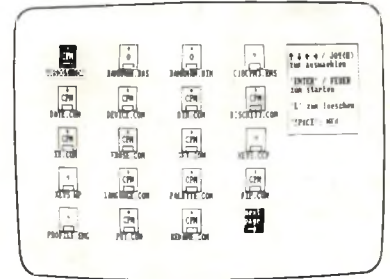


# DAS SONDERHEFT FÜR SCHNEIDER-FANS

SONDER  
PREIS  
öS 80,-  
1/87

**99 GEM<sup>66</sup>**  
auf dem Schneider

**180 Seiten**



**Disc-Operationen  
per Joystick**



## GRAPHIC PROCESSOR ANALOG/DIGITAL-WANDLER

**ANWENDUNGEN:** skyplot, zeichengenerator, word-processor, sprite-editor, apfelmännchen, hyperplot, grafik-macher

**WIE GEHT DAS?** farben erzeugen, grafikkurs, bilder codieren, hardcopies, computerwanzen

**SCHNEIDER**

DAS IST DIE LÖSUNG

**MAD  
GAMES**



Hero of the **GOLDEN** Talisman

WISE GUY EH!

I'VE GOT SECONDS TO GET THE CHECK OUT OF HERE!

BAM

FLOOM

GA

**MAD  
GAMES**

MASTERTRONICS ADDED DIMENSION

MASTERTRONICS

ST

C 64/128

FEATURES VOICE SYNTHESIS

wdh  
typographic  
design

MASTERTRONIC GmbH  
Kaiser-Otto-Weg 18  
D-4770 Soest  
Tel.: (0 29 21) 7 50 28-9

## Impressum

**Herausgeber:** Joachim Günster  
**Chefredakteur:** Joachim Günster  
**Stellvertr. Chefredakteur:** Klaus Weppler

**Autoren:** Klaus Weppler, Frank Thielen, Thomas Barndt, Thomas Binzinger, Beate Lang, Rudolf Petruck, Oliver Welsch, D. Schulze.

**Redaktionsassistentin:** Kornelia Hoffmann

**Titelfotos und -gestaltung:** Professional Photo, Koblenz

**Technische Herstellung:** Oberndorfer Druckerei, Oberndorf/Salzburg

**Datenkonvertierung, Fotosatz:** Dinges + Frick, Wiesbaden

**Vertrieb Handelsauflage:** Verlags-Union, Friedrich-Bergius-Str. 20, 6200 Wiesbaden.

**Manuskripteinsendungen:** Manuskripte und Programm Listings werden von der Redaktion gerne 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 den vom Verlag Joachim Günster herausgegebenen Publikationen und zur Vervielfältigung der Programm Listings auf Datenträger. Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte und Listings wird keine Haftung übernommen.

**Bezugspreis:** Das Einzelheft kostet DM 14,-.

**Urheberrecht:** Alle in dieser Publikation 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 1986 Joachim Günster  
 Anschrift für Verlag, Redaktion, Vertrieb, Anzeigenverwaltung und alle Verantwortlichen:  
 Joachim Günster, Mühlenstr. 12,  
 5431 Boden, 02602/ 60080.

Hallo, liebe Schneider-Fans,

über 170 Seiten Inhalt und 200 KByte Software, das ist unser Sonderheft für Schneider-Fans. Das ist zwar nicht die „Wende“, die Schneider mit ihrem neuen Textsystem Joyce publizieren, aber immerhin ein echt gutes Nachschlagewerk für den Anwender von CPC-Computern. Natürlich haben wir auch Joyce berücksichtigt, mit einem kurzen vorstellenden Artikel. Ansonsten über 200 KByte, wie z.B. das Superprogramm Skyplot zur Darstellung von allen Himmelsausschnitten, Planetenbahnen, Endklassen und Betrachtungswinkel. Die Unterschiede zwischen dem CPC 464 und 664 werden deutlich gemacht und ebenso das 512 Kilobyte-Konzept für die Speichererweiterung des CPC.

Nach dem Motto: Für jeden etwas, beinhaltet dieses Sonderheft für die Hardware-Bastler zwei Super-Projekte, ein 8 Bit-Interface sowie dazu passend einen Analog Digital-Wandler zur Eingabe von Analog-Signalen in den Schneider-Rechner.

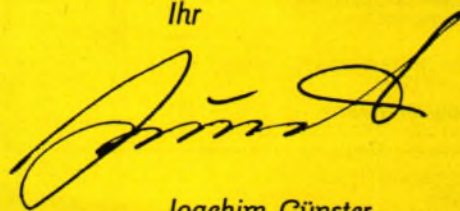
Für die Anwender bieten wir neben einem Zeichengenerator, dem Sprite-Editor und Wordprocessor ein Super-Hardcopy-Programm und vieles andere mehr. Auch die Spieler kommen nicht zu kurz mit unseren Programmen „Das fliegende Auge“ und „Zeno's Turtle“. Allgemeine Tricks und Tips finden Sie im Farbertip, den verborgenen Diskettenkommandos, in Computerwanzen und einigen anderen nützlichen Ratschlägen.

Ausführlich wird behandelt, wie man auf dem Schneider CPC Grafik erzeugt — ein Grafik-Kurs, der von Profis für Anfänger geschrieben wurde. Und unsere Titel-Story? Nun, dazu brauchen wir nicht viel zu sagen, denn GEM, das Oberflächenbenutzungsprogramm, ist sicher jedem bekannt von Rechnern wie Macintosh, von IBM-Rechnern und zu guter letzt von Atari. Aus rechtlichen Gründen haben wir unser GEM, welches ähnliche Funktionen bietet, als Grafik-Desk-Manager, GDM, für Sie abgedruckt.

Zum Schluß möchte ich noch darauf verweisen, daß alle Programme, die dieses Heft enthält, ebenfalls als Datenträger in Kassetten- oder Diskettenform vom Verlag bezogen werden können, das soll Ihnen über 200 KByte abzutippen ersparen.

Ich wünsche Ihnen viel Spaß beim Lesen und verbleibe mit freundlichen Grüßen

Ihr



Joachim Günster  
 Chefredakteur

**ANWENDUNG**

Erforschen Sie den Sternenhimmel!  
**Skyplot** 6  
 zeigt Ihnen den gesamten Himmel oder Ausschnitte und außerdem die Bewegung des Kometen Halley durch die Sternbilder.

**CP/M für Schneider** 47  
 Lesen Sie, wie es z.B. möglich ist, das Betriebssystem zu ändern.

Ein  
**Kalender** 61  
 zwischen 1581 und 2500 auf Ihrem CPC.  
 Vergrößerung und Verkleinerung von Bildern. Wie?

**Vektor-Grafik** 97  
 zeigt es Ihnen.  
 Die Mandelbrotmenge — das komplizierteste mathematische Objekt — erzeugt von einem einfachen Algorithmus.

**Apfelmännchen** 102  
 ... unzählbar viele Muster in den verschiedensten Farben auf dem Bildschirm Ihres CPC 464.  
 Auch über den Centronics-Port kann man Daten empfangen, und zwar über den Busy-Eingang.

**Bildübertragung per Telefon** 112  
 stellt Ihnen ein DFÜ-fähiges Gerät vor.  
 Auf allen drei CPCs können Sie Diagramme mit einer beliebigen Anzahl von Säulen erstellen.

**Säulendiagramme für Ihren CPC** 143  
 Sammeln Sie Mineralien? Das Programm **Mineralien bestimmen mit dem CPC 464** 145 hilft Ihnen, diese zu erkennen und zu ordnen.  
 Das Programm **3D-Plot** 160 erstellt ohne Probleme dreidimensionale Funktionen auf Ihrem Bildschirm.  
 Ein Programm zur Darstellung von Funktionen in einem Koordinatensystem ist **Hyperplot** 162  
 Eine Hilfe bei Ihrer Programmsuche bietet der **Label-Drucker** 173  
**Atari-Maus am CPC** 178  
 Nun auch an den Schneider-Computern 5 V. Spannung.

**UTILITY**

**Die Unterschiede zwischen CPC 464 und CPC 664** 24  
 ..., die nicht nur das Betriebssystem und BASIC betreffen.  
 Einen Befehlssatz, mit dem man komfortabel alle 256 Zeichen des CPC neu gestalten kann, bietet der **Zeichengenerator** 29  
**Schnelle Farbänderung** 33  
 Wann ändert das Betriebssystem die Farbe?  
 Lesen Sie u.a. über Pull-Down-Menüs, Pointer und die damit verbundene Windowtechnik.  
**Der Wordprocessor I** 35  
 Wie man auf einem CPC, der keine Sprites besitzt, trotzdem mit Sprites arbeiten kann.  
**Der Sprite-Editor** 41  
 zeigt die grafischen Fähigkeiten des CPC.  
**Oben 16 Farben im Bild und gleichzeitig unten 80 Zeichen Text** 63  
 Wie Sie in der Windowtechnik mehrere Bildschirmmodi verwenden können.  
 Machen Sie mehr aus Ihrer Floppy!  
**Verborgene Diskettenkommandos** 74  
 eröffnen neue Möglichkeiten.  
**Dirlist** 77  
 Bedienen Sie Ihre Diskettenstation ebenso komfortabel wie mit GEM.  
**CPC-Speicher optimal genutzt!** 123  
 Mit dem Optimizer für CPC-BASIC-Programme können Sie Speicherplatz sparen.  
 Ein Graphic-Utility mit 20 neuen und nützlichen Befehlen.  
**Graphic-Processor für den CPC 464** 128  
**Struklist** 134  
 In einem Struklist-Ausdruck werden Schleifen eingerückt und alle Zeilennummern rechtsbündig gedruckt.

Wie Sie Steuerzeichen mitten im Text für den Drucker plazieren können, lesen Sie in **Noch bessere Druckersteuerung** 148  
**Hardcopies** 151  
 ...mit einem Unterprogramm, das die notwendigen Druckdaten aus dem Bildschirminhalt erzeugt.  
 Textverarbeitung auf dem CPC  
**Der Wordprocessor II** 155  
 einfach - komfortabel — preiswert  
 Künstlerisch Interessierte kommen auf ihre Kosten.  
**Der Grafikmacher** 168  
 zeigt Ihnen, wie Sie sogar Portraits auf den Bildschirm zaubern können.  
**Die Wanze im Computer** 171  
 Hardcopies von jedem laufenden Programm jetzt auch auf den CPCs.  
**Die Uhr des CPC** 175  
 Fragen Sie Ihren Rechner nach der Uhrzeit.  
 Eine Befehlsweiterung, die die Simulation einer Maus ermöglicht.  
**Maussteuerung mit dem CPC 464** 177

**TEST**

**Das 512 KB-Konzept** 27  
 Lesen Sie über die Möglichkeit der Speichererweiterung für Ihren CPC 464.  
**Präsident 6005** 45  
 Der CPC im Test mit einem Typenradprinter  
**CPC 6128** 58  
 Der neue Schneider unter drei Aspekten betrachtet.  
**Schneider Joyce** 99  
 Ein Textverarbeitungssystem mit CP/M Plus und 256 KByte Hauptspeicher.

**KURS**

Wir bieten Ihnen einen dreiteiligen Grafikkurs an, in dem Sie zunächst mit den Grundlagen vertraut gemacht werden, um sodann in der Lage zu sein, z.B. Kugeln oder dreidimensionale Darstellungen zu zeichnen.  
**Der Grafikkurs** 67  
 für Einsteiger.  
**Der Grafikkurs II** 90  
**Der Grafikkurs III** 137



**HARDWARE**

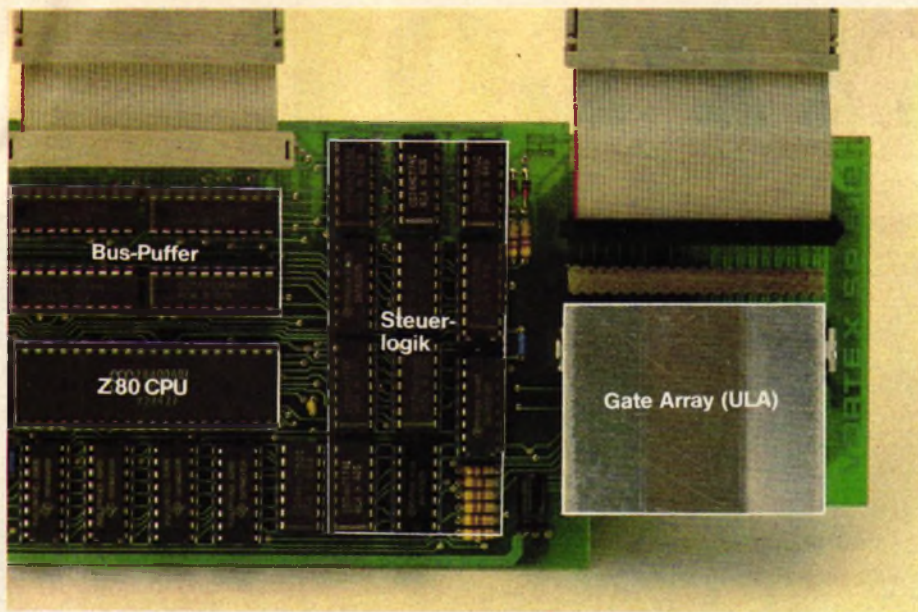
**Parallele Ein-Ausgabeschnittstelle für den CPC 464** 50  
 Eine vollwertige Parallelschnittstelle mit der dazugehörigen Software, die sowohl die Ein- als auch die Ausgabe von 8 Bit-Daten erlaubt.  
**Analoge Meßdatenerfassung mit dem CPC 464** 80  
 Durch eine einfache Schaltung können bis zu 16 analoge Meßwerte vom Computer erfaßt und verarbeitet werden.

**SPIEL**

**Zeno's Turtle** 20  
 Helfen Sie Zeno-boy beim Hüten seiner Schildkröten.  
**Maze** 94  
 Schaffen Sie es, einem dreidimensionalen Labyrinth zu entkommen? Versuchen Sie es.  
**Das fliegende Auge** 108  
 Das Superspiel für Freunde von Actiongames.



Eine 3-teilige Kurs Grafic ... und viele Listings.



### Das 512 Konzept

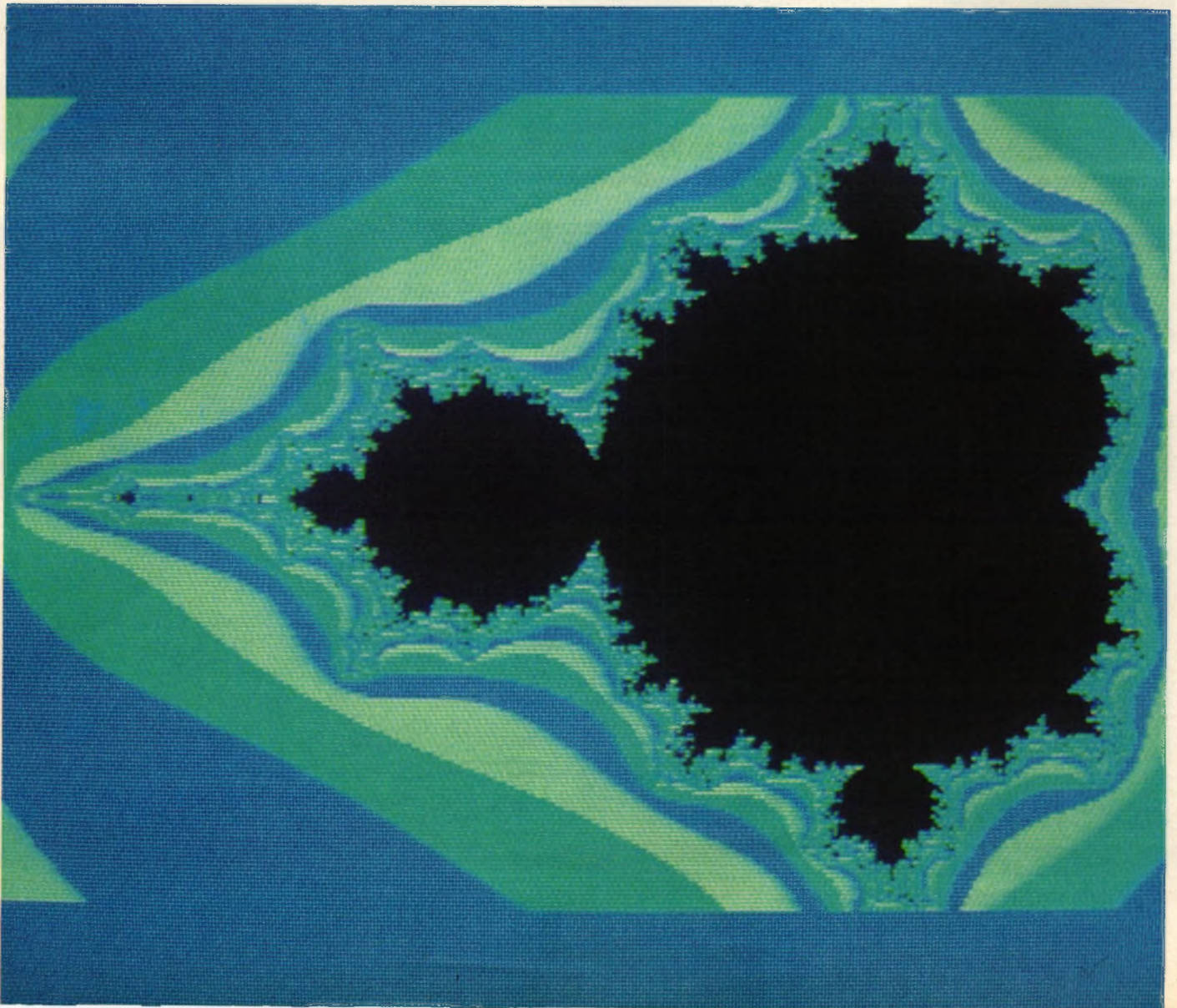
Speichererweiterungen für Ihren CPC

### Apfelmännchen

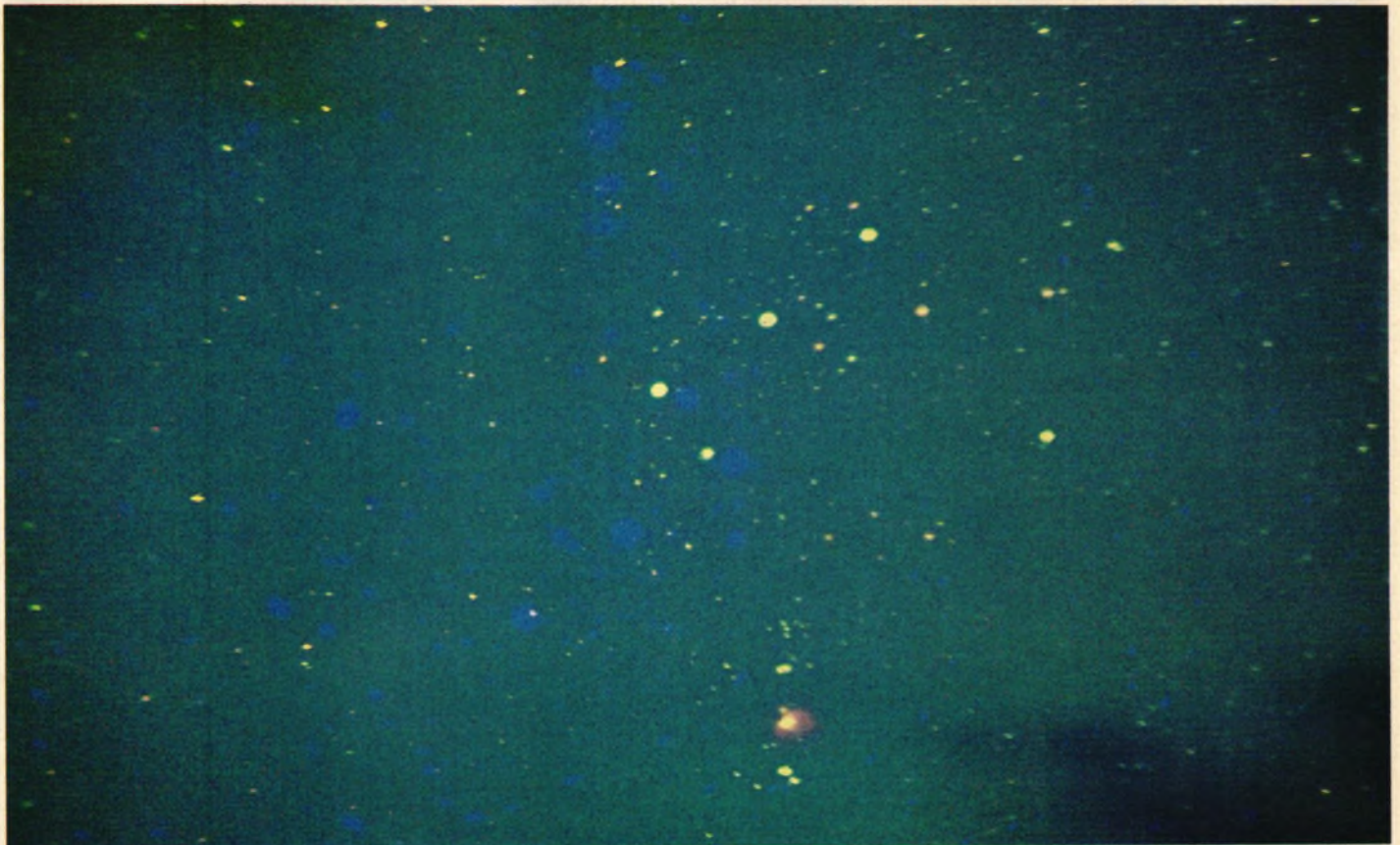
Durch einen einfachen Algorithmus die farbigsten Muster auf Ihrem Bildschirm

## Software — auf einen Blick

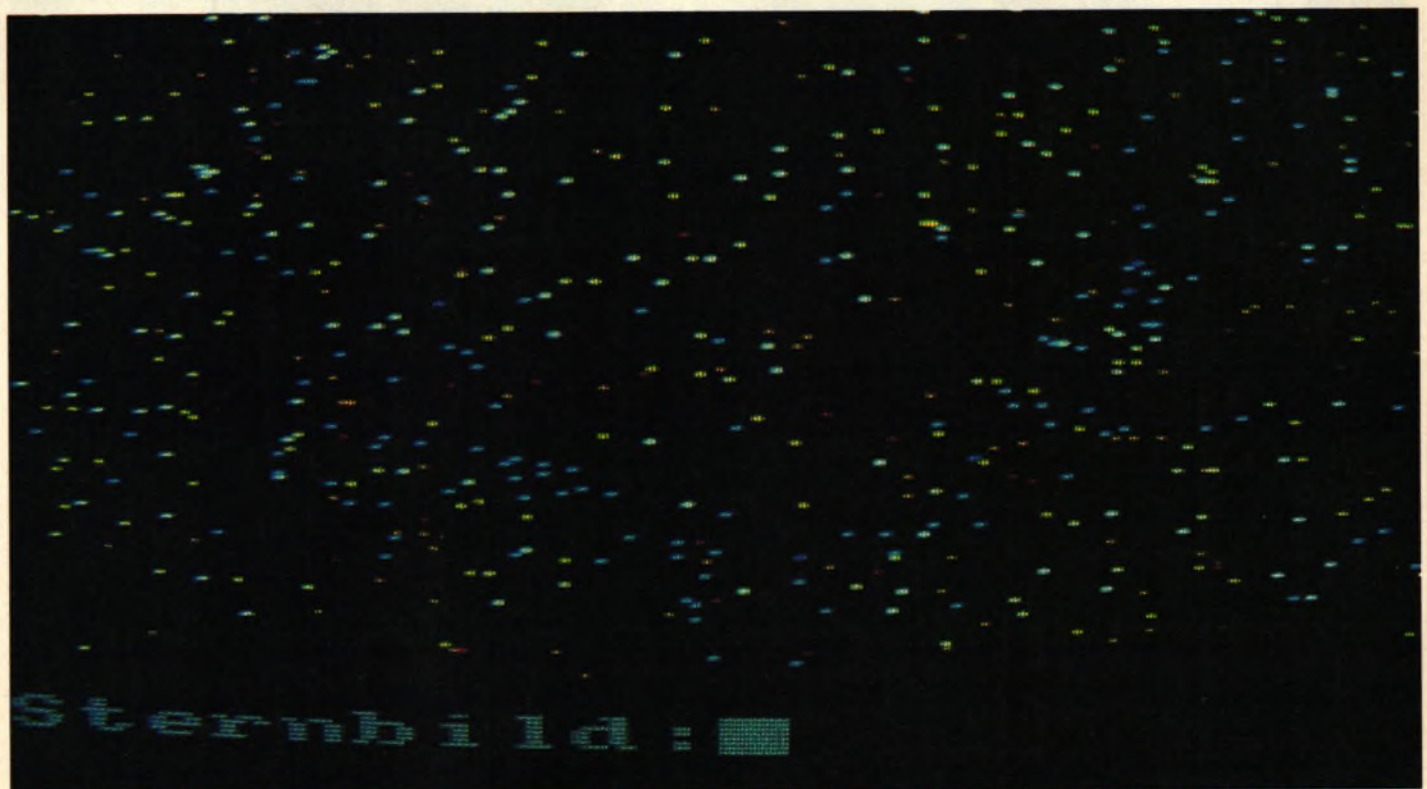
● Skyplot	S. 6
● Zeno's Turtle	S. 20
● Zeichengenerator	S. 29
● Der Wordprocessor	S. 35
● Sprite-Editor	S. 41
● Kalender	S. 61
● Dirlist	S. 77
● Maze	S. 94
● Vektorgrafik	S. 97
● Apfelmännchen	S. 102
● Das fliegende Auge	S. 108
● Bildübertragung	S. 112
● GDM oder "GEM"	S. 116
● Optimizer	S. 124
● Graphic-Processor	S. 128
● Struklist	S. 134
● Säulendiagramme	S. 143
● Mineralien bestimmen	S. 145
● 3D-Plot	S. 160
● Hyperplot	S. 162
● Wanze im Computer	S. 171
● Label-Drucker	S. 173
● Die Uhr des CPC	S. 175
● Mausteuerung	S. 177



# SKY PLOT



▲ Unser Bild zeigt den Originalhimmel mit dem südlichen Orion mit M42 (großer Orionnebel). Aufgenommen Fujichrome 400,  $f = 135$  mm, Blende 2,8, 600 s



▲ So sieht der Himmel auf dem Schneider aus.

## SKYPLOT

Mit diesem Programm, das auf allen Schneider-Computern (CPC 464, 664 und 6128) läuft, können Sie den Sternenhimmel erforschen. Es zeigt als Sternkarte den gesamten Himmel oder vergrößerte Ausschnitte. Außerdem kann der an einem beliebigen Ort auf der Erde sichtbare Himmelsausschnitt zu einem beliebigen Zeitpunkt dargestellt werden. Gerade jetzt zu Beginn des Jahres ist ja der Komet Halley wieder sichtbar; ein Ereignis, auf dessen nächste Wiederkehr Sie bis zum Jahr 2062 warten müssen. Mit der Hilfe dieses Programms können Sie seine Bewegung durch die Sternbilder verfolgen und die günstigsten Beobachtungszeitpunkte herausfinden.

### Planeten — die Geschwister der Erde

Am Himmel sind nicht nur Sterne sichtbar, sondern auch die Planeten des Sonnensystems, von denen Merkur, Venus, Mars, Jupiter und Saturn schon leicht zu beobachten sind. Sie gleichen auf den ersten Blick hellen Sternen und sind für den Laien nicht als Planeten zu erkennen. Sie bewegen sich aber durch die Sternbilder (wie die Sonne auf ihrem Lauf durch die Tierkreiszeichen), und sind bei einiger Kenntnis der Sternbilder leicht daran zu erkennen, daß sie an einer Stelle stehen, wo gar kein Stern stehen sollte. Wenn Sie also einen hellen vermeintlichen Stern sehen, der Ihnen unbekannt vorkommt, kann Ihnen SKYPLOT leicht Auskunft geben, indem Sie die Beobachtungszeit und Ihren Ort eingeben. Dann wird der gerade sichtbare Himmel mit den Planeten abgebildet, und Sie können das Objekt identifizieren.

Umgekehrt kann Ihnen das Programm zeigen, wann Sie etwa Merkur oder Venus beobachten können, da diese Planeten (besonders Merkur, der nahe an der Sonne steht) nur zu gewissen Zeiten und auch nie die ganze Nacht über zu sehen sind. Sie können dann den Teil des Himmels, in

```

5 DEG
8 DIM x%(611),y%(611),p(20)
11 DEF FN fr(x)=x-INT(x)
13 DEF FN dm(x)=SGN(x)*(INT(ABS(x))+INT(FN fr(ABS(x)))*60)/100+FN fr(ABS(FN fr(x))*60)*60/1000)
14 DEF FN aus$(x)=LEFT$(STR$(ROUND(FN dm(x),4)),8)
20 DEF FN dg(x$)=SGN(VAL(x$))*(ABS(VAL(LEFT$(x$,LEN(x$)-5)))+VAL(MID$(x$,LEN(x$)-3,2))/60+VAL(MID$(x$,LEN(x$)-1,2))/3600)
30 DEF FN set$(x$)=x$+STRING$(4-LEN(x$)+INSTR(x$,"."),"0")
40 DEF FN as(x)=ATN(x/SQR(1-x*x))
50 DEF FN ac(x)=90-FN as(x)
60 SYMBOL 255,&X1111000,&X11001100,&X11001100,&X1111000,0,0,0,0
70 SYMBOL 240,0,0,0,&X1110000,&X10001000,&X10101000,&X10001000,&X1110000
80 SYMBOL 241,&X10001000,&X1110000,&X10001000,&X10001000,&X1110000,&X100000,&X1110000,&X100000
90 SYMBOL 242,&X1110000,&X10001000,&X10001000,&X10001000,&X1110000,&X100000,&X1110000,&X100000
110 SYMBOL 244,&X1111,&X11,&X101,&X1111001,&X1001000,&X10001000,&X10001000,&X1110000
120 SYMBOL 245,&X1100000,&X10010000,&X10000,&X100100,&X100100,&X1111110,&X100,&X100
130 SYMBOL 246,&X100000,&X11111000,&X100000,&X101100,&X110010,&X110010,&X100100,&X11
140 SYMBOL 247,&X1110000,&X10101000,&X100000,&X1110000,&X10001000,&X10101000,&X10001000,&X1110000
150 SYMBOL 248,0,0,&X10001000,&X10101000,&X10101000,&X1110000,&X100000,&X100000
160 SYMBOL 249,0,&X11110000,&X10001000,&X10001000,&X11110000,&X10000000,&X10000000,&X11111000
170 SYMBOL 250,&X1,&X11,&X110,&X11110,&X1111100,&X11111000,&X11110000,&X1100000
190 SYMBOL 252,&X100110,&X1001001,&X1001001,&X1001001,&X1001001,&X1000,&X1000,&X1000
290 mz=-1:sh=-1:pz=-1:ka$="S":bm=0:GOSUB 6000:day=7:month=7:year=1959:zeit=12.5:phi=51.283333:zeit$="12.30":phi$="51.17":GOSUB 3100:GDSUB 4100
295 MODE 2:BORDER 13:INK 1,0:INK 0,13:PEN 1:PAPE R 0
300 GOTO 5000
330 IF INSTR(x$,".")=0 THEN x$=x$+"."
340 x=FN dg(FN set$(x$))
350 RETURN
1000 REM Einstellung Messier-Objekte
1010 INPUT "Messier-Objekte darstellen (J/N) ";a$:IF UPPER$(a$)="J" THEN mz=-1 ELSE mz=0
1020 INPUT "Sterne nach Helligkeiten oder Spektralklassen darstellen (H/S) ";a$:sh=UPPER$(a$)="H"
1030 INPUT "Planeten darstellen (J/N) ";a$:IF UPPER$(a$)="J" THEN pz=-1 ELSE pz=0
1900 RETURN
2000 REM Einstellen der Himmelsdarstellung
2010 INPUT "Sichtbarer Himmelsausschnitt (S), Polarkarte (P), Aequatorialkarte (Q) oder A

```

dem ein Planet steht, als Karte vergrößert darstellen und die Position in einem Sternbild bestimmen. Ein schönes Beispiel ist der 7. Juli 1959, wo ein äußerst seltenes Himmelsereignis stattfand: Der helle Stern Regulus (der hellste im Sternbild Löwe) wurde damals von dem Planeten Venus bedeckt. Da die Planeten nur sehr kleine scheinbare (d.h. von der Erde aus sichtbare) Durchmesser haben, sind solche Sternbedeckungen sehr selten (meistens laufen die Planeten mehr oder weniger weit an sichtbaren Sternen vorbei). Wie selten solch ein Ereignis ist, können Sie daran ersehen, daß diese Bedeckung zum letzten Mal am 11. 9. 1128 stattfand.

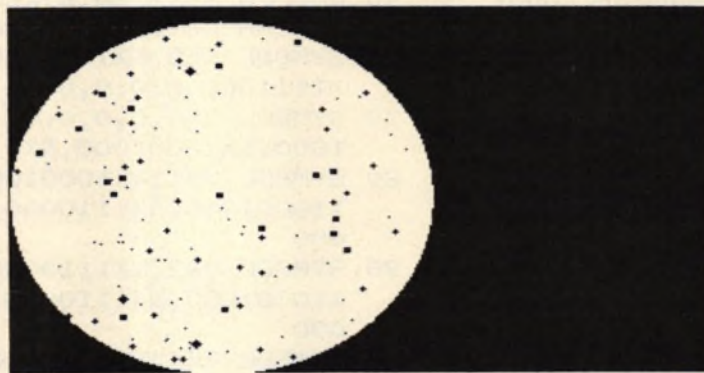
Wenn ein Planet vor einem Stern steht, wird (im Modus 0) ein blinkender Punkt dargestellt. Sie sollten dann jedoch einen Ausschnitt berechnen lassen, da meist nur wegen der mangelnden Auflösung der Planet vor dem Stern erscheint.

## Schleifen am Himmel

Wie schon gesagt wurde, bewegen sich die Planeten scheinbar zwischen den Sternen. Diese Bewegungen kommen dadurch zustande, daß sie selbst und die Erde um die Sonne laufen, und zwar mit unterschiedlicher Geschwindigkeit. Je näher ein Körper an der Sonne ist, desto schneller bewegt er sich. Da die Erde bei bestimmten Gelegenheiten (bei sogenannten Oppositionen oder Konjunktionen) von der Sonne aus gesehen in einer Linie mit dem Planeten steht, überholt sie gewissermaßen den Planeten auf der Innenbahn — oder sie wird von einem Planeten überholt, wenn es der Merkur oder die Venus ist. Normalerweise bewegen sich die Planeten und auch die Sonne zwischen den Sternen in einer Richtung, man nennt dies rechtläufig. Bei den eben angesprochenen Gelegenheiten aber werden die Planeten rückläufig und bewegen sich dann in der anderen Richtung. Da die Planeten außerdem nicht in einer Ebene um die Sonne laufen, entsteht bei einer solchen Rückläufigkeit meist eine Schleife, die durch die scheinbare Bewegung des Planeten zwischen den Sternen entsteht.



Der Sternhimmel vom Nordpol gesehen bis 30° Deklination



Derselbe Sternhimmel wie oben nur ohne Sternbilder

Es können die Bewegungen aller Planeten, der Sonne und des Kometen Halley über beliebige Zeiträume dargestellt werden. Auch die unterschiedlichen Geschwindigkeiten der um die Sonne laufenden Körper werden so deutlich.

## 88 sichtbare Sternbilder

Alle international festgelegten Sternbilder sind enthalten; es können also auch Himmelsansichten des Südhimmels dargestellt werden: z.B. kann der Sternhimmel am Südpol geplottet werden. Damit Sie Sternbilder auch finden und erkennen, kann nach der Berechnung einer Karte oder eines Himmelsanblickes der Name eines Sternbildes eingegeben werden. Die (am Himmel natürlich nicht sichtbaren) Sternbildhilfslinien, durch die die Bilder leichter erkennbar und merkbar werden, werden dann gezeichnet. Dabei kann die international standardisierte Abkürzung des Bildes (bestehend aus drei Buchstaben), der lateinische oder der deutsche Name eingegeben werden. Ist die Eingabe richtig und das Sternbild vollständig auf dem Ausschnitt, so werden die Hilfslinien gezeichnet. Dabei wird auch die Abkürzung,

die lateinische und die deutsche Bezeichnung ausgegeben (z.B. UMA, Ursa Major, Großer Bär). Damit Sie die Linien auch wieder entfernen können, werden sie durch erneute Eingabe des Sternbildes wieder gelöscht (die von ihnen verdeckten Sterne werden wieder sichtbar). Beim Zeichnen der Linien über die Sterne hinweg können im Modus 0 blinkende Farben auftreten, die aber beim Löschen wieder verschwinden.

Geben Sie auf die Frage nach dem zu zeichnenden Sternbild „Alle“ ein, werden alle sichtbaren Sternbilder gezeichnet; durch „Hardcopy“ wird eine solche erzeugt, und durch „Ende“ kommen Sie zurück ins Hauptmenue.

Außerdem können alle Objekte aus der Messier-Liste geplottet werden (für die Messier-Objekte siehe weiter unten). Dazu muß nur die Nummer des Nebels oder Sternhaufens (von 1 bis 109) eingegeben werden. Wenn das Objekt auf dem Bildschirmausschnitt darstellbar ist, wird seine Position durch ein Kreuz markiert. Außerdem wird die Art des Objektes, seine Helligkeit in Größenklassen und gegebenenfalls der Eigennamen angezeigt. Durch erneute Eingabe der Messier-Nummer wird das Kreuz wieder entfernt.

Die Helligkeiten von Sternen und anderen Objekten werden in der Astronomie in Größenklassen



angegeben. Ein kleinerer Wert entspricht dabei einem helleren Objekt. Die schwächsten Sterne, die Sie ohne optische Hilfsmittel sehen können, haben etwa 5 bis 6 Größenklassen. Hellere Sterne haben entsprechend weniger, der Polarstern hat etwa den Wert 2.1. Sehr helle Sterne müssen mit negativen Zahlen belegt werden; der hellste Stern des Himmels, Sirius im Großen Hund, hat -1.4 Größenklassen. Wenn Sie also die Helligkeit eines Nebels oder Sternhaufens wissen, können Sie davon ausgehen, daß Sie sie je nach Sichtverhältnissen dann sehen können, wenn sie heller als 5-6 Größenklassen sind. Hier noch eine kleine Tabelle der Planetenhelligkeiten, die aufgrund ihres teilweise sehr stark wechselnden Abstandes von der Erde kräftig schwanken:

**Merkur:** +3 bis -1.5 Größenklassen (schwer zu beobachten, da nur kurz vor Sonnenaufgang oder nach -untergang zu sehen)

**Venus:** -3.9 bis -4.7 (sehr hell, Morgen- oder Abendstern)

**Mars:** +1.8 bis -2.9 (im Juli 1986 sehr hell: -2.6 Größenklassen)

**Jupiter:** -1.7 bis -2.9

**Saturn:** +1.3 bis -0.5

**Uranus:** um 5.5 (ohne Feldstecher kaum zu sehen)

**Neptun:** um 7.9

**Pluto:** um 13.7 (also nur mit recht großen Teleskopen zu sehen)

## Die Farben der Sterne

Normalerweise werden die Sterne gemäß ihrer Helligkeit geplottet: Im Modus 1 als verschiedene Symbole und im Modus 0 als unterschiedliche Blaustufen (deshalb blau, weil davon die meisten Helligkeitsstufen auf dem Schneider vorliegen). Wenn Sie das Licht ausschalten (Computerfans sollen ja ohnehin meist nachts arbeiten, wie ein Gerücht wissen will) und im Mode 0 bei einem Farbmonitor die Helligkeit so weit herabdrehen, daß Sie die schwächsten Sterne gerade noch erkennen, haben Sie ein sehr natürliches Abbild des Himmels.

Sie können aber auch die Farben der Sterne darstellen; der Fachmann spricht hier von Spektralklassen. Die Sterne sind nicht

```

usschnitt aus Aequatorialkarte (A) ";a$:ka$=
UPPER$(a$)
2020 IF ka$="S" THEN RETURN
2030 IF ka$<>"P" THEN 2100
2035 INPUT "Nord- oder Suedpol als Kartenmitte (N
/S) ";a$:IF UPPER$(a$)="S" THEN mitte=-90 EL
SE mitte =90
2040 INPUT "Deklination des Kartenrandes ";rand:I
F ABS(rand)>90 THEN 2040
2050 RETURN
2100 IF ka$="Q" THEN l=24:r=0:u=-90:o=90:dr=r-l:d
d=o-u:rm=l+dr/2:dm=u+dd/2:RETURN
2105 IF ka$<>"A" THEN 2010
2110 INPUT "Grenzen des Kartenausschnittes: links
, rechts, unten, oben (hh.mm bzw. gg.mm) ";l
$,r$,u$,o$
2120 x$=l$:GOSUB 330:l=x:x$=r$:GOSUB 330:r=x:x$=o
$:GOSUB 330:o=x:x$=u$:GOSUB 330:u=x
2125 dr=r-l:dd=o-u:rm=l+dr/2:dm=u+dd/2
2130 IF r>l THEN h=l:l=r:r=h
2140 IF u>o THEN h=u:u=o:o=h
2150 v=ABS(dr*15/dd/1.66666667):IF ABS(v-1)>0.001
THEN IF v<1 THEN r=rm-dd*0.8333333333/15:l=r
m+dd*0.8333333333/15:GOTO 2125 ELSE o=dm+dr*1
5/3.333333333:u=dm-dr*15/3.333333333:GOTO 2125
2160 dr=r-l:dd=o-u:rm=l+dr/2:dm=u+dd/2
2900 RETURN
3000 REM Eingabe von Datum und Uhrzeit
3010 INPUT "Datum (Tag, Monat, Jahr) ";day,month,
year:IF year<1583 OR year>2499 THEN 3010
3020 INPUT "Uhrzeit (mittlere Ortszeit, Format hh
.mmss) ";x$:zeit=x$:GOSUB 330:zeit=x
3100 x=0.002737897:y=6.67011611:o=month+1:j=year:
IF o<4 THEN o=o+12:j=j-1
3110 nt=INT(30.6*o)+INT(365.25*j)+day
3120 IF nt<694098 THEN nt=nt+1
3130 IF nt<657574 THEN nt=nt+1
3140 IF nt<621050 THEN nt=nt+1
3150 IF nt<584526 THEN nt=nt+1
3160 IF nt>=767148 THEN nt=nt-1
3170 IF nt>=803672 THEN nt=nt-1
3180 IF nt>=840196 THEN nt=nt-1
3200 n=nt-722893:z=n*24*x+(x+1)*zeit+y:z=z/24:sz=
(z-INT(z))*24
3300 nt=nt+zeit/24:jd=nt+1720981.46:dnt=nt
3900 RETURN
4000 REM Eingabe des Ortes
4010 INPUT "Geographische Breite in Grad, noerdli
ch: positiv (Format gg.mmss) ";x$:phi=x$:GO
SUB 330:phi=x
4100 cp=COS(phi):sp=SIN(phi)
4900 RETURN
5000 REM Menue
5010 CLS
5011 PRINT "SKYPLOT"
5012 PRINT "FTCP 1980 / 1985"
5013 PRINT
5020 PRINT "H: Himmeldarstellung (Kartenart)"
5030 PRINT "S: Stern- und Objektfarben"
5040 PRINT "D: Datum und Uhrzeit"
5050 PRINT "O: Ort"
5055 PRINT "I: Status-Information"
5060 PRINT "K: Karte darstellen"
5062 PRINT "E: Bewegung eines Planeten"
5065 PRINT "B: Bildschirmmodus aendern"

```

nur unterschiedlich hell, sondern sie haben auch recht unterschiedliche Oberflächentemperaturen, wodurch ihre Farbe bestimmt wird. Ähnlich wie ein glühendes Stück Metall, das bei steigenden Temperaturen von Rot über Orange und Gelb eine weißglühende Farbe annimmt, verhalten sich auch die Sterne. Die kühlestern haben eine rote Farbe, bei zunehmenden Temperaturen werden sie orange, gelb, gelbweiß und schließlich blauweiß und bläulich. Die Sonne ist an der Oberfläche ca. 5500 Grad Celsius warm und hat eine gelbe Farbe (Spektraltyp G); rote Sterne des Typs M haben etwa 2500-4000 Grad und blaue Sterne mit dem Spektraltyp O oder W 50000 Grad oder mehr. Diese Farben können bei den hellen Sternen auch schon mit bloßem Auge wahrgenommen werden, bei den schwächeren ist ein Feldstecher oder ein Teleskop nötig (dies liegt daran, daß das menschliche Auge bei niedrigen Lichtstärken Farben nicht mehr unterscheiden kann: Nachts sind alle Sterne grau...).

Wenn Sie also statt der Helligkeiten der Sterne die Farben bzw. die Spektralklassen sehen wollen, können Sie dies veranlassen (Menuepunkt „S“). Der Verlauf der Milchstraße kann so leichter erkannt werden, da sich hier in der galaktischen Ebene die jungen und heißen Sterne der Spektralklassen O und B häufen. Die Identifizierung der Sternbilder ist dann allerdings erheblich schwieriger, da so alle Sterne gleich hell erscheinen. Eine Hardcopy sollten Sie übrigens nur im Modus 1 erzeugen, da der Drucker keine Farben darstellt und die Sternsymbole dann alle gleich erscheinen.

## Der Weltraum — unendliche Weiten...

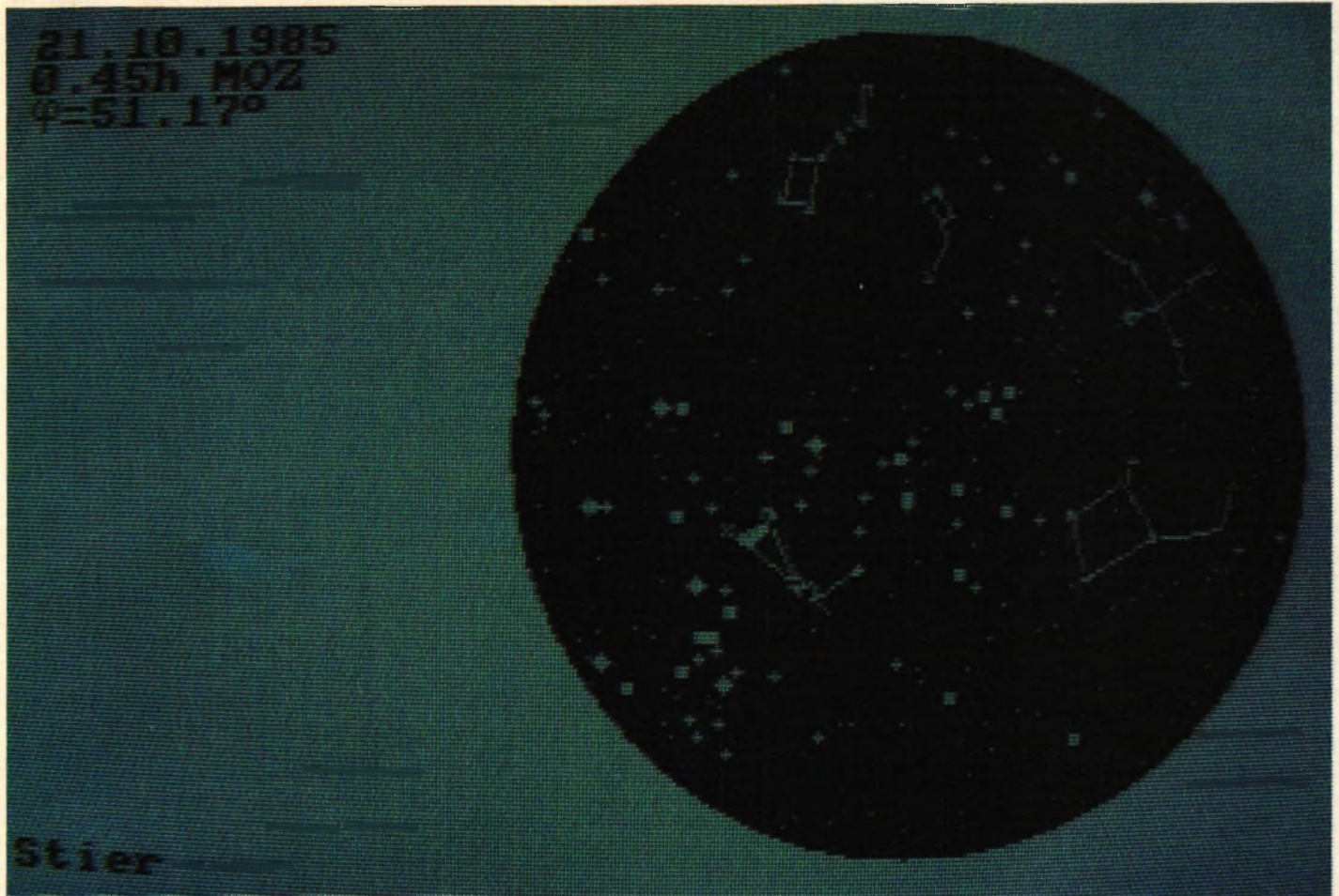
Nein, wir schreiben nicht das Jahr 2100, obwohl Sie den Sternenhimmel dieses Jahres sehr wohl berechnen können. Es soll vielmehr ein kleiner Exkurs in die Tiefen des Universums folgen:

Die Erde befindet sich mit dem Sonnensystem in einer Galaxis, von denen es im Universum ca. 100 Milliarden gibt. Diese Galaxis, genannt Milchstraße, enthält etwa

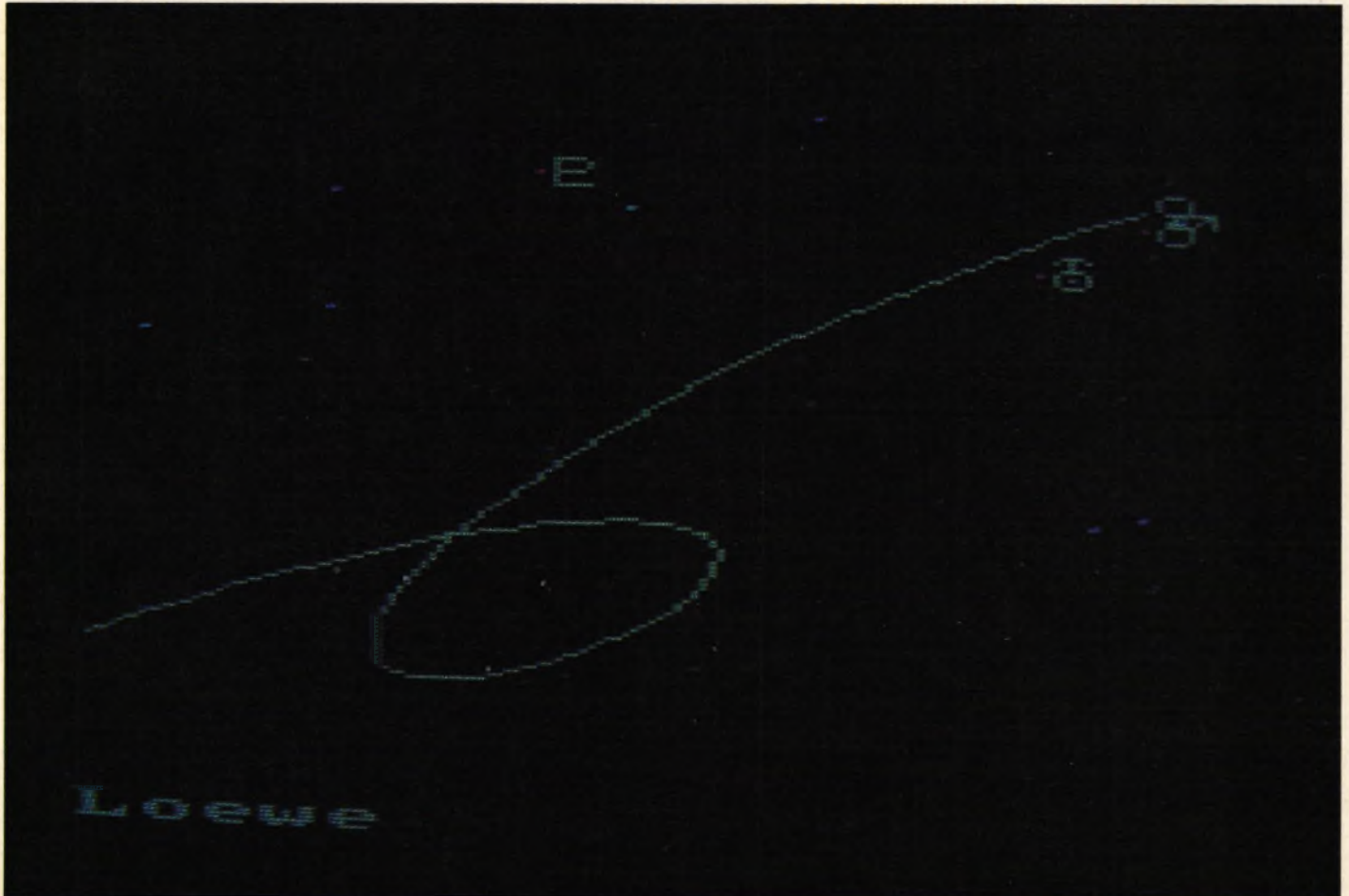
```

5070 PRINT "Z: Zeichenerklärung"
5072 PRINT "P: Programmende"
5080 PRINT
5090 INPUT a$:a$=UPPER$(a$)
5092 nt=dnt
5095 IF a$="I" THEN GOSUB 8000 ELSE IF a$="B" THEN
  N bm=1-bm ELSE IF a$="E" THEN GOSUB 9000 ELSE
  E IF a$="Z" THEN GOSUB 12000
5100 IF a$="P" THEN END ELSE IF a$="S" THEN GOSUB
  1000 ELSE IF a$="H" THEN GOSUB 2000 ELSE IF
  a$="D" THEN GOSUB 3000 ELSE IF a$="O" THEN
  GOSUB 4000 ELSE IF a$="K" THEN hc=0:GOSUB 60
  00
5110 GOTO 5000
6000 MODE bm:INK 0,0:IF bm=0 THEN INK 1,23:INK 2,
  11:INK 3,2:INK 4,1:INK 5,6:INK 6,24:INK 7,26
  :INK 8,25:INK 9,15:INK 10,4:INK 11,9:INK 12,
  13:INK 13,7:INK 14,2,6 ELSE INK 1,13:INK 2,4
  :INK 3,9
6001 IF year=0 THEN RETURN
6002 WINDOW #0,1,80,1,24:WINDOW #1,1,80,25,25:PRI
  NTCHR$(23);CHR$(0);
6005 IF bm=0 THEN hf=12 ELSE hf=1
6010 IF ka$="Q" OR ka$="A" THEN BORDER 0:ORIGIN 3
  20,208:PEN hf:PEN #1,hf:PAPER 0:PAPER #1,0:CLS
  ELSE BORDER 13:ORIGIN 444,208:PEN 0:PEN #
  1,0:PAPER hf:PAPER #1,hf:CLS:FOR i=0 TO 192
  STEP 2:x=SQR(36864-i*i):MOVE -x,-i:DRAW x,-i
  ,0:MOVE -x,i:DRAW x,i,0:NEXT
6020 i=1:CLS #1
6025 IF bm=1 THEN sh=-1
6030 IF mz THEN 6200
6100 RESTORE 60000
6110 READ ra$,de$,h$,typ$
6115 IF typ$="STERNENDE" THEN 6300
6120 IF sh THEN c=INT(VAL(h$)+0.5):IF c<1 THEN c=
  1 ELSE IF c>4 THEN c=4
6130 IF NOT sh THEN IF typ$="O" OR typ$="X" OR ty
  p$="W" THEN c=3 ELSE IF typ$="B" THEN c=2 EL
  SE IF typ$="A" THEN c=7 ELSE IF typ$="F" THE
  N c=8 ELSE IF typ$="G" THEN c=6 ELSE IF typ$
  ="K" THEN c=9 ELSE c=5
6140 x$=ra$:GOSUB 330:ra=x
6150 x$=de$:GOSUB 330:de=x
6155 g=c
6160 GOSUB 7000
6165 x%(i)=x:y%(i)=y:i=i+1:PRINT #1,612-i
6170 GOTO 6110
6200 RESTORE 50000
6210 READ nr,ra$,de$,h$,typ$
6215 IF typ$="ENDE" THEN 6100
6220 IF typ$="K" OR typ$="G" THEN c=11 ELSE c=10
6230 x$=ra$:GOSUB 330:ra=x
6240 x$=de$:GOSUB 330:de=x
6245 g=4
6250 GOSUB 7000
6260 GOTO 6210
6300 l1=1:IF NOT pz THEN 6880
6310 FOR p1=0 TO 10:IF p1=3 THEN 6400
6315 GOSUB 62000:l1=11
6320 g=4:c=13:GOSUB 7000:IF x=-1000 THEN 6400
6325 IF bm=1 THEN PLOT x+2,y+2:PLOT x+2,y-2:PLO
  T x-2,y+2:PLOT x-2,y-2
6330 PLOT -1000,0,hf:PRINT CHR$(23);CHR$(3);:TA

```



Unsere Bilder zeigen den Schneidermonitor mit den Sternzeichen ▲ oder den Planetenbahnen ▼



200 Milliarden Sterne. Es gibt wesentlich kleinere, aber auch erheblich größere Galaxien als unsere Milchstraße im Kosmos. Die Sonne ist darin ein recht durchschnittlicher Stern, die auf ein Leben von bisher etwa 5 Milliarden Jahren herabblicken kann und auch noch etwa so viel vor sich hat. Besagte Milchstraße hat man sich nun als eine flache Scheibe (ähnlich einer ziemlich platten fliegenden Untertasse) mit einer zentralen Verdickung vorzustellen, in der sich die Sonne ziemlich weit am Rand befindet (die Milchstraße hat etwa einen Durchmesser von 100000 Lichtjahren, und die Sonne ist ca. 30000 Lichtjahre vom Zentrum entfernt). Diese flache Scheibe kann als leuchtendes, nebliges Band besonders am Sommerhimmel beobachtet werden. Im Sommer blickt man etwa in Richtung Zentrum, wo dieses Band besonders hell ist.

Nun sind in unserer Milchstraße nicht nur viele Sterne (von denen der nächste im Sternbild Zentaur etwa 4.3 Lichtjahre entfernt ist), sie enthält auch viele für den Himmelsbeobachter äußerst interessante Objekte: (offene) Sternhaufen, Anhäufungen von Sternen mit meist fehlender Struktur; und besonders Anhäufungen von Gas oder Staub zwischen den Sternen, sogenannte Nebel. Diese Nebel werden von Sternen beleuchtet und sind im Feldstecher oder Teleskop als nebel- oder wolkenähnliche Gebilde (daher ihr Name) lohnende Beobachtungsobjekte. Sie können auch Überreste explodierter Sterne sein, sie werden dann als planetarische Nebel bezeichnet.

Diese Objekte haben die Gemeinsamkeit, daß sie abgesehen von ganz wenigen Ausnahmen ungefähr in der Ebene der Milchstraße stehen. Deshalb kann der Verlauf der Milchstraße auch an ihnen verfolgt werden. Dagegen gibt es noch andere Objekte, die sogenannten Kugelsternhaufen, die sich nicht direkt in der Ebene unserer Galaxis befinden. Sie bilden vielmehr eine lockere Ansammlung um die Milchstraße, das sogenannte Halo. Zwar befinden sich einige von ihnen in Richtung Milchstraßenzentrum, das sich im Sternbild Schütze befindet, doch sind die meisten von ihnen irgendwo am Himmel verteilt. Wie der Name schon sagt, handelt es sich um kugelförmige, sehr symmetrische Ansammlungen von

```

      G:MOVE x+8,y+8:PRINT CHR$(240+p1);:TAGOFF:
      PRINT CHR$(23);CHR$(0);
6400 NEXT p1
6800 IF ka$<>"S" THEN 6880
6810 LOCATE 1,1
6830 PRINT USING "##.";day,month;:PRINT USING "##
      ##";year
6840 PRINT LEFT$(zeit$,5);"h MOZ"
6860 PRINT CHR$(252);"=";LEFT$(phi$,6);CHR$(255)
6880 CLS #1
6900 IF beweg THEN RETURN
6905 IF hc THEN GOSUB 40000
6990 GOSUB 10000:MODE 2:BORDER 13:INK 1,0:INK 0,1
      3:PEN 1:PAPER 0:RETURN
7000 REM Rektaszension und Deklination in Bildsch
      irmkordinaten wandeln und Punkt setzen
7005 x=-1000
7010 IF ka$="Q" OR ka$="A" THEN x=(ra-rm)/dr*638:
      y=(de-dm)/dd*380:GOTO 7900
7020 IF ka$="S" THEN 7100
7030 IF mitte=90 AND de<rand OR mitte=-90 AND de>
      rand THEN RETURN
7040 r=ABS((mitte-de)/(mitte-rand))*190:w=-ra*15*
      SGN(mitte):x=COS(w)*r:y=SIN(w)*r
7050 GOTO 7900
7100 IF de<phi-90 OR de>90+phi THEN RETURN
7105 st=(sz-ra)/24:st=360*(st-INT(st)):sd=SIN(de)
      :cd=COS(de):cs=COS(st):ho=FN as(sd*sp+cd*cs*
      cp):IF ho<0 THEN RETURN
7110 az=FN ac(-(sd*cp-cd*cs*sp)/COS(ho)):IF st>18
      0 THEN az=360-az
7120 r=190*(90-ho)/90:w=az-90:x=COS(w)*r:y=SIN(w)
      *r
7900 IF bm=1 THEN IF c=11 THEN c=3 ELSE IF c=10 T
      HEN c=2 ELSE c=1
7905 x=INT(x+0.5):y=INT(y+0.5):IF ABS(x)>320 OR A
      BS(y)>192 THEN x=-1000:y=0:RETURN ELSE y=y+y
      MOD 2:IF c=13 AND TEST(x,y)<>0 THEN IF bm=0
      THEN PLOT x,y,14:RETURN ELSE RETURN
7908 PLOT x,y,c:IF g=4 OR bm=0 THEN RETURN
7910 PLOT x+2,y:PLOT x-2,y:PLOT x,y+2:PLOT x,y-2:
      IF g=3 THEN RETURN
7920 PLOT x+2,y+2:PLOT x-2,y+2:PLOT x+2,y-2:PLOT
      x-2,y-2:IF g=2 THEN RETURN
7930 PLOT x+4,y:PLOT x-4,y:PLOT x,y+4:PLOT x,y-4
7990 RETURN
8000 REM Anzeige Status-Information
8010 PRINT"Geographische Breite: ";phi$;CHR$(255)
8020 PRINT"Ortszeit: ";zeit$;"h"
8030 PRINT"Tag: ";day
8040 PRINT"Monat: ";month
8050 PRINT"Jahr: ";year
8052 PRINT"Sternzeit: ";FN aus$(sz);"h"
8054 PRINT"Julianisches Datum: ";jd
8055 PRINT"Bildschirmmodus: ";bm
8060 IF mz THEN PRINT"Mit"; ELSE PRINT"Ohne";
8070 PRINT" Messier-Objekte";
8075 IF mz THEN PRINT"n" ELSE PRINT
8080 PRINT"Sterne nach ";
8090 IF sh THEN PRINT"Helligkeiten" ELSE PRINT"Sp
      ektralklassen"
8100 IF pz THEN PRINT"Mit"; ELSE PRINT"Ohne";
8110 PRINT" Planeten"
8120 PRINT"Kartenart: ";

```

Sternen, die auch im Feldstecher schon als neblige Fleckchen gesehen werden können.

Schließlich stellt das Programm noch Galaxien dar, die ebenfalls mit kleinen Instrumenten beobachtet werden können. Eine dieser Galaxien, der Große Andromedanebel (im Sternbild Andromeda), kann bei guten Sichtverhältnissen schon mit bloßen Augen gesehen werden, mit dem Feldstecher ist er sehr auffällig als längliches Wölkchen sichtbar. Er ist ca. 2.3 Millionen Lichtjahre entfernt und stellt somit das entfernteste Objekt dar, das der Mensch ohne Hilfsmittel sehen kann. Da das Licht entsprechend der Entfernung 2.3 Millionen Jahre für seine Reise von dieser Galaxis benötigt, sehen wir bei seinem Anblick tief in die Vergangenheit. Alle anderen auf den Ausschnitten dargestellten Galaxien sind noch weiter entfernt, teilweise bis zu 50 Millionen Lichtjahren (ein Lichtjahr sind ca. 9.46 Billionen km).

Da die Erdachse nicht senkrecht zur Milchstraßenebene steht, zeigt das Band der Milchstraße auf der Übersichtskarte des Himmels (Äquatorialkarte) eine etwas seltsame Form: Sie beginnt oben links etwa in der Ecke, läuft dann schräg nach unten rechts, erreicht den tiefsten Punkt noch vor der Mitte der Karte, schwingt sich dann wieder aufwärts und erreicht etwa die rechte obere Ecke. Sie sollten einmal auf der Gesamtkarte folgende Sternbilder darstellen lassen, durch die sich die Milchstraße zieht:

Kepheus, Schwan, Adler, Schütze, Skorpion, Zentaur, Kreuz des Südens, Schiffssegel, Schiffsheck, Großer Hund, Einhorn, Fuhrmann, Perseus und Kassiopeia.

Die Sternhaufen, Nebel und Galaxien stammen aus der sogenannten Messier-Liste, einem Katalog von Objekten, die der französische Astronom gleichen Namens im 18. Jhd. zusammenstellte. Da Messier nur von der Nordhalbkugel aus beobachtete, sind ab einer bestimmten Grenze am Südhimmel keine Messier-Objekte mehr verzeichnet; jedoch gibt es dort noch etliche interessante Beobachtungsobjekte. Die Koordinaten und Daten der Sterne stammen aus dem „Handbuch der Sternbilder“ von H. Vehrenberg und D. Blank (Treugesell-Verlag), einem für die praktische Beobachtung sehr empfehlenswerten Werk.

```

8130 IF ka$="S" THEN PRINT"sichtbarer Himmel" ELS
E IF ka$="P" THEN PRINT"Polarkarte" ELSE IF
ka$="Q" THEN PRINT"Aequatorialkarte" ELSE PR
INT"Ausschnitt aus Aequatorialkarte"
8140 IF ka$<>"P" THEN 8200
8150 PRINT"Ausschnitt vom ";
8160 IF mitte=90 THEN PRINT"Nord"; ELSE PRINT"Sue
d";
8170 PRINT"pol bis ";FN aus$(rand);CHR$(255);" De
kination"
8180 GOTO 10900
8200 IF ka$<>"A" THEN 10900
8210 PRINT"Rektaszension von ";FN aus$(1);"h bis
";FN aus$(r);"h,":PRINT "Deklination von ";F
N aus$(u);CHR$(255);" bis ";FN aus$(o);CHR$(
255)
8220 GOTO 10900
9000 REM Bewegung eines Planeten
9005 IF ka$="S" THEN PRINT "Nicht bei Kartenart:
Sichtbarer Himmel":GOTO 10900
9010 PRINT:INPUT "Welches Objekt (0: Sonne, 1: Me
rkur, 2: Venus, 4: Mars, 5: Jupiter, 6: Satu
rn, 7: Uranus, 8: Neptun, 9: Pluto, 10: Kom
et Halley) ";p1
9020 p1=INT(p1):IF p1>10 OR p1<0 THEN 9010
9030 IF p1=3 THEN PRINT:PRINT "Nicht die Erde - d
a stehen Sie doch drauf!":GOTO 9010
9100 PRINT:INPUT "Zeit zwischen zwei Berechnungen
in Tagen ";dt
9110 PRINT:INPUT "Anzahl der Berechnungen ";ab
9115 beweg=-1:l1=1:planet=p1
9120 GOSUB 6000
9125 beweg=0:p1=planet
9130 GOSUB 62000
9135 PRINT #1,a$;:IF p1=0 THEN PRINT #1,"n";
9137 PRINT #1,"bahn";
9140 l=11:c=12:g=4:GOSUB 7000
9150 f1=(x<>-1000):xa=x:ya=y
9200 FOR be=2 TO ab
9210 nt=nt+dt
9220 GOSUB 62000
9225 l=11:c=12:g=4:GOSUB 7000
9230 IF f1 AND x<>-1000 AND ABS(x-xa)<300 THEN
MOVE xa,ya:DRAW x,y
9240 LOCATE #1,12,1:PRINT #1,ab-be
9300 f1=(x<>-1000):xa=x:ya=y
9310 NEXT be
9320 GOTO 6880
10000 REM Sternbildhilfslinien
10005 WINDOW SWAP 0,1:PRINT CHR$(23);CHR$(1);
10010 INPUT"Sternbild:",a$:IF a$="" THEN 10010
10020 a$=UPPER$(a$):IF a$="ENDE" THEN WINDOW SWAP
1,0:RETURN
10022 IF a$="HARDCOPY" THEN CLS:GOSUB 40000:GOTO 1
0010
10024 IF VAL(a$)>0 THEN GOSUB 11000:GOSUB 10900:GO
TO 10010
10030 RESTORE 30000:IF a$="ALLE" THEN READ b$,b$,b
$:abk$="AND":GOTO 10105
10040 READ b$:IF b$="ENDEBILDER" THEN PRINT"Sternb
ild unbekannt":GOSUB 10900:GOTO 10010
10050 IF LEFT$(b$,1)="*" THEN abk$=RIGHT$(b$,3):RE
AD latein$,deutsch$
10060 IF a$<>abk$ AND a$<>UPPER$(latein$) AND a$<>
UPPER$(deutsch$) THEN 10040

```

## Die Bedienung

Wichtig für den Gebrauch ist die Kenntnis der astronomischen Koordinaten: Ein Punkt auf der Erdoberfläche wird durch seine geographische Länge und Breite (die auch eingegeben werden muß) bestimmt, am Himmel geschieht dies ganz ähnlich. Die eine Richtung parallel zum Himmelsäquator (der Projektion des Erdäquators auf den Himmel) wird durch die Rektaszension bestimmt (entsprechend der Länge), rechtwinklig dazu wird die Deklination gemessen (entsprechend der Breite). Die Rektaszension wird in Zeitmaß, also von 0 Uhr bis 24 Uhr (entsprechend 0 bis 360 Grad), gemessen, die Deklination genau wie die geographische Breite von -90 Grad (Südpol) über 0 Grad (Äquator) bis zum Himmelsnordpol bei 90 Grad. Die Rektaszension läuft dabei auf der Gesamt- und den Ausschnittkarten von rechts nach links, die Deklination von unten nach oben. Wichtig ist, daß Sie bei einem Ausschnitt die linke Grenze (die erste Zahl) immer größer als die rechte eingeben, z. B. als Eingabe 12,8,0,36. Dann wird der Sternhimmel mit der Rektaszension von 12 bis 8 Uhr und der Deklination von 0 bis 36 Grad gezeichnet.

Bei dieser Art der Darstellung des Himmels treten an den Polen große Verzerrungen auf; deshalb werden auch polnahe Sternbilder wie der Kleine Bär (bzw. Wagen) mit dem Polarstern nicht gezeichnet; sie wären kaum zu erkennen. Darum gibt es für die Umgebung der beiden Pole eine besondere Darstellung, ähnlich Karten von den Polen der Erde. Dabei bildet die Karte einen kreisförmigen Ausschnitt mit dem Pol in der Mitte; die Deklination läuft radial vom Pol weg und die Rektaszension wird auf einem Kreis um den Pol gemessen. Die Nullmarke der Rektaszension ist dabei rechts; beim Nordpol läuft die Rektaszension im Uhrzeigersinn um den Pol, während sie beim Südpol entgegen dem Uhrzeigersinn läuft. Der Rand der Karte wird durch eine bestimmte Deklination begrenzt, die Sie eingeben können. Wenn Sie die Umgebung des Nordpols darstellen, sollten Sie als Rand keine südlichen (d.h. negativen) Deklinationen wählen, die Verzerrungen werden dann zu groß (entsprechend für den Südpol). Ein

```

10100 IF ka$="Q" OR ka$="A" THEN IF abk$="UMI" OR
abk$="PEG" OR abk$="PSC" OR abk$="SCL" OR ab
k$="TUC" OR abk$="OCT" THEN 10600
10105 i=1
10107 READ b$:p(i)=VAL(b$):IF x%(ABS(p(i)) MOD 100
0)=-1000 THEN 10600 ELSE IF p(i)>1000 THEN 1
0108 ELSE i=i+1:GOTO 10107
10108 PRINT abk$:i=1
10110 n=p(i):i=i+1:IF n>1000 THEN DRAW x%(n-1000),
y%(n-1000),hf:GOTO 10500
10120 IF n<0 THEN PLOT x%(-n),y%(-n) ELSE DRAW x%(
n),y%(n),hf
10130 GOTO 10110
10500 READ b$:IF VAL(b$)<>0 THEN 10500
10505 IF b$="ENDEBILDER" THEN 10010
10510 IF a$="ALLE" THEN abk$=RIGHT$(b$,3):READ b$,
b$:GOTO 10100 ELSE GOSUB 10900:PRINT latein$
:GOSUB 10900:PRINT deutsch$:GOSUB 10900:GOTO
10010
10600 IF a$="ALLE" THEN 10500 ELSE PRINT"Nicht dar
stellbar":GOSUB 10900:GOTO 10010
10900 IF INKEY$="" THEN 10900 ELSE RETURN
11000 REM Messier-Objekte
11010 n=ABS(INT(VAL(a$))):IF n>109 THEN PRINT"Exis
tiert nicht":RETURN
11020 RESTORE 50000
11030 READ nr,ra$,de$,h$,typ$
11040 IF typ$="ENDE" THEN RETURN
11050 IF n<>nr THEN 11030
11100 x$=ra$:GOSUB 330:ra=x
11110 x$=de$:GOSUB 330:de=x
11120 g=4:c=hf:GOSUB 7000:IF x=-1000 THEN PRINT"Ni
cht sichtbar":GOSUB 10900 ELSE MOVE x,y+8:DR
AW x,y-8,hf:MOVE x-8,y:DRAW x+8,y,hf
11130 PRINT"M";nr:GOSUB 10900:PRINT"Helligkeit ";h
$:GOSUB 10900
11140 IF typ$="0" THEN PRINT"Offener Sternhaufen"
ELSE IF typ$="K" THEN PRINT"Kugelsternhaufen
" ELSE IF typ$="D" THEN PRINT"Diffuser Nebel
" ELSE IF typ$="P" THEN PRINT"Planetarischer
Neb." ELSE PRINT"Galaxie"
11150 GOSUB 10900
11200 IF n=31 THEN PRINT"Andromedanebel" ELSE IF n
=33 THEN PRINT"Triangulumnebel" ELSE IF n=45
THEN PRINT"Plejaden" ELSE IF n=1 THEN PRINT
"Crabnebel" ELSE IF n=42 THEN PRINT"Orionneb
el" ELSE IF n=44 THEN PRINT"Praesepe" ELSE I
F n=97 THEN PRINT"Eulennebel"
11210 IF n=104 THEN PRINT"Sombreronebel" ELSE IF n
=20 THEN PRINT"Trifidnebel" ELSE IF n=8 THEN
PRINT"Lagunennebel" ELSE IF n=17 THEN PRINT
"Omeganebel" ELSE IF n=57 THEN PRINT"Ringneb
el" ELSE IF n=27 THEN PRINT"Hantelnebel"
11500 RETURN
12000 REM Zeichenerklaerung
12010 MODE 1:CLS:PRINT"Planeten:":PRINT:PRINT
12020 PRINTCHR$(240);" Sonne"
12030 PRINT:PRINT CHR$(241);" Merkur"
12040 PRINT:PRINT CHR$(242);" Venus"
12050 PRINT:PRINT CHR$(244);" Mars"
12060 PRINT:PRINT CHR$(245);" Jupiter"
12070 PRINT:PRINT CHR$(246);" Saturn"

```

guter Wert für den Rand ist eine Deklination von 30 bzw. -30 Grad. Der Himmelsnordpol wird ja bekanntlich durch den Polarstern (der Hauptstern im Kleinen Wagen) markiert, auf den die Verlängerung der hinteren Sterne des Wagenkastens des Großen Bären zeigt (natürlich hat ein Bär keinen Wagenkasten, allgemein werden die offiziell als Großer und Kleiner Bär bezeichneten Sternbilder als Wagen bezeichnet). Im Gegensatz zum Nordpol steht am Himmels-südpol kein heller Stern, nur das schwache Sternbild Oktant. Auf der Südhalbkugel der Erde ist aber das Kreuz des Südens zu sehen, dessen längere Achse etwa auf den Pol zeigt.

Die Darstellung des sichtbaren Himmels bildet ebenfalls einen kreisförmigen Ausschnitt ähnlich dem der Polarkarte; dabei ist das Zentrum des Kreises das Zenit, also der Punkt senkrecht über dem Beobachter. Die Begrenzung stellt den Horizont dar, wobei der unterste Punkt des Randes die Südrichtung markiert. Links, wo die Sterne aufgehen, ist Osten, rechts Westen und oben Norden. Wenn Sie eine Hardcopy erstellt haben und die Karte so drehen, daß die Himmelsrichtung, in die Sie blicken, auf der Karte unten ist, entspricht der Anblick des Himmels dem der Karte. Wenn Sie einmal eine drehbare Sternkarte gesehen haben (dabei können Sie auch Beobachtungsdatum und -zeit einstellen) oder eine besitzen, werden Sie feststellen, daß die Karte von SKY-PLOT im Prinzip dasselbe darstellt. Nur haben Sie auf Wunsch die Planeten mit dabei und können den Himmel für jede beliebige geographische Breite darstellen, während die drehbare Sternkarte für einen bestimmten Ort eingestellt ist.

Die Planetensymbole und die Sternfarben, die das Programm verwendet, können mit dem Menüpunkt „Z“ abgerufen werden. Die eingestellten Modi (Art der Kartendarstellung, Ausschnittsgrenzen usw.) können mit „I“ angezeigt werden. Ansonsten ist die Bedienung recht einfach und sollte Ihnen einen ungetrübten Genuß des Sternhimmels ermöglichen.

F. Th.

```

12080 PRINT:PRINT CHR$(247); " Uranus"
12090 PRINT:PRINT CHR$(248); " Neptun"
12100 PRINT:PRINT CHR$(249); " Pluto"
12110 PRINT:PRINT CHR$(250); " Kommet Halley"
12120 GOSUB 10900:CLS
12200 CLS:PRINT"Sternsymbole: ":PRINT:PRINT
12210 PRINT" erste Groesse (hellste Sterne)"
12220 PRINT:PRINT" zweite Groesse"
12230 PRINT:PRINT" dritte Groesse"
12240 PRINT:PRINT" vierte Groesse und schwaecher"
12250 x=8:y=342:g=1:PLOT x,y,1:GOSUB 7910
12260 y=310:g=2:PLOT x,y,1:GOSUB 7910
12270 y=278:g=3:PLOT x,y,1:GOSUB 7910
12280 PLOT x,246
12300 GOSUB 10900
12310 MODE 0:PRINT"Sternfarben: ":PRINT:PRINT"Spekt
ralklassen: "
12320 c=3:GOSUB 13000:PRINT"D bzw. W"
12330 c=2:GOSUB 13000:PRINT"B"
12340 c=7:GOSUB 13000:PRINT"A"
12350 c=8:GOSUB 13000:PRINT"F"
12360 c=6:GOSUB 13000:PRINT"G"
12370 c=9:GOSUB 13000:PRINT"K"
12380 c=5:GOSUB 13000:PRINT"M, R, N, S bzw. C"
12390 PRINT:PRINT:PRINT"Messier-Objekte: ":c=10:GOS
UB 13000:PRINT"Innergalaktische"
12400 c=11:GOSUB 13000:PRINT"Aussergalaktische":PR
INT" Objekte";
12900 GOSUB 10900:MODE 2:RETURN
13000 PRINT:PAPER c:PRINT" ";:PAPER 12:PRINT" ";:R
ETURN
30000 DATA *AND,Andromeda,Andromeda,-1,4,2,3,-2,10
07
30005 DATA *LAC,Lacerta,Eidechse,-11,10,1012
30008 DATA *AQR,Aquarius,Wassermann,-21,19,18,15,1
3,14,17,-15,20,1016
30010 DATA *AQL,Aquila,Adler,-27,28,25,23,24,29,-2
3,26,1031
30020 DATA *SCT,Scutum,Schild,-33,32,1034
30030 DATA *ARI,Aries,Widder,-37,36,35,1039
30040 DATA *TRI,Triangulum,Dreieck,-40,41,42,1040
30050 DATA *AUR,Auriga,Fuhrmann,-43,46,49,380,48,4
4,1043
30060 DATA *BOO,Bootes,Baerenhueter,-58,52,54,53,5
5,56,52,1057
30070 DATA *CRB,Corona Borealis,Noerdliche Krone,-
64,61,60,62,1063
30075 DATA *CAM,Camelopardalis,Giraffe,-68,66,65,6
7,1069
30080 DATA *CNC,Cancer,Krebs,-70,73,72,75,-71,73,1
074
30090 DATA *CVN,Canes Venatici,Jagdhunde,-77,1078
30100 DATA *CMA,Canis Major,Grosser Hund,-85,82,-8
3,82,79,-81,79,1080
30110 DATA *LEP,Lepus,Hase,-95,92,91,90,89,92,94,-
90,93,98,1089
30120 DATA *CMI,Canis Minor,Kleiner Hund,-99,1100
30130 DATA *MON,Monoceros,Einhorn,-107,102,105,103
,104,-105,1106
30140 DATA *CAP,Capricornus,Steinbock,-110,116,117
,113,112,111,115,114,110,109,1108
30150 DATA *CAS,Cassiopeia,Kassiopeia,-120,124,118
,119,126,120,121,1122

```

30160 DATA *CEP,Cepheus,Kepheus,-130,137,132,129,130,131,1137	30640 DATA *CEN,Centaurus,Zentaur,-493,491,497,496,495,494,493,490,489,-491,1492
30170 DATA *CET,Cetus,Walfisch,-139,141,142,149,143,147,140,144,145,143,-140,1146	30650 DATA *CRU,Crux,Kreuz des Suedens,-500,502,-499,1501
30180 DATA *CDM,Coma Berenices,Haar der Berenike,-150,151,1152	30660 DATA *CRA,Corona Australis,Suedliche Krone,-507,505,1506
30190 DATA *CRV,Corvus,Rabe,-157,154,156,155,157,1153	30670 DATA *LUP,Lupus,Wolf,-514,510,512,516,513,508,509,511,1514
30200 DATA *CRT,Crater,Becher,-159,160,161,162,1159	30680 DATA *NOR,Norma,Winkelmass,-520,1517
30210 DATA *SEX,Sextans,Sextant,-164,163,1165	30690 DATA *DOR,Dorado,Schwertfisch,-523,521,522,1524
30220 DATA *CYG,Cygnus,Schwan,-171,170,168,169,174,175,-166,168,172,1167	30700 DATA *MEN,Mensa,Tafelberg,-526,528,529,1527
30230 DATA *DEL,Delphinus,Delphin,-179,178,180,181,179,1182	30710 DATA *RET,Reticulum,Netz,-530,534,533,532,531,1530
30240 DATA *EQU,Equuleus,Fuellen,-184,187,186,188,1184	30720 DATA *FOR,Fornax,Ofen,-537,535,536,1538
30250 DATA *DRA,Draco,Drache,-611,190,191,611,192,193,194,195,196,197,189,1198	30730 DATA *GRU,Grus,Kranich,-545,544,540,543,542,541,-542,1539
30260 DATA *UMI,Ursa Minor,Kleiner Baer,-204,609,201,200,204,203,202,1199	30740 DATA *MIC,Microscopium,Mikroskop,-550,549,1548
30270 DATA *ERI,Eridanus,Fluss Eridanus,-206,215,207,208,209,210,213,211,212,214,1205	30750 DATA *PSA,Piscis Austrinus,-553,554,555,552,556,553,1557
30280 DATA *GEM,Gemini,Zwillinge,-218,221,219,217,216,220,1222	30760 DATA *HYI,Hydrus,Kleine Wasserschl.,-558,559,560,1558
30290 DATA *HER,Hercules,Herkules,-233,235,230,229,226,-235,232,231,227,228,-230,1231	30770 DATA *IND,Indus,Inder,-563,566,565,-566,1564
30300 DATA *HYA,Hydra,Noerdl. Wasserschl.,-245,246,238,237,244,243,236,241,240,239,1241	30780 DATA *OCT,Octans,Oktant,-567,568,569,1567
30310 DATA *LEO,Leo,Loewe,-249,250,248,254,247,253,249,252,1251	30790 DATA *TUC,Tucana,Tukan,-570,572,571,573,1570
30320 DATA *LMI,Leo Minor,Kleiner Loewe,-259,258,1260	30800 DATA *PHE,Phoenix,Phoenix,-575,579,577,576,575,580,1578
30330 DATA *LYN,Lynx,Luchs,-261,265,264,263,1262	30810 DATA *SCL,Sculptor,Bildhauer (atelier),-581,584,583,1582
30340 DATA *LIB,Libra,Waage,-267,266,268,1269	30820 DATA *PIC,Pictor,Maler (staffelei),-585,587,1586
30350 DATA *LYR,Lyra,Leier,-281,276,277,285,278,281,1275	30830 DATA *PUP,Puppis,Schiffsheck,-598,596,591,598,595,589,593,595,-589,592,1594
30360 DATA *OPH,Ophiuchus,Schlangentraeger,-293,292,291,290,296,287,288,293,1294	30840 DATA *VEL,Vela,Schiffsegel,-603,605,601,600,599,602,601,-602,1606
30370 DATA *ORI,Orion,Orion,-301,299,306,303,302,301,300,307,298,1303	31000 DATA "ENDEBILDER"
30380 DATA *PEG,Pegasus,Pegasus,-313,309,1,310,308,309,-308,312,314,1311	40000 MEMORY &9FFF
30390 DATA *PER,Perseus,Perseus,-323,322,321,318,319,328,-318,320,1324	40010 RESTORE 40060
30400 DATA *PSC,Pisces,Fische,-336,335,330,331,336,338,332,333,329,337,1334	40020 FOR i=&A000 TO &A0BF
30410 DATA *SGE,Sagitta,Pfeil,-341,342,339,-342,1340	40030 READ byte:POKE i,byte:s+byte:NEXT
30420 DATA *VUL,Vulpecula,Fuechslein,-344,343,345,1346	40040 CALL &A000:PRINT#8:PRINT#8
30430 DATA *SGR,Sagittarius,Schuetze,-349,348,353,357,356,-357,355,358,-355,351,350,-351,352,1354	40045 IF ka\$="Q" OR ka\$="A" THEN ORIGIN 320,208 ELSE ORIGIN 444,208
30440 DATA *SCD,Scorpius,Skorpion,-359,361,360,359,362,368,363,364,365,366,1367	40050 RETURN
30450 DATA *SER,Serpens,Schlange,-375,374,378,293,290,376,373,369,372,370,1371	40060 DATA &cd,&ba,&bb,&cd,&e7,&bb,&32,&bd
30460 DATA *TAU,Taurus,Stier,-384,379,380,-379,386,387,381,382,383,-385,381,1388	40070 DATA &a0,&cd,&6c,&a0,&21,&8f,&01,&22
30470 DATA *UMA,Ursa Major,Grosser Baer,-393,392,391,390,393,394,395,1396	40080 DATA &be,&a0,&11,&00,&00,&3e,&07,&32
30480 DATA *VIR,Virgo,Jungfrau,-400,407,402,403,405,400,408,409,-404,403,-402,406,1401	40090 DATA &c0,&a0,&cd,&7c,&a0,&0e,&00,&3a
30490 DATA *ANT,Antlia,Luftpumpe,-412,410,1411	40100 DATA &c0,&a0,&47,&e5,&d5,&c5,&cd,&f0
30500 DATA *PYX,Pyxis,Schiffskompass,-415,413,1414	40110 DATA &bb,&c1,&d1,&21,&bd,&a0,&be,&e1
30510 DATA *APS,Apus,Paradiesvogel,-416,418,1417	40120 DATA &37,&20,&01,&a7,&cb,&11,&2b,&2b
30520 DATA *CIR,Circinus,Zirkel,-422,421,1423	40130 DATA &10,&e9,&cd,&af,&a0,&79,&cd,&a6
30530 DATA *TRA,Triangulum Australe,Suedliches Dreieck,-424,425,426,1424	40140 DATA &a0,&13,&e5,&21,&7f,&02,&37,&ed
30540 DATA *ARA,Ara,Altar,-435,428,429,430,431,-429,433,1434	40150 DATA &52,&e1,&38,&05,&2a,&be,&a0,&18
30550 DATA *PAV,Pavo,Pfau,-436,437,440,441,442,443,439,437,1438	40160 DATA &cc,&23,&7c,&b5,&c8,&2b,&11,&00
30560 DATA *TEL,Telescopium,Fernrohr,-446,445,1447	40170 DATA &00,&22,&be,&a0,&3e,&07,&bd,&20
30570 DATA *CAE,Caelum,Grabstichel,-448,1450	40180 DATA &b9,&7c,&b4,&20,&b5,&3e,&04,&32
30580 DATA *COL,Columba,Taube,-452,451,455,452,453,1454	40190 DATA &c0,&a0,&18,&ae,&3e,&1b,&cd,&a6
30590 DATA *HOR,Horologium,Pendeluhr,-459,1457	40200 DATA &a0,&3e,&41,&cd,&a6,&a0,&3e,&07
30600 DATA *CAR,Carina,Schiffskiell,-464,465,461,467,469,464,468,462,466,1460	40210 DATA &cd,&a6,&a0,&c9,&e5,&3e,&42,&cd
30610 DATA *CHA,Chamaeleon,Chamaeleon,-471,473,472,474,475,1471	40220 DATA &1e,&bb,&e1,&28,&02,&e1,&cd,&3e
30620 DATA *MUS,Musca,Fliege,-476,477,479,478,476,480,1481	40230 DATA &0d,&cd,&a6,&a0,&3e,&0a,&cd,&a6
30630 DATA *VOL,Volans,Fliegender Fisch,-487,488,485,484,486,487,1482	40240 DATA &a0,&3e,&1b,&cd,&a6,&a0,&3e,&4c
	40250 DATA &cd,&a6,&a0,&3e,&7f,&cd,&a6,&a0
	40260 DATA &3e,&02,&cd,&a6,&a0,&c9,&cd,&2e
	40270 DATA &bd,&38,&fb,&cd,&2b,&bd,&c9,&3a
	40280 DATA &c0,&a0,&fe,&07,&c8,&af,&cb,&11
	40290 DATA &cb,&11,&cb,&11,&c9,&00,&00,&00
	50000 REM
	50010 DATA 31,0.4,41,4.8,6,32,0.4,40.35,8.7,6,33,1.31,30.24,6.7,6,74,1.34,15.32,10.2,6,76,1.39,51.19,10.8,P,103,1.30,60.26,7.4,0,34,2.39,42.34,5.5,0,77,2.40,-0.13,8.9,6,45,3.44,23.57,1.6,0,1,5.32,21.59,8.4,P,36,5.33,34.07,6.3,0
	50020 DATA 37,5.49,32.33,6.2,0,38,5.25,35.48,7.4,0,42,5.33,-5.25,2.9,0,43,5.33,-5.18,4.6,0,78,5.44,0.02,8,0,79,5.22,-24.34,7.9,K,35,6.06,24.21,5.3,0,41,6.45,-20.41,4.6,0,46,7.40,-14.42,6,0,47,7.52,-15.17,4.5,0,50,7,-8.16,6.3,0
	50030 DATA 93,7.42,-23.45,6,0,44,8.37,20.1,3.7,0,48,8.12,-1.48,5.3,0,67,8.48,12.6,1,0,81,9.52,69.18,7.9,6,82,9.52,69.56,8.8,6,95,10.41,11.58,10.4,6,96,10.44,12.05,9.1,6,105,10.45,12.51,9.2,6,65,11.16,13.22,9.3,6,66,11.18,13.16,8.4,6



50040 DATA 97,11.12,55.18,9.1,P, 108,11.09,55.57,1  
0,6, 109,11.55,53.39,9.9,G, 49,12.27,8.16,8.  
6,6, 58,12.35,12.05,9.2,G, 59,12.4,11.55,9.6  
,6, 60,12.41,11.5,8.9,G, 61,12.19,4.45,10.1,  
6, 64,12.54,21.57,8.8,G, 68,12.37,-26.28,8.2  
,K, 84,12.23,13.1,9.3,G

50050 DATA 85,12.23,18.28,9.3,G, 86,12.24,13.13,9.  
7,6, 87,12.28,12.4,9.2,G, 88,12.3,14.42,10.2  
,6, 89,12.33,12.5,9.5,G, 90,12.34,13.26,10,6  
, 94,12.49,41.24,7.9,G, 98,12.11,15.11,10.7,  
6, 99,12.16,14.42,10.1,G, 100,12.2,16.06,9.5  
,6, 104,12.37,-11.2,8.7,G

50060 DATA 106,12.17,47.35,8.6,G, 3,13.4,28.38,6.4  
,K, 51,13.28,47.27,8.1,G, 53,13.1,18.26,7.6,  
K, 63,13.14,42.18,9.5,G, 83,13.34,-29.37,9.5  
,6, 101,14.01,54.35,9.6,G, 5,15.16,2.16,6.2,  
K, 102,15.05,55.57,10.8,G, 4,16.22,-26.24,6.  
4,K, 10,16.54,-4.02,6.7,K

50070 DATA 12,16.45,-1.52,6.6,K, 13,16.4,36.33,5.7  
,K, 62,16.58,-30.02,6.6,K, 80,16.14,-22.51,7  
,7,K, 107,16.3,-12.57,9.2,K, 6,17.37,-32.11,  
5.3,0, 7,17.51,-34.48,3.2,0, 9,17.16,-18.28,  
7.3,K, 14,17.35,-3.13,7.7,K, 19,17,-26.12,6.  
6,K, 20,17.59,-23.02,7.5,D

