einer Handlung kommt. Dabei ist vor allem der zeitliche Ablauf wichtig, der durch die Personenwahl beeinflusst wird. Von Zeit zu Zeit muß jedoch auch eine direkte Entscheidung getroffen werden. Dann stehen zwei Antworten zur Auswahl, und nur eine führt zum Fortgang der Handlung.

Ort der Handlung ist Frankreich, am Ende des 18. Jahrhunderts. Agnes wuchs zusammen mit einem ihr aufs Haar gleichenden Waisenkind, Isabeau, bei einer Gouvernante auf. Als ihr Vater sie nach fünf Jahren der Trennung dort abholen will, tauscht Agnes zum Spaß ihre Rolle mit Isabeau. Doch aus Spaß wird Ernst; der Vater fährt mit seiner vermeintlichen Tochter davon, und Agnes kommt ins Waisenhaus. Nur der Haß auf Isabeau, die diese Verwechslung nicht aufklärt, läßt sie die schwere Zeit bei den strengen Nonnen überstehen. Erst nach fünf Jahren holt Isabeau, mittlerweile von Skrupeln geplagt, Agnes aus der Gefangenschaft heraus. Doch wegen des Skandals ist es unmöglich, die Verhältnisse aufzuklären, und Agnes heißt deshalb von nun an Isa. Um der Situation zu entfliehen, gehen beide an Bord eines französischen Kriegsschiffes. Bei einem Gefecht mit englischen Schiffen kommt Isabeau auf fürchterliche Weise ums Leben. Isa kann sich zusammen mit Hoel, der ihr das Leben gerettet hat, und dem Schiffsarzt Saint-Quentin auf eine Insel retten. Dort werden sie von Engländern aufgegriffen. Hoel und Saint-Quentin kommen

auf ein Gefangenenschiff. Nur Isa erhält die Freiheit und findet eine Anstellung bei Mr. Hereford, wo sie auf Mary trifft. Die hat die Möglichkeit, für die Befreiung von Hoel zu sorgen. Das ist Stand der Geschichte, die nun fortgesetzt werden muß.

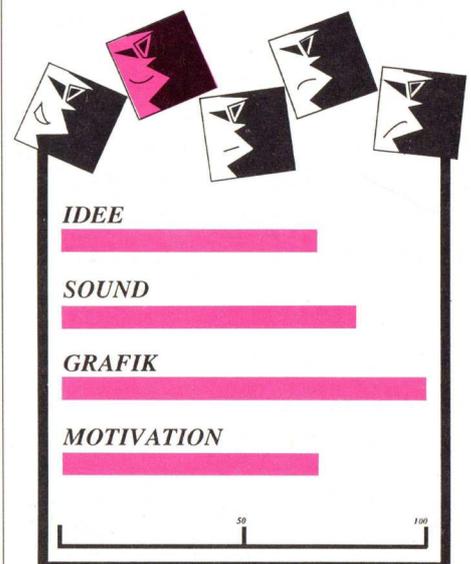
Die Serie “Reisende im Wind” ist ein sehr schöner und spannender Comic, und die Computerumsetzung steht dem nicht nach. Leicht und locker wandelt man durch die Szenen, ohne vor schwierige Aufgaben gestellt zu werden. Und wenn die Geschichte zu Ende ist, dann beginnt man nach einiger Zeit von vorne, genau wie bei einem guten Comic...

Der erste Teil hatte die ersten drei Bände der fünfteiligen Comiserie zum Inhalt. Im zweiten Teil folgt der Rest. Die spannende Geschichte um Isa, Mary, Hoel und die anderen wird in Afrika fortgesetzt, wo sie nach einer langen Überfahrt ankommen.

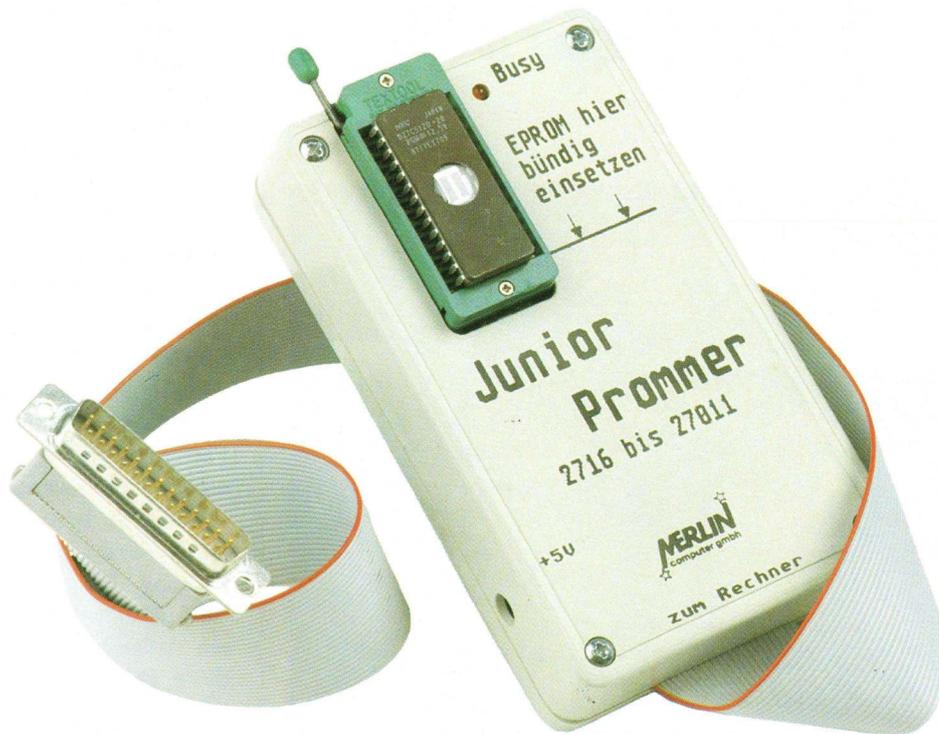
Auch Reisende im Wind 2 ist eine schöne Reise durch die atmosphärischen Bilder Bourgons, dessen Werk ein sehr großer Erfolg ist. Im Gegensatz zum ersten Teil ist jedoch die Lösung des Abenteuers etwas schwieriger. Trotzdem ist das Spiel mehr etwas “für’s Auge” als für den Geist.

(MN)

Vertrieb: ARIOLA-Soft



Klein, kompakt und leistungsstark- der JuniorPrommer



Der JuniorPrommer programmiert alle gängigen EPROM-Typen, angefangen vom 2716 (2 KByte) bis zum modernen 27011 (1 MBit). Aber nicht nur EPROMS, sondern auch einige ROM- und EEPROM-Typen lassen sich lesen bzw. programmieren.

Zum Betrieb benötigt der JuniorPrommer nur +5V, die am Joystickport Ihres Amiga abgenommen werden; alle anderen Spannungen erzeugt die Elektronik des JuniorPrommers. Die sehr komfortable Software, natürlich Menü-unterstützt, erlaubt alle nur denkbaren-Manipulationen.

Fünf Programmieralgorithmen sorgen bei jedem EPROM-Typ für hohe Datensicherheit. Im eingebauten Hex-/ASCII-Monitor läßt sich der Inhalt eines EPROMS blitzschnell durchsuchen und auch ändern.

Alles dabei!

Bemerkenswert ist der Lieferumfang, so wird z.B. das Fertigerät komplett aufgebaut und geprüft im Gehäuse mit allen Kabeln anschlussfertig geliefert.

Auf der Diskette mit der Treiber-
software befindet sich noch ein Programm, das die Kickstart-Diskette

in vier EPROM-Dateien für 27512 (64 KByte EPROM) zerlegt, die dann mit dem JuniorPrommer gebrannt werden können. Ferner wird der Source-Code für die Lese- bzw. Programmerroutinen mitgeliefert. Und last but not least ist im Bedienungshandbuch (deutsch) der Schaltplan abgedruckt.



Bestellcoupon Merlin Computer GmbH Industriestraße 26 6236 Eschborn Tel.: 06196/481811

Hiermit bestelle ich:

- Fertigerät wie oben beschrieben DM 249,00
- Leerplatine (o. Bauteile) und Software DM 59,00
- Leergehäuse (gebohrt und bedruckt) DM 39,90

Versandkosten: Inland DM 7,50

Ausland DM 10,00

Auslandsbestellungen nur gegen Vorkasse
Nachnahme zuzüglich DM 3,50 Nachnahmege-
bühr

WIZBALL



Es gilt, die graue Welt von Wiz wieder in eine farbenfrohe zu verwandeln.

