

20/64/128

Das unabhängige Commodore-Magazin

SUPER:
Grosses
Amiga-
Poster!

UMFANGREICH:
Neue Listings
für VC 20
und C 64

HILFREICH:
CP/M- Das
unbekannte
Wesen?

TRICKREICH:
Umgang mit
C16 und Plus 4

So koppeln
Sie eine
elektronische
Schreib-
maschine
mit Ihrem
Computer

KAUFBERATUNG -
SERVICE - TIPS & TRICKS



SONDERHEFT Nr. 1/86
DM 14,80/ÖS 124/SFR 14,80

**VC 20
+ C64
PLUS 4
C16/116
128 PC**

**DAS
GROSSE
SONDER
HEFT**



**Tests
Tips
Tricks
Kauf-
beratung**

**Ein Muss
für jeden
Commodore-
User!**

**Ab
11. April
an
Ihrem
Kiosk!**

**Rund
150 Seiten**

INHALT

COMMODORE
Nr. 4/86 WELT

TEST & TECHNIK

Der Amiga:

Es gibt ihn endlich – für unter 6000 DM!

Ab Seite 4

CP/M:

So schwer ist es nicht, mit CP/M zu arbeiten –

Eine neue Serie der COMMODORE-WELT

Ab Seite 9

BILDSCHIRMBRIEF:

Warum schreiben Sie noch Briefe?

Senden Sie Ihre Mitteilungen doch per Diskette oder Kassette!

Ab Seite 12

TIPS & TRICKS:

Der Cursor muß nicht immer blinken – und andere Farben können es auch sein

Ab Seite 16

STRUKTOGRAMME:

Eine nützliche Hilfe beim Programmieren

Auf Seite 17

GEWUSST WIE:

Die Schleife als Stolperstein

Ab Seite 18

EINSTELLHILFE:

Jetzt haben verstellte Tonköpfe keine Chance mehr

Auf Seite 19



Er wird die Sensation der Hannover-Messe: der Amiga. Auf unserem Titelbild in voller Aktion – auf der nächsten Doppelseite als Poster!

DATENSCHUTZ:

Damit niemand mehr in Ihren Programmen schnüffeln kann

Ab Seite 52

TIPS & TRICKS:

Diesmal für den C16 und Plus 4

Ab Seite 54

BÖRSE:

Zum Suchen und Finden – der Kleinanzeigenteil in der COMMODORE-WELT

Ab Seite 55

TIPS & TRICKS:

Mit zwei Floppies richtig programmieren – gewußt wie

Ab Seite 57

DRUCKER:

Welcher ist der richtige? COMMODOREWELT gibt handfeste Tips vor dem Druckerkauf

Ab Seite 58

SELBERMACHEN:

So bauen Sie sich Ihr Interface –, um eine elektronische Schreibmaschine an Ihren C64 anzukoppeln

Ab Seite 60

LISTINGS

FÜR DEN C 64:

Adreßverwaltung

Ab Seite 21

Weitsprung

Ab Seite 24

17 & 4

Ab Seite 27

Biorhythmus

Ab Seite 30

Vokabel-Pauker

Ab Seite 35

FÜR DEN VC 20:

Kunden

Ab Seite 37

Wüstenrallye

Ab Seite 39

Mathe-Künstler

Ab Seite 41

Kaleidoskop

Ab Seite 43

Donald Duck

Ab Seite 45

Und zum Schluß die ganz große Sensation: Ein Megabyte-Speicher für Ihren C64 – per Software!

IMPRESSUM

COMMODORE-WELT

erscheint monatlich in der AKTUELL-Gruppe Werner E. Seibt

VERANTWORTLICH FÜR DEN

INHALT: Anton Kult/Werner E. Seibt

REDAKTION UND STÄNDIGE

MITARBEITER: Senator-Presseservice: Brigitte Gerl, Helmut Gerl, Rosemarie Huber, Dipl.-Ing. Rainer Krampe, Torsten Seibt, Georg Ziemann

ANZEIGENLEITUNG: Bruno Redase

ANSCHRIFT FÜR ALLE VERANTWORTLICHEN:

Postfach 1107, 8044 Unterschleißheim
Tel.: 089/129 80 14
Es gilt Preisliste Nr. 6 vom 1.1.86
Media-Unterlagen bitte anfordern

GRAFISCHE GESTALTUNG:
AK-Design, SPS

VERTRIEB: Verlagsunion Wiesbaden

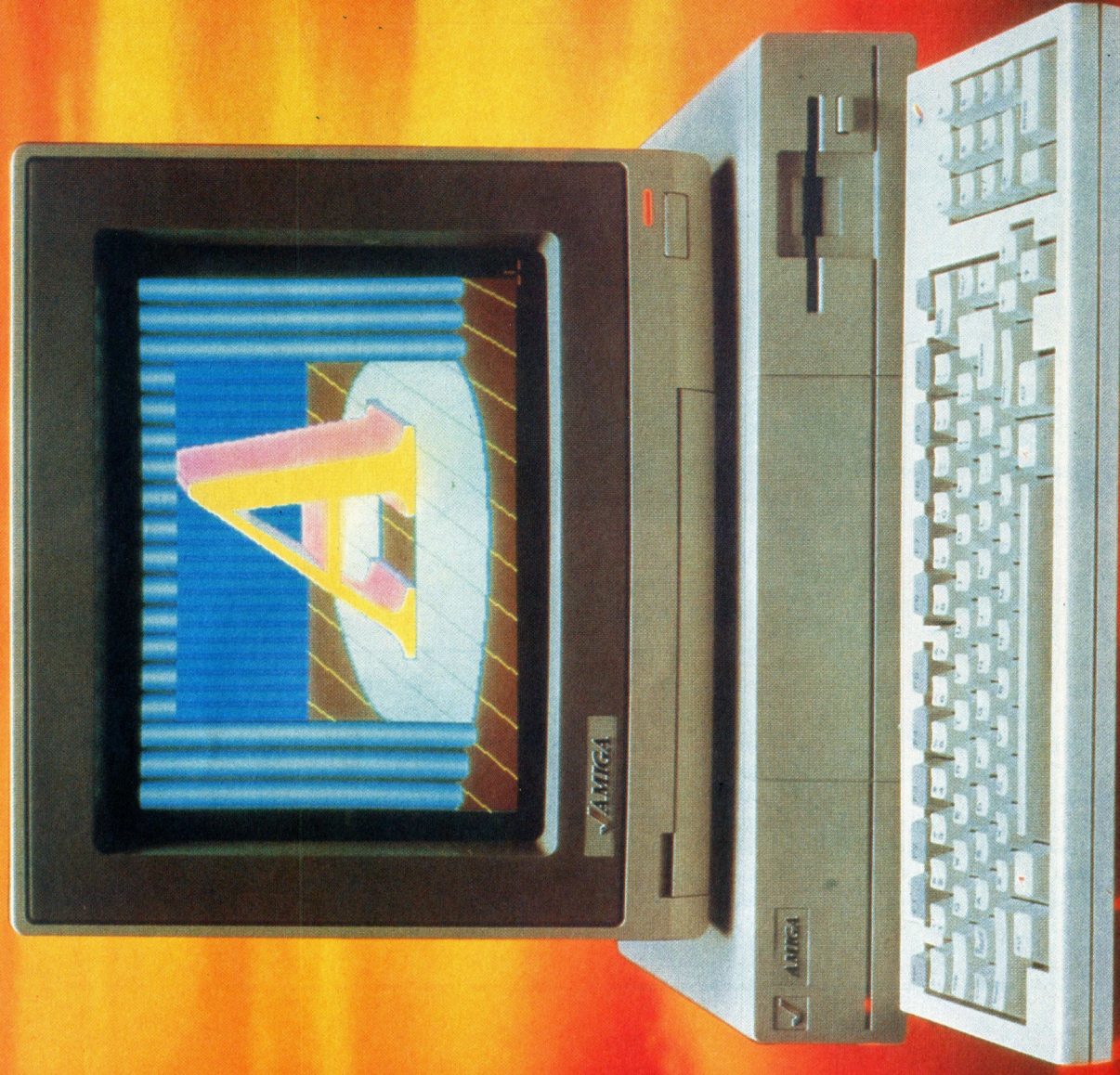
© 1986 by TI/CBM-Verlag Werner E. Seibt, Elisabethstr. 1, 8044 Unterschleiß-

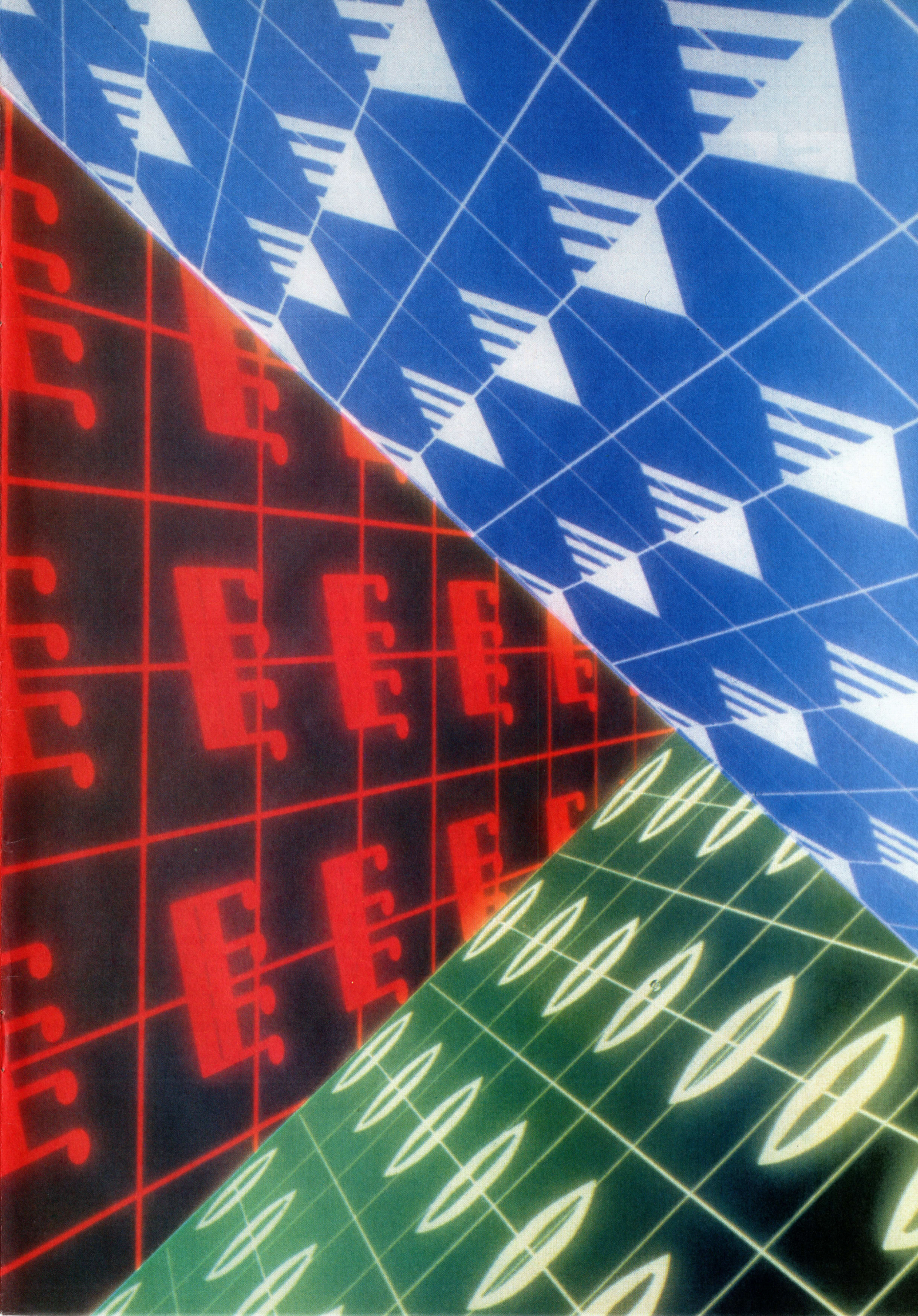
heim – SPS und Autoren. Für unaufgefordert eingesandte Manuskripte und Listings keine Haftung. Bei Einsendung von Texten, Fotos und Programmträgern erteilt der Autor dem Verlag die Genehmigung für einen einmaligen Abdruck und die Aufnahme in den Kassetten-Service. Alle in dieser Zeitschrift veröffentlichten Beiträge sind urheberrechtlich geschützt. Jedwede Verwendung ist untersagt. Namentlich gezeichnete Beiträge unserer Mitarbeiter stellen nicht unbedingt die Meinung der Redaktion dar.

Erstverkaufstag: 26. März 1986

© 1986 by TI/CBM-Verlag

COMMODOREWELT: DER AMIGA





ER IST DA

Lang ersehnt, heiß erfehlt: der Commodore Amiga, seines Zeichens Computer der neuen Generation. In Frankfurt wurde dieser mit Vorschlußparaden schon reichlich bedeckte Rechner nun endlich vorgestellt.

Winfried Hoffmann, Geschäftsführer bei Commodore Deutschland, meinte in seiner Eröffnungsrede zur Präsentation des Amiga selbstbewußt: „Der Amiga kann nicht mehr als andere Rechner. Er kann es lediglich viel besser.“ Und um dies zu verdeutlichen, wurde eine Show mit, von und über den Amiga veranstaltet, die eine solche Bezeichnung wirklich verdiente. Dem staunenden Publikum, rund 1600 Vertriebspartner, Händler und Journalisten, wurde in eindrucksvoller Weise die Qualität des neuen Mediencomputers vor Augen geführt. Angefangen mit der Begrüßungsmelodie, die mit Sphärenklängen die Gäste empfing (natürlich von Amiga gemacht) über Projektionen auf Großleinwänden bis hin zu Anwendungsbeispielen, welche keine Gelegenheit ausließen, den Zuschauer in Erstaunen zu versetzen. Da konnte die Entstehung eines Werbekonzeptes in hochauflösender Grafik genauso bewundert werden wie einige Kompositionsarbeiten auf dem Amiga. Und damit niemand meinte, dies wären Spielereien, kam eine junge Sängerin zum Zuge, welche alleine mit dem Amiga und einem angeschlossenen Keyboard ein ganzes Orchester auf die Bühne zauberte.

Die neue Computer-Form

Durch diese Vorführung setzte Commodore gleich die Anwendungsbereiche fest, für welche der Amiga konstruiert wurde. Er soll als ein Werkzeug für kreative Arbeit gesehen werden, und alleine dies stellt ein Novum auf dem Computermarkt dar. War es den Rechnern bislang vorbehalten, verwaltende Funktionen wie Textverarbeitung und Buchführung zu beschleunigen und zu erleichtern, soll nun der Amiga ein ganz neues Publikum vor die Tastatur bringen. Von seinen hervorragenden grafischen und

musikalischen Fähigkeiten können nun auch diejenigen Anwender profitieren, deren Arbeit auch heutzutage zumeist von Hand erledigt werden muß. Speziell angesprochen sind hier vor allem Designer jedweder Art oder beispielsweise Musiker.

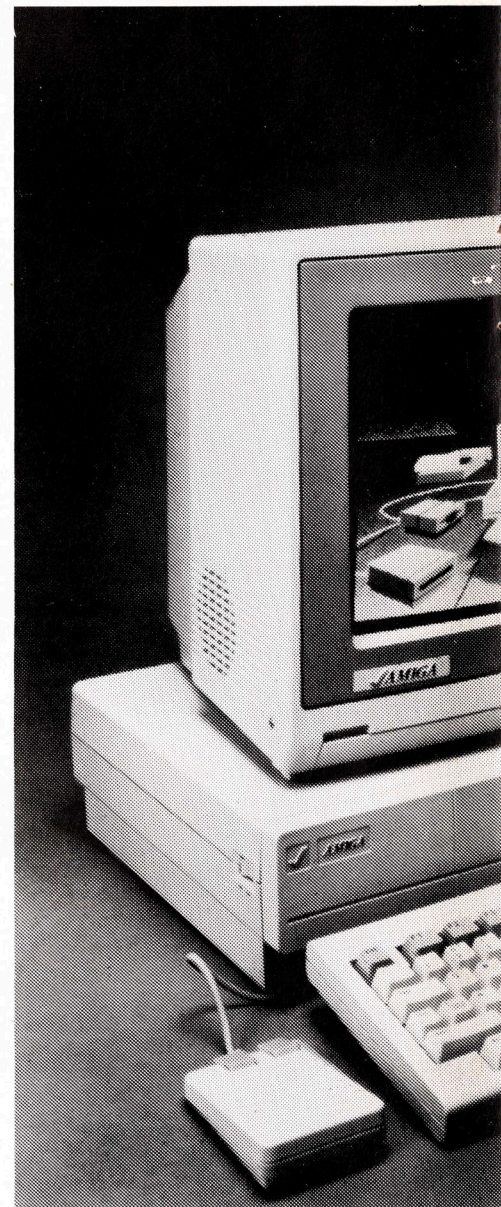
Jeder Ausgabe ihren Chip

Für diesen Zweck wurden Amiga neben dem bewährten Motorola 68000 Prozessor mit einer Taktfrequenz von 7,16 MHz drei von Commodore entwickelte Chips mit auf den Weg gegeben, welche sich um die Grafik-, Sound- und Rechenfähigkeiten des Amiga bemühen. Sie übernehmen gemeinsam die Arbeit, welche in herkömmlichen Rechnern bislang von einem streßgeplagten Chip alleine erledigt wird. Hier wurde der Begriff des „Multiasking“, der Fähigkeit, mehrere völlig verschiedene Arbeiten gleichzeitig durchzuführen, in Idealform in die Tat umgesetzt.

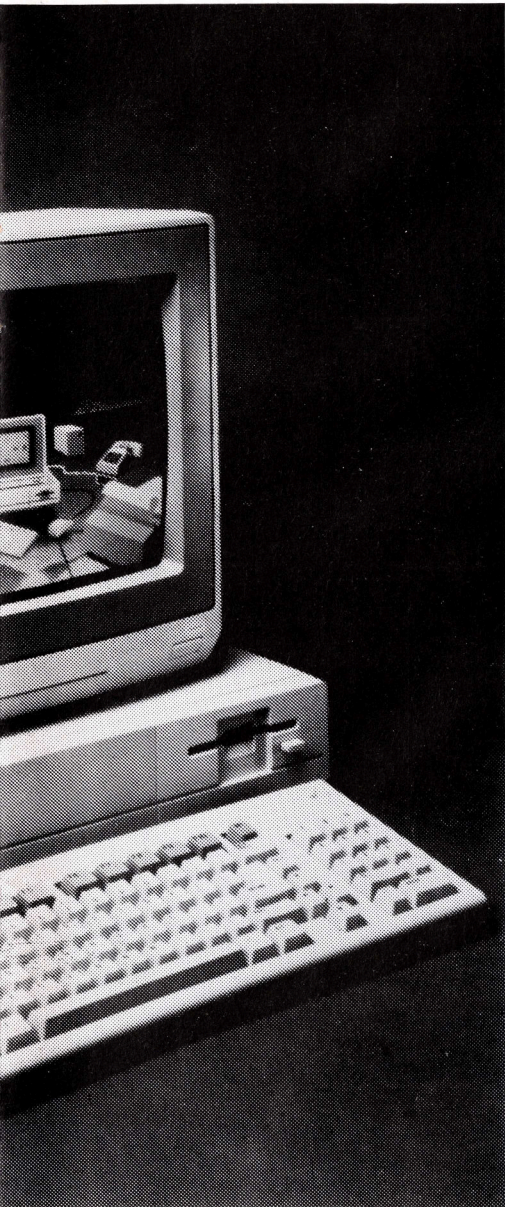
Betriebs-System: Amiga DOS

Um dies steuern zu können, wurde für den Amiga ein eigenes Betriebssystem, das Amiga-DOS, geschaffen. Es wird im Gegensatz zu den herkömmlichen Betriebssystemen in einen separaten 256 KB RAM Bereich geladen, welcher anschließend schreibgeschützt arbeitet wie ein ROM. Dadurch ergibt sich der Vorteil, daß das Betriebssystem offen ist und jederzeit aktualisiert werden kann. So bleiben in der Grundversion volle 512 KB im Arbeitsspeicher frei belegbar. Erweitert werden kann der Amiga auf bis zu 8,5 Megabyte im Direktzugriff.

Eine weitere Neuerung wird durch die Arbeitsteilung der Chips für Grafik und Rechnerfunktion ermöglicht: Auf dem Amiga können beliebig viele „Windows“, also



Bildausschnitte, dargestellt werden. Dadurch können verschiedene Arbeitsabläufe gleichzeitig, eben im Multiasking, auf dem Bildschirm verfolgt werden. Diese Windows können auf eine gewünschte Größe „gezoomt“ werden, ohne die dahinter ablaufenden Darstellungen zu beeinträchtigen. Auf diese Weise ist ein völlig neues Arbeiten möglich. Denn: Noch während der Rechner beispielsweise die Funktion eines Verbrennungsmotors simuliert, können darüber Berechnungen angestellt werden. Die Qualität der möglichen Bilder übertrifft dabei bei weitem die bisher von PC's bekannten Normen. In der Auflösung von 320x200 Punkten können sogar bis zu 4096 Farbtöne gleichzeitig dargestellt werden.



Dies erschließt dem Amiga ein weiteres Anwendungsgebiet: Bildschirmtext. Er kann nicht nur Btx-Seiten speichern und wiedergeben, sondern diese über den Videoport auch entwerfen oder umarbeiten. So wird der Amiga das preiswerteste und leistungsfähigste Bildschirmtext-Editiergerät auf dem Markt sein.

Audio-Visuelle Perspektiven

Im Videobereich liegt überhaupt eine der Stärken des Amiga. Über Zusatzgeräte ist es möglich, die Bilder von Videokameras auf dem Bildschirm darzustellen. Die so entstehenden Benutzermöglichkeiten liegen auf der Hand: Es können einzelne Bilder abgespeichert und be-

arbeitet werden, etwa durch farbliche oder formale Veränderung. In einfachster Weise können Details aus Fotos vergrößert oder verkleinert, vervielfältigt und coloriert werden. Auch im Audio-Bereich setzt der Commodore neue Maßstäbe. Natürlich ist es auch hier möglich, Töne von externen Geräten wie Mikrofonen, Verstärkern oder Tonbandgeräten in den Amiga einzuspeisen, zu verändern, speichern und wiederzugeben. Und dies ganz nebenbei über zwei Stereo-Ausgangskanäle und in HiFi-Qualität. So können beispielsweise eigene Tonbandaufnahmen mit einer anderen Akustik versehen werden. Auch der synthetischen Erzeugung von Tönen, Musik und Sprache steht nichts im Wege, der Amiga schafft einem Komponisten ebenso leicht ein Orchester im Westentaschenformat wie er Fehlermeldungen sprachlich ausgeben kann. Diese akustische Kommunikation wird der Anwender dankend begrüßen, wenn er mehrere Windows gleichzeitig beobachten und bearbeiten muß. Auch auf dem Ausbildungssektor und bei der Bedienung durch Behinderte, beispielsweise durch Blinde, ergeben sich durch diese Form des Dialoges völlig neue Perspektiven im Umgang mit dem Medium Computer.

Voll IBM-Kompatibel

Auch ohne diese Möglichkeiten gerät die Arbeit mit dem Amiga nicht zur Schufferei. Sinnigerweise hat Commodore der Tastatur einen Abstellplatz unter der Zentraleinheit beschert, was gleich darauf schließen läßt, wie die hauptsächlich Bedienung des Rechners von-statten gehen soll: mit der „Maus“. Und tatsächlich wurde in Frankfurt demonstriert, wie einfach und problemlos dies erledigt wird. Zur grafischen Anwendung des Amiga wird die Maus zum einzigen Eingabegerät. Mit ihr werden Diskettenkommandos gegeben, Farben ausgewählt, Bildschirmausschnitte bestimmt und vieles mehr.

Commodore weist auf die Entwicklung von etwa sechzig Programmpaketen speziell für den Amiga hin, welche sämtliche Anwendungsgebiete abdecken sollen. Dies ist noch lange kein Grund, mit dem

Prozessoren/Controller

CPU:

Motorola 6800
– 7,16 MHz, 16/32 bit

Grafik- und Zeichentrick-Chip „Agnus“

Bit-Blitter

- Hochgeschwindigkeits-Daten-transfer mit Verknüpfung von Daten aus drei verschiedenen Quellen für die Bewegung von Objekten
- schnelles Linien-Zeichnen und Flächen-Füllen mit 1 Mio Punkte pro Sekunde
- erkennt Kollisionen zwischen zwei bewegten Objekten

Cooper/Coprozessor

- ist mit Bildschirm synchronisiert, wird von Rasterstrahl gesteuert
- versorgt die Register von Agnus, Paula und Denise
- kontrolliert 25 DMA-Kanäle

Video Chip „Denise“:

Auflösungen

- 320 x 200
- 640 x 200
- 640 x 400

Farben

- 32 bei 320 Spalten
- 16 bei 640 Spalten aus 4096 Farbtönen
- bis zu 4096 Farbtöne je nach Betriebsart gleichzeitig darstellbar

Sprite-Controller/8

- faßt auf Wunsch zwei Sprites zusammen
- Kollisionsdetektor mit Prioritätenfestlegung bei Überschneidung

Textbildschirm

- wahlweise 60 oder 80 Zeichen breit, in Farbe

Port-Chip „Paula“:

DMA-Steuerung

- steuert den RAM-Zugriff aller berechtigten Bausteine

I/O-Steuerung

- serielle Schnittstelle
- parallele Schnittstelle
- Control-Ports
- Tastatur
- Audio-Ausgabe
 - 4 Tonkanäle für zwei Stereoausgänge
 - programmierbare Amplitude und Sampling-Rate
 - 9 Oktaven

- komplexe Wellenform
- Amplituden- und Frequenzmodulation
- Audio-Eingabe
- Steuerung eingeleiteter Naturstimmen oder von Musik-Originalen (über Spezialinterface)

Speicher

Arbeitsspeicher:

- 768 KB RAM
- 512 KB frei belegbar
- 256 KB separater, schreibgeschützter RAM-Bereich für Betriebssystem / Nach Laden wie ROM
- erweiterbar bis 8,5 Megabyte im Direktzugriff

Massenspeicher:

Integriert

- 3 1/2 Zoll Mikro Floppy-Disk
- Double sided, double density
- 880 KB formatiert

Optional

- 3 1/2 Zoll Mikro Floppy-Disk
- 5 1/4 Zoll MS-DOS Format
- Harddisk
- Fremde Systeme
- optische Speichersysteme
- Videorecorder

Eingabeelemente

Tastatur

- prozessorgesteuert
- 89 Tasten, numerische und Cursor-Tasten
- separater Rechenblock
- 10 Funktionstasten, zwei Amiga-Tasten
- Tastatur unter Gehäuse ein-schiebbar

Maus

- zwei Bedienungsknöpfe

Audio-Eingabe

- über HiFi-Geräte, Videorecorder, CD-Player
- Mikrofon/Verstärker
- Musikinstrumente mit Midi-Interface

Video-Eingabe

- mit Amiga-Genlock-Interface / Synchronisation von Computer mit Videoquellen und Darstellung als hintere Bildebene auf Monitor
- über Videokamera
- Bildplatte

- Fernseher
- Bildschirmtext
- Videotext
- Scanner
- Bildbe- und verarbeitung über Digitalisierung mit Amiga-Frame-Grabber/Videodigitizer

Schnittstellen

Floppy-Disk Ports:

Externe Diskettenstationen

- für 3 Laufwerke
- 3 1/2 oder 5 1/4 Zoll

Parallel:

programmierbarer Port

- für Ein- und Ausgabe
- normalerweise als Centronics konfiguriert

Seriell:

Programmierbarer Port

- Baudraten bis zu 31.250
- RS 232 C
- Midi über Adapterstecker

Controller-Ports:

2 Ports

- Maus
- Grafiktabellen
- Lightpen
- Steuerknüppel
- Drehregler

Video/Audio:

3 Ports

- Cnynch Stereo/Audio
- RGB analog, digital
- Composite Video

Tastatur:

1 Port

Expansion:

Herausgeführter Prozessorbus

- RAM-Erweiterungen
- Harddisk
- Spezialperipherie
- Meßwerterfassung
- Coprozessoren

Ausgabemedien

Monitor:

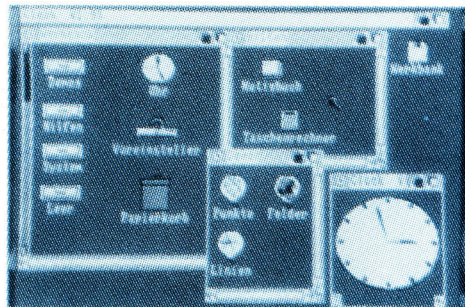
Colour Monitor

- 14 Zoll

Sprachausgabe:

- Eingebauter Lautsprecher oder über externe Tonquellen in HiFi-Stereo-Qualität

Kauf dieses Rechners zu zögern, denn mit einem MS-DOS Emulator ist der Amiga voll IBM-kompatibel und kann auf das umfangreiche Spektrum der erhältlichen Software zurückgreifen. So kann ein Amiga-Käufer in Ruhe mit den bekannten (und bewährten) Programmen für IBM und Kompatible arbeiten und abwarten, bis entsprechendes für den Amiga erhältlich ist. Bereits jetzt bzw. in den nächsten zwei Monaten können vom Ingenieurbüro Kalawsky in Weiterstadt diverse Hardware-Ergänzungen, von Verbindungskabeln über externe Laufwerke bis hin zu Speichererweiterungen erworben werden. Dies verdeutlicht, daß die Arbeit an Hard- und Software speziell für den Amiga auf Hochtouren läuft und bereits jetzt erste Früchte trägt. Ebenfalls jetzt schon erhältlich sind (leider nur in Englisch) vier Reference Manuals zu einem jeweiligen Preis von ca. 60 DM.



Im Lieferumfang des Grundpaketes, welches laut Commodore zu einem Preis von 5500 bis 6000 DM erhältlich sein wird, sind die Zentraleinheit mit eingebautem 3 1/2 Zoll Disk-Laufwerk, ein Farbmonitor, die Maus sowie natürlich die Tastatur enthalten, weitere Preise waren leider noch nicht zu erfahren. Das Laufwerk wird die doppelseitigen 3,5er Micro-Disketten bei doppelter Schreibdichte mit 800 KB versehen können, erhältlich sind zusätzliche Laufwerke im 5 1/4 und 3 1/2 Zoll Format.

Es bleibt abzuwarten, ob der Amiga nun wirklich in der von Commodore prophezeiten Weise seinen Siegeszug durch die Büros antreten wird und bald die Designer, Architekten, Komponisten und ... ihre Entwürfe statt in Aktenkoffern auf einer Microdiskette transportieren und ihren Kunden vorlegen werden.

(T. Seibt)

CP/M- DAS UNBEKANNTE WESEN

Da rief doch letztens ein Leser während des Leserservice am Mittwoch an und fragte allen Ernstes, wie er bitteschön eine Diskette formatieren soll. Genauso ernst antworteten wir ihm. Erst nach und nach stellte sich heraus, daß der gute Mann „versehentlich“ sein CP/M Betriebssystem geladen hatte. Seine Äußerungen gaben uns zu denken. Das Ergebnis sehen Sie hier: eine kleine Serie über CP/M.

Was CP/M ist, wurde an dieser Stelle schon des öfteren besprochen. Es ist keine, wie wir auch oft am Lesertelefon zu hören bekamen, Programmiersprache, sondern ein reines Betriebssystem. Man benötigt es, sehr allgemein gesagt, um mit der Außenwelt (hier sind natürlich die angeschlossenen Peripheriegeräte gemeint) in Kontakt zu treten.

CP/M ist im Vergleich zu anderen Betriebssystemen schon sehr alt. Man kann sogar sagen, daß es eigentlich der Urvater aller heutigen auf dem Markt befindlichen Betriebssysteme ist. Es wurde schon zu Beginn der siebziger Jahre von einem ehemaligen Mitarbeiter des Chipherstellers Intel entworfen. Er gründete die Firma Digital Research, die heute, neben der CP/M Serie für verschiedene Prozessortypen, auch die graphische Betriebssystemerweiterung GEM vertreibt.

Wozu ein Betriebssystem??

Die Frage ist berechtigt, kommt doch der 128'er augenscheinlich ohne aus. Aber nur auf den ersten Blick, denn die modernen Homecomputer haben das Betriebssystem schon fest eingebaut. Für die Hersteller hat das natürlich den Vorteil, daß sie keine Lizenzgebühren an die Programmierer zahlen müssen. Die Anwender halten allerdings ein starres System in der Hand und können nicht nach Herzenslust „patchen“, wie die Verbesserung eines Programmes auf neudeutsch heißt. Patchen bedeutet, daß auf die ursprünglich

„defekte“, d.h. schlechte Stelle, ein Flickchen aufgesetzt wird, der eine berichtigte Version enthält. Seit beispielsweise das Textprogramm WordStar auf dem Markt ist, sind in diversen Fachzeitschriften unzählige WordStar-Patches veröffentlicht worden.

System) regelt die Ansteuerung der an den Computer angeschlossenen Peripheriegeräte.

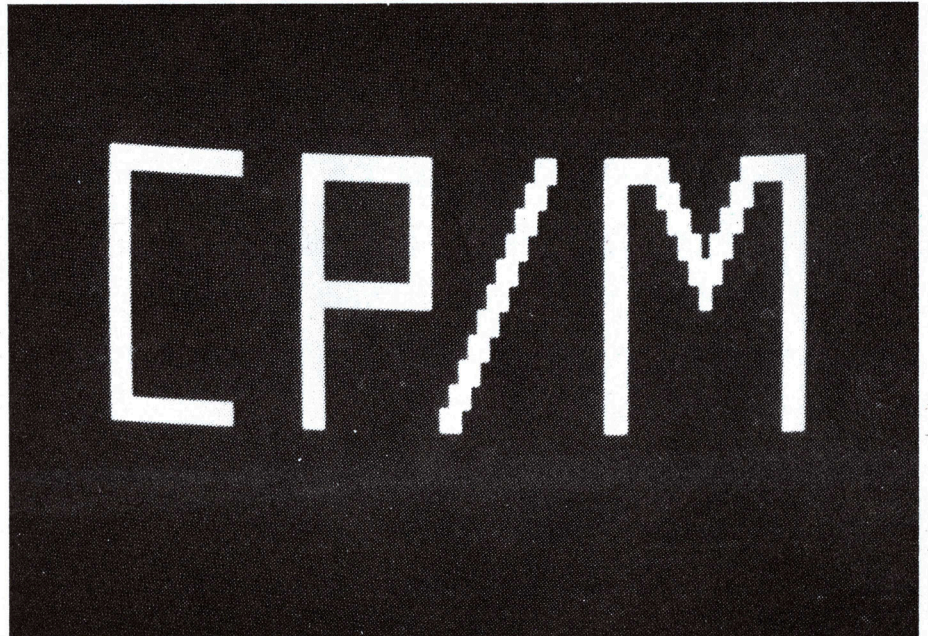
Das BDOS (Basic Disk Operating System) führt die vom BIOS kommenden Anweisungen aus. Es ist vollkommen hardwareunabhängig, während das BIOS auf jeden Rechner extra zugeschnitten werden muß.

Der Teil, mit dem Sie als CP/M Benutzer auf dem 128'er am meisten in Berührung kommen, ist der CCP, der Console Command Processor. Er nimmt Ihre Kommandos entgegen, übergibt entsprechende Parameter und weist Sie auf fehlerhafte Kommandos hin.

