

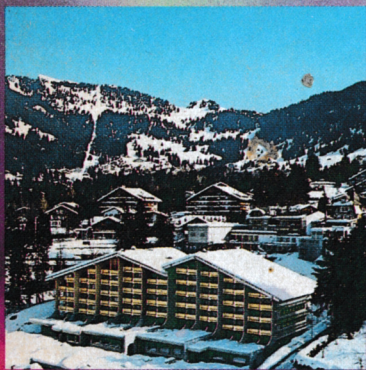
Nr. 12 Dezember 1985

DM 5,-

COMPUTER

TEAM

Das Kommunikations-Magazin für Computer-Fans.



Große Verlosung

1 Woche Schweiz

für zwei Personen



ATARI-TEAM

Das BASIC

SCHNEIDER-TEAM

TEAM OPTIMIZER

Macht BASIC schneller

WORD PROZESSOR

Profi-Textverarbeitung



COMMODORE-TEAM

TEAM ANIMATOR

von R. Leinfellner

Superlisting
Kalorienwächter
für Atari

ocean



**SOFTWARE
PROJECTS**

**ULTIMATE
PLAY THE GAME**



They sold a

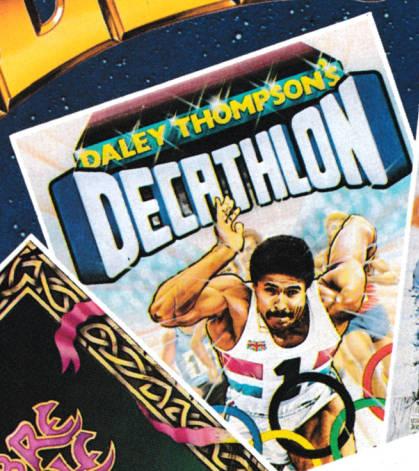
MILLION

They sold a

MILLION

EINE
GIGANTISCHE
ZUSAMMENSTELLUNG

Veröffentlichungsdatum
1. NOVEMBER
1985



Erhältlich für
**COMMODORE
64/128**

SPY HUNTER
DALEY THOMPSON'S DECATHLON
JET SET WILLY · STAFF OF KARNATH

SPECTRUM

SPY HUNTER
DALEY THOMPSON'S DECATHLON
JET SET WILLY · SABRE WOLF

SCHNEIDER

DALEY THOMPSON'S DECATHLON
SABRE WOLF
JET SET WILLY · ROCCO

IN DER COMMODORE 64-
VERSION WIRD SABRE WOLF DURCH
STAFF OF KARNATH

IN DER SCHNEIDER-VERSION WIRD SPY
HUNTER DURCH ROCCO ERSETZT

**U.S. GOLD
(GERMANY) LTD.**

Ander Gumpgesbrücke, 22
D-4044 Kaarst 2, Holzbuttg. Tel: 02101/6 84 99 + 6 85 61

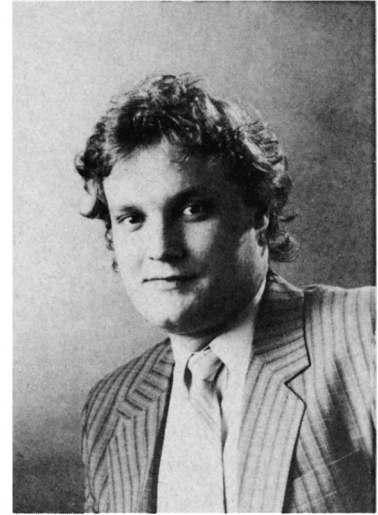


Telex: 17/2101 325 rush

Sehr geehrte Leserinnen,
sehr geehrte Leser,

der erste Mikrocomputer mit einem RAM-Speicher von 1 MByte, der MEGA-Atari, ist da. Und dieser zu einem Preis von unter 3000,— DM, inklusive Bildschirm, Tastur, Floppy und Maus.

Viele von Ihnen werden sich sicher noch an die Anfänge des Mikrocomputers erinnern. Die ersten Computer waren Einplatinenrechner mit einer HEX-Tastatur und einer mehrstelligen Siebensegmentanzeige. Zu erwähnen ist der wohl bekannteste, der KIM I von Commodore.



Commodore war es auch, die den ersten Mikrocomputer im Gehäuse produzierte, den PET 2001. Nach diesen Rechnern, die für den Anfänger noch viel zu teuer waren, kam der legendäre ZX80 von Clive Sinclair.

Damals sprach man von 1 K RAM oder 8 K; 16 K waren schon Spitzenwerte. Auch der erste Erfolgsschlagler von Commodore, der VC20 hatte nur ganze 3,5 KByte Schreib-/Lesespeicher.

Der Commodore 64 war dann mit 64 K RAM die echte Sensation.

Aber 64 K sind nichts gegen 512 KByte Speichergrößen, die nicht einmal serienmäßig im IBM PC zu finden sind. Trotzdem wurde Anfang '85 der Atari 520 ST mit 512 KByte vorgestellt.

Doch das war nur der Anfang. Der 520 ST ist tot! Es lebe der 520 ST Plus!

Auf der Systems in München konnte man ihn, den Atari 520 ST Plus, sehen und anfassen. (Dieses hat allerdings nichts mit der Lieferfähigkeit zu tun.) Er wartet mit einem Megabyte RAM auf und das zu einem Preis von nur 2998,— DM. Der „alte 520 ST“ heißt jetzt 260 ST und ist ca. 600,— DM billiger geworden.

Wenn man sich vor Augen hält, daß der Prozessor des Atari und des neuen Commodore Amiga 16 MB adressieren kann, dann werden wir Weihnachten 1986 sicher einen 10 MByte-Rechner unter dem Weihnachtsbaum erblicken.

Ein frohes Fest wünscht
Ihnen
Ihr

A large, stylized handwritten signature in black ink.

(Joachim Günster)

Impressum

Herausgeber: Joachim Günster.
 Chefredakteur: Joachim Günster.
 Titelfotos: Professional Photo, Koblenz.
 Titelgestaltung: Gesellschaft für Vertrieb und Werbegestaltung, Koblenz.

Technische Herstellung:
 Druckhaus Dierichs, Kassel
 Datenkonvertierung, Fotosatz:
 Dinges + Frick, Wiesbaden.

Vertrieb Handelsauflage: Inland (Groß-, Einzel- und Bahnhofsbuchhandel) sowie Österreich und Schweiz: Verlags-Union, Friedrich-Bergius Str. 20, 6200 Wiesbaden, Tel.: (0 61 21) 26 60

Manuskripteinsendungen: Manuskripte und Programmlistings werden gerne von der Redaktion angenommen. Sie müssen frei sein von Rechten Dritter. Sollten Sie auch an anderer Stelle zur Veröffentlichung oder gewerblichen Nutzung angeboten werden, so muß dies angegeben werden. Mit der Einsendung von Manuskripten gibt der Verfasser die Zustimmung zum Abdruck in von dem Verlag COMPUTER TEAM Joachim Günster herausgegebenen Publikationen und zur Vervielfältigung der Programmlistings auf Datenträger. Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte und Listings wird keine Haftung übernommen.

Anzeigen im Kleinanzeigenteil: Private Kleinanzeigen mit maximal 5 Zeilen Text DM 5,- je Anzeige
 Gewerbliche Kleinanzeigen: DM 11,- je Zeile Text. Auf alle Anzeigenpreise wird die gesetzliche MWSt. jeweils zugerechnet.

Erscheinungsweise: COMPUTER TEAM, Das Kommunikations-Magazin für Computer-Fans, erscheint monatlich, Ende des Vormonats.

Bezugspreise: Das Einzelheft kostet DM 5,-.

Urheberrecht: Alle im COMPUTER TEAM erschienenen Beiträge sind urheberrechtlich geschützt. Alle Rechte, auch Übersetzungen, Reproduktion gleich welcher Art, ob Fotokopie, Mikrofilm oder Erfassung in Datenverarbeitungsanlagen, nur mit schriftlicher Genehmigung des Verlages. Anfragen sind an Joachim Günster zu richten. Für Schaltungen und Programme, die als Beispiele veröffentlicht werden, können wir weder Gewähr noch irgendwelche Haftung übernehmen. Aus der Veröffentlichung kann nicht geschlossen werden, daß die beschriebenen Lösungen oder verwendeten Bezeichnungen frei von gewerblichen Schutzrechten sind.

Copyright 1985 Verlag COMPUTER TEAM Joachim Günster.

Anschrift für Verlag, Redaktion, Vertrieb, Anzeigenverwaltung und alle Verantwortlichen: Verlag COMPUTER TEAM Joachim Günster, Mühlenstraße 12, 5431 Boden, Telefon (0 26 02) 6 00 80.

SCHNEIDER TEAM

Seite

3-dimensionales Plotten mit dem CPC

Funktionen können Sie nun ohne Probleme 3-dimensional auf den Bildschirm zaubern.

39

TEAM-Texter

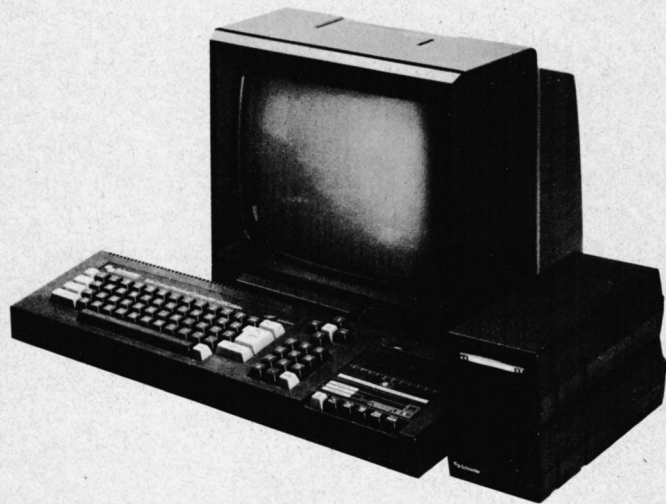
Eine neue Super-Textverarbeitung für den Schneider finden Sie in diesem Heft. Professionell mit Pull-DOWN-Menues.

47

Grafikkurs für den Schneider Teil 3

Um die Illusion von Bewegung und das Zeichnen von Kreisen und Ellipsen geht es im 3. Teil unseres Kurses.

54



Insider-TEAM

Wie man mehr aus seiner Floppy machen kann — und um verborgene Diskettenbefehle geht es in unserem Insider-TEAM.

58

Die Unterschiede zwischen 464 und 664

Wie man Programme an die beiden Rechner anpaßt und wie das Programm den Rechner erkennt, lesen Sie ab Seite

61

Der Schneider CPC 6128

Wo sind die Unterschiede, was kann er mehr, was ist nicht so gut?

63

512 KB für Ihren 464

Die Vortex-Speichererweiterung im Test

65

Machen Sie Ihre Programme schneller

Dieses Programm optimiert Ihre BASIC-Programme.

67

COMMODORE TEAM

Testbericht Commodore PC-128

Drei Betriebsarten in einem Rechner. Ein völlig neuer Stern fiel vom Computer-Himmel, Ausgeschlafene Konstrukteure bieten eine kompatible Maschine zum Vorgängertyp. Sollte auch dies bei IBM abgeguckt sein? Denn bei dieser Firma herrscht seit vielen Jahren eine Kompatibilität nach unten. Wie kompatibel der neue PC-128 ist und was drei Betriebsarten bringen, lesen Sie in diesem Bericht.

TEAM Animator

Exklusiv von R. Leinfellner. Passend zum TEAM Spriter der TEAM Animator. Das neue Superlisting, ein Direktimport aus England, finden Sie ab Seite

Repeat-Funktion

Tastatur-Repeat auf PC-128 und C-64 einfach durch POKEs

GRUNDLAGEN

COMPUTER TEAM-Projekt

Ein Animator für Z 80-Systeme umfaßt unser Dezember-Projekt. Lesen Sie, warum man einen Computer langsamer macht.

Mikroprozessoren kurz vorgestellt

In dieser neuen Serie wird monatlich ein Prozessor unter die Lupe genommen. Dieses Mal der Z 80 ab Seite

Die 68 000 Dimension

Um Coprozessor-Möglichkeiten geht es in der dritten und letzten Folge unserer Serie.

Compiler Technik

Der Compiler im Selbstbau 3. Teil

Logo Programmierkurs

Den ersten Teil unseres Programmierkurses finden sie ab Seite

Warum 16 Bit?

Mit dieser Frage beschäftigt sich unser Autor ab Seite

Seite

31

35

38

10

9

11

14

18

22

ATARI TEAM

Kalorien-Wächter

Planen Sie Ihr Abnehmem mit dem Computer

BASIC mit Bremse

Was leistet der Atari ST-Basic-Interpreter?

SPIELE TEAM

Neues vom Sordware Spy

Der neue Stern am Computer-Himmel: SS 63

Skyfox und Hotel

Zwei Superspiele von Arioasoft

MSX / TEAM

Der Computerstandard MSX

Über die MSX-Konferenz in Berlin lesen Sie ab Seite

ENTERPRISE TEAM

Strukturiertes Programmieren in BASIC

Einen Test des hervorragenden Enterprise-BASIC-Interpreters finden Sie ab Seite

INSERENTEN

dieser Ausgabe

Cordes, Willich	Seite 81
Data Service, Bonn	Seite 42
D & G Datentechnik, Koblenz	Seite 72
Dobbertin, Brühl	Seite 30
EDV, Worms	Seite 42
Effertz, Bedburg	Seite 81
Enterprise, München	Seite 51
Flesch, Neustadt	Seite 81
Heimcomputer-Shop, Delmenhorst	Seite 72
Heyns, Essen	Seite 30
Integral Hydraulik, Düsseldorf	Seite 30
C. ITHO Electronics, Düsseldorf	3. Umschlagseite
Janke, Wuppertal	Seite 36
Mastertronic, Soest	Seite 57
Media Plast, Dortmund	Seite 41
Orgasoft, Villingen	Seite 42
Palace Software, London	Seite 51
Probst, Duisburg	Seite 30
Soft Learning GmbH, München	4. Umschlagseite
Software Library, Frankfurt	Seiten 42, 81
Sun Soft, Eschwege	Seite 72
US-Gold, Kaarst	2. Umschlagseite

Seite

21

27

73

74

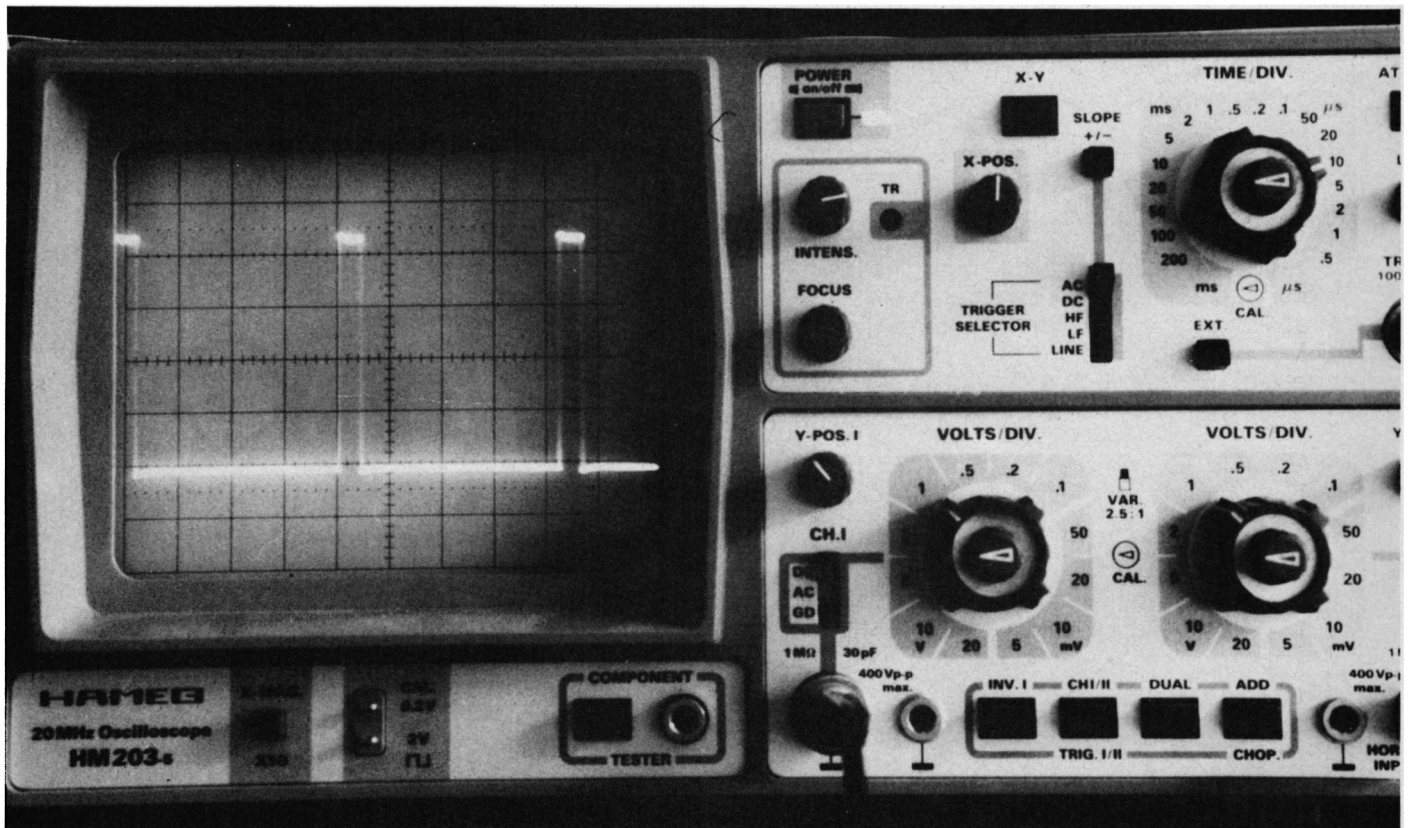
75

78

DIGITALIS 1

Manipulieren Sie Ihren Systemtakt

Wer kennt nicht das Problem: Es müssen Screenfotos angefertigt werden, jedoch erzeugen die schnellen Bewegungen der Objekte Bewegungsunschärfen. Kürzere Belichtungszeiten als eine achtel Sekunde dürfen nicht verwendet werden, da auf den Fotos sonst Balken erscheinen. Man benötigt folglich ein Gerät, das die Bildschirmbewegungen einfriert. Und diese Lösung heißt Digitalis 1! Doch damit ist die Leistungsfähigkeit dieses Gerätes noch lange nicht erschöpft.



Unser Bild zeigt einen typischen Systemtakt

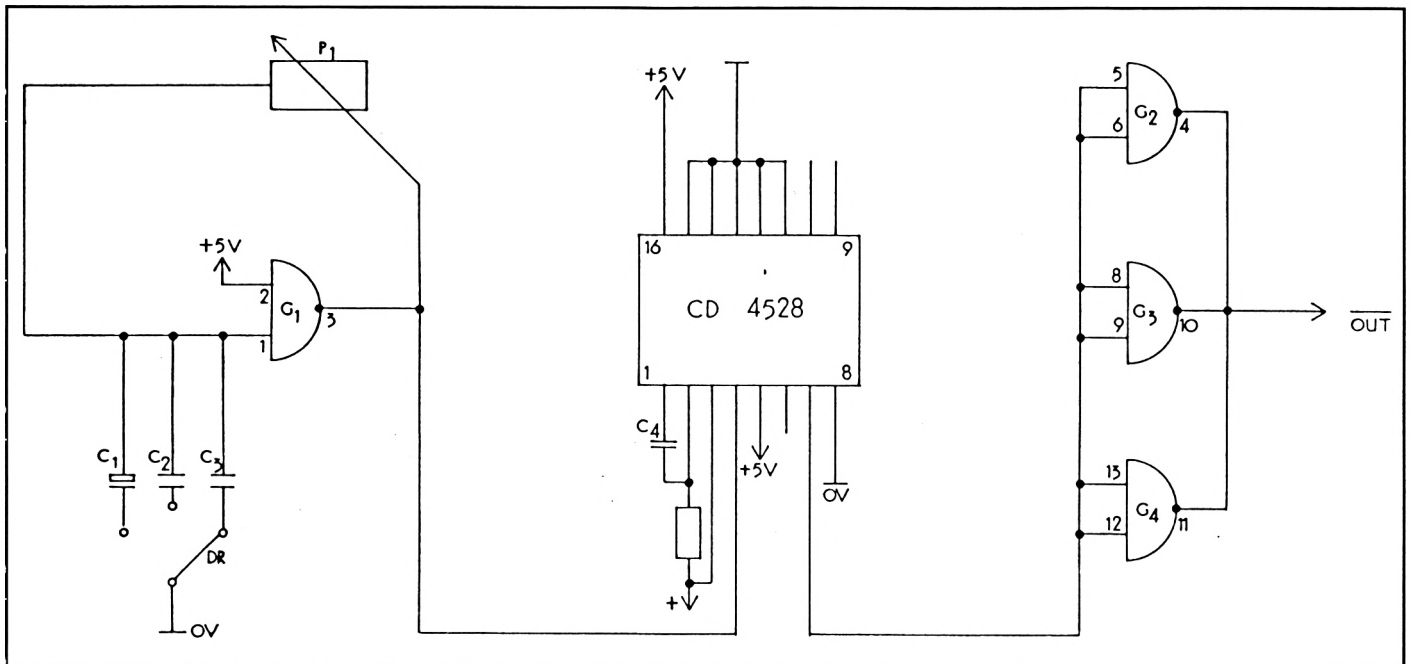
Allgemein dient es zur stufenlosen Herabsetzung der Arbeitgeschwindigkeit Ihres Spectrums (16 oder 48 Kb) bis zum Mindestfaktor 21000! Ebenfalls können alle Z 80-Systeme damit betrieben werden. Doch wozu wird die variable Geschwindigkeit noch benötigt? Eine weitere Anwendung ergibt sich wahrscheinlich bei der Fehlersuche in Programmen. Hier kann man Z 80-Systeme veranlassen, alle Programmschritte sehr langsam auszuführen, so daß man Fehler sofort nach Abarbeitung eines Befehls lokalisieren kann.

Eine andere Anwendungsmöglichkeit bietet sich den Spielernaturen. Durch langsamere Spielabläu-

fe ergeben sich hervorragende Übungsmöglichkeiten, so daß das Gerät in diesem Fall als Trainer fungiert. Bekanntlich erlaubt kaum ein Spiel die Möglichkeit zu üben, da die Geschwindigkeit der Abläufe nie festgelegt werden kann. Abhilfe schafft hier ausschließlich diese Schaltung. Aber auch bei der Steuerung von Peripherie, wie zum Beispiel Roboter, kann Digitalis 1 den Datentransfer regeln, ohne daß softwaremäßige Maßnahmen ergriffen werden müssen. Es ist übrigens funktionskompatibel zu der „Weltneuheit Slow-Motion“, kostet allerdings nur ein Drittel bis ein Fünftel davon!

Funktionsweise

Gatter 1 des Schmitt Triggers CD 4093 erzeugt in Verbindung mit dem Potentiometer und einem der Kondensatoren C1-C3 eine Frequenz von etwa 8-250000 Hertz. Diese wird durch den Drehstufenschalter und den Potentiometer festgelegt. Da die Impulsbreite viel zu groß ist (sie legt fest, wie lange der Mikroprozessor pro Takt arbeiten soll), muß sie mit Hilfe einer Impulskürzungsschaltung auf etwa 4,5 Mikrosekunden reduziert werden. Diese konstante Zeit wird durch C4 und R1 und einem wie-



dertriggerbaren Monoflop CD 4528 festgelegt. Bei jeder positiven Flanke des Taktgenerators erzeugt dieser Monoflop einen 4,5 Mikrosekunden langen Impuls. Je mehr pro Zeiteinheit nun folgen, also je höher die Frequenz ist, desto länger muß der Prozessor pro Sekunde arbeiten, und desto schneller wird er. Erreicht man allerdings eine Frequenz von etwa 222 KHz, so gibt der CD 4528 ein Dauersignal (High-Pegel) ab, das den Prozessor auf Höchstgeschwindigkeit rechnen läßt, also so, als wenn der Geschwindigkeitsregler nicht angeschlossen wäre.

Die parallel geschalteten Gatter 2-4 sorgen für einen kräftigen Ausgangsstrom, da diese beim Spectrum einen High-Pegel mit fünf Kiloohm Innenwiderstand auf ein eindeutiges Lowsignal ziehen müssen. Und ein CMOS Gatter wäre dabei hoffnungslos überfordert! Da der Stromverbrauch dank der CMOS-Gatter maximal nur 11 mA beträgt, kann selbst ein stark belastetes Computernetzteil dieses Gerät mit Strom versorgen.

Aufbau

Der Schaltplan und das Layout zeigen den Aufbau, der wegen der wenigen Bauteile keine Schwierigkeiten bereiten dürfte. Die drei Leitungen werden an Plus, Minus und Busrequest angeschlossen. Bitte fassen Sie die beiden CMOS-ICs

nicht an, wenn Sie sich nicht vorher an einer Heizung oder dem Schutzkontakt der Steckdose entladen haben. Diese integrierten Schaltungen sind trotz Schutzdioden an den Eingängen etwas empfindlich gegenüber elektrostatischen Ladungen. Sind diese Bausteine erst einmal in der Schaltung eingebaut, können Sie getrost meine Warnung vergessen. Ein kleines Problem bereitet der integrierte Baustein CD 4093. Je nach Hersteller schwankt die erzeugte Frequenz um den Faktor zwei. Die in Klammern stehenden Kapazitätswerte beziehen sich daher auf einen Schmitt Trigger von der Firma SGS.

Anschluss

Da ein Portstecker etwa 18 DM kostet und außerdem nicht erlaubt, weitere Interface aufzustecken, (Verteiler ergeben meistens nur Wackelkontakte, die zum Systemabsturz führen), liegt aus diesen Gründen die Lösung nahe, das Gerät mit einem schon vorhandenen Interface zu verbinden. Durch diese Idee sparen Sie cirka 50 DM, da ein zusätzlicher Portstecker entfällt, ebenso der Kauf eines unsicheren Verteilers. Ansonsten verweise ich auf das Spectrum-Handbuch, Seite 160. Ein Gehäuse für das Gerät ist nicht notwendig, da keinerlei Gefahr besteht, durch Anfassen der Platine den Rechner durch Elektrostatik zu zerstören!

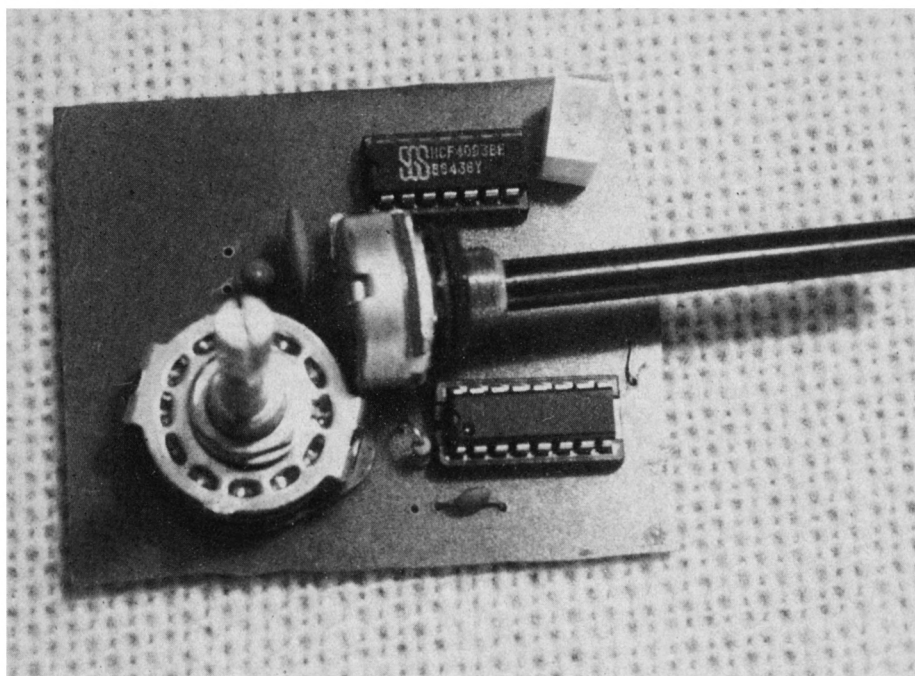
Bauteilliste

- 1 Monoflop CD 4528 = IC2
- 1 Schmitt-Trigger CD 4093 = G1-G4 = JC 1
- 1 Elektrolytkondensator 1 uF, 5V (2,2 uF) = C1
- 1 Kondensator 0,1 uF (0,15uF) = C2
- 1 Kondensator 1,5 nF (2,2 nF) = C3
- 1 Kondensator 1 nF = C4
- 1 Metallfilmwiderstand 5,6 Kiloohm, 1% Toleranz = R1
- 1 Potentiometer 100 Kiloohm, Positiv logarithmisch = P1
- 1 Drehstufenschalter 3*3 (hat 12 Anschlüsse) = DR1
- Je 1 IC-Fassung 16 polig und 14 polig
- 1 Mikromodul Steckverbinder Typ P 25/3

Alle aufgeführten Bauteile, insbesondere die mechanischen, sind erhältlich bei der Firma Völkner Elektronik, Postfach 5320 in 3300 Braunschweig. Die Kosten belaufen sich auf etwa 20 DM. Möchten Sie gerne dieses Gerät besitzen, können dieses aber wegen mangelnder Elektronikkenntnisse nicht nachbauen, so können Sie sich in diesem Fall an die Redaktion wenden, da noch ein paar wenige Testgeräte abgegeben werden können.

Fehler und ihre Beseitigung

Um die Wirkung bei den unvermeidlichen Bauteiltoleranzen zu te-



Unser Bild zeigt das fertig aufgebaute Gerät

sten, wurden acht Exemplare von Digitalis I hergeteilt. Dabei versagte nur ein einziges Gerät infolge von Toleranzen, so daß es insgesamt als recht nachbausicher bezeichnet werden kann. Trotzdem können in wenigen Fällen mal Störungen auftreten, die aber mit den nachfolgenden Hilfestellungen leicht beseitigt werden können.

Sollte es bei ziemlich schnellem Betrieb zu Systemabstürzen kommen, so ist wegen hoher Bauteiltoleranzen von R1 und C4 die Impulsbreite zu klein. Erhöhen Sie den Widerstandswert von R1 dann auf 6,8 Kiloohm. Kommt es bei langsamstem Betrieb zum Systemabsturz, muß der Elektrolytkondensator C1 in seiner Kapazität gesenkt werden, zum Beispiel von 1 uF auf 0,68 uF. Wer ein Z 80-System besitzt (keinen Sinclair Spectrum), kann einmal den Elektrolytkondensator in seiner Kapazität erhöhen, so daß der Rechner noch langsamer wird. Treten allerdings hierbei Störungen auf, verwenden Sie dann besser die Originalwerte.

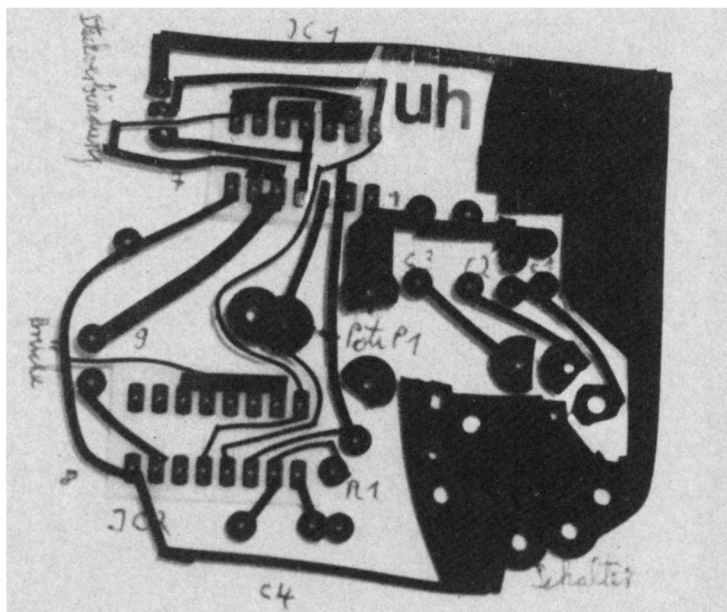
Bedienung von Digitalis I

Auf der Platine sehen Sie einen liegenden Regler (P1), der zur stufenlosen Geschwindigkeitsänderung der Schaltung dient. Drehen Sie dieses Potentiometer im Uhrzei-

bindung lösen müssen, empfiehlt es sich, Regler und Schalter an den rechtesten Einstellpunkt zu bringen. Nur so verhält sich das Modul, als ob es nicht angeschlossen wäre. Diese Stellung ist auch beim Laden und Saven von Programmen erforderlich.

Schaltplan und Layout

Das Gerät hat sich in der Praxis bisher als völlig problemlos erwiesen. Schon in einer der nächsten Ausgaben werden wir eine geringfügig modifizierte Version für 6502 (6510)-Prozessoren vorstellen, so daß auch Commodore- und Atari-besitzer demnächst diesen Geschwindigkeitsregler verwenden können. KUH



So kann das Schaltungs-Layout aussehen

Grob- und Feineinstellung

gersinn, so wird der Computer immer schneller. Ferner sehen Sie einen stehenden Stufenschalter, der drei Schalterstellungen besitzt, die den Geschwindigkeitsbereich festlegen. Auch hier gilt, daß die rechte Stellung den Computer am schnellsten arbeiten läßt. Man kann zusammenfassend sagen: Der Drehstufenschalter dient zur Grobeinstellung des Geschwindigkeitsbereiches, der andere Regler zur stufenlosen Feineinstellung. Damit Sie bei Nichtgebrauch des Gerätes nicht immer die Steckver-

**DIE
JANUAR-
AUSGABE
GIBTS
AM
27.12**

MIKROPROZESSOREN KURZ VORGESTELLT

Der Z 80

Unter den Mikroprozessoren der 8 Bit-Klasse dürfte der Z 80 von Zilog einer der leistungsfähigsten sein. Zusammen mit dem 6502 ist er in Heim- und Personalcomputern am häufigsten vertreten. Und das zu Recht!

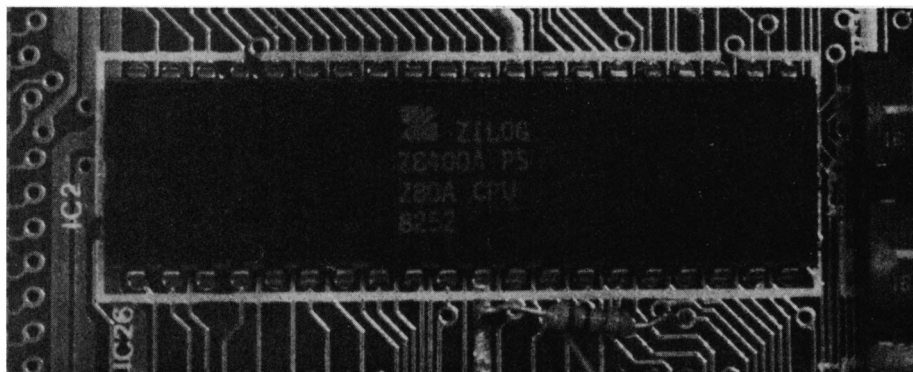
Für bestimmte Operationen stellt er 16 Bit-Register zur Verfügung, in denen die Einzelregister zu Paaren zusammgelegt werden (bc, de,

Doppelregister

hl). Mit einem einzigen Befehl können 16 Bit-Zahlen addiert und subtrahiert werden, wozu andere Prozessoren, wie der schon erwähnte 6502 (6510,8502), ungefähr doppelt so viele Kommandos benötigen. Natürlich wird dann auch zweimal soviel Speicherplatz verbraucht, so daß der Z 80 als speicherplatzschonend bezeichnet werden kann. Allerdings können die Doppelregister nur für die Addition, Subtraktion, Inkrementieren (= +1) und Dekrementieren (= -1) verwendet werden. Trotzdem, ein Hauch von Komfort.

Registersatz

In puncto Registervorrat (Register sind die Arbeitsplätze des Prozessors, an denen alle Daten verarbeitet werden) ist der Z 80 in der 8 Bit-Generation führend. Neben den Registern b,c,d,e,a,f,ix,iy, können über die Anweisungen 'ex af,af'' und 'exx' der Zweitregistersatz mit den Registern b'c',d'e',h'l',a'f', beziehungsweise die Paare bc', de' und hl' (f ist das Statusregister) aufgerufen werden. Jedoch ist es nicht möglich, beide Registersätze gleichzeitig zu benutzen. Was bedeutet aber ein solch gewaltiger Vorrat an Registern? Nun, da genug Arbeitsplätze vorhanden sind, müssen die Registerinhalte nicht jedesmal auf dem Stapel abgelegt werden, so daß wiederum Anweisungen eingespart werden und insbesondere an Ar-



Unser Bild zeigt den Z 80 A Prozessor

beitsgeschwindigkeit gewonnen wird. Leider besitzt das Zilog-Produkt auch einen gewaltigen Nachteil.

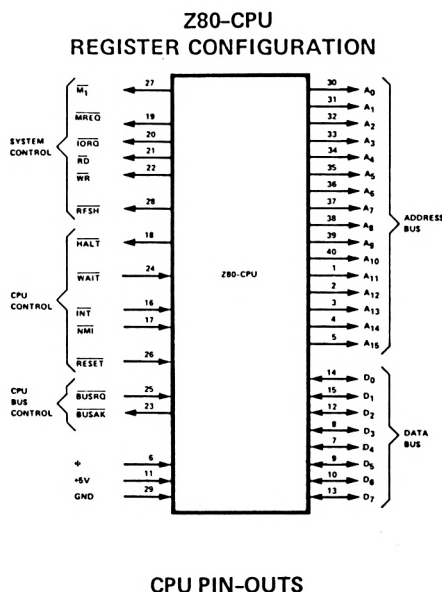
Befehlsvorrat

Sind bei den meisten Prozessoren 70 Befehle die Grenze, so schießt der Z 80 mit 159 Anweisun-

gen den Vogel ab. Der Grund hierfür liegt in der Eigenart der vielen Register. Sie sind nicht gleichberechtigt. So werden Summen und Differenzen nur in den Registern 'hl' und 'a' gebildet. Ebenfalls können nur die Paare 'de' und 'hl' vertauscht werden. Nicht nur die unterschiedliche Verwendbarkeit der Register ist verantwortlich für den großen Befehlssatz, sondern auch die zusätzlichen 16 Bit-Anweisungen und insbesondere die Macrobefehle. Die Macrobefehle sind in der 8 Bit-Klasse wohl einzigartig. Sie dienen hauptsächlich zum Blockladen. Sie sparen nicht nur Speicherplatz, sondern erhöhen auch die Verarbeitungsgeschwindigkeit!

Adressierung

Auch in diesem Punkt kann der Z 80 als fortschrittlich bezeichnet werden. Die beiden 16 Bit breiten Indexregister 'ix' und 'iy' ermöglichen ohne Umwege die Adressierung der gesamten 64 Kilobyte. Wer einmal den 6502 mit seinen acht Bit-Adressregistern programmieren mußte, wird Z 80 schätzen.



Taktfrequenzen

Auffallend sind die hohen Taktfrequenzen des Z 80. Sind es beim 6502 etwa ein bis zwei Megahertz, so muß der Z 80 mit 4 MHz getrieben werden. Der Grund hierfür liegt in der Menge der Taktzyklen, die ein Mikroprozessor benötigt, um einen Befehl abzuarbeiten. Reichen beim 6502 in einigen Fällen zwei Zyklen aus, so sind es beim Z 80 immerhin schon vier. Um aber trotzdem gleiche Verarbeitungsgeschwindigkeiten zu erzielen, muß folglich die Taktfrequenz doppelt so hoch angesetzt werden. Die hohen Frequenzen können bei bestimmten Speichern zu Problemen führen, so daß der 6502 in diesem Punkt unproblematischer ist. Daraus ergibt sich, daß die Zilog CPU bei gleicher Frequenz dem 6502 in Sachen Schnelligkeit hoffnungslos unterlegen ist, so etwa um den Faktor 2,5.

Insgesamt aber ist der Z 80 in der 8 Bit-Klasse führend und weist

Z 80 führend

schon einige wenige Merkmale der 16 Bit-Generation auf, wie etwa Macrobefehle und 16 Bit-Register. Besitzer eines Schneider- oder Sinclair-Computers, die in Assemblersprache programmieren müssen, werden trotz dieser nicht ganz einfachen Sprache ihre Freude am Z 80 haben. UH



BÜCHERECKE

Titel: 68000/68008 Maschinensprache, Autor: Ulrich Schollwöck, Verlag: iwt, 330 Seiten, Preis: 48,— DM

Unter dem relativ geringen Angebot an Literatur über den Motorola Prozessor 68000 gehörte dieses Buch zu den ersten auf dem Markt. Es behandelt sehr ausführ-

68000/68008

lich die allgemeinen Grundlagen der Assemblersprache. Da wird zum Beispiel dem Leser erst einmal die Umwandlung von Dezimalzahlen in andere Zahlensysteme beigebracht. Anschließend rechnet man mit Dualzahlen und bekommt damit Einblick in die Problematik der binären Arithmetik. Schnell lernt man dabei Begriffe wie Zweierkomplement oder Flags kennen. Sitten diese notwendigen Grundlagen, wird darauf folgend der Aufbau des Prozessors gezeigt, ohne daß man mit zu vielen Hardwaregrundlagen gequält wird. So ist man an dieser Stelle über den gesamten Registersatz informiert und nun in der Lage, die Maschinenprogrammierung zu beginnen. Es werden hauptsächlich die 4 Grundrechenarten als Beispiele verwendet. Dabei steigert sich langsam ihr Niveau, indem die einzelnen Operanden ihre Stellenan-

übersichtlich. Insgesamt hinterläßt diese Fachliteratur einen hervorragenden Eindruck und ist Einsteigern und Profis zu empfehlen, letzteren insbesondere als ausführliches Nachschlagewerk. UH

Gegen den Frust

Wirklich witzige Computer-Bücher können anscheinend nur Amerikaner schreiben. Dank der vortrefflichen Übersetzung von Vivian Marx kommt der Witz auch an in dem Buch von John Bear: Computerfrust — Ein Vermeidungshelfer, Rowohlt, 286 Seiten, 14,80 DM

Aus seiner jahrelangen Erfahrung und aus dem Material, das ihm von Freunden zugetragen worden ist, hat Bear eine lange Liste von Fehlern zusammengetragen. Diese können entweder vom Computer selbst, von seinem Besitzer oder von beiden zusammen begangen worden sein.

Sicherlich sollte der werte Leser nun nicht annehmen, ihm würden all diese bösen Dinge in solch gehäufte Form ebenfalls zustoßen (die Hersteller haben hoffentlich gelernt). Aber, da man auch aus Fehlern anderer lernen kann (nur dies meistens nicht einsehen will), vermag John Bears Buch den Frust bei der Computerei erheblich zu reduzieren. Vor allem durch einen wohlbedachten Kauf.

CPU

zahl vergrößern. Im zweiten Teil des Buches behandelt der Autor die 59 Befehle der 68000 CPU. Jede Anweisung wird gründlich erläutert, wobei alle möglichen Adressierungsarten aufgeführt werden. Obwohl jedem Kommando im Schnitt zwei Seiten gewidmet sind, fehlt leider immer ein praktisches Beispiel. Glücklicherweise findet man in diesem Werk genug Programme, die wohl diesem Problem abhelfen. Im letzten Teil stellt Ulrich Schollwöck noch ein Fließkommapaket vor, von dem selbst Profis noch einiges lernen können. Tabellen über die Laufzeiten der einzelnen Memmonics befinden sich im Anhang, sind aber allerdings etwas klein geraten und daher un-

**Das
COMPUTER
TEAM
ABO
für nur
50,— für
12 Ausgaben**

DIE 68000 DIMENSION

Coprozessormöglichkeiten

Bisher haben wir uns nur mit den hardwaremäßigen Eigenschaften diverser Mikroprozessortypen beschäftigt. Dabei setzten wir immer voraus, daß diese ohne jegliche Unterstützung Befehle abarbeiten. In der 16 Bit-Klasse eröffnet sich aber erstmals die Möglichkeit, den Hauptprozessor, bisher als CPU oder Mikroprozessor bezeichnet, mit einem Coprozessor kooperieren zu lassen. Das bedeutet, beide Prozessoren teilen sich die Arbeit so auf, daß jeder seine Stärken ausspielen kann und somit wahre Geschwindigkeitsrekorde aufgestellt werden können. Bei diesem Coprozessor handelt es sich, genauer gesagt, um einen Arithmetikchip.

Doch was bedeutet das? Dazu muß ich weiter ausholen und die Problematik bei der Verarbeitung von Gleitkomma-Zahlen schildern. Wenn man mit Integerzahlen rechnet, das sind Zahlen ohne Komma, kann ein Mikroprozessor jeder Klasse diese mit relativ wenigen Befehlen verarbeiten. Dadurch erreicht man hohe Verarbeitungsgeschwindigkeiten. Das erkennt man an den Integercompilern, die gegenüber ihren Floating-Point (Komma) - Kollegen wahre Geschwindigkeitsrekorde aufstellen. Unbequem wird es für die Prozessoren, wenn sie kommabehaftete Größen (Floating-Point) verarbeiten sollen. Betrachten wir uns einmal dazu eine normalerweise einfache Addition. Ein Summand (mindestens zwei existieren) besteht bei Gleitkommaverarbeitung aus zwei Größen, der Mantisse und dem Exponenten. Die Mantisse nimmt folgende Werte an: $0.1 \leq \text{Mant.} < 1$.

Der Exponent gibt dagegen die Position des Kommas an. Will man nun zwei kommabehaftete Zahlen addieren, funktioniert es nur mit Exponenten gleicher Größe. Also werden die Mantissen solange um-

geformt, bis die Exponenten übereinstimmen. Erst dann kann die eigentliche Rechenoperation ausgeführt werden. Es ist doch anhand des Beispiels leicht einzusehen, daß hierzu eine Menge Zeit benötigt wird, was die Rechengeschwindigkeit stark reduziert. Um diesen Nachteilen zu entgehen, wurden Arithmetikprozessoren entwickelt, die die verschiedensten mathematischen Operationen mit Gleitkomma in kürzester Zeit behandeln. Die Tabelle Seite 13 gibt hierzu ein paar Beispiele, wobei als gemeinsame Taktfrequenz der Fairness halber 8 MHz gewählt wurde.

Obwohl der 8087 unfairerweise eine größere Mantisse bearbeiten mußte, ist dieser Arithmetikprozessor in dem Vergleichstest trotzdem noch wesentlich schneller als seine Konkurrenten. Ein solches Rechengenie wird immer zusammen mit einem normalen Prozessor angewendet, die sich, abhängig von den Befehlen, die Arbeit aufteilen. Interessant ist die Art und Weise der Arbeitsteilung. Erhält der Hauptprozessor eine Arithmetikanweisung, versteht er zwar ihren Binärcode, weigert sich aber, die Daten

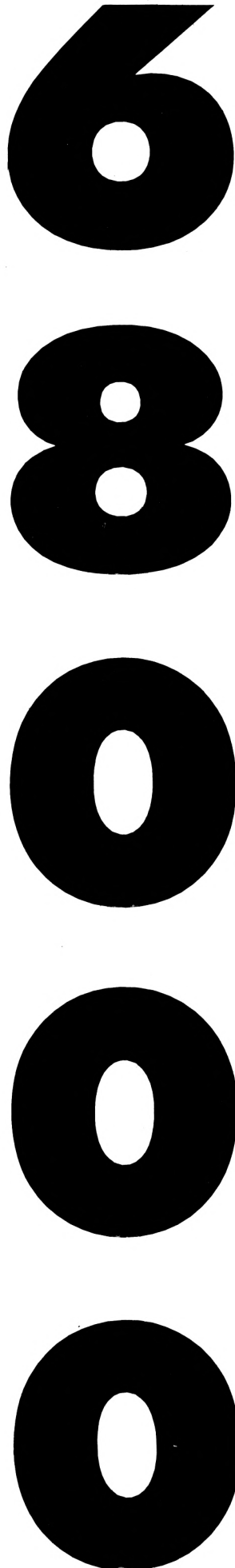
zu verarbeiten. Statt dessen legt er seinen Datenbus in den hochohmigen Zustand (Tri State = dritter Zustand) und gibt somit dem Arithmetikchip den Bus frei. Anschließend erhält er noch ein Signal, so daß das Rechengenie hiermit aktiviert wird und seine Arbeit aufnimmt. Wurden alle Berechnungen erledigt, so legt der Coprozessor seinen Datenbus in den Tri-State-Modus, übermittelt dem Hauptprozessor via Steuerleitung einen binären Wert (High oder Low, je nach Typ), der daraufhin seine Aktivitäten wieder aufnimmt. Beide erreichen durch diese Kooperation sagenhafte Rechengeschwindigkeiten, wie das nachfolgende Beispiel zeigt:

Ein Arithmetikprozessor arbeitet mit sehr breiten Registern, die beim 8087 achtzig Bit-Zahlen aufnehmen können! Ein Nachteil solcher Coprozessoren sei jedoch nicht verschwiegen. Zahlt man für einen 16 Bit-Prozessor so um die hundert DM, kommt man bei den Mathematikgenies in der Regel gerade mit 500-600 DM aus. Der in unseren Beispielen aufgeführte 8087-Arithmetikprozessor ist im

übrigen für eine Zusammenarbeit mit dem Intel 8086 und 8088 ausgelegt. Selbstverständlich läßt sich auch der Motorola 68000 mit einem solchen Rechengenie ausstatten, der nicht 8087 heißt, sondern Motorola 68881, aber gegenwärtig noch nicht lieferbar ist. Er kann für die gesamte 68000-Familie verwendet werden, also auch für den demnächst erscheinenden 32 Bit-Prozessor 68020. Dieser frühestens ab Winter 1985 erhältliche Arithmetikprozessor 68881 stellt wirklich alles bisher Dagewesene in den Schatten. So besitzt er neben den vier Grundrechenarten: Wurzel, arcos, arcsin, arctan, arctanh, cos, cosh, exp(x), log, ln, log mit festlegbarer Basis, sin sin und cos gleichzeitig berechnen, sinh, tan, tanh, Potenzen mit der Basis 10 und 2. Andere Arithmetikchips kennen die meisten der aufgeführten Funktionen nicht, so daß diese nicht mit der üblichen hohen Geschwindigkeit behandelt werden können. Hier nun einige Berechnungszeiten, die einmalig sind: siehe Tab. 2. Dieser Arithmetikprozessor besitzt acht Register mit einer Datenbreite von achtzig Bit. Er kann mit einer maximalen Taktfrequenz von sage und schreibe 16,67 MHz betrieben werden.

16,67 MHz

Da Haupt- und Coprozessor ein asynchrones System bilden dürfen, kann der 68881 mit knapp 17 MHz laufen, während der 68000 sich mit niedrigeren Taktfrequenzen abgibt. Nur so entsteht ein Maximum an Rechengeschwindigkeit und Unabhängigkeit. An dieser Stelle wird sich mancher Leser sagen, gut, dann statte ich einen Mikroprozessor der 8 Bit-Klasse mit einer Arithmetikeinheit aus, so daß durch die daraus resultierende höhere Rechengeschwindigkeit ein 16 Bit-Prozessor weitgehend überflüssig wird. Das ist leider nicht richtig. Der Haupt- und Coprozessor müssen sich nämlich den Datenbus teilen. Da nun der Datentransfer auf dem Datenbus eines 8 Bit-Prozessors sehr langsam abläuft, verursacht durch die vorhandenen acht Datenleitungen und der recht kleinen Taktfrequenz, wird dieser Bus des Hauptprozessors fast die ganze Zeit blockiert und läßt den Arithmetikprozessor kaum zum Zuge kommen. 16 Bit-



CPUs mit ihren meist sechzehn Datenleitungen holen sich immer gleich zwei Speicherzelleninhalte auf einmal. Und bei einer hohen Taktfrequenz von etwa acht bis zwölf Megahertz geschieht dies auch noch sehr geschwind. Kein Wunder also, daß die 16 Bit-Mikroprozessoren den Datenbus sehr schnell wieder freigeben können. Hinzu kommen Macrobefehle, die den Hauptprozessor lange beschäftigen und daher quasi nicht den Datenbus in diesem Zeitraum benötigen. Außerdem unterstützen bei einigen 16 Bit-Prozessoren bestimmte Befehle die Arbeitsteilung, so zum Beispiel der Maximum-Modus beim 8086 und 8088.