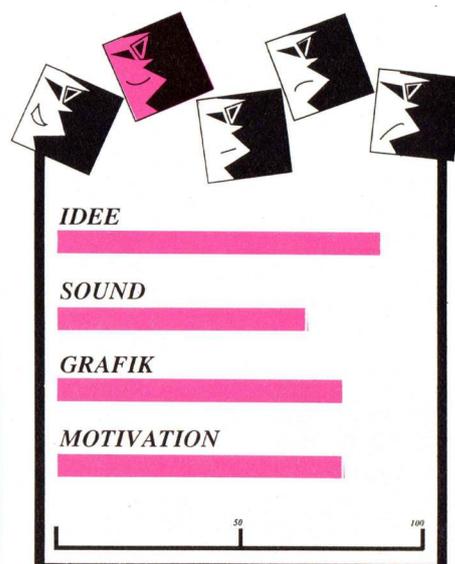
Von der 'Insel' kommt ein Spiel, das von anderen Rechnern her bereits bekannt ist. Jetzt ist es auch auf den Amiga umgesetzt - WIZBALL von OCEAN. WIZBALL liegt eine gelungene Spielidee zugrunde. In der vielfarbigem WIZwelt leben Wiz und seine phantastische Katze zufrieden und glücklich, bis Zark seine Kobolde ausschickt und die farbfrohe Welt in eine triste graue verwandelt. Wiz macht sich zur Aufgabe die farbenfrohe Welt wiederherzustellen und begibt sich in seinen Transporter, der treue Catelite begleitet Wiz bei seiner schwierigen Aufgabe. Soweit zur Rahmenhandlung. Zum Glück gibt es einige feindliche Kobolde, die sich nach Abschluß in

eine grüne Perle verwandeln, die, aufgesammelt, dem Transformer neue Optionen verleihen. Durch schnelles Links-, Rechts-rütteln des Joysticks kann die neue Option aktiviert werden. Durch Sammeln der Perlen ist es auch möglich, bessere Optionen zu erlangen. Die Welt von Wiz ist in viele Ebenen unterteilt, in denen die verschiedenen Farbchemikalien in Kessel gefüllt werden müssen, um die Welt wieder einfärben zu können. Sind die Kessel mit einer Farbe gefüllt, kommt der Spieler in eine Bonusszene. Am Anfang ist das Steuern recht schwierig, je mehr Optionen man eingesammelt hat, um so einfacher läßt sich aber Wiz steuern. Die Grafik ist ausgezeichnet, die Animation der Sprites ist ruckfrei und weich, auch das Scrolling ist vom Feinsten. Der Sound nutzt die Amiga-Fähigkeiten nicht

ganz aus, kann aber nicht als schlecht bezeichnet werden.

Bis zu vier Spieler können am Spiel teilnehmen, auch Teamwork ist möglich. In diesem Fall steuert der zweite Spieler den etwas später auftauchenden Catelite, was manchen Vorteil mit sich bringt. Beispielsweise können die Farbchemikalien wesentlich leichter eingesammelt werden.

WIZBALL besticht durch die neue Spielidee und die saubere Programmierung. Besonders das Spielen im Teamwork macht Spaß. Zunächst treten Anfangsschwierigkeiten auf, sind diese aber erst einmal überwunden, kann man durchaus vom Spiel gefesselt werden. Negativ ist lediglich, daß bei einem Scheitern alle gesammelten Optionen verlorengehen und man von vorne beginnen muß. WIZBALL wird sicher seine Freunde finden.





HEISSE PREISE



Kickstart 1.3 Umschaltung!

Endlich lieferbar für alle A500 und A2000 Besitzer!
Die Leerplatine mit Kabel und Schalter ist für DM 60,— erhältlich.

Modifiziertes Kickstart 1.3 mit Virusfinder auf vier 27512 Eproms als Programmierservice lieferbar.

Test in Amiga Special Extra Nr. 1/88

Lightspeed 1200

Das Luxusmodem für den Amiga! Anschlußfertig für alle Amigas mit RS-232-C Kabel und Treibersoftware.

- 100% Hayes kompatibel
- 300-1200 Baud
- Auto-Dial, Auto-Antwort
- Bell 212A, 1200bps, asynchron
- Bell 103, 0-300 bps, asynchron
- CCITT V.21 und CCITT V.22
- ohne FTZ-Nummer

nur **DM 398,—**

Coll-Card

Die 2 MB Epromkarte für Programmierer, Entwickler und alle die das exclusive für ihren A2000 lieben. Insgesamt mit 32 Eproms bestückbar!

Auf Wunsch auch gebrannte Eproms lieferbar.
Leerpreis der Karte beträgt

DM 498,—

Als Vorschau der nächsten Arbeit booten Sie von der Coll-Card.

Test in Amiga Special Extra Nr. 1/88

Grafikkarte

Maximale Auflösung 704 x 470 Pixel bei Darstellung von 4096 Farben gleichzeitig. Die Karte arbeitet unter anderem mit Overscan, wodurch ein völlig flimmerfreies Bild erreicht wird. Nur für A2000 B Modelle in Verbindung mit Multisync Monitoren. Der Preis für diese Karte inklusive Software und Manual beträgt

DM 1598,—

P.S. In Entwicklung ist die Karte für A1000 Modelle!
Test in Chip Nr. 6/88

Public Domain Software

Wir verfügen derzeit über das größte PD Angebot für den Amiga. Über 900 Disks vorhanden: alle Fish, RW 1-15, Panorama 70 Stück, Chiron 1-57, Auge 1-20, Faug 1-51, Kickstart 1-70, Amicus, ES-PD, ACS, TBAG, Slideshows, Sonstiges und vieles mehr.

2 Katalogdisketten mit Kurzbeschreibung aller Programme in deutsch gegen DM 5,— anfordern!

In letzter Sekunde die Bildserie RMS erhalten! Zur Zeit sind 25 Disketten erhältlich. Alle mit deutscher Anleitung, Tips usw.

AMIGA Hardware

Begrenzte Aktion! A500 DM 998,— A2000 DM 2100,—

Für A2000

- Speicherkarte 2 MB, autokonfig., voll bestückt nur DM 898,—
- Speicherkarte 8 MB, leer, mit Software und Manual nur DM 850,—
- 1 MB Chips dazu Stück nur DM 60,—

Für A1000

- Speicherkarte 1—4 MB mit 1 MB bestückt, inclusive modifizierter Kickstart und Manual nur DM 900,—
- Monitor Phillips CM 8833 Stereo nur DM 590,—
- CSS Multisync, 800 x 600 incl. Kabel nur DM 1190,—
- Star LC 10C, der preiswerte Farbdrucker nur DM 659,—
- Druckerkabel nur DM 15,—
- Colorband für LC 10C nur DM 29,—

3,5" MEDIA Diskettenbox für bis zu 160 Disks, stapelbar nur DM 45,—

Disketten 3,5" Qualitätsdisketten (2DD) eines namhaften Herstellers, neutral verpackt (Achtung Staffelpreise) ab DM 2,40,—

AMIGA Software

Programmiersprachen:

- M2 Amiga: Das Modula speziell für den Amiga, dt. und engl. Handbuch DM 349,—
- Atzteck C Developer: Ausgeliefert wird V.3.6a, mit Routine für 68881 DM 648,—

Tools:

- Quarterback: V.1.3 ist ein schnelles Backup Programm für Disk und Festplatte DM 139,—

Dateiverwaltung:

- GoAmiga Datei: Die komfortable Dateiverwaltung in deutsch für nur DM 199,—

Textverarbeitung:

- BECKERtext: Die gute und schnelle Textverarbeitung zum akzeptablen Preis von DM 199,—

Grafik:

- Photon Paint: Das 4096 Farbgrafikprogramm mit Supereigenschaften zum Preis von DM 199,—

WOLF

Computertechnik

Deipe Stegge 187 · 4420 Coesfeld

Telefon (0 25 41) 28 74

CSS

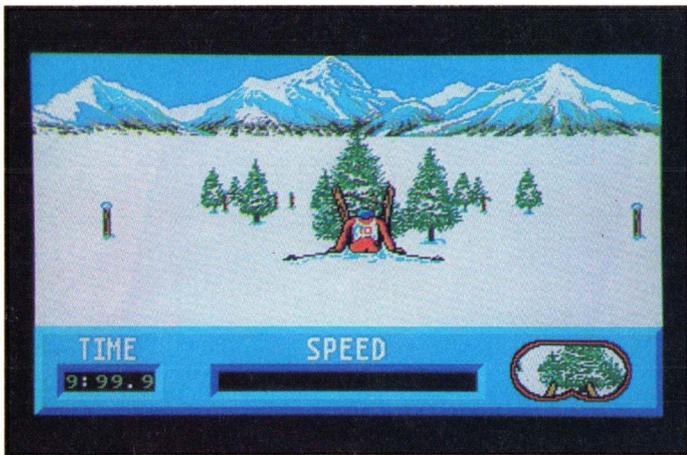
Consulting System Software

Auf der Warte 46 · 6367 Karben

Telefon (0 60 39) 57 76

SCHNELLER, HÖHER, WEITER

Winter-Olympiade '88



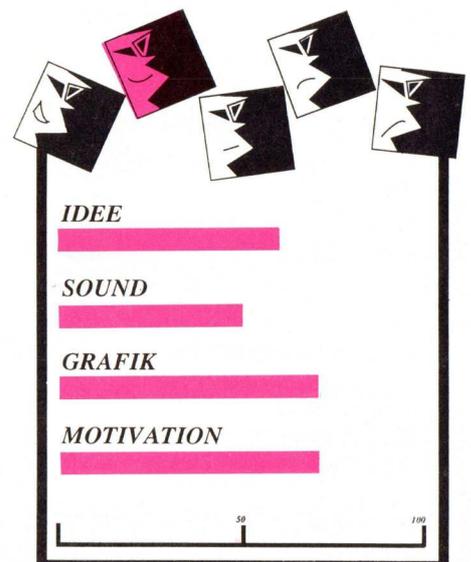
*Auch Pirmen
Zurbriggen hat Pro-
bleme beim
Abfahrtslauf.*