Der CCP kennt sechs Befehle, die er fest eingebaut hat. Dies sind:

DIR	Directory
ERA(SE)	Löschen von Dateien
REN(AME)	Umbenennen von Dateien
USER	Mehrplatzbetrieb
TYPE	Anzeigen von Dateien
SAVE	Abspeichern von Dateien

Mit fest eingebaut ist gemeint, daß CP/M diese Kommandos direkt



Ein ladbares Betriebssystem wie CP/M besteht in der Regel aus mehreren Programmen, von denen jeweils einige fest im Speicher stehen (resident), andere erst bei Bedarf nachgeladen werden. Die wesentlichen Teile von CP/M sehen Sie in Bild 1.

Gehen wir kurz auf die Bestandteile ein.

Das BIOS (Basic Input Output

ausführt, ohne auf der Diskette ein Programm zu suchen, das den Inhalt einer Datei ausgibt (TYPE). Wir werden auf diese Kommandos noch zurückkommen.

Sicher haben Sie gemerkt, daß in unserer Auflistung noch etwas fehlt. Richtig, die TPA oder Transient Program Area. Das ist der Bereich, in den Anwenderprogramme geladen und ausgeführt werden. Die

SERVICE

TPA hat auf Ihrem 128'er eine Größe von 59kB.

Jetzt wird's ernst: Die ersten Befehle

Verlassen wir aber die graue Theorie und wenden wir uns dem praktischen Arbeiten mit CP/M zu.

Als Erstes wollen wir uns einmal das Inhaltsverzeichnis der CP/M Diskette anschauen. Auf der Vorderseite befinden sich die Dateien aus Bild 2, die Rückseite enthält die Dateien aus Bild 3. Da die Diskette auch mit den Laufwerken

1541/1470 gelesen werden kann, ist sie einseitig ausgeführt, der Lesekopf der 1571 greift also nur auf eine Spur zu.

Die Dateien aus Bild 2 im einzelnen:

CPM+ SYS System
CCP COM Command Console Processor

HELP COM Hilfsprogramm

HELP HLP Datei zu HELP. HLP

KEYFIG COM Programm zum

Ändern jeder Taste auf dem Computer

KEYFIG HLP Hilfsdatei zu KEYFIG
FORMAT COM Formatierprogramm
PIP COM Perpheral Interchange Program

DIR COM Directory
COPYSYS COM Kopiert Systemdateien

Noch eine kurze Anmerkung zu DIR. Sicher haben sie noch in Erinnerung, daß DIR ein residentes Kommando ist. Da die meisten

Alle PIP-Befehle und Optionen auf einen Blick

Logische Ausgabegeräte unter CP/M:

CON — Konsole, Bildschirm
RDR — Lochstreifenleser, Fernschreiber

PUN — Lochstreifenstanzer, Fernschreiber

LST — Drucker

— Diskette vollständig kopieren: B:=A:*. *

— Dateien einer Gruppe kopieren: B:=A:*.TXT

— Datei kopieren und Namen ändern: B:=A:NEWNAME.TXT=OLDNAME.TXT

— Dateien kopieren und zusammenfügen: B:=A:NEW.DAT=FILE1.DAT,FILE2.DAT

— Kopieren zwischen Benutzerbereichen: Von 0 nach 3: B:CG3J=A:FILE1.DAT VON 3 NACH 0:

B:=A:FILE1 [G3]

Der Benutzerbereich 0 ist standardmäßig voreingestellt.

— Datei auf Drucker ausgeben: LST=FILE1.DAT

PIP-OPTIONEN

[A] Archiv-Funktion. Kopiert nur die Dateien, die nach dem letzten Kopiervorgang geändert wurden.

[C] Confirm. PIP erwartet bei jeder Datei eine Kopierlaubnis.

[Dn] Delete. Löscht alle Buchstaben nach der Spalte n. LST:FILE1.DAT[D40]

Listet alle Zeichen bis zur 40-ten Spalte, der Rest bleibt unberücksichtigt.

[E] Echo. Alle Zeichen werden während des Kopierens auf dem Bildschirm dargestellt.

[F] Filter. Entfernt sämtliche Form-Feed Zeichen aus einer Datei.

[Gn] Kopiert Daten aus dem angegebenen Benutzerbereich in den aktuellen Bereich.

[H] Hex-Format. Es findet während des Kopierens von Hexadezimalen Dateien eine Überprüfung auf einwandfreies Intel-Format statt.

[I] Ignore. Sämtliche Nullbytes einer Hex-Datei werden übergangen.

[K] Kill. Dateiname wird während des Kopierens nicht mit angezeigt.

[L] Lower-Case. Umwandlung einer Datei in Kleinbuchstaben.

[N] Number. Jede Zeile erhält eine Zeilennummer.

LST: Artikel. TXT[N]

→ 1: CP/M, das unbekannte Betriebssystem

2: oder:

3: Wie formatiere ich eine Diskette?

LST: Artikel. TXT[N2]

→ 000001 CP/M, das unbekannte Betriebssystem
000002 oder:

000003 Wie formatiere ich eine Diskette?

[O] Objekt-File. Wird nur dann verwendet, wenn keine .COM oder Text-files übertragen werden.

[Pn] Page length. Setzen der Seitenlänge. Die Voreinstellung

ist 60 Zeilen, nach DIN hat eine Schreibmaschinen-seite 72 Zeilen.

[QString^Z]

Kopieren einer Datei bis zu dem für „String“ angegebenen Begriff.

LST:ARTIKEL.TXT[QARTIKEL~Z]

[R] Read File. Dient zum kopieren von Systemdateien.

B:=ADIE AUCH[R]

Die Datei „DIE AUCH“ wird auf der Diskette im Laufwerk B als Systemdatei abgelegt und beim Booten mit geladen.

[SString^Z]

Kopieren einer Datei ab dem für „String“ angegebenen Begriff.

LST:ARTIKEL.TXT[SDIE AUCH^Z]

[Tn] Tabulator. Setzt den Tabulator bei einem Ausdruck auf den Wert n.

[U] Upper case. Wandelt Kleinbuchstaben in Großbuchstaben um.

[V] Verify. Die kopierte Datei wird mit der Originaldatei verglichen.

[W] Write. Mit dieser Option können Dateien, die schreibgeschützt sind, überschrieben werden.

B:=A:PROTECT.COM[W]

Kopiert und überschreibt auf dem Laufwerk B eine möglicherweise schreibgeschützte Datei PROTECT.COM.

[Z] Zero Parity Bit. Löscht das achte Bit jedes Zeichens einer ASCII-Datei, da das ASCII-Format nur sieben Bit benötigt.

CP/M+ Befehle aber mit Optionen „behängt“ werden können, also zum Beispiel **DIR*.COM**, wird ein erweitertes DIR-Programm von der Diskette geladen, das dann die Ausführung übernimmt. Übrigens, **DIR*.COM** listet nur Dateien mit dem Zusatz **.COM**.

Universell: FORMAT.COM

In unserer heutigen Folge über CP/M wollen wir uns auf die Programme **FORMAT.COM** und **PIP.COM** beschränken. Sie sind mit die wichtigsten Programme auf der Diskette, ihre Funktionen lassen sich, zumindest bei **FORMAT.COM**, schon aus dem Namen ableiten.

weder mit „N“ für NO oder mit „Y“ für YES beantworten sollten. Sie sehen, die Benutzerführung ist in der englischen Sprache gehalten. Das bereitet manchen Anwendern erhebliche Schwierigkeiten, wie wir aus Äußerungen am Lesertelefon schließen können.

Wir haben nun eine formatierte, aber leere Diskette. Darauf wollen wir das Betriebssystem kopieren, damit wir die Originaldiskette gestrost beiseite legen können. In Zukunft arbeiten wir dann nur noch mit der Sicherheitskopie.

Mit dem oben schon erwähnten **PIP**-Programm können wir uns die Kopie anlegen.

Alleine die vielfältigen Möglichkeiten des **PIP** würden schon eine

mischt), haben sie es wesentlich einfacher. Sie müssen dann nicht jedesmal die Diskette wechseln. Geben Sie ein:

B:=A:*.

Auch hier werden alle Dateien kopiert, Sie können sich aber genüßlich im Stuhl zurücklehnen und zuschauen, wie andere (und wenn es nur zwei Laufwerke sind) für Sie arbeiten.

Nachdem **Pip** sich zurückgemeldet hat, kopieren Sie bitte auch die zweite Seite der Systemdiskette. Ist dies geschehen und wollen Sie keine weiteren Dateien kopieren, drücken Sie **Return** und befinden sich wieder im normalen CP/M Modus.

Aber **PIP** kann noch mehr. Außer Dateien von einer Diskette auf eine

```
A>dir
A: CPM+      SYS : CCP      COM : HELP      COM : HELP      HLP : KEYFIG    COM
A: KEYFIG    HLP : FORMAT   COM : PIP      COM : DIR      COM : COPYSYS   COM

dir
A: DATE      COM : DATEC    ASM : DATEC    RSX : DEVICE    COM : DIR      COM
A: DIRLBL    RSX : DUMP      COM : ED      COM : ERASE     COM : GENCOM    COM
A: GET       COM : INITDIR  COM : PATCH    COM : PIP       COM : PUT      COM
A: RENAME    COM : SAVE     COM : SET      COM : SETDEF    COM : SHOW     COM
A: SUBMIT    COM : TYPE     COM : SETUP    COM
A>
```

Vorder- (oben) und Rückseite der CP/M-Diskette

Sicher haben Sie es schon erraten: Bei **FORMAT.COM** handelt es sich um ein universelles Formatierprogramm. Universell deshalb, weil alle drei Floppyformate (1541, 1570 und 1571) mit diesem Programm bedient werden können.

Laden Sie also jetzt das Programm, und legen Sie eine neue Diskette bereit.

Auf Ihrem Bildschirm sollte folgende Zeile stehen:

A)FORMAT

Nach Drücken von **Return** sollte sich das Formatierprogramm melden. Übrigens müssen Sie die Extensions, die am Ende der Namen stehen, nicht mit eingeben. Wählen Sie jetzt das Format an, das Ihrem Floppy-Laufwerk entspricht. Legen Sie dann die neue Diskette in das Laufwerk und bestätigen Sie mit „**\$**“. Nach geraumer Zeit erscheint dann die Frage nach weiteren zu formatierenden Disketten, die sie ent-

ganze Serie füllen. Wir wollen es aber bei den wichtigsten Befehlen belassen. Rufen Sie also jetzt **PIP** auf.

A)PIP

Das Programm meldet sich mit

PIP Version 3.0

*

Anstelle des bei CP/M üblichen „**>**“ (hier heißt der Cursor „Prompt“) erscheint nun der sog. Wildcart. Geben Sie nun folgende Befehlssequenz ein:

***E:=A:*. * (Return)**

Dieser Befehle weist **PIP** an, sämtliche Dateien auf der Betriebssystemdiskette auf eine neue Diskette im virtuellen Laufwerk **E** zu kopieren. Das virtuelle Laufwerk **E** benutzen wir deshalb, damit wir keine Fehlermeldung vom System bekommen, da wir ja kein zweites Laufwerk angeschlossen haben. Sollten Sie allerdings glücklicher Besitzer zweier Floppies sein, egal ob 1541, 1570 oder 1571 (auch ge-

andere zu übertragen, sind unter anderem noch folgende Operationen möglich:

– Dateien gleicher Art auf eine andere Diskette kopieren:

***B:=A:*.TXT**

– Kopieren und gleichzeitig umbenennen:

***B:=A:NEWNAME.TXT=OLD-NAME.TXT**

– Kopieren und verbinden mehrerer Dateien:

***B:=A:NEW.DAT=FILE1.DAT, FILE2.DAT**

– Umbenennen und gleichzeitig in einen anderen Userbereich kopieren:

***B:=A:NEWFILE.TXTCG1J=OLD-FILE.TXT**

– Zeilen durchnummerieren (!!):

***B:=A:NEWTEXT.TXT=TEXT.TXTCNJ**

In den Beispielen war schon die Rede von Optionen, die Sie an jeden **PIP**-Befehl anhängen können. Diese Optionen werden nach dem Dateinamen in eckigen Klammern

angegeben. Es können auch mehrere Optionen auf einmal, jeweils getrennt durch ein Leerzeichen, verwendet werden. Für folgende Anwendungen sind die Optionen sehr nützlich:

- Verändern des Inhalts von Textdateien
- Bearbeiten der Form von Textdateien
- Übertragen von Dateiausschnitten
- Festlegung bestimmter Übertragungsbedingungen

Tabelle 1 zeigt sämtliche PIP-Befehle und Optionen mit Beispielen.

Zum Schluß dieser ersten Folge möchten wir noch einige Tips zu CP/M auf dem 128'er geben.

Um Ihnen die Arbeit zu erleichtern, haben die Programmierer eine Funktion eingebaut, die jeweils die letzte Kommandozeile zurückholt. Dies erreichen Sie entweder mit „CONTROL-W“ oder mit der „CURSOR-UP“ Taste. Das zuletzt eingegebene Kommando erscheint und sie können es editieren.

Die Bedeutung der DISK STA-

TUS LINE unten rechts scheint noch vielen Lesern unklar zu sein. Das Format

O D T T S S

im einzelnen:

- O – Operation, entweder Read oder Write
- D – Floppyadresse, A,B,C oder D
- TT – die gerade gelesene Spur (Track)
- SS – der momentane Sektor (Sector)

Die DISK STATUS LINE ist mit der Tastenkombination RUN/STOP-CONTROL abschaltbar.

Irgendwann werden Sie sicher einmal in die Lage kommen, Daten oder Programm lesen zu müssen, die nicht mit einem 128'er geschrieben worden sind. Besitzer der 1570 und 1571 sind da fein raus, sind doch diese Floppies in der Lage, einige Fremdformate lesen zu können. Die 1541 schafft das nicht. Ohne Schwierigkeiten werden die Disketten der folgenden Computer verarbeitet (die 1570 schafft natürlich nur die einseitigen Formate):

EPSON QX 10	512 BYTES, DS, 10 S/T
IBM-8SS (CP/M86)	512 BYTES, SS, 8 S/T
IBM-8DS (CP/M86)	512 BYTES, DS, 8 S/T
KAYPRO II	512 BYTES, SS, 10 S/T
KAYPRO IV	512 BYTES, DS, 10 S/T
OSBORNE DD	1024 BYTES, SS, 5 S/T

Die erste Zahl gibt die Anzahl der Bytes pro Block an, DS/SS bedeutet Double bzw. Single Sided und S/T zeigt die Anzahl der verwendeten Sektoren pro Spur an.

Die Bildschirmausgabe auf dem 128'er geht recht schnell vonstatten. Anstatt jedesmal CONTROL-S zu drücken, können Sie ein Anhalten der Bildschirmausgabe auch mit der NO SCROLL Taste erreichen.

Soweit der erste Teil der Serie über CP/M auf dem 128'er. Im nächsten Heft werden wir etwas näher auf das Programm KEYFIG und einige kleinere Dateien der Utility-Diskette eingehen. (js)

BRIEFE PER BILDSCHIRM

Dieses Programm erlaubt das Schreiben eines Briefes am Bildschirm, wobei der Briefinhalt zusammen mit dem Programm abgespeichert und dem Empfänger zugesandt wird. Dieser sieht dann nach dem Starten des Programms gleich die erste Bildschirmseite und kann mit ‚w‘ weiterblättern. Nach der letzten Seite kommt wieder die erste ...

Der Platz reicht für 33 Bildschirmseiten, die Text oder Graphik enthalten können. Nachdem Sie das Programm abgetippt (wobei die REM-Zeilen und die Leerzeichen weggelassen werden können) und abgespei-

chert haben, erscheint nach Eingabe von RUN das Menue, mit dem Sie zwischen folgenden Punkten wählen können:

1: Bilder editieren:

Zuerst erscheint die Frage, ab welcher Seite auf Graphik umgeschaltet werden soll. Dieser Wert wird auch in die Speicherstelle 2056 (entspricht dem zweiten Doppelpunkt in Zeile 10) gespeichert. Beim Ansehen der Bildschirme wird dann immer an dieser Stelle umgeschaltet. Der Bildschirm wird nun gelöscht und kann beschrieben werden, wobei die Cursortasten, CLR/HOME und INST/DEL beliebig eingesetzt werden

können. Man muß nur zwei Dinge beachten: Die Stelle ganz unten rechts darf nicht beschrieben werden, da sonst der Bildschirm nach oben scrollt und statt der RETURN-Taste muß man SHIFT/RETURN verwenden, wenn man den Cursor an den Anfang der Folgezeile befördern will. Ist der Text (bzw. die Graphik) fertig, wird sie mit der RETURN-Taste in den Speicher verschoben und man hat nun die Möglichkeit, mit ‚w‘ die nächste Seite zu beschreiben oder mit ‚m‘ in das Menue zurückzukehren. Vorsicht!. Wenn die Briefseiten beschrieben sind, kann nicht mit Menuepunkt 1 weitergeschrieben werden, da hierbei immer mit der 1. Seite angefangen wird (Siehe Punkt 3).

2: Bilder ansehen

Hier werden die bereits

beschriebenen Seiten angezeigt. Mit ‚w‘ kann man weiterblättern. Wenn dabei ein kurzes Aufflackern des Bildschirms festgestellt wird, so liegt das an den Zeilen 220 und 230 und an der Tatsache, daß Sie einen C64 der älteren Bauart besitzen. Wenn nach einem Bildschirm löschen und POKE 1024,65 kein A oben links sichtbar wird, haben Sie das alte Kernal. Das ‚GOSUB220‘ kann bei neueren Computern durch ‚PRINT (CHR\$(147))‘ ersetzt werden.

3: Bilder verändern

Hiermit kann man Bilder verändern oder weiterschreiben. Man hat auch die Möglichkeit, sich mit dieser Funktion im Speicher umzusehen, um sich zum Beispiel anzusehen, wo Textstellen oder Graphik von zuvor geladenen Programmen abgespeichert waren. Die

Bildschirmseite wird angezeigt und (nach Veränderung) mit RETURN wieder zurückgespeichert.
p: Bilder drucken

Der Bildschirmcode wird in Zeile 208 in Commodore-ASCII umgerechnet und ausgedruckt. Wenn der Drucker nur Standard-ASCII versteht, muß man das REM in Zeile 212 weglassen.

e: Programm beenden

Die Werte für das Basicende in den Speicherstellen 45 und 46 werden berechnet und als POKE-Anweisung auf den Bildschirm gedruckt. Bestätigt man mit RETURN, wird dadurch das Basicende hinter die letzte Bildschirmseite verlegt und man kann alles zusammen abspeichern. In der Speicherstelle 2055 ist dann die Anzahl der gespeicherten Bilder enthalten. Wird das Programm nun gestartet, so überprüft Zeile 62, ob Bilder vorhanden sind und es wird gleich das erste Bild angezeigt.

Mit ,w' kann man weiterblättern und mit ,m' ins Menue gelangen.

Anwendungsbeispiele

- a) Briefe an C64-Besitzer schreiben
- b) Anlegen einer Wissenskartei und diese dann auch ausdrucken.
- c) Mit Menüpunkt 3 im Speicher blättern, wenn zuvor ein anderes Programm lief. Man kann z.B. Müllstrings erkennen ...
- d) Durch leichte Änderungen kann man mit dieser Methode leicht eigene Texte (Spielanleitung ...) oder Graphiken in selbst geschriebene Programme einblenden. Dazu muß man den Anfang des Bildspeichers (,ab' in Zeile 52) und die Werte für die Blockverschieberoutine (in

162-180) der Programmlänge anpassen (siehe e). Danach kann man die Bildschirmseiten beschreiben und das Programm mit ,e' verlassen. Zur Sicherheit abspeichern.

Die Werte für das Basicende in 45/46 merken! Lädt man nun das eigene Programm, so befindet sich der Text ja noch im Speicher. Man muß nun nur noch die Werte für 45/46 einPOKEn und alles zusammen abspeichern. Ins eigene Programm muß man natürlich nur den Teil, der für die Anzeige zuständig ist (38-46, 118-132 und 176-182) übernehmen.

e) Das Maschinenprogramm zur Speicher-verschiebung kann man für andere Zwecke sehr gut einsetzen. Es verwendet die Blockverschieberoutine (SYS41919). Hierzu werden in die Speicherstellen

861/862 das LO/HI-Byte des Anfangs des alten Blocks,

863/864 das LO/HI-Byte vom Ende+1 des alten Blocks und

865/866 das LO/HI-Byte vom Ende+1 des neuen Blocks

hineingePOKEt. Das LO- bzw. HI-Byte von X errechnet sich wie folgt:

$HI = INT(X/256); LO = X - 256 * HI$

Will man zum Beispiel den Interpreter ins RAM verlegen (40960-49151), so gibt man ein:

POKE861,0:POKE862,160:POKE863,0:
POKE864,192:POKE865,0:POKE866,192:
SYS828

Das Maschinenprogramm ist übrigens im Speicher frei verschiebbar.

```

10 rem"::bildschirmbrief=====64
12 rem (p)   cbm revue team           =
14 rem -----
16 rem (c) by
18 rem r.schmid-fabian
20 rem postfach 105027
22 rem 6900 heidelberg
24 rem c-64 + 1530/1541
26 rem -----
28 cl$=chr$(147):c4$=chr$(17)
29 up$=chr$(145)+chr$(145)+chr$(145)
30 rem steht in adresse 2055 kein
32 rem 'doppelpunkt' erscheint
34 rem die erste briefseite
36 rem ***** maschinenprogramm *
38 fori=828to860:reada:pokei,a:nexti
40 data 173,93,3,174,94,3,133,95,134,96
42 data 173,95,3,174,96,3,133,90,134,91
44 data 173,97,3,174,98,3,133,88,134,89
46 data 76,191,163
48 rem *****
50 rem *** syntax-variablenliste ***
52 ab=5632:rem anfang b-speicher
54 eb=ab:an=0:hn=an
56 mb=34:rem maximale bildernummer
58 gb=34:rem ab seite gb grafik
60 poke650,128:rem repeatfunktion
62 ifpeek(2055)<>58thenhn=peek(2055):goto118
64 printchr$(14)c1$c4$c4$"  Menue"
66 printc4$"  Bilder abspeichern  1"
68 printc4$"  Bilder ansehen      2"
70 printc4$"  Bilder veraendern   3"
72 printc4$"  Bilder drucken      p"
74 printc4$"  Programm beenden    e"
76 print:print
78 printho$"mit 'w' wird Bildnummer
   erhoeht"
80 print"mit 'm' zurueck ins Menue"
82 geta$:ifa$=""goto82
84 if a$="1" then 98
86 if a$="2" then an=0:goto 118
88 if a$="3" then input"Startnummer"
;a:an=a-1:goto136
90 if a$="e" then 158
92 if a$="p" then 186
94 goto 64
96 rem ***** abspeichern *
98 printcl$:input"Grafik ab Seite:";
gb:poke2056,gb:an=0
100 printcl$an+1:open1,0:input#1,q$:
close1
102 ifan>=peek(2056)-2thenprintchr$(
142)
104 hn=an:gosub 164:if hn=mb goto 64

```



```

106 printcl$"w weiter"an+2" m menue
"
108 get a$:if a$="" goto 108
110 if a$="w" then an=an+1:goto 100
112 if a$="m" goto 64
114 goto 108
116 rem ***** zeigen *
118 gosub220:if an>hn then an=0
120 if an>=peek(2056)-1 then printchr$(142)
122 if an<peek(2056)-1 then printchr$(14)
124 gosub176
126 get a$:if a$="" goto 126
128 if a$="w" then an=an+1:goto 118
130 if a$="m" goto 64
132 goto 126
134 rem ***** veraendern **
136 gosub220:if an>hn then hn=an
138 if an>=peek(2056)-1 then printchr$(142)
140 if an<peek(2056)-1 then printchr$(14)
142 gosub176
144 open1,0:input#1,q$:close1
146 gosub164:gosub220:print"'w' weiter"an+2"'m' Menue"
148 get a$:if a$="" goto 148
150 if a$="w" then an=an+1:goto 136
152 if a$="m" goto 64
154 goto 148
156 rem ***** programm beenden **
158 poke 2055,hn:eb=ab+(hn+1)*1024+3
:ha=int(eb/256):la=eb-256*ha
160 printcl$c4$c4$c4$:print"poke 45,"la":poke 46,"ha;up$:end
162 rem bildspeicher verschieben **
164 poke861,0:poke862,4
166 poke863,232:poke864,7
168 poke865,232:poke866,25+4*an:sys828
170 return
172 rem *** seite in bildspeicher **
174 rem ***** kopieren **
176 poke861,0:poke862,22+an*4
178 poke863,232:poke864,25+an*4
180 poke865,232:poke866,7 :sys828
182 return
184 rem***** drucken *****
186 gosub220:if an>hn then an=0
188 printchr$(14)
190 gosub176
192 get a$:if a$="" goto 192
194 if a$="w" then an=an+1:goto 186
196 if a$="m" goto 44
198 if a$="p" then gosub 202
200 goto 192
202 open1,4:fori=0to24:forj=0to39

```

```

204 bs=peek(1024+i*40+j)
206 rem**comodore ascii *****
208 ac=bs-64*(bs<32 or bs>95)-32*(bs>63 and bs<96)
210 rem**standard ascii (ohne rem) *
212 rem ac=ac+128*((ac>192)and(ac<220))-32*((ac>64)and(ac<91))
214 print#1,chr$(ac);:nextj:print#1:nexti:close1:return
216 rem** zeichenfarbe setzen beim *
218 rem** alten betriebssystem *****
220 if(peek(56295)and15)=peek(646) then printchr$(19):return
230 a=53281:b=peek(a):pokea,peek(646):printcl$:pokea,b:return

```

f) Trickfilmeffekte: Da die Maschinenroutine zur Speicherverschiebung sehr schnell ist, kann man trickfilmähnliche Effekte erzielen, indem man in einer FOR/NEXT-Schleife die Variable „an“ immer um 1 erhöht und die Bilder mit GOSUB 176 in den Bildspeicher

kopiert. Man kann die möglichen 33 Bilder in ca. 3 Sekunden durchlaufen lassen.

So, wir hoffen, daß diese Beispiele genügen, Ihnen das Abtippen des Programms schmackhaft zu machen und vor allem zu eigener Programmier-tätigkeit anregen.



Commodore 900

DATEN - ZAHLEN - FAKTEN

COMMODORE 128 PERSONAL COMPUTER IM 128 MODE

**** COMMODORE BASIC V7.0 122.365 BYTES FREE ****

(C) 1985 COMMODORE ELECTRONICS LTD

(C) 1977 MICROSOFT CORP.

ALL RIGHTS RESERVED

READY.

Bildschirmmeldung:

Commodore C-128

Micro-prozessor	Commodore MOS 8502 kompatibel mit MOS 6502
Taktfrequenz	1 oder 2 MHz
Speicher-RAM davon für Basic mit Assembler bis erweiterbar bis	128K-Byte RAM K-Byte RAM K-Byte RAM 512K-Byte RAM
Interpreter Betriebssystem	Basic V-7.0 48K-Byte ROM
Programmiersprache	Commodore-Basic 140 Kommandos Statements und Funktionen
Sprachen-Erweiterungen	k.A.
Toolkits:	k.A.
Zeichenauflösung	40 Spalten x 25 Zeilen umschaltbar auf 80 Spalten x 25 Zeilen
Farben	16 Grundfarben
Grafik	Groß- und Klein-Buchstaben und Grafiksymbole
Hochauflösende Grafik	320x200 = 64000 Punkte oder 640x200 = 128000 Punkte
Tastatur	QWERTY Schreibma- schinentastatur mit 64 Standard-ASCII- Zeichen und 62 Grafik- Zeichen, 4 Funktions- tasten mit 8er Be- legung, 4 Tasten für die Cursorsteuerung, 8 Tasten für Hilfsfunk- tionen, 14 Tasten (Block) für Zahlenein- gabe
Ton Musiksynthesizer	k.A.
Massenspeicher	Datasette-Recorder VC-1530 oder andere Kassettenrecorder mit Interface. Floppy 1541 mit 170K-Byte auf 5 1/2 Disketten Floppy 1571 mit 350K-Byte auf 5 1/4 Disketten

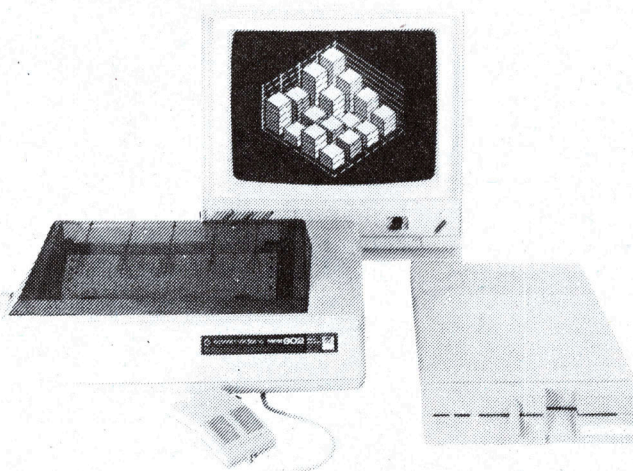
Schnittstellen

RS232C (V24) mit
TTL-Pegeln/Userport
8bit parallel freipro-
grammierbar / IEEE
Bus seriell Kassetten-
port für Datasette / Ex-
pansionsport für Er-
weiterungs- und Pro-
grammodule / Audio-,
Videoport für Monitor
u. TV-Anschlußport /
Zwei Anschlüsse für
Lichtgriffel, Joystick,
Paddles oder Maus

4XXX und 8XXX.
Centronics Drucker
mit Interface oder über
Userport mit Treiber-
Software, Disketten-
laufwerke VC-4040
oder VC-2031

Eingebaute Software

Optional nur über
Eprom-Steckmodule
z. B. Maschinenspra-
chemonitor, Grafik
und Editier-
Erweiterungen



Peripherie

Direkt anschließbar
Matrixdrucker VC-
1525, MPS 801/802
VC-1526 od. Plotter
VC1520, Disketten-
laufwerke VC-1541.
Über ein IEEE 488-
Interface, gesamte
höherwertige Peri. der
Commodore Serie

**Betriebsdaten
Netzspannung** 220 Volt

Kommentar

Preis ca. 800 DM

Hersteller Commodore USA

DER CURSOR DES C64 MUSS NICHT IMMER BLINKEN!

Es gibt Programmsituationen, bei denen man sich über den gewöhnlichen Cursor ärgert. Sei es nun, weil einen das Blinken nervös macht, oder weil man gerne eine andere Farbe hätte, oder und vor allem, weil er in bestimmten Fällen gar nicht zu sehen ist.

Diese letzte Situation tritt ein, wenn man die unzureichende INPUT-Funktion (Kein Komma oder Doppelpunkt, nur 80 Zeichen ...) mithilfe der GET-Funktion umgehen will. Zum Beispiel kann man dies so machen: 10 POKE204,0:GETA\$: IF A\$="" THEN 10 20 ?A\$:B\$=B\$+A\$:GOTO 10

Nach dem GET-Befehl ist der Cursor nämlich normalerweise unsichtbar. Man kann durch sorgfältige Manipulation der Speicherstellen 204 (Cursor an/aus), 205 (Zähler) und 207 (Cursor in Blinkphase) den Cursor zwar sichtbar machen und durch Einbauen einer Zeitschleife auch blinken lassen, das Verfahren ist aber umständlich und hat den Nachteil, daß manchmal ein reverses Leerzeichen stehen bleibt, wenn man die RETURN- oder die CRSR-Tasten betätigt.

Diesen ganzen Ärger hat man nicht, wenn man das Programm SPRITE-CURSOR einsetzt. Das

eigentliche Programm ist sehr kurz und kann überall eingebaut werden. Dabei sollte man aus Geschwindigkeitsgründen die REM-Zeilen weglassen und die Routine ziemlich an den Programmanfang setzen.

Da die Vorgehensweise im Programm etwas kompliziert ist, sollen nun ein paar Erklärungen zum Verständnis beitragen:

Die Zeilen 110–130 stellen ein Beispiel dar, wie die Routine verwendet werden kann. In Ihrem Programm können Sie dann aus den A\$ durch B\$=B\$+A\$ eine Zeichenkette zusammensetzen, die Sie dann weiterverarbeiten können. 50000: Hier wird festgelegt, wo der Sprite steht, der dann als Pseudocursor dient und welche Farbe er bekommt.

500010–50020: Der Sprite bekommt seine Form. Hier sind es einfach zwei senkrechte Striche. Denken Sie sich doch mal einen lustigen aus. Wie wäre es mit einem Männchen, das die Buchstaben auf seine Schippe nimmt? Einen hübschen Strichcursor erhält man, indem die Zeile 50020 ersetzt durch: 50020 POKE 725,255

50100–50120: Jetzt wird es kompliziert. In den Speicherstellen 209/210 steht die Cursorposition im Bildspeicher (a=

```

10 rem == spritecursor =====
12 rem (p)   cbm revue team      =
14 rem =====
16 rem (c)   r. schmid-fabian    =
18 rem c-64      =
20 rem =====
100 printchr$(147);:gosub50000:poke53
280,6:printchr$(14):printchr$(144)
107 rem*****
108 rem***  beispiel-get-schleife ***
109 rem*****
110 gosub 50100
120 gosub50200:gosub50400:geta$:ifa$=""
    then 120
130 printa$;:gosub50300:goto 110
49997 rem*****
49998 rem**  spritecursor in 11*64 **
49999 rem*****
50000 f=0:v=53248:pokev+21,1:pokev+39
    ,f:poke2040,11:rem f=farbe
50010 a=11*64:fori=atoa+62:pokei,0:next
50020 fori=atoa+18step3:pokei,129:next:
    return
50097 rem*****
50098 rem  spritecursor positionieren
50099 rem*****
50100 a=(peek(209)+256*peek(210)-1024)/
    40:b=peek(211)
50110 a=a+int(b/40):b=b+(b>=40)*40
50120 x=24+b*8:y=50+a*8:pokev,xand255
    :pokev+16,x/256:pokev+1,y
50197 rem*****
50198 rem**  spritecursor blinken ***
50199 rem*****
50200 bl=bl+1:if bl<15 then return
50210 if peek(v+21)=1 then pokev+21,0
    :bl=0:return
50220 if peek(v+21)=0 then pokev+21,1
    :bl=0:return
50297 rem*****
50298 rem**  spritecursor einschalten
50299 rem*****
50300 pokev+21,1:bl=0:return
50397 rem*****
50398 rem**  spritecursor einfaerben
50399 rem*****
50400 pokev+39,peek(646):return
50497 rem*****
50498 rem**  spritecursor blinken 2 **
50499 rem*****
50500 bl=bl+1:bl=bl+bl*(bl=15):poke(v
    +21),1+(bl>8):return

```


Bildschirmzeile), in 211 die Bildschirmspalte (b). Hier gilt es noch zu beachten, daß eine „logische“ Zeile mehr als 40 Spalten haben kann. Dies berücksichtigt Zeile 50110. Der Ausdruck (bÜ=40) ist 0 solange bÄ40 und wird zu -1 wenn bÜ=0 ist.

Dies muß man nun nur noch in die x- bzw. y-Position des Spritecursors umrechnen. 50200—50220: Dieses Unterprogramm läßt den Cursor blinken. Man kann die Blinkfrequenz einstellen (15 ändern). Statt 50200 kann man auch 50500 verwenden. Diese Routi-

ne ist kürzer aber langsamer.

50400: Der Spritecursor nimmt die Zeichenfarbe an.

Wenn Sie nun das Programm verstanden haben, werden Ihnen sicher noch viele neue Ideen kommen, was man damit noch alles anfangen kann.

Eine reizvolle und interessante Anwendung ist der Einsatz in Spielen, in denen der Sprite über die Cursortasten gesteuert wird. Hierbei kann man den Sprite ja statt in Sprüngen von 8 in Einerschritten bewegen. (rsf)

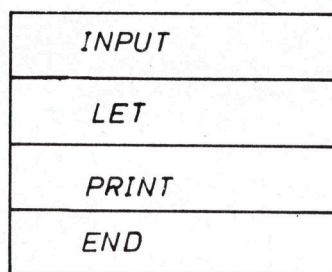
STRUKTOGRAMME

Da der Mensch Grafiken und Bilder schneller und besser überblickt als Text- und Zahlenkolonnen, bieten sich hier grafische Hilfsmittel besonders an. In der letzten Zeit wird der bisher übliche Programmablaufplan immer mehr von Struktogrammen (auch Nassi-Schneidermann-Diagramm oder kurz NS-Diagramm genannt) abgelöst.