Der Motorola 68000 intern

An dieser Stelle soll bevorzugt nur der 68000 betrachtet werden, da die 8 Bit-Klasse nichts Gleichwertiges anbietet und die anderen 16 Bit-Typen nur vereinzelt mithalten können. Die Motorola CPU besitzt auffallend viele Interruptvektoren (192), die eine Unterbrechung mit definiertem Sprung hervorrufen, sobald irreguläre Bedingungen, wie zum Beispiel eine Division durch Null, unzulässige Befehlscodes oder Überschreitung vorgegebener Wertegrenzen in Datenregistern auftreten. Solche Unterbrechungen werden auch als Traps bezeichnet. Bei der 8 Bit-Klasse können solche Interrupts nur durch aufwendige Software erreicht werden, die aber den Prozessor sehr verlangsamen. Erstmals tritt auch eine zweite arithmetisch-logische Einheit für die Adressierung auf. Eine ALU führt bekanntlich alle Rechenoperationen aus. Bei den früheren Mikroprozessoren mußten Daten und Adressen „Schlange stehen“, um verarbeitet zu werden. Dieser Zustand ist aber dem Motorola 68000 völlig unbekannt, da die Adressen eine eigene Rechenzentrale besitzen. Dies führt selbstverständlich zu einer erhöhten Arbeitsgeschwindigkeit. Eine solche sinnvolle zweite Arithmetikeinheit kennt der neue Intel 80286, auch unter der Bezeichnung iAPX 286 bekannt. Alle anderen Prozessoren haben dagegen nur eine ALU, was sie entsprechend auch verlangsamt. Kannten die 8 Bit-

Mikroprozessoren nur einen Stapel (= reservierte Speicherplätze für besondere Daten) und nur einen Stapelzeiger, der auf eine Adresse des Stapels zeigt, liegen bei Motorola dagegen völlig andere Strukturen vor. Ein lifo-orientierter Stapel (was zuletzt an Daten in den Stapel gelegt wurde, kommt auch bei Aufruf wieder zuerst heraus), und ein fifo-orientierter Stapel (= Gegenteil von lifo), auch Queue genannt, erlauben in Verbindung mit den beiden Stapelzeigern ein sehr elegantes Programmieren, was kürzere Programme erlaubt und somit kleinere Laufzeiten.

Im Gegensatz zur 8 Bit-Generation sind Befehle (Link und Unlink) beim 68000 vorhanden, die jedem Compiler erlauben, schnelleren Maschinencode zu produzieren. Das bedeutet, wer mit höheren Programmiersprachen wie Pascal, Fortran oder Forth arbeitet, also sogenannten Compilersprachen (Compiler=Name des Übersetzers), kann Dank der Unterstützung einiger Maschinenbefehle mit einer hohen Verarbeitungsgeschwindigkeit rechnen.

Kosten

Leider kann ein solches Maximum an Technik nicht gerade preiswert erworben werden. Kostet ein 8 Bit-Prozessor zwischen 10 und 15 DM, muß man für den 68000 einen Hunderter hinblättern. Auch das im Winter erscheinende Rechenwunder namens 68881 wird schätzungsweise mindestens 600 DM kosten, was für einen hochwertigen Arithmetikprozessor ein völlig normaler Preis ist (der 8087 ist nicht unter 600 DM erhältlich).

Softwarekompatibilität

Da der Motorola 68000 von seinen Hard- und Softwareeigenschaften her eine völlige Neuentwicklung darstellt, kann die bisherige Software für andere Mikroprozessoren leider nicht benutzt werden. Es sprechen aber alle Anzeichen dafür, daß der 68000 sich in der 16 Bit-Klasse durchsetzen wird. So existieren schon Betriebssysteme wie Unix oder CP/M/68K, die den gleichen Effekt haben werden wie CP/

	68000	Z 80	8087
Mantissen	32 Bit	32 Bit	52 Bit
Addition	0.06 ms	0.4 ms	0.009 ms
Subtraktion	0.06 ms	0.4 ms	0.011 ms
Multiplikation	0.06 ms	1.3 ms	0.017 ms
Division	0.11 ms	2.2 ms	0.024 ms
Quadratwurzel	0.65 ms	2.0 ms	0.023 ms
Sinus	1.40 ms	19 ms	-----
Exponentialfunktion	1.30 ms	14.5 ms	0.063 ms

Tabelle 1

Aufgabe (64 Bit Zahlen)	Ausführungszeit in µs
Addition und Subtraktion	4.6
Multiplikation	5.8
Division	7.7
sin	24.9
cos	24.9
sin und cos gleichzeitig	27.4
tan	29.1
arcsin	35.7
arccos	37.7
arctan	25.5

Tabelle 2

M beim Z 80, nämlich ein großes Softwareangebot. Mittlerweile erscheinen aber von bekannten Firmen wie Apple, Commodore und Atari Rechner mit dem Motorola-Prozessor, was in Kürze auch ein größeres Softwareangebot hervorrufen wird. So hat Atari laut Firmenaussage allein in den Vereinigten Staaten etwa 120 Firmen beauftragt, für den 520 ST (hat den 68000) Programme zu entwickeln. Existiert erst einmal ein solches für einen bestimmten Prozessor, so ist es nicht schwer, dieses für einen anderen Rechner mit der gleichen Zentraleinheit umzuschreiben. Das erlebt man doch fast jede Woche. Erscheint ein Programm für den Sinclair Spectrum, so ist es wenige Wochen später auch schon für den Amstrad (Schneider) erhältlich. Weitere zahlreiche Beispiele ließen sich noch anführen.

Aussichten

Der Motorola 68000, der inzwischen auch in Lizens von Valvo gebaut wird, stellt nicht den neuesten Stand der Technik dar. So wird Motorola frühestens im Winter 85 einen echten 32 Bit-Prozessor namens 68020 auf den Markt bringen. Er besitzt schon 64 Bit-Register und

einen Adreßbus, der Speicher in der Gigabytegröße adressieren kann und einen 32 Bit-Datenbus besitzt. Der Befehlssatz wurde nochmals wesentlich erweitert und schließt erstmalig Gleitkommabefehle ein. Daß die Geschwindigkeit gegenüber dem an sich schon sehr schnellen 68000 nochmals gesteigert wurde, liegt auf der Hand. Doch bis dieser Prozessor auf dem Markt eine große Verbreitung findet, werden noch Jahre vergehen. Insbesondere der Preis von mindestens 800 DM pro Stück wird die Verbreitung dieses Mikroprozessors nicht gerade fördern. Interessant ist, daß alle Programme für den 68000 auch auf dem 32 Bit-Giganten laufen werden. Also ein Grund mehr für die Softwarefirmen, massenweise 68000 Software zu entwickeln, um bei Markteinführung auch schon etwas für den 68020 zu haben! Außerdem stellt sich die Frage, ob ein 16 Bit-Rechner für die meisten Zwecke nicht schon völlig ausreicht. 16 Mbyte-Speicher in Verbindung mit hoher Verarbeitungsgeschwindigkeit dürfte selbst den Ansprüchen großer Unternehmen völlig genügen. Wo zum Beispiel durch Multitasking der Datendurchfluß zu klein wird, kann der Arithmetikprozessor 68881 den Kauf eines extrem teureren 32 Bit-Rechners ersparen.

KUH

COMPILERTECHNIK

3. Teil

von Frank Thielen

Wir kommen nun zum dynamischen Test der Sprünge, also der GOTOs (unbedingte) und der IF-THENs (bedingte Sprünge). Die syntaktische Richtigkeit der Sprachelemente haben wir ja getestet; falls etwas nicht in Ordnung war, beendete das Programm seine Arbeit in der Zeile 510. Es geht nun vielmehr darum festzustellen, ob die angesprungenen Zeilen auch existieren. Für jede Zeilennummer hinter einem GOTO und einem THEN muß also sichergestellt sein, daß sie auch als Zeilennummer vor einer Anweisung vorkommt. Die einfachste (und auch speicherplatzsparendste) Methode ist, die Zeilen zu suchen, die mit einem GOTO oder einem IF beginnen. Dann bestimmt man die Zeilennummer am Ende dieser Zeile und geht das gesamte Programm durch, um eben diese Zeilennummer zu finden. Hat man Erfolg, kann das nächste Sprungziel gesucht werden, sonst liegt ein Fehler vor (etwa: Sprungziel Zeile Nr. ... fehlt).

Im (Teile-) Listing 1 sehen Sie die Implementation des Sprungtestes. Zuerst werden alle Zeilen gesucht, die mit GOTO oder IF beginnen; ist dies nicht der Fall, wird nach 700 zur weiteren Überprüfung gesprungen. Dann werden von hinten alle Ziffern vom Text der Programmzeile entfernt, bis irgendwann ein Zeichen auftaucht, das keine Ziffer darstellt. Damit ist die Zielzeilennummer ermittelt, vor der ja das GOTO oder das THEN steht. Jetzt wird das gesamte Programm, beginnend von vorne, untersucht, ob eine Zeile den gleichen Wert (VAL) wie die Zielzeile hat. Da ja nach bestandener Syntaxprüfung im ersten Durchgang alle Zeilen mit einer Zeilennummer beginnen, ist diese mit der VAL-Funktion sehr leicht zu bestimmen. Wenn also der Wert einer Zeilennummer mit der Zielzeile übereinstimmt, ist die GOTO- bzw. IF-Zeile korrekt. Wenn man die Zeilennummer nicht mit der VAL-Funktion vergleicht, sondern direkt als String, so würde die Nummer 0010 nicht als Zeile 10 erkannt werden!

sich nur vor, das Programm würde in der ersten Zeile ein WEND und in der zweiten Zeile das WHILE enthalten; die Anzahl wäre zwar in Ordnung, aber die falsche Reihenfolge würde eine Compilation verhindern. Es muß also zu jedem Schleifenbegrenzer das dazugehörige andere Element existieren. Lassen Sie mich nun an einem Beispiel einer WHILE-WEND-Schachtelung die Methode demonstrieren, wie man recht einfach den jeweils anderen Schleifenbegrenzer findet:

```

WHILE      a  1
  WHILE    b  2
  ...
  WEND     b  1
  WHILE    c  2
    WHILE  d  3
    ...
    WEND   d  2
  WEND     c  1
WEND       a  0
    
```

Die Struktur ist korrekt, alle zusammengehörenden Elemente sind mit dem gleichen Buchstaben gekennzeichnet. Nehmen wir nun einmal an, wir wollten zu dem WHILE in der ersten Zeile (mit Buchstabe a) das Partnerelement finden. Dazu setzen wir einfach einen Zähler auf 1 (Spalte hinter dem Buchstaben). Nun suchen wir das Programm weiter nach WHILEs und WENDs ab, und jedesmal, wenn wir ein WHILE finden (das ja eine neue Schleife öffnet), setzen wir den Zähler höher. Umgekehrt

erniedrigen wir den Zähler, wenn wir ein Schleifenende (WEND) finden. So hat der Zähler immer den aktuellen Stand der Verschachtelungstiefe der Schleifen, und wenn er den Wert Null erreicht hat (letzte Zeile), dann ist auch die äußerste Schleife beendet und das passende WEND gefunden. Findet man das Programmende, bevor der Zähler Null annimmt, so fehlt das WEND, und das Programm ist fehlerhaft. Entsprechend umgekehrt läuft die Suche nach einem WHILE zu einem WEND; das Suchen läuft nun rückwärts (also von hohen zu niedrigen Zeilennummern), beim Auftreten eines WEND wird der Zähler erhöht und bei einem WHILE erniedrigt. Genauso funktioniert dieser Algorithmus auch bei FOR-NEXT- oder REPEAT-UNTIL-Schleifen. Sie sollten an obigem Beispiel einmal andere Fälle ausprobieren; Sie sehen dann leicht, wie es funktioniert.

Wir wollen nun ein Modul schreiben, das beide Suchrichtungen durchführt und bei Erfolg die Zeilennummer mit dem Partnerelement als „zn“ liefert. Wird das Element nicht gefunden, soll „zn“ den Wert Null annehmen, eine Zeilennummer, die nicht vorkommt. Wir müssen der Routine noch die Namen der Elemente liefern, die die Schleife begrenzen: „anfang\$“ und „ende\$“. Außerdem übergeben wir die Suchrichtung mit der Variable „vorw“, die -1 bei Suche vorwärts (d.h. von oben nach unten) und 0 bei Suche rückwärts (also nach dem Schleifenanfang) annehmen soll. Dieses Modul sehen Sie

WHILE und WEND

WHILE- und WEND- Schlüsselworte müssen im Programm immer paarweise vorkommen, damit es syntaktisch korrekt und damit übersetzbar ist. Es reicht aber nicht, einfach alle WHILEs und WENDs zu zählen und bei gleicher Anzahl die Korrektheit des Programms anzunehmen. Stellen Sie

Listing 1

```

600 FOR i=1 to lquell
610 a$=quell$(i)
620 GOSUB 1000
625 zeile$=a1$
627 ziel$=""
630 b$=RIGHT$(a$,1):IF b$>"0" AND b$<="9" THEN ziel$=b$+ziel$:a$=LEFT$(a$,LEN
(a$)-1):GOTO 630
640 j=1
650 IF VAL(quell$(j))=VAL(ziel$) THEN 900
660 j=j+1
670 IF j>lquell THEN PRINT "Nichtexistentes Sprungziel";:GOTO 880
680 GOTO 650
880 PRINT " in Zeile ";zeile$
890 ok=0
900 NEXT i
910 IF NOT ok THEN PRINT "Quellprogramm enthält Fehler - keine Übersetzung mögl:
ch":END
    
```

in Listing 2. Wichtig ist zuerst einmal, daß wir die Variable a\$ vorher in eine andere „retten“, damit sie nach Beendigung der Routine wieder unverändert zurückgegeben wird. Auch wenn a\$ nach Aufruf des Moduls gar nicht mehr gebraucht wird, sollte man immer auf Nummer Sicher gehen, da bei späteren Veränderungen des Programms die Variable irgendwann doch einmal benutzt werden sollte, und dann treten Fehler auf, die sehr schwer zu finden sind.

„j8“ ist die Indexvariable, die auf die gerade durchsuchte Zeile im Feld quell\$ zeigt. Wenn vorwärts gesucht werden soll, wird sie erhöht und sonst erniedrigt. Danach folgt der Test, ob schon das Ende bzw. der Anfang des Programms

deren Fall wird die Zeile aus dem Feld in a\$ gelesen und die Zeilennummer mit der schon bekannten Routine ab 1000 entfernt. Steht dann das Schlüsselwort für den Schleifenanfang anfang\$ als erstes in der Zeile, wird je nach Wert von „vorw“ der Zähler „i8“ inkrementiert oder dekrementiert. Steht das Schleifenelement in der Zeile, passiert das gleiche, nur umgekehrt. Hat der Zähler den Wert Null dann noch nicht erreicht (ist das Schleifenelement oder der -anfang also noch nicht erreicht), wird weitergesucht. Sonst erhält „zn“ den Wert der Zeilennummer, und der Rücksprung erfolgt.

Es fehlt jetzt noch das Programmsegment, das diese Suchroutine aufruft; Sie finden es im Listing 3.

Listing 2

```

3000 REM Schleifenanfang / -ende suchen
3005 a8$=a$
3010 i8=1
3020 j8=1
3025 zn=0
3030 IF vorw THEN j8=j8+1 ELSE j8=j8-1
3040 IF vorw AND j8>iquell THEN a$=a8$:RETURN
3050 IF not vorw AND j8<1 THEN a$=a8$:RETURN
3060 a$=quell$(j8)
3070 GOSUB 1000
3080 IF LEFT$(a$,LEN(anfang$))<>anfang$ THEN 8100
3090 IF vorw THEN i8=i8+1 ELSE i8=i8-1
8100 IF LEFT$(a$,LEN(ende$))<>ende$ THEN 8120
8110 IF vorw THEN i8=i8-1 ELSE i8=i8+1
8120 IF i8<>0 THEN 8030
8130 a$=a8$
8140 zn=VAL(a1$)
8150 RETURN
    
```

erreicht wurde (schließlich kann das Programm aus nur einer Zeile bestehen!). Wenn die Programmgrenzen überschritten sind, wird a\$ wiederhergestellt und der Rücksprung erfolgt. Die Variable „zn“ hat dann den Wert Null, der ihr vorher zugewiesen wurde. Im an-

Damit ist der Parser fertig; wenn wir keine Fehler gemacht und nichts vergessen haben, ist jedes Programm, das diese „Härtetests“ heil übersteht, fehlerfrei.

Wir wollen uns jetzt an die eigentliche Übersetzung machen. Wie zu Beginn angesprochen wurde, soll

der Compiler keinen direkten Maschinencode erzeugen, sondern Quellcode für einen Assembler. Da man aber normalerweise den Übersetzer und den Assembler in ein Programm packt und diese nicht getrennt aufruft, sieht es nach außen so aus, als würde der Compiler direkt und ohne Umwege den Objektcode generieren. Es gibt

Der Übersetzer

übrigens auch Compiler, die nur in einem File den Assembler-Quellcode erzeugen und dann die Arbeit einem Assembler überlassen, der mit dem Compiler gar nichts zu tun hat. Dies hat den Vorteil, daß man den Quellcode vor der Assemblierung noch optimieren, das heißt in Bezug auf Geschwindigkeit oder Speicherverbrauch verbessern kann. Es ist natürlich klar, daß kein Compiler der Welt so kurze bzw. schnelle Lösungen eines Problems bieten kann wie ein Mensch, der direkt in Assembler programmiert. Der Mensch wird hier durch gewisse Tricks hier und da Bytes einsparen können. Diese Tatsache wird durch den „ST-BASIC“-Interpreter für den Atari 520 ST belegt, der in der Programmiersprache BCPL geschrieben ist. Er ist lange nicht so schnell wie ein auf dem gleichen Prozessor laufender, in Maschinensprache geschriebener Interpreter; auch sein Speicherbedarf ist wesentlich höher.

Da wir ja nun Assembler-Quellcode erzeugen wollen, können wir symbolische Adressen benutzen. Dies soll gleich für die Variablen

Assembler-Code

getan werden. Jeder Variablen, die in unserer Sprache je aus einem Buchstaben (A..Z) bestehen, wird ein Label (eine symbolische Adresse) mit gleichem Namen und vorangestelltem „V“ zugeordnet. Der Variable „A“ entspricht also das Label „VA“ usw.. Da wir uns glücklicherweise auf Integervariablen mit je zwei Bytes beschränkt haben, ordnen wir jeder Variablen ein Doppelbyte im Speicher zu, genau wie es ein Interpreter mit Integervariablen auch tut. Wir müssen also jetzt für alle (vorkommenden!) Variablen ein Label deklarieren und zwei Bytes im Speicher reser-

vieren. Für diese Reservierung soll das Assemblerkürzel „DW“ (define word) benutzt werden, so daß ein paar Variablen im erzeugten File also so aussehen:

```
VA DB 0
VC DB 0
VV DB 0
```

Die Null hinter dem DB sagt dem Assembler, daß er das Doppelbyte (das Wort) mit Null füllen soll; dadurch sind dann automatisch alle Variablen mit Null vorbelegt. Manche Assembler kennen jedoch ein Kommando, das nur eine gewisse Anzahl von Bytes reserviert, ohne sie mit einem Wert zu belegen. Dann sind die Variablen zu Beginn des Programms undefiniert, sie haben den Inhalt, der in den Speicherzellen vor Erzeugen des Maschinencodes zufällig stand.

Da wir ja nur 26 Variablen zur Verfügung haben, könnten wir nun für sie alle je zwei Bytes definieren,

26 Variablen

egal ob sie nun irgendwo im Quellprogramm vorkommen oder nicht. Dies ist die einfachste Methode, die aber nur dann machbar ist, wenn nur wenige Variablen vorkommen können. Bedenken Sie dazu, daß auch einige BASIC-Taschencomputer diese Methode anwenden, die Standardvariablen von A bis Z sind fest vordefiniert, und ihr Speicherplatz kann anderweitig nicht genutzt werden.

Wir wollen hier aber eine andere Methode benutzen: Eine Tabelle, die alle verwendeten Variablen

Listing 3

```
700 anfang$="WHILE"
704 ende$="WEND"
706 vorw=-1
700 IF LEFT$(a$,5)<>"WHILE" THEN 800
710 GOSUB 8080
720 IF zn=0 THEN PRINT "Fehlendes WEND zum WHILE";:GOTO 880
730 GOTO 900
800 vorw=0
810 IF LEFT$(a$,4)<>"WEND" THEN 900
820 GOSUB 8000
830 IF zn=0 THEN PRINT "Fehlendes WHILE zum WEND in Zeile";:GOTO 880
840 GOTO 900
```

Listing 4

```
5 DIM tabelle(26)
2010 IF z$>="A" AND z$<="Z" THEN tabelle(ASC(z$)-64)=-1:GOSUB 10000:
fehler=0:RETURN
```

aufnimmt und anhand deren dann die Variablen reserviert werden. Normalerweise geschieht dies so, daß beim ersten Auftreten einer Variablen diese (d.h. ihr Name und evtl. ihr Typ) in die Tabelle eingetragen wird. Vorher muß aber nachgesehen werden, ob die Variable nicht schon eingetragen ist, denn sonst würde die Tabelle viel zu schnell voll werden, und wir würden womöglich Label für Variablen mehrmals definieren. Der Einfachheit halber soll es hier so geschehen, daß die Tabelle 26 Elemente haben soll. Es wird nun nicht die erste auftretende Variable in das erste Tabellenelement, die zweite (anderer Name vorausgesetzt) in das zweite usw. gesetzt. Vielmehr hat jeder Variablenname seinen Tabellenplatz, und hier wird eingetragen, ob diese Variable im Programm vorkommt. Vom BASIC her ist ein (numerisches) Feld ja bei der Dimensionierung mit Nullen vorbelegt, so daß die

Null das Zeichen für „noch nicht vorgekommen“ ist. Nun wird das gesamte Programm nach Variablen durchsucht, und bei einem Auftreten wird das der Variable zugeordnete Feldelement einfach auf -1 (logisch wahr) gesetzt: Bei „A“ das erste Element, bei „B“ das zweite usw.. Dann werden schließlich alle Labels, deren zugehöriges Tabellenelement gesetzt ist, deklariert.

Wir brauchen aber nicht das Programm schon wieder zu durchsuchen, schließlich existiert ja ein Modul, das auf Variablen testet und

Variablenerkennung

diese erkennt (ab Zeile 2000). Ist eine Variable erkannt (dies geschieht in 2010), brauchen wir nur den Eintrag in die Tabelle zu machen. Diesen Eintrag nimmt die neue Zeile 2010 aus Listing 4 vor, in dem auch die Tabelle dimensioniert wird.

Da das Zeichen „A“ den ASCII-Wert 65 hat, wird bei einem „A“ das erste Element der Tabelle auf „wahr“ gesetzt usw.. Jetzt können wir schon die Labels deklarieren und damit die Erzeugung des Quellfiles für den Assembler beginnen. Das File soll (sehr sinnig) in das Feld „files(.)“ geschrieben werden, das je nach Größe des Quellprogramms entsprechend groß definiert werden muß (hier sind 100 Elemente dimensioniert, evtl. kann dies zu wenig sein). Ein Assemblerbefehl besteht zu meist aus einem Label (einer Marke, ähnlich einer Zeilennummer), dem Opcode (dem eigentlichen Befehl) und einem Operanden (Werte, Labels oder andere Informationen, die den Opcode spezifizieren). Platz für Kommentare brau-

Listing 5

```
2 DIM file$(100)
11000 REM überträgt label$, opcode$ und operand$ in das Assembler-Quellfile
11010 file$(sz)=LEFT$(label$+" ",6)+" "+LEFT$(opcode$+" ",2)+" "+operan
d$
11015 label$=""
11020 sz=sz+1
11030 IF sz>100 THEN PRINT "Zwischencode-File läuft über, Quellprogramm zu lang.
":END
11040 RETURN
```

Listing 6

```
920 GOTO 20000:REM Platz vor 1000 für Übersetzer reicht nicht!
20000 sz=1
20010 label$="":opcode$="ORG":operands$="0":GOSUB 11000
20090 opcode$="DB":operands$="0"
20100 FOR i=1 TO 26
20110 IF tabelle(i) THEN label$="V"+CHR$(64+i):GOSUB 11000
20120 NEXT i
```

Listing 7

```
21000 FOR i=1 TO lowell
21010 a$=quell$(i)
21020 GOSUB 1000
21030 zeile=VAL(a1$)
21100 label$="Z"+MID$(STR$(a1$),2)
21110 IF a$="END"THEN opcode$="RET":operands$="":GOSUB 11000:GOTO 25000
25000 NEXT i
```

chen wir natürlich nicht, schließlich kann der Compiler aus einem Programm ohne Kommentare (unsere Sprache hat kein REM!) keine erzeugen. Deshalb implementieren wir eine kleine Routine, die jeweils ein „label\$“, ein „opcode\$“ und einen „operand\$“ in das „file\$“ einträgt und „sz“, den Schreibzeiger, um 1 hochsetzt (siehe dazu Listing 5). Normalerweise muß man dem Assembler sagen, wohin im Speicher er das zu erzeugen

Anfangsadresse

gende Maschinenprogramm setzen soll. Dies geschieht mit einer Anweisung wie „ORG“ (oder „PUT“), gefolgt von einem Operanden, der die gewünschte Anfangsadresse des Maschinenprogramms bezeichnet. Manche Assembler nehmen eine feste Anfangsadresse an, wenn kein ORG gegeben ist. Da die Anfangsadresse von der tatsächlichen Speicherorganisation Ihres Rechners abhängt, schreibe ich einfach Null als Anfangsadresse (obwohl ja bei einem Call 0 bei Z80- oder 8080/8085-Systemen ein Reset durchgeführt wird). Die günstige Lage des späteren Maschinenprogramms müssen Sie einfach selbst bestimmen.

Nun zur Erzeugung der Speicherreservierung

Speicherreservierung

cherreservierung für die Variablen: Listing 6. Damit sind die Variablen schon im Speicher reserviert, es wird einfach für jeden belegten Tabellenplatz ein „V“ mit dem Namen der Variablen als Label in das Assemblerfile geschrieben.

In Listing 7 finden Sie schon den Beginn des eigentlichen Übersetzers, genaueres dazu in der nächsten Ausgabe. Vielleicht überlegen Sie sich einmal, wieso der BASIC-Befehl „END“ in Assembler mit „RET“ (also Return) übersetzt wird.

COMPUTER TEAM
im Abo bequem
per Post
jeden Monat

BÜCHERECKE

Das Turbo Pascal Buch

Autor: Karl-Hermann Rollke
 Format: DIN A 5
 Seitenzahl: 288 mit zahlreichen Abb.
 Preis: 42,- DM

Zielgruppe

Mit diesem Buch erhalten sowohl Programmier-Anfänger in der Sprache Pascal wie auch Umsteiger auf Turbo Pascal alle notwendigen Informationen. Turbo Pascal läuft auf allen CP/M-Rechnern mit Z80-Prozessor, unter MS-DOS, PC-DOS, und ist schnell in der Entwicklungs- und Testphase.

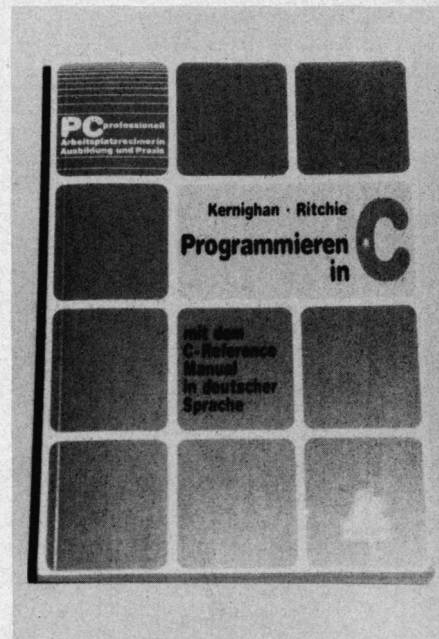
Inhalt

- Allgemeine Einführung in die Arbeitsweise des Turbo-Editors und des Systems.
- Viele Beispiele veranschaulichen die vorgestellten Programmierkonzepte, die leicht verständlich geschrieben sind.
- Die notwendigen Daten- und Kontrollstrukturen werden vom Autor ausführlich erläutert.
- Auch das Arbeiten mit dem Turbo Toolbox-System kommt zur Sprache.
- Übungen zu jedem Kapitel runden diese umfassende Einstiegslektüre ab.

Zusammenfassung

Karl-Hermann Rollke, Autor mehrerer SYBEX-Bücher über Pascal, zeigt Ihnen einen Pascal-Dialekt, der schnell, komfortabel, vielfältig einsetzbar und weitgehend kompatibel zu Standard Pascal und z.B. UCSD-Pascal ist. Dieses leicht verständlich geschriebene und didaktisch gut strukturierte Buch ermöglicht einen systematischen Einstieg in Turbo Pascal.

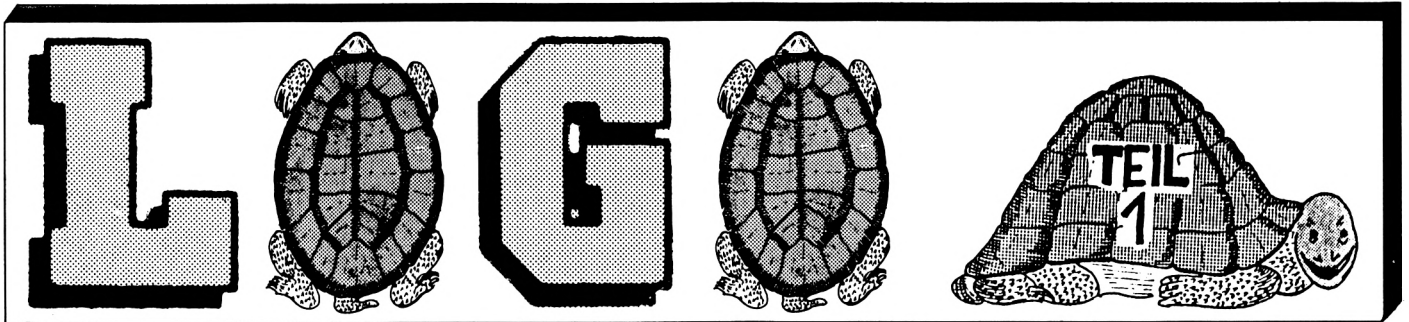
Programmieren in C



Programmieren in C, Autoren: Kernighan und Ritchie, Verlag Hanser, 262 Seiten, 48,- DM.

Das Buch läßt sich in zwei Teile zerlegen, einen 35-seitigen Schnellkurs und ein 227-seitiges Nachschlagewerk. Dieser von den Autoren angekündigte Schnellkurs erweist sich dennoch als sehr ausführlich. Es werden einfache Beispiele aus dem Bereich der Textverarbeitung aufgeführt. Jeder Schritt wird genauestens beschrieben. So lernt man mühelos die Arithmetik, Datentypen als auch Funktionen kennen. Insbesondere Leser mit soliden Pascalkenntnissen werden die Beispiele mühelos verstehen. Die nachfolgenden Kapitel erläutern alles bisher Gelernte noch einmal sehr ausführlich, und man lernt noch weitere Anweisungen und Funktionen kennen, die problemlos in eigenen Programmen integriert werden können. Dennis M. Ritchie, der geistige Vater dieser höheren Programmiersprache, als auch sein Kollege Brian W. Kernighan haben es mit diesem Werk geschafft, C einem breiten Kreis verständlich zu vermitteln. Diese Literatur ist für Anfänger und Experten empfehlenswert, für letztere als Nachschlagewerk.

KUH



Jeder, der sich einen 664 oder eine Floppy zulegt, bekommt die Programmiersprache Dr. Logo praktisch als Zugabe. Aber wenn der tatendurstige Neuling dann die „Einführung in Logo“ in seinem Handbuch aufschlägt, erlebt er eine herbe Enttäuschung, denn gleich der erste Satz lautet: Dieser Abschnitt... sollte jedoch nicht als erschöpfendes Lehrbuch oder Nachschlagewerk betrachtet werden...

Wie wahr. Denn noch nicht einmal die Übersicht der verfügbaren Befehle ist vollständig, von den scheinbar ohnehin unvermeidbaren Fehlern einmal abgesehen.

Und genau hier setzt dieser Kurs ein: Ob Sie nun BASIC- oder Maschinensprache-Profi sind und sich nur „noch eine Sprache einverleiben wollen“, oder ob Sie Anfänger sind: Hier wollen wir das nachholen, was das Handbuch versäumt, indem es so raffiniert auf den Logo-Führer (SW 160) verweist.

Logo ist... eigentlich eine Sprache für Kinder. Sie wurde nämlich unter der Leitung von Seymour Papert bis 1977 am MIT (Massachusetts Institute of Technology) entwickelt, um Kindern den Zugang und das Lernen mit dem Computer zu ermöglichen. Daher kommt auch die SCHILDKRÖTE (von der wir gleich noch mehr hören werden), für die Logo berühmt ist. Diese Schildkröte ist ein kleines Objekt, das man mit speziellen Befehlen auf dem Bildschirm herumdirigieren und zeichnen lassen kann. Bei unserem Logo sieht die Schildkröte allerdings eher aus wie das Raumschiff aus Meteorits. Gerade wegen der Zielgruppe Kinder wurde Logo anfangs oft unterschätzt, aber zum Glück wurde diese Fehleinschätzung in den letzten Jahren revidiert.

Die Schildkröte

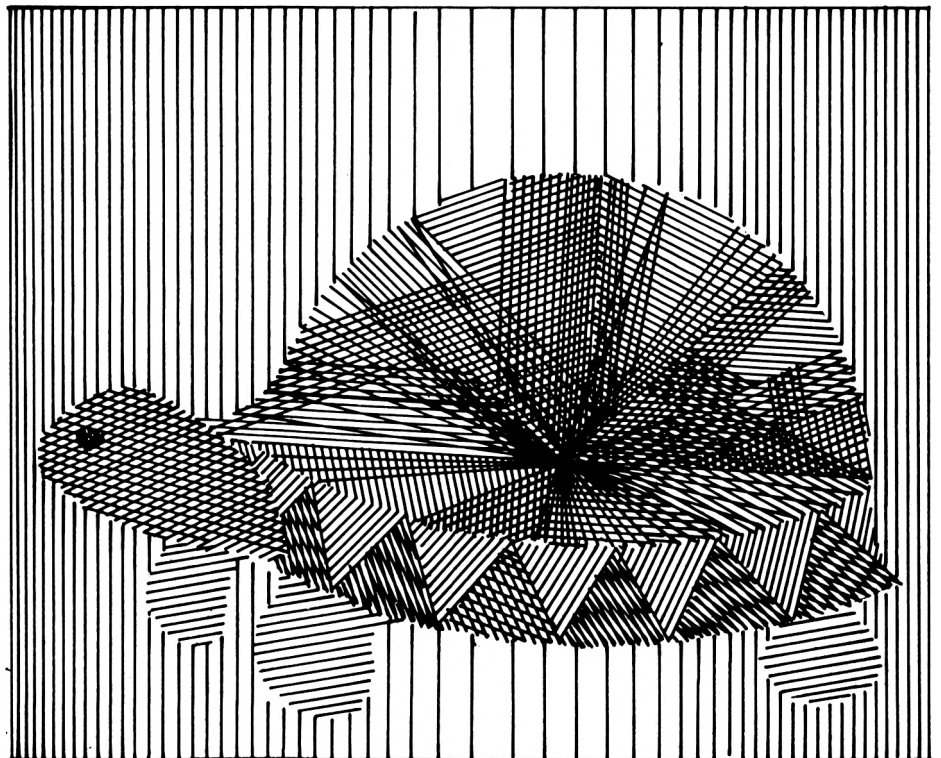
Logo wird gestartet..., indem man die mitgelieferte Diskette ins Laufwerk steckt, und zwar mit der B-Seite nach oben, und CPM ENTER eingibt. Relativ problemlos also. Die Frage, warum man den Logo überhaupt LADEN, also in den Computer holen muß, wird einem Profi sicher nur ein mitleidiges Grinsen ablocken, aber für einen

Anfänger ist sie durchaus berechtigt: Unser Schneider versteht nämlich „von Haus aus“ nur zwei Sprachen, BASIC, in dem man direkt nach dem Einschalten mit ihm „sprechen“ kann, und Maschinensprache, also die Sprache, die der Computer direkt versteht. Das Programm Logo, das wir von der Diskette laden, übersetzt unsere Logo-Befehle in eine dem Computer verständliche Sprache, also in die Maschinensprache. Dh., Logo auf der Diskette ist ein Maschinenspracheprogramm (das der Computer ja versteht), das unsere Eingaben untersucht und wieder in eine dem Computer verständliche Darstellung bringt. Wie es das allerdings macht, ist für diesen Kurs komplett unwesentlich, da wir uns mit Logo

und nicht mit Maschinensprache beschäftigen wollen.

Die ersten Schritte

Nachdem wir also Logo gestartet haben, erscheint eine Copyrightmeldung des Herstellers auf dem Bildschirm und verschwindet so schnell wieder, daß man kaum Zeit hat, sie zu lesen. Anschließend sehen wir einen leeren Bildschirm, leer bis auf ein Fragezeichen und den Cursor in der oberen Ecke. Das Fragezeichen sagt Ihnen, daß der Computer nun auf Ihre Befehle wartet. Probieren wir es doch ein-



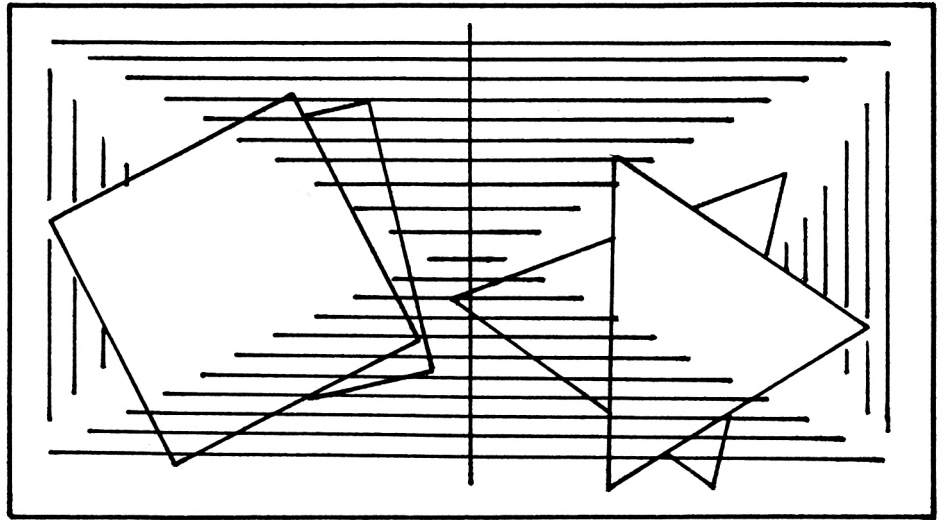
fach mal und schreiben vorwärts (vergessen Sie nicht, danach ENTER zu drücken und in Kleinbuchstaben zu schreiben, da Logo Sie sonst nicht versteht). Der Computer antwortet mit I DON'T KNOW HOW TO VORWAERTS, frei übersetzt also: ich weiß nicht, wie ich „vorwärts“ soll. Klar, wir sollten es vielleicht besser auf Englisch versuchen, also: Forward, aber es kommt wieder das gleiche Resultat. Und hier haben wir gleich eine unangenehme Eigenschaft des Schneider-Logos entdeckt, das will nämlich ABKÜRZUNGEN. Im Gegensatz zu fast allen anderen Logos, bei denen man sowohl die ausgeschriebenen Befehle wie forward als auch die Abkürzungen, hier fd, verwenden kann, müssen wir die Abkürzungen verwenden. Allerdings bringt auch fd ENTER keinen sichtbaren Erfolg, sondern nur die

Abkürzungen

lapidare Meldung, daß wir für fd nicht genug Angaben gemacht haben. Wenn Sie es aber zu guter Letzt mit fd 100 versuchen, ist der Erfolg um so überraschender: Das Fragezeichen samt Cursor wandert ins untere Bildschirm Drittel, und in der Bildschirmmitte erscheint ein gezacktes Dreieck, das blitzschnell nach oben wandert und dabei einen Strich auf seinem Weg zurückläßt. Mit etwas Phantasie haben Sie sicher schon erkannt, daß es sich um unsere berühmte Schildkröte handelt. Was ist jetzt passiert? Wir haben den Befehl fd 100, also vorwärts 100 gegeben, und unsere Schildkröte ist einhundert Schritte nach vorne (nach oben gegangen). Eigentlich ganz logisch. Aber das ist natürlich noch lange nicht alles, was sie kann. Probieren Sie es einmal mit lt 90. Unsere Schildkröte dreht sich um 90 Grad nach links. Und wenn wir jetzt rt 90 eingeben, steht sie wieder so wie vorher, hat sich also um 90 Grad nach rechts gedreht. Mit bk 100 bringen wir sie dann schließlich wieder an den Anfangspunkt der Linie zurück.

Die ersten Befehle

Wir kennen jetzt also schon vier Befehle, mit denen wir die Schildkröte steuern können: fd (von forward = vorwärts), bk (back = zu-



rück), lt (left = links) und rt (right = rechts). Mit diesen vier Befehlen können wir schon eine ganze Menge anfangen, z.B. ein Quadrat zeichnen. Dazu löschen wir erst mal den Bildschirm, um die Linie, das Abfallprodukt vom ersten Versuch „wegzuräumen“: Da „lösche Bildschirm“ in Englisch Clear Screen heißt, versuchen wir es logischerweise mit cs ENTER, und siehe: Es klappt. Der Schirm ist sauber, und die Schildkröte steht wieder in der Mitte. Mit fd 100 ziehen wir jetzt eine gerade Linie nach oben und drehen die Schildkröte mit lt 90 um 90 Grad nach links, um dann wieder mit fd 100 eine Linie zu ziehen, diesmal allerdings nach links, da die Schildkröte ja in diese Richtung zeigt. Jetzt wieder ein lt 90 und so weiter, bis die Schildkröte wieder am Ausgangspunkt angekommen ist. Wenn Sie bisher alles verstanden haben, sollte es Ihnen auch nicht allzu schwer fallen, ein Dreieck zu zeichnen. Ein Tip: Alle drei Winkel sollten 120 Grad betragen. Versuchen Sie es jetzt einmal, und lesen Sie bitter erst weiter, wenn Sie es geschafft haben, oder wenn Sie sicher sind, daß Sie es nicht schaffen.

FD 100

Mit cs fd 100 lt 120 fd 100 lt 120 fd 100 ENTER löschen wir den Bildschirm und zeichnen das Dreieck. Beachten Sie bitte, daß man in Logo, wie wir es gerade getan haben, auch mehrere Befehle in eine Zeile schreiben kann, ohne jedesmal zwischendurch ENTER zu drücken. Man muß allerdings einen Abstand dazwischen lassen, lt120fd100 wäre also unzulässig.

Logo lernt ein neues Wort. Stellen Sie sich vor, wie praktisch es wäre, wenn Sie Logo beibringen könnten, jedesmal ein Dreieck zu zeichnen, wenn Sie einfach nur dreieck eingeben. Nun, das haben sich die Entwickler von Logo wohl auch gedacht, und so ist das auch überhaupt kein Problem. Geben Sie ein: to dreieck ENTER. Das einzige, was sich dann ändert, ist, daß Fragezeichen, mit dem Logo normalerweise auf die Befehle des Benutzers wartet, durch ein > ersetzt wird. Damit sagt uns Logo: „Alles, was du jetzt eingibst, merke ich mir, aber ausführen werde ich es erst, wenn du es mit dreieck aufrufst.“ Wir schreiben also erneut fd 100 lt 120 fd 100 lt 120 df 100 ENTER. Das war schon alles, aber genau das müßten wir Logo auch sagen, und dazu tippen wir end ENTER. Wir bekommen die Meldung „DREIECK DEFINED“. Also hat >Logo das Wort

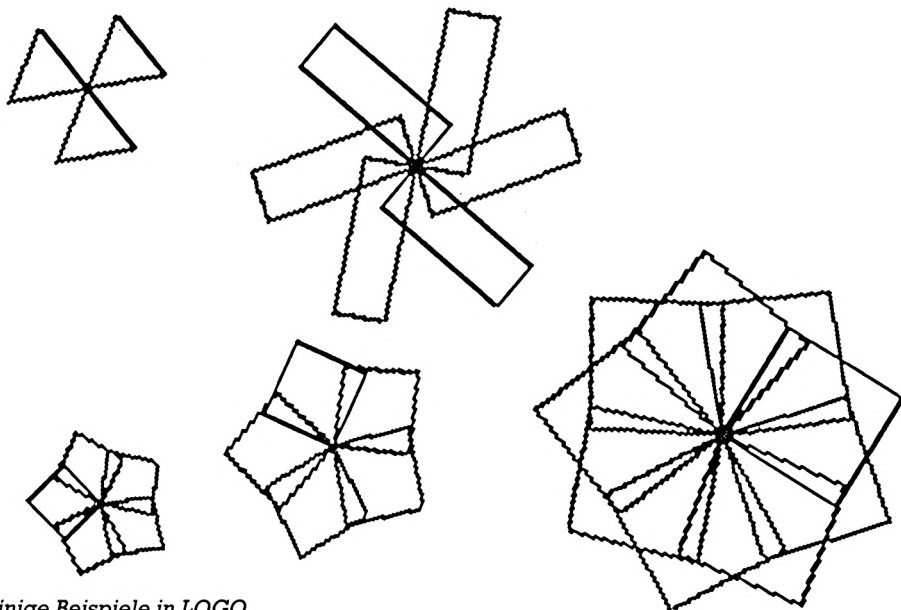
Dreieck

dreieck gelernt. Probieren Sie es aus : dreieck ENTER zeichnet tatsächlich das Dreieck! Das allgemeine Schema, mit dem man Logo Worte beibringt, ist also folgendes : to leitet die Definition eines neuen Wortes ein. Unmittelbar nach to muß der Name des neuen Wortes kommen. Das kann jede beliebige Buchstabenkombination sein, vorausgesetzt, diese Kombination ist nicht schon ein Logo-Wort. Jetzt müssen die Anweisungen kommen, was Logo bei Aufruf dieses Wortes tun soll (das können beliebig viele Zeilen sein), und zuletzt kommt ein end. Das war's schon. Wir hätten das neue Wort auch Quadrat nennen können, es wäre

trotzdem ein Dreieck gezeichnet worden, da sich Logo nicht um den Sinn (oder Unsinn) des Namens kümmert, sondern nur um die Anweisungen, die es dadurch ausführen soll. Ein kleiner Versuch zeigt auch die Richtigkeit dieser Behauptung: to Quadrat

```
dreieck  
end
```

Damit haben wir Logo das neue Wort `quadrat` beigebracht, das nichts anderes tut, als unser eben definiertes Wort `dreieck` auszuführen und somit ein Dreieck zu zeichnen!



Einige Beispiele in LOGO

Variablen. Was jetzt kommt, wird bei dem einen oder anderen unangenehme Erinnerungen an seine Schulzeit wachrufen, bzw. die bedauernswerten Schüler, die gerade damit traktiert werden, sich stöhnend abwenden lassen: Variablen, fast dieselben, die in mathematischen Gleichungen wie $x + 3 = 5$ vorkommen. Die sind nämlich ein wichtiger Bestandteil jeder Computersprache, also auch von Logo. Man kann sich eine Variable bildlich als eine Schublade vorstellen, auf der ein Name steht. Um an den Inhalt der Schublade heranzukommen, muß man diesen Namen angeben. In Logo kann man in diese Schubladen 1) Zahlen 2) Wörter und 3) Befehle hineinlegen. Die ersten beiden Möglichkeiten sind eigentlich nichts Besonderes, das kann man in jeder Programmiersprache, wie z.B. in BASIC. Die dritte Möglichkeit ist schon ungewöhnlicher, und deshalb werden wir uns auch erst in späteren Folgen damit beschäftigen. Aber auch bei den ersten beiden Möglichkeiten gibt

es in Logo noch eine Besonderheit — man kann mehrere Zahlen oder Wörter gleichzeitig in eine einzige Schublade hineinlegen. Doch dazu später mehr. Sehen wir uns erst einmal an, wie wir normale Zahlen oder Wörter in eine solche Schublade hineinlegen: Bezüglich des Namens der Variablen gilt dasselbe wie das für die neu definierten Wörter Gesagte. Wenn wir unsere Variable also `hugo` nennen wollen, so können wir das tun. Mit `make „(mache)“` sagen wir dem Computer, daß das folgende Wort eine Variable werden soll. Außerdem müs-

sen wir ihm auch noch sagen, was er in diese Variablen-Schublade hineinlegen soll. Mit `make „hugo 5“` erschaffen wir also eine Schublade, die den Namen `hugo` hat, und in der die Zahl 5 liegt. Um den Inhalt einer Schublade anzusehen, muß ein Doppelpunkt und unmittelbar dahinter der Name der Variablen angegeben werden, also liefert `:hugo ENTER` das Ergebnis 5. Mit dem, was in dieser Schublade liegt, können wir nun rechnen, wir können es vergleichen oder ersetzen.

HUGO + 2

`:hugo + 2 ENTER` liefert 7, da $5 + 2$ logischerweise 7 ist, auf `:hugo = 5` antwortet Logo mit TRUE (wahr) oder auf `:hugo = 9` mit FALSE (falsch).- Probieren Sie das auch mit größer und kleiner (`<`, `>`), Sie werden sehen, es geht auch. Wir können sogar `:hugo` mit 89.6 (z.B.) multiplizieren und das Ergebnis dann

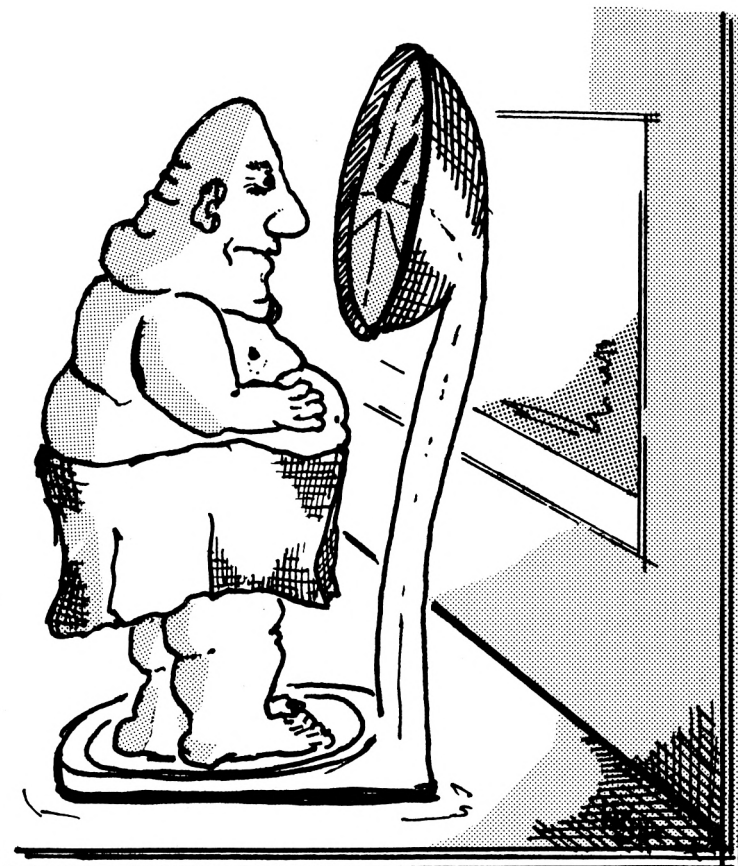
wieder in `:hugo` ablegen: `make „hugo :hugo + 89.6 ENTER`. Wenn Sie jetzt den Inhalt von `:hugo` überprüfen, werden Sie sehen, daß tatsächlich das neue Ergebnis in `:hugo` steht. Achten Sie beim `make`-Befehl darauf, daß vor dem ersten `hugo` ein Anführungszeichen, vor dem zweiten aber der Doppelpunkt steht. Das hat durchaus einen Sinn: Nach einem `make`-Befehl erwartet Logo einen Text, nämlich den Namen der Schublade, die es machen bzw. füllen soll. Ein Text wird mit einem Anführungszeichen (nur einem, nicht zwei wie in BASIC) gekennzeichnet. Danach soll aber eine Zahl kommen, nämlich die, die in die Schublade hineinkommen soll. Da der Doppelpunkt sich immer auf den Inhalt der angegebenen Schublade bezieht, also auf die Zahl 5 in unserem Falle, haben wir ja unsere Zahl. Der Befehl `make „hugo :hugo würde nachsehen, welche Zahl in :hugo steht und diese dann wieder in :hugo hineinschreiben`. Es würde sich überhaupt nichts ändern. Da wir aber noch die in `:hugo` gefundene Zahl multiplizieren, wird nicht die alte Zahl, sondern das Ergebnis der Multiplikation hineingeschrieben. Überlegen Sie einmal, was der Befehl `make „hugo :hugo + 1` wohl anstellen würde... (probieren Sie es aus).

Ein bißchen Grafik

Nach diesem zweifellos komplizierten Abschnitt über die Variablen möchte ich Ihnen zum Abschluß dieses Teils noch ein bißchen Entspannung gönnen. Im folgenden finden Sie ein neues Wort, das Sie abtippen sollten. Achten Sie bitte genau darauf, keine Tippfehler zu machen, da einige neue Wörter und Logo-Eigenschaften verwendet werden, die Sie noch nicht kennen. Aber dieses kleine Beispiel zeigt sehr gut, wie mit wenig Schreibarbeit sehr schöne Grafik in Logo erzeugt werden kann. Und es zeigt Ihnen auch, was Sie in zwei oder drei Folgen selber können werden...

```
to gebilde :g  
  cs  
  repeat :<fd :g lt 89 make „g :g - 1>  
end  
Und starten Sie mit gebilde 190 ENTER. Die Bedeutung dieser Befehle sollten Sie jetzt kennen: fd,bk,lt,rt,cs,make. T.B.
```

Kalorien- Wächter



Fast jeder hatte schon einmal Probleme mit dem Gewicht. Es gibt etliche Diätpläne. Das wird jetzt anders. Mit dem Atari-Computer ist es möglich, seinen ganz persönlichen Diätplan aufzustellen. Das Programm Kalorien-Wächter hat 11,5K und läuft somit mit allen Atari-Computern.