50080 DATA 23,17.54,-19.01,6.9,0, 92,17.16,43.11,6  
.2,K, 8,18.01,-24.23,5.9,D, 11,18.48,-6.2,6.  
3,0, 16,18.16,-13.48,6.4,0, 17,18.18,-16.12,  
7.7,D, 18,18.17,-17.09,7.5,0, 21,18.02,-22.3  
,6.5,0, 22,18.33,-23.57,6.9,K, 24,18.16,-18.  
27,4.6,0

50090 DATA 25,18.29,-19.17,6.5,0, 26,18.42,-9.26,9  
.3,0, 28,18.22,-24.54,7.3,K, 54,18.52,-30.32  
,7.1,K, 57,18.52,32.58,8.9,P, 69,18.28,-32.2  
3,8.9,K, 70,18.4,-32.21,9.6,K, 27,19.58,22.3  
5,7.6,P, 55,19.37,-31.04,6.3,K, 56,19.15,30.  
05,8.2,K

50100 DATA 71,19.52,18.39,8.3,K, 29,20.02,38.21,7.  
1,0, 72,20.51,-12.44,9.8,K, 73,20.56,-12.5,7  
.5,0, 75,20.03,-22.04,8,K, 2,21.31,-1.03,6.3  
,K, 15,21.28,11.57,6,K, 30,21.38,-23.25,8.4,  
K, 39,21.3,48.13,5.2,0, 52,23.22,61.19,7.3,0

50110 DATA 0,0,0,0,ENDE

60000 REM

60010 DATA 0.06,28.49,2.15,B, 1.07,35.21,2.37,M, 2  
.01,42.05,2.21,K, 0.37,30.35,3.49,K, 0.45,24  
,4.3,K, 23.35,46.11,4,6, 0.54,38.14,3.94,A,  
1.34,41.09,4.18,G, 1.35,48.23,3.77,K

60030 DATA 22.3,50.02,3.85,A, 22.22,51.59,4.58,K,  
22.14,37.3,4.22,K

60050 DATA 22.03,-0.34,3.19,G, 21.29,-5.48,3.07,G,  
22.19,-1.38,3.97,A, 22.52,-16.05,3.51,A, 20  
.45,-9.41,3.83,A, 22.26,-0.17,3.75,F, 22.33,  
-0.23,4.13,B, 22.14,-8.02,4.32,G, 22.5,-7.51  
,3.84,M, 23.07,-21.27,3.8,K

60070 DATA 19.48,8.44,0.8,A, 19.53,6.17,3.9,G, 19.  
44,10.29,2.8,K, 19.23,3.01,3.44,A, 18.57,15,  
4.21,K, 19.03,13.47,3.02,B, 20.09,-0.58,3.37  
,8, 19.34,-1.24,4.28,B, 19.04,-4.58,3.55,B

60090 DATA 18.33,-8.17,4.06,K, 18.45,-4.48,4.47,G,  
18.26,-14.36,4.73,A

60110 DATA 2.04,23.14,2,K, 1.52,20.34,2.72,A, 1.51  
,19.03,4,A, 3.09,19.32,4.53,K, 2.47,27.03,3.  
68,B

60130 DATA 1.5,29.2,3.58,F, 2.07,34.45,3.08,A, 2.1  
4,33.37,4.07,A

60150 DATA 5.13,45.57,0.09,G, 5.56,44.57,1.93,A, 5  
.55,54.17,3.88,G, 4.58,43.45,3.5,F, 5.03,41.  
1,3.28,B, 5.56,37.13,2.7,A, 4.54,33.05,2.9,K  
, 6.12,29.31,4.45,G, 5.48,39.08,4.18,K

60170 DATA 14.13,19.27,-0.06,K, 15,40.35,3.63,G, 1  
4.3,38.32,3.23,A, 15.14,33.3,3.54,G, 14.43,2  
7.17,2.59,K, 14.39,13.56,3.86,A, 13.52,18.39  
,2.8,G, 14.24,52.05,4.06,F

60190 DATA 15.33,26.53,2.21,A, 15.26,29.17,3.72,A,  
15.41,26.27,3.93,A, 15.56,27.01,4.22,K, 15.  
31,31.32,4.17,B

60210 DATA 4.49,66.16,4.38,0, 4.59,60.22,4.22,G, 3  
.45,71.11,4.67,A, 4.53,53.41,4.44,A, 3.25,59  
.46,4.42,B

60230 DATA 8.56,12.03,4.27,F, 8.14,9.2,3.76,K, 8.4  
,21.39,4.73,A, 8.42,18.2,4.17,K, 8.09,17.48,  
4.71,G, 8.44,28.57,4.09,G

60250 DATA 12.54,38.35,5.39,A, 12.54,38.35,2.78,A,  
12.31,41.38,4.32,G

60270 DATA 6.43,-16.39,-1.43,A, 6.21,-17.56,1.97,B  
, 7.02,-15.33,4.07,B, 7.06,-26.19,1.84,G, 6.  
57,-28.54,1.78,B, 6.18,-30.02,3.1,B, 7.22,-2

9.12,2.43,B, 6.52,-11.58,4.25,K, 6.54,-16.59  
,4.39,B, 6.48,-32.27,3.78,B

60290 DATA 5.31,-17.51,2.69,F, 5.26,-20.48,2.96,G,  
5.42,-22.28,3.8,F, 5.49,-20.53,3.9,G, 5.03,  
-22.26,3.29,K, 5.45,-14.5,3.67,A, 5.54,-14.1  
1,3.77,F, 5.11,-13.4,4,B, 5.17,-13.14,4.29,B  
, 5.11,-16.16,3.3,A

60310 DATA 7.37,5.21,0.37,F, 7.24,8.23,3.09,B, 7.2  
5,9.02,4.6,K

60330 DATA 7.39,-9.26,4.07,K, 6.26,-7,3.93,B, 6.12  
, -6.15,4.09,K, 7.09,-0.25,4.09,A, 6.21,4.37,  
4.33,A, 8.06,-2.5,4.41,G

60350 DATA 20.15,-12.4,4.53,G, 20.15,-12.42,3.77,G  
, 20.18,-14.56,3.25,F, 21.37,-16.53,3.8,F, 2  
1.44,-16.21,2.98,A, 21.24,-22.38,3.86,G, 21.  
03,-17.26,4.19,A, 21.2,-17.03,4.3,K, 20.43,-  
25.27,4.26,F, 20.49,-27.06,4.24,M

60370 DATA 0.38,56.16,2.47,K, 0.07,58.52,2.42,F, 0  
.54,60.27,1.6,B, 1.23,59.59,2.8,A, 1.51,63.2  
5,3.44,B, 0.34,53.37,3.72,B, 0.46,57.33,3.64  
,F, 1.08,54.53,4.52,A, 0.3,62.39,4.24,B, 1.5  
9,72.11,4.06,A, 23.52,57.13,4.1,G

60390 DATA 21.17,62.22,2.6,A, 21.38,70.2,3.32,B, 2  
3.37,77.21,3.42,K, 22.27,58.1,3.9,F, 22.13,5  
6.48,4.23,A, 22.09,57.57,3.62,K, 20.44,61.39  
,3.59,G, 20.29,62.5,4.28,A, 22.48,65.56,3.68  
,K, 21.42,58.33,3.6,M

60410 DATA 3,3.54,2.82,M, 0.41,-18.16,2.24,K, 2.41  
,3.02,3.58,A, 2.37,0.07,4.04,B, 1.49,-10.35,  
3.92,K, 1.06,-10.27,3.6,K, 1.22,-8.26,3.83,K  
, 0.17,-9.06,3.75,K, 1.42,-16.12,3.65,G, 1.5  
8,-21.19,4.18,M, 2.17,-3.12,2,M

60430 DATA 13.08,17.48,4.47,F, 13.1,28.08,4.32,G,  
12.24,28.33,4.56,K

60450 DATA 12.06,-24.27,4.18,F, 12.32,-23.07,2.84,  
6, 12.13,-17.16,2.78,B, 12.27,-16.14,3.1,A,  
12.08,-22.21,3.21,K, 12.3,-15.55,4.42,F

60470 DATA 10.57,-18.02,4.2,K, 11.09,-22.33,4.52,A  
, 11.22,-17.25,4.14,A, 11.17,-14.3,3.82,K

60490 DATA 10.05,-0.08,4.5,A, 10.28,-0.23,4.95,B,  
9.5,-7.52,5.16,A

60510 DATA 20.4,45.06,1.26,A, 19.29,27.51,3.1,B, 2  
0.2,40.06,2.32,F, 19.43,45.2,2.97,A, 20.44,33.  
47,2.64,K, 21.11,30.01,3.4,G, 19.54,34.57,4.  
03,K, 20.12,46.35,3.95,K, 19.28,51.37,3.94,A  
, 19.16,53.17,3.98,G, 19.49,32.47,3.3,S, 20.  
58,47.2,4.49,B

60530 DATA 20.37,15.44,3.86,B, 20.35,14.25,3.72,F,  
20.44,15.57,4.12,F, 20.41,14.54,4.53,A, 20.  
31,11.08,3.98,B, 20.33,14.3,4.69,A

60550 DATA 21.13,5.02,4.14,F, 21.2,6.36,5.14,A, 21  
.08,9.56,4.76,F, 21.12,9.48,4.61,F, 20.57,4.  
06,5.29,F

60570 DATA 14.03,64.37,3.64,A, 17.29,52.2,2.99,G,  
17.55,51.3,2.42,K, 19.13,67.34,3.24,G, 19.48  
,70.08,3.99,G, 17.09,65.47,3.22,B, 16.23,61.  
38,2.89,G, 16.01,58.42,4.11,F, 15.24,59.08,3  
.47,K, 12.31,70.04,3.88,B

60590 DATA 1.49,89.02,1.94,F, 14.51,74.22,2.02,K,  
15.21,72.01,3.14,A, 17.48,86.37,4.44,A, 16.5  
1,82.07,4.4,G, 15.46,77.54,4.34,A

60610 DATA 1.36,-57.29,0.6,B, 5.05,-5.09,2.92,A, 3  
.56,-13.39,3.19,M, 3.41,-9.56,3.72,K, 3.31,-  
9.38,3.81,K, 2.54,-9.06,4.05,K, 2.56,-40.3,3  
.06,A, 2.39,-40.04,4.06,G, 4.16,-33.55,3.59,  
B, 1.54,-51.51,3.73,G, 4.34,-3.27,3.4,B

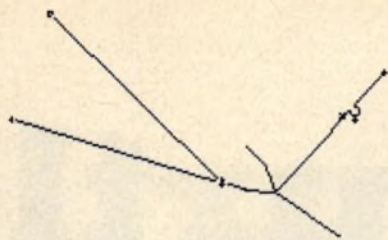
60630 DATA 7.31,32,1.58,A, 7.42,28.09,1.16,K, 6.35  
,16.27,1.93,A, 7.17,22.05,3.5,A, 6.41,25.11,  
3.18,G, 7.01,20.39,3.68,F, 6.12,22.31,3.27,M  
, 6.5,34.01,3.64,A, 7.23,27.54,3.89,G, 7.41,  
24.31,3.68,G

60650 DATA 17.12,14.27,2.9,M, 16.28,21.36,2.81,G,  
16.2,19.16,3.79,A, 17.13,24.54,3.16,A, 16.58  
,31,3.92,A, 16.39,31.42,3,G, 16.41,39.01,3.6  
1,G, 17.55,37.15,3.99,K, 17.45,27.45,3.48,G,  
17.13,36.52,3.36,K

60670 DATA 9.25,-8.26,1.98,K, 11.5,-33.38,4.4,B, 1  
3.16,-22.54,3.33,G, 8.35,5.53,4.18,A, 8.44,6  
.36,3.48,G, 8.53,6.08,3.3,G, 8.41,3.35,4.32,  
B, 10.47,-15.56,3.32,K, 11.31,-31.35,3.72,G,  
14.04,-26.27,3.48,K, 13.27,-23.01,3,M

60690 DATA 10.06,12.13,1.36,B, 11.47,14.51,2.23,A,  
10.17,20.06,2.3,K, 11.12,20.48,2.58,A, 9.43  
,24,3.12,G, 10.14,23.4,3.65,F, 10.05,17,3.58  
,A, 11.12,15.42,3.41,A, 11.21,10.48,4.03,F,  
10.3,9.34,3.85,B, 9.45,11.4,4.4,M

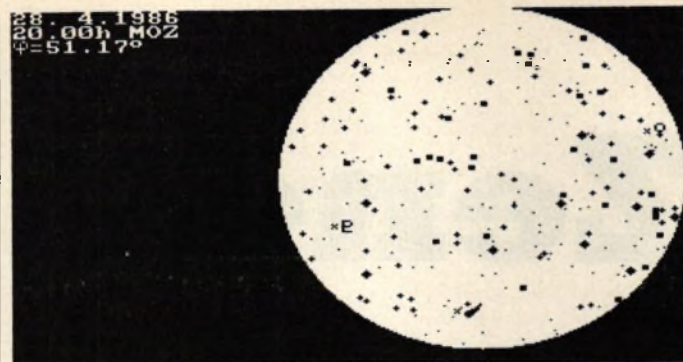
60710 DATA 10.25,36.58,4.41,G, 10.51,34.29,3.92,K, 10.05,35.29,4.47,A	61150 DATA 16.43,-68.56,1.88,K, 15.51,-63.17,3.04, F, 15.42,-68.3,3.06,A, 16.11,-63.34,4.03,G
60730 DATA 9.18,34.36,3.3,M, 7.23,49.19,4.45,A, 8.19,43.21,4.43,K, 8.58,41.59,4.09,F, 9.16,37.01,3.82,B	61170 DATA 17.28,-49.5,2.97,B, 17.21,-55.29,2.8,K, 17.21,-56.2,3.51,B, 17.27,-60.39,3.79,B, 16.56,-53.05,4.15,M, 16.55,-55.55,3.06,K, 16.46,-58.57,3.68,K, 18.03,-50.06,3.9,B
60750 DATA 14.48,-15.47,5.33,F, 14.48,-15.50,2.9,A, 15.14,-9.12,2.74,B, 15.33,-14.37,4.02,G, 14.58,-8.19,4.91,A, 15.51,-16.35,4.34,G, 15.01,-25.05,3.41,M, 15.36,-29.37,3.8,B, 15.34,-27.58,3.78,K	61190 DATA 20.22,-56.54,2.12,B, 20.41,-66.23,3.6,A, 21.22,-65.36,4.3,F, 20.04,-66.19,3.64,G, 19.55,-73.03,4.1,A, 18.37,-71.28,4.1,K, 17.41,-64.42,3.58,K, 18.19,-61.31,4.25,M, 18.52,-67.18,3.92,F
60770 DATA 18.35,38.44,0.04,A, 18.48,33.18,3.34,B, 18.57,32.37,3.3,B, 18.52,36.54,5.51,B, 18.43,39.37,3.83,A, 18.43,39.34,4.5,A, 18.43,37.33,4.06,A, 19.12,39.03,4.46,B, 19.15,38.03,4.46,K, 18.18,36.02,4.34,K, 18.53,36.5,5.98,M, 18.54,43.53,3.9,M	61210 DATA 18.23,-46,3.76,B, 18.08,-45.58,4.6,G, 18.25,-49.06,4.14,K
60800 DATA 17.33,12.36,2.14,A, 17.41,4.35,2.94,K, 17.45,2.43,3.74,A, 16.12,-3.34,3.03,M, 16.16,-4.34,3.34,G, 16.34,-10.28,2.70, 17.08,-15.4,2.63,A, 17.19,-24.57,3.37,B, 16.52,10.15,4.29,B, 16.55,9.27,3.42,K, 16.24,-18.21,4.18,B	61230 DATA 4.39,-41.58,4.52,F, 4.3,-37.14,5.08,F, 5.03,-35.33,4.62,K
60820 DATA 5.53,7.24,0.42,M, 5.12,-8.15,0.15,B, 5.22,6.18,1.64,B, 5.3,-0.2,1.94,0, 5.34,-1.14,1.7,B, 5.38,-1.58,1.78,B, 5.22,-2.26,3.14,B, 5.33,-5.56,2.87,0, 5.45,-9.41,2.2,B, 5.32,9.54,3.49,0	61250 DATA 5.38,-34.06,2.75,B, 5.49,-35.47,3.22,K, 5.56,-35.17,4.36,B, 6.2,-33.25,3.98,G, 5.29,-35.3,3.92,K
60840 DATA 23.02,14.56,2.57,B, 23.01,27.49,2.3,M, 0.11,14.54,2.87,B, 21.42,9.39,2.54,K, 22.39,10.34,3.61,B, 22.41,29.58,3.1,G, 22.08,5.57,3.7,A, 22.05,25.06,3.96,F, 21.42,25.25,4.27,F, 22.48,24.2,3.67,G	61270 DATA 4.12,-42.25,3.83,K, 2.58,-64.16,5.08,A, 4.09,-42.07,4.85,F, 2.36,-52.46,5.26,A
60860 DATA 3.21,49.41,1.8,F, 3.05,40.46,2.13,B, 3.01,53.19,3.08,F, 3.39,47.38,3.1,B, 3.55,39.52,2.96,B, 3.51,31.44,2.91,B, 2.47,55.41,3.91,K, 2.41,49.01,4.22,F, 3.05,49.25,4.17,G, 3.06,44.4,4.6, 3.02,38.39,3.3,M	61290 DATA 6.23,-52.4,-0.86,F, 9.13,-69.31,1.8,A, 8.22,-59.21,1.74,K, 10.41,-64.08,3.03,B, 9.16,-59.04,2.25,F, 9.46,-64.5,3.15,F, 7.56,-52.51,3.6,B, 10.13,-69.47,3.56,B, 9.1,-58.46,3.56,B, 10.15,-61.05,3.44,K, 9.31,-62.34,3.9,M
60880 DATA 2,2.31,3.94,A, 23.01,3.33,4.58,B, 23.15,3.01,3.85,G, 0.46,7.19,4.55,K, 1.7,37.4,4.5,G, 1.29,15.05,3.72,G, 23.25,6.06,4.45,K, 23.37,5.21,4.28,F, 1.43,8.54,4.5,G, 23.57,6.35,4.03,F	61310 DATA 8.2,-76.46,4.08,F, 12.15,-79.02,4.38,B, 10.35,-78.21,4.1,M, 10.45,-80.17,4.62,B, 8.22,-77.19,4.26,K
60900 DATA 19.38,17.54,4.37,F, 19.39,17.22,4.45,G, 19.57,19.21,3.71,M, 19.45,18.25,3.78,M	61330 DATA 12.34,-68.52,2.94,B, 12.43,-67.5,3.26,B, 12.3,-71.51,4.04,B, 12.59,-71.17,3.63,K, 12.15,-67.41,4.16,M, 11.43,-66.27,3.8,A
60920 DATA 19.27,24.34,4.63,M, 19.41,21.18,4.6,B, 19.51,23.57,4.5,A, 19.59,27.37,4.74,A	61350 DATA 9.02,-66.12,4.18,A, 8.25,-65.58,3.65,K, 7.08,-70.25,5.81,G, 7.11,-70.25,3.87,F, 7.17,-67.52,4.02,F, 8.08,-68.28,4.46,B, 7.43,-72.29,3.89,K
60940 DATA 19.2,-40.43,4.11,B, 19.19,-44.33,4.24,B, 19.2,-44.54,4.51,A, 18.03,-30.26,3.07,K, 18.18,-29.51,2.84,K, 18.21,-34.25,1.82,B, 18.59,-29.57,2.71,A, 18.14,-36.47,3.16,M, 18.25,-25.27,2.94,K, 18.55,-21.1,3.61,K, 18.52,-26.22,2.14,B	61370 DATA 14.36,-60.38,0.06,G, 14,-60.08,0.86,B, 12.39,-48.41,2.38,A, 12.06,-50.27,2.88,B, 13.37,-55.13,2.56,B, 13.52,-47.03,3.06,B, 14.32,-41.56,2.65,B, 14.04,-36.07,2.26,G, 13.18,-36.27,2.91,A, 14.56,-41.54,3.35,B
60950 DATA 18.11,-21.04,3.79,B	61390 DATA 12.24,-62.49,1.05,B, 12.45,-59.25,1.5,B, 12.28,-56.5,1.6,M, 12.13,-58.28,3.08,B, 12.19,-60.08,3.57,K, 12.52,-56.54,3.95,B
60970 DATA 16.26,-26.19,0.88,M, 16.03,-19.4,2.9,B, 15.57,-22.29,2.54,B, 16.47,-34.12,2.36,G, 16.51,-42.17,3.75,K, 17.09,-43.11,3.44,A, 17.34,-42.58,2.04,F, 17.39,-39.2,2.51,B, 17.3,-37.04,1.71,B, 16.49,-37.58,3,B	61410 DATA 19.06,-37.59,4.12,A, 19.07,-39.25,4.16,G, 19.03,-37.08,4.26,F
60990 DATA 15.42,6.35,2.75,K, 15.44,15.35,3.74,A, 15.54,15.49,3.86,F, 15.32,10.42,3.85,A, 15.48,4.38,3.75,A, 18.19,-2.55,3.42,G, 18.54,4.08,4.1,A, 15.47,-3.17,3.63,A, 17.18,-12.48,4.33,A, 17.35,-15.22,3.64,A	61430 DATA 14.39,-47.11,2.89,B, 14.55,-42.56,2.81,B, 15.32,-41.2,95,B, 15.18,-40.28,3.43,B, 15.19,-44.31,3.74,B, 15.09,-51.55,3.5,G, 15.57,-38.15,3.64,B, 14.16,-45.5,4.1,B, 15.09,-48.33,3.97,B
51010 DATA 4.33,16.25,0.85,K, 5.23,28.34,1.65,B, 4.17,15.31,3.86,G, 4.2,17.26,3.93,G, 4.26,19.04,3.63,G, 5.35,21.07,3,B, 3.45,23.57,2.96,B, 4.26,15.51,4.04,G, 4.26,15.46,3.62,A, 3.58,12.21,3.5,B, 4.13,8.46,4.32,B	61450 DATA 16.16,-50.02,4.14,G, 16.03,-45.02,4.84,A, 16.24,-47.27,4.8,B, 16,-49.06,4.74,G
51030 DATA 11.01,62.01,1.8,K, 10.59,56.39,2.44,A, 11.51,53.58,2.54,A, 12.13,57.19,3.44,A, 12.52,56.14,1.78,A, 13.22,55.11,2.17,A, 13.46,49.34,1.87,B, 9.3,51.54,3.26,F, 8.56,48.14,3.12,A, 11.07,44.46,3.15,K	61470 DATA 4.33,-55.09,3.47,A, 5.33,-62.31,4.03,F, 4.15,-51.37,4.36,F, 5.45,-65.45,4.52,A, 5.14,-67.14,4.78,K
51050 DATA 13.23,-10.54,1,B, 11.48,2.03,3.8,F, 12.39,-1.11,2.91,F, 12.53,3.4,3.66,M, 13,11.14,2.95,G, 13.32,-0.2,3.44,A, 12.17,-0.23,4,A, 13.07,-5.16,4.44,A, 14.13,-5.46,4.16,F, 14.41,-5.27,3.95,F	61490 DATA 6.12,-74.44,5.14,G, 5.03,-71.23,5.3,G, 5.34,-76.23,5.06,K, 4.57,-75.01,5.28,K
61070 DATA 10.25,-30.49,4.42,M, 9.27,-35.44,4.64,M, 10.54,-36.52,4.7,G	61510 DATA 4.14,-62.36,3.36,G, 4.44,-64.58,3.8,G, 4,-62.18,4.46,M, 3.58,-61.32,4.41,M, 4.16,-59.25,4.42,K
61090 DATA 8.42,-33,3.7,B, 8.38,-35.08,4.04,G, 8.48,-27.31,4.19,K	61530 DATA 3.1,-29.11,3.95,F, 2.47,-32.37,4.5,G, 3.4,-32.06,4.93,B, 2.02,-29.32,4.74,A
61110 DATA 14.42,-78.50,3.81,K, 16.36,-77.25,4.16,G, 16.26,-78.47,3.9,K, 16.13,-78.34,4.78,M, 16.13,-78.33,5.22,M	61550 DATA 22.05,-47.12,2.16,B, 22.4,-47.09,2.24,M, 21.51,-37.36,3.16,B, 22.26,-43.45,4.02,G, 22.27,-44,4.31,M, 22.46,-51.35,3.69,A, 22.58,-53.01,4.18,G, 23.04,-43.47,4.35,F, 23.08,-45.31,4.1,G
61130 DATA 14.38,-64.46,3.41,F, 15.14,-58.37,4.16,A, 15.19,-59.09,4.54,B	61570 DATA 20.47,-33.58,5.6, 20.58,-32.27,4.71,G, 21.15,-32.23,4.79,A, 21.18,-41.01,4.92,A



Ausschnitt des Gesamthimmels vom 28. 4. 1968 mit Stier Rektaszension: 6.096 h bis 2.5 h

```

61710 DATA 0.56,-29.38,4.39,B, 23.3,-38.06,4.46,B,
      23.16,-32.48,4.51,G, 23.46,-28.24,4.64,A
61730 DATA 6.48,-61.53,3.3,A, 5.46,-51.05,3.94,A,
      5.49,-56.11,4.38,K
61750 DATA 7.51,-40.27,3.76,G, 8.02,-39.52,2.27,O,
      7.37,-26.41,3.81,B, 6.36,-43.09,3.18,B, 7.4
      7,-24.44,3.47,G, 7.15,-37,2.74,K, 8.05,-24.1
      ,2.88,F, 7.28,-43.12,3.27,M, 6.49,-50.33,2.8
      3,G, 7.44,-37.51,3.72,K, 7.12,-44.33,2.6,M
61770 DATA 8.08,-47.12,2.12,B, 8.43,-54.31,2.01,A,
      9.21,-54.48,2.63,B, 9.06,-43.14,2.22,K, 10.
      45,-49.09,2.84,G, 8.39,-52.45,3.68,B, 9.55,-
      54.2,3.7,B, 9.29,-40.15,3.64,A, 8.39,-46.28,
      4.06,F, 9.02,-46.54,3.69,K
61772 DATA 16.18,76,5,X, 10.43,-59.25,4,X, 17.31,5
      5.13,4.26,A
61780 DATA 0,0,0,STERNEDE
62000 REM
62010 d=nt
62040 ON p1+1 GOSUB 62250,62260,62270,62250,62280,
      62290,62300,62310,62320,62330,62340
62050 s=j:p=k:GOSUB 62160:b=g+n:a=m+ATN(COS(1)*TAN
      (b)):IF COS(b)<0 THEN a=a+180
62060 b=ATN(SIN(1)*SIN(b)):la=a:br=b:j=1:k=0.01672
      :x=723625:rd=f:IF p1=0 THEN rd=0:la=0
62070 e=f*COS(b):GOSUB 62160:c=e:IF p1=0 THEN c=0:
      e=f:b=0.000001
62080 IF p1<>3 THEN 62110
62090 a$="Erde":br=0:ra=0:de=0:en=0:rd=f:la=g+101.
      22:e1=0:eb=0:IF la>360 THEN la=la-360
62100 RETURN
62110 d=g+101.22:x=c:y=a:GOSUB 62240:a=x:c=y:x=f:y
      =d:GOSUB 62240:c=c-y:a=a-x:x=a:y=c:GOSUB 621
      90
62120 d=x:f=y:c=ATN(e/d*TAN(b)):eb=c:e1=f:g=d/COS(
      c):en=g:c=ATN(SIN(f)/TAN(c)):IF f<0 THEN e1=
      f+360
62130 e=ATN(SIN(c-23.45)/SIN(c)*TAN(f)):IF COS(f)<
      0 THEN e=e+180
62140 j=e/15:IF e<0 THEN e=e+360:j=j+24
62150 d=ATN(SIN(e)/TAN(c-23.45)):ra=j:de=d:RETURN
62160 i=0.985609166/j^1.5*(d-x)/360+1000:i=360*(i-
      INT(i)):t=i
62170 c=SIN(t)*180/PI*k+i:IF ABS(c-t)>0.0001 THEN
      t=c:GOTO 62170
62180 g=TAN(FN ac(k)/2):g=2*ATN(TAN(c/2)/g):f=j*(C
      OS(c)-k)/COS(g):RETURN
62190 GOSUB 62200:x=rr:y=th:RETURN
62200 rr=SQR(x*x+y*y):IF x=0 THEN th=90*SGN(y):RET
      URN
62210 th=ATN(y/x):IF x<0 THEN th=th+180
62220 IF th>180 THEN th=th-360
62230 RETURN
62240 rr=x*COS(y):y=x*SIN(y):x=rr:RETURN
62250 a$="Sonne":j=1:k=0:l=0:m=0:n=0:x=0:RETURN
62260 a$="Merkur":j=0.3871:k=0.20563:l=7.004:m=48.
      04:n=29.03:x=723219.6:RETURN
62270 a$="Venus":j=0.72333:k=0.006785:l=3.394:m=76
      .44:n=54.79:x=723566.7:RETURN
62280 a$="Mars":j=1.5237:k=0.09338:l=1.85:m=49.37:
      n=286.24:x=721596.3:RETURN
62290 a$="Jupiter":j=5.20225:k=0.048:l=1.306:m=100
      .2:n=273.78:x=721659.3:RETURN
62300 a$="Saturn":j=9.55415:k=0.05635:l=2.488:m=11
      3.47:n=338.66:x=721090:RETURN
62310 a$="Uranus":j=19.1323:k=0.04526:l=0.772:m=73
      .93:n=94.32:x=718277:RETURN
62320 a$="Neptun":j=29.985:k=0.01158:l=1.771:m=131
      .59:n=273.92:x=687416:RETURN
62330 a$="Pluto":j=39.293:k=0.2456:l=17.14:m=109.9
      6:n=114.08:x=636923:RETURN
62340 a$="Halley":j=17.95:k=0.9672788:l=162.24:m=5
      8.145:n=111.848:x=725489.5:RETURN
    
```



Hardcopies von Skyplot



Gesamthimmel am 7. 7. 1959 Breite 51.17° 12.30 Uhr



Gesamthimmel am 28. 4. 1962 Breite 51.17° 12.00 Uhr mit großem Bär



# Zeno's Turtle

## Helfen Sie Zenoboy beim Hüten seiner Schildkröten

---

### Programmaufbau

---

'Zeno's Turtles' ist ein reines BASIC-Programm. Der einige Male auftauchende CALL &bb06 bewirkt nur ein Warten auf den nächsten Tastendruck. Dennoch wirken sich Tippfehler hier ärgerlich aus. Besondere Aufmerksamkeit ist auch dem Dateien-Teil zuzuwenden. Zeilen 2850-3090: In diesen Zeilen sind die Koordinaten für den Titelplot zusammengefaßt. Eine Fehlersuche ist hier weniger schlimm, als es aussieht. Die Daten haben das Format

x-Koord. (abs. Startwert), y-Koord. (abs. Startwert), Plotrichtung, Koord. Zielwert (x oder y abs). Da in 2-Pixel-Abständen geplottet wird und der Zielwert abgefragt wird, muß die Differenz durch 2 teilbar sein, sonst kommt es zu Ausbrüchen bei PLOT. Dies bedeutet aber in jedem Fall: Tippfehler! Zeilen 3480-3610: Das Format der LOCATE-Daten weist eine Besonderheit auf. Anders als beim LOCATE-Befehl wird hier erst die

---

### LOCATE

---

Zeile und dann die Spalte angegeben (mit führenden Nullen). Dahinter kommt die Nummer eines Charakters. Sollte Zeile 2100 eine Fehlermeldung verursachen, obwohl sie korrekt getippt ist, so ist der Fehler in den Daten zu suchen. Das Programm wurde in Mode 2

geschrieben. Bei Nachahmung empfiehlt sich, INK 0,25 und INK 1,0 zu wählen (PAPER und PEN je nach Geschmack und Beleuchtung). Beim Eintippen der Datas erleichtern Zwischenzeilen, die später entfernt werden können, die Übersicht. Nach dem Eintippen das Programm bitte erst abspeichern, die Anlage zurückset-

---

### Das Abenteuer

---

zen und neu laden. Sollten unerwünschte Farb- und Printeffekte auftreten, liegt ein Fehler in der Grafik-Text-Umschaltung vor. In diesem Fall bitte kontrollieren, ob alle TAG-, TAGOFF-, PRINT CHR\$(23); CHR\$(0)- und PRINT CHR\$(23); CHR\$(1)-Anweisungen vorhanden sind und richtig angesprungen werden.

---

### Zenoboy

---

Zenoboy, ein Aussteigertyp mit einer Nase, die Cyrano de Bergerac zur Ehre gereicht hätte, hatte wieder einmal kein Glück. Zenoboy's zu lang geratene Nase und seine spezielle Vorliebe zu den Geschichtchen um Zeno, den Älteren (irgendein alter Grieche), und seltsamen Wettläufen mit Schildkröten hatten ihm schon früh zahllose Hänseleien eingebracht. Als sich der dieserart Geplagte dann auch noch dem Zen-Buddhismus verschrieb, bekam er seinen Spitzna-

men: 'Zenoboy', was den Ärmsten nun endgültig ins Aussenseitertum trieb. Doch dann, nach langen Jahren der Suche, erfüllte sich sein langgehegter Wunsch, sein Leben in Ruhe und Besinnlichkeit auf einer einsamen kleinen Insel verbringen zu können. Pustekuchen! So etwas gibt es nur in schönen alten Geschichtchen. Die Insel seiner Wahl war vulkanischen Ursprungs, der Vulkan tüchtig aktiv geworden und Zenoboy fliegt die glühende Asche um die Ohren! Entsetzt rennt er los, um seine Lieblinge, die griechischen Landschildkröten, in Sicherheit zu bringen, als ihm kurz vor dem Ziel der

---

### Lavaglut

---

Weg durch glutflüssige Lavamassen abgeschnitten wird. Aber es gibt in der Lava einige kältere Stellen. Wenn man nun von 'Scholle zu Scholle' hüpfet... Genau dies ist die Aufgabe der Spieler. Mitspielen können bis zu 6 Personen. Gesteuert wird mit den Cursortasten. Bemerkungen: Der in der einen oder anderen Runde niedergehende Aschenregen ist nicht so harmlos, wie es zunächst scheint. Wenn Zenoboy oder Turtles verschwinden, so hat es sie an ihrer empfindlichsten Stelle erwischt. (?????) Sollte einmal keine Möglichkeit gegeben sein, die Lava zu passieren, bitte CTRLR drücken (nicht zu oft!). Das Hüpfen von Stein zu Stein auf horizontaler Ebene ist nicht möglich. Trotz 'Mensch-laß-dich-nicht-Nerven'-Effekt: Viel Spaß!

```

10 '-----
15 '-----
20 ' ZENO'S Turtles
25 ' Copyrights (1985) by
30 ' W.Ruschewyh
35 ' Bad Pymont
40 '-----
45 ' f)r Schneider CPC 464 ---
50 '-----
100 GOSUB 2550
110 MODE 0 :DEFINT a-z :fw=2
120 INK 0,0 :INK 1,6 :INK 2,3 :INK 3,25 :INK 4,15
:INK 5,18 :INK 12,24
130 BORDER 0 :PLOT 0,-2,4
140 ' --- dedicated ---
150 TAG :FOR m=370 TO 260 STEP -2
160 MOVE 244,m :PRINT"f)r";:MOVE 220,m+17:PRINT
";
170 NEXT :TAGOFF
180 LOCATE 4,13:PEN 3:PRINT "COMPUTER"
190 LOCATE 5,16:PEN 1:PRINT "T E A M";:LOCATE 1
6,15:PEN 5:PRINT CHR$(132);
200 FOR i=1 TO 8000 :NEXT
210 FOR y=120 TO 270 STEP 2 :PLOT 16,y:DRAW 624,y,
0 :SOUND 1,y-88,2,0,6:NEXT
220 EVERY 50,2 GOSUB 2810
230 ' --- Titel-Plot ---
240 RESTORE
250 READ x,y,idx$,ziel
260 IF idx$="xp" THEN GOSUB 300 ELSE IF idx$="xn"
THEN GOSUB 310
270 IF idx$="yp" THEN GOSUB 320 ELSE IF idx$="yn"
THEN GOSUB 330
280 IF idx$="ff" THEN GOTO 340
290 GOTO 250
300 PLOT x,y : x=x+2 :IF x=ziel THEN RETURN
ELSE GOTO 300
310 PLOT x,y,1 : x=x-2 :IF x=ziel THEN RETURN
ELSE GOTO 310
320 PLOT x,y : y=y+2 :IF y=ziel THEN RETURN
ELSE GOTO 320
330 PLOT x,y : y=y-2 :IF y=ziel THEN RETURN
ELSE GOTO 330
340 LOCATE 18,6 :PEN 2 :PRINT CHR$(64)
350 ' --- Copyrights ---
360 LOCATE 4,15:PEN 3:PRINT "T u r t l e s"
370 LOCATE 4,17:PEN 2:PRINT "Computer-Game";: LOCA
TE 9,20:PEN 4:PRINT "von"
380 LOCATE 3,22:PEN 6:PRINT "Willi Ruschewyh"
390 LOCATE 12,25:PRINT CHR$(164);"10/1985"
400 INK 1,26,6:INK 2,25,9:SPEED INK 30,50:FOR i=1
TO 10000:NEXT
410 TAG : FOR m=0 TO 15 STEP 2
420 MOVE 176,m :PRINT "-Taste-" :NEXT
430 SOUND 1,213,15,0,3,2:SOUND 3,63,15,0,3,2:SOUND
2,142,15,0,3,2
440 TAGOFF :PEN 0:PRINT CHR$(30);REMAIN (2)
450 :CALL &BB06
460 FOR y=400 TO 0 STEP -2 :SOUND 1,y+32,1,4:PLOT
0,y:DRAW 640,y,0 :NEXT
470 '---- Instruct. ---
480 MODE 1:INK 1,25 :INK 2,6 :INK 3,21

```



```

490 LOCATE 14,12:PEN 2:PRINT "Beschreibung"
500 FOR i=1 TO 3000 :NEXT :CLS
510 LOCATE 16,4:PEN 2:PRINT "Spielziel"
520 LOCATE 16,5:PEN 3:PRINT "-----";
530 RESTORE 3100 :PEN 1
540 FOR t=0 TO 16 :READ text$ :LOCATE 3,7+t
:PRINT text$:NEXT
550 LOCATE 16,25:PEN 3:PRINT "--Taste--";
:CALL &BB06
560 LOCATE 3,24: PRINT CHR$(19);:LOCATE
15,3:PEN 2 :PRINT "H)pf-Technik"

```

```

570 LOCATE 15,4:PEN 3
:PRINT "-----"
580 RESTORE 3280 :PEN 1
590 FOR t=0 TO 17 :READ text$ :LOCATE 3,6+t:PRINT
text$:NEXT
600 :CALL &BB06
610 ' --- Mitspieler ---
620 DIM nam$(6),scr(6)
630 CLS:LOCATE 4,4 :PEN 2:PRINT "Wieviele Mitspiel
er [max. 6]";:INPUT anz
640 FOR i=1 TO anz
650 LOCATE 4,6+2*i :PEN 3:PRINT "Name des ";i;". M
itspielers";:INPUT nam$
660 sp=i :nam$(sp)=nam$ :NEXT
670 PRINT:PRINT CHR$(19);:LOCATE 8,4:PEN 2:PRINT "
Mitspielerliste"
680 FOR i=1 TO anz :LOCATE 6,6+2*i:PEN 3:PRINT "Sp
ieler ";i:nam$(i);:NEXT
690 LOCATE 8,22:PEN 1:PRINT "Alles richtig ?
[j/n]"
700 a$=INKEY$:IF a$="" THEN 700
710 IF a$="j" OR a$="J" THEN 730 ELSE ERASE nam$,
scr:GOTO 620
720 ' --- Spielturnus ---
730 CLS:LOCATE 4,6:PEN 3:PRINT "Spielzeitbegrenzun
g"
740 LOCATE 4,7:PEN 1:PRINT "-----";:
PEN 2
750 LOCATE 4,10:PRINT "Anzahl der Runden ";:INPU
T rnz
760 LOCATE 4,13:PRINT "Anzahl der Chancen ";:INPU
T chanc
770 LOCATE 4,16:PRINT "Bonusabzug je Turtle";:INPU
T bnab
780 LOCATE 18,19:PEN 3:PRINT "Okay ? [j/n]"
790 a$=INKEY$:IF a$="" THEN 790
800 IF a$="J" OR a$="j" THEN 810 ELSE 730
810 sp=0 :rund=1
820 ' --- HAUPTPROGRAMM ---
830 MODE 0 :cc=chanc :snap=0
840 INK 0,0:INK 1,25:INK 5,3 :INK 7,9:INK 14,3,25
:SPEED INK 10,5
850 WINDOW#2,1,20,1,25 :PAPER#2,1
860 WINDOW#1,1,20,1,25
870 WINDOW#3,1,20,1,1 :PAPER#3,1
880 sp=sp+1 :IF sp=anz+1 THEN sp=1 :GOTO 2130 'Ru
ndenwechsel
890 fw2=fw2+1:IF fw2=2 THEN fw2=3 ELSE IF fw2>12 T
HEN fw2=1 :CLS
900 LOCATE 5,10:PEN fw2:PRINT nam$(sp);:LOCATE 5,1
3:PEN 3:PRINT "ist dran ...";
910 LOCATE#3,10,1:PEN#3,7:PRINT#3," Runde ";rund;
920 FOR i=1 TO 4000:NEXT:CLS
930 GOSUB 2010 :GOSUB 2060

```

```

940 IF rund MOD 2=0 THEN GOSUB 2430 ELSE IF rund M
OD 2=1 THEN GOSUB 2470
950 bn=300 :trx=-64 :try=271 :GOSUB 1950
960 ' === animation ===
970 PRINT CHR$(23);CHR$(1)
980 TAG
990 fa=249 :wem=250 :osm=248 :exm=250 :mx=640: ny=
63
1000 PLOT 0,402,4
1010 IF TEST (mx+16,ny-33)=2 THEN GOSUB 1610 :cc=
cc-1 :GOSUB 1950 :IF cc=0 THEN 830 ELSE 970
1020 IF ny>415 AND mx>575 THEN GOSUB 1880 :GOTO 83
0
1030 IF ny<48 AND mx<161 THEN GOSUB 1880 :GOTO 83
0
1040 IF ny<0 OR ny>415 THEN GOSUB 1610 :GOTO 830
1050 IF mx<-32 OR mx>672 THEN GOSUB 1610 :GOTO 830
1060 trx=trx+1:MOVE trx,try :PRINT CHR$(132); :MOV
E trx-1,try :PRINT CHR$(132);
1070 IF TEST(trx-2,try+2)=2 THEN GOSUB 1870 :GOS
UB 1920 :GOSUB 1950
1080 IF mx=trx AND ny-16=try THEN SOUND 1,32,6,7:M
OVE trx,try :PRINT CHR$(132); :GOSUB 1910 :GOSUB
1950

```

```

1090 IF INKEY(16)=0 THEN GOSUB 2470
1100 IF INKEY(8)=0 THEN 1170
1110 IF INKEY(1)=0 THEN 1230
1120 IF INKEY(0)=0 THEN 1290
1130 IF INKEY(2)=0 THEN 1350
1140 IF INKEY(9)=0 THEN 1730
1150 IF bn<0 THEN 1690 ELSE GOTO 1000
1160 ' ---- < west ----
1170 MOVE 603,382:PRINT "<";WHILE INKEY#="" :WEND:
IF INKEY(8)=0 THEN 1180 ELSE 1410
1180 mx=mx-16
1190 MOVE mx,ny :PRINT CHR$(wem):: MOVE mx,n
y-16 :PRINT CHR$(fa);
1200 MOVE mx+16,ny :PRINT CHR$(exm):: MOVE mx+1
6,ny-16 :PRINT CHR$(fa);
1210 SOUND 2,338,15,0,6,2,2 :exm=wem : GOTO 1
000
1220 ' ---- ost > ----
1230 MOVE 615,382:PRINT ">";WHILE INKEY#="" :WEND:
IF INKEY(1)=0 THEN 1240 ELSE 1510
1240 mx=mx+16
1250 MOVE mx,ny :PRINT CHR$(osm):: MOVE mx,n
y-16 :PRINT CHR$(fa);
1260 MOVE mx-16,ny :PRINT CHR$(exm):: MOVE mx-1
6,ny-16 :PRINT CHR$(fa);
1270 SOUND 1,239,15,0,6,2,2 :exm=osm : GOTO 1
000
1280 ' ---- down/up ----
1290 WHILE INKEY#="" :WEND:IF INKEY(0)=0 THEN 1300
ELSE 1000
1300 ny=ny+16
1310 MOVE mx,ny :PRINT CHR$(exm):: MOVE mx,n
y-16 :PRINT CHR$(fa);
1320 MOVE mx,ny-16 :PRINT CHR$(exm):: MOVE mx,n
y-32 :PRINT CHR$(fa);
1330 SOUND 1,63,2,6 : GOTO 1000
1340 ' ---- up/down ----
1350 WHILE INKEY#="" :WEND:IF INKEY(2)=0 THEN 1360
ELSE 1020
1360 ny=ny-16
1370 MOVE mx,ny :PRINT CHR$(exm):: MOVE mx,n
y-16 :PRINT CHR$(fa);
1380 MOVE mx,ny+16 :PRINT CHR$(exm):: MOVE mx,n
y :PRINT CHR$(fa);
1390 SOUND 2,225,2,6 : GOTO 1000
1400 ' ----
1410 IF INKEY(0)=0 THEN 1420 ELSE IF INKEY(2)=0 TH
EN 1460 ELSE 1490
1420 mx=mx-32 :ny=ny+16
1430 MOVE mx,ny :PRINT CHR$(exm):: MOVE mx,n
y-16 :PRINT CHR$(fa);
1440 MOVE mx+32,ny-16 :PRINT CHR$(exm):: MOVE mx+3
2,ny-32 :PRINT CHR$(fa);
1450 SOUND 1,124,2,12 : GOTO 1000
1460 mx=mx-32 :ny=ny-16
1470 MOVE mx,ny :PRINT CHR$(exm):: MOVE mx,n
y-16 :PRINT CHR$(fa);
1480 MOVE mx+32,ny+16 :PRINT CHR$(exm):: MOVE mx+3
2,ny :PRINT CHR$(fa);
1490 SOUND 1,56,2,6 : GOTO 1000
1500 ' ----
1510 IF INKEY(0)=0 THEN 1520 ELSE IF INKEY(2)=0 TH
EN 1560 ELSE 1590
1520 mx=mx+32 :ny=ny+16
1530 MOVE mx,ny :PRINT CHR$(exm):: MOVE mx,n
y-16 :PRINT CHR$(fa);
1540 MOVE mx-32,ny-16 :PRINT CHR$(exm):: MOVE mx-3
2,ny-32 :PRINT CHR$(fa);
1550 SOUND 2,67,2,6 : GOTO 1000
1560 mx=mx+32 :ny=ny-16
1570 MOVE mx,ny :PRINT CHR$(exm):: MOVE mx,n
y-16 :PRINT CHR$(fa);
1580 MOVE mx-32,ny+16 :PRINT CHR$(exm):: MOVE mx-3
2,ny :PRINT CHR$(fa);
1590 SOUND 1,1014,2,12 : GOTO 1000
:END
1600 ' ---- Bad ----
1610 MOVE mx,ny :PRINT CHR$(exm):: MOVE mx,ny-16 :
PRINT CHR$(fa);
1620 MOVE mx,ny :PRINT CHR$(exm):: MOVE mx,ny-16 :
PRINT CHR$(253);
1630 FOR i=1 TO 7 : d=d+16 :SOUND 1,142+d,10,5
,3,2
1640 MOVE mx,ny-16 :PRINT CHR$(254);:MOVE mx,ny-16
:PRINT CHR$(253);:NEXT
1650 MOVE mx,ny :PRINT CHR$(exm);
1660 TAGOFF :PRINT CHR$(23);CHR$(0);:DI:FOR i=
1 TO 3000 : NEXT
1670 adtu=0 :RETURN :END
1680 ' --- Bonus null ---
1690 TAGOFF :MODE 0
1700 FOR i=1 TO 3:SOUND 1,124,10,0,5,4 :SOUND 1,32
,15,0,3,2:SOUND 4,224,10,0,3,2:NEXT
1710 LOCATE 4,12:PEN 2:PRINT "Bonus 0";:LOCATE 4,1
4:PEN fw2:PRINT nam$(sp) "!!";:FOR i=1 TO 4000:NEXT
:GOTO 830
1720 ' ---- Catch und Set ----
1730 SOUND 2,198,15,12,3,2
1740 IF snap=1 AND TEST (mx+9,ny-33)=8 THEN 1750
ELSE 1000
1750 snap=0 :PLOT 666,444,1 :adtu=adtu+1 :pkt(sp)=
pkt(sp)+15
1760 MOVE mx,ny-16 :PRINT CHR$(132);:SOUND 1,213,1
5,12,5,4
1770 IF adtu=7 THEN GOTO 1800
1780 GOSUB 1950 :GOTO 1000 :END
1790 ' --- Turtle Canasta ---
1800 tse(sp)=tse(sp)+1 :pkt(sp)=pkt(sp)+((bnab*7)*
cc)+bn :bn=300 :MODE 0
1810 TAGOFF :LOCATE 3,12:INK 1,25,3:PEN 1:PRINT "T
urtle Canasta";
1820 LOCATE 3,14:PEN 3:PRINT "f)r";
1830 LOCATE 8,16:PEN fw2:PRINT nam$(sp);
1840 FOR i=1 TO 5:SOUND 1,73,15,0,5,4:SOUND 1,142,
15,0,5,4:SOUND 1,32,15,0,5,4:NEXT
1850 FOR i=1 TO 7000:NEXT : adtu=0:GOTO 830 :END
1860 ' --- Turtle weg + Schummelecke ---
1870 FOR i=1 TO 5 :SOUND 2,32*i,i,6:NEXT:MOVE trx,
try:PRINT CHR$(132);:RETURN
1880 FOR i=7 TO 1 STEP -1 :SOUND 4,224-i*32,30,0,3
,2:NEXT :MODE 0 :adtu=0
1890 LOCATE 6,12:PEN 3:PRINT "pfuuuu !!";:LOCATE 6
,14:PEN fw2:PRINT nam$(sp) "...";:FOR i=1 TO 4000:
NEXT:RETURN :END
1900 ' ---- new Turtle ----
1910 pkt(sp)=pkt(sp)+10 :snap=1
1920 bn=bn-bnab :trx=-64 :try=INT(RND(1)*7+10)*16-
1
1930 RETURN :END
1940 ' --- Score + Time ---
1950 LOCATE#2,1,25:PEN#2,7:PRINT#2,"Score";USING "
####":pkt(sp);
1960 LOCATE#2,12,25:PRINT#2,"Set ";USING " #":adt
u;
1970 SOUND 2,2024,3,4,,2
1980 LOCATE#3,1,1:PRINT#3,"Bonus ";USING "####":bn
1990 LOCATE#3,15,1:PRINT#3,cc; :RETURN :END
2000 ' ---- Bach ----
2010 x=2 :y=28
2020 rma=INT(RND(1)*5)-1
2030 PLOT x+rma,y,2 :DRAW x+230+rma,y
2040 x=x+2 :IF y>382 THEN RETURN ELSE y=y+2 :GOT
0 2020
2050 ' - - - READ ROUTINE - - -
2060 RESTORE 3470
2070 READ vy,hx,cha
2080 IF cha=333 THEN RETURN ELSE 2090
2090 IF cha=217 THEN fw=1 ELSE IF cha=137 THEN fw=
3 ELSE IF cha=148 OR cha=151 OR cha=152 THEN fw=2
ELSE IF cha=207 OR cha>220 THEN fw=8
2100 LOCATE#1, hx+1,vy:PEN#1, fw:PRINT#1, CHR$(cha
);:GOTO 2070
2110 END
2120 ' ---- Score/Player ----
2130 TAGOFF
2140 CLS :LOCATE 4,13 :PEN 3 :PRINT "Ende Runde ";
rund;:FOR i=1 TO 4000:NEXT
2150 MODE 1:PRINT CHR$(23);CHR$(0):adtu=0 :snap=0
:rund=rund+1
2160 FOR i=1 TO anz :scr(i)=pkt(i) :NEXT :CLS
2170 LOCATE 10,4:PEN 1:PRINT "High-Score"
2180 LOCATE 2,6:PEN 2:PRINT "Spieler Cana
sta' Punkte "
2190 FOR i=1 TO anz
2200 LOCATE 2,6+2*i:PEN 3:PRINT nam$(i) TAB(22) ts
e(i);TAB (30) USING "####":scr(i);:NEXT
2210 score(h)=MAX(scr(1),scr(2),scr(3),scr(4),scr(
5),scr(6)): te=0
2220 te=te+1 :IF score(h)=scr(te) THEN 2230 ELSE 2
220
2230 INK 1,3,25:LOCATE 1,6+2*te:PEN 1:PRINT CHR$(1
32);
2240 LOCATE#1,14,25:PEN#1,2:PRINT#1, "- Taste -"
2250 CALL &BB06 :CALL &BB06
2260 IF rund>rnz THEN 2300 ELSE 2270

```

```

2270 MODE 0:LOCATE 4,13:PEN 14:PRINT "nächste Rund
e!";:FOR i=1 TO 4000:NEXT
2280 sp=0:GOTO 830
2290 ' --- Endabfrage ---
2300 INK 1,25:LOCATE 1,24:PRINT CHR$(19);
2310 LOCATE 6,5:PEN 2:PRINT "Wie soll's weitergehe
n?"
2320 LOCATE 6,8:PEN 3:PRINT "-- nächste Runde
 [w]"
2330 LOCATE 6,10:PRINT "-- neues Spiel [n]"
2340 LOCATE 6,12:PRINT "-- Regeln ändern [a]"
2350 LOCATE 6,14:PRINT "-- Ende [e]"
2360 LOCATE 12,17:PEN 2:PRINT "Bitte wählen ..."
2370 a$=INKEY$:IF a$="" THEN 2370:GOTO 2380
2380 IF a$="w" THEN sp=0:GOTO 830
2390 IF a$="a" THEN GOTO 730
2400 IF a$="n" THEN ERASE nam$,scr:GOTO 620
2410 IF a$="e" THEN 2540 ELSE 2370:END
2420 ' --- 2. Runde ---
2430 FOR i=1 TO 200:rmb=INT(RND*590+16):rnc=INT(
RND*380+16):PLOT rmb,rnc,2
2440 BORDER rmb MOD 2:SOUND 2,rnc+800,1,6,,3:BORDE
R rnc MOD 2:BORDER 0:NEXT
2450 RETURN:END
2460 ' --- 1. Runde ---
2470 hx=11:vy=4
2480 FOR i=1 TO 10
2490 rmb=INT(RND(1)*3):IF rmb=0 THEN rmb=3
2500 rnc=INT(RND(1)*7):r2=rnc+rnc:IF r2=14 THE
N r2=13
2510 IF rmb=1 OR rmb=2 THEN cha=143 ELSE cha=148
2520 LOCATE#1, hx+rnc,vy:PEN#1,2:PRINT#1, CHR$(cha
);
2530 hx=hx-1:vy=vy+2:NEXT i:RETURN
2540 MODE 1:END
2550 SYMBOL AFTER 62
2560 SYMBOL 64,192,159,129,141,171,172,250, 3
2570 SYMBOL 96,112,216,208,248,204,204,248,192
2580 SYMBOL 123,102, 0, 60,102,102,102, 60, 0
2590 SYMBOL 124,102, 0,120, 12,124,204,118, 0
2600 SYMBOL 125,102, 0,102,102,102,102, 62, 0
2610 SYMBOL 132,136,112,250,255,250,112,136, 0
2620 SYMBOL 137, 32, 34, 36,149, 90, 52, 24, 60
2630 SYMBOL 148,227,133,170, 46, 48, 84,104,131
2640 SYMBOL 217, 0, 28, 38,109,221,165,162,124
2650 SYMBOL 220, 49, 83,142,109, 42, 52,168,112
2660 SYMBOL 221,168,196, 74, 41, 18, 13, 4, 3
2670 SYMBOL 222, 3, 4, 13, 18, 33, 90,164,136
2680 SYMBOL 223,192,160,176, 72,132, 90, 37, 17
2690 SYMBOL 248,124,247,124, 40,100,122, 42, 60
2700 SYMBOL 249, 60, 90,189,165,255,165,255, 60
2710 SYMBOL 250, 62,239, 62, 20, 38, 70, 94, 52
2720 SYMBOL 252,129,102,153,102, 24, 65, 18, 24
2730 SYMBOL 253,148,170,169, 85,137,126,211,126
2740 SYMBOL 254,136, 20, 8, 52,74, 163, 66, 60
2750 ENV 3,14,-1,1, 10,-1,1
2760 ENT 2,14,-1,2, 10,-1,2, 5,-1,2
2770 ENV 5,32,-2,2, 16,2,2, 32,-2,2
2780 ENT 4,15,-1,4, 12,-1,3, 10,-1,2, 8,-1,1
2790 ENV 6,10,1,2,15,-1,2,10,1,2
2800 RETURN:END
2810 SOUND 1,134,15,0,6,,6:SOUND 1,169,15,0,6,,6:S
OUND 1,142,15,0,6,,6:RETURN:END
2820 ' === Plot-Data's ===
2830 ' --- 'Z' ---
2840 DATA 160,192,xn,064,064,192,yp,272,064,272,xp
,140,140,272,yp,304
2850 DATA 140,304,xn,064,064,304,yp,320,064,320,xp
,160,160,320,yn,256
2860 DATA 160,256,xn,080,080,256,yn,224,080,224,xp
,160,160,224,yn,192
2870 ' --- 'e' ---
2880 DATA 288,192,xn,192,192,192,yp,288,192,288,xp
,288,288,288,yn,240
2890 DATA 288,240,xn,224,224,240,yp,256,224,256,xn
,208,208,256,yn,224
2900 DATA 208,224,xp,288,288,224,yn,192,272,256,xn
,240,240,256,yp,272
2910 DATA 240,272,xp,272,272,272,yn,256
2920 ' --- 'n' ---
2930 DATA 336,192,xn,320,320,192,yp,288,320,288,xp
,336,336,288,yn,272
2940 DATA 336,272,xp,352,352,272,yp,288,352,288,xp
,416,416,288,yn,192
2950 DATA 416,192,xn,400,400,192,yp,272,400,272,xn
,368,368,272,yn,256
2960 DATA 368,256,xn,336,336,256,yn,192

```

```

2970 ' --- 'o' ---
2980 DATA 544,192,xn,448,448,192,yp,288,448,288,xp
,544,544,288,yn,192
2990 DATA 528,224,xn,464,464,224,yp,272,464,272,xp
,528,528,272,yn,224
3000 ' --- Apostroph und 's' ---
3010 DATA 564,288,xp,326,564,326,xn,540,540,326,yn
,298,540,298,xp,554
3020 DATA 554,298,yn,288,554,288,xp,564
3030 DATA 598,288,xn,576,576,288,yn,268,576,268,xp
,592,592,268,yn,256
3040 DATA 592,256,xn,576,576,256,yn,250,576,250,xp
,598,598,250,yp,274
3050 DATA 598,274,xn,582,582,274,yp,282,582,282,xp
,598,598,282,yp,288
3060 ' --- Kasten ---
3070 DATA 065,150,xp,545,545,150,yn,120,545,120,xn
,065,065,120,yp,150
3080 DATA 0,0,ff,0
3090 ' --- Beschreibung ---
3100 DATA "Zenoboy, er erscheint unten in der"
3110 DATA "rechten Ecke wenn man die Cursortaste"
3120 DATA " [links] betitigt, mu' sieben Turtles"
3130 DATA "in die Grube setzen, ohne dabei ein-"
3140 DATA "mal 'baden' zu gehen."
3150 DATA "Wie nahe man diesem Ziel tats'chlich"
3160 DATA "gekommen ist, dokumentiert der 'Set'-"
3170 DATA "Z'ihler. Geht man 'baden', so verliert"
3180 DATA "zwar vorher gemachte Punkte nicht,"
3190 DATA "setzt den 'Set'-Z'ihler aber wieder"
3200 DATA "auf 0. Wie oft man sich da' leisten"
3210 DATA "darf, wird durch 'Chance' festgelegt."
3220 DATA "Um eine Turtle zu fangen gen'gt es."
3230 DATA "sie genau in den K'fig, den Zenoboy"
3240 DATA "mit hat, hinlaufen zu lassen."
3250 DATA "Die COPY-Taste dient als Lockruf und"
3260 DATA "zum Absetzen der Turtles in die Grube."
3270 '
3280 DATA "Die F'hrung des Zenoboy ist nicht"
3290 DATA "ganz einfach, denn mit den vier Cur-"
3300 DATA "sortasten werden acht Bewegungsrich-"
3310 DATA "richtungen angesteuert."
3320 DATA "
3330 DATA "Hierzu nur drei Tips : "
3340 DATA "
3350 DATA "1. Oben, rechts im Spielfeld ist ein"
3360 DATA "Indikator, der anzeigt ob ein Tas-"
3370 DATA "tendruck 'gebongt' ist oder nicht."
3380 DATA "2. Bevor man h'pft, zun'chst die Ge-"
3390 DATA "genrichtung testen. Reagiert der"
3400 DATA "Indikator und findet keine Bewegung"
3410 DATA "statt, dann gilt : "
3420 DATA "Bewegungsrichtung (links/recht) vor"
3430 DATA "H'pf-Richtung (hoch/runter) !!"
3440 DATA "3. Zenoboy mu' auf die h'chste Stel-"
3450 DATA "le der Steine springen ! "
3460 ' ---- Locate und CHR*-Data's ----
3470 DATA 03,04,137,03,07,137,03,12,148,04,13,148,
04,18,137,05,03,137
3480 DATA 06,02,137,06,06,137,06,10,148,06,17,137,
07,03,137,07,12,148
3490 DATA 07,16,137,08,06,137,08,18,137,09,08,148,
09,11,148,10,03,137
3500 DATA 10,09,148,11,07,148,11,10,148,11,12,148,
11,16,222,11,17,223
3510 DATA 12,02,217,12,06,148,12,08,148,12,11,148,
12,15,222,12,16,207
3520 DATA 12,15,222,12,16,207,12,17,207,12,18,223,
13,03,137,13,09,148
3530 DATA 13,14,222,13,15,207,13,16,207,13,17,207,
13,18,220,14,02,137
3540 DATA 14,02,137,14,08,148,14,10,148,14,14,221,
14,15,207,14,16,207
3550 DATA 14,17,220,15,03,137,15,06,148,15,09,148,
15,15,221,15,16,220
3560 DATA 16,04,148,16,06,148,16,07,148,16,08,148,
16,10,148,16,13,217
3570 DATA 17,05,148,17,07,148,17,12,137,18,02,137,
18,04,148,18,06,148
3580 DATA 18,10,137,18,12,217,18,14,137,18,16,217,
19,06,148,19,18,137
3590 DATA 20,03,148,20,08,148,20,12,217,20,17,137,
21,04,148,21,10,217
3600 DATA 21,14,137,21,16,217,22,15,217,22,17,217,
23,16,217,01,01,333
3610 END : '-----

```

# Die Unterschiede zwischen CPC 464 und CPC 664



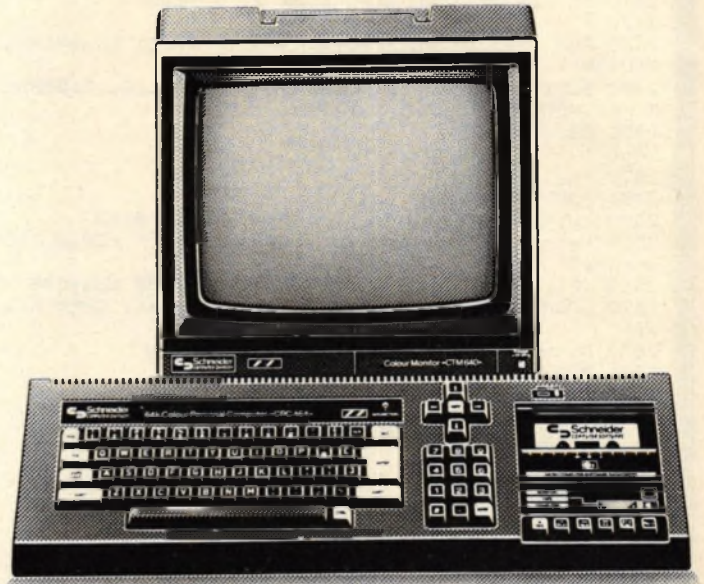
Schneider »CPC 664«  
mit Color-Monitor

Der einzige Unterschied zwischen den beiden Schneidercomputern scheint die Diskette zu sein, die beim 664 an der Stelle des Kassettenrekorders sitzt. Aber es gibt noch andere Unterschiede, so frappierende sogar, daß einige 464-Programme nicht auf dem 664 laufen.

Der Unterschied, der am meisten auffällt, nämlich die statt des Kassettenrekorders eingebaute Diskettenstation, ist noch der geringste. Denn obwohl in der Hardware des 664 einiges geändert wurde, so zum Beispiel eine leicht unterschiedliche Floppyschaltung, sind die beiden Computer hardwaremäßig absolut kompatibel (verträglich). „Na also, wo liegt dann das Problem?“, werden Sie jetzt vielleicht fragen. Tatsächlich gäbe es überhaupt kein Problem, wenn die Entwickler von Amstrad (dem englischen Hersteller) es damit belassen hätten. Leider aber beschloß man dort, das sowieso recht gute BASIC um einige neue Befehle zu erweitern, über deren Sinn man sich streiten kann. Das hatte zur Folge, daß viele Programme, die irgendwelche Unterprogramme aus dem BASIC benutzten, nicht mehr funktionierten. Ebenso solche Programme, die auf den Variablenbereich von BASIC zugreifen, d.h. auf den RAM-Block, in dem BASIC solche Daten wie HIMEM, Anfangs- und Endadresse eines BASIC-Programms etc. (zum Vergleich: die Startadresse eines BASIC-Programms steht beim 464 an den

Adressen &AE81 /&AE82, beim 664 an den &AE64/&AE65, die Endadresse folgt entsprechend in den beiden Bytes dahinter) ablegt. BASIC-Erweiterungen sind damit „prädestinierte Abstürzler“.

Auch das Betriebssystem wurde geändert. Vielen von Ihnen ist sicher bekannt, daß das ROM der Schneider-Computer in zweimal 16 K Byte aufgeteilt ist: Einmal das bereits erwähnte BASIC, das „oben“ im letzten Viertel des Speichers liegt und dann noch das Betriebssystem, das im ersten Viertel, also den untersten 16k Byte liegt. Und auch das Betriebssystem ist von den Änderungen nicht verschont geblieben. Um nämlich auch unterbrochene Linien zeichnen zu können (mit MASK), mußten die Grafikprogramme im Betriebssystem geändert werden, wodurch sich natürlich auch alle folgenden Unterprogramme verschoben. Daß die meisten alten 464-Programme trotz der Änderung im Betriebssystem noch immer auf dem 664 laufen, ist nur einem Trick zu verdanken: Die einzelnen Unterprogramme werden in beiden Computern nicht direkt angesprungen, sondern über Vektoren, also Sprungbefehle, die ihrerseits an



Schneider »CPC 464«  
mit Color-Monitor

die richtige Stelle springen. Wenn sich diese Ansprungsadresse ändert, macht das gar nichts, da zwar der Sprungbefehl geändert werden muß, aber immer noch an derselben Stelle steht.

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!"
```

Zusammenfassend kann man sagen, daß folgende Programmtypen besonders gefährdet sind, auf dem 664 nicht korrekt zu laufen, wenn sie für den 464 geschrieben wurden:

- BASIC-Erweiterungen, sofern sie nicht über RSX laufen,
- alle Programme, die Daten vom BASIC übernehmen,
- alle Programme, die nicht die Sprungvektoren benutzen.

BASIC-Programme werden in der Regel ohne Probleme laufen. Für den Diskettenbetrieb ergeben sich keine Probleme, da das DOS (Disc Operating System) gleich geblieben ist.



Wir haben 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. An dieser Stelle 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 Unterprogramme aus dem BASIC von &c000-&ffff verwenden, rufen beim 664 eine falsche Adresse auf, was die unvorhergesehensten Ergebnisse produzieren kann. Die

## BASIC-Routinen

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 Speicherplatz zu sparen und weil so eine höhere Geschwindigkeit erzielt werden kann. Eine solche Zahl nennt man TOKEN.) Nun, des Rätsels Lösung: Der 464 nutzte 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 Kennzeichnung, 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 mit dem Betriebssystem passiert? Auch dieses wurde 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-

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	C12B	NEW	B1
C0DF	COEA	AUTO	81	C7E3	C885	ON	B2
C221	C24B	BORDER	82	C8CB	C979	ON BREAK	B3
F1BA	F261	CALL	83	C8FB	CCCD	ON ERROR GOTO	B4
D246	D299	CAT	84	C940	C9FB	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	C2B3	CLS	8A	C212	C227	PEN	BB
CBC0	CC96	CONT	8B	C4D0	C546	PLOT	BC
EBEF	E9AB	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	EBF3	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	EBF3	E9AC	REM	C5
C4CB	C541	DRAWR	95	E7DF	E8A3	RENUM	C6
C052	C046	EDIT	96	DCD9	DCCD	RESTORE	C7
EBF3	E9B2	ELSE	97	CC03	CCDB	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	F7B4	SYMBOL	CF
C6ED	C78F	GOSUB	9F	C319	C346	TAG	D0
C6EB	C789	GOTO	A0	C320	C34D	TAG OFF	D1
C6C7	C76A	IF	A1	DDE4	DECA	TRON	D2
C22A	C254	INK	A2	DDE2	DEC6	TROFF	D3
DB2B	DB48	INPUT	A3	F17D	F22E	WAIT	D4
D439	D489	KEY	A4	C776	C81D	WEND	D5
D654	D691	LET	A5	C747	C7EA	WHILE	D6
DAFB	DB18	LINE	A6	C3E3	C42D	WIDTH	D7
EOF7	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	GRAFIKS	DE
C505	C532	MOVE	AE	n.v.	C5C3	MASK	DF
C50A	C537	MOVER	AF	n.v.	BD19	FRAME	EO
C5FB	C6A5	NEXT	BO	n.v.	C363	CURSOR	E1

# Die Adressen der Routinen von BASIC-Befehlen

Tabelle 2: Die neuen 664-Tokens.

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

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

49	DERR
7E	COPYCHR\$

alle Zahlen in hexadezimal

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

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-ADB9	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-AE82 AE64-AE65	Startadresse des Basicprogramms
AE83-AE84 AE66-AE67	Endadresse des Basicprogramms
AE85-AE86 AE68-AE69	Beginn Variablenbereich

solche Funktionen wie Kassettenschreibgeschwindigkeit oder Grafikhintergrund zu nehmen. In Ta-

belle Nr. 4 sind die wichtigsten Adressen für beide Computer aufgeführt. TMB

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

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

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 der man 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 Computer-

# Das 512 KB-Konzept



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	

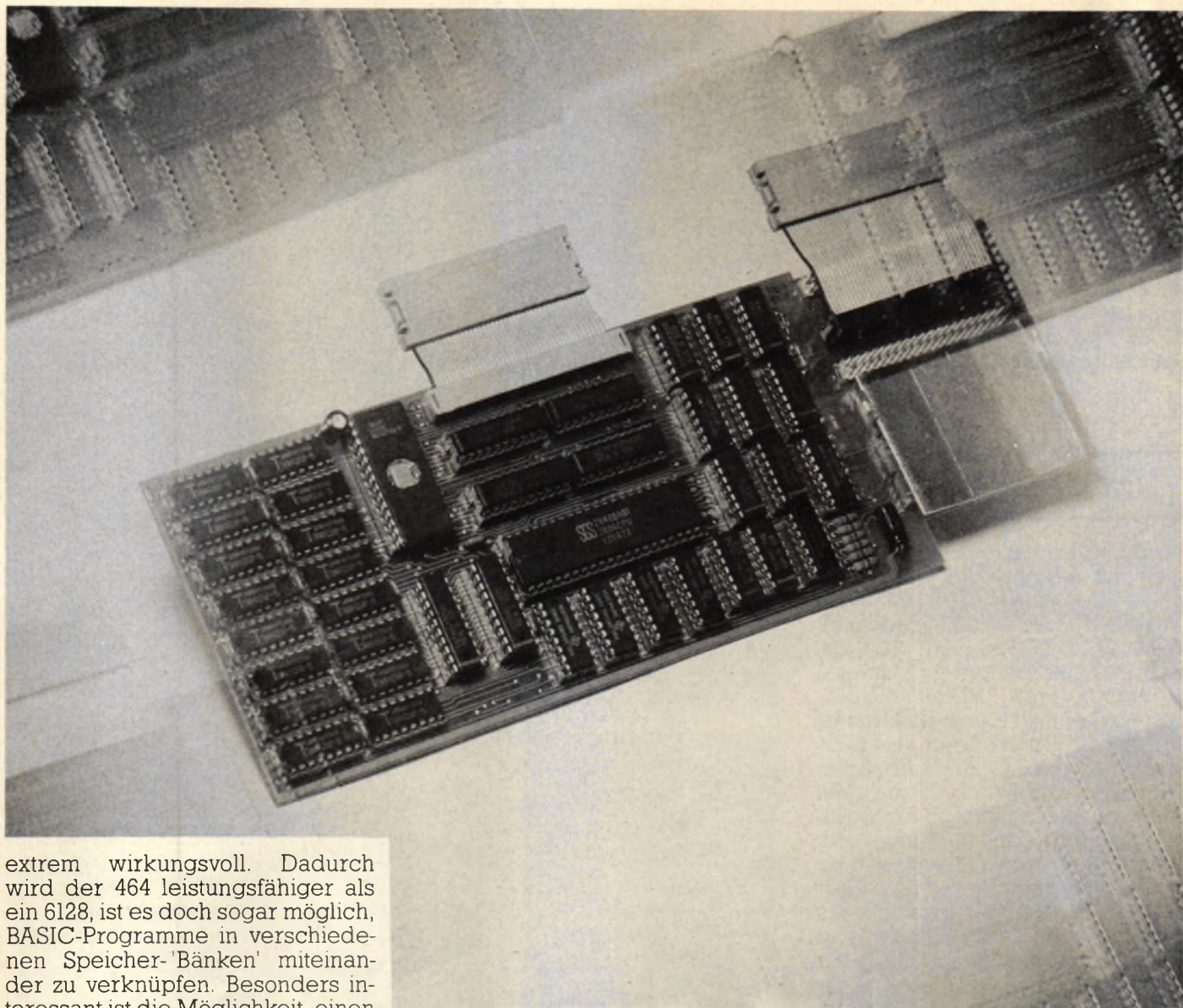
platine genommen, in die Speichererweiterung eingesetzt, zwei Kontakte angeklemt, und das war es dann auch schon. Bei sorgfältiger Arbeit und nochmaligem Check haben wir nur eine halbe Stunde gebraucht, bis der Computer wieder betriebsbereit war.

## 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 Verfü-

gung steht und wie diese sich aufteilt (Daten/Programm/Druckerspöler).

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 Speicherraum zur Verfügung, sondern kann jeweils auswählen, mit welchem Teil man arbeiten möchte. Dazu stehen eine Reihe von BASIC-Befehlen (siehe Tabelle 1) zur Verfügung (die notwendige 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' miteinander zu verknüpfen. Besonders interessant ist die Möglichkeit, einen Druckerspooier 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-)Sekundenschnelle 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.

# ZEICHENGENERATOR für den SCHNEIDER CPC 464

Das Programm bietet einen Befehlssatz, mit dem man komfortabel und doch einfach alle 256 Zeichen des CPC neu gestalten kann. Im einzelnen sind folgende Möglichkeiten vorhanden:

1. In einer 16 x 8 Felder großen Matrix werden jeweils auf Tastendruck entweder die Zeichen von 0 – 127 oder 128 – 255 ausgegeben. Das aktuelle Zeichen erscheint in einer 8 x 8 Matrix und kann hier geändert werden, wobei diese Änderung sofort in Originalgröße kontrolliert werden kann. Neben dem letztgenannten Feld erscheinen die SYMBOL-Werte in dezimaler Form.

2. Das aktuelle Zeichen kann Pixel für Pixel geändert oder neugestaltet, gespiegelt, gekippt, gedreht und invertiert sowie ein beliebiges anderes Zeichen an dessen Stelle kopiert werden.

3. Nach Löschen aller Zeichen (auf Tastendruck) in der großen Matrix können – aus einzelnen Zeichen zusammengesetzt – ganze Schriftzüge und sonstige Gemälde in hochauflösender Grafik erstellt werden.

4. Auf Wunsch generiert das Programm DATA-Zeilen mit den SYMBOL-Werten in hexadezimaler Form. Daraus können a) die Daten auf Kassette (oder Diskette) gesichert und später wieder in das Programm eingelesen werden, b) können die DATA-Zeilen gespeichert werden. Mit MERGE lassen sie sich dann an eigene Programme hängen und dort einlesen.

## Programmablauf

10 – 100: Initialisierung. Farb- festlegung in Zeile 40. Ferner Dimensionierung der Felder und Definition der CURSOR Zeichen.

110 – 160: Gestaltung der Über-

```

1 *****
2 * Z E I C H E N G E N E R A T O R *
3 *      fuer SCHNEIDER CPC 464      *
4 *****
5 * ==>      written 1985 by      <=== *
6 * ==>      Dietmar Schulze      <=== *
10 *****
20 *** INITIALISIERUNG ***
40 BORDER 13:INK 0,0:INK 1,15:INK 2,26:INK 3,6
50 MODE 1:CLEAR:char=0
60 DEFINT a-z:SYMBOL AFTER 0:KEY DEF 62,0
65 OPENOUT"dummy":MEMORY HIMEM-1:CLOSEOUT
80 SYMBOL 127,255,255,255,255,255,255,255
90 SYMBOL 126,60,126,231,195,195,231,126,60
100 DIM z(8,8),c1(255,8),z$(8),c$(8)
110 '=== TITEL ===
120 PAPER 1:PEN 2:PRINT STRING$(120,207):PEN 3
130 PAPER 0:LOCATE 3,2:PRINT CHR$(24)" Z E I C H
E N - G E N E R A T O R "CHR$(24)
140 PLOT 8,393,3:DRAW 631,393:DRAW 631,359:DRAW 8,
359:DRAW 8,393
160 GOSUB 4030:'==> zu MAIN SCREEN
1000 *****
1010 *** TASTATURABFRAGE ***
1020 *****
1030 ke$=INKEY$:IF ke$="" THEN 1030
1035 x1=0:y1=0:x2=0:y2=0
1040 '=== CURSOR 1 ===
1050 IF INKEY(1)=0 THEN x1=1 ELSE IF INKEY(8)=0 TH
EN x1=-1
1060 IF INKEY(0)=0 THEN y1=-1 ELSE IF INKEY(2)=0 T
HEN y1=1
1065 '=== CURSOR 2 ===
1070 IF INKEY(1)>31 THEN x2=1 ELSE IF INKEY(8)>31
THEN x2=-1
1080 IF INKEY(0)>31 THEN y2=-1 ELSE IF INKEY(2)>31
THEN y2=1
1085 '=== POSITION CURSOR 1 ===
1090 IF x1<>0 THEN xp1=xp1+x1:IF xp1<1 OR xp1>8 TH
EN xp1=xp1-x1
1100 IF y1<>0 THEN yp1=yp1+y1:IF yp1<1 OR yp1>8 TH
EN yp1=yp1-y1
1110 IF x1<>y1 THEN GOSUB 2820:GOTO 1030
1115 '=== POSITION CURSOR 2 ===
1120 IF x2<>0 THEN xp2=xp2+x2:IF xp2<1 OR xp2>16 T
HEN yp2=yp2-x2
1130 IF y2<>0 THEN yp2=yp2+y2:IF yp2<1 OR yp2>8 TH
EN xp2=xp2+y2
1140 IF xp2<1 OR xp2>16 THEN xp2=ABS(xp2-16)
1145 IF yp2<1 OR yp2>8 THEN yp2=ABS(yp2-8)
1150 IF x2<>y2 THEN GOSUB 2840:GOTO 1030

```

schrift, die während des Programmablaufs erhalten bleibt. Aufruf Routine Hauptscreen ab 4030.

**1000 — 1480:** Hauptschleife mit Tastaturabfrage für den gesamten Befehlssatz.

**1040 — 1150:** sind zuständig für die CURSOR-Pfeiltasten, und zwar ohne SHIFT für das einzelne Zeichen, mit SHIFT für den Zeichensatz.

**1320:** mit [u] Umschalten auf den jeweils anderen Zeichensatz.

**1330:** mit [ENTER] das fertige Zeichen sichern (SYMBOL).

**1340 — 1350:** [L] löscht ohne SHIFT das Einzelzeichen, mit SHIFT den Zeichensatz 128 bis 255.

**2000 — 2880:** sind für Unterroutinen zuständig:

**2270 — 2330:** berechnen binär die Werte für Spiegelung und rufen zur Ausgabe die benötigten Routinen auf.

**2340 -2410:** Dto. für das Drehen um 90 Grad.

**2420 — 2520:** kopieren mit „c“ ein gewähltes Zeichen in das kleine Fenster. Eingabe erfolgt hexadezimal = Zeilen- und Spaltennummer des großen Fensters. Abschluß der Eingabe mit ENTER.

**2530 — 2600:** generieren mit einem Trick eine DATA-Zeile im Bereich der Zeilen 10000 — 10255. Die zusammengesetzte DATA-Zeile auf Variablen wird auf die kleine ENTER-Taste gelegt, zusammen mit einem „GOTO 2600 + CHR\$(18)“. Da der CPC bei Programm-Abbruch keine Variablen löscht, ist dieser Trick möglich. Nach einem WINDOW-Swap nimmt das Programm seine Arbeit wieder auf.

**2610 — 2690:** schreiben die in DATA-Zeilen abgelegten Werte auf Kassette oder Diskette.

**2700 — 2760:** lesen die Daten von Kassette/Diskette und verzweigen zum Generieren der Zeichen mit dem SYMBOL-Befehl.

**2770 — 2820:** lesen die Daten aus generierten DATA-Zeilen.

**2830 — 2880:** Verzweigungen zu Unterroutinen je nach Bedarf.

**2840 — 2870:** legt Positionen für LOCATE, TEST, PLOT fest und verzweigt zu den benötigten Ausgaberroutinen.

**2880:** wandelt Dezimalwerte in hexadezimale Form um.

**3000 — 3240:** enthalten die PRINT- und PLOT-Routinen.

**3040 — 3060:** drucken Zeichen und Cursor im kleinen Fenster im Transparentmodus.

**3070 — 3120:** plottet Zeichenpixel an die entsprechenden Stellen.

```
1320 IF INKEY(42)=0 THEN char=ABS(char-128):c1=char:GOSUB 4120'==> ANDERER ZEICHENSATZ
```

```
1330 IF INKEY(18)=0 THEN GOSUB 2870:'==> AKTUELLES ZEICHEN SICHERN
```

```
1340 IF INKEY(36)=0 THEN SYMBOL c1,0,0,0,0,0,0,0,0,0:GOSUB 2850:'==> AKTUELLES ZEICHEN LOESCHEN
```

```
1350 IF INKEY(36)=32 AND char=128 THEN FOR a=128 TO 255:SYMBOL a,0,0,0,0,0,0,0,0,0:NEXT:GOSUB 4140:'==> ZEICHENSATZ 128-255 LOESCHEN
```

```
1360 IF INKEY(35)=0 THEN PAPER#2,0:LOCATE#2,yp2,yp2:PRINT#2,CHR$(24)CHR$(1)CHR$(c1)CHR$(24);:PAPER#2,2:GOSUB 2870:'==> ZEICHEN INVERTIEREN
```

```
1370 IF INKEY(37)=0 THEN SYMBOL c1,c1(c1,8),c1(c1,7),c1(c1,6),c1(c1,5),c1(c1,4),c1(c1,3),c1(c1,2),c1(c1,1):GOSUB 2850:'==> ZEICHEN KIPPEN
```

```
1380 IF INKEY(60)=0 THEN GOSUB 2280:'==> ZEICHEN S
```

PIEGELN

```
1390 IF INKEY(61)=0 THEN GOSUB 2350:'==> ZEICHEN DREHEN
```

```
1400 IF INKEY(62)=0 THEN a$=INKEY$:GOSUB 2430:'==> ZEICHEN COPIEREN
```

```
1410 IF INKEY(69)=0 THEN 2540:'==> DATA-ZEILE SCHREIBEN
```

```
1420 IF INKEY(9)=0 THEN z(xp1,yp1)=ABS(z(xp1,yp1)-1):GOSUB 3040:IF g1=0 THEN CLS#4:PRINT#4,,," >ENTER < druecken, wenn Zeichen fertig!":g1=1'==> PIXEL INVERTIEREN
```

```
1430 IF INKEY(69)=32 THEN 9010'==> DATA-ZEILEN AUF CASSETTE
```

```
1440 IF INKEY(54)=0 THEN GOSUB 5030'==> BEFEHLSLISTE
```

```
1450 IF INKEY(71)=0 THEN GOSUB 2620'==> AUSGABE DATA AUF CASSETTE
```

```
1460 IF INKEY(58)=0 THEN GOSUB 2710'==> ZEICHEN VON CASSETTE LESEN
```

```
1470 IF INKEY(58)=32 THEN GOSUB 2780'==> EINLESEN AUS DATA-ZEILEN
```

```
1480 fr=FRE(""):GOTO 1030
```

```
2000 ' *****
```

```
2010 ' *** UNTERPROGRAMME ***
```

```
2020 ' *****
```

```
2270 '=== ZEICHEN SPIEGELN ===
```

```
2280 FOR a=1 TO 8
```

```
2290 z$(a)=BIN$(c1(c1,a),8)
```

```
2300 z1$(a)="&x":FOR b=1 TO 8
```

```
2310 z1$(a)=z1$(a)+MID$(z$(a),9-b,1)
```

```
2320 NEXT b:c1(c1,a)=VAL(z1$(a)):NEXT a
```

```
2330 GOTO 2860
```

```
2340 '=== ZEICHEN UM 90 GRAD DREHEN ===
```

```
2350 FOR a=1 TO 8
```

```
2360 z$(a)=BIN$(c1(c1,a),8):NEXT
```

```
2370 FOR a=1 TO 8:z1$(a)="&x"
```

```
2380 FOR b=8 TO 1 STEP -1
```

```
2390 z1$(a)=z1$(a)+MID$(z$(b),a,1):NEXT b
```

```
2400 c1(c1,a)=VAL(z1$(a)):NEXT a
```

```
2410 GOSUB 2860:RETURN
```

```
2420 '=== ZEICHEN COPIEREN ===
```

```
2430 CLS#4:b$="&":PRINT#4," Welches Zeichen kopieren", " ( Hexadezimal = Zeile/Spalte ) ? ";CHR$(18);
```

```
2440 a$=UPPER$(INKEY$):IF a$="" THEN 2440
```

```
2450 IF INKEY(79)=0 AND LEN(b$)>1 THEN PRINT#4,CHR$(8) CHR$(16);:b$=LEFT$(b$,LEN(b$)-1):GOTO 2440
```

```
2460 IF INKEY(18)=0 THEN 2500
```

**3130 — 3190:** fragen Zeichen im großen Fenster mit TEST ab, setzen daraus in binärer Form die Werte für die Zeichen zusammen, drucken sie als Pixel und geben sie als Dezimalwert neben dem kleinen Fenster aus.

**3200 — 3240:** definieren das neue Zeichen mit SYMBOL und geben es als ASCII-Wert aus.

**4000 — 4180:** besorgen den Bildschirm Aufbau.

**4030 — 4090:** bauen Windows auf und färben sie ein.

**4100 — 4180:** für Texte, Numerierungen und Anfangswerte der CURSOR-Positionen.

**5000 — 5260:** enthalten die Befehlsliste zur Bedienung. Sie kann jederzeit mit „b“ abgerufen werden.

**9000 — 9070:** speichern die generierten DATA-Zeilen als ASCII-Datei „DATAZEIL“ auf Kassette oder Diskette ab. Von dort können sie mit „MERGE“ an eigene Programme angehängt werden.

**10000 — 10255:** enthalten schließlich die generierten DATA-Zeilen.

**11000:** dient zum Erkennen des Endes beim Einlesen der Daten.

Sämtliche im Programm eingefügten REM-Statements werden vom Programm nicht angesprungen und können weggelassen werden. Sie dienen nur der Verständlichkeit und Übersicht.

**z(n,m):** Pixelwerte des Zeichens (0 oder 1)

**z\$(n),zl\$(n):** für SYMBOL-Werte in binärer Form

**fw:** Dualwert für das PLOTTEN der Pixel

**gl:** Flag für Ausgabe in WINDOW 4

**xpl,ypl,xl,yl:** Positionswerte für Cursor in WINDOW 1

**xp2,yp2,x2,y2:** für WINDOW 2  
**xpla,ypla,xp2a,yp2a:** für alte Positionen

**xg,yg,xga,yga:** Positionen für den Grafik-CURSOR

**xy:** Relativwert für DRAW- und TESTR-Befehl

**a,b,c:** allgemeine Variablen für Schleifen usw.

**a\$,b\$,ke\$:** für INKEY\$-Abfrage und Aufbau der Eingaben

## WINDOWS

**0:** generelles Fenster ohne die Überschrift

**1:** kleines Fenster für das gewählte Zeichen

**2:** großes Fenster zur Ausgabe der gesamten Zeichensätze

**3:** für Ausgabe der dezimalen SYMBOL-Werte

**4:** für Kommentare und Anweisungen am unteren Bildschirmrand.

## Variablenliste

**char:** ist der Anfangswert der zwei Zeichensätze, entweder 0 oder 128. Das Wechseln erfolgt durch

„ABS(char-128),“

**cl,cla:** neue und alte ASCII-Werte des aktuellen Zeichens.

**cl(n,m):** SYMBOL-Werte für das Zeichen.