Das Struktogramm bezeichnet man auch als grafische Darstellung eines Algorithmus in Blockform. Der Hauptgrund war wohl die Entwicklung höherer Programmiersprachen, die eine Strukturierung zulassen, wie Pascal, Fortran und ähnliche. Auch verschiedene Basic-Versionen, wie das 3.5 oder 7.0 Basic der neuen Commodore-Homecomputer (C16/Plus 4 oder der C128) besitzen Befehle, die eine Strukturierung zulassen. Damit ist vor allem ein Verzicht auf die zeitaufwendigen GOTO Sprünge möglich. Das Struktogramm stützt sich vor al-

lem auf die Befehle IF... THEN... ELSE... DO WHILE/UNTIL... DO LOOP usw. Struktogramme zwingen den Program-

Sequenz



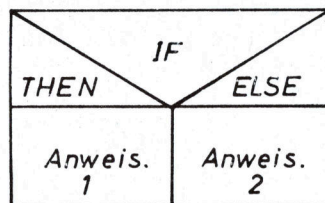
Sequenz:

Hier werden mehrere Anweisungen zusammengefaßt und hintereinander ausgeführt.

mierter zu übersichtlichen Programmen. Auch kann er nicht durch GOTO-Sprünge die Programme unübersichtlich machen. Die Struktogramme bestehen aus nur wenigen Grundstrukturen, die zu Blöcken zusammengesetzt und beliebig ineinander verschachtelt werden. Die Struktogramme bestehen aus Folge-, Auswahl-, Wiederholungs-, sowie Unterprogrammstrukturen.

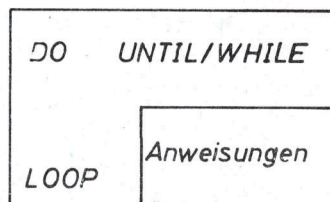
Auch werden Struktogramme ohne die vom Programmablaufplan her bekannte Ablaufflinie dargestellt. Beim Struktogramm wird sie durch die

IF.....THENELSE



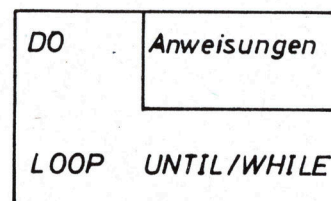
Bedingte Wiederholung (mit Abbruchbedingung am ANFANG der Schleife)

Bedingte Wiederholung

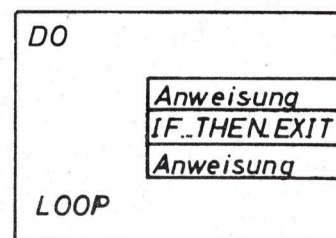


waagrechte Begrenzungslinie ersetzt. Struktogramme lassen sich immer ohne Schwierigkeiten in Programmablaufpläne um-

Bedingte Wiederholung



Bedingte Wiederholung (mit Abbruchbedingung am ENDE der Schleife)



Bedingte Wiederholung (Mit Abbruchbedingung im Wiederholungsteil der Schleife)

wandeln. Umgekehrt ist das nur mit erheblichem Aufwand möglich. Der Vorteil des Struktogrammes liegt darin, daß man die Programme leichter übersehen kann und es schwerer wird sogenannte Spaghetticode zu programmieren.

DIE SCHLEIFE ALS STOLPERSTEIN

Wofür brauche ich Schleifen beim Programmieren? Eine Frage, die immer wieder auftaucht. Obwohl viele nicht wissen, was eine Schleife beim Programmieren ist, sind fast alle schon darüber gestolpert. Fast kein Programm kommt ohne Schleifen aus. Sei es um zu zählen, den Computer für eine bestimmte Zeit anzuhalten, um eventuell auf Eingaben zu warten oder zum Einlesen von Daten aus Datazeilen und viele Anwendungen mehr.

Vielleicht haben Sie in einem Listing schon mal den Begriff FOR...TO...NEXT gelesen. Das ist schon eine Schleife. Heute wollen wir uns aber mit der Möglichkeit der Schleifenprogrammierung, die uns das Basic 3.5 bietet, beschäftigen. Obwohl dieses Basic nur in den

Gegenüber dem Basic der Computer VC-20 und C64 verfügt das Basic 3.5 über die Möglichkeit des strukturierten Programmierens. Durch die neuen Schleifenbefehle des Basic 3.5 erübrigen sich die für Unübersichtlichkeit sorgenden GOTO-Sprünge. Mit den Befeh-

und LOOP. Auf dem ersten Blick erscheint die Schleife auch nicht kürzer als die Kombination 100 GOTO 100. Da aber die Sprungbefehle immer von 9 an hoch gezählt werden, ist die Schleife mit 100 DO:LOOP viel schneller. Mit einer Endlosschleife kann man im allgemeinen nicht viel anfangen, denn Schleifen müssen irgendwann zum Ende kommen, also endlich sein. Eine endliche Schleife mit DO LOOP sieht etwa so aus.

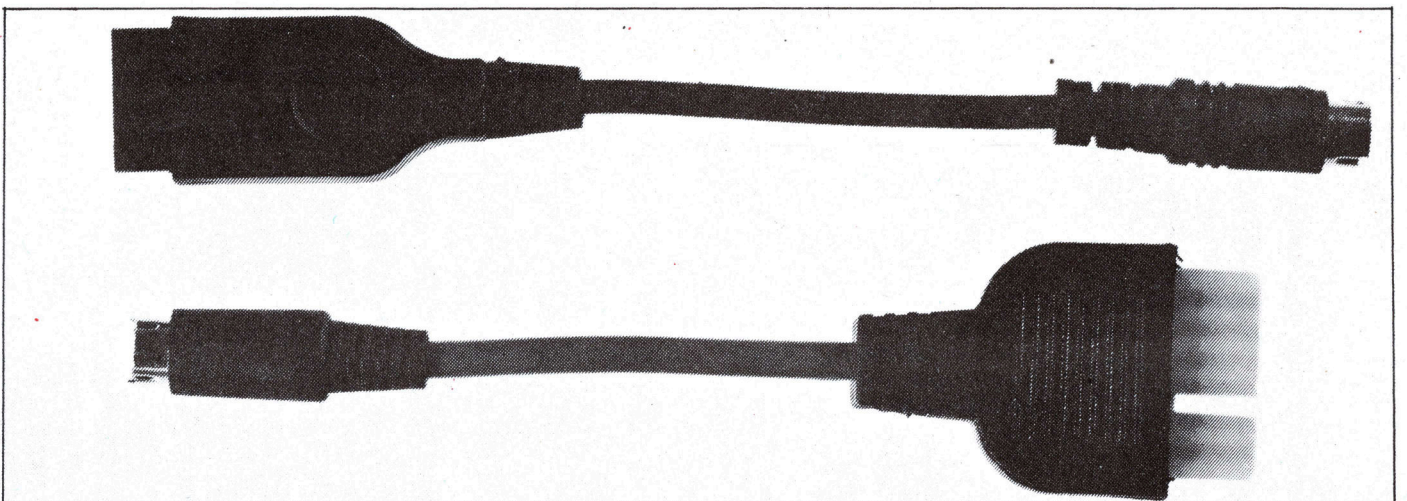
```
100 X=0
110 DO
120 X=X+1
130 LOOP UNTIL X=10
140 PrintX
```

In dieser Schleifenversion erfolgt der Austritt aus der Schleife am Ende derselben. Denn dort steht die Bedingungsabfrage „UNTIL X = 10“. Diese Schleife läuft also solange, bis X den Wert 10 hat. Man kann auch

Regeln) die Anweisung „wahr“ wird. WHILE stellt praktisch das Gegenteil von UNTIL dar, die Schleife wird solange durchlaufen wie die Anweisung „wahr“ ist.

Um einen Austritt am Schleifenanfang zu erreichen, setzt man den Befehl UNTIL einfach hinter den Befehl DO. Diese Schleifenart nennt man abweisende Schleife, denn wenn der Befehl WHILE am Anfang der Schleife steht, wird, falls die Bedingung hinter WHILE erfüllt ist, der Schleifendurchlauf nicht mehr durchgeführt, während bei unserer Beispielschleife die Bedingung erst am Ende erfüllt sein muß.

Befehl EXIT ermöglicht einen Ausstieg aus einer Schleife zu einem bestimmten Zeitpunkt. Diese erfolgt im Zusammenhang mit einer IF..THEN Abfrage z.B. IFA\$



Streßgeplagte Anwender des C16/Plus 4 müssen nicht länger verzweifeln. In „guten“ Computerfachgeschäften sind nun Interfaces erhältlich, die es ermöglichen, die handelsüblichen Joysticks mit neunpoligem Stecker sowie die Datasette 1530 an die beiden Commodore Rechner anzuschließen.

Commodore Computern C16/C116 und Plus 4 vorhanden ist, brauchen die Besitzer der Computer VC-20 und C64 nicht aufhören zu lesen, denn in vielen Befehlserweiterungen, z.B. Exbasic Level II, sind diese Befehle auch enthalten.

len DO, LOOP, UNTIL, WHILE und EXIT können jetzt Schleifen schnell und einfach programmiert werden. Um z.B. die oft verwendete, aber eigentlich unerwünschte Endlosschleife zu programmieren, genügen nun nur noch zwei Befehle: DO

die Umkehrung nehmen. Nämlich „WHILE () 10“. Diese Schleife läuft so lange wie X ungleich 10 ist.

Daraus resultiert: Wird in einer Schleife UNTIL verwendet, wird die Schleife solange durchlaufen, bis (nach Booleschen

=50 THEN EXIT. EXIT unterbricht also die Schleife, um irgendwas in einer Variablen zu speichern. Nach EXIT verzweigt das Programm in die der Anweisung LOOP folgende Zeile. Die Struktur einer DO LOOP EXIT Schleife sieht folgender-

maßen aus.

```
100 DO
110 .....
120 IF....THEN EXIT
130 .....
140 LOOP
```

Zum Schluß kommen wir zu der wohl schon allseits bekannten FOR..NEXT Schleife. Diese Schleifenart wird hauptsächlich als Zähl- oder Warteschleife angewendet. Auch zum Einlesen von Daten aus Datazeilen in den Speicher kann sie recht nützlich sein. Da die Überprüfung der Schleife am Ende erfolgt, wird eine Schleife mit

FOR X = 5T05: NEXT nur einmal durchlaufen.

Halt! Eine Schleifenart hätten wir bald noch vergessen. Die Warteschleife mit GET. Das Basic 3.5 stellt uns hier den Befehl GETKEY zur Verfügung. Das Programm wartet an dieser Stelle solange, bis ein Zeichen über die Tastatur eingegeben wird. Erst dann fährt das Programm in seiner Abarbeitung fort. Fazit: Durch die Schleifenbehandlung des Basic 3.5 wird eine Programmierung weitgehend ohne GOTO-Sprünge ermöglicht.

(B. Welte)

Das Aus für Hacker?

Am 27. Februar 1986 verabschiedete der Bundestag das zweite Gesetz zur Bekämpfung der Wirtschaftskriminalität. Ausgelöst wurde diese Initiative durch neue Formen der Kriminalität, die durch Entwicklungen im Wirtschaftsleben, wie etwa dem verstärkten Einsatz der Datenverarbeitung, möglich wurden. Solche Delikte konnten mit dem bisherigen Strafrecht nur unzureichend verfolgt werden. Der Schwerpunkt der beschlossenen Regelungen liegt auf dem Gebiet der Computerkriminalität. Wer also künftig Datenübertragungssysteme anzapft oder abhört, muß, wenn er erwischt wird, mit Strafe rechnen. Denn das ist jetzt verboten. War es früher erlaubt? Mit langjährigen Gefängnisstrafen oder hohen Geldbußen muß derjenige rechnen, der in fremde Datensysteme eindringt und Daten fälscht oder aber fremde Computer sabotiert. Es hat lange gedauert, bis der Gesetzgeber gehandelt hat. Obwohl sich hier die Frage

stellt, wer wen anklagt. Denn in fremde Computer eindringen und sich ein bißchen umsehen, soll nicht unter Strafe gestellt werden. Wie will man da unterscheiden, ob jemand nur schauen oder sabotieren will? Denn was ist ein fremder Computer? Etwa schon die Mailbox des Computer-Clubs XY oder erst der Großrechner der nächsten Universität? Vielleicht sollten sich einige maßgebliche Herren der Computer- bzw. der Postbranche mal Gedanken darüber machen, wie sorglos doch wichtige Daten über teilweise unzureichende und ungesicherte Leitungen kreuz und quer durch unser Land bzw. um die ganze Welt gejagt werden. Auch gab oder gibt es Computerfachleute, die, wenn sie sich von ihren Firmen trennen, sogenannte Viren in die Programme einbauen, die das Programm dann nach gewisser Zeit von selbst zerstören. Dieses wird dann meistens den „Hackern“ angelastet, die vielleicht in den meisten Fällen ganz unschuldig sind.

EINSTELLHILFE

Das Set besteht aus 2 Cassetten mit Schraubendreher und Anleitung im stabilem Aufbewahrungskasten.

Nie mehr Ladeprobleme, so wird dieses Ton-

sich ein Prüfton, mit dem der Tonkopf der Datensette eingestellt werden soll. Die 7seitige bebilderte englische Anleitung zeigt ganz genau, wie und wo mit dem beiliegenden



kopfjustage-Set angepriesen. Es ermöglicht die genaue Einstellung des Tonkopfes Ihrer Datensette. Nach dem Laden und Starten des Programmes erscheint eine Graphic, die aus einer großen, vierstelligen Digitalanzeige und einer Skala mit einem roten und grünem Endbereich besteht, auf der sich ein Zeiger in Abhängigkeit von der Tonkopfeinstellung bewegt. Auf der zweiten Cassette befindet

Schraubendreher die Einstellung vorgenommen werden muß. Wenn der Tonkopf genau eingestellt ist, sollte sich der Zeiger im grünen Bereich befinden und die Anzeige 3000 oder mehr anzeigen. Die sehr ausführliche Bedienungsanleitung führt Schritt für Schritt durch den Einstell- und Prüfvorgang. Es wird sogar darauf hingewiesen, wann der Schraubendreher aus dem Einstelloch gezogen werden muß.

WESTERNRALLYE

Zu Seite 38

Sie verfügen über einen Rennwagen, mit welchem Sie auf einer Straße durch die Wüste steuern müssen. Von Zeit zu Zeit tauchen Hindernisse auf, welche umfahren werden müssen. Gesteuert wird mit den Tasten >Z< und >/<. Im oberen Bildschirmteil wird die bereits gefahrene Zeit in Sekunden angezeigt. Kollidieren Sie mit einem Hindernis, so bekommen Sie die erreichte Fahrzeit extra angezeigt. Mit >SPACE< wird ein neues Spiel gestartet.

KALEIDOSKOP

Zu Seite 43

Das Programm zeichnet in den verschiedenen Farben des VC 20 ein Muster wie das eines Kaleidoskopes auf den Bildschirm. Dieses Muster wird ständig verändert, so daß es unzählig viele Variationen gibt. Um ein gewünschtes Muster länger anzusehen, kann der Bildschirm mit >F1< gestoppt und wieder gestartet werden. Mit >F3< kann der dazugehörige Ton ab- bzw. angeschaltet werden.

DONALD DUCK

Zu Seite 45

Nach Programmstart druckt der VC 20 einen Donald-Duck-Kopf auf dem Commodore Drucker aus.

KASSETTEN-ETIKETT

Zu Seite 39

Das Programm druckt gewünschte Etiketten für Kassetten aus. Sie können die Etiketten numerieren, eine Rückenbeschriftung angeben, den Interpreten und den Titel ausdrucken lassen. Mit >L< wird ein weiterer Ausdruck eingegeben, mit >*< wird das Programm beendet.

MATHEKÜNSTLER

Zu Seite 41

Mit dem Programm können alle Grundrechenarten bearbeitet werden.

Im Hauptmenü wird gewählt zwischen

- Addieren
- Subtrahieren
- Multiplizieren
- Dividieren
- Potenzieren
- Wurzel
- Fakultät (<33)
- Durchschnitt
- Primzahlen
- Geometrie

Mit >K< wird das Programm beendet. Unter dem Menüpunkt "Geometrie" können die Inhalte der wichtigsten geometrischen Flächen berechnet werden, für den Kreis kann auch der Umfang berechnet werden.

KUNDENKARTEI

Zu Seite 37

Mit dem Programm kann eine Kundenkartei auf dem Kassettenrekorder erstellt werden. Im Hauptmenü wird der gewünschte Punkt ausgewählt. Die Kundenkartei kann mit den Menüpunkten Speichern, Laden, neue Angaben, Suchen und Löschen bearbeitet werden. Jeder Adresse kann eine Kundennummer hinzugefügt werden.

17&4 KARTENSPIEL

Zu Seite 27

Das Programm simuliert das bekannte Kartenspiel 17&4 und zeichnet die Karten auf den Bildschirm. Zu Programmbeginn werden noch einmal ausführlich die Regeln erläutert. Sie können bis zu einem Limit von 25.000 DM setzen und gegen den C 64 spielen. Nach Programmende wird Ihr Gesamtkonto mit dem entsprechenden Kommentar angezeigt.

BIORHYTHMUS

Zu Seite 30

Sie müssen Ihr Geburtsdatum in der Reihenfolge Tag-Monat-Jahr eingeben. Nach allen Daten muß >ENTER< gedrückt werden. Danach werden Sie nach dem gewünschten Monat gefragt, für den die Berechnung erstellt werden soll. Der C 64 zeigt dann auf dem Bildschirm die Biorhythmus-Kurve für den ganzen Monat an.

VOKABELPAUKER

Zu Seite 35

Mit diesem Programm ist es möglich, einzelne Lektionen einzugeben, abzuspeichern und sich vom C 64 abfragen zu lassen. Im Hauptmenü kann gewählt werden zwischen

- Vokabeleingabe
 - Vokabelabfrage
 - Abspeichern der Vokabel
 - Laden der Daten vom Band
 - Liste der Vokabeln auf Drucker oder Bildschirm
- Durch die Auflistung auf dem Drucker kann die gerade bearbeitete Lektion schwarz auf weiß betrachtet und gelernt werden.

ADRESSENREGISTER

Zu Seite 20

Bei Programmbeginn wird eine ausführliche Beschreibung gegeben. Zu beachten ist, daß die einzelnen Angaben wie z.B. Name und Straße durch einen Schrägstrich getrennt werden. Mit diesem Programm können außer den üblichen Funktionen auch Adressen sortiert werden!

WEITSPRUNG

Zu Seite 24

An dem Spiel können bis zu fünf Spieler teilnehmen. Nach jedem Sprung wird das Ergebnis mit einem entsprechenden Kommentar angezeigt. Bei Spielende kann gewählt werden, ob mit den selben Spielern, mit anderen Spielern oder überhaupt nicht weitergespielt werden soll.


```

10 rem Adressen-Register=====64
20 rem (P) 09/85 CBM REVUE TEAM =
30 rem =====
40 rem (C) 05/85 by =
50 rem Axel Kohlmann =
60 rem =
70 rem VC 20 + mind. 3K + 1531 =
80 rem C-64 + 1530/1541 =
90 rem =====
100 goto 370
110 gf$=chr$(034):y3$=chr$(162)
120 y5$=chr$(166):kl$=chr$(091)
130 u4$=chr$(191):ss$=chr$(047)
140 kr$=chr$(093)
150 sl$=chr$(014):c4$=chr$(017)
160 rn$=chr$(018):he$=chr$(019)
170 ne$=chr$(028):c3$=chr$(029)
180 gr$=chr$(030):bl$=chr$(031)
190 sl$=chr$(032)
200 fa$=chr$(130):fb$=chr$(131)
210 bk$=chr$(144):c2$=chr$(145)
220 rf$=chr$(146):cl$=chr$(147)
230 pu$=chr$(156):cl$=chr$(157)
240 cy$=chr$(159)
250 s2$=chr$(160):s3$=chr$(224)
260 z1$=chr$(096):z2$=chr$(098)
270 z3$=chr$(123):z4$=chr$(171)
280 z5$=chr$(173):z6$=chr$(174)
290 z7$=chr$(176):z8$=chr$(177)
300 z9$=chr$(178):za$=chr$(179)
310 zb$=chr$(189):ml$=chr$(125)
320 d2$=c4$+c4$:d3$=d2$+c4$
330 d4$=d3$+d2$:d5$=d4$+c4$
340 d6$=d5$+c4$:d7$=d6$+c4$
350 d8$=d7$+c4$:d9$=d8$+c4$
360 return
370 poke 53280,6:poke 53281,1
380 gosub 110:goto 650:rem -> menu
e =
290 rem ===== dateneingabe =
400 z=1970
410 print cl$d7$re$z7$;:for i=1 to
19:print z1$;:next :print z6$
420 print ml$ " geben sie die "m
1$
430 print ml$ " einzelnen adress- "m
1$
440 print ml$ " angaben wie name, "m
1$
450 print ml$ "strasse, wohnort..."m
1$
460 print ml$ "jeweils durch einen"m
1$
470 print ml$ " schraegstrich "gf$
s$gf$ml$
480 print ml$ " voneinander "m
1$
490 print ml$ " getrennt ein. "m
1$
1$
500 print ml$ " die erste angabe "m
1$
510 print ml$ " ist entscheidend "m
1$
520 print ml$ " fuer suchen&sort. "m
1$
530 print ml$ " "m
1$
540 print ml$ " zur funktionswahl "m
1$
550 print ml$ " "rn$"return"rf$"-t
aste "ml$
560 print z5$;:for i=1 to 19:print
z1$;:next :print zb$
570 print he$bl$rn$ " geben sie date
n ein "rf$:l3$=c1$+c1$+c1$
580 if fre(0)<900 then 910
590 print u4$l3$;:input f$
600 if f$=u4$ then 650
610 print cl$d3$;z;"data ";f$
620 print "400 z=";z+10:print "goto
100"
630 poke 631,13:poke 632,13:poke 63
3,13:poke 198,3:print he$:end
640 rem ===== menue =
650 print cl$"...re$rn$".....
....."rf$
660 print "...rn$"adressen-register
."rf$
670 print "...rn$".....
."rf$
680 print "...rn$(c).axel.kohlmann
."rf$
690 print "...rn$".....
."rf$
700 print he$d6$bk$"....funktionswa
hl":print
710 print re$rn$;:for i=1 to 20:pr
nt y3$;:next :print rf$:print
720 print bk$ " druecke":print
730 print " "rn$"1"rf$ " dateneinga
be"
740 print " "rn$"2"rf$ " datenausga
be"
750 print " "rn$"3"rf$ " sortierung
.."
760 print " "rn$"4"rf$ " adressen s
uchen"
770 print " "rn$"5"rf$ " adressaend
erung"
780 print " "rn$"6"rf$ " programmen
de"
790 get a$:if a$="" then 790
800 on val(a$)goto 400,1470,980,130
0,1780,1660
810 goto 790
820 rem == s feld von b$ trennen =

```



```

830 dim n$(n):for i=0 to n
840 ll=0
850 ll=ll+1
860 if ll=len(b$(i))+1 then 890
870 if mid$(b$(i),ll,1)="/" then 890
880 n$(i)=n$(i)+mid$(b$(i),ll,1):go
to 850
890 next
900 return
910 print c4$b1$ " kein Speicherpla
tz      mehr verfuegbar!"
920 for i=1 to 5000:next :goto 650
930 print pu$rn$c2$c2$ " drucker ein
schalten!! und zur....."rf$;
940 print rn$ " bestaetigung return-
taste druecken....."rf$
950 print h$d4$ "      "rn$"drucker"
rf$
960 poke 198,0:wait 198,1:print d3$
".....ok!":return
970 h4$=c2$+c2$+c2$+c2$
980 print h4$ "      "pu$rn$"sortierun
g "rf$:print d6$ " "rn$"bitte etwa
s geduld "
990 n=1:gosub 1000:goto 1020
1000 read a$:if a$="***" then res
tore :n=n-1:return
1010 n=n+1:goto 1000
1020 n=n-1:dim b$(n):for x=0 to n:r
ead b$(x):next :restore :gosub 830
1030 print c1$:gosub 1040:goto 1050
1040 print h$re$rn$"alphabetische
sortierung der namen
"rf$:return
1050 for i=0 to n-1:k=i
1060 for j=i+1 to n
1070 if n$(j)<n$(k) then k=j
1080 next j
1090 x$=n$(k):n$(k)=n$(i):n$(i)=x$
1100 next i
1110 f$=c4$:l=1
1120 for i=1 to n:gosub 1210:print
ff$;n$(i):l=l+1:if l=17 then gosub
1450
1130 gosub 1040:ff$=ff$+f$:
1140 for k=1 to 300:next k
1150 next
1160 :print ff$re$rn$"alle namen so
rtiert! "rf$;
1170 print pu$rn$"ok? return --> en
de"rf$;
1180 poke 198,0:wait 198,1
1190 if peek(631)<>13 then 1180
1200 poke 198,0:clr :gosub 110:goto
650
1210 as=asc(n$(i))-64
1220 on asgoto 1240,1250,1260,1270,
1280,1290
1230 as=as-6:goto 1220
1240 print cy$;:return
1250 print re$;:return
1260 print bk$;:return
1270 print pu$;:return
1280 print gr$;:return
1290 print bl$;:return
1300 print c2$c2$c2$ "      "pu$rn$"ad
ressen suchen"rf$
1310 print d5$ "      "rn$"bitte etwas g
eduld"rf$
1320 gosub 1000:dim b$(n):for k=0 t
o n:read b$(k):next :gosub 830
1330 print c1$re$rn$" suchen einen
adresse (return-->ende) "rf$
1340 print d3$re$"name: ";
1350 print y5$c1$c1$c1$;:input na$:
if na$=y5$ then 1430
1360 for k=0 to n:if na$=n$(k) then
k2=k:goto 1400
1370 next
1380 print d3$re$" die adresse ist
nicht      gespeichert "
1390 for lt=1 to 2000:next :goto 13
30
1400 j=k2:k2=1:gosub 1610:print re$
rn$"ok? --> taste druecken"rf$
1410 poke 198,0:wait 198,1:poke 198
,0
1420 goto 1330
1430 clr :gosub 110:goto 650
1440 goto 1940
1450 l=1:print c4$re$rn$"weiter? ta
ste druecken"rf$;:poke 198,0:wait 1
98,1
1460 print c1$:gosub 1040:ff$="":re
turn
1470 print c1$re$rn$"ausgabe aller
adressen"rf$
1480 print d2$bk$ " ausgabegeraet:"
1490 print c4$ "      "rn$"1"rf$ " druck
er"
1500 print "      "rn$"2"rf$ " bildschi
rm"
1510 print d5$re$rn$"      bitte waeh
len! "rf$
1520 get d$:if d$="" then 1520
1530 on val(d$)goto 1550,1560
1540 goto 1520
1550 ag=4:gosub 930:open 4,4:gosub
1590:close 4:restore :clr :gosub 11
0:goto 650
1560 print h$pu$d5$ "      "rn$"bild
schirm"rf$d5$
1570 print "....."
c2$c2$c2$c2$
1580 ag=3:gosub 1590:restore :clr :
gosub 110:goto 650

```



```

1590 gosub 1000:dim b$(n+1):for j=1
  to n+1
1600 :read b$(j)
1610 for c=0 to 7:c$(c)=" ":next c
1620 i=0:c=1
1630 i=i+1
1640 if i=len(b$(j))+1 then 1670
1650 if mid$(b$(j),i,1)="/" then c=
c+1:goto 1630
1660 c$(c)=c$(c)+mid$(b$(j),i,1):go
to 1630
1670 if k2>0 then gosub 1750:return
1680 gosub 1720:next
1690 print re$rn$ " ok? ende --> re
turn "rf$
1700 poke 198,0:wait 198,1:if peek(
631)=13 then return
1710 goto 1700
1720 if ag=4 then goto 1770
1730 bi=bi+1:if bi=2 then bi=0:prin
t re$rn$"weiter? taste druecken!"rf
$;
1740 poke198,0:wait 198,1:print cl$
1750 for c=0 to 7:print bk$cl$c$(c)
:next c:return
1760 rem ===== drucker output =
1770 for c=0 to 7:print# 4,"
      ";c$(c):next :return
1780 print cl$re$rn$" adressenaende
rung "rf$;
1790 print rn$(angaben hinter den.
.datas aendern und dann"rf$;
1800 print rn$" programm erneut mi
t "rf$ "run"rn$" starten) "rf
$;
1810 print rn$"ctrl-taste zum verl
angsamem von list.....
"rf$
1820 print d3$"list1950- : "kl$"ret
urn"kr$
1830 for k=1 to 4000:next :poke 631
,13:poke 198,1
1840 print he$d6$:end
1850 goto 650
1860 v=0
1870 print cl$d3$"1810 v=";v+1:prin
t "v=";v;":goto1830"
1880 poke 631,13:poke 632,13:poke 1
98,2:print he$:end
1890 gosub 110:print cl$re$rn$"kass
ette einlegen und"rf$
1900 print rn$"die taste rec + play
"rf$
1910 print rn$"druecken !.....
"rf$
1920 print he$d6$"save"gf$"adressen
"v;gf$
1930 poke 631,13:poke 632,13:poke 1

```

```

98,2:print he$d3$bk$:end
1940 gosub 110:goto 650
1950 data cbm-revue/redaktion/post
fach 1107/8044 lohnhof
1960 data commodore/benutzerclub/p
ostfach 426/6078 neu-isenburg
-----

```

Ab hier schreibt der Computer, automatisch DATA-Zeilen mit Adressen ein, solange der Speicherplatz ausreicht.

```

9900 data "****"
9910 rem adressen-register =====64
9920 rem 38911 bytes memory =====
9930 rem 06068 bytes program =====
9940 rem 00343 bytes variables =====
9950 rem 00000 bytes arrays =====
9960 rem 00000 bytes strings =====
9970 rem 32410 bytes free =====
9980 rem =====

```



Wir verschenken! CBM-64/C-128 VC-20

Die PEEK/POKE-MAGICS!

Jetzt gratis zu jeder Katalog-Bestellung.

Über 300 PEEKS und POKES-Tips und Tricks als kompaktes Nachschlagewerk mit den letzten Geheimnissen des Betriebssystems. Da sollten Sie zugreifen!

NEU! Katalog "Winter 85/86" (64 Seiten). Schnell anfordern od. abholen für 2,50 DM (Briefmarken).

Stichwort "Katalog + PEEK/POKE-MAGICS!"