Nach dem Titel gibt man seine persönlichen Daten ein (Größe und Gewicht). Auch das Geschlecht ist wichtig, da der Computer nun das Idealgewicht berechnet. Es liegt bei Frauen 15% und bei Männern 10% unter dem Normgewicht (Normgewicht = Körpergröße in cm minus 100).

Nun wird der Kalorienbedarf berechnet. Nach einigen Texterklärungen beginnt das Hauptprogramm. Es erscheint zunächst ein Speisemenü. Mit Return wird in diesem Menü geblättert. Nach der letzten Seite wird wieder die erste Seite angezeigt. Jeder Speise ist ein Buchstabe zugeordnet. Drückt man den entsprechenden Buchstaben, wird umgeschaltet. Nun beantwortet man die Frage nach der Menge für die ausgewählte Speise. Die Menge wird in Gramm eingegeben. Es sei denn, es wird Stück oder eine andere Größenbezeichnung mit der Speise angezeigt. Als Anzeige erhält man nun den Kaloriengehalt der Speisemenge. Weiterhin wird die Kaloriengesamtannahme angezeigt.

Mit Return kommt man wieder in das Speisemenü. Wird hier „*“ gedrückt, endet das Programm. Zuvor wird jedoch noch angezeigt, wie man die zu sich genommenen Kalorien abbauen kann.

Liegt man unter seinem Kalorienbedarf, zeigt der Computer an, wieviel man ungefähr abnimmt.

Die Speisen kann man nach Belieben und Speichermöglichkeit ergänzen. Der letzte Datawert muß jedoch „ENDE“ und einen Zahlenwert enthalten!
W.B.

VARIABLENLISTE

A\$=	Auswahl Buchstabe
SPEISE\$=	Speise im Menü
MW\$=	männlich/weiblich
T=	Tastatureingabe
PX=	Position x im Menü
PY=	Position y im Menü
KL=	Kalorienwert
NX=	Buchstabe im Menü
GZL=	Eingabe der Speisemenge
AUSGABE=	Gesamtkaloriengehalt
TOTAL=	Totale Kalorieneinnahme
TKB=	Tatsächlicher Kalorienbedarf
KG=	Körpergewicht
SCHLAF=	Verbrauchswert beim Schlafen
BEWEGUNG=	Verbrauchswert bei Bew.
ANSTRENGUNG=	Verbrauchswert bei Anstren.
EB1=	Kalorienabbau bei Schlaf
EB2=	Kalorienabbau bei Bewegung
EB3=	Kalorienabbau bei Anstrengung
PZ=	Prozentabzug bei m/w
CM=	Körpergröße
NG=	Normgewicht
IG=	Idealgewicht
UG=	Unter- bzw. Übergewicht
PR=	Prozent Unter/Übergewicht
KB=	Kalorienbedarf
JB=	Joulebedarf
TJB=	tatsächlicher Joulebedarf
A,B,C,I,J,N=	Zählvariablen

Programmaufbau

```

10 - 1090 HAUPTPROGRAMM
    20- 30 Verzweigung Upro
    40- 100 Erklärung
    110- 360 Aufbau/Abfrage Speisemenü
    370- 750 Wiedergabe/Berechnung Kaloreinwert
    760-1040 Anzeige Abnahme/Verbrauchszeiten
    1060-1090 Upro Speisemenü

2000 - 2720 DATA FUER SPEISEN MIT KALORIENWERTEN

3000 - 3490 UPRO EINGABE
    3010-3090 Eingabe persönlicher Daten
    3100-3410 Ausgabe Gewichte/Kalorienbedarf
    3420-3490 Texterklärung

4000 - 4130 UPRO TITEL
    4010-4060 Aufbau Titel
    4080-4130 Farbwechsel
    
```

KALORIE 1 PG 1

```

1 REM *****
2 REM *   KALORIENMAECHTER *
3 REM *   (C) 1985 *
4 REM *   WOLFGANG BURGER *
5 REM *   *
6 REM *   *
7 REM *   *
8 REM *****
10 REM *** HAUPTPROGRAMM ***
20 GOSUB 4000:REM *** TITEL ***
30 GOSUB 3000:REM *** DATENEINGABE ***
40 GRAPHICS 0:SETCOLOR 4,4,5:SETCOLOR
  2,4,1:SETCOLOR 1,0,15:POKE 752,1:POKE
  82,2
50 ? :? :? :? " DIESES PROGRAMM ERKLAER
  RT DIR, WIEVIELE KALORIEN DU Z
  U DIR NIMMST."
60 ? :? " DU BEKOMMST GLEICH EIN
  SPESIMENU AUF DEM BILDS
  CHIRM"
70 ? " ZU SEHEN,DASS DIR DIE AUSWAHL V
  ON VERSCHIEDENEM SPEISEN ERLAUBT."
80 ? :? :? " ES WIRD DANK DER GESAMTWH
  ERT BERRECHNET UND MIT DEM KA
  LORIENBEDARF VERGLICHEN."
90 ? :? :? :? "DRUECKE DIE TASTE FUE
  R DAS MENUE"
100 GET #1,T
110 GRAPHICS 0:SETCOLOR 4,2,6:SETCOLOR
  1,0,15:SETCOLOR 2,2,2:POKE 752,1
120 CLOSE #1
130 OPEN #1,4,0,"K":POKE 82,0
140 DIM A$(1),SPEISE$(39)
150 GOTO 170
160 GRAPHICS 0:SETCOLOR 4,2,6:SETCOLOR
  1,0,15:SETCOLOR 2,2,2:POKE 752,1
170 PX=0:PY=3:GOSUB 1060
180 FOR I=1 TO 18
190 READ SPEISE$,KL
200 IF SPEISE$="ENDE" THEN 230
210 POSITION PX,PY:? CHR$(I+192);"-";S
  PEISE$:PY=PY+1
220 NEXT I
230 REM HINWEISZEILE AUSGEBEN
240 IF PEEK(20)>60 AND PEEK(20)<120 TH
  EN POSITION 2,23:? "DRUECKE BUECHER FU
    
```

```

ER SPEISE ";
250 IF PEEK(20)>120 AND PEEK(20)<180 T
  HEN POSITION 2,23:? "DRUECKE RETURN FU
  ER NAECHSTE SEITE";
260 IF PEEK(20)>180 THEN POSITION 2,23
  :? "DRUECKE ESC FUER ENDE
  ";
270 IF PEEK(764)=255 THEN 240
280 GET #1,A:A$=CHR$(A):IF (A$("<A" OR
  A$)"R") AND A$(<)CHR$(155) AND A$(<)"*"
  THEN 240
290 IF A$(<)CHR$(155) THEN 320
300 NX=NX+1:IF SPEISE$="ENDE" THEN RES
  TORE :NX=0
310 GOTO 170
320 RESTORE
330 IF A$="*" THEN 760
340 FOR I=1 TO NX*18+A$(A$)-64
350 READ SPEISE$,KL
360 NEXT I
370 GRAPHICS 0:SETCOLOR 4,3,2:SETCOLOR
  2,13,2:SETCOLOR 1,0,15:POKE 752,1
380 POSITION 2,2:? "SPEISE: ";SPEISE$
390 POSITION 0,3:FOR I=0 TO 39:? CHR$(
  18);:NEXT I
400 POSITION 1,5:? "Kaloriengehalt:
":POSITION 1,6:? "SpeGramm/Menge/S
":"
410 POSITION 1,7:? "
  |":REM CNTRL V,CNTRL B
420 POSITION 1,8:? "
  |":REM CNTRL M
430 POSITION 7,7:? KL;" Kcal"
440 POSITION 0,10:FOR I=0 TO 39:? CHR$(
  18);:NEXT I
450 POSITION 1,21:? "FALLS VERTAPPT:
NUMMERN EINGEBEN"
460 POSITION 1,22:? "GEBE GESAMTMENGE
  DER SPEISE EIN";
470 TRAP 470:POKE 752,0:INPUT GZL:POKE
  752,1:TRAP 40000
480 IF GZL=0 THEN 750
490 IF GZL<0 THEN ? "↓":GOTO 370
500 AUSGABE=GZL*KL
510 POSITION 23,5:? "Kaloriengehalt:
  :POSITION 23,6:? "gesamt="
  |
520 POSITION 23,7:? "
  |":REM CNTRL V,CNTRL B
530 POSITION 23,8:? "
  |":REM CNTRL M
540 POSITION 28,7:? AUSGABE;" Kcal"
550 TOTAL=TOTAL+AUSGABE
560 POSITION 1,12:? "TOTAL KALORIEN IN
  :POSITION 1,13:? "NAHME BIS ZUR
  |"
570 POSITION 1,14:? "
  |":POSITION 1,15:? "
  |"
580 POSITION 7,14:? TOTAL;" Kcal"
590 IF TOTAL<TKB THEN GOSUB 620
600 IF TOTAL>TKB THEN GOSUB 660
610 GOTO 720
620 POSITION 22,12:? "NOCH KEINE
  :POSITION 22,13:? "GEWICHTZUNAHME"
630 POSITION 22,14:? "
  |"
640 POSITION 22,15:? "SIND NOCH BEREI
  |"
650 POSITION 26,14:? TKB-TOTAL;" Kcal"
  :RETURN
    
```

```

660 POSITION 22,12:? " ACHTUNG!
":POSITION 22,13:? " GEWICHTZUNAHME"
670 POSITION 22,14:? " |
"
680 POSITION 22,15:? " ZURUECK!
"
690 POSITION 26,14:? TOTAL-TKB;" Kcal"
700 FOR I=15 TO 0 STEP -0.2:SOUND 0,35
,14,I:NEXT I
710 SOUND 0,0,0:RETURN
720 POSITION 1,21:? "
"
730 POSITION 1,22:? " DRUECKE RETURN
UM WEITERZUMACHEN...";
740 GET #1,A:A$=CHR$(A):IF A$( < > )CHR$(15
5) THEN 740
750 RESTORE :NX=0:GOTO 160
760 GRAPHICS 0:SETCOLOR 2,3,1:SETCOLOR
4,13,2:SETCOLOR 1,0,15:POKE 752,1:POK
E 82,2
770 SCHLAF=KG*1.39:BEWEGUNG=KG*1.71:AN
STRENGUNG=KG*2.5
780 GOSUB 800
790 GOTO 810
800 EB1=TOTAL/SCHLAF:EB2=TOTAL/BEWEGUN
G:EB3=TOTAL/ANSTRENGUNG:RETURN
810 GOSUB 820:GOTO 880
820 ? :? "UM DIESE KALORIENMENGE ABZUB
AUEN,
MUSST DU :?"
830 ? :? :? ;;(EB1):POSITION 16,5:? " 
STUNDENSCHLAFEN"
840 ? :? "ODER DICH"
850 ? :? :? ;;(EB2):POSITION 16,10:? " 
STUNDENBEWEGEN"
860 ? :? "ODER DICH"
870 ? :? :? ;;(EB3):POSITION 16,15:? " 
STUNDENANSTRENGEN":RETURN
880 ? :? "Dabei ist der taegliche Klal
orienbe- darf nicht beruecksichtigt!"
890 POSITION 8,22:? " DRUECKE EINE TAST
E"
900 GET #1,T
910 IF TOTAL>TKB THEN GOTO 930
920 IF TOTAL<TKB THEN GOTO 990
930 TOTAL=TOTAL-TKB:GOSUB 800
940 ? "K";:GOSUB 820
950 ? :? "Jetzt ist der taegliche Klal
orienbe- darf beruecksichtigt!"
960 POSITION 8,22:? " DRUECKE EINE TAST
E"
970 GET #1,T
980 GOTO 1050
990 ABNAHME=((TKB-TOTAL)/6250)*1000
1000 ? "K"
1010 POSITION 14,10:? " DU BIMMST
":POSITION 14,11:? " GIGA"
1020 POSITION 14,12:? " |
":REM CNTRL V,CNTRL B
1030 POSITION 14,13:? " GRAMM AB
":REM CNTRL M
1040 POSITION 20,12:? INT(ABNAHME):? :
?
1050 END
1060 POKE 82,0:? "K";
1070 ? "++++++"
"
1080 ? " SPETSE-MENU "
"
1090 RETURN

```

```

2000 DATA ALOKOHOL,7
2005 DATA ANANAS,1
2010 DATA APFEL,0.52
2015 DATA APFELKOESE,0.54
2020 DATA BANANE,0.9
2025 DATA BIER,0.48
2030 DATA BIER 1 GLAS (0.21),95
2035 DATA BIER 1 GLAS (0.21),235
2040 DATA BIRNE,0.39
2045 DATA BIMMELKOEHE,0.35
2050 DATA BOHNEN,0.2
2055 DATA BRATENSUSSE,3.5
2060 DATA BROCCOLI,0.33
2065 DATA BROTTARTEN,2.8
2070 DATA BROT (GRAHAM),2.5
2075 DATA BROT (KROKANT),2.5
2080 DATA BROT (VOLLKORN),2.4
2085 DATA BROT (KROKANT),2.6
2090 DATA BUTTER,7.6
2095 DATA BUTTERBROT (SCHNITZEN),150
2100 DATA BUTTERKEKS (1 STUECK),20
2105 DATA BUTTERKEKS,0.35
2110 DATA CHAMPIGNONS,0.26
2115 DATA KONNIGSBROT,1.54
2120 DATA DICKMILCH,0.43
2125 DATA FRUCHTSUSSE,88
2130 DATA EI 1 GRAMM,1.4
2135 DATA FRUCHTSUSSE,1.41
2140 DATA EISCREME (1 KUGEL),82
2145 DATA FRUCHTSUSSE,350
2150 DATA EIWEIS,0.5
2155 DATA ERBEN,0.66
2160 DATA ERDBEEREN,0.4
2165 DATA ERDUSSE,6.33
2170 DATA FETT (TIERISCH),9.3
2175 DATA FRUCHTSUSSE,1.12
2180 DATA FLEISCHWURST (MAGER),1.8
2185 DATA FRUCHTSUSSE,1.94
2190 DATA GEFLUEGEL (GANS),2.3
2195 DATA FRUCHTSUSSE,3.5
2200 DATA GELEEFRUECHTE,2.6
2205 DATA FRUCHTSUSSE,0.1
2210 DATA GRAPEFRUITSAFT,0.28
2215 DATA FRUCHTSUSSE,3.66
2220 DATA GUMMIBAERCHEN(1 ST.),5
2225 DATA FRUCHTSUSSE,0.1
2230 DATA HAFERFLOCKEN,4.02
2235 DATA FRUCHTSUSSE,0.99
2240 DATA HANDKAESE 10%,1.05
2245 DATA FRUCHTSUSSE,3.1
2250 DATA HIMMBEEREN,0.4
2255 DATA FRUCHTSUSSE,1.4
2260 DATA JAGDWURST 7.5%,1.7
2265 DATA FRUCHTSUSSE,2.8
2270 DATA KAESE (GOUDA 30%),2.8
2275 DATA FRUCHTSUSSE,1.25
2280 DATA KALBSSTEAK,1.03
2285 DATA FRUCHTSUSSE,4.8
2290 DATA KAROTTEN,0.36
2295 DATA FRUCHTSUSSE,0.86
2300 DATA KARTOFFELKL 5SE,1.6
2305 DATA FRUCHTSUSSE,30
2310 DATA KAUGUMMI (1ST.),7
2315 DATA FRUCHTSUSSE,5.2
2320 DATA KETCHUP,1
2325 DATA FRUCHTSUSSE,0.57
2330 DATA KNAECKEBROT,3.75
2335 DATA FRUCHTSUSSE,0.14
2340 DATA KONDENSMILCH 2%,0.6

```

```

2345 DATA KOPFSCHNITZ, 0.14
2350 DATA KRABBE, 1
2355 DATA KUCHENSCHNITZ, 1.44
2360 DATA LAKRITZSCHNECKE (1 ST.), 40
2365 DATA LÄMMEL, 0.4
2370 DATA LEBER, 1.43
2375 DATA MAGERBROT, 0.43
2380 DATA MAGERMILCHPULVER, 3.7
2385 DATA MARGARINE, 3.9
2390 DATA MANDEL gebrannt (1 ST.), 10
2395 DATA MARIENKUCHEN, 4.1
2400 DATA MARZIPAN, 4.6
2405 DATA MARIENKUCHEN, 0.4
2410 DATA MEHL, 3.7
2415 DATA MENTHOL, 0.24
2420 DATA METTWURST, 5.3
2425 DATA MICHEN (MAGER), 0.9
2430 DATA MIRABELLE, 0.63
2435 DATA NERKENKUCHEN (1 ST.), 60
2440 DATA NOUGAT, 5.8
2445 DATA OBERKUCHEN, 4.1
2450 DATA OEL, 9.2
2455 DATA ORANGE, 0.5
2460 DATA PAPRIKA rot & gruen, 0.3
2465 DATA PFEFFERMINZTAFELCHEN (1 ST.), 34
2470 DATA PFIRSICH, 0.46
2475 DATA PASTASCHNITZ, 625
2480 DATA POMMES FRITES (1 PORT.), 330
2485 DATA PRAUNE, 0.5
2490 DATA PRALINE (1 ST.), 55
2495 DATA PRALINE MIT MONTKIRSCHEN (1 ST.), 40
2500 DATA PREISSELBEEREN, 0.6
2505 DATA RINDENKUCHEN, 2.3
2510 DATA QUARK (MAGER), 0.88
2515 DATA RAUENKUCHEN, 0.2
2520 DATA REIS, 4
2525 DATA REISKORNERN, 3.7
2530 DATA RINDFLEISCH, 2.05
2535 DATA RINDFLEISCHERUHE, 0.03
2540 DATA ROSTBEEF (MAGER), 2.53
2545 DATA ROLLSCHNITZ, 4
2550 DATA ROSENKOHL, 0.52
2555 DATA ROSENBEEREN, 0.66
2560 DATA ROTKOHL, 0.26
2565 DATA ROTKOHLSCH, 1.12
2570 DATA ROTHEIN (1 GLAS 0.21), 157
2575 DATA SANDGURKE, 0.1
2580 DATA SANDKUCHEN, 4.1
2585 DATA SAUERKIRSCHEN, 0.6
2590 DATA SAUERKRAUT, 0.26
2595 DATA SCHEDEL, 1
2600 DATA SCHINKEN (GEKOCHT), 1.92
2605 DATA SCHINKEN (ROH), 1.44
2610 DATA SCHOKOLADE, 5.5
2615 DATA SCHOKOLADENKUGELN, 5.25
2620 DATA SCHWARZWURZEL, 0.74
2625 DATA SCHNITZ, 9.47
2630 DATA SCHWEINERUECKEN (MAGER), 1.8
2635 DATA SELBENKUCHEN, 105
2640 DATA SELLERIE, 0.38
2645 DATA SPARGEL, 1.95
2650 DATA SPARGEL, 0.2
2655 DATA SPINAT, 0.23
2660 DATA TOASTBROT, 2.6
2665 DATA TOMATEN, 0.2
2670 DATA TOMATENSaft, 0.22
2675 DATA TROCKENSCHNITZ, 4

```

```

2680 DATA WEINBRAND ( 5cl), 120
2685 DATA WEINBRANDBOHNE (1 ST.), 50
2690 DATA WEISSWEIN (0.15l), 105
2695 DATA WIESENKUCHEN, 250
2700 DATA WIRSING, 0.26
2705 DATA ZUCKER, 4
2710 DATA ZWIEBACK, 3.95
2715 DATA ZWIEBEL, 0.6
2720 DATA ENDE, 000
3000 REM *** DATENEINGABE UND BERECHNUNG
      G GEWICHT/KALORIENBEDARF ***
3010 GRAPHICS 0:SETCOLOR 2,6,2:SETCOLOR
      R 4,9,4:SETCOLOR 1,0,15:POKE 752,1:POKE
      E 82,2
3020 DIM MW$(1)
3030 ? :? "GUTEN TAG"
3040 ? "Zu Beginn brauche ich Deine Daten." :?
3050 ? "GESCHLECHT (M/W) : " : INPUT MW$
3060 IF MW$="M" THEN PZ=90
3070 IF MW$="W" THEN PZ=85
3080 TRAP 3080: ? "KOERPERGROESSE IN CM " : INPUT CM
3090 TRAP 3090: ? "GEWICHT IN KILOGRAMM " : INPUT KG:TRAP 40000
3100 POSITION 0,8:FOR I=0 TO 39: ? CHR$(18);:NEXT I
3110 NG=CM-100:IG=NG/100*PZ
3120 POSITION 2,10: ? "Idealgewicht "
3130 POSITION 2,11: ? "
      " : REM CNTRL V, CNTRL B
3140 POSITION 2,12: ? "
      " : REM CNTRL M
3150 POSITION 8,11: ? IG;"kg"
3160 UG=KG-IG:PR=INT(IG*UG/100)
3170 POSITION 22,10: ? "Lebergewicht "
3180 IF UG<0 THEN GOSUB 3210
3190 POSITION 22,11: ? "
      "
3200 GOTO 3230
3210 POSITION 22,10: ? "Untergewicht "
3220 RETURN
3230 POSITION 22,12: ? "
      "
3240 POSITION 25,11: ? UG;"kg =";PR;"%"
3250 KB=IG*32:TKB=KG*32:JB=INT(KB*4.184):TJB=INT(TKB*4.184)
3260 POSITION 2,14: ? "Kalorienbedarf "
3270 POSITION 2,15: ? "bei Normalgew."
3280 POSITION 2,16: ? "
      "
3290 POSITION 2,17: ? "
      "
3300 POSITION 2,18: ? "
      "
3310 POSITION 6,16: ? KB;"kcal"
3320 POSITION 4,17: ? "oder ";JB;"kJ"
3330 POSITION 22,14: ? "Kalorienbedarf "
3340 POSITION 22,15: ? "Kalorienbedarf "
3350 POSITION 22,16: ? "
      "

```

```

3360 POSITION 22,17:? "
|
3370 POSITION 22,18:? "
|
3380 POSITION 26,16:? TKB;"kcal"
3390 POSITION 24,17:? "oder ";TJB;"kJ"
3400 POSITION 0,19:FOR I=0 TO 39:? CHR
$(18);:NEXT I
3410 ? :? " DRUECKEN EINE TASTE
"
3420 OPEN #1,4,0,"K":GET #1,T
3440 ? "5":POSITION 2,10
3450 ? :? "Der in diesem Programm bere
chnete Kalorienbedarf bezieht sich
auf leich-te bis Mittlere Taetigkeit.
"
3460 ? :? "Um 1 kg abzunehmen, musst D
U 6000 bis 6500 Kalorien eins
paren."
3470 ? :? :? " DRUECKEN EINE TASTE
"
3480 GET #1,T

```

```

3490 RETURN
4000 REM TITEL
4010 GRAPHICS 2:SETCOLOR 2,0,0:SETCOLO
R 0,2,6:SETCOLOR 1,12,6:A=2:B=12:C=4:N
=0
4020 POSITION 0,1:? #6;"*****
*****"
4030 POSITION 6,3:? #6;"Ka[Or]En"
4040 POSITION 6,5:? #6;"Wa[Ch]Er"
4050 POSITION 0,7:? #6;"*****
*****"
4060 POKE 752,1:? :? "[C] 1985 "?: "Wo
lfgang Burger"
4070 REM WECHSEL FARBE
4080 FOR I=1 TO 40
4090 SETCOLOR 0,A,6:SETCOLOR 1,B,6:SET
COLOR 3,C,6:N=N+1:IF N>3 THEN N=1
4100 IF N=1 THEN A=12:B=4:C=2
4110 IF N=2 THEN A=4:B=2:C=12
4120 IF N=3 THEN A=2:B=12:C=4
4130 FOR J=1 TO 10:NEXT J:NEXT I:RETUR
N

```

BÜCHERECKE



Titel: Programmieren in Forth, Autor: Leo Brodie, Verlag: Hanser, Preis: 48,— DM, 310 Seiten.

Mit diesem Fachbuch macht es Spaß, diese in der letzten Zeit populär gewordene Programmiersprache zu erlernen. Dafür sorgen nicht nur die zahlreichen lustigen Karikaturen, sondern auch der saloppe Stil des Autors Leo Brodie. Bevor man auf die vier Grundrechenarten eingeht, werden sehr ausführlich der Stapel und seine Besonderheiten erklärt, denn er bildet sozusagen das Herz dieser Sprache. Neben der Arithmetik werden

alle wichtigen Sprachelemente gründlich behandelt. Insbesondere aber wird ab und zu zum besseren Verständnis die Struktur des Parameterstapels abgebildet. Dieser Extraservice ist sehr begrüßenswert, da der Stack etwas kompliziert ist, teilweise hervorgerufen durch die umgekehrte polnische Notation. Dieses Buch beschränkt sich nicht nur auf das Forth-Vokabular, sondern geht auch auf die interne Arbeitsweise dieser etwas ungewöhnlichen Hochsprache ein. Ohne überlastet zu werden, weiß der Leser sehr schnell, wie der Interpreter oder der Compiler funktioniert. Diese Grundkenntnisse werden noch vertieft, indem der Autor zeigt, wie man einen Forth-compiler erweitert. Hier nutzt Leo Brodie alle Vorteile dieser transparenten und ausbaufähigen Sprache aus. Hat man dieses Werk durchgelesen, so besitzt man Systemkenntnisse über Forth, die man quasi lückenlos bezeichnen kann. Jetzt aber zu den Nachteilen. Die Beispiele sind etwas einseitig gewählt worden. Spiele mit Worten sind zwar teilweise recht erheiternd, jedoch könnten kleine Arithmetikprogramme den Sprachumfang besser ausnutzen. Auch vermißt man komplexere Beispiele im Kapitel 'Forth Intern'. Man hat allgemein den Eindruck, daß der Autor

sich über den Standard nicht hinauswagt. Das ist mir insofern unverständlich, als er durch seine Arbeit bei der Forth Incorporation in den Vereinigten Staaten wirklich über ein enormes Fachwissen verfügt. Wie man allgemein Assembler in Forth-Software einsetzt, wird leider verschwiegen. Eigentlich schade, denn in diesem Punkt besitzt die Sprache der vierten Generation, Forth, Vorteile, die einzigartig sind. Warum Brodie so ausführlich die Editorfunktion behandelt hat, ist mir ebenfalls schleierhaft, da es hier keinen vorgeschriebenen Standard gibt, was die Hersteller der verschiedenen Forthimplementationen auch ausnutzen. Daher sind die zahlreichen Editorbefehle nicht nur überflüssig, sondern verwirren auch gleichzeitig den Leser. Darum kann ich nur jedem Käufer des Buches empfehlen, dieses Kapitel zu überschlagen und dafür seine Bedienungsanleitung ausführlich zu lesen.

Fazit: Wer den Einstieg in Forth erlangen will, ist mit diesem Werk hervorragend beraten. Sind Sie aber Fortgeschrittener und wollen auf einer höheren Ebene programmieren, so müssen Sie sich leider auf dem umfangreichen Markt der Fachliteratur umsehen, da Sie an dieser Stelle vom Autor im Stich gelassen werden. KUH



BASIC-COMPILER Für Ihren SCHNEIDER CPC 464

Version für CPC 664 und CPC 6128 in Kürze lieferbar

REPEAT
UNTIL

mit
FLIESSKOMMAARITHMETIK

Der neue Basiccompiler für den Schneider CPC erreicht bis dreißigfache Geschwindigkeitssteigerung. Ein Programm zum Löschen des Bildschirmspeichers benötigt im normalen Schneiderbasic etwa 55 Sekunden Laufzeit.

Dasselbe compilierte Programm läuft ca. 1,8 Sekunden. Der Compiler ist in der Lage, eindimensionale Felder zu definieren, Strings zu verarbeiten und Integerarithmetik von -32768 bis +65535 zu verarbeiten.

Neben den reinen Integerzahlen ist der Compiler auch in der Lage, Fließkommazahlen zu verarbeiten. Neben den üblichen arithmetischen Operationen stehen eine ganze Anzahl von Stringoperationen zur Verfügung. Der Compiler kennt FOR-NEXT-Schleifen, sowie WHILE-WEND und als zusätzliche Erweiterung REPEAT-UNTIL-Schleifen.

Die vom Compiler erzeugten Maschinenprogramme können als Objekt Code abgespeichert werden und sind ohne den Compiler selbständig lauffähig. Das durch den eigenen Editor erstellte Programm kann aber auch als normales Textfile abgespeichert werden. Der Compiler kann bestehende Dateien und Programme lesen, die im ASCII Format auf Kassette oder Diskette abgespeichert sind. So ist es ohne weiteres möglich, schon bestehende Programme zu compilieren.

Ja, Ihr Angebot hat mich überzeugt
und ich bestelle

Einsenden an:

_____ BASIC COMPILER (DISK) 89,— _____
Anzahl Einzel Gesamt

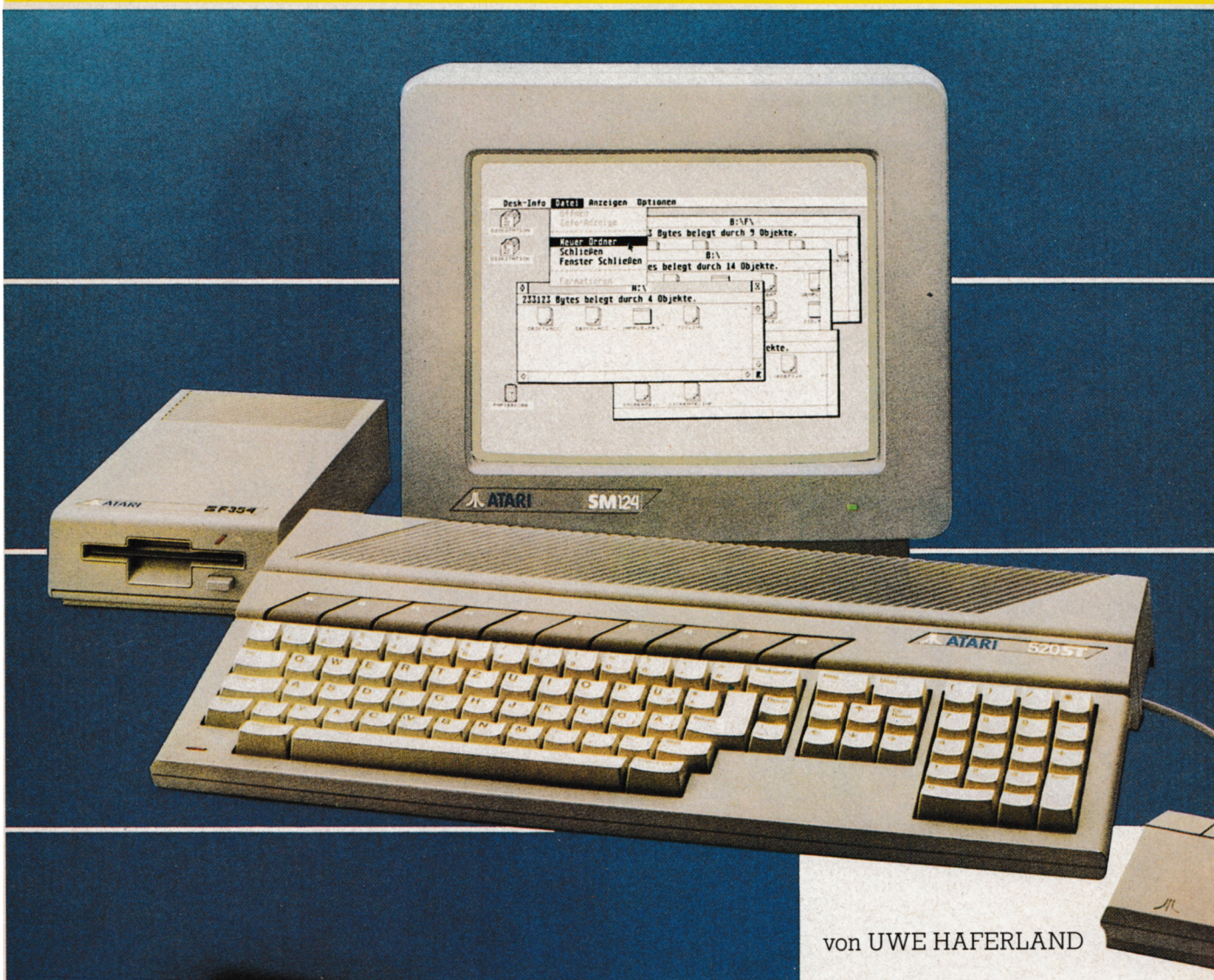
_____ BASIC COMPILER (CASS) 79,— _____
Anzahl Einzel Gesamt

SOFTWARE TEAM
Joachim Günster
Mühlenstr. 12
5431 BODEN

Versandwunsch bitte angeben: Bargeld liegt bei
 Verrechnungsscheck beigefügt per Nachnahme
Bei Versand per NN werden 5,— DM Versandkosten pauschal
erhoben

BASIC

mit Handbremse



von UWE HAFERLAND

Wir haben schon in Ausgabe Nr. 10 der COMPUTER TEAM vermutet, daß das 520 ST BASIC für Überraschungen sorgen wird. Leider haben wir ausnahmslos Recht behalten. Aber lesen Sie selbst, was die Vorabversion von Digital Research leistet.

Wir wollen es noch einmal betonen. Es wurde eine noch nicht auslieferungsreife Vorabversion getestet. Das bedeutet, wenn dieses

BASIC voraussichtlich im Dezember auf dem Markt erscheint, wird es noch einige zusätzliche Verbesserungen aufweisen. Selbstverständlich berücksichtigen wir diese Umstände und werden uns daher nicht auf Mängel beziehen, die gegenwärtig noch verbessert werden. Schließlich wäre es unfair, sich über nicht vorhandene Befehle und Anweisungen aufzuregen, die zum Systemabsturz führen. Sol-

che Mißstände sind normal und werden bis zur endgültigen Version behoben. Doch nun zu den einzelnen Eigenschaften.

16 Bit-Träume

Der Atari 520 ST weist hardwaremäßige Leistungsmerkmale auf, die, sieht man einmal von Amiga ab,

ihresgleichen suchen. Der schnelle 68000 Prozessor könnte BASIC-Programme auf Geschwindigkeiten beschleunigen, bei denen es sich kaum noch lohnt, Compilersprachen wie Pascal oder Fort zu erlernen. Ein BASIC-Programm auf einem 68000er System müßte ungefähr genauso schnell sein wie Pascal-Software auf 8 Bit-Computern.

Die traurige Wirklichkeit

Wir konnten es kaum erwarten, dieses BASIC auszuprobieren. Schnell wurde der 520 ST zusammen mit dem Rennpferd Schneider CPC 464 aufgebaut. Schließlich erwartete man ein hochinteressantes Rennen identischer BASIC-Programme. Auf solche zweifelhaften Routinen wie den Benchmarktest wurde verzichtet. Statt dessen verwendeten wir umfangreichere Software wie Primzahlen, Integralrechnung, Dualzahlen. Aber auch Grafiken wurden erstellt, um einen objektiven Eindruck zu gewinnen. Dann ging es los. Beide Computer wurden gleichzeitig gestartet. Was dabei herauskam, war vollendete Verwirrung. Man konnte nur schwer erraten, in welchem Gerät sich der 16 Bit-Motorola-Prozessor befand. Der 520 ST kroch dem tapferen Schneiderlein immer nur um Nasenlängen voraus. Genauer gesagt war der CPC 464 um den Faktor 1,6 langsamer. Ein Wert, mit dem sich das 520 ST BASIC nicht rühmen darf. Dabei verlief der Test insofern unfair, als der Atari drei bis vier Stellen ungenauer rechnete. Beide Computer mit gleicher Genauigkeit gegeneinander antreten zu lassen, war nicht realisierbar. Gott sei Dank auch! Angenommen, das Atari-BASIC hätte die gleiche Anzahl an Dezimalstellen wie der Schneider besessen, dann wäre der 520 ST seinem Konkurrenten mit Sicherheit hinterher gehinkt. Zwar läßt sich das Atari-BASIC auf eine gesteigerte Genauigkeit umstellen, jedoch wäre es in diesem Fall schon zu genau, um noch einen objektiven Vergleichstest liefern zu können. Was insbesondere auffiel, war die extrem langsame Ausgabe. Der Scrollvorgang verläuft hierbei so langsam, daß selbst der schon gemächlich

ABS	EOF	LOGIO	RSET
ALL	EQU	LPOS	RUN
AND	ERA	LPRINT	SAVE
AS	ERASE	LSET	SEG
ASC	ERL	MERGE	SIGN
ATN	ERR	MID\$	SIN
AUTO	ERROR	MKD\$	SPACES
BASE	EXP	MKI\$	SPC
BLOAD	FIELD	MKS\$	SQR
BREAK	FIX	MOD	STEP
BSAVE	FLOAT	NAME	STOP
CALL	FOLLOW	NEW	STRING
CDBL	FOR	NEXT	STR\$
CHAIN	FRE	NOT	SWAP
CHR\$	GET	OCT\$	SYSTEM
CINT	GO	OLD	TAB
Circle		ON	
CLEAR	GOSUB	OPEN	TAN
CLOSE	GOTO	OPTION	THEN
COMMON	HEX\$	OR	TO
CONT	IF	OUT	TRACE
COS	INKEY\$	PEEK	TRON
CSNG	INP	PLOT	TROFF
CVD	INPUT	POKE	UNBRK
CVI	INPUT\$	POS	UNFOLLOW
CVS	INSTR	PRINT (= '?')	UNTRACE
DATE	INT	PRINT#	USING
DEF	KILL	PUT	USR
DEFDEL	LEFT\$	RANDOMIZE	VAL
DEFINT	LEN	READ	VARPTR
DEFSNG	LET	REM	WAIT
DEFSTR	LINE	RENUM	WEND
DELETE	LIST	REPLACE	WHILE
DELETED	LLIST	RESET	WIDTH
DIM	LOC	RESTORE	WRITE
DIR	LOF	RESUME	WRITE#
DO	LOG	RETURN	XOR
EDIT	LPRINT	RIGHT\$	
ELSE		RND	
END		SOUND	

arbeitende Sinclair Spectrum dem 520 ST in diesem Punkt schon davonrennt. Wir hoffen, daß diese gravierende Schwachstelle verbessert wird, da sonst die Ausgabe von Zahlenkolonnen oder Texten das Gesamte enorm aufhält. Bedenkt man, daß sich im Schneider ein armseliger 8 Bit-Prozessor abquält, der neben seinen aufwendigen Berechnungen auch dauernd zwischen RAM und ROM hin und her schalten muß, hätte das 520 ST BASIC ungefähr um den Faktor 15-20 schneller sein müssen als sein Herausforderer. Man kann bloß hoffen, daß die Firma Schneider (genauer gesagt Amstrad) nicht eines Tages den CPC 464-Prozessor (= Z 80) mit acht Megahertz taktet, statt bisher vier. Dann möchten wir den Vergleichstest nicht noch einmal miterleben, da

sich hierbei eine Katastrophe für das Atari-BASIC anbahnen würde.

Die Gründe

Wie es schon die Überschrift ausdrückt, ist die Wurzel allen Übels im BCPL-Interpreter zu suchen. Normalerweise wird ein Interpreter in Maschinensprache geschrieben, damit die Ausführungsgeschwindigkeit ein Maximum erreichen kann. Schließlich muß er Ihre Befehle in Maschinencode übersetzen und anschließend diesen sofort ausführen. Sie können sich doch sicherlich leicht vorstellen, daß ein gemächlich arbeitender Übersetzer die Ausführungsgeschwindigkeit Ihrer Befehle senkt. Wahrscheinlich aus Zeit-

gründen konnte man ihn nicht in Maschinensprache definieren, sondern mußte auf eine höhere Programmiersprache ausweichen. Dabei entschied man sich für BCPL, eine Sprache, über die sich nur schwer Informationen finden lassen. Sie wurde 1970 von Martin Richards entwickelt und kann als Vorläufer von 'C' betrachtet werden. Dabei ist sie längst nicht so flexibel und leistungsfähig wie 'C'. Wie man ausgerechnet auf BCPL kam, bleibt uns ein Rätsel. Selbst wenn man eine andere Compilersprache genommen hätte, wäre das Atari-BASIC kaum schneller. Man darf einen Interpreter grundsätzlich nicht in einer Compilersprache definieren. Zwar entsteht nach der Compilierung Maschinencode, der aber niemals so schnell sein kann wie direkt programmierte Maschinensprache. Dabei entstehen Unterschiede um den Faktor zehn bis dreißig. Und

Aussichten

Zwar weist die Atari-Niederlassung in Raunheim darauf hin, daß die endgültige BASIC-Version eine gesteigerte Geschwindigkeit gegenüber dem Prototypen aufweisen wird. Nur, viel läßt sich nicht mehr verbessern, da die Grenzen eines Compilers generell nicht überwunden werden können und folglich dieser immer eine Bremse darstellen wird. Außerdem muß man die bis Dezember '85 vorhandene knappe Entwicklungszeit nutzen, die fehlenden Befehle zu implementieren und diejenigen, die bisher noch nicht korrekt funktionierten, zu korrigieren. Da bleibt also kaum noch Zeit übrig, den bisher vorliegenden umfangreichen (150 Kilobyte!) Maschinencode zu optimieren, um etwas bessere Ausführungszeiten zu erhalten. Sollte

Anweisungen. Die Neunummerierung der BASIC-Zeilen mit Hilfe von 'RENUM' ist ebenfalls möglich. Auch wird die Verarbeitung von Strings mit einer Vielzahl von Anweisungen unterstützt. Hervorragend ist die gegebene Option, die Genauigkeit von Gleitkommazahlen festzulegen. Insbesondere mathematisch orientierte Programme können hieraus großen Nutzen ziehen. Sollten sich mal in Ihrem BASIC-Programm Fehler befinden, keine Angst, an dieser Stelle werden Sie nicht verlassen. 'TRACE' und 'TRON' helfen Ihnen bei der Suche nach den Übeltätern. Insgesamt gewinnt man den Eindruck, daß das hervorragende Schneider-BASIC Pate stand und deshalb auch bei dieser Version keine Wünsche offen bleiben. Sehen Sie sich bitte auch die tabellarische Aufstellung der bisher vorhandenen Kommandos am Schluß dieses Berichtes an.

Einige typische Laufzeiten (Auszug)

	Spectrum(+)	CPC 464	Atari 520 ST
Primzahlen 3-300	248 s	103 s	60 s
Grafikprogramm	31 s	6 s	3 s
Dualzahlen 0-40	180 s	58 s	37 s

dies ist genau der Geschwindigkeitsverlust unserer BASIC-Version! Außerdem haben wir dem BCPL-Compiler zu verdanken, daß die gesamte BASIC-Implementierung sage und schreibe etwa 150 Kilobyte an Speicherplatz verschlingt! Es ist wirklich schade, daß dem BASIC eine Bremse namens 'BCPL' eingebaut wurde. Einen Compiler darf man in jeder beliebigen Programmiersprache schreiben, da die Compilierung eine einmalige Angelegenheit ist und nur der fertig übersetzte und abgespeicherte Maschinencode zum Laufen gebracht wird. Selbst wenn man während der Compilierung noch Dallas genießen kann, so hat dies auf die Ausführungsgeschwindigkeit des entstandenen Maschinenprogramms keinen Einfluß. Ein Interpreter dagegen, der den Maschinencode gar nicht erst abspeichert, sondern unverzüglich ausführen muß, ist gezwungen, mit Maximalgeschwindigkeit zu operieren. Und da ist ein in einer Hochsprache definierter Interpreter fehl am Platz!

die endgültige Version gravierende Verbesserungen aufweisen, werden wir selbstverständlich in einem Zusatztest darauf hinweisen. Allerdings können wir uns nur schwer vorstellen, daß in der verbleibenden kurzen Entwicklungszeit noch großartig Verbesserungen vorgenommen werden.

Befehlssatz

Dieser ist wirklich einsame Spitze. Grafik und Sound können ohne Peek und Poke angesprochen werden. Gleichzeitig sind Anweisungen vorhanden, die die Interrupttechnik unterstützen. So ist ohne weiteres Multitasking möglich, das heißt die scheinbare gleichzeitige Verarbeitung zweier verschiedener Programme. Strukturiertes Programmieren, wie man es in Pascal gewohnt ist, wird durch Anweisungen wie 'WHILE...WEND' (= END) unterstützt. Selbstverständlich fehlen auch keine 'IF..THEN.....ELSE'

Funktionstasten

Erstmals werden auch die zehn frei definierbaren Tasten vom BASIC benutzt. Eine aufrufbare Menükarte teilt Ihnen die genaue Belegung mit. Zum größten Teil sind diese zehn Tasten für Editierfunktionen reserviert, aber auch Laden und Abspeichern sind mit einem einzigen Tastendruck möglich.

Fazit: Sieht man mal von der reduzierten Geschwindigkeit des BASICs ab, kann man die Version insgesamt als gut beurteilen. Ein hervorragender Befehlssatz ohne Kompromisse und letztendlich der Komfort von GEM lassen einfach kein schlechteres Urteil zu. KUH

Die Januar-Ausgabe gibts ab 27. 12.

**VERGESSEN SIE ALLES,
WAS SIE BISHER KENNEN !
JETZT GIBT ES RH-DAT**

RH-DAT ist die Lösung Ihrer Probleme!
RH-DAT ist ein Dateiverwaltungsprogramm mit FREIER Maskendefinition (max. 21 Felder / max. 70 Zeichen pro Feld)
RH-DAT hält Ihre Daten im RELATIVEN Zugriff
RH-DAT kann nach jedem beliebigen Datenfeld suchen
RH-DAT findet einen Datensatz bei der Suche im Indexfeld selbst bei voller Daten (503/1003 Datensätze) in max. 4 Sekunden (Regelmaß 1-2 Sekunden)
RH-DAT ist ein CP/M-Programm



Sie werden staunen, was RH-DAT noch alles kann !!!

RH-DAT ist erhältlich für den CPC 464, 664, 6128, 8256 auf 3"- und 5 1/4"-Disk

PREIS *BAS GIBT'S DOCH NICHT!* 503 Datensätze 79,- DM
1003 Datensätze 89,- DM

RH-BÜRO

Textverarbeitung, Adressverwaltung und Mail-Merge
Testbericht in
CPC August 85, S. 17
NUR Disk 3"

*Preislenkung
inkl. neuem Programm
Laufbandwerbung
Neuer Preis 59,- DM*

Preis 99,- DM

TEL. BESTELLUNG
0211/5065-213



KOMPLETT-PAKETE	
1. RH - ADW (Komfortable Adressverwaltung)	40,- DM
2. RH - TEXT (Super-Textverarbeitungsprogramm)	40,- DM
3. RH - SYNTHESEIZER (Super-Synthesizerprogramm)	40,- DM
4. RH - VOKABELLA (Ausgerolltes Vokabel-Lernprogramm)	30,- DM
5. RH - GRAPH (Klasse Funktionsplotter)	30,- DM
	180,- DM

Jetzt als KOMPLETT-PAKET 50,- DM
Oder ein anderes Beispiel

1. Drucker-Kabel zum Anschluß eines Druckers mit CENTRONICS-Schnittstelle 58,- DM
 2. Wie Komplet Paket '5 für 1' 180,- DM
- 238,- DM

Jetzt als KOMPLETT-PAKET 75,- DM

INTEGRAL HYDRAULIK

INTEGRAL HYDRAULIK & CO
z. Hd. Herrn Hülshner, Am Hohenloen 108, 4000 Düsseldorf 11.
Tel. 0211/5065-213
Vertrieb von: RH-Software

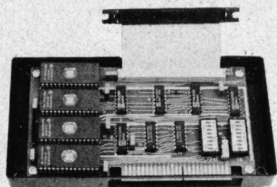
Programme und Softwareautoren für
Schneider 464/664/6128/8256 gesucht!



**Eprommer
Schneider
CPC 464/664**

Universeller EPROM-Programmer 4003

- Programmiert alle gängigen EPROM-Typen (z.B.: 2716,-32,-64,-128,2508,-16,-32,-64...)
- Voll menügesteuerte Software auf Kassette
- Kein Schalten, Stecken oder Löten nötig
- Programmierspannung wird im Gerät erzeugt
- Verbindung zum CPC über Flachbandkabel und Interface-Karte
- Gleichzeitiger Anschluß der Floppy möglich
- Rote und grüne Leuchtdiode zur Betriebs-Art-Anzeige
- Kompl. mit 28 poligem Textool-Sockel
- Fertigergerät DM 289,50 ■ Bausatz mit Anleitung DM 239,-



**EPROM
Karte
64 KByte**

Die ideale Ergänzung für jeden CPC

- Wahlweise bestückbar mit 2 - 64 KByte EPROM-Kapazität
- Arbeitet mit den EPROM-Typen 2716,-32,-64,-128
- Durchgeführter Erweiterungsbus (Floppy kompatibel)
- Autostart von BASIC- und/oder Assembler-Programmen
- Komplett mit umfangreicher und komfortabler Software
- Gleichermaßen für Profis und Einsteiger geeignet
- Fertigergerät DM 249,50 ■ Bausatz mit Anleitung DM 219,50

Drucker für alle CPC

EEDY 100-80 SPEEDY 100-80 SPEDDY 100-80 SP

- 100 Zeichen pro Sekunde schnell
- FX80 kompatibel
- Bis zu 142 Zeichen pro Zeile
- Optionaler Druckerpuffer
- Grafikfähig
- Kein doppelter Zeilenvorschub
- Direkt anschlussfähig
- Internationale Zeichensätze
- Gutes Preis-Leistungsverhältnis
- Kompl. mit deutschem und englischem Handbuch DM 739,-
- Druckerkabel CPC 464/664 DM 35,- ■ CPC 6128 DM 45,-

Preise inkl. Mehrwertsteuer. Alle Artikel ab Lager lieferbar.

**DOBBERTIN
INDUSTRIE-ELEKTRONIK**

Bremsstraße 9, 6835 Brühl, Tel.: (06202) 71417

**Neu Neu Neu Neu Neu Neu Neu Neu
MAXAM**

Assembler - Monitor - Editor
Maxam, der Star unter den Monitoren, zählt zu den modernsten Entwicklungssystemen für den CPC 464/664. (siehe Testbericht CPC Int. 9/85)

**Cassette 69,- Diskette 109,-
im Rom-Modul 239,- DM
TRANSMAT 464/664**

Übertragen Sie Ihre Software problemlos von Cassette auf Diskette. Das Programm beinhaltet einen automatischen Relocator.

Cassette nur 49,- DM
Wir weisen ausdrücklich darauf hin, das urheberrechtlich geschützte Programme nicht oder nur mit Genehmigung des Schutzberechtigten kopiert werden dürfen.

POWER-BASIC 464/664

47 zusätzl. RSX-Befehle
Cassette 49,- DM

C.A.D. 464

Super-Graphikprogramm **Cass. 49,- DM**

PARA (Vortex Diskmanager) **58,- DM**

VORTEX-RAM-SPEICHER-ERWEITERUNGEN

von 64 - 512 KB **ab 275,- DM**

DFÜ-PAKET: Teleterminal 300 S + Akkustikkoppler S 21 D (mit FTZ-Nr.) **nur 378,- DM**

Lightpen (Müller) mit Supermalprogramm **nur 95,- DM**

DRUCKERKABEL **45,- DM**

DISKETTEN-CONTAINER

(HAN) für 40 Disk. 3" **ab 24,- DM**

Ralf Probst

EDV-Service

Fr.-Eberf.-Str. 14, (02136) 32870
4100 Duisburg 17

PHASE 4

**TEXTPROGRAMM - PHASE 4
- DIE NEUE GENERATION**

- ★ Deutsches Programm, deutsches Logo
- ★ In Maschinensprache geschrieben
- ★ Umfangreiches Handbuch

PHASE 4 MIT MAIL MERGE:

- ★ 500 Datensätze
- ★ 40 Sortierkriterien
- ★ Individuelle Massenbriefe
- ★ Dateiverwaltung (z.B. Bibliotheken, Karteien, Register usw.)
- ★ Menügesteuert
- ★ Etikettendruck
- ★ ... und vieles mehr!

PHASE 4:

- ★ Zeichen löschen u. einfügen
- ★ Zeile(n) löschen u. einfügen
- ★ Absätze löschen u. einfügen
- ★ Begriffe suchen u. ersetzen
- ★ Wort oder Zeilenrest löschen
- ★ Zeilen aufbrechen u. anschließen
- ★ Bausteinverwaltung
- ★ Darstellungsbreite bis 240 Zeichen je Zeile
- ★ Fließtexteingabe
- ★ Randausgleich, Blocksatz
- ★ Linker Rand im Text frei definierbar
- ★ Gängige Druckertreiber sind installiert
- ★ Taschenrechnerfunktion u. Formelrechnung
- ★ Tabellenkalkulation im Text
- ★ Seitenumbruch, Zeilenanzahl frei wählbar
- ★ ... und vieles mehr!

Erhältlich auch in Computerfachgeschäften und den Fachabteilungen der Kaufhäuser.