```

2470 IF (a$<"0" OR a$>"9") AND (a$<"A" OR a$>"F")
THEN 2430
2480 b$=b$+a$: IF LEN(b$)>3 THEN b$=LEFT$(b$,3):GOT
O 2440
2490 PRINT#4,a$;:GOTO 2440
2500 b$=LEFT$(b$+"00",3):c=VAL(b$): IF c<0 OR c>255
THEN 2430
2510 LOCATE#2,xp2,yp2:PAPER#2,0:PRINT#2,CHR$(1) CH
R$(c);:PAPER#2,2
2520 GOSUB 2870:CLS#4:RETURN
2530 '=== DATA-ZEILE FUER ZEICHEN SCHREIBEN ===
2540 WINDOW SWAP 0,4
2550 PRINT,,," KLEINE >ENTER<-Taste druecken !":P
EN 3
2560 GOSUB 2880
2570 KEY 139,CHR$(13)+RIGHT$(STR$(10000+c1),5)+" D
ATA "+c1$+",""+c$(1)+",""+c$(2)+",""+c$(3)+",""+c$(4)+
",""+c$(5)+",""+c$(6)+",""+c$(7)+",""+c$(8)+CHR$(13)+
goto 2600"+CHR$(13)
2580 IF INKEY(6)<>0 THEN 2580
2590 END
2600 CLS:PEN 2:LOCATE 1,2:PRINT," Okay !":WINDOW S
WAP 0,4:GOTO 1030
2610 '=== ABSPEICHERN AUF CASSETTE ===
2620 OPENOUT "!Zdaten"
2630 CLS#4:PRINT#4,,," REC und PLAY druecken, dann
Leertaste"
2640 IF INKEY(47)<>0 THEN 2640 ELSE CLS#4:PRINT#4,
,,," DATA werden gespeichert";
2650 RESTORE 9100
2660 FOR a=0 TO 8:READ c$(a):NEXT
2670 IF VAL(c$(0))>255 THEN CLOSEOUT:CLS#4:PRINT#4
,,," Zeichen sind gesichert":RETURN
2680 FOR a=0 TO 8:PRINT#9,c$(a):NEXT
2690 GOTO 2660
2700 '=== DATEN VON CASSETTE ===
2710 CLS#4:PRINT#4,,," PLAY druecken, dann die L
eertaste"
2720 IF INKEY(47)<>0 THEN 2720 ELSE CLS#4:PRINT#4,
,,," DATEN werden eingelesen"
2730 OPENIN"!Zdaten"
2740 IF EOF THEN CLOSEIN:CLS#4:PRINT#4," DATEN
sind eingelesen":GOSUB 4120:RETURN
2750 INPUT#9,c1$:c1=VAL(c1$):FOR a=1 TO 8:INPUT#9,
c1$:c1(c1,a)=VAL(c1$):NEXT
2760 GOSUB 3210:GOTO 2740
2770 '=== ZEICHEN AUS DATA-ZEILEN ===
2780 RESTORE 9100:CLS#4:PRINT#4,,," DATEN werd
en eingelesen":FOR a1=1 TO 3000:NEXT
2790 READ c1:IF c1>255 THEN CLS#4:PRINT#4,,,"
DATEN sind eingelesen":GOSUB 4120:RETURN
2800 FOR a=1 TO 8:READ c1(c1,a):NEXT

```

```

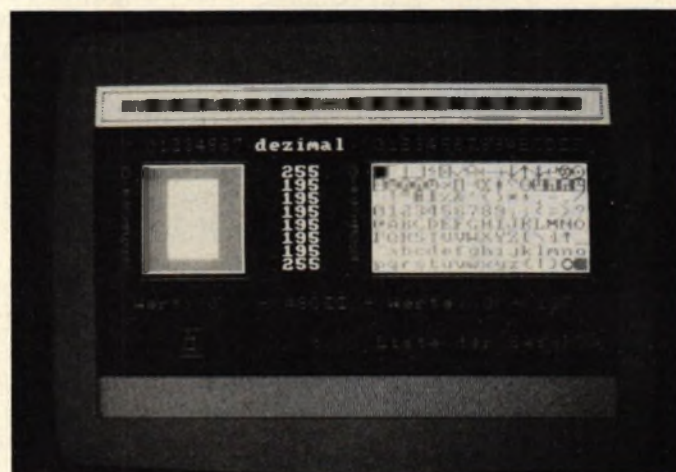
2810 GOSUB 3210:GOTO 2790
2820 GOSUB 3040:RETURN
2830 '=== VERZWEIGUNGEN ZU PRINTS & PLOTS ===
2840 c1=char+xp2-1+(yp2-1)*16:xg=(xp2-1)*16:yg=127
-(yp2-1)*16
2850 FOR a1=1 TO 3:ON a1 GOSUB 3110,3140,3210:NEXT
:RETURN
2860 FOR a1=1 TO 3:ON a1 GOSUB 3210,3110,3140:NEXT
:RETURN
2870 GOSUB 3140:GOSUB 3210:RETURN
2880 FOR a=1 TO 8:c$(a)="#"&HEX$(c1(c1,a)):NEXT:c1
$="#"&HEX$(c1):RETURN
3000 '*****
3010 '*** PRINTS & PLOTS ***
3020 '*****
3030 '=== CURSOR 1 DARSTELLEN ===
3040 fw=z(xp1,yp1):LOCATE#1,xp1a,yp1a:PRINT#1,CHR$(
(32+z(xp1a,yp1a)*95);
3050 LOCATE#1,xp1,yp1:PEN#1,3:PRINT#1,CHR$(22) CHR
$(1) CHR$(126) CHR$(22) CHR$(0);:PEN#1,1
3060 xp1a=xp1:yp1a=yp1
3070 '=== PIXELS PLOTTEN ===
3080 PLOT xg+(xp1-1)*2,yg-(yp1-1)*2,fw:ORIGIN 112,
81:PLOT (xp1-1)*2,15-(yp1-1)*2,fw
3090 ORIGIN 353,177:RETURN
3100 '=== ZEICHEN-2 DRUCKEN ===
3110 LOCATE#2,xp2a,yp2a:PRINT#2,CHR$(1):CHR$(c1a);
:PAPER#2,0:LOCATE#2,xp2,yp2:PRINT#2,CHR$(1):CHR$(c
1);:ORIGIN 353,177:PAPER#2,2
3120 xp2a=xp2:yp2a=yp2:c1a=c1:RETURN
3130 '=== PIXELS ABFRAGEN ===
3140 CLS#3:FOR a=0 TO 7:z$(a+1)="#"&x":FOR b=0 TO 7
3150 z(b+1,a+1)=TEST(xg+b*2,yg-a*2)
3160 z$(a+1)=z$(a+1)+RIGHT$(STR$(z(b+1,a+1)),1)
3170 LOCATE#1,b+1,a+1:PRINT#1,CHR$(32+z(b+1,a+1)*9
5);
3180 NEXT b:c1(c1,a+1)=VAL(z$(a+1)):FRINT#3,c1(c1,
a+1)
3190 NEXT a:MOVE xg,yg:xga=xg:yga=yg:GOSUB 3040:PR
INT CHR$(7);:RETURN
3200 '== ZEICHEN GENERIEREN UND DRUCKEN ==
3210 SYMBOL c1,c1(c1,1),c1(c1,2),c1(c1,3),c1(c1,4)
,c1(c1,5),c1(c1,6),c1(c1,7),c1(c1,8)
3220 LOCATE 8,17:PRINT CHR$(1) CHR$(c1);
3230 PEN 3:LOCATE 9,14:PRINT c1;SPACE$(4-LEN(STR$(
c1))):PEN 1
3240 CLS#4:g1=0:RETURN
4000 '*****
4010 '*** BILDSCHIRM - AUFBAU ***
4020 '*****
4030 ORIGIN 0,0:WINDOW 1,40,4,22:WINDOW#1,5,12,7,1
4:WINDOW#2,23,38,7,14:WINDOW#3,15,20,7,14:WINDOW#4
,1,40,23,25
4040 PAPER 0:PEN 1:PAPER#1,2:PEN#1,1:PAPER#2,2:PEN
#2,1:PAPER#3,0:PEN#3,2:PAPER#4,3:PEN#4,2:CLS:CLS#1
:CLS#2:CLS#3:CLS#4
4050 xy=16*8+3:PLOT 62,305,2:DRAWR xy,0:DRAWR 0,-x
y:DRAWR -xy,0:DRAWR 0,xy
4060 xy=xy+12:PLOT -6,4,3:DRAWR xy,0:DRAWR 0,-xy:
DRAWR -xy,0:DRAWR 0,xy
4070 xy=xy-12:PLOT 350,305,2:DRAWR xy*2-3,0:DRAWR
0,-xy:DRAWR -xy*2+3,0:DRAWR 0,xy
4080 xy=xy+10:PLOT -6,4,3:DRAWR xy*2-12,0:DRAWR 0
,-xy:DRAWR -xy*2+12,0:DRAWR 0,xy
4090 PLOT 104,104,3:DRAWR 32,0:DRAWR 0,-32:DRAWR -
32,0:DRAWR 0,32
4100 LOCATE 5,2:PRINT"01234567 ";:PEN 2:PRINT"dezi
mal ";:PEN 1:PRINT"0123456789ABCDEF"
4110 FOR a=4 TO 11:LOCATE 3,a:PRINT HEX$(a-4):NEXT
4120 CLS#2:LOCATE 4,14:PRINT"Wert:":LOCATE 14,14:P
RINT"- ASCII - Werte:"::PEN 3:PRINT char;:PEN 1:PR
INT"-"::PEN 3:PRINT char+127;CHR$(18)::PEN 1
4130 FOR a=4 TO 11:LOCATE 21,a:PRINT HEX$(a-4+SGN(
char)*8):NEXT
4140 CLS#2:c1=char:xg=0:yg=127:c1a=c1
4150 LOCATE 16,17:PEN 3:PRINT"< b > ";:PEN 1:PRIN
T"Liste der Befehle";
4160 FOR a=char TO char+127:PRINT#2,CHR$(1) CHR$(a
);:NEXT
4170 xp1=1:yp1=1:xp1a=1:yp1a=1:xp2=1:yp2=1:xp2a=1:
yp2a=1
4180 GOSUB 2850:RETURN
5000 '*****
5010 '*** BEFEHLSLISTE ***
5020 '*****
5030 WINDOW 1,40,4,25:CLS:PRINT

```

```

5040 RESTORE 5110
5050 READ a$,b$:IF a$="0"THEN 5090
5060 PEN 3:PRINT a$;:PEN 1:PRINT b$
5070 IF VPOS(#0)=5 THEN PRINT
5080 GOTO 5050
5090 LOCATE 8,21:PRINT"Zurueck mit";:PEN 3:PRINT"
LEERTASTE !";:PEN 1
5100 IF INKEY$("<") THEN 5100 ELSE GOSUB 4030:RETUR
N
5110 DATA"PFLEILTASTEN >>"," Cursor kleines Fenster
"
5120 DATA "mit SHIFT >>"," Cursor grosses Fenste
r"
5130 DATA "COPY >>","Invertieren Pixel"
5135 DATA "< c >>","Copieren des Zeichens"
5140 DATA "< d >>","Drehen des Zeichens um 90 Gra
d"
5150 DATA "< s >>","Spiegeln des Zeichens"
5160 DATA "< k >>","Kippen des Zeichens um 180 Gr
ad"
5170 DATA "< i >>","Invertieren des Zeichens"
5180 DATA "< u >>","Umschalten des Zeichensatzes"
5190 DATA "< l >>","Loeschen des Zeichens"
5200 DATA "< L >>","Loeschen Zeichensatz 128-255"
5210 DATA "< a >>","Ablegen Zeichen in DATA-Zeile
"

```



```

5220 DATA "< A >>","Abspeichern Zeilen auf Cass/D
isc"
5230 DATA "< z >>","Zeichen aus DATA-Zeilen auf"
5240 DATA "","Cass/Disc sichern"
5250 DATA "< e >>","Einlesen Zeichen von Cass/Dis
c"
5260 DATA "< E >>","Einlesen Zeichen aus DATA-Zei
len"
5270 DATA 0,0
9000 '=== ABSPEICHERN DATA-ZEILEN ===
9010 RESTORE 9100:CLS#4:PRINT#4,,," REC und PLAY g
edrueckt ? ( J / N)";:WHILE INKEY(45)=INKEY(46):W
END:CLS#4:IF INKEY(46)>-1 THEN 1030
9020 PRINT#4,,," DATA-Zeilen werden gesichert";:S
PEED WRITE 1:OPENOUT"datzeil"
9025 FOR a=0 TO 8:READ c$(a):NEXT
9030 IF VAL(c$(0))>255 THEN CLOSEOUT:CLS#4:PRINT#4
,,," DATA-Zeilen sind jetzt auf CASS/DISC";:FOR pa
=1 TO 4000:NEXT:frei=FRE(""):CLS#4:GOTO 1030
9040 zeil=10000+VAL(c$(0)):zeil$=RIGHT$(STR$(zeil)
,5)+" DATA "
9050 FOR a=0 TO 8:zeil$=zeil$+c$(a)+",":NEXT:zeil$
=LEFT$(zeil$,LEN(zeil$)-1)
9065 PRINT#9,zeil$
9070 GOTO 9025
9080 '*****
9090 '*** DATA - ZEILEN ***
9100 '*****
9110 '
9120 '== Hier entstehen die Datazeilen
9130 '== von 10000 bis 10255
11000 DATA 256,0,0,0,0,0,0,0,0

```



**Farben-Tip**

# Schnelle Farbänderung

Die Sache, um die es geht, kann man lapidar als schnelle Farbänderung bezeichnen. Aber das wäre wirklich zu lapidar, denn damit kann man ziemlich beeindruckende Effekte erzielen (die jedenfalls solange noch beeindruckend sind, bis sie in jedem beliebigen Programm auftauchen). Sicher haben Sie schon des öfteren, sei es durch ein BASIC-Programm, einen Programmabsturz oder durch sonst etwas, gesehen, wie es aussieht, wenn Ihr Schneider wild (und zufällig) die einer Ink zugeordneten Farben ändert, also z.B. Border oder Ink 0 wild flackern. Tatsache ist, daß einen so etwas nur beim allerersten Mal beeindruckt, wenn überhaupt. Und dafür wieder ist das Schneider-Betriebssystem ROM verantwortlich. Jedes Programm, auch BASIC, greift nämlich auf die entsprechenden Betriebssystemroutinen zu, um die Farben zu ändern. Das ist nichts Neues. Interessant wird es erst, wenn man sich anschaut, WANN dann das Betriebssystem die Farbe ändert.

## Erst kommt der Frame-Fly!

Es trägt nämlich den neuen Farbwert erst mal in die Farbtabelle ein und gibt dann die gesamte Tabelle aus, aber erst nach dem nächsten Frame-Fly, also dem Rücklauf des Elektrodenstrahls, der das Bild auf den Schirm zeichnet. Um wieder ein neues Bild zu zeichnen, muß der Strahl wieder oben links anfangen und wandert also von unten rechts nach oben links, um das

```

10 ; Listing 1
20 ;
30      org #a000          ;Startadresse &a000
A000    ent $
40
50 ;
60 ; ENT ist Anweisung fuer Schneider Devpac Assembler,
70 ; und gibt die Startadresse an
80 ;
A000 010CF0    90      ld bc,#f000      ;Zaehlschleife, Farben &F000* aendern
A003 C5       100     loop1: push bc      ;retten
A004 067F    110     ld b,#7f        ;Port &7Fxx=Farben
A006 0E10    120     ld c,#10        ;Farbe 16=Border
A008 ED49    130     out (c),c      ;Palette Pointer ansprechen
A00A ED5F    140     ld a,r          ;Farbe aus R-Register:Zufall
A00C E61F    150     and #1f        ;mit &1F maskieren und
A00E F640    160     or #40         ;mit &40 verknuepfen, damit Farbe legal
A010 ED79    170     out (c),a      ;Farbe ausgeben
A012 C1      180     pop bc         ;BC wieder holen
A013 0B      190     dec bc          ;BC-BC-1
A014 78      200     ld a,b          ;BC=0?
A015 B1      210     or c           ;Nein->alles nochmal
A016 20EB    220     jr nz,loop1     ;BC=0->fertig
A018 C9      230     ret            ;Pr. Ende
A019        240     end

Pass 2 errors: 00
Table used: 25 from 175
Executes: 40960

```

nächste Bild zu zeichnen. Während dieser „Wanderzeit“ wird auf dem Bildschirm nichts angezeigt, der Bildschirm leuchtet nur noch nach. Das heißt praktisch, daß

a) man die Farben maximal 50x pro Sekunde ändern kann (50 Bilder werden pro Sekunde angezeigt)

b) daß immer nur eine Farbe pro Ink und pro Bild angezeigt werden kann. Und genau hier wollen wir ansetzen. Wenn wir nämlich die Farbe einer Ink während des Bildaufbaus und nicht in der Pause dazwischen ändern, und wenn wir das auch noch oft genug tun — dann haben wir die ersehnten beeindruckenden Effekte.

## Listing 1!

...tut genau das. Die Farbe wird F000 Hex mal geändert (das dauert ca. eine Sekunde), und der Bildschirmrand wird mit einer zufälligen Farbe beschrieben. Die dafür notwendige Zahl wird einfach dem

R-Register entnommen, das dauernd irgendeine Zahl zwischen 0-255 enthält (für die Auffrischung des Speichers). Damit wir die Romroutinen zum Farbändern nicht benutzen müssen, schreiben wir den Farbwert direkt in den Palettenpointer, das Hardwareteil, das für die Farbgebung auf dem Bildschirm zuständig ist. Dazu muß das B-Register mit der Zahl 7F hex geladen werden, dann liegt auf der Portadresse 7Fxx das Farbregister (Bei einem Out (C)...Befehl wird als MSB ja auch der Inhalt des B-Registers adressiert. Da das LSB ja gleichgültig, eben xx, ist, interessiert für die Portadressierung nur der Inhalt von B). Auf Port 7F muß zuerst geschrieben werden, welche Ink bestimmt werden soll, und dann muß deren Wert folgen. Um also z.B. die Ink 0, die Hintergrundfarbe, auf 3 zu setzen, wäre folgende Befehlsfolge notwendig: LD BC, 7F00 (B'7FC'0) OUT (C),C LD C;3 OUT (C),C. Damit dürfte die Funktionsweise von Listing 1 wohl klar sein. Das Flackern der Borderfarbe kommt daher, daß die Farbe auch mitten im Bildaufbau geän-

```

10 ; Listing 2
20 ; bindet Farbwechsel in
    Interruptkette ein
A000 30 ;
    40 ORG #A000
    50 ;
A000 2107A0 60 ld hl,block
A003 CDE3BC 70 call #bce3
A006 C9 80 ret
    90 ;
A007 0000 100 block: defw 0
A009 0000 110 defw 0
A00B 00 120 defb 0
A00C B1 130 defb #B1
A00D 12A0 140 defw routi
A00F 00 150 defb 0
A010 0000 160 defw 0
    170 ;
A012 01107F 180 routi: ld bc,#7f10
A015 ED49 190 out (c),c
A017 ED5F 200 ld a,r
A019 E61F 210 and #1f
A01B F640 220 or #40
A01D ED79 230 out (c),a
A01F C9 240 ret
A020 250 end

```

Pass 2 errors: 00

Table used: 37 from 170

dert wird. Der aus dem R-Register geholte Wert muß mit 1F verknüpft und mit 40 geORt werden, damit sicher ist, daß ein gültiger Farbwert entsteht. Sollten Sie von Listing 1 nicht allzu beeindruckt sein, spielen Sie mal etwas damit herum. Ein paar DJNZ-Verzögerungsschleifen und NOP-Befehle wirken teilweise wahre Wunder (so könnten z.B. mehrere regenbogenfarbige Streifen entstehen, die langsam von unten nach oben wandern. In einem Programmtitel macht sich so etwas sehr gut!) Außerdem kann man auch noch eine weitere logische Operation (ein and/or etc.) nach dem or +40 einfügen, um so bestimmte Farben zu bevorzugen oder auszuschließen. Auch das kann teilweise überraschende Effekte bringen.

## Einbinden in die Interruptkette!

Ideal wäre es natürlich, wenn die normalen Programme weiterlaufen könnten, während diese Effekte erzeugt werden. Die Lösung für dieses Problem ist einfach und lautet — wie Sie schon in der Zwischenüberschrift lesen konnten: das Einbinden der Routine in die Interruptkette. Man kann dem Betriebssystem des Schneiders mitteilen, welche Programme (bzw. Adressen) er bei jedem Interrupt aufrufen soll, und man hat dabei sogar die Wahl, wie oft pro Sekunde, nach wievielen „Anstößen“ und wie oft insgesamt das Unterprogramm ausgeführt werden soll. Alle diese Informationen übergibt

man in einem sogenannten Interruptblock einigen Bytes im Speicher, die alle diese Informationen und die Adresse des Unterprogramms enthalten. Dann lädt man die Blockadresse ins HL-Register und ruft die entsprechende Betriebssystemroutine auf (siehe Programmlisting 2), und schon läuft die Sache (da wir uns ohnehin noch mit Interrupts genau beschäftigen werden, habe ich mich hier etwas kurz gefaßt.) Bei einem Aufruf durch Interrupts

seren Effekt zu erlangen. Aber auch hier gilt wieder die Devise: Herumexperimentieren! Denken Sie auch daran, daß ohne Umstände auch zwei Farben geändert werden können, z.B. Border (+10) und Hintergrund (+0). Wenn man in beide dieselben Farbwerte hineinschreibt, ergibt sich eine einheitliche Fläche. Auch müßten Sie beachten, daß Programmlisting 3 soviel Zeit braucht, daß andere Programme deutlich langsamer ablaufen. (Übrigens: Wenn Sie bei

```

10 ; Listing 3
20 ; mehrmals die Farbe wechseln in einem Interrupt
A000 30 ;
    40 ORG #A000 ;Start &A000
    50 ent $ ;Startadresse &A000
    60 ;
A000 2107A0 70 ld hl,block ;Adresse Interruptblock
A003 CDE3BC 80 call #bce3 ;KL_ADD_FAST_TICKER:bindet Block ein
A006 C9 90 ret ;ROUTI wird ueber Interrupt aufgerufen
    100 ;
A007 0000 110 block: defw 0 ;Interruptblock
A009 0000 120 defw 0
A00B 00 130 defb 0
A00C B1 140 defb #B1
A00D 12A0 150 defw routi
A00F 00 160 defb 0
A010 0000 170 defw 0
    180 ;
A012 F3 190 routi: di ;evtl. laenger als 1/300 Sek. !
A013 ED5F 200 ld a,r ;mehrmals in einem Interrupt
A015 E603 210 and 3 ;die Farbe aendern, aber max.3
A017 47 220 ld b,a ;Wert fuer DJNZ
A01B C5 230 rout2: push bc ;Zaehlwert sichern
A019 01107F 240 ld bc,#7f10 ;b=P.pointer,c=Border
A01C ED49 250 out (c),c ;Border ansprechen
A01E ED5F 260 ld a,r ;Zufallszahl ins A Register, und
A020 E61F 270 and #1f ;so maskieren, dass
A022 F640 280 or #40 ;ein gueltiger Farbwert entsteht
A024 F604 290 or 4 ;Farbe weiter maskieren
A026 ED79 300 out (c),a ;B ist immer noch &7F, a=Farbe
A028 C1 310 pop bc
A029 10ED 320 djnz rout2 ;...die Farbe beiter aendern
A02B FB 330 ei ;Interrupt wieder zulassen
A02C C9 340 ret ;fertig
A02D 350 end

```

Pass 2 errors: 00

Table used: 49 from 210

Executes: 40960

sind die Farbbalken zu regelmäßig. Wenn wir uns das von Programmlisting 2 gelieferte Resultat anschauen, könnten wir eigentlich zufrieden sein. Nach dem einem CALL &A000 erscheinen die erwarteten Farbbalken auf dem Border, aber andere Programme laufen normal weiter. Aber gerade diese Farbbalken sind nicht gerade „das Gelbe vom Ei“. Sie stehen immer starr an einer Stelle und wirken nicht sehr interessant. Diesem Problem hilft Listing 3 ab. Es ist, wie Sie feststellen werden, nur eine Variation von Listing 2. Dabei werden die Farben bei einem Interrupt mehrmals geändert, und so wird die starre Struktur durchbrochen. Ebenfalls werden die Farben noch einmal markiert, um so einen bes-

## CALL & A 000

Listings 2&3 etwas ändern und dann erneut assemblieren, während die Interruptroutine noch aktiv ist, riskieren Sie einen Crash, denn der Speicherbereich, den Sie

## Totales Chaos

beim Assemblieren überschreiben, wird ja über Interrupts noch dauernd aufgerufen. Wenn Sie also Ihren Computer im totalen Chaos versinken lassen wollen — Sie wissen, was Sie zu tun haben!). TMB

# Der Wordprocessor I

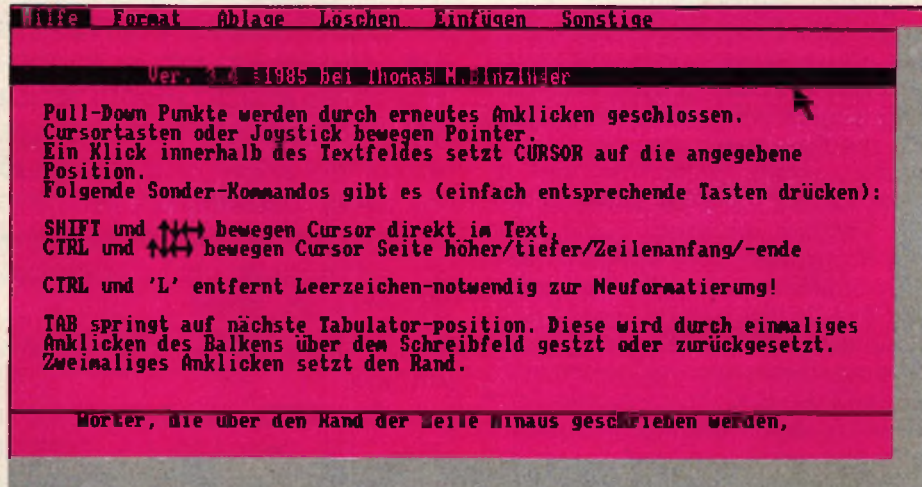
## Supertextverarbeitung mit Pull Down Menues

Schon wieder eine Textverarbeitung, werden Sie jetzt vielleicht sagen. Das ist zwar richtig, aber Wordprocessor 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 Cursortasten oder einem Joystick bewegt werden muß) und die damit verbundene Window-Technik und Bedienungstechniken.

### Zur Textverarbeitung

Wordprocessor 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 Wordprocessor? 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 Wordprocessor 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 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



FORMAT angewählten Punkt). Teilweise kann es aber auch nützlich sein, die eingefügten Leerzeichen wieder zu entfernen: Dazu einfach CTRL & L drücken. Mit CTRL & F wird eine Zeile wieder neu formatiert.

Wenn Sie die Funktion 'Texteinlesen' (unter Ablage) auswählen, muß das gewünschte Programm angeklickt werden, um es zu laden. Wird kein Programm, sondern irgendeine andere Stelle des Bildschirms angeklickt, wird die Funktion abgebrochen. Das kann man z.B. machen, wenn man nur das Directory sehen wollte. (Achten Sie aber darauf, daß Sie wirklich nur ein Textfile anklicken, nicht etwa ein Programm. Und Vorsicht: Sind so viele Files auf der Diskette, daß ein Teil des Directorys herausrollt, kann man die Files nicht mehr auswählen. Denn das Programm stellt anhand der Bildschirmposition des Pointers fest, welches Programm gemeint war). Bei der Farbwahl-Funktion unter Sonstiges muß man solange die Wörter 'Vordergrund' und 'Hintergrund' anklicken, bis man mit den Farben zufrieden ist. Dann 'OK' anklicken, um die Funktion zu beenden.

Bitte denken Sie daran, daß man so lange nicht schreiben kann, wie irgendein Extra-Window geöffnet ist, und daß man es erst durch erneutes Anklicken des entsprechenden Punktes schließen muß.

Ansonsten . . . probieren Sie herum! Es ist nicht allzu schwer, mit Wordprocessor zurechtzukommen.

## Wordprocessor erweitern . . .

Wie bereits gesagt, ist Wordprocessor nicht um der Textverarbeitung willen geschrieben worden, sondern als Demonstrationsmodell für GEM-ähnliche Techniken (GEM ist das Betriebssystem, das der Atari 520 ST verwendet, ähnlich wie Macintosh und Amiga). Wer also Teile aus Wordprocessor für seine eigenen Programme verwenden möchte, kann das gerne tun.

In diesem Zusammenhang sind vor allem die im ersten Programmteil enthaltenen Maschinencodieroutinen interessant: Sie stehen über sogenannte RSX-Befehle zur Verfügung und sorgen für den schnellen Bildschirmaufbau, die Darstellung des Pointers und die

```

5 '
6 ' Copyright 1985 Thomas M.Binzinger
10 DATA 01BB9021D490CDD1BCC9200038003E003F803FE03
FFB3FFE3FF03FE030700038001C000E00070000,3632
20 DATA C690C31891C35E91C38091C3AF915052494ED4494
ED64558C353C5000000000000000000000000000,3645
30 DATA DD6E00DD23DD6600DD2322DB90DD6E00DD660122D
E90DD6E02DD660322DC90DD6E04DD660522DA90,4731
40 DATA DD2AD890DD6E01DD660222DB90C90000CDE090CDO
6B92ADA901100C0192216912ADC90452ADE907D,4691
50 DATA 90472ADC90EB2AD89019ED5B1691C5E5D5E7CDA5B
B06087E1223E521000819EBE110F4D113E123C1,5094
60 DATA 10E4CD0CB9C9CDE0902ADA901100C019060BC5E5E
D5BDC90437E2F772310FAE111000819C110EBC9,4770
70 DATA CDE0902ADC907D329C912ADA901100C019119A900
60EC51A4F7E1279007723131A4F7E1279712BCD,3899
80 DATA 26BC13C110E8C9CDE0902ADC90ED5BDE90015000E
DB0C90000,3255
85 SYMBOL AFTER 90:OPENOUT"dummy":MEMORY &908F:CL
OSEDUT
90 RESTORE:ad=&9090
100 FOR l=10 TO 80 STEP 10:READ a$,cs:ls=0:FOR x=
1 TO LEN(a$) STEP 2:b$=MID$(a$,x,2):w=VAL("&"+b$)
:POKE ad,w:ad=ad+1:ls=ls+w:NEXT
110 IF ls<>cs THEN PRINT "Achtung!!! Falsche Date
n in Zeile:"l"!!!":STOP
120 NEXT:CALL &9090
130 RUN"tt2
140 '
150 ' Die Zeilennummern sollten nicht geaendert we
rden!
160 ' Dieser Programmteil laedt TT2 nach, muss da
her auf Kassette
170 ' vor TT2 stehen. Hat TT2 einen anderen Namen
, muss dieser in
180 ' Zeile 130 eingesetzt werden !

```

Verwaltung des Textspeichers (als Textspeicher wurde bei Wordprocessor nicht einfach ein Stringarray verwendet, da der Computer zu dessen Verwaltung zuviel Zeit braucht, sondern der Text steht über HI MEM). Die Befehle sind öPRINT, öINV, öEXC und öSE. Wie sie im einzelnen benutzt werden, ist aus dem Listing ersichtlich.

Das Programm ist für einen Epson MX-80 Drucker ausgelegt. Wenn man einen anderen Drucker

verwendet, muß man die Zeile 2270-2320 entsprechend ändern. Wenn man Wordprocessor mit Kassette benutzen will: Zeilen 1770-1820 löschen, Zeile 1830 sollte lauten: 1830 OPENIN"! Der Computer liest das nächste File von Kassette, das er findet.

Also: Wir wünschen viel Spaß mit Wordprocessor, und vielleicht finden Sie hier ja auch ein paar Anregungen für Ihre eigenen Programme. TMB

Beim eintippen beachten:  
Listing-Zeichen

ö  
ü  
ä  
ß  
§

Taste

] Klammeraffe mit SHIFT  
[  
↑ (unter £)  
] Klammeraffe ohne SHIFT

```

5 'Wordprocessor 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):istatus
$=" Hilfe Format Ablage Löschen Einfügen
Sonstiges":PLOT 0,358,0:DRAW 640,358 ELSE 280
260 FOR x=1 TO 80:IF tabs(x)=1 THEN PLOT 8*x-4,35
2,0:DRAW 8*x-4,360:PLOT 8*x-3,352:DRAW 8*x-3,360
270 NEXT
280 a$=SPACE$(77):öPRINT,82,0,76,8a$:öPRINT,80,0,
56,8status$
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)*80,0,79,8e$:LOCATE cx,cy
530 LOCATE cx,cy:CALL &BB8A:REM cursor on
540 GOSUB 810:in$=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$:GO
TO 520
580 IF i=6 THEN GOSUB 2540:qq=cx:cx=rrand:GOSUB 2
060:cx=qq:i=245:CALL &BB8D:GOTO 570
590 IF i=12 THEN GOSUB 2540:GOTO 520:REM Leerzei
chen entfernen
600 IF i=13 THEN öPRINT,(cy+2)*80,0,79,8e$: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)*80,0,79,8e$:c
x=cx+1:IF cx<rrand+1 THEN 530
730 IF in$="" 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+a1,e1-a1,1,1: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
=z$-screentop:screentop=MIN(lzeile,screentop+1
0):GOSUB 1170:zeile=screentop+d:GOSUB 2000:e$=z$:
GOSUB 1300
950 GOSUB 1010:REM Menuebalken
960 IF INKEY(9)=0 OR (JOY(0) AND 16)=16 THEN IF f
arben=1 THEN GOSUB 1680 ELSE IF loadp=1 THEN GOSU
B 1760 ELSE GOSUB 1090
970 IF u=ppos THEN 870
980 IF u<0 OR u>80*23+80 THEN 860
990 GOSUB 1300:ppos=u:GOTO 850
1000 DATA 2,7,10,16,19,25,28,35,38,46,49,57
1010 IF wopen=1 THEN RETURN ELSE IF ges=1 AND ppo
s-161>=a1-2 AND ppos-161<e1 THEN RETURN
1020 IF ges=1 THEN GOSUB 1300:ÖINV,79+a1,e1-a1,1,
1:GOSUB 1300:ges=0:a1=0:e1=0
1030 x=1:q=ppos-160:RESTORE 1000
1040 IF ppos<161 OR ppos>258 THEN RETURN
1050 READ a,e:IF (q>a-2 AND q<e) THEN nr=x:GOTO 1
080
1060 x=x+1:IF x=7 THEN RETURN
1070 GOTO 1050
1080 GOSUB 1300:ÖINV,79+a,e-a,1,1:a1=a:e1=e:ges=1
:GOSUB 1300:RETURN
1090 REM ** Open window **
1100 IF wopen=0 AND ppos>240 AND ppos<320 THEN 18
80:REM Schreibzone
1110 IF wopen=0 AND ppos>319 THEN z$=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$:LOCATE cx,cy:CALL ÖBBA:GOSUB 1300:RETURN
1120 GOSUB 2000:z$=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 p1=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 m1=minute:s1=stunde:minute=m:stunde=s:PLOT 1
58,191,1:GOSUB 1410:stunde=s1:minute=m1: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=p1: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$(20,127):FOR p=800 TO 1600 STEP 80:ÖPRINT,p
*1,0,19,$A$: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$:GOSUB 2030:ho=20:screentop=1:cx=1rand:cy=1:ze
ile=1:GOSUB 1170:LOCATE cx,cy:CALL &BB8A:wopen=0:
GOSUB 1300:GOSUB 2000:e$=z$:RETURN
1550 IF nr=6 AND w=6 THEN GOSUB 1300:GOSUB 240:z$
=e$:GOSUB 2030:ho=20:cx=1rand:zeile=lzeile:GOSUB
2000:screentop=lzeile-20:cy=20:IF screentop<1 THE
N screentop=1
1560 IF nr=6 AND w=6 THEN GOSUB 1170:CALL &BB8A:w
open=0:GOSUB 1300:GOSUB 2000:e$=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))+l1)*14:n$="":FOR
z=l1+ba TO l1+ba+10:n$=n$+CHR$(PEEK(z)):NEXT:n$=
LEFT$(n$,8)+". "+RIGHT$(n$,3):zeile=1:GOSUB 1300:0
PENIN n$
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:G
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 &BB8D: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$:NEXT:zeile=x
1980 CLOSEDOUT
1990 WINDOW SWAP 0,3:GOSUB 240:ho=20:GOSUB 1170:w
open=0:GOSUB 1300:RETURN
2000 REM z$ ← 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$(1
8)CHR$(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$:zeile=x:RETURN
2330 REM Hilfstext anzeigen
2340 WINDOW #0,3,78,3,22
2350 PRINT:PRINT:PRINT" Ver. 3.4 "CHR$(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 gesetzt
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=1zeile TO x+1 STEP -1
2470 zeile=y:GOSUB 2000:zeile=MIN(y+sz,maxi):GOSUB
B 2030:IF zeile>1zeile THEN 1zeile=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 1zeile: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

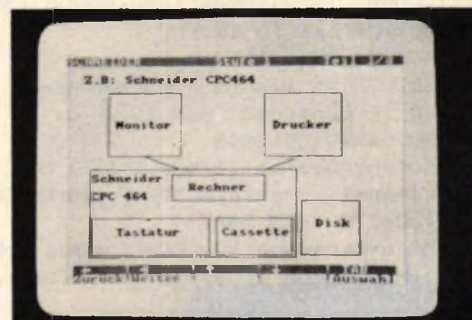
```

## 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 Komplett-Anlage gehört, also die erweiterte Computeritis. 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 herbeigezauberten 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



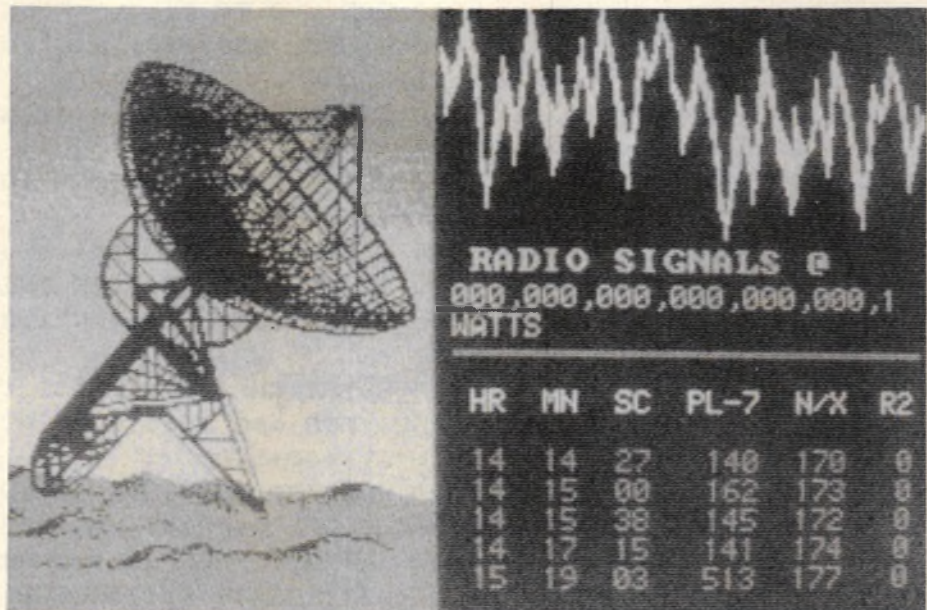
numerische und alphanumerische Variablen angesprochen und erklärt. So richtig schön rund um die BASIC-Sprache, bei der man nach kurzer Zeit schon in der Lage ist, seine eigenen kleinen Programme zu schreiben. Ergebnisse der Grafik-Programmierung sind z.B. ein Balkendiagramm, welches die Umsatzzahlen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



# Sprite Editor

## Das Top Listing



Viele Computer, wie zum Beispiel der C-64 können etwas, das die Schneidercomputer, obwohl sonst in fast allen Bereichen überlegen, nicht können: Sprites darstellen. Sprites sind mehrfarbige Figuren, die man beliebig über den Bildschirm bewegen kann, ohne den Hintergrund zu zerstören (wie es z.B. bei einem PRINT geschehen würde). Sie sind ein unentbehrlicher Bestandteil fast aller Computerspiele.

Mit dem Sprite Editor kann jetzt auch der Schneider Sprites darstellen. Sie können ganz einfach die Figur(en), die Sie brauchen, auf den Bildschirm zeichnen, den Rest erledigt der Sprite Editor. Nur noch Ihr Programm müssen Sie selbst schreiben. Die Bedienung ist denkbar einfach: Nach dem Start erscheint das Titelbild, und Sie werden nach der Größe Ihres Sprites gefragt. Die maximale Größe ist 16 Punkte breit und 25 Punkte hoch, aber Sie sollten daran denken, daß um so mehr Zeit benötigt wird, um das Sprite zu zeichnen, je größer es ist. Anschließend haben Sie noch die Gelegenheit, die Farben zu ändern. Geben Sie dazu erst

```

10 REM *****
20 REM ***** Sprite Editor *****
30 REM *** (c) Thomas M.Binzinger ***
40 REM *****
50 REM
60 GOTO 670:REM Zur Hauptschleife
70 REM ***Tastatur abfragen & Pos aendern
80 IF INKEY(0)=0 OR (JOY(0) AND 1)=1 THEN y=MAX(y-
1,1)
90 IF INKEY(2)=0 OR (JOY(0) AND 2)=2 THEN y=MIN(y+
1,25)
100 IF INKEY(8)=0 OR (JOY(0) AND 4)=4 THEN x=MAX(x
-1,1)
110 IF INKEY(1)=0 OR (JOY(0) AND 8)=8 THEN x=MIN(x
+1,xmax+1)
120 REM *** Blink Cursor ***
130 PEN RND*15:FOR t=1 TO 20:NEXT:LOCATE x,y:CALL
&BB8A
140 FOR t=1 TO 20:NEXT:CALL &BB8D
150 RETURN
160 REM *** Bildschirm aufbauen ***
170 PEN 1
180 MODE 0:LOCATE 1,1:FOR a=1 TO ymax:PRINT STRING
$(xmax,144):NEXT

```

## Sprites

die Nummer der entsprechenden Ink an und dann die dazugehörigen Farben (die Farben finden Sie auf G3/Seite 2 des CPC 464 Handbuchs). Sie werden dann so lange wieder gefragt, ob Sie die Farben ändern möchten, bis Sie irgendwann mit N für Nein antworten. Dann erscheint das leere Feld, in dem Sie das Sprite zeichnen können. Der blinkende Punkt kennzeichnet die Position, an der Sie malen. Mit dem Joystick oder den Pfeiltasten bewegen Sie ihn, mit Space, COPY oder dem Feuerknopf am Joystick setzen Sie einen Punkt. Um eine andere Farbe anzuwählen, brauchen Sie nur auf die Farbpalette an der Seite zu fahren und über der entsprechenden Farbe Space/COPY/Feuer zu drücken. Um einen Punkt zu löschen, müssen Sie die Hintergrundfarbe nehmen, die in der Farbpalette eine Position über der ersten Farbe liegt. Rechts unten können Sie das Sprite in Originalgröße sehen. Wenn Sie fertig sind, brauchen Sie nur auf ENDE zu fahren und dort auf den Knopf zu drücken.

## Der Sprite Editor erzeugt das komplette Programm

Der Sprite Editor erzeugt nun das komplette Programm, das notwendig ist, um die Spritedaten in einen String zu schreiben. Sie können dabei Zeilennummern und Schrittweiten bestimmen, um das Sprite-Programm möglichst optimal in Ihr eigenes Programm einfügen zu können. Der Spritename, den Sie angeben müssen, ist der Name des Strings, in dem das Sprite später steht. Wenn Sie kein weiteres Sprite mehr definieren wollen, können Sie noch entscheiden, ob Sie die Sprite-Routine anhängen wollen. Wenn Sie die erzeugten Sprites in einem Programm verwenden wollen, sollten Sie das auf jeden Fall tun, da das der Programmteil ist, der die neuen BASIC-Befehle für die Sprites erzeugt. Nun brauchen Sie nur noch einen Filenamen anzugeben, und das Programm wird auf

```

190 FOR f=0 TO 15:LOCATE xmax+1,f+2:PEN f:PRINT CHR$(233);:NEXT
200 PEN 1:a$="ENDE":FOR x=1 TO 5:LOCATE xmax+1,21+x:PRINT MID$(a$,x,1);:NEXT
210 RETURN
220 REM *** Drucke A$ doppelt hoch ***
230 FOR a=1 TO LEN(a$):b=ASC(MID$(a$,a,1)):adr=(b-32)*8+1+HIMEM
240 FOR s=0 TO 7:s(s)=PEEK(adr+s):NEXT s
250 SYMBOL 254,s(0),s(0),s(1),s(1),s(2),s(2),s(3),s(3)
260 SYMBOL 255,s(4),s(4),s(5),s(5),s(6),s(6),s(7),s(7)
270 PRINT CHR$(254)CHR$(10)CHR$(8)CHR$(255)CHR$(11);:NEXT:RETURN
280 REM *** Startbild anzeigen ***
290 IF PEEK(&B295)<>32 THEN SYMBOL AFTER 32
300 MODE 1:INK 3,6:INK 1,24:INK 2,15:INK 0,1:BORDE R 1
310 PRINT CHR$(22)CHR$(1);:PEN 1:a$="Sprite Editor":DEG:FOR w=0 TO 360 STEP 10:x2=COS(w)*150+320:y2=SIN(w)*100+300
320 PLOT 320,300,3:DRAW x2,y2:NEXT:LOCATE 15,6:PEN 1:GOSUB 220:PRINT CHR$(22)CHR$(0);
330 INK 2,15:PEN 2:a$=CHR$(164)+"1985 Thomas M.Binzinger":LOCATE 9,17:GOSUB 220
340 LOCATE 5,23:PRINT SPC(34);:LOCATE 9,23:PEN 1:PRINT"X-Breite des Sprites: ";:INPUT "",x:IF x<0 OR x>16 THEN 340 ELSE IF x MOD 2<>0 THEN x=x+1
350 LOCATE 5,24:PRINT SPC(34);:LOCATE 9,24:PEN 1:PRINT"Y-Breite des Sprites: ";:INPUT "",y:IF y<0 OR y>25 THEN 340
360 IF x/2*y>250 OR x<3 OR y<3 THEN PRINT CHR$(7);:GOTO 340
370 xmax=x:ymax=y:x=1:y=1:farbe=1
380 LOCATE 1,23:PRINT STRING$(80," ");:LOCATE 4,23:PRINT"Moechten Sie eine Farbe aendern?";
390 aa$=INKEY$:IF aa$="" THEN 390 ELSE aa$=LOWER$(aa$):IF aa$<>"j" THEN RETURN ELSE PRINT" Ja"
400 LOCATE 9,24:INPUT " Ink (0-15)? ",i:IF i<0 OR i>15 THEN 400
410 LOCATE 9,24:INPUT "nach Farbe (0-27)? ",f1:IF f1<0 OR f1>27 THEN 410
420 LOCATE 2,24:INPUT "zweite Farbe (nur ENTER wenn keine):",f2$
430 IF f2$=""THEN f2=f1 ELSE f2=VAL(f2$):IF f2<0 OR f2>27 THEN 420
440 INK i,f1,f2:WINDOW #5,1,40,23,25:CLS #5:GOTO 380
450 REM *** Sprite Edit ***
460 GOSUB 70:GOSUB 120:REM Cursor bewegen und blinken
470 IF (JOY(0) AND 16)<>16 AND INKEY(9)<>0 AND INKEY(47)<>0 THEN 460
480 IF x>xmax OR y>ymax THEN 510

```

Ein Beispiel: Wir wollen das Sprite BALL horizontal über den Bildschirm bewegen. Wir zeichnen also zuerst den Ball mit Sprite-Editor und hängen an die Balldaten auch noch die Sprite-Routine an. Mit folgendem Programm können wir dann den Ball unseren Wünschen entsprechend bewegen (diese Zeilen werden an das vom Sprite-Editor erzeugte Programm angehängt):

(In den Ausdrücken entspricht das ö dem I, also dem SHIFT-Klammeraffen.)

```

300 MODE 0:REM Sprites arbeiten nur in Mode 0
310 x=1:REM Sprite soll von links nach rechts wand
    ern
320 y=100:REM in Bildschirmmitte
330 h$=ball$:REM immer einen HILFSSTRING verwenden
    , da bei ,3 nicht zweimal EXChanged werden kann!
340 öEXC,x*1,y*1,sh$,3:REM Ball zeichnen
350 CALL &BD19:REM dadurch ist Bewegung besser
360 öPUT,x*1,y*1,sh$,1:REM Hintergrund wieder schr
    eiben, ,1 verwenden da Sprite ja wieder vollstaend
    ig ueberschrieben werden soll
370 x=x+1:IF x<70 THEN 330:REM 1 nach rechts
380 END:REM fertig

```

Kassette oder Diskette abgelegt. Und das können Sie jetzt mit Ihren eigenen Programmen MERGEN und so z.B. Top-Spiele schreiben. Achten Sie dabei darauf, daß das ganze Spriteprogramm einmal durchlaufen worden sein muß (z.B. indem Sie es an den Anfang Ihres eigenen Programmes stellen oder es als Unterprogramm aufrufen), bevor Sie die Sprites oder die folgenden neuen Befehle verwenden können:

PUT,x,y, spritename\$,a  
 GET,x,y, spritename\$,a  
 EXC,x,y,spritename\$,a  
 PUT schreibt einfach das Sprite auf den Bildschirm, ohne es zu verändern, GET holt den Inhalt des Bildschirms ins Sprite (damit wird der alte Inhalt überschrieben), und EXC tauscht den Inhalt von Sprite und Bildschirm aus. X und Y sind die Koordinaten, an denen das Sprite erscheinen soll, dabei muß X von 0-79 und Y von 0-199 gehen (diese Koordinaten gelten für MODE 0, in dem man mit den Sprites arbeiten sollte). Für X und Y müssen ZAHLEN eingesetzt werden! Wenn man trotzdem Variablen verwenden will, muß man mit ihnen irgendeine Rechnung durchführen, z.B. +0 oder \*1, also:

x+0 RICHTIG  
 12 RICHTIG, aber  
 x FALSCH!!!

Spritename ist der Name des Strings, in dem das Sprite steht (ist jeweils in einer REM-Zeile über den Datenzeilen angegeben), und a ist eine Zahl (auch hier gilt das für X und Y Gesagte), die angibt, wie das Sprite auf dem Bildschirm dargestellt werden soll, nämlich:

1= überschreibt Hintergrund  
 2= wird mit Hintergrund ge-OR-t  
 3= wird mit Hintergrund gemischt.

TMB

```

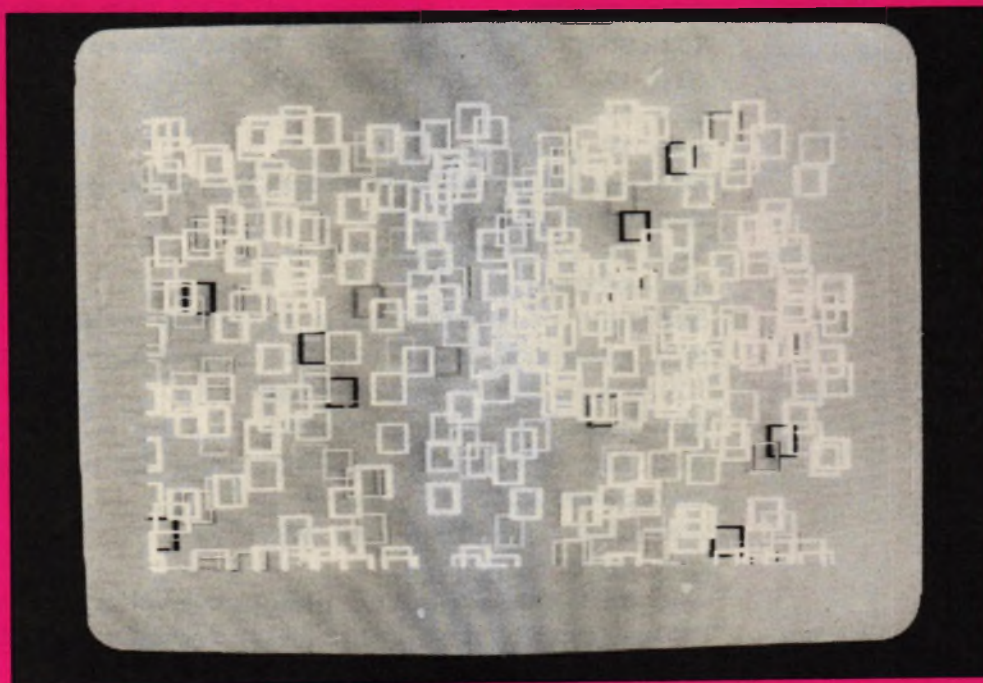
490 x1=556+(x*4):y1=128-(y*2):PLOT x1,y1,farbe:PEN
    farbe
500 LOCATE x,y:IF farbe=0 THEN PEN 1:PRINT CHR$(14
    4);:GOTO 460 ELSE PRINT CHR$(233);:GOTO 460
510 IF x<>xmax+1 THEN 550
520 IF y>17 THEN 560
530 IF y=1 THEN 460
540 farbe=y-2:PRINT CHR$(7);:GOTO 460
550 GOTO 460
560 IF y<20 THEN 460
570 RETURN
580 REM *** Bildschirm -> Array ***
590 ad=&C000+240+71+1120-1:sp=4:sprite(1)=1:sprite
    (2)=xmax/2:sprite(3)=ymax
600 xm=xmax/2:xm=xm-1:z=1
610 FOR y1=0 TO ymax:FOR x1=ad TO ad+xmax/2-1:spr
    ite(sp)=PEEK(x1):sp=sp+1
620 NEXT x1:GOSUB 640:NEXT y1
630 RETURN
640 REM *** errechne naechstunteres Byte ***
650 z=z+1:IF z=9 THEN z=1:ad=ad-7*2048+80:RETURN
660 ad=ad+2048:RETURN
670 REM *** Hautprogramm **
680 DIM sprite(300),pr$(400):prz1=1:zlnr=10:schrit
    t=10:ez=0
690 GOSUB 280:REM startscreen
700 GOSUB 160:REM Zeichenschablone
710 GOSUB 450:REM Zeichnen
720 GOSUB 580:REM in array schreiben
730 CALL &BB03:REM Tastaturpuffer loeschen
740 MODE 2:PEN 1:CALL &BC02:PRINT:INPUT "Bitte den
    Namen des Sprites eingeben:",sname$:IF LEN(sname$
    )>39 THEN 740
750 PRINT"Die erste Programmzeile waere"zlnr", die
    Schrittweite"schritt"."
760 PRINT"Moechten Sie diese Werte aendern?"CHR$(1
    43);
770 WHILE a$<>"j" AND a$<>"n":a$=INKEY$:a$=LOWER$(
    a$):WEND:PRINT CHR$(8)a$
780 IF a$="j" THEN INPUT"Startzeile,Schrittweite:"
    ,zlnr,schritt
790 IF z1=0 THEN z1=zlnr:z1$=sname$

```

```

800 dzeile=zlnr+schritt:wpz=0:z%=STR$(zlnr)+" REM
-----Daten fuer "+sname$
810 FOR x=1 TO sp
820 IF wpz=0 THEN PRINT z$:pr$(prz1)=z$:prz1=prz1+
1:zlnr=zlnr+schritt:z%=STR$(zlnr)+" DATA "
830 IF wpz<>0 THEN z%=z$+", "
840 z%=z$+"&"+HEX$(sprite(x),2)
850 wpz=wpz+1:IF wpz=17 THEN wpz=0
860 NEXT x
870 pr$(prz1)=STR$(zlnr)+" Data 9999"
880 PRINT pr$(prz1):prz1=prz1+1:zlnr=zlnr+schritt
890 pr$(prz1)=STR$(zlnr)+" "+sname$+"$="+STRING$(
"+STR$(sp)+",0)"
900 PRINT pr$(prz1):prz1=prz1+1:zlnr=zlnr+schritt
910 pr$(prz1)=STR$(zlnr)+" Restore"+STR$(dzeile)+
:a=5"+sname$+"$:a=peek(a+2)*256+peek(a+1):w=0:whil
e w<>9999:read w:if w<>9999 then poke a,w"
920 PRINT pr$(prz1):prz1=prz1+1:zlnr=zlnr+schritt
930 pr$(prz1)=STR$(zlnr)+" a=a+1:wend"
940 PRINT pr$(prz1):prz1=prz1+1:zlnr=zlnr+schritt
1050 FOR x=1 TO prz1-1: PRINT #9,pr$(x):NEXT:CLOSE
OUT
1060 PRINT"Programm komplett abgespeichert."
1070 END
1080 d$(1)="010BA52120A5CDD1BCC916A5C3C7A5C3CCA5C3
D1A54558C34745D45055D4000000000000000000000000000000
FE032813FE04C287A5DD6E00DD23DD6600DD237D322CA5DD6E
00DD66012228A5DD6E02DD"
1090 d$(2)="66032226A5DD6E04DD66052224A5DD2A28A5DD
6E01DD66022228A5DD2A28A5DD7E01322AA5DD7E02322BA5DD
23DD23DD23DD2228A5C9E1C9000000000000C5E5DDE5E106004F
7CCBBFFE4038077E4779CD"
1100 d$(3)="38A67778E1C1C93A24A56F2600328CA529EB3A
26A56F3ECB9D328DA52600CD1DBC228AA5C900F53E011B08F5
3E021803F53E0332C6A5F1CD2DA5CDA8A52A28A5DD2A8AA53A
2BA547C53A8CA54FDD2ABA"
1110 d$(4)="A53A2AA547C53AC6A5FE01CC24A63AC6A5FE02
CC2AA63AC6A5FE03CC33A623DD23C110E1E52A8AA5CD26BC22
8AA5E1C110C6C97ECD8EA577C9AFCD8EA577CD8EA5C97ECD8E
ASC9F53A2CA5FE012B0CFE"

```



```

950 PRINT:PRINT"-Daten komplett-":PRINT"Moechten Sie
ein weiteres Sprite definieren?"CHR$(143);a$="
"
960 WHILE a$<>"j" AND a$<>"n":a%=INKEY$:a%=LOWER$(
a%):WEND:PRINT CHR$(8)a$
970 IF a$="j" THEN 690
980 PRINT"Soll an die Daten die Sprite-Routine ang
ehaengt werden?"CHR$(143);a$=""
990 WHILE a$<>"j" AND a$<>"n":a%=INKEY$:a%=LOWER$(
a%):WEND:PRINT CHR$(8)a$
1000 IF a$="j" THEN GOSUB 1080
1010 PRINT"Das Programm hat als hoechste Zeilenumm
er"zlnr","
1020 PRINT"es besteht aus"prz1"Programmzeilen."
1030 PRINT"Bitte geben Sie den Namen an,":INPUT"un
ter dem es abgespeichert werden soll (bei Disk:ohn
e Extension!!):",name$
1040 name$=name$+".bas":PRINT"schreibe "name$"...
":OPENOUT name$
1120 d$(5)="03280BFE02280BF1AE77C9F177C9F1B677C9F1
C54FE6AAB728065F78E655B34779E655B728065F78E6AAB347
78C1C900"
1130 uzl=zlnr:FOR x=1 TO 5:pr$(prz1)=STR$(zlnr)+
Data "+d$(x)
1140 PRINT pr$(prz1):zlnr=zlnr+schritt:prz1=prz1+1
:NEXT
1150 pr$(prz1)=STR$(zlnr)+" restore"+STR$(uzl)+":a
d=&a501:for x=1 to 5:read d$(x):for y=1 to len(d$(
x)) step 2:w=val("+CHR$(34)+"&"+CHR$(34)+"+mid$(d$(
x),y,2)):poke ad,w:ad=ad+1:next y,x:call &a501:":
PRINT pr$(prz1)
1160 prz1=prz1+1:zlnr=zlnr+schritt
1170 pr$(1)=STR$(z1)+" memory &a500:rem Daten fuer
-----"+z1$:RETURN

```

# Des Schneiders Präsident

## Testbericht Präsident 6005

von Rudolf Petruck

Eine Computer-Anlage ohne Drucker ist wie ein Auto ohne Benzin. Möchte der Anwender ein Textverarbeitungsprogramm betreiben, ist meist ein Drucker mit einem sauberen Schriftbild erwünscht. Das Gedruckte soll eben aussehen, als sei es mit der Schreibmaschine geschrieben. Selbstverständlich muß so ein Drucker auch preiswert sein. Für die Schneider-Anlage ist z.B. der NLQ-401 (NLQ = Near Letter Quality / nahe Schreibmaschinen-Qualität) eine preiswerte Lösung. Für den C-64 wäre z.B. der SEIKOSHA SP 1000 VC ein Drucker mit einem guten NLQ-Schriftbild. Fabriziert man hiermit einen sogenannten Ausdruck, meint man auf den ersten Blick, so ein Schriftstück käme aus einer Schreibmaschine. Allerdings bei näherem Hinsehen erkennt man sofort den Matrixdrucker. Der Usus-Knaktus ist und bleibt also immer noch ein Typenraddrucker. Wenn diese Geräte nur nicht so teuer wären.

---

### Schreibmaschine als Drucker

---

Schaut man sich in den Computer-Shops oder in den Computer-Abteilungen der Warenhäuser einmal um, so findet man eine optimale Lösung. Die Firma Grubert bietet eine Typenrad-

schreibmaschine inklusive Interface (serielle bzw. Centronics-Schnittstelle) für unter DM 1000.-. Sehr vorteilhaft ist natürlich, daß man diese Typenradmaschine auch ohne Computer gebrauchen kann. Sicherlich taucht jetzt die Frage auf: Welche Software nimmt man denn hierzu? Nun, dazu kann ich nur sagen: Data-Becker's TEXTOMAT z.B. ist ein Programm, das sich sehr leicht an diese Schreibmaschine anpassen läßt. Sowohl bei Commodore, als auch bei Schneider. Wir testeten den CPC 6128 an der PRÄSIDENT 6005. Ich kann nur sagen: einfach phantastisch, wirklich toll! Es lief vollkommen problemlos. Natürlich Hochstellen, Tiefstellen, komprimierte Schrift usw. sind nicht möglich. Genauso gut wie Grafikzeichen, die sind auch nicht auf einem Typenrad. Also immer nur das, was auf einem Typenrad an Zeichen vorhanden ist, gelangt zum Ausdruck und nichts mehr oder weniger. Man hat natürlich bei Typenradmaschinen die Möglichkeit, diese Typenräder auszutauschen, und ist somit in der Lage, in verschiedenen Schriftarten und -zügen seine Texte zu erstellen. Nicht nur als Drucker, sondern auch extern als reine Schreibmaschine ist die PRÄSIDENT 6005 gut.

Abschließend noch etwas Technik!

**DER ANTRIEB:** Der moderne Antrieb mit Linearschrittmotor — das große Plus für einen leisen Druckwagenlauf und die exakte Positionierung des Typenrades.

**DER KURZTEXTSPEICHER:** Da-

mit können Sie Texte bis zu 165 Zeichen und Funktionen einspeichern und danach beliebig oft automatisch ausdrucken.

**KORREKTURSPEICHER:** Dieser hat eine Kapazität von 165 Zeichen. Innerhalb einer Schreibzeile können Sie jeden Tippfehler löschen. Nach der Korrektur drücken Sie die Relocate-Taste, und der Druckwagen fährt automatisch in die letzte Schreibposition zurück.

**FLIESSTEXT:** gestattet ein flüssiges Schreiben ohne Überschreiten des rechten Randes. Nach dem akustischen Warnsignal „Zeilenende“ löst die Betätigung des Trennungsstriches oder der Leertaste den Rücklauf des Druckwagens aus.

**ZENTRIEREN:** Überschriften, Hervorhebungen u.a. werden automatisch in die Mitte gerückt.

**DIE TABULATORSTASTE:** läßt die eingegebenen elektronischen Stopps schnell und zugleich weich ansteuern.

**DER DEZIMALTABULATOR:** ermöglicht es, Zahlenwerte stengerecht untereinander zu schreiben — unabhängig von der Anzahl einzugebender Stellen.

**DIE HALBSCHRITTSTASTE:** Haben Sie einen Buchstaben vergessen, kann dieser mittels der Halbschritt-Taste nachträglich eingefügt werden.

**DAUERFUNKTIONEN:** Alle Schreib Tasten haben Dauerfunktion, wenn sie länger als eine halbe Sekunde gedrückt werden, ebenso Rücktaste, Halbzeilenschaltung vor- und rückwärts sowie Leertaste.



#### WEITERE VORTEILE:

- Elektronischer Papiereinzug und -austrieb
- Sperrschrift
- Elektronischer Anschlagstärkeregler
- Halbzeilenschaltung vor- und rückwärts: ideal für manuelle Korrektur außerhalb der aktuellen Schreibzeile.

**DIE MEHRFACHBELEGTASTE** KB: wird zur Umschaltung der Tastatur benötigt, um die zusätzlichen elektronischen Funktionen auszulösen oder um Sonderzeichen auszudrucken, die als dritte Belegung auf den Tasten 1 bis 7

vorhanden sind. In Kombination mit der KB-Taste können Sie unter anderem folgende Funktionen ausführen:

- Einschreiben in den Kurztextspeicher
- Ausdrucken aus dem Kurztextspeicher
- Einstellen Dauerausdruck aus dem Kurztextspeicher
- Zentrieren einstellen
- Text zentriert ausdrucken
- Fließtext einstellen bzw. ausschalten
- Farbbandanhebung zum leichten Auswechseln der Farbbandkassette

— Korrekturbandanhebung zum leichten Auswechseln

**DAS TYPENRAD:** hat einen Vorrat von 100 Buchstaben bzw. Zeichen. Durch das Drop-in-System ist es leicht und problemlos auszuwechseln. Sie können verschiedene Schriftarten wählen. Schreibschritte: 1/10, 1/12 und 1/15 Zoll.

Den letzten „CHECK-UP“ bei all diesen Funktionen, brauche ich diese alle oder nicht, dürfen und müssen Sie natürlich durchführen. Ich meine: Für unter 1000,- DM bietet dieser Drucker bzw. diese Schreibmaschine allerhand.

# CP/M!

# FÜR

# SCHNEIDER

Aus den mitgelieferten Handbüchern zu den Schneider-Computern ist es leider nur mühsam oder gar nicht möglich, sich mit dem Betriebssystem CP/M einigermaßen vertraut zu machen. Dadurch bleiben einige interessante Möglichkeiten dieser Computer unentdeckt. Da es nun nicht jedermanns Sache ist, sich teure Bücher (bis über 80,- DM) zu diesem Thema zu-



zulegen, wollen wir in dieser Serie einige Feinheiten ausarbeiten, die über die Anwendung des Editors hinausgehen. So soll es zum Beispiel ermöglicht werden, das Betriebssystem zu verändern und eigene COM-Dateien zu erstellen. Voraussetzung sollte etwas Erfahrung auf dem Gebiet der Assemblerprogrammierung sein. Wir

werden hier speziell auf die 8080-Maschinensprache eingehen, da unter CP/M ein 8080-Assembler mitgeliefert wird.

## Der Aufbau von CP/M

CP/M besteht im wesentlichen aus fünf Teilen, nämlich:

- Systemparameterbereich

Dieser Bereich unterhalb der Adresse 100hex enthält wichtige Informationen wie Laufwerknummern und Sprungadressen.

- Anwenderprogrammabereich

Hier werden ab 100hex die .COM-Anwenderprogramme geladen und ausgeführt.

- Kommandoprozessor CCP

Dieser Speicherbereich enthält die Programme, welche nicht von Diskette nachgeladen werden müssen (DIR, ERA, REN, SAVE, TYPE UND USER).

- Basis- Diskettenbetriebssystem (BDOS)

- Basis-Ein/Ausgabe-System (BIOS)

An dieser Stelle fällt auf, daß im Handbuch zwei Befehle des CP/M

vergessen wurden, nämlich USER und SAVE. Der Befehl USER 3 zum Beispiel bewirkt eine Änderung

## Unbekannte Befehle

der Usernummer von 0 in 3. Dadurch können mehrere Benutzer dieselbe Diskette mit einem eigenen Inhaltsverzeichnis versehen. Wenn Sie den Befehl USER 3 eingegeben haben und anschließend den Befehl DIR, dann werden Sie bemerken, daß der Computer behauptet, die Diskette sei leer. Dies liegt daran, daß der Rechner versucht, das Inhaltsverzeichnis des Benutzers mit der Nummer drei auszugeben. Wenn Sie nun USER 0 eingeben, wird der Rechner wieder Ihre normale Directory (engl.: Adressbuch) ausgeben.

Mit dem Befehl SAVE ist es möglich, den Speicherbereich ab Adresse 100hex auf Diskette zu speichern. Dies ist der Bereich, in dem die CP/M-Programme abgelegt werden. Wenn Sie zum Beispiel 'SAVE 4 test.com' tippen, dann werden vier Blöcke mit je 256 Byte Länge abgespeichert. Versuchen Sie es einmal mit SAVE 0 'cont.com'. Auf der Diskette befindet sich danach eine Datei mit dem Namen CONTCOM. Diese Datei hat natürlich die Länge null. Führen Sie nun das Programm STATCOM aus, indem Sie die entsprechende Diskette einlegen und STAT tippen. Anschließend führen Sie Ihr neues Programm CONT aus. Da dieses die Länge null hat, der Rechner dies jedoch nicht merkt, versucht er, es an Adresse 100hex zu laden, und führt es aus. An dieser Stelle befindet sich allerdings noch das Programm STAT, welches durch diese Maßnahme erneut gestartet wird, ohne daß es neu geladen werden mußte.

## Lokalisieren des BIOS mit dem DDT

Wir wollen mit dem Systemparameterbereich beginnen und uns dort ein wenig näher umsehen. An Adresse 0 befindet sich ein Sprung ins BIOS. Da die Adresse des Bios von Rechner zu Rechner unterschiedlich ist, wollen wir diese herausfinden. Zu diesem Zweck booten wir CP/M und legen eine Diskette mit der Datei DDTCOM ein. Dies ist ein Disas-

sembler und Monitor. Wir starten diesen einfach durch Eingabe von DDT. Der Disassembler meldet sich daraufhin mit seinem Prompt; in diesem Fall ein kleiner Bindestrich. Um nun ab Adresse null zu disassemblieren, tippen wir L0. Wir erhalten eine Liste von Assemblerbefehlen, welche etwa so anfängt:

```
0000 JMP AD03
0003 ADD C
0004 NOP
0005 JMP 8F00
```

Hier sehen wir deutlich den Sprung (JMP) an Adresse AD03. Dies ist die Warmstartadresse unseres BIOS. Da wir die Kaltstartadresse des BIOS suchen, geben wir LAD00 ein. Als Ergebnis erhalten wir eine Reihe von Sprungbefehlen:

```
AD00 JMP C1B2 (Kaltstart-Adresse)
AD03 JMP C2BE (Warmstart-Adresse)
AD06 JMP C2E1 (Konsol-Status)
AD09 JMP C2C3 (Konsol-Eingabe)
AD0C JMP C2C8 (Konsol-Ausgabe)
AD0F JMP C2D2 (Drucker-Ausgabe)
AD12 JMP C2D7 (Stanzer-Ausgabe)
AD15 JMP C2DC (Leser-Eingabe)
```

An dieser Stelle erleben wir nun unsere erste Enttäuschung. Da die Sprünge alle auf Adressen oberhalb C000hex verzweigen, wo sich der Bildschirmspeicher des Rechners befindet, und das BIOS sich auch schlecht im BASIC-ROM befinden kann, müssen wir schließen, daß es sich im Disketten-Controller befindet. Dort können wir natürlich nichts ändern. Wir müssen uns also mit unseren Änderungen auf die Sprungtabelle beschränken.

Wir wollen an dieser Stelle versuchen, die Ausgaben des Computers an Bildschirm und Drucker gleichzeitig zu senden. Diese Möglichkeit bietet sich an, da der Sprung zur Druckerausgabe unmittelbar hinter dem Sprung zur Konsolenausgabe steht. Wenn wir al-

der Bildschirmausgaberroutine an die Adresse ADOF zurückgekehrt und zur Druckerausgabe verzweigt.

Unsere Änderungen können wir leicht mit dem DDT durchführen. Starten Sie ihn also wie vorhin und geben Sie nach dem Prompt 'AAD0C' ein. Sie erhalten folgendes Bild:

```
AD0C CALL C2C8 (alles, was nach ADOC steht, geben Sie ein).
AD0F (an dieser Stelle drücken Sie nur ENTER).
```

Sollten Sie jetzt keinen Drucker angeschlossen haben, wird Ihr Rechner streiken. Er wartet nämlich so lange, bis er ein Zeichen an den Drucker ausgeben kann. Betätigen Sie daher die Tasten CTRL und C gleichzeitig. Dadurch wird ein Warmstart von CP/M ausgelöst und die Änderung wieder rückgängig gemacht.

Diese Methode ist natürlich etwas zu umständlich, um sie jedesmal durchzuführen, wenn man etwas ausdrucken will.

Wir haben festgestellt, daß ab Adresse &AD00 eine Sprungtabelle steht, welche zu bestimmten Ein- und Ausgaberroutinen verzweigt. Wir haben gelernt, den DDT zu benutzen, um in dieser Tabelle Änderungen vorzunehmen. Außerdem haben wir mit dem Befehl 'SAVE 0 CONTCOM' ein Programm erzeugt, mit dessen Hilfe wir Programme, die wir bereits verlassen haben, die sich aber noch im Speicher befinden, erneut starten können. Dies erweist sich zum Beispiel als sinnvoll, wenn wir LOGO versehentlich verlassen haben, ohne unsere Arbeit zu sichern. In diesem Fall können wir durch Eingabe von CONT unser LOGO erneut starten, ohne daß unser Programm gelöscht wird.

Wir wollen nun den DDT nutzen, um selbst kleinere Programme zu schreiben. Dazu starten wir diesen und geben hinter dem Prompt 'al00' ein. Dies bedeutet, daß wir nun ab Adresse &l00 ein Assemblerprogramm schreiben können. Der DDT zeigt uns nun also die erste Adresse an, und wir können unser kleines Programm eingeben:

```
LHLD 0001
LXI D,0009
DAD D
MVI M,CD
RET
```

Der entsprechende Z80-Code würde so aussehen:

## Änderungen im BIOS

so den JMP-Befehl an die Adresse AD0C in einen CALL-Befehl ändern, wird nach der Ausführung



```
LD HL,(0001)
LD DE,0009
ADD HL,DE
LD (HL),CD
RET
```

Wenn Sie Ihr Programm fertig eingegeben haben, drücken Sie noch einmal Enter, um den Assemblermodus des DDT zu verlassen. Unser Programm lädt zunächst den Wert an Adresse 1 in das HL-Register. Dieser Wert stellt ja, wie wir bereits gesehen haben, die Warmstartadresse des CP/M dar. Um nun zur Adresse für die Konsoleneingabe zu gelangen, müssen wir zum HL-Register 9 hinzuaddieren. Dies erreichen wir über den Umweg mit dem DE-Register und dessen Addition zum HL-Register. Unser HL-Register enthält jetzt die Adresse, an der der JMP-Befehl für die Konsolenausgabe steht. An diese Adresse schreiben wir nun einfach einen CALL-Befehl (&CD). Dadurch kehrt das Programm nun nach jeder Konsolenausgabe zu unserer Sprungvektortabelle zurück und trifft auf den Vektor zur Drucker Ausgabe. Dadurch werden alle auf den Bildschirm ausgegebenen Zeichen auch an den Drucker geschickt. Verlassen Sie nun den DDT mit 'G0' oder 'CTRL C'. In beiden Fällen wird ein Warmstart von CP/M ausgelöst.

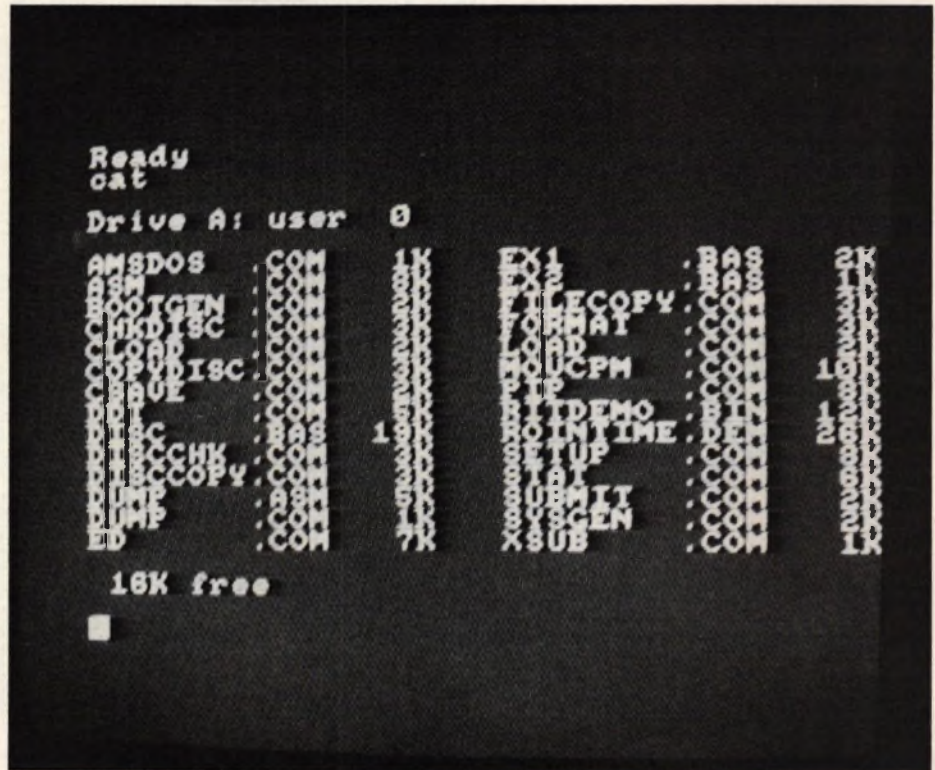
## Warmstart

Der G-Befehl des DDT startet das Programm an der angegebenen Adresse. In unserem Fall ist dies die Adresse 0, an der ja der Sprung zum Warmstart steht. Um unser erzeugtes Programm nun auf Diskette zu sichern, tippen wir 'SAVE 1 DRUCKAN.COM'. Mit diesem Programm können Sie nun zwar jederzeit den Drucker einschalten, Sie wollen ihn aber schließlich auch irgendwann wieder ausschalten. Dazu müssen Sie das gleiche Programm noch einmal schreiben; diesmal ersetzen Sie den Befehl 'MVI M,CD' jedoch durch 'MVI M,C3'. Dadurch ersetzen Sie den CALL-Befehl wieder durch einen JMP-Befehl. Dieses Programm nennen Sie dann einfach DRUCKAUS.COM.

## ASM.COM

Diese beiden Programme hätten wir auch mit dem mitgelieferten 8080-Assembler ASM.COM erzeugen können. Bei so kurzen Programmen geht es jedoch schneller, wenn man direkt die Möglichkeiten des DDT nutzt. Zur Entwicklung von längeren Programmen erstellt man in einem Editor zunächst den Programmtext. Für die

Um das Programm zu strukturieren, ohne ständig speicherfressende Leerzeichen benutzen zu müssen, können Sie TABs 'CHR\$(9)' verwenden. Auch manche Editoren unterstützen diese Möglichkeit (z.B. der EDCOM). Nach dem Abspeichern des Programms starten Sie nun den Assembler mit 'ASM filename'. Als Filename ist der Name der zu assemblierenden Datei einzugeben. Diese muß die Extension .ASM besitzen. Der Assembler erzeugt nun zwei neue Datei-



sen Zweck kann man den mitgelieferten Editor EDCOM benutzen. Wesentlich komfortabler sind jedoch einfache Textverarbeitungsprogramme, wie zum Beispiel Tasword, Wordstar oder der Turbo-Pascal Editor. Wenn Sie den Assembler ASM.COM benutzen, ist es notwendig, daß Sie die Memnics in Großbuchstaben schreiben. Unser kleines Programm würde dann etwa so aussehen:

```
ORG 100H
LHLD 1
LXI 9
DAD D
MVI M,CDH
ET
```

en mit den Erweiterungen .HEX und .PRN. Das .PRN-File enthält das assemblierte Quellprogramm mit dem erzeugten Code. Dieses kann man sich mit TYPE 'filename.PRN' ansehen. Das .HEX-File kann man sich ebenfalls ansehen. Es enthält den erzeugten Code im Intel-Hex Format. Mit 'LOAD filename' kann man das ausführbare .COM-File erzeugen.

Diese Methode hat den Vorteil, daß man beim Auftreten eines Fehlers jederzeit das Quellfile ändern kann. Bei einem Absturz eines unausgereiften Programms geht so auch das Programm nicht so leicht verloren. (tb)

# Parallele Ein-Ausgabeschnittstelle für den CPC 464



Von BEATE LANG

Ein universeller Einsatz des Computers, z.B. für Steuer- und Regelaufgaben, wird erst durch einen vollständigen Datenaustausch mit anderen Geräten ermöglicht. Die im Schneider CPC 464 eingebaute 7 Bit Parallelschnittstelle ist leider sehr unzulänglich und nur zur Datenausgabe geeignet. Im folgenden wird eine vollwertige Parallelschnittstelle mit der zugehörigen Software vorgestellt, die sowohl die Eingabe als auch die Ausgabe von 8 Bit Daten erlaubt. Die Anwendungspalette ist unbegrenzt. Sie ermöglicht beispielsweise den Anschluß eines Analog-Digitalwandler (ADC) zur Meßwerterfassung oder eines Digital-Analogwandler (DAC) zur Ausgabe, den Anschluß eines zweiten Computers, eines Plotters oder grafikfähigen Druckers. Preiswerte, auf dem Markt erhältliche Parallel-Seriell-Umsetzer (z.B. Elektor) ermöglichen auch den direkten Anschluß eines Akustikkopplers oder Modems. Im folgenden werden der Aufbau und die Funktion der Schnittstelle ausführlich beschrieben.

## A) Hardware-Aufbau der Schnittstelle

Das Kernstück der Schnittstelle ist der programmierbare PIO-Baustein 8255A (Bild 1), dessen 24 E/A-Anschlüsse in drei Betriebsarten verwendet werden können.

### 1. Die Betriebsarten des PIO 8255

#### 1.1 Betriebsart 0

Sie dient zur einfachen Ein- und Ausgabe. Die vorhandenen Anschlüsse werden in zwei Gruppen zu je 8 Leitungen (Port A und Port B) und eine Gruppe mit  $2 * 4$  Leitungen (Port C) unterteilt. Jede Gruppe kann unabhängig von der anderen als Eingang oder Ausgang pro-

grammiert werden. Ausgabedaten werden im Gegensatz zu Eingabedaten im Baustein zwischengespeichert.

#### 1.2 Betriebsart 1

Sie dient zur getasteten Ein- und Ausgabe mit Handshake-Signalen. Es stehen zwei Gruppen mit je ei-

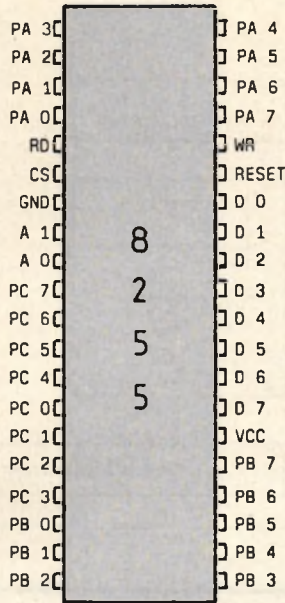


Bild 1  
Anschlußbelegung des PIO 8255A

$D_0 - D_7$	Data	Datentransfer zwischen Baustein und Prozessor
CS	Chip Select	Low - Baustein wird ausgewählt
RD	Lesen	Low - Daten oder Zustandsinformationen werden zum Prozessor gesendet
WR	Schreiben	Low - Daten oder Steuerbefehle werden in den Baustein eingelesen
Reset	Reset	High - alle internen Register einschließlich Steuerregister werden zurückgesetzt; alle Ports arbeiten als Eingabekanäle in Betriebsart 0
$A_0, A_1$	Adressleitungen zur Registerwahl	$A_1 A_0$ Register 0 0 Port A 0 1 Port B 1 0 Port C 1 1 Steuerregister
$PA_0 - PA_7$	Port A	Bit 0 - Bit 7
$PB_0 - PB_7$	Port B	Bit 0 - Bit 7
$PC_0 - PC_7$	Port C	Bit 0 - Bit 7
VCC	Versorgungsspannung	+ 5V
GND	Masse	0V

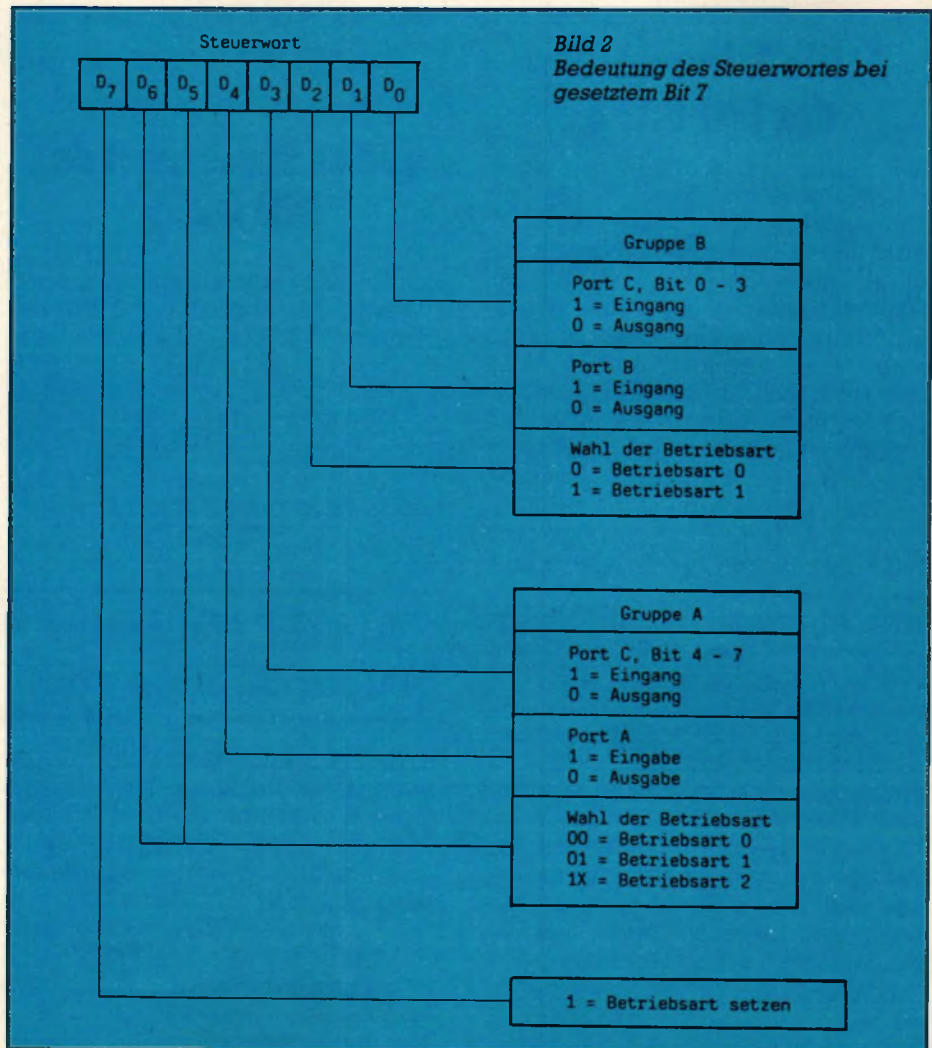
nem 8Bit breiten Datenport (Port A und Port B) und einem 4Bit breiten Steuerport (Port C) zur Verfügung. Jeder Datenport sowie die nicht verwendeten Leitungen des Steuerports können unabhängig voneinander als Ein- oder Ausgang programmiert werden. In dieser Betriebsart werden bei den Ports A und B alle Ein- und Ausgabedaten zwischengespeichert.

### 1.3 Betriebsart 2

Sie ermöglicht einen Datenaustausch auf einem einzigen 8 Bit breiten Port (Port A, Zweiwegbus). Den Austausch der Steuersignale übernimmt ein 5 Bit breiter Steuerport (Port C). Alle drei Betriebsarten können auch miteinander kombiniert werden.

## 2. Die Programmierung des PIO-8255

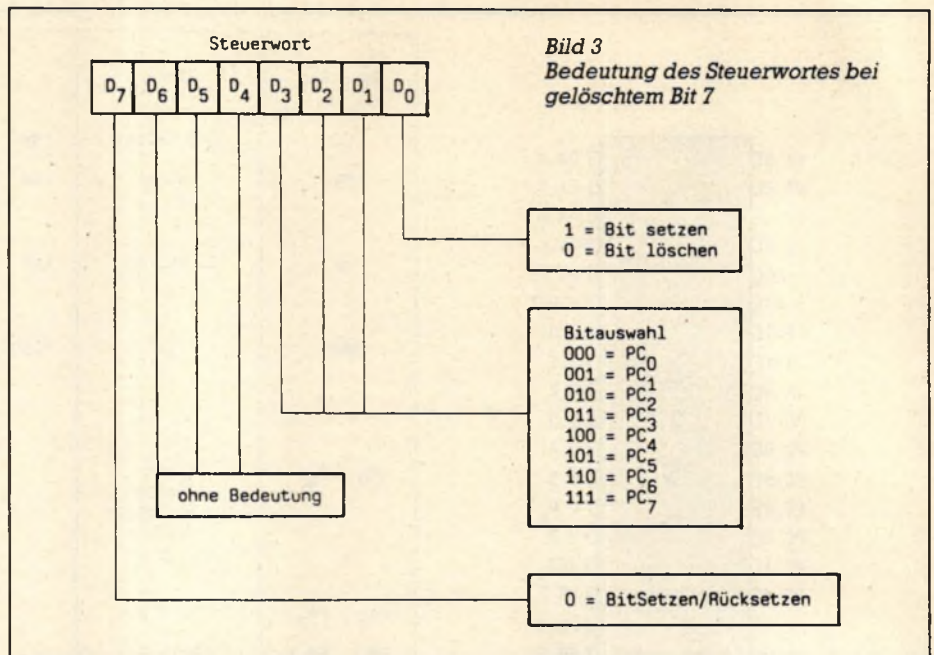
Die Programmierung des Bausteins ist verhältnismäßig leicht und mit einfachen In- und Out-Befehlen durchführbar. Intern besitzt der 8255 vier Register, die über die



Adressleitungen A0 und A1 ausgewählt werden können. Drei dieser Register dienen zur Datenübertragung, das vierte ist das Steuerregister. Das in dieses Register geschriebene Steuerwort bestimmt bei gesetztem 7. Bit die Arbeitsweise der Schnittstelle. Die Bedeutung der einzelnen Bits ist in Bild 2 zusammengestellt. Bei gelöschtem 7. Bit ermöglicht das Steuerwort außerdem das Setzen bzw. Löschen einzelner Bits des Registerinhalts von Port C (Bild 3). Da die meisten Peripheriegeräte mit Quittungssignalen arbeiten, bietet sich für die Schnittstelle die Betriebsart 1 an, welche daher auch für die beschriebene Hard- und Software gewählt wurde. Dabei wird Port A zur Datenausgabe und Port B zur Eingabe verwendet (Bild 4). Hierzu muß die Schnittstelle mit dem Steuerwort  $\&x1010A11$  initialisiert werden. Mit Bit 3 (A) können die zur Steuerung nicht benötigten Anschlüsse C4 und C5 des Ports C als Eingang (1) oder Ausgang (0) programmiert werden. Bit 0 ist in dieser Betriebsart ohne Bedeutung.

### 3. Die Adressierung des PIO 8255

Die serienmäßige Druckschnittstelle des CPC 464 wird über die Adresse  $\&EFxx$  angesprochen. Damit die volle Funktionsfähigkeit dieser Schnittstelle erhalten bleibt, bekam die neue Schnittstelle eine eigene Adresse. Wie aus dem CPC-Handbuch ersichtlich ist, stehen dem Anwender für eigene Erweiterungen die Adressen  $\&F8xx$  bis  $\&FBxx$  mit  $xx = \&E0 - \&FF$  zur Verfügung. Mit Rücksicht auf spätere Erweiterungen erfolgt die korrekte Adressierung der Schnittstelle über einen einfachen Adressendekodierer, der die Adressen  $\&F8Fx$  bis  $\&FbFx$  zuläßt (siehe Schaltplan Bild 5). Die Ein-Ausgabeschnittstelle wurde auf die Adresse  $\&F8Fx$  gelegt. Dabei dienen die oberen 12 Adressbits zur Bausteinauswahl und die untersten 2 Bits (A0 und A1) zur Registerauswahl (A2 und A3 werden zur Adressierung nicht herangezogen, was evtl. bei nachfolgenden Erweiterungen berücksichtigt werden muß). Damit erhält man die Registeradressen: Datenregister Port A  $\&F8F0$ , Datenregister Port B  $\&F8F1$ , Datenregister Port C  $\&F8F2$ , Daten-



register  $\&F8F3$ . Die Initialisierung der Schnittstelle erfolgt dann mit den Befehlen: LD A, $\&x10101111$ ; Steuerwort LD BC,  $\&F8F3$ ; Adresse des Steuerregisters OUT (C),A; Ausgabebefehl an das Steuerregister. Auf die gleiche Weise werden auch die Datenregister beschrieben bzw. gelesen.

### 4. Das Steuerport des PIO 8255

Der Registerinhalt des Steuerports C hat in der Betriebsart 1 die Bedeutung: OBF = Zustand des Ausgabe-Puffer Flipflops, INTE und INTR = Interrupt-Anforderung, E/A = Eingang/Ausgang, IBF = Zustand des Eingabe-Puffer Flipflops, von diesen Bits werden nur Bit 1 (IBF) und Bit 7 (OBF) ausgewertet. (Siehe Tabelle 3).

#### 4.1 OBF (Ausgabe-Puffer Flipflop voll)

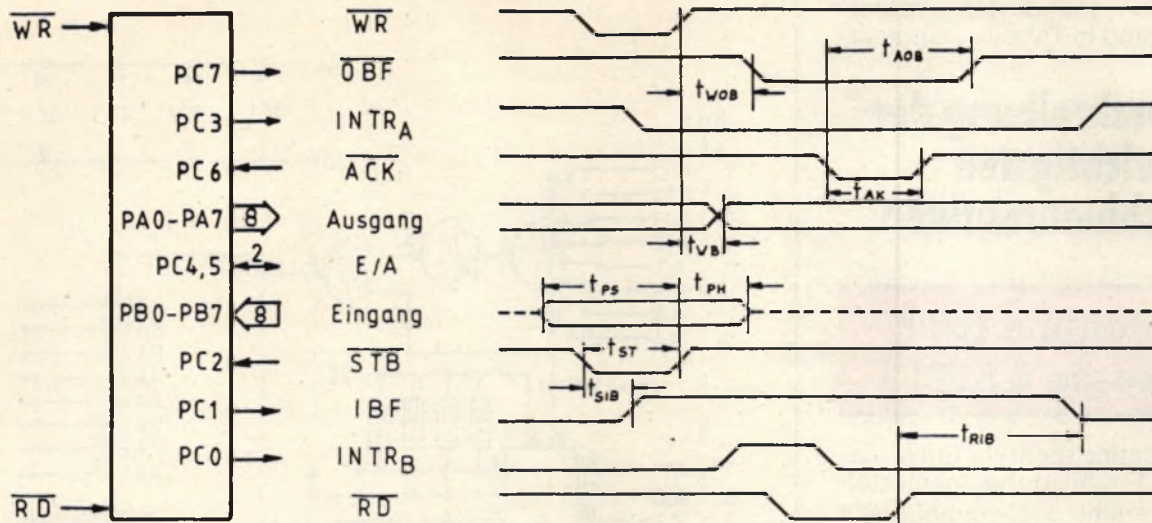
Der OBF-Ausgang wird Low, wenn Daten zur Ausgabe im Datenregister des Port A bereitstehen. Das OBF-Signal entspricht somit dem Strobe beim Ausgabevorgang. Es wird von der ansteigenden Flanke des WR-Signals gesetzt und von der fallenden Flanke des vom Peripheriegerät kommenden ACK-Signals zurückgesetzt (siehe Impulsdiagramm Bild 4).

### IBF (Eingabe-Puffer Flipflop voll)

Der IBF-Ausgang wird High, wenn Daten in das Datenregister des Port B eingelesen wurden. Dieses Signal entspricht somit dem ACK-Signal beim Einlesevorgang. Es wird durch die fallende Flanke des Strobe-Signals gesetzt und durch die ansteigende Flanke des RD-Signals zurückgesetzt.

### Hardware-Aufbau der Schnittstelle

Bild 5 zeigt den Schaltplan einschließlich der Bestückungsliste. Die Schaltung wurde auf einer Lochrasterplatine in Wire Wrap-Verdrahtung aufgebaut. Die Verbindung mit dem Rechner geschieht über den Erweiterungsbus an der Gehäuserückseite, an dem auch das Floppy-Laufwerk angeschlossen wird. Hierfür wurde ein Adapterstück hergestellt, bestehend aus einem 50-poligen Platinenstecker, der an ein ca. 2,5 cm langes Platinenstück mit Leiterbahnen im 2,54 mm Raster angelötet wird. An das Platinenstück wird möglichst dicht am Stecker das Verbindungskabel zur Schnittstellenplatine gelötet. Auf das freie Platinenende kann nun wieder der Floppy-Controller aufgesteckt werden. Die Pin-Belegung des Erwei-



$t_{AOB}$  = Zeit von ACK=0 bis OBF=1  
 $t_{AK}$  = ACK-Impulsbreite  
 $t_{PS}$  = periphere Daten vor STB  
 $t_{ST}$  = STB-Impulsbreite  
 $t_{RIB}$  = Zeit von RD=1 bis IBF=0  
 $t_{WOB}$  = Zeit von WR=1 bis OBF=0  
 $t_{WB}$  = Zeit von WR=1 bis Ausgabe  
 $t_{PH}$  = periphere Daten nach STB  
 $t_{SIB}$  = Zeit von STB=0 bis IBF=1

Bild 4  
Impulsdiagramm bei Ein- und Ausgabe in der Betriebsart 1

terungsbusses ist im CPC-Handbuch, Anhang V/Seite 2 oben angegeben. Der Anschluß der Eingabegeräte an das Port B des Schnittstellenbausteins erfolgt z.B. über einen 25-poligen Cannonstecker. Für den Anschluß des Druckers wurde ein Platinenstück mit 2X17 Leiterbahnen mit dem Port A des Bausteins verbunden und auf der Platine befestigt, so daß das vorhandene Druckerkabel weiterverwendet werden kann. Die Pin-Belegung des Platinenstücks ist aus dem CPC-Handbuch, Anhang V/Seite 2 unten ersichtlich.

nach dem Ausgabertext gibt als Eingabesignal ein Fragezeichen aus. Ein Komma unterdrückt das Fragezeichen.

b) Ein Strichpunkt nach Line Input +8 unterdrückt den Zeilenvorschub nach der Eingabe.

c) Print +8, (<Ausgabeliste>)(<Using Format >)(<Trennzeichen>). Ein Komma als Trennzeichen gibt den folgenden Ausdruck in der nächsten Druckzone aus. Ein Strichpunkt erzeugt ein Leerzeichen.

d) Write +8, (<Ausgabeliste). Die in der Ausgabeliste angegebenen

Schnittstelle. Nach der Initialisierung ist &A500 = 0, d.h., die Ausgabe erfolgt auf der neuen Schnittstelle.

## 2. Programmaufbau

Wird bei der Ausführung eines BASIC-Befehls eine Routine des Betriebssystems benötigt, so springt der BASIC-Interpreter diese im allgemeinen nicht direkt an, sondern über die sog. Indirections, die im RAM untergebracht sind. Da der RAM-Bereich beschrieben werden kann, besteht hier die Möglichkeit, den Befehlsablauf zu beeinflussen. Als günstig erweist sich hierbei, daß alle für die Parallelschnittstelle wichtigen Ausgabebefehle die Vektoradresse &BD2B benutzen, die normalerweise einen Einsprung in die Druckroutine des Betriebssystems bewirkt. Die Eingabe-Routinen benutzen für Kanalnummern <+9 die Vektoradresse &BD3a, welche einen Sprung in die Edit-Routine des Betriebssystems enthält. Werden diese zwei Vektoradressen auf die Routinen der neuen Schnittstelle umgeleitet, dann können alle Ein- und Ausgabe-Routinen des BASIC-Interpreters von den Routinen der neuen Schnittstelle übernommen werden. Auf diese Weise bleibt das Maschinenprogramm kurz und be-

# B) Software der Schnittstelle

## 1. Beschreibung der BASIC-Befehle

Nach dem Laden des Programms (über BASIC-Lader oder Maschinenprogramm) muß die Schnittstelle einmalig mit 'Call &A500' initialisiert werden. Danach kann die Schnittstelle mit folgenden BASIC-Befehlen angesprochen werden:

a) Ein Strichpunkt nach Input +8 unterdrückt den Zeilenvorschub nach der Eingabe. Ein Strichpunkt

Werte werden durch Kommata getrennt ausgegeben.

e) List (<Zeilenbereich>), +9. Es wird der angegebene Zeilenbereich ausgelistet. Die Input-Befehle (a und b) sprechen immer die neue Schnittstelle an. Die Ausgabebefehle können wahlweise auf der neuen oder alten Schnittstelle ausgegeben werden. Zur Unterscheidung dient die Speicherstelle &A500. Es gilt: Inhalt von &A500 = 0 Ausgabe auf der neuen Schnittstelle. Inhalt von &A500 > 0 Ausgabe auf der alten

nötigt wenig Speicherplatz im RAM. Die geänderten Vektoradressen sind in Tabelle 1 angegeben.

### 3. Beschreibung der wichtigsten Maschinenroutinen

#### 3.1 Routine INIT Adresse &A501

Diese Routine dient zur Initialisierung der Ein-Ausgabe-Schnittstelle. Sie braucht nur einmal nach dem Laden des Maschinenprogramms mit CALL &A501 aufgerufen zu werden.

#### 3.2 Routine COMPI Adresse &A50D Vektoradresse &BD3A

Dieser Programmteil entscheidet für alle BASIC-Eingabebefehle mit Kanalnummern < +9 (z.B. Line Input +6, a\$), zu welcher Eingaberoutine verzweigt werden soll. Ist die Kanalnummer, die bei Adresse &AC21 abgespeichert ist, kleiner als +8, dann springt das Programm in das Betriebssystem zur SCREEN EDIT-Routine (&2A98), die die Eingabe von der Tastatur verwaltet. Ist die Kanalnummer +8, dann wird zur Routine ICHAR verzweigt, die die Eingabe über die Parallelschnittstelle steuert.

#### 3.3 Routine ICHAR Adresse &A51A

Diese Routine liest Daten von der Eingabeschnittstelle in den Akku, speichert sie in einem 255 Zeichen großen Eingabepuffer (&ACA4-&ADA3) ab und gibt sie am aktuellen Bildschirmfenster aus. Steuerzeichen werden dabei nicht geführt, sondern als Symbol ausgegeben. Wenn das letzte Zeichen ein Linefeed war oder die ESC-Taste gedrückt wurde, kehrt das Programm zu der aufrufenden Routine zurück. Linefeed und Carriage Return werden dabei nicht in dem Puffer zwischengespeichert. Ist eine Zeichenfolge länger als 255

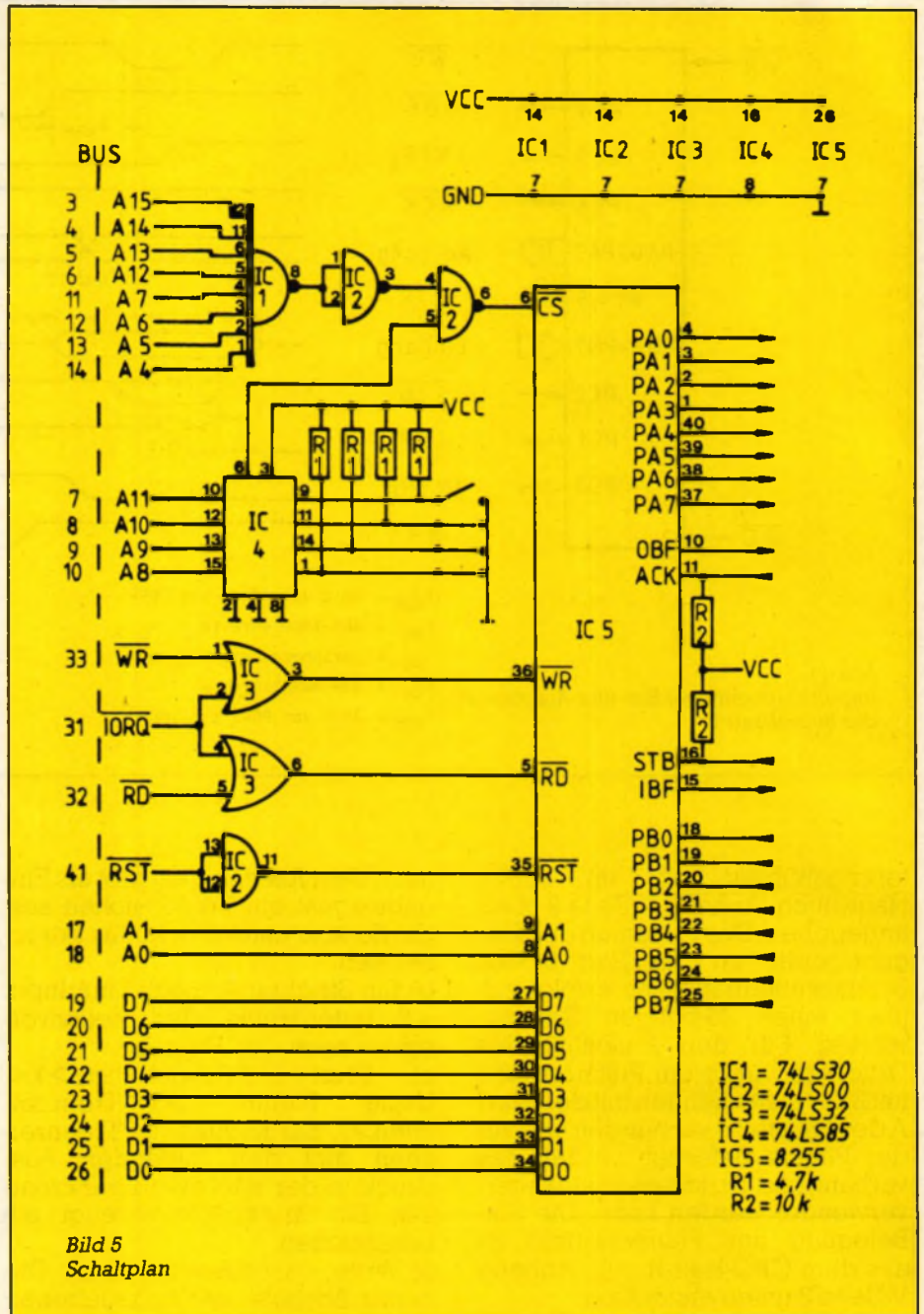


Bild 5  
Schaltplan

	OBF	INTE <sub>A</sub>	E/A	E/A	INTR <sub>A</sub>	INTE <sub>B</sub>	IBF	INTR <sub>B</sub>
Bit	7	6	5	4	3	2	1	0
	Gruppe A				Gruppe B			

Tabelle 3

- OBF = Zustand des Ausgabe-Puffer Flipflops
- INTE und INTR = Interrupt-Anforderung
- E/A = Eingang/Ausgang
- IBF = Zustand des Eingabe-Puffer Flipflops

Zeichen (d.h., das 255te Zeichen ist noch kein LF), dann wird die Fehlermeldung „over flow“ ausgegeben, und die Routine kehrt ebenfalls zurück. Die Beeinflussung des Carry-Flags läßt sich aus Tabelle 2 entnehmen.

#### 3.4 Routine IBUSY Adresse &A57C

Diese Routine prüft, ob Daten in

Tabelle 1

VEKTORADRESSE (RAM)	ALTE SPRUNGADRESSE (BETRIEBSSYSTEM)	NEUE SPRUNGADRESSE (RAM)
&BD3A &BD2B &BD2E &BD31	&2A98 SCREEN EDIT &07F2 PRINT CHAR &081B BUSY PRINTER &0807 SEND PRINTER	& A50D COMP1 & A592 COMP2 & A5CE OBUSY & A5C6 SEND

das Portregister B des Schnittstellenbausteins eingelesen wurden. Wenn Daten eingelesen wurden, kehrt sie mit gesetztem Carry-Flag zur aufrufenden Routine zurück, sonst ist das Carry-Flag gelöscht.

### 3.5 Routine INPUT Adresse &A589

In diesem Programmteil wird der Inhalt des Portregisters B in den Akku gelesen. Es wird dabei nicht geprüft, ob eine Eingabe über die Schnittstelle erfolgte. Das Carry-Flag wird gesetzt. Die Routine IBUSY enthält eine Verzögerungsroutine (&BD19), Warten auf Bildrücklauf, damit sichergestellt ist, daß vor Aufruf der Routine INPUT das Strobe-Signal des Eingabegerätes zurückgesetzt ist.

### 3.6 Routine COMP2 Adresse &A592, Vektoradresse &BD2B

Diese Routine hat verschiedene Prüffunktionen. Der BASIC-Interpreter gibt bei den Input-Befehlen alle Kommentare, Fragezeichen u.s.w. auf dem Gerät mit der Kanalnummer des Input-Befehls aus. Dieses ist aber bei den Befehlen, welche die Ein- Ausgabeschnittstelle ansprechen, nicht sinnvoll. Die COMP2-Routine überprüft deshalb anhand aktueller Tokens, ob es sich um einen Eingabebefehl handelt, und gibt in diesem Fall alle Zeichen auf dem aktuellen Bildschirmfenster aus. Handelt es sich um einen Ausgabe-Befehl, dann verzweigt die Routine je nach dem Inhalt der Speicherstelle &A400 entweder zur Druckroutine des Betriebssystems (alte Drucker-schnittstelle) oder zur Routine OCHAR (neue Ein- Ausgabeschnittstelle).

### 3.7 Routine OCHAR Adresse &A5AE

Diese Routine gibt das im Akku befindliche Zeichen über die Ein-Ausgabeschnittstelle aus und kehrt bei gelungener Übertragung mit gesetztem Carry-Flag zur aufrufenden Routine zurück.

### 3.8 Routine SEND Adresse &A5C6, Vektoradresse &BD2E

Diese Routine liest das im Akku befindliche Zeichen in das Datenregister des Port A, wodurch das dem Strobe entsprechende OBF-Signal auf Low-Pegel gesetzt wird.

### 3.9 Routine OBUSY Adresse &A5CE, Vektoradresse &BD2E

Diese Routine überprüft das OBF-Signal. Solange dieses Signal Low ist, ist der Inhalt des Portregisters A noch nicht vom Ausgabegerät übernommen worden. Ist OBF High, dann können Daten zur Ausgabe in das Datenregister des Port A geschrieben werden. In Tabelle 2 sind noch einmal die wichtigsten Maschinenroutinen zusammengefaßt.

Tabelle 2

ROUTINE	ADRESSE	RÜCKSPRUNG-BEDINGUNG	CARRY-FLAG	BEMERKUNGEN
ICHAR	&A51A	LF	C=1	Pufferinhalt wird bei Basic-Programmen der angegebenen Variablen übergeben in Basic wird das Programm abgebrochen in Basic wird das Programm nicht abgebrochen, der Pufferinhalt wird der angegebenen Variablen zugewiesen
		ESC	C=0	
		OVERFLOW	C=1	
IBUSY	&A57C	keine	C=0 C=1	Daten wurden nicht eingelesen Daten wurden eingelesen
INPUT	&A589	keine	C=1	Registerinhalt von Port B wird in den Akku geschrieben
OCHAR	&A5AE	keine	C=1 C=0	Zeichenausgabe gelungen Zeichen konnte nicht ausgegeben werden
SEND	&A5C6	keine	unbekannt	Akkuinhalt wird in das Register Port A geschrieben
OBUSY	&A5CE	keine	C=1 C=0	Datenausgabe noch nicht beendet es können Daten ausgegeben werden

# BASIC-LADER