Wir haben alles für COMMODORE 64, C-128, VC-20! Geräte, Hardware, Software, Zubehör und ... und ... über 500 Artikel zu Tiefstpreisen!

mükra
DATEN-TECHNIK

Schöneberger Str. 5
(Am Berlinicke Platz)
1000 Berlin 42/RE
☎ 030-752 91 50/60

Öffnungszeiten:
Mo-Fr: 10-18 Uhr
Sa.: 10-13 Uhr

Händleranfragen erwünscht. · 24 Std. Schnellversand.


```

10 rem Weitsprung =====64
20 rem (P) 09/85 CBM REVUE TEAM =
30 rem =====
40 rem (C) 09/85 Ali-Osman Koese =
50 rem =
60 rem =
70 rem =
80 rem C-64 + 1530/1541 =
90 rem =====
100 wh$=chr$(5):dc$=chr$(3)
110 ec$=chr$(9):rt$=chr$(13)
120 sl$=chr$(14):c4$=chr$(17)
130 rn$=chr$(18):he$=chr$(19)
140 de$=chr$(20):ee$=chr$(27)
150 re$=chr$(28):c3$=chr$(29)
160 gr$=chr$(30):bl$=chr$(31)
170 sl$=chr$(32):oe$=chr$(129)
180 fa$=chr$(130):fb$=chr$(131)
190 fl$=chr$(133):f3$=chr$(134)
200 f5$=chr$(135):f7$=chr$(136)
210 f2$=chr$(137):f4$=chr$(138)
220 f6$=chr$(139):f8$=chr$(140)
230 sr$=chr$(141):su$=chr$(142)
240 bk$=chr$(144):c2$=chr$(145)
250 rf$=chr$(146):cl$=chr$(147)
260 in$=chr$(148):br$=chr$(149)
270 lr$=chr$(150):gl$=chr$(151)
280 g2$=chr$(152):lg$=chr$(153)
290 lb$=chr$(154):g3$=chr$(155)
300 pu$=chr$(156):cl$=chr$(157)
310 ye$=chr$(158):cy$=chr$(159)
320 s2$=chr$(160):s3$=chr$(224)
330 z1$=chr$(096):z2$=chr$(098)
340 z3$=chr$(123):z4$=chr$(171)
350 z5$=chr$(173):z6$=chr$(174)
360 z7$=chr$(176):z8$=chr$(177)
370 z9$=chr$(178):za$=chr$(179)
380 a2$=s1$+s1$:a3$=a2$+s1$
390 a4$=a3$+s1$:a5$=a4$+s1$
400 be$=chr$(045):lu$=chr$(175)
410 e9$=chr$(186):l3$=chr$(180)
420 l8$=chr$(182):lo$=chr$(183)
430 s9$=chr$(114):kl$=chr$(110)
440 l4$=chr$(165):l5$=chr$(163)
450 l1$=chr$(116):l2$=chr$(167)
460 l6$=chr$(164):y3$=chr$(162)
470 d2$=c4$+c4$:d3$=d2$+c4$
480 d4$=d3$+c4$:d5$=d4$+c4$
490 h2$=c2$+c2$:h3$=h2$+c2$
500 h4$=h3$+c2$:h5$=h4$+c2$
510 y8$=chr$(184):y9$=chr$(185)
520 y1$=chr$(121):m1$=chr$(125)
530 o2$=y9$+y9$:o3$=o2$+y9$
540 o4$=o3$+y9$:o5$=o4$+y9$
550 y2$=chr$(161)
560 sx$(2)=15$+15$:sx$(3)=sx$(2)+15$
570 sx$(4)=sx$(3)+15$:sx$(5)=sx$(4)
+15$
580 sz$(2)=lo$+lo$:sz$(3)=sz$(2)+lo$
590 sz$(5)=sz$(2)+sz$(3)
600 lz$=a5$+a5$+a5$+a5$+a5$+a5$
+15$
610 goto 2020
620 f=0:j=1
630 poke 53281,6:poke 53280,0:print
cl$wh$
640 input "wieviel spieler <1-5>";
sp
650 if sp<1 or sp>5 then 640
660 for a=1 to sp:print cl$"name de
s ";a;:input "spielers ";sp$(a):nex
t a:
670 v=53248:poke v+21,0:poke 2040,1
3:poke 2041,14:poke 2042,15:poke 20
43,204
680 for x=0 to 62:read q:poke 832+x
,q:next x
690 for x=0 to 62:read q:poke 896+x
,q:next x
700 for x=0 to 62:read q:poke 957+x
,q:next x
710 for x=0 to 62:read q:poke 204*6
4+x,q:next x
720 poke v+23,255:poke v+29,255:pok
e v+38,1:poke v+39,1:poke v+40,1
730 poke v+41,1:poke v+42,1:poke v+
43,1
740 goto 750
750 print cl$gr$d5$d5$d4$:ge=20
760 for x=1 to 8:for i=1 to 40:prin
t rn$sl$rf$;:next :next
770 print h5$h2$
780 for x=1 to 40:print rn$br$y8$rf
$;:next x:for x=1 to 40
790 print br$rn$sl$rf$;:next :print
c2$
800 for x=1 to 40:print rn$br$y9$rf
$cy$;:next x
810 print he$sl$"spieler"a5$a5$a2$rf
$"meter"
820 for i=1 to 38:print bs$;:next :
print bs$
830 for x=1 to sp:print sl$sp$(x);:
print tab(19);"";me(x):next
840 for i=1 to 38:print bs$;:next :
print bs$
850 f=f+1:if f>sp then goto 1410:j=
j+1
860 print sl$"springer = "sp$(f)
870 for a=10 to 255 step 6
880 j=peek(56321):poke v+21,1
890 poke v,a:poke v+1,168:for t=1 t
o 30:next :if (j and 4)=0 then gosub
b 1340

```



```

900 ge=ge-1
910 poke v+21,2:poke v+2,a:poke v+3
,168
920 for t=1 to 30:next :j=peek(5632
1):if (j and 8)=0 then gosub 1340
930 gosub 1350:ge=ge-1:gosub 1350:n
ext a
940 for a=0 to 63 step 6
950 poke v+21,1:poke v+16,1:j=peek(
56321)
960 poke v,a:poke v+1,168:for t=1 t
o 30:next t:if (j and 4)=0 then gos
ub 1340:ge=ge-1
970 gosub 1350
980 poke v+21,2:poke v+16,2:poke v+
2,a:poke v+3,168:for t=1 to 30:next
t:
990 j=peek(56321):if (j and 8)=0 th
en gosub 1340:ge=ge-1:gosub 1340
1000 next a
1010 poke v+21,0:print cl$h5$h2$
1020 print gr$d5$d5$c4$
1030 for x=1 to 9:print rn$a5$a5$a5
$a5$a5$a5$a5$a5$a5$a4$rf$:next :print
1040 print tab(24)h5$h5$c4$rn$:for
i=1 to 15:print lu$:next :print r
f$
1050 print tab(23)rn$gr$yl$ye$a5$a5
$a5$gr$
1060 print rn$:for i=1 to 23:print
lu$:next:print e9$ye$a5$a5$a5$bk$
1070 printbr$rn$a5$a5$a5$a4$wh$13$1
3$gr$br$sl$18$gr$ye$a5$a5$a5$sl$gr$
1080 printbr$rn$a5$a5$a5$a4$wh$13$1
3$gr$br$sl$18$gr$ye$a5$a5$a5$sl$gr$
1090 printbr$rn$a5$a5$a5$a4$wh$13$1
3$gr$br$sl$18$gr$ye$a5$a5$a5$gr$
1100 print rn$:for i=1 to 23:print
lo$:next :printe2$ye$a5$a5$a5$gr$
1110 print tab(23)rn$18$ye$a5$a5$a5
$gr$
1120 print tab(24)rn$:for i=1 to 1
5:print lo$:next :print wh$
1130 poke v+16,0:for a=10 to 180 st
ep 6
1140 poke v+21,1:j=peek(56321)
1150 poke v,a:poke v+1,177:for t=1
to 30:next t:if (j and 4)=0 then go
sub 1340
1160 ge=ge-1:gosub 1350:j=peek(563
21):
1170 poke v+21,2:poke v+2,a:poke v+
3,177:for t=1 to 30:next t
1180 if (j and 8)=0 then gosub 1340
:ge=ge-1:gosub 1340
1190 j=peek(56321):if (j and 16)=0
then goto 1320
1200 next a:goto 1330

```

```

1210 poke v+21,4:for a=177 to 255:p
oke v+4,a:poke v+5,177:next
1220 poke v+21,8:poke v+6,255:poke
v+7,177
1230 me=ge/22:if me>7 then print he
$"ein sehr guter sprung!":goto 1290
1240 if me>6 then print he$"ein gut
er sprung!":goto 1290
1250 if me>5.5 then print he$"naja?
!nicht schlecht.":goto 1290
1260 if me>4.5 then print he$"aber,
aber!!bitte anstrengen.":goto 1290
1270 if me>.5 then print he$"sie bl
amieren sich doch nur!":goto 1290
1280 print he$"lassen sie dieses sp
iel":goto 1290
1290 print "meter = ";me:me(f)=me
1300 for t=1 to 1400:next t:poke v+
21,0
1310 goto 750
1320 if a>156 and a<165 then 1210
1330 print he$"falschsprung"a5$a5$a
5$sl$:poke v+21,0:for t=1 to 600:ne
xt t:goto 750
1340 ge=ge+3
1350 print he$d5$d5$d5$d5$d3$"gesch
windigkeit =";ge;c2$cl$cl$sl$
1360 if ge<2 then 1380
1370 return
1380 print cl$"sie sollten mal biss
chen laufen!"
1390 poke v+21,0:for x=1 to 1400:n
ext
1400 goto 750
1410 for t=1 to 1000:next
1420 print cl$d5$d5$d3$
1430 print tab(16)sl$s9$s9$s9$s9$
1440 print tab(15)sl$kl$a3$kl$14$
1450 printtab(15)sl$rn$a4$rf$sl$14$
1460 printtab(6)kl$sx$(4)sx$(4)15$rn
$a2$"1"sl$rf$sl$sx$(4)sx$(3)kl$14$
1470 print tab(6)rn$a5$"2"a5$a5$sl$
"3"a4$rf$sl$14$
1480 print tab(6)rn$a5$a5$a5$a5$a2$
rf$kl$
1490 poke 2042,204:poke 2043,204:po
ke 2044,204
1500 poke v+21,28:poke v+6,142:poke
v+7,132
1510 poke v+4,85:poke v+5,149:poke
v+8,199:poke v+9,149:
1520 print he$:for i=1 to 26:print
s9$:next :print s9$
1530 print "nummer"a3$m1$"name"a5$m
1$"meter"sl$12$
1540 print sx$(5)sx$(5)sx$(5)sx$(5)
sx$(5)sx$(2)
1550 for x=1 to sp

```



```

1560 print x,sp$(x),me(x)
1570 next
1580 for i=1 to 26:print bs$;:next
:print bs$
1590 for x=1 to 3000:next :get a$:i
f a$="" then 1590
1600 rem ===== abfrage =
1610 poke v+21,0:print cl$wh$"noch
einmal versuchen?"
1620 print c4$lg$rn$"j"rf$wh$" ja,
mit gleichen Spielern"
1630 print c4$lg$rn$"a"rf$wh$" ja,
aber mit anderen Spielern"
1640 print c4$lg$rn$"n"rf$wh$" nei
n,keine lust mehr!"
1650 get a$:if a$="" then 1650
1660 if a$="j" then f=0:goto 1710
1670 if a$="a" then restore :f=0:go
to 1710
1680 if a$="n" then print cl$:end
1690 goto 1650
1700 for x=1 to sp:me(x)=0:next :re
store :goto 670
1710 for x=1 to sp:me(x)=0:next :re
store :goto 630
1720 rem ===== laeuer 1 =
1730 data 0,0,0,0,0,0,0,0,0,0,0
1740 data 0,0,0,0,0,0,0,0,0,0,0
1750 data 0,0,0,0,1,0,0,0,128,0
1760 data 0,64,0,0,64,0,0,96,0,0
1770 data 240,0,0,248,0,0,240,0,0,9
6
1780 data 0,0,32,0,0,16,0,0,12,0,0,
0,0
1790 rem ===== laeuer 2 =
1800 data 0,0,0,0,0,0,0,0,0,0,0
1810 data 0,0,0,0,0,0,0,0,0,0,0
1820 data 0,0,0,0,0,0,0,0,0,0,0
1830 data 0,28,0,0,32,0,0,96,0,0
1840 data 240,0,0,248,0,0,240,0,0,9
6
1850 data 0,0,64,0,0,64,0,0,128,0,0,
0,0
1860 rem ===== springer =
1870 data 0,0,0,0,0,0,0,0,0,0,0
1880 data 0,0,0,0,0,0,0,0,0,0,0
1890 data 0,0,0,0,0,24,0,0,32,0,0
1900 data 64,0,0,64,0,0,127,128,0
1910 data 240,0,0,248,0,0,240,0,0,1
27
1920 data 128,0,64,0,0,64,0,0,32,0
1930 data 0,24,0
1940 rem ===== sieger =
1950 data 0,0,0,0,0,0,0,0,0,0,0
1960 data 0,0,0,0,0,0,0,0,0,112
1970 data 0,0,168,0,0,248,0,4,217,0
1980 data 4,113,0,4,33,0,2,114,0,1
1990 data 252,0,0,112,0,0,112,0,0,1

```

```

12
2000 data 0,0,216,0,0,216,0,0,216,0
2010 data 1,220,0
2020 rem ===== titelbild =
2030 poke 53280,0:poke 53281,14:pok
e 646,1:print cl$
2040 wx$=16$+16$+1u$+y9$+y3$+y9$+y3
$+y3$+y9$+y9$+y3$+y9$+1u$+16$+16$
2050 print c4$tab(17)wx$
2060 print tab(17)bl$sz$(5)sz$(5)sz
$(5)
2070 print wh$a3$16$1u$y9$y3$y3$y9$
1u$16$1$
2080 print tab(3)bl$sz$(3)sz$(3)lo$
bk$rn$bk$a5$a5$a5$a4$rf$
2090 wk$=16$+1u$+y9$+y3$+y3$+y9$
$+y9$+1u$+16$
2100 print tab(10)rn$bk$a4$"weitspr
ung"a5$rf$wh$wk$bk$
2110 print tab(10)rn$bk$a4$sx$(5)sx
$(5)a5$rf$bl$sz$(3)sz$(3)sz$(3)lo$b
k$
2120 print tab(10)rn$bk$a4$"copyrig
ht by"a3$rf$
2130 print tab(10)rn$bk$a3$"ali-osm
an koese"sl$rf$
2140 print tab(10)gl$sz$(3)lo$bk$y2
$gl$sz$(5)sz$(5)bk$y2$gl$sz$(3)bk$
2150 print tab(14)y2$tab(25)y2$
2160 print cy$rn$1z$
2170 print gl$c2$o5$o4$sl$o5$o5$o5$
o4$sl$o5$o5$
2180 print c2$o5$o3$sl$o5$o5$o5$o5$
y9$sl$o5$o4$
2190 print c2$o5$o2$sl$o5$o5$o5$o5$
o3$sl$o4$o4$
2200 print c2$o5$y9$sl$o5$o5$o5$o5$
o5$sl$o5$y9$
2210 print c2$rn$wh$::for i=1 to 40
:print sl$;:next :print rf$
2220 print rn$bl$c2$::for i=1 to 40
:print sl$;:next :print rf$
2230 print rn$bl$c2$::for i=1 to 40
:print sl$;:next :print rf$
2240 print rn$bl$c2$::for i=1 to 40
:print sl$;:next :print rf$
2250 print c2$rn$br$::for i=1 to 40
:print sl$;:next :print rf$
2260 print c2$rn$gr$::for i=1 to 40
:print sl$;:next :print rf$
2270 print c2$rn$gr$::for i=1 to 39
:print sl$;:next :print rf$wh$
2280 for x=0 to 5000:next :goto 620
2290 rem weitsprung =====64
2300 rem 32711 bytes memory =====
2310 rem 07430 bytes program =====
2320 rem 00672 bytes variables =====
2330 rem 00080 bytes arrays =====

```



```

10 rem 17+4 Kartenspiel =====64
20 rem (P) 01/86 CBM REVUE TEAM =
30 rem =====
40 rem (C) 11/85 by =
50 rem Ph. Bertschg =
60 rem Idee von A. Weingard =
70 rem Version 2.0 40Z/ASCII =
80 rem C-64 + 1530/1541 =
90 rem =====
100 pk$=chr$(097):ra$=chr$(122)
110 hz$=chr$(115):kr$=chr$(120)
120 cl$=chr$(147):he$=chr$(019)
130 c4$=chr$(017):rn$=chr$(018)
140 d2$=c4$+c4$:d3$=d2$+c4$
150 d4$=d3$+c4$:d5$=d4$+c4$
160 rf$=chr$(146):sl$=chr$(032)
170 a2$=s1$+s1$:a3$=a2$+s1$
180 a4$=a3$+s1$:a5$=a4$+s1$
190 aa$=a5$+a5$:ab$=aa$+aa$
200 c2$=chr$(145):h2$=c2$+c2$
210 c1$=chr$(157):c3$=chr$(029)
220 w2$=c3$+c3$:w3$=w2$+c3$
230 w4$=w3$+c3$:w5$=w4$+c3$
240 l4$=chr$(165):e4$=chr$(169)
250 l5$=chr$(163):e3$=chr$(127)
260 gosub 2830
270 s9$=pk$+ra$+hz$+kr$
280 c8$="23456789tbdka"
290 c9$=" 2 3 4 5 6 7 8 91010101011
"
300 gosub 2460,2520
310 print chr$(142);cl$
320 dim c(52),p(10),d(10),s$(52),c$
(52)
330 gosub 1220:t=0
340 j8=j
350 if a9<4 then 370
360 gosub 1220
370 if j<42 then 390
380 gosub 1220
390 print he$d5$d5$d3$"wieviel dm s
etzen sie";
400 input b1
410 k$=b$
420 if b1=0 then 1110
430 if b1>0 then 450
440 gosub 1220:b1=-b1
450 if b1>=1 then 480
460 print "mindestens 1 dm setzen!"
470 goto 340
480 if b1<=25000 then 520
490 if m1=1 then 1680
500 print he$d5$d5$d2$rn$"das limit
beträegt 25000 dm"rf$a5$a5$sl$
510 print ab$ab$c4$:goto 390
520 p=0:d=0:p9=0:d9=0:v=0:h=0
530 print cl$:m1=1:gosub 1370
540 gosub 1370,1450,2370
550 rem fre=1209#7-78-01#
560 j1=j:m1=2:gosub 1370,1450,1610
570 if p=21 then 620
580 if d<21 then 680
590 m1=1:gosub 1610
600 k$=h2$+rn$+"c64"+rf$+" hat 17 u
nd 4"
610 gosub 1580:goto 1010
620 m1=1:gosub 1610
630 if d<21 then 660
640 k$=h2$+"wir beide haben 17 und
4"
650 gosub 1580:goto 950
660 k$=h2$+"sie haben 17 und 4"
670 gosub 1580:b1=1.5*b1:goto 970
680 if p<22 then 720
690 k$=h2$+"sie haben sich totgekau
ft"
700 gosub 1580:m1=1:gosub 1610:m1=2
710 goto 1010
720 if v=0 then 740
730 goto 860
740 k$=c2$+"wollen sie noch eine ka
rte"
750 gosub 1580
760 print rn$"? "cl$rf$;:for i=1 to
150:next
770 print sl$cl$;:for i=1 to 150:ne
xt
780 get a$:if a$="" then 760
790 if a$="j" then 840
800 if a$="d" then 830
810 k$=c2$+ab$+sl$
820 gosub 1580:goto 860
830 b1=2*b1:v=1
840 k$=b$:m1=m1+1
850 gosub 1370:goto 680
860 m1=1:gosub 1610:m1=2
870 if d>16 then 890
880 m1=m1+1:gosub 1450,1610:goto 87
0
890 if d<=21 then 930
900 k$=h2$+"der c64 hat sich totgek
auft ----- "
910 k$=k$+"sie haben daher "+rn$+"g
ewonnen"+rf$
920 print he$d5$d5$c4$k$c4$:goto 990
930 if d<p then 970
940 if d>p then 1010
950 k$="wir haben karten-gleichstan
d!"
960 gosub 1580:goto 1040
970 k$="sie haben soeben "+rn$+"ge
wonnen"+rf$+"!"
980 gosub 1580
990 t=t+b1:goto 1040
1000 gosub 1580
1010 k$="sie haben gerade "+rn$+"ve

```



```

rloren"+rf$+"!"
1020 gosub 1580
1030 t=t-b1
1040 print :if t>0 then 1070
1050 if t<0 then 1090
1060 print h2$"ihr kontostand ist a
usgeglichen (null)":goto 340
1070 print h2$"sie besitzen jetzt d
m ";t
1080 goto 340
1090 print h2$"ihre schulden betrag
en dm",-t
1100 goto 340
1110 print c1$
1120 if t=0 then 1170
1130 if t<0 then 1200
1140 print "herzlichen glueckwunsch
! sie duerfen"c4$
1150 print t" dm nach hause tragen!
"
1160 goto 1210
1170 print "sie haben mit ausgeglic
henem kontostand":print
1180 print "(null) das spiel beende
t."
1190 goto 1210
1200 print "wir bedauern sehr, aber
sie verlieren "c4$dm";-t;c4$
1210 print :print : "vielen dank fue
rs mitspielen!":end
1220 rem ***** misch-routine *
1230 print he$d5$d5$d3$"ich mische
die karten!
1240 for i=1 to 4:for j=1 to 13
1250 w1=13*(i-1)+j
1260 s$(w1)=mid$(s9$,i,1)
1270 c$(w1)=mid$(c8$,j,1)
1280 c(w1)=val(mid$(c9$,2*j-1,2))
1290 next j:next i
1300 for i=1 to 52:j=52*rnd(1)+1
1310 tt$=s$(i):s$(i)=s$(j)
1320 s$(j)=tt$:tt$=c$(i)
1330 c$(i)=c$(j):c$(j)=tt$
1340 tt=c(i):c(i)=c(j):c(j)=tt
1350 next i:a9=0:j=0:j1=2:j8=0
1360 return
1370 j=j+1:p(m1)=j
1380 if c(j)<11 then 1400
1390 p9=p9+1:a9=a9+1
1400 p=p+c(j)
1410 if p<22 then 1440
1420 if p9=0 then 1440
1430 p=p-10:p9=p9-1
1440 print he$d5$d5$d3$:goto 1630
1450 j=j+1:d(m1)=j
1460 if c(j)<11 then 1480
1470 d9=d9+1:a9=a9+1
1480 d=d+c(j)
1490 if d<22 then 1520
1500 if d9=0 then 1520
1510 d=d-10:d9=d9-1
1520 return
1530 for i=1 to 52
1540 p(i)=0:d(i)=0:next i:return
1550 for i=1 to 52
1560 print c$(i)+s$(i);
1570 next i:print :return
1580 print he$d5$d5$c4$;
1590 print ab$aa$a5$a2$
1600 print k$;:return
1610 p5=0:print he$;:goto 1640
1620 goto 1640
1630 p5=1
1640 for l=1 to 8
1650 if m1=1 then 1680
1660 for i5=1 to m1-1
1670 print w5$w3$;:next i5
1680 if p5=1 then 1700
1690 i=d(m1):goto 1710
1700 i=p(m1)
1710 print rn$;:r=c(i)
1720 if l>1 then 1740
1730 goto 2290
1740 if l>2 then 1800
1750 if c$(i)="t" then 1780
1760 print right$(c$(i),1);s$(i);a5
$;
1770 goto 2360
1780 print "10";s$(i);a4$;
1790 goto 2360
1800 if l>3 then 1870
1810 if c(i)<11 then 1830
1820 goto 2290
1830 if c(i)>2 then 1850
1840 goto 2290
1850 if c(i)>3 then 2330
1860 goto 2310
1870 if l>4 then 2020
1880 if c(i)<11 then 1900
1890 goto 2310
1900 if c(i)<9 then 1920
1910 goto 2350
1920 if c(i)>3 then 1940
1930 goto 2310
1940 if c(i)>4 then 1960
1950 goto 2290
1960 if c(i)>5 then 1980
1970 goto 2310
1980 if c(i)>6 then 2000
1990 goto 2330
2000 if c(i)>7 then 2330
2010 goto 2310
2020 if l>5 then 2120
2030 if c(i)<11 then 2050
2040 goto 2290
2050 if c(i)<10 then 2070

```



```

2060 goto 2350
2070 if c(i)<6 then 2090
2080 goto 2330
2090 if c(i)<4 then 2110
2100 goto 2290
2110 goto 2310
2120 if l>6 then 2210
2130 if c(i)<11 then 2150
2140 goto 2290
2150 if c(i)<7 then 2170
2160 goto 2330
2170 if c(i)=4 then 2200
2180 if c(i)=5 then 2200
2190 goto 2290
2200 goto 2330
2210 if l>7 then 2280
2220 print " ";
2230 if c$(i)="t" then 2260
2240 print s1$;s$(i);right$(c$(i),1
);
2250 goto 2360
2260 print s$(i);"10";
2270 goto 2360
2280 goto 2290
2290 print a5$a2$;
2300 goto 2360
2310 print a3$;s$(i);a3$;
2320 goto 2360
2330 print a2$;s$(i);s1$;s$(i);a2$;
2340 goto 2360
2350 print a2$;s$(i);s$(i);s$(i);a2
$;
2360 print :next i:print :return
2370 print he$rn$a5$a2$rf$14$
2380 print rn$a5$a2$rf$14$
2390 print rn$s1$rf$e4$a2$rn$a3$rf$
14$
2400 print rn$s1$rf$s1$rn$a2$rf$s1$
rn$e4$s1$rf$14$
2410 print rn$s1$rf$s1$rn$a2$rf$15$
e3$rn$s1$rf$14$
2420 print rn$s1$e3$rf$a2$rn$a3$rf$
14$
2430 print rn$a5$a2$rf$14$
2440 print rn$a5$a2$rf$14$
2450 return
2460 print cl$:for i=1 to 100
2470 print he$d5$d5$d3$;tab(15)
2480 print "1"rn$7"rf$s1$rn$"u"rf$
"n"rn$d"rf$s1$rn$"0"rf$4"
2490 print he$d5$d5$d2$;tab(15)
2500 print rn$1"rf$7"rn$s1$rf$"u"
rn$"n"rf$d"rn$s1$rf$"0"rn$4"
2510 next :return
2520 print cl$w5$w3$rn"spielregel 1
7 und 4"
2530 print chr$(14):print :print
2540 print "gespielt wird mit einem

```

```

kartenspiel, das
2550 print "folgende karten enthael
t:"c4$
2560 print "2 3 4 5 6 7 8 9 b d k a
"c4$
2570 print "dabei zaehlen die bilde
r (b,d,k) jeweils
2580 print "zehn punkte, as 11 punk
te, aber wenn 21"c4$
2590 print "ueberschritten werden,
nur 1 punkt."c4$
2600 print "die zahlen zaehlen ihre
n eigenwert. wer"c4$
2610 print "die hoechste punktezahl
(max.21!) er-"c4$
2620 print "reicht (c64 oder sie),
gewinnt.
2630 print c4$a5$a3$"verstanden (j/
n)"
2640 get a$:if a$="" then 2640
2650 if a$="j" then 2670
2660 goto 2520
2670 print cl$"sie erhalten zwei ka
rten und koennen so-
2680 print "viele nachziehen wie s
ie wollen, aber"c4$
2690 print "sie duerfen"rn$ 21 pun
kte nicht ueberschrei"
2700 print rn$ten,"rf$" sonst verl
ieren sie. gleiches gilt"c4$
2710 print "fuer den c64. wenn sie
das spiel been-"c4$
2720 print "den wollen, setzen sie
bitte null dm!"c4$
2730 print "wenn sie die frage, ob
sie noch eine"c4$
2740 print "karte wollen, statt mit
'j' wie ja mit"c4$
2750 print "'d' wie doppelt beantwo
rten, dann erhal-
2760 print "ten sie eine weitere ka
rte, zusaetzlich"c4$
2770 print "wird ihr einsatz verdop
pelt.
2780 print d2$aa$s1$"verstanden? (j
/n)"
2790 get a$:if a$="" then 2790
2800 if a$="j" then return
2810 if a$="n" then 2520
2820 goto 2790
2830 data 169,0,141,32,208,141,33,
208,141,0,4,141,1,4
2840 data 141,2,4,169,149,141,0,22
1,169,5,141,24
2850 data 208,169,128,141,136,2,13
3,56,96,96
2860 for i=50000 to 50035:read a:po
ke i,a:next

```



```

2870 sys 50000:print chr$(147):prin
t chr$(5)
2880 return
2890 rem 17+4 Kartenspiel =====64
2900 rem 038911 Bytes Memory      ===
2910 rem 006823 Bytes Program     ==
2920 rem 000224 Bytes Variables  ==
2930 rem 000000 Bytes Arrays      ==
2940 rem 000091 Bytes Strings     ==
2950 rem 031773 Bytes Free (0)    ==
2960 rem =====

```

Biorythmus 64 - 1 -

```

10 rem Biorythmus =====64
20 rem (P) 01/86 CBM REVUE TEAM =
30 rem =====
40 rem (C) 01/86 by =
50 rem Daniel Rügemer =
60 rem Rottenburg =
70 rem Version 2.0 40Z/ASCII =
80 rem C-64 + 1530/1541 =
90 rem =====
100 c4$=chr$(017):rn$=chr$(018)
110 s1$=chr$(032):z1$=chr$(045)
120 rf$=chr$(146)
130 pi=3.14159265:m1$=chr$(125)
140 e2$=chr$(112):s7$=chr$(103)
150 for q=1 to 39:q1$=q1$+z1$:next q
160 print chr$(147);chr$(014):poke
59468,14
170 print q1$
180 print " Biorythmus fuer alle Ge
legenheiten"
190 print q1$
200 print c4$"Bitte Name eingeben:"
210 print :input n$
220 print c4$"Geburts- Tag ..., Mona
t ..., Jahr ...."
230 print :input gt,gm,gj
240 if gj<100 then print c4$"Sie me
inen wohl "1900+gj:gj=gj+1900
250 print c4$"Biorhythmus fuer Monat
..., Jahr ...."
260 print :input bm,bj
270 if bj<100 then print c4$"Sie wo
llten sagen "bj+1900:bj=bj+1900
280 print c4$" s=seelisch
290 print c4$s1$rn$"G"rf$="geistig"
300 print c4$" P=physisch
310 def fny(x)=int(15%sin((t+n1)*dd
)+18)
320 a$=chr$(13)+chr$(145)
330 ge=23:s1=28:ph=33
340 pb=2*pi/ge:sb=2*pi/s1:gb=2*pi/p
h

```

```

350 rem ***** Perioden der *
360 rem *** verschiedenen Rythmen *
370 if gm>2 or (gt=2 and gt=29) the
n 380
380 restore
390 for i=1 to bm:read en:next :res
tore
400 if bj/4=int(bj/4) and bm=2 then
en=29
410 n1=1:for a=1 to gm:read x:next
420 n1=n1-gt+x
430 if gm<>12 then for a=gm+1 to 12
:read x:n1=n1+x:next
440 if bj-gj<2 then 490
450 for a=gj-1899 to bj-1901
460 dz=365:if a/4=int(a/4) then dz=
366
470 n1=n1+dz
480 next
490 if bm=1 then 510
500 restore :for a=1 to bm-1:read x
:n1=n1+x:next
510 f1=int(bj/4):f2=bj/4
520 if bm>2 and f1=f2 then n1=n1+1
530 print c4$" Druecken Sie eine Ta
ste.":gosub 690
540 print chr$(147):t=0
550 t=t+1:if t>en then 630
560 dd=pb:aa=fny(x):dd=sb:bb=fny(x)
:dd=gb:cc=fny(x)
570 t1=aa+3:t2=bb+3:t3=cc+3:if t=25
then gosub 690
580 print tab(t1)e2$a$tab(t2)"s"a$
tab(t3)rn$s7$a$tab(21)m1$a$tab(4)m1
$a$;
590 print tab(37)m1$:goto 550
600 t=t+1:if t>en then 630
610 data 31,28,31,30,31,30,31,31,30
,31,30,31
620 rem ***** Tage pro Monat *
630 print c4$::input " Weitere Bere
chnungen":f$
640 if f$<>"j" and f$<>"n" then pri
nt c4$" j/n":goto 630
650 if f$="n" then 730
660 print c4$" Fuer "n$::input f$:i
f f$<>"j" and f$<>"n" then 660
670 if f$="n" then run
680 if f$="j" then goto 250
690 for i=1 to 10:get g$:next
700 rem * Routine warte auf Taste *
710 get g$:if g$="" then 710
720 return
730 end
740 rem Biorythmus =====64
750 rem 38911 Bytes Memory      =====
760 rem 02270 Bytes Program     =====
770 rem 00301 Bytes Variables  =====

```


20/64/128

Das unabhängige Commodore-Magazin

KOMMT REGELMÄSSIG ZU IHNEN INS HAUS

Finden Sie Ihre COMMODORE-WELT nicht am Kiosk? Weil sie schon ausverkauft ist? Oder „Ihr“ Kiosk nicht beliefert wurde? Kein Problem! Für ganze 30 DM liefern wir Ihnen per Post sechs Hefte ins Haus (Ausland 40 DM). Einfach den Bestellschein ausschneiden – fotokopieren oder abschreiben, in einen Briefumschlag und ab per Post (Achtung: Porto nicht vergessen). COMMO-DORE-WELT kommt dann pünktlich ins Haus.



WICHTIGE RECHTLICHE GARANTIE!

Sie können diesen Abo-Auftrag binnen einer Woche nach Eingang der Abo-Bestätigung durch den Verlag widerrufen – Postkarte genügt. Ansonsten läuft dieser Auftrag jeweils für sechs Ausgaben, wenn ihm nicht vier Wochen vor Ablauf widersprochen wird, weiter.

NUTZEN SIE UNSEREN BEQUEMEN POSTSERVICE

ABO SERVICE-KARTE

4/86

Ich nehme zur Kenntnis, daß die Belieferung erst beginnt, wenn die Abo-Gebühr dem Verlag zugegangen ist.

COMMODORE WELT

Abo-Service 4/86
Postfach 1107
8044 UNTERSCHLEISSHEIM

Coupon

Ja, ich möchte von Ihrem Angebot Gebrauch machen.

Bitte senden Sie mir bis auf Widerruf ab sofort jeweils die nächsten

sechs Ausgaben an untenstehende Anschrift. Wenn ich nicht vier Wochen vor Ablauf kündige, läuft diese Abmachung automatisch weiter.

Name _____ (4)

Vorname _____

Straße/Hausnr. _____

Plz/Ort _____

Ich bezahle:

☐ per beiliegendem Verrechnungsscheck

☐ gegen Rechnung

☐ bargeldlos per Bankeinzug von meinem Konto

bei (Bank) und Ort _____

Kontonummer _____

Bankleitzahl _____

(steht auf jedem Kontoauszug)

Unterschrift _____

Von meinem Widerspruchsrecht habe ich Kenntnis genommen.

Unterschrift _____

ANZEIGENSERVICE

Die große Börse für jeden Zweck in der CBM REVUE / COMMODORE-WELT. Kostenlos für Privat-Inserenten. Spottbillig für gewerbliche Anbieter. Einfach Coupon ausschneiden, fotokopieren o.ä., ausfüllen und ab die Post — Freimachen nicht vergessen! — Unsere Adresse steht auf dem Coupon, ebenso die Preise für gewerbliche Anbieter! Achtung! Wir weisen ausdrücklich darauf hin, daß wir offensichtlich gewerbliche Anzeigen nicht kostenlos veröffentlichen und uns jedweden Abdruck kostenloser Anzeigen vorbehalten müssen, insbesondere, wenn deren Inhalt nicht CBM-typisch ist oder gegen geltendes Recht verstößt. Private Chiffreanzeigen werden nicht aufgenommen. Für Privatanbieter: maximal acht Zeilen à 28 Anschläge. Für gewerbliche Anbieter: 5 DM p. mm.

[illegible]**CW**

Anzeigenabteilg.

Postfach 1107

8044 Lohhof

Name _____

Vorname _____

Straße/Hausnr. _____

PLZ/Ort _____

DAS SONDERANGEBOT: PRIVATE KLEINANZEIGEN KOSTENLOS!

Das bietet Ihnen COMMODORE-WELT: KLEINANZEIGEN SIND KOSTENLOS FÜR PRIVATANBIETER! Suchen Sie etwas, haben Sie etwas zu verkaufen, zu tauschen, wollen Sie einen Club gründen? Coupon ausfüllen, auf Postkarte kleben oder in Briefumschlag stecken und abschicken. So einfach geht das. Wollen Sie das Heft nicht zerschneiden, können Sie den Coupon auch fotokopieren. Oder einfach den Anzeigentext uns so schicken, auf Postkarte oder im Brief. Aber bitte mit Druckbuchstaben oder in Schreibmaschinenschrift!

Und: Einschließlich Ihrer Adresse und/oder Telefonnummer sollten acht Zeilen à 28 Anschläge nicht überschritten werden.

ACHTUNG: WICHTIGER HINWEIS!

Wir veröffentlichen nur Kleinanzeigen privater In-

**serenten, keine gewerblichen Anzeigen. Die kosten
pro Millimeter DM 5.00 plus Mehrwertsteuer!**

Wir versenden für Privat-Inserenten keine Beleg-Exemplare!

Chiffre-Anzeigen sind nicht gestattet! Wir behalten uns vor, Anzeigen, die gegen rechtliche, sittliche oder sonstige Gebote verstoßen, abzulehnen!

Anzeigenabdruck in der Reihenfolge ihres Eingangs,
kein Rechtsanspruch auf den Abdruck in der nächsten
Ausgabe!

Die Insertion ist nicht vom Kauf des Heftes abhängig!

Wir behalten uns vor, Anzeigen, die nicht zum Themenkreis des Heftes – Computer – gehören, nicht abzurufen oder sie nur insoweit zu berücksichtigen, wie es der Umfang des kostenlosen Anzeigenteils zuläßt.

PROGRAMMSERVICE

Hiermit bestelle ich

die Listings dieses Heftes auf

☐ Kassette (10 DM) ☐ Diskette (25 DM)

Zutreffendes bitte ankreuzen!

Ich zahle:

per beigefügtem Scheck / Schein ()

Gegen Bankabbuchung am Versandtag ()

Meine Bank (mit Ortsname)

Meine Kontonummer

Meine Bankleitzahl (steht auf jedem Bankauszug)

Vorname Nachname

Str./Nr. PLZ / Ort

Hiermit bestätige ich, Ihre Verkaufsbedingungen gelesen zu haben und zu akzeptieren. (3)

Unterschrift

Bitte ausschneiden und einsenden an

COMMODORE-WELT

KASSETTENSERVICE 4/86

Postfach 1107

8044 Unterschleißheim

LESER WERBEN LESER

GEWINNEN SIE EINE COMPUTER UHR! Und zusätzlich eventuell noch ein großes Commodore-Buch. Oder ein Paket Disketten. ODER AUCH EINEN COMMODORE-DRUCKER — ODER EINE DISKETTENSTATION! Wie? Sie werben einen Abonnenten. Dann haben Sie auf jeden Fall schon die Computer-Uhr gewonnen. Zusätzlich verlosen wir unter allen, die mitmachen, jeden Monat vier weitere wertvolle Preise. Und alle sechs Monate gibt es einen Hauptpreis unter allen Abo-Werbern zu gewinnen. Also: Mitmachen. Mitgewinnen.

Ja, ich mache mit beim Abo-Wettbewerb. Ich habe



Herrn/Frau.....

.....

Straße/Hausnr.

.....

PLZ/Ort

.....

als neuen Abonnenten der COMMODORE WELT gewonnen.

Der neue Abonnent war bisher noch nicht Bezieher dieser Zeitschrift.

Als Prämie erhalte ich nach Eingang des Abo-Entgeltes auf jeden Fall eine Computer-Uhr, wie abgebildet, und nehme zusätzlich noch an der Verlosung des Monats sowie der halbjährlichen Hauptpreise teil. Mir ist bekannt, daß der Rechtsweg bei den Verlosungen ausgeschlossen ist.

Meinen Preis senden Sie an

Name

Straße/Hsnr.

PLZ/Ort

(Bitte ausschneiden und zusammen mit der Abobestellkarte unten einsenden!)

VC 20 * 64 * 128
VC 20 * C 64 * 128 PC

**Rund
150
Seiten
Listings
TIPS +
TRICKS**

**Anwender-
Programme
Utilities
Schul-Programme
Spiele
Adventures**

Rund 150 Seiten, voll mit Listings, Service, Tips und Tricks. Ein Muß für jeden Commodore-User, ob mit dem VC 20 oder dem 128 — selbstverständlich erst recht für den 64er Fan! Dazu ein attraktives Angebot: Reservieren Sie sich Ihre Ausgabe mit untenstehendem Coupon! Sie erhalten das Heft nicht nur früher per Post ins Haus geliefert, sie sparen auch noch glatte 2 Mark, also rund 15 Prozent!