PHASE 4: DM 225,72, PHASE 4 inkl. MAIL MERGE: DM 282,72 zzgl. Verpackung und Versandgebühr (9,50 DM)

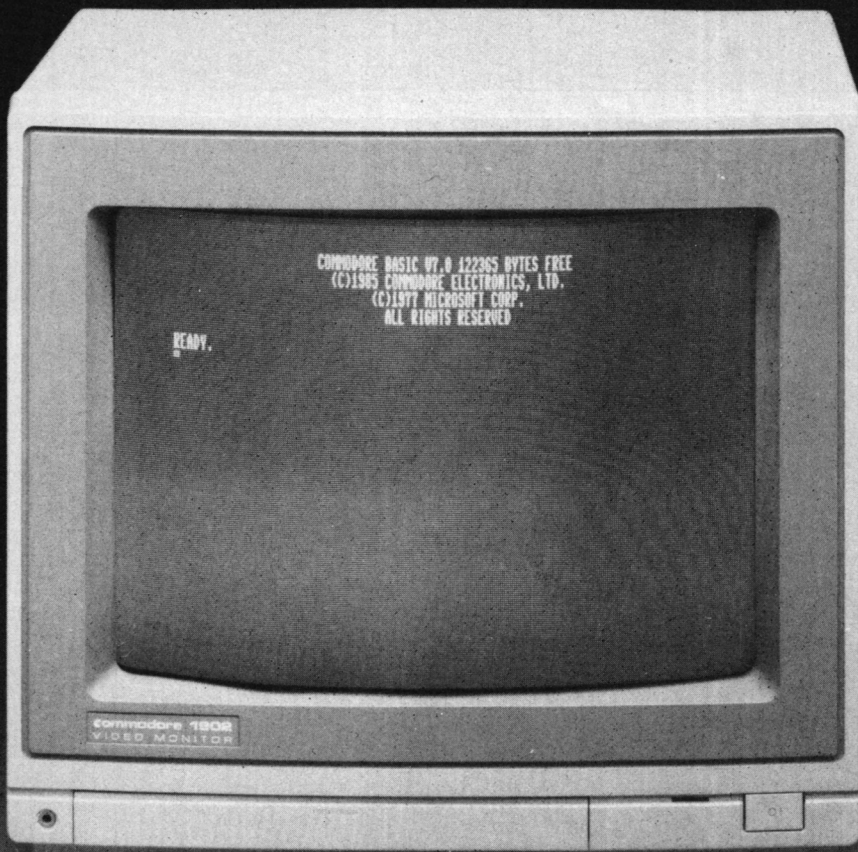
SOFTWAREAGENTUR HEYNS
LIMBECKER PLATZ 7 · 4300 ESSEN 1 · TEL. 0201 / 22 64 17



Für ganz Eilige:
02 01 / 22 64 17

SOFTWAREAGENTUR HEYNS
LIMBECKER PLATZ 7
4300 ESSEN 1

Testbericht Commodore PC-128



Drei Betriebsarten in einem Rechner. Ein völlig neuer Stern fiel vom Computer-Himmel. Ausgeschlafene Konstrukteure bieten eine kompatible Maschine zum Vorgängertyp. Sollte auch dies bei IBM abgeguckt sein? Denn bei dieser Firma herrscht seit vielen Jahren eine Kompatibilität nach unten. Wie kompatibel der neue PC-128 ist und was drei Betriebsarten bringen, lesen Sie in diesem Bericht.

Die Spannung der Interessenten, durch spektakuläre Ankündigung der Presse, wird nun endlich gebrochen. Er ist jetzt lieferbar, der PC-128!

Mit dem PC-128 hat die Firma Commodore wohl einen neuen Clou gelandet. Voll kompatibel zum C-64, ist der 128er wohl die am besten geeignete Aufsteigermaschine für den C-64-Anwender. Neben dem 64er Modus, welcher tatsächlich 100% kompatibel zum alten C-64 ist, bietet der 128er auch noch einen eigenen Modus mit Commodore BASIC 7.0. Dieses 7.0 BASIC stellt alles bisher Dagewesene in den Schatten, selbst die Büromaschinen der Serie 600, 700, 2000, 3000, 4000 und 8000 kommen nicht mehr mit und erblassen vor Neid.

Im Herbst 1984 lautete nicht nur in Las Vegas die Werbung: „Schlechte Nachrichten für IBM und Apple“ (BAD NEWS FOR IBM AND APPLE), sondern fast in ganz USA.

Mit der schlechten Nachricht war natürlich der PC-128 gemeint, der interessanteste der neuen Commodore-Rechner, die auf der CES (Consumer Electronics Show) vorgestellt wurden. Der PC-128 dürfte wohl mit seinen Qualitäten den vielbewährten alten C-64 so leiseweinend ablösen. Wann es nun soweit sein wird, steht wohl noch im Raume! Immerhin sollte man auch an die Einsteiger denken, bei denen der Geldbeutel einem „Zwiebelportemonnaie“ (schaut man hinein, beginnt man zu weinen) ähnelt, und in diesem Fall ist wohl der C-64 interessanter, da dieser bereits für 489,- DM angeboten wird. Sollten Sie jedoch ein C-64-Besitzer sein, nebst reichhaltiger Software, so gibt es keinen besseren Rechner für Sie als den PC-128! Da er nicht nur die Software des 64er verarbeitet, sondern auch noch hardwaremäßig voll kompatibel ist.

Design und Innenleben gut

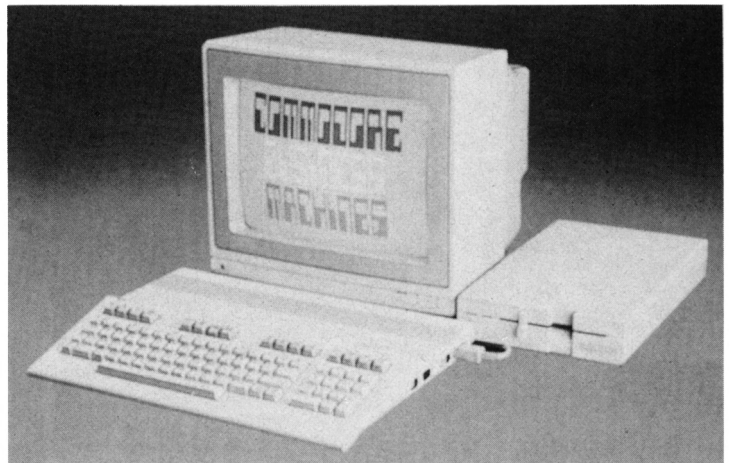
Nicht nur die Computerkonstrukteure, erst recht die Designer waren wohl gut ausgeschlafen, als es hieß: Wir bauen einen neuen Computer. Denn diese Herren dürften wohl nicht nur einen Vogel abgeschossen haben, das war schon ein

mächtiger ADLER. Immerhin läßt sich über Technik und Design nicht so schnell meckern! .. Sie schmunzeln, ... warum? Ach, Ihnen war noch nicht bekannt, daß Sie beim Kauf eines PC-128 gleich drei Computer haben!?! Es ist aber so! Die CPU der PC-128 könnte man Hattric bezeichnen! 64er, 128er und CP/M-Modus (8502 aufwärtskompatibel zum 6510, Z80 B, CP/M-Modus).

128 KByte RAM, als RAM-Disk bis auf 512 KByte erweiterungsfähig. 64 KByte ROM mit einem komfortablen BASIC 7.0 (ca. 140 Anweisungen). Maschinensprache und DOS-Monitor. Umschaltbar von 40 auf 80 Zeichen pro Zeile. Grafikauflösung: 640 mal 200 Punkte. 16 Farben, Sprites einfacher entwerfen

sich natürlich durch eine Komma-Taste in diesem Zehnerblock ergeben. Ich denke dabei an das Eintippen von DATA-Zeilen, wobei man auch hier immer noch zurück in das Haupttastenfeld muß, um das Komma einzugeben, was auf mich etwas störend wirkt. Die oberste Tastenreihe enthält dann noch 4 Funktionstastenblöcke, Block 1: ESC, TAB, ALT, ASCII/DIN-CAPS/LOCK. Der zweite Funktionstastenblock hat die Tasten: HELP, LINE-FEED, 40/80 DISPLAY, NO SCROLL! Der vorletzte Block enthält CURSOR-STEUERTASTEN und der letzte Viererblock die f-Tasten von 1 bis 8. Da die Schreibmaschinentastatur auch der des C-64 100%ig entspricht, hat man großzü-

Der Commodore 128 mit Bildschirm und Floppy



durch den eingebauten Sprite-Editor. 92 Tasten, separater Zehnerblock und acht programmierbare Funktionstasten. Höhe der mittleren Tastenreihe: 3cm. Tiefe: 32,4 cm. Breite: 43,2 cm. Höhe: 5,6 cm. Gewicht: ca. 2,5 kg.

Besser als C-64

Insgesamt 92 Tasten enthält die klar gegliederte und übersichtliche Tastatur. Ein „HÄMMERN“ auf dieser weichgängigen, mit einem kaum spürbaren Anschlag, aber mit Fingermulden versehenen Tastatur wäre nicht angebracht. Aber alleine der Anblick dieser Tastatur läßt die Augen eines jeden Commodore-Fan's strahlen. Rechts neben dem Haupttastenfeld der lang ersehnte abgesetzte Zehnerblock, mit Plus, Minus, Punkt und Enter-Taste. Ja, Sie haben richtig gelesen, mit ENTER-Taste! Im Haupttastenfeld heißt diese natürlich RETURN-Taste. Ein vollendeter Traum hätte

gigerweise beim PC-128 nun 6 Cursor-Steuertasten.

Extras ohne Aufpreis

Kommen wir zum berühmten DRUM-HERUM. An der rechten Seite finden wir 100% C-64. Zwei Joy-Stick-Ports. Aber nun kommt der USUS-KNACKTUS: Nicht nur für Joy-Sticks sind diese Ports gedacht. Man munkelt, daß bei Commodore Mäuse gezüchtet werden. Zur Zeit ist jedoch noch keine Commodore-Maus für den PC-128 erhältlich. Weiterhin finden wir, wie bei dem C-116, C-16 und Plus/4, auf der rechten Seite einen nicht für jeden Finger leicht erreichbar angebrachten Reset-Taster. Daneben der Ein-/Ausschalter und eine fünfpolige quadratische Buchse für die Stromzufuhr. Die Rückseite, auch hier wieder 100% C-64 mit Extras. Von links nach rechts: Expansion-Port, für die altbekannten Module beziehungsweise Steckkarten



mit Eproms. Daneben Cassetten-Port, Serial-Buchse, Video-Buchse, TV-Buchse, RGB/I-Anschluß und der User-Port. Die Heizung des Systems (Netzteil im Bügeleisenformat) wurde, „Gott sei es hundertmal gedankt“, auch beim PC-128 wieder extern gehalten. Nur leider doppelt so groß.

eins, zwei, drei, 64- 128- CP/M

Schaltet man den PC-128 ein, passiert schon eine ganze Menge in seinem Innenleben. Denn für ihn gilt es, auf die Schnelle abzuklären, welche Peripherie und Software auf ihn einwirken sollen. Ist das Diskettenlaufwerk ausgeschaltet, meldet er sich im 128er-Modus und je nachdem, welche Bildschirmdarstellung gewählt wurde, 40 oder 80 Zeichen pro Zeile. Bedingung ist natürlich, der entsprechende Monitor ist angeschlossen. Liegt eine CP/M-Diskette im eingeschalteten Laufwerk, wird sofort das Betriebssystem in den Computer geladen, sobald man ihn einschaltet. Auch

Unser Bild zeigt den Commodore 128 in Weihnachtsstimmung. Der Computer hat alle Voraussetzungen um das Geschenk Nr. 1 zu werden.

hier kommt es wieder darauf an, wieviel Zeichen pro Zeile voreingestellt wurden. Das CP/M-System läßt sich nämlich erstaunlicherweise auch im 40-Zeichen-Modus darstellen. Hält man hingegen beim Einschalten die Commodore-Taste fest, befindet man sich sofort im C-64-Modus. Übrigens, nicht daß Sie meinen, man könnte wie in einer Metzgerei HALB und HALB bestellen, also Mixbetrieb ist nix, alles klar ... ja !?! Der PC-128 besitzt ganz eindeutig „DREI“ Computer im Inneren, die nicht untereinander gemischt betrieben werden können. Vom 128er Modus gibt es allerdings die Möglichkeit, sich mit „GO 64“ in den C-64 Betrieb zu schalten. Hat man den Rechner eingeschaltet, ohne das Diskettenlaufwerk einzuschalten, genügt ein einfaches „BOOT“, um in CP/M zu kommen. Läuft der Rechner hingegen einmal im 64er oder CP/M-

Modus, gibt es kein Zurück mehr. Der Computer muß ausgeschaltet werden. Zugegeben, am Anfang mag das etwas verwirrend sein, aber der Mensch ist schließlich ein Gewohnheitstier!?! Viele tausend C-64-Besitzer mögen sich wohl immer noch die Frage stellen: Wie kompatibel ist denn nun der PC-128 wirklich, läuft denn nun tatsächlich meine Software auf dem Ding? Egal ob man die C-64-Programme aus dem schnellen „Döschen“ oder von Cassette lädt, egal ob BASIC oder Maschinensprache, es läuft! Es läuft wirklich ohne jegliche Probleme. Bei der Verwendung des Laufwerks „1541“ gibt es ebenfalls keine Probleme, sogar CP/M läßt sich mit der „1541“ laden. Also kein Grund zur Besorgnis. Ein letzter Beweis sind die Tasten! Nämlich die Tasten, die beim 64er nicht vorhanden sind. Wenn man auf solch eine Taste drückt, passiert überhaupt



Auch im Profieinsatz ist der C128 denkbar

nichts. Im 64er Modus weiß der Computer gar nicht, was er damit anfangen soll.

Comfort ist tot, es lebe Comfort!

Die Funktionstasten im 64er Modus arbeiten ganz normal, das heißt, sie sind mit nichts belegt. Im 128er Modus sieht dies natürlich anders aus. Hier gibt's Comfort. F-Tasten-Funktionen gibt es acht an der Zahl: GRAPHICS, DLOAD, DIRECTORY, RUN, DSAVE, LIST, SCNCLR und MONITOR. Diese können nach Belieben vom Anwender neu programmiert werden. Übrigens, wenn Sie versuchen möchten, ein Programm, das auf dem 64er mit f-Tasten gesteuert wird, auf dem 128er umzuschreiben, dürfte es leichte Problemchen geben, die f-Tasten im 128er-Modus „LEER“ zu bekommen. Aber nur keine Bange, wir zeigen Ihnen auch, wie das funktioniert. Comfort bietet ebenfalls das BASIC 7.0 mit über 140 Anweisungen bzw. Befehlen! Diese Version stellt natürlich alles Bisherige in den Schatten, egal ob C-16, VC-20, C-64, 600er, 3/4000er oder 8000er Rechner von Commodore. Es gibt sogar noch einige Befehle mehr. Alleine in der Grafik gibt es 13 neue Befehle: BOX, CIRCLE, COLOR, GRAPHIC, MOVSPR, SCALE, SPRSAV, CHAR, LOACATE, PAINT, GSHAPE, SSHAPE und DRAW. Wenn man einmal etwas genauer hinsieht, kann man

eine gewisse Ähnlichkeit mit MSX-BASIC feststellen, dort gibt es ebenfalls solche Macrobefehle. So zeichnet BOX z.B. ein Rechteck, wenn man die dazugehörigen Koordinaten eingibt. CIRCLE z.B. einen Kreis, und mit PAINT füllt man eine umrandete Fläche mit Farbe aus. Genauso toll sieht es bei Musik aus. Man braucht nicht mehr zu „peeken“ und „poken“, sondern man bedient sich der neuen Befehle: SOUND, ENVELOPE, VOL, TEMPO, PLAY und FILTER. Der Soundchip (8581) kann damit viel, viel leichter unter Strom gesetzt werden. Es sind sogar schon zehn voreingestellte Instrumente im 128er enthalten.

CP/M-Modus Profiwelt

Was man beim C-64 nur mit akrobatischen Kunststücken und zusätzlichen Steckmodulen, nebst Diskette und was es sonst noch so gibt, schafft, erledigt der PC-128 bereits in der Grundversion. CP/M! Wird dieses Betriebssystem geladen, so dürfte man Zugriff auf ca. 2000 Standard- und Spezialprogramme im Profibereich haben. Ich denke da an Textverarbeitung, Kalkulation, Datenverwaltung und Datenfernübertragung. Dies ermöglicht der „Z 80 B“ (4 MHz Taktfrequenz). Die Ladezeit des Betriebssystems dauert ca. 30 Sekunden von der Diskette. Danach meldet sich der Rechner mit dem typischen A>

Prompt, daß er bereit ist, Befehle wie z.B.: DIR (Disketteninhaltsverzeichnis lesen und anzeigen), FORMAT (Diskette formatieren), PIP (Kopierbefehl) oder TYPE (schnelle Durchsicht der Diskettendateien) usw. zu empfangen, um diese dann abzuarbeiten. Das neue Diskettenlaufwerk 1571, das Anfang bis Mitte November erscheinen soll, lag uns bei Redaktionsschluß auch noch nicht vor. Übrigens soll zwischen Commodore, Keypro und Osborn das gleiche Diskettenformat vorliegen. Wenn dies stimmt, dann sage ich nur: Gute Nacht, Marie“. Denn somit ist ja schon ein großer Teil der Frage geklärt, woher die Software? Ein Freund von mir berichtete: Ich habe CP/M geladen auf dem PC-128, gewartet, bis das A> Prompt auf dem Bildschirm zu sehen war, nahm die Kaypro-Word-Star-Diskette und gab „WS“ ein, um diese zu laden. Das Diskettenlaufwerk arbeitet, und siehe da: Das Wordstar-Eröffnungsmenü erscheint. Man kann jetzt mit dem Commodore-Computer genauso arbeiten wie mit dem Kaypro. Jeder Befehl wird ausgeführt — eine Überraschung. Nach „Adam Riese“ werden dann natürlich auch andere CP/M-Programme wie Multiplan, dBaseII usw. geschluckt; ... der Fachmann staunt, der Laie wundert sich ...!

Alles schreit nach IBM! Kein Problem!

Nicht nur eine Formatierung der Disketten im IBM-Format ist möglich, sondern auch einen IBM-Monitor kann man an den RGB/I-Port des PC-128 anschließen. Apropos Diskette, wer an seinem Arbeitsplatz eine Datei auf einem IBM erstellt, kann diese dann zu Hause auf seinem 128er in der Textverarbeitung weiterverwenden. Die Datei wird also mühelos gelesen. Sinnvoll ist allerdings, wie bereits erwähnt, ein 80-Zeichen-Monitor. Der Color-Monitor von Commodore (1902) ist jedoch noch nicht lieferbar. Was aber kein allzu großes Problem ist, denn es gibt ja auch noch Fremdhersteller, die in der Lage sind, etwas schneller als Commodore einen 80-Zeichen-Monitor, sogar in Farbe, auf den Markt zu bringen.

Zusammenfassend:

Sucht man einen schnellen Bürocomputer, ist man mit anderen besser bedient. Ist man jedoch C-64-Aufsteiger oder sucht eine Kombination zwischen Home-Computer (Spiele!) mit professionellen Qualitäten und Bürocomputer, so ist der PC-128 genau das richtige. Da dürfte es wohl zur Zeit nichts Besseres und Preiswerteres geben.

Minuspunkte:

- Leider nicht als Komplettsystem erhältlich (siehe ATARI 520 ST).
- Netzteil zu wuchtig.
- Strombuchse rechts (besser an der Rückseite).
- Reset-Taste zu klein.
- Ein/Aus-Schalter zu klein.

Pluspunkte:

- + 100% kompatibel zum C-64 (software- und hardwaremäßig).

- + Tastatur sehr gut.
- + Separater Zehnerblock.
- + Vier doppelt belegte Funktionstasten.
- + CP/M (3.0) und CP/M.
- + Design sehr gut (kein Spiel-Computeraussehen). 40/80 Zeichen-Modus.
- + RAM-Disk (bis 512 KByte)
- + Kaypro- und Osbornkompatibel im CP/M-Modus.
- + Preis 998,- DM.

R.P.

In einem Satz: TEAM ANIMATOR ist ein Programm, bei dem Sie alle 8 Sprites in Richtung, Geschwindigkeit und Laufdauer ganz bequem steuern können. TEAM ANIMATOR ist das Werkzeug, mit dem es kein Problem mehr darstellt, umfangreiche Spiele mit verschiedenen Sprites zu programmieren.

Von Richard Leinfellner

Sie sehen hier 2 verschiedene Listings. Das eine Listing mit den vielen Zahlen in den Datazeilen ist ein BASIC-Programm, das über die Datazeilen ein Maschinenprogramm, das eigentliche Spriteutility, erzeugt. Die Handhabung dieses Programms ist sehr einfach, Sie brauchen es nur abzutippen und laufen zu lassen. Das BASIC-Programm erzeugt das Maschinenprogramm und speichert es zu allererst auf Cassette oder Diskette ab. Die Abspeicherung ist menügesteuert. Zusätzlich sind gegen

evtl. Abtippfehler in den langen Zahlenreihen Prüfsummen in das Programm eingebaut. Das bedeutet, wenn Sie sich in den Datazeilen bei einer Zahl vertippen sollten, merkt das Programm dieses und gibt Ihnen eine Fehlermeldung: Fehler in den Maschinensprache-Databefehlen, raus.

Nach dem Abspeichern steht Ihnen dieses Tool dann zur Verfügung. Das Programm ist dann bei Hexadezimal \$ C000 (49152) abgespeichert. In diesem Bereich ist eine 4 K breite Spalte zwischen dem BASIC-ROM und dem INPUT, OUTPUT des C 64 frei. Das Programm ist 364 Bytes lang und benutzt den Bereich \$ C000 — \$ C140 und \$ CFE0 — CFFF. Das Maschinenprogramm steht dem Benutzer jederzeit zur Verfügung und läuft unabhängig von allen anderen Programmen.

Eine Anwendung dieser Spriteroutine finden Sie in Listing 2. Dieses Listing ist ein BASIC-Pro-

ogramm, das ermöglicht, die 8 gesetzten Sprites auf dem Bildschirm, sie sind alle als Quadrate gesetzt, durch Abfrage weniger Eingaben zu bewegen. So fragt das Programm nach Sprite-Nummer, nach Richtung, nach horizontaler Geschwindigkeit, vertikaler Geschwindigkeit und der Laufdauer. Dazu ein Beispiel:

Sprite-Nr. = 0, Richtung = NR, horizontale Geschwindigkeit = 1, vertikale Geschwindigkeit = 1, Laufdauer = 100.

Das bedeutet, daß das Sprite, Farbe weiß, sich nach unten rechts für 100 Pixels bewegt. Die Richtungsangaben erfolgen per Buchstabe H für hoch, N für nieder, L für links und R für rechts. Das BASIC-Programm ist eine Demonstration, wie man die Spriteroutine am einfachsten nutzen kann. Dazu muß man noch bemerken, die Sprite-Nummern liegen zwischen 0 — 7, somit insgesamt 8 Sprites. Die Eingabe für die Geschwindigkeit in

```

5 REM PROGRAM BEI RICHARD LEINFELLNER,PALACE SOFTWARE
10 IF A=1 THEN 30
15 POKE 52,63:POKE56,63:CLR
20 GOSUB 700
25 REM POSITIONEN FUR DIE SPRITES WERDEN GESETZT
30 GOSUB 800
35 REM ALLE ALTEN BEWEGUNGS TABELLEN WERDEN AUF 0 GESETZT
40 GOSUB 600
45 REM SCHWARZER HINTERGRUND
50 POKE 53280,0:POKE53281,0:POKE53275,255
55 REM ALL SPRITES EINGESCHALTEN
70 POKE 53269,255
75 REM STARTET DAS MASCHINENSPRACHE PROGRAM
80 SYS 12*4096
90 PRINT"(CLR)(WHT)(HOM)";
95 REM NEUE WERTE FUR SPRITE BEWEGUNG
100 INPUT"SPRITE NUMBER";SN
120 INPUT"RICHTUNG(H,N,L,R)";A#
130 INPUT"HORIZONTALE GESCHWINDIGKEIT";X
140 INPUT"VERTICALE GESCHWINDIGKEIT";Y
150 INPUT"LAUFDAUER";DUR
155 REM ARBEITET DIE WERTE AUS
160 GOSUB 400
170 GOTO 90
400 IF LEN(A#)=0 THEN RETURN
410 DIR=0
420 FOR I = 1 TO LEN(A#)
430 B#=MID$(A#,I,1)
440 IF B#="H" THEN DIR=DIR OR 1
450 IF B#="N" THEN DIR=DIR OR 2
460 IF B#="L" THEN DIR=DIR OR 4
470 IF B#="R" THEN DIR=DIR OR 8
480 NEXT
485 REM HORIZONTALE GESCHWINDIGKEIT
490 POKE 53232+SN,X
495 REM VERTICALE GESCHWINDIGKEIT
500 POKE 53240+SN,Y
505 REM RICHTUNG
510 POKE 53224+SN,DIR
515 REM LAUFDAUER
520 POKE 53216+SN,DUR
530 RETURN
600 FOR I=0 TO 7
605 REM ALLE SPRITES WERDEN AUF $3FC0 GESETZT
610 POKE I+2040,255
615 REM NEUE FARBEN 1-8
620 POKE I+53287,I+1
625 REM ALLE ALTEN TABELLEN AUF 0
630 POKE53216+I,0:POKE53224+I,0
640 POKE53232+I,0:POKE53240+I,0
650 NEXT
660 FORI=0 TO 63
665 REM SPRITES WERDEN ALS QUADRATE GESETZT
670 POKE I+16320,255
680 NEXT
690 RETURN
700 PRINT"LEGEN SIE DIE KASSETTE ODER DISK"
710 PRINT"MIT DEM MASCHINENSPRACHE PROGRAMM"
730 INPUT"K=KASSETTE,D=DISK";A#
740 IF A#="D" THEN D=8:GOTO 760
750 D=1
760 A=1:LOAD"SPRITES",D,1
770 RETURN
800 POKE53269,0
810 FOR I=0 TO7
820 READ X,Y
830 POKE 53248+I*2,(X AND 255):POKE 53249+I*2,Y

```

Anzeige

<i>DFÜ Paket: Akustikkoppler (FTZ-Nr) Netz., Software und Kabel</i>		<i>450,- DM</i>
<i>Turbo-Pascal auf 3"-Diskette</i>		<i>220,- DM</i>
<i>CPC 464; 664 zu Superpreisen Vortex Diskstat. Speichererweiter. 64k und 256KByte Lightpen Prof-Software Spiele Zubehör und vieles mehr</i>	<i>Liste mit über 400 Artikeln 2,50 DM H.-J. Janke Postfach 150 173 56 Wuppertal 12 0202/47 55 21 Händleranfragen erwünscht!</i>	

```

840 IF INT(X/256)=1 THEN POKE 53264,PEEK(53264)OR 2^I
850 NEXT
860 RETURN
895 REM POSITIONEN FUR SPRITES AM ANFANG
900 DATA 172,140,320,50,320,229,24,229
910 DATA 172,50,172,229,24,140,320,140
995 REM DATA FUR MASCHINENSFRACHE

```

READY.

horizontale und vertikale Richtung sollte zwischen 0 und höchstens 20 liegen, da sonst die Bewegung zu schnell wird. Bei der Laufdauer besteht die Möglichkeit, eine Zahl zwischen 1 — 255 einzugeben. Außerdem muß man darauf achten, daß Sprites sich nicht weiter als 24 — 320 in der Horizontalen und 50 — 229 in der Vertikalen bewegen können. Wichtig ist es, daß man die Werte für die Laufdauer zuletzt eingibt, weil diese Werte die Bewegung starten. Wenn die Laufdauer auf 0 zugeht, bleibt das Sprite still stehen.

Die Zeilen 400 — 530 übertragen die Werte für die Geschwindigkeit, Richtung und Laufdauer zu dem Maschinensprache - Programm. In den Speicherstellen 53232 bis 53239 sind die Werte für die X-Geschwindigkeit. In den Speicherstellen 53240 bis 53247 sind die Werte für die Y-Geschwindigkeit.

In 53224 bis 53231 sind die Werte für die Richtung. In 53216 bis 53223 sind die Werte für die Laufdauer. Jedes Sprite hat innerhalb dieser 4 Parameter eine eigene Tabelle. So benutzt das Sprite 0 die Speicherstellen 53232, 52340, 53224, 53216. Hingegen benutzt das Sprite Nr.1 dieselben Speicherstellen + 1, also 53233, 53241, 53225 und 53217. Die Adressierbarkeit der Sprites wird dadurch recht einfach. Man braucht nur noch den Befehl POKE 53232 + N,G eintippen. N ist die

einfach ein RETURN stehen muß:
660 RETURN.

Mit diesen kleinen Änderungen benutzt das Programm TEAM ANIMATOR die Sprites Nr. 200 — 207, die durch den TEAM SPRITER aus COMPUTER TEAM Ausgabe 11/85 erzeugt werden.

An dieser Stelle möchten wir Ihnen noch einen kleinen Trick verraten. Wenn man das Programm TEAM SPRITER aus Ausgabe 11/85

, Richtung und laufdauer zu dem Maschinensprache Program.

53232 bis 53239 sind die Werte für X Geschwindigkeit,
53240 bis 53247 sind die Werte für Y Geschwindigkeit,
53224 bis 53231 sind die Werte für Richtung und
53216 bis 53223 sind die Werte für Laufdauer.

Jede Sprite hat seine eigene Tabelle so das Sprite 0
53232,52340,53224,53216 benützt und Sprite 1 benützt
53233,53241,53225,53217 ect.

Nummer des jeweiligen Sprites und G ist die neue Geschwindigkeit, resultiert a. Basiswert 53232 für X-Geschwindigkeit + Sprite-Nummer mit neuer Geschwindigkeit. Dazu noch ein Beispiel: Der Befehl POKE 53224 + N,R würde dem Sprite N eine neue Richtung geben. Die Richtung ist in demselben Format wie bei der Joystickabfrage, z.B. herauf = 1, herab = 2, links = 4, rechts = 8, rechts herab = 10. Dieses Programm soll nur ein Beispiel dafür sein, wie man mit TEAM ANIMATOR ohne Probleme 64er Spiele erstellen kann.

Der TEAM ANIMATOR läuft in Verbindung mit dem TEAM SPRITER aus der Ausgabe 11/85. Um beide Programme miteinander benutzen zu können, müssen allerdings zwei Änderungen in diesem Programm vorgenommen werden. Die Zeile 610, die bisher lautete:

```
610 POKE i + 2040, 255 muß folgendermaßen geändert werden:
610 POKE i + 2040, i + 200. Außer der Zeile 610 muß noch die Zeile 660 geändert werden, bei der anstelle:
```

```
660 FOR i = 0 TO 63
```

gestartet hat, beginnt das Editor-Menü mit dem Sprite Nr. 128. Durch Betätigen der Plustaste kann man die Sprites hochzählen bis 200, die für den TEAM ANIMATOR interessant sind. Um dieses Verfahren abzukürzen, kann man allerdings auch den TEAM SPRITER mit BREAK unterbrechen und danach die Variable S auf 100 setzen, indem man einfach:

```
S = 100
```

eintippt. Allerdings muß man jetzt darauf achten, das Programm erneut zu starten, indem man nicht den Befehl RUN nimmt, sondern CONT. Danach erscheint das Editor-Menü mit der Sprite-Nr. 200. Der TEAM ANIMATOR benutzt die Sprites Nr. 200 — 207 aus dem TEAM SPRITER.

Wir wünschen allen Lesern von COMPUTER TEAM nun viel Spaß bei der Anwendung von TEAM SPRITER und TEAM ANIMATOR, auf daß Sie selbst in der Lage sind, Hexen über den Bildschirm des C 64 wandern zu lassen, genauso wie es Richard Leinfellner bei dem erfolgreichen Spiel Hexenküche gemacht hat.

```

10 GOSUB 700
20 PRINT "{CLR}MASCHINENSFRACHE PROGRAM IST JETZT AUF #C000 VERSETZT"
30 PRINT
40 PRINT "JETZT KÖNNEN SIE DAS PROGRAM AUF"
50 PRINT "KASSETTE ODER DISK AUFNEHMEN"
60 INPUT "K=KASSETTE,D=DISK";A$
70 IF A$="D" THEN POKE49473,B:POKE49475,8
80 SYS 49472
90 PRINT "PROGRAM IST JETZT AUFGENOMMEN"
500 END
700 K=49152:PRINT "{CLR}{HOM}";:PRINT "BITTE WARTEN SIE FÜR EINEN MOMENT"
710 READ A
720 IF A=-1 THEN 760
725 REM MASCHINENSFRACHE POKE AUF #C000
730 POKE K,A

```

```

740 K=K+1:C=C+A
750 GOTO 710
760 IF C<>46483 THEN 780
770 RETURN
780 PRINT"FEHLER IN DEM MASCHINENSFRACHE DATA      BEFEHL"
790 PRINT"IN LINIEN 1000-1170":STOP
895 REM POSITIONEN FUR SPRITES AM ANFANG
1000 DATA 120,169,31,141,20,3,169,192,141,21,3,169,1,141,26,208,169,250
1010 DATA 141,18,208,173,17,208,41,127,141,17,208,88,96,173,25,208,48,3
1020 DATA 76,49,234,173,17,208,41,128,208,31,173,18,208,201,250,144,24
1030 DATA 32,101,192,169,50,141,18,208,173,17,208,41,127,141,17,208,169,1
1040 DATA 141,25,208,76,95,192,169,250,141,18,208,173,17,208,41,127,141
1050 DATA 17,208,169,1,141,25,208,104,168,104,170,104,64,162,7,138,10,168
1060 DATA 189,224,207,240,41,222,224,207,189,232,207,141,62,193,78,62
1070 DATA 193,144,3,32,156,192,78,62,193,144,3,32,183,192,78,62,193,144,3
1080 DATA 32,210,192,78,62,193,144,3,32,0,193,202,16,204,96,185,1,208
1090 DATA 201,50,144,11,185,1,208,56,253,248,207,153,1,208,96,189,232,207
1100 DATA 9,128,157,232,207,96,185,1,208,201,230,176,11,185,1,208,24,125
1110 DATA 248,207,153,1,208,96,189,232,207,9,128,157,232,207,96,173,16
1120 DATA 208,61,46,193,208,7,185,0,208,201,24,144,22,185,0,208,56,253,240
1130 DATA 207,153,0,208,176,9,173,16,208,61,54,193,141,16,208,96,189
1140 DATA 232,207,9,128,157,232,207,96,173,16,208,61,46,193,240,7,185,0
1150 DATA 208,201,65,176,22,185,0,208,24,125,240,207,153,0,208,144,9,173
1160 DATA 16,208,29,46,193,141,16,208,96,189,232,207,9,128,157,232,207,96
1170 DATA 1,2,4,8,16,32,64,128,254,253,251,247,239,223,191,127,0,36,169,1,162,1
1180 DATA 160,255,32,186,255,169,7,162,100,160,193,32,189,255,169,0,133,251,169
1190 DATA 192,133,252,162,63,160,193,169,251,32,216,255,96,83,80,82,73,84,69,83
1200 DATA 0,-1

```

READY.

Repeat-Funktion für Commodore

PC-128
POKE 2594,0 Repeat wie bei

C-64.
POKE 2594,64 keinerlei Repeat
POKE 2594,128 alles wieder nor-

mal, Repeat auf allen Tasten.
POKE 2595,x Tastenverzögerung
einstellen, x = die Dauer der Ver-

zögerung.
C-64
POKE 650,128 Repeat für alle Ta-

sten.
POKE 650,64 keinerlei Repeat.
POKE 650,0 Repeatfunktion nor-

mal.
POKE 650,127 Cursorrepeat aus-

schalten.
Beep- bzw. Piepton

PC-128
PRINT CHR\$(T);... läßt einen Piep
erklängen. (Programmier-Mode)

CTRL + G im Direkt-Mode läßt
ebenfalls einen Piep erklingen.
POKE 249,128 stellt den Piepmatz

aus.
POKE 249,0 erweckt ihn wieder
zum Leben.

C-64

folgende POKES mit den entspre-
chenden Zeilennummern verse-
hen und durch GOSUB aufgerufen,
lassen beim C-64 auch einen Piep-

matz erklingen:
54296,13:POKE 54277,25:POKE
54278,68:POKE 54273,99:POKE
54272,100:POKE 54276,33:FOR T=1

TO 120:NEXT T: POKE 54276,0:
REM evtl. RETURN

Groß/Kleinschrift-Umschaltung:
PC-128

POKE 245,64 verriegelt die Um-
schaltung mit den C= + Shift-

Tasten.
POKE 245,0 und alles ist wieder
beim alten bzw. Einschaltmodus.

C-64
POKE 657,128 verriegelt die Um-
schaltung
POKE 657,0 schaltet wieder alles

auf normal.
PRINT CHR\$(8) verriegeln.
PRINT CHR\$(9) normal.

3D-Plot

Dieses Programm plottet dreidimensionale Funktionen (d.h. Funktionen von zwei Veränderlichen) auf dem Bildschirm und fertigt auf Wunsch auch eine Hardcopy an. Die Funktion liegt dabei in der Form $z=f(x,y)$ als Unterprogramm ab Zeile 1000 vor. Zu Beginn werden der zu plottende Bereich und die Anzahl der zu zeichnenden parallelen Linien abgefragt.

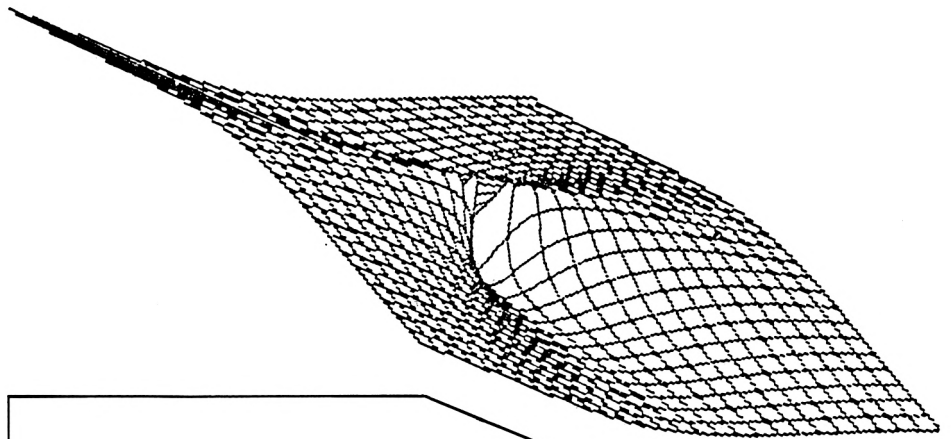
Bei der Darstellung der Funktion läuft die x-Achse nach hinten rechts, die y-Achse nach hinten links und die z-Achse nach oben. Die Funktion wird dann durch jeweils zur x- und y-Achse parallele Linienscharen dargestellt. Im Programm wird dabei für beide Richtungen die gleiche Schleife benutzt, die von einer äußeren Schleife gesteuert wird.

Verdeckte Linien werden dabei unterdrückt; dies läuft so, daß alle Punkte, die zwischen den bisher gezeichneten Linien liegen (d.h. alle Punkte zwischen den bisherigen

Verdeckte Linien unterdrücken

Minima und unterhalb des Maximums), nicht gezeichnet werden. Die Minima und Maxima werden in den Feldern `maxp` und `minp` gespeichert, die vor jedem der beiden Durchläufe mit einem speziellen Wert gekennzeichnet werden, weil zu Beginn noch keine Minima und Maxima existieren.

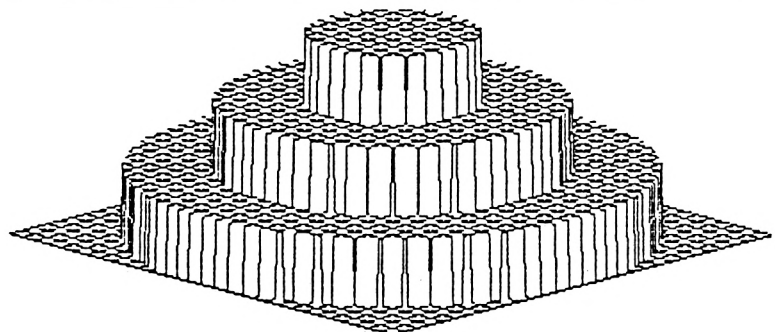
Der Bereich, in dem die Funktion geplottet wird, wird durch die Grenzen `xmin` und `xmax`, `ymin` und `ymax`, `zmin` und `zmax` angegeben. Alle außerhalb dieses Bereiches fallenden Punkte werden trotzdem gezeichnet, falls sie noch auf dem Bildschirm sichtbar sind.



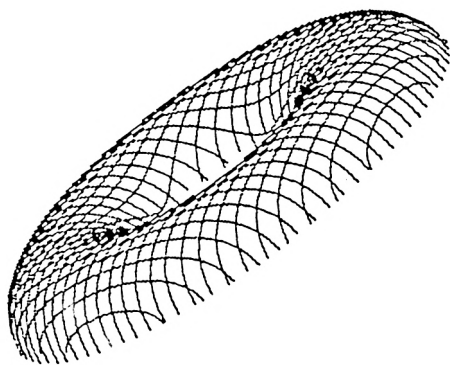
```

1 REM Frank Thielen
4 REM      &
5 REM Thomas Barndt
8 REM
10 MODE 2
20 BORDER 0
30 INK 1,26:PEN 1
40 INK 0,0:PAPER 0
50 MEMORY &9FFF
60 FOR i=&A000 TO &A0BF
70 READ byte:POKE i,byte:s=s+byte:NEXT
80 IF s<>23767 THEN PRINT"Fehler in datas !":EN
  D
90 CLEAR
100 DEFINT d,i,j,r,f,g,n,m
110 CLS
120 PRINT"
                                3D-Plot":PRINT
    :PRINT "
    2":PRINT
130 PRINT
140 PRINT"
                                Realisiert durch FTCP & THB
    CS 1985":PRINT:PRINT
150 PRINT"
                                Eingabe der Funktion als z=f(x,y)
    ab Zeile 1000"
160 PRINT
170 PRINT"Eingabe von  xmin, xmax,"
180 PRINT"
                                ymin, ymax,"
190 PRINT"
                                zmin, zmax,"
200 INPUT "
                                Anzahl Linien";xmin,xmax
    ,ymin,ymax,zmin,zmax,n
210 IF xmin>=xmax OR ymin>=ymax OR zmin>=zmax OR
    n<2 THEN 110
220 m=320
    Anzahl der Punkte in beiden Haupta
    chsrichtungen
230 d1=320
240 d2=100
250 d3=200

```



Die meisten der Beispielfunktionen sind punktsymmetrisch zum Ursprung, dies muß jedoch nicht sein, wie die „Wellen“ zeigen. Für symmetrische Funktionen sollte auch der x- und y-Bereich symmetrisch gewählt werden, so z.B. in x- und y-Richtung von -1.7 bis 1.7 wie beim „Sombbrero“ und anderen Funktionen. Um eine unbekannte Funktion darzustellen, sollte man zu Beginn einen ziemlich großen Bereich wählen und dann nach Augenmaß einen kleineren Ausschnitt eingeben.



```

260 c1=(xmax-xmin)/(n-1)
270 c2=d1/(n-1)
280 c3=d1/m
290 c4=d3/(zmax-zmin)
300 c5=(xmax-xmin)/m
310 c6=(ymax-ymin)/(n-1)
320 c7=(ymax-ymin)/m
330 c8=d2/(n-1)
340 c9=d2/m
350 DIM maxp(2*d1),minp(2*d1)
360 MODE 2
370 ORIGIN 320,0
380 FOR r=-1 TO 0
390   FOR i=0 TO 2*d1
400     maxp(i)=-10000:minp(i)=-10000
410   NEXT i
420   FOR i=0 TO n-1
430     IF r THEN x=xmin+c1*i ELSE y=ymin+c6*i
440     f=-1
450     FOR j=0 TO m
460       IF r THEN y=ymin+c7*j ELSE x=xmin+c5*j
470       GOSUB 1000
480       xp%=ROUND(i*c2-j*c3):IF NOT r THEN xp%
         =-xp%
490       yp%=ROUND(i*c8+j*c9+(z-zmin)*c4)
500       index=xp%+d1
510       IF maxp(index)=-10000 THEN maxp(index)
         =yp%:minp(index)=yp%:g=-1:GOTO 530

```

```

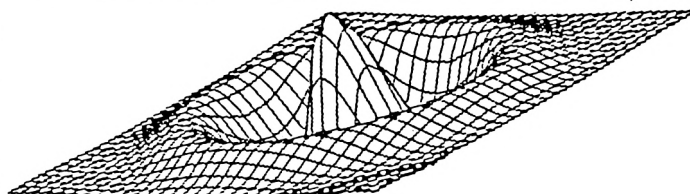
520   IF yp%>maxp(index) THEN maxp(index)=yp%
     %:g=-1 ELSE IF yp%<minp(index) THEN minp
     (index)=yp%:g=-1 ELSE g=0
530   IF j=0 THEN MOVE xp%,yp%:GOTO 560
540   IF f AND g THEN DRAW xp%,yp%
550   IF g THEN MOVE xp%,yp%
560     f=g
570   NEXT j
580 NEXT i
590 NEXT r
600 DIM rs(639)
610 FOR i=0 TO 639
620   rs(i)=PEEK(49152+i MOD 80+(i\80)*2048)
630 NEXT
640 LOCATE 1,1:PRINT"H)ar'dcopy B)eenden N)eustar
  t"
650 s$=LOWER$(INKEY$):IF s$="b" THEN CLS:END ELS
  E IF s$="n" THEN ERASE minp,maxp,rs:GOTO 110
  ELSE IF s$<>"h" THEN 650
660 FOR i=0 TO 639
670   POKE 49152+i MOD 80+(i\80)*2048,rs(i)
680 NEXT
690 PRINT#8,CHR$(27);"@";:CALL &A000:PRINT#8:PRI
  NT#8,CHR$(27);"e"
700 GOTO 640
990   *** Huegel ***
1000 'z=EXP(-x*x-y*y)
1010   *** Sombbrero ***
1020 z=EXP(-x*x-y*y)*COS(SQR(x*x+y*y)*5)
1030   *** Spitze ***
1040 'z=sqr(x*x+y*y)
1050 'if z>2 then z=0 else z=2-z
1060   *** Torte ***
1070 'z=sqr(x*x+y*y)
1080 'if z>2 then z=0 else z=2
1090   *** Hochzeitstorte ***
1100 'z=sqr(x*x+y*y)
1110 'if z>3 then z=0 else z=3-int(z)
1120   *** Kuppel ***
1130 'z=x*x+y*y
1140 'if sqr(z)>3 then z=0 else z=sqr(9-z)
1150   *** Wellen ***
1160 'z=COS(x):z=z*z*y*y

```

```

1170 '   *** Schwarzes Loch ***
1180 'z=SQR(x*x+y*y)
1190 'IF z=0 THEN z=0.01
1200 'z=-ABS(1/z)
1210   *** Kreuzgewoelbe ***
1220 'z=EXP(-(x)*x)+EXP(-(y)*y)-EXP(-(x)*x-y*y)
3000 RETURN
3010 '
3020 '
3030 DATA &cd,&ba,&bb,&cd,&e7,&bb,&32,&bd
3040 DATA &a0,&acd,&6c,&a0,&21,&8f,&01,&22
3050 DATA &be,&a0,&11,&00,&00,&3e,&07,&32
3060 DATA &c0,&a0,&cd,&7c,&a0,&0e,&00,&3a
3070 DATA &c0,&a0,&47,&e5,&d5,&c5,&cd,&f0
3080 DATA &bb,&c1,&d1,&21,&bd,&a0,&be,&e1
3090 DATA &37,&20,&01,&a7,&cb,&11,&2b,&2b
3100 DATA &10,&e9,&cd,&af,&a0,&79,&cd,&a6
3110 DATA &a0,&13,&e5,&21,&7f,&02,&37,&ed
3120 DATA &52,&e1,&38,&05,&2a,&be,&a0,&18
3130 DATA &cc,&23,&7c,&b5,&c8,&2b,&11,&00
3140 DATA &00,&22,&be,&a0,&3e,&07,&bd,&20
3150 DATA &b9,&7c,&b4,&20,&b5,&3e,&04,&32
3160 DATA &c0,&a0,&18,&ae,&3e,&1b,&cd,&a6
3170 DATA &a0,&3e,&41,&cd,&a6,&a0,&3e,&07
3180 DATA &cd,&a6,&a0,&c9,&e5,&3e,&42,&cd
3190 DATA &1e,&bb,&e1,&28,&02,&e1,&c9,&3e
3200 DATA &0d,&cd,&a6,&a0,&3e,&0a,&cd,&a6
3210 DATA &a0,&3e,&1b,&cd,&a6,&a0,&3e,&4c
3220 DATA &cd,&a6,&a0,&3e,&7f,&cd,&a6,&a0
3230 DATA &3e,&02,&cd,&a6,&a0,&c9,&cd,&2e
3240 DATA &bd,&38,&fb,&cd,&2b,&bd,&c9,&3a
3250 DATA &c0,&a0,&fe,&07,&c8,&af,&c5,&11
3260 DATA &cb,&11,&cb,&11,&c9,&00,&00,&00

```



Alles aus einer Hand

- Software auf die Sie schon lange warten
- Hardware die es nicht überall gibt
- Branchenlösungen maßgeschneidert

Abdeckhauben und Druckertische



..... für Schneider CPC 464



..... für Commodore C 64



..... für Commodore Floppy



..... für Drucker aller Typen

..... sowie für viele andere Computer und Floppies

und dazu die Software von CARAT SOFT



Für CPC 464 / 664 / 6128

und für ATARI, für APRICOT, für IBM, für KOMPATIBLE und, und, und

Dateiverwaltung

Multidatei zur Erstellung individueller, beliebig vieler Dateien, z.B. Video - Briefmarken, - Schallplattendaten, Ausdruck von Listen oder Hardcopy sowie Such- und Sortierkriterien

System: Schneider CPC 664 / 664 / 6128

Adressenverwaltung

Kommerzielle Adressverwaltung zur Speicherung von ca. 300 Adressen, sowie Firmen oder Person mit Familien-Nr., Telefon, Telex, Land und Bezeichnungen, Ausdrucken als Listen- und Hardcopydruck.

System: Schneider CPC 664 / 664 / 6128

Adressenverwaltung

Kommerzielle Adressverwaltung zur Speicherung von ca. 300 Adressen, sowie Firmen oder Person mit Familien-Nr., Telefon, Telex, Land und Bezeichnungen, Ausdrucken als Listen- und Hardcopydruck.

Kassenbuch

Kassenbuch mit Geschäftsschub, Einnahmen, Ausgaben von automatischen Buchungsbetragnummern, Tageskassenbericht mit Einnahmen und Ausgaben, Ausdruck des monatlichen Kassenbuches sowie einzelner Kassenblätter.

System: Schneider CPC 664 / 664 / 6128

Kassenbuch

Kassenbuch mit Geschäftsschub, Einnahmen, Ausgaben von automatischen Buchungsbetragnummern, Tageskassenbericht mit Einnahmen und Ausgaben, Ausdruck des monatlichen Kassenbuches sowie einzelner Kassenblätter.

Vereinsverwaltung

Bearbeitung von ca. 250 Vereinsmitgliedern und deren persönliche Daten: Eintrittsdatum, Geburtsdatum, Geburtsdatum, Funktion im Verein und Mitglieds-Nr., Beitragszahlungen, Briefkopiedruck, Mahnungstext, Mitgliederstatistik, Anzeige oder Ausdruck neuer Mitglieder, Altersstruktur des Vereins sowie Listen- und Hardcopydruck.

System: Schneider CPC 664 / 664 / 6128

Vereinsverwaltung

Bearbeitung von ca. 250 Vereinsmitgliedern und deren persönliche Daten: Eintrittsdatum, Geburtsdatum, Funktion im Verein und Mitglieds-Nr., Beitragszahlungen, Briefkopiedruck, Mahnungstext, Mitgliederstatistik, Anzeige oder Ausdruck neuer Mitglieder, Altersstruktur des Vereins sowie Listen- und Hardcopydruck.

..... und vieles andere mehr

Fragen Sie nach den Produkten von **MEDIA PLAST GmbH**

oder direkt bei MEDIA PLAST GmbH, Lübecker Straße 32
4600 Dortmund, Telefon (02 31) 52 78 45 / 46

ATARI, APRICOT, IBM, SCHNEIDER, COMMODORE sind eingetragene Warenzeichen

DER ERFOLG GAB UNS RECHT

EASY-TEXT plus

für Ihren Schneider
CPC 464, 664, 6128
Funktionen (Auszug)

1. Volleditierung mit DEL, CLR und Pfeiltasten in allen Richtungen.
2. Text laden und speichern auf Diskette oder Kassette.
3. Blocksatz einrichten oder auflösen.
4. Links- und rechtsbündig ausrichten oder zentrieren.
5. Text zentrieren.
6. Suchen und Ersetzen.
7. Wortwrap ausschaltbar.
8. Trennvorschläge + Trennung.
9. Adressverwaltung auch separat nutzbar.
10. Drucker-Steuerung im Text Groß-, Hoch-, Tief-, Fettdruck, und Unterstreichen.
11. Löschen und Versetzen von Zeilen, oder Textteilen.
12. Text vor- und rückwärts verketten
13. Grafikausdruck mit NLQ 401.
14. Hilfsmenü jederzeit aufrufbar und vieles mehr!

Komplett mit Bedienungsanleitung zu einem Preis von nur...

89,00 DM

auf 3 oder 5 1/4 Zoll Discette.

EASY-DISC

Ein Disketten-Manager, der keine Wünsche offen läßt.
15 Diskettenfunktionen und ein-nige Kassettenfunktionen.
Komplett mit Bedienungsanleitung zu einem Preis von nur...

79,00 DM

auf Kassette, oder Diskette in 3 oder 5 1/4 Zoll + 10,00 DM

Bei Bestellung eines Programms bis zum 24.12.1985 gibt's bei uns eine Programmkassette zum Preis von nur 5,00 DM (kein Druckfehler) dazu.