```

10 MEMORY %A4FF:z=0:RESTORE
20 FOR i=%A500 TO %A5E2
30 READ a#:a=VAL("&"+a#):s=s+a
40 POKE i,a
50 NEXT
60 IF s<>%7BEB THEN PRINT"Fehler in Data-Zeilen":END
70 POKE %ED3A,%C3:POKE %ED3B,%D:POKE %ED3C,%A5
80 POKE %ED2B,%C3:POKE %ED2C,%92:POKE %ED2D,%A5
90 POKE %ED2E,%C3:POKE %ED2F,%CE:POKE %ED30,%A5
100 POKE %ED31,%C3:POKE %ED32,%C6:POKE %ED33,%A5
110 CALL %A501:END
120 DATA 00,3e,af,01,f3,f9,ed,79,97,32,00,a5,c9
130 DATA 3a,21,ac,fe,08,20,03,c3,1a,a5,cf,98,aa
140 DATA c5,d5,e5,21,a4,ac,06,ff,e5,c5,cd,7c,a5,dc,89,a5,dc,67,a5,c1,e1,28,14,d
,6b,a5,28,05,dc,6f,a5,28,10,cd,09,bb,fe,fc,28,03,c3,22,a5
150 DATA 36,00,e1,d1,c1,c9
160 DATA e5,21,5c,a5,7e,23,b7,c4,5a,bb,20,f8,e1,37,c3,45,a5
170 DATA 07,20,4f,76,65,72,66,5c,6f,77,00
180 DATA fe,0a,37,c9
190 DATA fe,0d,37,c9
200 DATA 77,c5,d5,e5,cd,5d,bb,e1,d1,c1,23,05,c9
210 DATA c5,01,f2,f9,ed,79,1f,1f,cd,19,bd,c1,c9
220 DATA c5,01,f1,f9,ed,79,c1,37,c9
230 DATA d5,e5,5f,2a,34,ae,23,7e,e1,fe,a3,28,3d,fe,a6,29,39,3a,00,a5,b7,7b,d1,2
,03,cf,f2,87
240 DATA c5,cd,b4,a5,c1,c9
250 DATA 01,32,00,cd,c6,a5,cd,ce,a5,3f,d8,10,f9,0d,20,f6,b7,c9
260 DATA c5,01,f0,f9,ed,79,c1,c9
270 DATA c5,d5,5f,01,f2,f9,ed,79,17,7b,d1,c1,3f,c9
280 DATA 7b,d1,cd,5a,bb,37,c9

```

```

A500 20 ORG %a500
A500 30 INIT EQU %a501
A500 40 COMP1 EQU %a50d
A500 50 EDIT EQU %a517
A500 60 ICHAR EQU %a51a
A500 70 S2 EQU %a52b
A500 90 RUECK EQU %a545
A500 90 FEHLER EQU %a54b
A500 100 ERROR EQU %a55c
A500 110 LFEED EQU %a567
A500 120 CR EQU %a56b
A500 130 PUFFER EQU %a56f
A500 140 IBUSY EQU %a57c
A500 150 INPUT EQU %a599
A500 160 COMP2 EQU %a592
A500 170 OCHAR EQU %a5ae
A500 180 QWRIT EQU %a5b4
A500 190 SEND EQU %a5c6
A500 200 DBUSY EQU %a5ce
A500 210 ISET EQU %a5dc
A500 00 220 STELLE DB 0
A501 230
A501 3EAF 240 INIT LD a,%X10101111:Port a = Ausgabe, Port b = Eingabe
A503 01F3F8 250 LD bc,%F8F3:Steuerregister
A506 ED79 260 OUT (c),a
A509 97 270 SUB a
A509 3200A5 280 LD (stelle),a
A50C C9 290 RET
A50D 300
A50D 3A21AC 310 COMP1 LD a,(%ac21):Kanalnummer in Akku laden
A510 FE02 320 CF %9:Kanalnummer = 9?
A512 2083 330 JR nz,edit:CR, dann springe ins Betriebssystem
A514 C21AA5 340 JP 1char
A517 CF 350 EDIT RST %08
A519 99AA 360 ON %aa98:screen edit (Betriebssystem)
A51A C5 370 ICHAR PUSH bc
A51B D5 380 PUSH de
A51C E5 390 PUSH hl
A51D 21A4AC 400 LD hl,%aca4:Startadresse Puffer
A520 06FF 410 LD b,%ff:Zaehler fuer Puffer
A522 E5 420 S1 PUSH hl:Startadresse retten
A523 C5 430 PUSH bc:Zaehler retten
A524 C07CA5 440 CALL 1busu:Einsabegeraet bereit?
A527 DC99A5 450 CALL c,input:wenn ja, dann Zeichen holen
A52A DC67A5 460 CALL c,lfeed:LF?
A52D C1 470 POP bc
A52E E1 490 POP hl
A52F 2B14 490 JR z,rueck:Ruecksprung, wenn LF
A531 DC2FA5 500 CALL c,cr:CR?
A534 2085 510 JR z,s2:CR nicht abspeichern
A536 DC6FA5 520 CALL c,puffer:Zeichen in Puffer ablesen
A539 2B10 530 JR z,fehler:springe, wenn Puffer voll

```

```

A53B C09BB 540 S2 CALL %bb09:Zeichen von Tastatur holen
A53E FEFC 550 CF %fc:Break?
A540 2903 560 JR z,rueck
A542 C322A5 570 JP s1
A543 3600 580 RUECK LD (hl),%00:letztes Zeichen = 0
A547 E1 590 POP hl
A548 D1 600 POP de
A549 C1 610 POP bc
A54A C9 620 RET:Ruecksprung
A54B E5 630 FEHLER PUSH hl
A54C 215CA5 640 LD hl,error
A54F 7E 650 S3 LD a,(hl)
A550 29 660 INC hl
A551 87 670 OR a
A552 C45AB2 690 CALL nz,%bb5a:Fehlerbezeichng. ausgeben
A553 20F8 690 JR nz,s3:bis Zeichen = 0
A557 E1 700 POP hl
A558 37 710 SCF
A559 C345A5 720 JP rueck
A55C 07 730 ERROR DE %07
A55D 20 740 DE %20
A55E 4F 750 OR %4f
A55F 76 760 DE %76
A560 65 770 DE %65
A561 72 780 DE %72:" overflow"
A562 66 790 DE %66
A563 6C 900 DE %6c
A564 6F 910 DE %6f
A565 77 920 DE %77
A566 00 930 DE %00
A567 FE0A 940 LFEED CP %0a:LF?
A569 37 950 SCF:carry=1
A56A C9 960 RET
A56B FE0D 970 CR CP %0d:CR?
A56D 37 980 SCF:carry=1
A56E C9 990 RET
A56F 77 900 PUFFER LD (hl),a:Zeichen in Puffer laden
A570 C5 910 PUSH bc
A571 D5 920 PUSH de
A572 E5 930 PUSH hl
A573 CD%DBB 940 CALL %bb5d
A576 E1 950 POP hl
A577 D1 960 POP de
A578 C1 970 POP bc
A579 23 980 INC hl:naechste Adresse
A57A 05 990 DEC b:Zaehler um eins erniedrigen
A57B C9 1000 RET
A57C C5 1010 IBUSY PUSH bc
A57D 01F2F8 1020 LD bc,%f2f8:Port c adressieren
A57E ED79 1030 IN a,(c):Port c einlesen
A57F 1F 1040 RRA
A580 1F 1050 RRA:Port c, Bit 1 ins Carry (IBF-Signal)

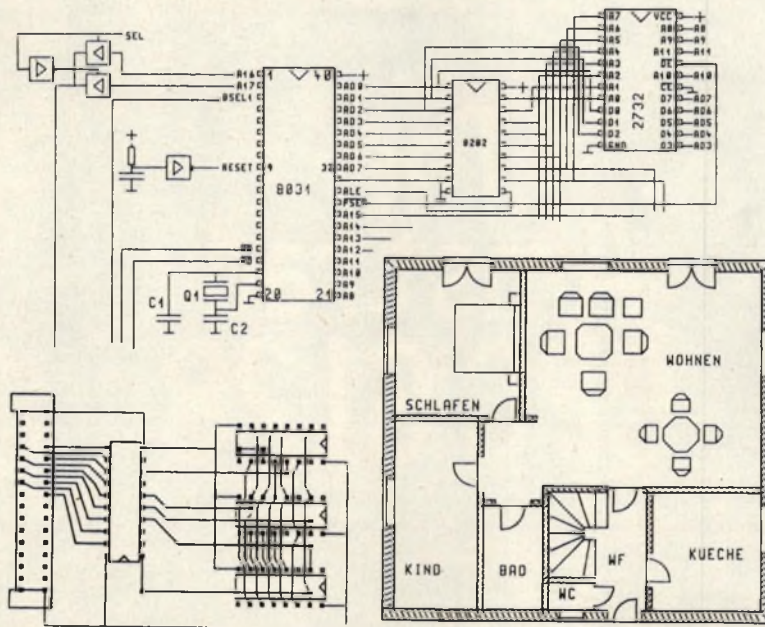
```



<pre> AS99 05 1090 INPUT PUSH bc AS9A 01F1F9 1100 LD bc,%9f1 Port b adressieren AS9B ED79 1110 IN a(c) Port b einlesen AS9C C1 1120 POP bc AS9D 37 1130 SCF Carry-Flag = 1 AS9E C9 1140 RET AS9F 1150 -----Print-Routine AS92 D5 1160 COMP2 PUSH de AS93 E5 1170 PUSH hl AS94 5F 1180 LD e,a AS95 2604RE 1190 LD hl,(%ae34) Adresse des aktuellen Statements AS96 23 1200 INC hl AS97 7E 1210 LD a,(hl) Token des aktuellen Statements AS98 E1 1220 POP hl AS99 FE9C 1230 CP %a3 Input? AS9A 293D 1240 JR z,iset AS9B FE96 1250 CP %a6 Line? AS9C 2939 1260 JR z,iset AS9D 3A00A5 1270 LD a,(stelle) AS9E 27 1280 OR a AS9F 7B 1290 LD a,e AA00 D1 1300 POP de AA01 290C 1310 JR z,ochar AA02 CF 1320 RST %00 AA03 F297 1330 DN %87f2 AA04 C5 1340 OCHAR PUSH bc AA05 CDE4RE 1350 CALL await AA06 C1 1360 POP bc AA07 C9 1370 RET AA08 013200 1380 ONRIT LD bc,%00C2 Zaehler AA09 CDC6RE 1390 CALL send Zeichen an Port a bereitstellen AA0A CDC6RE 1400 S4 CALL obusy Uebertragung gelungen? AA0B 3F 1410 CCF AA0C D8 1420 RET c wenn ja, dann Ruecksprung (Carry = 1) AA0D 10F9 1430 DJNC #4 </pre>	<pre> AS01 86 1440 DEC c AS02 26F6 1450 JR %z,s4 AS03 B7 1460 OR a keine Ausgabe moeglich (Carry = 0) AS04 C9 1470 RET Ruecksprung AS05 C5 1480 SEND PUSH bc AS06 01F0F9 1490 LD bc,%9f0 Port a adressieren AS07 ED79 1500 OUT (c),a Daten an Port a anlesen AS08 C1 1510 POP bc AS09 C9 1520 RET AS0A C5 1530 OBUSY PUSH bc AS0B 05 1540 LD e,a AS0C 5F 1550 LD bc,%9f2 port c adressieren AS0D 01F2F9 1560 IN a(c) Port c einlesen AS0E 17 1570 RLA Bit 7 von Port c in Carry (UBF-Signal) AS0F 7B 1580 LD a,e AA10 D1 1600 POP de AA11 C1 1610 POP bc AA12 C9 1620 CCF AA13 C9 1630 RET AA14 7B 1640 ISET LD a,e AA15 D1 1650 POP de AA16 CDE4RE 1660 CALL %bb5a Ausgabe auf laufendes Bildschirmfenster AA17 37 1670 SCF Carry = 1 fuer gelungene Ausgabe AA18 C9 1680 RET AA19 1690 -----Sprungvektoren AA20 1700 ORG %bd0a AA21 C30DA5 1710 JP come1 AA22 1720 ORG %bd2b AA23 C392A5 1730 JP come2 AA24 EDCE 1740 ORG %bd2e AA25 C3C6A5 1750 JP obusy AA26 1760 ORG %bd31 AA27 C3C6A5 1770 JP send </pre>
---	---

# MICA

Das CAD-Programm für Ihren Computer



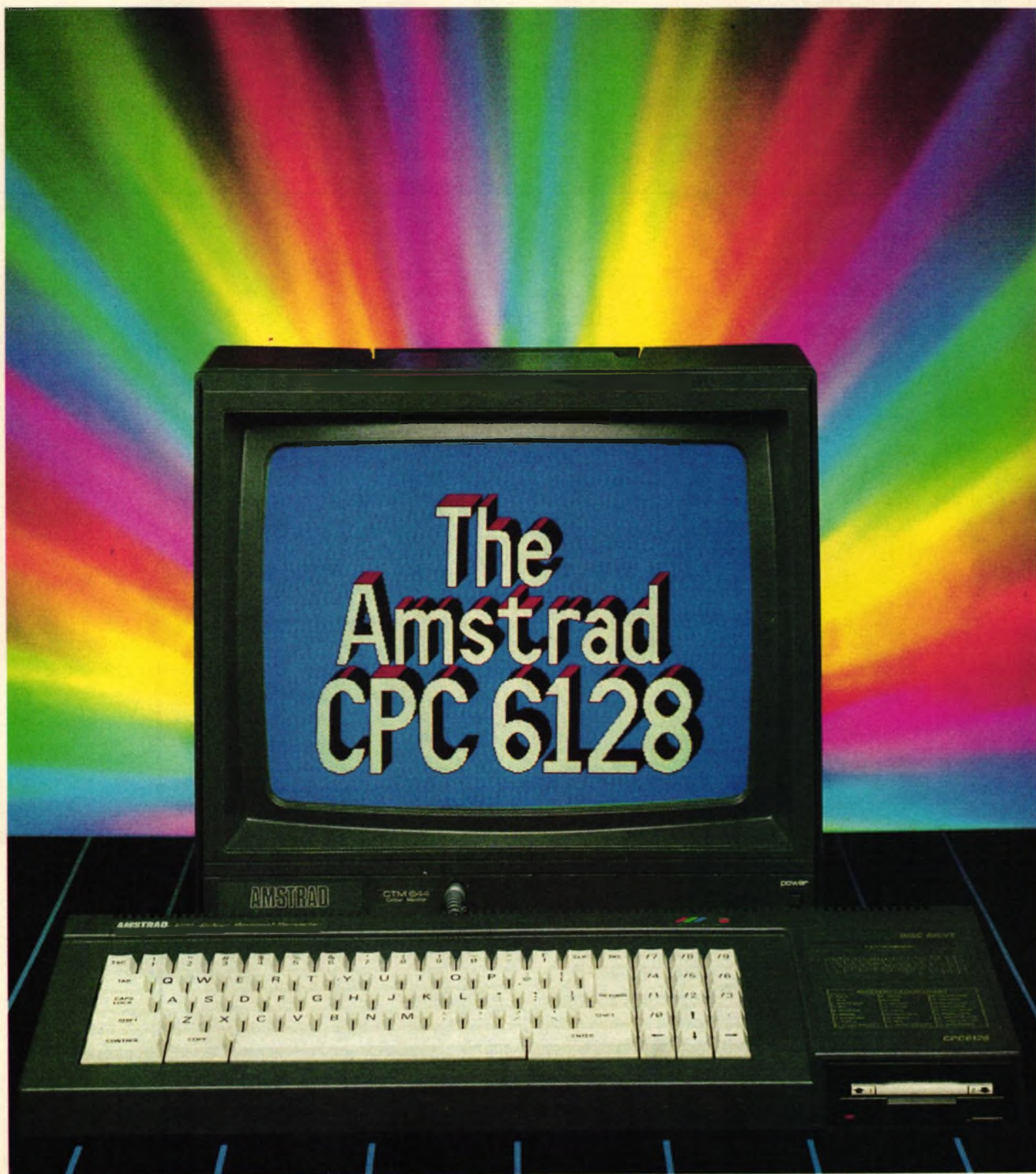
- maßstabsgerechtes Erstellen von Zeichnungen und Layouts in Zoll und mm
- Symbolbibliotheken
- Symbole vergrößern, verkleinern, drehen, spiegeln
- Ausgabe auf Drucker und Plotter
- 6 Zeichenebenen und ... und ... und ...

Lieferbar für: CPC664, CPC6128, IBM-PC und kompatibel, MC-Term1 Apple II, C128, PMS88, Atari 260/520 ST, Joyce

Mica wird für 198.- DM geliefert, für Atari 260/520 ST 298.- DM V-Scheck oder zuzüglich 5.- DM bei Nachname

Alleinvertrieb: E & C Ruppert Zellmeier Dompfaffstr. 127a  
8520 Erlangen Tel. 09131/440303

# CPC 6128



# Mit diesem Artikel wollen wir einmal die wichtigsten Neuerungen des CPC 6128 gegenüber seinen Vorgängern herausstellen.

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: SETDFCOM / DEVICE.COM / LANGUAGE.COM / SHOWCOM / DATE.COM / SETCOM / SETLSTCOM

SETDEF [PAGE] bewirkt das Anhalten 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 bestimmte zu spezifizierende Files zu suchen hat. So kann zum Beispiel veranlaßt werden, daß alle .COM-Dateien zuerst auf Laufwerk C gesucht 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 Bild-



Unser Bild zeigt den 6128 mit der anderen Tastatur.

schirmausgabe 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 ini-

tialisiert 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 FILE-COPY 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.

sitzer 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

sprechen 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.



Unser Bild zeigt die Schnittstellen des CPC 6128.

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 Be-

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 An-

## Drei Bilder pro Sekunde

Der CPC 6128 besitzt 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. .

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)

# KALENDER

von FRANK THIELEN

Kalender des Jahres 1986

## JANUAR

MO	DI	MI	DO	FR	SA	SO
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

## FEBRUAR

MO	DI	MI	DO	FR	SA	SO
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28		

## MAERZ

MO	DI	MI	DO	FR	SA	SO
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

## APRIL

MO	DI	MI	DO	FR	SA	SO
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

## MAI

MO	DI	MI	DO	FR	SA	SO
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

## JUNI

MO	DI	MI	DO	FR	SA	SO
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

## JULI

MO	DI	MI	DO	FR	SA	SO
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

## AUGUST

MO	DI	MI	DO	FR	SA	SO
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

## SEPTEMBER

MO	DI	MI	DO	FR	SA	SO
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

## OKTOBER

MO	DI	MI	DO	FR	SA	SO
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

## NOVEMBER

MO	DI	MI	DO	FR	SA	SO
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

## DEZEMBER

MO	DI	MI	DO	FR	SA	SO
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

Mit dem Programm Kalender kann man sich für die Jahre zwischen 1582 und 2500 die zwölf Monate auf Drucker oder Bildschirm

ausdrucken lassen. Schaltjahre werden dabei selbstverständlich automatisch berücksichtigt. Nach dem Start des Programms und der

Eingabe des gewünschten Jahres erhält man einen Ausdruck wie in dem obenstehenden Beispiel für das Jahr 1986.



Farben-Tip

# Oben 16 Farben im Bild und gleichzeitig unten 80 Zeichen Text

Mehr Mode  
in einem  
Bild

Die Schneider-Computer können bekanntlich auf 3 verschiedene Arten Grafik und Text darstellen: 16 Farben und eine so geringe Auflösung, daß nur noch 20 Zeichen/Zeile dargestellt werden ODER ein Kompromiß aus 40 Zeichen/Zeile bei 4 Farben pro Bildschirmpunkt ODER die höchste, aber einfarbige Auflösung von 80 Zeichen/Zeile. Wäre es nicht schön, wenn man z.B. im oberen Bildschirm Drittel ein 16farbiges Bild UND im unteren Drittel den Begleittext mit 80 Zeichen/Zeile darstellen könnte? Sicher haben Sie es schon vermutet: Weil es so schön ist, werden wir es jetzt auch machen.

Unser Problem ist es also, verschiedene MODEs gleichzeitig auf dem Bildschirm darzustellen. Prinzipiell gibt es zwei Lösungen dafür: Wir könnten einen Computer, z.B. einen Atari 130 kaufen, dessen Videoprozessor das schon von Haus aus kann, oder wir müßten uns eines Software-Tricks bedienen. Da die erste Möglichkeit aus praktischen Erwägungen ausscheidet, beschäftigen wir uns besser mit



der notwendigen Software. Bei jedem Interrupt wurde das Farb-Änder-Programm aufgerufen, wodurch 6 farbige Balken untereinander entstanden. Damals störte es uns, daß die Balken genau an der gleichen Stelle stehenblieben, ohne zu flackern oder sich zu bewegen, aber heute ist das genau das, was wir brauchen.

## Interrupts sind die Lösung

Angenommen, wir wollten die obere Hälfte des Bildes in Mode 0 (16 Farben) darstellen und die untere Hälfte in Mode 2 (80 Zeichen).

Wir brauchen dazu lediglich Mode 0 einzuschalten, zu warten, bis der Elektronenstrahl das halbe Bild gezeichnet hat, und dann Mode 2 einzuschalten. So weit, so gut. Aber wir wollen ja nicht nur das tun, sondern zwischendurch sollen ja auch noch unsere normalen Programme weiterlaufen, also rufen wir das Programm über Interrupts auf. Das hat noch einen weiteren Vorteil: 300 Interrupts (Unterbrechungen) werden pro Sekunde erzeugt, 50 mal pro Sekunde wird ein Bild gezeichnet. Wir haben also 6 Unterbrechungen in einem Bild, können theoretisch also 6 mal den Bildschirmmodus ändern. Vorausgesetzt, daß der erste Interrupt zeitlich mit dem Beginn eines neuen Bildes übereinstimmt, brauchen wir also nur bei jedem 3. Interrupt nach dem Bildanfang auf Mode 2 und synchron mit dem Bildanfang auf Mode 0 zu schalten, und schon sind wir da, wo wir hin wollten.

Gesagt, getan. Listing 1 ist das Ergebnis unserer bisherigen Überlegungen. Es wird auf die übliche Weise ein Interruptblock eingebunden, so daß ROUTI dreihundertmal pro Sekunde aufgerufen wird. (Zeichen 70/80. Im Listing entspricht + bd19 &bd19, ist also hexadezimal. Das Zeichen + wird vom Schneider-Devpac-Assembler verwendet; wer einen anderen Assembler hat, muß das Symbol entsprechend ändern.) Bei jedem Aufruf von ROUTI wird mitgezählt, der

## Listing 1

wieviele Aufruf es war. Handelt es sich um Nr. 1, wird Mode 0 eingeschaltet, bei Nr. 3 wird Mode 2 eingeschaltet, bei Nr. 6 wird der Zähler wieder zurückgesetzt, weil ja 6 Interrupts pro Bild auftreten. Allerdings wird der Zähler abwärts, also nicht von 0-5 gezählt. Zeilen 240-380 erledigen das und wählen den entsprechenden Mode an. Sehr wichtig sind aber die Zeilen vor und nach dem Einbinden in die Interruptkette: Da die obere Bildschirmhälfte in Mode 0 dargestellt werden soll, schalten wir erst einmal auf diesen um und warten dann auf den Strahlrücklauf (40-60). Erst dann wird KL-ADD-FASTTICKER aufgerufen. Anschließend werden die Interrupts gesperrt, da wir jetzt noch einmal warten müssen, und zwar länger als die Zeitperiode, die zwischen zwei Interrupts

```

A000          10      org  #a000          ;LISTING Nr.2
A000          20      ent  $              ;Startadresse &a000
A000 210AA0    30  init: ld  hl,free4      ;4 freie Bytes zur Verwaltung
A003 010EAO    40      ld  bc,tab1       ;Sprungadressentabelle
A006 CDD1BC    50      call #bcd1        ;RSX'es einbinden
A009 C9        60      ret              ;das war es
A00A          70  free4: defs 4
A00E 16A0     80  tab1: defw ntab
A010 C31CA0    90      jp  init2         ;Modeswitch einschalten
A013 C335A0   100     jp  aus          ;' ausschalten
A016 4FCE     110  ntab: defb "0","N"+#80   ;Tabelle der Namen
A01B 4F46C6   120     defb "0","F","F"+#80
A01B 00       130     defb 0            ;Ende der Tabelle
                140 ;
A01C 3E00     150  init2: ld  a,0          ;Mode 0
A01E CD1CBD   160     call #bd1c        ;einschalten
A021 CD19BD   170     call #bd19        ;MC_WAIT_FLYBACK
A024 2141A0   180     ld  hl,block      ;binde in Interruptkette ein
A027 CDE3BC   190     call #bce3        ;KL_ADD_FAST_TICKER
A02A F3       200     di              ;keine Stoerung bitte...
A02B 3E06     210     ld  a,6          ;Zaehler auf Startwert
A02D 3269A0   220     ld  (auf),a       ;(wird heruntergezählt)
A030 CD19BD   230     call #bd19        ;warte auf Strahlruecklauf
A033 FB       240     ei              ;ist synchronisiert
A034 C9       250     ret
                260 ;
A035 2141A0   270  aus:  ld  hl,block      ;Block wieder aus...
A03B CDE6BC   280     call #bce6        ;...Interruptkette herausnehmen
A03B 3E02     290     ld  a,2
A03D CD1CBD   300     call #bd1c        ;Mode 2 einschalten
A040 C9       310     ret
                320 ;
A041 0000     330  block: defw 0          ;Interruptblock
A043 0000     340     defw 0
A045 00       350     defb 0
A046 B1       360     defb #B1
A047 4CA0     370     defw routi
A049 00       380     defb 0
A04A 0000     390     defw 0
                400 ;
A04C 3A69A0   410  routi: ld  a,(auf)        ;Das eigentliche Programm
A04F 3D       420     dec  a          ;erniedrige Zaehler und...
A050 3269A0   430     ld  (auf),a
A053 FE00     440     cp  0            ;...ueberpruefe welcher Interrupt
A055 280B     450     jr  z,mod0      ;bei Nummer 0 ->Mode 0
A057 FE02     460     cp  2
A059 C0       470     ret  nz        ;auch nicht Nr.2
A05A 3E02     480     ld  a,2
                490 ;sonst...
A05C C31CBD   490     jp  #bd1c        ;Mode 2!
A05F 3E06     500  mod0: ld  a,6          ;Zaehler wieder auf Anfang
A061 3269A0   510     ld  (auf),a
A064 3E00     520     ld  a,0
A066 C31CBD   530     jp  #bd1c        ;Mode 0 einschalten
A069 06       540  auf:  defb 6          ;Zaehlvariable
A06A          550     end

```



liegt. Nun werden der Zähler auf seinen Startwert gesetzt und die Interrupts — und somit der Aufruf unseres Programms — wieder eingeschaltet, sobald ein neuer Frame-Fly aufgetreten ist (90-140). Diese komplizierte Prozedur ist notwendig, um eine möglichst gute Synchronisation bzgl. des Umschaltprogramms und der tatsächlichen Darstellung auf dem Bildschirm zu erreichen. Sonst könnte es leicht passieren, daß unser Mode-0-Bereich irgendwo in der Bildmitte „hängt“ und die Bereiche darunter und darüber in Mode 2 angezeigt werden. Wenn Sie nun Listing 1 laufen lassen, werden Sie allerdings feststellen, daß wir keine exakte Teilung des Bildschirms in der Mitte haben, sondern daß unten 7 Zeilen Text und oben 18 Zeilen Grafik angezeigt werden. Das liegt aber nicht etwa daran, daß nun beim falschen Interrupt umgeschaltet wurde, sondern daran, daß ja nicht nur die eigentliche Bildfläche, sondern die gesamte BILDSCHIRMfläche, also auch der Border vom Elektronenstrahl mitgezeichnet wird, so daß die Interrupts nicht synchron zur Bildschirmposition 0,0, sondern zur oberen Bildschirmcke liegen. Um also tatsächlich in der Bildmitte umzuschalten, muß die Nummer in Zeile 290 entsprechend erniedrigt werden.

## Andere Modes/Farben...

Listing 1 ist natürlich noch nicht das Nonplusultra. Ärgerlich ist z.B., daß Mode 0 noch etwas in die erste Textzeile hineinreicht. Man kann hier noch nachregulieren, obwohl der nächste Interrupt erst ein ganzes Stück tiefer kommt: Nach Setzen des neuen Modes noch ein bißchen mit einer Verzögerungsschleife a'la LD Bx:DJNZ \$ warten. 50 wäre hier ein guter Wert für x, aber die genaue Zahl zu ermitteln, überlasse ich Ihnen (denken Sie aber daran, daß dadurch andere Programme, die normal ablaufen, verzögert werden können).

Kein Problem ist es auch, andere Modes einzuschalten (einfach die Zahlen in Zeile 40,310 und 350 ändern) oder noch mehr Modes auf dem Bildschirm zu mischen. Ebenfalls eine interessante Idee wäre

Hisoft GENA3.1 Assembler. Page 1.

Pass 1 errors: 00

```

A000      10      org #a000      ;LISTING NR.1
A000      20      ent $          ;hier Pr-Start
                                30 ;
A000 3E00      40 init: ld a,0      ;Initialisieren:erst Mode 0
A002 CD1CBD      50      call #bd1c      ;einschalten
A005 CD19BD      60      call #bd19      ;warte auf Strahlruecklauf
A008 2119A0      70      ld hl,block      ;binde in Interruptkette ein
A00B CDE3BC      80      call #bce3      ;KL_ADD_FAST_TICKER
A00E F3          90      di          ;dauert laenger als 1/300 Sek
A00F 3E06      100     ld a,6      ;Zaehler auf Startwert
A011 3241A0      110     ld (count),a      ;(wird heruntergezaehlt)
A014 CD19BD      120     call #bd19      ;warte auf Strahlruecklauf
A017 FB          130     ei
A018 C9          140     ret
                                150 ;
A019 0000      160 block: defw 0          ;Block mit Daten fuer
A01B 0000      170     defw 0          ;Interrupthandler
A01D 00          180     defb 0
A01E 81          190     defb #81
A01F 24A0      200     defw routi
A021 00          210     defb 0
A022 0000      220     defw 0
                                230 ;
A024 3A41A0      240 routi: ld a,(count)      ;Das eigentliche Program
A027 3D          250     dec a          ;erniedrige Zaehler und...
A028 3241A0      260     ld (count),a
A02B FE00      270     cp 0          ;...ueberpruefe welcher Interrupt
A02D 2808      280     jr z,mod0      ;bei Nummer 0 ->Mode 0
A02F FE02      290     cp 2
A031 C0          300     ret nz          ;auch nicht Nr.2
A032 3E02      310     ld a,2          ;sonst...
A034 C31CBD      320     jp #bd1c      ;Mode 2!
A037 3E06      330 mod0: ld a,6      ;Zaehler wieder auf Anfang
A039 3241A0      340     ld (count),a
A03C 3E00      350     ld a,0
A03E C31CBD      360     jp #bd1c      ;Mode 0 einschalten
A041 06          370 count: defb 6      ;Zaehlvariable
A042           380     end

```

Pass 2 errors: 00

Table used: 71 from 211

Executes: 40960

```

1 REM LISTING Nr.3
2 DATA 21,0A,A0,01,0E,A0,CD,D1,BC,C9,00,00,00,00,1
6,A0,C3,1C,A0,C3,35,A0,4F,CE,4F,46,C6
3 DATA 00,3E,00,CD,1C,BD,CD,19,BD,21,41,A0,CD,E3,B
C,F3,3E,06,32,69,A0,CD,19,BD,FB,C9
4 DATA 21,41,A0,CD,E6,BC,3E,02,CD,1C,BD,C9,00,00,0
0,00,00,81,4C,A0,00,00,00,3A,69,A0,3D
5 DATA 32,69,A0,FE,00,28,08,FE,02,C0,3E,02,C3,1C,B
D,3E,06,32,69,A0,3E,00,C3,1C,BD,06
6 MEMORY &9FFF:MODE 2
7 RESTORE 2:FOR x=&A000 TO &A069:READ w$:POKE x,VA
L("&"+w$):NEXT:CALL &A000
8 PRINT"öON - schaltet Modesplit ein":PRINT"öOFF -
schaltet Modesplit aus"

```

es, gleichzeitig mit dem Umschalten des Modus auch die angezeigten Farben zu ändern, so daß man in Mode 0 andere Farben verwenden könnte als in Mode 1, die aber doch gleichzeitig auf dem Bildschirm erscheinen. Es gibt hier noch viele weitere Möglichkeiten, und ich bin sicher, Sie haben genug Phantasie, um das Programm selbst weiterzuentwickeln.

## Neue BASIC-Befehle

Damit wären wir bei Listing 2. Hiermit ist es möglich, die „gesplittete“ Darstellung mit den Befehlen ION ein- und mit IOFF auszuschalten. INIT erfüllt hier eine ganz andere Funktion als in Listing 1: Es werden lediglich die beiden Befehle ON und OFF generiert, und erst INIT2 bindet dann ROUTI in die Interruptkette ein. Bei einem OFF-Kommando wird dann AUS aufgerufen und der Interruptblock wieder entfernt, so daß wieder alles einheitlich in einem Mode dargestellt wird.

Das letzte Programm, Nr. 3, ist eine absolute Ausnahme: Hier haben wir auch mal einen „Happen“ für reine BASIC-Programmierer: Es tut im Prinzip dasselbe wie Nr. 2, aber man braucht keinen Assembler, um es zu nutzen. Also: Listing 3 an den Anfang des eigenen Programmes, und man kann mit ION den Modesplit ein- und mit IOFF wieder ausschalten.

Sicherlich fällt jedem beim Herumexperimentieren mit dem Modesplit schon nach sehr kurzer Zeit auf, daß die Schrift recht merkwürdig aussieht, wenn Sie in den Mode 0-Bereich gerät: Der Grund dafür ist klar: Der Zeichengenerator nimmt natürlich immer noch an, daß auf dem Bildschirm Mode 2 (beim Herumexperimentieren mit all unseren Beispiellistings sollte in Mode 2 gearbeitet werden) dargestellt wird, und so werden nicht die richtigen Bitmuster für Mode 0 erzeugt. Um das zu verhindern, kann man WINDOW +0,1,80,19,25 definieren oder mit POKE &b1c8,0 die Schrift an Mode 0 anpassen (an dieser Adresse steht der Bildschirm-

modus bei 464. Siehe Vergleich zwischen 464 und 664), aber leider funktioniert die Grafik deshalb trotzdem nicht. Mit einem POKE &b1c8,0:call &bbba ist es schon besser, aber immer noch nicht perfekt. Man muß also den Zeichengenerator komplett an Mode 0 anpassen, um oben normal zeichnen zu können. Die Lösung dafür werde ich nachreichen. Allerdings wird

## Vorschau

man normalerweise auch direkt Mode 0-Bilder von Diskette oder Tape in den Bildschirm laden und so keine Probleme mit dem Zeichengenerator bekommen. Man sollte Modesplit nach Tape- oder Diskettenbenutzung neu einschalten, da bei Tape/Disk-Betrieb die Interrupts ausgeschaltet werden und die Synchronisation gestört wird. TMB

# Der Schneider Tip

ROM-CALL's ohne Einspruchbedingungen zum Ausschneiden und Sammeln

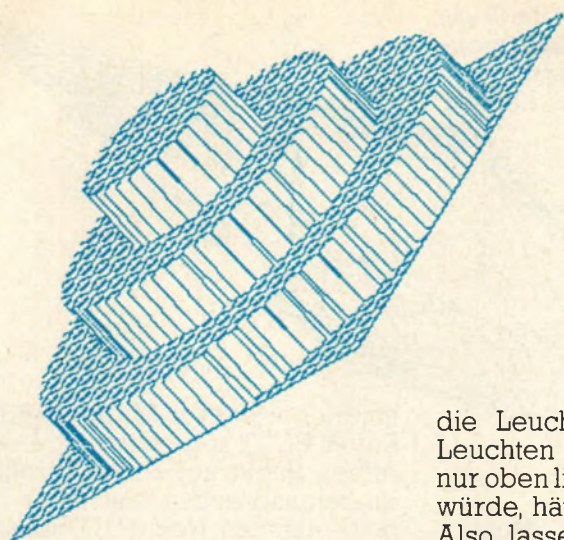
CALL &BB00 Initialisierung der Tastatur. CALL &BB03 Rücksetzung des Tastaturpuffers.  
CALL &BB06 Warte auf das nächste Zeichen von der Tastatur.  
CALL &BB19 Warte auf nächste Tastatureingabe.

CALL &BB3F in Direkteingabe: keine Repeatfunktion mehr.  
CALL &BB48 BREAK ausschalten.  
CALL &BB45 BREAK einschalten.  
CALL &BB4E Setzt Textcursor auf Bildschirmfang, oben links.

CALL &BB57 Verbieta, daß Zeichen am Bildschirm dargestellt werden.  
CALL &BB54 Erlaubt, daß Zeichen am Bildschirm dargestellt werden.

CALL &BB7B Lasse Cursordarstellung (Anwender) zu. Z.B. bei Input.  
CALL &BB7E Verbieta Cursordarstellung (Anwender). Z.B. bei Input.

CALL &BB9C Tausche Ink's für Stift und Papier.  
CALL &BB14 Löscht Bildschirm bis Cursorposition.  
CALL &BB6B Verhindert Bildschirmausgabe folgender Meldungen: Press PLAY then any Key: Press REC and Play then any Key: Found „FILENAME“ Block „N“, Loading „FILENAME“ Block „N“, Saving „FILENAME“ Block „N“.



# Der Grafik-Kurs für Einsteiger, Teil I

Die Schneider-Computer sind in der Lage, großartige Grafiken zu erzeugen, und das BASIC bietet auch dementsprechende Befehle an. Aber gerade wegen der vielen Möglichkeiten sind die entsprechenden Befehle auch schwer zu verstehen.

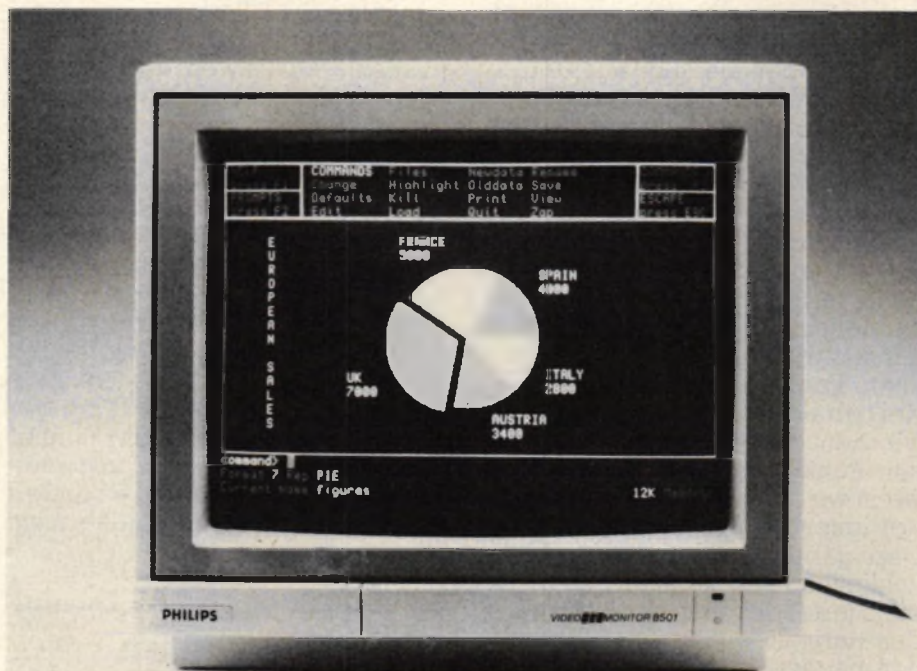
In diesem Kurs werden wir uns zunächst einmal mit den Grundlagen, also den Grafikbefehlen und verschiedenen Möglichkeiten zum Zeichnen auf dem Bildschirm, beschäftigen. Später wenden wir uns dann den Verfahren zu, mit denen wir mit eben diesen Grafikbefehlen Kreise, Kugeln, 3-dimensionale Darstellungen und so weiter erzeugen können. Auch, oder besser: gerade Anfänger werden hier auf ihre Kosten kommen, umfangreiche BASIC-Kenntnisse sind nicht notwendig.

## Für Anfänger

Alles auf dem Bildschirm besteht aus kleinen Punkten. Schalten Sie ihren Schneider einmal an und sehen sich die Schrift ganz genau an (nah heran gehen!). Wenn Sie genau genug hinsehen, werden Sie feststellen, daß alle Buchstaben aus lauter kleinen Punkten zusammengesetzt sind. Um zu verstehen, warum das so ist, muß man etwas über die Art wissen, wie der Computer das Bild, das Sie auf dem Monitor sehen, aufbaut: Stellen Sie sich vor, daß noch gar nichts auf dem Bildschirm steht, er also komplett schwarz ist. Jetzt schießt von hinten gegen die Leuchtfläche der sogenannte Elektrodenstrahl, der

die Leuchtfläche zum (logisch...) Leuchten bringt. Wenn die aber nur oben links in der Ecke leuchten würde, hätten wir nicht viel davon. Also lassen wir den Strahl blitzschnell von links nach rechts, und von oben nach unten rasen (genau wie beim Lesen). Überall da, wo der Strahl auf die Fläche trifft, leuchtet sie, und so leuchtet schließlich der ganze Bildschirm. Da der Strahl so schnell über den Bildschirm zum Leuchten gebracht wird, haben wir den Eindruck eines ruhigen, stehenden Bildes. Von einem leeren Bildschirm allein hätten wir allerdings nicht viel, und so müßten wir den Strahl auch noch aus- und einschalten, um nur bestimmte Teile zum Leuchten zu bringen, eben jene Teile, die dann Buchstaben oder andere Symbole

dieses Prinzips ist notwendig, um die Grafikfähigkeiten optimal nutzen zu können; (in Wirklichkeit ist das Ganze sogar noch etwas komplizierter, da ja auch noch verschiedene Farben dargestellt werden sollen, aber das braucht uns jetzt wirklich nicht zu kümmern). Um zu wissen, welche Punkte dargestellt werden sollen, schaut der Computer in einem bestimmten Teil seines Speichers nach, und von der Größe dieses Teils hängt es ab, wieviele Punkte wir auf dem Bildschirm darstellen können. Beim Schneider sind es maximal  $640 \times 200 = 128\,000$  Punkte. Eine ganz schöne Menge. Je mehr Punkte, um so weniger Farben. Sicher ist Ihnen das Wörtchen „maximal“ aufgefallen. Werden denn evtl. auch weniger Punkte dargestellt? So ist es. Denn die An-



Unser Bild zeigt die bekannte Kuchendiagrammdarstellung.

darstellen sollen. All das ist normalerweise für uns völlig unwichtig, da der Computer sich selber darum kümmert, aber die Kenntnis

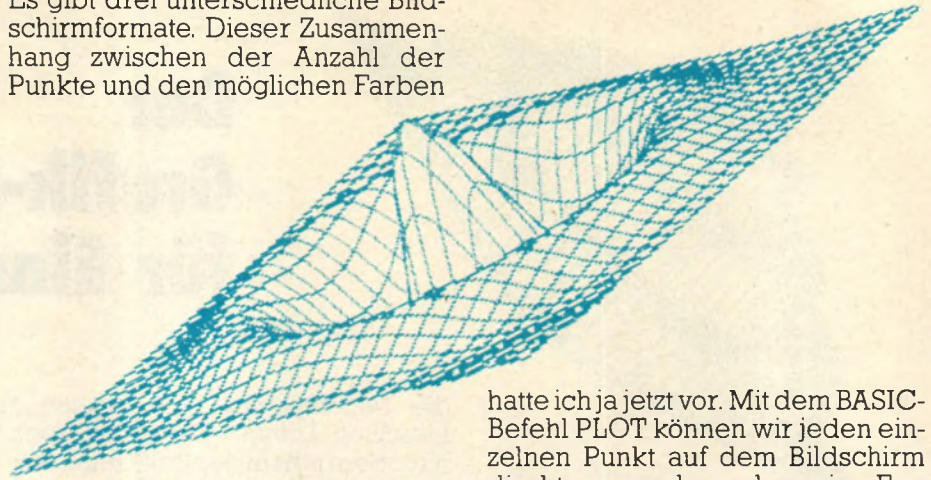
zahl der Punkte hängt, wie schon gesagt, von der Größe des Speicherteils (ab jetzt „Grafikspeicher“ genannt) ab, in dem steht, welche

Punkte gesetzt werden sollen und welche nicht. Stellen Sie sich vor, daß dieser Teil aus 128 000 Schaltern (Bits) besteht (für jeden Punkt einen), und wenn der entsprechende Schalter an ist, dann leuchtet auch der Punkt. Mit einem einzigen Schalter können wir so zwei Farben darstellen: Ist der Schalter aus, leuchtet der Punkt nicht, oder eben in der ersten Farbe. Ist der Schalter aber an, dann leuchtet der Punkt, und zwar in der anderen Farbe. Beim Einschalten wären diese beiden Farben zum Beispiel blau, wenn der Schalter aus ist und gelb, wenn er an ist (gelbe Schrift auf blauem Grund). Aber zwei Farben sind nicht gerade üppig, vor allem, wo doch jedes Computerspiel von bunten Explosionen und vielen farbigen Einzelheiten lebt. Wie also mehr Farben auf den Bildschirm bringen? Der Trick dafür ist ebenso einfach wie genial: Wir nehmen zwei Schalter für nur einen Punkt!

## Zwei oder vier Farben

Damit können wir dann den einen Punkt in vier verschiedenen Farben darstellen: 1 Farbe: beide Schalter aus; 2 Farben: nur Schalter 1 an; 3 Farben: nur Schalter 2 an; 4 Farben: beide Schalter an. Leider hat dieser geniale Trick auch einen Nachteil: Wir können nur noch die Hälfte der Punkte darstellen, da wir ja ZWEI Schalter pro Punkt brauchen. Diese Methode kann natürlich noch beliebig weiter ausgebaut werden: Mit vier Schaltern können wir schon einen einzigen Punkt in 16 verschiedenen Farben darstellen, aber wir haben auch nur noch ein Viertel der ursprünglichen Punkte zur Verfügung. Sie werden sich nun sicher fragen, wie wir den ganzen Bildschirm noch mit Punkten vollkriegen wollen, wenn wir nur noch (z.B.) ein Viertel der ursprünglichen Anzahl haben. Dazu gäbe es prinzipiell zwei Möglichkeiten: Wir könnten nur noch ein Viertel des ursprünglichen Bildes darstellen, also das Bild kleiner machen, oder die Punkte viermal größer. Die zweite Möglichkeit ist wohl sinnvoller, deshalb wird sie auch praktiziert: Dadurch, daß die Punkte breiter werden, bleibt auch das Bild genauso breit wie vorher, obwohl nur noch ein Viertel der Punkte pro Zeile dargestellt wird.

Es gibt drei unterschiedliche Bildschirmformate. Dieser Zusammenhang zwischen der Anzahl der Punkte und den möglichen Farben



ist allerdings nicht nur Theorie, sondern genauso wird es in Ihrem Computer gemacht! Wir haben die Wahl, ob wir 2,4 oder 16 Farben gleichzeitig darstellen wollen. Dazu gibt es in BASIC den Befehl MODE. Nach MODE muß eine Nummer von 0 bis 2 folgen, um anzugeben, welches Bildformat gewünscht ist. Die drei Formate sind: MODE 0: Der Bildschirm ist 160 Punkte breit, jeder Punkt kann in einer von 16 verschiedenen Farben leuchten. MODE 1: Der Bildschirm ist 320 Punkte breit, jeder Punkt kann 4 verschiedene Farben haben. Dieser MODE wird automatisch beim Einschalten des Computers gewählt. MODE 2: Der Bildschirm ist 640 Punkte breit, jeder Punkt kann in 2 Farben erscheinen (Vordergrund oder Hintergrund). In allen MODES werden 200 Punkte untereinander dargestellt (also zweihundert Zeilen). Und so kommt auch die maximale Anzahl von 640x200 Punkten zustande, nämlich im MODE 2. Probieren Sie es doch einmal: MODE 0 ENTER bringt eine ganz breite Schrift hervor, mit MODE 2 ENTER ist sie viel schmaler. Die Anzahl der Punkte, die ein Buchstabe breit ist, ist bei beiden MODES gleich, nur sind die Punkte in MODE 0 breiter. Die Höhe der Buchstaben ist jedoch immer gleich, da ja immer 200 Punkte vertikal dargestellt werden. Es ändert sich also nur die Punktbreite, nicht die Punkthöhe.

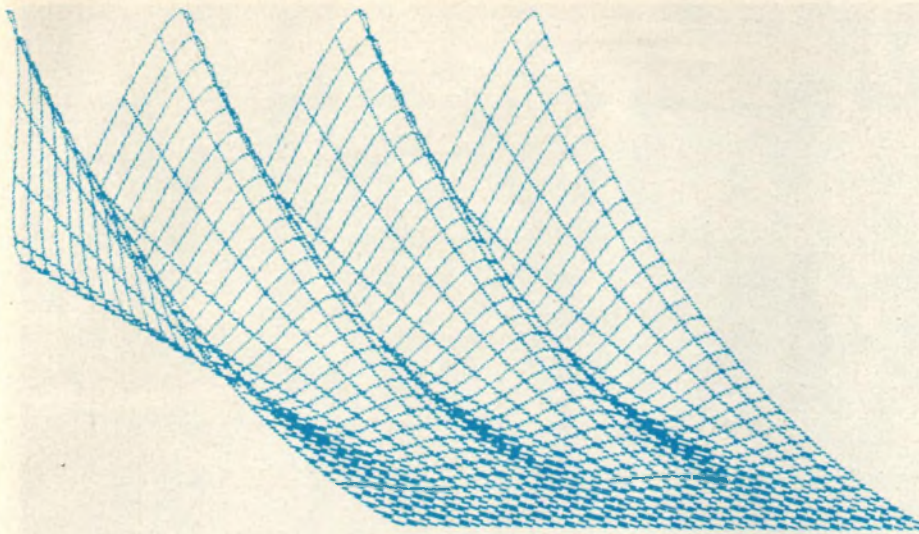
## Die ersten Zeichenversuche

Sie haben die Nase voll von dem theoretischen Kram und wollen endlich praktisch was machen? OK, ok, nicht aufregen, genau das

hatte ich ja jetzt vor. Mit dem BASIC-Befehl PLOT können wir jeden einzelnen Punkt auf dem Bildschirm direkt ansprechen, also seine Farbe bestimmen. Nach PLOT müssen mindestens zwei Zahlen kommen (durch ein Komma getrennt), die angeben, welcher Punkt gemeint ist. Die erste Zahl gibt dabei seine horizontale Lage an, die zweite Zahl seine vertikale Lage, genau wie bei einem Koordinatensystem. Eine Null bei der horizontalen Lage bedeutet dabei ganz links, 639 ganz rechts. Für die zweite Zahl gilt: 0 ist ganz unten, 399 ganz oben. Keine Frage, daß Sie jetzt gemerkt haben, daß hier etwas nicht stimmt: Die erste Zahl sollte doch eigentlich je nach MODE verschieden sein, und die zweite sollte doch nur bis 200 gehen...? Stimmt EIGENTLICH, aber da wir jetzt erst etwas Praktisches machen wollten, lassen Sie mich die Erklärung auf gleich verschieben. Also weiter: PLOT 320,200 ENTER müßte eigentlich

## Nur ein Punkt

einen Punkt genau in die Mitte des Bildschirms setzen ... aber es passiert nichts! Nicht aufregen, es hat geklappt! Nur haben wir den Punkt in der Farbe Null, also Blau, gesetzt. Da er aber vorher schon blau war, kann man auch weiter nichts sehen. Die Farbe des Punktes kann man durch eine dritte Zahl bestimmen. 0 ist dabei die erste, 3 die vierte Farbe (oder maximal 1 bei MODE 2 und maximal 15 bei MODE 0). Also noch ein Versuch: PLOT 320,200,1 ENTER und...VIOLA! Es hat geklappt. Mit einer starken Leserbrille oder einem gewöhnlichen Haushaltselektronenmikroskop kann man den Punkt ohne weiteres sehen. (Wenn keine Farbe angegeben wird, wird die zuletzt angegebene Farbe verwendet). Probieren Sie es mit verschiedenen Zahlen (und in den 3MODES),



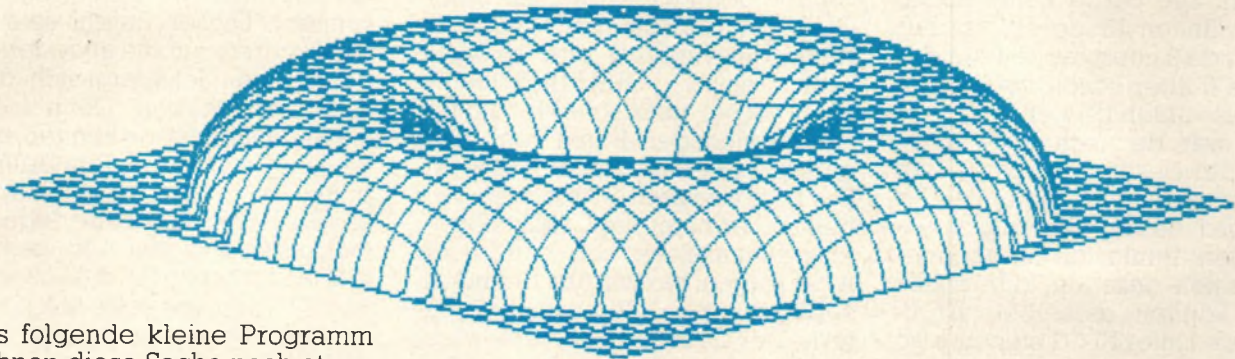
aber rechnen Sie sich vorher ungefähr aus, wo der Punkt erscheinen muß, da Sie ihn sonst vielleicht nicht finden. Vor allem in MODE 2 muß man sich da schon ziemlich anstrengen.

Computer tut nur so, als hätten wir 640 Punkte!! Das hat nämlich den unschätzbaren Vorteil, daß wir nicht komplett mit neuen Zahlen rechnen müßten, nur weil wir einen anderen MODE eingeschaltet ha-

pro Punkt! Und nach genau demselben Prinzip geht das auch bei den vertikalen Zahlen, nur hat es hier einen anderen Grund: Wenn wir nur Koordinaten von 0.199 hätten, würde jeder Kreis, den wir auf den Bildschirm zeichnen, wie ein plattgedrückter Ball aussehen, denn das Verhältnis von der tatsächlichen Bildschirmbreite zur Höhe (in cm) entspricht eben mehr 640/400 als 640/200! Es ist also nur ein Trick, um das Zeichnen zu erleichtern. So einfach ist das.

Sie kennen jetzt zwei Grafikbefehle: MODE und PLOT. Experimentieren Sie etwas damit herum, um damit vertraut zu werden. Und damit Sie auch noch etwas zu experimentieren haben, versuchen Sie es mal mit dem Befehl DRAW. Dieser Punkt ist dann das ENDE der Linie. Diese Linie hat die angegebene oder die vorherige Farbe (wenn Sie keine Farbe angeben). Ersetzen Sie doch z.B. das Plot im obigen

## Ein erstes Programm



Das folgende kleine Programm soll Ihnen diese Sache noch etwas verdeutlichen. Dabei müßten Sie wissen, daß der Befehl RND + Zahl irgendeine zufällige Zahl zwischen 0 und der angegebenen Zahl erzeugt. PLOT RND\*639, RND\*399, RND\*3 setzt also einen Punkt in irgendeiner Farbe irgendwo auf den Bildschirm.

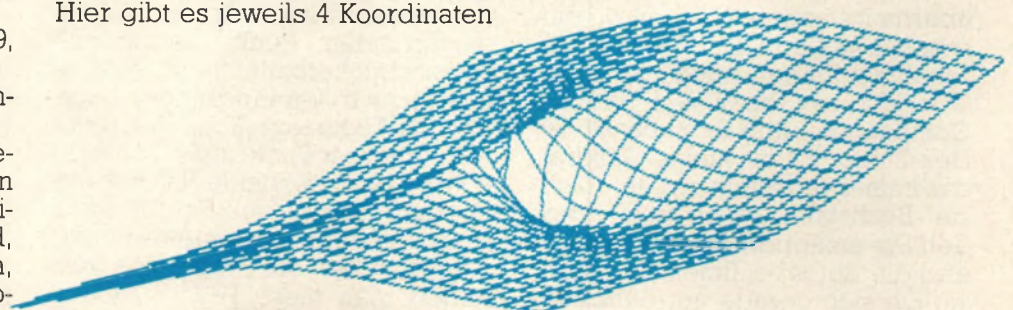
```
10 INPUT „Wieviele Punkte?“,anzahl
20 For i=1 to anzahl
30 PLOT RND*639, RND*399, RND*3
```

```
40 NEXT i: PRINT „Fertig“; anzahl; „Punkte gesetzt.“, GOTO 10
```

Die Koordinaten bleiben in jedem MODE gleich. Wenn Sie nun wieder etwas aufnahmefähig für einen neuen Happen Theorie sind, wollen wir noch die Frage klären, wieso wir immer 639 als größte horizontale, und, noch merkwürdiger, 399 als vertikale Zahl haben. Der

ben. Dazu teilt er die Zahl einfach durch 2 oder 4. Deshalb werden Sie auch keinen Unterschied sehen, wenn Sie in MODE 1 (320 Punkte horizontal) erst Punkt 0,0 und dann Punkt 1,0 plotten. Es handelt sich nämlich um denselben Punkt. Erst bei 2,0 wird der nächste Punkt sichtbar. In MODE 0 (160 Punkte) ist es sogar noch extremer: Hier gibt es jeweils 4 Koordinaten

Programm durch DRAW ... Anschließend werden wir uns mit weiteren Grafikbefehlen beschäftigen und einige sehr schöne Bilder erstellen.



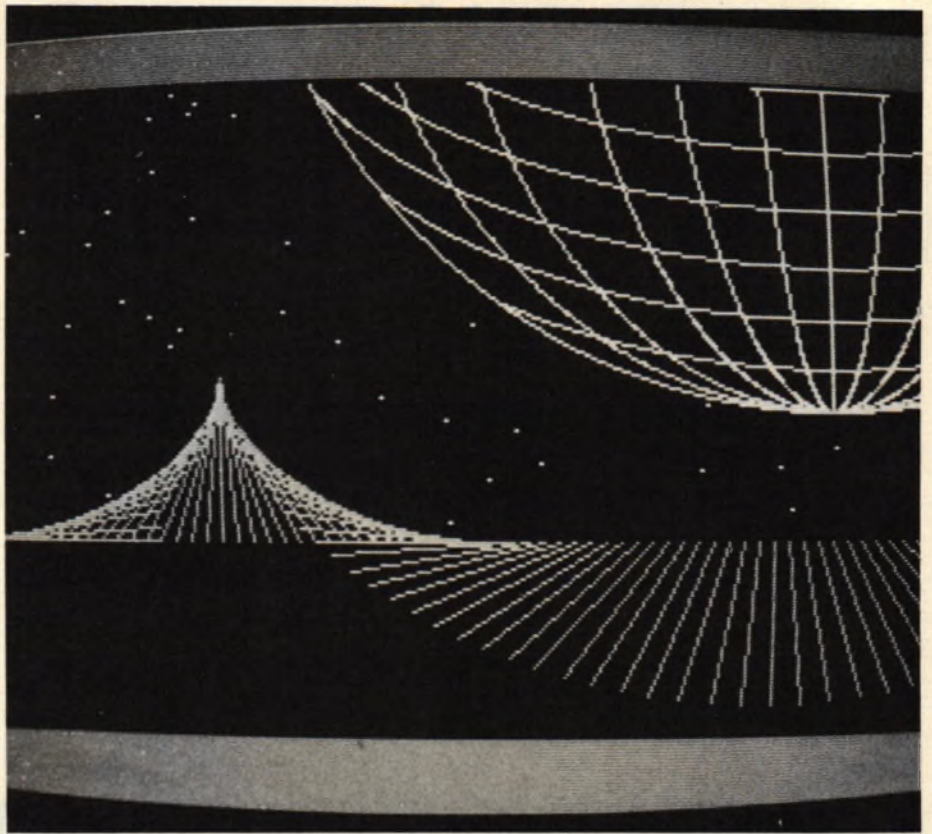
Wir haben herausgefunden, daß der Computer das Bild, das er auf dem Bildschirm darstellt, aus vielen kleinen Punkten zusammensetzt. Die Anzahl und somit die Größe dieser Punkte hängt vom gewählten Bildschirmmodus ab, den wir mit MODE bestimmen können.

Obwohl es verschieden viele Punkte in den einzelnen Modes sind, tut der Computer so, als hätten wir immer 640\*400 Punkte, um das Programmieren zu erleichtern. Die Anzahl der Farben, die wir darstellen können, hängt ebenfalls vom Mode ab, je weniger Punkte dargestellt werden, um so mehr Farben können angezeigt werden. Soviel zum Theoretischen. Aber wir haben auch schon zwei, genauer, drei Befehle kennengelernt: MODE, mit dem wir einen Bildschirmmodus von 0 bis 2 wählen können, wobei 2 die höchste Auflösung darstellt. PLOT, mit dem wir die Farbe jedes einzelnen Punktes auf dem Bildschirm bestimmen können, und zwar indem wir seine x-Koordinate von 0-639, seine y-Koordinate von 0-399 und die Farbe, in der er erscheinen soll, angeben. Die Farbe hängt dabei vom gewählten Mode ab, so daß wir in Mode 2 nur zwischen 0 und 1, in Mode 0 aber schon zwischen 0 bis 15 auswählen können.

Dann war da noch der Befehl DRAW, der besser „LINE“ heißen sollte, denn er zieht eine Linie. Die Linie wird dabei vom zuletzt geploTTeten Punkt zur angegebenen Position gezogen, d.h., hinter DRAW können dieselben Argumente wie hinter PLOT stehen, also x,y und Farbe.

## Ein Cursor für die Grafik

Wenn Sie Ihren Schneider, oder überhaupt jeden Computer, einschalten, sehen Sie außer den Einschaltmeldungen auch noch den CURSOR, der Ihnen sagt, daß der Computer auf Ihre Eingabe wartet. Der Cursor ist das kleine Quadrat, das immer dort steht, wo der nächste Buchstabe erscheinen wird, den Sie eintippen. Der Cursor hat also die Aufgabe, Ihnen zu zeigen, wo Sie sich gerade auf dem Bildschirm „befinden“. Unser Schneider hat aber noch einen zweiten



Cursor, denn außer diesem TEXT-CURSOR (der uns hier aber nicht weiter interessiert) gibt es auch noch einen GRAFIKCURSOR. Aber dieser unterscheidet sich in einem wichtigen Punkt vom Textcursor: Man kann ihn nicht sehen! Dafür gibt es zwei Gründe: Zum einen soll er nicht die Grafik stören, aber zum anderen soll er uns auch nicht zeigen, wo wir uns befinden, sondern nur das Zeichnen erleichtern. Der Grafikkursor steht, wenn wir ihn sonst nicht bewegen, immer auf dem zuletzt gesetzten Punkt (z.B. mit PLOT oder bei DRAW auf dem letzten Punkt der Linie). Wenn wir nun den Befehl PLOT verwenden, haben wir mit dem Grafikkursor nicht viel zu tun, da wir ihn einfach nicht brauchen. Ganz anders sieht es aber bei DRAW aus. Erinnern Sie sich, daß wir weiter oben gesagt hatten, DRAW würde den angegebenen Punkt mit dem zuletzt gezeichneten Punkt verbinden? Das stimmt normalerweise, vorausgesetzt, wir verändern die Lage des Grafijkursors nicht, der dann ja auf diesem Punkt steht. In Wirklichkeit aber verbindet DRAW die Position, an der der Grafikkursor steht, mit dem angegebenen Punkt. Nun stellt sich die Frage: Wie verändert man diese Position? Ganz einfach: Dazu haben wir den Befehl MOVE x,y. (Denken Sie daran, daß

Sie hier KEINE Farbe angeben können). Dieser Befehl setzt den Grafikkursor auf die angegebenen Koordinaten. Jetzt zeigt sich, ob Sie mitgedacht haben: Dann müßten Sie jetzt ZWEI Wege kennen, um eine Linie zwischen dem Anfangspunkt x<sub>a</sub>y<sub>a</sub> und dem Endpunkt x<sub>e</sub>y<sub>e</sub> zu zeichnen! Überlegen Sie mal einen Moment und lesen Sie erst weiter, wenn Sie die Lösung haben — nun, die erste Möglichkeit ist, einen Punkt an x<sub>a</sub>y<sub>a</sub> zu setzen und dann die Linie zu ziehen: PLOT x<sub>a</sub>y<sub>a</sub>:DRAW x<sub>e</sub>y<sub>e</sub>, aber es gibt auch noch die Möglichkeit, nur den Grafikkursor an den Anfangspunkt zu setzen. Dazu brauchen Sie nur PLOT durch MOVE zu ersetzen.

## Alles ist relativ

Bisher haben wir bei allen Grafikbefehlen, also PLOT, DRAW und MOVE mit ABSOLUTEN Koordinaten arbeiten müssen. Das kann jedoch im Einzelfall ziemlich unpraktisch sein. Nehmen wir einmal an, Sie haben ein Programm geschrieben, das ein einfaches Strichlinienhaus auf den Bildschirm zeichnet. (Das wäre übrigens eine gute Übung für Sie — versuchen Sie es doch einmal, wenn Sie etwas Zeit

haben). Die Spitze des Dachs liegt z.B. auf dem Punkt 320, 200 (in der Bildschirmmitte), die Kanten des Daches entsprechend tiefer rechts und links davon. Eine Weile sind Sie mit dem Haus auch ziemlich glücklich, aber dann wollen Sie eine ganze Stadt auf den Bildschirm bringen (der Mensch kann doch nie genug kriegen!), d.h., Sie dürfen sich der freudigen Aufgabe hingeben, für jedes einzelne Haus die entsprechenden Punkte zu bestimmen und zusammen mit den entsprechenden Befehlen an Ihr ursprüngliches Programm anzuhängen. Aber es geht auch einfacher, und wenn Ihnen die Nützlichkeit des Grafikcursors bisher nicht so recht einleuchtete, dann wird Ihnen jetzt bestimmt ein Licht aufgehen: Von allen Grafikbefehlen gibt es nämlich auch RELATIVE Ausführungen, bei denen wir nicht direkt die Punkte angeben müssen, sondern die Entfernung, in der der Punkt von der augenblicklichen Position des Grafikcursors liegt. Wir können also sagen: „Zeichne eine Linie von hier (Cursorposition) nach dem Punkt, der 40 Punkte rechts und 50 Punkte tiefer liegt“. Wenn wir das ganze Haus auf diese Art beschreiben, brauchen wir nur den Grafikcursor am Anfang auf eine andere Start-Position zu setzen, um das Haus an einer anderen Stelle zu zeichnen. Die Namen der relativen Befehle bestehen aus dem jeweiligen alten Namen + einem „R“ für Relativ. Also: PLOT<sub>R</sub>, DRAW<sub>R</sub>, MOVER. Einige Beispiele dazu: PLOT<sub>R</sub> -4,0 zeichnet den neuen Punkt 4 Punkte links vom alten Punkt (-4, d.h. nach links, nach rechts wäre 4!!), MOVER 4,-8 bewegt den Grafikcursor 4 Punkte nach rechts und 8 Punkte nach unten, DRAW<sub>R</sub> 20,30 zeichnet eine Linie nach links oben, der Endpunkt liegt 20 Punkte höher und 30 Punkte rechts von der alten Grafikcursorposition. Denken Sie daran, daß alle diese Befehle die Position des Grafikcursors wieder verändern, daß sie also z.B. nach DRAW 20,30 auf dem Endpunkt der Linie liegt!

```
10 REM Faecher
20 MODE 2
30 FOR x=0 TO 639 STEP 3
40 MOVE 320,0
50 DRAW x,399,1
60 NEXT
70 GOTO 70:REM ESC fuer STOP!
```

```
10 REM Diamant
20 MODE 0
30 zaehler=0:tinte=1
40 INK 0,0:BORDER 0:INK 1,1:INK 2,2:INK 3,11:INK 4,14
50 FOR x=0 TO 639
60 MOVE 320,399
70 DRAW x,200,tinte:DRAW 320,0,tinte
90 zaehler=zaehler+1:IF zaehler=160 THEN zaehler=0:tinte=tinte+1
100 NEXT
110 GOTO 110:REM ESC fuer Stop
```

```
10 REM quadrate
20 MODE 0
30 x=RND*639:y=RND*639
40 MOVE x,y
50 tinte=RND*15
60 GOSUB 70:GOTO 30
70 DRAWR 25,0,tinte:DRAWR 0,25:DRAWR -25,0:DRAWR 0,-25:RETURN
```

```
10 MODE 0
20 tinte=1:xm=0:FOR ym=0 TO 399 STEP 29:GOSUB 70:xm=xm+40:tinte=tinte+1:NEXT
25 INK 0,1
30 FOR tinte=1 TO 15:INK tinte,1:NEXT
40 FOR tinte=1 TO 15:INK tinte,6:FOR verz=1 TO 20:NEXT verz:INK tinte,1:NEXT tinte
50 FOR tinte=15 TO 1 STEP -1:INK tinte,6:FOR verz=1 TO 20:NEXT verz:INK tinte,1:NEXT tinte
60 GOTO 40
70 REM einen Kreis zeichnen
80 DEG:MOVE 100+xm,ym
90 FOR winkel=0 TO 360 STEP 12
100 x=COS(winkel)*100+xm
110 y=SIN(winkel)*100+ym
120 DRAW x,y,tinte:NEXT winkel
130 RETURN
```

## 27 Farben

Wie wir in jedem Werbeprospekt für den Schneider nachlesen können, hat unser Computer 27 Farben. Aber selbst in Mode 0 kön-

```

5 REM Weg in die Ferne
10 MODE 1
11 INK 1,11
20 x=0:breite=80:FOR y=0 TO 200
30 MOVE x,y:DRAW x+breite,y,1
40 breite=breite-0.4
41 x=x+2.3
50 NEXT
60 FOR t=1 TO 100:PLOT RND*630,RND*400,RND*2+2:NEXT
70 INK 0,0:BORDER 0
80 INK RND*2+2,RND*27:FOR t=1 TO RND*400:NEXT:GOTO 80

```

nen wir nur 16 darstellen, von den 2 in Mode 2 gar nicht erst zu reden. Wo sollen da 27 Farben herkommen? Wenn Sie sich daran erinnern, wie wir unterschiedliche Farben darstellen (die Sache mit den 2 und 4 Schaltern), ist schon klar, wie so nur maximal 16 Farben GLEICHZEITIG dargestellt werden können. Aber unser Computer bietet uns die Möglichkeit, diese 16, 4 oder zwei, je nach Mode, aus einer PALETTE von 27 Farben auszuwählen! Wir können also schon 27 Farben darstellen, aber eben nicht alle gleichzeitig! Dazu haben wir den Befehl INK (Tinte). Je nach Mode haben wir 2 bis 16 TINTEN, mit denen wir zeichnen können. Aber welche Farbe diese Tinten haben, können wir selbst bestimmen. So können wir zwei oder mehr Tinten auch dieselbe Farbe geben oder sogar eine Tinte in zwei Farben blinken lassen (jetzt geht der Bezug zur Realität doch etwas verloren). Das geht ganz einfach so: INK, Tintennummer, Farbe 1, Farbe 2. Wir

## Alle Farben Tabelle 1

brauchen keine zwei Farben anzugeben, aber wenn wir es tun, dann blinkt alles, was in der entsprechenden Tinte gezeichnet wird, abwechselnd in der ersten und zweiten Farbe. Wie groß die Tintennummer ist, hängt vom Mode ab, aber die Farben können immer einen Wert von 0 bis 27 annehmen. Welche Nummer welche Farbe ergibt, können Sie aus Tabelle Nr. 1 ablesen. Probieren Sie es doch einmal: INK 1,3,6-. Die Buchstaben auf dem Bildschirm beginnen zwischen dunkel- und hellrot zu blinken, sie wurden also in Tinte Nummer 1 ge-

druckt (denn die haben wir ja geändert). Richtig effektiv wird es, wenn wir jetzt die Farbe des Papiers verändern (das ist Tinte Nr. 0 !!!) INK 0,6,3-. Daß wir überhaupt noch etwas erkennen können, liegt daran, daß die Farben von INK 0 genau „andersherum“ blinken als die von INK 1. Probieren Sie etwas herum! Wenn Sie überhaupt nichts mehr erkennen können, hilft CALL & BC02: Dieser Befehl weist allen Tinten die Farben zu, die sie normalerweise beim Einschalten haben. Übrigens können Sie sogar die Farbe des Bildschirmrandes ändern, der um das eigentliche Bild herumliegt und auf dem man

## Call & BC02

nichts zeichnen oder schreiben kann: Mit BORDER farbel,farbe2 können Sie seine Farben festlegen, ABER: Mit INKS haben Sie hier nichts am Hut, für den Border müssen Sie die Farben so festlegen, als ob er selbst eine INK wäre! Und zu guter Letzt können Sie sogar noch die Geschwindigkeit festlegen, in

der Farben auf dem Bildschirm blinken sollen (wenn Sie für irgendeine INK oder den BORDER zwei Farben angegeben haben): SPEED INK dauer1,dauer2 legt die Zeit fest, in der eine Farbe jeweils auf dem Bildschirm steht. Dauer1 ist die Zeit für die erste, Dauer2 die Zeit für die zweite Farbe (alle Farben richten sich danach, Zeiten für einzelne Tinten kann man nicht individuell festlegen). Normalerweise sind sowohl Dauer1 als auch Dauer2 = 10. Je kleiner diese Werte werden, umso kürzer ist die jeweilige Zeit!

## Die ersten Programme

Vielleicht haben Sie angesichts dieser vielen Befehle innerlich aufgestöhnt, aber ich kann Sie beruhigen: Sie kennen jetzt so ziemlich alle Grafikbefehle (die auch noch mal in Tabelle 2 zusammengestellt sind), und wir können jetzt endlich anfangen zu programmieren. Sie finden im folgenden einige Programme, die Sie abtippen können. Aber damit Ihnen das auch etwas nützt, sollten Sie sich VORHER überlegen, welche Grafik wohl durch das jeweilige Programm erzeugt wird und wie das Programm funktioniert. Wenn Sie irgendwelche Befehle im Programm nicht kennen, schlagen Sie diese im Handbuch nach! Und denken Sie daran: Wenn Sie Programme einfach nur abtippen, schaden Sie sich nur selbst, vor allem, da Sie im folgenden selbst schon einiges programmieren sollen. Also mitdenken! (Teil 2 ab S. 90.)

Tabelle 1: Diese Nummern haben die einzelnen Farben:

0 Schwarz	10 Blaugrün	20 helles Blaugrün
1 Blau	11 Himmelblau	21 Limonengrün
2 Hellblau	12 Gelb	22 Pastellgrün
3 Rot	13 Weiß	23 Pastellblaugrün
4 Magenta	14 Pastellblau	24 Hellgelb
5 Hellviolett	15 Orange	25 Pastellgelb
6 Hellrot	16 Rosa	26 Leuchtendweiß
7 Purpur	17 Pastellmagenta	
8 Helles Magenta	18 Hellgrün	
9 Grün	19 Seegrün	

Tabelle 2: Alle bisher behandelten Befehle:

-MODE	-PLOT	-BORDER
-DRAW	-DRAWR	-SPEED INK
-MOVE	-MOVER	-CALL & BC02
	-INK	



# Jetzt ist er endlich da

# **BASIC COMPILER 2.0**

— für Schneider CPC —  
464 + 6128

```
170 MODE 1
180 '** Farben fuer die Border, Ink 0 un
190 '** sechs moeglichen Balken
200 %MULTICOLOR, 1, 1, 26, 2, 11, 26, 11, 11, 3, 1
210 PRINT TAB(9) "Demonstrationsprogramm"
220 PRINT TAB(16) "fuer den"
230 %CHARSIZE, 0 '* elongierte Zeichen
240 PRINT " GRAPHIC-PROCESSOR"
250 %CHARSIZE '* elongierte Zeichen wie
260 INK 2, 0: INK 3, 23 '* "globalen" Inks
270 %GPAPER, 0 '* Ink des Grafikhintergru
280 '** Huegel
```

- Eigener Editor
- Lesen von ASCII-Dateien  
(bestehende Programme können  
gelesen werden)

```
380 '** Knopf auf Fahnenmast
390 %GPEN, 2
400 %CIRCLE, 158, 355, 12, 5, 200, 160
410 MOVE 158, 355: %PAINT
420 '** Flagge
430 MOVE 162, 350: DRAWR 20, -50
440 %RECTANGLE, 182, 304, 206, -112
450 MOVE 182, 200: DRAWR -16, -122
460 '** Schneiderlogo
470 %GPEN, 1
480 %CIRCLE, 240, 267, 30, 30, 180, 0
490 %CIRCLE, 240, 267, 5, 5, 180, 0
```

- erzeugter Object-Code ist ohne  
Compiler lauffähig
- Stringverarbeitung
- bis 30-fache Geschwindigkeit
- veränderbarer Anfangszeiger
- Object-Files können von BASIC  
aufgerufen werden
- 5-pass-Compiler

## Die Highlights:

- Fließkommaarithmetik
- Integerarithmetik
- über 60 K frei für Quelltext  
(über 20 K frei bei CPC 464)
- REPEAT UNTIL - Schleifen

```
500 %POLYGON, 240, 297, 280, 297, 280, 272, 240
510 %POLYGON, 240, 260, 280, 260, 280, 237, 240
520 %CIRCLE, 330, 230, 30, 30, 0, 180
530 %CIRCLE, 330, 230, 5, 5, 0, 180
540 %POLYGON, 330, 260, 290, 260, 290, 237, 330
550 %POLYGON, 330, 225, 290, 225, 290, 200, 330
560 %GPEN, 1 ' rot
570 MOVE 240, 250: %PAINT '* linken Teil d
580 %GPEN, 2 ' schwarz
590 MOVE 330, 220: %PAINT '* rechter Teil
600 %GPEN, 3 ' Pastellblaugruen
610 MOVE 184, 200: %PAINT '* Farbe der Fla
```

**Sofort  
bestellen**

Einsenden an: **SOFTWARE TEAM, Joachim Günster, Mühlenstraße 12, 5431 Boden**

**Ja**, Ihr Angebot hat mich überzeugt und ich bestelle:  
 Compiler 6128 99.—/Stk.  Compiler CPC 464 99.—/Stk.

Name: \_\_\_\_\_ Vorname: \_\_\_\_\_  
PLZ/Ort: \_\_\_\_\_ Straße: \_\_\_\_\_

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

**Disketten-Tip**

# 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 gewuenschten 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,zwl          ;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 #1B          ;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
    
```

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).