ACHTUNG: DIESES SUBSKRIPTIONSANGEBOT IST BEFRISTET BIS ZUM 10. April!

RESERVIERUNGS-KARTE

Bitte reservieren Sie mir () Exemplar(e)
COMMODORE-WELT SPECIAL Nr. 1/86
zum Preis von DM 12,80

Ich zahle

() Per beigefügtem Scheck*

() Per Bankabbuchung am Versandtag*

Name _____

Straße/Hausnr. _____

Ort/PLZ _____

Bei Bankabbuchung:

Kto-Nr.: _____ Bankleitzahl _____

Bei (Bank und Ort) _____

Unterschrift _____

Ausschneiden und einsenden

* = Abbuchung erst am Versandtag!

**COMMODORE
WELT
Reservierungs-
Service 5
Postfach 1107
8044 Lohhof**


```

10 rem Vokabelpauker =====64
20 rem (P) 01/86 CBM REVUE TEAM =
30 rem =====
40 rem (C) 01/86 by =
50 rem Daniel Rügemer =
60 rem Rottenburg =
70 rem Version 2.0 40Z/ASCII =
80 rem C-64 + 1530/1541 =
90 rem =====
100 c1$=chr$(147):c4$=chr$(017)
110 d2$=c4$+c4$:d3$=d2$+c4$
120 d4$=d3$+c4$:z1$=chr$(045)
130 c3$=chr$(029):rn$=chr$(018)
140 w2$=c3$+c3$:w3$=w2$+c3$
150 w4$=w3$+c3$:w5$=w4$+c3$
160 rf$=chr$(146):xe$=chr$(046)
170 sl$=chr$(032):hk$=chr$(039)
180 a2$=sl$+sl$:a3$=a2$+sl$
190 rt$=chr$(013):xf$=chr$(047)
200 for q=1 to 39:q1$=q1$+z1$:next q
210 print c1$c4$xe$:p=peek(32808):p
rint c1$;
220 print chr$(14):dim a$(999)
230 f=0:print c1$q1$
240 print " "rn$"Menuwahl"
250 print q1$
260 print c4#w2$"1."sl$rn$"Eingabe
"rf$sl$"Vokabeln eingeben"
270 print c4#w2$"2."sl$rn$"Abhoeren
"rf$
280 print c4#w2$"3."sl$rn$"save....
"rf$"Lektion auf Band/Disk"
290 print c4#w2$"4."sl$rn$"load....
"rf$"Vokabeln vom Band/Disk"
300 print c4#w2$"5."sl$rn$"print...
"rf$"Anlegen einer Liste"
310 print tab(14)"aller Vokabeln au
f"
320 print tab(14)"Drucker oder Moni
tor"
330 print c4#w2$;:input "Welche Nr.
waelhlen Sie ";a
340 on agoto 350,500,740,820,1000
350 i=0
360 rem ***** eingabe *
370 print d2$w5$rn$"Eingabe"
380 print c4#w3$;:input "neue Lekti
on (j/n)";xx$:if xx$="j" then 440
390 print c4#w3$;:input "ab welchem
Wort (Fremdspr.) ";xx$
400 i=0
410 i=i+1:if a$(i)=xx$ then i=i+1:g
oto 440
420 if a$(i)<>" " then 410
430 print c4$"das Wort ' "xx$" ' gi
bt es gar nicht!":goto 370
440 print c4#w3$"Wenn Sie fertig si
nd , geben Sie
450 print c4#w3$" 'menu' ein .
460 print c4$;:input "Wort";a$(i)
470 if a$(i)="menu" then goto 230
480 i=i+1:print c4$;:input "Wort (F
remdspr.)";a$(i)
490 i=i+1:goto 460
500 i=0:rem ***** abhoeren *
510 print c4#w2$;:input "ganz von v
orn (j/n) ";xx$:if xx$="j" then i=0
:goto 580
520 print c4#w2$;:input "ab welchem
Wort (Fremdspr.) ";xx$
530 i=0
540 i=i+1:if a$(i)=xx$ then i=i+1:g
oto 570
550 if a$(i)<>" " then 540
560 print c4#w2$hk$xx$hk$"gibt es g
ar nicht !":goto 510
570 i=i-2
580 print c1$c4$"1. Deutsch --
-> Fremdsprache"
590 print c4$"2. Fremdsprache --->
Deutsch"
600 print c4$;:input "Welche Nummer
waelhlen Sie ";x
610 y=1:if x=2 then i=i+1:y=-1
620 print c1$d4$w4$rn$"Abhoeren
630 print c4$"Was heisst "a$(i);:in
put a$
640 if a$=a$(i+y) then print c4$"Ri
chtig !"
650 if a$<>a$(i+y) then print c4$"F
alsch !!!":f=f+1
660 print c4$"das war der "f".Fehle
r!"
670 if a$<>a$(i+y) then goto 1130
680 i=i+2:if a$(i)="" or a$(i)="men
u" then 700
690 goto 630
700 print c4$"Das war der Test."
710 print c4$"Sie haben "f" Fehler
gemacht."
720 for i=1 to 3000:next :goto 230
730 rem ***** save *
740 print c4$;:input " Was war das
fuer eine Lektion";z$
750 print c4#c3$;:input "Cassette o
der Disk";cd$:if cd$<>"c" and cd$<>
"d" then 750
760 a=0:if cd$="c" then open 1,1,2,
z$
770 if cd$="d" then open 1,8,2,"$0:
"+z$+",s,w"
780 print# 1,a$(a);rt$;:if a$(a)<>"
" then a=a+1:goto 780
790 close 1
800 goto 230
810 rem ***** load *

```



```

820 print c4$;:input "Welche Lektio
n ";z$
830 i=0:print c4$c3$;:input "Casset
te oder Disk";cd$
840 if cd$<>"d" and cd$<>"c" then 8
30
850 if a$(0)="" and a$(1)="" then 9
20
860 print c4$c3$"Soll ich "z$" an d
ie alte Lektion
870 print c4$c3$;:input "dranhaenge
n (j/n)";an$:if an$<>"j" and an$<>"
n" then 870
880 if an$="n" then 920
890 for i=0 to 999:if a$(i)="menu"
or a$(i)="" then 910
900 next
910 rem ***** Floppy *
920 if cd$="d" then open 1,8,2,"0:"
+z$+",s,r":goto 950
930 print c4$w2$"ich suche .."
940 open 1,1,0,z$
950 print c4$"ich hab's"
960 print d2$w2$"bitte noch etwas G
eduld !
970 input# 1,a$(i):if a$(i)<>" the
n i=i+1:goto 970
980 close 1:goto 230
990 rem ***** print *
1000 print c4$w3$;:input "Auf Druck
er oder Bildschirm (b/d)";xx$
1010 if xx$<>"d" and xx$<>"b" then
1000
1020 if xx$="b" then 1130
1030 print cl$d3$w5$c3$"Bitte Druck
er einschalten.

1040 print c4$c3$;:input "Tab";tb
1050 print c4$w3$;:input "Welche Le
ktion war das";xx$
1060 xx=xx+1:if a$(xx)<>" then 1060
1070 open 1,4
1080 print# 1,tb(tb+3)c4$xx$
1090 for i=0 to 2:print# 1,c4$:next
1100 for i=0 to xx-3 step 2
1110 print# 1,tb(tb)c4$a2$a$(i)s1$
xf$s1$a$(i+1):next
1120 close 1:goto 230
1130 i=0
1140 for x=0 to 30:next
1150 print a$(i)s1$xf$s1$a$(i+1):i=
i+2:if a$(i)<>" and a$(i)<>"menu"
then 1150
1160 goto 230
1170 rem *** verbesserung mit rn$ *
1180 b$a$(i+y):for x=1 to len(b$)
1190 if mid$(b$,x,1)<>mid$(a$,x,1)
then 1220
1200 c$c$+mid$(b$,x,1)+rf$:next
1210 goto 1230
1220 c$c$+rn$:goto 1200
1230 print c4$w2$"Es heisst"xf$c$xf
$:c$="":goto 680
1240 rem vokabelpauker =====64
1250 rem 38911 bytes memory ====
1260 rem 04022 bytes program ====
1270 rem 00203 bytes variables ====
1280 rem 03007 bytes arrays ====
1290 rem 00820 bytes strings ====
1300 rem 30859 bytes free ====
1310 rem =====

```

Variablenliste Vokabelpauker
Daniel Rügemer

A\$(0) bis A\$(999) Vokabel "1000"

CL\$ = CHR\$(147)	Clear Home
C4\$ = CHR\$(017)	Cursor down
D2\$ = 2*CHR\$(017)	2x Cursor down
D3\$ = 3*CHR\$(017)	3x Cursor down
D4\$ = 4*CHR\$(017)	4x Cursor down
Z1\$ = CHR\$(045)	Grafikzeichen (-)
C3\$ = CHR\$(029)	Cursor right
RN\$ = CHR\$(018)	Reverse on
W2\$ = 2*CHR\$(029)	2x Cursor right
W3\$ = 3*CHR\$(029)	3x Cursor right
W4\$ = 4*CHR\$(029)	4x Cursor right

W5\$ = 5*CHR\$(029)	4x Cursor right
RF\$ = CHR\$(146)	Reverse off
XE\$ = CHR\$(042)	Punkt
S1\$ = CHR\$(032)	Space
HK\$ = CHR\$(039)	Grafikzeichen (')
A2\$ = 2*CHR\$(32)	Space
A3\$ = 3*CHR\$(32)	3x Space
RT\$ = 35*CHR\$(32)	35x Space
XF\$ = CHR\$(047)	Grafikzeichen (/)
Q = 40	
Q1\$ = 40*CHR\$(45)	40x (-)
P = 77	
F = 0	
I = 1151	
XX\$ = "J"	
X = 1	
Y = 1	


```

10 rem Kundenkartei =====20
20 rem (P) 1986 Commodore Welt ts =
30 rem =====
40 rem (C) by Jerzy Komorowski =
50 rem Version 2.0 22Z/ASCII =
60 rem =====
70 rem VC 20 =
80 rem 1530/1541 =
90 rem =====

100 c4$=chr$(017):cl$=chr$(147)
110 rn$=chr$(018):rf$=chr$(146)
120 wh$=chr$(005):c3$=chr$(029)
130 sl$=chr$(032):he$=chr$(019)
140 c2$=chr$(145):cl$=chr$(157)
150 zc$=chr$(184)
160 goto290
170 poke198,0:wait198,1:getq$:return
180 printc4$"weiter mit "rn$"space"
: return
190 printcl$c4$"bitte band position
ie-ren"
200 printc4$"wenn fertig-"rn$"space"
: print
210 gosubl70:ifasc(q$)<>32then210
220 return
230 printc4$k$(i):printc4$n$(i):pri
ntc4$s$(i):printc4$w$(i):printc4$t$
(i)
240 ife=1thenreturn
250 printc4$c4$rn$"f1=weiter.....f3
=ende"
260 gosubl70:ifasc(q$)<>133andasc(q
$)<>134then260
270 ifasc(q$)=134theni=z
280 f=1:return
290 kd$="*****kundenkartei*****"
300 dimk$(20),n$(20),s$(20),w$(20),
t$(20)
310 printcl$c4$kd$::fori=0to21:prin
tzc$::next
320 printspc(48)"1.speichern":print
spc(26)"2.laden":printspc(26)"3.neu
e angaben"
330 printspc(26)"4.suchen":printspc
(26)"5.loeschen":printspc(26)"6.end
e"
340 gosubl70:x=val(q$):ifx<1orx>6th
en340
350 ifx=6thenend
360 onxgoto370,410,450,560,800
370 gosubl90:open1,1,2:print#1,z
380 fori=1toz
390 print#1,k$(i):print#1,n$(i):pri
nt#1,s$(i):print#1,w$(i):print#1,t$
(i)
400 next:close1:poke198,0:goto310

410 gosubl90:open1,1,0:input#1,z
420 fori=1toz
430 input#1,k$(i):input#1,n$(i):inp
ut#1,s$(i):input#1,w$(i):input#1,t$
(i)
440 next:close1:poke198,0:goto310
450 printcl$:z=z+1
460 printc4$"kund.nr.":printc4$"na
me":printc4$"str.":printc4$"ort.:"
"
470 printc4$"tel.:"
480 printhe$spc(52):inputk$(z):prin
tspc(26):inputn$(z):printspc(26):in
puts$(z)
490 printspc(26):inputw$(z):printsp
c(26):inputt$(z)
500 printc4$rn$"f1=korektur":printc
4$rn$"f3=weitere angaben"
510 printc4$rn$"f5=menue"
520 gosubl70:ifasc(q$)<>133andasc(q
$)<>134andasc(q$)<>135then520
530 ifasc(q$)=133then480
540 ifasc(q$)=134then450
550 ifasc(q$)=135then310
560 printcl$tab(4)"suchen nach:"
570 printspc(46)"1.kundennummer":pr
intspc(24)"2.name":printspc(24)"3.w
ohnort"
580 printspc(24)rn$"4.hauptmenue"
590 gosubl70:x=val(q$):ifx<1orx>4th
en590
600 onxgoto610,670,720,310
610 printcl$c4$"kundennummer-":inpu
tk$
620 fori=1toz:f=0
630 ifleft$(k$(i),len(k$))=k$thengo
sub230
640 next:iff=0thenprintc4$rn$"numme
r nicht gefunden"
650 gosubl80
660 gosubl70:ifasc(q$)<>32then660:g
oto310
670 printcl$"name:":inputn$
680 fori=1toz:f=0
690 ifleft$(n$(i),len(n$))=n$thengo
sub230
700 next:iff=0thenprintc4$rn$"name
nicht gefunden":gosubl80
710 gosubl70:ifasc(q$)<>32then710:g
oto310
720 printcl$"wohnort:":inputw$
730 fori=1toz:f=0
740 p=val(left$(w$,1)):ifp=0then760
750 ifleft$(w$(i),len(w$))=w$thengo
sub230:goto770
760 ifmid$(w$(i),5,len(w$)-1)=right
$(w$,len(w$)-1)thengosub230
770 next:iff=0thenprintc4$rn$"wohno

```



```

rt nicht gefunden":gosub180
780 gosub170:ifasc(q$)<>32then780
790 goto310
800 printcl$"name des kunden,der ge
loescht werden soll:":inputn$
810 fori=1toz:e=0:f=0
820 ifn$(i)=n$thene=1:gosub230
830 ife=1thenprintc4$rn$"f1=korektu
r...f3=ok.":goto880
840 next:iff=0thenprintc4$rn$"kunde
nicht gefunden"
850 gosub180
860 gosub170:ifasc(q$)<>32then860
870 e=0:goto310
880 gosub170:ifasc(q$)<>133andasc(q
$)<>134then880
890 ifasc(q$)=133theni=z:next:goto8
00
900 goto310
910 k$(i)="" :n$(i)="" :s$(i)="" :w$(i
)="" :t$(i)="" :gosub250:goto840
920 rem lottozahlen =====20
930 rem 28079 bytes memory =====
940 rem 02939 bytes programm =====
950 rem 00091 bytes variables =====
960 rem 00350 bytes arrays =====
970 rem 00011 bytes strings =====
980 rem 24688 bytes fre =====
990 rem =====

```

TIPS & TRICKS

Tips und Tricks VC-20

Diese kleine Routine ist geschrieben für den VC-20 + 16 K. Sie bewirkt, daß der VC-20 auf einer Matrix von 22 Spalten und 16 Reihen, doppelt hohe Zeichen darstellt. Leider aber können keine Reversen-Zeichen dargestellt werden.

```

100
FORI=0TO16*130:POKE5120+I,PEEK(32768+I/2)
:NEXT
110
FORI=0TO11*130-1:POKE5120+I+48*16,PEEK(32
768+I/2+8*48):NEXT
150
POKE36869,205:POKE36867,27:POKE36867,33:P
OKE36865,20:PRINTCHR$(147):PRINTCHR$(18);
CHR$(30):POKE36879,25
200 PRINTCHR$(145)"**** CBM BASIC V2
****"
202 PRINTCHR$(157);FRE(0);"BYTES FRE"

```

(c) Markus Neumann

Variablenliste Biorythmus Daniel Rügemer

```

C4$ = CHR$(017)      Cursor down
RN$ = CHR$(018)      Reverse on
S1$ = CHR$(032)      Space
Z1$ = CHR$(045)      Grafikzeichen (-)
RF$ = CHR$(146)      Reverse off
PI = 3.14159265
M1$ = CHR$(098)      Grafikzeichen ( )
E2$ = CHR$(080)      Grafikzeichen (p)
S7$ = CHR$(071)      Grafikzeichen (g)
Q = 40
Q1$ = 40*CHR$(045)  40x      (-)
n$ = "an"
GT = Geburtstag
GM = Geburtsmonat
GJ = Geburtsjahr
Bm = Monat Biorythmus
BJ = Jahr Biorythmus
Y% = 29173
X = 31
"$ = "
GE = 23 Periode geistig
SL = 28 Periode seelisch
PH = 33 Periode physisch
PB = .2731
SB = .2243
GB = .1903
i = 11
en = (31)(28)(30) Tag
n1 = 365 Jahr
a = 13
f1 = 475
f2 = 475.25
G$ = " "
T = 32
bb = 29
cc = 17
t1 = 35
t2 = 32
t3 = 20
f$ = "n" Abfrage Schleife

```

**Die nächste
COMMODOREWELT
erscheint am
25. April 86**


```

10 rem Wuestenrallye =====20
20 rem (P) 1986 Commodore Welt ts =
30 rem =====
40 rem (C) by F. Schomborg =
50 rem Version 2.0 22Z/ASCII =
60 rem =====
70 rem VC 20 + 8K =
80 rem 1530/1541 =
90 rem =====

100 c4$=chr$(017):c1$=chr$(147)
110 rn$=chr$(018):rf$=chr$(146)
120 wh$=chr$(005):c3$=chr$(029)
130 c1$=chr$(157):re$=chr$(028)
140 y1$=chr$(117):y2$=chr$(105)
150 y3$=chr$(106):y4$=chr$(107)
160 y5$=chr$(110):y6$=chr$(109)
170 y7$=chr$(164):y8$=chr$(191)
180 y9$=chr$(166):y0$=chr$(168)
190 c2$=chr$(145):he$=chr$(019)
200 forq=1to2:q2$=q2$+c4$:nextq
210 forq=1to3:q3$=q3$+c4$:nextq
220 forq=1to4:q4$=q4$+c4$:nextq
230 forq=1to5:q5$=q5$+c4$:nextq
240 forq=1to6:q6$=q6$+c4$:nextq
250 forq=1to7:q7$=q7$+c4$:nextq
260 forq=1to8:q8$=q8$+c4$:nextq
270 forq=1to9:q9$=q9$+c4$:nextq
280 forq=1to6:q0$=q0$+c1$:nextq
290 forq=1to8:qa$=qa$+c1$:nextq
300 forq=1to3:x3$=x3$+c3$:nextq
310 forq=1to5:x5$=x5$+c3$:nextq
320 forq=1to7:x7$=x7$+c3$:nextq
330 forq=1to8:x8$=x8$+c3$:nextq
340 forq=1to9:x9$=x9$+c3$:nextq
350 forq=1to10:x0$=x0$+c3$:nextq
360 forq=1to6:xa$=xa$+y8$:nextq
370 forq=1to8:xb$=xb$+y8$:nextq
380 forq=1to4:xc$=xc$+y7$:nextq
390 forq=1to2:xd$=xd$+y7$:nextq
400 qf$=qf$+re$+c4$+y1$+y2$+c4$+c1$
+c1$+y3$+y4$
410 qg$=qg$+c4$+y5$+y6$+y7$+y5$+y6$
+y5$+y6$+xd$+c4$+y6$
420 qg$=qg$+xd$+y5$+c2$+y5$+y6$+xc$
+y5$+y6$
430 printcl$
440 printq9$"*** wuestenrallye ***"
:print:printc3$(c)1985 f.schomborg
"
450 forq=0to3000:next
460 printcl$:gosub2160
470 fori=0to505
480 poke37888+i,0
490 nexti
500 t1=36874:t2=36875:t3=36876:gr=3
6877:ls=36878
510 pokels,8:pokegr,128:poket1,129

520 rw=4436:gosub930
530 poke36879,28
540 a=4260:b=4261
550 pokea,78:pokeb,77
560 a=a+21:b=b+23
570 ifa>4470then590
580 goto550
590 forq=0to21
600 poke4228+q,100
610 nextq
620 printspc(15)qf$
630 printqg$
640 ti$="000000"
650 h=int(rnd(1)*20)
660 ifh<5thengosub730
670 ifh=9orh=10thengosub1490
680 ifh>14thengosub830
690 ifh=7orh=8thengosub1660
700 ifh=12orh=13thengosub1770
710 gosub1050
720 goto650
730 k=4260
740 pokek,88
750 forw=0to20:next
760 pokek,78
770 pokery,230:pokery+5,230:pokerr+
21,230:pokerr+24,230
780 k=k+21
790 gosub1080
800 ifk>4470then820
810 goto740
820 return
830 p=4261
840 pokep,88
850 forw=0to20:next
860 pokep,77
870 pokery,230:pokery+5,230:pokerr+
21,230:pokerr+24,230
880 p=p+23
890 gosub1080
900 ifp>4491then920
910 goto840
920 return
930 rr=rw:pokerr,85:pokerr+1,73
940 pokerr+21,102:pokerr+22,115:pok
err+23,107:pokerr+24,102
950 rt=rr+42
960 pokert,85:pokert+1,105:pokert+2
,113:pokert+3,113:pokert+4,95:poker
t+5,73
970 ry=rt+22
980 pokery,102:pokery+1,114:pokery+
2,64:pokery+3,64:pokery+4,114:poker
y+5,102
990 ru=ry+22
1000 pokeru,160:pokeru+1,105:pokeru
+2,127:pokeru+3,255:pokeru+4,95:pok
eru+5,160

```



```

1010 ri=ru+22
1020 pokeri,160:pokeri+1,223:pokeri
+2,215:pokeri+3,215:pokeri+4,233:po
keri+5,160
1030 rp=ri+22:pokerp,160:pokerp+5,1
60
1040 return
1050 t=int(rnd(1)*2)
1060 ift=0then1270
1070 ift=1then1380
1080 ifti$="000300"then1910
1090 ifpeek(197)=33then1130
1100 ifpeek(197)=30then1200
1110 pokery,102:pokery+5,102:pokerr
+21,102:pokerr+24,102:print"s"right
$(ti$,3)
1120 return
1130 gosub1560:rw=rw-1:gosub930:ifr
w=4428then1150
1140 return
1150 pokegr,0:poket1,0:forq=180to23
0:poket3,q:nextq:poket3,0
1160 poket1,129:pokegr,128
1170 gosub1200
1180 fori=0to2:poke4428+i*21,78:nex
ti
1190 return
1200 gosub1610:rw=rw+1:gosub930:ifr
w=4444then1220
1210 return
1220 pokegr,0:poket1,0:forq=180to23
0:poket3,q:nextq:poket3,0
1230 poket1,129:pokegr,128
1240 gosub1130
1250 fori=0to2:poke4445+i*23,77:nex
ti
1260 return
1270 s=int(rnd(1)*10)
1280 n=4250+s
1290 fory=0tos
1300 pokery,230:pokery+5,230:pokerr
+21,230:pokerr+24,230
1310 poken,46
1320 foru=0to30-3*s:nextu
1330 poken,32
1340 n=n+21
1350 gosub1080
1360 nexty
1370 return
1380 s=int(rnd(1)*10)
1390 n=4271-s
1400 fory=0tos
1410 pokery,230:pokery+5,230:pokerr
+21,230:pokerr+24,230
1420 poken,46
1430 foru=0to30-3*s:nextu
1440 poken,32
1450 n=n+23
1460 gosub1080
1470 nexty
1480 return
1490 pokeri+2,209:pokeri+3,209
1500 pokegr,160
1510 fori=15to5step-.1
1520 pokels,i:nexti
1530 pokegr,128:pokels,8
1540 pokeri+2,215:pokeri+3,215
1550 return
1560 pokerr+1,32:pokerr+24,32
1570 pokert+5,32:pokery+5,32
1580 pokeru+5,32:pokeri+5,32
1590 pokerp,32:pokerp+5,32
1600 return
1610 pokerr,32:pokerr+21,32
1620 pokert,32:pokery,32
1630 pokeru,32:pokeri,32
1640 pokerp,32:pokerp+5,32
1650 return
1660 s=int(rnd(1)*2)
1670 d=4282:i=0:e=d+i
1680 fora=dtoe:pokea,81:gosub1880:n
exta
1690 gosub1080
1700 fory=dtoe:pokey,32:nexty
1710 d=d+21:ifd>4345thend=d+1+s
1720 i=i+1:ifd>4345theni=i-1
1730 pokery,230:pokery+5,230:pokerr
+21,230:pokerr+24,230
1740 e=d+i
1750 ifd>4580thenreturn
1760 goto1680
1770 s=int(rnd(1)*2)
1780 d=4283:i=0:e=d+i
1790 fora=dtoe:pokea,81:gosub1880:n
exta
1800 gosub1080
1810 fory=dtoe:pokey,32:nexty
1820 d=d+22:ifd>4349thend=d-s
1830 i=i+1:ifd>4349theni=i-1
1840 pokery,230:pokery+5,230:pokerr
+21,230:pokerr+24,230
1850 e=d+i
1860 ifd>4592thenreturn
1870 goto1790
1880 ifpeek(a)=peek(rr+21)orpeek(a)
=peek(ry)then2040
1890 ifpeek(a)=peek(rr+24)orpeek(a)
=peek(ry+5)then2040
1900 return
1910 pokegr,0:poket1,0:pokels,12
1920 printhe$q6$x0$y0$y0$:gosub2030
1930 printhe$q6$x9$y9$y9$y9$y9$:gos
ub2030
1940 printhe$q6$x8$xa$c4$q0$xa$:gos
ub2030
1950 printhe$q6$x7$xb$c4$qa$y8$y8$"

```



```

ziel"y8$y8$c4$qa$xb$:gosub2030
1960 fori=1to3000:nexti
1970 poke36879,187:print"S"
1980 printq4$"gratuliere,sie haben
    "c4$"es geschafft !"
1990 printq3$"fuer neues spiel bitt
e"c4$x7$"<space>"
2000 printc4$"druecken !"
2010 geta$:ifa$<>" "then2010
2020 goto430
2030 poket2,200:fori=1to1000:nexti:
poket2,0:return
2040 wq$=ti$
2050 poket1,0:pokegr,150
2060 forl=15to0step-2:pokels,1:form
=1to300:nextm:nextl
2070 pokegr,0
2080 poke36879,42:print"S"
2090 printwh$q2$"sie sind kollidier
t !"
2100 printq3$"sie fuhren : "
2110 printc4$x3$mid$(wq$,4,1)c3$"mi
n."c3$right$(wq$,2)c3$"sek."
2120 printc4$"viel glueck naechstes
mal !"
2130 printq2$"fuer neues spiel.....
.c4$".....<space>....."c4$"dr
uecken !"
2140 getq$:ifq$<>" "then2140
2150 goto430
2160 printq5$"steuern sie ihren....
.c4$"rennwagen mit den....."c4$"ta
sten : "
2170 printc4$x5$"'z' und '/'"
2180 printc4$"und umfahren sie die
    "c4$"hindernisse !"
2190 forg=1to5000:next:printcl$
2200 return
2210 rem wuestenrallye          =====20
2220 rem 28079 bytes memory     =====
2230 rem 05127 bytes programm   =====
2240 rem 00301 bytes variables=====
2250 rem 00000 bytes arrays     =====
2260 rem 21793 bytes free       =====
2270 rem 21892 bytes free       =====
2280 rem =====

```

```

10 rem Mathekuestler =====20
20 rem (P) 1986 Commodore Welt ts =
30 rem =====
40 rem (C) by Thomas Splett      =
50 rem Version 2.0 22Z/ASCII     =
60 rem =====
70 rem VC 20 + 1530/1541         =
80 rem                           =
90 rem =====

100 c4$=chr$(017):cl$=chr$(147)
110 rn$=chr$(018):rf$=chr$(146)
120 wh$=chr$(005):c3$=chr$(029)
130 c2$=chr$(145):c1$=chr$(157)
140 printcl$:print"...mathekuenstl
er":print:print:ifol=0thendimp(50)
150 print"a=multiplizieren"
160 print"b=dividieren"
170 print"c=addieren"
180 print"d=subtrahieren"
190 print"e=potenzieren.....f=w
urzel.....g=fakultaet
(<33)"
200 print"h=durchschnitt.....i=p
rimzahlen.....j=geometrie.....
.....k=ende"
210 geta$:ifa$=""then210
220 ifasc(a$)<65orasc(a$)>75then210
230 printcl$:z1=1:onasc(a$)-64goto2
50,310,360,410,450,510,560,600,660,
720,960
240 rem multiplizieren
250 z=0:input" faktor";z:ifz=0anda=
0thenz1=0:goto280
260 ifz=0then280
270 a=1:z1=z1*z:goto250
280 print" produkt"z1
290 goto990
300 rem dividieren
310 input" dividend";v:input" divis
or ";d:input" kommastellen";k
320 b%=v/d:printb%:z=v-d*b%:fori=1t
ok
330 x%=z*10/d:printcl$x%::z=10*z-d*
x%:next
340 goto990
350 rem addieren
360 input" summand";x:ifx=0then380
370 x1=x1+x:x=0:goto360
380 print" summe "x1
390 goto990
400 rem subtrahieren
410 input" minuend...";g:input" sub
trahend";h
420 print" differenz "g-h
430 goto990
440 rem potenzieren
450 input" basis...";t:input" expon

```

☎ TELEFONSERVICE ☎

Alle Experten der COMMODORE-WELT stehen unseren Lesern jeden Mittwoch zwischen 16.30 und 20.00 Uhr zur Beantwortung aller Fragen zur Verfügung unter der Telefonnummer 089/1298013. Ebenso der Abo- und Kassettenservice. Einfach anrufen 089/129803/4!


```

10 rem Kaleidoskop =====20
20 rem (P) 1986 Commodore Welt ts =
30 rem -----
40 rem (C) by Jerzy Komorowski =
50 rem Version 2.0 22Z/ASCII =
60 rem -----
70 rem VC 20 =
80 rem 1530/1541 =
90 rem -----
100 c4$=chr$(017):cl$=chr$(147)
110 rn$=chr$(018):rf$=chr$(146)
120 wh$=chr$(005):c3$=chr$(029)
130 c2$=chr$(145):cl$=chr$(157)
140 re$=chr$(028):gr$=chr$(030)
150 bl$=chr$(031):sl$=chr$(032)
160 he$=chr$(019):pu$=chr$(156)
170 ye$=chr$(158):cy$=chr$(159)
180 goto200:printtab(5)re$
190 print"k"cy$a"pu"l"gr$e"bl$i
"ye$d"wh$o"re$s"cy"k"pu"o"bl$
p":return
200 printcl$:poke36879,8:gosub190
210 printspc(44)gr$"druecke"re$-f1
-, "gr$um bild zu stopen"
220 printc4$bl$"druecke"re$-f1-"bl
$"nochmal, um bild wieder zu starte
n"
230 printgr$"druecke"re$-f3-"gr$"u
m geraeusche auszuschalten"
240 printc4$bl$"druecke"re$-f3-"bl
$"wieder um"
250 printbl$"geraeusche einzuschalt
en"
260 printspc(66)"weiter mit-"rn$sp
ace"
270 pokel98,0:wait198,1:getq$:ifasc
(q$)<>32then270
280 r=rnd(1)
290 gf=1:kf=1:n=int(rnd(1)*5+1):cn=
int(rnd(1)*7+1)
300 ba=7680
310 s3=36876:l=36878:f=30720
320 deffnmod(a)=int((a/b-int(a/b))*
b+.05)*sgn(a/b)
330 printcl$
340 fori=0to1000000
350 forj=0to10
360 on n gosub540,570,600,630,660,6
90
370 a=ba+22*y+x:pokea,160:pokea+f,c
n
380 a=ba+22*(21-y)+x:pokea,160:poke
a+f,cn
390 a=ba+22*y+21-x:pokea,160:pokea+
f,cn
400 a=ba+22*(21-y)+21-x:pokea,160:p
okea+f,cn
410 a=ba+22*x+y:pokea,160:pokea+f,cn
420 a=ba+22*x+21-y:pokea,160:pokea+
f,cn
430 a=ba+22*(21-x)+y:pokea,160:poke
a+f,cn
440 a=ba+22*(21-x)+21-y:pokea,160:p
okea+f,cn
450 getq$:ifq$=chr$(134)thengf=1-gf
:ifgf=0thenpokel,0
460 ifgf=0then480
470 pokes3,128+(x+y)*2.8:pokel,15
480 ifq$=chr$(133)thenkf=1-kf
490 ifkf=0thenpokel,0:getq$:goto480
500 ifrnd(1)<.1thencn=int(rnd(1)*8)
510 ifrnd(1)<.07thenn=int(rnd(1)*6+
1)
520 ifrnd(1)<.065then gosub720
530 next:next:end
540 b=15:x=fnmod(abs(i-sgn(j-6)*(j+
2)))
550 b=21:y=fnmod(j*j+2*j+7)
560 return
570 b=18:x=fnmod(i*j)
580 b=12:y=fnmod(abs(abs(i-abs(2*j-
2*j))))
590 return
600 b=20:x=fnmod(i)
610 b=20:y=fnmod(j)
620 return
630 b=12:x=fnmod(abs(y-j))
640 b=20:y=fnmod(abs(2*j-abs(i-abs(
2*i-j))))+rnd(1)*3)
650 return
660 b=16:x=fnmod(abs(i-sgn(j-10)*j)
)
670 b=21:y=fnmod(i*j)
680 return
690 b=22:x=fnmod(abs(3*j-abs(2*i-ab
s(2*i-j))))
700 b=22:y=fnmod(abs(2*j-abs(2*x-ab
s(2*x-j))))
710 return
720 cb=int(rnd(1)*7)
730 poke36879,peek(36879)and248orcb
740 poke646,cb
750 printhe$spc(255)spc(229);
760 printrn$sl$sl$sl$sl$sl$sl$sl$sl
$sl$sl$sl$sl$sl$sl$sl$sl$sl$sl$sl
1$sl$;
770 pokeba+505,160:pokeba+31225,cb:
return
780 rem kaleidoskop =====20
790 rem 28079 bytes memory =====
800 rem 02505 bytes programm =====
810 rem 00112 bytes variables =====
820 rem 00000 bytes arrays =====
830 rem 00016 bytes strings =====
840 rem 25446 bytes free =====
850 rem =====

```



```

ent";f:ift=0orf=0thent=0:goto480
460 ift=1orf=1thent=t:goto480
470 f1=t:forf2=2tof:t=t*f1:next
480 print" ergebnis"t
490 goto990
500 rem wurzel
510 input" zahl.....";z:input" w
urzelanzahl";w:w=int(w)
520 er=z^(1/w)
530 print" ergebnis"er
540 goto990
550 rem fakultaet
560 input" zahl";s2:ifs2=0thenprint
" ergebnis"0:goto580
570 s1=2:for s3=1 to s2:s1=s1*s3:next:
print" ergebnis"s1/2
580 goto990
590 rem durchschnitt
600 a=0:input" zahl...";a:ifa=0then
620
610 s=s+a:n=n+1:goto600
620 print" durchschnitt";:ifs=0then
print" 0":goto640
630 prints/n
640 goto990
650 rem primzahl
660 input" zahl";z:ifz=1orz=0thenpr
int" keine primzahl":goto710
670 fori=2 to sqr(z)+2:k=z/i
680 ifk=int(k)andi<>zthenprint" kei
ne primzahl":goto710
690 next
700 print" primzahl"
710 goto990
720 cl$=chr$(147):printcl$:print"...
..mathekuenstler.....(geometrie
)"
730 print:print"a=hauptmenue"
740 print"b=inhalt des dreiecks c=i
nhalt des trapez...d=";
745 print"inhalt des parallelo
gramms"
750 print"e=umfang/inhalt des....kr
eises.....f=ende"
760 geta$:ifa$=""then760
770 ifasc(a$)<65orasc(a$)>70then760
780 printcl$:onasc(a$)-64goto1000,8
00,840,880,920,960
790 rem dreieck
800 input" grundseite";g:input" hoe
he.....";h
810 print" flaecheninhalt";g*h/2
820 goto970
830 rem trapez
840 input" grundseite";g:input" dec
kseite ";d:input" hoehe.....";h
850 print" flaecheninhalt"(g+d)/2*h
860 goto970

```

```

870 rem parallelogramm
880 input" grundseite";g:input" hoe
he.....";h
890 print" flaecheninhalt"g*h
900 goto970
910 rem kreis
920 input" radius";r
930 print" flaecheninhalt" *r^2
940 print" umfang"2* *r
950 goto970
960 end
970 geta$:ifa$=""then970
980 run720
990 geta$:ifa$=""then990
1000 run
1010 rem mathekuenstler =====20
1020 rem 28079 bytes memory =====
1030 rem 02926 bytes programm =====
1040 rem 00063 bytes variables=====
1050 rem 00262 bytes arrays =====
1060 rem 00009 bytes strings =====
1070 rem 24819 bytes free =====
1080 rem =====

```



Wir verschenken! CBM-64/C-128 VC-20

Die PEEK/POKE-MAGICS!
Jetzt gratis zu jeder Katalog-Bestellung.