LAGER & RECHNUNG

Lager und Rechnung ist eine Programm zur Eingabe von Waren aller Art. Das Rechnungsprogramm dazu ermöglicht die Rechnungserstellung dazu. Der Warenbestand wird automatisch mit dem Verkauf reduziert. Diese zwei Programme kosten zusammen nur 88,00 DM auf Kassette oder 98,00 DM auf 3 oder 5 1/4 Zoll Diskette.

ANWALT

Anwalt ist ein Programm zur automatischen Berechnung der Zwangsvollstreckung für Anwaltsbüros. Nach Eingabe der Hauptdaten wie Hauptforderung, Zinssatz, Datum, Gebühren, Zinssatz, Datum wird eine Erstberechnung mit Ausdruck durchgeführt. Danach wird nur noch das Datum bis, weitere Kosten oder Zahlung eingegeben und die Berechnung geht weiter. Der Ausdruck erfolgt in Sätzen vom Beginn der Hauptforderung bis zur nächsten Veränderung durch Zahlung usw. Auf Wunsch wird ein gespeicherter Brief direkt mit ausgedruckt. Das Programm ist seit mehr als 1 Jahr in praktischem Einsatz. Preis auf Anfrage. Computertyp und Drucker angeben.

SOUND-EDITOR

Mit Sound-Editor können Sie Ihre eigene Musik erstellen, spielen und speichern. Kassette 39,00 DM Diskette 3 - 5 1/4 Zoll 49,00 DM
Sonder-Dienste: auf Wunsch übersetzen wir Ihr englisches Original Textprogramm in deutscher Sprache mit Bedienungshandbuch und deutschen Umlauten. Preis auf Anfrage. Preisbeispiel Tasword DM 35,00, Tasprint DM 35,00. Sondertextprogramm zur Anwendung von Tasprint nur 49,00 DM ohne Tasprint.

SUCHMAX

Suchmax ist ein Programm zum Suchen, Anzeigen und Austauschen von Buchstaben, Wörtern usw.. Damit können Sie endlich Ihre Programme selbst schnell beliebig verändern. Preis 39,00 DM auf Kassette oder 49,00 DM auf 3 Zoll / 5 1/4 Zoll Diskette.

Paketpreise: EASYTEXT + EASYDATA auf Diskette nur 148,00 DM.

Software Library H. Krohn Mailänder Str. 9
6000 Frankfurt/M 70

BTX Nr.069/681087 Seite #92059191#
Telex 4 189 410 ASU Teletext 69 97 316

Wir wünschen allen Lesern ein recht frohes Weihnachtsfest!

EASY-DATA

für Ihren Schneider
CPC 464, 664, 6128
Funktionen (Auszug)

1. Volleditierung mit DEL, CLR und Pfeiltasten in allen Richtungen.
2. Bildschirmfarben wählbar.
3. Deutsche oder internationale Tastatur einstellbar.
4. Feldlöschung ab Cursorposition nach rechts oder links.
5. Automatische Feldduplizierung.
6. Plausibilitätsprüfung nach vorgegebenen Feldattributen.
7. Satzeinfügung zwischen zwei bestehenden Datensätzen.
8. Satzlöschung (nachfolgende Datensätze rücken vor).
9. Satzumsetzung an eine andere Position in der Datei.
10. Satzsuche nach Satznummer oder Satzinhalt (auch Inhaltssuche in mehreren Feldern gleichzeitig).
11. Sortieren bis 5 Sortierbegriffe gleichzeitig.
12. Aufrufbare Hilfsfunktion.
13. Frei wählbare Eingabemasken für Aufkleber, Karteikarten und vieles mehr.

Komplett mit Bedienungsanleitung zu einem Preis von nur.....

79,00 DM

auf Kassette, oder Diskette in 3 oder 5 1/4 Zoll + 10,00 DM

KASSE

Kasse ist ein Programm als Ersatz für das Durchschreibekas-

senbuch für alle Bargeschäfte (für Gewerbebetriebe geeignet).
Komplett mit Bedienungsanleitung zu einem Preis von nur.....
49,00 DM

auf Kassette, oder Diskette in 3 oder 5 1/4 Zoll + 10,00 DM

1 MB ab 659,-

Das DSB-51 System: 5 1/4 Zoll Drive B mit 1 MB für CPC 464 und 664

2 MB ab 1398,-

Das DSB-52 System: 2 x 5 1/4 Zoll Drive mit je 1 MB für CPC 464 und 664

3 Zoll 180 KB

399,-



5 1/4 Zoll 180 KB

544,-

Data-Service Bonn, 5300 Bonn, Kessenicherstraße 13, Telefon: 0228/235003

Das ist Spitze!

Spiele

Harrier Attack	C	36,-
Ghostbusters	C	45,-
Football Manager	C	36,-
Fighter Pilot	C	36,-
007 James Bond	C	45,-
Karate	C	45,-
Gremlins	C	45,-
Bruce Lee	C	45,-
Daley Thomsons Decathlon	C	32,-
3D-Invaders	C	36,-
Night Lore	C	46,-
Pyjarama	C	36,-
Future World	D	47,-
Strip-Poker	C	37,-
Nazca (Adventure)	D	47,-
Enterprise	D	57,-

Kommerzielle Software

ORGAFAKT (Fakturierung)	D	168,-
ORGAFIBU	D	248,-
Multiplan	D	195,-
DBase II	D	195,-
Wordstar	D	195,-

MegaCad	D	77,-
Basic Compiler	D	95,-
Turbo Pascal	D	225,-
CAD 464	C	67,-
Vereinsverwaltung	D	87,-
Hausverwaltung	D	290,-

Hardware + Zubehör

Drucker Panasonic 1090	890,-
Drucker Quen-Data 1100	850,-
Disketten 3" 10er-Pack	145,-
Diskettenbox (10er) 3"	20,50
Abdeckhaubensatz klar	
(2 Stück Tastatur + Monitor)	35,-
Druckerkabel 2 m	65,-
64 kB Speichererweiterung	125,-

Bitte fordern Sie unsere umfangreiche Preisliste an!
Alle Preise inkl. MwSt.
Lieferung erfolgt per NN zuzügl. Porto

ORGASOFT® GMBH

Rietgasse 6-8 · 7730 VS-Villingen
☎ 0 77 21 / 31 01 · Telex 7 921 500

SCHLUSS MIT DER SEQUENTIELLEN DATEI!

Relative Dateiverwaltung mit DEISYS!

DEISYS verwaltet Ihre kompletten Daten und ist unabhängig vom freien Speicherplatz Ihres Rechners!

Universelle Dateiverwaltung mit frei definierbarer Bildschirmmaske für CPC 464, 664 und 6128!

Leistungsmerkmale in Stichworten:

Integrierter Terminkalender

- nach dem Starten des Programmes werden automatisch die aktuellen Tagetermine angezeigt

- Datenaufnahme unabhängig vom freien Speicherplatz

- relative Datenstruktur

- frei definierbare Bildschirmmaske

- schnellstmögliche Abarbeitungsgeschwindigkeit

- deutscher Zeichensatz, auf ASCII umschaltbar

- max. 20 Datenfelder pro Datensatz

- diverse Suchkriterien

- durchgehende Menüführung, dadurch:

- hoher Bedienungskomfort

DEISYS gibt es auf 3"- und 5 1/4"-Diskette zum Preis von **198,- DM**

VERIS

Das Datenbanksystem für den modernen Versicherungskaufmann! Verwaltet Ihre kompletten Bestände und unterstützt gezielte Verkaufaktionen (z.B: Altersaktion)!

Terminkalender und Datenbank in einem Programm!

- frei definierbare Bildschirmmaske

- druckt Bestände

- max. 20 Datenfelder pro Datensatz

- durchgehende Menüführung

VERIS ist auf 3"- und 5 1/4"-Diskette erhältlich. Preis **348,- DM**

Preis auf Anfrage

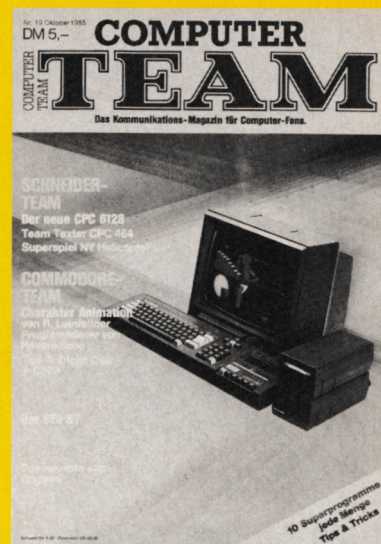
Deitext

Händleranfragen erwünscht.



EDV-Beratung Worms
Rheinbergstraße 14
Postfach 280108
6520 Worms 28
☎ 06242/4597

Teuflich schnell
Enorm bequem
Alle Ausgaben
Mit Sicherheit



**Kündigungsfrist
 NUR
 1 MONAT**

Ihr persönliches **COMPUTER TEAM** Jahres-Abonnement

Nutzen Sie den Preisvorteil von über 16%

COMPUTER TEAM erhalten Sie im Jahresabonnement bequem per Post frei Haus! Dazu sparen Sie noch DM 10,—. Sie zahlen für 12 Hefte nur DM 50,— statt DM 60,— incl. Porto und Verpackung*. Also ein Preisvorteil von über 16%! Nutzen Sie diese Vorteile, indem Sie untenstehenden Coupon ausfüllen und an COMPUTER TEAM-Leser-Service, Mühlenstr. 12, 5431 Boden, einsenden.

* Im Ausland zuzüglich Porto.

Ein **COMPUTER TEAM**-Abo mit über 16% Preisvorteil

Ja, bitte schicken Sie mit ab der nächst erreichbaren Ausgabe **COMPUTER TEAM** zum Jahresvorzugspreis von DM 50,— (statt DM 60,—) incl. Porto und Verpackung*. Sollte ich **COMPUTER TEAM** nicht mehr lesen wollen, kann ich das Abonnement jederzeit mit einer Frist von nur einem Monat schriftlich kündigen und erhalte das Geld für nicht bezogene Hefte zurück.

Bei Lieferung ins Ausland: zuzüglich Porto. Die regelmäßige Lieferung der Hefte erbitte ich an folgende Adresse:

 Name

 Str./Nr.

 PLZ/Ort

 Datum

 Unterschrift

Den Jahresbezugspreis von DM 50,— zahlen Sie erst nach Erhalt der Rechnung

oder bequem und zeitsparend durch bargeldlosen Bankeinzug.

Die Abbuchungsermächtigung gilt bis auf Widerruf.

 Konto-Nr.

 Bankleitzahl

 Bankinstitut

Garantie:

Sie garantieren mir, daß ich die Vereinbarung innerhalb von 10 Tagen schriftlich widerrufen kann.

Zur Wahrung der Frist genügt die rechtzeitige Absendung des Widerrufs (Poststempel) bei **COMPUTER TEAM**, Mühlenstr. 12, 5431 Boden.

Dieses Recht bestätige ich durch meine zweite Unterschrift.

 Unterschrift

COMPUTER TEAM LESERUMFRAGE

**Gewinnen Sie eine Woche Urlaub für
2 Personen in einem 4-Sterne Hotel
am Genfer See**





Leserumfrage

Liebe COMPUTER TEAM-Leser,

Ihre Meinung ist uns etwas wert. Auf den folgenden 2 Seiten finden Sie eine Leserbefragung, mit der wir erfahren möchten, was Ihnen an unserer Zeitschrift besonders gefällt und welche Punkte verbessert werden können. Auch möchten wir wissen, mit welchem Computer-System Sie arbeiten, um das redaktionelle Konzept für die Zukunft daraufhin abzustimmen.

Jeden Tag erreichen uns Leserbriefe, in denen die unterschiedlichsten Wünsche geäußert werden. Heutzutage ist es sicherlich sehr problematisch, jedem gerecht zu werden, zumal immer mehr Computerhersteller die unterschiedlichste Hardware und immer mehr Softwarehersteller die unterschiedlichsten Programme auf dem deutschen Markt anbieten. Durch rege Teilnahme an unserer Leserumfrage helfen Sie uns, COMPUTER TEAM in der Zukunft noch mehr Ihren Wünschen entsprechend zu gestalten.

Deshalb die Bitte an Sie: Beantworten Sie alle Fragen auf den folgenden 2 Seiten und senden Sie diese an COMPUTER TEAM, Leserbefragung, Mühlenstr. 12, 5431 Boden.

Als kleines Dankeschön werden wir unter allen Einsendungen eine Woche Urlaub in einem 4-Sterne-Hotel in der Schweiz für zwei Personen verlosen. Der Gewinner hat die Gelegenheit, eine Woche in dem Sporthotel Panoramic in Villars in der Schweiz zu verbringen. So viel ist Ihre Meinung uns wert.

Gestiftet und organisiert wird der einwöchige Schweizaufenthalt von einem begeisterten Computer-Fan, der in Rheinland-Pfalz und Nordrhein-Westfalen den Verkauf dieses Hotels im Timesharingverfahren anbietet. Für Interessenten sei folgende Kontaktadresse gegeben: Jürgen Wirth, Timesharevertrieb, Steineckstr. 8, 5431 Ruppach-Goldhausen, Tel. 02602-8953.

Zum Schluß sei noch eins bemerkt. Der Fragebogen ist so aufgebaut, daß Sie ihn lediglich aus dem Heft herauszutrennen brauchen und in einen Fensterbriefumschlag stecken müssen. Dann noch eine Achtzigpfennigbriefmarke und ab an den COMPUTER TEAM-Verlag. Wir möchten darauf hinweisen, daß nur ordnungsgemäß frankierte Sendungen an der Verlosung teilnehmen und alle falsch frankierten Sendungen von uns leider nicht angenommen werden können.

Unsere Fragen an Sie:

1 Alter

- Bis 9 Jahre
- 10—19 Jahre
- 20—29 Jahre
- 30—39 Jahre
- 40-49 Jahre
- 50 und älter

2 Beruf

- Schüler/Student/Azubi
- Freier Beruf/Selbständig
- Leitender Angestellter/Beamter
- Facharbeiter
- Arbeiter
- Ohne Beruf

3 Schulbildung

- Volksschule
- Mittlere Reife/Handelsschule
- Abitur/Fachoberschule/Studium

4 Familienstand

- Alleinlebend
- Verheiratet
- Verheiratet mit Kind(ern)

5 Haushalts-Nettoeinkommen

- Bis DM 1000
- Bis DM 2000

- Bis DM 2500
- Bis DM 3000
- Bis DM 4000
- Über DM 4000
- Das geht Sie nichts an

6 Arbeiten Sie beruflich mit einem Computer? ja nein
Wenn ja, mit welchem?

7 Besitz

Haben Sie einen Computer? ja nein

8 Welchen?

9 Wann sitzen Sie am häufigsten am Computer?
Tags Nachts Wochenende

10 Wollen Sie einen Computer kaufen Ja Nein
Wenn ja, welchen?

11 Wenn Sie mehrere Computer haben, welchen benutzen Sie hauptsächlich?

12 Haben Sie zum Computer noch Peripherie?

	Ich besitze	Ich will kaufen
Drucker	<input type="checkbox"/>	<input type="checkbox"/>
Plotter	<input type="checkbox"/>	<input type="checkbox"/>
Floppy	<input type="checkbox"/>	<input type="checkbox"/>
Kassettenrecorder	<input type="checkbox"/>	<input type="checkbox"/>
Interface	<input type="checkbox"/>	<input type="checkbox"/>
Speichererweiterung Typ	<input type="checkbox"/>	<input type="checkbox"/>
Toolkit	<input type="checkbox"/>	<input type="checkbox"/>
Akustik-Koppler	<input type="checkbox"/>	<input type="checkbox"/>
Modem	<input type="checkbox"/>	<input type="checkbox"/>
Joystick	<input type="checkbox"/>	<input type="checkbox"/>
Monitor	<input type="checkbox"/>	<input type="checkbox"/>
Fernseher	<input type="checkbox"/>	<input type="checkbox"/>
Schreibmaschine	<input type="checkbox"/>	<input type="checkbox"/>
Typenrad-Drucker	<input type="checkbox"/>	<input type="checkbox"/>
Sonstiges	<input type="checkbox"/>	<input type="checkbox"/>



13 Können Sie schon programmieren? ja Nein
 Wenn ja, in welcher Sprache?

14 Wollen Sie noch eine Sprache lernen? Ja Nein
 Wenn ja, welche?

15 An welchen Programmen sind Sie hauptsächlich interessiert:
 Spiele
 Grafik
 Anwenderprogramme
 Utilities

16 Was machen Sie mit Ihrem Micro-Computer?
 Spielen
 Lernen
 Archivieren (Sammlungen usw.)
 Terminkalender führen
 Briefe schreiben
 Berufliche Fortbildung
 Programmieren lernen
 Weiß ich noch nicht
 Sprechen/Verstehen Sie englisch Ja
 Nein

17 Nutzen (oder wollen Sie nutzen) Sie Ihren Computer überwiegend Haushalt usw.? Ja Nein
 Beruflich für die eigene Firm Ja Nein
 Beruflich, um an Ihrem Arbeitsplatz weiterzukommen? Ja Nein

18 Lesegewohnheiten:
 Lesen Sie außer COMPUTER TEAM noch andere Fachzeitschriften? Ja
 Nein
 Wenn ja, welche

19 Wenn Sie den Inhalt von COMPUTER TEAM vorschlagen könnten, welchen Themen würden Sie mehr Raum geben oder besonders vorziehen?

20 Wie viele Personen lesen außer Ihnen noch COMPUTER TEAM?

21 Meinen Computerezubehör kaufe ich vorwiegend:
 Im Warenhaus
 beim Computerhändler
 im Versandhandel
 auf dem Gebrauchtmart über Kleinanzeigen

22 Was veranschlagen Sie an Neuananschaffungskosten, um Ihre Ausrüstung zu vervollkommen (pro Jahr)?
 _____ DM

23 Was würden Sie ausgeben für neue Computer _____ DM
 Drucker/Plotter _____ DM
 Diskettenstation _____ DM
 Fertige Programme _____ DM

24 Wenn Sie einen Personal-Computer kaufen wollen, wonach richten sie sich?
 Berichte in Fachzeitschriften?
 Rat des örtlichen Fachhändlers?
 Beratung durch einen unabhängigen Experten

25 Geschlecht
 Männlich
 Weiblich

Teilnahmebedingungen: Unter den eingesandten Fragebogen werden — unter Ausschluß des Rechtsweges — die ausgelobten Preise ausgelost. Einsendeschluß ist der 31.12.1985 (Datum des Poststempels). Die Gewinner werden in der Ausgabe 3/4 1986 veröffentlicht und/oder persönlich benachrichtigt.

Wichtige rechtliche Garantie!

Wir garantieren, daß die von Ihnen gemachten Angaben ohne jegliche personenbezogenen Daten (Name, Anschrift usw.) lediglich zu statistischen Zwecken elektronisch gespeichert und verarbeitet werden. Sämtliche Einzeldaten fallen zudem unter das Redaktionsgeheimnis. Zusätz-

liche Sicherheit: Sie müssen diesen Fragebogen nicht mit Ihrer Anschrift versehen, es genügt uns auch lediglich Ihre Postleitzahl! (Dabei verzichten Sie allerdings auf die Teilnahme an unserem Gewinnspiel.)

COMPUTER TEAM
Mühlenstraße 12

5431 Boden

Vorname _____
 Name _____
 Straße/Hausnr. _____
 Postleitzahl/Ort _____
 Telefon (mit Vorwahl) _____

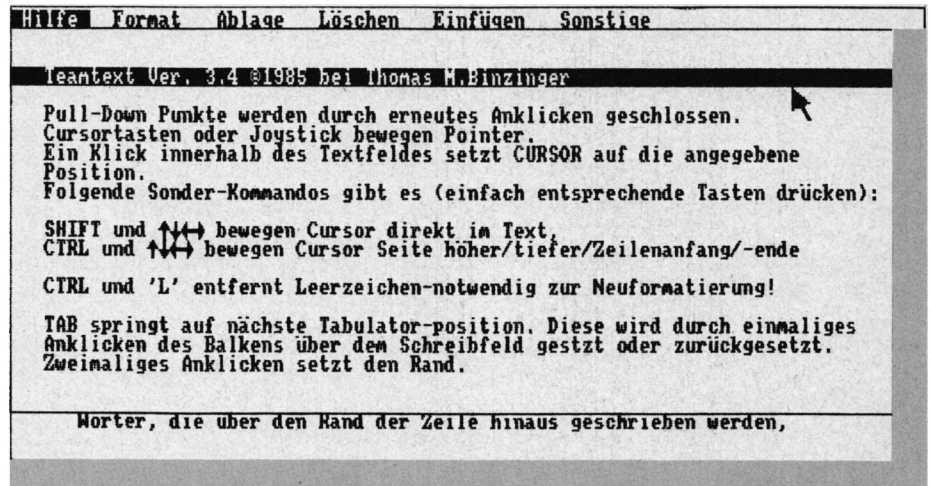
TEAM-TEXTER

So einfach schreiben wie noch nie!

Schon wieder eine Textverarbeitung, werden Sie jetzt vielleicht sagen. Das ist zwar richtig, aber Team-Text ist nicht eines der üblichen Textverarbeitungsprogramme. Denn dieses Programm wurde nicht um seiner selbst willen, d.h. nur als Textverarbeitungsprogramm, geschrieben. Vielmehr soll damit demonstriert werden, daß „Techniken, wie sie auf den modernsten Computern wie Macintosh, Amiga und Atari 520 Verwendung finden, auch auf dem Schneider realisierbar sind. Das heißt im einzelnen: Pull-Down-Menüs, ein Pointer (der aber aus verständlichen Gründen nicht mit einer Maus, sondern mit den Cursor-tasten oder einem Joystick bewegt werden muß) und die damit verbundene Window-Technik und Bedienungstechniken.

Zur Textverarbeitung

Team-Text ist in der Lage, 200 Zeilen Text zu verwalten. Wörter, die über den Rand der Zeile hinaus geschrieben werden, werden automatisch in die nächste Zeile geschoben, ohne daß man beim Schreiben darauf achten muß. Wenn eine Zeile vollgeschrieben ist, wird Sie automatisch formatiert, und zwar je nach dem angewählten Format links- oder rechtsbündig,

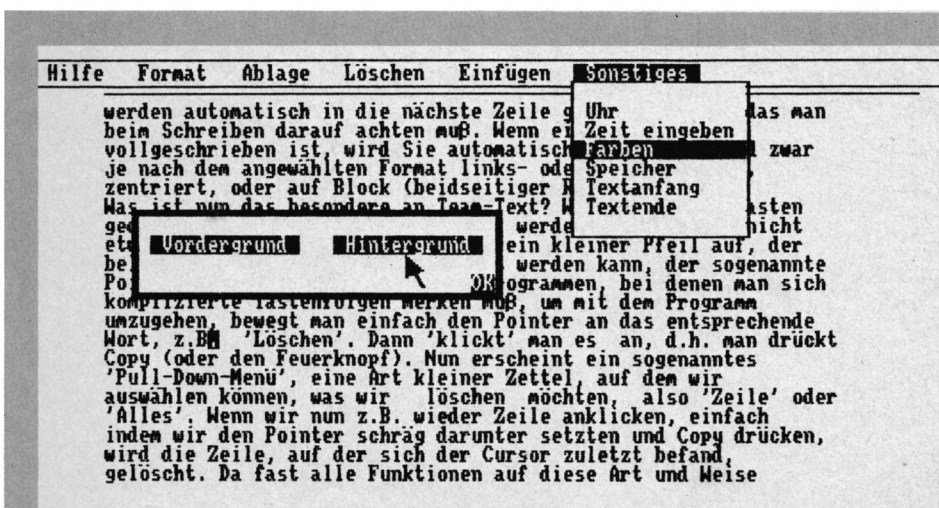


zentriert oder auf Block (beidseitiger Randausgleich). Was ist nun das Besondere an Team-Text? Wenn die Cursortasten gedrückt (oder der Joystick bewegt) werden, bewegt sich nicht etwa der Cursor, sondern es taucht ein kleiner Pfeil auf, der beliebig über den Bildschirm bewegt werden kann, der sogenannte Pointer. Im Gegensatz zu anderen Programmen, bei denen man sich komplizierte Tastenfolgen merken muß, um mit dem Programm umzugehen, bewegt man einfach den Pointer an das entsprechende Wort, z.B. 'Löschen'. Dann 'klickt' man es an, d.h., man drückt Copy (oder den Feuerknopf). Nun erscheint ein sogenanntes 'Pull-Down-Menü', eine Art kleiner Zettel, auf dem man aus-

wählen kann, was gelöscht werden soll, also 'Zeile' oder 'Alles'. Wenn wir nun z.B. Zeile anklicken, einfach indem wir den Pointer schräg darunter setzen und Copy drücken, wird die Zeile gelöscht, auf der sich der Cursor zuletzt befand. Da fast alle Funktionen auf diese Art und Weise ausgewählt werden, kann man schon nach sehr kurzer Eingewöhnungszeit mit dem Programm umgehen. Selbst Personen mit wenig Computererfahrung können schnell mit dem Programm arbeiten, da das Zeigen und Auswählen mit dem Pointer sehr viel anschaulicher ist, als sich endlose Tastenfolgen zu merken.

Folgendes sollte beachtet werden, wenn man mit Team-Text arbeitet: Der Pointerpfeil muß rechts unter der Stelle stehen, auf die man mit ihm zeigt. Ein 'Klick' innerhalb des Textes setzt den Cursor auf die angegebene Stelle. Der Cursor läßt sich auch selber bewegen, und zwar indem man SHIFT und die Cursortasten benutzt. Mit CTRL und den Cursortasten kann man den Cursor 10 Zeilen höher/tiefer bzw. an den Zeilenanfang und das Zeilenende stellen.

Der Pointer verschwindet automatisch, wenn man weiterschreibt. Dann taucht nämlich der Cursor wieder an der Textstelle auf, an der er vorher war. Normalerweise wird die Zeile, die man gerade geschrieben hat, automatisch formatiert (entsprechend dem unter




```

5 ' TeamText 2
6 ' Copyright 1985 Thomas M.Binzinger
10 MEMORY HIMEM-(200*80)-1:DEFINT a-y
20 GOSUB 60:GOSUB 240:a=REMAIN(0):EVERY 3000,0 GO
SUB 1460
30 GOSUB 2000:GOSUB 480:GOSUB 2060:REM Zeile hole
n und formatieren
40 z$=e$:GOSUB 2030:zeile=MIN(zeile+1,200):lzeile
=MAX(zeile,lzeile):cy=cy+1:GOTO 30
50 a$=INKEY$:IF a$="" THEN 50 ELSE PRINT ASC(a$):
GOTO 50
60 REM ** Initialisieren **
70 REM
80 maxi=200:DIM tabs(80),pd$(6):z$=SPACE$(80)
90 lrand=7:rrand=77:FOR x=1 TO 80 :tabs(x)=-1:NEX
T:z$=SPACE$(80):FOR zeile=1 TO maxi:GOSUB 2030:NE
XT:zeile=1:screentop=1:lzeile=1:cx=1:cy=1:ppos=84
0:vorder=24:hinter=1:farben=0:sk=0
100 PEN 1:PAPER 0:MODE 2:WINDOW #0,1,80,4,25:WIND
OW #1,1,80,1,3
110 a$=CHR$(13)+CHR$(10)+" ":pd$(2)=" keins"+a$+"
rechts"+a$+"zentriert"+a$+"Blocksatz":pd$(3)=" Te
xt abspeichern"+a$+"Text einlesen"+a$+"Text druck
en"+a$+"Ab Cursor drucken":pd$(4)=" Zeile"+a$+"Al
les":pd$(5)=" Zeile"+a$+"Seite"
120 pd$(6)=" Uhr"+a$+"Zeit eingeben"+a$+"Farben"+
a$+"Speicher"+a$+"Textanfang"+a$+"Textende":pd$(1
)=" "
130 w(1)=2:w(2)=1:w(3)=1:w(4)=1:w(5)=1:w(6)=1:w(7
)=1
140 SYMBOL 123,198,0,120,12,124,204,118,0
150 SYMBOL 125,198,0,102,102,102,102,62,0
160 SYMBOL 124,198,0,60,102,102,102,60,0
170 SYMBOL 126,120,198,198,252,198,198,248,192
180 SYMBOL 91,219,60,102,102,126,102,102,0
190 SYMBOL 93,198,0,198,198,198,198,60,0
200 SYMBOL 92,198,56,198,198,198,108,56,0
210 KEY DEF 17,1,123,91:KEY DEF 19,1,125,93:KEY D
EF 26,1,124,92:KEY DEF 24,1,126:KEY 128,CHR$(64):
RETURN
220 GOTO 2310
230 REM
240 REM ** Statuszeile **
250 LOCATE #1,1,3:PRINT #1,STRING$(80,32);:status
$=" Hilfe Format Ablage Löschen Einfügen
Sonstiges":PLOT 0,358,0:DRAW 640,358 ELSE 280
260 FOR x=1 TO 80:IF tabs(x)=1 THEN PLOT 8*x-4,35
2,0:DRAW 8*x-4,360:PLOT 8*x-3,352:DRAW 8*x-3,360
270 NEXT
280 a$=SPACE$(77):öPRINT,82,0,76,5a$:öPRINT,80,0,
56,5status$
290 PLOT lrand*8-8,358,1:DRAW rrand*8,358
300 PLOT 0,387:DRAW 639,387,1:DRAW 639,364:DRAW 0
,364:DRAW 0,387
310 FOR x=1 TO 80:IF tabs(x)=1 THEN IF x>=lran
d AND x<=rrand THEN PLOT 8*x-4,352:DRAW 8*x-4,360
,1:PLOT 8*x-3,352:DRAW 8*x-3,360
320 NEXT
330 RETURN
340 REM
350 REM ** Pull down Menue **
360 GOTO 380
370 DATA 2,76,20,Hilfe,10,11,6,Format,19,19,6,Abl
age,28,8,4,"Löschen",38,9,4,Einfügen,49,14,8,Sons
tiges
380 RESTORE 370:q=nr:z=1
390 READ rr,br,ho,u$:nr=nr-1:IF nr<>0 THEN 390
400 rr=rr-1
410 WINDOW #2,rr,rr+br,3,2+ho:CLS #2
420 PLOT rr*8-9,400-2*16,1:DRAW rr*8-9,400-(2+ho)
*16-2
430 IF rr=1 THEN PLOT rr*8-8,400-2*16,1:DRAW rr*8
-8,400-(2+ho)*16-2
440 PLOT rr*8-10,400-2*16:DRAW rr*8-10,400-(2+ho)
*16-2
450 DRAW (rr+br)*8+1,400-(2+ho)*16-2
460 DRAW (rr+br)*8+1,400-2*16:PLOT (rr+br)*8+2,40
0-2*16:DRAW (rr+br)*8+2,400-(2+ho)*16-2:PLOT rr*8
-2,400-2*16:DRAW (rr+br)*8+2,400-2*16:LOCATE #1,r
r,2:PRINT #1,CHR$(24)" "u$" "CHR$(24);:WINDOW SWA
P 0,2:PRINT:PRINT pd$(q);:nr=q:IF nr=1 THEN GOSUB
2340
470 w(7)=w(nr):öINV,(w(nr)+2)*80+rr-1,br+1,1,1:WI
NDOW SWAP 0,2:RETURN
480 REM
490 REM ** Zeile editieren **
500 IF cy>20 THEN IF screentop+10<=230 THEN ho=20
:screentop=screentop+10:cy=cy-10:GOSUB 1170
510 cx=lrand:e$=LEFT$(z$,80):IF wort$<>" THEN MI
D$(e$,lrand,LEN(wort$))=wort$:cx=cx+LEN(wort$):wo
rt$=""
520 öPRINT,(cy+2)*80,0,79,5e$:LOCATE cx,cy
530 LOCATE cx,cy:CALL &BB8A:REM cursor on
540 GOSUB 810:in$=i$:i=ASC(in$)
550 IF i=224 THEN PRINT CHR$(7);:GOTO 540
560 IF i=244 THEN IF zeile=1 THEN 520 ELSE CALL &
BB8D:z$=e$:GOSUB 2030:IF zeile=screentop THEN scr
eentop=MAX(screentop-20,1):zeile=screentop+19:cy=
20:GOSUB 2000:e$=z$:GOSUB 1170:GOTO 520 ELSE zeil
e=zeile-1:cy=cy-1:GOSUB 2000:e$=z$:GOTO 520
570 IF i=245 THEN IF zeile=lzeile THEN 520 ELSE C
ALL &BB8D:z$=e$:GOSUB 2030:IF zeile=screentop+19
THEN screentop=MIN(screentop+20,lzeile-20):zeile=
screentop:cy=1:GOSUB 2000:e$=z$:GOSUB 1170:GOTO 5
20 ELSE zeile=zeile+1:cy=cy+1:GOSUB 2000:e$=z$:GO
TO 520
580 IF i=6 THEN GOSUB 2540:qq=cx:cx=rrand:GOSUB 2
060:cx=qq:i=245:CALL &BB8D:GOTO 570
590 IF i=12 THEN GOSUB 2540:GOTO 520:REM Leerzei
chen entfernen
600 IF i=13 THEN öPRINT,(cy+2)*80,0,79,5e$:wort$=
"":RETURN
610 IF i=127 THEN CALL &BB8D:MID$(e$,cx,1)=" ":IF
cx>lrand THEN MID$(e$,cx,rrand-cx)=MID$(e$,cx+1,
rrand-cx):MID$(e$,rrand,1)=" ":cx=cx-1:GOTO 520 E
LSE 520
620 IF i=16 THEN MID$(e$,cx,rrand-cx)=MID$(e$,cx+
1,rrand-cx):MID$(e$,rrand,1)=" ":GOTO 520
630 IF i=250 THEN cx=lrand:GOTO 520 ELSE IF i=251
THEN cx=rrand:WHILE cx>lrand AND MID$(e$,cx,1)="
":cx=cx-1:WEND:GOTO 520
640 IF i=246 AND cx>1 THEN CALL &BB8D:cx=cx-1:GOT
O 530
650 IF i=247 AND cx<80 THEN CALL &BB8D:cx=cx+1:GO
TO 530
660 IF i=248 THEN z$=e$:GOSUB 2030:screentop=MAX(
screentop-20,1):zeile=screentop:cy=1:GOSUB 2000:e
$=z$:GOSUB 1170:GOTO 520
670 IF i=249 THEN z$=e$:GOSUB 2030:screentop=MIN(
screentop+20,lzeile-20):zeile=screentop+19:cy=20:
GOSUB 2000:e$=z$:GOSUB 1170:GOTO 520
680 IF i=9 AND cx<>80 THEN CALL &BB8D:x=cx+1:WHIL
E tabs(x)<>1 AND x<=rrand:x=x+1:WEND:cx=x:GOTO 530
690 IF i<32 THEN RETURN
700 IF cx<lrand OR cx>rrand THEN CALL &BB8D:PRINT
CHR$(7);:GOTO 530
710 IF cx>=rrand THEN 720 ELSE MID$(e$,cx,rrand-c
x)=" "+MID$(e$,cx,rrand-(cx+1))
720 MID$(e$,cx,1)=in$:öPRINT,(cy+2)*80,0,79,5e$:c
x=cx+1:IF cx<=rrand+1 THEN 530
730 IF in$="" THEN RETURN
740 u$="" :x=rrand: WHILE x>lrand AND u$<>" ":u$=M
ID$(e$,x,1):x=x-1:WEND

```

```

750 IF x=1rand THEN RETURN
760 wort$=MID$(e$,x+1,rrand-x-1)+in$:MID$(e$,x+1,
rrand-LEN(wort$))=SPACE$(LEN(wort$)):LOCATE c+1,c
y:PRINT SPACE$(LEN(wort$))
770 IF LEFT$(wort$,1)=" " THEN wort$=RIGHT$(wort$,
LEN(wort$)-1)
780 öPRINT, (cy+2)*80,0,79,5e$
790 RETURN
800 REM
810 REM ** Keys abfragen& Pointer bewegen **
820 wopen=0:ges=0:ppos=cy*80+cx+240-1:n$=CHR$(9)+
CHR$(10)+CHR$(8)+CHR$(11)+CHR$(88)+CHR$(240)+CHR$(
241)+CHR$(242)+CHR$(243)+CHR$(224)
830 i$=INKEY$:IF i$<" " AND JOY(0)=0 THEN 830
840 IF JOY(0)<>0 THEN 850 ELSE i=ASC(i$):IF i<240
OR i>243 THEN RETURN
850 GOSUB 1300
860 u=ppos
870 j=JOY(0):IF INKEY(0)=0 OR (J AND 1)=1 THEN u=
u-80
880 IF INKEY(2)=0 OR (J AND 2)=2 THEN u=u+80
890 IF INKEY(8)=0 OR (J AND 4)=4 THEN u=u-1
900 IF INKEY(1)=0 OR (J AND 8)=8 THEN u=u+2
910 IF INKEY(1)=0 OR (J AND 8)=8 THEN u=u+1
920 a$=INKEY$:IF a$<" " THEN IF wopen=0 AND INSTR
(n$,a$)=0 THEN öEXC,ppos*1,0,1,1:i$=a$:IF ges=1 T
HEN GOSUB 1300:öINV,79+a,1,e1-a,1,1:GOSUB 1300:ge
s=0:a1=0:e1=0:RETURN ELSE RETURN
930 IF loadp=0 AND farben=0 THEN GOSUB 1230
940 IF wopen=0 AND farben=0 THEN IF u>80*23+80 TH
EN GOSUB 1300:z$=e$:GOSUB 2030:ho=20:GOSUB 1170:d
=zeile-screentop:screentop=MIN(lzeile,screentop+1
0):GOSUB 1170:zeile=screentop+d:GOSUB 2000:e$=z$:
GOSUB 1300
950 GOSUB 1010:REM Menuebalken
960 IF INKEY(9)=0 OR (JOY(0) AND 16)=16 THEN IF f
arben=1 THEN GOSUB 1680 ELSE IF loadp=1 THEN GOSU
B 1760 ELSE GOSUB 1090
970 IF u=ppos THEN 870
980 IF u<0 OR u>80*23+80 THEN 860
990 GOSUB 1300:ppos=u:GOTO 850
1000 DATA 2,7,10,16,19,25,28,35,38,46,49,57
1010 IF wopen=1 THEN RETURN ELSE IF ges=1 AND ppo
s-161>=a1-2 AND ppos-161<e1 THEN RETURN
1020 IF ges=1 THEN GOSUB 1300:öINV,79+a,1,e1-a,1,
1:GOSUB 1300:ges=0:a1=0:e1=0
1030 x=1:q=ppos-160:RESTORE 1000
1040 IF ppos<161 OR ppos>258 THEN RETURN
1050 READ a,e:IF (q>a-2 AND q<e) THEN nr=x:GOTO 1
080
1060 x=x+1:IF x=7 THEN RETURN
1070 GOTO 1050
1080 GOSUB 1300:öINV,79+a,e-a,1,1:a1=a:e1=e:ges=1
:GOSUB 1300:RETURN
1090 REM ** Open window **
1100 IF wopen=0 AND ppos>240 AND ppos<320 THEN 18
80:REM Schreibzone
1110 IF wopen=0 AND ppos>319 THEN z$=e$:GOSUB 203
0:GOSUB 1300:öPRINT, (cy+2)*80,0,80,5z$:GOSUB 1210
:cx=px:cy=py-4:zeile=screentop+cy-1:GOSUB 2000:e$
=z$:LOCATE cx,cy:CALL &BB8A:GOSUB 1300:RETURN
1120 GOSUB 2000:z$=e$
1130 IF wopen=1 AND ppos-161>=a1-2 AND ppos-161<e
1 AND farben=0 THEN GOSUB 1300:GOSUB 240:GOSUB 11
70:GOSUB 1300:wopen=0:RETURN
1140 IF ges=0 THEN RETURN
1150 GOSUB 1300:öINV,79+a,e-a,1,1:a1=a:e1=e:ges=0
:GOSUB 350:GOSUB 1300:wopen=1:RETURN
1160 REM
1170 REM **Window Rest weg **
1180 p=240:x=zeile:FOR zeile=screentop TO screent

```

```

op+ho-1:GOSUB 2000:öPRINT,p*1,0,80,5z$:p=p+80:NEX
T:zeile=x:GOSUB 2000
1190 RETURN
1200 REM
1210 REM ** Errechne px/py **
1220 py=INT(ppos/80):px=ppos-py*80:px=px+1:py=py+
1:RETURN
1230 REM ** Mark Pull-Down Punkte **
1240 IF wopen=0 THEN RETURN
1250 GOSUB 1210:IF px<rr OR px>rr+br OR py<5 OR p
y>ho+2 THEN RETURN
1260 IF JOY(0)=16 OR INKEY(9)=0 THEN w(nr)=w(7):G
OSUB 1500:PRINT CHR$(7);
1270 q=nr:IF w(7)=py-4 THEN RETURN
1280 GOSUB 1300:öINV, (w(7)+2)*80+rr-1,br+1,1,1
1290 w(7)=py-4:öINV, (py-2)*80+rr-1,br+1,1,1:GOSUB
1300:RETURN
1300 REM ** Exchange Pointer&Screen **
1310 öEXC,ppos*1,0,1,1:RETURN
1320 REM ** Uhr **
1330 REM mathematische Unterstützung:"Jörg Rohde"
1340 pl=ppos:GOSUB 1300:WINDOW #2,10,30,9,18:PAPE
R #2,1:PEN #2,0:CLS #2:PEN #2,1:PAPER #2,0
1350 DEG:PLOT 218,191,0:FOR w=0 TO 360 STEP 12:DR
AW COS(w)*60+158,SIN(w)*60+191:NEXT:FOR w=0 TO 36
0 STEP 30:x1=COS(w)*55+158:y1=SIN(w)*55+191:x2=CO
S(w)*70+158:y2=SIN(w)*70+191:PLOT x1,y1:DRAW x2,y
2:NEXT:m=minute:s=stunde:ppos=1389:GOSUB 1300
1360 ml=minute:s1=stunde:minute=m:stunde=s:PLOT 1
58,191,1:GOSUB 1410:stunde=s1:minute=ml:PLOT 158,
191,0:GOSUB 1410:GOTO 1370
1370 m=minute:s=stunde
1380 IF m=minute AND INKEY(9)<>0 AND JOY(0)<>16 T
HEN 1380
1390 IF INKEY(9)=0 OR JOY(0)=16 THEN GOSUB 1300:p
pos=pl:GOSUB 1300:RETURN
1400 GOTO 1360
1410 w=(11-stunde)*30+90+(60-minute)/2
1420 PLOT 158,191:x1=COS(w-20)*15+158:y1=SIN(w-20
)*15+191:x2=COS(w+20)*15+158:y2=SIN(w+20)*15+191:
x3=COS(w)*53+158:y3=SIN(w)*53+191:DRAW x1,y1:DRAW
x3,y3:DRAW x2,y2:DRAW 158,191:w=(60-minute)*6+90
:x3=COS(w)*53+158:y3=SIN(w)*53+191:DRAW x3,y3:RET
URN
1430 REM Freien Speicher darstellen
1440 FOR y=9 TO 17:LOCATE 9,y:PRINT " ";NE
XT:LOCATE 9,9:PRINT"100%";:LOCATE 12,17:PRINT"0%"
:FOR x=116 TO 120:PLOT x,80,1:DRAW x,80+(maxi-lz
eile)*(144/maxi):NEXT:RETURN
1450 REM Uhr weiterzaehlen
1460 minute=minute+1:IF minute=60 THEN minute=0:s
tunde=stunde+1:IF stunde=12 THEN stunde=0
1470 RETURN
1480 REM ** Schreibe linke Haelfteneu **
1490 IF lp=0 THEN RETURN ELSE p=240:x=zeile:FOR z
eile=screentop TO screentop+19:GOSUB 2000:öPRINT,
p*1,0,lp*1,5z$:p=p+80:NEXT:zeile=x:GOSUB 2000:RET
URN
1500 REM ** Bearbeite geklickte Pulldownpunkte **
1510 w=w(7):IF nr=6 AND w=1 THEN GOSUB 1340:lp=35
:GOSUB 1480:RETURN
1520 IF nr=6 AND w=4 THEN GOSUB 1300:p1=ppos:a$=S
TRING$(20,127):FOR p=800 TO 1600 STEP 80:öPRINT,p
*1,0,19,5A$:NEXT:GOSUB 1430:ppos=1618:GOSUB 1300:
lp=20:WHILE INKEY(9)<>0 AND JOY(0)<>16:WEND:GOSUB
1300:GOSUB 1480:ppos=p1:GOSUB 1300:RETURN
1530 IF nr=6 AND w=3 THEN farben=1:x=10:y=10:b=30
:h=3:GOSUB 1660:WINDOW SWAP 0,3:PRINT:PRINT "CHR
$(24)" Vordergrund "CHR$(24)" "CHR$(24)" Hinter
grund ":LOCATE 30,4:PRINT"OK"CHR$(24);:WINDOW SWA
P 0,3:RETURN

```



```

1540 IF nr=6 AND w=5 THEN GOSUB 1300:GOSUB 240:z$
=e$:GOSUB 2030:ho=20:screentop=1:cx=1rand:cy=1:ze
ile=1:GOSUB 1170:LOCATE cx,cy:CALL &BBBA:wopen=0:
GOSUB 1300:GOSUB 2000:e#=z$:RETURN
1550 IF nr=6 AND w=6 THEN GOSUB 1300:GOSUB 240:z$
=e$:GOSUB 2030:ho=20:cx=1rand:zeile=lzeile:GOSUB
2000:screentop=lzeile-20:cy=20:IF screentop<1 THE
N screentop=1
1560 IF nr=6 AND w=6 THEN GOSUB 1170:CALL &BBBA:w
open=0:GOSUB 1300:GOSUB 2000:e#=z$:RETURN
1570 IF nr=3 AND w=2 THEN GOSUB 1300:GOSUB 1170:G
OSUB 240:x=4:y=7:b=70:h=15:GOSUB 1660:WINDOW SWAP
0,3:CAT:WINDOW SWAP 0,3:GOSUB 1300:loadp=1:RETUR
N
1580 IF nr=4 AND w=2 THEN FOR zeile=1 TO lzeile:z
$=SPACE$(80):GOSUB 2030:NEXT:GOSUB 1300:GOSUB 240
:ho=21:GOSUB 1170:GOSUB 1300:wopen=0:lzeile=1:cx=
rrand:cy=1:zeile=1:GOSUB 2000:e#=SPACE$(80):RETUR
N
1590 IF nr=3 AND w=1 THEN GOTO 1930
1600 IF nr=2 THEN GOSUB 1300:GOSUB 240:GOSUB 1170
:wopen=0:GOSUB 1300:RETURN
1610 IF nr=6 AND w=2 THEN 2140
1620 IF nr=3 THEN IF w=3 THEN 2210 ELSE IF w=4 TH
EN 2230
1630 IF nr=5 THEN IF w=1 THEN 2420 ELSE 2440
1640 IF nr=4 THEN IF w=1 THEN 2510
1650 RETURN
1660 REM ** Open extra Window
1670 WINDOW #3,x,x+b,y,y+h:CLS #3:FOR a=1 TO 5:PL
OT (x-1)*8-a,416-y*16+a,1:DRAW (x-1)*8-a,399-(y+h
)*16-a:DRAW (x+b)*8+a,399-(y+h)*16-a:DRAW (x+b)*8
+a,416-y*16+a:DRAW (x-1)*8-a,416-y*16+a:NEXT:RETU
RN
1680 REM ** Handle Farbklick
1690 IF ppos<890 THEN RETURN ELSE IF ppos<903 THE
N vorder=vorder+1:IF vorder>27 THEN vorder=0
1700 IF ppos>905 AND ppos<918 THEN hinter=hinter+
1:IF hinter>27 THEN hinter=0
1710 IF vorder=hinter THEN vorder=vorder+1
1720 INK 0,hinter:BORDER hinter:INK 1,vorder
1730 IF ppos>1077 AND ppos<1081 THEN farben=0:lp=
41:GOSUB 1300:GOSUB 1480:GOSUB 1300
1740 FOR t=1 TO 100:NEXT
1750 RETURN
1760 REM ** Waehle Programm aus
1770 IF ppos<803 THEN 1860
1780 ba=&91CD+2048
1790 anz=0:z=ba+11:WHILE PEEK(z)<>0 AND z<&BA+641
:anz=anz+1:z=z+14:WEND:IF ppos>ROUND((anz+1)/3+0.
5)*80+779 THEN 1860
1800 u=0:p=ppos MOD 80:l1=ppos-p-80:IF p>2 AND p<
15 THEN GOSUB 1300:öINV,11+3,12,1,1:GOSUB 1300:u=
1
1810 IF p>22 AND p<35 THEN GOSUB 1300:öINV,23+11,
12,1,1:GOSUB 1300:u=2
1820 IF p>42 AND p<55 THEN GOSUB 1300:öINV,11+43,
12,1,1:GOSUB 1300:u=3
1830 IF u=0 THEN 1860 ELSE GOSUB 1300:l1=11-720:l
1=11/80:l1=((u-1)*ROUND(anz/3))+11)*14:n$="":FOR
z=11+ba TO 11+ba+10:n$=n$+CHR$(PEEK(z)):NEXT:n$=
LEFT$(n$,8)+". "+RIGHT$(n$,3):zeile=1:GOSUB 1300:ö
PENIN n$
1840 WHILE EOF<>-1 AND zeile<251:LINE INPUT #9,a$
:z$=SPACE$(80):MID$(z$,1,LEN(a$))=a$:GOSUB 2030:z
eile=zeile+1:IF zeile>lzeile THEN lzeile=zeile
1850 WEND:x=zeile:WHILE zeile<lzeile:z$=SPACE$(80
):GOSUB 2030:zeile=zeile+1:WEND:lzeile=x:zeile=1:
GOSUB 2000:e#=z$:screentop=1:cx=1:cy=1
1860 GOSUB 1300:ho=20:GOSUB 1170:loadp=0:wopen=0:
GOSUB 1300

```

```

1870 RETURN
1880 REM Schreibzone aendern
1890 GOSUB 1210:IF ppos<>sk THEN sk=ppos:IF tabs(
px)=1 THEN tabs(px)=0:GOTO 1910 ELSE tabs(px)=1:ö
OTO 1910
1900 sk=0:pm=1rand+(rrand-1rand)/2:IF px<pm THEN
1rand=px ELSE rrand=px
1910 PLOT 0,358,0:DRAW 639,358:GOTO 240
1920 REM Abspeichern
1930 GOSUB 1300:x=10:y=7:b=55:h=5:GOSUB 1660
1940 WINDOW SWAP 0,3:PRINT" Bitte geben Sie den F
ilenamen an:"
1950 PRINT" ";CALL &BBBD:INPUT N$:IF LEN(N$)>8 T
HEN PRINT" Filename zu lang.":GOTO 1940
1960 IF N$="" THEN 1990
1970 N$=N$+".TXT":OPENOUT N$:x=zeile:FOR zeile=1
TO lzeile:GOSUB 2000:PRINT #9,z$:NEXT:zeile=x
1980 CLOSEOUT
1990 WINDOW SWAP 0,3:GOSUB 240:ho=20:GOSUB 1170:w
open=0:GOSUB 1300:RETURN
2000 REM z$ ← Text
2010 IF LEN(z$)<80 OR zeile<1 THEN RETURN
2020 öSE,0,HIMEM+1+((zeile-1)*80),PEEK($z$+2)*256
+PEEK($z$+1),0:RETURN
2030 REM z$ → Text
2040 IF LEN(z$)<80 OR zeile<1 THEN RETURN
2050 öSE,0,PEEK($z$+2)*256+PEEK($z$+1),HIMEM+1+((
zeile-1)*80),0:RETURN
2060 REM Zeile formatieren
2070 IF w(2)=1 THEN RETURN
2080 IF MID$(e$,1rand,rrand-1rand+1)=SPACE$(rrand
-1rand+1) THEN RETURN
2090 IF w(2)=2 THEN x=rrand:WHILE x>=1rand AND MI
D$(e$,x,1)=" ":x=x-1:WEND:IF x=1rand THEN RETURN
ELSE MID$(e$,1rand,rrand-1rand+1)=SPACE$(rrand-x)
+MID$(e$,1rand,x-1rand+1):GOTO 2130
2100 IF w(2)=3 THEN x=rrand:WHILE x>=1rand AND MI
D$(e$,x,1)=" ":x=x-1:WEND:IF x=1rand THEN RETURN
ELSE MID$(e$,1rand,rrand-1rand+1)=SPACE$((rrand-x
)/2)+MID$(e$,1rand,x-1rand+1)+SPACE$((rrand-x)/2)
:GOTO 2130
2110 IF cx<1rand+(rrand-1rand)/2 THEN RETURN ELSE
IF INSTR(e$," ")=0 THEN RETURN ELSE IF MID$(e$,r
rand,1)<>" " AND MID$(e$,1rand,1)<>" " THEN 2130
2120 i=RND*(rrand-1rand):i=i+1rand:i=INSTR(i,e$,"
"):a$=MID$(e$,i+1,rrand-i):MID$(e$,i+1,rrand-i)=
" "+a$:GOTO 2110
2130 öPRINT,(cy+2)*80,0,79,5e$:RETURN
2140 REM Uhr stellen
2150 x=10:y=6:b=55:h=10:GOSUB 1300:GOSUB 1660:WIN
DOW SWAP 0,3:PRINT" Uhr stellen.":PRINT
2160 stunde=-1:minute=-1
2170 INPUT " Stunden (0-12) (ENTER):",stunde
2180 INPUT " Minuten (0-60) (ENTER):",minute
2190 IF stunde<0 OR stunde>12 OR minute<1 OR minu
te>59 THEN PRINT" Falsche Eingabe!":GOTO 2160
2200 WINDOW SWAP 0,3:ho=20:GOSUB 240:GOSUB 1170:w
open=0:GOSUB 1300:RETURN
2210 REM Text ganz drucken
2220 sz=1:GOTO 2240
2230 sz=zeile:REM Text ab Cursor drucken
2240 x=10:y=5:h=10:b=55:GOSUB 1300:GOSUB 1660:WIN
DOW SWAP 0,3:PRINT:PRINT" Machen Sie bitte den Dr
ucker bereit und drücken Sie.":PRINT:PRINT" 'F' f
ür Fettdruck":PRINT" 'K' für Komprimiert":PRINT"
'N' für Normal":a$=""
2250 a$=INKEY$:IF a$="" THEN 2250 ELSE a$=UPPER$(
a$):IF a$<>"N" AND a$<>"F" AND a$<>"K" THEN PRINT
" Falsche Eingabe!":FOR t=1 TO 800:NEXT:GOTO 232
0
2260 REM

```

```

2270 WIDTH 255:PRINT #8,CHR$(27)"A"CHR$(12)CHR$(1
B)CHR$(27)CHR$(70)CHR$(27)CHR$(72)CHR$(27)"F";:IF
a$="K" THEN PRINT #8,CHR$(15); ELSE IF a$="F" TH
EN PRINT #8,CHR$(27)CHR$(69)CHR$(27)CHR$(71);
2280 PRINT:PRINT "CHR$(24)"KLICK"CHR$(24)" für
Abbruch...":x$=z$:x=zeile:zeile=sz:WHILE zeile<=1
zeile AND JOY(0)<>16 AND INKEY(9)<>0
2290 GOSUB 2000
2300 PRINT #8,z$;CHR$(13);
2310 zeile=zeile+1:WEND
2320 ho=20:GOSUB 1170:wopen=0:GOSUB 240:GOSUB 130
0:WINDOW SWAP 0,3:z$=x$:zeile=x:RETURN
2330 REM Hilfstext anzeigen
2340 WINDOW #0,3,78,3,22
2350 PRINT:PRINT:PRINT" Teamtext Ver. 3.4 "CHR$(1
64)"1985 bei Thomas M.Binzinger"
2360 PRINT:PRINT" Pull-Down Punkte werden durch e
rneutes Anklicken geschlossen.":PRINT" Cursorast
en oder Joystick bewegen Pointer.":PRINT " Ein Kl
ick innerhalb des Textfeldes setzt CURSOR auf die
angegebene":PRINT" Position."
2370 PRINT" Folgende Sonder-Kommandos gibt es (ei
nfach entsprechende Tasten drücken)":PRINT
2380 PRINT" SHIFT und ";CHR$(240)CHR$(241)CHR$(24
2)CHR$(243)" bewegen Cursor direkt im Text,":PRIN
T" CTRL und ";CHR$(240)CHR$(241)CHR$(242)CHR$(243
)" bewegen Cursor Seite höher/tiefer/Zeilenanfang
/-ende":PRINT:PRINT" CTRL und 'L' entfernt Leerze
ichen."
2390 PRINT" CTRL und 'F' formatiert die Zeile, in
der Cursor sich befindet, neu."
2400 PRINT:PRINT" TAB springt auf nächste Tabulat
or-position. Diese wird durch einmaliges":PRINT"
Anklicken des Balkens über dem Schreibfeld gestzt
oder zurückgesetzt.":PRINT" Zweimaliges Anklicke
n setzt den Rand."