Die Olympiade in Calgary ist schon lange vorbei, und die Sportfans unter Ihnen werden das vielleicht bedauern. Doch mit der WINTER OLYMPIADE von TYNESOFT können jetzt einige der in Kanada gezeigten olympischen Disziplinen nachgespielt oder - vielleicht sollte man das so sagen - simuliert werden, denn Grafik und Animation des Programms sind einfach bestechend. Gewisse Ähnlichkeiten einiger Szenarien mit den WINTERGAMES von EPYX sind allerdings nicht von der Hand zu weisen, obwohl festzustellen ist, daß die Grafik von WINTER OLYMPIADE '88 doch um einiges besser wirkt und mit mehr Liebe zum Detail ausgestattet ist. Also, auf die Plätze, fertig und los geht es zur Jagd nach Medaillen und Punkten, in insgesamt fünf verschiedenen Disziplinen. Mit Abfahrtslauf, Biathlon, Skispringen, Slalom und Bobfahren stehen dabei wirklich interessante Disziplinen zur Auswahl an, die sich Dank der perfekten Animation der Athleten und des sauber programmierten Bild-

schirmscrollings sehr bald einer großen Beliebtheit erfreuen dürften. Nach jeder Runde wird ein Gesamtsieger ermittelt, und die zu spielenden Sportarten der nächsten Runde können danach neu ausgewählt werden. Neben den genannten Sportarten beinhaltet das Programm natürlich auch eine Eröffnungszeremonie. Auf eine Abschlußfeier wurde zwar verzichtet, dies sollte aber für die Beurteilung des Programms nicht weiter relevant sein. Wichtiger erscheint da die Tatsache, daß kein Trainingsmodus gewählt werden kann, aber auch daß sollte den guten Eindruck, den das Programm hinterließ, nicht sonderlich trüben. Die auf der Diskette mitgelieferten Weltrekorde sind ausnahmslos von hohem Niveau, so daß schon etwas Übung und Geschick notwendig ist, sollen die Rekordhalter von ihrer Spitzenposition verdrängt werden. Die Weltrekorde können optional eingeladen werden, so daß auch Anfängern die Möglichkeit gegeben wird, ihre Bestleistungen in eine eigene Liste einzutragen und ab-

zuspeichern. Der Sound ist recht realistisch programmiert und dient nicht nur zur Untermalung der einzelnen Wettkämpfe, sondern gibt auch Hinweise auf stattfindende Aktionen. Die WINTEROLYMPIADE '88 kann mit bis zu sechs Spielern gleichzeitig gespielt werden, was den sowieso schon recht hohen Unterhaltungswert dieses Programms noch weiter steigert. Leider können in keiner der Disziplinen zwei Spieler gegeneinander antreten; so bleibt also nur der Kampf gegen die Uhr bzw. um Punkte. Trotzdem ist die WINTEROLYMPIADE '88 eines der bestprogrammierten und spannendsten Sportspiele, die es für den Amiga zur Zeit gibt, und ein Muß für jeden sportbegeisterten Computerfreak. Das Programm läuft auf allen Amigas mit einem Laufwerk und mindestens 512 Kbyte Speicher. (JS)

Hersteller: TYNESOFT



Einkaufsführer

Hier finden Sie Ihren
Commodore/Amiga Fachhändler

1000 Berlin



RUNOW

Büroelektronik

Keithstraße 26 · 1000 Berlin 30

☎ 26 111 26

2000 Hamburg

GMA mbH
Systemhändler
Wandsbeker Chaussee 58
2000 Hamburg 76

2390 Flensburg



Norderstraße 94-96 · D-2390 Flensburg
☎ (04 61) 281 81 + 281 93

COMPUTER-STUDIO

Schlichting

... die etwas andere Computerei

ATARI-Fachmarkt
MS-DOS Fachmarkt · NEC-Fachhandel

Katzbachstraße 8 · 1000 Berlin 61
☎ 030/7864340

Computer

Hardware · Software · Zubehör

Lilienstraße 32
(beim Mönckebergbrunnen)
2000 Hamburg 1
Tel. (0 40) 33 67 08



SYSTEMSHOP

2900 Oldenburg

GOLDT
Computerhaus

Donnerschwer Straße 127-129
(gegenüber Weser Ems Halle)
2900 Oldenburg
Telefon (04 41) 88 47 06

2000 Hamburg

Bit Computer Shop

Osterstraße 173 · 2000 Hamburg 20
Telefon: 040/494400

Createam
Computer Hard & Software

Bramfelder Chaussee 300 · 2000 Hamburg 71
Telefon: 040/ 6416473+6416861

Bei uns werben bringt

GEWINN

Sprechen Sie mit uns.

Heim Verlag ☎ 06151/56057 BUF

2940 Wilhelmshaven

Radio Tiemann

Commodore-Systemfachhändler

Marktstraße 52
2940 Wilhelmshaven
Telefon (0 44 21) 2 61 45

Gerhard u. Bernd Waller GbR
Computer und Zubehör-Shop

Kieler Straße 623
2000 Hamburg 54

Telefon (0 40) 5 70 60 07 + 5 70 52 75

2160 Stade

BERGAU
Büromaschinen · EDV-Systeme
Neue Straße 5 · 2160 Stade
Telefon (0 41 41) 23 64 u. 23 84

3000 Hannover

COM DATA

Am Schiffgraben 19 · 3000 Hannover 1
Telefon (05 11) 32 67 36

3000 Hannover

COMPUTERSOFT GmbH
HANNOVER'S
 SOFTWARETHEK NR. 1
AMIGA
ATARI
IBM PC

An der Tiefenriede 27 · 3000 Hannover 1
 Tel. (05 11) 88 63 83 · 24 Stunden Service

3500 Kassel

Hermann Fischer GmbH

Commodore-Systemfachhändler

Rudolf-Schwander-Str. 5-13
3500 Kassel
Telefon (05 61) 70 00 00



**Erfolgreich werben
im Amiga-Einkaufsführer**

Sprechen Sie mit uns.
Heim-Verlag ☎ (061 51) 560 57 BUF

4650 Gelsenkirchen-Horst



Hard- und Software, Literatur
Bauteile, Service, Versand
Groß- und Einzelhandel

Poststraße 15 · 4650 Gelsenkirchen-Horst
Telefon (02 09) 5 25 72

5000 Köln

**BÜRO MASCHINEN
braun**
AM RUDOLFPLATZ GmbH
5000 KÖLN 1
RICHARD-WAGNER-STR. 39
TEL. (02 21) 21 91 71

6000 Frankfurt

**COMPUTER
bco**
 Büro-Computer + Organisations GmbH

Oederweg 7-9
6000 Frankfurt/Main 1
☎ (0 69) 55 04 56 - 57

Commodore OKI ATARI TOSHIBA

6200 Wiesbaden

Poststraße 25
6200 Wiesbaden-Bierstadt
Telefon (061 21) 56 00 84
Telefax (061 21) 56 36 43

Werbung und EDV GmbH

AUTORISIERTER
COMMODORE
SYSTEM-HÄNDLER

Commodore

6380 Bad-Homburg

PDC GmbH
Produkte u. Details Computerverband
 Louisenstraße 115
 Ladenpassage Alter Bahnhof
 6380 Bad-Homburg
 Telefon (0 61 72) 2 47 48

Autorisierter Commodore-Systemhändler!

6457 Maintal

Landolt-Computer

Beratung · Service · Verkauf · Leasing

Autorisierter Commodore-Händler
 Wingertstr. 112 · 6457 Maintal/Dörnigheim
 Telefon (061 81) 4 52 93

6551 Fürfeld

**MICHAEL
WEISGERBER**

**HARD
+
SOFT** Rathausstraße 2
6551 Fürfeld
Telefon (0 67 09) 7 78

6680 Neunkirchen

SHOP 64 Homburg
Computer GmbH Kaiserslautern
 Neunkirchen Neustadt
 Telefon (068 21) 2 37 13 Saarbrücken
 Saarlouis
 St. Ingbert
 Trier

Commodore
Systemhändler

6700 Ludwigshafen

COMPUTING & SOUD
 Dieter Hieske
 Schillerstraße 36 · 6700 Ludwigshafen
 Telefon (06 21) 67 31 05

AMIGA USER ONLY

6800 Mannheim

GAUCH+STURM
 Computersysteme + Textsysteme

6800 Mannheim 24
 Casterfeldstraße 74-76
 Telefon (06 21) 85 00 40 · Teletex 6 211 912

7000 Stuttgart

»If AMIGA, go to Schreiber«
 Stuttgart's starker Computer-Laden.

SCHREIBER COMPUTER
 HANDELS-GMBH

Motorstraße 39
 7000 Stuttgart 31
 Tel. (07 11) 8 30 94 - 0

Neu · Neu · Neu · Neu
 Im SUBWAY
 BREUNINGER CITY

7140 Ludwigsburg

BDT
 BÜRO-DATEN-TECHNIK-VERTRIEBS GMBH
 Kurfürstenstraße 18 · 7140 Ludwigsburg
 Telefon (071 41) 2 50 74

7890 Waldshut-Tiengen

hettler-data
 service gmbh
 Lenzburger Straße 4
 7890 Waldshut-Tiengen
 Telefon (0 77 51) 30 94

7900 Ulm

Systemhaus:
 Frauenstraße 28
 7900 Ulm/Donau
 Tel. (07 31) 2 80 76
 Telex 7 12 973 csulm-d

EDV-Systeme
 Software-
 erstellung
 Schulung

COMPUTER
STUDIO

8000 München

Ludwig
 COMPUTER + BÜROTECHNIK
 COMPUTER · SOFTWARE · PERIPHERIE
 BERATUNG · TECHN. KUNDENDIENST
 INGOLSTÄDTER STRASSE 62L
 EURO-INDUSTRIE-PARK · 8000 MÜNCHEN 45
 TELEFON 089/3 11 30 66 · TELETEX 89 83 41

AMIGA

MACHINE
 Computergraphic
Gabriele Lechner
 · Beratung
 · Schulung
 · Verkauf

Planeggerstraße 1 · 8000 München 60
 Telefon (0 89) 8 34 05 91

8400 Regensburg

**Zimmermann
elektroland**

8400 Regensburg 8390 Passau
 Dr.-Gessler-Str. 8 Kohlbruck 2a
 ☎ 09 41 / 9 50 85 ☎ 08 51 / 5 20 07

Bei uns werben bringt

GEWINN

Sprechen Sie mit uns.
 Heim Verlag ☎ 0 61 51 / 5 60 57

BUF

8700 Würzburg

**SCHILL
BÜROTEAM**

Hardware · Software
Service · Schulung

computer center

am Dominikanerplatz
Ruf (09 31) 308 08-0



**DISCOVERY
SOFTWARE
INTERNATIONAL**

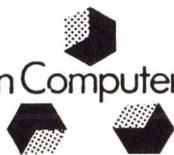
Die Firma, die Euch
MarauderII® und GRABBIT®
gebracht hat, hat jetzt in
Deutschland ein Büro.
Schaut mal nach unseren
neuesten Titeln.