Über 300 PEEKS und POKES-Tips und Tricks als kompaktes Nachschlagewerk mit den letzten Geheimnissen des Betriebssystems. Da sollten Sie zugreifen!

NEU! Katalog "Winter 85/86" (64 Seiten). Schnell anfordern od. abholen für 2,50 DM (Briefmarken).

Stichwort "Katalog + PEEK/POKE-MAGICS!"

Wir haben alles für COMMODORE 64, C-128, VC-20! Geräte, Hardware, Software, Zubehör und ... und ... über 500 Artikel zu Tiefpreisen!

mükra
DATEN-TECHNIK

Schöneberger Str. 5
(Am Berlinicke Platz)
1000 Berlin 42/RE
☎ 030-752 91 50/60

Öffnungszeiten:
Mo-Fr: 10-18 Uhr
Sa.: 10-13 Uhr

Händleranfragen erwünscht. · 24 Std. Schnellversand.


```

10 rem Cassettenetikett =====20
20 rem (P) 1986 Commodore Welt ts =
30 rem =====
40 rem (C) by Joachim Riedel =
50 rem Version 2.0 22Z/ASCII =
60 rem =====
70 rem VC 20 + MPS 801 =
80 rem 1530/1541 =
90 rem =====

100 rem =====
110 c4$=chr$(017):rn$=chr$(018)
120 re$=chr$(028):c3$=chr$(029)
130 bl$=chr$(031):s1$=chr$(032)
140 zc$=chr$(061):z1$=chr$(096)
150 z2$=chr$(125):rf$=chr$(146)
160 cl$=chr$(147):pu$=chr$(156)
170 z4$=chr$(171):z5$=chr$(173)
180 z6$=chr$(174):z7$=chr$(176)
190 za$=chr$(179):zb$=chr$(189)
200 forq=1to22:q1$=q1$+z1$:nextq:fo
rq=1to22:q2$=q2$+s1$:nextq
210 forq=1to9:q3$=q3$+c4$:next
220 forq=1to3:q4$=q4$+s1$:nextq:for
q=1to3:q5$=q5$+c4$:nextq
230 forq=1to39:q6$=q6$+z1$:next
240 forq=1to39:q7$=q7$+zc$:next
250 poke36879,28
260 printcl$;
270 printq3$;bl$;rn$;q1$;q2$;
280 printre$;rn$;q4$"cassettenetike
tt";q4$;
290 printbl$;rn$;q2$;q1$
300 fort=1to3000:next
310 dimt$(20):dimi$(20)
320 printcl$bk$
330 input"cassettennr.":nr$
340 printc4$"rueckenaufschrift"
350 printc4$
360 inputr$
370 poke36879,26
380 fort=1to20
390 printcl$;bk$"ausdruck-nr.":rn$;
t
400 printc4$pu$;q1$;
410 printrn$"ö = eingabe wiederh.
";
420 printrn$q2$;
430 printrn$" * = eingabe beenden
";
440 printrf$q1$
450 printbk$q5$"titel:"
460 printc4$bl$::inputt$(t)
470 ift$(t)="$"thennextt
480 ift$(t)="*"thent$(t)=" ":goto53
0
490 ift$(t)="ö"thent=t-2:goto520
500 printq5$bk$"interpret:"

510 printc4$re$::inputi$(t)
520 nextt
530 open4,4
540 print#4,z7$;q6$;z6$
550 fort=1to15
560 gosub720
570 ift$(t)="$"thenprint#4,z2$;q7$;
z2$:nextt
580 gosub730
590 nextt
600 print#4,z4$;q6$;za$
610 r1$=left$(r$,29):r2$=mid$(r$,30
,29)
620 print#4,z2$chr$(14)nr$chr$(15)c
hr$(16)"10"r1$chr$(16)"40";z2$
630 print#4,z2$chr$(16)"10";r2$chr$
(16)"40";z2$
640 print#4,z4$;q6$;za$
650 fort=16to20
660 gosub720
670 ift$(t)="$"thenprint#4,z2$;q7$;
z2$:nextt
680 gosub730
690 nextt
700 print#4,z5$;q6$;zb$
710 close4:end
720 t$(t)=left$(t$(t),20):i$(t)=lef
t$(i$(t),16):return
730 print#4,z2$;t$(t);s1$:s1$;chr$(
16)"23";i$(t)chr$(16)"40";z2$:retur
n
740 rem cassettenetikett =====20
750 rem 28079 bytes memory =====
760 rem 02046 bytes programm =====
770 rem 00189 bytes variables =====
780 rem 00000 bytes arrays =====
790 rem 02141 bytes strings =====
800 rem 23703 bytes fre =====
810 rem =====

```




```

10 rem Donald Duck =====20/64
20 rem (P) 1986 Commodore Welt ts =
30 rem =====
40 rem (C) by Joachim Brosig =
50 rem Version 2.0 22/40Z ASCII =
60 rem =====
70 rem VC 20 + C 64 + Drucker =
80 rem 1530/1541 =
90 rem =====

```

```

100 open4,6,0
110 rem*****
120 rem*zeile 100 muss*
130 rem*bei einem druk*
140 rem*ker 'open4,4'*
150 rem***  lauten ***
160 rem*****
170 open3,6,3:print#3,0
180 rem nur bei plotter
190 rem vc-1520 die zei
200 rem le 170 eintippe
210 print#4,".....
.....xxxxxx"
220 print#4,".....
.....xx.....xx"
230 print#4,".....
.....x.....x"
240 print#4,".....xxxx
.....x.....x"
250 print#4,".....xx....
xx....x.....x"
260 print#4,".....x.....
..xx.x.....x"
270 print#4,".....x.....
...xx.....x.....x.xx"
280 print#4,".....x.....
..x.....xx.....xxxx"
290 print#4,".....x.....
xx.....x.....xxxxx.x"
300 print#4,".....x.....
.....xxxxxxxxxxxxxx"
310 print#4,".....
.....xxxxxx.x.....xx....xxx"
320 print#4,".....x.xx....
.....xxx.x.....x.xxx.xxxx.x"
330 print#4,".....x....xx...
....xx....x.xxxx.xx....x.xxx.x"
340 print#4,".....x..x..x.xxx
xxxx....x.x....xx....xx..x.x"
350 print#4,".....x..x..x...x
.....xx....x.....xx....xx"
360 print#4,".....x.xx..x...x
.....x.x....x.....x....xx"
370 print#4,".....x.xx..x...x
.....xx....x.....x....x"
380 print#4,".....x.x.x..x...x
.....x....x.....x....x"
390 print#4,".....x.x.x.x...x

```

```

.....x.....x.....x.....x"
400 print#4,".....x.x..x.x.....x
.....x.....x.....x.....x"
410 print#4,".....xx...x.x.....x
.....x.....x.....x.....x"
420 print#4,".....xx...xx.....
x.....x.....x.....x.....x"
430 print#4,".....x....xx.....
x.....x.....x.....x.....x"
440 print#4,".....x.....
x.....x.....x.....x.xxx.x"
450 print#4,".....
.x.....x...xx.x.....xxxxxxxx"
460 print#4,".....
.x..x....x..xxxxx.....xxxxxxxx"
470 print#4,".....
..xx.x...x.xxxxxx.....x.xxxx"
480 print#4,".....
..xxx.x..x...xxx....xxxx.xxxxxx"
490 print#4,".....
...xx.x..x.xxxxxx..xx....xxxxxxxx"
500 print#4,".....
...x.x.x.x..xxxxxxx..xx....xxx.x"
510 print#4,".....
...x.xx.xxxxx..x....xxx"
520 print#4,".....
...xx.xx.....x"
530 print#4,".....
...x.x..x.....x"
540 print#4,".....
...x.x..x.....x"
550 print#4,".....
...x.x..x.....xxxxxxxxxxx
xxx"
560 print#4,".....
...x.x...x.....xx....xxxxxx
xx.x"
570 print#4,".....
...x.x....x.....xx.....
.x.x"
580 print#4,".....
.....x.x....x.....xxx....
.x.x"
590 print#4,".....
.....x.x....x.....xx...x....
x..x"
600 print#4,".....
.....x.x.....xxxxxxx.x....xx..xx
..xx"
610 print#4,".....
.....x.x.....xx.....xxxx
x"
620 print#4,".....
.....x.x.....xx"
630 print#4,".....
.....x.x.....xx"
640 print#4,".....
.....xx.....xx"

```



```

650 print#4,".....
.....x.x.....xx"
660 print#4,".....
.....x.xxxxxxxxxxx.x"
670 print#4,".....
.....x.....x"
680 print#4,".....
.....xxxxxxxxxxxxx"
690 print#4:print#4:print#4:print#4
,"(c) 1985 by joachim brosig"
700 close3:close4
710 rem den 'close3' -
720 rem befehl nur bei
730 rem   vc 1520
740 rem   eingeben
750 rem donald duck   =====20
760 rem 28079 bytes memory   =====
770 rem 03917 bytes programm   =====
780 rem 00000 bytes variables   =====
790 rem 00000 bytes arrays   =====
800 rem 00000 bytes strings   =====
810 rem 24162 bytes free   =====
820 rem =====

```

MatheProgramm

-1-

```

10 rem Matheprogramm 2.0 =====cbm
20 rem (P) 09/85 CBM REVUE TEAM   =
30 rem =====
40 rem (C) 05/85 by Lars Zimmer   =
50 rem   =
60 rem   =
70 rem   =
80 rem VC-20 + 1530/1541   =
90 rem =====
100 c4$=chr$(17):cl$=chr$(147)
110 he$=chr$(19)
120 for a=1 to 22:he$=he$+c4$:nexta
130 rem ===== hauptmenue =
140 print cl$ "*** matheprogramm ***
*"
150 print left$(he$,5)"bitte waehle
n sie:"left$(he$,6)
160 print c4$"grundrechenarten...1"
170 print c4$"logarithmentafel...2"
180 print c4$"programmende.....3"
190 get a$
200 if a$="1" then x=1:gosub 260
210 if a$="2" then x=1:gosub 610
220 if a$="3" then end
230 if x=1 then x=0:goto 140

```

```

240 goto 190
250 rem = untermenue grundrechnen =
260 print cl$ "*** grundrechnen ***
*"
270 print left$(he$,5)"bitte waehle
n sie:"left$(he$,6)
280 print c4$"addition.....1"
290 print c4$"subtraktion.....2"
300 print c4$"multiplikation...3"
310 print c4$"division.....4"
320 get a$
330 on val(a$)gosub 370,430,490,550
340 if y=1 then y=0:return
350 goto 320
360 rem ===== addition =
370 y=1
380 gosub 960
390 er=z1+z2
400 gosub 1150
410 return
420 rem ===== subtraktion =
430 y=1
440 gosub 960
450 er=z1-z2
460 gosub 1150
470 return
480 rem ===== multiplikation =
490 y=1
500 gosub 960
510 er=z1*z2
520 gosub 1150
530 return
540 rem ===== division =
550 y=1
560 gosub 960
570 er=z1/z2
580 gosub 1150
590 return
600 rem == untermenue log.-tafel =
610 print cl$ "** logarithmentafel
**"
620 print left$(he$,5)"bitte waehle
n sie:"left$(he$,6)
630 print c4$"sinus.....1"
640 print c4$"cosinus.....2"
650 print c4$"tangens.....3"
660 print c4$"wurzel.....4"
670 get a$
680 on val(a$)gosub 720,780,840,900
690 if y=1 then y=0:return
700 goto 670
710 rem ===== sinus =
720 y=1
730 gosub 1070
740 er=sin(z1)
750 gosub 1150
760 return
770 rem ===== cosinus =

```


EIN MEGABYTE- SPEICHER IM C64!

Endlich ist es dem CW-Team gelungen, in nächtelanger, schweißtreibender Arbeit aus dem C64 das letzte herauszuholen. Im Commodore ist noch Speicherplatz für Basicprogramme nutzbar, von dem die meisten keine Ahnung haben. Das folgende Programm simuliert einen Atari 520 ST+ mit 1MByte (sprich Megabit) Speicher. Nun

```

780 y=1
790 gosub 1070
800 er=cos(z1)
810 gosub 1150
820 return
830 rem ===== tangens =
840 y=1
850 gosub 1070
860 er=tan(z1)
870 gosub 1150
880 return
890 rem ===== wurzel =
900 y=1
910 gosub 1070
920 er=sqr(abs(z1))*sgn(z1)
930 gosub 1150
940 return
950 rem ===== eingabe 2 zahlen =
960 print cl$
970 open 1,0
980 print left$(he$,2)" 1.zahl : ";
990 input# 1,z1$
1000 z1=val(z1$)
1010 print :print chr$(13)do$ " 2.za
hl : ";
1020 input# 1,z2$
1030 z2=val(z2$)
1040 print :close 1
1050 return
1060 rem ===== eingabe 1 zahl =
1070 print cl$
1080 open 1,0
1090 print left$(he$,2)" zahl : ";
1100 input# 1,z1$
1110 z1=val(z1$)
1120 print :close 1
1130 return
1140 rem ===== ausgabe =
1150 print do$
1160 for a=1 to 22
1170 print chr$(163);
1180 next a:print
1190 print tab(7)er
1200 get a$
1210 if a$="" goto 1200
1220 return
1230 rem Matheprg. v2.0 =====cbm
1240 rem 28079 Bytes Memory =====
1250 rem 02563 Bytes Program =====
1260 rem 00035 Bytes Variables =====
1270 rem 00000 Bytes Arrays =====
1280 rem 00278 Bytes Strings =====
1290 rem 25203 Bytes Free =====
1300 rem =====

```

```

10 rem == atarisimulator =====
15 rem (p) cbm revue team =
20 rem =====
25 rem (c) r. schmid-fabian =
30 rem c-64 =
35 rem =====
40 rem * datazeilenkontrolle *****
45 for i=1 to 254:read a:s=s+a:next:i
if s<>27699 then print"fehler":stop
50 rem *****
55 rem * simulatormaschinenprogram **
60 rem *****
62 restore
65 for i=0 to 173:read a:poke51200+i,
a:next
70 data120,160,0,132,3,169,208,133,4,
162,16,169,51,133,1,177,3
75 data41,60,72,234,169,48,133,1,104,
145,3,200,208,236,230,4,202
80 data208,231,32,145,200,169,55,133,
1,88,96,234,120,162,199,160,76
85 data173,14,220,41,254,9,144,141,14,
220,142,4,220,140,5,220,173
90 data14,220,9,1,168,173,18,208,234,
234,16,249,173,17,208,16,244
95 data140,14,220,169,192,141,136,2,1
73,0,221,41,252,141,0,221,173
100 data24,208,41,15,141,24,208,173,1,
221,9,3,141,1,221,88,96
105 data0,169,6,141,32,208,234,169,11
5,160,228,32,30,171,169,92,160
110 data228,32,30,171,32,224,228,96,0,
169,34,141,104,208,141,107,208
115 data141,108,208,141,109,208,141,1
10,208,169,54,141,105,208,169,42,141

```



```

92:sys828                                     uf ram umschalten
180 fori=861to866:pokei,0:nexti             190 poke64982,53:rem ram ist restore-
185 poke862,224:sys828:poke1,53:rem a stabilisiert
120 data106,208,96,0                         195 rem *****
125 printchr$(147):sys51200:rem bilds 200 rem *veraenderungen im interpr. *
chirm nach c000                             205 rem * und im betriebssystem ***
130 rem*****                               210 rem *****
135 rem*interpreter+betr.sys ins ram*       215 for i=41848 to 41862:reada:pokei,
140 rem*****                               a:next:poke58460,50:poke58462,50
145 fori=828to860:reada:pokei,a:nexti       220 data5,65,80,82,73,76,32,65,80,82,
150 data 173,93,3,174,94,3,133,95,134 73,76,154,13,0
,96                                           225 for i=58494 to 58525:reada:pokei,
155 data 173,95,3,174,96,3,133,90,134 a:next:poke58461,94:poke58463,48
,91                                           230 data32,65,84,65,82,73,45,83,84,45
160 data 173,97,3,174,98,3,133,88,134 ,83,73,77,85,76,65,84,79,82,32,42,42,
,89                                           42,42
165 data 76,191,163                         235 data13,13,49,77,66,89,84,69
170 fori=861to866:pokei,0:nexti             240 rem * start des simulators *****
175 poke862,160:poke864,192:poke866,1 245 sys51246:sys51320

```

steht im Handbuch und in der ganzen C64-Literatur, daß nur 2^{16} (sprich 2 hoch 16) = 65536 Adressen ansprechbar sind. Nach etlichen Litern guten bayrischen Bieres kam jedoch einem unserer Teammitglieder sensationelle Einfall:

Wir nutzen den virtuellen Speicher des C64!

Durch Multiplexen von 8 Speicherbausteinen werden diese so miteinander vernetzt, daß sich daraus das 16fache Speichervermögen ergibt. Da die Daten im Computer durch Be- und Entladen von winzigen Kondensatoren abgespeichert sind, konnte das nichtlineare Ver-

halten dieser Speicherbausteine dazu genutzt werden, 3 weitere Energieniveaus zu nutzen. Ein genauer Zeitabgleich beim Multiplexen im Gigahertzbereich ist dazu nötig. Die Daten müssen bei der Weitergabe über den Bus so schnell hintereinander eingelesen werden, daß beständig 4 Bits gleichzeitig auf einer Leitung transportiert werden.

Tippen Sie bitte das Programm ein und speichern es auf einen Datenträger ab. Dann schalten Sie den Computer erst einmal aus und lassen ihn abkühlen. Entfernen Sie (im abgeschalteten Zustand) alle Erweiterungen (Module ...) (Halt! Die Floppy lassen sie dran, die brauchen Sie ja!) und laden nach dem

Einschalten das Programm und starten es.

Nach 4 Sekunden erscheint die Einschaltmeldung des Atarisimulators und Sie haben nun einen neuen Zeichensatz (dieser mußte etwas schmaler ausfallen, um bei der Darstellung auf dem Bildschirm Zeit zu sparen) und 2^{20} (sprich 2 hoch 20) = 1048576 Byte zur Verfügung. Bei manchen Computern dauert es etwas, bis der Cursor wieder erscheint. Um dies zu beschleunigen, können Sie auf die Commodoretaste links unten drücken. Wie Sie den zusätzlichen Speicher nutzen können, erfahren Sie in der nächsten Commodore-Welt. (rsf).

COMMODORE-WELT

MSX REVUE

suchen kompetente

COMPUTER-JOURNALISTEN.

Die künftigen Kollegen sollten nicht nur ihr Handwerk beherrschen, sondern auch die Rechner, die dazu gehören. Außerdem erwarten wir ein ausgeprägtes Interesse für Themen rund um diese Computer-Typen. Ihre Bewerbung mit den üblichen Unterlagen bitte an
**AKTUELL-GRUPPE, Postfach 1107,
8044 Lohhof, z.Hd. H. Seibt.**

MERGE C 128

Durch folgende Befehle, im Direktmodus eingegeben, können Sie zwei Basic-Programme

MERGEN, allerdings muß das anzuhängende Programm höhere Zeilennummern als das erste besitzen.

1. Erstes Programm laden
2. PRINT PEEK(45), PEEK(46) <RETURN>
3. Bitte notieren Sie die beiden Zahlen. (Im Normalfall sind das die '1' und '28')
4. POKE45, (PEEK(4624) +256*PEEK(4625)-2)

AND 255

5. POKE46, (PEEK(4624) +256*PEEK(4625)-2)/256
 6. Zweites Programm nachladen.
 7. POKE 45, (Zahl 1): POKE 46 (Zahl 2)
- Nach LIST werden Sie feststellen, daß die beiden BASIC-Programme gemERGET sind.

Harald Beiler

ZEICHENFARBE BEIM C 128:

POKE 241, (Farbnummer 0 - 15)

	C64+VC20 POKE 43/44/45/46,x		C64+VC20 PEEK (61)/(62)
	Basicanfang und -ende Diese Speicherstellen geben Anfang und Ende des zur Zeit im Speicher befindlichen Basicprogramms an. Basic-Anfang berechnen: PRINT PEEK (43)+256*PEEK(44) Basic-Ende berechnen: PRINT PEEK(45)+256*PEEK(46)		Zeiger auf Basic-Zeile f. CONT Mit PeEEK (62)+256*PEEK(62) erhält man die Adresse, die nach der zuletzt ausgeführten Zeile kommt. Hier holt sich der Interpreter bei CONT den nächsten Befehl.
	C64+VC20 POKE 657,128		POKE53281/646
	Blockiert SHIFT/C= (Umschaltung Groß/Klein) andere Möglichkeiten: a) PRINT CHR\$(14):CHR\$(8) Schaltet auf Kleinbuchstaben und blockiert) b) Drücken von CTRL/H		Setzen von Bildschirm und Rahmenfarbe: POKE 53280, RF (Rahmenfarbe) POKE 53281, HF (Hintergrund) Setzen der Cursorfarbe: POKE 646,CF
	C64+VC20 POKE 19,64		C64+VC20 POKE 46325,x
	INPUT ohne Fragezeichen: wird eine Input-Anweisung ausgeführt, so erscheint normalerweise ein ? am Ende des Textes. Z.B.: EINGABE:? Will man dies nicht, so kann man das ? durch POKE 19,64 verhindern. Nachteil: Der Cursor springt nach Return nicht mehr in die nächste Zeile.		Zeitlupe: x=1: Cursor wird schnell Listen wird langsam CRL/HOME langsam (von unten nach oben) x=0: Cursor und Listen und RUN langsam Zeitlupe für VC20: POKE 37877,0 normal mit: RUN/STOP-RESTORE
	C64XVC20 SYS 42291		C64+VC20 SYS 58460
	Basiczeilen neu binden Die Linkadressen werden neu berechnet. Die Linkadresse der ersten Basiczeile in 2049 und 2050 (LO- und HI-Byte) zeigt auf die Linkadresse der zweiten Basiczeile. (a=pE (2049) = 256*pE(2040) bei A beginnt die Linkadresse der zweiten Zeile (VC:20SYS 50483)		Cursor positionieren POKE 214,z : POKE 211,s : SYS 58460 Setzt den Curosr in die z-te Zeile und die s-te Spalte. Für VC20: POKE 781, z: POKE 782,s : SYS 58636

Merge: (2 Programme verbinden) <pre>POKE43, (PEEK(45)+256*PEEK(46) -2)AND255 (RETURN) POKE44, (PEEK(45)+256*PEEK(46) -0/256 (RETURN) zweites Programm laden POKE43,1:POKE44,8 (RETURN)</pre>	Abschneiden des hinteren Teils eines Programms, Beispiel: 1000 STOP (einfügen) RUN 1000 (nach 1000 soll alles abgeschnitten werden.) AD=PEEK (61)+256*PEEK(62)-3 HB=INT(AD/256):LB=AD-256*HB POKEAD-1,0:POKEAD-2,0 PRINT LB:POKE46,HB (merken10 poke45,LB:POKE46,HB (LB und HB müssen eingesetzt werden!))
Noch eine Möglichkeit beim LIST: <pre>0 REM:: POKE2054,14:POKE2055,8 Macht nach Eingabe von LIST das gleiche wie PRINT CHR\$(14):CHR\$(8)</pre>	Beispiel: POKE 53280,0 (Rahmen schwarz) POKE 53281,0 (Hintergr. schwarz) POKE 646,2 (Cursor rot)
Diesen Nachteil kann man beheben, indem man nach der INPUT-Anweisung die Speicherstelle 19 wieder auf 0 setzt: <pre>10 POKE 19,64:INPUT "EINGABE:" 20 POKE 19,0</pre>	<pre>0 REM REAKTIONSSPIEL 1 TIS\$="000000":PRINT CHR\$(147) 2 X%=RND(0)*1000+1024: 3 POKE X%,43:POKE56325,1 4 INPUT A\$:IF A\$=" "GOTO4 5 IF PEEK (X%)=42 THEN PRINT TIS\$:END 6 GOTO 4 + MUSS DURCH * ÜBERSCHRIEBEN WERDEN</pre>
RENEW nach versehntl. NEW oder RESET: POKE2050,1:SYS42291 POKE45,a:POKE46,b:LIST Hierbei ist $a+256*b-2048$ die Länge des Basicprogramms. Sind die Werte nicht bekannt, setzt man z.B. $a=1$ und $b=100$. Um die richtigen Zahlen für 45/46 zu finden siehe PEEK(61) und(62). (VC20:SYS50483, statt 2050: Einschaltwert von $pE(44)*256+2$)	1. Beispiel: 110 INPUT "ZEILE,SPALTE";Z,S 120 GOSUB 1000:PRINT "TEXT":END 1000 POKE211,S:POKE214,Z:SYS58640 1010 RETURN 2. Beispiel: 0 REM ZUFALLSGRAPHIK (STERNEN-HIMMEL) 1 PRINT CHR\$(147) 2 S=RND(1)*40:Z=RND(1)*25:F=RND(1) 3 POKE211,S:POKE214,Z:SYS58460:POKE 646,F 4 PRINT "":GOTO2

COM- PUTER TEST- JAHR- BUCH

**RUND 150
SEITEN
EINKAUFS-
BERATER**

**Die wichtigsten
Computer**

**Die beste
Software**

**Die schnellsten
Drucker**

**ALLES
ÜBER
COMPUTER
& ZUBEHÖR**

**WAS SIE
BEIM COM-
PUTERKAUF
BEACHTEN
MÜSSEN!**

**JETZT
AN
IHREM
KIOSK**

SO VERHINDERN SIE DATEN- SCHNÜFFELEI

In der letzten **COMMODORE WELT** berichteten wir über verschiedene Möglichkeiten, Daten und Programme vor Unbefugten zu schützen. Wir setzen diese Serie hier fort.

Man kann mit den „Gemeinheiten“ noch weitergeben. Ein künstliches Basicende kann man auch an dem Ende von REM-Zeilen erzeugen. Hierzu dient folgendes kleine Programm (für C64 + VC20):

```
2 goto4
3 rem test φφ
3 rem test ##
4 print "test":end
63000 rem* nullen in remzeilen***
63001 for i=256*peek(44)+5to256
*peek(46)
63002 ifpeek(i)=0theni=i+5
63004 ifpeek(i)=35andpeek(i+1)=
35 then
pokei, 0:pokei+1,0:i=i+1
63005:next:end
```

Mit RUN63000 starten. Listet man das Programm anschließend, so sind die beiden # verschwunden, das Programm sieht ganz normal aus und verhält sich bei RUN wie erwartet, es druckt TEST. Versuchen Sie aber mal das END in Zeile 4 zu löschen und listen Sie dann das Programm. Es steht dann nur noch da:

```
2 goto4
3 rem test
```

Sie können nun die Zeilen 63000–63005 an ein anderes Programm anhängen. Wenn Sie dann an das Ende mehrerer REM-Zeilen 2 mal # schreiben, werden diese durch Nullen ersetzt. Beim Listen fällt dies nicht auf. Will man aber eine Zeile verändern, löschen oder das Programm abspeichern und wieder laden, so wird wieder die Binde-routine aufgerufen und der Rest vom Programm ist weg.

Zwei Dinge muß man bei dieser Methode unbedingt beachten:

- a) Diese REM-Zeilen dürfen vom Programm nicht durchlaufen werden, da es sonst hier abbricht.

Am besten eignen sich REM-Zeilen vor Unterprogrammen, wobei der Einsprung mit GOSUB hinter der REM-Zeile erfolgt.

- b) Man muß bei einer kleineren Zeilennummer bereits ein künstliches Basicende durch Nullsetzen der Linkpointer erzeugt haben (wie oben beschrieben).

Der Trick mit dem Basicanfang:

Hierbei setzt ein kleines Maschinenprogramm den Basicanfang auf den eigentlichen Beginn des Basicprogramms hoch und startet es. Dies soll hier an einem Beispiel gezeigt werden:

Bitte folgende Zeile eingeben:

```
10 sys2078
```

Dann den Basicanfang hochsetzen (HI-Byte genügt):

```
poke44,10:pole10*256,0:new
(RETURN)
```

Das folgende Programm (für C64) eingeben, starten und dann mit NEW wieder löschen.

```
10 for i=2078to2101:read a:pokei,
a:next
20 data169,10,133,44 :rem poke
44,10
30 data169,82,141,119,2 :rem
poke631,82
40 data169,213,141,120,2:rem
poke632,213
50 data169,13,141,121,2 : rem
poke633,13
60 data169,3,133,198,96: rem
poke198,3
```

Es erzeugt ab der Adresse 2078 ein kleines Maschinenprogramm. Hinter REM steht jeweils die entsprechende Basicanweisung, damit es leicht auf andere Rechner übertragen werden kann.

Nun ein kleines Programm ein-

geben oder laden. Die letzte Anweisung, die ausgeführt wird, muß lauten:

```
...:poke44,8: end
```

Setzt man jetzt den Basicanfang wieder mit POKE44,8 zurück, so steht dort nur noch der SYS-Befehl. Das Programm kann nun so abgespeichert werden. Mit RUN wird das Maschinenprogramm aufgerufen, setzt den Basicanfang hoch und schreibt die Abkürzung für RUN in den Tastaturpuffer.

Das ist natürlich leicht zu knacken (wenn man's weiß), indem vor dem Start der Basicanfang hochgesetzt wird. (Speicherstellen 43 und 44).

Die entsprechenden Adressen findet man, indem man mit dem MINIMONITOR nach der ersten 0 sucht:

```
10 a=2048:for i=a to a+1000:print
i,peek(i),chr$(peek(i)),chr$(154):
sys58592:next
```

Hiermit kann man sich die Speicherinhalte genau ansehen.

SYS58592 wartet auf die C-Taste. Beim VC20 setzt man statt 2048: „peek(44)*256“ und statt sys58592 kann man „poke198,0: wait198,1“ schreiben.

Der Trick mit dem Basicende:

Hiermit kann man zwar keinen direkten Listschutz machen, es kann aber hilfreich sein, um Änderungen des Programms zu erschweren. Setzt man da Basicende hoch, z.B. mit:

```
POKE 46, PEEK (46+4)
```

so wird ein Kilobyte zusätzlich mit abgespeichert. Hier kann man z.B. Daten, die das Programm braucht oder ein Maschinenprogramm unterbringen. Wird nun im Programm etwas geändert, so verschieben sich die Speicherinhalte entsprechend der Änderung und das Programm läuft nicht mehr.

Nun komme ich zu den Methoden b), die immer wirksam und daher (unseres Erachtens) besser wird.

Wir wollen das Listen verhindern, also schauen wir mal nach, was dem C64 Probleme bereitet. Was er uns auf den Bildschirm zaubert, ist im Speicher in Form von Zahlenwerten

zwischen 0 und 255 abgelegt. Er kennt zwei Modi, mit und ohne Gänsefüßchen (sprich Anführungszeichen). Sehen wir uns das doch einmal näher an: Ohne Anführungszeichen werden die Codezahlen von 32 bis 128 als CHR\$-Codes (siehe Handbuch S. 135) behandelt. Ab 128 werden sie als TOKEN dargestellt, mit einer Ausnahme und zwar dem Code 204.

Machen Sie mal folgenden Test:
1:.....
.....
(eine 1 und 78 Doppelpunkte.)

Jetzt geben Sie ein:

```
I=2054:FOR K=128 TO 105:POKE
I, K:I=I+1: NEXT:LIST
```

Jetzt haben Sie die ganzen TOKEN aufgelistet und am Ende einen Syntax-Error stehen. Der Code 204 entspricht dem geschifteten L.

Dies gibt uns die einfachste und zugleich häufigste Methode. Wird eine Zeile mit REM SHIFT/L erreicht, so bricht der Computer mit Syntax Error ab. Die darauffolgende Zeile kann aber immer noch gelistet werden. POKEn wir jedoch den Wert 255 in die Speicherstellen 2051 und 2052, so erhalten wir die maximale Zeilennummer (255+256*255) und die nachfolgenden Zeilen können nicht mehr gelistet werden. Außerdem lassen sich Zeilen mit Nummern größer 63999 nicht löschen. (Aber nicht vergessen, sie im Programm auf den alten Wert zurückzusetzen.)

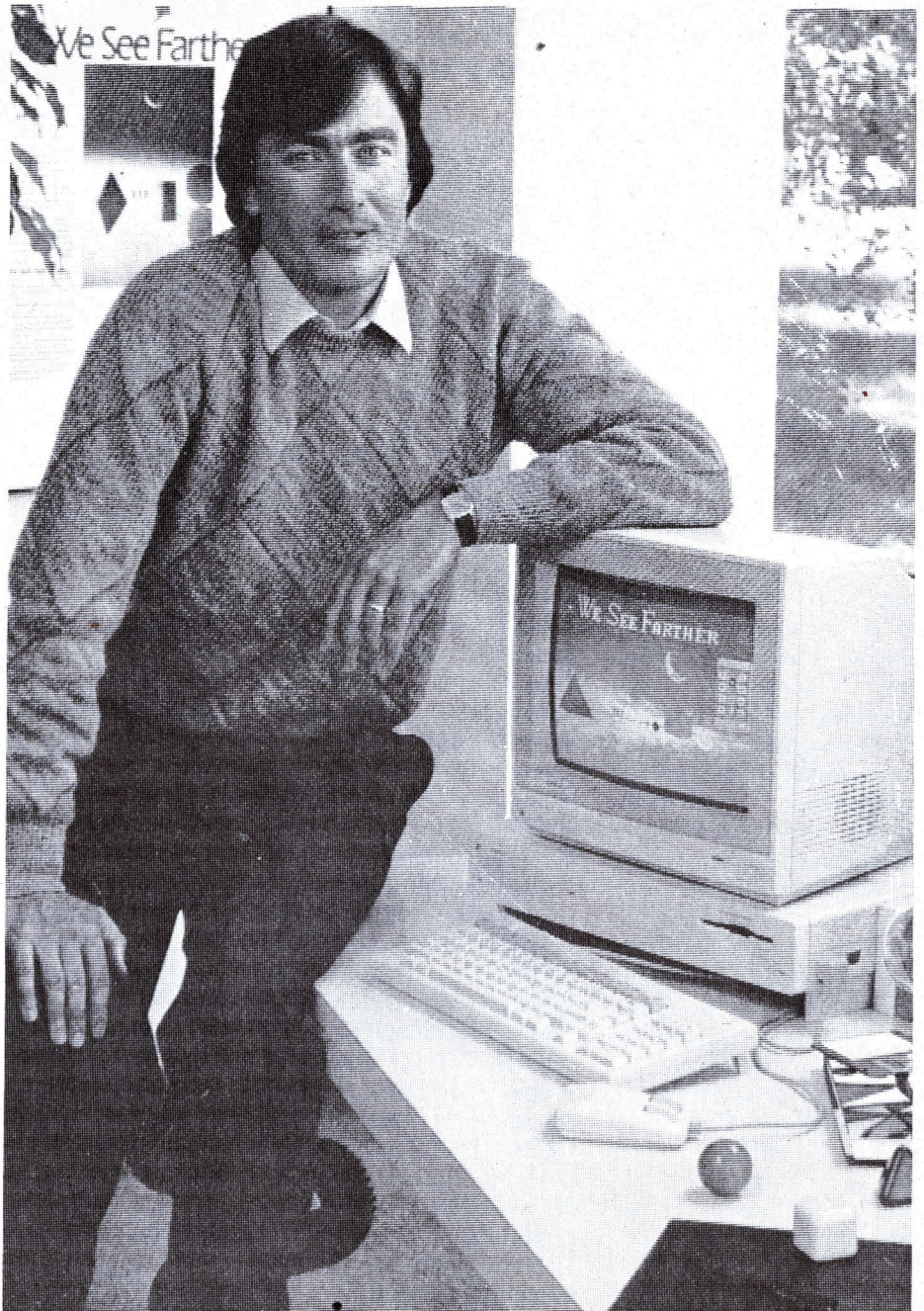
Durch POKE 2130,58 wird das geschiftete L durch einen „:“ ersetzt, und wir sehen, daß es ab dem Code 205 wieder mit FOR...NEXT weitergeht. Diese Codes erzeugen zwar beim Listen entsprechende TOKEN, welche aber beim RUN mit SYNTAX ERROR bestraft werden.