```

```

2410 RETURN
2420 REM Einfuegen Zeile
2430 sz=1:GOTO 2460
2440 REM Einfuegen Seite
2450 sz=10:z$=e$:GOSUB 2030
2460 x=zeile:FOR y=lzeile TO x+1 STEP -1
2470 zeile=y:GOSUB 2000:zeile=MIN(y+sz,maxi):GOSU
B 2030:IF zeile>lzeile THEN lzeile=zeile
2480 NEXT:FOR zeile=x+1 TO x+sz:z$=SPACE$(80):GOS
UB 2030:NEXT
2490 zeile=x
2500 zeile=x:GOSUB 1300:wopen=0:GOSUB 240:ho=20:G
OSUB 1170:GOSUB 1300:RETURN
2510 REM Loeschen Zeile
2520 MID$(e$,lrand,rrand-lrand+1)=SPACE$(rrand-lr
and+1):z$=e$:GOSUB 2030
2530 x=zeile:FOR y=x+1 TO lzeile:zeile=y:GOSUB 20
00:zeile=y-1:GOSUB 2030:NEXT:zeile=x:GOSUB 2000:e
$=z$:GOSUB 1300:ho=20:GOSUB 1170:GOSUB 1300:wopen
=0:GOSUB 240:RETURN
2540 REM Leerzeichen aus Zeile entfernen
2550 ab$=MID$(e$,lrand,rrand-lrand+1):IF INSTR(ab
$," ")=0 THEN RETURN
2560 p=INSTR(ab$," ") +1
2570 IF MID$(ab$,p,1)=" " THEN ab$=LEFT$(ab$,p-1)
+RIGHT$(ab$,LEN(ab$)-p):GOTO 2570
2580 p=INSTR(p+1,ab$," "): IF p<>0 THEN p=p+1:GOT
O 2570
2590 IF MID$(ab$,LEN(ab$),1)=" " THEN IF LEN(ab$)
>1 THEN ab$=LEFT$(ab$,LEN(ab$)-1):GOTO 2590
2600 IF LEFT$(ab$,1)=" " THEN ab$=RIGHT$(ab$,LEN(
ab$)-1)
2610 MID$(e$,lrand,rrand-lrand+1)=SPACE$(rrand-lr
and+1):MID$(e$,lrand,rrand-lrand+1)=ab$+SPACE$(rr
and-lrand-LEN(ab$)):RETURN

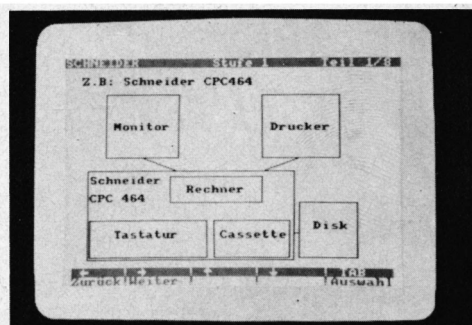
```

Computerkurs für Schneider CPC

BASIC-Kurse gibt es mittlerweile wie Sand am Meer, egal ob auf Kasette, Diskette oder in Buchform und natürlich auch in den Computer-Magazinen. Nur für den Schneider CPC hat Schneider sich einen ganz besonderen Leckerbissen einfallen lassen: DREI IN EINEM könnte man diesen Computerkurs bezeichnen, da er nicht nur einen BASIC-Lehrgang per Bildschirm beinhaltet, sondern auch noch sehr „Lehrreiches“ über den Computer nebst Tastatur. Der gesamte Kurs benötigt zwei Diskettenseiten, nämlich Vorder- und Rückseite.

Startet man den Computerkurs, so erhält man erst einmal wertvolles Informationsmaterial über die Hardware, sogar mit etwas Begleitmusik. Erklärt wird nicht nur CPU, ROM und RAM, sondern auch das, was zu einer Complett-Anlage gehört, also die erweiterte Computerritis. Im Übungsteil, TASTATUR, beginnt der Computer mit seinem Anwender schon fleißig zu arbeiten,

ohne daß es dem Bediener der Tastatur schwer fallen wird. Zwischendurch wird auch noch erklärt, wofür z.B. die Tasten, ESC, CTRL, f-Tasten oder die große und kleine Entertaste sind, bzw. welche Funktionen oder Bedeutung diese haben. Wo sich die durch Software herbeigezauerten Umlaute, ß, usw. befinden, wird selbstverständlich auch gezeigt. Der Abschnitt, Tastaturübung dürfte sich für den Anwender bis zu einem 10-Finger-blind-Schreiben ausdehnen lassen. Also, alles kein Problem. Erst jetzt, nachdem jedem Anwender Rechner, Peripherie und Tastatur in Fleisch und Blut übergegangen sind, geht es zum BASIC-Kurs über. Also der interessante Teil für jenen, der einmal ein Programmierer werden möchte. Ein BASIC-Kurs im Frage- und Antwort-Spiel. Nicht nur Print und Input, sondern auch FOR-NEXT bzw. GOSUB oder on GOTO bzw. on GOSUB usw.. Selbstverständlich werden auch



numerische und alphanumerische Variablen angesprochen und erklärt. So richtig schön rund um die BASIC-Sprache, bei der man nach kurzer Zeit schon in der Lage ist, seine eigenen kleinen Programme zu schreiben. Ergebnisse der Grafik-Programmierung sind z.B. ein Balkendiagramm, welches die Umsatzzahlen der letzten fünf Jahre übersichtlich anzeigt.

Fazit: Computerkurs von Schneider ist schon fast ein „MUSS“ für den Einsteiger, aber für den Fortgeschrittenen auch nicht uninteressant! P.R.



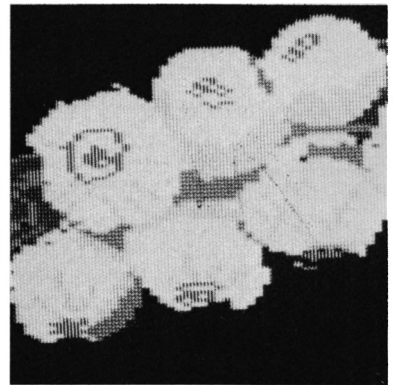
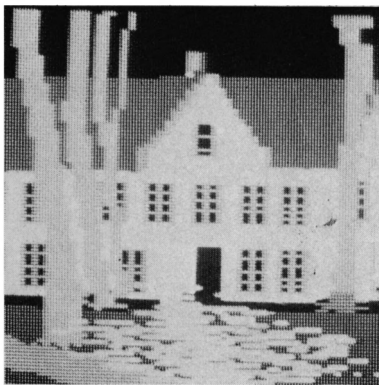
Für Ihren Schneider CPC

AVAILABLE
NOW

Thomas Binzingers Grafik-Test-Adventure

DARK POWERS

Version für Schneider 6128 in Kürze lieferbar



Ein dunkler, nebliger Novemberabend im Jahre des Herrn 1890. Ein einsamer Reiter auf einem engen Hohlweg. In der Ferne die spärlichen Lichter eines kleinen Dorfes. Nur das müde Stampfen des Pferdes ist zu hören. Plötzlich eine Bewegung im Dickicht — zerlumpte, wilde Gestalten stürzen sich auf den Reiter. Der Kampf dauert nicht lange, zu groß ist die Übermacht der Fremden. Der Reiter wird niedergeschlagen, alles wird dunkel um ihn. Als er mit dröhnenden Kopfschmerzen wieder erwacht, befindet er sich in einem hohen, offensichtlich schon lange nicht mehr

betretenen Raum, und er erinnert sich an den letzten Satz des Anführers seiner Fänger: „Befreie uns von dem Dämon, und bringe uns Beweis für seinen Tod — er oder du . . .“. Sicherlich eine ganz gewöhnliche Geschichte, wie sie jeden Tag passiert, und die einen auch nicht besonders interessiert. Außer natürlich, wenn man einen Schneider CPC sein eigen nennt, und wenn man das Grafikadventure DARK POWERS gekauft hat. Dann ist man nämlich selber der betreffende Herr mit den Kopfschmerzen und hat die schaurig schöne Aufgabe vor sich, einen Vampir killen zu müssen.

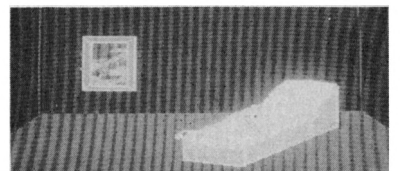
Dark Powers ist ein Grafikadventure. Aber es hebt sich vor allem durch zwei Eigenschaften anderen Programmen dieses Genres gegenüber heraus: Es versteht **DEUTSCH**, d. h. also, Befehle wie **TÖTE VAMPIR** werden ohne Komplikationen verstanden, und es hat eine **TOP-GRAFIK**.

Jeder Raum, in dem man sich befindet (wieviele es genau sind, konnte ich noch nicht herausfinden) wird als Bild in den oberen zwei Dritteln des Bildschirms dargestellt, im letzten Drittel steht dabei der Text. Das Programm versteht sehr viele Worte und hat sogar auf so sinnlose Eingaben wie **ESSE SCHRANK** eine passende Erwiderung bereit.

Im Gegensatz zu manchen anderen Adventures, wo der Spieler tagelang daran herumknobelt, wie er aus dem ersten Raum herauskommen soll, kann man bei Dark Powers überall herumwandern und sich umsehen (und das im wahrsten Sinne des Wortes, da man ja von seiner Umgebung das entsprechende Bild sieht), vorausgesetzt, man hat erst mal den Hauptschlüssel gefunden — was allerdings nicht besonders schwierig ist. Schwierig wird es erst später, z. B. wenn es dunkel wird. Dann wird nämlich

jemand (Sie können sich sicher denken, wer) ungewöhnlich durstig. Auch sollte man sich nicht zuviel Zeit damit lassen, den Vampir zu finden, weil einen nämlich schon am zweiten Spieltag ein Blick in den Spiegel belehrt, daß die eigenen Schneidezähne auch schon länger geworden sind . . .

Und wenn man dann erst mal die diversen Werkzeuge wie Kreuze, Knoblauch, Silberpfähle etc., die ein professioneller Vampirjäger nun mal braucht, zusammen hat, dann scheint der Vampir nicht besonders begeistert von dem Plan zu sein, sich töten zu lassen . . .



Spitzen-Software
aus
Deutschland

Ja, Ihr Angebot hat mich überzeugt
und ich bestelle

Einsenden an:

_____	DARK POWERS CPC 464	79,—	_____
Anzahl		Einzel	Gesamt
_____	DARK POWERS CPC 664	79,—	_____
Anzahl		Einzel	Gesamt
_____	DARK POWERS CPC 6128	79,—	_____
Anzahl		Einzel	Gesamt

SOFTWARE TEAM
Joachim Günster
Mühlenstr. 12
5431 BODEN

Nur als Diskette lieferbar

Versandwunsch bitte angeben:

Verrechnungsscheck beigelegt

Bargeld liegt bei

per Nachnahme

Bei Versand per NN werden 5,— DM Versandkosten pauschal erhoben

so die Koordinaten noch multiplizieren, und zwar mit dem **Radius**, den der Kreis haben soll. Zu guter Letzt muß man noch die Koordinaten des Kreismittelpunkts dazu addieren, damit der Kreis an der richtigen Stelle steht. In Listing 1 wird diese allgemeine Methode gezeigt.

Eine Ellipse ist nichts anderes als ein 'plattgedrückter' Kreis, und sie entsteht, wenn wir die X- und Y-Koordinaten mit einem unterschiedlichen Radius multiplizieren. In Listing 2 wird eine Ellipse gezeichnet, indem Y noch einmal durch 2 dividiert wird, wodurch der Kreis nur halb so hoch wie breit ist. Natürlich kann auch X dementsprechend modifiziert werden, man muß nur darauf achten, daß man nicht etwa die 'fertigen' Koordinaten verändert, sondern bevor der Kreismittelpunkt dazu addiert wird.

Speed-Up...

Das Ergebnis, das die Listings 1 und 2 liefern, ist zwar der erwartete hübsche Kreis bzw. die Ellipse, aber die Programme sind **LAA-ANGSAAAM**. Wie bekommt man sie nun schneller? Die Lösung zeigt das %QQ-Programmlisting: Wir errechnen nicht für alle 360 Winkel die Koordinaten, sondern nur jeden 12ten. Dazu muß allerdings vorher auf Winkelgradmaß umgeschaltet werden (DEG, probieren Sie das auch einmal bei den ersten beiden Beispielen aus), denn wir müssen die Punkte ja auch noch durch Linien miteinander verbinden. Das ist aber nur möglich, wenn die Punkte auch entlang einer Kreisbahn gezeichnet werden und nicht in der scheinbar wirren Reihenfolge wie bei Bogenmaß (RAD, Standard). Und auch wichtig: Vorher muß der Grafikcursor an die richtige Stelle gesetzt werden, damit die erste Verbindungslinie nicht einfach aus irgendeiner Bildschirmcke in den Kreis kommt (also den Grafikcursor an die Stelle setzen, die sich bei $\cos(0)$ und $\sin(0)$ ergibt: $\text{mittex} + \text{radius}$, mittey)!

Durch diese Technik wird der Kreis so schnell gezeichnet, daß man schon fast glaubt, man habe es mit einem Maschinenprogramm zu tun. Die Schrittweite für den Winkel kann natürlich vergrößert (dann geht es noch schneller) oder verkleinert werden, aber bei einer zu großen Schrittweite wird deutlich,

daß man eigentlich gar keinen Kreis mehr zeichnet, sondern ein Vieleck.

Die ersten Aufgaben

Sie sollten auf jeden Fall versuchen, die Technik des Kreise-Zeichnens zu verstehen, schon al-

lein weil wir sie in der nächsten Folge wieder brauchen werden. Um Ihnen dabei zu helfen (und um die 'alten' Sachen zu wiederholen), haben wir hier ein paar Testaufgaben abgedruckt. Versuchen Sie bitte, diese bis zum nächsten Monat zu lösen; schlagen Sie in den alten Ausgaben nach, wenn Sie irgendwelche Befehle vergessen haben. Ansonsten... viel Spaß beim Computern! Ihr COMPUTER TEAM.

```
10 REM Listing 1 - Kreis zeichnen
20 MODE 1
30 mittex=320:mittey=200:radius=100
40 FOR winkel=0 TO 360
50 x=COS(winkel)
60 y=SIN(winkel)
70 x=x*radius
80 y=y*radius
90 x=x+mittex
100 y=y+mittey
110 PLOT x,y,1
120 NEXT winkel
```

```
10 REM Listing 2 - Ellipse zeichnen
20 MODE 1
30 mittex=320:mittey=200:radius=100
40 FOR winkel=0 TO 360
50 x=COS(winkel)
60 y=SIN(winkel)
70 x=x*radius
80 y=y*(radius/2) ' <--- Das formt die Ellipse
90 x=x+mittex
100 y=y+mittey
110 PLOT x,y,1
120 NEXT winkel
```

```
10 REM Listing 3 - Schnelles Kreisprogramm
20 MODE 1
25 DEG:REM Auf Winkelgradmass umschalten!
30 mittex=320:mittey=200:radius=100
35 PLOT mittex+radius,mittey
40 FOR winkel=0 TO 360 STEP 12
50 x=COS(winkel)
60 y=SIN(winkel)
70 x=x*radius
80 y=y*radius
90 x=x+mittex
100 y=y+mittey
110 DRAW x,y,1 ' <-- nun eine Linie ziehen
120 NEXT winkel
```

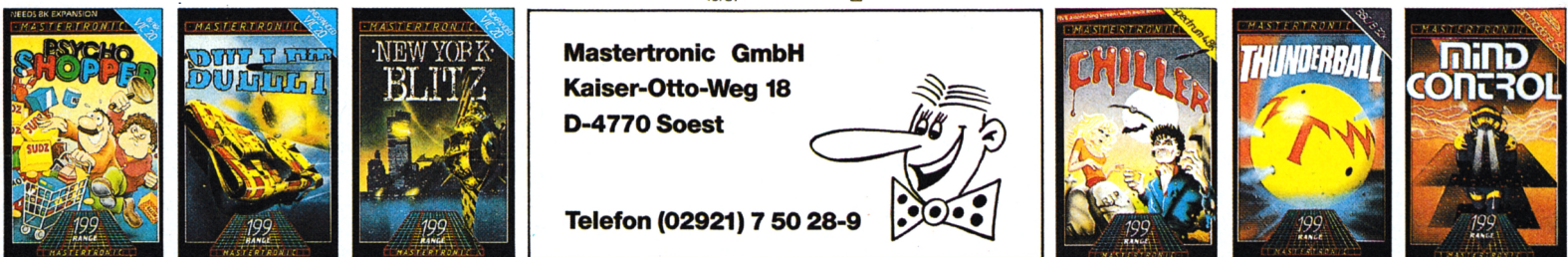
MASTERTRONIC COMPUTER GAMES



DUCK SHOOT 1 **VEGAS JACKPOT** 2 **SQUIRM** 3 **BULLSEYE/TANK TRAX** 5/8 **BIONIC GRANNY JUNGLE STORY** 9/21 **SPECTIPEDE GNASHER** 10/16 **SPACE WALK** 6 **BMX RACERS (48K)** 7 **PHANTOM ATTACK** 11



3D MAZE/STAR MAZE 2 12 **ORBITRON** 14 **SPACE SCRAMBLE** 19 **SPECTIPEDE** 21 **WHODUNNIT ALCATRAZ HARRY** 15/37 **MUNCHMANIA** 22 **HEKTIK** 23 **SUB HUNT** 26 **RIFLE RANGE** 28



PSYCHO SHOPPER 31 **BULLET** 33 **NEW YORK BLITZ** 34 **CHILLER** 36 **THUNDERBALL** 38 **MIND CONTROL** 40

Mastertronic GmbH
Kaiser-Otto-Weg 18
D-4770 Soest



Telefon (02921) 7 50 28-9



DARK STAR 41 **MAGIC CARPET** 42 **CHALLENGER** 44 **VIPER 3** 45 **STAR RACE** 46 **CITY FIGHTER** 48 **APOLLO II** 49 **CHUCKMAN** 51 **WIZARD'S WARRIOR** 52

NEW NEW NEW NEW NEW NEW NEW NEW NEW NEW NEW NEW NEW NEW NEW NEW NEW NEW NEW NEW



FINDERS KEEPERS 53 **FORMULA ONE SIMULATOR** 58 **DOODLE BUG** 59 **1985** 60 **BIG MAC THE MAD MAINTENANCE MAN** 57 **SPOOKS** 69 **SKYJET** 66 **R.I.P. THE GAME** 67 **ROCKMAN** 68

- GAMES FOR YOUR COMPUTER**
- COMMODORE 64
 - CBM 64 DISKETTE
 - COMMODORE 16
 - MSX
 - SPECTRUM
 - VIC 20
 - BBC
 - DRAGON



MAKE MUSIC WITH MISTERTRONIC 61

TYPE-ROPE WITH MISTERTRONIC 66

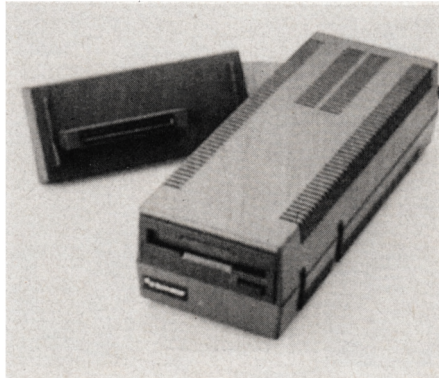
Insider Team — Schneider für Profis

Von T. BINZINGER

Verborgene Diskettenkommandos

Im Disketten-ROM steckt mehr, als das Handbuch verrät. Es gibt nämlich noch 9 zusätzliche Befehle, die allerdings aus dem BASIC nicht nutzbar sind. Aber Assembler-Programmierer können damit sehr interessante Sachen anstellen, z.B. relative Dateien aufbauen oder ganz neue Diskettenformate konstruieren.

Wenn man an den 464 eine Diskettenstation anschließt, bekommt man auch die notwendige Software in einem ROM dazu, dieselbe Software, die auch der 664 verwendet. Diese Diskettensoftware steckt in einem sogenannten Background-ROM. Nach dem Einschalten untersuchen alle Schneider-Computer, oder besser: deren Betriebssystem, ob zusätzliche ROMs an den Computer angeschlossen sind. Handelt es sich um ein Vordergrund-ROM, (das erste Byte im ROM kennzeichnet, um welchen Typ es sich handelt), wird ihm die Kontrolle übergeben. Das ist zum Beispiel bei BASIC der Fall. Background-ROM wird lediglich die Kontrolle zum Initialisieren ihres benötigten RAM-Speichers übergeben, und es wird erwartet, daß sie danach wieder die Kontrolle ans Betriebssystem übergeben. Der Aufbau jedes ROMs entspricht in etwa dem der hinlänglich bekannten RSX-Kommandos, die man aus BASIC mit dem SHIFT-Klammeraffen aufrufen kann, und so besteht auch die Möglichkeit, ROM-Routinen so aufzurufen (versuchen Sie einmal !BASIC). Beispiele hierfür sind alle zusätzlichen Disc-Kommandos wie !DIR. So weit, so gut. Bis jetzt war alles nur 'Wiederholung'. Interessant wird es aber, wenn man sich den Anfang des Disc-ROMs ansieht: Dort stehen nämlich mehr Sprünge, als im Handbuch Disc-Befehle beschrieben werden. Logische Folgerung: Es muß noch mehr Disc-Kommandos geben!



Low-Level-Befehle ...

Auf Adresse C072 stehen die normalen Befehlsnamen, wie man sie auch erwartet (bis auf den Befehl CPM-ROM, den man aus BASIC auch nicht aufrufen kann), aber dann ... dann folgen 9 Zahlen von #81 bis #89. Da das Ende jedes Befehls durch ein gesetztes Bit 7 im Namen gekennzeichnet ist, handelt es sich um die Befehle 1, 2, 3 ... 9, die jeweils ein Zeichen lang sind. Da die Zahlen 1 ... 9 aber keine ASCII-Zeichen sind, kann man sie aus BASIC auch nicht aufrufen (ebenso wie Namen, die Leerzeichen enthalten). Die nächste Frage, die wir uns nun stellen, lautet natürlich: Was tun sie?

Nun, nach langer Forschungsarbeit ist das in Tabelle 1 wiedergegebene (hoffentlich richtige) Ergebnis herausgekommen. Wie man sieht, handelt es sich ausschließlich um Low-Level-Befehle wie Sektor — lesen / schreiben / formatieren etc. Dadurch wird es möglich, eigene Anwenderprogramme mit solchen Routinen wie >Disketten formatieren< auszustatten und so besonders anwenderfreundlich zu machen. Man kann aber auch solche Scherze, wie z.B. eine Diskette in mehreren Formaten formatieren etc., damit treiben und so seine Programme schützen. Mit genug Phantasie tun sich hier ungeahnte Möglichkeiten auf. Beschäftigen wir uns nun mit den Befehlen im einzelnen (Listing 1 zeigt Ihnen, wie man die Kommandos aus einem Assemblerprogramm heraus aufrufen muß), aber denken Sie beim Herumprobieren daran, daß ein Write (oder Format)-Sektor mit irgendwelchen Nonsense-Daten Ihrer Lieblings-Diskette vielleicht nicht so gut bekommt!

Die Kommandos im einzelnen.

#81 dient nur dazu, die Disc-Meldungen ein- oder auszuschal-

Tabelle 1:

#81	Discmeldungen ein/aus
#82	Discparameter festsetzen
#83	Format festsetzen
#84	Sektor lesen
#85	Sektor schreiben
#86	Spur formatieren
#87	Kopf positionieren
#88	Laufwerk Nr (A) einschalten
#89	Leseversuch-Anzahl festsetzen

```

10 ; Listing 1
20
A000 30      org  #a000
40 ;
50 ; ROMCAL ruft die Disc-Kommandos auf:
60 ; Nummer des gewünschten Kommandos ins
70 ; B-Register!!
80 ;
90 ;
A000 E5      100 romcal: push hl
A001 C5      110      push bc
A002 D5      120      push de
A003 211EA0  130      ld  hl,zw1          ;Namen...
A006 70      140      ld  (hl),b          ;...ablegen und
A007 CDD4BC  150      call #bcd4         ;Adresse ermitteln
A00A 300E    160      jr   nc,ready2      ;Fehler->zurueck
A00C 221FA0  170      ld  (zw2),hl       ;Adresse->zw2
A00F 2121A0  180      ld  hl,zw3         ;und deren Romselect
A012 71      190      ld  (hl),c         ;nach zw3
A013 D1      200      pop  de
A014 C1      210      pop  bc
A015 E1      220      pop  hl          ;alten Wert von HL holen
A016 DF      230      rst  #18         ;Far-Adress aufrufen
A017 1FA0    240      defw zw2
A019 C9      250 ready: ret
A01A D1      260 ready2: pop de
A01B C1      270      pop  bc
A01C E1      280      pop  hl
A01D C9      290      ret
A01E 00      300 zw1:  nop             ;enthalt 1Byte Name
A01F 0000    310 zw2:  defw 0          ;Platz fuer Adresse
A021 00      320 zw3:  nop             ;Platz fuer rom-Auswahl
A022                330      end

```

ten. Ein POKE an der richtigen Speicherstelle (#be 78) ist hier einfacher. A muß mit dem gewünschten Wert geladen werden, #FF=aus, ansonsten ein. Beim Rücksprung liefert die Routine den alten Wert in L.

#82 erlaubt es, die Laufwerkdaten festzulegen. Der Anfang der Tabelle, in der diese stehen, muß

Laufwerkdaten

im HL-Register übergeben werden. Die Tabelle setzt sich folgendermaßen zusammen: 2 Bytes, für Wartezeit vor dem ersten Zugriff nach Einschalten des Motors. 50 entspricht dabei einer Sekunde. Dann folgen wieder 2 Bytes, sie ge-

ben die Wartedauer vor dem Abschalten des Motors an. Wenn dieser groß ist, kann der Diskettenzugriff evtl. beschleunigt werden, da der Motor bei kleineren Pausen nicht jedesmal ausgeht. Anschließend folgen noch Werte, die interne Controllerzeiten bestimmen. Es sollten folgende Werte folgen (in dezimal): 175, 15, 12, 7, 64. Die Tabelle steht übrigens an Adresse #be44.

#83 legt das Format fest, mit dem

IBM-Format

gearbeitet werden soll. Es muß in A übergeben werden und muß lauten: #0:IBM-Format, #40:CPM-Format, #C0:AMSDOS(BASIC/Daten-Format).

#84 liest einen Sektor, ist damit wohl einer der interessanteren Befehle. Die Anwendung ist ganz einfach: Laufwerk ins E-Register (0 oder 1), Sektor nach C (es sind 9 Sektoren auf einem Track, also eine Nummer von 1-9. ABER ACHTUNG: Die bei #83 angegebenen Werte müssen dazu addiert werden, also hat der erste Sektor auf einer CP/M-Diskette die Nummer #41) und natürlich der Track 0-39 ins D-Register. Nun noch HL mit der Adresse laden, wo die 512 Bytes hin sollen und dann aufrufen (s. Listing 1).

#85 schreibt einen Sektor auf Diskette, ansonsten alle Daten wie bei #84 (HL enthält die Adresse, an die 512 Bytes gehen).

```

10 ; Listing 2
20 ; liest alle Sektoren einer Diskette und zeigt Inhalt an
30 ;
9DFE 40      org  #a000-514          ;Platz fuer Buffer
9DFE 50      buf:  defs 514          ;Platz fuer Daten
A000 60      ent  $                  ;Startadresse hier
A000 0600    70      ld  b,0          ;von Track 0 bis 39
A002 0E41    80      ld  c,#41       ;Track 1 fuer CPM (!!) Disk
A004 C5      90      ll:  push bc
A005 1E00    100     ld  e,0          ;Laufwerk A
A007 50      110     ld  d,b          ;Tracknummer nach D
A008 21FE9D  120     ld  hl,buf       ;Buffer
A00B 0684    130     ld  b,#84       ;Read Sektor
A00D CD34A0  140     call romcal      ;ausfuehren
A010 21FE9D  150     ld  hl,buf       ;Daten nun ausgeben
A013 010002  160     ld  bc,512      ;512 Bytes
A016 E5      170 12:  push hl
A017 C5      180     push bc

```

```

A018 7E      190      ld  a,(hl)          ;Zeichen holen und
A019 CD5DBB  200      call #bb5d         ;anzeigen
A01C C1      210      pop  bc
A01D E1      220      pop  hl
A01E 23      230      inc  hl
A01F 0B      240      dec  bc            ;und weiter,bis alle 512...
A020 78      250      ld  a,b
A021 B1      260      or   c
A022 20F2    270      jr   nz,12
A024 C1      280      pop  bc            ;Track und Sektor
A025 0C      290      inc  c            ;naechsten Sektor
A026 79      300      ld  a,c            ;ist es Nummer 10?
A027 FE4A    310      cp  #4a           ;wenn nein,->lesen
A029 38D9    320      jr   c,11
A02B 0E41    330      ld  c,#41         ;Sektor 1,Track=Track+1
A02D 04      340      inc  b
A02E 78      350      ld  a,b            ;Track=40
A02F FE2B    360      cp  40            ;ja,fertig
A031 C8      370      ret  z
A032 18D0    380      jr   11           ;sonst anzeigen
          390 ;
A034 E5      400 romcal: push hl
A035 C5      410      push bc
A036 D5      420      push de
          430 ;
          440 ; WEITER: Siehe LISTING 1 !
          450 ;
A037          460      end

```

Sektor schreiben

#86 formatiert einen der 40 Tracks (Spuren) auf der Diskette, also 9 Sektoren. Track nach D, E=Laufwerknummer, C=das entsprechende Sektoroffset (siehe #83) plus 1 (also #41 für CP/M-Format), HL muß auf einem Datenblock folgende Formate zeigen: 1 Byte Tracknummer wie in D, 0, Sektornummer mit Offset, Sektorgrößen in 128 Byte, also 2. Und das für alle 9 Sektoren!

Listing 2

#87 bewegt den Kopf auf die angegebene Spur: Laufwerk ins E-Register, Track ins D-Register.

#88 schaltet Laufwerk mit Nr. in A ein und gibt Inhalt von +be4c zurück?

#89 setzt die Nummer der Leseversuche fest. Die Nummer muß in A stehen und ist standardmäßig 10.

Nun, der Rest ist Ihre Sache — was Sie mit diesen neuen Befehlen anfangen. Auf jeden Fall viel Spaß dabei, Ihr COMPUTER TEAM.

Pass 2 errors: 00

Table used: 54 from 211

Executes: 40960

Listschutz ade!

von BEATE LANG

Ein mit SAVE „NAME“, P abgespeichertes Programm läßt sich nicht mehr ohne weiteres listen. Hat man versäumt, eine ungeschützte Kopie herzustellen, so ist das sehr ärgerlich. Mit folgendem kleinen Programm läßt sich jedoch der Listschutz beheben:

Dieses Programm muß vor dem geschützten Programm geladen und gestartet werden. Danach kann das geschützte Programm z.B. mit LOAD „NAME“ geladen und wie jedes andere Programm gelistet werden.

Ob ein Programm geschützt ist

oder nicht, wird allein von der Speicherstelle &AE45 bestimmt.

Bei jedem READY-Durchlauf wird unter anderem der Inhalt dieser Speicherstelle geprüft. Ist er Null, so handelt es sich um ein ungeschütztes Programm, sonst ist es geschützt, und alle Variablen sowie das Programm selbst werden gelöscht.

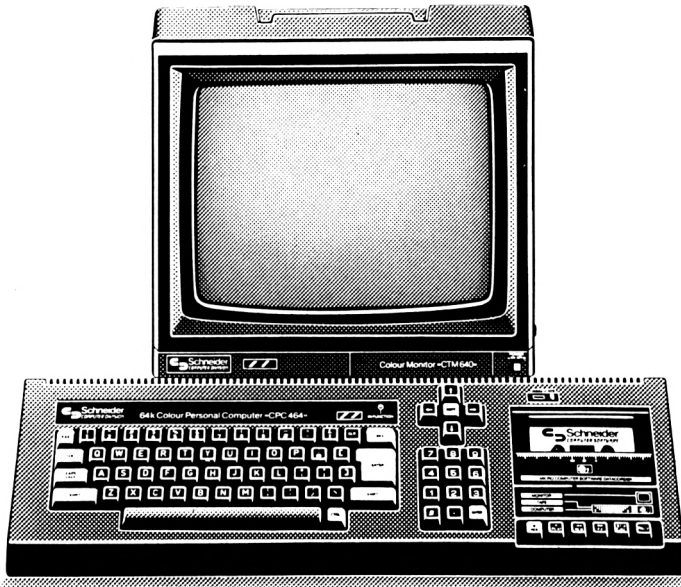
Obiges Programm nutzt die Tatsache, daß vor Ablauf der READY-Routine zur RAM-Adresse &AC01 gesprungen wird, wo normalerweise nur ein RET-Befehl steht. Hier wird nun ein Sprung zu einer Routine eingetragen, die in die Speicherstelle &AE45 eine Null (= ungeschützt) lädt. Diese Routine steht ab Adresse &a600 im Speicher.

```

10 FOR i=&A600 TO &A604
20 READ a:POKE i,a:NEXT
30 FOR i=&AC01 TO &AC03
40 READ a:POKE i,a:NEXT
50 DATA %97,%32,%45,%ae,%c9
60 DATA %c3,%00,%a6

```

Die Unterschiede zwischen CPC 464 und CPC 664 Teil 2



Schneider »CPC 464«
mit Color-Monitor



Schneider »CPC 664«
mit Color-Monitor

Beim letzten Mal haben wir gezeigt, daß der Hauptunterschied zwischen den beiden Schneider-Computern CPC 464 und CPC 664 hauptsächlich in einem geänderten Betriebssystem und in einem geänderten BASIC zu suchen ist. In diesem zweiten (und letzten) Teil werden wir mehr ins Detail gehen und zeigen, welche Unterprogramme und Speicheradressen sich geändert haben.

Beginnen wir mit dem Offensichtlichsten: dem geänderten BASIC. Wie allgemein bekannt, hat der CPC 664 einige neue Befehle dazu bekommen, die natürlich auch Platz im BASIC-ROM beanspruchen. Dadurch hat sich die Lage der einzelnen BASIC-Routinen geändert, und Programme, die direkt

cherplatz zu sparen und weil so eine höhere Geschwindigkeit erzielt werden kann. Eine solche Zahl nennt man TOKEN.) Nun, des Rätsels Lösung: Der 464 nutzte gar nicht alle Werte von +80 — +FF, denn so viele Befehle waren nicht vorhanden, vor allem da es für die gesamten Funktionen (also Befehle, die eine Zahl oder einen String zurückliefern) eine extra Tabelle gibt, so daß ein Funktions-Token aus zwei Bytes besteht: +FF zur Kennzeich-

So können Programme unterscheiden, auf welchem der beiden Computer sie gerade laufen:

```
10 a=peek(&BBFA)
20 if a=54 then print "Das ist ein 464!"
30 if a=162 then print "Es ist ein 664!"
```

nung, daß nun ein Funktionstoken folgt und dann erst das eigentliche Token. So blieben einige Werte zwischen +DB und +E3 ungenutzt, die der 664 jetzt verwenden kann. Tabelle Nr. 2 zeigt die neuen Tokens. Übrigens: Wenn ein 464 in einem Programm auf ein solches 664 Token stößt, meldet er einen Syntax-Error, und Zeilen, die ein 664 Token enthalten, können auf einem 464 nicht geLISTet werden. So viel zum BASIC.

Was ist aber nun mit dem Betriebssystem passiert? Wie bereits letzten Monat erwähnt, wurde auch dieses geändert. Durch Änderungen bei den Grafikroutinen haben sich die folgenden Routinen verschoben und liegen an anderen Stellen. Auch wurde der Code optimiert, und so ist z.B. die Routine KL EVENT ein Byte kürzer geworden. Da aber alle Betriebssystem-Funk-

BASIC-Routinen

Unterprogramme aus dem BASIC von &c000-&ffff verwenden, rufen beim 664 eine falsche Adresse auf, was die unvorhergesehensten Ergebnisse produzieren kann. Die Adressen der BASIC-Befehle bei beiden Computern finden Sie in Tabelle Nr. 3. Bei den neuen Befehlen stellt sich die Frage, woher die Amstrad-Programmierer die zugehörigen Tokens bekommen haben. (Der Computer speichert BASIC-Befehle nicht als einzelne Buchstaben, sondern als Zahl ab, um Spei-

Tabelle 2: Die neuen 664-Tokens.

Wert	Token
DD	FILL
DE	GRAPHICS
DF	MASK
E0	FRAME
E1	CURSOR

neue Funktionen: (vor dem Token steht bei einer Funktion FF)

49	DERR
7E	COPYCHR\$

alle Zahlen in hexadezimal

Tabelle 1: Die wichtigsten Basicadressen beim 464/664

Adresse bei Cpc 464 / Cpc 664	Bedeutung der Adresse
AB80-ABFF	AB80-ABFF Zeichenmatrix CHR\$(240)-CHR\$(255)
AC00	AC00 Flag:Blanks ignorieren
AC1C	AC01 Flag:Auto Mouds
AC1D-AC1E	AC02-AC03 Zeilennummer für AUTO
AC1F-AC20	AC04-AC05 Zeilenplus für AUTO
AC21	AC06 I/O Out-Nummer
AC22	AC07 I/O In-Nummer
AC23	AC08 Position des Druckkopfs
AC24	AC09 Wert von WIDTH
AC34-AC35	AC19-AC1A Adresse von ON BREAK
ACA4-ADA3	AC8A-AD89 Eingabepuffer
ADAA	AD8D Letzte Fehlernummer
AE45	AE2C Flag:Geschütztes Programm
AE7B-AE7C	AE5E-AE5F HIMEM
AE7D-AE7E	AE60-AE61 Ende freies Ram
AE7F-AE80	AE62-AE63 Anfang freies Ram
AEB1-AEB2	AE64-AE65 Startadresse des Basicprogramms
AE83-AE84	AE66-AE67 Endadresse des Basicprogramms
AE85-AE86	AE68-AE69 Beginn Variablenbereich

tionen über die Vektoren (spezielle Sprungbefehle, die immer an der gleichen Stelle stehen) aufgerufen werden, spielt es im Endeffekt keine Rolle, was wo im ROM liegt, Hauptsache, die Vektoren stimmen. Für uns sind die vom Betriebssystem genutzten RAM-Adressen wesentlich interessanter, da sich hier die Möglichkeit bietet, direkt vom BASIC — mit ein paar simplen Pokes — Einfluß auf solche Funktionen wie Kassettenschreibgeschwindigkeit oder Grafikhintergrund zu nehmen. In Tabelle Nr. 4 sind die wichtigsten Adressen für beide Computer aufgeführt. Viel Spaß beim „POKEN“ wünscht Ihnen Ihr
COMPUTER TEAM.

Tabelle Nr.3

Die Adressen der Routinen der Basicbefehle. Beim 464 steht die Tabelle, in der diese Adressen nach aufsteigenden Tokens sortiert sind, von #DE01-#DEB9, beim 664 von #DEE5-#DFAB.

464-Adresse	664-Adresse	Befehlsroutine von...	Token	464-Adresse	664-Adresse	Befehlsroutine von...	Token
C971	CA25	AFTER	80	C12B	C128	NEW	B1
C0DF	C0EA	AUTO	81	C7E3	C885	ON	B2
C221	C24B	BORDER	82	C8CB	C979	ON BREAK	B3
F1BA	F261	CALL	83	CBFB	CCCD	ON ERROR GOTO	B4
D246	D299	CAT	84	C940	C9F8	ON SQ	B5
EA3C	EB02	CHAIN	85	D25F	D2B7	OPENIN	B6
C132	C12F	CLEAR	86	D256	D2AB	OPENOUT	B7
C4B5	C509	CLG	87	C48C	C4E1	ORIGIN	B8
D298	D2F0	CLOSEIN	88	F177	F228	OUT	B9
D2A1	D2F8	CLOSEOUT	89	C20A	C23C	PAPER	BA
C25A	C283	CLS	8A	C212	C227	FEN	BB
CB0C	CC96	CONT	8B	C4D0	C546	PLOT	BC
E8EF	E9A8	DATA	8C	C4D5	C54B	PLOTR	BD
D117	D174	DEF	8D	F15F	F214	POKE	BE
D618	D657	DEFINT	8E	F1FD	F2A9	PRINT	BF
D61C	D65B	DEFREAL	8F	E8F3	E9AC	'	CO
D614	D653	DEFSTR	90	D4EB	D530	RAD	C1
D4E7	D52C	DEG	91	D559	D59C	RANDOMIZE	C2
E728	E7F3	DELETE	92	DCEB	DCDF	READ	C3
D67D	D6B9	DIM	93	D31E	D373	RELEASE	C4
C4C6	C53C	DRAW	94	E8F3	E9AC	REM	C5
C4CB	C541	DRAWR	95	E7DF	E8A3	RENUM	C6
C052	C046	EDIT	96	DCD9	DCCD	RESTORE	C7
E8F3	E9B2	ELSE	97	CC03	CCD8	RESUME	C8
CB65	CC34	END	98	C70F	C7B3	RETURN	C9
D385	D3D7	ENT	99	E9BD	EA7D	RUN	CA
D34E	D3A1	ENV	9A	EC09	ECE1	SAVE	CB
D9C0	D9F4	ERASE	9B	D2C0	D316	SOUND	CC
CABF	CB54	ERROR	9C	D494	D4DE	SPEED	CD
C979	CA2D	EVERY	9D	CB5A	CC29	STOP	CE
C529	C5D7	FOR	9E	F69D	F784	SYMBOL	CF
C6ED	C78F	GOSUB	9F	C319	C346	TAG	D0
C6E8	C789	GOTO	A0	C320	C34D	TAG OFF	D1
C6C7	C76A	IF	A1	DDE6	DECA	TRON	D2
C22A	C254	INK	A2	DDE2	DEC6	TROFF	D3
DB2B	DB48	INFUT	A3	F17D	F22E	WAIT	D4
D439	D489	KEY	A4	C776	C81D	WEND	D5
D654	D691	LET	A5	C747	C7EA	WHILE	D6
DAF8	DB18	LINE	A6	C3E3	C42D	WIDTH	D7
E0F7	E1D2	LIST	A7	C2E1	C311	WINDOW	D8
E9F6	EABA	LOAD	A8	F47B	F50D	ZONE	D9
C2D2	C302	LOCATE	A9	F1F6	F2A2	WRITE	DA
F4EF	F570	MEMORY	AA	C8E1	C99A	DI	DB
EAA6	EB59	MERGE	AB	C8E7	C9A0	EI	DC
F993	FA07	MID\$	AC	n.v.	C515	FILL	DD
C24F	C278	MODE	AD	n.v.	C59D	GRAFIK	DE
C505	C532	MOVE	AE	n.v.	C5C3	MASK	DF
C50A	C537	MOVER	AF	n.v.	BD19	FRAME	E0
C5FB	C6A5	NEXT	B0	n.v.	C363	CURSOR	E1

Tabelle Nr.4: Die wichtigsten Betriebssystem-RAM-Adressen bei beiden Computern.

Adresse bei 464	664	und ihre Bedeutung	Adresse bei 464	664	und ihre Bedeutung
b1c8	b7c3	aktueller MODE	b32a	b695	Grafik Y-Origin
b1ca	b7c5	Bildschirmadresse	b32c	b697	Grafikcursor X-Koordinate
b1cb	b7c6	High-Byte Bildschirmadresse	b32e	b699	Grafikcursor Y-Koordinate
b1cf	b7ca	Bit-Masken für Punkte:Abhängig von Mode	b330	b69b	Grafikfensterposition links
			b332	b69d	" " rechts
b1d7	b7d2	Blinkzeiten für Farben	b334	b69f	" " oben
b288	b729	akt. Textfenster oben	b336	b6a1	" " unten
b289	b72a	akt. Textfenster links	b338	b6a3	Grafikfarbstift
b28a	b72b	akt. Textfenster unten	b339	b6a4	Kodierte Farbe für Grafik-Paper
b28b	b72c	akt. Textfenster rechts	b4e7	b631	Status SHIFT-Lock
b28d	b72e	Cursor Flag	b4e8	b632	Status CAPS-Lock
b28f	b72f	aktuelle Pen-Farbe	b800	b118	Flag:Kassettenmeldungen an/aus
b290	b730	Farbe für Paper	b8d1	b1e9	Kassettschreibgeschwindigkeit
b328	b693	Grafik X-Origin	b8d2	b1ea	" "

Neues vom CPC 6128

von THOMAS BARNDT

In der Ausgabe 10/85 haben wir bereits die wichtigsten Neuerungen des neuen Schneiders angesprochen. Wir wollen in dieser Ausgabe ein wenig mehr ins Detail gehen. Vor allem die folgenden drei Themen wollen wir hier besonders ansprechen:

- neue Möglichkeiten mit CP/M plus
- Die Erweiterung des Dr. LOGO
- Anwendung des zusätzlichen Speicherplatzes

CP/M plus

Es soll nicht Aufgabe des Artikels sein, das gesamte CP/M plus des CPC 6128 zu erklären. Es sind daher nur einige der neuen Dienstprogramme ausschnittsweise beschrieben. Insgesamt sind über dreißig neue .COM-Dateien auf den mitgelieferten Disketten (im Vergleich mit CPC 464). Allerdings fehlen dafür auch ein paar von den alten. Folgende Dienstprogramme sind neu oder erweitert und sollen an dieser Stelle erklärt werden: SETDEF.COM / DEVICE.COM / LANGUAGE.COM / SHOW.COM / DATE.COM / SET.COM / SETLST.COM
SETDEF [PAGE] bewirkt das An-



halten der Bildschirmausgabe, wenn der Bildschirm voll ist. Das bedeutet, daß jedesmal, wenn der Bildschirm beschrieben ist, in der letzten Zeile die Meldung „Press RETURN to continue“ erscheint und der Computer erst nach dem Drücken der Return- oder Entertaste mit der Ausgabe fortfährt. Dies

ist zum Beispiel sehr nützlich, wenn man mit den HELP-Texten arbeitet. Außerdem kann mit SETDEF bestimmt werden, auf welchen Diskettenlaufwerken der Computer be stimmte zu spezifizierende Files zu suchen hat. So kann zum Beispiel veranlaßt werden, daß alle .COM-Dateien zuerst auf Laufwerk C ge-

sucht werden und danach erst auf dem Standardlaufwerk. Dabei können bis zu vier Laufwerke angesprochen werden.

Der Datentransfer zwischen den Ein- und Ausgabegeräten kann mit DEVICE manipuliert werden. Hiermit können z.B. Baudraten eingestellt werden. DEVICE CONSOLE [COLUMNS= 40] stellt die Bildschirmausgabe auf 40 Spalten ein.

Mit dem Befehl LANGUAGE kann einer von acht internationalen Zeichensätzen angewählt werden. LANGUAGE 2 belegt die Klammern mit deutschen Umlauten, den Klammeraffen (§) mit dem Paragraphenzeichen und die Tilde mit dem scharfen S. Der deutsche Zeichensatz der meisten Drucker liegt auf demselben Zeichen.

SHOW bewirkt die Anzeige von Informationen zu den Diskettenlaufwerken. So können z.B. die Anzahl der freien Directoryeinträge oder andere Speicherkapazitäten angezeigt werden.

Über dem DATE-Befehl kann eine implementierte Uhr angespro-

DATE-Befehl

chen werden. Sie wird mit DATE SET eingestellt und zeigt nach Eingabe von DATE C eine laufende Uhr mit Datum an.

SET arbeitet ähnlich wie STAT, kann jedoch zusätzlich Files und Disketten mit Passwörtern und Datum versehen („stempeln“). Dazu muß das Directory der Diskette mit INITDIR formatiert werden. Leider wird dadurch der Diskettenaustausch zwischen CP/M plus und AMSDOS bzw. CP/M 2.2 erschwert.

Mit SETLST kann ein Drucker initialisiert werden. Zu diesem Zweck muß ein File erstellt werden, welches die nötigen Fluchtsequenzen für den Drucker enthält.

Das war's also zu den Dienstprogrammen. Eine Sache hat mich etwas geärgert: Das Programm FILECOPY existiert nur noch unter CP/M 2.2 auf Seite vier der mitgelieferten Disketten. Da dieses Programm tatsächlich nur unter CP/M 2.2 läuft, mußte ich, um einzelne Dateien von Diskette auf Diskette zu kopieren, erst CP/M 2.2 booten. Beim Versuch, die Help-Texte auf eine andere Diskette zu kopieren, behauptete das Programm gar dreist, ich hätte die falsche Ziel-Diskette eingelegt, und brach den Kopiervorgang ab.

Einen Lerneffekt hat die Tatsache, daß die Help-Texte in Englisch verfaßt sind. Dadurch müssen nämlich alle diejenigen, die in dieser Sprache nicht so versiert sind, sehr viele Fachbegriffe nachschlagen.

Noch etwas Positives: Durch Drücken von [CONTROL] W wird die letzte Befehlseingabe wiederholt. Bei Tippfehlern braucht man nun nicht mehr die gesamte Eingabe, sondern nur noch die letzte zu wiederholen und zu berichtigen.

Dr. LOGO

Die neue LOGO-Version des CPC 6128 enthält viele Befehle, die beim CPC 464 noch nicht vertreten waren. Diese sind vernünftigerweise im Handbuch mit einem Sternchen versehen, so daß man als Besitzer eines CPC 6128 wenigstens die Möglichkeit besitzt, kompatible Programme zu erstellen.

Die Diskettenoperationen sind durch sechs Befehle erweitert worden. Leider befindet sich unter diesen noch immer keiner zum Löschen von Dateien. Denn wenn man an einem Programm arbeitet, ist es sinnvoll, dieses zwischen durch mehrmals abzuspeichern. Dies kann man bei Dr. LOGO jedoch nicht unter gleichem Namen tun, da man dadurch eine Fehlermeldung des Systems erhält. Mit dem Befehl CHANGEF ist es jedoch nun möglich, Filenamen in der Directory zu ändern.