**Friedrich-Spee-Straße 11
8700 Würzburg
Telefon (09 31) 88 48 22**

Schweiz

CH-8021 Zürich

Senn Computer AG



Langstrasse 31
Postfach
CH-8021 Zürich
Tel. (01) 2 41 73 73
Telex 8 14 193 seco

CH-4054 Basel

**Wir sind
Amiga-Freaks.**

Unsere Öffnungszeiten:
Von Dienstag bis Freitag
9.30—12.30 und 14.00—18.30 und
am Samstag 9.30—16.00 Uhr

**SYSAG
COMPUTERCENTER**

Basel: Tel. 061/39 25 25 · Holeestrasse 87 · 4054 Basel
Aarau: Tel. 064/22 63 33 · Kasernenstrasse 26 · 5000 Aarau

AMIGA ★ Public-Domain ★ AMIGA
STEFAN OSSOWSKI

Ca. **550** Disketten lieferbar:
Fisch 1—**148**, Panorama 1—**68**, Fang
1—**51**, Amicus 1—**20**, Auge 4000 1—**19**,
Taifun 1—**60**, Chiron Conception 1—**68**,
RPD 1—**113**, KICK 1—**60** u.v.a.

Bei Bestellung von mindestens **10** Dis-
ketten wird die PD-Disk **CLI-Help**
— unentbehrlich für Anfänger und
Einsteiger — **Kostenlos** mitgelie-
fert! — Stichwort: **CLI-Help**

Einzeldisk DM 7,00
ab 10 Stück DM 6,50
ab 20 Stück DM 6,00
ab 30 Stück DM 5,50
ab 50 Stück DM 5,00
ab 100 Stück DM 4,70
ab 200 Stück DM 4,50

Wir kopieren selbstverständlich auf
2-DD-Disketten!

2 Katalogdisks
mit Kurzbeschreibung aller Programme
gegen **DM 5,-** (Scheck/Briefmarken)
anfordern!

Garantie: Versand erfolgt am
gleichen Tage des
Bestelleinganges!

10 % Abo-Rabatt bei Neuerschei-
nungen. (Alle oder bestimmte Serien)

★★ **Taifun** ★★
Super-PD-Software
Taifun = Auslese der besten
auf dem Markt befindlichen PD-
Programme.

Eigenentwicklung
— **Exklusivvertrieb**

Sonderangebot:
Nr. 1—Nr. 30 V-Scheck DM 160,—
Nr.1—Nr. 40 V-Scheck DM 205,—
Nr. 1—Nr. 50 V-Scheck DM 245,—
Die **echten Taifun** erkennt man an
der Original-Seriennummer!

**Das große
Public-Domain-Buch**
Ausführliche deutsche Beschreibung
zu ca. 100 Public-Domain-Program-
men auf ca. 320 Seiten.
DM 49,— zuzüglich Versandkosten

Super-Grafik-Paket

Inhalt: Ray-Tracing (DBW-Render), 1 Zeichenprogramm, 1 Spriteeditor,
28 Zeichensätze, Apfelmännchen, Fractals, viele IFF-Grafiken,
Slideshow-Programme...

Sonderpreis: DM 62,— Scheck **DM 66,—** Nachnahme
— incl. Porto- und Verpackungskosten —

STEFAN OSSOWSKI

Veronikastraße 33
D-4300 Essen 1
Tel.: 02 01 / 78 87 78

IHR PD-SPEZIALIST

INSERENTEN-VERZEICHNIS

	Seite		Seite
AIT M. RÖNN	95	MERLIN	50, 120
ALCOMP	55	MUSIK- UND	
COMPUTING UND		GRAPHIKSHOP	37
SOUND	66	OSSOWSKI	125
C.V.S.	63	OPTIVISION	95
CWTG	54	PBC	103
C.S.S.	117	PDC	66, 79, 109
DATA-BECKER 2, 38, 39		PERS. COMPUTER	
DATA COM	45	GENTELE	72
DONAU SOFT	103	PHILGERMA	25
DTM	9	RAINBOW DATA	66
EDOTRONIK	63	RECK	29
EDV TECHNIK		SOFTWARE 2000	34
BEHRENDT	37	SCHMIELEWSKI	11
ELEKTRONIK-		SIGIS SOFTWARE	103
ZUBEHÖR	54	STALTER	11
FISCHER	37	TRÖPS	47
HEIM	115, 131	UPC	72
IDEE SOFT	114	VIDEO LOFT	11
IM	54, 66	WALLER	34
KUPKE	132	YELLOW	63

KICKSTART PU

Liebe Leser,

in diesem Monat präsentieren wir Ihnen wieder einmal unser komplettes PUBLIC DOMAIN PROGRAMM, das wir aus den besten PD-Programmen zusammengestellt haben. Neu in diesem Heft ist z.B. eine UTILITY-Diskette, die aus vier Disketten, deren Programme uns von unseren Lesern zugeschickt, zusammengestellt wurden. Nach dem Aufruf in der letzten Zeitung sind einige interessante PD-Programme bei uns eingegangen. Wir werden die anderen Programme im Laufe der Zeit auf themenbezogenen Disketten veröffentlichen. Allen Einsendern sei an dieser Stelle Dank ausgesprochen, wir werden uns um jede Einsendung speziell kümmern. Bitte denken auch Sie auch weiterhin daran, wie nützlich und interessant solche Programme für die AMIGA-Benutzer sein können. In der Hoffnung, daß auch Sie sich rege an dieser Aktion beteiligen, wünsche ich Ihnen bis zum nächsten Mal viel Erfolg beim Programmieren Ihrer Programme.

(M. Nerding)

DIE NEUHEITEN

Diskette 60: RAY-TRACER

DBW-Render ist ein leistungsfähiges Programm zum Erstellen von Bildern nach dem Ray-Tracing-Verfahren. Die Daten der Bilder werden mit einem Texteditor eingegeben, wobei spezielle Befehle zur Verfügung stehen. Die Berechnung der Bilder durch das Programm kann, wegen des aufwendigen Verfahrens, mehrere Stunden dauern, aber die Ergebnisse sind hervorragend.

PD 61: GRAFIK-DEMOS

Auf dieser Diskette sind die Beiträge von Allen Hastings, Hobie Oris und Phil Burk zum BADGE KILLER DEMO CONTEST enthalten.

RIPPLES: eine eindrucksvolle Sequenz, die mit VideoScape 3D erstellt wurde.

WAVEBENCH: bringt Ihren Bildschirm gewaltig ins Wanken;

VIACOM: ist auch nicht viel besser!

DROPSHADOW 2.0: von nun an hat jedes Fenster eine Schattenseite, mit Einstellregler für Schattenstärke und -länge.

PD 62: GRAFIK-DEMOS



Eine weitere Diskette mit Werken vom BADGE KILLER DEMO CONTEST. Dieser Wettbewerb hat so viele gute Demos hervorgebracht, die man einfach nicht verstecken kann.

HAGENDEMOS: Die Filme, die Joel Hagen zeigt, sind unbestritten Kunstwerke, denn nicht umsonst wurde "RGB" der Gewinner des BADGE KILLER DEMO CONTEST. Die beiden Filme wurden mit "The Director" von "The Right Answers Group" erstellt. (benötigt 1 MB RAM !)

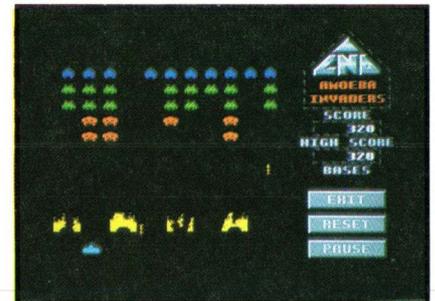
BULLY: Diese Demo ist der absolute Wahnsinn. Auf der Workbench läuft ein Boxen-Demo, auf einem anderen Bildschirm blinkt ein Bild von Zipp mit Colorcycling und auf einem weiteren hüpfert der AMIGA-Ball. Doch nicht genug, die drei Bildschirme sind auch noch in Bewegung. Sie gleiten

auf und nieder und treten dabei abwechselnd in den Vordergrund. Da zeigt der AMIGA was Multitasking bedeutet.

LINEDRAWER: zeichnet Bilder, deren Daten einer Datei entnommen werden.

POPLIFE: ist eine Variante von LIFE, die sich direkt auf der Workbench abspielt.

PD 63: (Action-) SPIELE



AMOEBAS: hinter diesem Namen verbirgt sich eine spielhallenmäßige SPACE INVADERS-Adaption. Obwohl dies ein Spiel der ersten Computergeneration ist, hat es nichts von seiner Spannung verloren. Ein sehr gutes Ballerspiel.

ASTEROIDS: ist ein Weltraumspiel, das ebenfalls von Spielhallencomputern umgesetzt wurde. Es ist nicht ganz einfach, die Kontrolle über das Raumschiff zu behalten und es sicher im Asteroidenhafen zu manövrieren.