POKEt man jetzt in 2053 den Code für das Anführungszeichen, also POKE 2053,34, so passieren beim Listen seltsame Dinge. Jetzt wird es nämlich interessant. Geben Sie zunächst noch POKE 2067,58: LIST ein. Sie erhalten Graphikzeichen und einen Doppelpunkt (wegen der 58). Der Übeltäter ist der Code 141, SHIFT/RETURN. Die anderen Steuerzeichen wirken nicht beim Listen. Normalerweise! Sie haben aber ja gemerkt, daß sich

seltsame Dinge auf dem Bildschirm tun. Vor allem wurde anscheinend der Code 147, CLR/HOME wirksam.

Was sich hier genaueres tut, erfahren Sie in der nächsten Commodore-Welt.
(rsf)

AMIGA-EIN HOMECOMPUTER?



„Der Amiga wird die Heimcomputer-Industrie revolutionieren. Er ist der erste Heim-Rechner, welcher alles für die Hauptverwendungen eines solchen Gerätes besitzt, einschließlich Unterhaltung, Erziehung und Produktivität. Die Software, welche wir für den Amiga entwickeln, wird Ihnen die Schuhe ausziehen. Wir glauben, daß der Amiga, mit seiner unvergleichlichen Leistung, Grafik und Soundmöglichkeit der Elektronischen Kunst und der angegliederten Industrie eine verheißungsvolle Zukunft beschert.“

Trip Hawkins, Präsident Electronic Arts (unser Foto)

C16/PLUS4: TIPS & TRICKS

Vorneweg ein paar Bemerkungen zum Literaturangebot für diese Computertypen. In vielen Publikationen wird behauptet, es gäbe so gut wie keine. Das stimmt so nicht ganz, denn außer den neuerdings besser gewordenen Handbüchern gibt es meines Wissens 6 Bücher, die sich mit diesen Computern beschäftigen. Im Vergleich zum VC-20 oder C64 ist das nicht besonders viel. Aber auch die haben ja mal klein angefangen.

Auch das Angebot an Software wird, wie es aussieht, immer besser. Auf dem Spielesektor ragen vor allem die Firmen Kingsoft und Mastertronic hervor. Die Firma Kingsoft bietet außerdem auch Hardwareerweiterungen für die kleinen 16K von Commodore an. Es sind eine 16K und eine 64K Erweiterung, wobei letztere in den Computer eingebaut wird und damit immer präsent ist. Im Verkaufsprogramm sind außerdem auch noch eine Datenbank, eine Textverarbeitung und ein Musikprogramm. Auch die manchmal schwer zu erhaltenden Joystickadapter kann man bekommen. Eine 16K Hardwareerweiterung gibt es auch bei der Firma Jeschke. Schmeißt man sich mal in Unkosten und investiert in ein ausländisches Computermagazin, gehen einem beim Studium die Augen über. Da annoncierten C16/Plus4 Center, es werden Spiele besprochen und auch Artikel über diese Computer findet man dort. In einer Anzeige zählte ich über 100 Programme für den C16 bzw. Plus4. Selbst in der Spielehitparade werden die Spiele für diesen Computer berücksichtigt. Dieses natürlich alles nur in Großbritannien.

Nun ein paar Tips, die

den Umgang mit dem C16/Plus4 erleichtern sollen. Viele, die von einem anderen Commodore Computer umgestiegen sind, vermissen vielleicht

Reset:

die Restoretaste, die ein Aussteigen aus dem Programm und damit einen Warmstart ermöglicht. Auch beim C16/Plus4 ist dieses möglich. Drücken Sie zuerst die RUN/STOP-Taste, halten diese gedrückt und betätigen dann die RESET-Taste. Damit landen Sie im eingebauten Maschinensprachmonitor, den Sie dann mit der Befehlsfolge x „Return“ wieder verlassen können. Damit ist das Programm wieder da. In einem Sonderheft einer renommierten Computerzeitung steht es leider umgekehrt. Wenn Sie nämlich zuerst die RESET-Taste drücken, verabschiedet sich Ihr Programm von Ihnen. Wichtig ist nämlich, daß Sie die RUN/STOP-Taste solange gedrückt halten, bis Sie die RESET-Taste wieder loslassen.

Der eingebaute Maschinensprachemonitor ist zwar sehr nützlich, aber meiner hat einige Mucken. Mir passiert es leider öfters, daß das Programm aus unerklärlichen Gründen in den Monitor springt.

Auch für die Besitzer der Diskettenstation hält der Computer gelegentlich eine Überraschung bereit. Vielleicht ist es Ihnen auch schon einmal passiert, daß plötzlich, während des Programmierens, ein Programm geladen wurde. Wahrscheinlich haben Sie ganz aus Versehen die SHIFT und die RUN/STOP-Taste betätigt. Da die Computer C16/Plus4 ganz auf Diskette ausgerichtet sind, kann mit dieser Tastenkombination das erste Programm der Diskette geladen und gestartet werden. Beim VC-20 und C64 wurde hiermit der Cassettenrecorder angesprochen, der aber nicht, wie die Floppy, von alleine losläuft. Ich jedenfalls nehme die Diskette nach dem Laden oder Saven immer heraus, denn nichts ist ärgerlicher als ein schon fast fertiges und dann überschriebenes Programm.

1. Das Rom-Listing für den Plus 4

Auf 260 Seiten stellt das Buch eine unerläßliche Informationsquelle für alle diejenigen dar, die sich näher mit dem Computer, insbesondere der Assemblerprogrammierung, beschäftigen. Auch der normal sterbliche Programmierer findet hier eine Menge Informationen, die er zum Programmieren, bzw. zum Umschreiben von Programmen benötigt. Auf den ersten 6 Seiten befindet sich die Speicherbelegung mit allen notwendigen Erklärungen dieser Speicherstellen, ebenso eine Auflistung der Speicherbereiche, die beim Einschalten der Graphikbe-

reiche verschoben werden. Die restlichen Seiten enthalten das komplette Rom Listing als Assembler-Quellcode mit nebenstehenden in deutscher Sprache kommentierten Codesequenzen. Leider trübt der Preis von 59,— DM die Freude etwas. Zu beziehen ist dieses Buch über den Computer-Fachhandel oder über Buchhandlungen.

2. Der Commodore 16 in der Praxis

Dieses Buch befaßt sich im einzelnen mit Zeitmessung, Graphic, Entwicklung und Gebrauch von Diagrammen, Musikprogrammierung und Büroarbeiten einschließlich einer Datenbank, eines einfachen Textverarbeitungssystems und einer Finanzverwaltung. Der Benutzer wird Schritt für Schritt durch die einzelnen Programmabschnitte geführt, wobei mit Erklärungen und Erläuterungen nicht gespart wird. Die einzelnen Module, die dann zu einem Programm zusammengesetzt werden, bestehen im Schnitt aus ca. 10 bis 30 Zeilen und sind somit leicht übersehbar. Auch wird jedesmal der genaue Ablauf des Moduls erklärt. Durch die sehr genauen Erläuterungen des Modules bzw. der Programmzeilen ist dieses Buch vor allem für den Anfänger empfehlenswert, wobei er noch zwei Fliegen mit einer Klappe schlägt. Zum ersten wird er Schritt für Schritt in die Programmierung des C16 eingeführt, zum anderen erhält er dadurch einige recht interessante Programme für seinen persönlichen Gebrauch. Das Buch kostet ca. 20,— DM.

Suche C-64. Zahle ca. 100 DM für Gerät, das äusserlich beschädigt sein kann, ansonsten aber o.k. ist. Angebot mit Angabe des Baujahres u. Beschädigung an: R. Salzner, Ingolstädterstr. 12, 8832 Weissenburg. Antworte bei Annahme innerhalb 10 Tagen.

Suche für VC-20: Modulbox VC 1020. 40/80; Zeichenkarte oder -Modul. R. Ritter, Am See 22, 7520 Bruchsal 5

Suche Service-Unterlagen für CBM 4023 B, CBM 4040 und Epson MX 80, sowie Interface VC 128; CBM 4040 (IEEE 488) oder Bausatz gegen Kostenbeitrag. H. Kastenmeier, 8401 Irnkofen 16. Tel. 09454/244, möglichst nach 19 Uhr

Verkaufe für VC 20: 16K Erw. und Schreibmaschinenkurs = VB 60 DM. Suche für C16: Netzadapter, Software (Tausch). Listen und Angebote an: Thomas Weiß, Meissener Bruch 3, 4950 Minden.

Tel: 0471/36159.

Hallo.
Suche Vioce; Master bis 50 DM. Interessierte Leser bitte melden. Schreibt oder ruft an. Tel: 04334/706. Marc Pedex, ab 17 h. Katzheide 20. 2371 Bredenbek.

VC 20 + 3K + 16K + Datensette + Joyst Ick + 10 Kassetten und 2 Bücher + 2 Spielebücher. VB = 300 DM. Tel: 0241/871058.

Suche dringend Rollen- und Strategiespiele (C 64) z.B. Ultima, Colonial Conq., Battle for Normandy. Corg. + Anl.) Nur Disk-Liste mit Preisvorst. an: Andreas Morlok, Hegelstr. 27, 7400 Tübingen (C 64).

Suche C 16-Programme. Angebote an M. PSchner, Tel.: 02371/42869. PS: Nur auf Kassette.

Computer-Olivetti-P 6040 mit Floppy + Drucker P 6040 SBS zu verkaufen. Preis 1.000,- DM. Tel.: 02932-29179.

Gute C;64 Spiele sind teuer – nicht bei mir: Bomb Jack 12 DM, Karate; Man (!!!) 12 DM, Donkey KÖng 20.3 9 DM, Dragonwar Ld (!) 10 DM, usw. Katalog anfordern. S. Kleina. Tel: 05405/1604.

Shinwa CP 80 II. Suche ver-zweifelt HC-Routine Hires-Graphic. Text für C 64. Klaus Weiß, Am Grenzgraben 26a, 4000 Düsseldorf 30.

Suche Kontakt zu C 64/1541 Benutzern, im Raum HH, auch mit Gel. zusammentreffen. Bin über die 40. Daher ält. Personen erwünscht. P. Hille, Brunnenhofstr. 5, 2 HH 50.

Suche CBM 8032. Ggf. auch Software "0ZZ" etc. Preisangebote an U. Finke, Stade. Tel.: 04141-64235.

Suche Commodore Pet 2001. Kleine oder große Tastatur. Möglichst billig oder geschenkt, da vom Taschengeld Neugerät nicht finanzierbar. Gerät wird abgeholt. Christrian Finke, Stade. Tel: 04141-64235 ab 14 h.

Verkaufe: VC 20 Programmierhandbuch" deutsche Ausgabe), VB 9,99 DM. "Programme für VC 20/ Spiele, Utilities, Hardware-Erweiterungen" (W. Hofacker), VB: 9,99 DM. Jörg Güttel, Kölner Straße 99, 5650 Solingen 1.

C-64 Software-Angebot: Alles neu und original, aber bis zu 60% billiger! Info Sfr. 2 - Briefmarke) bei: Beat Biddiger, Dammweg 25, CH - 3904 Nafers.

Computer-Camp.
Wer – im Alter zwischen 15 und 17 Jahren – war schon alleine in einem 64-er Computercamp (z.B. in den Ferien) und könnte mir seine Erfahrungen und Empfehlungen mitteilen. Schreibt bitte an: Dietmar Splitt, Stettiner Str. 51, 7144 Asperg.

Verkaufe 64 Anwender, Hilfsprogramme und Spiele. Listen gegen frankierten Rückumschlag (1,80). W. Schreiner, In der Laach 27, 5400 Koblenz 33.

IG-C + T. Zeitschrift, DFÜ, Kurse, FREE-Soft. Kein Mitgliederbeitrag. Info bei: IG-C + T, P.O. Box 18, CH - 6312 Steinhausen.

--- VC - 20 ---
Ich verkaufe: VC-20 für 100,- (VB) 3K-Speichererw. 30,- (VB). Data-Becker (Tips + Tricks VC-20)-Lektüre 5,-. Spielekassette umsonst. Alexander Bohndorf 2 HH 63. Tel.: 040/590991. Alles VB!!

Suche Disketten.
Wer hat alte und mindestens 95% fehlerfreie, 5 1/4 Disketten zum Preis von weniger als 1,50 DM pro Stück? Kaufe bis zu 25 Stück. Schreibt bitte an: Dietmar Splitt, Stettiner Str. 51, 7144 Asperg. Eilt!

Gebe ab: Simons Basic; incl. Handb. + Lit.: 100 DM/ ZX 81 16 K, Handb., Lit. + Software.: 100 DM/ Discsorter f. 5000 Prg., mit Anlgt.: 70 DM/ U.C. Parpart, Sachsenhäuserstr. 21, 3578 Schwalmstadt 1/ 06691-2401.

Wegen Systemwechsel zu verk.: Original PRGC 64. Superbase Fr. 50, Datamat Fr. 40, Ada Fr. 40, Basic 64, Fr. 40, Profimat 40, Call Result Fr. 80, Eckert, Militärstr. 105, 8004 Zürich.

Suche defekte SX-64 und VC-1541. Angebote an Horst Herschberger: 06198/33635.

VC-20 Achtung! Wer von euch hat ein Programm (auf Papier) welches er selbst geschrieben hat? Meldet euch bei: H. Stadelmann, Seestr. 43a, 8805 Richterswil.

Verkaufe Rechnungsstellung (Invoicing) (Diskette, neu DM 60,-). Tel: 0511/5555

Software gesucht. Für CBM 710 (600/700) und für CBM 8xxx, auch Erfahrungsaustausch angenehm. Dr. Pabst, 06171-24826.

Suche für VC-20 billigen Drucker. Zahle – 200 DM. Tausche o. kaufe Programme – 16 K. Schickt eure Angebote bzw. Listen an Christian Niedgrer, Kirchstraße 134, 6974 Gaissau, VLBG, Österreich.

Über 150 Spielepokes für nur DM 10. Anzufordern bei: Jens Günther, Oberhof 1, 3501 Cuxhafen. Tel.: 05665/1275.

Wer hat das MBS (Mehrbenutzer System) von Commodore und würde sich davon trennen? Zu einem fairen Preis. Angebote mit Preis an B. Welte, Brackeler Hellweg 150. 46 Dortmund 12.

Achtung – PFÜ-Freaks! Suche Leute, die mit in Regensburg eine Malibox gründen. Bei Interesse bitte unter 09404/1814 melden (von 18 bis 22 Uhr).

Suche C-64 + Datensette. Zahle DM 300,-, wenn beide Geräte noch gut sind. Verkaufe Tischrechner HP 97 mit viel Zubehör für DM 800,-. Tel.: (07031) 84513 ab 18 Uhr.

Verkaufe 12 Originalspiele-Kassette C 64 – 10 DM/ Pro Spiel (Pastf, Shamus, Fort Apoc. 4A). + Computerhefte: Run, Happy Computer, 64'er: Telematch sowie Trakball – Preis VB –. Suche Drucker (Matrix) gebr. Tel.: 0202/591987, Wuppertal.

C 16 – C 16 – C 16 –
Suche Hardware-Unterlagen spez. Schaltbilder sowie Datenblätter der ICs z.B. 7501 etc. und speziellen Instruction-Set des 7501 bzw. Abweichungsliste zum 6502/6510. Uwe Peters, Tannenweg 9, 2351 Trappenkamp.

--- Umsonst ---
Tausche kostenlos Programme für VC-20 + Basic-Kurs am VL-20. Besitze 3K-Erweiterung. Achim Bergweiler, Händelstr. 20a, 556 Wittlich, Tel.: 06571/7655.

Suche gebrauchte Floppy VC 1541 für C 64. (unter 300 CM). Verkaufe CP 1 (VB) vom Kosmos! (neuwertig). Tel.: 09734/219, erreichbar ab 18.30 Uhr.

Suche Grafikprg. für Epson-Drucker LX-90 (C-64). Ährlich Anlgt. GX-80 Seite 27. J. Gänge (06186) 2509.

Verkaufe VC-20 + Datensette + 16 K Modul und 4 andere Modulen für 400 DM. Thomas May, Siemensstraße 31, A – Mettingen.

VC 20 mit folgendem Zubehör zu verkaufen: Programmierhilfe Modul, Maschinensprache Modul 40/80 Zeichenkarte 64 K Erweiterung (28159 fre) Datensette, Softw. und 12 Büch VHB 550 DM. Tel.: 0431/62216.

Tausche, kaufe und verkaufe Spiele für den VC-20. Christoph Krohn, Goethestr. 35a, 5980 Werdohl.

Verkaufe: C64 + IEEE-Bus + 1001 + 80-Zeichenkarte + HPS 801 + Textverarbeiter + Multiplan etc. + Koppler + Datamat + Profi-Pascal + Oxford-Pascal + Pascal 64 + Forth 64 + Fig.-Forth + Comal + Basic 64 W. Guldenberg, Quellstr. 166, 5180 Eschweiler

Wir — der CC-CLUB — AACHE — suchen noch Leute, die eigene Progr. haben + diese vermarkten möchten. Aber nicht wissen wie. Wir helfen euch. Ruft uns an oder schreibt uns: Commodore-Club — AC - 5100 Aachen, Mittelstr. 13 oder Tel.: 0241-13947, von 8 bis 13 Uhr. Neudecker.

Suche dringend Tauschpartner für VC-20 Prgr. ab 8 K u. 8 K Modul aufwärts. Suche auch Sprachen u.a. gute Progr. !! Liste an: Ralf Koenig, Wuragstr. 23, 58 Hagen 5. T.: 02334/2053 nach 16 h.

C — 64 C — 64 C — 64 C — 64 Schüler sucht C-64, 1530, 1541, technisch einwandfrei, faire Bezahlung (auch einzeln) unter (09561) 15 399 nach Edi fragen. C — 64 C — 64 C — 64 C — 64

PL	4	*Austria*
US	4	SUCHE
4Plus4	4 4	Software
Plus 4 P	444	(Tapes)
LU	4	100%ige
S4	4	Antwort

Ch. Mautner A-4191
Vorderweissenbach 92

Suche zuverlässigen Tauschpartner für C64 Programme nur Kassette. Listen an Peter Wevelsiep, Kaiserstr. 69, 4330 Mülheim a.d. Ruhr. Ich antworte garantiert. Suche billigen Drucker MPS 801.

+4	Suche alles über
+4	C16/Plus4. Tausche
+4	und kaufe.
+4	100%ige Antwort
+4	Bernhard Ömer
+4	Weichstetten 28
+4	4502 St. Marien
+4	Österreich

Verkaufe
Floppy 1570 ungebraucht
Anfragen an: E. Schoenberg, Maierhofenerstr. 56
7972 Isny
Tel. 07562/74660

Suche C-64 bis DM 280.
Rumänischen Zeichensatz für Drucker Eposn FX-80 mit Görlitz-Interface für C-64
Suche weiter für C64 od. VC20 Originalspielemodule max. 7.—. Tel. 09732/4297

*** ACHTUNG ***
C16/116 sucht noch Mitglieder!! Schreibt an: Helge Mierisch, Eugen-Richter-Str. 60
5800 Hagen 1

Wegen Systemwechsel zu verkaufen: CBM PLUS/4. VB SFR. 280 incl. Porto. 1 Monat alt, 5 Mt. Garantie. Schreiben an: Reto Fäs, Schlattweg, CH-5040 Schöftland

Wer hat 128-er-Programme?

Wer hat Erfahrung beim Umbau von einem Epson-Drukker FX80 mit Görlitz-Interface auf NLQ-Schrift wie beim FX-85. Tel. 09732/42 97. Rufe zurück. Suche Vizawrite, nur Original, für C64 od. VC20 als Modul od. auf Disk. bis DM 100,—

Verk. VC-20 DM 120. Monitor grün oh. Ton 179; 40/80 Zeichkarte 79; 64K Erweiterer. 89; Stechplatzerweit. 6-fach 79; Eprokarte mit Masch.Sprach. Monitor 39; alles zusammen DM 500,—
Tel. 09732/4297

Suche Anwenderprogramme für C64 — Angebote an: Harald Henning, Bonifatius 8, 6742 Herxheim b. Landau

Verkaufe: „Das Schulbuch zum C-64 (Data-Becker) VB 23,45 DM, „C64, Der Computer für Einsteiger und Aufsteiger“ (Riedl/Quinke), VB: 12,34 DM, Jörg Güttel, Kölner Straße 99, 5650 Solingen 1

CBM 4032/8032 mit 80Z Karte 600,— DM
CBM 4040 Doppel Floppy-Disk auch für C64 geeignet 800,— DM
CBM 8032/8050/4023 3.000,— DM
Tel. 05251-26213

Verkaufe für VC-20: 3KB (20 DM) 16 KB (60 DM), 32 KB (80 DM) 40/80-KZ-Karte (150 DM), Modul Clowns (15 DM); Suche Spielmodule VC-20/C64; Tel.: 07053-7618 ab 18 Uhr

*** C-64 ***
*** Suche / tausche ***
Spiele und Lernprogramme (nur Disk)
Tel.: 02303/87 83, Niessen, Goethestr. 21, 4750 Unna
*** C-64 ***

Neuer Computerclub — „ATCO“ Atco beschäftigt sich mit Atari, C-64, VC-20, CG-16/116/+4, Schneider CPC 464. Eigene Clubzeitschrift. Info gegen Rückporto ATCO-Zentrale, c/o F. Baumann, Wacholderweg 4, 6236 Eschborn 2

C-16 116 User. Achtung! Ich biete 5 Topspiele für nur 20 DM. Diese sind: Dallas, Pachan, Terminator, Terror, Wibryl, Keine Raubkopien! 20 DM. Bitte auf folgendes Konto überweisen: Kontonr. 3603768, BLZ 41450075
Bestellungen an: Roland Fuchs, Blanckenagelweg 9, 4770 Soest-Westfalen

C64 — Verkäufe RS232 Modul neu 50 DM, Disketten DS DD Hubring 10 St. 15 Anschluß an Club im Raum 6000 gesucht. Suche neue Software und Anl. Kauf oder Tausch, Tel. 069/78 27 18 ab 18 Uhr

*** Achtung ***
Suche C64 + Floppy 1541 (ohne Schaden) biete bis 700,— DM. Melden bei Stefan Behringer, Tel. 0911/313238

Hey Freaks! Suche Tauschpartner für gute Disk-Software. Michael Haun, Markusweg 4a, 495 Minden, Tel.: 0571-33 444. P.s.: Verkäufe auch Org. Five aSide Football!

Wir haben sooo viel! Für Commodore-Fans + Freaks *
Gratislisten anfordern U HSW-Vertrieb — Lindenstr. 30, 3012 Langenhagen 5
0511-73 43 32

Suche PRG'e für Plotter 1520 Jede Art (z.B. Treiber, Grafik, Mal, Textverarbeitung etc.)
Wolfgang Behring, Gewerbeschulstr. 64, 5600 Wuppertal 2, Tel. 0202/59 19 87

VC20 — 64KB Vollausbau nur 95 DM, gebrauchte Originalsoftware, billig abzugeben sowie Mas-Programm-listings — Bitte Liste anfordern bei Harald Bartz, Ewaldstr. 1, 4100 Duisburg 14 — Rheinhausen

VC-20 mit 3K, 8K, 16K Erweiterung und einige Spiele zu verkaufen — VB 300 DM. Floppy 1570 gesucht. Angebote an Horst Zens, Rhein-str. 20, 4150 Krefeld

CBM-2001, 8K mit Datensette und Drucker — Epson TX80B mit Interface F. CBM + div. Programme. Preis VB
W. Heisslein, Tel.: 0911/345072

C64 und Monitor gesucht
Angebote: Tel. 09072/5224

VC-20 mit 3K, 8K, 16K Erweiterung und einige Spiele zu verkaufen. VB 300 DM. Floppy 1570 gesucht. Angebote an: Horst Zehs, Rhein-str. 20, 4150 Krefeld

VC-20 VC-20 VC-20 VC-20 Suche Kontakt zu Club und zu Tauschpartner (Spiele, Anwendungen und, und) Suche auch Fotokopien von Listings. Sendet Eure Liste an:
Peter Hammer, Leihgesterner Weg 16/404 6300 Giessen

VC-20 VC-20 VC-20 VC-20 Suche preiswerte Supererweiterung und 8K-RAM
Peter Hammer, Leihgesterner Weg 16/404 6300 Giessen
0641/74573 ab 20 h

VC-20 VC-20 VC-20 VC-20 Suche die Supererweiterung VC1211 A + (CBM Revue 5/85) + VC-20 Topgames
Antwort an: Olaf Wolnd, Wildgansstr. 16b, 2000 Hamburg 73
VC-20 VC-20 VC-20 VC-20

8096 + FL4040 verk.
0421-555091

Unidat — universelle Datei für den C64. Viele Funktionen! Freie Erstellung einer Datei! Disk incl. Anleitung: 20 DM!!
Frank Hoffmann, Paderbornerstr. 12, 47 Hamm 4

Gepflegter Computertisch samt Papierständer für nur DM 600,— (NP DM 1300,—) abzugeben.
Tel.-Nr. 08092/6570

Suche Amateurfunkprogramme für C16 RTTY-Packet-Radio-Rufzeichendatei, Beyer, Amselstr. 4, 8551 Hemhofen, Tel. 09195-7472

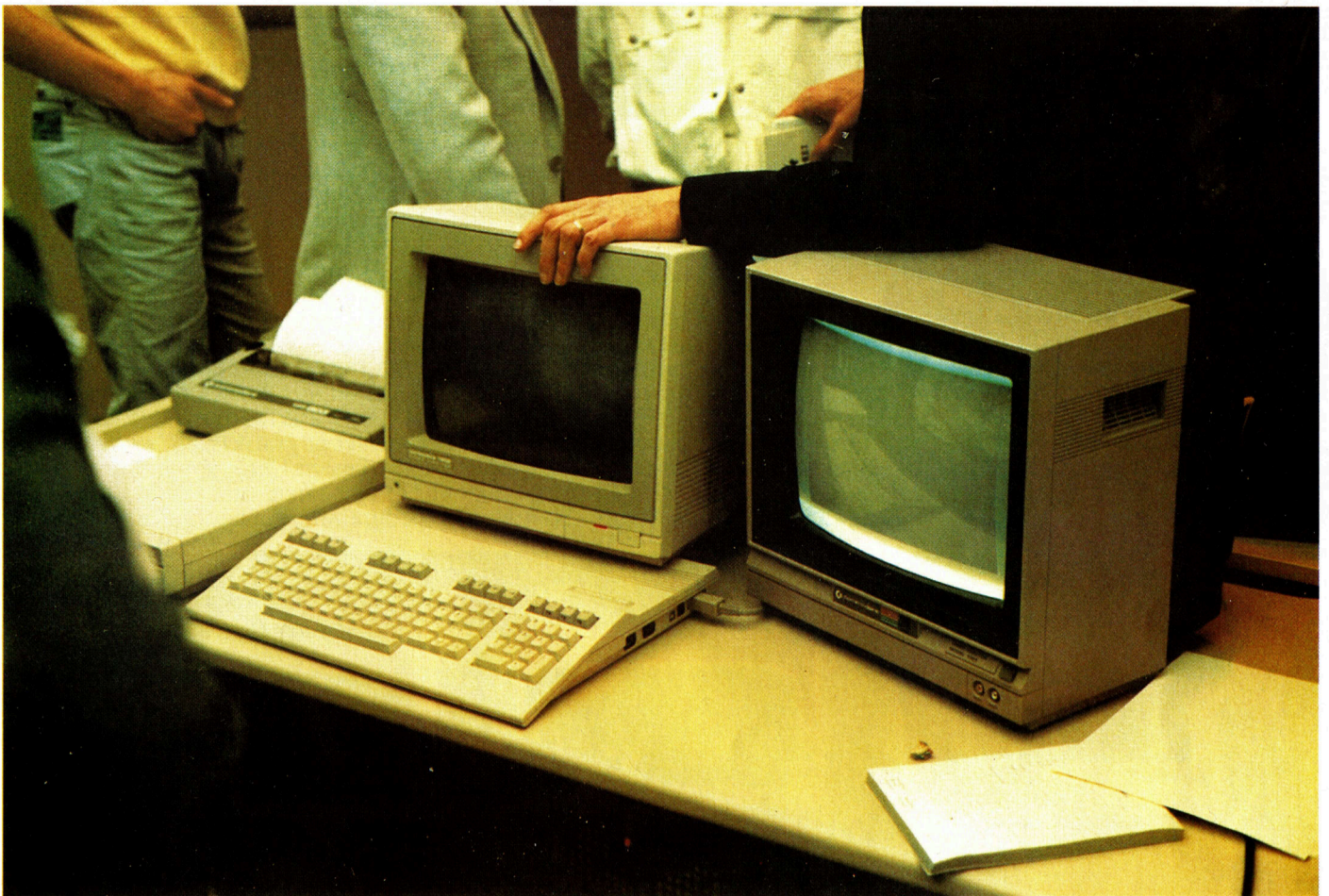
Verk. G-T00L 2000 Plus für C-64, 170 Makros für Grafik, Sound, Sprites, strukt. Prog. usw. Disk für DM 35,—
Hans G. Müller, Am Erlengbach 4, 8618 Leesten

Suche Plotter 1520 ca. 100 DM oder Tausch gegen Floppyspieder (16fache Gesch. Toolkits u.v.m. Einfach in C64 u. Floppy stecken)!. Suche auch gute Programme und alles andere für den C64.
Reiner Ruse, Gladenbacherstr. 35, 3555 Fronhausen
T. 06426/7418

MIT 2 FLOPPIES RICHTIG PROGRAM- MIEREN!

Wieso das? Sie sind ganz einfach von Ihrer 'F3'-Taste verwöhnt, denn diese Taste bringt Ihnen die Directory ohne Programmverlust. Allerdings nur von Laufwerk Nr. 8. Wenn Sie also den Fehler gemacht haben, Ihr Pro-

tenbank '! Wissen Sie nach der fünfzigsten Adresse noch, ob Sie Fam. H. Schmitz schon eingegeben haben oder nicht? Also Directory. Ich brauche nichts zu sagen, Ihr Programm ist schon wieder 'futsch'.



Manchmal ist es durchaus sinnvoll, während des Programmierens zwei Laufwerke einzusetzen. Angenommen, Sie wollen eine Adressdatei erstellen. Die Daten sollen auf eine separate Disk gespeichert werden. Sie sind also gezwungen, zwischendurch Probeläufe zu machen, um Schwachstellen Ihrer Entwicklung aufzuspüren. Sie sind also ständig dabei zu ändern und zu ergänzen. Es entstehen mehrere Versionen, welche Sie immer wieder abspeichern. Sie kommen mit Sicherheit an einen Punkt, wo Sie nicht mehr

wissen, unter welchem Namen Sie Ihre letzte Version abgespeichert haben. Macht ja nichts, werden Sie sich sagen. Der Aufruf der Directory wird es mir zeigen. Tut es auch. Version 'V1.8' war der letzte Eintrag. Na bitte, dann wird eben jetzt unter Version 'V1.9' abgespeichert. Gedacht, so getan. Nach Eingabe von 'RUN' und 'RETURN' erscheint 'Syntax Error in 0'. Der 'LIST'-Befehl bringt Ihnen die Directory. Sie ahnen schon richtig, Sie haben nicht Ihr Programm, sondern die Directory abgespeichert.

Programm auf Laufwerk Nr. 9 zu speichern, ist Ihr Programm von der Directory überschrieben worden. Denn das Laufwerk 9 können Sie nur manuell auswählen. Wenn Sie jetzt sagen, also gut, dann wird eben als Programmlaufwerk die Nummer 8 gewählt, so muß ich Ihnen sagen, daß dies auch nicht der Weisheit letzter Schluß ist. Was passiert wohl, wenn Sie Ihr Programm fertig haben, und Sie geben laufend die Adressen Ihrer Bekannten, Verwandten usw. ein? In diesem Falle benutzen Sie Laufwerk Nr. 9 als 'Da-

Es ist also doch sinnvoll, das Programm auf Laufwerk 9 und die Daten auf Laufwerk 8 zu legen. Denn programmiert wird in der Regel nur einmal und da muß man sowieso mehr aufpassen und denken. Sie können natürlich auch beim Programmieren das Laufwerk 8 benutzen, aber dann sollten Sie spätestens nach Fertigstellung des Programmes die Geräteadressen austauschen. Dies ist zwar etwas mehr Aufwand, aber man braucht in diesem Fall die Directory nicht zu fürchten.

P. B.

SO FINDEN SIE DEN RICHTIGEN DRUCKER

Neben der Floppy ist der Drucker das wichtigste Werkzeug für einen Computer. COMMODORE WELT sagt Ihnen, was Sie vor dem Kauf eines Druckers beachten sollten.

Drucker gibt es mittlerweile schon wie Sand am Meer. Ob für den Heimgebrauch, zum Ausdruck für kleinere Programme und Listen oder aber für den Profi-Bereich: Für jede Nutzung gibt es mindestens ein halbes Dutzend Anbieter mit verschiedenen Leistungen und verschiedenen Preisen.

Selbstverständlich muß diese Frage im Endeffekt jeder einzelne für sich selbst entscheiden. Und gerade im Fall eines Druckerkaufes sollte man sich für diese Entscheidung besonders viel Zeit nehmen. Das erste, was man tun sollte, ist, in den eigenen Geldbeutel zu schauen, um sich damit klarzuwer-

Typen heraus, die mehr kosten, als Sie bereit oder fähig sind, dafür auszugeben.