Nachteilig war bei der bisherigen LOGO-Version, daß man nicht alle Prozeduren gleichzeitig editieren konnte, wie dies bei einem Pascal-Programm z.B. selbstverständlich ist. Man mußte immer erst die Änderung einer Prozedur ab-

ED ALL

brechen und konnte dann erst die nächste aufrufen. Mit dem Befehl EDALL ist dieses nun behoben. Der Befehl TRACE erlaubt das Verfolgen einer Prozedur während ihrer Ausführung. Mit MEMBERP und WHERE kann wie bei INSTR im BASIC ein String auf seinen Inhalt hin überprüft werden. PIECE und ITEM sind mit MID\$ vergleichbar. Mit COPYON und COPYOFF kann die Ausgabe auf einen angeschlossenen Drucker ein- bzw. ausgeschaltet werden. Auch das Ansprechen des Ports ist nun mit .IN und .OUT realisiert worden.

Die Arithmetik ist um den Arcustangens und eine REMAINDER-Funktion, welche den ganzzahligen Rest einer Division ergibt, erweitert worden.

Interessant ist noch der Befehl TOWARDS, mit dem die Schildkröte auf einen bestimmten Koordinatenpunkt ausgerichtet werden kann.

Dies sind nur einige Beispiele aus einer Palette von insgesamt etwa 45 neuen Befehlen. Die in meinen Augen wesentlichste Verbesserung stellt jedoch die größere Anzahl der zur Verfügung stehenden Knoten dar. 3761 Knoten stehen nun nach dem Aufruf von LOGO3 zur Verfügung; beim CPC 464 waren es nur 2105. Aber der hatte ja auch nur 64K.

Drei Bilder pro Sekunde

Wie ja schon in der letzten Ausgabe zu lesen war, besitzt der CPC 6128 die Möglichkeit, fünf Bildschirmseiten gleichzeitig im Arbeitsspeicher unterzubringen. Mit dem Listing „GRAFDEMO“ habe ich einmal versucht, mit welcher Geschwindigkeit sich diese Bilder austauschen lassen. Das Programm lädt fünf Bilder von Diskette und tauscht diese dann gegeneinander aus. Sobald das Programm unterbrochen wird, erfahren wir, wieviele Bilder pro Sekunde angezeigt wurden. Nachdem ich dieses Programm einige Minuten laufen ließ, erhielt ich einen Wert von ca. 3 Bildern/sec.

Da das Arbeiten mit relativen Dateien schon in der vorletzten Ausgabe erklärt wurde, will ich an dieser Stelle nicht näher darauf eingehen.

Ich habe einige Spiele auf dem 6128 ausprobiert und festgestellt, daß viele Programme ohne besondere Anpassung vom CPC 464 auf den CPC 6128 übertragen werden können. Nur bei einigen maschi-

Kompatibel?

nensprachorientierten Programmen kam es zu Fehlern. Aber die meisten Softwarehäuser werden wohl auch angepaßte Versionen ihrer Programme für den CPC 6128 anbieten. Der SOFTWARE-TEAM-BASIC-Compiler läuft leider auf dem neuen nicht. An einer Anpassung wird jedoch gearbeitet. (tb)

TEAM-TEST: Vortex 512 KB-Speichererweiterung

Galten vor ein paar Jahren noch Speicherkapazitäten von 16KB als guter Durchschnitt, so müssen es heute 64KB sein. Der Schneider hat diese 64 KB freies RAM, aber u.U. reicht auch das nicht aus. Bei Anwendungen wie Datenbanken erscheint schnell die Meldung 'Out of Memory' und läßt einen ratlosen Benutzer zurück. Auch ist ein ernsthaftes Arbeiten mit dem Schneider-CP/M, das dem Anwenderprogramm nur noch 38KB Speicher übrig läßt, nur sehr schwer vorstellbar.

Die Lösung: Vortex

Schneider hat dieser Entwicklung mit seinem 6128, der 128KB bietet, Rechnung getragen, aber was macht derjenige, der seinen 464 zu Hause stehen hat? Nun, obwohl sich die Schneider-Computer aufgrund ihrer 'Minimalkonzeption' nur sehr schwer erweitern lassen, hat die Firma Vortex das Kunststück fertiggebracht, eine Speicherkarte zu entwickeln, mit dem Anwendern bis zu 512KB zur Verfügung stehen. Man kann dabei zwischen mehreren Karten wählen und seinen Rechner auf den Wert aufstocken, den man braucht (was natürlich auch davon abhängt, wie 'flüssig' man gerade ist. Kleinere Karten kann man dann auch nachträglich weiter ausrüsten, bis zum maximalen Wert von einem halben Megabyte, mit dem wohl jeder auskommen dürfte.

Die Karte selber (uns stand die SP512, also die Maximalausführung

Leichter Einbau

zur Verfügung) macht einen sehr sauber verarbeiteten und professionellen Eindruck. Platine und Verarbeitung sind durchaus Industriequalität, wenn auch noch einige Kontakte über Drahtbrücken hergestellt wurden. Der Einbau ist auch für den Bastel-Unerfahrenen relativ einfach, da jeder Schritt im

Handbuch genau beschrieben und auch mit Bildern illustriert wird. Bemerkenswert ist, daß man praktisch mit einem Schraubenzieher als Werkzeug auskommt, da kein Löten notwendig ist. Es werden einfach zwei ICs aus der Computerplatine genommen, in die Speichererweiterung eingesetzt, zwei Kontakte angeklemt, und das war es dann auch schon. Bei sorgfälti-

wendige Software liegt auf einem ROM auf der Karte, so daß sie gleich nach dem Einschalten zur Verfügung steht, ohne daß man etwas von Kassette/Diskette lesen müßte). Diese Befehle sind zum Teil extrem wirkungsvoll. Dadurch wird der 464 leistungsfähiger als ein 6128, ist es doch sogar möglich, BASIC-Programme in verschiedenen Speicher-'Bänken' miteinan-

Tabelle 1: Zusätzliche Basic-Befehle.

BANK	LIST	RECORDS
BASIC	LOAD	RETURN
BOS	MASK	RUN
CALL	MON	SAVE
COMMON	NEW	SCREEN.IN
DEV	PEEK	SCREEN.OFF
FAST	POKE	SCREENS
FRAME	RAMCLOSE	UNMASK
GOSUB	RAMFIELD	VIDEO.ON
GPAPER	RAMOPEN	VIDEO.OFF
GPEN	RAMREAD	
ID	RAMWRITE	

ger Arbeit und nochmaligem Check haben wir nur eine halbe Stunde gebraucht, bis der Computer wieder betriebsbereit war.

Zusätzliche Software

Schaltet man den Computer ein, erscheint außer der Einschaltmeldung auch noch ein Logo der Speicherkarte. Darin wird der Benutzer darüber informiert, wieviel Speicherkapazität ihm nun zur Verfügung steht und wie diese sich aufteilt (Daten/Programm/Druckerspoo-ler).

Die Vortex-Erweiterungen verwenden zum Verwalten des Speichers dasselbe Prinzip, wie es der 6128 tut: Bank-Switching. Das heißt, man hat nicht immer gleichzeitig den gesamten Speicher zum Verfü- gung, sondern kann jeweils auswählen, mit welchem Teil man arbeiten möchte. Dazu stehen eine Reihe von BASIC-Befehlen (siehe Tabelle 1) zur Verfügung (die not-

der zu verknüpfen. Besonders interessant ist die Möglichkeit, einen Druckerspoo-ler zuzuschalten. Dadurch muß der Computer nicht mehr auf den Drucker (und der User auf den Computer) warten, sondern man kann, während der Drucker noch emsig an einem langen Listing arbeitet, schon das neueste Computerspiel spielen. Doch die Aufzählung der neuen Möglichkeiten ist noch nicht zu Ende: Man kann mit Hilfe der neuen BASIC-Befehle Bilder im Speicher ablegen und in (Drittel-)Sekunden-schnelle wieder auf den Bildschirm holen oder eine relative Datei im RAM anlegen, wodurch sehr komfortable und schnelle Dateiprogramme möglich sind.

Ein neues CP/M

Neben der Erweiterungskarte selber und dem Handbuch bekommt der Käufer auch noch eine Kassette, auf der sich Hilfspro-

gramme befinden. Damit ist es dann möglich, eine neue CP/M-Version zu erstellen, bei der 61KB TPA (Transient Programm Area) zur Verfügung stehen. Erst dadurch wird der CPC auch wirklich 100% CP/M-fähig. Gleichzeitig kann man nun aber auch noch ein drittes (simuliertes) Diskettenlaufwerk unter dem Namen C: ansprechen, das im RAM liegt. Da dieses 'Laufwerk' sehr viel schneller ist als die echten, bedeutet das bei Disk-intensiven Arbeiten (z.B. Kopieren) eine große Zeitersparnis. Der Inhalt der Pseudo-Diskette kann auch wieder auf andere Disketten geschrieben werden.

Pseudo-Floppy

Zu guter Letzt soll auch noch ein Extra nicht unerwähnt bleiben, das man beim Kauf noch dazu bekommt: Auf dem ROM, in dem die Kartensoftware steckt, war wohl noch etwas Platz, und so kommt der Benutzer in den Genuß eines sehr guten Maschinensprache-Monitors und einiger BASIC-Befehle, die den neuen 664 Befehlen entsprechen (MASK etc.). Der Monitor ist, wenn auch sicherlich nur für eine kleinere Gruppe der Käufer interessant, ein professionelles Programm, das sogar einen Zeilenassembler enthält.

Jeder, der sich für die Vortex-Speichererweiterung interessiert, sollte daran denken, daß einige der oben beschriebenen Möglichkeiten u.U. nur ab einer bestimmten Erweiterungsstufe arbeiten. Die Fa. Vortex gibt diesbezüglich sicher gerne Auskunft. Zusammenfassend kann man nur sagen: Die RAM-Karte ist ein gut durchdachtes und konstruiertes System, das man jedem, der daran denkt, seinen Computer zu erweitern, nur empfehlen kann.

**COMPUTER
TEAM
jeden Monat
NEU**

CPC-Speicher optimal genutzt!

Optimizer für CPC-BASIC-Programme

von THOMAS BARNDT

Lange Variablenamen sind sehr komfortabel. Dies wissen sicher auch alle CPC-Benutzer zu schätzen. Dennoch: Komfort kann Nachteile haben. Variablenamen benötigen Speicherplatz und zwar für jeden weiteren Buchstaben ein Byte mehr. Bei einem langen Textverarbeitungsprogramm kann dies schon einiges an verschwendetem Speicherplatz ausmachen. Der Text muß ja schließlich auch irgendwo gespeichert werden. Auch REM-Zeilen kosten Speicherplatz und sind für das Funktionieren des Programms nicht notwendig. Dieser Optimizer schafft da Abhilfe. Mit seiner Hilfe können alle REM-Anweisungen aus einem Programm entfernt werden. Variablenamen werden automatisch auf maximal zwei Buchstaben plus Typkennzeichen gekürzt.

Kleine Einschränkungen

Ein paar Einschränkungen hat die Sache jedoch. In dem zu optimierenden Programm dürfen nie mehr als 26 Variablen mit gleichem Typkennzeichen und Anfangsbuchstaben vorkommen, da der Optimizer den Anfangsbuchstaben beibehält (falls man das Programm vielleicht doch noch einmal bearbeiten will!) und die jeweils zweiten Buchstaben alphabetisch verteilt.

Es kann auch passieren, daß der Optimizer Variablen wie IF oder TO verwendet. Dies muß dann jeweils durch einen Probelauf korrigiert werden. Der Computer meldet ja Fehler, wenn er solche Zeilen bearbeitet.

String-Feldvariable pro\$(). Das Programm arbeitet in drei Schritten und zeigt an, in welchem es sich befindet und welche Zeile gerade bearbeitet wird. Wenn das Programm einwandfrei läuft, kann es ohne Schwierigkeiten auch auf sich selbst angewandt werden. Nach Beendigung der Arbeit wird ein ASCII-File mit dem Namen OPTI.File erzeugt. Darin befindet sich das fertig optimierte Programm. Es kann nun ganz normal geladen und gestartet werden. In Zeile 20 des Optimizers sollte die Anzahl der Zeilen und Variablen mit möglichst niedrigen Werten vereinbart werden.

Die Anwendung

Das zu bearbeitende Programm muß als ASCII-File abgespeichert sein. Zu diesem Zweck lädt man es und speichert es erneut mit SAVE-"name".A. Nachdem man die Cassette an den Anfang des Files zurückgespult hat, lädt man den Optimizer und startet ihn. Dieser erwartet dann die Eingabe des Filenamens und lädt das Programm in die

Optimierter TEAM- TEXTER

Das Programm Team-Texter aus der vorletzten Ausgabe ließ sich einwandfrei optimieren. Es wurden ca. 3K Speicherplatz gespart, wodurch die Anzahl der Zeilen, die mit diesem Textverarbeitungsprogramm bearbeitet werden können, von 230 auf 270 erhöht werden konnte. (tb)

```

1 '*****
2 '** Optimizer **
3 '** (c) by THOMAS BARNDT **
4 '*****
5 '
10 MODE 2:DEFINT a-z:BORDER 13
20 anzvar=100:anzzeil=500
30 DIM pro$(anzzeil),var$(anzvar,1),hochpos(6,1)
40 LOCATE 30,3:PRINT CHR$(24);" Optimizer by THBCS ";CHR$(24):LOCATE 1,6
50 '
60 ' *** Programm einlesen ***
70 '
80 i=0:vnr=0
90 OPENIN""
100 WHILE NOT EOF
110 IF i>anzzeil THEN PRINT CHR$(7);
    "*** Programmfeld zu klein ***":
    END
120 LINE INPUT#9,pro$(i)
130 i=i+1
140 WEND
150 CLOSEIN
160 '
170 ' *** REM's entfernen und Variablen-
    lentabelle erstellen ***
180 '
190 zeilen=i-1
200 LOCATE 1,10:PRINT "Pass 1 working
    on line"
210 FOR j=0 TO zeilen
220 LOCATE 23,10:PRINT VAL(pro$(j))
230 zeile$=pro$(j)
240 GOSUB 1190
250 t$=LEFT$(zeile$,1)
260 IF t$="" OR LEFT$(zeile$,3)="REM" OR t$=":" THEN pro$(j)="" :GOTO 290
270 IF LEFT$(zeile$,4)="DATA" THEN 290
280 GOSUB 1090
290 NEXT j
300 '
310 ' *** GOTO's und GOSUB's aendern *
    **
320 '
330 LOCATE 1,12:PRINT "Pass 2 working
    on line"
340 FOR j=0 TO zeilen
350 LOCATE 23,12:PRINT VAL(pro$(j))
360 in=0
370 in=INSTR(in+1,pro$(j),"GOSUB")
380 IF in>0 THEN in=in+6:GOSUB 1560:
    GOTO 370
390 in=INSTR(in+1,pro$(j),"GOTO")
400 IF in>0 THEN in=in+5:GOSUB 1560:
    GOTO 390
410 in=INSTR(in+1,pro$(j),"THEN")
420 IF in>0 THEN in=in+5:GOSUB 1560:
    GOTO 410
430 in=INSTR(in+1,pro$(j),"ELSE")
440 IF in>0 THEN in=in+5:GOSUB 1560:
    GOTO 430
450 NEXT j
460 '

```

```

470 ' *** Sortieren der Variablen ***
480 '
490 FOR i=1 TO vnr-1
500 f1=0
510 FOR j=vnr-1 TO i STEP -1
520 IF var$(j-1,0)>var$(j,0) THEN
    h$=var$(j,0):var$(j,0)=var$(j-1,0):var$(j-1,0)=h$:f1=1
530 NEXT j
540 IF f1=0 THEN 570
550 NEXT i
560 '
570 ' *** Neue Variablen erzeugen ***
580 '
590 merk$=""
600 FOR i=0 TO vnr-1
610 typ$=RIGHT$(var$(i,0),1)
620 typ=0
630 IF typ$="$" THEN typ=1
640 IF typ$="%" THEN typ=2
650 IF typ$="!" THEN typ=3
660 var$(i,1)=LEFT$(var$(i,0),1)
670 IF var$(i,1)<>merk$ THEN chr(0)=
    97:chr(1)=97:chr(2)=97:chr(3)=97
680 merk$=var$(i,1)
690 IF INSTR(pr$(typ),var$(i,1))=0 THEN pr$(typ)=pr$(typ)+var$(i,1):
    GOTO 710
700 var$(i,1)=var$(i,1)+CHR$(chr(typ)):chr(typ)=chr(typ)+1:IF chr(typ)>122 THEN PRINT CHR$(7);"* Variablenfehler *":END
710 IF typ>0 THEN var$(i,1)=var$(i,1)+typ$
720 NEXT i
730 '
740 ' ** Neue Variablen einsetzen **
750 '
760 GOSUB 1870
770 LOCATE 1,14:PRINT "Pass 3 working
    on line"
780 FOR i=0 TO zeilen
790 wert=VAL(pro$(i))
800 LOCATE 23,14:PRINT wert
810 IF LEFT$(RIGHT$(pro$(i),ABS(LEN(pro$(i))-LEN(STR$(wert)))),4)="DATA" THEN 960
820 GOSUB 1780
830 FOR j=vnr-1 TO 0 STEP -1
840 varpos=0
850 varpos=INSTR(varpos+1,pro$(i),var$(j,0))
860 IF varpos=0 THEN 950
870 q=0
880 IF varpos>hochpos(q,0) AND varpos<hochpos(q,1) THEN 850
890 q=q+1
900 IF q<=k1 THEN 880
910 t$=MID$(pro$(i),varpos-1,1)
920 IF t$>="a" AND t$<="z" OR INSTR("0123456789.",t$)>0 THEN 850
930 pro$(i)=LEFT$(pro$(i),varpos-1)+var$(j,1)+RIGHT$(pro$(i),LEN(pro$(i))-(varpos-1)-LEN(var$(j,0)))
940 GOSUB 1780:GOTO 850
950 NEXT j

```

```

960 NEXT i
970 '
980 ' ** Leerzeichen und : entfernen *
    *
990 '
1000 FOR i=0 TO zeilen
1010   t%=RIGHT$(pro$(i),1)
1020   IF t%=":" OR t%=" " THEN pro$(i)
      =LEFT$(pro$(i),LEN(pro$(i))-1):G
      OTO 1010
1030 NEXT
1040 '
1050 '
1060 '
1070 GOTO 1950
1080 '
1090 ' *** Variablenamen suchen ***
1100 '
1110 IF zeile$="" THEN RETURN
1120 t%=LEFT$(zeile$,1)
1130 IF t%=CHR$(34) THEN GOSUB 1250:GOT
    O 1110
1140 IF t%="'" OR LEFT$(zeile$,3)="REM"
      THEN GOSUB 1490:RETURN
1150 IF t%>="a" AND t%<="z" THEN GOSUB
    1330:GOTO 1110
1160 zeile%=RIGHT$(zeile$,LEN(zeile$)-1
    )
1170 GOTO 1110
1180 '
1190 ' *** Zeilennummer entfernen ***
1200 '
1210 t%=LEFT$(zeile$,1)
1220 IF t%>="0" AND t%<="9" OR t%=" " T
    HEN zeile%=RIGHT$(zeile$,LEN(zeile
    $)-1):GOTO 1210
1230 RETURN
1240 '
1250 ' *** Strings entfernen ***
1260 '
1270 zeile%=RIGHT$(zeile$,LEN(zeile$)-1
    )
1280 IF LEFT$(zeile$,1)<>CHR$(34) THEN
    1270
1290 zeile%=RIGHT$(zeile$,LEN(zeile$)-1
    )
1300 RETURN
1310 '
1320 '
1330 ' *** Variablenname in Tabelle ein
    tragen ***
1340 '
1350 variable$=""
1360 variable%=variable%+t%
1370 zeile%=RIGHT$(zeile$,LEN(zeile$)-1
    )
1380 IF zeile$="" THEN 1410
1390 t%=LEFT$(zeile$,1)
1400 IF t%>="a" AND t%<="z" OR INSTR("0
    123456789.?!$",t%)>0 THEN 1360
1410 FOR k=0 TO vnr-1
1420   IF var$(k,0)=variable% THEN RETU
    RN
1430 NEXT
1440 IF vnr>anzvar THEN PRINT CHR$(7);"
    * Variablenfeld zu klein *":END
1450 var$(vnr,0)=variable%
1460 vnr=vnr+1
1470 RETURN

1480 '
1490 ' *** REM's entfernen ***
1500 '
1510 in=INSTR(pro$(j),zeile$)-1
1520 pro$(j)=LEFT$(pro$(j),in)
1530 RETURN
1540 '
1550 '
1560 ' *** Zeilennummer ? ***
1570 '
1580 IF VAL(MID$(pro$(j),in,1))>0 THEN
    1600
1590 RETURN
1600 '
1610 ' *** Zeilennummer suchen und erse
    tzen
1620 '
1630 inn=in:znr$=""
1640 znr%=znr%+MID$(pro$(j),in,1)
1650 in=in+1
1660 IF MID$(pro$(j),in,1)>="0" AND MID
    $(pro$(j),in,1)<="9" THEN 1640
1670 znr=VAL(znr%)
1680 FOR i=0 TO zeilen
1690   lznr=VAL(pro$(i))
1700   IF lznr=znr THEN RETURN
1710   IF lznr>znr THEN 1740
1720 NEXT i
1730 '
1740 pro$(j)=LEFT$(pro$(j),inn-1)+RIGHT
    $(STR$(lznr),LEN(STR$(lznr))-1)+RI
    GHT$(pro$(j),LEN(pro$(j))-(inn-1)-
    LEN(znr%))
1750 in=inn+LEN(STR$(lznr))-1:IF MID$(p
    ro$(j),in,1)="," THEN in=in+1:GOTO
    1600
1760 RETURN
1770 '
1780 ' ** Zeichenkettenpositionen ermit
    teln **
1790 '
1800 k=0:k1=0:hochpos(k,1)=0
1810 hochpos(k,0)=INSTR(hochpos(k1,1)+1
    ,pro$(i),CHR$(34))
1820 hochpos(k,1)=INSTR(hochpos(k,0)+1,
    pro$(i),CHR$(34))
1830 IF hochpos(k,0)>0 THEN k=k+1:k1=k-
    1:GOTO 1810
1840 RETURN
1850 '
1860 '
1870 ' ** Gleichbleibende Variablen aus
    Tabelle **
1880 '
1890 k=0
1900 FOR i=0 TO vnr-1
1910   IF var$(i,0)<>var$(i,1) THEN var
    $(k,0)=var$(i,0):var$(k,1)=var$(
    i,1):k=k+1
1920 NEXT
1930 vnr=k
1940 RETURN
1950 '
1960 ' *** Variablentabelle ausgeben **
    *
1970 '
1980 PRINT:PRINT" Variablen:":PRINT
1990 FOR l=0 TO vnr-1

```

```

2000 PRINT var$(1,0);TAB(20);" - ";va
      r$(1,1)
2010 FOR warten=0 TO 500:NEXT
2020 NEXT
2030 PRINT
2040 '
2050 ' ** Programm abspeichern **
2060 '

```

```

2070 SPEED WRITE 1
2080 OPENOUT"opti.file"
2090 FOR i=0 TO zeilen
2100 IF pro$(i)<>"" THEN PRINT#9,pro$(
      i)
2110 NEXT
2120 CLOSEOUT

```

Maschinensprache von BASIC erzeugt

Computer : CPC 464

Sprache : BASIC

Peripherie : evtl. Drucker, Diskette oder Kassette

Um Programme in Maschinensprache auch Anwendern, die keinen Assembler besitzen, zugänglich zu machen, sollte zu jedem Assemblerprogramm auch ein in BASIC geschriebener Lader gehören. Dieser besteht im wesentlichen aus Data-Zeilen, die das Assemblerprogramm im Maschinencode enthalten, und einer For-Next-Schleife, in der der Code in die entsprechenden Speicherstellen gepoket wird. Bei längeren Ladern sind Tippfehler in den Data-Zeilen fast unvermeidlich. Es liegt also nahe, den Computer diese Arbeit selbst durchführen zu lassen.

Das vorliegende Programm liefert von jedem im Speicher befindlichen Maschinenprogramm innerhalb kürzester Zeit den dazugehörigen BASIC-Lader einschließlich Prüfsumme. Der Lader kann auf dem Drucker ausgegeben oder unter einem frei wählbaren Namen auf Diskette oder Kassette abgespeichert werden.

Der auf Diskette/Kassette abgespeicherte BASIC-Lader wird wie gewohnt, z.B. mit 'RUN „Name“', geladen und gestartet. Während des Programmablaufs wird der Inhalt der Data-Zeilen aufaddiert, mit der Prüfsumme verglichen und eventuell eine Fehlermeldung ausgegeben. Bei fehlerfreiem Ablauf wird die korrekte Speicherung des Maschinenprogramms bestätigt und der Speicherbereich angegeben.

```

10 REM          *** ERSTELLEN EINES BASIC-LADERS ***
20 REM          *** copyright 1985 by Beate Lang ***
30 REM
40 MODE 2:DEFINT a-z
50 INPUT"Name des Maschinenprogramms (max. 8 Zeichen)":na#
60 IF LEN(na#)>>8 THEN 50
70 INPUT"Anfangsadresse des Maschinenprogramms ":ad1
80 INPUT"Endadresse des Maschinenprogramms ":ad2
90 INPUT"Bei welcher Zeilennummer soll der Lader beginnen ":z
100 n=(ad2+1-ad1)/17+2.49:DIM feld$(n):i=ad1:s=0
110 FOR k=0 TO n-4
120 ad=PEEK(i):s=s+ad:feld$(k)=STR$(z)+" DATA "&"+HEX$(ad)
130 FOR j=i+1 TO i+16
140 ad=PEEK(j):s=s+ad:feld$(k)=feld$(k)+"",&"+HEX$(ad):NEXT
150 i=j:z=z+10:NEXT
160 ad=PEEK(i):s=s+ad:feld$(k)=STR$(z)+" DATA "&"+HEX$(ad)
170 WHILE i<ad2
180 i=i+1:ad=PEEK(i):s=s+ad:feld$(k)=feld$(k)+"",&"+HEX$(ad)
190 WEND
200 z=z+10:feld$(n-2)=STR$(z)+" s=0:FOR i=&"+HEX$(ad1)+" TO "&
+HEX$(ad2)+":READ a:poke i,a:s=s+a:next"
210 z=z+10:feld$(n-1)=STR$(z)+" ?:IF s<>"+STR$(s)+" THEN ?"
+CHR$(34)+"Fehler in den Data-Zeilen!"+CHR$(34)+":END"
220 z=z+10:feld$(n)=STR$(z)+" ?"+CHR$(34)+"Programm "+na#
+" korrekt geladen, Startadresse : "&"+HEX$(ad1)
+" Endadresse : "&"+HEX$(ad2)+CHR$(34)
230 w=0:PRINT:GOSUB 290:PRINT"Ausgabe auf Drucker (j/n) ?"
240 CALL &BB06:IF INKEY(45)=0 THEN w=8:GOSUB 290
250 PRINT"Ausgabe auf Diskette/Kassette (j/n) ?":CALL &BB06
260 IF INKEY(45)=0 THEN OPENOUT na#+".bas":w=9:GOSUB 290:CLOSEOUT
270 PRINT"Soll noch ein Lader erstellt werden (j/n) ?"
280 CALL &BB06:IF INKEY(45)=0 THEN ERASE feld$:GOTO 40 ELSE END
290 FOR k=0 TO n:PRINT#w,feld$(k):NEXT:RETURN

```

Mineralien bestimmen mit dem CPC 464

Dieses Programm erleichtert die Bestimmung von Mineralien. Wenn Sie schon einmal auf einer Wanderung oder im Urlaub einen ungewöhnlichen Stein gefunden haben, ist es sehr wahrscheinlich, daß es sich um ein Mineral handelt. Oft versucht man dann, das Mineral zu bestimmen, indem man in entsprechenden Büchern nach ähnlich aussehenden Mineralien sucht und die Eigenschaften vergleicht. Da es sehr viele Mineralien gibt, ist dies oft eine sehr aufwendige Arbeit.

Wenn Sie dieses Programm abgetippt haben, brauchen Sie nur noch die Eigenschaften Ihres gefundenen Minerals einzugeben. Das Programm sucht dann alle Mineralien mit diesen Eigenschaften aus den DATA-Zeilen und zeigt sie an. Sie brauchen dann nur noch die vorgeschlagenen Mineralien mit dem gefundenen zu vergleichen. Es werden vier Kriterien zur Bestimmung verwendet:

- Kristallsystem
- Härte
- Farbe
- Strich

Wenn Sie nicht alle vier Merkmale Ihres Minerals kennen, so ist dies nicht weiter schlimm. Es ist nur wahrscheinlich, daß der Computer Ihnen dann mehr Vorschläge ausgibt. Die bereits abgedruckten Mineralien sind dem Kosmos-Steinführer entnommen. Damit das Programm optimal arbeiten kann, sollten noch möglichst viele Mineralien in DATA-Zeichen angehängt werden. Dazu müssen Sie wissen, in welcher Reihenfolge die Informationen auftreten. Als erstes steht der Name des Minerals. Es folgt das Kristallsystem. Es können durchaus mehrere Kristallsysteme vorkommen; daher werden diese mit einem großen „K“ abgeschlossen. Die Kristallsysteme sind nach folgendem Schlüssel codiert:

Danach müssen zwei Zahlen für die Härte stehen; die kleinere zuerst. Sollte die Härte nicht angegeben sein, so tippen Sie für jede Zahl einfach zwei Fragezeichen. Dies war zum Beispiel bei Arsen der Fall. Nun kommen die Farben, in denen das Mineral vorkommt. Diese sind wie folgt geschlüsselt:

Die Farben müssen mit einem großen „F“ abgeschlossen werden. Als letzte Angabe folgt der Strich. Dort können ebenfalls mehrere Farben vorkommen. Sie sind nach demselben Schlüssel codiert. Die Angaben für den Strich schließt man mit einem großen „S“ ab. Den Strich eines Minerals ermittelt man mit einem Strichtäfelchen aus Keramik, welches eine raue Oberfläche besitzt. Daran reibt man das zu bestimmende Mineral. Die Farbe des so gewonnenen Abriebs ist der Strich.

1 rhombisch	6 hexagonal
2 monoklin	7 triklin
3 trigonal	8 oktaedrisch
4 kubisch	9 amorph
5 tetragonal	

2 weiß	9 grau
3 grün	10 schwarz
4 gelb	11 silberweiß
5 rot	12 goldgelb
6 blau	13 kupferrot
7 violett	14 hellgrau
8 braun	15 farblos

```

80 .
90 . ** Rahmen **
100 .
110 LOCATE 1,3
120 PRINT CHR$(150);STRING$(42,CHR$(154));CHR$(156);CHR$(150);STRING$(34,CHR$(154));CHR$(156);
130 FOR i=4 TO 20
140 LOCATE 1,i:PRINTCHR$(149);
150 LOCATE 44,i:PRINTCHR$(149);
160 LOCATE 45,i:PRINTCHR$(149);
170 LOCATE 80,i:PRINTCHR$(149);
180 NEXT
190 LOCATE 1,21:PRINT CHR$(147);STRING$(42,CHR$(154));CHR$(153);CHR$(147);STRING$(34,CHR$(154));CHR$(153);
200 .
210 . ** Farben und Kristallsysteme lesen **
220 .
230 i=0
240 READ ksys$(i)
250 IF ksys$(i)="ende" THEN kmax=i-1:GOTO 270
260 i=i+1:GOTO 240
270 i=0
280 READ farbe$(i)
290 IF farbe$(i)="ende" THEN fmax=i-1:GOTO 310
300 i=i+1:GOTO 280
310 .
320 . ** Menue **
330 .
340 PRINT#1,"M - Suchen nach Merkmalen"

```

```

10 MODE 2:BORDER 0:INK 1,26:INK 0,0:PEN 1:PAPER 0
20 WINDOW #1,3,43,5,20:WINDOW #4,2,80,23,25
30 WINDOW #2,47,79,5,20
40 DEFINT a-z
50 DIM ksys$(30),farbe$(30),strich$(30)
60 PRINTTAB(30);"-- Minerale bestimmen --"
70 PRINT TAB(36);CHR$(164);" by THBCS"

```

```

350 PRINT#1,"S - Suchen nach Namen"
360 PRINT#1,"A - Ausgabe aller Minerale"
365 PRINT#1,"E - Ende des Programms"
370 wahl$=UPPER$(INKEY$):IF wahl$="" THEN 370
380 IF INSTR("MSAE",wahl$)<1 THEN 370
390 ON INSTR("MSAE",wahl$) GOSUB 550,1440,460,425
5
400 CLS#4:PRINT#4,TAB(8);"< ENTER >"
410 CALL &BB06
420 CLS#1:CLS#4:CLS#2:GOTO 320
425 MODE 2:END
430 '
440 ' ** Ausgabe aller Minerale **
450 '
460 RESTORE 30000
470 PRINT#4,TAB(8);"< ENTER >"
480 READ name$
490 CLS#1
500 GOSUB 1550
510 READ name$
520 IF name$="ende" THEN RETURN
530 CALL &BB06 :GOTO 490
540 '
550 ' *** Merkmale eingeben ***
560 '
570 CLS#2:PRINT#2,TAB(10);"MERKMALE":PRINT#2
580 CLS#1:PRINT#1,TAB(10);"KISTALLSYSTEM":PRINT#1
590 FOR i=0 TO kmax
600 PRINT #1,i;ksys$(i)
610 NEXT
620 CLS#4:INPUT#4,"Ihre Wahl: ",wahl$:ksys=VAL(wahl$)
630 IF ksys<0 OR ksys >i-1 THEN 620
640 PRINT#2,"Kristallsystem: ";ksys$(ksys);
650 CLS#1
660 '
670 CLS#1:PRINT#1,TAB(10);"FARBE":PRINT#1
680 FOR i=0 TO fmax
690 LOCATE#1,(i\10)*15+1,i MOD 10+3
700 PRINT #1,i;farbe$(i);
710 NEXT
720 CLS#4:INPUT#4,"Ihre Wahl: ",wahl$:farbe=VAL(wahl$)
730 IF farbe<0 OR farbe>i-1 THEN 720
740 PRINT#2,"Farbe : ";farbe$(farbe);
750 farbe=farbe+1
760 '
770 CLS#1:PRINT#1,TAB(10);"STRICH":PRINT#1
780 FOR i=0 TO fmax
790 LOCATE#1,(i\10)*15+1,i MOD 10+3
800 PRINT #1,i;farbe$(i);
810 NEXT
820 CLS#4:INPUT#4,"Ihre Wahl: ",wahl$:strich=VAL(wahl$)
830 IF strich<0 OR strich>i-1 THEN 820
840 PRINT#2,"Strich : ";farbe$(strich)
850 strich=strich+1
860 '
870 CLS#1:PRINT#1,TAB(10);"HAERTE (nach F.Mohs)":PRINT#1
880 PRINT#1," 1 Talk O Unbekannt"
890 PRINT#1," 2 Gips"
900 PRINT#1," 3 Kalzit"
910 PRINT#1," 4 Fluorit"
920 PRINT#1," 5 Apatit"
930 PRINT#1," 6 Orthoklas"
940 PRINT#1," 7 Quarz"
950 PRINT#1," 8 Topas"
960 PRINT#1," 9 Korund"
970 PRINT#1,"10 Diamant"
980 PRINT#1:PRINT#1," Jedes angegebene Mineral ritzt das"
990 PRINT#1," Vorangegangene bzw. wird von dem"
1000 PRINT#1," Nachfolgenden geritzt."
1010 CLS#4:INPUT#4,"Ihre Wahl: ",wahl$:haerte=VAL(wahl$)
1020 IF haerte!<1 AND haerte!<>0 OR haerte!>10 THEN 1010
1030 PRINT#2,"Haerte : ";
1040 IF haerte!<1 THEN PRINT#2,"Unbekannt" ELSE PRINT#2,haerte!
1050 CLS#4
1060 '
1070 ' ** Hier evtl. weitere Bestimmungskriterien **
1080 ' *
1090 ' **
1100 '
1110 ' ** Suchen der in Frage kommenden Minerale **
1120 '
1130 CLS#1:PRINT#1,"Vorschlaege:":PRINT#1
1140 RESTORE 30000
1150 READ name$
1160 IF name$="ende" THEN RETURN
1170 ksysflag=0
1180 READ k$
1190 IF k$="K" THEN 1230
1200 k=VAL(k$)
1210 IF k=ksys OR ksys=0 THEN ksysflag=-1
1220 GOTO 1180
1230 haerteflag=0
1240 READ h1$,h2$
1250 IF h1$="??" THEN haerteflag=-1:GOTO 1280
1260 h1!=VAL(h1$):h2!=VAL(h2$)
1270 IF haerte!>h1! AND haerte!<h2! OR haerte!=0 THEN haerteflag=-1
1280 farbflag=0
1290 READ f$
1300 IF f$="F" THEN 1340
1310 f=VAL(f$)
1320 IF f=farbe OR farbe=1 THEN farbflag=-1
1330 GOTO 1290
1340 strichflag=0
1350 READ s$
1360 IF s$="S" THEN 1410
1370 s=VAL(s$)
1380 IF s=strich OR strich=1 THEN strichflag=-1
1390 GOTO 1350
1400 '
1410 IF ksysflag AND haerteflag AND farbflag AND strichflag THEN PRINT#1,name$
1420 GOTO 1150
1430 '
1440 ' ** Suchen nach Namen **
1450 '
1460 RESTORE 30000
1470 CLS#1:INPUT#4,"Welches Mineral suchen Sie";suchname$
1480 suchname$=UPPER$(suchname$)
1490 CLS#4
1500 READ name$
1510 IF name$="ende" THEN PRINT#1,suchname$;" ist nicht eingetragen":RETURN
1520 IF name$=suchname$ THEN GOSUB 1550:RETURN
1530 READ wegdamit$: IF wegdamit$="S" THEN 1500 ELSE 1530
1540 '
1550 ' ** Mineral ausgeben **
1560 '
1570 PRINT#1,TAB(14);name$
1580 PRINT#1
1590 PRINT#1,"Kristallsysteme:"
1600 READ k$
1610 IF k$="K" THEN 1640
1620 PRINT#1,TAB(4);ksys$(VAL(k$))
1630 GOTO 1600
1640 PRINT#1,"Haerte: ";
1650 READ h1$,h2$
1660 IF h1$="??" THEN PRINT#1,"unbekannt":GOTO 1690
1670 IF h1$=h2$ THEN PRINT#1,VAL(h1$):GOTO 1690
1680 PRINT#1,VAL(h1$);"-";VAL(h2$)
1690 PRINT#1,"Farben:"
1700 READ f$
1710 IF f$="F" THEN 1740
1720 PRINT#1,TAB(4);farbe$(VAL(f$)-1)
1730 GOTO 1700
1740 PRINT#1,"Strich:"
1750 READ s$
1760 IF s$="S" OR s$="??" THEN RETURN
1770 PRINT#1,TAB(4);farbe$(VAL(s$)-1)
1780 GOTO 1750
1790 '
1800 '

```

Schneider Team

10000 *** Kristallsysteme ***

10010 DATA Unbekannt, Rhombisch, Monoklin, Trigonal, Kubisch

10020 DATA Tetragonal, Hexagonal, Triklin, Oktaedrisch, Amorph, ende

10030

20000 *** FARBE ***

20010 DATA Unbekannt, Weiss, Gruen, Gelb, Rot, Blau, Violet, Braun, Grau

20020 DATA Schwarz, Silberweiss, Goldgelb, Kupferrot, Hellgrau, Farblos, ende

20030

30000 *** Minerale ***

30010 DATA GOLD, 4, K, 2, 5, 3, 12, F, 12, S

30020 DATA SILBER, 4, K, 2, 5, 3, 11, F, 11, S

30030 DATA KUPFER, 4, K, ??, ??, 13, F, 13, S

30040 DATA EISEN, 4, K, 4, 3, 4, 7, 9, 10, F, S

30050 DATA ARSEN, 3, K, ??, ??, 14, 9, F, 14, S

30060 DATA ANTIMON, 3, K, 3, 3, 5, 14, F, 9, S

30070 DATA WISMUT, 3, K, 2, 2, 5, 11, F, 11, S

30080 DATA SCHWEFEL, 1, K, 1, 5, 2, 5, 4, F, 2, S

30090 DATA DIAMANT, 4, K, 10, 10, 15, 4, 8, 5, 10, F, 2, S

30100 DATA GRAPHIT, 6, K, 1, 2, 10, F, 10, S

30110 DATA SILBERGLANZ, 4, 1, K, 2, 2, 5, 10, F, 10, S

30120 DATA ARGENTIT, 4, K, 2, 2, 5, 10, F, 10, S

30130 DATA ARKANTHIT, 1, K, 2, 2, 5, 10, F, 10, S

30140 DATA BORNIT, 4, K, 3, 3, 5, 8, F, 14, S

30150 DATA COVELLIN, 6, K, 1, 5, 2, 6, 5, F, 9, 10, S

30160 DATA CHALKOSIN, 1, K, 2, 5, 3, 9, 10, F, 10, S

30170 DATA KUPFERGLANZ, 1, K, 2, 5, 3, 9, 10, F, 10, S

30180 DATA SPHALERIT, 4, K, 3, 5, 4, 4, 8, 10, F, 8, 4, 2, S

30190 DATA ZINKBLENDE, 4, K, 3, 5, 4, 4, 8, 10, F, 8, 4, 2, S

30200 DATA KUPFERKIES, 5, K, 3, 5, 4, 12, F, 3, 10, S

30210 DATA CHALKOPYRIT, 5, K, 3, 5, 4, 12, F, 3, 10, S

30220 DATA WURTZIT, 6, K, 3, 5, 4, 8, 10, F, 8, S

30230 DATA BLEIGLANZ, 4, K, 2, 5, 2, 5, 9, F, 9, S

30240 DATA GALENIT, 4, K, 2, 5, 2, 5, 9, F, 9, S

30250 DATA MAGNETKIES, 6, K, 3, 5, 4, 5, 4, 5, F, 9, 10, S

30260 DATA PYRRHOTIN, 6, K, 3, 5, 4, 5, 4, 5, F, 9, 10, S

EPSON-Drucker

RX-80	799,-	RX-90 F/T	919,-
FX-85	1329,-	FX-105	1739,-
LQ-1500	3198,-		
KX-80	1899,-	HI-80	1245,-

Seikosha-Drucker

GP-500 A.o. VC	538,-	GP-500 AS, CPC, AT	598,-
GP-700 A, VC, o. CPC	998,-	SP 800 A oder I	798,-
SP 1000 A, AS, o. VC	828,-	SP 1000 CPC	878,-
SP 1000 AP, Imagewriter kompatibel für Macintosh u. Apple Iic direkt anschließb.			948,-

Panasonic

KX-P 1090	675,-	KX-P 1091	848,-
KX-P 1092	1068,-	KX-P 3151 Typenradr	1528,-

Star-Drucker

Gemini 10 xl	sagenhafte 698,-		
SG 10	nur 889,-	SG 10 C	839,-
SD 10	1198,-	SR 10	1598,-
SG 15	1225,-	SD 15	1598,-
SR 15	1999,-		

Okidata MICROLINE

Typ 182, 183, 192, 193 Sonderpreise

Vortex Laufwerke und Karten, alle Software vom M & T Verlag lieferbar

D&G Datentechnik, Postf. 431, Bogenstr. 53, ☎ 02 61/40 88 39, 5400 Koblenz

Commodore

C 128, PC 10, PC 20 a. Anfr.

Schneider PC

CPC-464 grün grünem Monitor	725,-
CPC-464 mit farbigem Monitor	1198,-
Floppy-Laufwerk 3"	725,-
Disketten 3" f. Schneider CPC	10 St. 99,-
CPC-664 mit grünem Monitor	1248,-
CPC-664 mit Color-Monitor	1699,-
CPC-6128 mit grünem Monitor	1448,-
CPC-6128 mit Color-Monitor	1898,-
NLQ-401 Drucker	699,-

Hewlett Packard

HP-41 CV	439,-
HP-41 CX	648,-

Disketten u. Zubehör DM/10 St.

3 1/2" 1 D Markendiskette	75,-
3 1/2" 2 D Markendiskette	129,-
5 1/4" 1 D No name	23,-
5 1/4" 2 D No name	29,-

Sharp

PC 1401	215,-	PC 1402	299,-
PC 1500 A	399,-	PC 1500A+E150	699,-
PC 515 P	645,-	PC 2500	780,-



FÜR CPC 464 & 664

Die Tiefe
Ein deutsches Abenteuer-Spiel.
Ein Wrack aus dem 2. Weltkrieg, in dem ein Edelstein von unermesslichem Wert verborgen ist. Sie alleine gegen die Gefahren der Tiefe.
Mit Lösungscode. **C 19,90/D 29,90**

Basic-Kit
Ein Programmierool mit deutschem Zeichensatz, Basic-Befehlen auf Tastendruck und einer Fast Tape-Routine. Mit Tastatur-Schablone.
C 39,90/D 49,90

Nazca
Ein deutsches Grafik-Adventure. Im Dschungel Südamerikas, auf der Suche nach einer versunkenen Kultur. Doch Vorsicht, mannigfaltige Gefahren erwarten Sie. Mit Lösungscode.
C 39,90/D 49,90

Versand erfolgt per Vorkasse (Scheck) oder Nachnahme, zzgl. DM 3,- für Porto + Verpackung bzw. + Nachnahmegebühr.

Nina Franke Software Postfach 273
3440 Eschwege
Tel.: 0 56 51 / 6 05 78
von 19-21 Uhr

Händleranfragen erwünscht · Softwareautoren gesucht

Schneider CPC Software-Schnell-Versand

LOCOMOTION 9,95

LORDS OF MIDNIGHT 39,00

MACADAM BUMPER 29,90

MARSFOT 29,90

MASTER OF THE LAMPS 37,90

MESSAGE FROM ANDROMEDA 27,90

MOONBUGGY 27,90

MR FREEZE 9,95

MULTIDATA 69,00

MULTIVOKABEL 49,00

NEVERENDING STORY 34,90

NIGHTSHADE 35,90

NONTERRAQUEOUS 9,95

ONE MAN & DROID 29,90

ON THE RUN 29,90

PINBALL WIZARD 27,90

PROJECT FUTURE 34,90

RAID 111 (OVER MOSCOW) 35,90

RED ARROWS 35,90

RED MOON 28,90

RETURN TO EDEN 37,90

ROBIN OF SHERWOOD 37,90

ROCKY HORROR SHOW 29,90

SHORTS FUSE 9,95

SLAPSHOT 31,90

SORCERY 39,90

SOUL OF A ROBOT 9,95

SOULS OF DARKON 32,90

SOUTHERN BELLE 29,90

SPACE HUNTER 9,95

SPY VS SPY 34,90

STARION 34,90

STAR - NO! 39,90

STATISTIC - STAR 59,90

STRIP POKER 34,90

TALES OF ARABIAN NIGHTS 27,90

TASCOPI DEUTSCH 29,90

TASPRINT DEUTSCH 29,90

TASWORD DEUTSCH 49,90

THE HOBBIT 46,90

THE QUILL 59,00

THEY SOLD A MILLION 36,90

WARLORD 29,90

WAY OF EXPLODING FIST 36,90

WILD BUNCH 9,95

WIZARDS LAIR 29,90

WORLD CUP 34,90

WORLD SERIES BASEBALL 29,00

ZAPP 49,90

3 D STARSTRIKE 12,95

3 D STUNTRIDER 34,90

A VIEW TO A KILL 47,90

ACTION BIKER 9,95

AIRWULF 32,90

ALIEN 8 35,90

ARMHEM MOUSE 9,95

ASSEMBLER KURS (SYBEX) 39,90

BATTLE FOR MIDWAY 36,90

BATTLE OF BRITAIN 34,90

BEACH HEAD 37,90

BOULDERDASH 38,90

BRIAN JACKS SUPERSTAR 29,90

CENTRE COURT 31,90

CHILLER 38,90

ASSEMBLER KURS (SYBEX) 9,95

CODE NAME MAT 2 49,00

COLOUR STAR II 29,90

COMBAT LYNX 29,90

CONFUZION 29,00

COPY STAR II 39,90

CYRUS II SCHACH 39,00

DANGER MOUSE 38,90

DECATHLON 29,00

DESIGNER STAR 59,90

DEVILS CROWN 29,90

DEVPAC 80 129,00

DOPPLEGANGER 29,90

DRAGONTORC 34,90

DUN DARAGH 37,90

EVERYONE'S DAN 29,90

ERIC THE VIKING 36,90

EVERYONE'S A WALLY 33,90

FANTASTIC VOYAGE 34,90

FIGHTER PILOT 34,90

FINDERS KEEPERS 9,95

FOREST AT WORLDS END 27,90

FORMULA ONE 39,90

FORMULA ONE SIMULATOR 9,95

FRANK BRUNOS BOXING 29,00

GHOSTBUSTERS 38,90

GREMLINS 36,90

HACKER 38,90

HARD HAT MAC 37,90

HIGHWAY ENCOUNTER 35,90

KAISER 59,00

KNIGHT LORE 35,90

AMX movie
MAUS, GRAFIKPAKET und BASIC-ERWEITERUNG für alle CPC's 298,- DM

DISKETTE 3 1/2"

AIRWULF	49,00
CODE NAME MAT II	49,00
COM PACK	798,00
CYRUS II SCHACH	49,00
DATAMAT	139,00
DATEI STAR	98,00
GBASE II	299,00
DEVPAC 80	145,00
FANTASTIC VOYAGE	49,00
FIBU STAR	98,00
FIGHTER PILOT	49,00
GEWISS	49,00
GRAPHIK MASTER	89,00
LAGER STAR	98,00
LORDS OF MIDNIGHT	49,00
MASTERFILE	139,00
MULTIPLAN	249,00
MULTIVOKABEL	59,00
SLAPSHOT	49,00
SORCERY PLUS	49,00
SPY VS SPY	49,00
STAR WRITER II	198,00
STAR MON	89,00
STATISTIC STAR	79,90
SUPERCALC II	198,00
TASWORD 6128	96,00
TASWORD	96,00
TEX PACK	198,00
TEXTOMAT	139,00
TEXTOMAT PLUS	189,00
WORDSTAR	299,00

***** DRUCKER FÜR DEN CPC *****

MELCHERS CP 80	698,00
MELCHERS CPA 80 P.	698,00
NEU-SPEZIELL FÜR SCHNEIDER	
MELCHERS CPA 80 GS	749,00
TAXAN KP 810	1149,00
TAXAN KP 910	1449,00
SCHNEIDER NLQ 401.	798,00
PANASONIC KX-P 1090.	798,00
PANASONIC KX-P 1091.	998,00
PANASONIC KX-P 1092.	1249,00

Unentschlossen ?? Einfach Datenblätter anfordern !!!

Heimcomputer-Shop Bahnhofstraße 10 Hotline
Waldeck-Automaten Vertriebsgesellschaft mbH 2870 Delmenhorst (0 42 21) 164 64

Ja, schicken Sie mit umgehend folgende Artikel aus Ihrem Angebot

Stück _____ DM
Stück _____ DM
Stück _____ DM
Stück _____ DM

per Nachnahme
 per Vorkasse (Scheck) anheft.
 Barzahlung

Versandsumme _____ DM
Gesamtsumme _____ DM

Kategorie (bitte beschreiben) _____

Spezial Report

Neues von der Sordware Spy Ltd.



Nachdem wir Ihnen nun schon einmal diese junge aufstrebende Firma vorgestellt haben, möchten wir Ihnen in dieser Ausgabe den ersten selbstentwickelten Computer dieses Hauses vorstellen, den SS 63. Der Sordware Spy 63 ist die erste nennenswerte Neuentwicklung der Hardware-Abteilung. Denn die ersten 62 Versuchsmodelle explodierten beim ersten Anschluß an das Stromnetz ... Der SS 63 wurde deshalb vollkommen neu konzipiert: Um den Anschluß an das Stromnetz zu umgehen, wird der SS 63 mit Solarenergie betrieben.

Der SS 63! Verkaufshit

Dies hat mehrere Vorteile, denn erstens wird der User nie wieder über ein lästiges Stromkabel stolpern, zweitens konnte damit der kostenträchtige Einbau eines EIN/AUS-Schalters gespart werden.

Man stülpt einfach einen leeren Schuhkarton über den Computer, und mangels Licht stellt der SS 63 den Betrieb ein. Diese Methode hat zusätzlich den Vorteil, daß plötzlich eintreffender Besuch nicht sofort sieht, daß Sie, wie viele andere

auch, auf die Verkaufspraktiken der SORDWARE SPY, Ltd hereingefallen sind.

Tasten mit Solarzellen

Da allerdings die Solarzellen auf den Tasten angebracht sind, aus platz- und materialsparenden Gründen, sollte man nur mit einer Hand über dem Tastenfeld arbeiten, da sonst zuwenig Strom im Computer erzeugt wird, was Funktionsstörungen nach sich zieht. Somit hat der geübte Programmierer, der schon im 'Zehn-Finger-System' schreibt, eine schwierige Umgewöhnungsphase durchzumachen. Die Tastatur beinhaltet alle Buchstaben von A-Z, einige Sonderzeichen, wie z.B. \$ + <> , ; + = ? / * ... , sowie alle Ziffern von 0-9 und zwei Zusatz Tasten für die 12 und die 25, was dem SS 63 erhebliche Vorteile gegenüber der Konkurrenz verschafft. Damit man die Reset-Taste nicht unbeabsichtigt drückt, was ja normalerweise die Arbeit mehrerer Stunden zunichte macht, wurde diese unter eine Spannung von 20 V gesetzt, so daß derartige 'Programmier-Unfälle' auf ein absolutes Minimum reduziert werden.