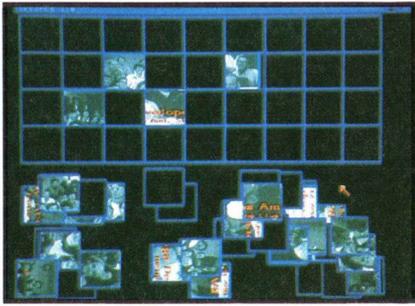
EGYPTIAN RUN: nennt sich ein Actionspiel, bei dem ein Geländewagen durch die Wüste gesteuert werden muß. Dabei muß man den Sandhügeln und besonders dem Sandstrahl ausweichen, der den Wagen zu unkontrollierten Bewegungen veranlaßt.

PD 64: (Strategie-) SPIELE

BACKGAMMON: bei diesem Strategiespiel können Sie zeigen, ob Sie besser sind als der Computer.

PUSH OVER: hier wird so lange geschoben bis plötzlich und unerwartet einer der Spieler fünf Steine in einer Reihe hat. Ein spannendes Strategiespiel.

BLIC DOMAIN



PUZZLE PRO, IFF2PICS: zwei Programme, die ein beliebiges IFF-Bild in mehr oder weniger viele Einzelteile zerlegen. Dann liegt es an Ihnen, wie lange Sie brauchen, um es wieder zusammenzupuzzeln. Der Schwierigkeitsgrad ist einstellbar.

PD 65: ICON-TOOLS

Mit diesen Programmen ist es jedem möglich, eigene Icons zu erstellen. Diese können sogar gegen alte ausgetauscht werden.

DPICON: ermöglicht es, eine mit Deluxe Paint erstellte Brush beliebiger Größe in ein Icon umzuwandeln. Damit kann man Bilder mit dem Malprogramm entwerfen und dann einfach in ein großes Icon umwandeln.

SNAP: ist ebenfalls ein sehr interessantes Hilfsprogramm, denn damit kann ein beliebiger Bildschirm in ein Icon umgewandelt werden. Die Größe des Icons ist dabei einstellbar. Bei mehrfarbigen Screens wird außerdem eine Farbreduzierung durchgeführt.

ALTICON: gibt dem eigenen Icon den letzten Pfiff, denn hiermit werden zwei Icon-Bilder zu einem 'animierten' Icon zusammengefügt.

Und außerdem noch eine Vielzahl schöner Icons, damit nicht alles gleich aussieht.

PD 66: M2-Modula

M2 AMIGA: Dies ist die funktionsfähige Demo-Version des M2-Modula-Compilers. Beschränkungen gibt es nur hinsichtlich der Größe der compilierbaren Programme, außerdem sind nur wenige Schnittstellen und Standard-Bibliotheken vorhanden. Auf der Diskette sind einige Demo-Programm-

me enthalten, die z.T. direkt für diesen Compiler entwickelt wurden.

Der MODULA-Kurs, der in der nächsten Ausgabe der KICKSTART beginnen wird, kann fast vollständig mit diesem Compiler nachvollzogen werden.

PD 67: UTILITIES

PR: ein Programm zum Ausdrucken von Texten. Das Inhaltverzeichnis wird in einem Fenster dargestellt und das File kann mit der Maus ausgewählt werden. Auf dem Ausdruck erscheint dann eine fortlaufende Seitennummierung, der Name und das Datum.

SHOW entspricht dem Programm PR jedoch werden die Texte nur angezeigt. **FUEL GAUGE:** ist eine grafische Anzeige für den vorhandenen und benutzten Speicherbereich.

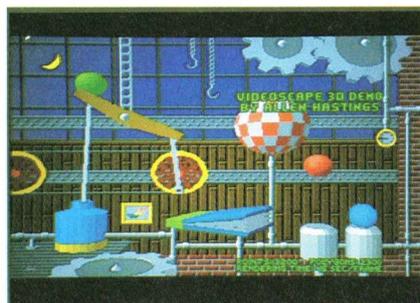
SIMCPM: CP/M-Emulator, der den 8080-Prozessor emuliert. Mit H19 Terminal.

LOGO: ein Interpreter der bekannten-Turtle-Sprache, die mehr kann als nur malen.

MICROSPELL ist ein flexibler Spellchecker mit einem englischen Grundwortschatz von 43.000 Wörtern.

PD 68: GRAFIK-DEMO

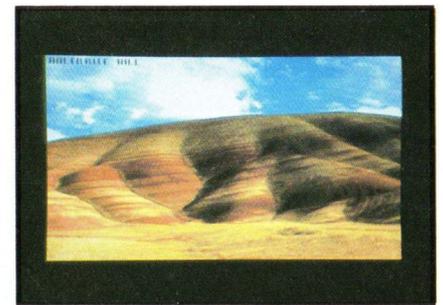
WBLANDER: versuchen Sie das Raumschiff sicher auf einem Workbench-Fenster zu landen. Drei Triebwerke können zum Abbremsen und Festlegen der Richtung eingesetzt werden. Bei erfolgreicher Landung wird der Kontrolltower eingblendet, der Ihnen zur Landung gratuliert. Auch dieses Programm hat am BADGE KILLER DEMO CONTEST teilgenommen.



MACHINE: eine hervorragende Demo von Allen Hastings erstellt mit VideoScape 3D.

DRUNKENMOUSE: wenn Sie nicht betrunken sind, dann liegt es sicherlich an diesem Programm, daß Sie Schwierigkeiten haben, etwas anzuklicken.

PD 69: GRAFIK-DEMOS



AMUC_DEMO: Ein Bild mit 200 x 2000 Bildpunkten scrollt vertikal über den Bildschirm. Ein sehenswertes Erlebnis.

HBHILL: Dies ist Kevin Sullivans Beitrag zum BADGE KILLER DEMO CONTEST. Die Demo nutzt den Extra-Half-Brite-Modus (64 Farben) des AMIGA, der jedoch auf sehr alten Modellen nicht vorgesehen war.

HAMmmm: Eine mit Musik unterlegte Linendemo von Phil Burk. Inclusive Source-Code in JForth.

STARS: Ein interessanter Flug durch einen Sternenhimmel.

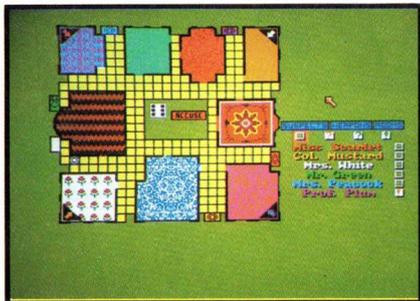
WIREDMO: zeigt einen rotierenden Linienkörper, mit Source in C.

BRAND NEU

PD 70: SPIELE

CLUE: das bekannte Brettspiel für pfiffige Detektive in einer ansprechenden Computerversion. Verschiedene Räume müssen untersucht, Personen befragt und die Ereignisse kombiniert

KICKSTART PUBLIC DOMAIN



werden um den Mörder herauszufinden. Die gute Grafik und die einfache Mausbedienung machen eine Anleitung weitgehend unnötig.

WHEEL: Bei diesem Spiel muß ein Ausdruck erraten werden. Diese Glücksrad-Simulation ist grafisch gut aufgemacht und komplett mausgesteuert. Da die zu erratenden Begriffe in englisch sind, ist dieses Spiel gut zum Erlernen dieser Sprache geeignet.

MASTERMIND: bekanntes Kombinationspiel, bei dem eine Farbkombination erraten werden muß. Mausgesteuert!

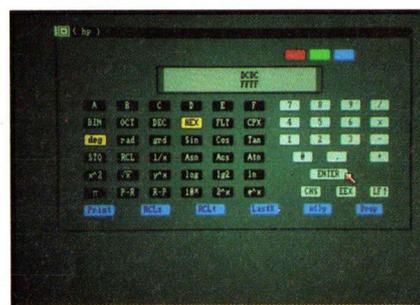
RISTINOLLA: Eine spielstarke GO-Muko-Variante. Mausgesteuert und sehr schnell!

PD 71: UTILITIES

PRTDRVGEN 2: der Druckertreiber-generator zum Anpassen der exotischsten Typen an den AMIGA. Sehr einfache Benutzerführung mit ausführlicher (englischer) Erläuterung aller Optionen (siehe Bericht in dieser Ausgabe).

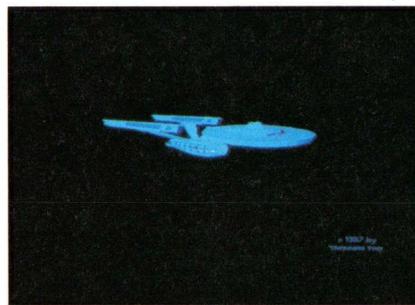
VIRUS X: läuft im Hintergrund und prüft den Bootsektor jeder eingelegten Diskette auf Viren und mögliche Veränderungen. Mit Source.

OVERSCAN: ein Patch für die Intuition-Library, mit der amerikanische Programme das PAL-Format ausnutzen könnten, ohne daß sie modifiziert werden müssen.



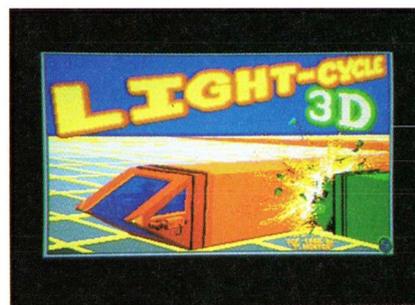
HP: ein RPN-Taschenrechner mit vielen Funktionen (BIN, OCT, DEC, HEX, SIN, COS, TAN, ATN, LOG, LN, LG2, ATN, y^x ...) und 32 Registern. Mit Source in C.

PD 72: ANIMATIONEN



Für alle STAR TREK-Fans hat Tobias Richter zwei Animationen mit VideoScape 3D erstellt. Die erste zeigt ein Klingonenschiff, die zweite die U.S.S. ENTERPRISE in voller Pracht.