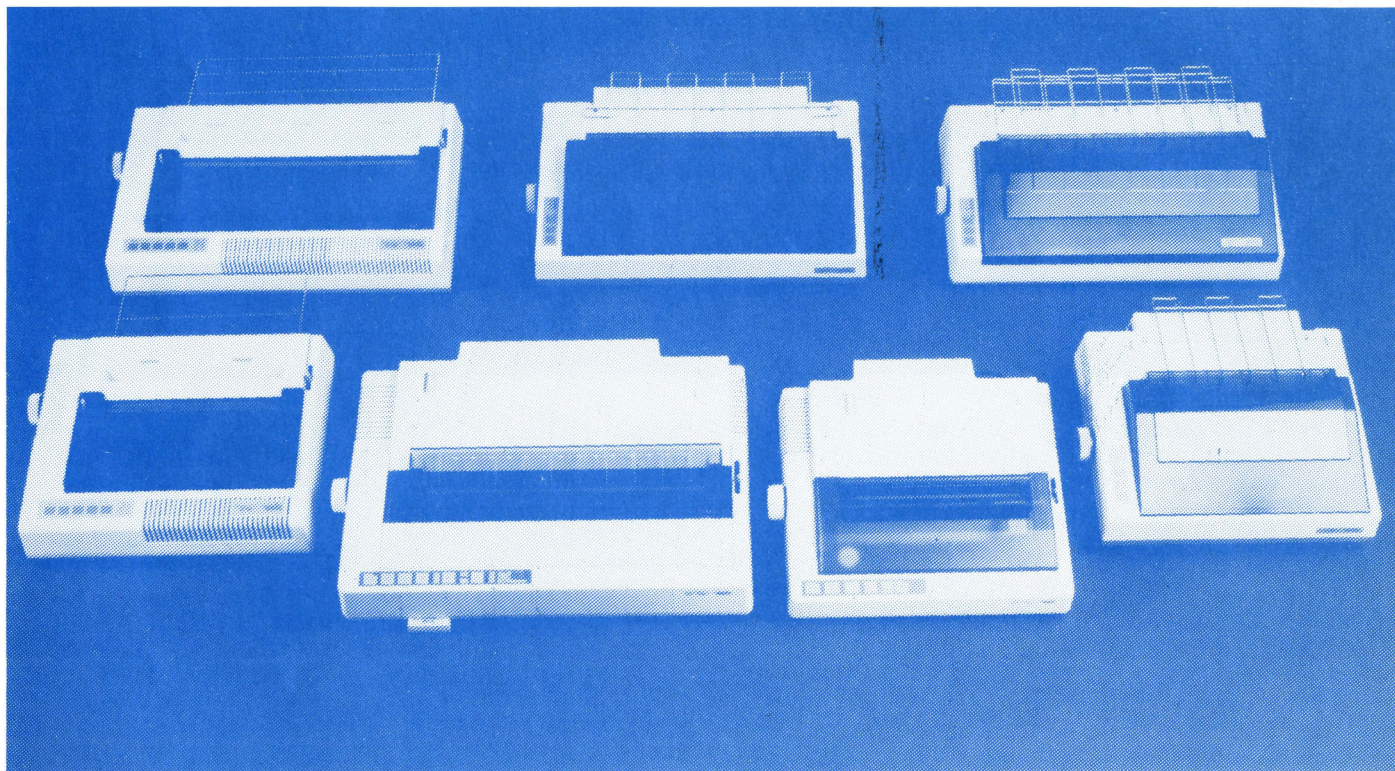
In der zweiten Auswahl sollten Sie sich die Frage stellen, ob Ihr Drucker einen bestimmten Standard erfüllen soll.

Gängige Standards sind zum Beispiel Epson- und IBM-Druckformate. Die für den C 64 geschriebene Druckersoftware ist entweder für einen original Commodore Drucker (MPS-Serie) oder aber für einen Drucker mit Epson-Standard bestimmt; ebenso verhält es sich beim Apple IIe.

Haben Sie nun zum Beispiel eben einen C 64 und Sie sind sich sicher, daß Sie keinen IBM oder einen IBM-ähnlichen Rechner in nächster Zukunft kaufen wollen, so brauchen Sie einen Drucker mit Commodore- oder Epson-Standard.

Ansonsten sollte Ihr neuer Drucker sowohl den Epson- als auch den IBM-Standard beherrschen.

Ein weiteres Auswahlkriterium sollte die Grafikfähigkeit des Druckers sein. Ist er grafikfähig, so kann man mit diesem Gerät sowohl normalen Text als auch, mit einem Zusatzprogramm, hochaufgelöste Grafik-



„Welche Marke und welchen Typ nehme ich, was ist für mich am Besten?“ Diese oder ähnliche Fragen stellen sich jedem Computer-Benutzer, der sich einen Drucker kaufen will.

den, welche Summe überhaupt ausgegeben werden kann. Jetzt erst sollte man sich informieren, welche Drucker für welchen Preis angeboten werden. In der ersten Auswahl fallen alle

Es gibt sie wie Sand am Meer: Drucker. Hier allein das Angebot eines einzigen Unternehmens: Star.

bilder ausdrucken. (Die meisten modernen Drucker sind grafikfähig.) Ein weiterer, nicht minder wichtiger Punkt ist die Art der Schnittstelle, die der Printer besitzt.

Schnittstellen gibt es allgemein nur zwei verschiedene: Die serielle und die parallele Schnittstelle.

Die meisten Drucker im Home- und Personalcomputerbereich besitzen parallele Schnittstellen, eine Ausnahme bildet der Commodore 64.

Will man einen Drucker mit paralleler Schnittstelle an den C 64 anschließen, so benötigt man hierfür noch eine besondere Schnittstelle, ein Druckerinterface.

Bei manchen Angeboten ist diese Karte schon dabei, bitte beachten Sie dies!!! Ansonsten sind Mehrausgaben bis zu 250 und mehr Mark möglich!

Hat man diesen wichtigen Punkt auch beachtet, kann man endlich dazu übergehen, den Drucker nach seinen eigenen Leistungen zu testen. Beginnen wir mit:

a) Einzelblatteinzug:

Einzelblatteinzug bedeutet nichts anderes, als daß Sie mit diesem Drucker ganz normales Papier (meist bis DIN A4-Format) benutzen können. Ein nicht zu unterschätzender Vorteil.

b) Traktor bzw. Endlosblattverarbeitung:

Dieser Zusatz bedeutet, daß der Drucker Endlospapier mit Lochrand verarbeiten kann.

Auch diese Eigenschaft ist von großer Bedeutung.

Selbstverständlich gibt es viele Drucker, die sowohl den Punkt (a) als auch den Punkt (b) erfüllen; sie sind den anderen Geräten vorzuziehen.

c) Druckgeschwindigkeit:

Die Druckgeschwindigkeit wird in Z/sec. = Zeichen pro Sekunde angegeben. Hat ein Drucker die Angabe 180 Z/sec. so bedeutet dies, daß man von einem Text mit 80 Zeichen pro Zeile jede Sekunde 180 Zeichen abdruckt.

Da die meisten Texte jedoch kürzere Zeilen haben, ist dieser Wert nicht hundertprozentig verbindlich, sondern zu hoch.

Dennoch ist ein Drucker mit einer Geschwindigkeit von 180 Z/sec. um einiges schneller als ein Drucker mit nur 120 Z/sec.

d) NLQ=near letter quality=

Schönschriftmodus:

Ein durchschnittlicher, guter Drucker schreibt normalerweise mit einer 8x9 Punkte Matrix.

Schlechtere Drucker, wie zum Beispiel der Commodore MPS 801, ha-

ben nur eine 5x7 Punkte Darstellung.

Die Zeichen eines solchen Druckers sind weit schlechter zu lesen, also unsauberer, während die 8x9 Punkte-Drucker weitaus schönere Druckbilder erzeugen.

Achten Sie also vor dem Kauf auf die Druckmatrix Ihres Druckers!!! Einige Drucker haben nun einen Extra-Modus, NLQ genannt. Dieser Modus simuliert eine 18x24 Punkte Matrix; der Zeichenausdruck in diesem Modus ist, gegenüber dem Normalmodus, umwerfend; es ist kaum noch ein Unterschied zu einer Schreibmaschine feststellbar.

Dafür ist die Druckgeschwindigkeit auch erheblich verringert, was man aber für besonders wichtige Briefe in Kauf nehmen sollte.

Wichtig ist auf jeden Fall, daß der Drucker die Schönschreibmöglichkeit besitzt.

e) Druckerpuffer:

Ein Druckerpuffer ist ein Speicher, in dem Daten, die vom Computer kommen und die ausgedruckt werden sollen, zwischengespeichert = gepuffert werden.

langsam und die Wartezeit dementsprechend auch länger ist.

Ein vernünftiger Druckerpuffer faßt mindestens 5 KBytes, mehr Speicher ist natürlich besser, aber auch teurer.

f) Zeichenentwicklung möglich:

Einige Druckerfabrikate bieten dem Benutzer noch einen ganz besonderen Leckerbissen.

Auf ihnen kann man eigene Zeichen entwickeln und dann abspeichern. So wie man auf dem C 64 den Zeichensatz ziemlich leicht ändern kann, so ist dies auch bei solchen Druckern möglich.

Zwar wird diese Eigenschaft fast nie gebraucht, sie ist aber dennoch eine nette und für manche bestimmt nicht unnütze Spielerei.

g) Anzahl der Zeichensätze:

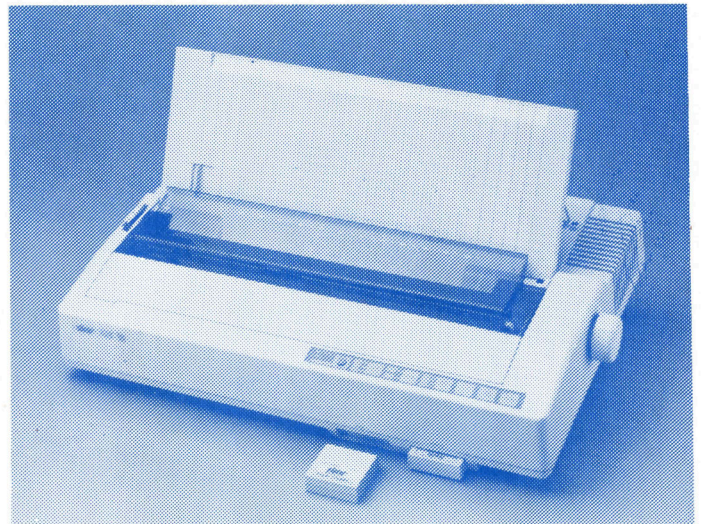
Ein guter Drucker sollte den amerikanischen ASCII-Zeichencode sowie den deutschen Zeichensatz mit Umlauten (äöüß) besitzen.

Viele andere Drucker haben überdies noch französisch, norwegisch, griechisch und andere Zeichencodes fest integriert.

h) Farbband:

Das Farbband stellt für vieldrucken-

Moderne Drucker ändern ihre Schriften heute per Einschub-Kassette, Dip-switcher umstellen ist nicht mehr „in“.



Hat man zum Beispiel ein 3seitiges Listing auszudrucken, so nimmt der Puffer, je nach Größe, alle 3 Seiten auf und man kann, während der Drucker arbeitet, auch am Computer weiterarbeiten.

Bei kleineren Ausdrucken entsteht daher keine „Wartezeit“, bis der Drucker alles gedruckt hat und sich der Computer erst dann wieder meldet.

Besonders im NLQ-Druck lernt man einen großen Druckerpuffer schätzen, da der Drucker in diesem Modus, wie erwähnt, besonders

de Computerbesitzer eine stetige Ausgabe dar.

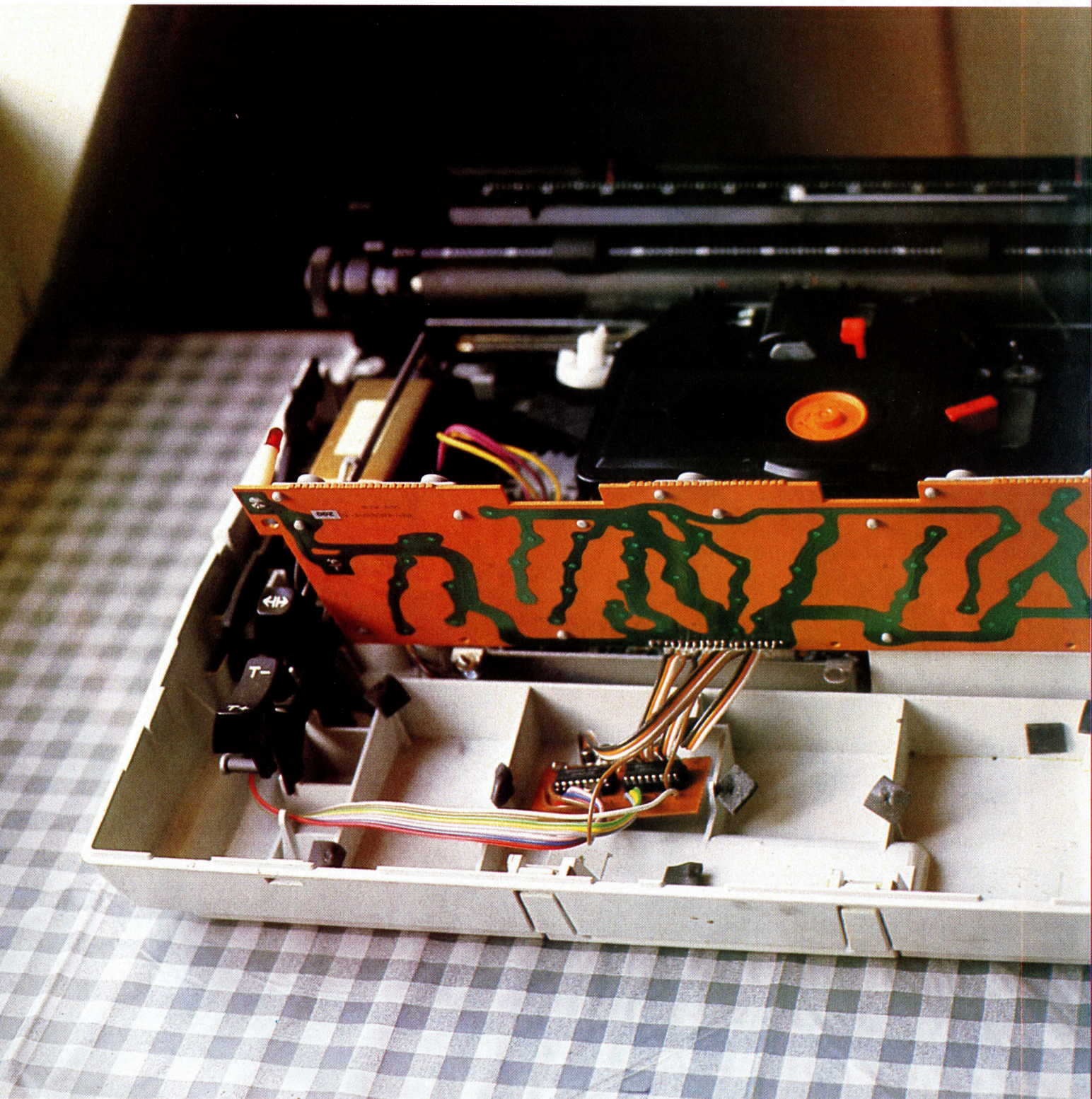
Deshalb sollte man sich vor dem Kauf des Druckers nach den Preisen für die Farbbänder erkundigen.

Es ist ebenfalls gut zu wissen, wieviel Zeichen lang ein Farbband hält, d.h. nach wie vielen Druckseiten die Druckintensität nachläßt.

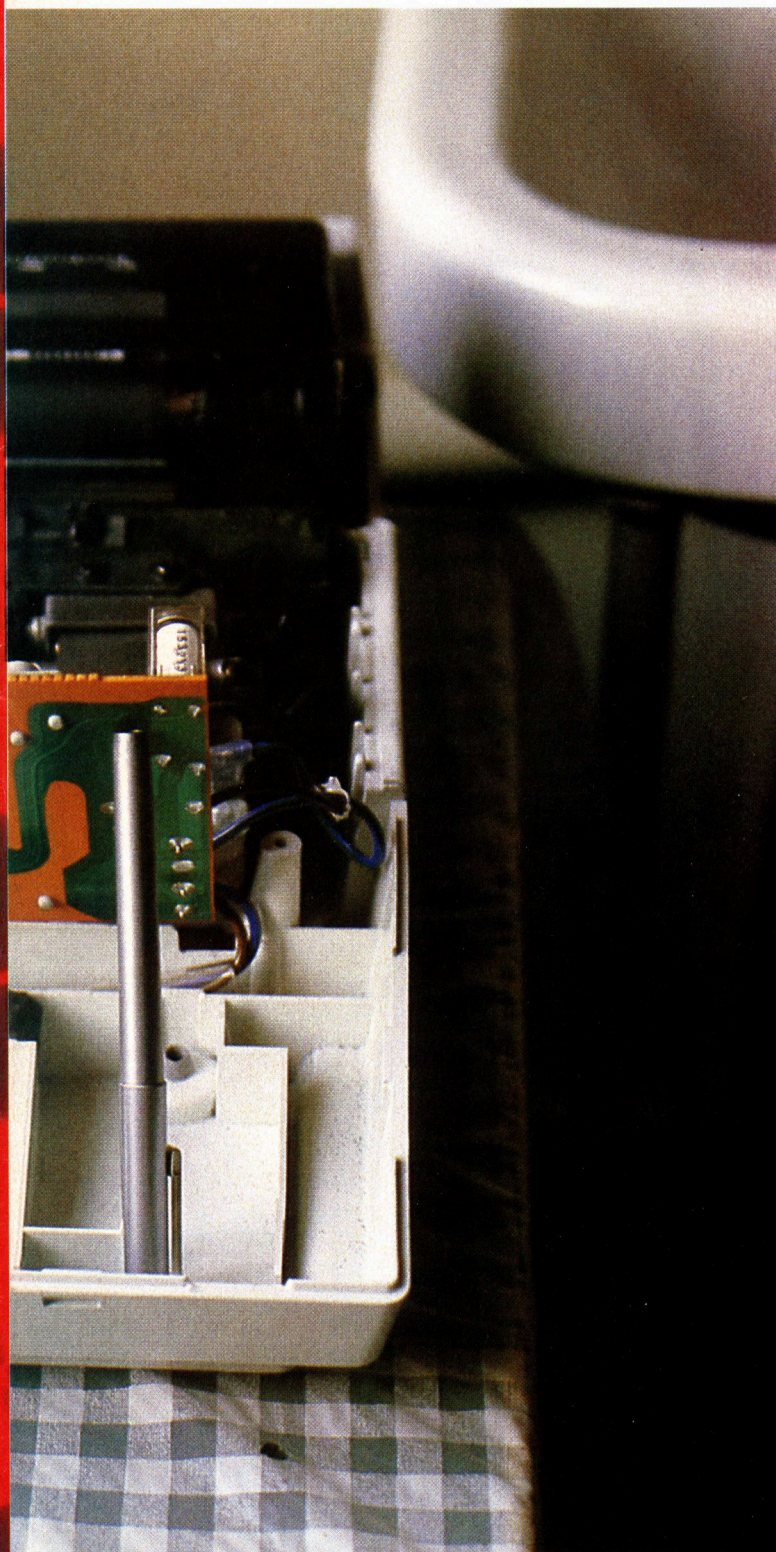
(Manche Farbbänder haben einen Nachfülltank, durch den man man die Farbbandkosten erheblich senken kann. Bedenken Sie also auch dies vor dem Kauf!)

Ph. Bertschig

SO KOPPELN SIE EIN NISCHE SCHREIBMA



E ELEKTRO- SCHINE AN DEN C64



Vor einigen Jahren kaufte ich mir meinen ersten Heimcomputer und wenig später eine Typenradschreibmaschine Marke Brother Electronic 8300. Damals war ich noch der Meinung, daß alle Schreibmaschinen schon einen Computeranschluß besitzen, und ein Interface sei nur ein mehradriges Kabel mit passenden Steckern. So war ich doch ein wenig erstaunt, als ich nirgendwo eine Buchse für einen Computeranschluß fand. Vielleicht war der Anschluß irgendwo innen auf der Platine und nur aus Kostengründen nicht nach außen geführt worden?

Bis heute nichts gefunden

Nun, leider habe ich bis heute nichts gefunden. Dafür sah ich, daß von der Tastaturplatine 17 Leitungen in die Elektronik führen. Durch Drücken einer Taste werden immer nur zwei bestimmte Drähte verbunden, und wenn man bei eingeschalteter Schreibmaschine solch ein Drähtepaar mit einem Widerstand zwischen 0 und 1000 Ohm kurzschließt, wird ein Buchstabe gedruckt. Durch Probieren konnte so die ganze Tastaturmatrix entschlüsselt werden.

Dabei haben die Anschlüsse 1–8 ein positives,

die Drähte A–G ein negatives Potential. Eine Diode als Kurzschluß funktioniert also nur, wenn sie richtig herum gehalten wird.

Um die Tastatur mit dem Computer ansteuern zu können, braucht man Schaltelemente, die ihren Innenwiderstand beim Anlegen einer Spannung erniedrigen. Geeignete Bauelemente sind der Transistor, das Relais und der Optokoppler. Im Falle eines npn-Transistors (bei einem pnp-Transistor ist es umgekehrt) werden z.B. die Anschlüsse 1 A der Tastatur kurzgeschlossen, wenn ein Bit am Userport high ist; es wird dann ein Buchstabe gedruckt. Ist das Bit low, dann sperrt der Transistor – kein Buchstabe wird gedruckt. Verwendet man ein Relais, so muß man noch einen Treiber, z.B. das IC 75469, zwischenschalten, da der Userport nicht genug Strom liefert. Solche Miniinterfaces bringen natürlich noch nichts. Schließlich will man ja alle Buchstaben drucken können.

Einfache Lösung

In Abb. 3 sehen Sie nun das fertig entwickelte Schaltbild. Die Schalter der oberen Reihe wurden hier mit pnp-Kleinsignaltransistoren realisiert. Für die anderen Schalter wurden Opto-



Klein, aber fein: das selbstgebaute Schreibmaschineninterface

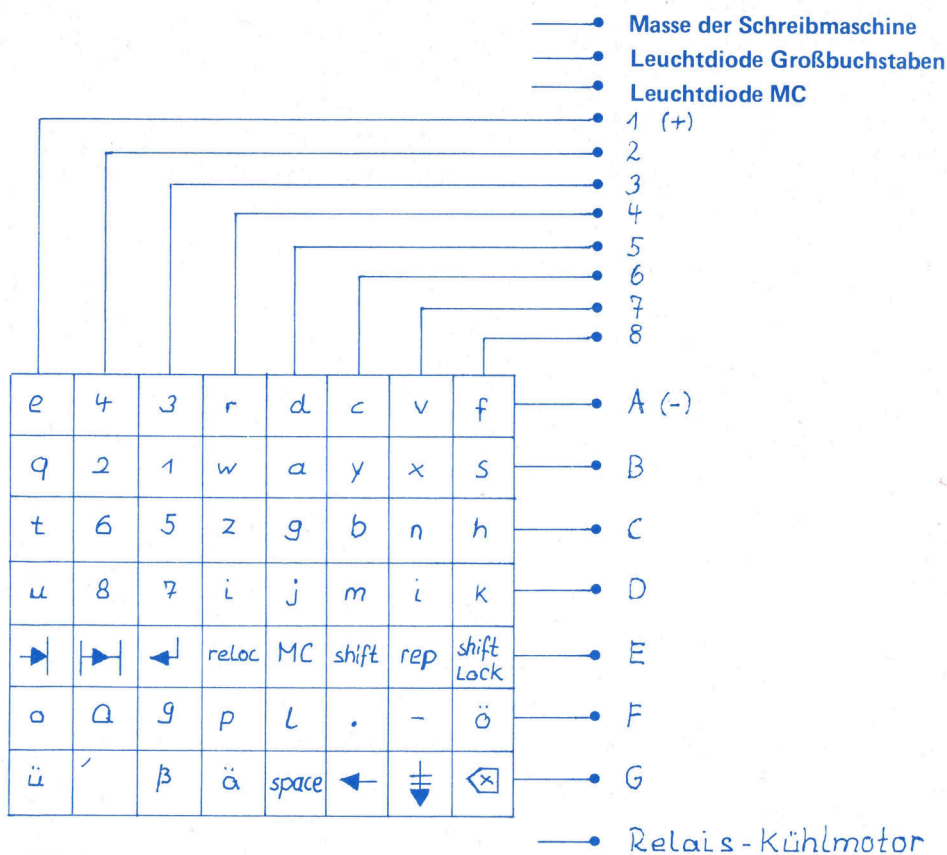


Abb.1.
Tastaturmatrix der Typenrad Schreibmaschine

koppler IL74 genutzt. Durch die Verwendung von zwei 3 zu 8 Binärcodecodern (Demultiplexer) 74LS138 können die 15 Schalter mit nur 6 Bits des Userports angesteuert werden. Es funktioniert so: Durch Anlegen eines Bitmusters an die drei Eingänge des 74LS138 wird einer der 8 Ausgänge low. Damit wird auch nur einer der pnp-Transistoren leitend. Bei den Optokopplern wäre es aber genau umgekehrt. Ein Optokoppler würde sperren und alle anderen würden leiten. Um dies zu verhindern, sind zwischen die Ausgänge des unteren Demultiplexers und den Kopplern noch schaltet.

Das eben vorgestellte Interface ist zwar ganz gut — es konnte mit Erfolg bei der oben erwähnten Brother 8300 und bei einer Olympia Schreibmaschine eingesetzt werden — aber ziemlich aufwendig.

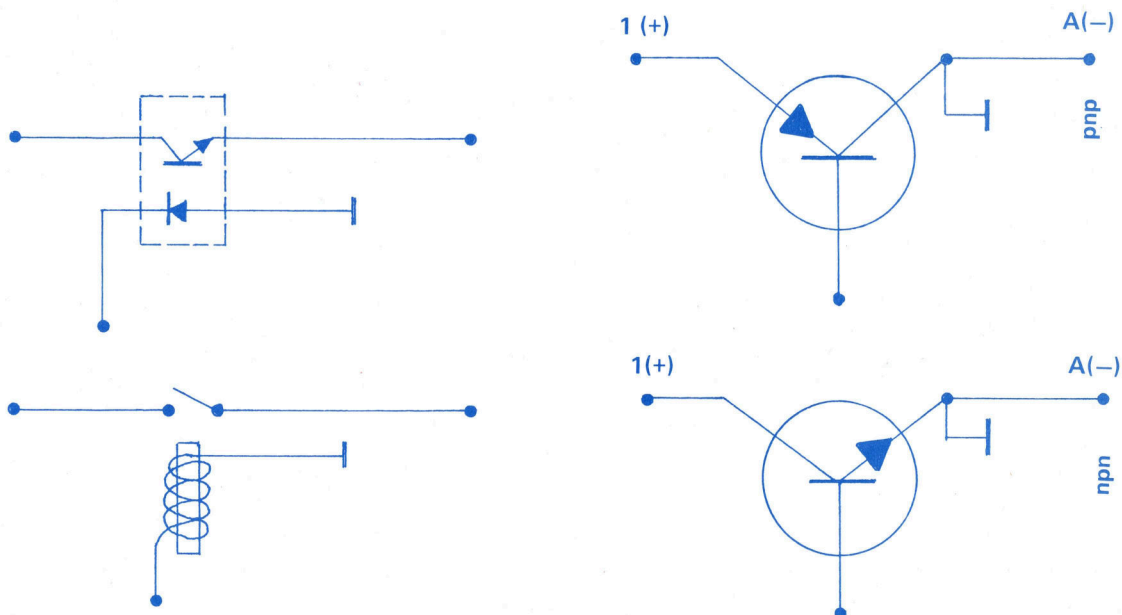


Abb2.:
Prinzip der Transistoren und des Optokopplers

Durch Probieren konnte jedoch noch eine simple Lösung entwickelt werden, die allerdings nur bei der Brother-Maschine funktioniert. Das Interface besteht nur aus zwei IC's 74LS138, 8 npn-Transistoren (zum Beispiel 2N2221A) und einem Trimmer von 10K. Über PB0 bis PB5 des Userports wird der zu druckende Buchstabe ausgewählt. PB6 und PB7 steuern einen Transistor an, der die gesamte Stromversorgung des Interface einstellt. Sind diese Bits auf low, so kann man die Tastatur der Schreibmaschine benutzen, als ob kein Interface vorhanden wäre.

Ein Beispiel:

poke56579,255 Daten-
richtungsregister
poke56577,32 32 ist der
Code für ein 'e'
poke56577,32+64+128

das 'e' wird nun
gedruckt

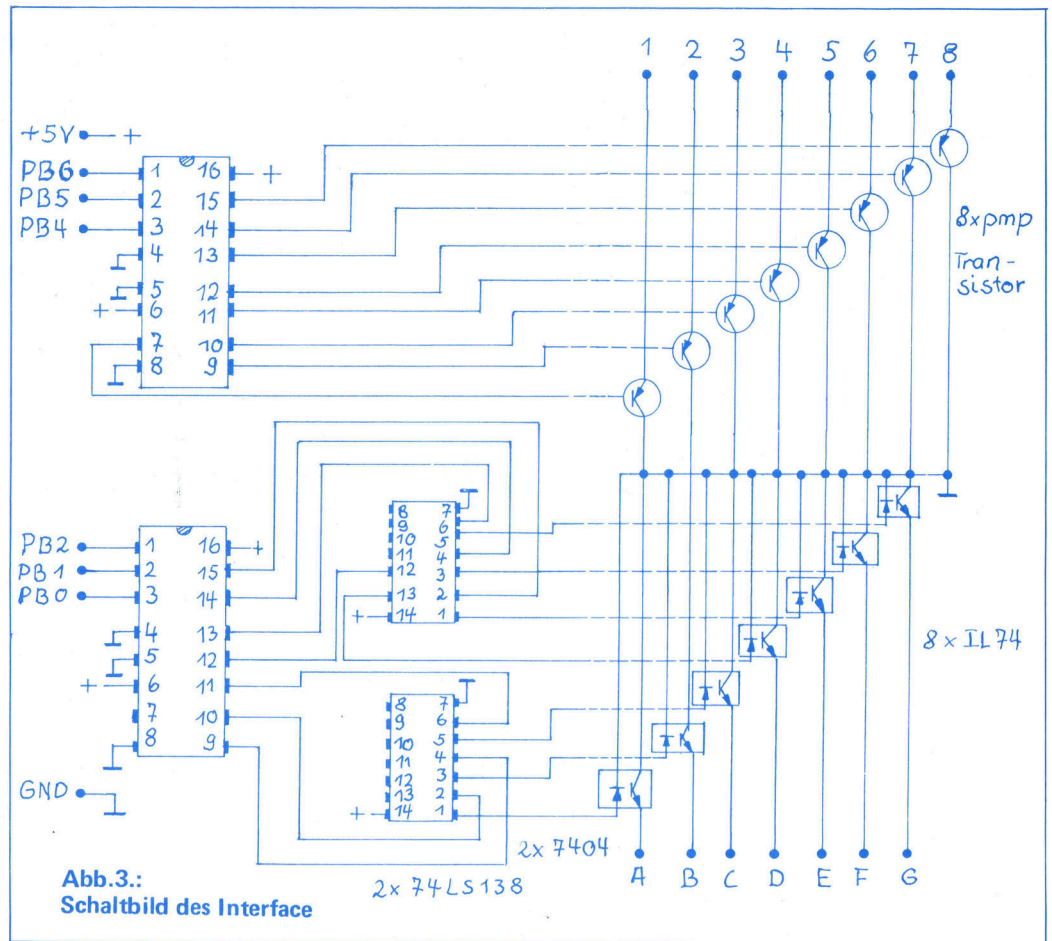


Abb. 3.:
Schaltbild des Interface

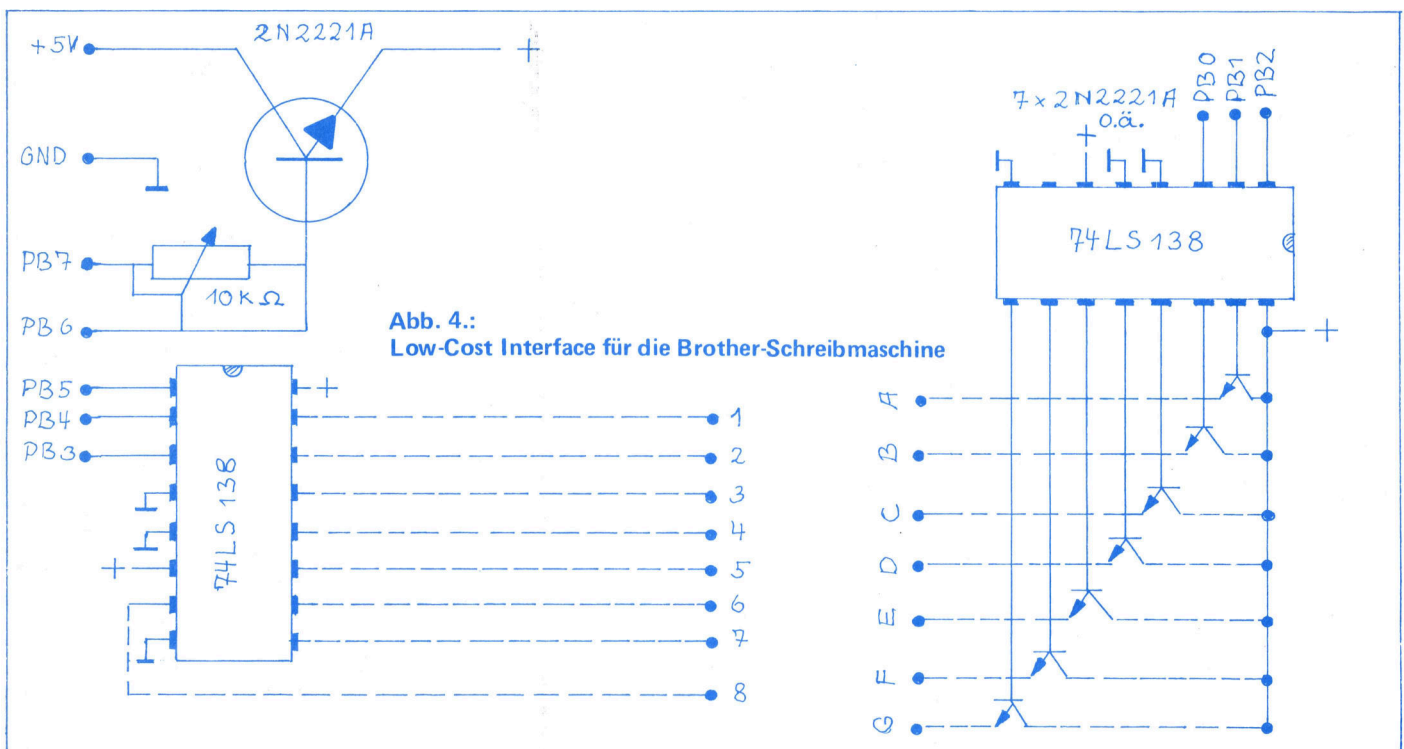


Abb. 4.:
Low-Cost Interface für die Brother-Schreibmaschine

Auf diese Art wäre es ein recht mühsames Unterfangen, einen ganzen

Brief zu poken – Entschuldigung – zu schreiben. Aus diesem Grunde

habe ich auch ein kleines Textverarbeitungsprogramm geschrieben, das

ich in einer der nächsten Ausgaben vorstellen werde.

MANNESMANN
TALLY

Computerperipherie der Mannesmann-Tally- Klasse



Drucker in jeder Leistungsklasse, zwischen 100 Zeichen/Sekunde und mehr als 600 Zeilen/Minute Druckgeschwindigkeit, für Home Computer, Personal Computer, Bürocomputer, Textsysteme, EDV-Systeme. Drucker mit vielen Zusatzausstattungen wie Einzelblattzufuhr, Stapelzufuhr, Mehrfarbdruck, Etikettendruck oder Schneidevorrichtung. Drucker mit vielen, wählbaren Schriftarten.

mannesmann technologie 

Schnellschrift, Schönschrift, OCR-Schrift, Plakatschrift oder Barcode.

Mannesmann-Tally-Klasse – das sind erfolgreiche Drucker, kompatibel zu erfolgreichen Computersystemen und dazu der qualifizierte, kundennahe Service.

Mannesmann Tally GmbH

Bottroper Str. 10, 7000 Stuttgart 50
Telefon 07 11 / 5 03 90, Telex 7 254 672