Das Herz des Computers besteht aus einem vollkommen neuartig gestalteten 32 Bit-Prozessor. Dieser setzt sich aus acht im Bank-Switching-Verfahren verbundenen 4 Bit-Prozessoren zusammen, die wahlweise angesteuert werden können.

Unter diesen CPU's befindet sich auch der berühmte 4711 von Motorola, der sich absolut in den Schatten der Konkurrenz stellt.

Es ist eine phantastische Herausforderung für jeden Programmierer, die Wirkung der verschiedenen Befehle herauszufinden, da die SORDWARE SPY, Ltd es nicht für nötig hielt, ein Handbuch zu erstellen.

Kein Handbuch

Dies garantiert eine spannende Freizeitgestaltung, bis man die Grenzen der Befehlsmöglichkeiten des SS 63 erforscht hat. Wir konnten drei Befehle herausfinden, die wir aber nicht veröffentlichen, damit Ihnen der Spaß beim Experimentieren nicht vorenthalten bleibt. Der Marketing-Manager der SORDWARE SPY, Ltd erwägt, um die Verkaufszahlen zu steigern, einen Wettbewerb für die SS 63-Besitzer auszuschreiben:

Wer meint, alle Befehle herausgefunden zu haben, kann seine Ergebnisse an die SORDWARE SPY, Ltd einschicken. Die richtigen Lösungen nehmen an einer Verlosung teil:

1. Preis: Die Firma nimmt den SS 63 zum vollen Preis zurück.

2. Preis: Die Firma nimmt den SS 63 zum halben Preis zurück.

3. Preis: Die Firma verrät keinem, daß Sie einen SS 63 besitzen. Einsendeschluß ist der 31.12.86, alle Einsendungen bitte an die Adresse, die auf dem weißen Schild am Boden des SS 63 steht. jr



SKYFOX — FIGHTER

Datenträger: Cassette/Diskette
Rechnertyp: C-64 / Apple II
Vertrieb: Bertelsmann-Software
GmbH, Abt. Ariolasoft
Preis: DM 49,— / DM 69,—

Nachdem Skyfox in den Rechner eingeladen wurde, kann man gleich zu Beginn zwischen 5 Schwierigkeitsstufen und 15 verschiedenen Umgebungen auswählen. Die entsprechende Auswahl geschieht per Joystick in Port 1. Die ersten 7 Umgebungen dienen der Übung. Hierbei trifft man nur auf einige feindliche Panzer, während man bei der Wahl zwischen 8 und 15 schon auf feindliche Flugzeuge stößt. Einziger Trost: In den Übungsteilen kommen keine feindlichen „Flaggschiffe“ zum Einsatz. Außerdem kann der Feind Ihre Heimatbasis nicht zerstören. In allen anderen Fällen müssen Sie Ihre Heimatbasis verteidigen oder aber alles aufgeben, was Sie an Jägern und Steuerungscomputern besitzen. Ziel ist es, alles, was sich bewegt und dem Gegner gehört, zu vernichten. Am Ende des Spiels können Sie mittels Joystick von der gleichen Situation aus oder durch irgendeinen Tastendruck neu beginnen. Drückt man nach Auswahl der Rangstufen den Feuerknopf, beginnt das aufregende Abenteuer als Weltraumfuchs. Der erste Weg führt auf die Abschubrampe. Wie hoch die Abschubgeschwindigkeit sein wird, hängt von Ihrer jeweiligen Auswahl zu Beginn des Spiels ab. Von nun an verwandelt sich Ihr Joystick in den wichtigsten Steuerknüppel für Seiten- und Hö-

henleitwerk. Selbst ein „Nachbrenner“ ist im Skyfox vorhanden, der die MACH 1 in Sekundenschnelle erreichen läßt! Hat man den Feind erst einmal auf dem Radarschirm, der sich zwischendurch als nützliche Hilfe aufrufen läßt, haben Sie die Entscheidung zu treffen, welche Waffen jetzt eingesetzt werden, um den Sieg zu erreichen. An Bord sind: Laser-Kanonen, Infrarot-Raketen und ferngesteuerte Lenkraketen. Sollten Sie in der Aufregung und Hektik durch den Feind einmal eine Rakete zu früh scharf gemacht haben, besteht selbstverständlich die Möglichkeit, diese wieder zu sichern, um eine evtl. Selbsterstörung zu verhindern. Bei einer totalen Erschöpfung können Sie sogar den Autopiloten einschalten, dieser wird Sie sicher und heil wieder zur Heimatbasis zurück führen. Abschließend kann man sagen: Verlieren Sie nicht die Übersicht über die reichhaltigen Instrumente. Vielleicht kommen Sie sogar das

erste Mal in den Genuß, mit einer Geschwindigkeit von stolzen MACH 4,8 (4.800 Km/h) zu fliegen.

Hotel

Datenträger: Diskette
Rechnertyp: C-64 / Atari 800 XL
Vertrieb: Bertelsmann-Software
GmbH, Abt. Ariolasoft
Preis: DM 79,-

Träumen auch Sie, einmal Super-Hotelier zu sein wie Conrad Hilton? Dann ist das Spiel „HOTEL“ genau das richtige für Sie. Haben Sie Hotelkaufmann gelernt oder nicht? Das ist hier die Frage! — Sehen Sie, Conrad Hilton ist nur deshalb Super-Hotelier geworden, weil er wußte, was Hotelgäste erwarten und worauf es bei ihnen ankommt. Nicht daß Sie jetzt denken, Conrad Hilton hätte sich ins gemachte Nest gesetzt. Nein, er hat ganz klein angefangen, genauer gesagt: mit nichts.

Ob Sie es schaffen, Super-Hotelier zu werden, liegt natürlich ganz bei Ihnen. Nicht daß Sie denken, dies ginge einfach so, aus dem linken Ärmel. Nein, „HOTEL“ wird mit mindestens zwei Mann, höchstens mit vier, gespielt. Da es ja sonst wohl kaum ein Wettbewerb ist.

Apropos Wettbewerb: Wissen Sie eigentlich, was bei „HOTEL“ alles auf Sie zukommt? Das reinste Streß-Management. Wenn ich an all die Kleinigkeiten denke: Aschenbecher, Bibel, Taxidienst und Prospekte. Und dann z.B. das Personal, der Service, die Auslastung der Zimmer usw....! Sie bestimmen beim Start des Spiels, welche Position Ihr Ziel sein wird. Ob Sie vom Pensionsbesitzer zum Drei-Vier- oder Fünf-Sterne-Hotel aufsteigen möchten oder sich in eine satte Pleite stürzen werden. Am besten, Sie behaupten sich einfach mal mit „HOTEL“!



Das MSX-Forum signalisierte den Totalangriff auf breiter Front

Der neue Computerstandard auf der IFA

Fast zwei Jahre lang plätscherte MSX müde vor sich hin. Literatur und Software waren Mangelware. Doch das soll jetzt ganz anders werden. Auf einem gemeinsamen Forum während der Internationalen Funkausstellung in Berlin wurde zum Angriff auf den deutschen Markt geblasen. COMPUTER TEAM-Mitarbeiter Dieter Hurcks berichtet.

Einziges Weltstandard

Kay Nishi vom Softwareentwickler Microsoft gilt als Erfinder des MSX-Standards. Microsoft, das 1975 das BASIC für die ersten und später für viele weitere 8080-Computer kreierte, entwickelte auch das leistungsstarke MSX-BASIC. Für den neuen Standard sei der Z 80 ausgewählt worden, weil damit auf Maschinensprachniveau der Zugriff zu den verschiedensten Computern möglich sei. Damit auch Kompatibilität bei den höheren Programmiersprachen möglich wurde, entstand der MSX-Standard.

Ausbaufähiges System

Neben der Software-Kompatibilität zählt die Möglichkeit der Hardware-Erweiterung zu den Stärken von MSX. Jedes Gerät eines jeden MSX-Anbieters kann mit jedem anderen kombiniert werden. Wer sich einen neuen Computer kauft, braucht seine Software nicht mehr wegzuerwerfen. „Hardware-Erweiterung“, sagte Nishi, „umfasst aber auch die Möglichkeit, daß ein Computer mit anderen elektronischen Geräten kommunizieren kann.“



Unser Bild zeigt den Sony HIT BIT

Neue Entwicklungen

Die Geräte der zweiten Generation werden noch leistungsfähiger sein durch neuentwickelte Chips (mehr dazu in einem Extra-Bericht). Diese sogenannten MSX 2-Computer sind von ihren technischen Möglichkeiten her bereits für den professionellen Einsatz geeignet. Künftig werden Audio, Video, TV, Bildplatte, CD und die Telekommunikation noch enger mit dem Computer verwachsen. Nishi: „Deshalb ist es ein großer Vorteil, wenn Firmen, die Audio oder Video produzieren, auch Computer bauen.“ In Japan sei dieses Konzept schon aufgegangen. Dort habe MSX auf dem Homecomputermarkt bereits einen Anteil von über zwei Dritteln. Der millionste verkaufte Computer wird dort für Weihnachten erwartet.

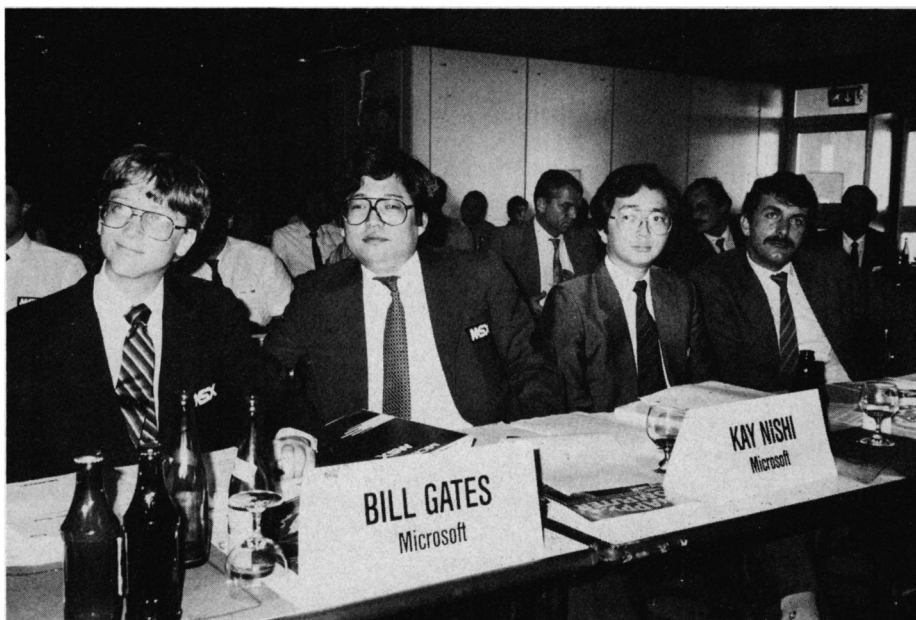
MSX-Vormarsch in Europa

Philips als europäischer Anbieter sieht in MSX die treibende Kraft seines neuen Medienkonzepts. Frank Lech aus der Firmenzentrale in Eindhoven nannte als Zielgruppe für New Media Systems (NMS) kleinere Gewerbebetriebe, Schüler und Studenten. Wichtig sei vor allem die leichte Handhabung der Hard- und Software. Ein Standard ermögliche zudem den Austausch von Software ohne Probleme.

Tim Cole von der deutschen MSX-Entwicklungsgruppe in Fellbach zeigte die bereits geleisteten Schritte zur Etablierung des Systems in Europa auf. Allein in Großbritannien hätten bereits 140 Softwarehäuser an MSX-Programmen gearbeitet, noch ehe die ersten Geräte auf den Markt kamen.

Katalog mit 300 Titeln

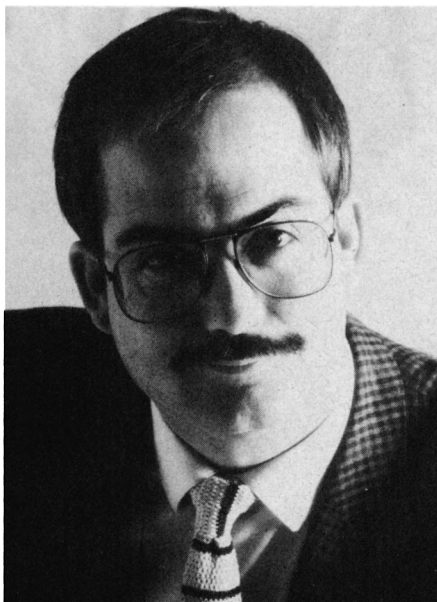
Cole: „Die deutsche MSX-Arbeitsgemeinschaft, auf der diesjährigen Hannover-Messe ins Leben gerufen, hat jetzt einen Katalog mit mehr als 300 Softwaretiteln herausgegeben, die alle nach und nach auf den Markt kommen. Der Katalog wird an Händler und Konsumenten kostenlos verteilt.“ In Frankreich, Belgien, der Schweiz und anderen Ländern sind ebenfalls AGs gegründet worden. Bill Gates von Microsoft/USA: „MSX passiert heute, weltweit. Der Standard wird den Horizont für Heimcomputer erweitern – heute und zukünftig.“



Die MSX-Konferenz auf der Funkausstellung

Superspiele aus Japan

Das japanische Softwarehaus Konami, seit 16 Jahren in Fernost im Geschäft, wird in Europa nur bei MSX aktiv. Wie Deutschland-Sprecher Peter McKenna gegenüber COMPUTER TEAM ankündigte, kommen jetzt eine ganze Reihe grafisch sehr anspruchsvoller, aber dennoch leicht erlernbarer Spiele heraus. Über die Sportspiele von Konami wird COMPUTER TEAM in einem späteren Heft ausführlich berichten. Alle Programme sind auf Cartridge. McKenna: „Dadurch haben wir keine Probleme mit illegalen Kopien.“ Konami-Spiele sind keine Billigware. Aber: „Spielfans,



Unser Bild zeigt Tim Cole

das haben Umfragen ergeben, sind bereit, ein teureres Spiel zu kaufen, wenn es gut ist.“ Damit kein Mangel an Nachschub herrscht, arbeiten 250 Entwickler in Japan ständig an neuen Programmen.

Schritt zur einfacheren Handhabung.

Anwender-Programme

Buchhaltung, Rechnungsschreibung und Lagerverwaltung – das sind drei von fünf Anwenderprogrammen aus dem Hause Philips. Dort gibt's auch schon ein Grafikprogramm mit Maus. Ein weiterer Softwareschub ist aus Holland zu erwarten. Dr. Guurt A.Kok von Aackosoft kündigte eine ganze Palette nützlicher Programme an. Kok: „Ein Geschäftsmann, der sich irgendwann einen größeren Computer kauft, verliert dank MSX nicht seine gesamten Daten.“ Datenbank, Textverarbeitung und Kalkulation mit interaktiver Datenstruktur bringt Aackosoft heraus.

Wachstum durch Weiterbildung

Vor allem im Bereich Weiterbildung sehen die MSX-Anbieter große Wachstumschancen. 14 Millionen Studenten, 11 Millionen bildungswillige Erwachsene und 575.000 Schulklassen in den deutschsprachigen Ländern seien ein Potential, so Felix Irninger von Intus Lernsysteme, das Anlaß zu Optimismus gebe. Irninger: „Amerikanische Experten prophezeien, daß der Computer in gut 20 Jahren Schulklassen und Bücher abgelöst haben wird.“ In den USA bestünden schon heute zwei Drittel der verkauften Software aus Lernprogrammen. Irninger: „Keiner kann so billig unterrichten wie ein Computer.“

Identische Befehle

IDS – wieder ein neues Kürzel, aber eines, das man sich merken sollte. IDS besagt ganz einfach, daß künftig alle Anwenderprogramme mit identischen Befehlen arbeiten. Die gleiche Taste bewirkt also bei Kalkulation oder Textverarbeitung immer das gleiche. Ein wichtiger

Programmier-Hilfen

Die letzte Gruppe der Computeranwender sprach während des MSX-Forums Andreas Müller-Hermann von der RVS Datentechnik München an: die Programmierer. Jeder zweite Anwender, sagte er, „verwendet seinen Computer zum Programmieren oder zum Erlernen einer Programmiersprache.“ Hierzulande sei die Gesell-

schaft allerdings gegenüber dem Programmieren eher skeptisch, was die Ausbreitung doch behindere. Die RVS werde durch ein Angebot an Compilern und Interpreten den Anwendern Hilfen an die Hand geben, selbst zu programmieren. Diese seien auf den Privat-User zugeschnitten und weniger kompliziert zu handhaben als die bisher üblichen. Ein neuer Macro-assembler für die Programmiersprache Forth werde in Kürze auf den Markt kommen. Müller-Hermann: „Die MSX-Hardware ist das ideale Vehikel für neue Produkte im Bereich der Compiler und Interpreter. Und die werden sicherlich der Computerwelt neue Impulse bringen.“

Einige zögern noch

Nach über zwei Stunden konnte man den Eindruck gewinnen, daß MSX von einer einheitlichen Front vorangetrieben wird. Auch wenn einige der MSX-Mitstreiter sich noch zurückhielten, deuten Namen wie Sony, Philips, Panasonic und Sanyo darauf hin, daß hier ein Pflänzlein gesetzt worden ist, das eines nicht fernen Tages den gesamten Homecomputer-Markt dominieren könnte. Vor dieser wirtschaftlichen Power wird manch anderes Unternehmen weichen müssen.

Grafik der Spitzenklasse

Die sowieso schon hervorragende Grafik des MSX-Standards wird weiter verbessert. MSX 2 erhält einen neuen Videochip von Yamaha, der für eine Farbauflösung von 512x212 Punkte sorgt — ein toller Wert. Bei Philips zeigte der VG 8240, der mit einer verstellbaren Tastatur ausgestattet ist. Was damit möglich ist? Zum Beispiel die Darstellung von 256 Farben gleichzeitig.

Digitalisierte Bilder

Mit einem eingebauten Grafik-Superimposer können Videobilder digitalisiert werden. Die tolle Auflösung verwischt dabei fast den Unterschied zum Foto. Für diese Leistung ist der VG 8240 mit 192 K RAM und 64 K ROM ausgestattet. Ein 3 1/3-Zoll-Laufwerk (inzwischen MSX-Standard) ist integriert, ein zweites Laufwerk kann direkt angeschlossen werden. Ein- und Ausgänge über Scartbuchsen ermöglichen fast unbegrenzten Ausbau des Systems.

80 Zeichen darstellbar

Textverarbeiter werden über die 80-Zeichen-Darstellung erfreut sein, die jetzt ohne Umwege möglich ist. Als externen ROM-Speicher zeigte Philips am IFA-Stand einen CD-Player. Auf der Silberscheibe war der komplette Katalog der zur Zeit angebotenen (rund 5000) Compact Discs abgelegt. Das Speichermedium der Zukunft für langlebige Daten. Kapazität: 600 Megabyte. Das sind 150 000 oder 15 Meter gestapelte Schreibmaschinenseiten voll Text.

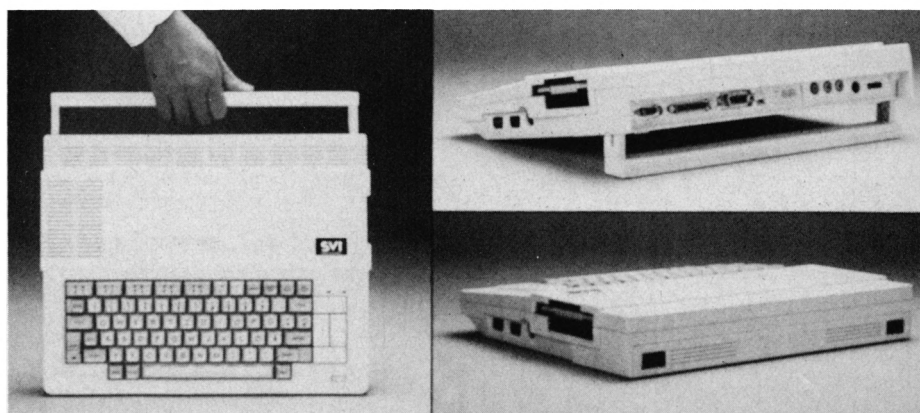
Trend zu höherer Leistung

SVI will bereits im Frühjahr seinen 128 K-Rechner auf den Markt bringen, den 738 X'PRESS. Der SVI 728 wurde jetzt im Preis gesenkt, von 1000 auf 700 DM. Auch die anderen Anbieter gehen herunter. Neue Modelle am Horizont, der Druck der Konkurrenz sowie höhere Stückzahlen und der gefallene Dollarkurs machen's möglich. URC

MSX II Einführung 1986

Erweiterung nach oben

Wer nun meint, die jetzt angebotenen MSX-Computer gehörten schon bald zum alten Eisen, der ist auf dem Holzweg. MSX 2 ist nämlich lediglich eine Erweiterung des Systems nach oben hin, in den semiprofessionellen Bereich. Sicher: Die neue Generation leistet mehr, aber sie wird natürlich auch mehr kosten. COMPUTER TEAM-Mitarbeiter Dieter Hurcks hat sich auf der Funkausstellung die MSX 2-Prototypen angeschaut.



Der tragbare SVI 738; fast schon MSX II

```

IS-BASIC Program 0
30 LET X=640:LET Y=360
40 FOR R=1 TO 255
50   SET INK R
60   LET A=X-R-220:LET A1=Y-R-50
70   LET C=X+R+220:LET C1=Y+R+50
80   PLOT A,A1;A,C1;C,C1;C,A1;A,A1
90   PRINT R
100  NEXT
110  FOR BALL=1 TO 100
120    CALL FIREBALL(256,X,Y)
130  NEXT
140  DEF FIREBALL(COLOURS,A,B)
150    SET LINE MODE 3
160    SET INK RND(COLOURS)
170    FOR GO=1 TO 2
180      FOR AROUND=1 TO 650 STEP 30
190        PLOT A,B,ELLIPSE AROUND,
          AROUND
200      NEXT
210    NEXT
220    SET LINE MODE 0
230  END DEF
ok

```

Strukturiertes Programmieren mit ENTERPRISE-BASIC

BASIC ist nicht gleich BASIC. Und BASIC ist nicht mehr das, was es einmal war, nämlich eine Sprache zum Produzieren von Spaghetti-Code. Den Vorwurf, daß alle, die mit BASIC das Programmieren erlernt haben, für immer verdorben seien, habe ich schon oft gehört. Mit der Begründung, daß diese „Volksprogrammiersprache“ nur dazu verführen würde, mit GOTOs wild im Programm umherzuspringen und damit nur unübersichtlichen Code zu erzeugen, wurde BASIC mit Recht schon oft in den Boden gestampft. Doch heute gibt es Dialekte dieser Einsteigersprache, die strukturiertes und überschaubares Programmieren unterstützen.

Was nützt es, wenn ich ein Programm schreibe, das nach einem halben Jahr nur noch mit größter Mühe erweiter- oder veränderbar ist, weil keiner mehr „durchblickt“. Da steht in Zeile 1230 ein GOTO 9870, und in Zeile 930 steht GOSUB 5120. Und jedesmal muß man erst einmal nachsehen, was dort überhaupt gemacht wird. Von Übersichtlichkeit kann da gar keine Rede sein. Nun gibt es aber BASIC-Dialekte, bei denen man ganz auf GOTOs und GOSUBs verzichten kann, weil andere (bessere)

Ablauf-Steuer-Befehle angeboten werden, etwa nach Pascal-Art.

Warum nicht direkt in Pascal programmieren? Ganz einfach! BASIC hat für Anfänger und während der Programmentwicklung einen entscheidenden Vorteil. BASIC ist in den meisten Fällen eine Interpretersprache, d.h. die lästige Compilerzeit während der Programmbearbeitung entfällt. Die Abarbeitungsgeschwindigkeit spielt nur eine untergeordnete Rolle. Außerdem stehen oft auch BASIC-Compiler zur Verfügung, die einem ferti-

gen Programm zu einer höheren Geschwindigkeit verhelfen können.

Einer der neuen Dialekte ist das IS-BASIC des Enterprise. Doch was ist hier neu? Wenn man in „normalem“ BASIC strukturiert programmiert, verwendet man GOTO und IF/THEN GOTO nur zum Erzeugen von Schleifen oder zum Steuern von bedingten Verzweigungen. Das geht natürlich, es besteht aber die Gefahr, daß man mit GOTO „über Kreuz“ springt. Schleifen dürfen aber nur die Form von ineinander

Bild 1:

```

120 REM Schleife 1
.
.   Anweisungen
.
230 IF Bedingung1 THEN
.   510
.   Anweisungen
.
360 GOTO 120
370 REM Schleife 2
.
.   Anweisungen
.
470 IF Bedingung2 THEN
.   120
.   Anweisungen
.
680 GOTO 370
    
```

bedingung oben und die Schleife mit Abbruchbedingung unten. Alle werden vom ENTERPRISE-BASIC unterstützt. Die allgemeine Form lautet DO/LOOP, der Abbruch erfolgt mit WHILE oder UNTIL. Steht die Bedingung unten, wird der Schleifenkörper, so nennt man das Innere einer Schleife, mindestens einmal durchlaufen. Steht sie oben, wird zuerst die Bedingung überprüft. Die drei Formen sehen Sie in Bild 2.

Die Zeilen 20 bis 170 bilden eine Endlosschleife, die nicht verlassen werden kann. Zwischen den Zeilen 40 und 120 läuft eine Schleife mit Abfrage oben. Sie wird durchlaufen, solange (WHILE) die Bedingung erfüllt ist. In Schleife 130 bis 150 ist die Abfrage unten. In dem hier gezeigten Beispiel wird solange nichts getan, bis (UNTIL) die Taste „Y“ gedrückt wird. Der Schlei-

Rechners frei Haus, d.h. voll automatisch.

Ein weiterer wichtiger Aspekt beim strukturierten Programmieren ist die Konstruktion von Zwei- und Mehrfachentscheidungen. Die Ausgänge der einzelnen Wege müssen auf der gleichen Ebene liegen, d.h., „wildes“ Springen ist verboten. Die Struktur sollte wie folgt aussehen: Wenn die Bedingung erfüllt ist, bearbeite Block 1, wenn nicht, bearbeite Block 2. Nach Durchlaufen von Block 1 oder 2 muß an der gleichen Programmstelle fortgefahren werden. In den Beispielen (Bild 3) ist der gemeinsame Knotenpunkt die Kommentarzeile „Weiter“.

Bei Mehrfachverzweigungen ist analog zu verfahren. Im IS-BASIC gibt es dafür aber einen besonderen Befehl, die SELECT-CASE-Anweisung (Bild 4). In Zeile 30 wird

Bild 2:

10	! ENTERPRISE-BASIC	10	REM Normales BASIC
20	DO	20	REM Anfang der
30	! Anfang der Endlosschleife	30	REM Endlosschleife
40	DO WHILE Bedingung	40	IF NOT Bedingung THEN 110
50	! Anfang der Schleife	50	REM Anfang der Schleife
60	! mit Bedingung oben	60	REM mit Bedingung oben
.	.	.	.
.	Anweisungen	.	Anweisungen
.	.	.	.
100	! Ende der Schleife	100	REM Ende der Schleife
110	! mit Bedingung oben	110	REM mit Bedingung oben
120	LOOP	120	GOTO 40
130	DO ! Anfang der Schleife	130	REM Anfang der Schleife
140	! mit Bedingung unten	140	REM mit Bedingung unten
150	LOOP UNTIL INKEY\$="Y"	150	IF INKEY\$<>"Y" THEN 130
160	! Ende der Endlosschleife	160	REM Ende der Endlosschleife
170	LOOP	170	GOTO 20

oder nebeneinander gestellten Schachteln haben. Mitten aus einer Schleife heraus- oder mitten in eine Schleife hineinspringen, ist beim strukturierten Programmieren tabu. Eine Konstruktion wie in Bild 1 ist verboten.

Es gibt mehrere Arten von erlaubten Schleifen: Die Endlosschleife, die Schleife mit Abbruch-

fenkörper besteht nur aus Kommentaren, die beim ENTERPRISE mit einem Ausrufezeichen (!) beginnen.

Beim ENTERPRISE-Listing ist zu erkennen, daß die einzelnen Strukturebenen entsprechend eingerückt sind. Diesen Service, der die Lesbarkeit enorm verbessert, liefert der Editor bzw. der Lister des

festgestellt, nach welchem Ausdruck entschieden werden soll, hier ist das einfach die Variable A\$. In den Zeilen 40, 60 und 80 wird der festgelegte Ausdruck mit einem weiteren verglichen, hier sind das die Strings „A“, „B“ und „C“. Bei Gleichheit wird der entsprechende Block ausgeführt, d.h., hier, wenn A\$=B" ist, wird der Block 2

Bild 3:

10	! ENTERPRISE-BASIC	10	REM Normales BASIC
20	IF Bedingung THEN	20	IF Bedingung THEN 50
30	! Block 1	30	REM Block 2
40	ELSE	40	GOTO 60
50	! Block 2	50	REM Block 1
60	END IF	60	REM Weiter
70	! Weiter		

abgearbeitet. Wenn keine Übereinstimmung erkannt wird, z.B. $A\$=X$, trifft Zeile 100 in Aktion. Sie bewirkt dann die Bearbeitung von Block 4. Egal welcher Block durchlaufen wird, die weiteren Programmschritte beginnen in Zeile 130. Um den gleichen Effekt in normalem BASIC zu erreichen, muß von einem IF zum anderen gesprungen oder mehrere IF/THEN/ELSE verschachtelt werden.

In den Beispielen ist der Sinn für die dem Programmierer auferlegte Disziplin nur schwer zu verstehen, da es sich um gut überschaubare Programmstückchen handelt. Bei komplexeren Programmen wird die Lesbarkeit aber enorm verbessert. Durch sinnvolle Kommentare übrigens auch. Das Programmieren selbst wird ebenfalls vereinfacht. Wie deutlich zu erkennen ist, muß man sich beim ENTERPRISE-BASIC nicht um Zeilennummern kümmern. So auch nicht beim Sprung in ein Unterprogramm. Der GOSUB-Befehl wird durch CALL ersetzt. CALL ist in der Lage, Unterprogramme mit ihrem Namen aufzurufen. Der Name eines UPs wird mit DEF festgelegt (Bild 5). Beim Aufruf können auch Parameter übergeben werden. Der Vorteil dieser Möglichkeit liegt eindeutig darin, daß man bei sinnvoll gewähltem UP-Namen schon am Aufruf erkennen kann, was das Unterprogramm macht. Bei vorhandener Parameterliste sieht man auch gleich, welche Werte (Variablen) vom UP be- oder verarbeitet werden.

Parameter-Liste

Ebenso wie Prozeduren können beim ENTERPRISE Funktionen frei definiert werden. Das folgende Beispiel legt Y als eine Funktion von X fest:

```
DEF Y(X)=(SIN(X)+COS(X))/X
```

Im Programm könnte der Aufruf nun so aussehen:

```
A=Y(0.5)
```

Dabei ist es völlig egal, wo im Programm die Definition steht. Zugunsten der Überschaubarkeit sollten jedoch alle Definitionen am Programmanfang oder Programmende stehen.

Ein ähnliches BASIC wie der neue ENTERPRISE-Rechner bietet auch der SINCLAIR QL. Ansonsten muß man aber lange suchen. Es bleibt zu hoffen, daß noch mehr Computerhersteller genauso leistungsfähige BASIC-Versionen entwickeln. kw

Bild 4:

10 ! ENTERPRISE-BASIC	10 REM Normales BASIC
20 INPUT A\$	20 INPUT A\$
30 SELECT CASE A\$	30 IF A\$<>"A" THEN 60
40 CASE "A"	40 REM Block 1
50 ! Block 1	50 GOTO 130
60 CASE "B"	60 IF A\$<>"B" THEN 90
70 ! Block 2	70 REM Block 2
80 CASE "C"	80 GOTO 130
90 ! BLOCK 3	90 IF A\$<>"C" THEN 120
100 CASE ELSE	100 REM Block 3
110 ! Block 4	110 GOTO 130
120 END SELECT	120 REM Block 4
130 ! Weiter	130 REM Weiter

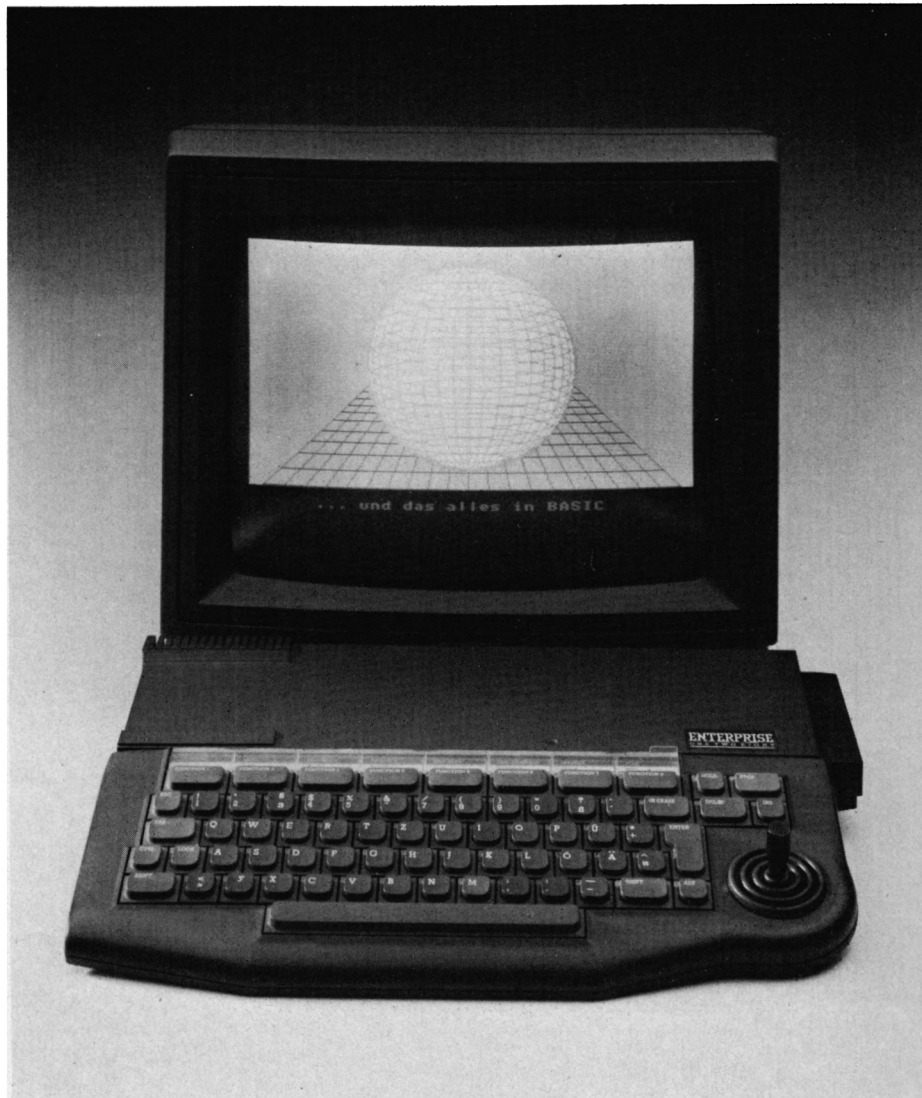


Bild 5:

10 ! ENTERPRISE-BASIC	10 REM Normales BASIC
:	:
:	90 A=10
100 CALL BLANK (10)	100 GOSUB 1000
:	:
:	:
1000 DEF BLANK (ANZAHL)	1000 REM Blanks ausgeben
1010 FOR K=1 TO ANZAHL	1010 FOR K=1 TO A
1020 PRINT " ";	1020 PRINT " ";
1030 NEXT K	1030 NEXT K
1040 END DEF	1040 RETURN

Computer-Zubehör nur vom Fachmann!

Disketten:
 3" Maxell CF2 1 - 9 Stück DM 12,50 10 Stück DM 120,--
 3" Neutral CF2 1 - 9 Stück DM 12,-- 10 Stück DM 117,--
 3,5" Fuji MF100-67,5 tpi 10 Stück DM 80,--
 3,5" Fuji MF100-135 tpi 10 Stück DM 95,--
 3,5" Fuji MF200-67,5 tpi 10 Stück DM 95,--
 3,5" Fuji MF200-135 tpi 10 Stück DM 125,--

3,5" Sony MF100-135 tpi 10 Stück 105,--
 3,5" Sony MF200-135 tpi 10 Stück 135,--

5,25" Neutral einseitig, doppelte Datendichte 10 Stück DM 32,--
 5,25" Neutral zweiseitig, doppelte Datendichte 10 Stück DM 38,--
 5,25" Neut. zwes. dop. Datend. 80 track/96 tpi 10 Stück DM 59,--

Diskettenboxen:
 3" 40er Box ohne Schloß DM 39,--
 3" 40er Box mit Schloß DM 48,--

3,5" 12er Box, aufstellbar DM 21,--
 3,5" 40er Box mit Schloß DM 48,--
 3,5" 50er Box mit Schloß DM 59,--
 3,5" 60er Box mit Schloß DM 65,--

5,25" 10er Box, aufstellbar DM 5,50
 5,25" 60er Box mit Schloß DM 45,--
 5,25" 85er Box mit Schloß (DX85) DM 35,--
 5,25" 90er Box ohne Schloß DM 25,--
 5,25" 100er Box mit Schloß DM 49,--

Neo-Datensatzboxen:
 10er Box mit Schloß DM 40,--
 10er Box ohne Schloß DM 32,--

Wartares Zubehör:
 Druckerstände, Schallschluckhauben, EDV-Möbel, Datensicherungs-schranke, Bildschirmarbeitsplätze, Vortex-Systeme, Data Media, Software usw. Anfragen lohnt sich

Gegen Einsetzung von von DM 3,00 (bar oder Briefmarken) übersenden wir Ihnen unsere gesamten Unterlagen und Preislisten. Händleranfragen erwünscht!
 Alle Preise incl. MWST
 Versand gegen Nachnahme plus Versandkosten oder Vorkasse (versandkostenfrei)

EDV-Zubehör-Großhandel
Erika Effertz
 Goethe Straße 8
 5012 Bedburg
 Tel.: 02272-1088
 und 02274-5693

Schneider CPC 464/664/6128

**Kaufen Sie keine Software
 "Katze" im Sack!!!
 Erst testen, dann kaufen.**

**Achtung!!
 Bei Typ CPC 6128 jeweils 80K
 Arbeitsspeicher frei.**

Superdatei »Pro-Dat«
 Freier Maskengenerator, fast unbegrenzte Kriterieneingabe und Kriterien-Suchroutinen, 40- und 80-Zeichenmodi, Schnittstelle zu »Pro-Text«
 Universaldatei (Bücher, Adressen, Schallplatten, Briefmarken und und und...) plus

Supertextprogramm »Pro-Text«
 Schnittstelle zu »Pro-Dat«. Druckerwahl und Schriftarten nach Wunsch, deutscher Zeichensatz.
 Selbstverständlich:
 Blocksatz, Formatieren, Rechts-links-bündig, Einfügen sofort im Text.
 Abblendung der Steuerzeichen, Suchen/Ersetzen, und und und...

Sie bestellen die Demo-Diskette in Originalverpackung Inklusiv Handbuch per Vorkasse DM 14,50 oder per Nachnahme. Bei Kauf tauschen wir Ihre Demo-Diskette in Originalprogramm-Diskette zum Aufpreis von DM 85,40 plus NN oder 85,40 bei Vorkasse, aus!!!

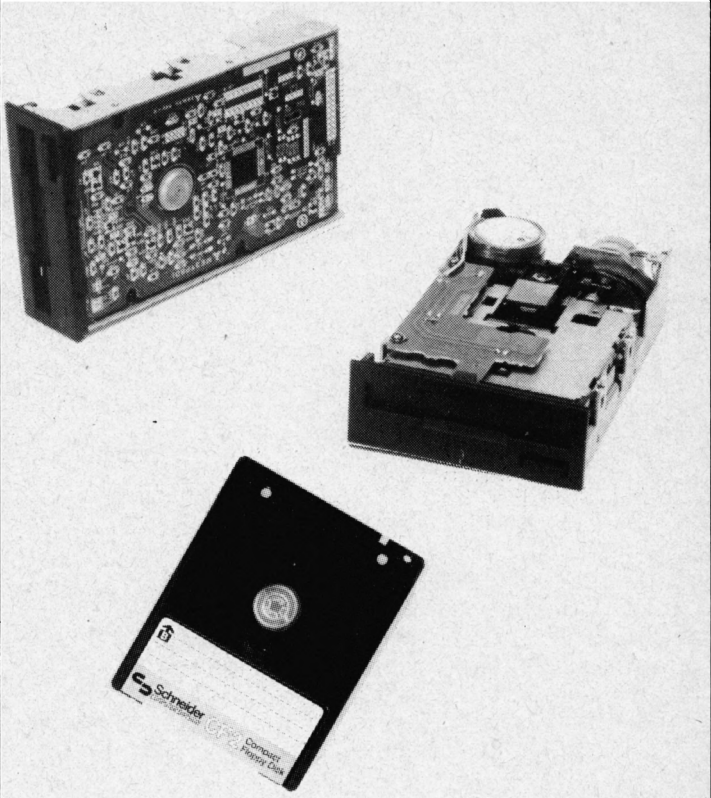
**Kein Risiko, da Sie eine Qualitätsdiskette bei Nichtkauf des Originalprogramms erwerben.
 Auch Kassettenversion möglich.**

Interstate Ltd., Abt. Software
Bestellanschrift:
M. Cordes
Anrather Str. 18
4156 Willich 1
Tel.: 02154/1752

E N D L I C H lieferbar!

Diskettenlaufwerke 3 Zoll mit 1,4 MB formatierter Speicherkapazität.

Die Diskette braucht nicht mehr gewendet zu werden, denn alle Laufwerke sind Slimline 80 Track Doppelkopflaufwerke eines führenden Herstellers.



Wir liefern folgende Ausführungen:

- Einzellaufwerke ohne Gehäuse
428,00 DM (unverbindliche Preisempfehlung)
- Einzellaufwerk als Laufwerk B mit Software zum Betrieb am Schneider CPC 464/664/6128 mit Gehäuse und Netzteil
798,00 DM (unverbindliche Preisempfehlung)
- Einzellaufwerk als Laufwerk A mit Betriebssystem, Gehäuse, Netzteil und CPM 2.2
1198,00 DM (unverbindliche Preisempfehlung)
- Doppellaufwerk mit Betriebssystem und CPM 2.2 komplett betriebsbereit
1698,00 DM (unverbindliche Preisempfehlung)

Wegen der zu erwartenden großen Nachfrage kann es zu kurzen Wartezeiten kommen. Wir werden jedoch bemüht sein, ihre Bestellung innerhalb einer Woche auszuführen.

Schriftliche Bestellung erbeten:

Software Library H. Krohn Mailänder Str. 9
 6000 Frankfurt/M 70

BTX Nr.069/681087 Seite #92059191#
 Telex 4 189 410 ASU Teletext 69 97 316

Wir wünschen allen Lesern ein recht frohes Weihnachtsfest!

Händleranfragen erwünscht. Zwischenverkauf vorbehalten.

**ATARI 520 ST
 4x FORTH**

Ein 32-Bit-FORTH mit Grafikerunterstützung Assembler u. Editor, das unter TOS läuft. Die Vorteile des Systems liegen bei Übersetzungszeiten im Sekundenbereich, übersetzte Programme sind sehr schnell, z. B. werden 100 000 Leerschleifen pro Sek. abgearbeitet. Erweiterungspakete unterstützen das GEM sowie Floatingpoint, das Acceleratorpaket optimiert die Übersetzung noch zusätzlich.

4x FORTH Level 1 **DM 498,-**
 4x FORTH Level 2, GEM, Floatingpoint **DM 750,-**
 Accelerator **DM 398,-**

FORTH-SYSTEME Angelika Flesch

Postfach 1226, 7820 Titisee-Neustadt, ☎ 0 76 51 / 16 65

CPC 464 FORTH

Erleben auch Sie auf Ihrem CPC 464 die Geschwindigkeit eines schnellen FORTH-Compilers mit Turtlegrafik, Editor, Assembler, Tracer und De-Compiler. Dieses System ist im neuesten FORTH83 Standard geschrieben und erzeugt kompakte Programme. Die Grafik ist um Kreis- und Füllbefehle erweitert, das System setzt ebenfalls Windows ein. Das Programm wird mit einem 180-seitigen deutschen Handbuch geliefert.

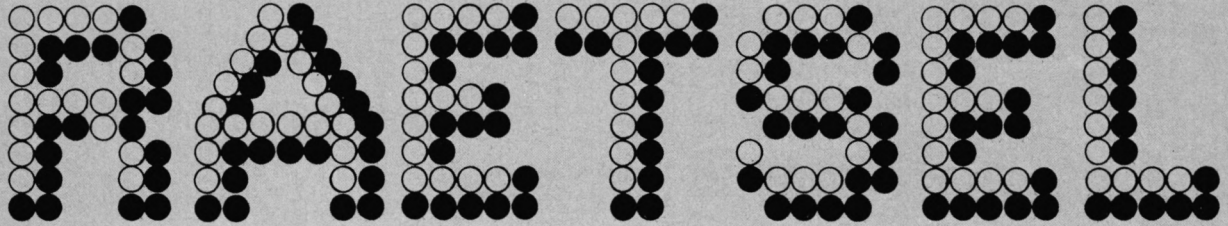
Preis auf Cassette **DM 148,-**
 auf Disk 3", 5,25" **DM 178,-**

Ausführliches Prospektblatt bei:

FORTH-SYSTEME Angelika Flesch

Postfach 1226, 7820 Titisee-Neustadt, ☎ 07651/1665

TEAM



Das Neue Rätsel

Jeder, der einen Home-Computer besitzt, hat doch auch sicherlich ein Spielchen, das zu Beginn erst einmal ein schönes Musikstück ableiert. Da bei den meisten von uns Computer-Freaks der morgendliche Wecker ein ziemlich monotones Geschrei von sich gibt, wäre es doch sicherlich toll, wenn uns der Computer mit Spiele-Musik wecken würde. Die Frage ist nur, wie stellen wir so etwas an? Ge-

Der musikalische Wecker!

sucht wird ein kleines Programm, bei dem man die tatsächliche Zeit und die Weckzeit einstellen muß. Ist die Weckzeit erreicht, sollte der Computer das entsprechende Programm einladen und selbständig starten, damit die Melodie ertönt. (Ghost Busters wäre hierfür gut geeignet.) Viel Spaß bei der Ausarbeitung dieses Mini-Programms wünscht die Redaktion.

Auflösung
aus Heft
11/85

```
100 REM *** HEXA / DEZI ***
110 CLS
120 INPUT "Zahl: ",n$
130 PRINT n$;:l=LEN(n$):n=0
140 FOR x=1 TO l
150 y=ASC(MID$(n$,x,1))-48
160 IF y > 9 THEN y=y-7
170 n=16*n+y: NEXT x
180 PRINT "=" ;n
```

VORSCHAU JANUARHEFT '86

Das ABC der Programmiersprachen

Ein Überblick über Programmiersprachen gibt dieser Artikel

Raubkopieren — Ja oder Nein?

Das Für und Wider wird hier abgewogen

Große Druckerübersicht

Um alle möglichen Drucker geht es in unserer großen Übersicht

GEM-Schnellkurs

Wie man mit GEM umgeht, zeigen wir in der Januar-Ausgabe

Textverarbeitung für Schneider

Welche Textverarbeitung für welchen Schneider und für welche Zielgruppe?

**am Kiosk
oder als Abo per Post**

ab
27.12.
85

Einfache Handhabung – vom PC bis zum Großsystem!

RITEMAN F+

Der kleine Riese mit den vielen Talenten



1.140.-DM

Unverbindliche Preisempfehlung

Das auffälligste Merkmal des Matrixdruckers Riteman F+ ist seine kompakte Bauweise mit der geringen Stellfläche, das Ergebnis eines völlig neuen Druckerkonzepts. Rundherum ein aufgeräumter Drucker, innen wie außen. Das Papier liegt griffbereit unter dem Drucker und Sie legen es von vorne in die verstellbaren Traktoren. Das Papier wird waagrecht zum Druckkopf geführt – einfacher geht es nicht. Etikettenbahnen und Einzelblätter handhaben Sie ebenso leicht.

Unproblematisch ist auch die Papierablage: die Anschlußkabel liegen außerhalb der Papierbahn. Fummeln Sie nicht mehr herum. Der erste Test beim Händler überzeugt Sie. Rite!

Diese kleine Druckstation – kompatibel zu Epson® FX-80 – liefert erstaunliche Leistungen: 105 Zeichen pro Sekunde bzw. 45 Zeilen pro Minute schnell, 96 ASC II-Zeichen, 96 Italic-Zeichen, 9 internationale Zeichensätze, 32 Grafik-Symbole, Puffer und dazu

noch Schönschrift – eben alles was Sie an Ihrem Arbeitsplatz brauchen.

Der Riteman F+ zeigt Ihnen, wie einfach Drucken ist. Nutzen Sie jetzt Ihre Chance: Fragen Sie uns nach Einzelheiten.

C. ITOH

C. ITOH ELECTRONICS GMBH
Roßstr. 96 · 4000 Düsseldorf 30
Telefon: 0211/4 54 98-0 · Telex: 8 584 102

Sprachen schaffen Kontakte. In Urlaub, Freizeit und Beruf.

Lernen Sie entspannt, schnell und erfolgreich. Mit Softlearning.



SOFTLEARNING ist die völlig neue Art der Wissensvermittlung: Durch Anwendung moderner Entspannungstechniken werden während der Lernphase natürliche und anerzogene Lernhemmschwellen abgebaut und dadurch eine erstaunliche Steigerung der Lerngeschwindigkeit, bei gleichzeitiger Erhöhung der Behaltensquote, ermöglicht.

Sprachen sind das Tor zur Welt.



SOFTLEARNING Grund- und Aufbaukurse
Unsere erfolgreiche Sprachkurs-Reihe bietet für jeden den passenden Einstieg. Jeder Kurs vermittelt einen Wortschatz von ca. 1200 bis 1500 Vokabeln.

- ENGLISCH Grundkurs ENGLISCH Aufbaukurs
- MANAGEMENT ENGLISCH
- FRANZÖSISCH Grundkurs FRANZÖSISCH Aufbaukurs
- ITALIENISCH Grundkurs ITALIENISCH Aufbaukurs
- SPANISCH Grundkurs SPANISCH Aufbaukurs

NEU NEU NEU NEU NEU NEU NEU NEU
SOFTLEARNING Zusatzwortschatz

Vokabeltrainer mit je ca. 1000 Vokabeln – kompatibel mit den Übungen der SYSTEMBASIS 'S'.

- ENGLISCH 1 ENGLISCH 2
- ENGLISCH 3
- FRANZÖSISCH 1 FRANZÖSISCH 2

NEU NEU NEU NEU NEU NEU NEU NEU
SOFTLEARNING Wortschatzeditor

Programm zum Erfassen und Trainieren eigener Vokabeln – kompatibel mit den Übungen der SYSTEMBASIS 'S'.

WORTSCHATZEDITOR

Softlearning läuft derzeit auf COMMODORE 64 (128) und ATARI 130 XE (800 XL), in Verbindung mit einem Floppy-Laufwerk und einem handelsüblichen Kassettenrecorder mit DIN-Audio-Ausgang und Anschluß für Start-Stop-Fernbedienung.



NEU NEU NEU NEU NEU NEU NEU NEU NEU NEU

SOFTLEARNING Intensivkurse "Amadeus".
Die idealen Kurzlehrgänge für Urlaub und Freizeit.

- INTENSIV ENGLISCH INTENSIV FRANZÖSISCH
- INTENSIV ITALIENISCH INTENSIV RUSSISCH
- INTENSIV SCHWEDISCH INTENSIV SPANISCH

Entspannung, Erholung und Spaß daran,
so ganz nebenbei eine neue Sprache
zu lernen. Mit Softlearning. Der Lern-
Revolution des 20. Jahrhunderts.

SOFTLEARNING-SPRACHKURSE erhalten Sie bei Ihrem Fachhändler, den Fachabteilungen der Warenhäuser Horten, Karstadt, Quelle, beim Otto-Versand und beim Großversandhaus Quelle.
Nähere Informationen erhalten Sie auch von
SM SOFTTRAINING GMBH, Fasangartenstraße 4, 8000 München 83, Telefon (089) 6380339.

So beurteilen Anwender Softlearning.

Handhabung, Kursaufbau und Bedienung beurteilen von den befragten Personen mit:	
sehr gut	41 %
gut	53 %
befriedigend	6 %
gesamt	100 %

Auf die Frage, ob sie glauben, daß sie mit SOFTLEARNING ihre Lerngeschwindigkeit gegenüber anderen Lernmethoden steigern konnten, antworteten mit:

ja	89 %
nein	0 %
keine Angabe	11 %
gesamt	100 %

* Auswertung der ersten hundert Rückläufe einer Fragebogenaktion 1985

Als Zeitaufwand zum Durcharbeiten eines Kurses wurden angegeben:	
weniger als 30 Std.	11 %
30 bis 40 Std.	24 %
40 bis 50 Std.	65 %
gesamt	100 %