PD 73: SPIELE



LIGHT CYCLE 3D: ist eine sehr gut programmierte TRON-Variante, bei der zwei Fahrer versuchen, dem Gegenspieler den Weg abzuschneiden. Das Geschehen wird aus der Sicht des Fahrers verfolgt. Die 3D-Grafik ist dabei ein wirkliches Erlebnis. Da jeder Fahrer eine undurchdringbare Bahn hinterläßt wird das Spielfeld schnell

eng. Dann hilft nur noch ein energieko-stender Schuß, um sich Platz zu schaffen. Mit vielen Anzeigeinstrumenten und Anleitung. Autor: Tobias Richter
ARTILLERIE-DUELL: in einer zerklüfteten Berglandschaft sehen sich zwei Kanonen gegenüber. Jeder der zwei Spieler muß nun den Abschußwinkel und die Stärke einstellen, mit der er seine Kugel abfeuern will. Das Spiel ist nicht ganz so einfach wie es aussieht, denn die Spitze des gegnerischen Kegels muß ziemlich genau getroffen werden, außerdem kann die Windstärke eingestellt werden. Autor: Peter Kliem

TREFFER: eine Kniffel-Variante, die komplett mit der Maus gesteuert wird. Bis zu vier Spieler, mit Zwischenstand, Hi-Score und Statistik. Autor: Michael Jänecke

PD 74: DIGI-SOUNDS

The Sound of Musik (Volume 1): Digitalisiert Musikstücke zur Verwendung in Vorspannen, Demos und eigenen Programmen. Sehr gute Qualität; mit Player; 11 Samples; Länge 30-100kB. Sampler: Harald Schneider

PD 75: SUPER-SOUNDS

KICKSTART SOUND DEMO: Nicht digitalisierte Sounds, sondern programmierte Musikstücke schickt uns Tom Becke, damit wir "Abstand nehmen von den schönsten Digi-Sounds (kann doch jeder, oder?)". 17 actiongeladene Musikstücke; Länge 12-95kB. Komponisten: TOB & TWP

KICKSTART PD-SAMMLUNG:

Die einzelnen Disketten werden nach festen Kriterien zusammengestellt, d.h. daß jede Diskette einen Schwerpunkt hat (z.B. Lehrgänge (Tutor), Bilder-Show, C-Programme, Utilities, Spiele, u. ä.). Außerdem werden Angaben über die Programmiersprache, den verwendeten Interpreter oder Compiler usw. gemacht.

Die Programme laufen auf allen AMIGA-Computern mit Kickstart/Workbench 1.2, allerdings sollten mindestens 512k Speicher vorhanden sein, die meisten Grafikdemos benötigen sogar 1MB Speicher. Sollten dennoch Einschränkungen gelten, wird dies bei den betreffenden Programmen angegeben.

KICKSTART PUBLIC DOMAIN

DAS AKTUELLE ANGEBOT

Diskette 1: C-Source

Eine Sammlung von Programmen, die besonders den Anfänger zeigen, wie man Intuition programmiert. Die Programme liegen sowohl als C-Quellcode als auch als fertige Programme vor, die sofort gestartet werden können.

Diskette 2: Spiele

YachtC (Würfelspiel für 4 Personen), Puzzle, Missile (verteidigen Sie Ihre Stadt, 3D-Sound), Triclops (sehr schönes 3D-Spiel), Breakout (3D-Effekt mit Brille), Trek73 (bekannte Star Trek-Variante)

Diskette 3: Spiele

HACK: das bekannte Textadventure, das ursprünglich auf UNIX-Rechnern erstellt wurde, liegt hier als spezielle Grafikversion für den Amiga vor.

Diskette 4: Terminal-Programme

KERMIT: bekanntes, luxuriöses Terminalprogramm (drei verschiedene Versionen, mit Source-Code in C)

Diskette 5: Terminal-Programme

WOMBAT (VT102/52 Emulator, XModem, autodial), VT100 (grafikfähig, Source in C), TermPlus (XModem, Source in C), DG210 (Data General D-210 Emulator), AHost (XModem, Kermit), TEK4010 (XModem, VT100)

Diskette 6: Terminal-Programme

Speech Term (spricht den empfangenen Text, XModem), StarTerm (mit Phone, Duplex, XModem), Argo Term, PD Term (Source in C), AmigaDisplay

Diskette 7: UTILITIES

- QuickCopy (gutes Kopierprogramm)
- DirUtil (File-Copy)
- FileZap (File-Monitor)
- DiskZap (Disk-Monitor)
- DiskSalv (Diskettenretter)
- System-Monitor
- CSH (UNIX-ähnliche Shell)

Diskette 8: Spiele

Monopoly: das bekannte Brettspiel mit sehr schöner Grafik, einfacher Mauseingabe, bis zu vier Spieler (Source in ABasic)

Diskette 9: Grafik-Show

Grafik-Show mit bekannten Cartoons und schönen Landschaftsbildern

Diskette 10: Grafik-Show

JUGGLER DEMO: ein bewegliches Männchen jongliert mit drei verspiegelten Kugeln, sehr schöne Demo

Diskette 11: Grafik-Show

RAY TRACERS: wunderschöne räumliche Bilder, die auf einer VAX berechnet und auf den AMIGA übertragen wurden

Diskette 12: Grafik

Digitalisierte Bilder mit erstaunlicher Qualität (IFF-Format)

Diskette 13: Grafik

Sehr schöne Bilder-Show (IFF-Format)

Diskette 14: EDITOR

Bekannter Texteditor MICROEMACS Version 30 (viele Features: Search/Replace/Copy)

Diskette 15: Grafik-Animation

Verschiedene Filme, die mit dem AEGIS-ANIMATOR erstellt wurden, incl. PLAYER zum Abspielen der Filme. INFO: Einige Filme benötigen auf dem AMIGA 1000 mehr als 512 KB Speicher. Bei AMIGA 500/2000 mit 1MB Speicher muß vorher 'NoFastMem' gestartet werden!

Diskette 16: Sprachen

XLISP 1.7 (neueste Version) mit ausführlicher Anleitung (über 50k)

Diskette 17: Sprachen

MODULA-2: Pre-Release eines Modula-Compilers mit verschiedenen kleineren Beispielprogrammen, die als Source-Code vorliegen.

Diskette 18: Grafik

MANDELBROT-Generator

Diskette 19: Grafik-Show

Sehr schöne digitalisierte H.A.M.-Bilder

Diskette 20: Grafik-Show

'Fred the Baker and Rose's Flower Shop' COMIC-Film, der die Multitasking Fähigkeiten des AMIGA erklärt

Diskette 21: AMIGA-Tutor

Einführung in die Bedienung des AMIGA 500. Ein farbenfroher Lehrgang, der ganz am Anfang beginnt und mit vielen Bildern und Grafiken die Grundbegriffe des AMIGA erklärt. (für Anfänger, komplett in deutsch)

Diskette 22: Sprachen

MVP-FORTH und C-FORTH (C-Forth ist ein recht leistungsfähiger FORTH-Interpreter, der auch als Quelltext vorliegt)

Diskette 23: Grafik-Show

Viele abwechslungsreiche Motive in verschiedenen Auflösungen, verpackt in einer Grafik-Show.

Diskette 24: Grafik-Show

Sehr schöne, digitalisierte Frauengestalt.

Diskette 25: UTILITIES

CLOCK, PORTAR, MACView, Kickbench, Disassembler, Tracker, Checkmodem, POPCLI und vieles mehr.

Diskette 26 & 27: Grafik-Show

Auf zwei randvollen Disketten erleben Sie eine einmalige Dia-Show mit hervorragend digitalisierten futuristischen Bildern in voller PAL-Auflösung. Dazu gibt es stimmungsvolle, sphärische Musik.

Diskette 28: Editoren

Auf dieser Diskette befinden sich einige schöne Editoren (UEDIT, MED, BLITZ) mit dazugehörigen Zeichensatz-Utilities.

Diskette 29: UTILITIES

PriDrvGen: erstellt Drucker-Treiber
DropShadow: jedes Fenster bekommt einen Schatten
MemClear: löscht den Speicher
ScreenSave: speichert den Bildschirm auf Diskette
Compress: komprimiert Programme

Diskette 30: SOUND-DEMOS

Digitalisierte Songs: Changing Minds, Joan Lui, Miami Vice II, Respectable, Holiday

Diskette 31: SOUND-DEMO

Dieses Programm erzeugt naturgetreue Geräusche, die über die Tastatur, wie auf einem Klavier, angespielt werden können.

Diskette 32: SOUND-DEMOS

Mit einer Demo-Version von SoundScape können digitalisierte Musikstücke abgespielt werden. Die Qualität ist wirklich erstaunlich!

Diskette 33: GRAFIK-SHOW

Einige sehr gute mit Deluxe Video erstellte Filme. Der benötigte PLAYER ist auch auf der Diskette. INFO: bei AMIGA 500/2000 mit 1MB Speicher erst 'NoFastMem' starten!

Diskette 34: SPIELE

TUNNEL VISION: Werden Sie den Weg durch das Labyrinth finden?
REVERSI: eine spielstarke Version des

bekanntes Brettspiels

KLONDIKE: ein Patience-Kartenspiel

Diskette 35: UTILITIES

ASDG (resetteste RAM-Disk)
FixDisk, ErrorCk (zur Fehlersuche auf der Diskette)
DiskCat (erstellt eine Übersicht über die Programme Ihrer Disketten)

Diskette 36: CAD

mCAD ist ein wirklich gut gemachtes CAD-Programm, daß im Interlace-Modus läuft. Es bietet die einfachen Zeichenfunktionen und Features wie Zoom, Group, Ungroup, Grid, Move, Rotate). Auf der Diskette sind mehrere Dokumente, die das Programm erklären.