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-

```

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 11:  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 06B4    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,l2
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 3BD9    320      jr   c,l1
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   l1        ;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. TMB

Pass 2 errors: 00

Table used: 54 from 211

Executes: 40960

# Listenschutz 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.

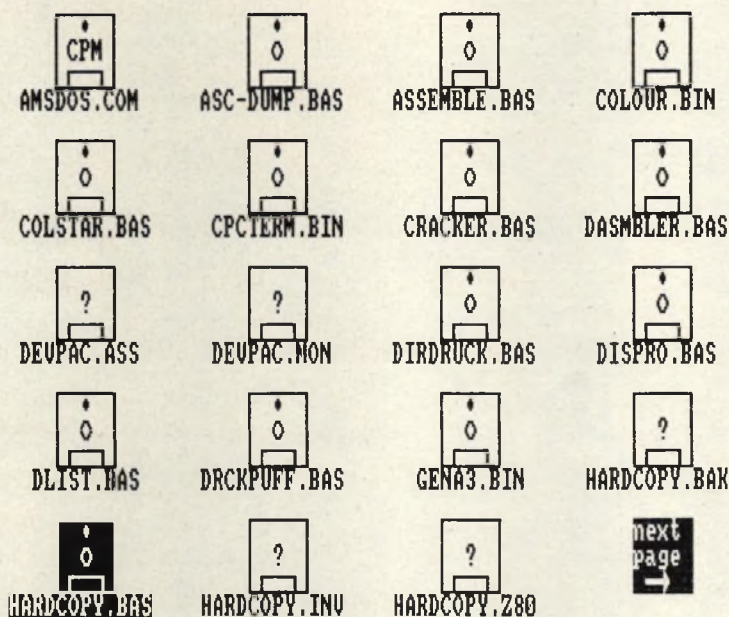
Obiges Programm nutzt die Tatsache, daß vor Ablauf der READY-Routine zur RAM-Adresse &AC01 gesprungen wird, wo normalerweise nur ein RET-Befehl steht. Hier wird nun ein Sprung zu einer Routine eingetragen, die in die Speicherstelle &AE45 eine Null (= ungeschützt) lädt. Diese Routine steht ab Adresse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

```

10 FOR i=&A600 TO &A504
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

```

# DIRLIST



↑ ↓ ← → / JOY(0)  
zum auswahlen

'ENTER' / FEUER  
zum starten

'L' zum loeschen

'SPACE': NEU

Der neue Superrechner, der Mega Atari, war einer der Renner des Jahres 1985.

Woher kommt das? Nur weil er einen riesigen Speicher hat? Weil er einen sehr schnellen Prozessor hat? Oder liegt es an seinem extrem günstigen Preis? Nun, diese Gründe spielen sicher auch eine sehr wichtige Rolle, aber einen Faktor darf man nicht unterschätzen: Die Faszination, die von GEM, dem Betriebssystem des Ataris, ausgeht. Dadurch ist es erstmals möglich, daß auch Leute, die absolut keine Ahnung von Computern haben, damit arbeiten können. Mit Dirlist haben wir versucht, das Einlesen und Starten von Programmen auf den Schneider-Computern ähnlich einfach zu machen, wie bei GEM. Dadurch können nun auch einmal die anderen Familienmitglieder die neuen Programme ausprobieren, die sonst auf den guten Willen des jeweiligen 'Fachmanns' angewiesen sind.

## Was ist Dirlist?

Wenn Sie Dirlist eingetippt haben, sollten Sie praktischerweise

```

10 REM *****
20 REM **          Dirlist          **
30 REM ** von Thomas M.Binzinger **
40 REM ** fuer CPC 464/664/6128 **
50 REM *****
60 GOTO 190:REM Zur Hauptschleife
70 SYMBOL 255,255,0,0,0,0,0,0,0,0:SYMBOL 253,192,192
,192,192,192,192,192,192
80 SYMBOL 254,3,3,3,3,3,3,3,3:SYMBOL 252,0,0,0,0,0
,0,0,255
90 SYMBOL 251,255,0,24,60,60,24,0,0:SYMBOL 247,255
,192,192,192,192,192,192,192
100 SYMBOL 248,255,3,3,3,3,3,3,3:SYMBOL 249,192,19
2,192,192,192,192,255
110 SYMBOL 250,3,3,3,3,3,3,3,255:SYMBOL 244,0,0,25
5,192,192,192,192,255
120 SYMBOL 245,0,0,255,0,0,0,0,255:SYMBOL 246,0,0,
255,3,3,3,3,255
130 bs=CHR$(10)+STRING$(5,8)
140 picon$=CHR$(247)+CHR$(255)+CHR$(251)+CHR$(255)
+CHR$(248)+bs+CHR$(253)+""+CHR$(230)+""+CHR$(254
)+bs+CHR$(249)+CHR$(244)+CHR$(245)+CHR$(246)+CHR$(
250)
150 dicon$=CHR$(247)+STRING$(3,255)+CHR$(215)+bs+C
HR$(253)+""+DAT""+CHR$(254)+bs+CHR$(249)+STRING$(3,25
2)+CHR$(250)
160 ticon$=CHR$(247)+STRING$(3,255)+CHR$(215)+bs+C
HR$(253)+""+CHR$(254)+bs+CHR$(249)+STRING$(3,25
2)+CHR$(250)
170 uicon$=CHR$(247)+STRING$(3,255)+CHR$(248)+bs+C
HR$(253)+"" ? ""+CHR$(254)+bs+CHR$(249)+CHR$(244)+C
HR$(245)+CHR$(246)+CHR$(250)
180 RETURN
190 REM
200 MODE 2:INK 0,13:INK 1,0:BORDER 13

```

eine Kopie davon auf jede Ihrer Disketten schreiben (Dirlist arbeitet nur mit Diskette!), und zwar am besten unter einem Einbuchstabe-Namen, wie 'D'. Hängen Sie jetzt noch einen Zettel der Art 'Computer wird folgendermaßen eingeschaltet...', Diskette so einlegen, RUN, "D" ENTER eingeben... Programm auswählen' an den Computer. Nun weiß jedes, auch noch so technik-ferne Familienmitglied, wie es Programme starten kann. Denn sobald Dirlist gestartet wurde, zeigt es in 'GEM-Manier' das Inhaltsverzeichnis der Diskette an, und das entsprechende Programm kann dann mit einem Joystick oder den Pfeiltasten ausgewählt werden. Dann nur noch ENTER (oder den Knopf am Joystick) drücken, und das Programm läuft. Dirlist sorgt dafür, dass nur BASIC- oder Binär-Files gestartet werden können, um jede Fehlbedienung zu vermeiden. Bilder (also Binär-Files, die in den Bildschirmspeicher geladen werden) werden nicht gestartet, sondern nur angezeigt, wenn sie ausgewählt werden. Dirlist erkennt Bilder an den Extensions. SCR oder .PIC (.PIC-Dateien werden zum Beispiel vom Grafikeditor erzeugt).

## Auch für den Profi

Damit Dirlist auch dem Profi, der sich ja schließlich die Mühe gemacht hat, es einzutippen, etwas nutzt, kann man damit auch Dateien löschen, und zwar wesentlich bequemer als mit 'a\$="... öera,....'. Dazu einfach das entsprechende Symbol anwählen, und 'L' für Löschen drücken. Das Programm fragt noch einmal ab, ob man auch sicher ist, und wenn ja, wird das File gelöscht. Zum Abschluss hier noch einmal alle Datei-Typen, für die Dirlist ein spezielles Symbol hat (handelt es sich um keines davon, wird die Diskette mit dem Fragezeichen angezeigt): BASIC und BIN-Files: Eine normale Diskette, Bilder (.PIC/.SCR): Diskette mit Gesicht, CPM-COM-Files: Diskette mit 'CPM', Dateien (.DAT oder keine Extension): Blatt mit 'DAT', Texte (.TXT): Papier mit Eselsohr, alles andere: Diskette mit Fragezeichen.

```

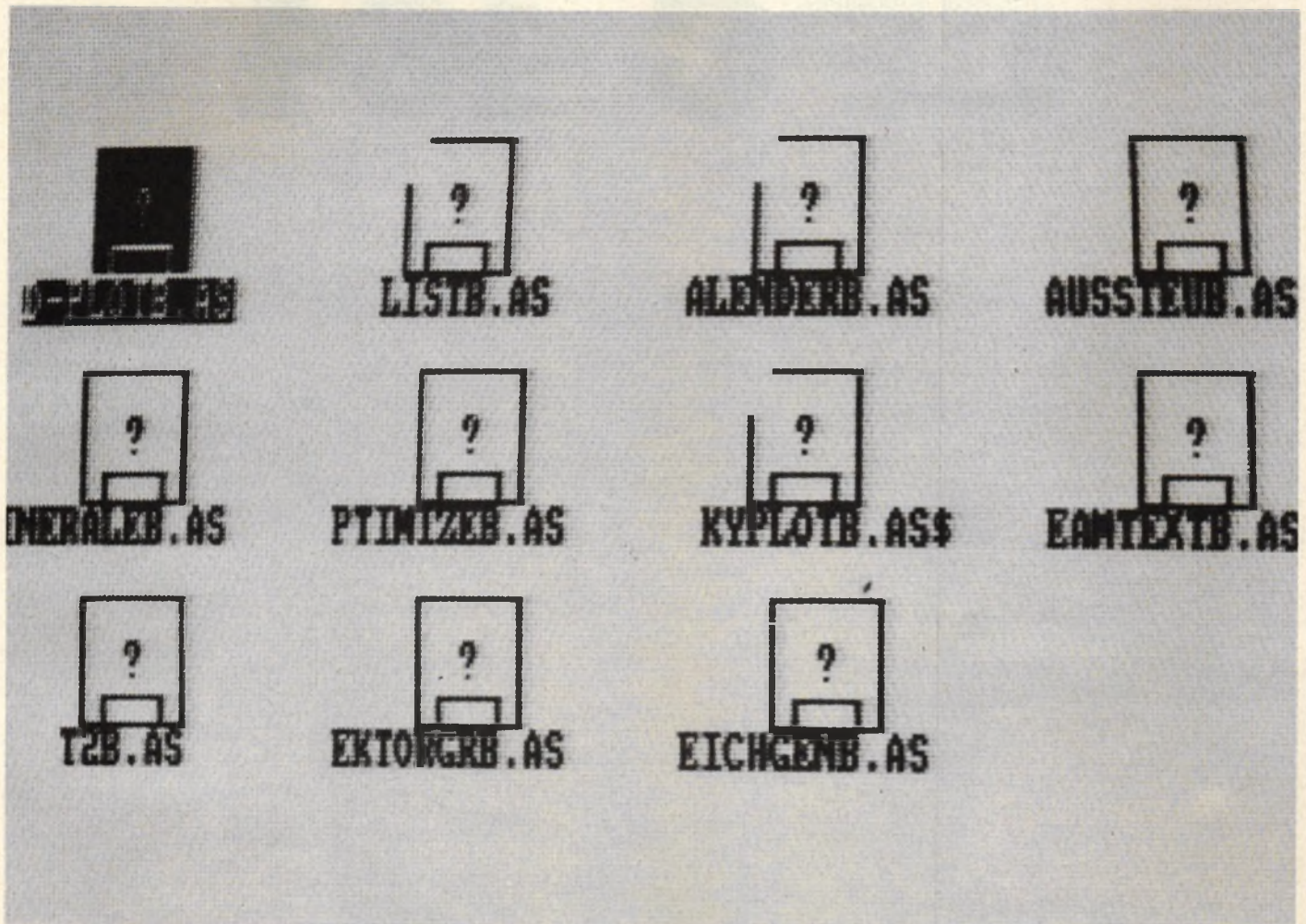
210 OPENOUT"d":MEMORY HIMEM-1:ba=HIMEM+2048:CLOSED
UT:DIM i$(70)
220 WINDOW #2,64,79,2,10:WINDOW #3,1,61,1,25
230 ERASE i$:DIM i$(70):WINDOW #1,1,80,22,25:LOCAT
E 1,21:PRINT SPACES(80);:LOCATE 30,21:PRINT" Lese
Directory...":WINDOW SWAP 0,1:CLS:PRINT CHR$(21);:
CAT:PRINT CHR$(6);:WINDOW SWAP 0,1:GOSUB 770:REM B
egleittext
240 ad=ba:WHILE PEEK(ad)<65 OR PEEK(ad)>90:ad=ad+1
:WEND
250 z=1:p=ad:WHILE PEEK(p)<>0
260 i$(z)="" :FOR x=p TO p+10:i$(z)=i$(z)+CHR$(PEEK
(x)):POKE x,0:NEXT
270 i$(z)=LEFT$(i$(z),8)+". "+RIGHT$(i$(z),3)
280 z=z+1:p=p+14:WEND
290 FOR i=MIN(70,z+1) TO 70:i$(i)="" :NEXT
300 ofs=1
310 GOSUB 70:REM Symbole definieren
320 GOSUB 500:REM Seite anzeigen
330 i=ofs:x=1:y=1
340 PRINT CHR$(24);:GOSUB 570:PRINT CHR$(24);
350 IF INKEY(47)=0 THEN 230:REM Neu anzeugen
360 IF INKEY(36)=0 THEN CLS #2:PRINT #2:PRINT #2,"
SICHER (j/n) ?":b$="" :FOR t=1 TO 100:b$=b$+INKEY$:
NEXT:s$="" :WHILE s$="" :s$=INKEY$:WEND:s$=UPPER$(s$
):IF s$="J" THEN PRINT #2,"Loesche:" :PRINT #2,i$(i
):!ERA.@i$(i):GOTO 230 ELSE 230
370 IF INKEY(18)<>0 AND JOY(0)<>16 THEN 420:REM ke
in Programm ausgewaehlt->320
380 IF i=ofs+19 AND i$(ofs+ep)<>"" THEN ofs=ofs+19
:GOTO 320:REM Next Page
390 REM
400 REM ** Programm starten oder ein Bild (.PIC/.S
CR) anzeigen
410 ex$=RIGHT$(i$(i),3):IF ex$="PIC" OR ex$="SCR"
THEN MODE 1:LOAD i$(i),&C000:WHILE JOY(0)=0 AND IN
KEY(18)<>0:WEND:MODE 2:GOTO 220 ELSE IF ex$="BIN"
OR ex$="BAS" THEN CALL &BBOO:CALL &BB4E:RUN i$(i)
ELSE PRINT CHR$(7);
420 a=JOY(0):IF a=0 AND INKEY(0)<>0 AND INKEY (1)<
>0 AND INKEY(2)<>0 AND INKEY(8)<>0 THEN 350
430 GOSUB 570:p=i:xa=x:ya=y
440 IF ((a AND 1)=1 OR INKEY(0)=0) AND y<>1 THEN y
=y-5:i=i-4
450 IF ((a AND 2)=2 OR INKEY(2)=0) AND y<>21 THEN
y=y+5:i=i+4
460 IF ((a AND 4)=4 OR INKEY(8)=0) AND x<>1 THEN x
=x-16:i=i-1
470 IF ((a AND 8)=8 OR INKEY(1)=0) AND x<>49 THEN
x=x+16:i=i+1
480 IF i$(1)="" THEN i=p:x=xa:y=ya
490 GOTO 340
500 CLS #3:x=1:y=1:i=ofs:ep=0:FOR y=1 TO 25 STEP 5
:FOR x=1 TO 64 STEP 16
510 ep=ep+1:IF ep=20 AND i$(i+1)<>"" THEN GOSUB 74
0:RETURN
520 IF i$(i)="" THEN RETURN
530 GOSUB 570:REM ein Symbol anzeigen
540 i=i+1:NEXT:NEXT:RETURN
550 REM
560 REM ** Drucke je nach ex(tension)$ das entspre
chende Symbol
570 IF i=ofs+19 THEN 730:REM 'Next Page' anstelle
des Prs anzeigen
580 i$=i$(i):IF LEN(i$)=INSTR(i$,".") THEN i$=i$+S
PACES(3)

```

```

590 ex$=RIGHT$(i$,3):LOCATE x+4,y
600 IF ex$="BAS" OR ex$="COM" OR ex$="BIN" THEN PR
INT picon$;:GOTO 640
610 IF ex$="DAT" OR ex$=" " THEN PRINT dicon$;:G
OTO 640
620 IF ex$="TXT" OR ex$="DOC" THEN PRINT ticon$;:G
OTO 640
630 PRINT uicon$;
640 IF ex$="COM" THEN LOCATE x+5,y+1:PRINT"CPM";
650 IF ex$="PIC" OR ex$="SCR" THEN LOCATE x+6,y+1:
PRINT CHR$(224);
660 i$=i$(i)
670 IF INSTR(i$," ")=0 THEN 690 ELSE sp=INSTR(i$,"
")
680 i$=LEFT$(i$,sp-1)+RIGHT$(i$,LEN(i$)-sp):GOTO 6
70
690 i$(i)=i$
700 LOCATE x+6-LEN(i$(i))/2,y+3:PRINT i$(i);

```



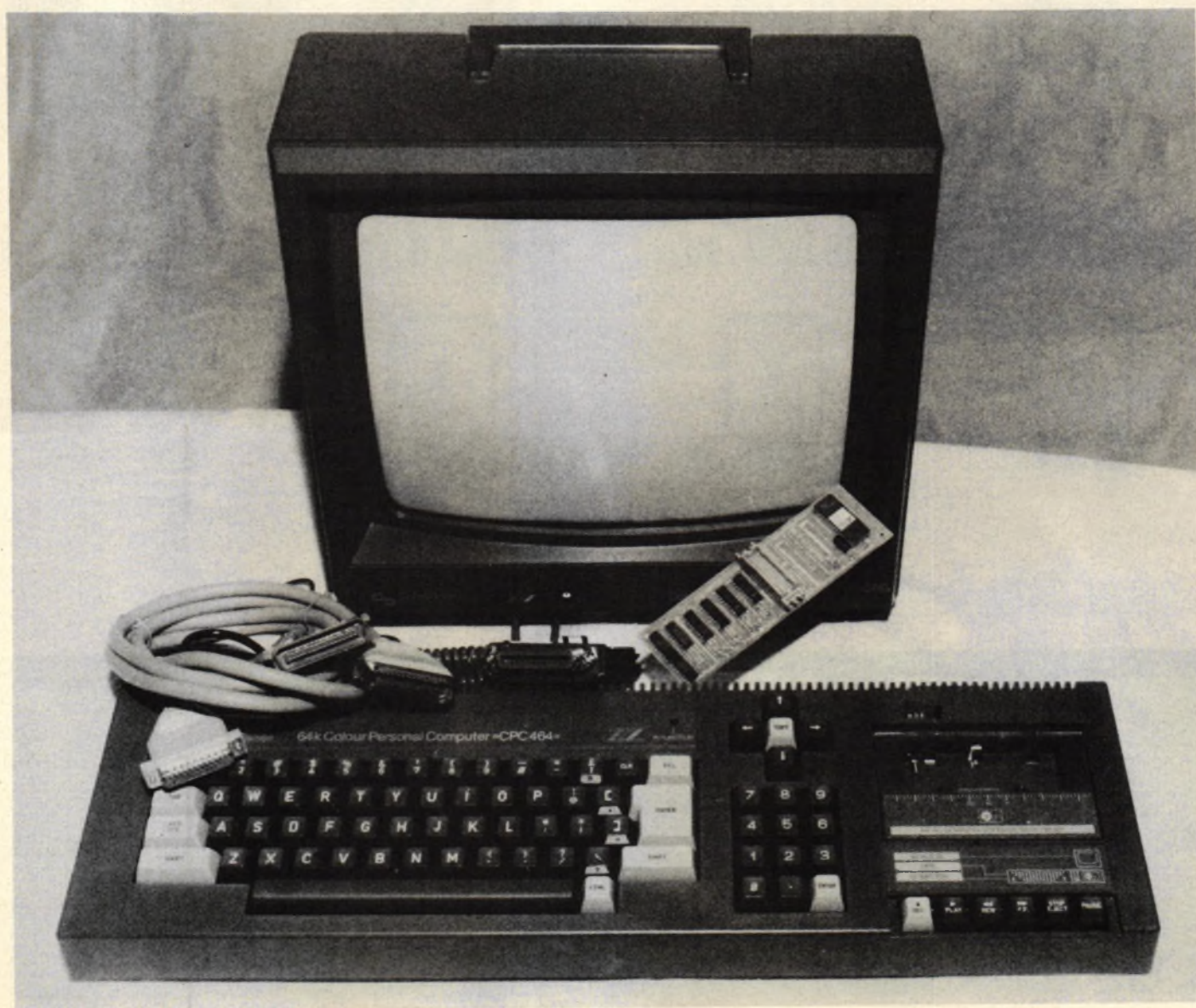
```

710 RETURN
720 REM
730 REM ** Drucke Symbol 'Next Page'
740 LOCATE 53,21:PRINT CHR$(24)+"next "+CHR$(24);:
LOCATE 53,22:PRINT CHR$(24)+"page "+CHR$(24);:LOCA
TE 53,23:PRINT CHR$(24)+ " "+CHR$(154)+CHR$(243)+
"+CHR$(24);:RETURN
750 REM
760 REM ** Zeige Begleittext an rechter Seite an
770 CLS #2:PLOT 499,399,1:DRAW 499,224:DRAW 639,22
4:DRAW 639,399:DRAW 499,399
780 WINDOW SWAP 0,2:FOR x=240 TO 243:PRINT CHR$(x)
" ";:NEXT:PRINT"/ JOY(0)";:PRINT"zum auswaschen":P
RINT:PRINT"ENTER" / FEUER":PRINT"zum starten":PRI
NT:PRINT"L" zum loeschen":PRINT"SPACE": NEU"
790 WINDOW SWAP 0,2:RETURN

```

# Analoge Meßdatenerfassung mit dem

# CPC 464





von Beate Lang

Im vorliegenden Beitrag wird gezeigt, wie mit Hilfe einer einfachen Schaltung bis zu 16 analoge Meßwerte vom Computer erfaßt und verarbeitet werden können. Obgleich die Steuerung des Meßwandlers am Beispiel des CPC 464 gezeigt wird, läßt sie sich mit geringen Änderungen an jedem Home-Computer betreiben.

Im allgemeinen können Computer nur digitale Größen verarbeiten. Es ist daher erforderlich, die analogen Werte zunächst zu digitalisieren.

Dies geschieht mit Hilfe eines Analog/Digital-Wandlers. Die Steuerung des Wandlers sowie die Übergabe der digitalisierten Daten an den Computer erfolgt über eine parallele Ein-Ausgabeschnittstelle.

Das hier vorgestellte Programm arbeitet mit der PIO-Schnittstelle, die den PIO-Baustein 8255 enthält.

Als Analog/Digital-Wandler wird der DAS-952R von Intersil/Datel eingesetzt. Dieser ermöglicht die Digitalisierung von 16 Analogeingängen in 8 Bit-Datenwörter. Bild 1 zeigt die Anschlüsse und den internen Aufbau. Bild 2 zeigt das Impuldiagramm für die wichtigsten Signale dieses Bausteins laut Datenblatt.

Die Auswahl der 16 Eingangskanäle geschieht über die vier Adressleitungen A1 bis A8 und den im Baustein integrierten Adressdekoder. Die an den Adressleitungen anliegende Adresse wird vom Dekoder mit dem nach High gehenden Adress-Enable-Signal (AE) übernommen. In Bild 3 sind die Adressen der jeweiligen Eingänge aufgeführt.

Die Digitalisierung des ausgewählten Kanals beginnt, wenn das Signal „Start of Conversion“ gesetzt wird. Dabei setzt die ansteigende Flanke dieses Signals den 8 Bit-ADC zurück, während die fallende Flanke die eigentliche Digitalisierung startet.

Die Digitalisierung kann durch einen neuen „Start of Conversion“-Impuls unterbrochen und neu gestartet werden.

Nach 64 Clock-Perioden ist die A/D-Umwandlung beendet und der Ausgang „End of Conversion (EOC)“ nimmt High-Pegel an. Der in einem Three-State-Ausgabepuffer zwischengespeicherte digitale Wert kann gelesen werden,

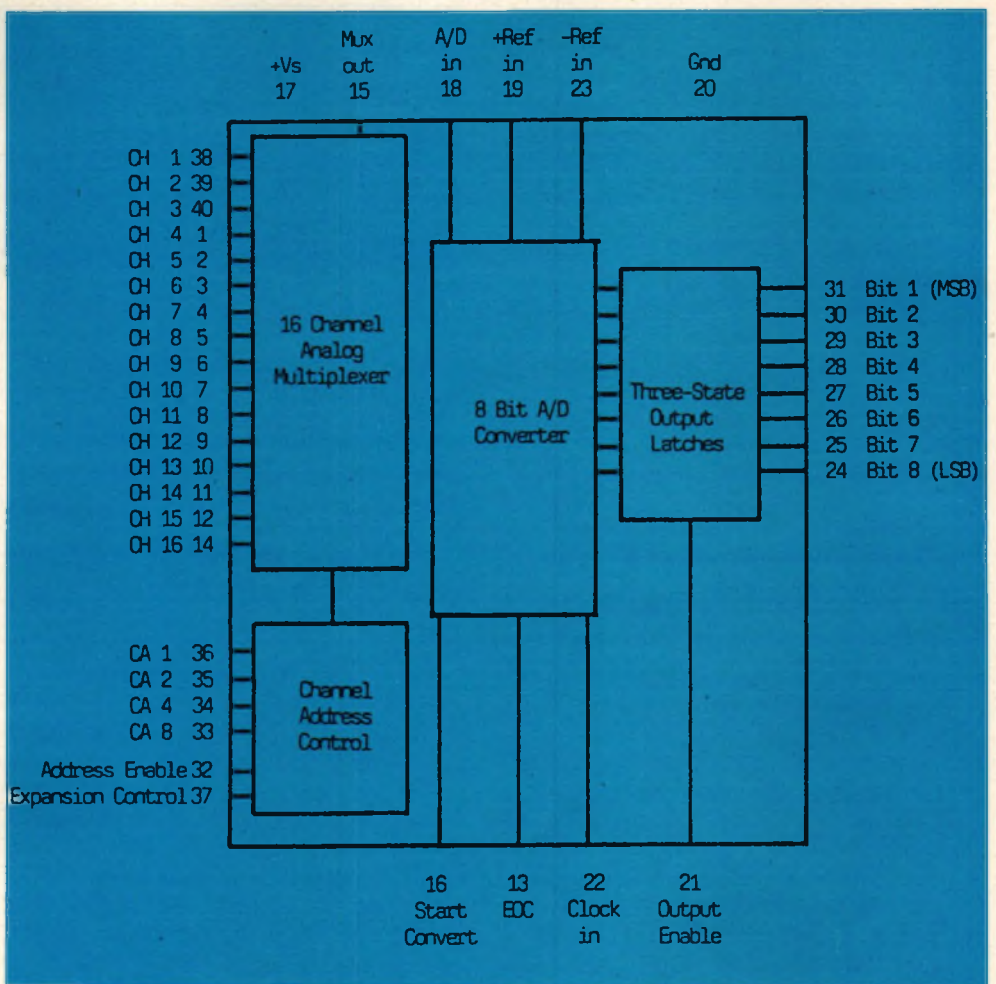


Bild 1 Aufbau des DAS-952R

wenn das Signal „Output Enable (OE)“ auf High-Pegel gesetzt wird.

Bild 4 zeigt die komplette Schaltung des A/D-Wandlers und den Anschluß an die PIO-Schnittstelle.

In der gezeigten Schaltung wurde OE fest auf +5V gelegt, wodurch das Ergebnis der Umwandlung stets an den Ausgängen D1-D8 anliegt, bis es durch einen neuen Wert überschrieben wird. Die Zuordnung zwischen digitalem Wert und analoger Meßgröße erfolgt durch die Spannung an den Referenzeingängen des A/D-Wandlers. In der vorliegenden Schaltung liegt -Ref In an GND und +Ref In an +5V. Dadurch erzeugt eine Eingangsspannung von 0V das Datenwort 00 und eine Eingangsspannung von +5V das Datenwort &FF. Damit ergibt sich bei einem 8 Bit breiten Datenwort eine Auflösung von  $5V/255 = 20mV/Bit$ . Durch eine Trennung des Ref In-Eingangs von der +5V-Versorgung und eine Stabilisierung auf z.B. +2V wird eine andere Auflösung, z.B.  $8mV/Bit$ , erreicht.

Zur Spannungsstabilisierung werden die Stabilisatoren VR-182C, zur Erzeugung des Taktsignals der SG-10/153.6K, beide von Intersil/Datel, verwendet.

Die Schaltung wurde für eine Versorgungsspannung von 9-18V ausgelegt. Für weitere Informationen sei auf das Datenblatt der Herstellerfirma General Elektrik, Bereich Datel, Bavariaring 8/1, Postfach 150826, 8000 München 15, verwiesen.

Der Schnittstellenbaustein 8255 wird in der kombinierten Betriebsart 0/1 betrieben.

Die Betriebsart 0 ermöglicht eine einfache Ein- oder Ausgabe ohne Quittungssignale, in der Betriebsart 1 wird die Ein- bzw. Ausgabe durch Quittungssignale gesteuert.

Bild 5 zeigt die Funktion und Arbeitsweise der einzelnen Schnittstellen-Ports.

Das Port A (A0-A3) wird zur Kanaladressierung des A/D-Wandlers verwendet, über das Port B (B0-B7) werden die digitalisierten Daten vom ADC in den

Computer eingelesen, das Port C dient zur Steuerung des ADC und der Schnittstelle. Über C7 wird der „Adress-Enable“-Impuls und über C6 der „Start Conversion“-Impuls an den ADC gesendet. Auf C2 wird das invertierte EOC-Signal gelegt, das als Strobe-Signal für den Port B der Schnittstelle dient.

Die freien Portleitungen C3 bis C5 sind als Ausgänge programmiert und könnten für Steuerzwecke (z.B. Schrittmotoren in x-, y-, z-Richtung) verwendet werden.

## BASIC-Programm

Das folgende BASIC-Programm initialisiert die Schnittstelle und steuert den Programmablauf.

Es ermöglicht die Digitalisierung der an den Analogeingängen anliegenden Spannungen in beliebiger Reihenfolge und mit beliebigen Zeitverzögerungen. Weiterhin können die Eichfaktoren, mit denen die digitalen Werte multipliziert werden sollen, eingegeben werden.

Während des Digitalisierungsvorgangs werden der momentan aktive Analogeingang und der ungeeichte digitale Wert am Bildschirm ausgegeben.

Ferner wird die Zeit seit Start des Meßprogramms angezeigt. Nach beendeter Messung können für jeden Analogeingang die geicheten Meßdaten in Tabellenform dreispaltig auf dem Bildschirm oder Drucker ausgedruckt und auf Diskette oder Kassette abgespeichert (oder geladen) werden.

Die Steuerung des A/D-Wandlers erfolgt über ein Maschinenprogramm, das mit einem Call-Befehl aus dem BASIC-Programm aufgerufen wird (Zeile 1080).

Dieser Call-Befehl enthält zwei Parameter:

Der erste übergibt an das Maschinenprogramm die Speicheradresse der Variablen  $k(i)$ , welche die Kanalnummer des aktuellen Analogeingangs enthält, der zweite umfaßt die Adresse der Variablen  $x(j,i)$ . In diese Variable speichert das Maschinenprogramm den digitalisierten Wert ab und übergibt so diesen Wert an das BASIC-Programm.

# Programmbeschreibung

Um das Programm möglichst anwenderfreundlich zu gestalten, wurde es vollständig in Menütechnik aufgebaut. Der Bildschirm wird dabei während des Programmablaufs in drei Bereiche eingeteilt.

Der untere Teil zeigt das Hauptmenü, über das die einzelnen Programmpunkte ausgewählt werden können.

Im oberen Teil werden der momentan angesprochene Programmpunkt sowie das dazugehörige Untermenü angezeigt.

Der mittlere Bildschirmbereich dient zur Programmausführung. Nach dem Ausdruck „Bitte wählen Sie...“ ist es möglich, entweder einen Punkt des Untermenüs oder aber einen neuen Programmpunkt aus dem Hauptmenü durch Eingabe

des angegebenen Zahlenwertes bzw. Buchstabens zu wählen.

## 1. Programmpunkt: I-INITIALISIERUNG

Unter diesem Programmpunkt können alle für die Messung erforderlichen Daten eingegeben oder geändert werden.

### 1-Kanalfolge

In diesem Unterprogramm wird die Reihenfolge festgelegt, mit der die gewünschten Kanäle des ADC abgefragt werden sollen. Es sind max. 16 Werte möglich, wobei in der Kanalfolge ein Kanal auch mehrmals abgefragt werden kann.

Sollen z.B. die ersten drei Kanäle und der letzte Kanal des ADC in der Reihenfolge 1, 2, 16, 3, 2 eingelesen und digitalisiert werden, dann muß als Kanalfolge '1, 2, 16, 3, 2 ENTER' eingegeben werden.

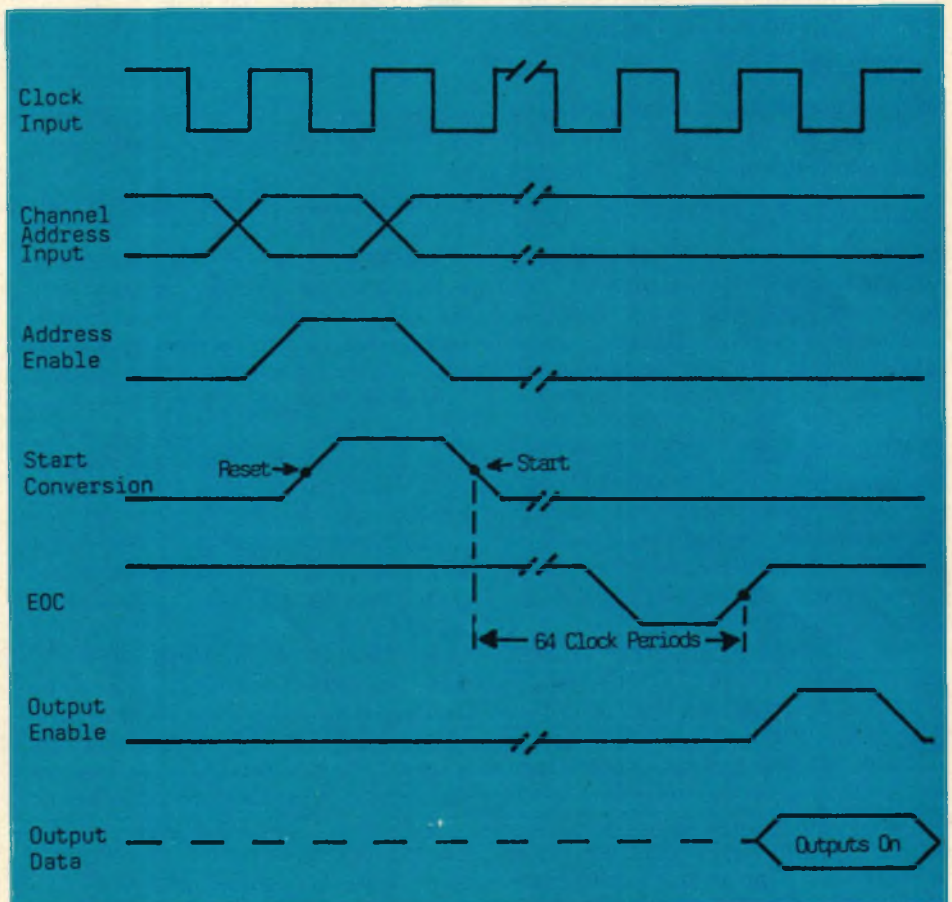


Bild 2 Impulsdiagramm des DAS-952R

KANALADRESSE				KANALNUMMER
A8	A4	A2	A1	
0	0	0	0	1
0	0	0	1	2
0	0	1	0	3
0	0	1	1	4
0	1	0	0	5
0	1	0	1	6
0	1	1	0	7
0	1	1	1	8
1	0	0	0	9
1	0	0	1	10
1	0	1	0	11
1	0	1	1	12
1	1	0	0	13
1	1	0	1	14
1	1	1	0	15
1	1	1	1	16

Bild 3 Kanaladressen des DAS-952R

Solange die Eingabezeile noch nicht mit ENTER abgeschlossen wurde, kann sie mit den Cursor-, CLS- und DEL-Tasten korrigiert werden.

2-Zeitfolge

Dieses Unterprogramm dient zur Festlegung der zeitlichen Verzögerungen zwischen den einzelnen Kanalabfragen. Die Verzögerungen werden in Sekunden eingegeben. Für die Kanalfolge 1, 2, 16, 3, 2 bedeutet z.B. die Eingabe '0, 1, 10 ENTER', daß der Kanal 1 sofort, der Kanal 2 eine Sekunde nach dem Kanal 1, der Kanal 16 zehn Sekunden nach dem Kanal 2 und die Kanäle 3 und 2 ohne Verzögerung vom ADC gelesen und digitalisiert werden.

3-Eichung

In diesem Unterprogramm können die Eichfaktoren eingegeben werden, mit denen die digitalisierten Werte der einzelnen Kanäle multipliziert werden sollen.

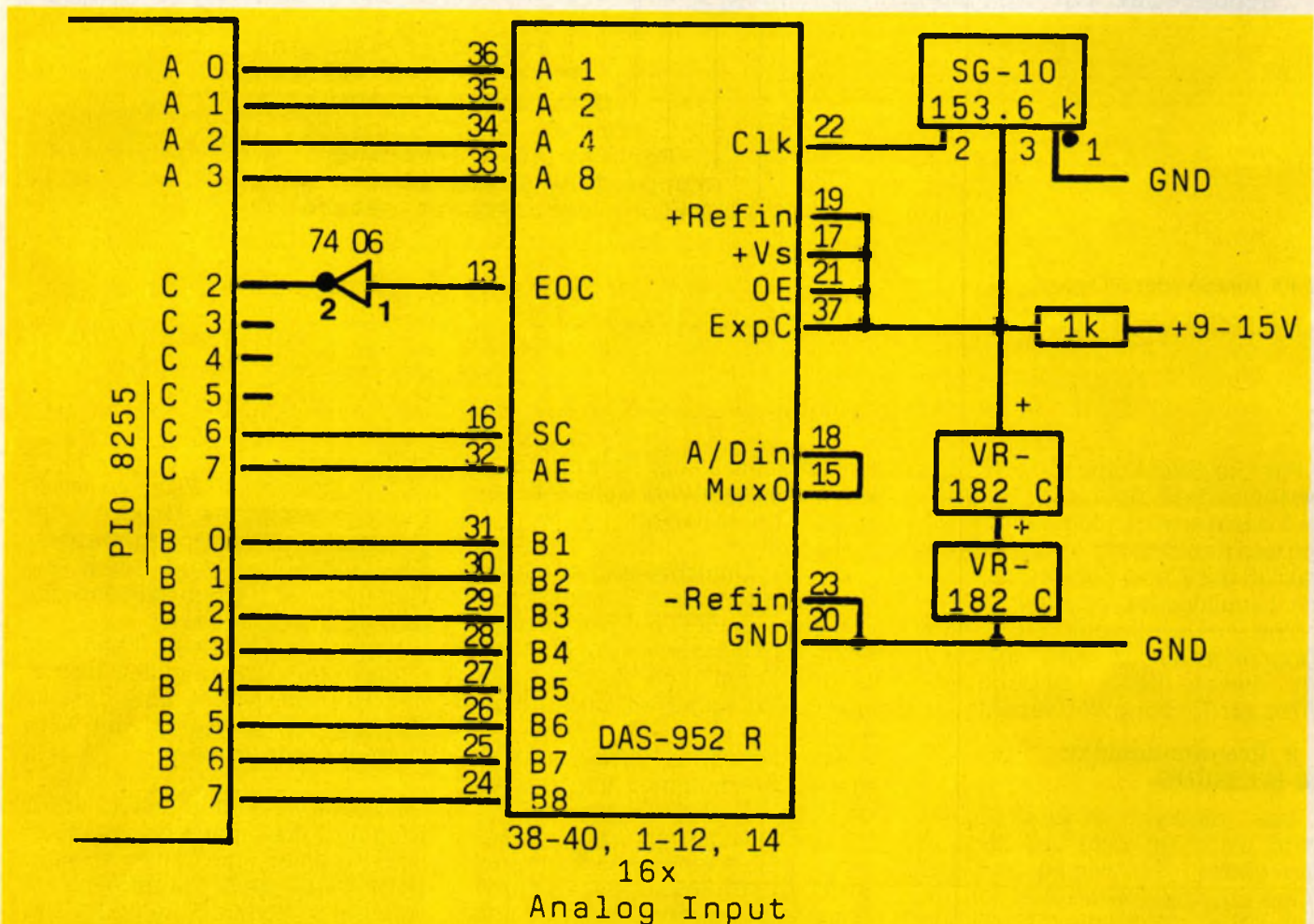
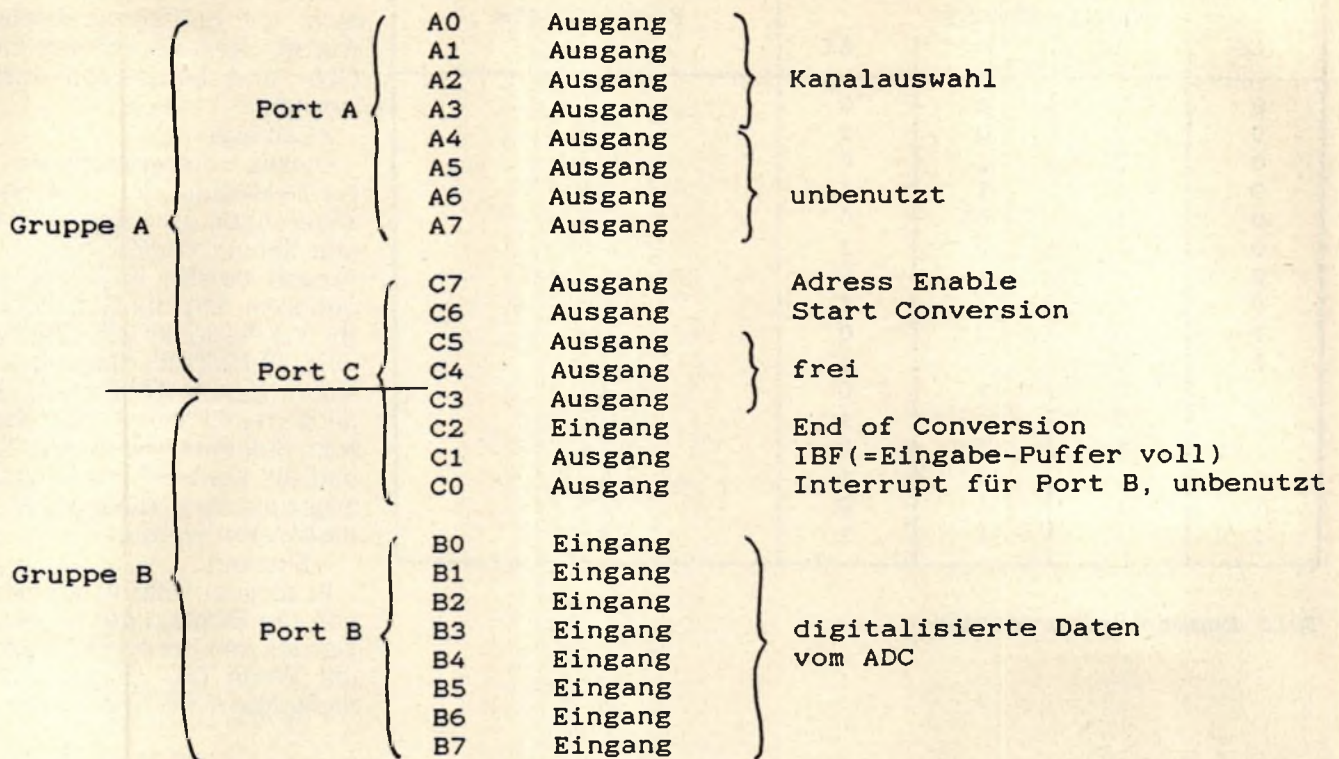


Bild 4 Schaltung des ADC und Anschluß an die Schnittstelle



Steuerwort für den 8255 :

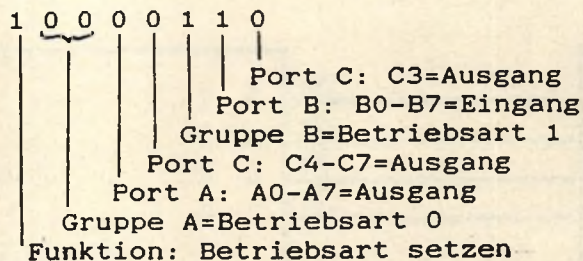


Bild 5 Funktion der PIO-Ports

Für die Kanalfolge 1, 2, 16, 3, 2 bedeutet die Eingabe 0,1, 10, 2 z.B., daß die Werte des Kanals 1 mit 0,1, die des Kanals 2 mit 10, die des Kanals 16 mit 2 und die des Kanals 3 mit 1 multipliziert werden.

Tritt ein Kanal mehrmals in einer Kanalfolge auf, so wird der erste für diesen Kanal eingegebene Wert zur Eichung verwendet.

## 2. Programmpunkt: M-MESSUNG

Bei diesem Programmpunkt wird nach der Zahl der Messungen gefragt, d.h., wie oft die Kanalfolge durchlaufen werden soll, die ungefähre Meßdauer berechnet und ausgedruckt und alte Messungen gelöscht.

Weiterhin wird die maximal mögliche Anzahl der Messungen

am Bildschirm ausgegeben. Sie ist von der Anzahl der Kanäle in der Kanalfolge abhängig.

### 1-Start

Dieses Unterprogramm startet das Meßprogramm. Zunächst wird die Uhr des CPC auf Null gestellt. Dann werden die einzelnen Kanäle der Kanal- und Zeitfolge entsprechend abgefragt und digitalisiert.

Am Bildschirm werden die aktuelle Kanalnummer und der ungeeichte Wert ausgegeben.

Die Digitalisierung eines Kanals mit dem Bildschirmausdruck dauert etwa 0,07 Sekunden. Wenn die Kanäle in kürzeren Abständen abgefragt werden sollen, dann muß der Print-Befehl in Programmzeile 1090 weggelassen werden, der ca. 0,03 Sekunden dauert.

### ESC-Stop

Der Meßablauf kann jederzeit durch zweimaliges Drücken der ESC-Taste unterbrochen werden. Am Bildschirm wird dann die Nummer der letzten vollständigen Messung ausgegeben.

Nach einem Stop kann mit '1'(Start) die Messung neu begonnen werden. Dabei wird die alte Eingabe für die Anzahl der Messungen übernommen.

Es ist aber auch möglich, durch Eingabe des angegebenen Buchstabens einen anderen Programmpunkt aus dem Hauptmenü zu wählen, so kann z.B. nach der Eingabe 'M' (Messung) die Anzahl der Messungen geändert und dann die Messung neu gestartet werden.

Zu beachten ist, daß nach dem Start einer neuen Messung oder nach dem Ändern der Meßanzahl alle alten Meßwerte gelöscht werden (evtl. vorher abspeichern).

### 3. Programmpunkt: A-AUSDRUCK

Dieser Programmpunkt dient zur tabellarischen Ausgabe der Meßwerte auf dem Bildschirm oder Drucker.

1-Bildschirm 2-Drucker

Nach der Eingabe der gewünschten Kanal-Nummer werden die zugehörigen geeichten Meßwerte dreispaltig ausgedruckt.

Kommt der Kanal in der Kanalfolge mehrmals vor, dann werden alle Messungen dieses Kanals zeitlich geordnet in der Tabelle ausgegeben.

### 4. Programmpunkt: S-SAVE/LOAD

Unter diesem Programmpunkt können die Meßwerte auf Diskette/Kassette unter einem frei zu wählenden Namen abgespeichert oder geladen werden.

1-Speichern 2-Laden

Abgespeichert werden folgende Daten:

Die Zahl der Messungen, m, die Zahl der Kanäle in der Kanalfolge, n, die Kanalfolge k(n), die Zeitfolge z(n), die Eichfaktoren f(n), die Meßwerte x(n,m) und ihre zeitliche Zuordnung t(n,m).

### 5. Programmpunkt: E-PROGRAMMENDE

Unter diesem Programmpunkt kann das Programm durch Drücken einer Taste außer i, m, a, s, e beendet werden. Über die Tasten i, m, a, s, e wird dem Hauptmenü entsprechend ein anderer Programmpunkt angewählt.

```

10 REM *****
20 REM * Steuerprogramm fuer den A/D-Wandler DAS-952R *
30 REM * copyright 1985 by Beate Lang *
40 REM *****
50 REM
60 MEMORY &A5FF:DEFINT a-z:DEFREAL f,t
70 MODE 2:BORDER 0:INK 0,0:INK 1,25
80 DIM x(3,50),t(3,50),f(15),k(15),z(15),g(15)
90 FOR i=0 TO 15:f(i)=1:NEXT:n=-1
100 s=0:FOR i=&A600 TO &A64C:READ a:POKE i,a:s=s+a:NEXT
110 IF s<> 10924 THEN PRINT"Fehler in den Data-Zeilen!":END
120 CALL &A600:'----- *** INITIALISIERUNG PIO ***
130 ON BREAK GOSUB 490
140 WINDOW#0,1,80,3,23:WINDOW#1,1,70,1,1
150 WINDOW#2,1,80,25,25:WINDOW#3,71,80,1,1
160 CLS:CLS#1:CLS#2:EVERY 50,0 GOSUB 230
170 MOVE 0,376:DRAWR 639,0:MOVE 0,20:DRAWR 639,0
180 PRINT#2," I-Initialisierung";SPC(6);"M-Messung";
190 PRINT#2,SPC(6);"A-Ausdruck";SPC(6)"S-Save/Load";SPC(6);"E-Ende"
200 GOSUB 260:IF l=0 THEN 200
210 ON l GOTO 510,810,1120,1420,1720
220 REM ----- *** SEKUNDENANZEIGE ***
230 PRINT#3,USING "#####";TIME/300
240 RETURN
250 REM ----- *** TASTATURABFRAGE ***
260 PRINT:PRINT"Bitte waehlen Sie..."
270 in$=INKEY$:IF in$="" THEN 270
280 l=0:in$=UPPER$(in$)
290 IF in$="I" THEN l=1:RETURN
300 IF in$="M" THEN l=2:RETURN
310 IF in$="A" THEN l=3:RETURN
320 IF in$="S" THEN l=4:RETURN
330 IF in$="E" THEN l=5
340 RETURN
350 REM ----- *** EINGABE-ZERLEGUNG ***
360 s=0:LINE INPUT in$:sm=LEN(in$)
370 e$="":WHILE s<sm:s=s+1:w$=MID$(in$,s,1):IF w$="," THEN RETURN
380 e$=e$+w$:WEND
390 RETURN
400 REM ----- *** PARAMETER-AUSDRUCK ***
410 CLS:PRINT"Kanalfolge: ";:FOR i=0 TO n:PRINT STR$(k(i));",,";:NEXT
420 PRINT CHR$(8);CHR$(16)

```

```
430 PRINT"Zeitfolge : ";:FOR i=0 TO n:PRINT STR$(z(i)/50);",",":NEXT
440 PRINT CHR$(8);CHR$(16)
450 PRINT"Eichung : ";:FOR i=0 TO n:PRINT STR$(f(i));",",":NEXT
460 PRINT CHR$(8);CHR$(16):PRINT
470 RETURN
480 REM ----- *** BREAK ***
490 esc=-1:RETURN
500 REM ----- *** INITIALISIERUNG ***
510 CLS#1:PRINT#1,CHR$(24);" I N I T I A L I S I E R U N G ";CHR$(24);
520 PRINT#1,SPC(2);"1-Kanalfolge";SPC(2);"2-Zeitfolge";SPC(2);"3-Eichung"
530 CLS:IF n>-1 THEN GOSUB 410
540 GOSUB 260:IF l THEN 210
550 IF in$="1" THEN GOSUB 600:GOTO 530
560 IF in$="2" THEN GOSUB 670:GOTO 530
570 IF in$="3" THEN GOSUB 740
580 GOTO 530
590 REM ----- *** KANALFOLGE ***
600 CLS:PRINT CHR$(24);"Kanalfolge";CHR$(24)
610 PRINT:PRINT"Bitte die Kanalfolge eingeben (1...16,durch Komma
620 GOSUB 360 getrennt):"
630 FOR n=0 TO 15:k(n)=VAL(e$):IF k(n)<1 OR k(n)>16 THEN 610
640 IF s=sm THEN RETURN ELSE GOSUB 370:NEXT
650 n=n-1:RETURN
660 REM ----- *** ZEITFOLGE ***
670 PRINT CHR$(24);"Zeitfolge";CHR$(24)
680 IF n=-1 THEN GOSUB 610:GOSUB 410
690 PRINT:PRINT"Bitte Wartezeiten fuer die Kanale eingeben (in [s],
durch Komma getrennt):"
700 GOSUB 360
710 FOR i=0 TO n:z(i)=50*VAL(e$):IF s=sm THEN FOR j=i+1 TO n:z(j)=0:NEXT
ELSE GO SUB 370:NEXT
720 RETURN
730 REM ----- *** EICHUNG ***
740 PRINT CHR$(24);"Eichung";CHR$(24)
750 IF n=-1 THEN GOSUB 610:GOSUB 410
760 PRINT:PRINT"Bitte die Eichfaktoren der Kanale - durch Kommas
getrennt - eingeben:"
770 GOSUB 360
780 FOR i=0 TO n:f(i)=VAL(e$):IF s=sm THEN FOR j=i+1 TO n:f(j)=1:NEXT
ELSE GOSUB 370:NEXT
790 RETURN
800 REM ----- *** MESSUNG ***
810 CLS#1:PRINT#1,CHR$(24);" M E S S U N G ";CHR$(24);
820 IF n=-1 THEN GOSUB 610
830 CLS:GOSUB 410:a=FIX(4000/(n+1))
840 PRINT:PRINT"Wieviele Messungen sollen gemacht werden
( 1 -";a;")";:INPUT m
850 IF m<1 OR m>a THEN 840 ELSE m=m-1
860 ERASE x,t:DIM x(n,m),t(n,m)
870 t=0:FOR i=0 TO n:t=t+z(i)/50:NEXT
880 t=(m+1)*(t+(n+1)*0.07)
890 PRINT"gesamte Messzeit ca.";ROUND(t,0);"s"
900 PRINT#1,SPC(15);"1-Start";SPC(8);"ESC-Stop":esc=0
910 GOSUB 270:IF l AND esc THEN esc=0:m=j:GOTO 210
920 IF l THEN 210
930 IF in$="1" THEN GOSUB 960:GOTO 910
940 GOTO 910
950 REM ----- *** START/STOP ***
960 CLS:esc=0:FOR i=0 TO 3:POKE &B187+i,0:NEXT
970 FOR j=0 TO m
```

```

980 FOR i=0 TO n
990 IF esc THEN 1030
1000 IF z(i) THEN AFTER z(i),3 GOSUB 1080 ELSE GOSUB 1080
1010 IF l=0 THEN 1010
1020 l=0:NEXT:NEXT:PRINT:PRINT"*** Messung beendet ***":GOTO 1060
1030 PRINT:PRINT"Programm gestoppt";
1040 IF j THEN PRINT" nach Messung Nr.";j
1050 j=j-1
1060 PRINT:PRINT"Bitte waehlen Sie...":RETURN
1070 REM ----- *** DIGITALISIERUNG ***
1080 CALL &A608,@k(i),@x(i,j)
1090 t(i,j)=TIME/300:PRINT USING "####";k(i);x(i,j),
1100 l=1:RETURN
1110 REM ----- *** AUSDRUCK ***
1120 CLS#1:PRINT#1,CHR$(24);" A U S D R U C K ";CHR$(24);
1130 PRINT#1,SPC(10);"1-Bildschirm";SPC(5);"2-Drucker":CLS
1140 IF m=-1 THEN PRINT"Keine Messwerte vorhanden!"
1150 GOSUB 260:IF l THEN 210
1160 IF in$="1" THEN l=1:w$="BILDSCHIRM ":w=0:a=16:GOSUB 1200:GOTO 1150
1170 IF in$="2" THEN l=1:w$="DRUCKER ":w=8:a=s*(m+1)/3+0.4:GOSUB 1200
1180 GOTO 1150
1190 REM ----- *** TABELLEN=AUSDRUCK ***
1200 CLS:PRINT CHR$(24);" AUSDRUCK ";w$;CHR$(24)
1210 PRINT"Moegliche Kanalnummern:";:FOR i=0 TO n:PRINT k
(i);", ";:NEXT:PRINT CHR$(8);CHR$(16)
1220 PRINT:INPUT"Ausdruck fuer Kanal-Nr.";in
1230 s=0:FOR i=0 TO n:IF k(i)=in THEN g(s)=i:s=s+1
1240 NEXT:IF s=0 THEN 1210
1250 e$="#.#":b=LEN(STR$(FIX(t(g(s-1),m))))-1:FOR j=1 TO b:e$="#" +e$
:NEXT:b=8-b
1260 c=FIX(s*m/a):FOR i=0 TO c STEP 3
1270 CLS:PRINT#w,"Kanal-Nr.:";k(g(0));SPC(45);"Eichfaktor=";
f(g(0)):PRINT#w
1280 FOR l=i TO i+2:IF l>c THEN 1300
1290 PRINT#w," NR. | WERT | ZEIT(s) ";:NEXT
1300 PRINT#w,CHR$(13);:PRINT CHR$(22);CHR$(1);
1310 FOR l=i TO i+2:IF l>c THEN 1330
1320 PRINT#w," ";STRING$(23,"_");" ";:NEXT
1330 PRINT CHR$(22);CHR$(0);:PRINT#w
1340 FOR j=1 TO a:FOR l=i TO i+2
1350 e=l*a+j-1:x=e MOD s:y=FIX(e/s):IF y>m THEN 1380
1360 PRINT#w,USING "####";e+1;:PRINT#w,"|";USING "##.##^";
x(g(x),y)*f(g(x));
1370 PRINT#w,"|";USING e$;t(g(x),y);:PRINT#w,SPC(b);
1380 NEXT
1390 PRINT#w:NEXT:IF l-1<c THEN PRINT:PRINT"Bitte Taste druecken !":CALL
&BB06
1400 NEXT:RETURN
1410 REM ----- *** SAVE/LOAD ***
1420 CLS#1:PRINT#1,CHR$(24);" S A V E / L O A D ";CHR$(24);
1430 PRINT#1,SPC(10);"1-Speichern";SPC(5);"2-Laden":CLS
1440 GOSUB 260:IF l THEN 210
1450 IF in$="1" THEN GOSUB 1490:GOTO 1440
1460 IF in$="2" THEN GOSUB 1600
1470 GOTO 1440
1480 REM ----- *** SPEICHERN ***
1490 CLS:PRINT CHR$(24);" Speichern ";CHR$(24):PRINT
1500 INPUT "Name der Datei (max. 8 Buchstaben)";in$:IF LEN(in$)>8 OR
in$="" THEN 1500

```

```

1510 in$=in$+".dat":PRINT:PRINT"Bitte warten..."
1520 OPENOUT in$
1530 PRINT#9,m,n
1540 FOR i=0 TO n:PRINT#9,k(i),z(i),f(i):NEXT
1550 FOR i=0 TO n:FOR j=0 TO m:PRINT#9,x(i,j),t(i,j):NEXT:NEXT
1560 CLOSEOUT
1570 PRINT"Die Daten wurden unter dem Namen ' ";in$;" ' gespeichert."
1580 RETURN
1590 REM ----- *** LADEN ***
1600 CLS:PRINT CHR$(24);" Laden ";CHR$(24):PRINT
1610 INPUT "Name der Datei";in$:IF LEN(in$)>8 OR in$="" THEN 1610
1620 in$=in$+".dat":PRINT:PRINT"Bitte warten..."
1630 OPENIN in$
1640 INPUT #9,m,n
1650 ERASE x,t:DIM x(n,m),t(n,m)
1660 FOR i=0 TO n:INPUT#9,k(i),z(i),f(i):NEXT
1670 FOR i=0 TO n:FOR j=0 TO m:INPUT#9,x(i,j),t(i,j):NEXT:NEXT
1680 CLOSEIN
1690 PRINT"Es wurde die Datei ' ";in$;" ' geladen."
1700 RETURN
1710 REM ----- *** PROGRAMMENDE ***
1720 CLS#1:PRINT#1,CHR$(24);" E N D E ";CHR$(24);
1730 CLS:PRINT"Zum Beenden eine Taste (ausser i,m,a,s,e) druecken!"
1740 GOSUB 270:IF 1 THEN 210
1750 END
1760 REM      ***              MASCHINENPROGRAMM ZUR              ***
1770 REM      *** INITIALISIERUNG PIO UND STEUERUNG A/D-WANDLER ***
1780 DATA &3E,&86,&1,&F3,&F8,&ED,&79,&C9,&CD,&1B,&A6,&CD,&37,&A6,&DC,&3F,&A
1790 DATA &D8,&CD,&9,&BB,&FE,&FC,&C8,&C3,&B,&A6,&DD,&66,&3,&DD,&6E,&2,&7E
1800 DATA &3D,&1,&F0,&F8,&ED,&79,&3E,&80,&1,&F2,&F8,&ED,&79,&3E,&40,&ED,&79
1810 DATA &97,&ED,&79,&C9,&ED,&78,&1F,&1F,&CD,&19,&BD,&C9,&1,&F1,&F8,&ED,&7
1820 DATA &DD,&56,&1,&DD,&5E,&0,&12,&37,&C9

```

Liste der wichtigsten Variablen:

a	Zeilenzahl bei Tabellenausdruck
e\$	Teil einer Eingabe
esc	Unterbrechungsflag
f(i)	Eichfaktor
g(i)	Wiederholung des Kanals in der Kanalfolge
i,j	Laufvariablen
in,in\$	Eingabe
k(i)	Kanal
l	Flag,Laufvariable
m	m+1=Anzahl der Messungen
n	n+1=Anzahl der Kanäle in der Kanalfolge
t(i,j)	Zeit
w	Window
x(i,j)	digitalisierter Wert
z(i)	Verzögerung



## ASSEMBLERLISTING ZUM MASCHINENPROGRAMM (BASICPROGR. ZEILE 1780-1820)

```

A600          20          ORG  &a600
A600          30          ;----- Analog/Digitalumwandlung
A600          40          ;----- copyright 1986 by B.Lang
A600          50          ;----- Initialisierung PIO: CALL &A600
A600          60          ;----- Befehl CALL &A608,@KANAL,@WERT
A600          70          ;
A600          80          ;----- Initialisierung PIO
A600 3E86     90  INIT  LD   a,&x10000110 ; PIO Steuerwort
A602          100         ; Port A = Ausgabe,Betriebsart 0
A602          110         ; Port B = Eingabe,Betriebsart 1
A602          120         ; Port C = C0-C2 Quittungssignale
A602          130         ; C3-C7 Ausgang
A602 01F3F8  140         LD   bc,&f8f3 ; PIO Steuerregister
A605 ED79    150         OUT  (c),a ; Steuerwort ausgeben
A607 C9      160         RET
A608          170         ;----- Steuerung ADC
A608 CD0000  180         CALL address
A60B CD0000  190  S1    CALL endcon
A60E DC0000  200         CALL c,input
A611 D8      210         RET  c ; wenn A/D-Umwandlung beendet,
A612          220         ; dann Ruecksprung
A612 CD09BB  230         CALL &bb09 ; Tastaturabfrage
A615 FEFC    240         CP   &fc ; Break-Taste gedrueckt?
A617 C8      250         RET  z ; wenn ja, dann Ruecksprung
A618 C30BA6  260         JP   s1
A61B          270         ;----- Kanalauswahl A/D
**** Zeile 180 : ADRESS=&A61B
A61B DD6603  280  ADRESS LD  h,(ix+3) ; Adresse der Variablen Kanal
A61E DD6E02  290         LD  l,(ix+2) ; nach hl laden
A621 7E      300         LD  a,(hl) ; Kanalnr. laden
A622 3D      310         DEC  a
A623 01F0F8  320         LD  bc,&f8f0 ; PIO Port a adressieren
A626 ED79    330         OUT  (c),a ; Kanal ausgeben
A628          340         ;----- Adress-Enable A/D
A628 3E80    350  ADREN LD  a,&x10000000 ; Adress-Enable A/D=high setzen
A62A 01F2F8  360         LD  bc,&f8f2 ; PIO Port c adressieren
A62D ED79    370         OUT  (c),a ; Adress-Enable ausgeben
A62F          380         ;----- Start Conversion A/D
A62F 3E40    390  STCON LD  a,&x01000000 ; Adress-Enable A/D=low,
A631          400         ; Start conv. A/D=high setzen
A631 ED79    410         OUT  (c),a
A633 97      420         SUB  a ; Start conv. A/D=low
A634 ED79    430         OUT  (c),a
A636 C9      440         RET
A637          450         ;----- End of Conversion A/D
**** Zeile 190 : ENDCON=&A637
A637 ED78    460  ENDCON IN  a,(c) ; Port C einlesen
A639 1F      470         RRA
A63A 1F      480         RRA ; Bit 1 (IBF PIO) ins Carry
A63B CD19BD  490         CALL &bd19 ; Verzoeigerung, damit ADC-Daten stabil
A63E C9      500         RET
A63F          510         ;----- Einlesen in Prozessor
**** Zeile 200 : INPUT=&A63F
A63F 01F1F8  520  INPUT LD  bc,&f8f1 ; PIO Port b adressieren
A642 ED78    530         IN  a,(c) ; Inhalt lesen
A644 DD5601  540         LD  d,(ix+1) ; Adresse der Variablen x(i,j)
A647 DD5E00  550         LD  e,(ix+0) ; nach de laden
A64A 12      560         LD  (de),a ; digitalen Wert in Variable x ablegen
A64B 37      570         SCF ; Carry setzen
A64C C9      580         RET

```

Programm :A/D-Wandler

Start : &A600 Ende : &A64C

Laenge : 004D

0 Fehler

Variablentabelle :

INIT A600 S1 A60B ADRESS A61B ADREN A628 STCON A62F ENDCON A637  
INPUT A63F



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 Grafikkursor an die richtige Stelle gesetzt werden, damit die erste Verbindungslinie nicht einfach aus irgendeiner Bildschirmcke in den Kreis kommt (also den Grafikkursor an die Stelle setzen, die sich bei  $\cos(0)$  und  $\sin(0)$  ergibt:  $\text{mittex} + \text{radius}$ ,  $\text{mittey}$ )!

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

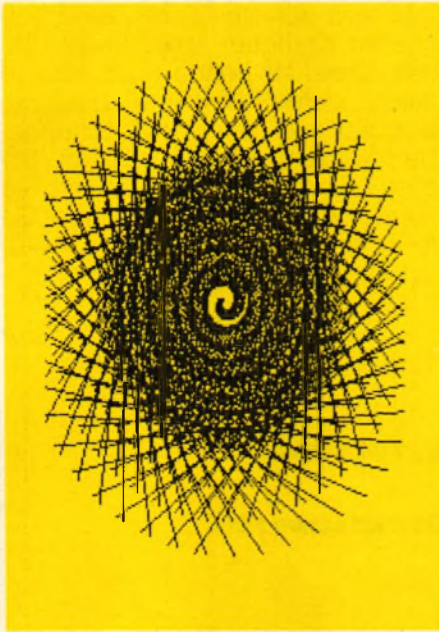
Sie sollten auf jeden Fall versuchen, die Technik des Kreise-Zeichnens zu verstehen, schon al-

lein weil wir sie im nächsten Teil 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 zu lösen; schauen Sie nach, wenn Sie irgendwelche Befehle vergessen haben. Ansonsten ... viel Spaß beim Computern!

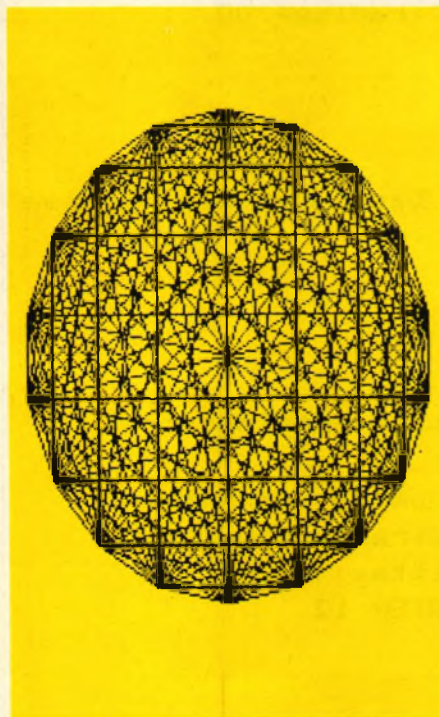
```
10 REM Listing 1 - Kreis zeichnen
20 MODE 1
30 mittex=320:mittey=200:radius=100
40 FOR winkel=0 TO 360
50 x=COS(winkel)
60 y=SIN(winkel)
70 x=x*radius
80 y=y*radius
90 x=x+mittex
100 y=y+mittey
110 PLOT x,y,1
120 NEXT winkel
```

```
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
```



Bisher haben wir uns sehr ausführlich damit beschäftigt, wie man Kreise zeichnet, und das hatte natürlich auch seinen Grund: Denn mit den Funktionen Sinus und Kosinus, die dafür notwendig waren, kann man auch noch ganz andere Gebilde erzeugen, nicht nur Kreise. Weil diese Sachen aber ziemlich kompliziert sind, hier noch einmal das Prinzip in



Kürze: Um einen Kreis auf dem Bildschirm (oder in einem beliebigen Koordinatensystem) zu zeichnen, müssen wir irgendwie die Punkte ermitteln, durch die der Kreis verläuft, d.h. natürlich deren X- (also horizontale) und Y- (vertikale) Koordinaten. Diese Koordina-

ten bekommt man, indem man Sinus und Kosinus eines Winkels ermittelt. Der Winkel ist dabei ein beliebiger Winkel von 0-360, aus denen sich der Kreis ja zusammensetzt. Das heißt aber noch nicht, dass wir nun tatsächlich direkt die Koordinaten herausbekommen, mit denen wir Punkte auf den Bildschirm PLOTten könnten. Es gibt nämlich noch zwei andere Faktoren, die das Aussehen des Kreises mitbestimmen: Die Größe (also der Radius) und seine Lage (also die Lage des Mittelpunktes). Um diese beiden Faktoren richtig beeinflussen zu können, muß man wissen, daß Sinus und Kosinus immer die Koordinaten für den sogenannten **Einheitskreis** ergeben, daß heißt: Für einen Kreis, dessen Mittelpunkt auf 0,0 liegt und der den Radius 1 hat. Um den Kreis auf die gewünschte Größe 'aufzublähen', muß man Sinus/Kosinus vom Winkel-Koordinaten mit dem Radius (mit der Größe) multiplizieren. Da der Mittelpunkt des Einheitskreises 0,0 ist, können einfach die gewünschten Mittelpunkt- Koordi-

naten hinzu addiert werden, und... die Koordinaten stimmen. Grafik 1 macht das noch einmal deutlich. Genau das gerade beschriebene Prinzip wird übrigens in Programm Nr. 1 des vorigen Teils verwendet.

## Andere Anwendungen

Daß wir mit den beiden Funktionen nicht nur Kreise zeichnen können, haben wir ja schon gesehen, als Ellipsen erstellt wurden. Aber Ellipsen sind ja praktisch auch nur gedehnte oder gestauchte Kreise, die entstehen, wenn Sinus- und Kosinus-Werte nicht mit demselben Radius multipliziert werden. Aber mit etwas Phantasie lassen sich damit noch ganz andere Sachen machen: Denken Sie sich einmal ein ganz normales Quadrat. Genau in der Mitte, also

### Listings zum Grafikkurs:

```

1 REM Listing Nr.1
2 REM Polygon
3 MODE 1:INPUT "Anzahl der Ecken (z.B. 18):",ecken
4 PRINT"Ich errechne die Werte..."
5 DIM ex(ecken),ey(ecken)
6 FOR winkel=0 TO 2 *PI-0.001 STEP 2*PI/ecken
7 z=z+1
8 ex(z)=COS(winkel)*198+320:ey(z)=SIN(winkel)*198+
200
9 NEXT winkel
10 CLS
11 FOR a=1 TO ecken-1:farbe=INT(RND*3)+1:FOR b=a+1
TO ecken
12 PLOT ex(a),ey(a),farbe:DRAW ex(b),ey(b)
13 NEXT b,a

```

```

1 REM Listing Nr.2
2 REM Spirale
3 MODE 1
4 farbe=1
5 FOR winkel=0 TO 15*PI STEP PI/23
6 winkel2=winkel+2*PI/3
7 IF winkel MOD PI=0 THEN farbe=farbe+1:IF farbe=4
THEN farbe=1
8 PLOT COS(winkel)*5*winkel+320,SIN(winkel)*5*wink
el+200,farbe
9 DRAW COS(winkel2)*5*winkel2+320,SIN(winkel2)*5*w
inkel2+200
10 NEXT winkel

```

auf dem Punkt, den man erhält, wenn man alle 4 Ecken durch Linien miteinander verbindet, liegt der Mittelpunkt eines Kreises. Dieser Kreis schneidet die Kanten des Quadrates. Wenn uns der Mittelpunkt des Kreises bekannt ist, können wir das Quadrat auf zwei Arten beschreiben: Einmal durch die Koordinaten der 4 Eckpunkte, das ist der Normalfall. Wir können es aber auch durch die Winkel beschreiben, die man für Sinus und Cosinus braucht, um die Koordinaten der 4 Eckpunkte zu erhalten: 45, 135, 225 und 315 Grad. Es erscheint auf den ersten Blick umständlicher, immer erst die entsprechenden Koordinaten ausrechnen zu müssen, aber es hat auch einen riesigen Vorteil: Wir können das Quadrat um sich selbst drehen, einfach indem wir jeden der 4 Grundwinkel vergrößern: Die 4 Eckpunkte werden einfach auf einer Linie entlang der Kreisbahn geführt! Versuchen Sie einmal, ein Programm zu erstellen, das 10 Quadrate zeichnet, jedes um 30 Grad dem anderen gegenüber versetzt (verwenden Sie DEG). Listing Nr.4 sollte Ihnen eine Hilfe dabei sein: dort wird ein Objekt (es handelt sich allerdings nicht ein um Quadrat, da die 4 Eckwinkel anders sind) um sich selbst gedreht und bei jeder Drehung vergrößert; es entsteht der Eindruck eines in die Ferne laufenden Tunnels.

---

## Polygon

---

Eine beliebte Computergrafik wird von Programm Nr.1 erstellt: Sie wird erzeugt, indem alle Diagonalen in einem n-seitigen (im Programm: ecke-seitigen) Polygon gezeichnet werden. Auch das ist nur mit Hilfe von Sinus und Kosinus möglich, auch wenn die Rechnung hier schon wesentlich komplizierter aussieht. Aber im Grunde genommen ist das Prinzip relativ einfach: Es werden lediglich die Koordinaten einer Anzahl von Punkten, die auf der Kreisbahn liegen, errechnet, und diese Punkte werden dann durch gerade Linien miteinander verbunden. Die Punkte sind in der fertigen Zeichnung die Stellen, an denen sich die Linien am Außenrand des Kreises treffen. Das Programm errechnet zuerst die Koordinaten der Punkte

```
1 REM Listing Nr.3
2 REM "Zelt"
3 MODE 2:CALL &BCO2:INK 0,0:BORDER 0
4 FOR winkel=0 TO 2*PI STEP PI/100
5 hv=COS(2*winkel)*200:winkel2=winkel+PI/3
6 hv2=COS(2*winkel2)*200
7 PLOT COS(winkel)*hv+320,SIN(winkel)*hv+200,1
8 DRAW COS(winkel2)+hv2+320,SIN(winkel2)*hv2+200
9 NEXT winkel
```

```
1 REM Listing Nr.4
2 REM Tunnel
3 MODE 2:CALL &BCO2:INK 0,0:BORDER 0
4 radius=1
5 DEG:FOR ofset=10 TO 370 STEP 5
6 PLOT COS(135+ofset)*radius+320,SIN(135+ofset)*radius+200
7 DRAW COS(45+ofset)*radius+320,SIN(45+ofset)*radius+200
8 DRAW COS(-45+ofset)*radius+320,SIN(-45+ofset)*radius+200
9 DRAW COS(270+ofset)*radius+320,SIN(270+ofset)*radius+200
10 DRAW COS(135+ofset)*radius+320,SIN(135+ofset)*radius+200
11 radius=radius+10
12 NEXT ofset
```

und speichert sie im Feld CX0 für die X-Koordinaten, und CY0 für die Y-Koordinaten ab. Erst dann werden sie verbunden. Dabei wird auch noch die Farbe (zufällig) geändert, in der die einzelnen Linien gezeichnet werden sollen. Dadurch wirkt die Zeichnung interessanter. Auf eine Besonderheit (die auch noch für die Programme 2/3 zutrifft), muß allerdings hingewiesen werden: Das Programm arbeitet nicht im Winkelgradmaß, wie wir es bisher immer getan haben (im Winkelgradmaß besteht der Kreis aus den Winkelmaßen von 0-360, und es wird durch DEG eingeschaltet), sondern im Bogenmaß. Dabei hat ein Kreis die Periode  $2\pi$ , nicht 360. Da unser Rechner normalerweise im Bogenmaß rechnet, ist kein spezieller Befehl zum Aktivieren nötig, außer wir hatten vorher auf Winkelgradmaß umgeschaltet: Verwenden Sie dann RAD. Wenn Sie in Winkelgradmaß arbeiten möchten, ersetzen Sie die Schleife durch: *DEG:FOR winkel=0 TO 360 STEP 360/ecke*

---

## Programm Nr. 2

---

Auch Programm Nr.2 erstellt eine populäre Computergrafik. Es

handelt sich um eine Art Schnecke, zusammengesetzt aus einzelnen, schräg liegenden Linien. Das Prinzip ist einfach: Jede gezeichnete Linie liegt in ihrem Anfangspunkt auf einem Kreis, dessen Radius sich allerdings dauernd vergrößert, so daß eine Spirale entsteht. Für den Endpunkt der Linie gilt genau dasselbe, nur dass der Radius größer und der Winkel ein wenig verschoben ist. Überlegen Sie einmal, wie Sie aus dem Programm ersehen können, wieviele Windungen die Schnecke hat (schauen Sie sich dazu die WINKEL-Schleife an und bedenken Sie, daß eine Windung ein Kreis ist, also die Periode  $2\pi$  hat).

---

## $2\pi$

---

Jetzt bleibt nur Programm Nr.3 übrig, aber dabei handelt es sich um etwas ganz Besonderes. Um ehrlich zu sein: Das, was es zeichnet, sollte eigentlich ganz anders aussehen, aber das jetzige Ergebnis sieht besser aus, als das, was ich eigentlich im Sinn hatte. Es entsteht eine Art Zelt, und das ist auch schon alles, was ich dazu sagen möchte.

Bei Maze handelt es sich um ein 3-dimensionales Labyrinth, dem der Spieler entgegen kommen muß. Dabei erscheint das Innere des Labyrinths so auf dem Bildschirm, als würde der Spieler durch den Monitor 'hindurch' die Gänge sehen. Es geht darum, den Ausgang mit möglichst wenigen Schritten zu erreichen. Der Ausgang ist dabei ein Teil irgendeines Ganges, der an seiner sich dauernd ändernden Farbe erkennbar ist. Das ist aber noch nicht alles: Man wird nämlich auch noch von irgendwelchen unangenehmen Zeitgenossen verfolgt, die hinter rücks auf einen schießen. Sobald man den Knall hört bzw. den Explosionssblitz sieht, muß man sich

```

10 REM *****
20 REM ** MAZE (C) **
30 REM *****
40 REM
50 MEMORY &AFFF:DEFINI a-z
60 DIM g(6):PEN 1:PAPER 0:score=1000
70 GOSUB 880:GOSUB 880:REM Titelbild und Maze erzeugen
80 PLOT 0,0,1:MODE 1:INK 0,0:BORDER 0
90 REM
100 scrad=&B1C8: REM &b7cs fuer den 6 b 4 1
110 REM
120 dis=0:POKE scrad,&40
130 RESTORE 190
140 FOR z=0 TO 6:READ x1(z),y1(z),x2(z),y2(z),x3(z),y3(z),x4(z),y4(z):NEXT
150 GOTO 540
160 FOR t=1 TO 6:CLS:FOR nr=6 TO t+1 STEP
290:GOSUB 320:NEXT:nr=t:GOSUB 270:GOSUB 380:NEXT
170 FOR t=6 TO 1 STEP -1:CLS:FOR nr=6 TO t+1 STEP
-1:GOSUB 290:GOSUB 320:NEXT:nr=t:GOSUB 270:GOSUB 3
80:NEXT
180 GOTO 160
190 DATA 319,199,320,199,320,200,319,200
200 DATA 319,199,320,200,319,200
210 DATA 307,191,332,191,332,208,307,208
220 DATA 280,173,359,173,359,226,280,226
230 DATA 210,126,429,126,429,273,210,273
240 DATA 37,9,602,9,602,390,37,390
250 DATA 24,0,615,0,615,399,24,399
260 REM wand nr 1-5 darstellen
270 REM wand-mitte darstellen
280 PLOT x1(nr),y1(nr):DRAW x2(nr),y2(nr):DRAW x3(nr),y3(nr):DRAW x4(nr),y4(nr):DRAW x1(nr):R
ETURN
290 REM wand links 1-5 darstellen
300 PLOT x1(nr),y1(nr):DRAW x1(nr-1),y1(nr-1):DRAW x4(nr-1),y4(nr-1):DRAW x4(nr):DRAW x1(nr)
310 IF nr=1 THEN RETURN
320 REM wand 1-5 rechts darstellen
330 PLOT x2(nr-1),y2(nr-1):DRAW x2(nr),y2(nr):DRAW x3(nr-1),y3(nr-1):DRAW x3(nr),y3(nr):DRAW x2(nr-1)
340 REM ausgang links 1-5

```

```

100 scrad=&B1C8: REM &b7cs fuer den 6 b 4 1
110 REM
120 dis=0:POKE scrad,&40
130 RESTORE 190
140 FOR z=0 TO 6:READ x1(z),y1(z),x2(z),y2(z),x3(z),y3(z),x4(z),y4(z):NEXT
150 GOTO 540
160 FOR t=1 TO 6:CLS:FOR nr=6 TO t+1 STEP
290:GOSUB 320:NEXT:nr=t:GOSUB 270:GOSUB 380:NEXT
170 FOR t=6 TO 1 STEP -1:CLS:FOR nr=6 TO t+1 STEP
-1:GOSUB 290:GOSUB 320:NEXT:nr=t:GOSUB 270:GOSUB 3
80:NEXT
180 GOTO 160
190 DATA 319,199,320,199,320,200,319,200
200 DATA 319,199,320,200,319,200
210 DATA 307,191,332,191,332,208,307,208
220 DATA 280,173,359,173,359,226,280,226
230 DATA 210,126,429,126,429,273,210,273
240 DATA 37,9,602,9,602,390,37,390
250 DATA 24,0,615,0,615,399,24,399
260 REM wand nr 1-5 darstellen
270 REM wand-mitte darstellen
280 PLOT x1(nr),y1(nr):DRAW x2(nr),y2(nr):DRAW x3(nr),y3(nr):DRAW x4(nr),y4(nr):DRAW x1(nr):R
ETURN
290 REM wand links 1-5 darstellen
300 PLOT x1(nr),y1(nr):DRAW x1(nr-1),y1(nr-1):DRAW x4(nr-1),y4(nr-1):DRAW x4(nr):DRAW x1(nr)
310 IF nr=1 THEN RETURN
320 REM wand 1-5 rechts darstellen
330 PLOT x2(nr-1),y2(nr-1):DRAW x2(nr),y2(nr):DRAW x3(nr-1),y3(nr-1):DRAW x3(nr),y3(nr):DRAW x2(nr-1)
340 REM ausgang links 1-5

```

```

100 scrad=&B1C8: REM &b7cs fuer den 6 b 4 1
110 REM
120 dis=0:POKE scrad,&40
130 RESTORE 190
140 FOR z=0 TO 6:READ x1(z),y1(z),x2(z),y2(z),x3(z),y3(z),x4(z),y4(z):NEXT
150 GOTO 540
160 FOR t=1 TO 6:CLS:FOR nr=6 TO t+1 STEP
290:GOSUB 320:NEXT:nr=t:GOSUB 270:GOSUB 380:NEXT
170 FOR t=6 TO 1 STEP -1:CLS:FOR nr=6 TO t+1 STEP
-1:GOSUB 290:GOSUB 320:NEXT:nr=t:GOSUB 270:GOSUB 3
80:NEXT
180 GOTO 160
190 DATA 319,199,320,199,320,200,319,200
200 DATA 319,199,320,200,319,200
210 DATA 307,191,332,191,332,208,307,208
220 DATA 280,173,359,173,359,226,280,226
230 DATA 210,126,429,126,429,273,210,273
240 DATA 37,9,602,9,602,390,37,390
250 DATA 24,0,615,0,615,399,24,399
260 REM wand nr 1-5 darstellen
270 REM wand-mitte darstellen
280 PLOT x1(nr),y1(nr):DRAW x2(nr),y2(nr):DRAW x3(nr),y3(nr):DRAW x4(nr),y4(nr):DRAW x1(nr):R
ETURN
290 REM wand links 1-5 darstellen
300 PLOT x1(nr),y1(nr):DRAW x1(nr-1),y1(nr-1):DRAW x4(nr-1),y4(nr-1):DRAW x4(nr):DRAW x1(nr)
310 IF nr=1 THEN RETURN
320 REM wand 1-5 rechts darstellen
330 PLOT x2(nr-1),y2(nr-1):DRAW x2(nr),y2(nr):DRAW x3(nr-1),y3(nr-1):DRAW x3(nr),y3(nr):DRAW x2(nr-1)
340 REM ausgang links 1-5

```

### So wird gespielt

blitzschnell drücken, um der Kugel auszuweichen, sonst darf man seinen eigenen Grabstein bewundern.

Um sich zu bewegen, kann man die 4 Cursorstasten (die 4 Pfeile oben rechts) oder einen Joystick benutzen. Jeder der 4 Pfeile dreht den Spieler in die entsprechende Himmelsrichtung, d.h. also nicht zwangsläufig, daß einen der Pfeil nach links auch vor eine evtl. linksliegende Abzweigung dreht: Wenn man nach Süden schaut, muß man dazu den rechten Pfeil drücken. Die jeweilige Himmelsrichtung wird im oberen Teil des Bildes angezeigt. Um dann einen Schritt nach vorn zu machen, muß man COPY oder (beim Joystick) den Feuerknopf drücken. Es bringt nichts, wenn man wie ein hämmerer auf den Tasten herum-hämmert, dadurch wird man nicht schneller, sondern bringt höchstens das Programm durcheman-der (es handelt sich nun mal um ein BASIC-Programm, und eine 3D-Darstellung braucht ihre Zeit). Kennt man gegen eine Wand, wird dies angezeigt (BAMM!). Um zu entkommen, braucht man nur den blinkenden Gang zu finden und auf die entsprechende Stelle zu gehen. Je nachdem, wieviele Schritte man gebraucht hat (jeder Schritt bringt Minuspunkte), bzw. wie oft man den Feinden entkommen konnte (bringt Pluspunkte), ist das erzielte Ergebnis. Die Feinde machen sich durch Schußgeräusche und Explosionsblitze bemerkbar. Man muß so schnell wie



Druecken Sie eine beliebige Taste...

möglich die ENTER-Taste drücken, um der Kugel auszuweichen, sonst ist das Spiel(-leben) zu Ende.

des Spielers und die des Ausgangs durch entsprechende Symbole angezeigt. Nun kann man wählen, ob man ein neues Labyrinth haben möchte ('N' drücken), oder ob man sich hineinwagen will ('W' drücken). Dass der Spieler diese Wahlmöglichkeit hat, hat zwei Gründe: Zum einen kann man sich so ein einfacheres oder schwierigeres Labyrinth, ganz nach Geschmack und Geschick, auswählen (es ist natürlich lächerlich, wenn der Ausgang nur einen

Schritt vor der Startposition des Spielers ist), zum anderen generiert der Computer aber auch keinesfalls immer sinnvolle Gebilde: So ist es u.U. nicht möglich, den Ausgang zu erreichen, weil Verbindungsgänge fehlen, oder der Spieler ist nicht sichtbar: Es wäre natürlich Unsinn, sich in ein solches Labyrinth zu begeben. Ansonsten... wenn Sie jetzt noch keine Platzangst haben, nach diesem Spiel haben Sie sie bestimmt. Viel Spaß dabei. tmb

## Zufällige Labyrinth

Der Computer erzeugt vor dem eigentlichen Spiel das jeweilige Labyrinth. Diesen Vorgang kann man auf dem Bildschirm mitverfolgen. Im Labyrinth wird die Position

```

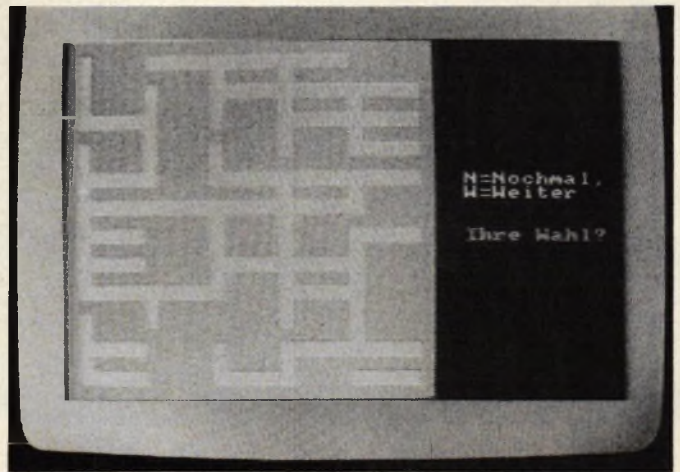
350 IF sw=1 THEN RETURN ELSE PLOT x1(nr-1),y1(nr-1)
: DRAW x1(nr),y1(nr-1): PLOT x4(nr-1),y4(nr-1): DRAW
x1(nr),y4(nr-1): PLOT x1(nr-1),y1(nr-1): DRAW x4(nr-1),y4(nr-1): RETURN
360 REM ausgang rechts 1-5
370 IF sw=1 THEN RETURN ELSE PLOT x2(nr-1),y2(nr-1)
: DRAW x2(nr),y2(nr-1): PLOT x3(nr-1),y3(nr-1): DRAW
x2(nr),y3(nr-1): PLOT x2(nr-1),y2(nr-1): DRAW x3(nr-1),y3(nr-1): RETURN
380 IF dis=0 THEN dis=1: OUT &BC00,12: OUT &BD00,&10
: POKE scrad,&CO ELSE dis=0: POKE scrad,&40: OUT &BC00,12: OUT &BD00,&30
390 RETURN
400 REM ** Bild aus den 6 Schritten in g(..) erzeugen
410 CLS: sw=0: FOR t=6 TO 1 STEP -1: REM g(6)=vorderer schritt
420 IF t=zs THEN PLOT 640,400,3 ELSE PLOT 640,400,1
430 IF sw=1 THEN 480: REM sicht versperrt
440 a=g(t): nr=t
450 IF (a AND 64)=64 THEN GOSUB 270: sw=1: GOTO 480: REM wand mitte
460 IF (a AND 128)=128 THEN GOSUB 290 ELSE GOSUB 340: REM wand links
470 IF (a AND 32)=32 THEN GOSUB 320 ELSE GOSUB 360: REM wand rechts
480 NEXT
490 IF du=1 THEN PEN 2: LOCATE 18,2: PRINT "Puhh..": PEN 1
500 LOCATE 18,1: IF hr=1 THEN PRINT "West"; ELSE IF hr=2 THEN PRINT "Nord"; ELSE IF hr=3 THEN PRINT "Ost"; ELSE IF hr=4 THEN PRINT "Sued";
510 SOUND 2,400,5,7,0,0,4
520 RETURN
530 REM Hauptschleife *****
540 INK 3,RND*27,RND*27: GOSUB 1060: GOSUB 400: GOSUB 380: du=0
550 IF INT(RND*905)-1 THEN GOSUB 1400: IF du=0 THEN GOTO 1260 ELSE 540: REM erschossen?
560 IF ox=zx AND oy=zy THEN 1490
570 IF JOY(0)=0 AND INKEY(0)<>0 AND INKEY(1)<>0 AND INKEY(2)<>0 AND INKEY(8)<>0 AND INKEY(9)<>0 THEN 550
580 IF JOY(0)<>16 AND INKEY(9)<>0 THEN 610
590 IF (g(5) AND 64)=64 THEN GOSUB 380: PLOT 0,0,1: CLG 1: LOCATE 8,14: PEN 2: INK 2,6,3: PAPER 1: PRINT "BAMM!": PEN 1: PAPER 0: SOUND 1,200,20,7,0,0,5: FOR ws=1 TO 100: NEXT: INK 1,6: FOR ws=1 TO 100: NEXT: INK 1,24: GOSUB 380: GOTO 610: REM kein Weiterkommen, Wand 1 m Weg
600 IF hr=1 THEN ox=MAX(ox-1,2) ELSE IF hr=2 THEN oy=MAX(2,oy-1) ELSE IF hr=3 THEN ox=MIN(ox+1,26) ELSE IF hr=4 THEN oy=MIN(26,oy+1)
610 IF (JOY(0)=4 OR INKEY(8)=0) THEN hr=1
620 IF (JOY(0)=8 OR INKEY(1)=0) THEN hr=3
630 IF (JOY(0)=2 OR INKEY(2)=0) THEN hr=4
640 IF (JOY(0)=1 OR INKEY(0)=0) THEN hr=2
650 score=score-5
660 IF score<=0 THEN MODE 2: INK 0,0: INK 1,26: PEN 1: PAPER 0: PRINT: PRINT "Ahhhh...": PRINT "Sie sind zu lange im Labyrinth herumgelaufen.": PRINT "Nun sind Sie verhungert.": GOTO 1340
670 GOTO 540
680 REM Labyrinth erzeugen
690 DIM l$(35): MODE 1

```

```

700 BORDER 1: INK 3,3: PAPER 3: CLS: PAPER 0: INK 0,12: INK 1,6: PEN 1
710 FOR y=1 TO 25: LOCATE 1,y: l$(y)=STRING$(27,207): PRINT l$(y);: NEXT
720 FOR x=2 TO 26 STEP 5
730 FOR y=2 TO 24 STEP 2
740 IF INT(RND*2)-1 THEN LOCATE x,y: PRINT " " : MIDS(l$(y),x,5)=SPACE$(5): SOUND 1,200,2,7,0,0,10
750 NEXT: NEXT
760 FOR y=2 TO 22 STEP 2
770 FOR x=2 TO 26 STEP 5
780 IF INT(RND*5)>1 THEN FOR t=y TO y+2: LOCATE x,t: PRINT " " : MIDS(l$(t),x,1)=" " : NEXT: SOUND 1,200,2,7,0,0,11
790 NEXT: NEXT: INK 2,15,3: SPEED INK 5,5: PAPER 3: PEN 2: LOCATE 30,10: PRINT "N-Nochmal, " : LOCATE 30,11: PRINT "W-Weiter": LOCATE 30,14: PEN 1: PRINT "Ihre Wahl?": PAPER 0

```



```

800 oy=INT(RND*20)+3: IF INSTR(l$(y)," ")=0 THEN 800
810 ox=INSTR(l$(y)," "): LOCATE ox,oy: PEN 2: PRINT CHR$(250);
820 zy=INT(RND*20)+3: IF INSTR(l$(y)," ")=0 THEN 820
830 zx=INSTR(l$(y)," "): IF xz=ox AND xy=oy THEN 820 ELSE LOCATE zx,zy: PEN 2: PRINT CHR$(240);: PEN 1
840 xa=0: ya=-1: hr=2: REM Himmelsrichtung Nord
850 a$=INKEY$: IF a$="" THEN 850
860 IF UPPER$(a$)="W" THEN RETURN
870 GOTO 700
880 REM Titelbild
890 MODE 1: INK 0,3: BORDER 3: INK 1,24: INK 2,5: INK 3,6
900 FOR y=-15 TO 399 STEP 40: PLOT 0,y: DRAW 640,y,3: NEXT
910 FOR x=0 TO 710 STEP 40: PLOT x-70,0: DRAW x,399,3: NEXT

```

```

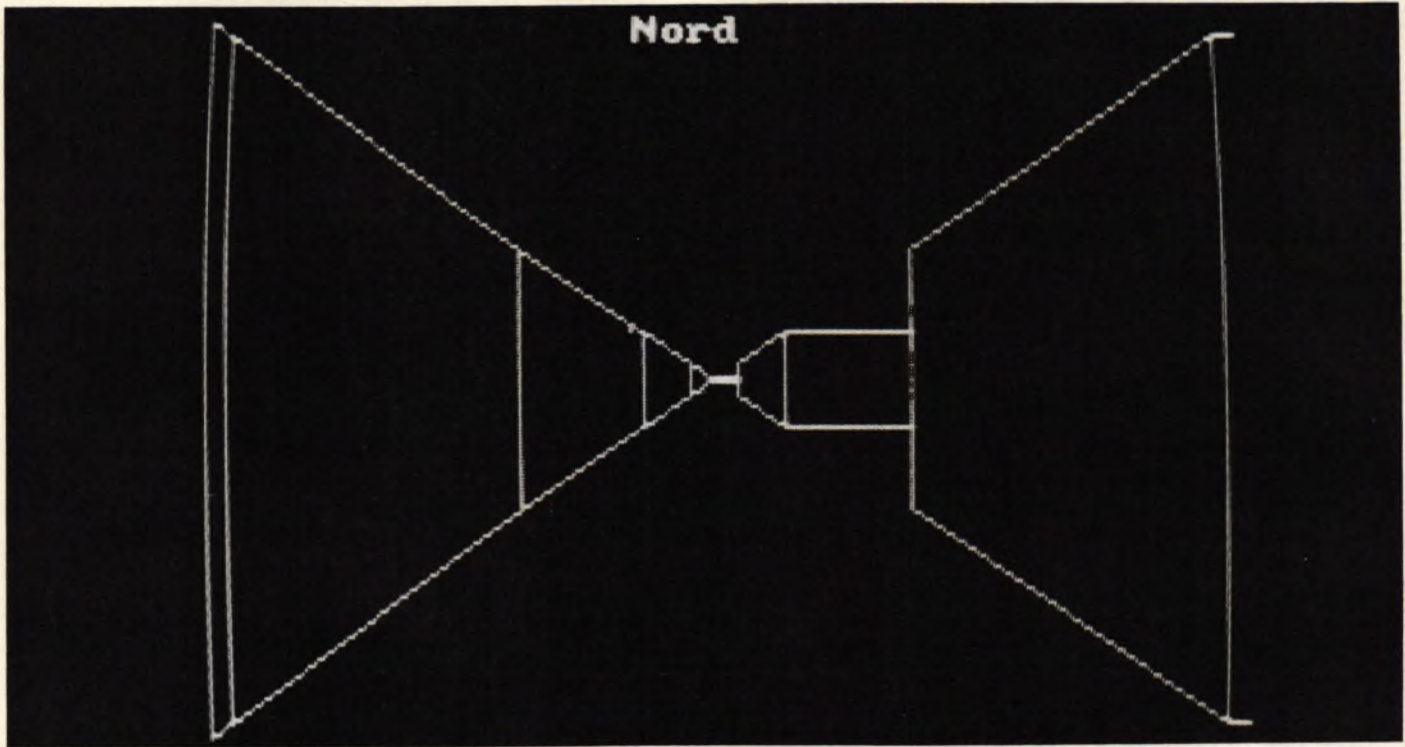
920 DATA 8,2,-4,2,4,2,-8,2,0,0,5,8,4,-4,2,3,0,-4,0
,4,-4,2,3,0,0,5,0,8,0,-8,8,8,0,-8,0,8,-8,-8,0,8,5,
8,0,0,8,0,-8,-3,0,0,8,0,-8,-5,0,0,8,99,99
930 RESTORE 920:z=0
940 xs=150:FOR ys=230 TO 210 STEP -2:PLOT xs,ys,IN
T(RND*3)+1:xs=xs+2:RESTORE 920
950 READ y,x:IF x<>99 THEN DRAWR x*10,y*20:GOSUB 1
030:GOTO 950
960 NEXT
970 PEN 1:LOCATE 11,15:PRINT CHR$(164)"TMB & Compu
ter Team";
980 LOCATE 3,22:PEN 2:PRINT"Druecken Sie eine beli
ebige Taste...";
990 z=1:a$="":WHILE z<3000 AND a$="":a$=INKEY$:z=z
+1:WEND
1000 IF a$<>" " THEN RETURN
1010 IF INKEY$=" " THEN INK INT(RND*1)+1,RND*27:GOT
O 1010
1020 RETURN
1030 REM Sound bei Titelbild
1040 z=z+1:IF z=5 THEN z=1 ELSE RETURN
1050 SOUND 1,200+RND*400,5,RND*15:RETURN
1060 REM Daten fuer 1 Bild erzeugen,
1070 REM je nach Plan und Himmelsrichtung
1080 zs=99:FOR t=1 TO 6:g(t)=64:NEXT:z=6
1090 IF hc<>2 AND hc<>4 THEN 1160

```

```

1290 FOR x=250 TO 390 STEP 4:PLOT x,90:DRAW x,250,
2:NEXT:FOR x=310 TO 330:PLOT x,250:DRAW x,360:NEXT
:FOR y=300 TO 320:PLOT 270,y:DRAW 370,y:NEXT
1300 FOR x=0 TO 640 STEP 4:PLOT x,0:DRAW x,RND*30+
105,1:NEXT
1310 ENT 1,200,2,1:SOUND 1,284,200,15,0,1
1320 LOCATE 7,23:PAPER 1:INK 4,6:PEN 4:PRINT"Sorry
...":PEN 0:PAPER 1:CALL &BBO6
1330 MODE 2
1340 PRINT:PRINT:INK 1,26:BORDER 0:INK 0,0:PRINT"E
s hat sie erwischt.":PRINT"Seien Sie nicht zu depr
imiert!":PRINT:PRINT"Bitte waehlen Sie:":PRINT"1)
Noch ein Versuch im selben Feld,":PRINT"2) Spiel v
on vorn beginnen,":PRINT"3) Spiel beenden.":a$="
1350 a$=INKEY$:IF a$=" " THEN 1350
1360 IF a$="1" THEN score=500:INK 0,0:BORDER 0:INK
1,26:PEN 1:PAPER 0:MODE 1:GOTO 540
1370 IF a$="2" THEN RUN
1380 IF a$="3" THEN PRINT"Ok...":STOP
1390 GOTO 1350
1400 REM Auf Spieler wird geschossen
1410 u=INT(RND*10)+30:SOUND 1,400,20,7,0,0,12
1420 FOR t=1 TO 10:BORDER RND*27:INK 0,RND*27:NEXT
:INK 0,0:BORDER 0
1430 du=0:FOR t=1 TO u*10
1440 IF INKEY(18)=0 THEN du=1

```



```

1100 x=ox:y=oy:WHILE z<>1 AND y>=2:g(z)=0:IF x=zX
AND y=zY THEN zs=z
1110 IF MIDS(1$(y),x-1,1)<>" " THEN g(z)=128
1120 IF MIDS(1$(y),x,1)<>" " THEN g(z)=g(z)+64
1130 IF MIDS(1$(y),x+1,1)<>" " THEN g(z)=g(z)+32
1140 z=z-1:IF hc=2 THEN y=y-1 ELSE IF hc=4 THEN y=
y+1
1150 WEND:RETURN
1160 IF hc=3 THEN 1210 ELSE x=ox:y=oy:WHILE z<>1 A
ND x>=2:g(z)=0:IF x=zX AND y=zY THEN zs=z
1170 IF MIDS(1$(y+1),x,1)<>" " THEN g(z)=128
1180 IF MIDS(1$(y),x,1)<>" " THEN g(z)=g(z)+64
1190 IF MIDS(1$(y-1),x,1)<>" " THEN g(z)=g(z)+32
1200 z=z-1:x=x-1:WEND:RETURN
1210 x=ox:y=oy:WHILE z<>1 AND x<=24:g(z)=0
1220 IF MIDS(1$(y-1),x,1)<>" " THEN g(z)=128
1230 IF MIDS(1$(y),x,1)<>" " THEN g(z)=g(z)+64
1240 IF MIDS(1$(y+1),x,1)<>" " THEN g(z)=g(z)+32
1250 z=z-1:x=x+1:WEND:RETURN
1260 REM Verloren-Bild anzeigen
1270 bs="":FOR t=1 TO 100:bs=bs+INKEY$:NEXT::MODE
0:INK 3,9:INK 4,1:INK 1,12:INK 2,0:BORDER 1
1280 FOR x=0 TO 640 STEP 4:PLOT x,90:DRAW x,RND*30
+200,3:DRAW x,399,4:NEXT

```

```

1450 NEXT
1460 IF du=1 THEN score=score+100
1470 INK 0,0:BORDER 0
1480 RETURN
1490 REM geschafft...
1500 bs="":FOR t=1 TO 100:bs=bs+INKEY$:NEXT
1510 PAPER 0:PEN 1:ENV 1,100,1,3:ENT 1,100,-2,3:SO
UND 1,284,300,1,1,1
1520 MODE 0:PRINT CHR$(22)CHR$(1);:FOR t=1 TO 40:P
EN RND*15:LOCATE RND*14+1,RND*24+1:PRINT"Sieg!!";:
BORDER RND*27:NEXT:BORDER 0:INK 0,0:PRINT CHR$(22)
CHR$(0);:LOCATE 2,24:PEN 1:INK 1,26:PRINT" Mit" sco
re"Punkten "
1530 CALL &BBO6:PEN 1:MODE 0:INK 1,0:PAPER 3:CLS: I
NK 3,6:LOCATE 7,10:PRINT"Nochmal?";
1540 CALL &BBO6
1550 a$=INKEY$:IF a$=" " THEN 1550 ELSE a$=UPPER$(a
$)
1560 IF a$="J" THEN RUN
1570 MODE 2:PEN 1:PAPER 0:INK 1,26:PRINT"Ok...":ST
OP

```



# VEKTOR- GRAFIK mit dem CPC

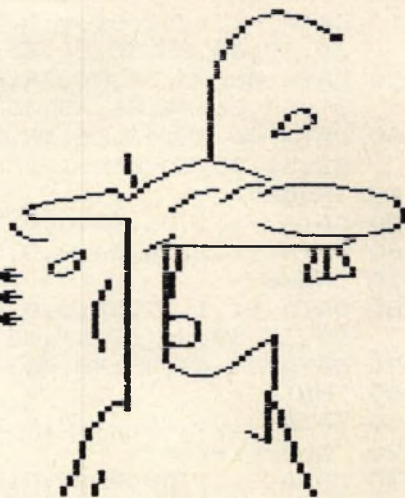
Dieses Programm demonstriert die Anwendung von Vektorgrafik. Das kleine grüne Männchen entstand durch Markierung von markanten Punkten in einem Koordinatensystem. Die Koordinaten dieser Punkte wurden in Datenzeilen an das Programm angefügt. Das Programm erledigt die Verbindung der Punkte mit Linien. Dabei ist zu beachten, daß das Männchen nicht in einem Zug gezeichnet werden kann. Der Stift muß also während des Zeichnens an einen neuen Anfangspunkt geführt werden. Das Programm erkennt einen solchen neuen Anfangspunkt an einem vorangestellten Koordinatenpaar (-1,-1). Ab diesem Punkt setzt das Programm solange Punkte, bis es das Koordinatenpaar (0,0) liest. Ab dieser Stelle werden die Punkte wieder mit DRAW verbunden.

Das Ende der Daten wird mit (999,999) markiert. Dadurch ist der Anzahl der Daten, außer durch den Speicherplatz, keine Grenze gesetzt. Auf diese Art und Weise können auch komplexere Grafiken dargestellt werden. Es eignen sich zum Beispiel Landkartenausschnitte, da dort die Koordinaten direkt am Gradnetz abgelesen werden können.

Eine auf diese Weise erstellte Grafik kann theoretisch in jeder beliebigen Größe dargestellt werden, da die Koordinaten der Punkte vor dem Plotten nur mit einem Vergrößerungsfaktor multipliziert werden müssen.

Für Farbmonitorbesitzer ergibt sich auch die Möglichkeit, den y-Wert der Markierungspunkte zur Farbcodierung zu verwenden, da diese nur an ihrem x-Wert erkannt werden. So kann dann ab einem neuen Anfangspunkt auch mit einer neuen Farbe weitergezeichnet werden.(tb)

```
XXXXXXXXXXXXX
# 1986 #
XXXXXXXXXXXXX
```



```
[ ( c ) THBCS ]
```

```
10 '** (c) Thomas Barndt **
20 '**      1985          **
30 '
40 MODE 1
50 BORDER 13
60 DEFINT g-z
70 INPUT"Mode:",m
80 INPUT"Vergrößerungsfaktor (1 - 4.4):",f
90 MODE m:m=(m+1)*3-2
100 ORIGIN 440-40*f,240-50*f
110 CLS
120 '
130 'Malschleife
140 READ x,y
150 IF x=999 THEN 200
160 IF x=-1 THEN plotten=-1:PEN y:GOTO 140
170 IF x=0 THEN plotten=0:GOTO 140
180 IF plotten THEN PLOT x*f,y*f ELSE DRAW x*f,y
*f
190 GOTO 140
200 '
210 LOCATE 4*m,4
220 PRINT"
230 LOCATE 5*m,6
240 PRINT"
250 LOCATE 2*m,13
251 PRINT"*****"
260 LOCATE 2*m,14
270 PRINT"* 1986 *"
280 LOCATE 2*m,15
290 PRINT"*****"
300 LOCATE 6*m,25
310 PRINT"[ (c) THBCS]"
320 CALL &BB06:RUN
330 '
340 '      *** DATEN fuer Punkte ***
350 'Ohr
360 DATA -1,1,20,51,0,0,6,51,5,52,6,53,7,53,11,5
4
370 DATA 15,55,16,55,17,56,18,56,19,55,20,54
380 'Schaedel
390 DATA 21,55,23,57,26,59,29,60,32,61,36,60,38,
59,40,58
```

```

400 'Antenne
410 DATA -1,1,32,61,0,0,32,75,33,76,33,77,33,78,
34,79,34,80,35,81,35,82
420 DATA 36,83,37,84,38,85,39,86,40,86,41,87,42,
87,43,86,44,86,45,86
430 DATA 46,85,47,85,48,84,49,84,50,83,50,82,51,
82,51,83,50,83
440 'Augen
450 DATA -1,1,26,52,0,0,29,55,33,56
460 DATA -1,1,34,54,0,0,35,55,36,56,38,57,39,57
470 'Nase
480 DATA -1,1,39,55,0,0,40,56,44,57,49,57,54,56,
57,55,59,54,60,52,60,51,58,49
490 DATA 54,48,45,48,46,48,41,49,39,50
500 'Hals
510 DATA -1,1,20,51,0,0,20,18
520 'Oberkiefer
530 DATA -1,1,26,48,0,0,23,45
540 DATA -1,1,25,46,0,0,51,46,52,47,51,48
550 'Zaehne
560 DATA -1,1,47,46,0,0,46,40,52,41,51,46
570 DATA -1,1,49,46,0,0,49,41
580 'Mund

```

```

*****
* 1986 *
*****

```



[(c) THBCS]

```

590 DATA -1,1,25,46,0,0,25,27,26,25,29,24,32,24
600 DATA 35,25,37,27,41,31,40,12,38,14
610 'Zunge
620 DATA -1,1,25,33,0,0,26,34,28,34,30,33,30,31
630 DATA 29,29,26,29,25,31,25,33,27,46
640 'Koerper
650 DATA -1,1,20,18,0,0,19,19,18,19,16,17,14,14,
13,11,12,8,11,3
660 DATA -1,1,16,7,0,0,15,3
670 DATA -1,1,41,19,0,0,42,17,44,11,46,5,47,3
680 'Haar,Tropfen + Striche
690 DATA -1,1,22,56,0,0,21,57,20,60,20,62
700 DATA -1,1,13,45,0,0,11,44,9,43,9,42,10,41,11,
41,12,42,13,45
710 DATA -1,1,41,65,0,0,43,69,45,70,46,70,47,69,
47,68,45,66,41,65
720 DATA -1,1,17,40,0,0,15,35,15,32,16,28
730 DATA -1,1,3,50,0,0,4,48,6,47,8,47
740 DATA 999,999

```

## MULTIPLAN für den Schneider CPC

Professionelles Planen und Kalkulieren mit dem Schneider CPC. September 1985, 226 Seiten mit 2 Abbildungen und 89 Tabellen. DM 49,-.

Multiplan wurde ursprünglich für das 16 Bit-Betriebssystem MS-DOS entwickelt. Inzwischen ist aber auch die in diesem Buch beschriebene CP/M-Version für den Schneider CPC auf dem Markt, die den vollen Leistungsumfang der 16 Bit-Version enthält.

Das vorliegende Buch soll eine praktische Einführung in den Umgang mit Multiplan auf dem Schneider CPC geben. Für den Leser, der sich erstmals mit einem Tabellenkalkulationsprogramm beschäftigt, wird zunächst eine kurze Einführung in die Arbeitsweise solcher Planungssysteme gegeben. Nach einigen Vorbemerkungen zur Inbetriebnahme von Multiplan werden anhand von praxisnahen Beispielen alle Befehle und Funktionen beschrieben, und zwar in der Reihenfolge, die der Arbeit in der Praxis entspricht. In weiteren Abschnitten werden die Befehle im Detail beschrieben und Leistungen aufgezeigt, die für den fortgeschrittenen Anwender interessant sind, wie zum Beispiel das Verknüpfen mehrerer Multiplan-Tabellen.

Die vielfältigen Einsatzmöglichkeiten von Multiplan werden in einem größeren Abschnitt anhand von Beispielen vorgeführt. Sie sollen den Leser vor allem dazu anregen, eigene Anwendungen mit Multiplan zu erarbeiten. Den praktischen Vorfürungen schließt sich eine Zusammenstellung aller Befehle und Funktionen sowie einiger wichtiger CP/M-Befehle für den schnellen Überblick an.

Der Autor: Dr. Peter Albrecht, geb. 1944, studierte Nachrichtentechnik an der Technischen Universität München und promovierte dort mit einer Arbeit über Mikrowellenhalbleiter zum Dr.-Ing.. Danach war er 12 Jahre in verschiedenen Funktionen bei der NCR GmbH tätig. Heute ist Dr. Peter Albrecht als Unternehmensberater in Augsburg tätig.

# SCHNEIDER JOYCE



Der neue Schneider Joyce ist ein Textverarbeitungssystem mit dem Betriebssystem CP/M Plus und 256 KByte Hauptspeicher. Im Lieferumfang enthalten sind die Sprachen BASIC und Logo sowie das Textverarbeitungsprogramm LocoScript.

## Die Hardware

Der eigentliche Rechner befindet sich mitsamt allen Netzteilen im Monitorgehäuse, wodurch nur ein Netzkabel benötigt wird. Dies hat leider zur Folge, daß die Schrift etwas flackert, was zusätzlich mit der Tatsache, daß der Monitor nicht entspiegelt ist, eine eher negative Beurteilung des Bildes zur Folge hat. Positiv ist die Darstellung von 90x32 Zeichen, wodurch das Editieren von Texten an Übersichtlichkeit gewinnt.

Die beiden Handbücher ermöglichen dem Einsteiger eine schnelle Einarbeitung in das System. Als Nachschlagewerk für Fortgeschrittene sind sie jedoch weniger geeignet.

## Das Laufwerk

Das Diskettenlaufwerk ist ein 3-Zoll-Gerät in gewohnter Schneider-Manier mit 180 KByte pro Seite. Das optionale Zweitlaufwerk kann bis zu 760 KByte auf eine Diskette aufzeichnen.

Es ist jedoch nicht ohne weiteres möglich, die mitgelieferten Disketten mit einem CPC 464 zu lesen. Umgekehrt ist es möglich, z.B. Logo-Programme vom 464 problemlos auf den Joyce zu übertragen. Wenn man Programme vom Joyce übertragen will, so muß man diese zunächst auf eine Diskette vom CPC 464 aufzeichnen. Dies ist für den Joyce kein Problem.

## Die Tastatur

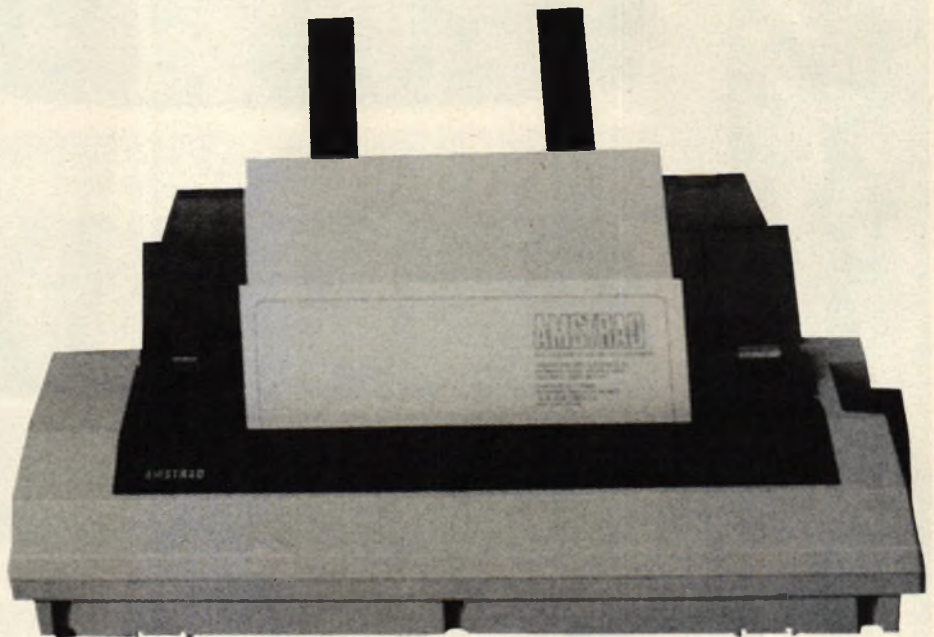
Der Joyce hat eine deutsche QWERTZ-Tastatur mit deutschen Umlauten und Sonderzeichen. Das Anschlußkabel zum Monitor ist lang genug, so daß das Platzieren der Tastatur keine Probleme bereiten dürfte.

Die Tastatur ist leider ähnlich wie beim CPC 6128 nicht in separate Funktionsblöcke unterteilt. Außerdem wird man durch abwechselnd deutsche und englische Bezeichnungen irritiert. Das Klappern der Tasten ist lauter als beim CPC 464.

sowie Proportionalschrift. Im Textverarbeitungsprogramm eingegebene, über Tastatur erreichbare Sonderzeichen werden ausgedruckt. Die Stromversorgung erhält er durch ein extra Kabel vom Monitor.

Leider konnten wir in keinem der beiden Handbücher die Steuer-codes finden, über welche die verschiedenen Schriftarten eingestellt werden können. So dürfte es wohl einem BASIC-Programmierer etwas schwerfallen, die Möglichkeiten des Druckers voll auszunutzen.

Schwer herauszufinden war auch die Möglichkeit, über die Tastenkombination EXTRA + PTR ei-



## Der Drucker

Der unscheinbar aussehende Drucker weist alle üblichen Schriftarten auf. Zusätzlich bietet er eine akzeptable Schönschrift

Unser Bild zeigt den Drucker des PCW8256

ne Hardcopy des Bildschirms anzufertigen.

Den Joyce kann man mit dem Modul CPS-8256, welches eine V.24- und eine Centronicsschnittstelle besitzt, nachrüsten.

## Die Software

Nach dem Einschalten des Rechners muß zunächst das Betriebssystem von der Diskette geladen werden. Das mitgelieferte CP/M Plus besitzt zahlreiche Dienstprogramme, die das Arbeiten mit dem Joyce vereinfachen. Unter anderem verfügt man über den Macroassembler MAC von Digital Research.

Wie schon der CPC 6128, besitzt auch der Joyce einen HELP-Befehl, mit dessen Hilfe man sich über die wichtigsten CP/M-Dienstprogramme informieren kann.

Der Joyce stellt sofort nach dem Einschalten eine virtuelle RAM-Floppy mit 112 KByte zur Verfügung, welche ein sehr schnelles Laden und Speichern von Dateien ermöglicht.

## Das BASIC

Das BASIC der Firma Locomotive ist nicht mehr mit den bisherigen Versionen kompatibel. Es wird hierzu ein mehr als 380 Seiten umfassendes Handbuch mitgeliefert. Das sogenannte Mallard-BASIC besitzt die komfortable Möglichkeit, relative Dateien mit Schlüsselwörtern zu verwalten. In diesem Zusammenhang ist uns als erstes der Befehl SEEKRANK, welcher zum Suchen von Schlüsseln verwendet wird, durch seinen etwas irreführenden Namen aufgefallen.

Die Verarbeitung von doppelt genauen Zahlen mit einer Genauigkeit von 16 Dezimalstellen liefert einen weiteren Pluspunkt für das BASIC.

Leider stehen keine Grafikbe- fehle zur Verfügung, so daß die Auflösung von 720x256 Punkten nur in dem etwas langsameren Logo genutzt werden kann. Auch Sound-Befehle sucht man vergeblich; der einzige Ton, den wir dem Joyce entlocken konnten, war ein helles Piepsen, welches ertönt, wenn man versehentlich eine falsche Taste getroffen hat.

Das BASIC belegt etwa 28 KByte des Hauptspeichers, der Bildschirm Speicher etwa 22,5K. Wenn man die 112K für das Laufwerk M abrechnet, bleiben also von den gesamten 256K etwas über 90K

übrig. Im BASIC bleiben aber nur etwas über 30K frei. Man muß sich also fragen, warum der restliche Speicher nicht auch genutzt wird.

## Logo

Das Logo des Joyce ist weitgehend identisch mit dem des CPC 6128, jedoch steht die wesentlich höhere Auflösung von 720x256 Punkten zur Verfügung. Die Geschwindigkeit hat gegenüber dem 6128 auch etwas zugenommen. Positiv ist, daß auch hier jederzeit eine Hardcopy des Bildschirms angefertigt werden kann.

## Textverarbeitung

Die Textverarbeitung hat den großen Vorteil einer Menüführung, wodurch das Arbeiten etwas erleichtert wird. Dennoch benötigt man zu Anfang etwas Eingewöhnungszeit, um die vielfältigen Möglichkeiten voll nutzen zu können. So haben wir zum Beispiel ei-

nige Zeit gebraucht, bis wir herausfanden, wie man einen Text formatiert.

Es werden zahlreiche Textschablonen und vorformulierte Sätze für Briefe und Formulare mitgeliefert, welche man sich auch selbst erstellen kann.

Leider ist es nicht möglich, von LocoScript in CP/M zu wechseln oder umgekehrt, da man, um die Textverarbeitung zu starten, den Rechner erst ausschalten muß. Leider sind die erzeugten Textdateien nicht kompatibel zu anderen Textverarbeitungsprogrammen, so daß man beim Umsteigen von einem anderen Schneider auf den Joyce seine Texte nicht weiterbearbeiten kann. Auch umgekehrt ist dies nur schlecht möglich, da auch Steuerzeichen mit gespeichert werden.

Überhaupt ist die Diskettenverwaltung mit LocoScript einer der schwerwiegendsten Kritikpunkte. Man muß nämlich, um Disketten zu formatieren oder zu kopieren, immer erst CP/M booten. Aber wenn man sich erst einmal daran gewöhnt hat, kommt man auch hiermit zurecht.

Dichte 10 in doppeltem Abstand

kursiv in Dichte 12  
kursiv in Dichte 17

ganz klar kann man Hoch<sup>2,3</sup> und Tief<sub>abcd</sub>stellen

Man kann voll unterstreichen

oder einzelne Worte unterstreichen

Fettdruck mit und ohne Unterstreichung

Doppelter Anschlag  
an kann auch Sonderzeichen darstellen

•"¥+2138!;†«»08|¿\™"0e p t t0wπ[ ] (e0λ+→+0+eYΓδΔσΣαγχβ±µç••±{ }=κελδ+πλζ  
ζκπλσµνξ

## Fazit

### Vorteile:

Ausführliche Handbücher (für Einsteiger)

BASIC mit relativen Dateien und doppelt genauen Zahlen

CP/M Plus mit vielen Dienstprogrammen

Drucker mit NLQ und Proportional- schrift

Kein Kabelsalat

### Nachteile:

Nicht entspiegelter Monitor

Tastatur nicht in Blöcke aufgeteilt

BASIC ohne Grafik

Speicher wird in BASIC und Logo

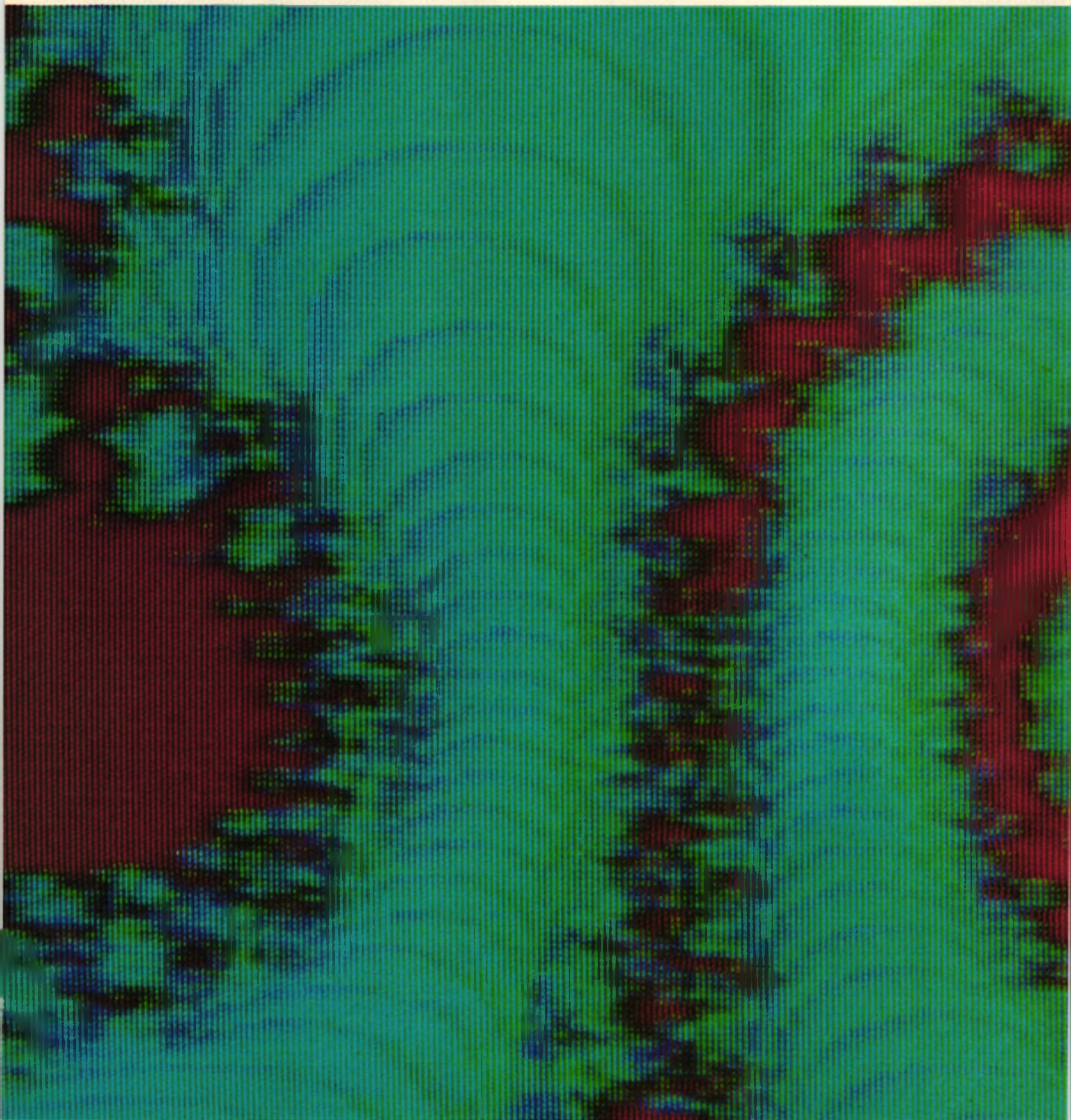
nicht voll genutzt

Textdateien nicht kompatibel zu anderen Systemen

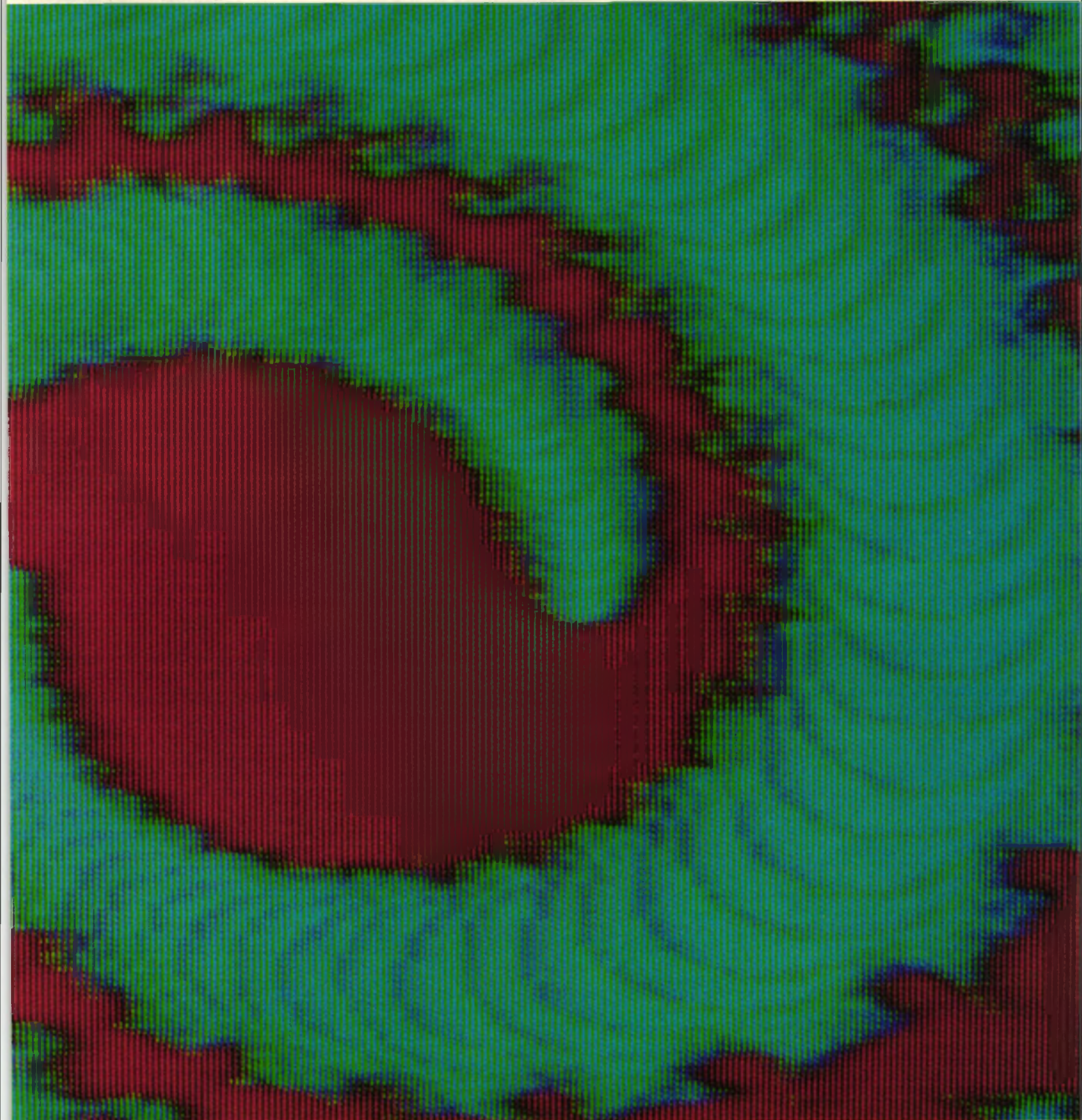
Alles in allem ist der Joyce wohl, aufgrund des unzureichenden Monitors, kein Rechner für die Sekretärin, welche den ganzen Tag Briefe zu schreiben hat, sondern eher für Leute, die nur hin und wieder etwas schreiben, aber nicht auf den Komfort eines guten Textverarbeitungs-systems verzichten wollen.

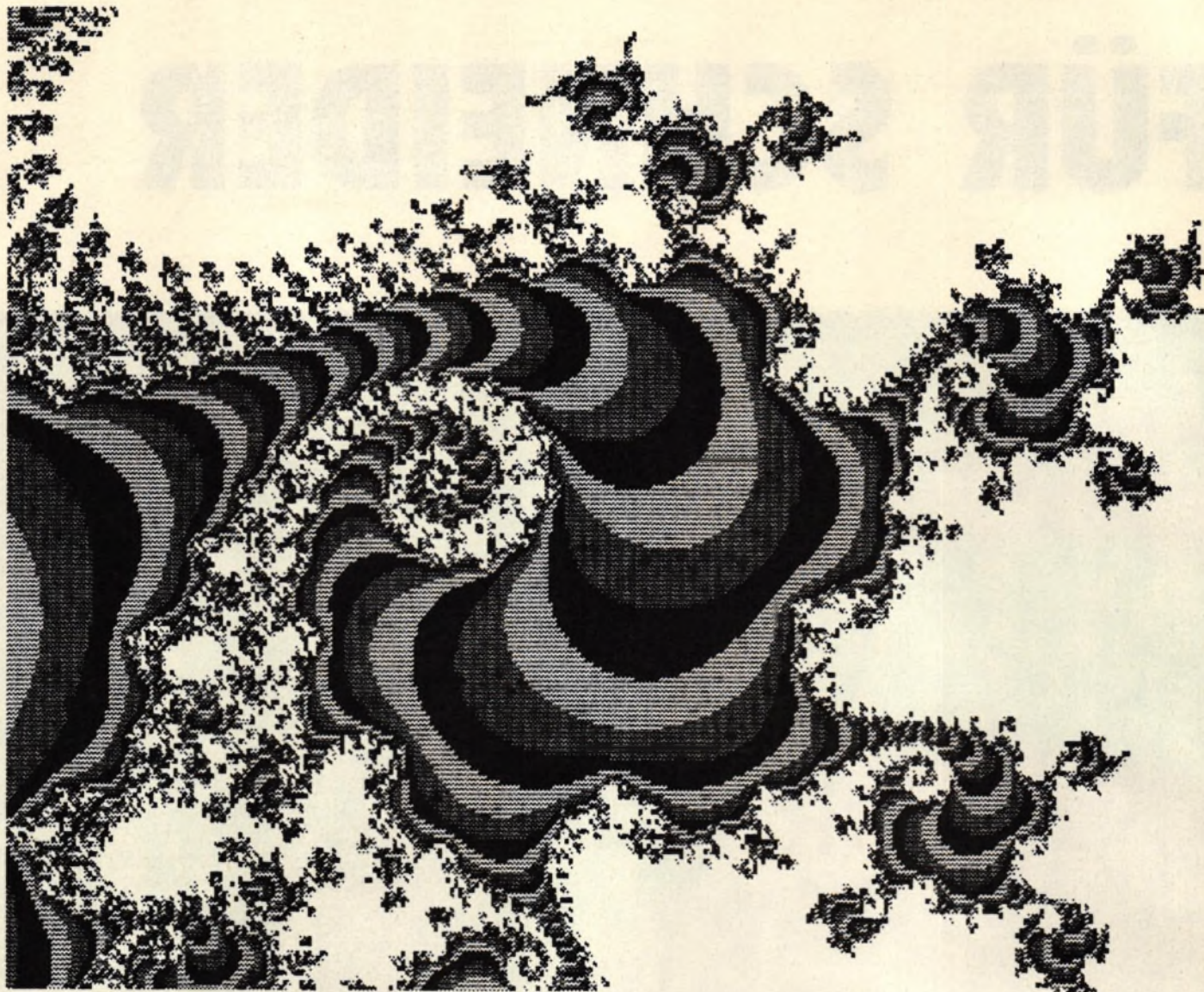
Schließlich haben Logo, CP/M und BASIC auch noch einiges zu bieten. (tb)

# APFELMÄNNCHEN



# FÜR SCHNEIDER





**Die Mandelbrotmenge wird als das komplizierteste mathematische Objekt bezeichnet. Erzeugt wird sie jedoch von einem einfachen Algorithmus. Auf dem Bildschirm dargestellt, ergibt sich eine unzählbare Anzahl von Mustern, welche, mit geeigneten Farben versehen, durchaus als Computerkunst aufgefasst werden können.**

So einfach der Algorithmus ist, so langsam wird er auch vom Computer abgearbeitet. Also etwas Geduld sollten Sie schon aufbringen. Der Algorithmus läßt sich etwa wie folgt darstellen:

```
10 c = eine komplexe Zahl
20 c = c2 + c
30 if c < = 2 then 20
```

Die Iteration wird die Zahl  $c$  sehr schnell, ohne absehbare Grenze, ansteigen lassen. Es gibt jedoch eine Menge von Zahlen, welche sich anders verhalten. Zum Beispiel die reellen Zahlen zwischen  $-2$  und  $0,25$ . Solche Zahlen sind Elemente der Mandelbrotmenge.

---

## Die Entstehung

---

Um nun eine dieser Grafiken zu erzeugen, berechnet man für jeden Punkt des Bildschirms eine Farbe. Ich habe mich in meinen Programmen für MODE 1 entschieden, aber man kann die Äpfelmännchen natürlich auch in MODE 0 mit bis zu 15 Farben erzeugen. Die Berechnung geht dann auch schneller, da ja nur noch die Hälfte der Punkte berechnet werden muß.

Ein Punkt erhält eine Farbe, wenn er nach einer bestimmten Anzahl (Tiefe) von Durchläufen ei-

ne bestimmte Grenze nicht überschreitet. Überschreitet er diese Grenze, so wird mit der bis dahin erreichten Tiefe die Farbe nach folgendem Algorithmus berechnet:

$$\text{Farbe} = (\text{Tiefe} \text{ MOD } \text{'Anz. mögl. Farben'}) + 1$$

In MODE 1 stehen drei Farben zur Verfügung, da alle Punkte, welche keine Farbe erhalten, schwarz (Farbe 0) bleiben. Die Tiefe kann willkürlich gewählt werden. Als Grenze wurde die Acht gewählt, Sie können jedoch auch mit diesem Wert experimentieren.

Komplexe Zahlen setzen sich aus zwei Komponenten zusammen, ei-



nem Realteil und einem Imaginärteil. Der Realteil besteht aus einer reellen Zahl. Der Imaginärteil besteht aus dem Produkt einer reellen Zahl und der Wurzel aus -1. Zur Darstellung verwendet man die Formel 'a + ib', wobei i die Wurzel aus -1 ist.

Komplexe Zahlen lassen sich einfach in einem Koordinatensystem darstellen, indem man den Realteil auf der einen und den Imaginärteil auf der anderen Achse abträgt. Um nun die Größe ei-

ner komplexen Zahl zu ermitteln, bestimmt man einfach die Länge der Strecke vom Ursprung (0+i0) des Koordinatensystems bis zu dem die Zahl beschreibenden Punkt. Nach Pythagoras ergibt sich dann:

$$\text{Länge} = \text{sqr}(a^2 + b^2)$$

Das Quadrat einer komplexen Zahl erhält man auch auf recht einfache Weise. Man rechnet:

$$(a + ib)^2 = a^2 + 2aib + i^2b^2$$

Da  $i^2 = -1$ , können wir schreiben:

$$(a + ib)^2 = a^2 - b^2 + 2iab$$

Jetzt ist  $a^2 - b^2$  der Realteil und  $2iab$  der Imaginärteil. Um komplexe Zahlen zu addieren, addiert man einfach Real- und Imaginärteil getrennt.

```

100 'Apfelmaennchen fuer den CPC 464
110 '(c) 1985
120 ' Thomas Barndt
130 '
140 'Der Code wurde durch den Software-Team
150 'Basic-Compiler erzeugt und mit dem
160 'Hexladergenerator weiterbearbeitet.
170 '
180 PRINT"PLEASE WAIT ..."
190 MEMORY &A04B
200 DEFSTR c
210 DEFINT i,j
220 DIM c(55)
230 c( 0)=""21 00 00 39 22 4C A0 21 01 00 7D CD 0
    E BC 21 00 00 EB 21 00"
240 DATA 1227
250 c( 1)=""00 7B 45 4D CD 32 BC 21 01 00 EB 21 0
    E 00 7B 45 4D CD 32 BC"
260 DATA 1740
270 c( 2)=""21 02 00 EB 21 14 00 7B 45 4D CD 32 B
    C 21 03 00 EB 21 1A 00"
280 DATA 1365
290 c( 3)=""7B 45 4D CD 32 BC CD 6C BB 21 03 00 7
    D CD 90 BB 3E 00 32 21"
300 DATA 2054
310 c( 4)=""AC C3 18 A1 21 20 20 20 20 41 70 6
    6 65 6C 6D 61 65 6E 6E"
320 DATA 1760
330 c( 5)=""63 68 65 6E 20 62 79 20 54 48 42 43 5
    3 20 31 39 38 35 21 F6"
340 DATA 1595
350 c( 6)=""A0 CD B8 A4 CD AE A4 3E 00 32 21 AC C
    D AE A4 3E 00 32 21 AC"
360 DATA 2433
370 c( 7)=""CD AE A4 3E 00 32 21 AC CD AE A4 3E 0
    0 32 21 AC C3 48 A1 06"
380 DATA 2154
390 c( 8)=""58 6D 69 6E 20 3A 21 41 A1 CD B8 A4 3
    E 00 21 A4 AC 77 32 21"
400 DATA 1947
410 c( 9)=""AC CD 3A BD CD AE A4 21 01 A0 CD DB A
    4 3E 00 32 21 AC C3 72"
420 DATA 2575
430 c(10)=""A1 06 58 6D 61 78 20 3A 21 6B A1 CD B
    8 A4 3E 00 21 A4 AC 77"
440 DATA 2075
450 c(11)=""32 21 AC CD 3A BD CD AE A4 21 06 A0 C
    D DB A4 3E 00 32 21 AC"
460 DATA 2354
470 c(12)=""C3 9C A1 06 59 6D 69 6E 20 3A 21 95 A
    1 CD B8 A4 3E 00 21 A4"
480 DATA 2176
490 c(13)=""AC 77 32 21 AC CD 3A BD CD AE A4 21 0
    B A0 CD DB A4 3E 00 32"
500 DATA 2445
510 c(14)=""21 AC C3 C6 A1 06 59 6D 61 78 20 3A 2
    1 BF A1 CD B8 A4 3E 00"
520 DATA 2270
530 c(15)=""21 A4 AC 77 32 21 AC CD 3A BD CD AE A
    4 21 10 A0 CD DB A4 3E"
540 DATA 2597
550 c(16)=""00 32 21 AC C3 F0 A1 06 54 69 65 66 6
    5 3A 21 E9 A1 CD B8 A4"

```

```

560 DATA 2388
570 c(17)=""3E 00 21 A4 AC 77 32 21 AE CD 3A BD C
    D AE A4 CD CB A4 22 15"
580 DATA 2427
590 c(18)=""A0 21 06 A0 CD EF A4 EB 21 01 A0 EB C
    D 5B BD EB 21 17 A0 CD"
600 DATA 2772
610 c(19)=""3D BD 21 17 A0 CD EF A4 EB C3 2F A2 0
    0 00 80 1F 89 21 2A A2"
620 DATA 2246
630 c(20)=""EB CD 64 BD EB 21 17 A0 CD 3D BD 21 1
    0 A0 CD EF A4 EB 21 0B"
640 DATA 2731
650 c(21)=""A0 EB CD 5B BD EB 21 1C A0 CD 3D BD 2
    1 1C A0 CD EF A4 EB C3"
660 DATA 3050
670 c(22)=""61 A2 00 00 00 47 8B 21 5C A2 EB CD 6
    4 BD EB 21 1C A0 CD 3D"
680 DATA 2204
690 c(23)=""BD 21 01 A0 EB 21 21 A0 CD 3D BD 21 1
    0 A0 EB 21 26 A0 CD 3D"
700 DATA 2240
710 c(24)=""BD 21 00 00 22 2B A0 EB 21 8F 01 EB C
    D C4 BD CA AC A2 F5 21"
720 DATA 2510
730 c(25)=""02 00 11 00 00 CD C4 BD FA A8 A2 F1 F
    2 7D A4 C3 AC A2 F1 FA"
740 DATA 2981
750 c(26)=""7D A4 00 21 00 00 22 2D A0 EB 21 7F 0
    2 EB CD C4 BD CA D6 A2"
760 DATA 2361
770 c(27)=""F5 21 02 00 11 00 00 CD C4 BD FA D2 A
    2 F1 F2 4D A4 C3 D6 A2"
780 DATA 2804
790 c(28)=""F1 FA 4D A4 00 21 00 00 22 2F A0 C3 E
    5 A2 00 00 00 28 00 21"
800 DATA 1665
810 c(29)=""E0 A2 EB 21 31 A0 CD 3D BD C3 F7 A2 0
    0 00 00 28 00 21 F2 A2"
820 DATA 2399
830 c(30)=""EB 21 36 A0 CD 3D BD C3 09 A3 00 00 0
    0 28 00 21 04 A3 EB 21"
840 DATA 1812
850 c(31)=""3B A0 CD 3D BD C3 1B A3 00 00 00 28 0
    0 21 16 A3 EB 21 40 A0"
860 DATA 1809
870 c(32)=""CD 3D BD 21 3B A0 CD EF A4 EB 21 40 A
    0 EB CD 5B BD EB 21 45"
880 DATA 2861
890 c(33)=""A0 CD 3D BD 2A 2F A0 EB 2A 15 A0 EB C
    D C4 BD F2 03 A4 21 45"
900 DATA 2658
910 c(34)=""A0 EB C3 54 A3 00 00 00 00 84 21 4F A
    3 EB CD 6A BD F2 03 A4"
920 DATA 2388
930 c(35)=""C3 66 A3 00 00 00 00 82 21 61 A3 CD E
    F A4 EB 21 36 A0 EB CD"
940 DATA 2413
950 c(36)=""61 BD EB 21 36 A0 CD 3D BD 21 36 A0 C
    D EF A4 EB 21 31 A0 EB"
960 DATA 2790
970 c(37)=""CD 61 BD EB 21 36 A0 CD 3D BD 21 36 A
    0 CD EF A4 EB 21 26 A0"

```

Damit jeder Besitzer eines Schneiders die Apfelmännchen erzeugen kann, veröffentlichten wir drei Versionen. Die BASIC-Version sollte auf allen Schneidern laufen. Ebenso die Pascal-Version, wenn Sie ein grafikfähiges Pascal besitzen.

Das Maschinensprachprogramm wurde mit dem BASIC-Compiler (Software-Team) für den CPC 464 erzeugt und läuft auch nur auf diesem. Sie läuft wesentlich schneller als die reine BASIC-Version und sollte, wenn Ihr Pascal

nicht schneller ist, zur Erzeugung von Apfelmännchen mit grösserer Tiefe verwendet werden, da diese oft mehrere Stunden in Anspruch nehmen. Dabei ist unbedingt darauf zu achten, daß die ersten vier einzugebenden Werte *mit* Dezimalpunkt eingegeben werden. Die Tiefe muß *ohne* Punkt eingegeben werden.

Die Farben können im BASIC- und im Pascalprogramm leicht an einen Farbmonitor angepasst werden. Im Maschinensprachprogramm ist dies nur möglich, wenn

Sie die Stellen, an welchen die Farben definiert werden, herausfinden und durch „poken“ verändern.

Um die „Urform“ aller Apfelmännchen zu erhalten, geben Sie folgende Koordinaten ein:

-0.7/ 2.1/ -1/ 1/ 30

Nun können Sie, wie mit einer Lupe, einzelne Ausschnitte dieses

```

980 DATA 2749
990 c(38)="EB CD 5B BD EB 21 36 A0 CD 3D BD 21 3
  B A0 CD EF A4 EB 21 40"
1000 DATA 2849
1010 c(39)="A0 EB CD 5B BD EB 21 31 A0 CD 3D BD 2
  1 31 A0 CD EF A4 EB 21"
1020 DATA 2930
1030 c(40)="21 A0 EB CD 5B BD EB 21 31 A0 CD 3D B
  D 21 31 A0 CD EF A4 EB"
1040 DATA 2930
1050 c(41)="21 31 A0 EB CD 61 BD EB 21 3B A0 CD 3
  D BD 21 36 A0 CD EF A4"
1060 DATA 2765
1070 c(42)="EB 21 36 A0 EB CD 61 BD EB 21 40 A0 C
  D 3D BD 2A 2F A0 23 22"
1080 DATA 2473
1090 c(43)="2F A0 C3 25 A3 2A 2F A0 EB 2A 15 A0 E
  B CD C4 BD C2 81 A4 21"
1100 DATA 2654
1110 c(44)="00 00 22 4A A0 21 21 A0 CD EF A4 EB 2
  1 17 A0 EB CD 5B BD EB"
1120 DATA 2505
1130 c(45)="21 21 A0 CD 3D BD 2A 4A A0 7D CD DE B
  B 2A 2D A0 EB 2A 2B A0"
1140 DATA 2423
1150 c(46)="CD EA BB 2A 2D A0 EB 21 02 00 CD AC B
  D 22 2D A0 C3 B3 A2 00"
1160 DATA 2484
1170 c(47)="21 01 A0 EB 21 21 A0 CD 3D BD 21 26 A
  0 CD EF A4 EB 21 1C A0"
1180 DATA 2405
1190 c(48)="EB CD 5B BD EB 21 26 A0 CD 3D BD 2A 2
  B A0 EB 21 02 00 CD AC"
1200 DATA 2533
1210 c(49)="BD 22 2B A0 C3 89 A2 00 C3 99 A4 2A 2
  F A0 EB 21 03 00 EB CD"
1220 DATA 2392
1230 c(50)="BB BD 22 4A A0 2A 4A A0 23 22 4A A0 C
  3 17 A4 2A 4C A0 F9 C9"
1240 DATA 2333
1250 c(51)="E5 21 56 C3 22 4E A0 CD AA A4 E1 C9 D
  F 4E A0 C9 3E 0D CD 9E"
1260 DATA 2880
1270 c(52)="A4 3E 0A C3 9E A4 7E B7 C8 23 47 7E 2
  3 C5 E5 CD 9E A4 E1 C1"
1280 DATA 2900
1290 c(53)="05 C2 BD A4 C9 21 A3 EC 22 4E A0 21 A
  4 AC CD AA A4 2A C2 B0"
1300 DATA 2777
1310 c(54)="C9 E5 21 A3 EC 22 4E A0 21 A4 AC CD A
  A A4 11 C2 B0 E1 C3 3D"
1320 DATA 2910
1330 c(55)="BD EB 21 C2 B0 3E 05 32 C1 B0 C3 3D B
  D 00 00 00 00 00 00"
1340 DATA 1758
1350
1360 RESTORE
1370 adr=&A0A2
1380 FOR i=0 TO 55
1390   c(i)="00"+c(i)
1400   sum=0:READ pruefsumme
1410   FOR j=1 TO 20
1420     wert=VAL("&"+MID$(c(i),3*j,2))
1430     POKE adr,wert
1440     adr=adr+1
1450     sum=sum+wert
1460   NEXT j
1470   IF sum<>pruefsumme THEN PRINT"Fehler in c (
  ";i;")":END
1480 NEXT i
1490 CALL &A0A2

100 'Apfelmaennchen
110 '(c) 1985
120 ' THOMAS BARNDT
150 MODE 1
160 DEFINT z,s,t,p
170 INK 0,0
180 INK 1,14
190 INK 2,20
200 INK 3,26
210 CLS:PEN 3
220 PRINT TAB(5)"Apfelmaennchen by THECS 1985"
230 PRINT:PRINT:PRINT
240 INPUT"Xmin :",xmin
250 INPUT"Xmax :",xmax
260 INPUT"Ymin :",ymin
270 INPUT"Ymax :",ymax
280 INPUT"Tiefe:",tmax
290 dx=(xmax-xmin)/319
300 dy=(ymax-ymin)/199
310 cx=xmin
320 cy=ymax
330 FOR zeile=0 TO 399 STEP 2
340   FOR spalte=0 TO 639 STEP 2
350     tiefe=0
360     xwert=0:ywert=0
370     xquad=0:yquad=0
380     WHILE (tiefe<tmax) AND (xquad+yquad<8)
390       ywert=2*xwert*ywert-cy
400       xwert=xquad-yquad-cx
410       xquad=xwert*xwert
420       yquad=ywert*ywert
430       tiefe=tiefe+1
440     WEND
450     IF tiefe=tmax THEN pcol=0 ELSE
460       fe=MOD 3)+1 pcol=(tie
470     PLOT spalte,zeile,pcol
480   NEXT
490   cx=xmin
500   cy=cy-dy
510 NEXT
520 WHILE INKEY$="" :WEND

```



Apfelmännchens vergrößern, indem Sie einfach die neuen Koordinaten schätzen und ausprobieren. Auffällig ist hier die Selbstähnlichkeit der entstehenden Figuren. Probieren Sie zum Beispiel einmal

den folgenden Ausschnitt aus der Spitze des Apfelmännchens:

1.67/1.86/-0.75/0.75/40

Deutlich ist hier das Apfelmännchen wiederzuerkennen. Bei der

Wahl eines neuen Ausschnitts sollten Sie darauf achten, daß Sie einen Rand der jeweils vorhergehenden Grafik vergrößern, da dort die schönsten Bilder entstehen. tb

```

1 PROGRAM apfelmaennchen(input,output);
2
3 VAR xmax,xmin,
4     ymax,ymin,
5     dx,dy,
6     cx,cy,
7     xwert,ywert,
8     xquad,yquad :real;
9     tiefe,
10    spalte,zeile,
11    punkte,
12    col,
13    tiefemax      :integer;
14
15 {ab hier werden die Prozeduren fuer die }
16 {Grafikbefehle eingeschlossen          }
17 {$F plot      .chn}
18 {$F mode      .chn}
19 {$F ink        .chn}
20 {$F pen        .chn}
21
22
23 BEGIN
24   mode(2);
25   ink(3,26,26);
26   ink(2,20,20);
27   ink(1,14,14);
28   ink(0,0,0);
29   pen(3);
30   writeln('Apfelmaennchen
31   by THBCS 1985');
32   write('xmin: ');read(xmin);
33   write('xmax: ');read(xmax);
34   write('ymin: ');read(ymin);
35   write('ymax: ');read(ymax);
36   write('Tiefe: ');read(tiefemax);
37   mode(1);

```

```

37   dx:=(xmax-xmin)/319;
38   dy:=(ymax-ymin)/199;
39   cx:=xmin;
40   cy:=ymax;
41   FOR zeile:=0 TO 199 DO
42     BEGIN
43       FOR spalte:=0 TO 320 DO
44         BEGIN
45           tiefe:=0;
46           xwert:=0; ywert:=0;
47           xquad:=0; yquad:=0;
48           WHILE (tiefe<tiefemax)

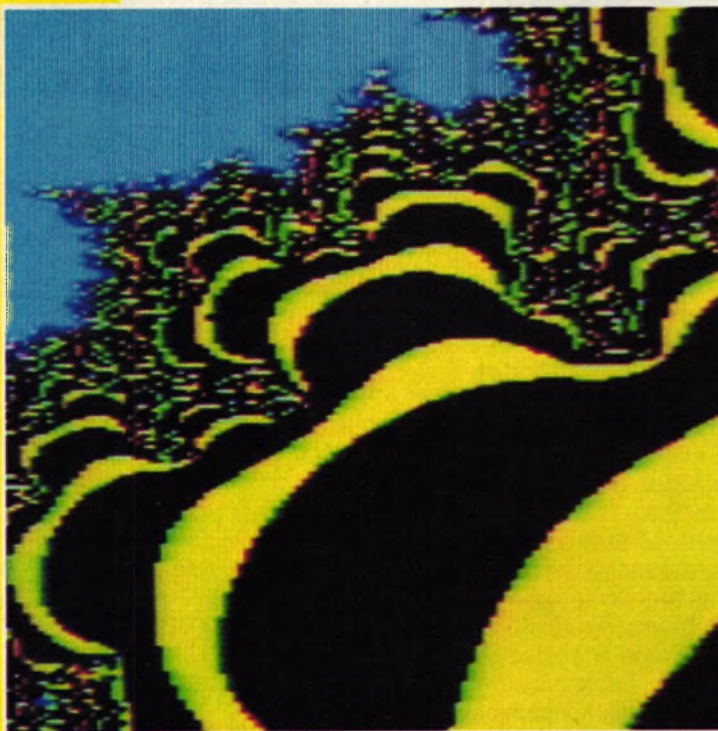
```

Xmin /Xmax /Ymin /Ymax /Tiefe

0.5665/0.5737/0.5602/0.5665/85

-0.4 /-0.35/ 0.1/ 0.2/ 100

0.7425/0.74825/0.09621/0.10067/150



```

49     BEGIN
50       ywert:=2*xwert*ywert-cy;
51       xwert:=xquad-yquad-cx;
52       xquad:=xwert*xwert;
53       yquad:=ywert*ywert;
54       tiefe:=tiefe+1
55     END;
56     IF tiefe=tiefemax THEN col:=0
57     ELSE col:=(tiefe MOD 3) +1;
58     cx:=cx+dx;
59     plot((spalte)*2,zeile*2,col);
60   END;
61   cx:=xmin;
62   cy:=cy-dy
63 END
64 END.

```

# Das fliegende Auge

## Das Super-Arcade-Spiel für Hubschrauber-Fans

Ziel des Spiels ist es, möglichst viele Männchen zu retten und dabei möglichst viel Geld zu verdienen. Man steuert dazu den Hubschrauber so neben das Männchen, daß dieses von vorne in den Hubschrauber einsteigen kann. Nach Betätigen des Feuerknopfes muß man dann zur Erde fliegen und das Männchen, wieder durch Betätigen des Feuerknopfes, aussteigen lassen. Wenn man die oben im Bild angezeigte Mindestanzahl von Männchen gerettet hat und sich keines mehr auf dem Dach befindet, wechselt das Bild. Die Schwierigkeitsstufe bestimmt die Zeit, die von der Aufnahme eines Männchens bis zum Erscheinen eines neuen verstreicht. Für jedes gerettete Männchen erhält man 200,— DM. Jeder zerstörte Hubschrauber kostet 1500,— DM. Das Spiel ist für einen Grünmonitor geschrieben, jedoch ist es für Farbmonitorbesitzer nicht schwer, in Zeile 120 die Farben zu ändern. Erfahrenere Programmierer können sich mit dem im Handbuch abgedruckten Text- und Windowplaner auch leicht neue Screens ausdenken. Dazu muß nur die Variable anzscreen erhöht werden und die entsprechenden Datazeilen eingefügt werden. Die Datazeilen für die Häuser bestehen jeweils aus den Koordinaten links, rechts, oben, unten,...(usw.). Die obere Zeile ist für

```

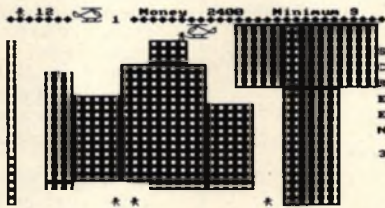
10 ' *****
*
20 ' **      Copyright by      *
*
30 ' **                Thomas Barndt      *
*
40 ' **                                &      *
*
50 ' **                Frank Thielen      *
*
60 ' *****
*
70 '
80 SYMBOL AFTER 199
90 GOSUB 1530:GOSUB 1430
100 ENV 1,15,8,15,1,1,0
110 MODE 1
120 INK 0,0:INK 1,26:INK 2,17:INK 3,22
130 PEN 1:BORDER 0:PAPER 0
140 LOCATE 10,12:PRINT"H E L I C O P T E R"
150 LOCATE 20,15:PRINT"by THBCS"
160 PEN 3
170 LOCATE 1,23:PRINT CHR$(164);" 4/85"
180 LOCATE 15,10:PRINT"N.Y.City"
190 FOR i=0 TO 2000:NEXT
200 CLS
210 FOR i=1 TO 24
220   READ inst$
230   PRINT inst$
240 NEXT
250 WHILE INKEY$<>" ":WEND
260 CLS
270 LOCATE 3,5:PRINT"Schwierigkeitsstufe
Zeit"
280 LOCATE 3,9:PRINT"          1          1
1 sec"

```

```

290 LOCATE 3,11:PRINT"          2
      9 sec"
300 LOCATE 3,13:PRINT"          3
      7 sec"
310 PEN 1
320 LOCATE 13,23:PRINT"Ihre Wahl ?"
330 '
340 '
350 '   *** Variablen initialisieren ***
360 '
370 DEFINT a-g,i-z
380 DEFSTR h
390 WHILE htest$>"3" OR htest$<"1":htest$=INKEY$
      :WEND
400 ON VAL(htest$) GOTO 410,420,430
410 zeit=550:GOTO 440
420 zeit=450:GOTO 440
430 zeit=350
440 CLS
450 '
460 DIM bild(39,25)
470 anzscreen=4
480 maxhub=3
490 FOR screen=0 TO anzscreen
500   FOR i=0 TO 7
510     READ li(i,screen)
520     READ re(i,screen)
530     READ ob(i,screen)
540     READ un(i,screen)
550   NEXT i
560 NEXT screen
570 FOR i=0 TO anzscreen
580   READ x(i),y(i)
590 NEXT
600 FOR i=0 TO anzscreen
610   READ xm(i),ym(i)
620 NEXT
630 FOR i=0 TO anzscreen
640   READ anz(i)
650 NEXT
660 screen=0:anzhub=0:konto=0
670 h$=hub$(2)
680 htex$="SCREEN"
690 '
700 '   *** Screen aufbauen ***
710 '
720 DI
730 IF screen>anzscreen GOTO 2660
740 ERASE bild
750 DIM bild(39,25)
760 FOR i=0 TO 34
770   bild(i,1)=-1
780   bild(i,2)=-1
790 NEXT
800 FOR j=0 TO 25
810   bild(0,j)=-1
820   bild(34,j)=-1
830 NEXT
840 CLS
850 '
860 '   Ausgabe der Informationen
870 PEN 3
880 LOCATE 1,2:PRINT STRING$(40,CHR$(227))
890 PEN 1
900 LOCATE 2,1:PRINT CHR$(250):gerettet
910 LOCATE 8,1:PRINT hub$(2):anzhub+1
920 LOCATE 15,1:PRINT"Money ";konto
930 LOCATE 29,1:PRINT"Minimum";anz(screen)
940 FOR i=2 TO 12 STEP 2
950   LOCATE 40,i+4
960   PRINT MID$(htex$,i/2,1)
970   LOCATE 40,i+5
980   PRINT " "
990 NEXT
1000 LOCATE 40,19:PRINT USING"#";screen+1
1010 '
1020 '   Haeuser aufbauen
1030 FOR z=0 TO 7
1040   IF z>3 THEN PEN 3:ELSE PEN 2

```



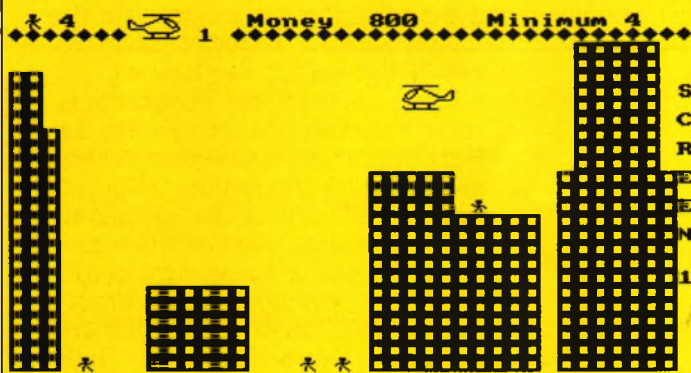
die dunklen und die untere für die hellen Häuser. In jeder Zeile stehen 16 Zahlen für vier Häuser.

Interessant an der Programmierung ist sicher die Entstehung des Hubschraubers. Dieser ist einfach in einer Stringvariablen abgelegt und besteht aus sechs selbstdefinierten Zeichen und vier Steuerzeichen. Zuerst kommen drei selbstdefinierte Zeichen, dann einmal Cursor nach unten (CHR\$(10)) und dreimal Cursor nach links (CHR\$(8)). Darauf folgen wieder drei sichtbare Zeichen, wodurch der Hubschrauber insgesamt scheinbar aus sechs Zeichen besteht. Die relativ hohe Geschwindigkeit des Hubschraubers resultiert aus der kurzgehaltenen Hauptschleife ab Zeile 1200 und der bevorzugten Verwendung von Integervariablen. Integervariablen benötigen weniger Speicherplatz als Realvariablen und können daher vom Rechner schneller verarbeitet werden. Das Männchen wird in einem durch einen Interrupt aufgerufenen Unterprogramm auf das Dach gesetzt.

```

1050 links=li(z,screen)
1060 rechts=re(z,screen)
1070 oben=ob(z,screen)
1080 unten=un(z,screen)
1090 GOSUB 2560
1100 NEXT z
1110 BORDER 10
1120 FOR i=1 TO 34
1130   bild(i,25)=0
1140 NEXT
1150 PEN 1
1160 x=x(screen):y=y(screen)
1170 GOSUB 1930
1180 EI
1190 '
1200 '   *****
1210 '   **** HAUPTPROGRAMM ****
1220 '   *****
1230 '
1240 LOCATE x,y:PRINT h$
1250 x1=x:y1=y
1260 jo=JOY(0)
1270 v=(1 AND jo=2)-(1 AND jo=1)+(1 AND jo=6)-(1
      AND jo=5)+(1 AND jo=10)-(1 AND jo=9)
1280 w=(1 AND jo=8)-(1 AND jo=4)+(1 AND jo=9)-(1
      AND jo=5)+(1 AND jo=10)-(1 AND jo=6)
1290 hub$(1)=h$
1300 h$=hub$(w+1)
1310 y=y+v
1320 x=x+w
1330 IF y>24 THEN y=24
1340 SOUND 1,700+v*300,8,0,1,,1
1350 IF bild(x,y) OR bild(x+1,y) OR bild(x+2,y) O
      R bild(x,y+1) OR bild(x+1,y+1) OR bild(x+2,y
      +1) THEN GOTO 2300
1360 IF jo=16 THEN 2110
1370 LOCATE x1,y1:PRINT hub$
1380 GOTO 1240
1390 '
1400 '
1410 '   **** HUBSCHRAUBER ZUSAMMENSETZEN ****
1420 '
1430 ht$=CHR$(10)+STRING$(3,CHR$(8))
1440 hub$=STRING$(3,CHR$(32))+ht$+STRING$(3,CHR$(
      32))
1450 hub$(2)=CHR$(200)+CHR$(201)+CHR$(202)+ht$+CH
      R$(203)+CHR$(204)+CHR$(205)
1460 hub$(0)=CHR$(207)+CHR$(208)+CHR$(209)+ht$+CH
      R$(210)+CHR$(211)+CHR$(212)

```



```

1470 *
1480 *      *** EXPLOSION ZUSAMMENSETZEN ***
1490 *
1500 hx$(0)=CHR$(213)+CHR$(214)+CHR$(215)+ht$+CHR
$(216)+CHR$(217)+CHR$(218)
1510 hx$(1)=CHR$(219)+CHR$(220)+CHR$(221)+ht$+CHR
$(222)+CHR$(223)+CHR$(224)
1520 RETURN
1530 *
1540 *
1550 *      **** SYMBOLE DEFINIEREN ****
1560 *
1570 *      HUBSCHRAUBER RECHTS
1580 SYMBOL 200,1,0,0,192,160,144,143,128
1590 SYMBOL 201,255,1,3,12,48,192,0,0
1600 SYMBOL 202,255,0,224,88,68,34,18,13
1610 SYMBOL 203,127,0,0,0,0,1,0,0
1620 SYMBOL 204,0,128,127,16,16,255,0,0
1630 SYMBOL 205,1,6,248,32,33,254,0,0
1640 *
1650 *      HAUS
1660 SYMBOL 206,255,255,195,195,195,195,255,255
1670 *
1680 *      HUBSCHRAUBER LINKS
1690 SYMBOL 207,255,0,7,26,34,68,72,176
1700 SYMBOL 208,255,128,192,48,12,3,0,0
1710 SYMBOL 209,128,0,0,3,5,9,241,1
1720 SYMBOL 210,128,96,31,4,132,127,0,0
1730 SYMBOL 211,0,1,254,8,8,255,0,0
1740 SYMBOL 212,254,0,0,0,0,128,0,0
1750 *
1760 *      EXPLOSION
1770 SYMBOL 213,8,68,34,17,8,4,0,1
1780 SYMBOL 214,0,0,0,24,36,129,153,36
1790 SYMBOL 215,32,72,144,32,64,0,0,159
1800 SYMBOL 216,249,0,0,2,4,9,18,4
1810 SYMBOL 217,36,153,129,36,24,0,0,0
1820 SYMBOL 218,128,0,32,16,136,68,34,16
1830 SYMBOL 219,4,18,9,4,2,0,0,124
1840 SYMBOL 220,0,0,0,0,24,36,66,153
1850 SYMBOL 221,16,34,68,136,16,32,0,0
1860 SYMBOL 222,0,0,4,8,17,34,68,8
1870 SYMBOL 223,153,66,36,24,0,0,0,0
1880 SYMBOL 224,62,0,0,64,32,144,72,32
1890 RETURN
1900 *
1910 *      *** MANN AUF'S DACH ***
1920 *
1930 LOCATE #1,xm(screen),ym(screen):PRINT #1,CHR
$(251)
1940 bild(xm(screen),ym(screen))=-1
1950 mann=-1
1960 RETURN
1970 *
1980 *      *** Mann aufnehmen ***
1990 *
2000 *
2010 IF inhalt THEN GOSUB 2950:RETURN
2020 SOUND 3,500,30,12
2030 mann=0:inhalt=-1
2040 LOCATE #1,xm(screen),ym(screen):PRINT #1,CHR
$(32)
2050 AFTER zeit GOSUB 1930
2060 bild(xm(screen),ym(screen))=0
2070 RETURN
2080 *
2090 *      *** Mannpos. testen ***
2100 *
2110 IF (x=xm(screen)-3 AND h$=hub$(2) OR x=xm(sc
reen)+1 AND h$=hub$(0)) AND y=ym(screen)-1 A
ND mann THEN GOSUB 2000
2120 IF y<>24 OR NOT inhalt THEN 1370
2130 IF h$=hub$(0) THEN xm=x-1:ELSE xm=x+3
2140 IF bild(xm,y+1)=-1 THEN GOSUB 2950:GOTO 1370

```

```

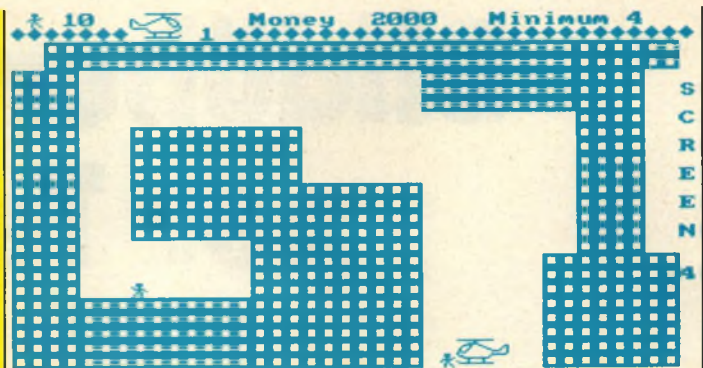
2150 *
2160 *      *** Mann aussetzen ***
2170 *
2180 LOCATE xm,y:PRINT CHR$(22)+CHR$(1)
2190 LOCATE xm,y+1:PRINT CHR$(250);
2200 bild(xm,y+1)=-1
2210 PRINT CHR$(22)+CHR$(0);
2220 inhalt=0
2230 gerettet=gerettet+1
2240 konto=konto+200
2250 LOCATE 3,1:PRINT gerettet
2260 LOCATE 21,1:PRINT konto
2270 IF gerettet>=anz(screen) AND NOT mann THEN s
creen=screen+1:GOTO 720
2280 GOTO 1370
2290 *
2300 *      **** EXPLOSION ****
2310 *
2320 LOCATE x1,y1:PRINT hx$(1)
2330 inhalt=0
2340 SOUND 2,2000,50,15,,5
2350 LOCATE x1,y1:PRINT hx$(0)
2360 SOUND 3,700,40,15,,1
2370 LOCATE x1,y1:PRINT hx$(1)
2380 ex=0
2390 FOR n=15 TO 1 STEP -1
2400   SOUND 1,426,4,10,,n
2410   LOCATE x1,y1:PRINT hx$(ex)
2420   ex=1-ex
2430 NEXT
2440 LOCATE x1,y1:PRINT hub$
2450 anzhub=anzhub+1
2460 konto=konto-1500
2470 LOCATE 21,1:PRINT konto
2480 IF anzhub=maxhub THEN 2640
2490 LOCATE 11,2:PRINT anzhub+1
2500 x=x(screen):y=y(screen)-1
2510 GOTO 1240
2520 *
2530 *
2540 *      **** HAEUSER SETZEN ****
2550 *
2560 FOR i=links TO rechts
2570   FOR j=oben TO unten
2580     bild(i,j)=-1
2590     LOCATE i,j
2600     PRINT CHR$(206)
2610   NEXT
2620 NEXT
2630 RETURN
2640 *
2650 *
2660 *      **** GAME OVER ****
2670 *
2680 *
2690 DI
2700 BORDER 10,15
2710 LOCATE 12,11
2720 PRINT"GAME OVER"
2730 GOSUB 2950:GOSUB 2950:GOSUB 2950
2740 FOR i=1 TO 3000:NEXT
2750 CLS
2760 BORDER 0
2770 LOCATE 3,8:PRINT"Wollen Sie weiterspielen (j
/n) ?"
2780 *
2790 htest$=""
2800 WHILE htest$="" :htest$=INKEY$:WEND
2810 htest$=LOWER$(htest$)
2820 IF htest$="j" THEN screen=0:anzhub=0:gerette
t=0:GOTO 700
2830 IF htest$<>"n" THEN 2790
2840 *
2850 CLS
2860 PRINT:PRINT:PRINT:PRINT
2870 IF konto>0 THEN 2900
2880 PRINT"Bitte ueberweisen Sie"ABS(konto)"DM au
f"
2890 PRINT:PRINT"Konto.Nr.: 14519 Blz.: 37069600
":GOTO 2910
2900 PRINT"Lassen Sie sich"konto"DM auszahlen !"
2910 FOR i=0 TO 3000:NEXT
2920 LOCATE 1,20
2930 END
2940 *
2950 FOR i=0 TO 30
2960   SOUND 2,100+i*20,3,12
2970 NEXT
2980 RETURN
2990 *
3000 DATA"Sie verdienen Ihr Geld als Pilot eines"
3010 DATA"Rettungshubschraubers in New York !"
3020 DATA""

```

```

3030 DATA""
3040 DATA"Ihre Aufgabe ist es, die Menschen von"
3050 DATA"den Daechern zu holen und auf der Erde"
3060 DATA"abzusetzen."
3070 DATA"Dieses erreichen Sie durch das"
3080 DATA"Betaetigen des Feuerknopfes."
3090 DATA""
3100 DATA"Wenn Sie die erforderliche"
3110 DATA"Mindestanzahl von Menschen gerettet"
3120 DATA"haben und sich Niemand mehr auf dem"
3130 DATA"Dach befindet, wechselt die Szene."
3140 DATA""
3150 DATA"Sie erhalten fuer jede gerettete Seele"
3160 DATA"200,-DM Praemie."
3170 DATA""
3180 DATA"Jeder zerstoerte Helicopter kostet"
3190 DATA"1500,-DM."
3200 DATA""
3210 DATA""
3220 DATA""
3230 DATA"                <Leertaste>"
3240 "
3250 "    *** Werte fuer Haeuser ***
3260 "
3270 DATA 9,14,20,25,34,38,3,11,22,31,15,25,1,3,9
      ,25
3280 DATA 33,39,12,25,1,2,5,8,22,26,12,19,35,37,6
      ,8
3290 "
3300 DATA 1,2,7,25,7,21,18,25,34,38,4,25,11,16,7,
      17
3310 DATA 33,39,11,15,15,20,9,19,9,12,10,17,10,12
      ,6,9
3320 "
3330 DATA 1,1,5,25,8,15,12,22,13,21,8,12,30,35,11
      ,25
3340 DATA 5,7,9,23,16,26,13,23,16,19,5,7,25,39,3,
      10
3350 "
3360 DATA 1,4,5,25,15,24,13,25,25,33,5,7,32,39,18
      ,25

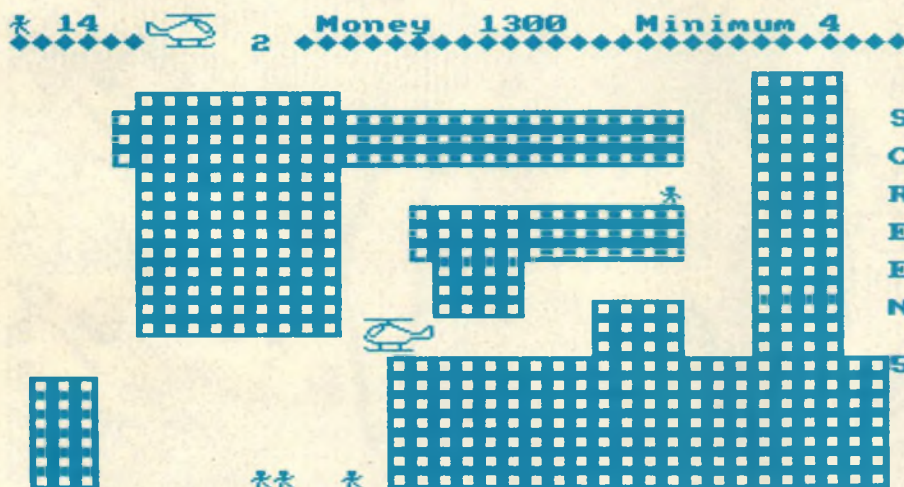
```



```

3370 DATA 3,16,21,25,3,39,3,4,8,17,9,16,34,37,5,2
      0
3380 "
3390 DATA 1,3,20,25,7,15,5,17,18,39,19,25,19,30,1
      1,13
3400 DATA 6,30,6,8,20,23,12,16,27,30,16,25,34,37,
      4,21
3410 "
3420 "    *** Startpos.Hubschrauber ***
3430 "
3440 DATA 6,24,25,24,21,5,25,24,14,24
3450 "
3460 "    *** Dachpos. Mann ***
3470 "
3480 DATA 28,14,9,9,19,4,8,20,30,10
3490 "
3500 "    *** Maenner pro Screen ***
3510 "
3520 DATA 1,1,1,1,1,1,1,1,1
3530 DATA 7,15,24,30,38
3540 "

```



## Variablentabelle:

**inst\$:** Instruktionen;  
**anzscreen:** Anzahl der Screens (0-4);  
**screen:** Momentane Screen;  
**maxhub:** Anzahl der Hubschrauber;  
**anzhub:** Nummer des momentanen Hubschraubers;  
**li, re, ob, un:** Koordinaten der Häuser;  
**konto:** Kontostand;  
**hub\$(0):** nach links fliegender Hubschrauber;  
**hub\$(2):** nach rechts fliegender Hubschrauber;  
**h\$:** Momentaner Hubschrauber;  
**hx\$:** Explosion;  
**bild:** Boolesche Variable für Hindernis;  
**gerettet:** Anzahl der geretteten Männchen;  
**v:** vertikale Bewegung;  
**w:** waagerechte Bewegung;  
**x,y:** aktuelle Hubschrauberkoordinaten;  
**xly:** alte Koordinaten;  
**xm, ym:** Koordinaten für Mann auf Dach;  
**mann:** Boolesche Variable für Mann auf Dach;  
**inhalt:** Boolesche Variable für Mann in Hubschrauber;  
**zeit:** Zeiteinheit für Schwierigkeitsgrad

# Bildübertragung per Telefon

Datenfernübertragung (DFÜ) per Telefon ist seit einiger Zeit ein Thema für jedermann. Leider besitzen die Schneidercomputer keine serielle Schnittstelle zum Anschluß eines Akustikkopplers. Findige Leute haben jedoch herausgefunden, daß man auch über den Centronics-Port Daten empfangen kann, nämlich über den BUSY-Eingang, über welchen der Drucker normalerweise mitteilt, ob er empfangsbereit ist oder nicht. Mit einem selbstgebastelten Kabel, der entsprechenden Software und einem Akustikkoppler besitzt man nun ein voll DFÜ-fähiges Gerät, mit dem man zum Beispiel auch Bilder übertragen kann.

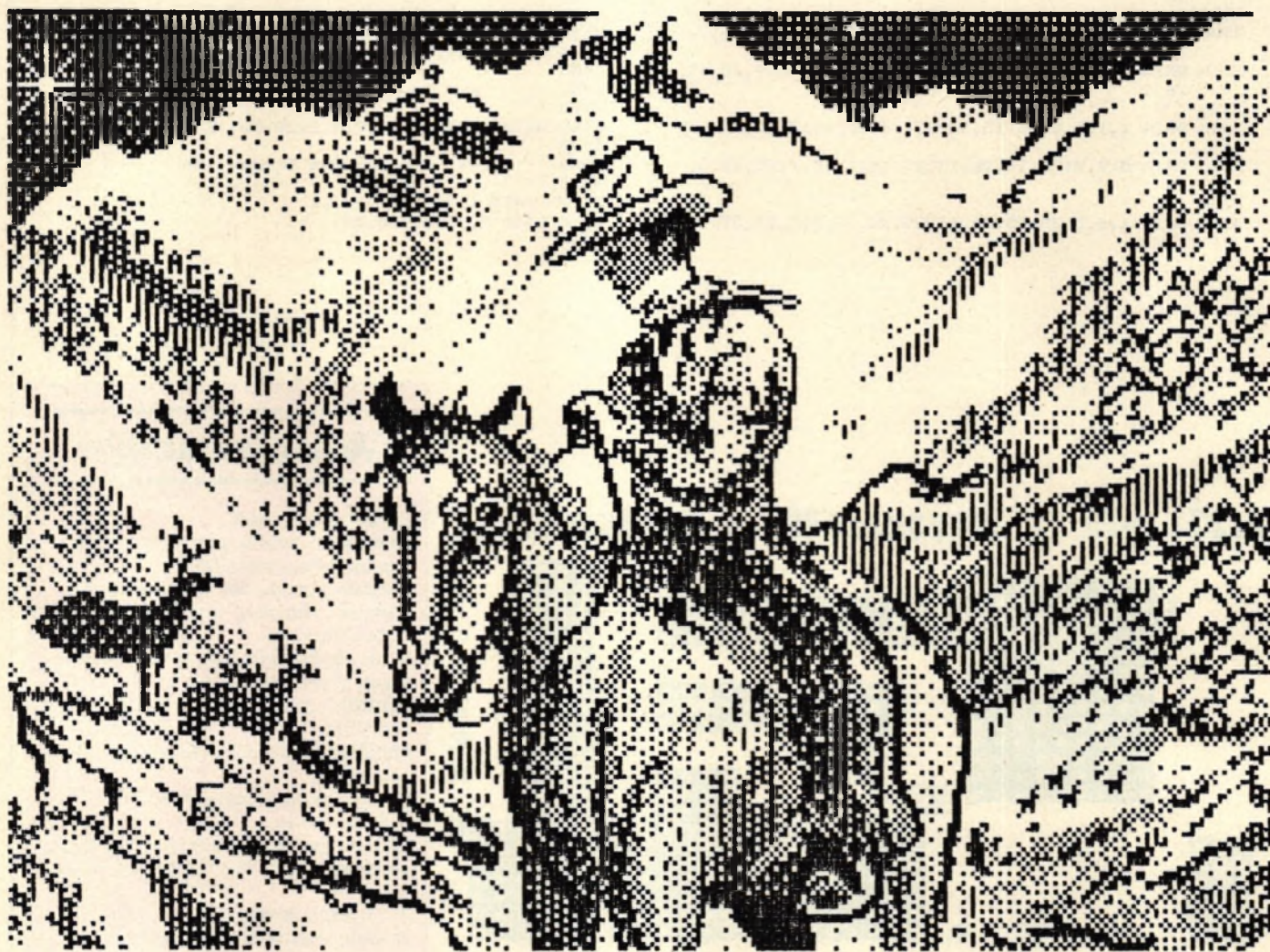


Bild vom Atari 520 ST auf CPC 464 übertragen

Zu diesem Zweck hat der WDR-Computerclub einen Standard zur Bildcodierung entwickelt, mit dessen Hilfe man Bilder von beliebigen Rechnern austauschen kann. Dieser Standard hat sehr schnell eine weite Verbreitung gefunden. So gibt es inzwischen außer der

WDR-Mailbox auch andere Mailboxen, welche Bilder in ihrem Angebot haben.

Man kann ein Bild nämlich nicht so einfach senden. Ein erstes Hindernis ist die Bilddarstellung bei verschiedenen Rechnern. Manche Rechner haben ihren Koordinaten-

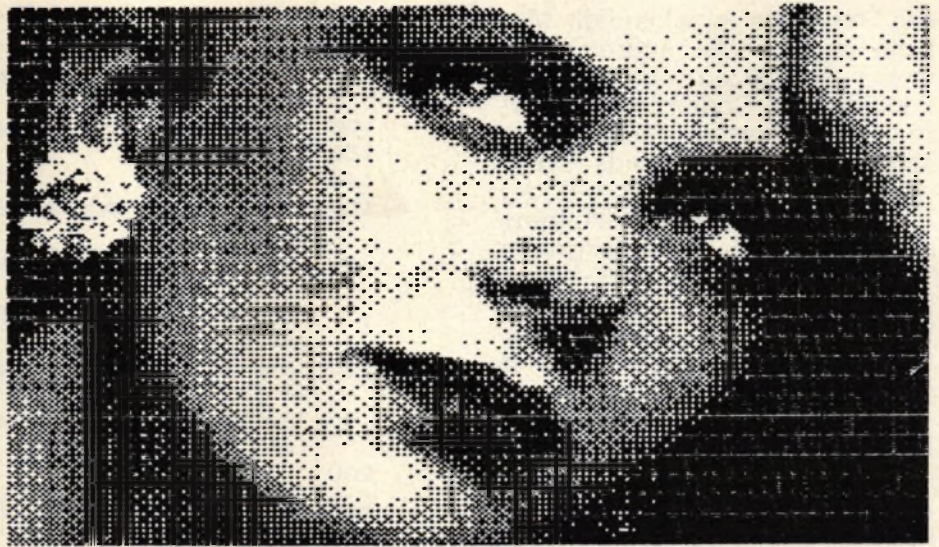
ursprung links oben auf dem Bildschirm, andere wiederum, wie z.B. auch der Schneider, bevorzugen die linke untere Ecke. Außerdem muß man darauf achten, daß man der Gegenstelle keine Kontrollzeichen schickt, da diese sonst eventuell den Kontakt abbricht.



## Die Codierung

Man hat sich also als erstes überlegt, zu jedem berechneten Code 32 zu addieren, so daß keine Kontrollzeichen (unter 32) mehr entstehen können. Nun ist der ASCII-Code, welcher zur Übermittlung verwendet wird, jedoch ein 7 Bit-Code und erlaubt die Darstellung von Codes bis zu 127. Addiert man jedoch zu Codes ab 96 die 32 hinzu, so landet man schon bei Codes, die man nicht mehr senden kann. Also entschloß man sich, auf ein Bit zu verzichten und nur jeweils sechs Punkte zu einem Code zusammenzufassen. Sind jetzt zum Beispiel sechs nebeneinander liegende Punkte auf dem Bildschirm gesetzt, so erhält man folgende Binärzahl: 111111. Also für jeden gesetzten Punkt setzt man eine Eins. Dezimal wäre dies eine 63, plus 32 ergibt 95; also einen Code, den wir senden können. Für sechs ungesetzte (also schwarze) Punkte ergibt sich  $0 + 32 = 32$  als zu sendender Code. Dies entspricht im ASCII-Code dem Leerzeichen.

Man braucht jetzt also nur noch den Bildschirm von links oben ze-



*High Resolution per Telefon, auch das ist möglich*

turn(CR)) in unsere Datei übernommen wird. Zum Abschluß der Datei fügen wir noch einmal zwei CRs in die Datei ein.

Damit das Ganze etwas einfacher zu decodieren ist, wird der Datei noch ein Kopf vorangestellt. Ähnlich einem Brief. Dieser könnte zum Beispiel so aussehen:

```
BILD
C-64-Schneegestöber
H320
V200
```

den folgenden Zeilen bleiben frei für eine spätere Erweiterung. Erst dann folgt das eigentliche Bild.

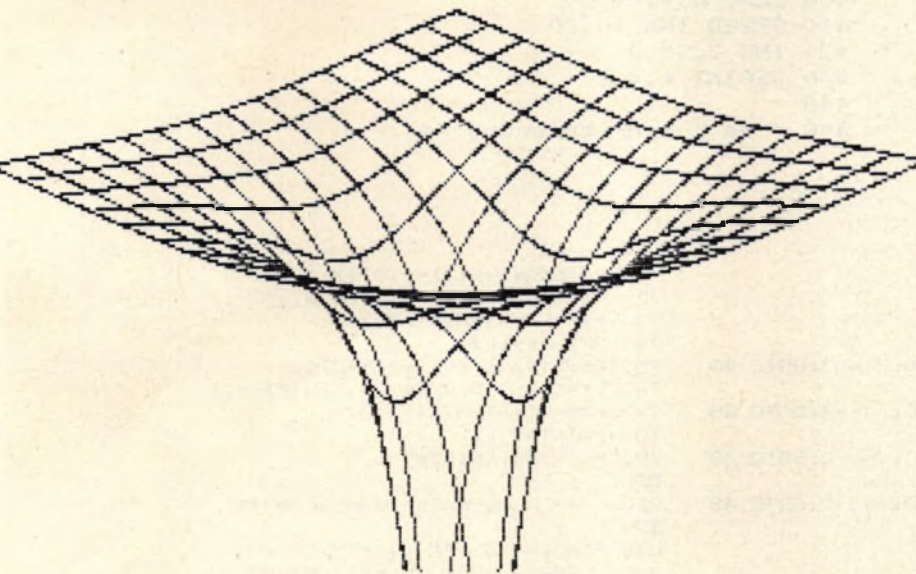
## Die Programme

Mit den abgedruckten Listings ist es möglich, Bilder zu codieren und zu decodieren. Beim Codieren ist es möglich, einen Ausschnitt zu wählen, da man evtl. Bilder zu einem Rechner übermitteln will, der keine so hohe Auflösung hat wie unser Schneider. Die codierten Bilder belegen übrigens weniger Speicherplatz auf der Diskette als normal abgespeicherte Bilder. Man kann diese Methode also auch nutzen, um teuren Diskettenspeicherplatz zu sparen. Sparen kann man dabei etwa 6K pro Bild.

Die Programme laufen in BASIC natürlich relativ langsam. In Mode 1 sind immerhin  $320 \times 200 = 64000$  Punkte zu verschlüsseln. In der WDR-Mailbox befindet sich jedoch von einem gewissen Don Camillo ein Programm in Maschinencode, welches Bilder um einiges schneller decodieren kann.

Schade ist, daß man Bilder nicht in alle Mailboxen senden kann. Viele Mailboxen, die zum Beispiel mit einem C-64 betrieben werden, erlauben nur das Senden von bis zu 99 Zeilen. Da wir jedoch für jeden vertikalen Bildpunkt eine Zeile benötigen, dürfen die Bilder in diesem Fall keine so hohe Auflösung haben.

Der WDR-Computerclub hat übrigens eine Verbesserung die-



lenweise abzutasten (scannen) und jeweils sechs nebeneinanderliegende Punkte nach obigem Schema in ein Zeichen zu verwandeln. Man erhält eine wild aussehende Ansammlung von Zeichen, welche man am besten in einer Datei auf Diskette oder Kassette ablegt. Zu beachten ist, daß nach jeder fertigen codierten Bildschirmzeile ein Zeilenendezeichen (Carriage Re-

Als erstes steht dort in Großbuchstaben „BILD“, um die Datei als Bilddatei kenntlich zu machen. In der nächsten Zeile folgt eine Rechner- und Bildbeschreibung. Wichtig sind die beiden folgenden Zeilen. Dort wird die Auflösung in horizontaler und vertikaler Richtung dargestellt. Zu beachten ist, daß die Zahlen dreistellig angegeben werden müssen. Die bei-

ses Standards angekündigt. Man kann nämlich bei Auftreten von vielen schwarzen oder weißen Punkten die Codierung optimieren, indem man zum Beispiel sagt, ab jetzt kommen soundso viele weiße Punkte. Dies ist bei der Übertragung von Schaltbildern sehr sinnvoll, da hier ein großer Teil der Bildfläche weiß bleibt. Diese Methode spart dann natürlich auch Telefonkosten. Das Senden eines Bildes dauert immerhin ca. sieben Minuten.

Wer sich aus der WDR-Box ein Bild holen möchte, dem sei hier die „LADY“ empfohlen. Leider gelingt es nur noch sehr selten, in diese Box einzudringen. Obwohl diese von zwei Usern gleichzeitig benutzt werden kann, scheint sie in letzter Zeit hoffnungslos überlastet zu sein. Zu empfehlen ist noch der Krefelder Info Service (KIS), welcher auch ganz nette Bilder bereithält. Eine Liste mit den Telefonnummern dieser Boxen kann man sich aus vielen Mailboxen senden lassen. tb

```

100 'CPC 464, 664, 6128
110 'Programm zum Codieren von Bildern in Mode 1

120 'nach dem WDR-Standard
130 '
140 'Anwendung:
150 'Mit den Cursortasten wird der blinkende Cur
    sor
160 'auf die linke obere Ecke des zu codierenden
    Ausschnittes
170 'gefahren. Mit <ENTER> wird diese Position b
    estaetigt.
180 'Genauso verfaehrt man mit der rechten unter
    en Ecke.
190 'Jetzt kann es etwas dauern bis das Bild cod
    iert ist
200 '
210 'Danach wird unter dem vorher eingegebenen N
    amen das
220 'codierte Bild gespeichert.
230 '
240 '(c) 1985
250 '    Thomas Barndt
260 '
290 '
300 MODE 2
310 '
320 ' ** Eingabe der Dateinamen **
330 '
340 a$="Name der Datei, auf der "
350 PRINT a$;:INPUT"sich das zu codierende Bild
    befindet: ",n1$
360 PRINT
370 PRINT a$;:INPUT"das codierte Bild gespeicher
    t werden soll: ",n2$
380 LINE INPUT"Bildbezeichnung: ",bez$
390 MODE 1
400 LOAD n1$,&C000
410 SPEED INK 10,10
420 INK 2,26,0
430 DEFINT x,y,z
440 '
450 ' ** Cursorsteuerung **

```

```

460 '
470 x=320:y=200
480 z=TEST(x,y)
490 PLOT x,y,2
500 a$=INKEY$
510 IF a$=CHR$(242) THEN GOSUB 570:x=x-2:GOTO 48
    0
520 IF a$=CHR$(243) THEN GOSUB 570:x=x+2:GOTO 48
    0
530 IF a$=CHR$(241) THEN GOSUB 570:y=y-2:GOTO 48
    0
540 IF a$=CHR$(240) THEN GOSUB 570:y=y+2:GOTO 48
    0
550 IF a$=CHR$(13) THEN 620
560 GOTO 500
570 PLOT x,y,z
580 RETURN
590 '
600 ' ** Koordinaten speichern **
610 '
620 x(i)=x:y(i)=y
630 IF i=0 THEN i=i+1:GOTO 500
640 PLOT x,y,z
650 '
660 ' ** Bildkopf erzeugen **
670 '
680 n2$="!" + n2$:OPENOUT n2$
690 PRINT#9,"BILD"
700 PRINT#9,bez$

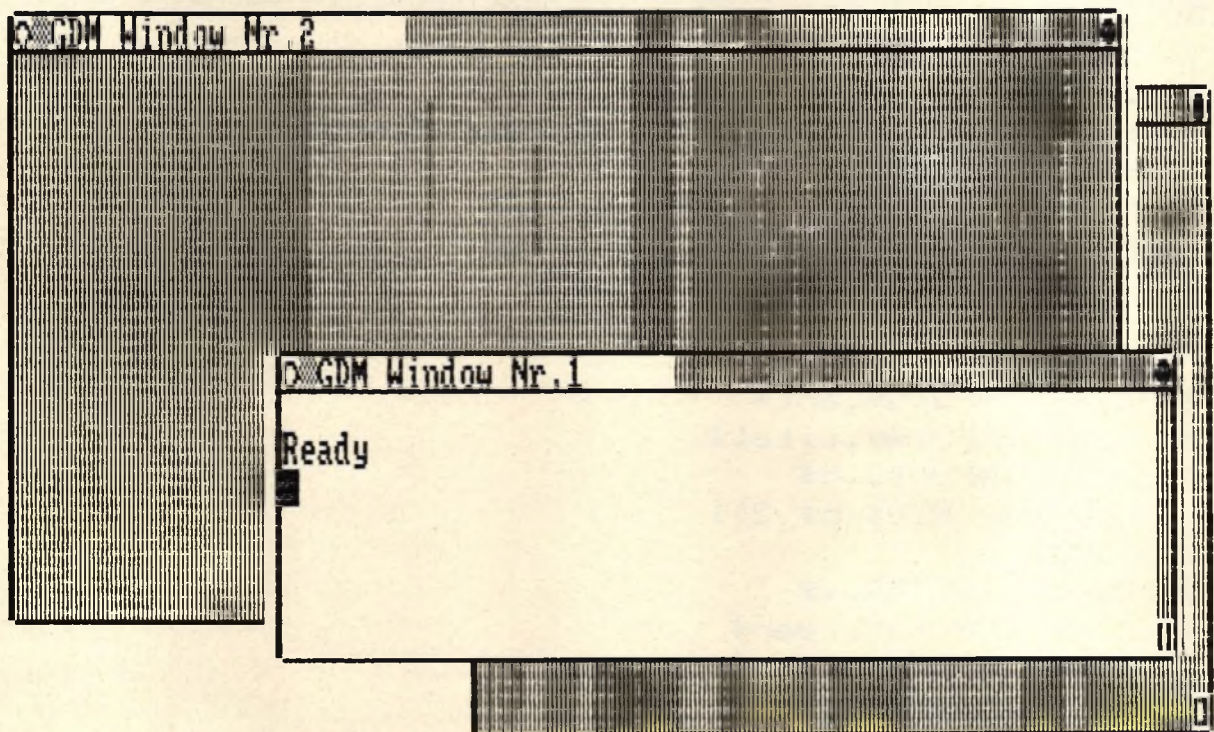
710 k$=STR$((x(1)-x(0)+2)\2)
720 k$="000"+RIGHT$(k$,LEN(k$)-1)
730 k$="H"+RIGHT$(k$,3)
740 PRINT#9,k$
750 k$=STR$((y(1)-y(0)+2)\2)
760 k$="000"+RIGHT$(k$,LEN(k$)-1)
770 k$="V"+RIGHT$(k$,3)
780 PRINT#9,k$
790 PRINT#9:PRINT#9
800 '
810 ' ** Ausschnitt codieren **
820 '
830 FOR y=y(0) TO y(1) STEP -2
840   FOR x=x(0) TO x(1) STEP 12
850     code=0
860     FOR i=0 TO 5
870       IF x+2*i>x(1) THEN 900
880       IF TEST(x+2*i,y)>0 THEN t=1 ELSE t=0
890       code=code+t*(2^i)
900     NEXT i
910     code=code+32
920     PRINT#9,CHR$(code);
930   NEXT x
940   PRINT#9
950 NEXT y
960 PRINT#9:PRINT#9
970 CLOSEOUT

```

```
100 'CFC 464, 664, 6128
110 'Programm zur Bilddecodierung nach dem WDR-S
    tandard
120 '
130 '(c) 1985
140 '    Thomas Barndt
150 '
180 '
190 MODE 2
200 DEFINT a-z
210 PRINT:PRINT TAB(10)"(c) Thomas Barndt"
220 PRINT TAB(10)"
230 PRINT TAB(10)"    1985":PRINT
240 CAT' Nur bei Diskettenbetrieb
250 INPUT"Bilddateiname:",name$
260 CLS
270 OPENIN name$
280 LINE INPUT#9,art$
290 LINE INPUT#9,titel$
300 LINE INPUT#9,h$
310 hor=VAL(MID$(h$,2))
320 h=hor/6
330 LINE INPUT#9,v$
340 LINE INPUT#9,leer$
350 LINE INPUT#9,leer$
360 v=VAL(MID$(v$,2))
370 v1=2*v
380 IF art$<>"BILD" OR h*v<=0 THEN PRINT"Kopffeh
    ler":END
390 MODE 1
400 FOR i=1 TO v
410     hor=0
420     IF NOT EOF THEN LINE INPUT#9,zeile$
430     zeile$=LEFT$(zeile$+SPACE$(h),h)
440     FOR j=1 TO h
450         w$=BIN$(ASC(MID$(zeile$,j,1))-32,6)
460         FOR k=0 TO 5
470             IF MID$(w$,5-k+1,1)="1" THEN PLOT hor+
                2*k,v1-2*i,1
480         NEXT
490         hor=hor+12
500     NEXT
510 NEXT
520 CLOSEIN
530 a$=UPPER$(INKEY$):IF a$="" THEN 530
540 IF a$="S" THEN SAVE"picture.pic",b,&C000,&40
    00 'Bild speichern
550 IF a$="N" THEN 190 'Neustart
560 IF a$="H" THEN RUN"hardcopy" 'hier eigene Ha
    rdcopy einsetzen
570 GOTO 530
```

# GDM oder „GEM“ auf dem Schneider

Hilfe Ablage Druck Ende



**Dieses Programm hat es in sich: GEM für die Schneider-Computer, fertig zum Abtippen. Um jeden Irrtum zu vermeiden, nennen wir unser Programm GDM (Graphic Desk Manager).**

Was ist GDM? Nun, GDM ist eine BASIC-Erweiterung, mit deren neuen Befehlen man im kleinen Stil eine Atari ST, Macintosh- oder eben GEM-Benutzerschnittstelle simulieren kann, und zwar auf 464, 664 oder 6128. GDM verwaltet dabei einen Pointer (ein kleiner Pfeil zur Objekt-Auswahl), WINDOWS, die vergrößert, verkleinert und umhergeschoben werden können, DROP-DOWN-Menüs und Menü-Balken, Knöpfe, mit denen der Benutzer durch „Drücken“ Programmfunktionen auslösen kann, und nicht zuletzt kann der Benutzer jederzeit eine Hardcopy-Funktion auslösen.

## Wie benutzt man GDM?

Am Anfang steht natürlich die Eingabe des Programms, das hier

in Hex-Darstellung wiedergegeben ist. Jede Zahl steht hier für ein Byte im Speicher, wobei vor jeder Zeile die Adresse steht, und dann jeweils 14 Bytes.

Um das Programm einzugeben, sollte man das BASIC-Programm-Listing Nr. 1 benutzen. Dieses Programm gibt die Adressen vor, es müssen dann nur noch die Zahlen eingegeben werden.

Um also eine lauffähige Version von GDM zu erhalten, sind folgende Schritte notwendig:

\*BASIC-Listing Nr. 1 eingeben & abspeichern

\* BASIC-Programm starten und Hex-Zahlen eingeben

Das BASIC-Programm schreibt nach der letzten Adresse automatisch eine Kopie von GDM auf Kassette oder Diskette, die dann mit LOAD"GDM nachgeladen werden kann (siehe unten).

## Anwendung

Hat man nun eine Kopie von GDM auf Kassette/Diskette, gibt

es zwei Möglichkeiten, GDM in eigenen Programmen zu verwenden:

Man kann GDM zuerst starten und dann sein eigenes Programm laden. Dazu erst den Speicher mit MEMORY &95FF (GDM beginnt ab Adresse &9600) reservieren, dann GDM mit LOAD"GDM laden, und mit CALL &9600 initialisieren. Alle hier abgedruckten Beispielprogramme setzen voraus, daß Sie diese Schritte vorher ausgeführt haben.

Oder, und das wird wohl die Hauptanwendung sein, man kann GDM von seinem eigenen Programm nachladen. Dazu müssen die oben angegebenen Befehle am Anfang des BASIC-Programms stehen, das GDM verwendet. Die erste Zeile des jeweiligen Programms sollte also lauten:

```
10 MEMORY &95FF:LOAD
"GDM: CALL &9000
```

Das Fenster Nr. 1, das dabei erscheint, muß mit der Nummer 0 (s.u.) angesprochen werden.

Der Befehl :INIT führt die Anfangssequenz von GDM erneut durch und kann jederzeit aufgerufen werden. Window-Größen und -Lagen werden nicht verändert.

Hilfe **Alt+F2** Druck Ende

Speichern

Einlesen

GDM Window Nr. 3

GDM Window Nr. 1

GDM unterstuetzt PULL-DOWN-MENUES!!

:INIT sollte nur aufgerufen werden, wenn Window 0 (GDM Window Nr.1) aktiv ist.

**Außerdem wird von :INIT Mode 2 eingeschaltet, da GDM nur in Mode 2 korrekt arbeitet.**

## Der Pointer

Der Pointer ist ein kleiner Pfeil, den der Benutzer mit den Pfeiltasten oder dem Joystick beliebig über den Bildschirm bewegen kann. Durch den Befehl **:p.move** wird die komplette Steuerung des Pointers übernommen, so daß das eigentliche Programm sich überhaupt nicht darum zu kümmern braucht. Es gibt aber noch mehr Befehle, um den Pointer zu beeinflussen, und zwar folgende:

**:PON** zeigt den Pointer wieder an, falls er mit **:POFF** abgeschaltet wurde. Der Pointer muß vorher abgeschaltet sein, sonst passiert bei diesem Befehl gar nichts. In diesem Zusammenhang sollte man daran denken, daß der Pointer *zwar nach einem CLS oder Überschreiben verschwindet, aber erst nach einem :POFF wieder neu angezeigt werden kann.*

**:POFF** läßt den Pointer vom Bildschirm verschwinden, so lange, bis er mit **:PON** wieder eingeschaltet wird.

**:P.POS,X\*1,Y\*1** setzt den Pointer auf die angegebene Position, wobei er x-Werte von 0-79, y-Werte von 0-49 annehmen kann.

**:P.ASK,@x,@y** legt die augenblickliche Position des Pointers in die Variablen x und y, wobei es sich um Integervariablen handeln muß.

**:P.MOVE** schaltet den Pointer ein (mit **:PON**) und bewegt ihn synchron zu den Bewegungen des Joysticks. Sobald der Benutzer den Joystick-Feuerknopf drückt, verschwindet der Pointer wieder, und der Befehl wird beendet. Das Programm sollte nun mit **:P.ASK** überprüfen, wo der Pointer zuletzt war, und entsprechend handeln. Der Befehl wird auch durch Drücken der ESC-Taste abgebrochen.

## Windows

GDM kann Windows verwalten, die diese Bezeichnung sehr viel mehr verdienen als die des Schneider-BASICs. GDM-Windows

```

10 ' Dieses Programm dient zum eingeben   Listing Nr. I
20 ' und abspeichern von GDM
30 '
40 INK 0,0:BORDER 0:INK 1,26:MODE 2:LOCATE 1,25
50 MEMORY &95FF:DIM 1(20)
60 INPUT "Soll ich einen bereits eingetippten Teil
   einlesen (j/n)";a$
70 IF LOWER$(LEFT$(a$,1))="j" THEN INPUT "Bitte Di
   skette/Kassette mit File GDM.BIN einlegen, dann EN
   TER ",a$:LOAD"gdm.bin",&9600
80 INPUT "Bitte geben Sie die Adresse in Hexadezim
   al ein, ab der Sie eingeben moechten (Beim Anfa
   ng 9600):",a$:IF LEN(a$)<>4 THEN PRINT"Falsche Adr
   esse":GOTO 80 ELSE adr=VAL("&"a$)
90 PRINT"Sie koennen nun eingeben. Wenn Sie aufhoe
   ren moechten, drucken Sie nur ENTER":PRINT
100 PRINT HEX$(adr,4)": ";:INPUT "",l$:IF l$="" TH
   EN 190 ELSE IF LEN(l$)<>4 THEN PRINT CHR$(7)"Fehl
   er in der Eingabe. Bitte nochmal!":GOTO 100
110 cs=0:zz=1
120 FOR x=1 TO 41 STEP 3
130 b$=MID$(l$,x,2):b=VAL("&"b$):cs=cs+b:l(zz)=b:
   zz=zz+1
140 NEXT
150 IF cs>255 THEN cs=cs-255:GOTO 150
160 c$=RIGHT$(l$,2):c=VAL("&"c$):IF c<>cs THEN PR
   INT CHR$(7)"Falsche Pruefsumme. Bitte nochmal!":G
   OTO 100
170 FOR t=1 TO 14:POKE adr,l(t):adr=adr+1:NEXT
180 GOTO 100
190 PRINT CHR$(7):INPUT"Moechten Sie aufhoeren? (j
   /n)",a$:IF LOWER$(LEFT$(a$,1))="j" THEN PRINT"Beim
   naechstenmal muessen Sie mit dieser Adresse begin
   nen:"HEX$(adr,4):PRINT"Schreiben Sie Sich das auf!
   Druucken Sie bitte ENTER, wenn die Kassette" ELSE
   100
200 INPUT "Oder Diskette einliegt, auf der ich den
   bisherigen Teil von GDM.BIN abspeichern kann.",a$
210 SAVE "GDM.BIN",b,&9600,adr-&9600,&9600
220 PRINT:PRINT"Bis zum naechsten mal..."

```

## Der Pointer



**IP.MOVE endet auch bei Neue-Beruehrung**

können nämlich vom Benutzer beliebig vergrößert oder verkleinert werden, verschoben oder über ein anderes Objekt bzw. darunter gelegt werden, vorausgesetzt, im Programm befindet sich ein Unterprogramm, das auf einen KLICK reagiert. Nur das aktuelle Window (das an oberster Stelle liegt) enthält Text, alle anderen sind innen einfach grau schraffiert. Bei allen Windowbefehlen ist zu beachten, daß der eigentliche Arbeitsbereich im Fenster in x- und y-Richtung jeweils 1 Zeichen (bzw. Zeile) kleiner ist als die Außenmaße, da auch noch Platz für das Kontrollfeld benötigt wird. Nach einem .W.ACTIVE laufen alle normalen PRINT-, INPUT-Befehle über das augenblickliche Fenster. Der BASIC-Window-Befehl kann weiter benutzt werden, aber man sollte sich darüber klar sein, daß dadurch evtl. die Umrahmung eines Fensters zerstört werden kann. Wenn der Benutzer die Größe oder Lage eines Fensters verändern möchte, so kann das Programm das ganz einfach feststellen, indem es die Position des Pointers mit der linken oberen Ecke (Bewegungsfeld), der oberen rechten Ecke (Vollfeld) oder der rechten unteren Ecke (Größenfeld) des aktuellen Fensters vergleicht. Dann sollte ein :FRAME...-Befehl aufgerufen und das Fenster neu gezeichnet werden. Wenn der Pointer sich nicht über einem Fenster befindet, sollte das Programm überprüfen, ob er evtl. über einem anderen Fenster steht, das aktiviert werden soll. Das Beispielprogramm Nr. 1 zeigt, wie es geht.

**:W.ACTIVE, NR\*1** zeichnet ein Fenster neu bzw. legt das Fenster über alle anderen. Der Inhalt des Fensters wird angezeigt, und so lange, bis ein anderes Fenster aktiviert wird, gehen alle Bildschirmausgaben zu diesem Fenster. Nr bezeichnet dabei die Nummer des Fensters, 0-3. *Auf gar keinen Fall sollte die Nummer >3 werden, da nur 4 Fenster verwaltet werden können.*

**:W.DESACTIVE, NR\*1** deaktiviert das Fenster mit der Nummer nr, d.h., das Fenster verschwindet vom Bildschirm. So lange, bis das Fenster (oder ein anderes) mit .W.ACTIVE wieder eingeschaltet wird, sind keine Bildschirmausgaben möglich. In diesem Fall wird in das nächste offene Window geschrieben. Bei einem .W.ACTIVE erscheint das Fenster an der alten

## GDM Graphic Desk Manager Version 1.0

```

9600: C3 69 96 00 02 4E 02 14 47 44 4D 20 57 69 E3
960E: 6E 64 6F 77 20 4E 72 2E 31 20 20 20 20 20 9A
961C: 02 4E 02 14 47 44 4D 20 57 69 6E 64 6F 77 D9
962A: 20 4E 72 2E 32 20 20 20 20 20 02 4E 02 14 48
9638: 47 44 4D 20 57 69 6E 64 6F 77 20 4E 72 2E 82
9646: 33 20 20 20 20 20 02 4E 02 14 47 44 4D 20 33
9654: 57 69 6E 64 6F 77 20 4E 72 2E 34 20 20 20 1E
9662: 20 20 00 00 00 00 04 01 8D 97 21 51 98 CD 43
9670: D1 BC 3E 02 CD 0E BC CD 14 BC 3E 00 01 00 45
967E: 00 CD 32 BC 3E 01 06 1A 0E 1A CD 32 BC 01 02
968C: 00 00 CD 38 BC 21 E8 23 22 55 98 3E 0C 32 7C
969A: 59 99 3E 28 32 58 99 3E 00 32 57 98 32 58 68
96A8: 98 26 00 16 4F 2E 01 1E 19 CD 66 BB 3E 04 BC
96B6: 32 68 96 3E 00 32 64 96 3E 00 CD F1 99 CD 02
96C4: EC 98 CD 02 99 21 D5 96 7E B7 CB CD 5A BB 5F
96D2: 23 18 F7 0D 0A 20 47 44 4D 20 2D 20 47 72 6A
96E0: 61 70 68 69 63 20 44 65 73 6B 20 4D 61 6E EC
96EE: 61 67 65 72 0D 0A 20 43 6F 70 79 72 69 67 87
96FC: 68 74 20 31 39 38 36 20 54 68 6F 6D 61 73 64
970A: 20 4D 2E 42 69 6E 7A 69 6E 67 65 72 0D 0A 5E
9718: 0D 0A 00 3D DD 6E 00 DD 66 01 DD 23 DD 23 E7
9726: C9 C3 EC 98 C3 02 99 CD 02 99 CD 1B 97 E5 42
9734: CD 1B 97 C1 65 69 7C 32 58 99 7D 32 59 99 54
9742: CD 1A BC 11 00 20 19 11 00 C0 ED 52 22 55 78
9750: 98 C3 EC 98 CD 1B 97 3A 59 99 77 23 36 00 60
975E: CD 1B 97 3A 58 99 77 23 36 00 C9 28 63 29 FB
976C: 54 4D 42 20 00 38 00 3E 00 3F 80 3F E0 3F 99
977A: F8 3F FE 3F F0 3F E0 30 70 00 38 00 1C 00 7C
9788: 0E 00 07 00 00 CE 97 C3 27 97 C3 2A 97 C3 47
9796: 2D 97 C3 54 97 C3 76 9B C3 17 9C C3 A6 9C C8
97A4: C3 D4 9C C3 1B 9D C3 55 9D C3 A0 9D C3 B3 E1
97B2: 9D C3 5B 99 C3 09 9E C3 49 9E C3 D9 9E C3 6D
97C0: 46 9F C3 6E 9F C3 76 9F C3 87 9F C3 69 96 40
97CE: 50 2E 4F CE 50 2E 4F 46 C6 50 2E 50 4F D3 69
97DC: 50 2E 41 53 CB 46 52 41 4D 45 53 49 5A C5 08
97EA: 46 52 41 4D 45 4D 4F 56 C5 57 2E 41 43 54 83
97F8: 49 56 C5 57 2E 4E 41 4D C5 57 2E 53 49 5A 0A
9806: C5 53 43 52 45 45 4E 4E 45 D7 57 2E 44 45 02
9814: 53 41 43 54 49 56 C5 57 2E 41 53 CB 50 2E F5
9822: 4D 4F 56 C5 57 2E 50 4F D3 42 2E 4F CE 4D 8D
9830: 2E 4F CE 4D 2E 4F 46 C6 4D 2E 52 45 53 45 CF
983E: D4 4D 2E 41 53 CB 48 41 52 44 43 4F 50 D9 8D
984C: 49 4E 49 D4 00 51 98 8D 97 E8 23 00 00 00 00
985A: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
9868: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
9876: 00 00 00 00 E5 D5 C5 2A 55 98 11 00 C0 19 84
9884: 11 5A 98 06 0F C5 7E 12 23 13 7E 12 13 CD 17
9892: 26 BC 2B C1 10 F1 C1 D1 E1 C9 E5 D5 C5 3E D0
98A0: 01 32 58 98 2A 55 98 11 00 C0 19 11 6F 97 3F
98AE: 06 0F C5 1A B6 77 23 13 1A B6 77 13 CD 26 A8
98BC: BC 2B C1 10 EF C1 D1 E1 C9 E5 D5 C5 3E 00 A8
98CA: 32 58 98 2A 55 98 11 00 C0 19 11 5A 98 06 30
98DB: 0F C5 1A 77 13 23 1A 77 13 CD 26 BC 2B C1 DE
98E6: 10 F1 C1 D1 E1 C9 F5 F3 3A 57 98 B7 20 0B 38
98F4: 3E 01 32 57 98 CD 7A 98 CD 9C 98 F1 FB C9 FC
9902: F5 F3 3A 57 98 B7 28 0B CD C5 98 CD 7A 98 0C
9910: 3E 00 32 57 98 F1 FB C9 F5 CD C5 98 F1 C9 F5
991E: D5 11 50 00 47 3A 59 99 CB 40 28 06 B7 28 C5
992C: 03 ED 52 3D CB 48 28 06 FE 17 28 02 19 3C 58
993A: 32 59 99 3A 58 99 CB 50 28 05 B7 28 02 2B A7
9948: 3D CB 58 28 06 FE 4E 28 02 23 3C 32 58 99 8A
9956: D1 C9 28 0C 00 E5 D5 CD EC 98 3A 57 98 B7 C0
9964: 28 34 CD 24 BB E5 3E 42 CD 1E BB E1 C2 9D 5A
9972: 99 3A 58 98 B7 CD C5 98 3A 59 99 B7 CC A3 FD
9980: 99 7C CB 67 20 14 2A 55 98 CD 1E 99 22 55 92
998E: 98 CD 7A 98 CD 9C 98 CD DD 99 18 CC CD E8 5D

```

```

999C: 99 CD 02 99 D1 E1 C9 3A CC 9E B7 C8 3A D7 B8
99AA: 9E B7 C0 DD E1 3E 01 32 D7 9E DD 21 CD 9E 2A
99B8: 0E 01 DD 7E 00 B7 20 06 32 D8 9E C3 9A 99 EA
99C6: 47 3A 58 99 3C B8 38 07 28 06 DD 23 0C 18 FA
99D4: E5 0D 79 32 D8 9E C3 9A 99 C5 01 10 27 0B 17
99E2: 78 B1 20 FB C1 C9 06 64 CD 1B BB 10 FB C9 B6
99FO: 00 F3 32 FO 99 CD 92 9D FE 00 C8 3A FO 99 3B
99FE: F5 21 00 00 16 4F 1E 19 CD 66 BB F1 21 03 B9
9A0C: 96 23 FE 00 28 07 11 18 00 47 19 10 FD E5 65
9A1A: DD E1 DD 7E 01 B7 FB C8 F3 3E 55 DD 66 00 65
9A28: DD 56 01 DD 6E 02 DD 5E 03 E5 D5 CD 44 BC 4D
9A36: DD 66 00 DD 6E 02 2C 24 CD 75 BB 3E E6 CD D4
9A44: 5A BB 3E CF CD 5A BB DD 7E 00 47 DD 7E 01 09
9A52: 98 06 14 FE 18 30 05 3D 3D 3D 47 DD E5 FE
9A60: E1 11 04 00 19 7E CD 5A BB 23 10 F9 DD 66 E3
9A6E: 01 DD 6E 02 2C CD 75 BB 3E E7 CD 5A BB DD 62
9A7C: 66 01 DD 6E 03 2C CD 75 BB 3E E8 CD 5A BB EC
9A8A: DD 66 00 DD 6E 02 CD 1A BC 11 00 38 19 E5 7F
9A98: DD 7E 00 47 DD 7E 01 98 4F 06 00 D1 13 36 0A
9AA6: FF ED B0 D1 E1 2C 15 15 CD 66 BB 3A FO 99 5D
9AB4: CD 67 9B 3A FO 99 47 3A 03 96 B8 CC 6C BB 5E
9AC2: FB C9 01 3A FO 99 DD 21 03 96 DD 23 B7 28 05
9ADO: 08 11 18 00 47 DD 19 10 FC DD 7E 00 47 DD FD
9ADE: 7E 01 98 47 C5 DD 66 00 DD 6E 02 CD 1A BC 5C
9AEC: C1 C5 11 50 00 ED 52 11 00 38 19 CD 3F 9B 34
9AFA: DD 66 00 DD 6E 03 2C CD 1A BC C1 CD 3F 9B CE
9B08: DD 66 00 DD 6E 02 CD 1A BC DD 7E 02 47 DD BA
9B16: 7E 03 98 3C CB 27 CB 27 CB 27 47 C5 3E 07 81
9B24: 32 3E 9B 2B CD 50 9B DD 66 01 DD 6E 02 CD 52
9B32: 1A BC C1 3E C8 32 3E 9B CD 50 9B C9 C8 3A 32
9B40: C4 9A B7 28 04 36 FF 18 03 7E 2F 77 23 10 EC
9B4E: FO C9 3A C4 9A B7 28 06 3A 3E 9B 77 18 03 E0
9B5C: 7E 2F 77 C5 CD 26 BC C1 10 EA C9 3E 01 32 93
9B6A: C4 9A C3 C5 9A AF 32 C4 9A C3 C5 9A CD EC A3
9B78: 98 CD C5 98 3A 03 96 32 FO 99 DD 21 03 96 ED
9B86: DD 23 B7 28 08 11 18 00 47 DD 19 10 FC DD 3B
9B94: 66 02 DD 6E 03 E5 3A 58 99 DD 77 01 3A 59 B3
9BA2: 99 DD 77 03 CD 24 BB 7C CB 67 20 35 F5 2A C4
9BB0: 55 98 22 EA 9B 3A 58 99 32 EC 9B 3A 59 99 AA
9BBE: 32 ED 9B F1 CD 1E 99 22 55 98 CD EE 9B CD 69
9BCC: 6F 9B CD 7A 98 CD 9C 98 CD 9B 9C CD C5 98 21
9BDA: CD 6F 9B E1 DD 74 02 18 B0 CD E8 99 E1 C3 CD
9BE8: 02 99 00 00 00 00 DD 7E 00 3C 47 3A 58 99 A7
9BF6: B8 38 0B DD 7E 02 47 3A 59 99 B8 38 01 C9 8A
9C04: 3A EC 9B 32 58 99 3A ED 9B 32 59 99 2A EA E4
9C12: 9B 22 55 98 C9 CD EC 98 CD C5 98 3A 03 96 C8
9C20: 32 FO 99 DD 21 03 96 DD 23 B7 28 08 11 18 67
9C2E: 00 47 DD 19 10 FC DD 7E 00 47 DD 7E 01 98 E4
9C3C: 47 3A 58 99 FE 02 38 2A 80 FE 50 30 08 98 77
9C4A: DD 77 00 80 DD 77 01 DD 7E 02 47 DD 7E 03 31
9C58: 98 47 3A 59 99 FE 02 38 0D 80 FE 19 30 08 24
9C66: 98 DD 77 02 80 DD 77 03 CD 24 BB 7C CB 67 26
9C74: C2 95 9C 2A 55 98 CD 1E 99 22 55 98 CD 6F DF
9C82: 9B CD 7A 98 CD 9C 98 CD 9B 9C CD C5 98 CD 7F
9C90: 6F 9B C3 34 9C CD E8 99 C3 02 99 C5 01 88 9E
9C9E: 13 08 78 B1 20 FB C1 C9 CD 02 99 DD E5 3A 57
9CAC: 03 96 F5 3E 63 32 03 96 F1 CD F1 99 DD E1 08
9CBA: CD 1B 97 7D 32 03 96 32 64 96 21 68 96 47 5E
9CC8: B7 28 03 28 10 FD 36 01 CD F1 99 C9 CD 1B 5F
9CD6: 97 E5 CD 1B 97 DD 21 08 96 7D FE 00 28 08 48
9CE4: 11 18 00 47 DD 19 10 FC FD E1 FD 46 00 FD 96
9CF2: 6E 01 FD 66 02 78 FE 15 38 02 06 14 0E 00 C4
9D00: 7E DD 77 00 23 DD 23 0C 10 F6 79 FE 14 D0 68
9DOE: DD 36 00 CF DD 23 0C 79 FE 14 38 F4 C9 CD 42
9D1C: 1B 97 E5 CD 1B 97 E5 CD 1B 97 E5 CD 1B 97 E5
9D2A: E5 CD 1B 97 7D 47 11 18 00 DD 21 03 96 DD CA
9D38: 23 B7 28 04 DD 19 10 FC E1 7D DD 77 00 E1 A1
9D46: 7D DD 77 02 E1 7D DD 77 01 E1 7D DD 77 03 42

```

Position und mit der alten Größe von neuem.

**:W.ASK, NR\*1, @X1, @Y1, @X2, Y2** ergibt die augenblicklichen Ecken des Fensters mit der Nummer nr. Das Fenster muß angezeigt sein. Mit dieser Funktion kann überprüft werden, ob der Pointer auf eines der Aktionsfelder des Fensters zeigt. Die Variablen x1, y1, x2, und y2 müssen Integervariablen sein.

**:W.POS, NR\*1, X1\*1, Y1\*1, X2\*1, Y2\*1** legt die Lage und Größe (also die obere linke und untere rechte Ecke) des Fensters fest.

**:W.NAME, NR\*1, @N\$** legt eine neue Überschrift für das Fenster nr fest. N\$ sollte dabei nicht länger als 20 Zeichen sein.

**:SCREENNEW** dient dazu, den Bildschirm „aufzuräumen“. D.h., der Bildschirm wird komplett gelöscht, und alle Fenster werden in der richtigen Größe und Lage über/untereinander gezeichnet. Dadurch verschwinden alle nicht gespeicherten Objekte wie Drop-Down-Menüs und Knöpfe vom Bildschirm.

Der Benutzer kann Windows vergrößern oder verkleinern und deren Position beliebig ändern, indem er die im Bild wiedergegebenen Kontrollen bedient. Das Programm muß überprüfen, ob eine dieser Kontrollen bedient wurde (indem es deren Position mit der Position des Pointers vergleicht). Sollte das der Fall sein, sollten die folgenden Maßnahmen unternommen werden:

1) Neue Position mit :FRAMEPOS oder neue Größe mit :FRAMESIZE wählen lassen

2) Fenster mit :SCREENNEW neu zeichnen

**:FRAMEPOS** zeichnet ein Rechteck in der Größe und Lage des aktuellen Fensters auf den Bildschirm. Dieses bewegt sich so lange synchron mit den Bewegungen des Pointers (der von dieser Funktion eingeschaltet wird), bis ein KLICK ausgeführt wird. Die neuen Koordinaten werden dann als die Position des aktuellen Fensters abgespeichert, und das Rechteck verschwindet.

**:FRAMESIZE** zeichnet ein Rechteck auf den Bildschirm, dessen untere rechte Ecke sich mit dem Pointer bewegt. Die obere linke Ecke steht fest und stimmt mit der Position des aktuellen Fensters überein. Nach einem Klick verschwindet der Rahmen wieder, und die x2- und y2-Werte des aktuellen Fensters werden entspre-



chend den letzten Rechteckausmaßen korrigiert. Das Fenster soll dann neu gezeichnet werden.

## Menüzeile

GDM unterstützt automatisch eine Menüzeile. Diese besteht aus mehreren Einträgen in der ersten Zeile des Bildschirms. Wenn der Benutzer den Pointer (beim Befehl :PMOVE) auf eines dieser Wörter führt, wird dieses invertiert dargestellt, und :PMOVE wird beendet, ohne daß der Benutzer einen KLICK ausgeführt hat. Das Programm sollte mit :PASK überprüfen, ob der Pointer auf die oberste Zeile ( $y=0$ ) zeigt und dann ein Drop-Down-Menü anzeigen.

Welcher Menüpunkt angewählt wurde, kann das Programm anhand des Befehls :M.ASK herausfinden. Erst nach dem Befehl :M.RESET ist es wieder möglich, ein Menü anzuwählen. Also vor Schließen des Pull-Down-Menüs unbedingt :M.RESET ausführen und dann mit :SCREENNEW das Pull-Down-Menü selber verschwinden lassen (evtl. auch noch Menü-Bar neu zeichnen). Wie das geht, zeigt Beispiel-Programm 2.

**:M.ON, @M\$** zeigt die Menüzeile an. @m\$ ist dabei die Adresse des Menüstrings m\$, der folgendermaßen aufgebaut sein muß:

m\$ =

Die einzelnen Menütitel (max. 9) getrennt durch / und beginnend und endend mit /,

z.B. m\$ = „Hilfen/ Ablage/Ende“

Dieser Befehl kann beliebig oft aufgerufen werden, z.B., um einen Eintrag invertiert darzustellen.

**:M.RESET** hebt die Menüsperierung auf. Erst nach diesem Befehl ist es möglich, erneut ein Menü anzuwählen. Vorher reagiert der Pointer nicht mehr auf Menü-Berührung. Der Pointer sollte nicht mehr auf den Menübalken zeigen.

**:M.OFF** schaltet die Menüzeile wieder ab.

**:M.ASK, @P** ergibt die Nummer des zuletzt angewählten Menüs. P muß eine Integer-Variable sein. Ist das Ergebnis 0, wurde kein Menü gewählt.

## Knöpfe

GDM unterstützt Knöpfe, die der Benutzer durch „Drücken“ mit dem Pointer anwählen kann.

```

9D54: C9 26 00 16 4F 2E 01 1E 18 CD 66 BB CD 6C E4
9D62: BB 3A CC 9E B7 20 0D 21 00 F8 11 01 F8 01 6C
9D70: 4F 00 36 00 ED B0 06 04 21 65 96 C5 E5 7E 75
9D7E: B7 28 05 78 3D CD F1 99 C9 E5 21 68 96 47 B7 28 03 4E
9D8C: 64 96 CD F1 99 C9 E5 21 68 96 47 B7 28 03 4E
9D9A: 2B 10 FD 7E E1 C9 CD 1B 97 7D 47 21 68 96 C8
9DA8: B7 28 03 2B 10 FD 36 00 C3 55 9D CD 1B 97 89
9DB6: E5 CD 1B 97 E5 CD 1B 97 E5 CD 1B 97 E5 CD E6
9DC4: 1B 97 7D DD 21 03 96 DD 23 B7 28 08 47 11 0A
9DD2: 18 00 DD 19 10 FC FD E1 DD 7E 00 FD 77 00 CD
9DE0: FD 36 01 00 FD E1 DD 7E 02 FD 77 00 FD 36 1D
9DEE: 01 00 FD E1 DD 7E 01 FD 77 00 FD 36 01 00 E8
9DFC: FD E1 DD 7E 03 FD 77 00 FD 36 01 00 C9 CD 81
9EOA: 1B 97 E5 CD 1B 97 E5 CD 1B 97 E5 CD 1B 97 E5
9E18: E5 CD 1B 97 DD 21 03 96 DD 23 7D B7 28 08 65
9E26: 47 11 18 00 DD 19 10 FC E1 7D DD 77 00 E1 0B
9E34: 7D DD 77 02 E1 7D DD 77 01 E1 7D DD 77 03 42
9E42: C3 55 9D 00 00 00 00 CD 1B 97 22 47 9E CD 0D
9E50: 1B 97 7D 32 46 9E CD 1B 97 7D 32 45 9E 67 C2
9E5E: 3A 46 9E 6F CD 75 BB 3E DE CD 5A BB 2A 47 FF
9E6C: 9E 7E 47 3E CF CD 5A BB 10 F9 3E DF CD 5A A6
9E7A: BB 3A 45 9E 67 3A 46 9E 3C 32 46 9E 6F CD FO
9E88: 75 BB 3E CF CD 5A BB 2A 47 9E 23 E5 DD E1 FB
9E96: DD 6E 00 DD 66 01 DD 46 FF 7E CD 5A BB 23 3B
9EA4: 10 F9 3E CF CD 5A BB 3A 45 9E 67 3A 46 9E AO
9EB2: 3C 6F CD 75 BB 3E DD CD 5A BB DD 46 FF 3E OD
9ECO: CF CD 5A BB 10 F9 3E DC CD 5A BB C9 00 00 86
9ECE: 00 00 00 00 00 00 00 00 00 00 00 00 CD 1B 97 80
9EDC: E5 DD E1 CD 69 BB E5 D5 21 00 00 16 50 1E F9
9EEA: 01 CD 66 BB CD 6C BB DD 46 00 DD 6E 01 DD 36
9EF8: 66 02 DD 21 CD 9E DD 36 00 00 3E 00 32 D8 31
9F06: 9E 7E FE 2F CC 2D 9F CD 5A BB 23 10 F4 D1 C2
9F14: E1 CD 66 BB 3E 01 32 CC 9E 21 00 F8 11 01 DA
9F22: F8 01 4F 00 36 FF ED B0 C3 55 9D E5 C5 3E BE
9F30: 20 CD 5A BB CD 78 BB 24 DD 74 00 DD 23 DD 5B
9F3E: 36 00 00 C1 E1 3E 20 C9 3E 00 32 CC 9E 32 10
9F4C: D8 9E 32 CD 9E 32 D7 9E CD 69 BB E5 D5 21 8E
9F5A: 00 00 16 50 1E 01 CD 66 BB CD 6C BB D1 E1 1F
9F68: CD 66 BB C3 55 9D AF 32 D7 9E 32 D8 9E C9 72
9F76: CD 1B 97 E5 DD E1 3A D8 9E DD 77 00 DD 36 41
9F84: 01 00 C9 CD 2E BD D8 C3 00 AO 00 00 00 00 C1
9F92: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
9FA0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
9FAE: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
9FBC: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
9FCA: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
9FDB: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
9FEB: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
9FF4: 00 00 00 00 00 00 00 00 00 00 00 00 CD BA 88
A002: BB CD E7 BB 32 C9 AO CD 75 AO 21 8F 01 22 81
A010: CA AO 11 00 00 3E 03 32 CC AO CD 85 AO 0E 5F
A01E: 00 3A CC AO 47 E5 D5 C5 CD FO BB C1 D1 21 9F
A02C: C9 AO BE E1 37 20 07 A7 CB 11 CB 11 18 05 E7
A03A: CB 11 37 CB 11 2B 2B 10 EO CD BD AO 79 CD AB
A048: B4 AO 13 E5 21 7F 02 37 ED 52 E1 38 05 2A B1
A056: CA AO 18 C3 23 7C B5 C8 2B 11 00 00 22 CA 8E
A064: AO 3E 03 BD 20 B0 7C B4 20 AC 3E 02 32 CC AD
A072: AO 18 A5 3E 1B CD B4 AO 3E 33 CD B4 AO 3E AD
A080: 11 CD B4 AO C9 E5 3E 42 CD 1E BB E1 28 02 18
A08E: E1 C9 3E OD CD B4 AO 3E OA CD B4 AO 3E 1B DE
A09C: CD B4 AO 3E 2A CD B4 AO 3E 04 CD B4 AO 3E 52
A0AA: 7F CD B4 AO 3E 02 CD B4 AO C9 CD 2E BD 38 C1
A0BB: FB CD 2B BD C9 3A CC AO FE 03 C8 AF CB 11 7B

```

```

A0C6: CB 11 C9 00 00 00 00 00 00 00 00 00 00 A6
A0D4: 00 00 00 00 00 00 00 00 00 00 00 00 00

```

**:BON, NR\*1, X\*1, Y\*1, @B\$** gibt einen Knopf mit dem Text in b\$ an der Stelle x, y aus.

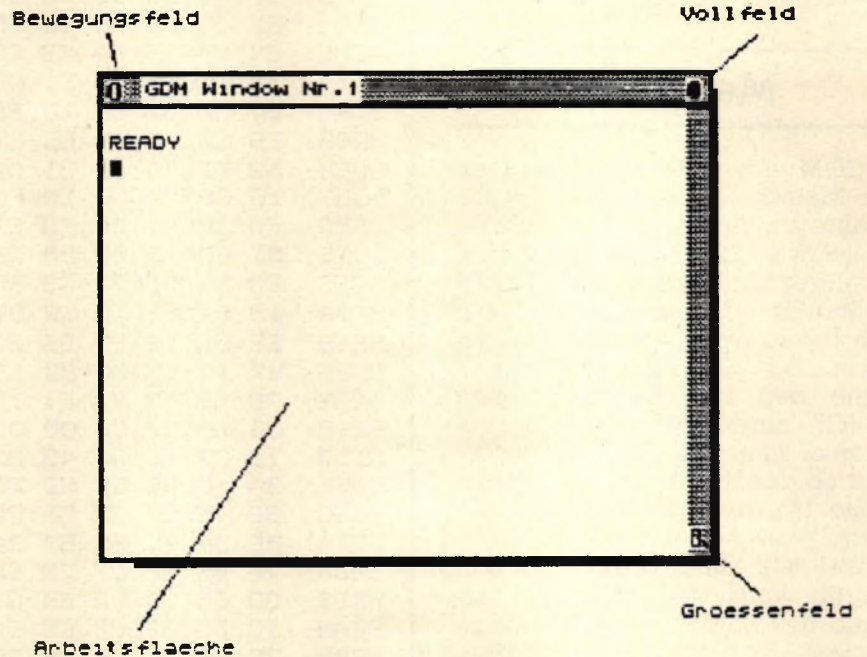
## Sonstiges

Der Benutzer kann jederzeit eine Hardcopy, also einen Bildschirmausdruck, anfertigen. Dafür ist der Befehl **:hardcopy** zuständig. Wenn der Drucker zum Zeitpunkt des Aufrufs nicht bereit ist, passiert gar nichts. Die Hardcopy funktioniert nur auf Epson-kompatiblen Druckern, z.B. Star SG-10.

Folgende Aktionen sollten vom Programm überwacht werden:

Windows verschieben, vergrößern, verkleinern, schließen und obenauflegen und alle Benutzeraktivitäten überwachen. Außerdem sollte ein Drop-Down-Menü angezeigt werden, wenn der Menübalken sichtbar ist und der Pointer die x-Position 0 hat. tmb

## Die Fensterkontrollen:



Das Programm sollte die Felder in entsprechender Weise interpretieren.

### Listing Nr. 1

```

10 REM Dieses Programm erlaubt es, ein Fenster
20 REM beliebig zu vergrössern und zu verschieben
.
30 REM
40 DEFINT p,x,y
50 x1=0:y1=0:x2=0:y2=0:px=0:py=0 'sonst liefert @
x1... Improper argument
60 !W.POS,0,20,3,70,20:!W.POS,1,14,8,76,23:!W.ACTI
VE,1:!W.ACTIVE,0:GOTO 80 'Fenstergrösse setzen
, aktivieren
70 !SCREENNEW 'Fenster zeichnen
80 PRINT:PRINT " Bewegen Sie das Fenster!":!P.MOU
E 'auf KLICK warten
90 !P.ASK,@px,@py 'Pointerposition ermitteln
100 !W.ASK,0,@x1,@y1,@x2,@y2 'Fensterposition erm
itteln
110 IF px=x1 AND py=y1 THEN GOSUB 150:!FRAMEMOVE:G
OTO 70 'Fenster bewegen
120 IF px=x2-1 AND py=y1 THEN !W.POS,0,1,1,78,23:G
OTO 80 'ganzer Bildschirm
130 IF px=x2-1 AND py=y2 THEN GOSUB 150:!FRAMESIZE
:GOTO 70 'Fenstergrösse ändern
140 GOTO 80
150 b$="":FOR t=1 TO 100:b$=b$+INKEY$:NEXT:RETURN
'Tastepuffer entleeren

```

GDM Window Nr. 1

GDM - Graphic Desk Manager  
Copyright 1986 Thomas M. Binzinger

Ready  
|hardcopy  
Ready  
|hardcopy  
Ready  
? #8,"a  
Ready  
|hardcopy

Listing Nr. 2

```

10 ' Dieses Programm zeigt, wie die einzelnen
20 ' Menueintraege abgefragt werden koennen
30 '
40 m$="/Hilfe/Ablage/Druck/Ende/":DEFINT p:p=0:px=
0:py=0
50 !M.ON,@m$
60 !P.ASK,@px,@py:!P.POS,px*1,MIN(py+1,23)
70 !M.RESET:!P.MOVE:!SCREENNEW:!M.ASK,@p
80 IF p=0 OR p>4 THEN 60
90 ' diese Datazeile enthaelt x-Position, Laenge u
nd Hoehe der Drop-Downs
100 DATA 3,15,20,10,20,5,18,13,10,25,25,10
110 RESTORE:FOR t=1 TO p:READ x,b,h:NEXT:x1=(x-1)*
8-1:x2=x1+b*8+1:y1=383:y2=y1-(h*16+2)
120 FOR a=0 TO 1:PLOT x1-a,y1,1:DRAW x1-a,y2:DRAW
x2+a,y2:DRAW x2+a,y1:NEXT a:WINDOW #7,x,x+b-1,2,2+
h-1:CLS #7:PRINT #7,CHR$(10)" Pull-Down!!"
130 GOTO 60

```

Knoepfe:

!B.ON



Hilfe Ablage Druck Ende

GDM Window Nr. 1

Pull-Down!!



GDM Wind

GDM Window Nr. 1

# CPC-Speicher optimal genutzt!

## Optimizer für CPC-BASIC-Programme

**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.**

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.

---

### 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

---

### • OPTI

---

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 OP-

TI.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.

---

### Optimierter WORDPROCESSOR

---

Das Programm Wordprocessor II aus dieser 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),h
   ochpos(6,1)
40 LOCATE 30,3:PRINT CHR$(24);" Optim
   izer by THBCS ";CHR$(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 Variab
   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$,3)="RE
   M" OR t$=":" THEN pro$(j)="" :GOT
   O 290
270 IF LEFT$(zeile$,4)="DATA" THEN 2
   90
280 GOSUB 1090
290 NEXT j
300
310 *** GOTO's und GOSUB's aendern *
   **
320
330 LOCATE 1,12:PRINT "Pass 2 working
   on line"
340 FOR j=0 TO zeilen
350 LOCATE 23,12:PRINT VAL(pro$(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 fl=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$(j-1,0)=h$:fl=1
530 NEXT j
540 IF fl=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 T
   HEN pr$(typ)=pr$(typ)+var$(i,1):
   GOTO 710
700 var$(i,1)=var$(i,1)+CHR$(chr(ty
   p)):chr(typ)=chr(typ)+1:IF chr(ty
   p)>122 THEN PRINT CHR$(7);"* Var
   iablenfehler *":END
710 IF typ>0 THEN var$(i,1)=var$(i,1
   )+typ$
720 NEXT
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)="D
   ATA" 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 var
   pos<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$<="z" OR INST
   R("0123456789.",t$)>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 ' *** Variablen-tabelle 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$(34)
230 w=0:PRINT:GOSUB 290:PRINT:PRINT"Ausgabe auf Drucker (j/n) ?"
240 CALL &BB06:IF INKEY(45)=0 THEN w=0:GOSUB 290
250 PRINT"Ausgabe auf Diskette/Kassette (j/n) ?":CALL &BB06
260 IF INKEY(45)=0 THEN OPENOUT na$+".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

```

# Graphic-Processor für den Schneider CPC 464

Ich möchte Ihnen hier ein Graphic-Utility für den Schneider CPC 464 vorstellen. Darin wurden 20 neue und nützliche Befehle implementiert. Als einige Höhepunkte wären die Circle-, die Paint-, die Shaperoutine und die Möglichkeit, mehr Farben als normalerweise gleichzeitig benutzbar, zu nennen.

Zuvor, bevor ich mich an die Erläuterung dieser neuen Kommandos mache, möchte ich bemerken, daß in dieser Graphikhilfe kein byte in irgendeiner Weise von einem anderen Programmierer abgekupfert wurde. Die Programmlogik entstand unabhängig von anderen auf dem Markt oder in Zeitschriften erschienenen Programmierhilfen. Lediglich einige Ideen hierzu wurden aufgegriffen.

## Grafikhilfe

Nach dem Eingeben des BASIC-Laders und dem Start desselben sollte man das solchermaßen generierte Maschinenprogramm mit 'SAVE"name", b, 39000, 1945, 39000' abspeichern, bevor man es aufruft.

Das Maschinenprogramm meldet sich, sofern es korrekt erzeugt wurde, mit seinem Namen und einem Hilfsmenü.

## RSX

Als RSX (Resident System eXtension — nicht 'Resistent' System eXtension, wie manche glauben) stehen nun die folgenden Befehle zur Verfügung.

- !GPEN, <graphicspen>: setzt die Farbe des Graphikstiftes. Parameter werden nicht auf ihre Korrektheit hin überprüft.

Beispiel: !GPEN,3: TAG: PRINT"  
Nun wird Pen 3 verwendet": TAG-OFF

## GPAPER

- !GPAPER, <graphicspaper>: setzt die Farbe des Graphikfensters/-hintergrundes. Parameter werden nicht überprüft.

Beispiel: !GPAPER,":CLG:REM  
Graphikfenster wird mit Ink 2 gelöscht

- !GMODE, <graphicsmode>: setzt die Zugriffszeit des Graphik-

stiftes auf den Graphikhintergrund; dabei sind folgende Möglichkeiten gegeben:

graphicsmode = 0 ==> Force-Modus,

1 ==> Xor-Modus,

2 ==> And-Modus,

3 ==> Or-Modus.

Parameter werden nicht überprüft.

Beispiel: !GMODE; 1: MOVE 0,0: DRAW 639, 399: DRAW 0,0: REM  
Linie wird durch einmaliges Dar-

```

100 '
110 ' Beispielprogramm fuer die Nutzung des Graphic-Processors
120 '
130 ' Autor: O. Welsch,
140 ' Stand: 25/10/86
150 '
160 '
170 MODE 1
180 '** Farben fuer die Border, Ink 0 und Ink 1 setzen fuer jeden der
190 '** sechs moeglichen Balken
200 $MULTICOLOR,1,1,26,2,11,26,11,11,3,14,11,19,19,11,18,9,11,18
210 PRINT TAB(9)"Demonstrationsprogramm"
220 PRINT TAB(16)"fuer den"
230 $CHARSIZE,0 '* elongierte Zeichen einschalten
240 PRINT" GRAPHIC-PROCESSOR"
250 $CHARSIZE '* elongierte Zeichen wieder ausschalten
260 INK 2,0:INK 3,23 '* "globalen" Inks 2 und 3 setzen (schwarz, blaugruen)
270 $GPAPER,0 '* Ink des Graphikhintergrunds festlegen
280 '** Huegel
290 $GPEN,2
300 $CIRCLE,200,-220,640,300,340,45
310 $GPEN,1 '* Ink 1 des Grafikvordergrunds ist abhaengig von Multicolor !
320 MOVE 320,0:$PAINT '* den Huegel mit dem Pen 1 fuellen
330 '** Fahnenmast
340 $GPEN,2
350 $POLYGON,150,80,154,350,162,350,166,80
360 $GPEN,3
370 MOVE 158,90:$PAINT '* Fahnenmast ausfuellen
380 '** Knopf auf Fahnenmast
390 $GPEN,2
400 $CIRCLE,158,355,12,5,200,160
410 MOVE 158,355:$PAINT
420 '** Flagge
430 MOVE 162,350:DRAWR 20,-50
440 $RECTANGLE,182,304,206,-112
450 MOVE 182,200:DRAWR -16,-122
460 '** Schneiderlogo
470 $GPEN,1
480 $CIRCLE,240,267,30,30,180,0
490 $CIRCLE,240,267,5,5,180,0
500 $POLYGON,240,297,280,297,280,272,240,272
510 $POLYGON,240,260,280,260,280,237,240,237
520 $CIRCLE,330,230,30,30,0,180
530 $CIRCLE,330,230,5,5,0,180
540 $POLYGON,330,260,290,260,290,237,330,237
550 $POLYGON,330,225,290,225,290,200,330,200
560 $GPEN,1 ' rot
570 MOVE 240,250:$PAINT '* linken Teil des Logos ausfuellen
580 $GPEN,2 ' schwarz
590 MOVE 330,220:$PAINT '* rechter Teil des Logos ausfuellen
600 $GPEN,3 ' Pastellblaugruen
610 MOVE 184,200:$PAINT '* Farbe der Flagge
620 GOTO 620

```



```

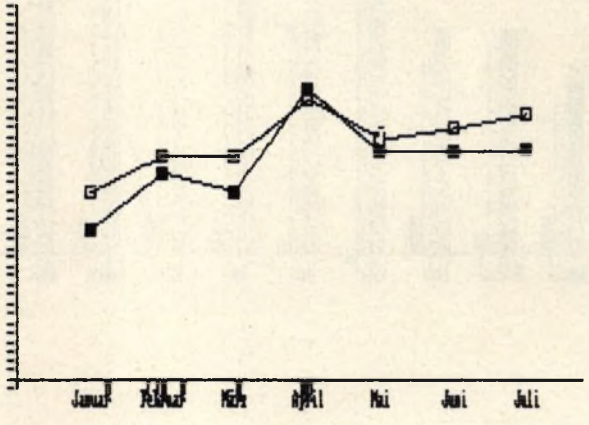
100 '
110 ' GRAPHIC - PROCESSOR 1 . 0
120 '
130 ' Autor: O.Welsch
140 ' Stand: Oktober 1985
150 '
160 '
170 IF HIMEM>38999 THEN MEMORY 38999
180 DEFINT a-z:DEFREAL s
190 RESTORE
200 DIM psum(31)
210 sum=0
220 FOR lin=1 TO 31
230 READ psum(lin)
240 sum=sum+psum(lin)
250 NEXT
260 IF sum<>207974 THEN PRINT"Error in checksums":PRINT"
270 sum=0:count=0:lin=0:er=0 *ERROR*":END
280 FOR adr=UNT(39000) TO &9FF0
290 READ byte$
300 byte=VAL("&"+byte$)
310 sum=sum+byte
320 POKE adr,byte
330 count=count+1
340 IF count=64 THEN GOSUB 420:sum=0:count=0
350 NEXT
360 GOSUB 420
370 IF er THEN PRINT"*ERROR*":END
380 SAVE"grprocessor.code",b,39000,1945,39000
390 PRINT"Link: CALL 39000"
400 END
410 '
420 ' PROCEDURE checksum
430 lin=lin+1
440 IF sum+psum(lin) THEN RETURN
450 er=-1
460 PRINT"Data error between";1000+(lin-1)*4;"and";
465 STOP 1000+(lin-1)*4+3
470 RETURN
480 '
490 ' Checksums
500 DATA 8833,6791,5516,7283,7269,4232,7617,6932
510 DATA 8462,5695,2090,7891,9053,9663,10185,8673
520 DATA 5527,13126,7721,8120,6980,7297,7289,7405
530 DATA 5536,4541,4261,4084,4271,4446,1185
540 '
1000 DATA 3E,C9,21,58,98,77,23,01,68,98,CD,D1,BC,C3,7B,9E
1001 DATA A6,98,C3,41,99,C3,47,99,C3,4D,99,C3,53,99,C3,73
1002 DATA 89,C3,DA,99,C3,16,9A,C3,1A,9A,C3,2A,9A,C3,2F,9A
1003 DATA C3,19,BD,C3,39,9A,C3,48,BB,C3,41,9A,C3,0C,9B,C3
1004 DATA 07,8B,C3,53,9B,C3,90,8B,C3,EF,9C,C3,7B,9E,47,50
1005 DATA 45,CE,47,50,41,50,45,D2,47,4D,4F,4A,C5,43,48,41
1006 DATA 52,52,45,41,C4,43,48,41,52,53,49,5A,C5,53,48,41
1007 DATA 50,C5,53,43,52,4F,4C,4C,55,DO,53,43,52,4F,4C,4C
1008 DATA 44,4F,57,CE,53,43,52,4F,4C,4C,45,46,D4,53,43,52
1009 DATA 4F,4C,4C,52,49,47,48,D4,46,52,41,4D,C5,4F,4E,2E
1010 DATA 42,52,45,41,4B,2E,53,54,4F,DO,4F,4E,2E,42,52,45
1011 DATA 41,4B,2E,43,4F,4E,D4,4D,55,4C,54,49,43,4F,4C,4F
1012 DATA D2,50,4F,4C,59,47,4F,CE,50,4F,4C,59,47,4F,4E,D2
1013 DATA 52,45,43,54,41,4E,47,4C,55,43,49,52,43,4C,55,50
1014 DATA 41,49,4E,D4,48,45,4C,DO,00,DD,7E,00,C3,DE,BB,DD
1015 DATA 7E,00,C3,E4,BB,DD,7E,00,C3,59,BC,11,FF,FF,3D,20
1016 DATA 10,CD,78,BB,CD,87,BB,30,08,CD,60,BB,30,03,16,00
1017 DATA 5F,DD,6E,00,DD,66,01,73,23,72,C9,4F,3A,1B,AC,FE
1018 DATA C9,20,03,3A,C8,B1,B7,C8,47,79,B7,28,50,OD,CO,DD
1019 DATA 7E,00,E6,01,B8,DO,4F,78,32,1B,AC,79,32,C8,B1,48
1020 DATA OD,OD,28,14,FE,01,20,08,11,88,44,21,22,11,18,25
1021 DATA 11,CC,33,21,22,11,18,1D,FE,02,20,08,11,80,40,21
1022 DATA 20,10,18,11,3D,20,08,11,CO,30,21,0C,03,18,06,11
1023 DATA FO,OF,21,20,10,ED,53,CF,B1,22,D1,B1,C9,3E,C9,48
1024 DATA 18,B6,3D,F8,DD,6E,00,DD,66,01,46,23,5E,23,56,1A
1025 DATA 13,05,32,02,9A,4F,OC,C5,D5,CD,13,9A,CD,1D,BC,CB
1026 DATA F4,CB,FC,D1,C1,E5,OD,20,07,OE,00,E1,CD,26,BC,E5
1027 DATA 1A,AE,77,13,CD,20,BC,10,ED,E1,C9,CF,1A,96,06,FF
1028 DATA 18,02,06,00,3A,90,B2,4F,CD,69,BB,79,D2,4D,BC,C3
1029 DATA 50,BC,11,02,00,18,03,11,FE,FF,CD,0B,BC,19,C3,05
1030 DATA BC,11,5E,C4,0E,FD,C3,45,BB,B7,CA,CE,9A,FE,13,DO
1031 DATA 4F,3D,07,16,00,5F,DD,19,21,D5,9A,EA,05,12,3E,14
1032 DATA 77,23,10,FC,E1,FD,21,E7,9A,41,DD,7E,00,E6,1F,32
1033 DATA 6C,9A,FD,7E,00,F6,40,77,23,DD,2B,DD,2B,10,EB,AF
1034 DATA 32,D4,9A,01,07,BC,ED,49,01,1F,BD,ED,49,21,98,9A
1035 DATA E5,CD,E6,BC,E1,11,A1,9A,06,81,CD,19,BD,C3,E0,BC
1036 DATA 00,00,00,00,00,00,00,00,00,00,D4,9A,34,7E,FE,06
1037 DATA 20,01,AF,77,06,16,10,FE,50,27,07,80,5F,23,19,01
1038 DATA 10,7F,ED,49,4E,ED,49,23,AF,ED,79,4E,ED,49,23,3C
1039 DATA ED,79,4E,ED,49,C9,21,98,9A,C3,E6,BC,00,00,00,00
1040 DATA 00,00,00,00,00,00,00,00,00,00,00,00,00,00,14
1041 DATA 04,15,1C,18,1D,OC,05,OD,16,08,17,1E,00,1F,OE,07
1042 DATA OF,12,02,13,1A,19,1B,0A,03,OB,01,08,09,10,11,11
1043 DATA 18,F9,18,03,11,00,F6,B7,C8,4F,OF,DF,47,7A,32,4E
1044 DATA 9B,7B,3E,34,9B,79,3D,07,16,00,5F,DD,19,DD,6E
1045 DATA 00,DD,66,01,DD,5E,02,DD,56,03,C5,18,CD,CO,BB
1046 DATA C1,11,FC,FF,DD,19,05,C8,DD,6E,00,DD,66,01,DD,5E
1047 DATA 02,DD,56,03,C5,CD,F6,BB,C1,18,E6,FE,04,CO,DD,E5
1048 DATA E1,06,04,5E,23,56,23,D5,10,F9,D1,E1,CD,CO,BB,D1
1049 DATA D5,21,00,00,CD,F9,BB,D1,E1,E5,D5,11,00,CO,CD,F9
1050 DATA BB,D1,AF,67,6F,ED,52,EB,67,6F,CD,F9,BB,D1,AF,67

```

```

1051 DATA 6F,ED,52,57,5F,C3,F9,BB,01,00,00,C5,D1,FE,04,28
1052 DATA 1D,FE,06,CO,DD,E5,E1,5E,23,56,23,4E,23,46,23,E5
1053 DATA DD,E1,C5,E1,CD,38,9C,E5,C1,EB,CD,38,9C,EB,DD,E5
1054 DATA E1,D5,C5,06,04,5E,23,56,23,D5,10,F9,CD,CO,BB,ED
1055 DATA 53,74,9E,22,76,9E,D1,E1,CD,C9,BB,D1,E1,ED,53,70
1056 DATA 9E,22,72,9E,E1,E5,CD,48,9C,ED,5B,70,9E,CD,D4,9C
1057 DATA EB,E1,E5,D5,CD,44,9C,ED,5B,72,9E,CD,D4,9C,D1,CD
1058 DATA CO,BB,C1,03,21,68,01,B7,ED,42,20,02,47,4F,C5,E1
1059 DATA E5,CD,48,9C,ED,5B,70,9E,CD,D4,9C,EB,E1,E5,D5,CD
1060 DATA 44,9C,ED,5B,72,9E,CD,D4,9C,D1,CD,F6,BB,C1,E1,E5
1061 DATA B7,ED,42,20,CE,E1,ED,5B,74,9E,2A,76,9E,C3,C9,BB
1062 DATA D5,11,68,01,B7,ED,52,30,FC,19,D1,C9,11,5A,00,19
1063 DATA 11,68,01,B7,ED,52,30,01,19,E5,11,B4,00,B7,ED,52
1064 DATA 30,01,19,7D,FE,5B,38,03,3E,B4,95,32,77,9C,AF,21
1065 DATA B4,00,D1,ED,52,30,01,3D,67,DD,21,79,9C,DD,6E,00
1066 DATA C9,00,04,08,OD,11,16,1A,1F,23,28,2C,30,35,39,3D
1067 DATA 42,46,4A,4F,53,57,5B,5F,64,68,6C,70,74,78,7C,80
1068 DATA 83,87,8B,8F,92,96,9A,9D,A1,A4,A7,AB,AE,B1,B5,B8
1069 DATA BB,BE,C1,C4,C6,C9,CC,CF,D1,D4,D6,D9,DB,DD,DF,E2
1070 DATA E4,E6,E8,E9,EB,ED,EE,FO,F2,F3,F4,F6,F7,F8,F9,FA
1071 DATA FB,FC,FD,FE,FF,FF,FF,FF,FF,FF,FF,E5,7D,21,00
1072 DATA 00,06,08,OF,30,01,19,CB,3C,CB,1D,10,F6,F1,3C,CO
1073 DATA EB,67,6F,B7,ED,52,C9,CD,06,B9,32,7A,9E,ED,73,78
1074 DATA 9E,CD,FC,15,CD,FC,16,2D,56,9E,ED,53,70,9E,22,72
1075 DATA 9E,CD,04,18,4F,CD,0A,18,B9,CA,56,9E,32,72,9D,32
1076 DATA 22,9D,32,CC,9D,DD,2A,78,9E,01,B0,BE,ED,5B,8D,B0
1077 DATA A1,00,42,ED,52,30,1F,2A,89,AE,ED,52,D5,11,00,02
1078 DATA ED,52,D1,38,11,01,20,40,2A,89,AE,ED,52,38,04,ED
1079 DATA 4B,89,AE,D5,DD,E1,DD,F9,ED,43,5D,9D,AF,E5,F5,33

```



```

1080 DATA .21,00,00,39,11,18,BE,ED,52,DA,56,9E,ED,5B,70,9E
1081 DATA 2A,72,9E,CD,A9,OB,CD,82,OC,FE,00,28,0A,2A,70,9E
1082 DATA 23,22,70,9E,C3,38,9E,3A,38,B3,47,CD,68,OC,ED,5B
1083 DATA 70,9E,ED,53,74,9E,2A,30,B3,B7,ED,52,28,1A,2A,72
1084 DATA 9E,1B,D5,CD,A9,OB,CD,82,OC,FE,00,20,0A,3A,28,B3
1085 DATA 47,CD,68,OC,D1,18,OB,D1,ED,5B,70,9E,ED,53,76,9E
1086 DATA 2A,32,B3,B7,ED,52,28,1A,2A,72,9E,13,D5,CD,A9,OB
1087 DATA CD,82,OC,FE,00,20,0A,3A,38,B3,47,CD,68,OC,D1,18
1088 DATA DB,D1,ED,5B,72,9E,2A,34,B3,B7,ED,52,28,1F,21,72
1089 DATA 9E,34,2A,74,9E,22,70,9E,2A,76,9E,ED,5B,70,9E,B7
1090 DATA ED,52,38,05,OE,10,C3,60,9E,21,72,9E,35,ED,5B,72
1091 DATA 9E,2A,36,B3,B7,ED,52,28,1F,21,72,9E,35,2A,74,9E
1092 DATA 22,70,9E,2A,76,9E,ED,5B,70,9E,B7,ED,52,38,05,OE
1093 DATA 20,C3,60,9E,21,72,9E,34,2A,76,9E,23,23,22,70,9E
1094 DATA 3B,F1,E1,5C,4F,E6,03,57,ED,53,74,9E,79,E6,OC,OF
1095 DATA OF,67,22,76,9E,79,E6,OF,FE,10,28,9C,CO,C5,ED,7B
1096 DATA 78,9E,3A,7A,9E,C3,OC,B9,2A,76,9E,7C,ED,5B,74,9E
1097 DATA 63,07,07,B2,B1,C3,55,9D,00,00,00,00,00,00,00
1098 DATA 00,00,00,21,87,9E,7E,B7,C8,CD,5A,BB,23,18,F7,OD
1099 DATA 0A,OD,0A,3C,20,47,52,41,50,48,49,43,50,52,4F,43
1100 DATA 45,53,53,4F,52,20,31,2E,30,20,3E,OD,0A,77,72,69
1101 DATA 74,74,65,6E,20,31,39,38,35,20,62,78,20,4F,2E,57
1102 DATA 65,6C,73,63,68,OD,0A,OD,0A,43,6F,6D,6D,61,6E,64
1103 DATA 73,3A,OD,0A,2D,20,47,50,45,4E,2C,67,70,65,6E,OD
1104 DATA 0A,2D,20,47,50,41,50,45,52,2C,67,70,61,70,65,72
1105 DATA OD,0A,2D,20,47,4D,4F,44,45,2C,67,6D,6F,64,65,OD
1106 DATA 0A,2D,20,43,48,41,52,52,45,41,44,2C,40,76,61,72
1107 DATA 25,OD,0A,2D,20,43,48,41,52,53,49,5A,45,20,5B,2C
1108 DATA 6D,6F,64,65,5D,OD,0A,2D,20,53,48,41,50,45,2C,40
1109 DATA 76,61,72,24,OD,0A,2D,20,53,43,52,4F,4C,4C,55,50
1110 DATA 2F,44,4F,57,4E,2F,4C,45,46,54,2F,52,49,47,48,54
1111 DATA OD,0A,2D,20,46,52,41,4D,45,OD,0A,2D,20,4F,4E,2E
1112 DATA 42,52,45,41,4B,2E,53,54,4F,50,2F,43,4F,4E,5A,OD
1113 DATA 0A,2D,20,4D,55,4C,54,49,43,4F,4C,4F,52,20,5B,62
1114 DATA 2C,69,2C,69,5D,5B,2C,2E,2E,2E,5D,OD,0A,2D,20,50
1115 DATA 4F,4C,59,47,4F,4E,2F,52,2C,78,31,CD,79,31,20,5B
1116 DATA 2C,2E,2E,2E,2C,78,31,36,2C,79,31,36,5D,OD,0A,2D
1117 DATA 20,52,45,43,54,41,4E,47,4C,45,2C,78,2C,78,2C,64
1118 DATA 78,2C,64,79,OD,0A,2D,20,43,49,52,43,4C,45,2C,6D
1119 DATA 78,2C,6D,79,2C,78,2C,72,79,20,5B,2C,77,31,2C
1120 DATA 77,32,5D,OD,0A,2D,20,50,41,49,4E,54,OD,0A,2D,20
1121 DATA 48,45,4C,50,OD,0A,OD,0A,00,00

```

überschreiben trotz (oder gerade wegen) der gleichen Farbe wieder gelöscht.

- !CHARREAD, <var%>: Hiermit kann eine Position auf dem Bildschirm ausgelesen werden. Vor Aufruf von Charread muß die gewünschte Position mittels 'LOCATE x,y' lokalisiert und die Variable 'var' vom Typ Integer deklariert werden. Nach dem Ausprung aus Charread steht in der

ßer oder gleich dem momentanen Mode entsprechend dargestellt werden sollen. Dies bedeutet: in Mode 2 sind Zeichen der Größe Mode 1 oder Mode 0 möglich, in Mode 1 nur Zeichen der Größe Mode 0. Das Zurückstellen auf den Original-Modus erfolgt durch '!CHARSIZE' ohne Parameter.

Die Anzahl der Parameter (eines oder keiner) werden überwacht.

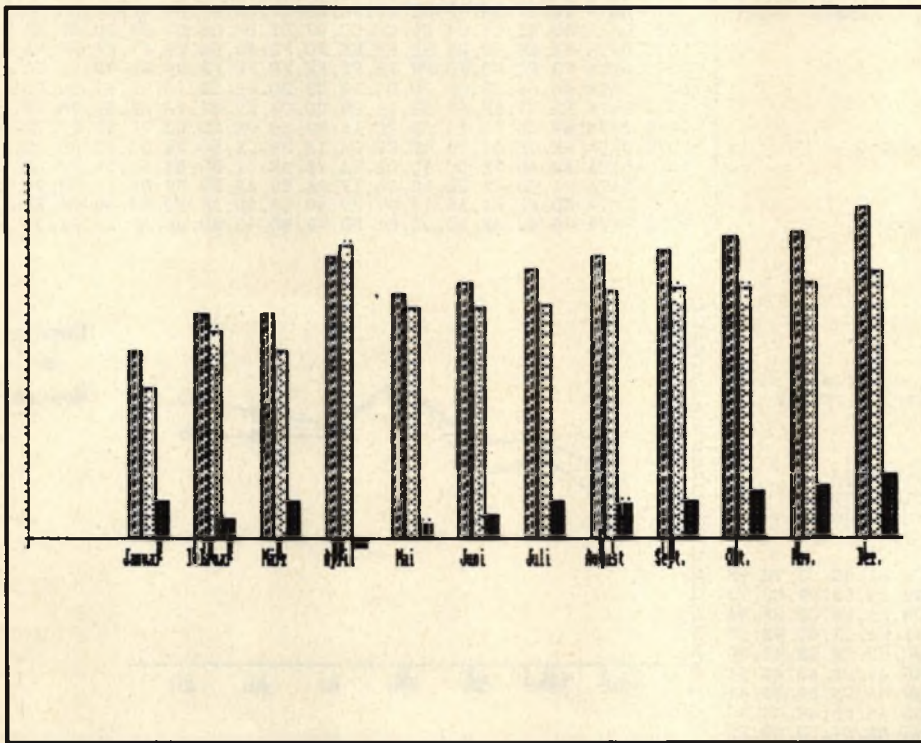
Werden mehr als zwei Parame-

pe auf max. 255. Im ersten Byte dieses Datenblocks steht die Anzahl der horizontalen Bildpunkte, dann folgen die eigentlichen Datenbytes.

## DATA-Zeilen

Um ein Shape zu generieren, sollte man der Einfachheit halber den beigegeführten Shape-Generator benutzen. Nach dem Einfügen der DATA-Zeilen und dessen Start fragt der Generator nach dem Mode, in dem das Shape später dargestellt werden soll. Zu beachten ist, daß in der ersten DATA-Zeile die Größe des Shapes in Bildpunkten anzugeben ist. Diese Größe wird dann als Berechnungsgrundlage für die Größe des Datenblocks genommen, wobei noch zusätzlich der Bildschirmmode berücksichtigt wird. Nach Beendigung dieser Programmhilfe steht in der Variablen 'shape\$' das Shape zur Verfügung.

Ausgehend von der Grafik-Cursorposition wird nach dem Kommando 'SHAPE' der Datenblock in dem angegebenen String im Xor-Modus mit dem darunterstehenden Bildschirminhalt verknüpft und nach links und unten ausgegeben. Da dies, wie schon gesagt, im



Variablen der ASCII-Wert des vorgefundenen Zeichens. Konnte das Zeichen nicht identifiziert werden, oder wurde eine Cursorposition außerhalb des aktuellen Bildschirmfensters angegeben, so steht in der Variablen der Wert -1. Die Anzahl der mitgegebenen Parameter wird nicht überwacht, ebenso, ob nun der Inhalt der Variablen oder ein Zeiger auf die Variable angeführt wurde. In diesem Falle ist ihr Inhalt ungültig.

Beispiel: LOCATE 1,1: !CHAR READ, \$char% : IF char%=-1 THEN PRINT " Zeichen nicht erkannt!" ELSE weiter...

## CHARSIZE

- !CHARSIZE, <mode>: Dank dieses Befehles können Sie nun innerhalb eines Bildschirmmodus mehrere Zeichengrößen zusammenmischen.

Beschränkt ist diese Methode allerdings nur auf Zeichen, die grö-

ter angegeben, so wird die Routine erfolglos verlassen. Parameterauswertung: Inhalt MODE 1.

Beispiel: MODE 2: PRINT "Normal": !CHARSIZE,1:PRINT "Elongiert": !CHARSIZE,0:PRINT "Abermals elongiert": !CHARSIZE:PRINT " und wieder normal"

## MODE

Etwas gilt es zu beachten: Vor jeder Modusänderung mittels 'MODE' sollte die ursprüngliche Zeichengröße durch '!CHARSIZE' wiederhergestellt werden. Ansonsten kann es zu unangenehmen Effekten kommen.

- !SHAPE; <\$var%>: Diese Routine gestattet es Ihnen, auf dem Schneider Shapes darzustellen. Die Daten für diese Shapes werden in einem String übergeben. Somit beschränkt sich die Anzahl der möglichen Bildpunkte je Sha-

## SHAPE

Xor-Modus geschieht, bleibt der Inhalt unter dem Shape restaurierbar. Schreibt man nämlich das Shape ein zweites Mal über die gleiche Position, so wird der alte Inhalt — sofern in der Zwischenzeit nichts anderes normal darüber geschrieben wurde — wieder rekonstruiert. Bedingt durch den Xor-Modus kann es zu seltsamen chaotischen Effekten kommen. Um eine hohe Schreibgeschwindigkeit zu erreichen, wird das Shape in ganzen Bytes in den Bildspeicher geschrieben. So vermeidet man langwierige Schiebe- und Rotieroperationen. Dadurch kann es zu 'ruckenden' Shapes kommen.

Wird kein Parameter beim Aufruf dieser Routine angegeben, so wird diese erfolglos verlassen. Die Gültigkeit der Parameter wird nicht überprüft.

Beispiel: MOVE 320,200:SHAPE;  
\$shape\$ (verwendet werden kann  
das Beispiel, das im Shape-  
Generator steht)

## SCROLLUP

- !SCROLLUP!/SCROLLDOWN:  
Mittels der hier angegebenen  
Kommandos kann der Inhalt des  
aktuellen Textfensters hinauf- und  
hinuntergeschoben werden. Die  
neu erscheinende Zeile wird da-  
bei immer gelöscht.

Die alte Cursorposition bleibt  
bestehen. Eventuelle Parameter  
werden nicht beachtet.

Beispiel: LOCATE 1,1: !SCROLL-  
DOWN:PRINT"Neue Zeile!"

- !SCROLLEFT!/SCROLLRIGHT:  
Hiermit wird der gesamte Bild-  
schirm um jeweils zwei Bytes nach  
links oder rechts gescrollt. Ge-  
scrollt wird 'um die Eck herum'. Es  
wird folglich keine Spalte dabei  
gelöscht.

Eventuelle Parameter werden  
nicht beachtet.

Beispiel: 10 !SCROLLEFT:!FRA-  
ME:GOTO 10:REM Spielerei

```

100 *
110 ** SHAPE-GENERATOR **
120 *
130 * Autor: O.Welsch
140 * Stand: Oktober 1985
150 *
160 *
170 PRINT"SHAPE-Generator"
180 PRINT"by O.Welsch"
190 INPUT"Modus : ",modus
200 bits=2^(modus+1)-1
210 DIM color(15),mask(bits)
220 IF modus=0 THEN RESTORE 530:GOSUB 470:RESTORE 430
230 IF modus=1 THEN RESTORE 540:GOSUB 470:RESTORE 440
240 IF modus=2 THEN RESTORE 550:GOSUB 470:RESTORE 450
250 FOR i=0 TO bits
260   READ code$
270   mask(i)=VAL("&"+code$)
280 NEXT
290 RESTORE 570 'Restore auf Datenblock fuer Shape
300 READ x,y
310 x=x/(bits+1)
320 count=0:value=0:p=2
330 shape$=CHR$(x)+STRING$(x*y,CHR$(0))
340 READ a$
350 IF a$="" THEN PRINT"Shape defined":END
360 FOR i=1 TO LEN(a$)
370   value=value OR (color(ASC(MID$(a$,i,1)) AND 15) AND mask(count))
380   count=count+1
390 IF count>bits THEN count=0:MID$(shape$,p,1)=CHR$(value):p=p+1:value=0
400 NEXT
410 GOTO 340
420 * Bitmasken der jeweiligen Modi
430 DATA aa,55
440 DATA 88,44,22,11
450 DATA 80,40,20,10,08,04,02,01
460 * Codierte Inks einlesen
470 FOR i=0 TO 15
480   READ code
490   color(i)=code
500 NEXT
510 RETURN
520 * Codierte Inks der jeweiligen Modi
530 DATA 0,192,12,204,48,240,60,252,3,195,15,207,51,243,63,255
540 DATA 0,240,15,255,0,240,15,255,0,240,15,255,0,240,15,255
550 DATA 0,255,0,255,0,255,0,255,0,255,0,255,0,255,0,255
560 * Beispiel-Shape: ein abgewandelter Commodore-Ballon
570 DATA 24,21
580 DATA "000000001111111100000000"
590 DATA "00000111111111111100000"
600 DATA "0001111111111111111000"
610 DATA "00111122221111111111100"
620 DATA "01111122221111111111110"
630 DATA "11111221111111111111111"
640 DATA "1111122221333311111111"
650 DATA "1111112222133331111111"
660 DATA "111111111111111133111111"
670 DATA "05111111111133331111150"
680 DATA "05011111111133331111050"
690 DATA "00500111111111111100500"
700 DATA "000500011111111110005000"
710 DATA "0000500011111111000050000"
720 DATA "000005000011110000500000"
730 DATA "000000500001100005000000"
740 DATA "0000000500011000050000000"
750 DATA "000000005000000500000000"
760 DATA "000000000555555000000000"
770 DATA "000000000555555000000000"
780 DATA "000000000555555000000000"
790 DATA ""

```

- !FRAME: Diese Routine wartet  
lediglich auf den nächsten Bild-  
rücklauf. Dies ist dann wichtig,  
wenn ein Bildaufbau flimmerfrei  
erfolgen soll.

Eventuelle Parameter werden  
nicht beachtet.

## SCROLLEFT

Beispiel: siehe '!SCROLLEFT'

-!ON.BREAKCONT: Diese Funk-  
tion ermöglicht es Ihnen, die  
Break-Taste nun nahezu völlig ab-  
zuschalten. Nur nahezu völlig, da  
im Betriebssystem des Schneider  
der Cassetten-Manager und der  
Editor direkt auf die Tastatur zu-  
rückgreifen und darum bei Kasset-  
tenoperationen oder bei Input-  
Befehlen die Esc-Taste weiterhin  
berücksichtigt wird. Mit 'ON.  
BREAK.STOP' können Unterbre-  
chungen wieder zugelassen wer-  
den.

Angegebene Parameter werden  
nicht beachtet.

Beispiel: 10 !ON.BREAKCONT  
20 WHILE INKEY\$=,," :PRINT"  
Unterbrechungen gesperrt  
":WEND

30 !ON.BREAK:STOP  
40 PRINT" Unterbrechungen  
wieder zugelassen": GOTO 40

!ON.BREAK:STOP: Mit dem Auf-  
ruf dieser Funktion wird eine Un-  
terbrechung durch die ESC-Taste  
wieder zugelassen, nachdem sie  
durch 'ON:BREAK:CONT' verhin-  
dert wurde.

Angegebene Parameter werden  
nicht beachtet.

Beispiel: siehe 'ON.BREAK.  
CONT'

## MULTICOLOR

--!MULTICOLOR, border1, ink1,  
ink2, border2, ink21, ink22,...  
border6, ink61, ink62: Hiermit ist  
Ihnen nun die Möglichkeit zur Ver-  
fügung gestellt, mehr Farben als  
normalerweise üblich darzustel-  
len (14 Farben in Mode 2, 16 Far-  
ben in Mode 1 und 27 Farben in  
Mode 0). Dieses Mehr an Farben  
äußert sich in max. sechsfarbigem  
Streifen auf dem Bildschirm. Ohne  
mich in die technischen Details  
verlieren zu wollen, möchte ich  
nur etwas zu der Parameteranga-  
be und -übergabe sagen. Jeder  
dieser sechs Farbbalken enthält  
die Information zur Bildrandfarbe

(Border) und der Farbe der Pen 0 und 1. Somit ergeben sich maximal  $3 \times 6 = 18$  zu übergebende Parameter. Diese Anzahl darf nicht überschritten, wohl aber unterschritten werden. Angegeben werden jeweils die Inksfarben (0-31). Bei der Parameterübernahme durch das Programm werden diese Parameter Modulo 32 genommen, um auf diese Weise eine gültige Ink zu erhalten. Anschließend werden diese Inks in eine interne Farbaufbauliste übernommen. Alle Stifte mit einer anderen Nr. als 0 oder 1 werden während der Arbeit der Routine

x- und y-Koordinate eines Eckpunktes und den jeweiligen Kantenlängen in x- und y-Richtung, wobei durch die Vorzeichen die Ecke dieses Rechtecks festgelegt wird (also links/unten, rechts/oben etc.). Anschließend wird dieses Rechteck gezeichnet. Der Graphik-Cursor steht beim Aussprung auf der Koordinate der angegebenen Ecke. Ist die Parameteranzahl ungleich vier, so wird die Routine erfolglos verlassen.

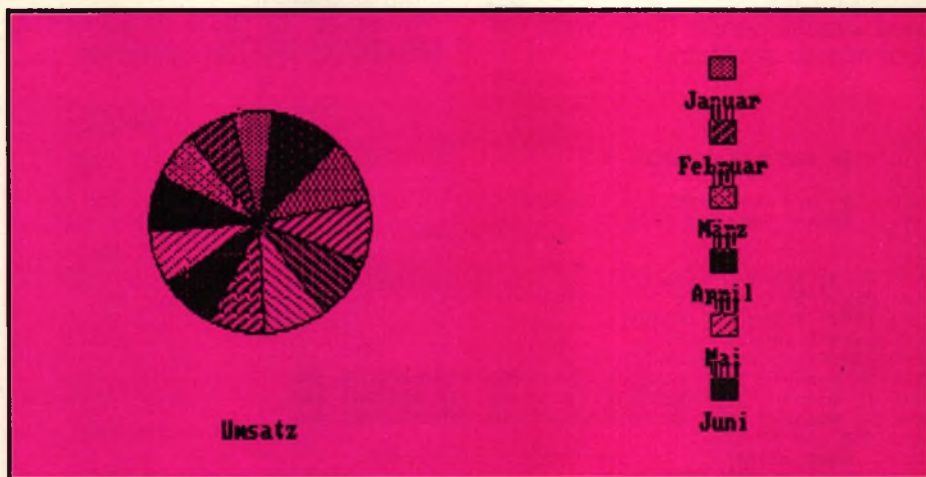
steckte' Flächen werden erkannt. Bevor die Routine aufgerufen wird, muß der Graphik-Cursor positioniert, der Graphik-Pen und das Graphik-Paper selektiert sein. Ausgehend von der momentanen Cursor-Position wird dann eine begrenzte Fläche, die die Farbe des Graphik-Papiers besitzt, mit der Farbe des Graphik-Stiftes gefüllt. Die Komplexität und die Größe der zu füllenden Fläche ist abhängig von dem zur Verfügung stehen-

## POLYGON

ne verändert. Werden keine Parameter angegeben, so wird die Routine wieder 'ausgehängt', ihre Abarbeitung beendet. Bei mehr als 18 Parametern wird die Routine ohne Ergebnis verlassen.