Diskette 37: UTILITIES

AddMem: zum Konfigurieren von Speichererweiterungen
MemView: zeigt den Speicherinhalt als Grafik an
GetRom: schreibt das Betriebssystem-ROM des AMIGA 500/2000 als bootfähige Kickstart für den AMIGA 1000 auf Diskette.
MegaPatch: paßt die Kickstart des AMIGA 1000 für das autom. Erkennen von internen Speichererweiterungen an.

Diskette 38: GRAFIK

NoFFP Mandelbrot Set Explorer V.2.1 (neue Version)
von ABC Softarts in Braunschweig

Diskette 39: GRAFIK-SHOW

Stimmungsvolle Landschaftsbilder, die sich gut zum Weiterverarbeiten eignen und einige digitalisierte Bilder.

Diskette 40: GRAFIK-DEMOS

Boing!, Rotate, Sparks, Moire, Dazzle, 3DCube, Scales, Sizzlers.
Sehenswert ist der Film 'Atari meets AMIGA', der die erste und einzige Begegnung der beiden Computer dokumentiert.

Sehr schön ist das Programm LANDSCAPE, das wunderschöne fraktale Berg- und Tallandschaften erzeugt.

Diskette 41: UTILITIES (Grafik)

Alles, was Sie zu dem von ELECTRONIC ARTS entwickelten Grafik-Standard (IFF-Format) wissen müssen: Laden, Speichern, Komprimieren, Dekomprimieren. Mit Dokumentationen und Source-Codes in C.

Diskette 42: GRAFIK-SHOW

Vielfältige nach dem RAY-TRACER-Verfahren erstellte Bilder. Lassen Sie sich von den realistischen Spiegelungen beeindrucken! Mit digitalisierter Musik!

Diskette 43: GRAFIK-SHOW

Eine einmalige Show, bei der eine digitalisierte Katze in gleitenden Bewegungen über den Bildschirm trabt. Erstellt wurde diese faszinierende Animation mit einem Digitizer, DPaint und VideoScape 3D.

Diskette 44: SPIELE

Adventurefans kennen sicherlich das Grafikadventure HACK (siehe PD 3). Hier gibt es nun die Fortsetzung: LARN. In unüberschaubaren unterirdischen Gängen müssen Gold und Schätze gesucht werden. Aber auch an einem Krafttrunk oder einem magischen Spruch sollte man nicht achtlos vorbeigehen, denn die benötigt man im Kampf gegen Gnome, Vampire und andere Gestalten.
Wirklich sehr empfehlenswert!

Diskette 45: SPIELE

GRAVITYWARS ist ein interessantes Weltraumspiel, bei dem sich zwei Raumschiffe im Kampf gegenüberstehen. (viele Optionen!).
OTHELLO, eine sehr schöne Reversi-Variante (mit eigenem Fenster!).
STREITPATIENCE, eine Patience-Variante von Hellmut Voelcker (Berlin)
CHESS: spielstarkes Schachprogramm
ADVENTURE: ein Textadventure

Diskette 46: GRAFIK-SHOW

Eine weitere Diskette (siehe auch PD 42) mit phantastischen RAY-TRACER-Bildern, unterlegt mit digitalisierter, fetziger Musik.

Diskette 47: UTILITIES

SECTORAMA: ein sehr nützlicher Disketten- und Festplatten-Monitor, mit dem verlorene oder zerstörte Daten wiederhergestellt werden können.
SILICON: ein sehr komfortabler CLI-Ersatz mit separatem Ausgabe- und Eingabefenster.
DBUG: maschinenunabhängiger Debugger von Fred Fish (Source in C)
TIMER: eine Stoppuhr für die Workbench

Diskette 48: CRAZY

Auf dieser Diskette befinden sich nur verrückte Programme, deren Sinn absolut zweifelhaft ist. Allerdings sollten Sie sich diesen Spaß nicht entgehen lassen!

Diskette 49: ICONS

Utility-Programme, die sich mit der Erstellung und Manipulation von Icons beschäftigen. z.B.: XICON 2.0 (mit diesem Programm können Dateien ausgeführt werden, die CLI-Kommandos enthalten)

Diskette 50: BASIC

Eine Diskette voll Programmen (Spiele, Grafikprogramme uva.) in AmigaBAS-

CI zum Reinschauen, Verändern, Lernen.

Diskette 51: C-Compiler

Ein einfacher C-Compiler, in den man etwas Arbeit stecken muß, damit er läuft. Eignet sich für Interessenten am Compilerbau, da der Source-Code vorliegt.

Diskette 52: UTILITIES

CONMAN V1.0 (sehr nützlicher CLI-Ersatz)
MOUSEREADER (Texte lesen mit Mauseingabe)
TIMERAM, MEMWATCH, DISKMAN V1.0 (Bedienung fast aller Diskettenoperationen mit der Maus, sehr umfangreich!)
DIRMASTER V1a (sehr schönes Disketten-Verwaltungsprogramm, viele Funktionen)

Diskette 53: COMPILER

Auf dieser Diskette befindet sich die Sprache ADL (Adventure Definition Language). Das System besteht aus Compiler, Interpreter und Debugger, wobei alle Teile als Source in C und auch ablauffertig vorliegen. Kreieren Sie Ihr eigenes Adventure.

Diskette 54: Anwenderprogramme

MICROSPELL (überprüft die Rechtschreibung)
ACCESS (Terminal)
QBASE (Dateiverwaltung) uvm.

Diskette 55: Grafik/Utilities

Einige schöne Grafikdemos und Utilities zu diesem Thema

Diskette 56: ASSEMBLER

ASM68K (Macro Assembler mit guter Dokumentation)
ASM (68010 Macro Assembler wie im AmigaDOS Manual beschrieben)
BLINK (bekannter Linker)
AS6502 (portabler 6502-Assembler mit Source in C)

Diskette 57: UTILITIES

Wieder eine Diskette mit hilfreichen Programmen, die wir speziell dafür aus dem großen Angebot auswählen.

Diskette 58: Grafik-Animation

Mit drei herrlichen Grafikdemos stellt Eric Graham seine Programme Sculpt 3-D und Animate 3-D vor. Die Bilder der Animationen sind nach dem Ray-Tracer-Verfahren berechnet.

Diskette 59: Grafik-Animation

Mit drei herrlichen Grafikdemos stellt Eric Graham seine Programme Sculpt 3-D und Animate 3-D vor. Die Bilder der Animationen sind nach dem Ray-Tracer-Verfahren berechnet.

JETZT NUR NOCH DM 8.- PRO DISKETTE

Dank eines neuen Kopierservice können wir die Disketten jetzt günstiger anbieten. Außerdem verwenden wir nur doppelseitige Markendisketten der Firmen JVC und MAXELL.

VERSANDBEDINGUNGEN:

Um einen schnellen und problemlosen Versand zu gewährleisten, beachten Sie bitte folgende

Punkte:

- Für jede Diskette ergibt sich ein Unkostenbeitrag von DM 8.-
- Pro Sendung kommt ein Versandkostenbetrag (für Porto und Verpackung) von DM 5.- (Ausland DM 10.-) hinzu.

Auslandsbestellungen nur gegen Vorkasse möglich

BEI EINER BESTELLUNG VON 5 ODER MEHR DISKETTEN ENTFÄLLT DER VERSANDKOSTENBEITRAG

MERLIN Computer GmbH, KICKSTART Redaktion

Postfach 55 69, 6236 Eschborn

VORSCHAU

HARDWARE

Eine RAM-Disk die gleich nach dem Einschalten verfügbar ist - ein Wunschtraum diskettengeplagter Amiga-Besitzer? Beleibe nicht! In der nächsten Ausgabe stellen wir eine Eprom-Disk vor, die alle Vorzüge einer Ramdisk (Zugriffszeit), mit den Vorteilen einer normalen Diskette (Dateinerhalt bei Stromausfall), verbindet. Die Karte findet ihren Platz in einem Amiga 2000 und muß dort zeigen was in ihr steckt.

SOFTWARE

Demon, ein 3D-Objekt-Editor. Mit Demon sollen sich auf einfachste Weise 3D-Objekte erzeugen und edieren lassen. Was leistet ein solches Programm? Wie sind die erzeugten Objekte zu verwenden? Diesen Fragen gehen wir unter anderem in der nächsten Ausgabe auf den Grund.

Programmiersprachen

Modula-2 wird immer beliebter. Mit Benchmark Modula-2 stellt sich ein weiteres Produkt dem Markt. Benchmark Modula-2 fällt vor allem durch die umfangreiche Dokumentation auf. Das Grundpaket enthält ein 1.5 kg schweres Handbuch, das auf den ersten Blick wie eine Neuauflage der Addison-Wesley Reference-Manuals aussieht. Insgesamt eine Programmiersprache die wir uns genauer ansehen werden.

Sonstiges

Neben einigen vielversprechenden Neuigkeiten und den KICKS für INSIDER, bringen wir auch wieder Berichte über einige neue und gute(?) Spiele.

Impressum

KICKSTART

Chefredakteur:
Uwe Bärtels
(Chefredakteur)(UB)
Markus Nerding
(Stellvertreter)(MN)

Redaktion:
Andreas Krämer (AK)
Gerald W. Carda (GC)
Harald Schneider (HS)
Marcelo Merino (MM)
Harald Egel (HE)

Herausgeber:
'MERLIN'-Computer
GmbH
Industriestraße 26
Postfach 5569
6236 Eschborn
Tel.: 06196/481811
FAX: 06196/41137

Redaktionelle Mitarbeiter:
Christian Keller (CHK)
Andreas Diebold (AD)

Wolf Dietrich (WD)
Jobst Hermeier (JH)

Autoren dieser Ausgabe:
Thomas Becker
Florian Du Bois
Carsten Borgmeier
Garry Glendown
Ralf Görlach
Jobst Hermeier
Stefan Kemeny
Herbert Kunz
Cristoph Münzer
Frank Neumann
Christian Rogge
Frank Schäfer
Holger Schemel
Lars Schmitt
Michael Sistig

Public Relations:
Claus Peter Lippert

Auslandskorrespondenz:
D. dela Fuente (GB)
L. Hennely (USA)

Verlag:
Heim Verlag
Heidelberger Landstraße
194
6100 Darmstadt 13

Tel.:06151/56057
FAX:06151/55689
FAX:06151/56059

Verlagsleitung:
Hans-Jörg Heim

Anzeigenverkauf:
Kyriakulla Margaritis
Uwe Heim (Ltg.)

Anzeigenpreise:
nach Preisliste Nr.3,
gültig ab 1.1.88

Layout:
Fabian & Mayer
Titel:
Fabian & Mayer

Produktion:
Karl-Heinz Hoffmann
Susanne Failer
Bernd Failer

Fotographie:
Rainer Spirandelli,
Archiv

Titelfoto:
Rainer Spirandelli

Druck:
Ferling Druck, Darmstadt

Bezugsmöglichkeit:
Zeitschriftenhandel,
Kauf- und Warenhäuser,
Commodore-Fachhändler
oder direkt beim Verlag.

KICKSTART erscheint
11 mal im Jahr
Einzelpreis: DM 7,- ,
ÖS 56,-
SFr 7,-
Jahresabonnement
Inland: DM 70,-
Europ. Ausland DM 90,-
Luftpost DM 120,-
Alle in der KICKSTART
erscheinenden Beiträge
sind urheberrechtlich
geschützt. Reproduktionen
gleich welcher Art,
ob Übersetzung,
Nachdruck, Vervielfältigung
oder Erfassung in
Datenverarbeitungsanlagen,
sind nur mit schriftlicher
Genehmigung des
Herausgebers und des
Heim Verlages erlaubt.
Programmlistings,
Bauanleitungen und
Manuskripte werden von
der Redaktion gerne
entgegenommen. Sie

müssen frei von Rechten
Dritter sein. Mit ihrer
Einsendung gibt der
Verfasser die Zustimmung
zum Abdruck und
der Vervielfältigung.
Honorare nach Vereinbarung.
Für unverlangt
eingesandte Manuskripte
wird keine Haftung
übernommen.
Sämtliche Veröffentlichungen
in KICKSTART
erfolgen ohne Berücksichtigung
eines eventuellen Patentschutzes,
auch werden Warennamen
ohne Gewährleistung
einer freien
Verwendung benutzt.
Für Fehler in Text, in
Schaltbildern, Aufbauskizzen,
Stücklisten, usw.,
die zum Nichtfunktionieren
oder evtl. zum Schadhaftwerden
von Bauelementen führen,
wird keine Haftung
übernommen.

(c) Copyright Heim
Verlag

PROGRAMMIEREN AUF DEM AMIGA

Das richtige Buch für
Auf- und Umsteiger



VON
ERNST HEINZ

PROGRAMMIEREN AUF DEM AMIGA

WICHTIGE MERKMALE

- ▶ Dieses Buch weist Programmierern den Weg, wie Sie die fantastischen Fähigkeiten Ihres AMIGA auch von **AmigaBASIC** aus nutzen können. Es schließt die in den Handbüchern entstandenen Lücken, indem es die offengebliebenen Fragen in anschaulicher Weise beantwortet.
- ▶ Damit ist dieses Buch insbesondere für lernwillige Aufsteiger, d. h. mehr oder weniger erfahrene BASIC-Programmierer, sowie für Umsteiger, die bisher mit anderen Computersystemen gearbeitet haben, konzipiert.
- ▶ Besonderer Wert wurde auf guten Programmierstil gelegt. Anhand von über 40 sehr ausführlich dokumentierten Übungs- u. Beispielprogrammen kann der Leser trainieren, fremde Programme zu lesen und zu verstehen und richtige und gute Programmierung erlernen.
- ▶ Alle offenen Fragen und Probleme in Bezug auf AMIGA-spezifische Programmier- und -befehle wird Ihnen dieses Buch zu beantworten versuchen.
- ▶ Besonders werden folgende Hauptthemen im Buch behandelt:
Fenster- und Menü-Technik · Unterbrechung-Technik · Animation in Amiga-Basic · Grafik-Befehle · Normale Maussteuerung
- ▶ Zum Buch gibt es eine Programm-Diskette mit allen aufgeführten Übungs- und Beispielprogrammen

AUS DEM INHALT

- ▶ **CLI** und Möglichkeiten der **Execute-Programmierung**
- ▶ Erstellen einer eigenen **Startup-Diskette**
- ▶ Spezielle AmigaBASIC-Arbeitsdiskette erstellen
- ▶ Grafikprogrammierung (ohne Animation)
- ▶ Sound- und Sprachprogrammierung (Erzeugung menschlicher Sprache/Erzeugung von Tönen und mehrstimmigen Melodien/Tonerzeugung gemäß musikalischer Notation)
- ▶ **Unterbrechungsfähigkeiten** von AmigaBASIC
- ▶ **Maussteuerung**
- ▶ **Menüprogrammierung** von Pull-Down-Menüs
- ▶ **Fenster- und Bildschirmtechnik**
- ▶ **Grafische Animation mit Sprites und Bobs**
- ▶ Ein **Potpurri** von AmigaBASIC-Programmen
- ▶ Kleine Einführung zur **Spezialhardware** des AMIGA

über 300 Seiten **DM 49,-***

PROGRAMMDISKETTE

AUS DEM INHALT

- ▶ Zur Unterstützung der praktischen Übung und Arbeit am Computer gibt es die Programm-Diskette.
- ▶ An über 40 Übungs- und Beispielprogrammen können Sie guten Programmierstil nachvollziehen und trainieren.
- ▶ Damit hat die lästige Tipparbeit ein Ende; Diskette laden und los geht's...
- ▶ Die Diskette ist beim Verlag erhältlich und kostet nur **DM 29,-***

* Preise sind unverbindlich empfohlene Verkaufspreise

BESTELL-COUPON

an Heim-Verlag
Heidelberger Landstraße 194
6100 Darmstadt-Eberstadt

Ich bestelle: _____ St. *Programmieren auf dem AMIGA* á DM 49,—

_____ St. *Programmdiskette zum Buch* á DM 29,—

zzgl. DM 5,— Versandkosten (unabhängig von bestellter Stückzahl)

per Nachnahme Verrechnungsscheck liegt bei

Name, Vorname _____

Straße, Hausnr. _____

PLZ, Ort _____

Benutzen Sie auch die in KICKSTART vorhandene Bestellkarte.

Heim Verlag

Heidelberger Landstraße 194
6100 Darmstadt-Eberstadt
Telefon 061 51-5 60 57

GOLEM

KUPKE



02 31/81 83 25-27

Telefax 02 31/81 74 29

D-4600 Dortmund 1

Burgweg 52a

Wir
liefern im
3-Tage-Rhythmus



1 Golem Drive 3,5

NEC 1036a mit heller Frontblende ● Amiga-farbenes Metallgehäuse ● Abschalter ● Busdurchführung bis DF3 ● PC-Karten und Sidecar kompatibel !!! neu !!! jedes Drive mit Trackdisplay zur aktuellen Spur- und Kopfanzeige

mit Display
ohne Display

DM 359,-
DM 339,-

2 Golem Drive 5.25

5,25 Zoll Laufwerk mit heller Frontblende ● Amiga-farbenes Metallgehäuse ● Abschalter ● Busdurchführung bis DF3 ● 40/80 Track Umschalter Amiga und MS-Dos kompatibel !!! neu !!! Drive mit Trackdisplay wie Golem 3,5

mit Display
ohne Display

DM 419,-
DM 399,-

3 Golem Drive 3,5 intern

modifiziertes NEC 1036a mit heller Blende ● Staubschutzklappe zum Einbau in den A 2000 incl. Einbausatz und Einbauanleitung

DM 249,-

4 Golem Ram Box 1000

2 MB Erweiterung ansteckbar ● in Amiga-farbenem Metallgehäuse ● Abschalter ● Busdurchführung ● auto konfigurierend ● Betriebskontrollanzeige durch LED ● erweitert den Grundspeicher auf 2,5 Megabyte

DM 1198,-

5 Golem 500

Ram Erweiterung speziell für den Amiga 500 ● technische Einzelheiten wie Golem Ram Box 1000 ● beide Erweiterungen ohne Wait States

DM 1198,-

6 Kickstart / Uhr Modul

"Bitte Workbench einlegen", so meldet sich ihr Amiga 1000 mit dem Kickstart Eprom Modul ● Ansteckbar am Systembus ● Amiga-farbenes Metallgehäuse ● durchgeführter Systembus ● abschaltbar sodasß andere Kickstart Versionen wieder gebootet werden können.

DM 199,-

Amiga 2000 u. 500 kompatibleles, externes Uhrenmodul ansteckbar am Systembus ● Software, die die 2000/500 Uhr anspricht, benutzt auch die Golem Clock für den A 1000

DM 149,-
DM 299,-

Uhr und Kickstart in einem Gehäuse

7 Golem Sound

Audio Digitizer der Spitzenklasse, kompatibel zu aller gängigen Software mit DIN- und Cinch-Anschluß auch für Micro Anschluß geeignet ● optische Aussteuerung über ein LED Display ● STEREO ● Wandlungsfähig ● 1MHz getaktet bietet der Golem Sound unglaubliche Sample Qualität.

Mono

Stereo

Software zum Golem Sound, stereofähig

DM 139,-
DM 189,-
DM 29,-

Technische Änderungen vorbehalten