Beispiel: !MULTICOLOR, 26, 0, 26, 22, 4, 22, 18, 8, 18, 14, 12, 14, 10, 16, 10, 6, 20, 6

- !POLYGON/!POLYGONR, x1, y1, x2, y2, ...x16, y16: Ziel dieser Funktion ist es, mittels eines Kommandos ganze Linienzüge zeichnen zu können. Dazu folgt dem Namen der Routine eine Parameterliste, in der paarweise x- und y-Koordinaten folgen. Auf die erste Koordinate wird der Grafik-Cursor entweder absolut oder relativ gesetzt, zu allen weiteren angegebenen Koordinaten werden dann absolute oder relative Linien gezogen. Im wesentlichen entspricht diesem Kommando also eine Abfolge von DRAWs bzw. DRAWRs. Vorteil dieses Kommandos gegenüber der üblichen Methode: bei



Beispiel: !RECTANGLE, 80, 50, 560, 350

:REM zentriertes Rechteck im Bildschirm

- !CIRCLE, mx, my, rx, ry, w1, w2: Diese Routine gestattet es, eine Ellipse — einen Kreis dann natürlich auch — oder einen Kreisbogen zu zeichnen. Die Koordinate des Mittelpunktes wird durch die beiden ersten Parameter bestimmt, die Radien in x- und y-Richtung durch

## CIRCLE

die beiden folgenden. Die Angabe dieser vier Parameter ist Pflicht. Zusätzlich können noch zwei Winkel (in Altgrad) angegeben werden. Null Grad liegen dabei in zwölf Uhr, die folgenden Winkel wandern im Uhrzeigersinn rechts herum. Zwischen diesen beiden Winkeln wird ein Kreissegment gezeichnet. Bei einer falschen Anzahl von Parametern wird die Routine erfolglos verlassen.

Beispiel: !CIRCLE, 320, 0, 159, 199, 270, 90 :REM zeichnet einen nach unten offenen Halbkreis über den ganzen Bildschirm.

- !PAINT: Mit PAINT können endlich beliebige (!) Flächen mit einer Farbe gefüllt werden. Auch 'ver-

den Speicherplatz. Anmerken möchte ich hier, daß sich das Programm selbstständig einen freien Speicherbereich sucht. Standardmäßig ist dieser Speicherbereich der Prozessorstack, ansonsten der freie Bereich zwischen dem Ende des Variablenspeichers und dem Anfang der Strings. So kann es bei einem großen BASIC-Programm und einer komplexen Figur zweckmäßig sein, vorher den Speicher mit Hilfe der BASIC-Funktion FRE zu entrümpeln. Kann eine Figur nicht ganz gefüllt werden, so wird diese Routine unterbrochen.

## GPEN

Eventuelle Parameter werden nicht beachtet. Beispiel:

!GPEN,1:GPAPER,0:CLG!  
CIRCLE, 320,200,50,199:MOVE  
320,200:PAINT

-!HELP: Mit dem Kommando HELP können Sie ein Hilfsmenü aufrufen. Darin enthalten sind sämtliche hier erläuterten Befehle und ihre Formate. Parameter werden nicht beachtet.

Beispiel: !HELP

Ich hoffe, daß dieses Utility vielen eine hilfreiche Ergänzung ist. Einige Mängel des Schneiders sind dadurch allemal beseitigt.

O. W.

## DRAW

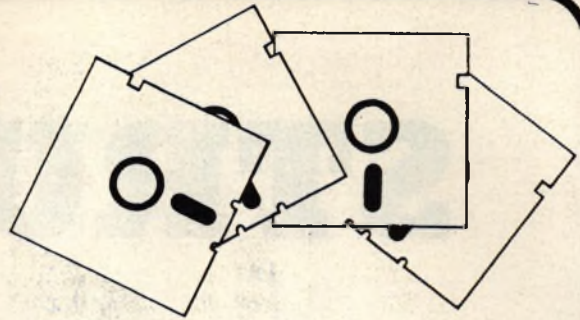
guter Ausnutzung der größtmöglichen Anzahl von Parametern ist diese Methode schneller. Bei ungerader Anzahl der Parameter wird diese Routine ohne Ergebnis verlassen.

Beispiel: !POLYGON, 80, 50, 560, 350, 80, 350, 560, 50, 80, 50

:REM zwei Dreiecke

- !RECTANGLE, x, y, dx, dy: Wie der Name vielleicht schon verrät, wird mit Hilfe dieses Kommandos ein Rechteck gezeichnet. Die vier Parameter, die stets angegeben werden müssen, bestehen aus der

# Die Software



## zum Schneider Sonderheft

Folgende Listings finden Sie in unserem Software-Service. Sparen Sie sich das eintippen.

### Die Themen:

- New York
- Der Grafikmacher
- Wordprocessor
- Spriteeditor
- Hintergrund-Uhr
- Banner
- Struk-List
- Wordprocessor II
- Optimizer
- Mineralien
- Skypolt
- Zeichen-Generator
- D-List
- Kalender
- Maussteuerung
- Rekursion in BASIC
- »GEM« GDM
- Apfelmännchen
- Labeldrucker
- Maze
- Hyperplot
- Bild/Tel.
- Grafikprocessor

Über  
200 Kilobyte  
für nur  
49.— Diskette (3''-Disk)  
45.— Cassette

23 Superprogramme  
für nur  
49.— Diskette (3''-Disk)  
45.— Cassette

**Sofort bestellen**

Einsenden an: **SOFTWARE TEAM, Joachim Günster, Mühlenstraße 12, 5431 Boden**

**Ja, Ihr Angebot hat mich überzeugt und ich bestelle:**

Softwarepaket I (Disk) 49.—/Stk.

Softwarepaket I (Cassette) 45.—/Stk.

Name: \_\_\_\_\_ Vorname: \_\_\_\_\_  
PLZ/Ort: \_\_\_\_\_ Straße: \_\_\_\_\_

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

# STRUKLIST

## Benutzte Variablen:

device : Nummer des Ausgabekanals  
 nleer : Anzahl der Leerzeichen, die vor jeder Zeile  
 ausgegeben werden  
 token\$(126) : Basic-Befehle als Strings  
 funktion\$(127) : Basic-Funktionen als Strings  
 i,k : Laufvariablen  
 a : Anfangsadresse des Basic-Speichers  
 zn : Aktuelle Zeilennummer  
 aneu : Anfangsadresse der nächsten Zeile  
 zeile\$ : Ausgabezeile als String (wird am Schluß ausge-  
 geben)  
 anfang : Flag, das führende Leerzeichen unterdrückt  
 diff : Gibt an, um wieviele Zeichen ein- bzw. ausgerückt  
 wird  
 aus\$ : Nimmt das gerade gelesene Basic-Kommando usw. auf  
 byte : Gerade gelesenes Byte  
 druck\$ : Wird zum Ausgeben der Zeile am Schluß benutzt, da  
 der Schneider nicht in der Lage ist, Strings über  
 40 Zeichen vernünftig auszugeben

## Funktionen:

fn deek : Liest zwei aufeinanderfolgende Bytes als 16 Bit-Zahl  
 fn druck\$ : Wandelt Zahlen in Strings um; bei positiven Zahlen  
 wird kein führendes Leerzeichen ausgegeben

```

60000 REM Strukturiertes Listing fuer Drucker mit
      Zeilenbreitenbegrenzung
60010 REM Copyright FTB Software Cooperation
60020 REM
60030 REM Frank Thielen
60060 REM
60070 REM Thomas Barndt
60100 REM
60110 REM
60120 REM Kontrollzeichen (in Stringkonstanten) ko
      ennen nicht ausgegeben werden
60130 REM
60140 REM
60150 REM Programm listet sich nicht selbst; soll
      dies doch geschehen, die
60160 REM Bedingung "zn<60000" (Zeilennummer unter
      60000) in der Zeile 60450
60170 REM entfernen
60180 REM
60190 REM
60200 REM
  
```

Das  
nützliche  
Listing

Dieses Programm listet Programme auf Bildschirm der Drucker. Dabei werden Zeilennummern rechtsbündig ausgegeben sowie For-Next- und While-Wend-Schleifen eingerückt, wenn dies möglich ist.

Man lädt das Programm zu diesem Zweck mit MERGE zu dem zu listenden Programm und startet es mit RUN 60000. Für die Ausgabe auf Drucker sollte in Zeile 60210 device=8 stehen, andernfalls device=0 für Ausgabe auf den Bildschirm. In Zeile 60450 sollten Sie das erste REM und den Doppelpunkt entfernen, damit sich das Programm nicht selbst listet. In den Zeilen 60220 und 60230 müssen Soll- und Istbreite für die Ausgabe angegeben werden.

Struklist liest den BASIC-Speicher aus und wandelt dabei die Tokens (Kürzel für BASIC-Befehle) in die tatsächliche Zeichendarstellung um (sie stehen am Ende des Programms). Werte von Konstanten, wie z.B. in „a=3.14“, speichert der Schneider nicht

## Pascalstrukturen in BASIC

in Form von ASCII-Zeichen ab, d.h. nicht als fortlaufende Zeichen wie „3“, „1“, „4“, wie dies die meisten Rechner tun, er wählt vielmehr eine von der Größe der Konstanten abhängige Darstellung und vermindert so den Speicherbedarf und erhöht die Ausführungsgeschwindigkeit. Alle dezimalen Konstanten unter 10 werden dabei in nur einem Byte abgelegt; dezimale Werte unter 256 werden durch ein Byte gekennzeichnet, auf das dann der eigentliche Wert folgt. Dies geschieht analog mit dezimalen, hexadezimalen und binären Konstanten, die mit einem Kennbyte beginnen und in zwei weiteren Bytes den eigentlichen Wert enthalten.

Fließkommazahlen werden grundsätzlich als 5 Byte-Sequenzen abgelegt, wobei die Darstellung

## 5 Byte Sequenz

der Ablage der Variableninhalte entspricht. Auch dies ist ein Grund für die hohe Geschwindigkeit der Programmausführung des Schneiders, weil bei einer Zuweisung diese fünf Byte nur übertragen werden brauchen, während die meisten anderen Computer die ASCII-Darstellung von Zahlen erst noch in das Variablen-Format umwandeln müssen.

## Der Trick

Auch bei der Ablage der Zeilennummer hinter GOTO wird ein Trick angewendet: Hier steht nicht etwa die Zeilennummer, sondern ein Zeiger, der auf den Beginn der angesprungenen Zeile im Speicher zeigt. Bei der Ausführung kann also dann der Ausführungszeiger direkt auf diese Speicherstelle gesetzt werden; viele andere Rechner müssen in dieser Situation den gesamten BASIC-Speicher nach der anzuspringenden Zeile absuchen. Deshalb werden Sprünge an den Programmanfang bei den meisten Rechnern schneller durchgeführt als an das Ende (die gesuchte Zeile wird schneller gefunden), während der Schneider immer gleich schnell springt.

```

60210 device=8 ' Fuer Ausgabe auf Bildschirm device=0
60220 sollbreite=50 ' Anzahl der gewünschten Zeichen pro Zeile auf dem Ausgabegeraet (mindestens 18)
60230 istbreite=80 ' Anzahl der tatsaechlichen (physikalisch vorhandenen) Zeichen pro Zeile auf dem Ausgabegeraet (mindestens sollbreite)
60240 INK 0,26:INK 1,0:PAPER 0:PEN 1:BORDER 26
60250 nleer=0 ' Anzahl der Leerzeichen, die vor jeder Zeile ausgegeben werden
60260 DIM token$(126),funktion$(127)
60270 RESTORE 61360
60280 FOR i=0 TO 126
60290   READ token$(i)
60300 NEXT i
60310 FOR i=0 TO 29
60320   READ funktion$(i)
60330 NEXT i
60340 FOR i=64 TO 72
60350   READ funktion$(i)
60360 NEXT i
60370 FOR i=113 TO 127
60380   READ funktion$(i)

60390 NEXT i
60400 DEF FN deek(i)=PEEK(i)+PEEK(i+1)*256
60410 DEF FN druck$(i)=MID$(STR$(i),1-(i>=0))
60420 a=368
60430 zn=FN deek(a+2)
60440 aneu=a+FN deek(a)
60450 WHILE zn>0 : 'AND zn<60000 ' Das erste ' muss entfernt werden
60460   zeile$=""
60470   anfang=0
60480   diff=0
60490   i=a+4
60500   WHILE i<=aneu-2
60510     aus$=""
60520     byte=PEEK(i)
60530     IF NOT anfang AND byte=32 THEN 61160 ELSE anfang=-1 ' Leerzeichen am Zeilenanfang ueberlesen (kann entfallen, falls selbst eingegebene Leerzeichen stehen bleiben sollen)
60540     IF byte<>192 AND byte<>197 THEN 60610
60550     IF byte=192 THEN aus$="" ELSE aus$="REM "
60560     FOR k=i+1 TO aneu-2 ' Text hinter REM oder Hochkomma, unveraendert ausdrucken
60570       aus$=aus$+CHR$(PEEK(k))
60580     NEXT k
60590     i=aneu-2
60600     GOTO 61160

```

```

60610 IF byte=1 THEN IF PEEK(i+1)<>192 AND PEE
K(i+1)<>151 THEN aus$="":GOTO 61160 ELS
E GOTO 61160
60620 IF byte<>124 THEN 60700
60630 REM RSX-Erweiterungen
60640 i=i+1:aus$="ö"
60650 WHILE PEEK(i)<128
60660   aus$=aus$+CHR$(PEEK(i)):i=i+1
60670 WEND
60680 aus$=aus$+CHR$(PEEK(i)-128)
60690 GOTO 61160

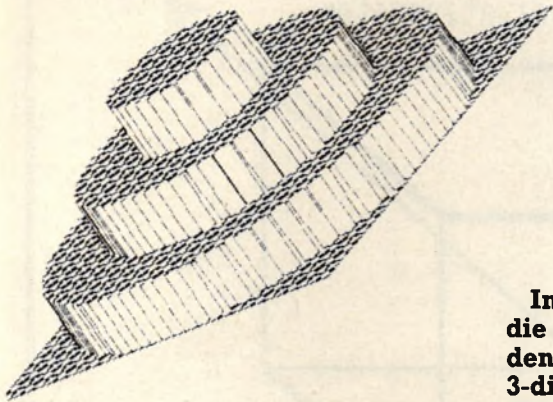
60700 IF byte>=14 AND byte<=23 THEN aus$=FN dr
uck$(byte-14):GOTO 61160 ' im Byte direk
t enthaltene Information (ohne Postbyte)
60710 IF byte=29 THEN aus$=FN druck$(FN deek(F
N deek(i+1)+3)):i=i+2:GOTO 61160 ' Point
er auf Sprungziel (Zeile) bei GOTO
60720 IF byte<>34 THEN 60820
60730 aus$=CHR$(34) ' Stringkonstante, alle fo
lgenden Zeichen bis zum naechsten Anfueh
rungszeichen oder zum Zeilenende direkt
verarbeiten
60740 i=i+1
60750 WHILE i<=aneu-2 AND PEEK(i)<>34
60760   byte=PEEK(i)
60770   IF byte>=32 THEN aus$=aus$+CHR$(byte)
ELSE aus$=aus$+"." ' nicht ausgebare
Kontrollzeichen (wie z.B. Bell (chr$(7
)) usw.) werden als Punkte ausgedruckt
60780   i=i+1
60790 WEND
60800 IF PEEK(i)=34 THEN aus$=aus$+CHR$(34)
60810 GOTO 61160
60820 IF byte>31 AND byte<128 THEN aus$=CHR$(b
yte):GOTO 61160 ' druckbares Zeichen
60830 IF byte=255 THEN aus$=funktion$(PEEK(i+1
)):i=i+1:GOTO 61160 ' Funktion
60840 IF byte<=127 THEN 60890
60850 aus$=token$(byte-128) ' Basic-Kommando
60860 IF byte=158 OR byte=214 THEN diff=diff+2
60870 IF byte=176 OR byte=213 THEN diff=diff-2
60880 GOTO 61160
60890 IF byte<2 OR byte>4 AND byte<11 OR byte>
13 THEN 61080
60900 REM Bezeichner (fuer Variable, Feld oder
Benutzerfunktion)
60910 REM 2: REAL ohne Typzeichen
60920 REM 3: STRING ohne Typzeichen
60930 REM 4: INTEGER ohne Typzeichen
60940 REM 11: REAL mit Typzeichen
60950 REM 12: REAL mit Typzeichen

60960 REM 13: INTEGER mit Typzeichen
60970 i=i+3
60980 aus$=""
60990 WHILE PEEK(i)<128
61000   aus$=aus$+CHR$(PEEK(i))
61010   i=i+1
61020 WEND
61030 aus$=aus$+CHR$(PEEK(i)-128)
61040 IF byte=4 THEN aus$=aus$+"!"
61050 IF byte=3 THEN aus$=aus$+"$"
61060 IF byte=2 THEN aus$=aus$+"%"
61070 GOTO 61160
61080 IF byte=25 THEN aus$=FN druck$(PEEK(i+1
)):i=i+1:GOTO 61160 ' Ein-Byte-Integer
61090 IF byte=27 THEN aus$="&X"+BIN$(FN deek(i
+1)):i=i+2:GOTO 61160 ' Zwei-Byte-Binaer
zahl
61100 IF byte=28 THEN aus$="&"+HEX$(FN deek(i+
1)):i=i+2:GOTO 61160 ' Zwei-Byte-Hexzahl
61110 IF byte=26 OR byte=30 THEN aus$=FN druck
$(FN deek(i+1)):i=i+2:GOTO 61160 ' Zwei-
Byte-Integer

61120 IF byte<>31 THEN 61160
61130 REM Reelle Zahl
61140 aus$=FN druck$((2^(PEEK(i+5)-145))*(6553
6+(PEEK(i+2)/128)+(PEEK(i+3)*2)+(PEEK(i+
4)*512)+(PEEK(i+1)/32800)))
61150 i=i+5
61160 zeile$=zeile$+aus$
61170 i=i+1
61180 WEND
61190 PRINT #device, USING "#####";zn;
61200 PRINT #device, " ";
61210 IF diff<0 THEN nleer=nleer+diff:IF nleer<0
THEN nleer=0
61220 leeranz=MIN(nleer+6,sollbreite-12)
61230 PRINT #device,STRING$(leeranz-6," ");
61240 WHILE LEN(zeile$)>sollbreite-leeranz
61250   PRINT #device,LEFT$(zeile$,sollbreite-le
eranz);:IF sollbreite<istbreite THEN PRI
NT #device
61260   zeile$=MID$(zeile$,sollbreite-leeranz+1)
61270   IF zeile$>"" THEN PRINT #device,STRING$(
leeranz," ");
61280 WEND
61290 IF zeile$>"" THEN PRINT #device,zeile$;:IF
LEN(zeile$)<sollbreite-leeranz OR sollbre
ite<istbreite THEN PRINT #device
61300 IF diff>0 THEN nleer=nleer+diff
61310 a=aneu
61320 zn=FN deek(a+2)
61330 aneu=a+FN deek(a)
61340 WEND
61350 END
61360 DATA "AFTER","AUTO","BORDER","CALL","CAT","C
HAIN","CLEAR","CLG","CLOSE IN","CLOSEOUT","CL
S","CONT","DATA","DEF","DEFINT","DEFREAL","D
EFSTR"
61370 DATA "DEG","DELETE","DIM","DRAW","DRAWR","ED
IT","ELSE","END","ENT","ENV","ERASE","ERROR"
,"EVERY"
61380 DATA "FOR","GOSUB","GOTO","IF","INK","INPUT"
,"KEY","LET","LINE","LIST","LOAD","LOCATE","
MEMORY","MERGE"
61390 DATA "MID$","MODE","MOVE","MOVER","NEXT","NE
W","ON","ON BREAK","ON ERROR GOTO"
61400 DATA "ON SG","OPENIN","OPENOUT","ORIGIN","OU
T","PAPER","PEN","PLOT","PLOT R","POKE","PRIN
T","RAD","RANDOMIZE","READ","RELEASE","R
EM","RENUM","RESTORE","RESUME","RETURN","RUN
"
61410 DATA "SAVE","SOUND","SPEED","STOP","SYMBOL"
,"TAG","TAGOFF","TROFF"
61420 DATA "TRON","WAIT","WEND","WHILE","WIDTH","W
INDOW","WRITE","ZONE","DI","EI","","","",""
,"","ERL","FN","SPC","STEP","SWAP","",""
,"TAB","THEN","TO","USING",">","=",">=","<","<
>","<=","+","-","*","/","^","ö"
61430 DATA "AND","MOD","OR","XOR","NOT"
61440 DATA "ABS","ASC","ATN","CHR$","CINT","COS","
CREAL","EXP","FIX","FRE","INKEY","INP","INT"
,"JOY","LEN","LOG","LOG10","LOWER$","PEEK","
REMAIN","SGN","SIN","SPACES$","SQ","SQR","STR
$","TAN","UNT","UPPER$","VAL"
61450 DATA "EOF","ERR","HIMEM","INKEY$","PI","RND"
,"TIME","XPOS","YPOS"
61460 DATA "BIN$","","HEX$","INSTR","LEFT$","MAX"
,"MIN","POS","RIGHT$","ROUND","STRING$","TEST
","TESTR","","VPOS"

```





# Grafikkurs für den Schneider, Teil III

In diesem Teil stoßen wir in die dritte Dimension vor: Es werden die Grundlagen der 3-dimensionalen Abbildung von Objekten auf dem Computerbildschirm besprochen und die Mög-

lichkeiten, die sich daraus ergeben: Drehung und Betrachtung von Objekten aus verschiedenen Blickwinkeln, Vergrößerung und Verkleinerung etc.

FÜR EINSTEIGER FÜR EINSTEIGER FÜR EINSTEIGER FÜR EINSTEIGER FÜR EINSTEIGER FÜR EINSTEIGER FÜR EINSTEIGER FÜR EINSTEIGER

Alle Grafiken, die wir bisher im Verlauf des Grafikkurses erstellt haben, hatten eines gemeinsam: Sie waren 2-dimensional. Das heißt, wir haben von vorneherein der Tatsache Rechnung getragen, daß wir auf einem Bildschirm, also einer zweidimensionalen Fläche zeichnen. Dieses Mal werden wir uns nun damit beschäftigen, wie 3-dimensionale Gebilde auf unserem 2-dimensionalen Bildschirm abgebildet werden können und welche Möglichkeiten sich dabei ergeben.

Betrachten Sie einmal Bild 1. Es zeigt einen perspektivisch dargestellten Würfel in Gitterdarstellung. Es wäre natürlich kein Problem, den Würfel mit unserem bisherigen Wissen auf dem Bildschirm zu zeichnen, indem wir die Koordinaten der einzelnen Linienanfänge und -enden bestimmen und einfach 'DRAW-en'. Doch dann hätten wir nichts Neues erreicht, da zwar der Betrachter den Eindruck von TIEFE hätte, aber wir programmäßig keine Möglichkeit hätten, z.B. die Perspektive, in der der Würfel dargestellt wird, zu ändern. Wir müssen also einen anderen Weg finden, das Aussehen des Würfels zu speichern.

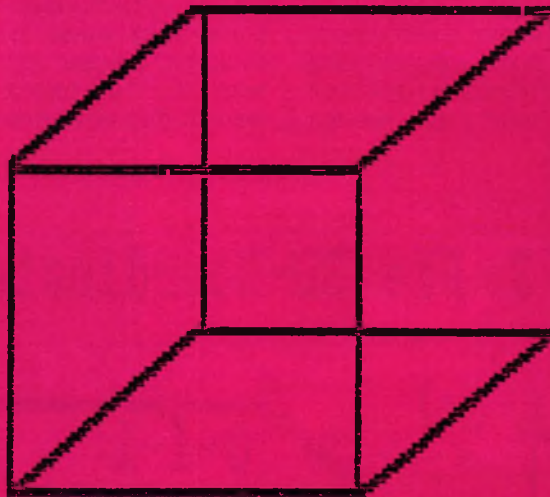
---

## TIEFE

---

Normalerweise arbeiten wir in einem Koordinatensystem mit 2-Achsen,  $x$  und  $y$ . Deshalb nennen wir es zwei-dimensional. Nun kommt aber noch eine dritte Dimension hinzu, die Tiefe. Wir brauchen also noch eine dritte

Bild 1: 3-dimensionaler Würfel



Achse dafür, die wir als Z-ACHSE bezeichnen werden. Aus Bild 2 wird ersichtlich, daß diese Achse (vorläufig jedenfalls) den rechten Winkel zwischen X- und Y-Achse teilt. Daß das aber keineswegs immer der Fall zu sein braucht, werden wir später noch sehen. Wenn wir in dem drei-dimensionalen Koordinatensystem arbeiten, hat jeder Punkt nicht mehr zwei, sondern drei Koordinaten: also nicht mehr  $P(x,y)$ , sondern  $P(x,y,z)$  (in dieser Reihenfolge werden die drei Koordinaten immer angegeben: X-Horizontal, Y-Vertikal, Z-Perspektive).

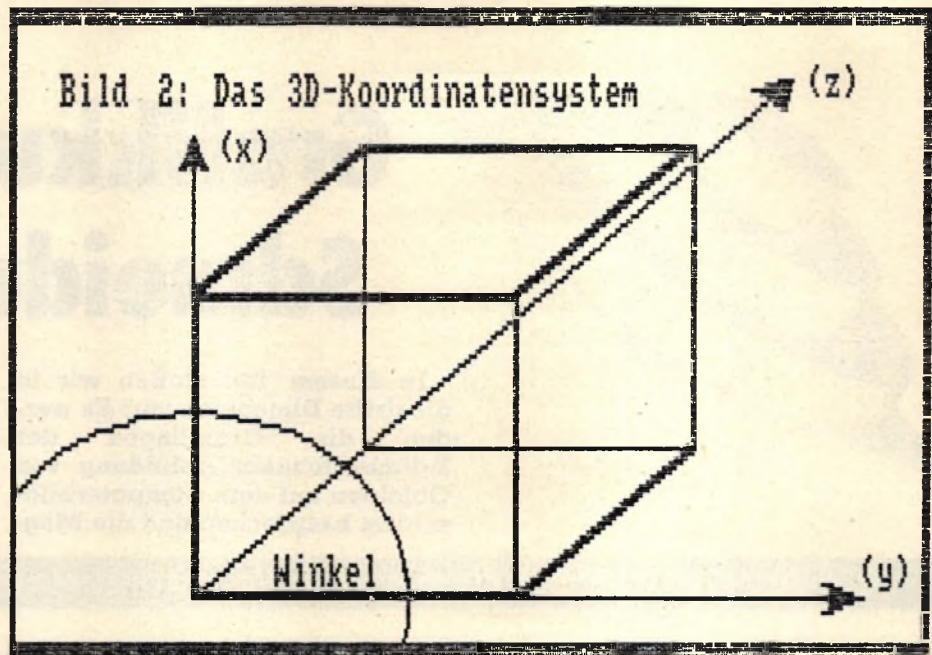
Die linke untere Ecke des Würfels stimmt mit dem Nullpunkt des Koordinatensystems überein  $(0,0,0)$ . Das ist zur Bestimmung der einzel-

nen Eck-Koordinaten günstiger, aber es ist keine Notwendigkeit; dieselbe Ecke könnte auch eine andere  $y$ -Koordinate haben, so daß der gesamte Würfel höher oder tiefer läge (z.B.  $0,2,0$ ), oder eine andere  $x$ -Koordinate, so daß der Würfel mehr rechts oder links läge (z.B.  $-2,0,0$ ). Wenn dieser Punkt eine andere  $z$ -Koordinate hätte, würde der Würfel näher oder weiter vom Betrachter entfernt liegen. Natürlich müssen dann auch alle anderen Ecken entsprechend verschoben werden. Die 3-D-Koordinaten für einige Ecken des Würfels zeigt Bild 3. 'Einige', weil die verborgenen Koordinaten nicht gezeigt werden. Alle Koordinaten finden Sie in den Data-Zeilen in Listing Nr. 1.

Der dargestellte Würfel ist eine Art 'Einheitswürfel' (in Anlehnung an den Einheitskreis), weil seine Seiten die Länge 1 haben. Diese Länge wurde gewählt, weil wir die einzelnen Koordinaten so leicht per Augenmaß bestimmen können. Wir hätten aber auch willkürlich jede andere Länge nehmen können (bei einem Rechteck, das nur halb so tief ist wie der Würfel, ergeben sich so Werte von 0,5 für die Z-Achse, anstelle der 1-Werte).

'Alles schön und gut, aber wie zeichnen wir etwas mit 3 Koordinaten mit den Grafikbefehlen, die doch nur 2 Argumente haben?', werden Sie jetzt berechtigterweise fragen. Das ist der Punkt, an dem die Mathematik ins Spiel kommt, und hier wird auch klar, warum wir bisher so auf Sinus und Kosinus herumgeritten sind. Diese beiden Funktionen brauchen wir nämlich in einer FORMEL, UM 3-D-KOORDINATEN IN 2-D-KOORDINATEN UMZURECHNEN.

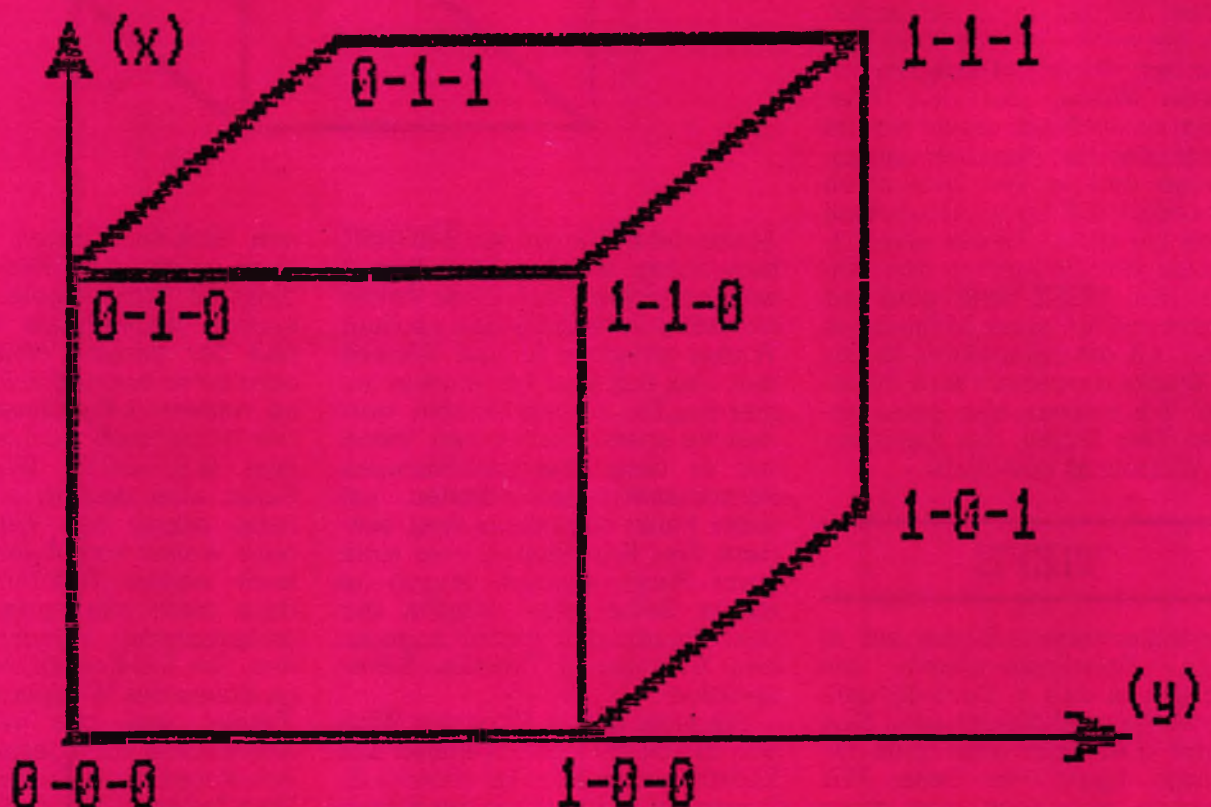
Um an diese Formel zu kommen, betrachten wir noch einmal ganz genau Bild 2: Wir sehen, daß der Eindruck, ein Punkt liege 'weiter



hinten', dadurch zustande kommt, daß wir den Punkt entlang der Z-Achse verschieben. Die Z-Achse kann aber auch in eine andere Richtung zeigen, als sie es in diesem Bild tut, indem wir einfach den im Bild eingetragenen Winkel

ändern. Uns sind nun also die  $x$ ,  $y$  und  $z$ -Koordinaten bekannt.  $x$ - und  $y$ -Koordinaten stimmen dabei bereits, in bezug auf eine zwei-dimensionale Darstellung jedenfalls. Wenn wir einfach den Quader mit diesen beiden Koordinaten zeich-

### Bild 3: Die 3D-Koordinaten



nen würden, und die Z-Achse unter den Tisch fallen lassen würden, erhielten wir ein Quadrat. Wir müssen nun aber zu diesen beiden Grund-Koordinaten noch eine Verschiebung hinzuaddieren, die sich aus dem Wert des Winkels und aus dem Wert Z (also der Entfernung selbst) ergibt. Der Wert für Z entspricht dem Radius eines Kreises. Der Winkel ist dabei IRGEND EIN Winkel dieses Kreises, und der Mittelpunkt des Kreises liegt auf dem Punkt x, y. Somit können wir den neuen 2-D-Punkt mit folgenden zwei Formeln ausrechnen:

$$x_{\text{neu}} = x + z \cdot \cos(\text{winkel})$$

$$y_{\text{neu}} = y + z \cdot \sin(\text{winkel})$$

### 3-dimensional

Das ist schon alles, was wir brauchen, um 3-dimensionale Koordinaten in 2-dimensionale Koordinaten umzurechnen. Vorausgesetzt, wir haben in einem Maßstab von 1 (wie in Bild 3) gearbeitet, müssen wir die neuen Werte nur noch an die aktuellen Bildschirmwerte anpassen. Dazu multiplizieren wir die Werte mit einem passenden Wert, um das gezeichnete Objekt in einer ausreichenden Größe darzustellen (z.B. mit 100), und addieren 320 zu den x-, und 200 zu den y-Werten, um das Objekt in die Bildschirmitte zu verschieben. In Listing Nr. 2 wird die praktische Anwendung für all das demonstriert.

Die Funktionsweise des Programmes ist leicht erklärt: Am Anfang stehen die Daten für die zwölf Linien, aus denen sich der Würfel zusammensetzt. In jeder Data-Zeile stehen dabei die Daten für eine Linie, d.h. Anfangspunkt in 3D-Koordinaten und der Endpunkt in 3D-Koordinaten. Das Programm wählt Bildschirm-Modus 1 und wählt einen Winkel von 45 Grad für die Z-Achse. Anschließend werden für alle zwölf Linien die Anfangs- und Endpunkte gelesen und in die entsprechenden Werte für ein zweidimensionales Koordinatensystem umgerechnet (Unterprogramm ab 350). Dazu wird unsere oben entwickelte Formel benutzt. Grafikcursor wird an die errechnete 2D-Startkoordinate gesetzt und eine Linie zur Endkoordinate gezogen (Zeile 310). Nachdem das 12 mal geschehen ist, wird nach einem neuen Darstellungswinkel gefragt, und das ganze beginnt von vorn.

Wenn Sie sich den Würfel unter Verwendung von verschiedenen Winkeln angeschaut haben, gibt es verschiedene Möglichkeiten, das Programm zu verändern: Wenn Sie folgende Zeilen einfügen, haben Sie die Möglichkeit, die Tiefe des Würfels frei zu bestimmen, indem Sie einen Faktor angeben, mit dem die jeweilige z-Koordinate multipliziert wird, bevor in 2D-Koordinaten umgerechnet wird:

```
205 zfaktor=1
```

```
235 z1=z1*zfaktor;z2=z2*zfaktor
```

```
325 locate 1,23:INPUT Z-Faktor=
,zfaktor
```

Ein ähnliches Ergebnis können Sie natürlich erzielen, indem Sie direkt in den DATA-Zeilen jeweils die 3. und 6. Zahl verändern, also Z. Eine weitere Manipulationsmöglichkeit bietet sich in bezug auf Vergrößerung/Verkleinerung des Objekts an. Dazu müssen lediglich ALLE Koordinaten vor der Umrechnung in 2D-Koordinaten mit einem Faktor multipliziert werden. Ist dieser Faktor >1, wird das Objekt vergrößert, ist er <1, wird es verkleinert. Der nächste Schritt ist, die Figur selbst zu verändern. Fügen Sie z.B. einmal folgende Data-Zeilen ein, und ändern Sie die Linien-Anzahl in Zeile 220 in 16:

```
185 DATA 0,0,0,0.5,0.5,-0.5
```

```
186 DATA 1,0,0,0.5,0.5,-0.5
```

```
187 DATA 0,1,0,0.5,0.5,-0.5
```

```
188 DATA 1,1,0,0.5,0.5,-0.5
```

Dadurch setzen Sie dem Würfel eine Spitze auf. Und wenn Sie sich daran satt gesehen haben, können Sie ja auch ganz neue Figuren definieren: In Bild 4 ist beispielsweise ein 'T' dargestellt. Denken Sie daran, auch die Linien-Anzahl in Zeile 220 zu ändern. In eigenen Kreationen können natürlich auch größere (oder kleinere) Zahlen als 0 und 1 verwendet werden.

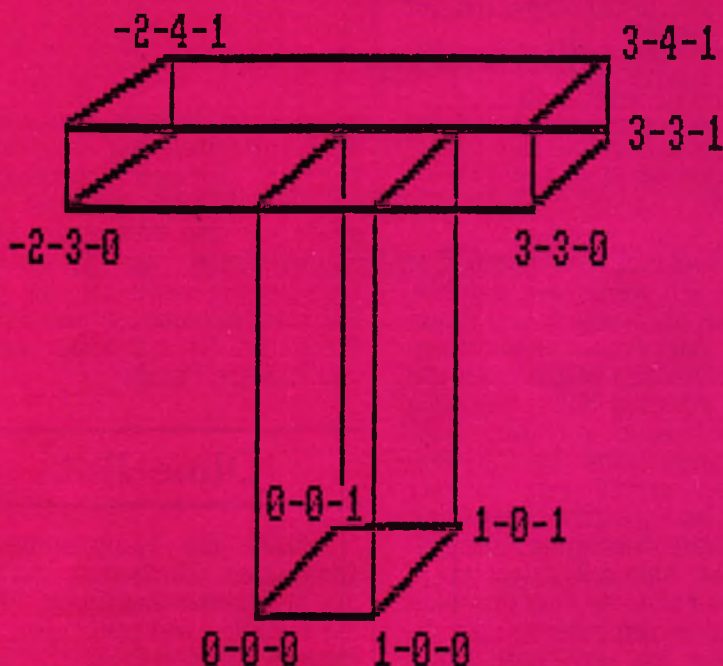
Damit Ihr Objekt optimal auf den Bildschirm paßt, sollten evtl. die Parameter in den Zeilen 260 bis 290 geändert werden (100 bestimmt dabei die Größe, 320/200 rücken das Bild in die Bildschirmitte). Im folgenden werden wir uns abschließend damit beschäftigen, wie man verdeckte Linien unterdrücken kann (so daß man nicht mehr durch die Objekte durchsehen kann), und wir werden mathematische Funktionen in 3D plotten. Bis jetzt haben wir 3-dimensionale Körper auf dem (2-dimensionalen) Bildschirm abgebildet. Dazu wurden natürlich Umrechnungsformeln benötigt, und zwar

$$sx = x + z \cdot \cos(\text{winkel})$$

$$sy = y + z \cdot \sin(\text{winkel})$$

Wir wollen diese Formeln nun dazu benutzen, um 3-D-Funktionen darzustellen. Es gibt natürlich noch andere (und sicher auch bessere) Methoden, um eine solche

Bild 4: 3D-Koordinaten eines 'T' (teilweise)



## Listing Nr. 1:

```

Die 3D-Koordinaten fuer den Wuerfel...
10 REM Datenformat:
20 REM x1,y1,z1,x2,y2,z2
30 REM
40 DATA 0,1,0,1,1,0
50 DATA 1,1,0,1,0,0
60 DATA 1,0,0,0,0,0
70 DATA 0,0,0,0,1,0
80 REM das war das vordere Quadrat
90 DATA 0,1,1,1,1,1
100 DATA 1,1,1,1,0,1
110 DATA 1,0,1,0,0,1
120 DATA 0,0,1,0,1,1
130 REM das war das hintere Quadrat
140 DATA 0,1,0,0,1,1
150 DATA 0,0,0,0,0,1
160 DATA 1,0,0,1,0,1
170 DATA 1,1,0,1,1,1
180 REM das waren die 4 Verbindungslinien

```

```

10 REM Testaufgabe 2 aus Teil 2:
20 REM Spirale
30 REM
40 MODE 1:INK 0,0:BORDER 0:INK 1,25
45 radius=1
50 DEG:FOR winkel=0 TO 360*10'(10 kompl.Kreise)
60 x=COS(winkel)*radius+320
70 y=SIN(winkel)*radius+200
80 PLOT x,y,1
90 radius=radius+0.1
100 NEXT winkel

```

## Listing Nr. 2:

```

Und das komplette Programm:
10 REM Datenformat:
20 REM x1,y1,z1,x2,y2,z2
30 REM
40 DATA 0,1,0,1,1,0
50 DATA 1,1,0,1,0,0
60 DATA 1,0,0,0,0,0
70 DATA 0,0,0,0,1,0
80 REM das war das vordere Quadrat
90 DATA 0,1,1,1,1,1
100 DATA 1,1,1,1,0,1
110 DATA 1,0,1,0,0,1
120 DATA 0,0,1,0,1,1
130 REM das war das hintere Quadrat
140 DATA 0,1,0,0,1,1
150 DATA 0,0,0,0,0,1
160 DATA 1,0,0,1,0,1
170 DATA 1,1,0,1,1,1
180 REM das waren die 4 Verbindungslinien
190 REM
200 MODE 1:DEG:winkel=45 '45 Grad Winkel
210 RESTORE 40
220 FOR linie=1 TO 12 '12 Linien sind zu zeichnen
230 READ x1,y1,z1,x2,y2,z2
240 GOSUB 350 'errechne 2-D Werte
250 REM nun Werte fuer Bildschirm korrigieren
260 xn1=xn1*100+320
270 xn2=xn2*100+320
280 yn1=yn1*100+200
290 yn2=yn2*100+200
300 REM und nun zeichnen
310 PLOT xn1,yn1,1:DRAW xn2,yn2
320 NEXT linie
330 LOCATE 1,24:INPUT "Winkel=",winkel
340 MODE 1:GOTO 210
350 xn1=x1+(z1*COS(winkel))
360 xn2=x2+(z2*COS(winkel))
370 yn1=y1+(z1*SIN(winkel))
380 yn2=y2+(z2*SIN(winkel))
390 RETURN

```

```

10 REM Testaufgabe 1 aus Teil 2:
20 REM Sinuswelle
30 REM
40 MODE 2:INK 0,0:BORDER 0:INK 1,25
50 x=0:DEG
60 FOR winkel=0 TO 639
70 y=SIN(winkel)*150+200
80 PLOT x,y,1
90 x=x+1
100 NEXT winkel

```

leicht nachvollziehbar zu sein. Die Figur, die ich darstellen möchte, könnte man als ausgebeulte Platte oder als Zuckerhut bezeichnen. Da ich am Anfang selbst nicht die geringste Ahnung hatte, wie man einen Zuckerhut auf den Bildschirm bringt, habe ich mit dem Einfachsten angefangen, mit der geraden Grundplatte selber. Eine gerade Platte hat immer denselben y-Wert, also z.B. 0 (sie ist ja eben). Dann braucht man nur noch eine Schleife, um alle x- und z-Koordinaten zu erzeugen, und

wenn dann diese so entstandenen 3D-Koordinaten mit unseren bekannten Formeln umgerechnet werden, können wir die Platte schon auf den Bildschirm plotten, vorausgesetzt, wir gleichen die Koordinaten noch an die realen Bildschirmkoordinaten an (100/110). Programm Nr. 1 arbeitet nach genau diesem Prinzip.

---

## Ohne Hut...

---

Aber die Hauptsache fehlt noch. Der Zuckerhut muß nun durch unterschiedliche Höhen dargestellt werden, denn die Punkte, aus denen er sich zusam-

menetzt, liegen auf einer anderen Höhe als die restlichen Punkte der Platte. Der Zuckerhut stellt eine Ausbeulung in der Platte dar, wobei sein Mittelpunkt am weitesten von der Platte entfernt ist und er am Rand immer weniger hoch wird. Wir brauchen also eine passende Funktion, die uns ein solches Bild liefert. Wenn Sie sich den Zuckerhut im Querschnitt von der Seite vorstellen, werden Sie feststellen, daß sein Profil eine ziemliche Ähnlichkeit mit der Nor-

---

## Normalparabel

---

malparabel hat. Es liegt also nahe, die Höhe jedes einzelnen Punktes mit Hilfe der Normalparabel (die Funktion  $x^2$  liefert die Normalparabel) auszurechnen. Wir haben dabei zwei Probleme. Das erste Problem: Die Normalparabel ist 'genau falsch herum', d.h., sie ist an der Stelle am niedrigsten, an der

der Zuckerhut am höchsten ist. Dieses Problem können wir lösen, indem wir die Formel umstellen, die Normalparabel praktisch spiegeln:

$$y = x^2$$

$$\Rightarrow y = (-\text{ABS}(x) + (x_{\text{max}} - x_{\text{min}})/2)^2$$

Und da  $x_{\text{max}}$  (5) und  $x_{\text{min}}$  (-5) feststehen, können wir die Formel noch weiter vereinfachen.

$$\Rightarrow y = (-\text{abs}(x) + 5)^2$$

Damit wäre das erste Problem gelöst. Diese neue Formel bringt uns (bei entsprechender Streckung/Stauchung) die richtige Zuckerhut-Form. Aber unser zweites Problem besteht noch, nämlich: Diese Formel liefert nur ein 2-D-Resultat ( $x$  und  $y$ ). Es geht also nun darum, wie man 3-D-Koordinaten erzeugen kann. Nun, das ist auch nicht so schwer, wie es aussieht (wenn man einmal weiß, wie's geht). Bei der obigen Formel fehlt die Tiefe, die  $z$ -Koordinate völlig, und nun stellt sich die Frage, wie man die  $y$ -Koordinate aus  $x$  und  $z$  errechnet.

## Die Lösung

Die Lösung ist so einfach, daß sie schon ein bißchen primitiv erscheint: Man multipliziert die beiden Koordinaten miteinander (das Produkt aus  $x$ - und  $z$ -Koordinaten ergibt die Höhe). Die komplette Formel für den Zuckerhut lautet demnach:

$$\Rightarrow y = (-\text{abs}(x) + 5)^2 + (-\text{abs}(z) + 5)^2$$

Für die  $z$ -Koordinate können wir die Formel unverändert übernehmen, da auch  $z_{\text{min}}$  und  $z_{\text{max}}$  -5 und 5 sind. Mit Programm Nr. 2, das mit unserer gerade herausgefundenen Formel arbeitet, kann man den fertigen Zuckerhut bewundern, allerdings müssen einige Besonderheiten im Programm beachtet werden: Erst einmal haben wir die Schrittweite (Zeilen 40,50) stark herabgesetzt. Dadurch werden mehr Punkte gezeichnet, und das Bild wird deutlicher. Außerdem wird  $y$  durch 400 dividiert, da sonst viel zu große Werte von der Formel kommen und wir nur einen kleinen Ausschnitt vom Zuckerhut sehen würden. Dadurch, daß wir  $y$  verkleinern, erreichen wir den Eindruck, der Betrachter würde aus sehr viel größerer Höhe auf den Zuckerhut heruntersehen (oder eben: der

```

10 ' Unterdrueckung von verdeckten Linien
20 ' bei einem Wuerfel in 3D Darstellung.
30 '
40 MODE 2:CALL &BC02:DIM f(8,5)
50 CLS
60 PRINT "Bitte geben Sie den Betrachterstandpunkt
   ein:"
70 PRINT"(z.B. x=50,y=50,z=20)":PRINT
80 INPUT "Bitte geben Sie die x-Koordinate ein:",b
   x
90 INPUT "Bitte geben Sie die y-Koordinate ein:",b
   y
100 INPUT "Bitte geben Sie die z-Koordinate ein:",
   bz
110 CLS
130 ' suche nach der verdeckten Kante
150 RESTORE
160 vdl=0 'verdeckte Linien Nummer
170 FOR t=1 TO 8 'alle 8 Kanten untersuchen
180   READ x,y,z,l1,l2
190   entf=SQR((x-bx)^2+(y-by)^2+(z-bz)^2)
200   'das war die Entfernungsformel
210   IF entf>vdl THEN vdl=t:vd=entf
220 NEXT
240 ' Hilfswerte aus Betrachterstandpunkt errechnen
260 bx=bx+0.000001 ' gegen Divison/0
270 h1=ATN(by/bx):h2=ATN(bz/bx) 'Hilfswerte h1 und
   h2
290 RESTORE
310 FOR t=1 TO 8 ' errechnen der 2D Koord.
320 '
330 READ x,y,z,l1,l2
350 ' liest x/y/z Koordinate und
360 ' Nummer der Verbindungslinien
380   xa=x
390   x=-xa*SIN(h1)+y*SIN(h1) ' x errechnen
400   ya=y
410   y=-xa*COS(h1)*SIN(h2)
420   y=y-ya*SIN(h1)*SIN(h2)+z*COS(h2)
440 ' nun die errechneten Werte in ein Feld schreib
   en
460 f(t,1)=x:f(t,2)=y:f(t,3)=z:f(t,4)=l1:f(t,5)=l2
470 NEXT t ' fuer alle 8
490 ' nun die Linien zeichnen
510 FOR t=1 TO 8
530 IF t=vdl THEN 620 'diese Linie ist nicht sic
   htbar!
540 xs=f(t,1):ys=f(t,2)
560 ' nur zeichnen, wenn auch die beiden Verbindun
   gs-
570 ' linen sichtbar sind!
590 nr=f(t,4):IF nr<>vdl THEN xe=f(nr,1):ye=f(nr,
   2):GOSUB 810
600 nr=f(t,5):IF nr<>vdl THEN xe=f(nr,1):ye=f(nr,
   2):GOSUB 810
620 NEXT t
630 ' das war es schon
650 WHILE INKEY$="":WEND:GOTO 50
670 ' nun folgen die Daten fuer den Wuerfel
680 ' x,y,z wie gehabt, dann die Nummern der
690 ' beiden Linien, die mit dieser Kante
700 ' verbunden sind
720 DATA 0,0,0,2,3
730 DATA 0,0,200,1,5
740 DATA 0,200,0,6,5
750 DATA 200,0,0,1,7
760 DATA 0,200,200,3,8
770 DATA 200,200,0,4,8
780 DATA 200,0,200,4,2
790 DATA 200,200,200,7,6
810 ' Linie von xs,ys nach xe,ye
820 ' in Bildmitte zentriert zeichnen
840 PLOT xs+320,(400-ys)-200:DRAW xe+320,(400-ye)-
   200,1:RETURN

```

Hut wäre viel kleiner). Und nicht zuletzt wenden wir auch noch einen Trick an, um das Bild plastischer erscheinen zu lassen. Grundsätzlich gilt: Je mehr Punkte wir zeichnen, um so plastischer wird das Bild, um so mehr Einzelheiten sind zu erkennen (deshalb haben wir ja auch die Schrittweite der beiden Schleifen geändert). Leider dauert das Zeichnen bei mehr Punkten auch länger, teilweise *sehr viel länger*. Wir behelfen uns hier mit einem Trick: Wir verbinden den Punkt, den wir gerade zeichnen, mit dem zuletzt gezeichneten Punkt und überbrücken so die Leerräume dazwischen. In Zeile 110 wird die Linie gezeichnet (aber nicht, wenn die Schleife das Plattenende erreicht hat, sonst würden ja zwei nicht nebeneinander liegende Punkte verbunden), und in Zeile 120 wird die Position des neuen Punktes abgespeichert, damit beim nächsten Punkt eine Linie dorthin gezogen werden kann. (Es ginge übrigens auch ohne diese Variablen, — überlegen Sie einmal, wie).

```

10 ' Listing 2: Eingedruckte Platte
20 DEG:MODE 2
30 winkel=2
40 FOR x=-5 TO 5 STEP 0.3
50 FOR z=-5 TO 5 STEP 0.5
60 y=((-ABS(z)+5)^2)*((-ABS(x)+5)^2)
70 y=y/400 'dadurch ganze Platte sichtbar
80 sx=x+z*COS(winkel)
90 sy=y+z*SIN(winkel)
100 sx=sx*32+320
110 sy=sy*200+50
120 PLOT sx, sy, 1: IF z <> -5 THEN DRAW xa, ya
130 xa=sx: ya=sy
140 NEXT z, x

```

## Verdeckte Linien

... sind ein Problem bei der 3D-Darstellung. Die notwendigen Formeln zum Unterdrücken dieser eigentlich nicht sichtbaren Linien sind teilweise sehr kompliziert und langsam, so daß man sehr leistungsstarke Rechner braucht, vor allem bei komplexeren Gebilden. Weil BASIC, auch das des Schneiders, nicht gerade Geschwindigkeitsrekorde aufstellt, solange man das Programm nicht kompiliert (in Maschinensprache über-

setzt), wollen wir uns mit einem einfacheren Beispiel begnügen, mit unserem altbekannten Würfel. Beim Würfel wird die Sache allerdings wirklich kinderleicht, denn es hat sich herausgestellt, daß der Eckpunkt des Würfels, der am weitesten vom Betrachter entfernt ist, gleichzeitig auch die nicht sichtbare Ecke ist. Alle drei mit diesem Eckpunkt verbundenen Linien sind dann auch nicht sichtbar. Da das Programm sehr ausführlich dokumentiert ist, möchte ich mir hier jeden weiteren Kommentar dazu sparen, schließlich haben wir die Grundprinzipien der 3D-Programmierung schon in der letzten Folge 'durchgekaut'. Nur auf eines sollten Sie achten: Wir drehen diesen Würfel *nicht*, indem ein *Winkel* geändert wird, *sondern* indem wir den *Betrachterstandpunkt* ändern. Daraus ergeben sich dann auch neue Umrechnungsformeln, die sich aus den alten Formeln herleiten lassen. Allerdings würde es zu weit führen, diese nun zu erklären.

Damit wären wir am Ende des Grafikkurses angelangt. Hoffentlich hat es Ihnen etwas Spaß gemacht.  
tmb

```

10 ' Listing Nr.1: Eine einfache Platte
20 DEG:MODE 2:INK 0,0:BORDER 0
30 winkel=2 'Betrachterwinkel
40 FOR x=-5 TO 5
50 FOR z=-5 TO 5
60 y=0 'da die Platte eben ist!!
80 sx=x+z*COS(winkel)
90 sy=y+z*SIN(winkel)
100 sx=sx*32+320
110 sy=sy*200+50
120 PLOT sx, sy, 1
140 NEXT z, x

```

## Listing 3

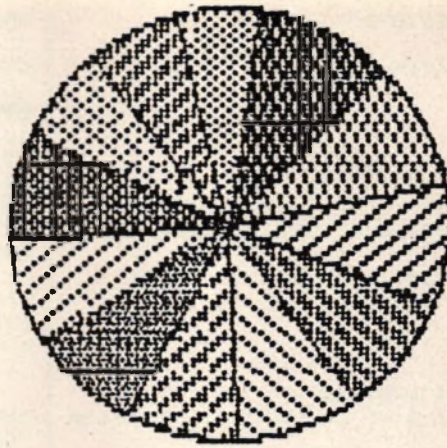
Listing Nr.3 stellt eine weitere Variante des Zuckerhutes dar, nur werden hier noch die einzelnen Höhenstufen unterschiedlich koloriert, einfach indem abgefragt wird, wie groß y ist, und dann eine dementsprechende Farbe ausgewählt wird.

```

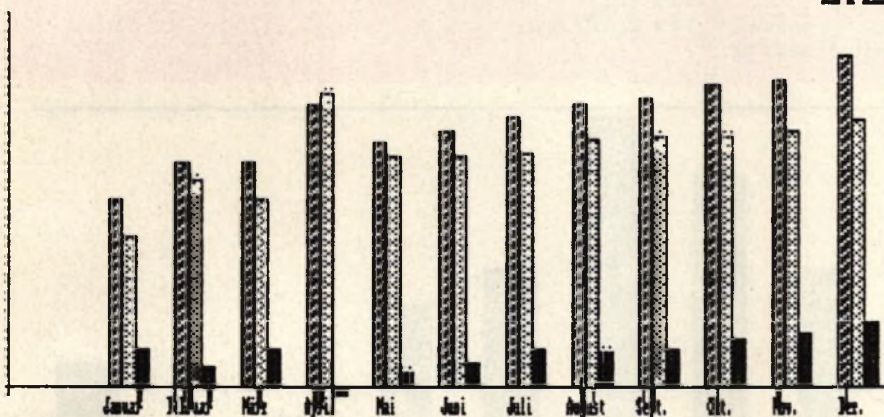
10 ' Listing 3: Platte in Farbe
20 DEG:MODE 1
30 winkel=2
40 FOR x=-5 TO 5 STEP 0.1
50 FOR z=-5 TO 5 STEP 0.2
60 y=((-ABS(z)+5)^2)*((-ABS(x)+5)^2)
70 y=y/400
80 sx=x+z*COS(winkel)
90 sy=y+z*SIN(winkel)
100 sx=sx*32+320
110 sy=sy*200+50
120 IF y>0.7 THEN f=3 ELSE IF y>0.05 THEN f=2 ELSE f=1 'Farbwahl
130 PLOT sx, sy, f
140 NEXT z, x

```

# Säulen- diagramme für Ihren CPC



## Umsatz



Mit diesem Programm können Sie Säulendiagramme mit einer beliebigen Anzahl von Säulen erzeugen. Sie können wählen, ob die eingegebenen Daten als Prozentwerte aufgefasst werden sollen. In diesem Fall warnt das Programm Sie vor einer Überschreitung der 100%-Grenze. Im Fall der Unterschreitung wird automatisch eine weitere Säule mit der Bezeichnung „sonst.“ erzeugt. Sie stellt die feh-

lenden Prozente zu 100% dar.

Bei normaler Auswertung der Daten werden schon bei der Eingabe das Maximum und das Minimum bestimmt. Diese bestimmen die Einteilung der Skala am linken Rand des Diagramms. Ist das Minimum größer als Null, so beginnt die Skala bei Null. Säulen mit negativen Werten werden von der Grundlinie nach unten gezeichnet. Jede Säule kann mit einem Text

bezeichnet werden. Dabei ist selbstverständlich darauf zu achten, daß dieser bei einer großen Anzahl von Säulen nicht allzu lang sein darf. Der einzugebende Kommentar wird unter dem Diagramm ausgegeben. Eine Ausgabe auf Drucker ist nicht vorgesehen, kann jedoch mit einer Hardcopy vorgenommen werden.

```

1 'Saeulendiagramme
2 '(c) 1986
3 'Thomas Barndt
4
10 MODE 2
20 minreal=-1E+38
30 ORIGIN 0,18,0,639,399,0
40 WINDOW #0,2,79,2,22
50 WINDOW #1,1,80,24,25
60 SYMBOL 255,255
70 PRINT:PRINTTAB(10)"Saeulendiagramme by THBCS
"
80 PRINT
90 INPUT"Anzahl der Saeulen:",anz
100 INPUT"Angaben in Prozent (J/N)";p$
110 p$=UPPER$(p$)
120 IF p$="J" THEN proz=-1:GOTO 140 ELSE proz=0
130 IF p$<>"N" THEN 100
140 IF proz THEN ybez$="Prozent" ELSE INPUT"Bezei-
chnung der Y-Achse:",ybez$

```

```

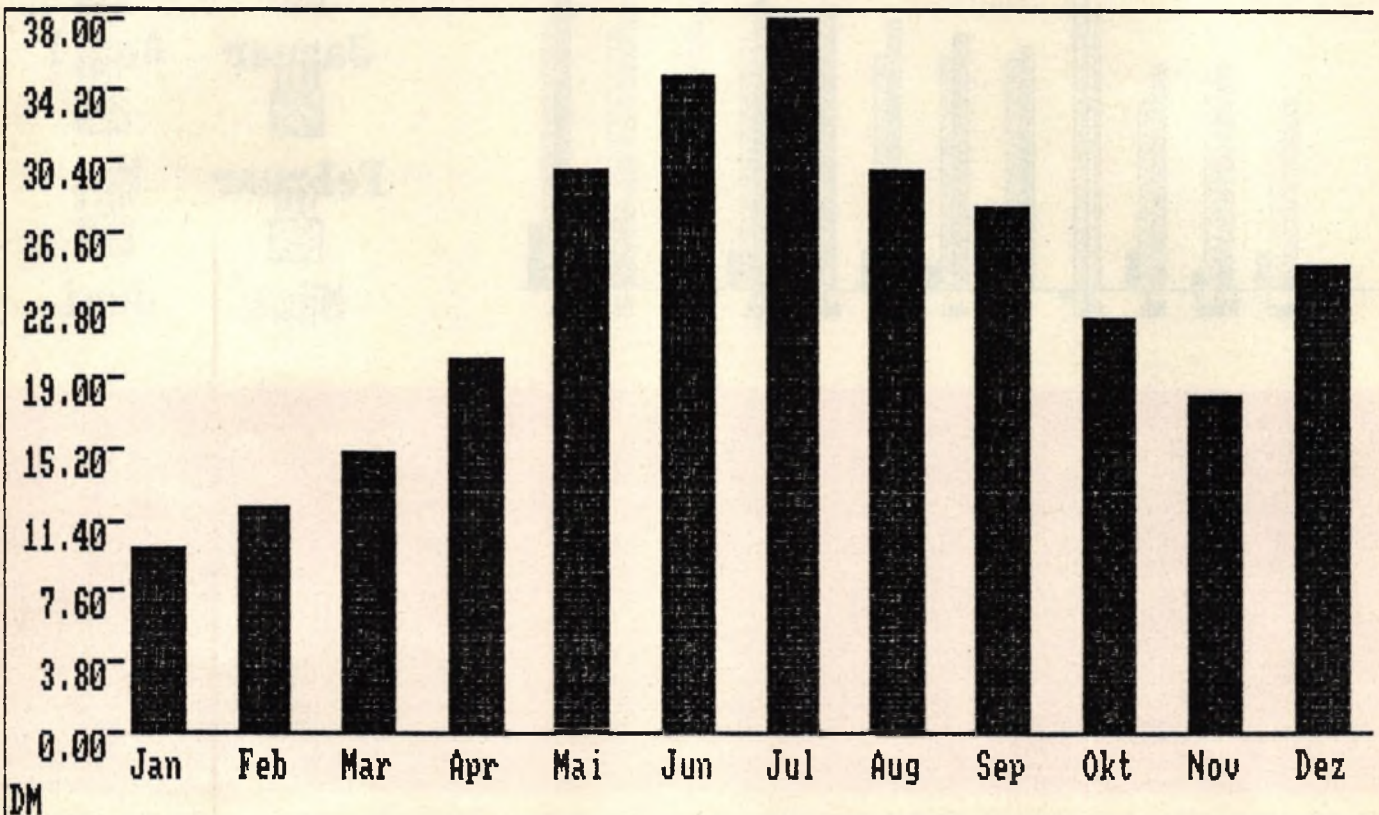
150 LINE INPUT"Kommentar:",kom$
160 '
170 ' ** Werte einlesen **
180 '
190 DIM wert(anz),bez$(anz)
200 anz=anz-1
210 sum=0:maxi=minreal:mini=-minreal
220 FOR i=0 TO anz
230 PRINT"Wert"i+1":";
240 INPUT"",wert(i)
250 INPUT"Bezeichnung:",bez$(i)
260 sum=sum+wert(i)
270 IF wert(i)>maxi THEN maxi=wert(i)
280 IF wert(i)<mini THEN mini=wert(i)
290 NEXT
300 sonstflag=0
310 IF NOT proz THEN 370
320 IF sum<100 THEN anz=anz+1:wert(anz)=100-sum:
bez$(anz)="sonst."
330 IF sum>100 THEN PRINT"Summe > 100% !! Einga-

```

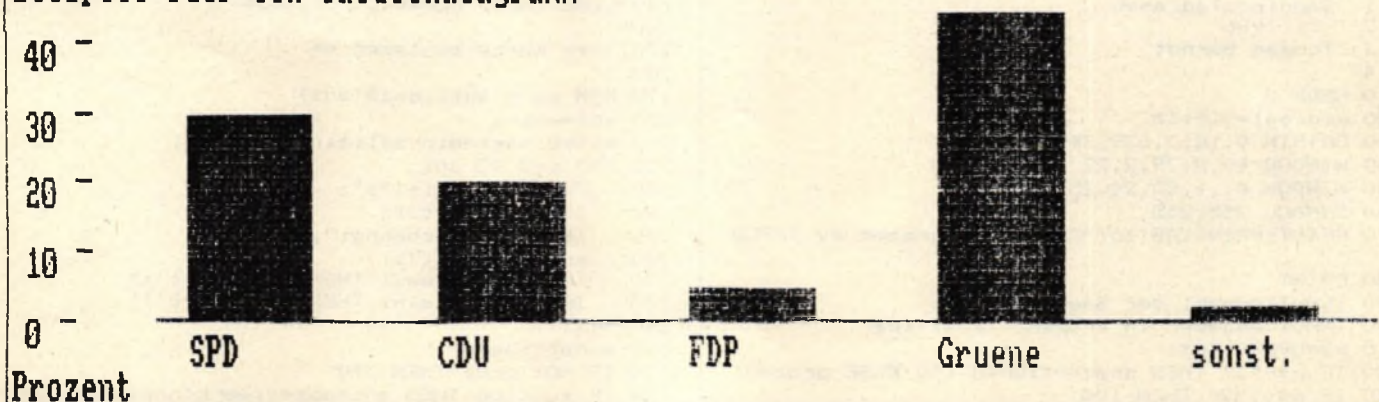
```

be wiederholen.":GOTO 210
340 '
350 ' ** Y-Achse einteilen **
360 '
370 CLS
380 PRINT#1,kom$
390 TAG
400 MOVE 3,34:PRINTybez$;
410 nul=58
420 MOVE 0,18:DRAW 0,380:DRAW 639,380:DRAW 639,1
      B:DRAW 0,18
430 IF proz THEN maxi=100:mini=0
440 IF mini>0 THEN mini=0
450 hoehe=376-nul
460 pro10=hoehe/10
470 TAG
480 mi$=STR$(mini):ma$=STR$(maxi)
490 lmi=INSTR(mi$,".")-1:IF lmi THEN lmi=LEN(mi$
      )
500 lma=INSTR(ma$,".")-1:IF lma THEN lma=LEN(ma$
      )
510 form$=STRING$(MAX(lmi,lma),"#")+".##"
520 FOR i=0 TO 10
530   MOVE 1,nul+i*pro10
540   IF NOT proz THEN PRINT USING form$;mini+((
      maxi-mini)/10)*i;:ELSE PRINT mini+((maxi-m
      ini)/10)*i;
550   PRINT CHR$(255);
560 NEXT
570 '
580 ' ** Diagramm zeichnen **
590 '
600 bereich=maxi-mini
610 xanf=(MAX(lmi,lma)+3)*8
620 breite=639-xanf
630 sbreite=breite/((anz+1)*2)
640 xanf=xanf+sbreite/2
650 IF mini<0 THEN nul=nul+hoehe/(bereich/(-mini
      ))
660 MOVE 70,nul:DRAW 639,nul
670 FOR s=0 TO anz
680   IF wert(s)<>0 THEN shoeh=nul+hoehe/(berei
      ch/wert(s)):schritt=SGN(wert(s))*2 ELSE sh
      oehe=nul:schritt=1
690   MOVE xanf+sbreite*(s*2),50
700   PRINTbez$(s);
710   FOR y=nul TO shoeh STEP schritt
720     MOVE xanf+sbreite*(s*2),y
730     DRAW xanf+sbreite*(s*2+1),y
740   NEXT
750 NEXT
760 GOTO 760

```



Beispiel fuer ein Saeulendiagramm



Wahlen 1987



# 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. tb

- |              |               |
|--------------|---------------|
| 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    |

```
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"
```

```
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"
```

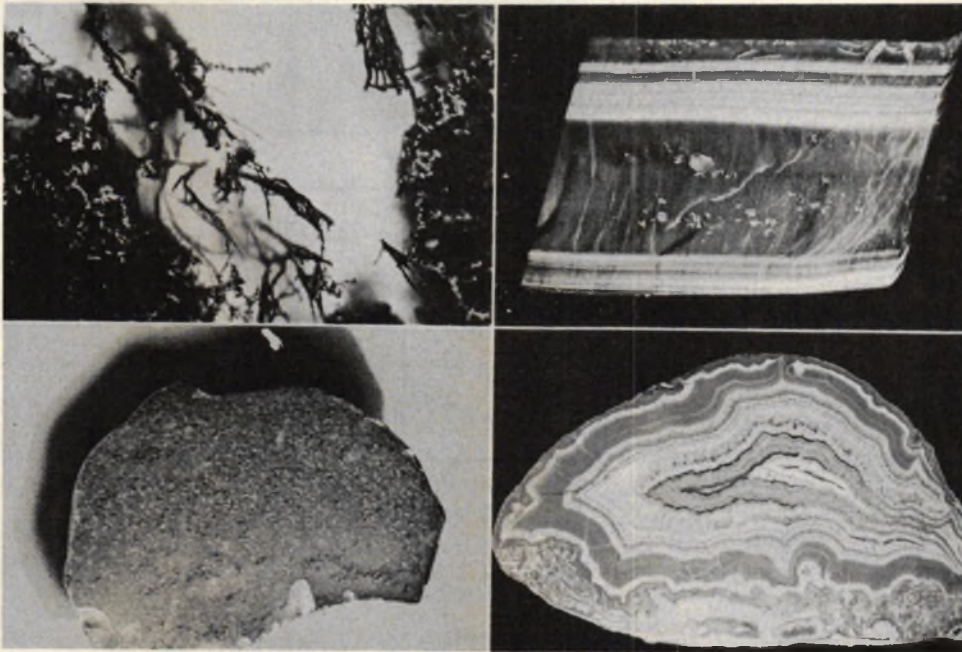
```

350 PRINT#1,"S - Suchen nach Namen"
360 PRINT#1,"A - Ausgabe aller Minerale"
370 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 550,1440,460,42
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$:ksys=VAL(w
ahl$)
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$(strich)
850 strich=strich+1
860 '
870 CLS#1:PRINT#1,TAB(10);"HAERTE (nach F.Mohs)"
:PRINT#1
880 PRINT#1," 1 Talk O Unbekannt"
890 PRINT#1," 2 Gips"
900 PRINT#1," 3 Kalzit"
910 PRINT#1," 4 Fluorit"
920 PRINT#1," 5 Apatit"
930 PRINT#1," 6 Orthoklas"
940 PRINT#1," 7 Quarz"
950 PRINT#1," 8 Topas"
960 PRINT#1," 9 Korund"
970 PRINT#1,"10 Diamant"
980 PRINT#1:PRINT#1," Jedes angegebene Minera
l ritzt das"
990 PRINT#1," Vorangegangene bzw. wird von de
m"
1000 PRINT#1," Nachfolgenden geritzt."
1010 CLS#4:INPUT#4,"Ihre Wahl: ",wahl$:haerte!=VA
L(wahl$)
1020 IF haerte!<1 AND haerte!<>0 OR haerte!>10 TH
EN 1010
1030 PRINT#2,"Haerte : ";
1040 IF haerte!<1 THEN PRINT#2,"Unbekannt" ELSE P
RINT#2,haerte!
1050 CLS#4
1060 '
1070 ' ** Hier evtl. weitere Bestimmungskriterien
**
1080 ' *
1090 ' **
1100 '
1110 ' ** Suchen der in Frage kommenden Minerale
**
1120 '
1130 CLS#1:PRINT#1,"Vorschlaege:":PRINT#1
1140 RESTORE 30000
1150 READ name$
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";s
uchname$
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 E
LSE 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$="???" THEN PRINT#1,"unbekannt":GOTO 16
90
1670 IF h1$=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 '

```

10000 ' \*\*\* Kristallsysteme \*\*\*  
 10010 DATA Unbekannt,Rhombisch,Monoklin,Trigonal,K  
 ubisch  
 10020 DATA Tetragonal,Hexagonal,Triklin,Oktaedrisch,  
 h,Amorph,ende  
 10030 '  
 20000 ' \*\*\* FARBE \*\*\*  
 20010 DATA Unbekannt,Weiss,Gruen,Gelb,Rot,Blau,Vio  
 lett,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 ARGENIT,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



## Wußten Sie schon, daß...

... bei PRINT CHR\$(7) ein Ton erklingt?

... mit der Schleife: For I=0 TO 31:PRINT CHR\$(1)CHR\$(I):NEXT I auch die Systemsteuerzeichen sichtbar werden?

... mit PRINT CHR\$(19) der Bildschirm oberhalb des CURSORS gelöscht wird? Und mit PRINT CHR\$(20) löschen wir bis in die untere rechte Ecke? Wichtig ist hierbei die aktuelle Cursorposition.

... die Cursorsteuerung mit Print CHR\$(8) oder PRINT „<CTRL-H>“ links, PRINT CHR\$(9) oder PRINT „<CTRL-I>“ rechts, PRINT CHR\$(10) oder PRINT „<CTRL-J>“ unten, PRINT CHR\$(11) oder PRINT „<CTRL-K>“ oben ist?

... CLR/HOME bzw. CLS = PRINT CHR\$(12) ist?



Schneider Computer-Station  
 Ergonomisch konzipierter Arbeitstisch  
 für den »CPC« mit eingebauter  
 5fach-Steckdose

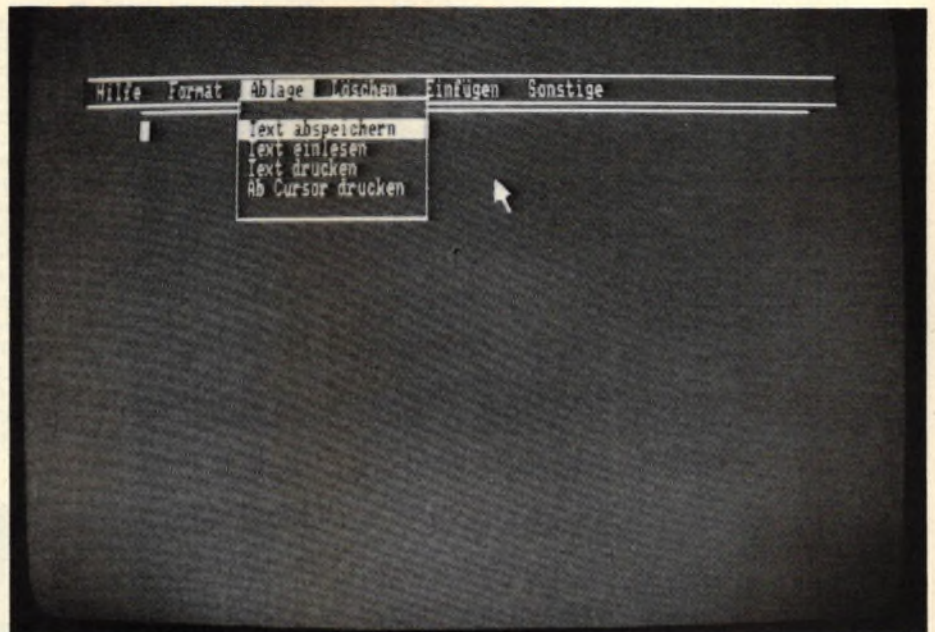
... mit PRINT CHR\$(24) oder PRINT „<CTRL-X>“ die REVERS-Darstellung erreicht wird?

... mit PRINT CHR\$(30) oder PRINT „>CTRL->“ ein Cursor-Home ausgeführt wird?

Hier noch einmal die Gesamtübersicht zum Ausschneiden und Sammeln:

CHR\$(1) macht Steuerzeichen in For Next Schleife sichtbar (0-32); CHR\$(7) Bell; es ertönt ein Piepton; CHR\$(8) Cursor links, CHR\$(9) Cursor rechts, CHR\$(10) Cursor hinunter, CHR\$(11) Cursor hinauf, CHR\$(12) CLEAR/HOME; entspricht CLS CHR\$(19) Löschen von Anfang bis Cursor, CHR\$(20) Löschen von Cursor bis Ende, CHR\$(30) Home-Pos.; Cursor auf Bildschirm anfang.

# NOCH BESSERE DRUCKER- STEUERUNG



Unser Leser Heinz Wienhoven hat sich die Arbeit gemacht, das Textverarbeitungsprogramm mit einer erheblich verbesserten Druckersteuerungsroutine zu versehen. Damit ist es nun möglich, mitten im Text Steuerzeichen für den Drucker zu plazieren, wodurch dann Textausschnitte unterstrichen, hervorgehoben usw. werden können. Um diese Verbesserungen in Ihr Programm einbauen zu können, müssen Sie das ursprüngliche Programm laden und

## Erläuterungen und Steuercodes

Das Programm wurde auf den Drucker NLQ 401 umgestellt. Desweiteren ist es nun möglich, einzelne Worte oder Textpassagen im Ausdruck hervorzuheben.

Das Formatieren des Textes über CTRL+F sollte durchgeführt werden, bevor die unten erläuterten

Der Ausdruck geht natürlich auch ohne Schriftartwechsel, egal ob REMARKs gesetzt wurden oder nicht. Die Anzahl der REMARKs entspricht der Anzahl der max. gewählten Zeilen.

Desweiteren besteht die Möglichkeit, den Zeilenabstand in 1/72 Zoll einzustellen. Die Anzahl der Zeilen je DIN A4-Seite wird in Abhängigkeit des Zeilenabstandes errechnet.

Die Zeilenlänge ergibt sich aus der gewählten Schriftart.

SHIFT+Cursor rechts	==> Fügt vor dem Cursor ein Zeichen ein<a>
SHIFT+Cursor links	==> Löscht das Zeichen auf Cursorposition<a>
SHIFT+Cursor unten	==> Fügt über dem Cursor eine Zeile ein<a>
SHIFT+Cursor oben	==> Löscht die Zeile auf Cursorposition<a>
COPY	==> Setzt Cursor auf Textanfang<a>
SHIFT+COPY	==> Setzt Cursor auf Textende<a>
CTRL+Cursor rechts	==> Löscht Rest der Zeile rechts vom Cursor<a>
CTRL+Cursor links	==> Löscht Zeile bis zur Cursorposition<a>
CTRL+COPY	==> Lädt die Zeile in den Puffer<a>
CTRL+S	==> Holt die Zeile aus dem Puffer<a>
TAB	==> Bewegt Cursor je 10 Stellen vorwärts<a>
SHIFT+TAB	==> Bewegt Cursor je 10 Stellen rückwärts<a>

die neuen Zeilen dazutippen. Da diese Erweiterungen Speicherplatz benötigen, war es notwendig, die maximal schreibbare Zeilenzahl auf 180 zu begrenzen. Mit dem Programm Optimizer ist es jedoch möglich, sämtliche Kommentare zu entfernen und alle Variablennamen zu kürzen, so dass dann wieder mehr Speicherplatz für Text zur Verfügung steht.

ten REMARKs gesetzt werden.

Im Druckmenü kommt man über -H- zur Hervorhebung der Texte. Vorher muß jedoch im Text überall dort, wo die Schriftart umgeschaltet werden soll, ein REMARK, ^ (SHIFT+7), gesetzt sein. Im Ausdruck wird das REMARK durch ein BLANC ersetzt. Die REMARKs und die Steuerzeichen werden nicht mit abgespeichert.

Weitere Steuercodes im Programm erreichbar über CTRL+H.

Um das Programm in der neuen Version 2.1 auf den Drucker Epson FX80 oder FX100 anzupassen, müssen nachfolgende Änderungen vorgenommen werden:

Die anderen Schriftarten werden über die gleichen Steuerzeichen gewählt wie beim NLQ 401.

```

3280 PRINT CHR$(i1);" N ";CHR$(i1);" Kursivschrift"<a>
3300 PRINT CHR$(i3);" E ";CHR$(i3);" Elite"<a>
3430 PRINT #8,CHR$(27);"4";:i1=24:GOTO 3260' Kursivsch.<a>
3450 PRINT #8,CHR$(27);"M";:i3=24:GOTO 3260' Elite<a>
3511 IF i1=24 THEN PRINT #8,CHR$(27);"5";:i1=0' Loeschen Kurs.<a>
3513 IF i3=24 THEN PRINT #8,CHR$(27);"P";:i3=0' Loeschen<a>
Elite<a>

```

```

1 REM ***** Version 2.1 *****
***
2 REM ***** (c) Thomas Barndt & Frank Thielen
***
3 REM *****
***
4 REM *****
***
5 REM ***** Ausdruck geaendert *****
***

380 IF maxze>180 THEN PRINT:PRINT "Dokument kann
max. 180 Zeilen lang sein":GOTO 350
395 DIM dnr(maxze),tcode(10)
400 FOR i=0 TO maxze:text(i)=STRING$(80," "):dnr
(i)=10:NEXT
410 FOR i=0 TO 9:stack(i)=STRING$(80," "):NEXT
411 tcode(0)=CHR$(27)+"-"+CHR$(1):tcode(1)=CHR$(
27)+"-"+CHR$(0)
412 tcode(2)=CHR$(27)+"G":tcode(3)=CHR$(27)+"H"
413 tcode(4)=CHR$(27)+"S"+CHR$(0):tcode(5)=CHR$(
27)+"T"
414 tcode(6)=CHR$(27)+"S"+CHR$(1):tcode(7)=CHR$(
27)+"T"
415 tcode(8)=CHR$(27)+"W"+CHR$(1):tcode(9)=CHR$(
27)+"W"+CHR$(0)
416 tcode(10)=""
3220 'Diese Zeile loeschen
3230 'Diese Zeile loeschen
3240 'Diese Zeile loeschen
3245 abst=12:druckbreite=80
3250 'Diese Zeile loeschen
3280 PRINT CHR$(i1);" N ";CHR$(i1);" Schoenschri
ft (NLQ)"
3290 PRINT CHR$(i2);" S ";CHR$(i2);" Schmalschri
ft"
3300 PRINT CHR$(i3);" E ";CHR$(i3);" Entwurfschr
ift"
3310 PRINT CHR$(i4);" F ";CHR$(i4);" Fettdruck"
3320 PRINT CHR$(i5);" B ";CHR$(i5);" Breitschri
ft (max. 40 Zeichen/Zeile)"
3330 PRINT CHR$(i6);" K ";CHR$(i6);" Kleinschri
ft"
3340 PRINT " ^ : Seitenvorschub"
3350 PRINT " - : Zeilenvorschub"
3355 PRINT " Z : Zeilenvorschub einstellen. Einst
ellung: ";abst;"/72 Zoll"
3360 PRINT " I : Initialisieren"
3365 PRINT " H : Hervorheben einzelner Worte"
3370 PRINT " R : Menue"
3380 PRINT " P : ";CHR$(24);" DRUCKEN ";CHR$(24)
3390 ts=UPPER$(INKEY%):IF INSTR(" NSEFBK^-ZIHPR",
ts)<2 THEN 3390
3400 CLS
3410 PRINT "Der Drucker ist nicht in Ordnung !"
3420 ON INSTR("NSEFBK^-ZIHPR",ts) GOTO 3430,3440,
3450,3460,3470,3480,3490,3500,3422,3510,3525
,3520,3530
3422 CLS:PRINT:PRINT:PRINT "Der Zeilenabstand wir
d in x/72 Zoll eingestellt. Die Normaleinste
llung ist 12/72 Zoll. Zeilenabstand
(1-85): ";
3423 INPUT " ";abst:IF abst<1 OR abst>85 THEN abst
=12
3424 PRINT #8,CHR$(27);"A";CHR$(abst);CHR$(27);"2
";
3425 GOTO 3260
3430 PRINT #8,CHR$(27);"I";CHR$(3);:i1=24:GOTO 32
60'---NLQ
3440 PRINT #8,CHR$(15);:i2=24:druckbreite=110:GOT
0 3260'---Schmalsch.
3450 PRINT #8,CHR$(27);"x";CHR$(0);:i3=24:GOTO 32
60'---Entwurfsch.
3460 PRINT #8,CHR$(27);"E";:i4=24:GOTO 3260'---Fe
ttdruck-
3470 PRINT #8,CHR$(27);"W";CHR$(1);:i5=24:druckbr
eite=40:GOTO 3260'---Breitsch.
3480 PRINT #8,CHR$(27);"S";CHR$(0);CHR$(15);:i6=2
4:abst=6:GOTO 3424'---Kleinsch.
3490 PRINT #8,CHR$(12);:GOTO 3260
3500 PRINT #8,CHR$(10);:GOTO 3260'---Zeilenvorsch
:
3510 '*** Drucker Initialisier
en ****
3511 IF i1=24 THEN PRINT #8,CHR$(27);"I";CHR$(1);
:i1=0' --- Loeschen NLQ
3512 IF i2=24 THEN PRINT #8,CHR$(18);:i2=0'--- Lo
eschen Schmal
3513 IF i3=24 THEN i3=0' --- Loeschen Entwurf
3514 IF i4=24 THEN PRINT #8,CHR$(27);"F";:i4=0' -
--- Loeschen Fett
3515 IF i5=24 THEN PRINT #8,CHR$(27);"W";CHR$(0);
:i5=0'--- Loeschen Breit
3516 IF i6=24 THEN PRINT #8,CHR$(27);"T";:i6=0'--
--- Loeschen Hoch
3517 PRINT #8,CHR$(27);"x";CHR$(0);'--- Draftmode
3518 druckbreite=80:abst=12:GOTO 3260
3520 RETURN
3525 GOSUB 5000:GOTO 3260
3530 'Diese Zeile loeschen
3535 WIDTH druckbreite
3540 CLS
3550 druckhoehe=INT(720/abst)'DIN A4 hat 720/72 Z
oll
3560 'Diese Zeile loeschen
3570 'Diese Zeile loeschen
3580 INPUT "Anzahl der Ausdrücke: ",anzk
3590 CLS:LOCATE 5,10:PRINT "Ausdruck abbrechen ==
> A"
3600 FOR j=1 TO anzk
3605 asz=0
3610 FOR i=1 TO laenge
3615 ver=UPPER$(INKEY%)
3616 IF ver="A" THEN i=laenge:j=anzk:PRINT "D
ruck abgebrochen":GOTO 3670
p=1
3621 WHILE p<LEN(text(i))
3622 r=INSTR(p,text(i),"")
3623 ON SGN(r-1)+2 GOTO 3625,3626,3627
3624 r=LEN(text(i)):tdru=tdru+MID$(text(i),
p,r-p):p=r:GOTO 3628
3625 asz=asz+1:tdru=" "+tcode(dnr(asz)):p=2
:GOTO 3628
3626 asz=asz+1:tdru=tdru+MID$(text(i),p,r-p
)+" "+tcode(dnr(asz)):p=r+1
3627 WEND
3628 PRINT #8,LEFT$(tdru+STRING$(80,32),druck
breite);:tdru=""
3630 IF i MOD druckhoehe>0 OR i=0 OR i=laenge
THEN 3670
3640 IF s="A" THEN PRINT #8,CHR$(12);:GOTO 36
70
3650 PRINT "Naechstes Blatt vorbereiten. Weit
er mit -ENTER-":CALL &B06

```

```

3660 CLS
3670 NEXT i
3680 IF j=anzk THEN PRINT #8;GOTO 3730
3690 IF s="A" THEN PRINT #8,CHR$(12);:GOTO 3730
3700 PRINT #8
3710 IF j<anzk THEN PRINT "Naechste Blatt vorbe
reiten. Weiter mit -ENTER-":CALL &BB06
3720 CLS
3730 NEXT j
3740 GOTO 3260
3750 '
5000 ' *** Druckarten einstelle
n ****
5005 CLS
5010 IF i1+i2+i3+i4+i5+i6=0 THEN PRINT "Zuerst Sc
hriftart wAhlen!":FOR i=1 TO 2000:NEXT i:GOT
O 5250
5020 CLS:CLS #2
5030 PRINT #2
5040 PRINT #2," mOgliche Hervorhebungen:"
5045 PRINT #2
5050 PRINT #2," Ein Aus"
5060 IF i6<>24 THEN PRINT #2," Unterstreichen ->
0 1"
5070 IF i1<>24 AND i6<>24 THEN PRINT #2," Doppeld
ruck -> 2 3"
5080 IF i1<>24 AND i6<>24 THEN PRINT #2," Hochsch
rift -> 4 5"

```

```

5090 IF i1<>24 AND i6<>24 THEN PRINT #2," Indexsc
hrift -> 6 7"
5100 IF i1<>24 AND i5<>24 THEN PRINT #2," Breitsc
hrift -> 8 9"
5110 asz=0:CLS #1
5120 FOR i=1 TO laenge
5125 LOCATE #1,10,1:PRINT #1,"Ich untersuche Ze
ile ";i;
FOR j=1 TO 80
5130 IF MID$(text(i),j,1)<>" " THEN 5230
5140 asz=asz+1:IF asz=maxze-2 THEN PRINT #1,"
Es koennen noch zwei Codes definiert wer
den ";
5160 LOCATE j,5:PRINT CHR$(241)
5170 LOCATE 1,6:PRINT text(i);
5175 PRINT:PRINT "Eingestellt : ";STR$(dnr (as
z))
5180 PRINT:PRINT "Druckartnr. : ";:INPUT " ",d
nr (asz):IF dnr (asz)<0 OR dnr (asz)>9 THEN
dnr (asz)=10
5200 LOCATE j,5:PRINT " ":PRINT:PRINT
5210 PRINT "Eingestellt : ":PRINT:PRINT
"Druckartnr. : "
5220 PRINT "Code Nr. ";asz;" = ";dnr (asz)
5230 NEXT j
5240 NEXT i
5250 RETURN

```

# Riesige Zeichen auf dem Drucker

Mit Banner kann man ganz leicht Spruchbänder oder eine neue Tapete für's Wohnzimmer herstellen. Man braucht nur einzugeben, was man gedruckt haben möchte, für genug Papier im Drucker zu sorgen, die Buchstabenhöhe und — breite zu bestimmen, und schon quillt der Text aus dem Drucker, wenn man will, 20 Meter lang.

```

1 REM >>BANNER<<
2 REM von Thomas M. Binzinger
3 REM
4 MODE 2:DEFINT b-o,r-z
5 SYMBOL AFTER 32:ad=HIMEM+1
6 PRINT:INPUT "Was soll gedruckt werden? ",d$
7 INPUT "Welche Hoehe (1-10):",ho
8 INPUT "Welche Breite:",br
9 FOR x=LEN(d$) TO 1 STEP -1
10 e$=d$+" "
11 PRINT CHR$(13);LEFT$(e$,x-1);CHR$(24);MID$(e$,x
,1);CHR$(24);RIGHT$(e$,LEN(e$)-x);
12 GOSUB 14:NEXT "Buchstabe drucken
13 GOTO 6
14 a=ASC(MID$(d$,x,1)):adr=ad+((a-32)*8)'adr=Adres
se des Buchstaben-Bitmusters
15 FOR q=adr TO adr+7:s$(q-adr)=BIN$(PEEK(q),8):
NEXT
16 FOR o=8 TO 1 STEP -1:t$(o)="" :FOR p=1 TO 8
17 t$(o)=t$(o)+MID$(s$(p),o,1):NEXT p,o
18 FOR o=8 TO 1 STEP -1:FOR r=1 TO br:FOR p=1 TO 8
:FOR i=1 TO ho
19 IF MID$(t$(o),p,1)<>"0" THEN PRINT #8,CHR$(a);
ELSE PRINT #8," ";
20 NEXT i,p:PRINT #8,"":NEXT r,o
21 RETURN

```

## Banner

**Drucker-Tip**

# Hardcopies

Eigentlich kann man für alle, die einen Drucker an ihren CPC angeschlossen haben, nur ein mitleidiges Grinsen übrig haben: Was Schneider sich mit der 7-Bit-Selbstbau-Schnittstelle gedacht haben mag, wird für die meisten von uns wohl immer ein Rätsel bleiben. Solange es nur um Text geht, funktioniert das ganze ja auch noch halbwegs, aber bei Hardcopies: Ohje!

Was allerdings bei Hardcopies produziert wird, dreht einem dann doch ganz den Magen um: Da wird jede Zeile 2 mal gedruckt, weil jeweils nur 4 Nadeln anschlagen (logische Begründung: die 8te funktioniert ja nicht?!?), und daß (auch wegen des fehlenden 7. Bits) die letzte Punktreihe vom Bildschirm fehlt, ist auch allgemein üblich (aber darüber lasse ich mich lieber nicht aus, weil ich so etwas auch schon selbst gemacht habe).

## Es geht auch anders

Daß es auch anders geht, möchte ich zeigen. Das Kernstück dieses Versuches stellt Programm Listing Nr. 1 dar (die assemblierte Fassung davon wird von den anderen Programmen nachgeladen). Dabei handelt es sich um das Unterprogramm, das die notwendigen Druckdaten aus dem Bildschirminhalt erzeugt.

'Nichts Besonderes', werden Sie nun vielleicht sagen, aber es gibt doch etwas Besonderes: Das Programm ist nämlich extrem flexibel und läßt sich leicht in Ihre eigenen Programme einbinden. Damit können Sie leicht beliebige Grafiken, nicht nur Hardcopies, erzeugen, weil Sie jeden beliebigen Teil des Bildschirms 'in Daten umwandeln' können. Das eigentliche Unterprogramm steht dabei in Zeile 840 (Adresse +A07B). Diese Subroutine erzeugt in einem Buffer aus dem Inhalt des Bildschirms, der an der angegebenen x- und y-Koordinate steht, die Bytes im richtigen Format zum An-den-Drucker-senden.

```

HiSoft GENAS.1 Assembler, Page 1.

Pass 1 errors: 00

      10 ; Listing Nr. 1
      20 ;
A000      30      org #a000
      40 ;
A000 00      50 add:  nop          ;xadd
      60 ;
A001 0000    70 buffer: defw 0      ;Adresse des 640-Byte Puffers
      80 ;
A003 00      90 colour: nop       ;Anzahl der Farben
     100 ;
A004 00     110 high:  nop         ;Anzahl Punkte horizontal
     120 ;
A005 00     130 points: nop       ;wie oft wird jeder Punkt gedruckt
     140 ;
A006 0000    150 buff2: defw 0      ;intern
     160 ;
A008 0000    170 pnum:  defw 0      ;Anzahl der zu druckenden Punkte
     180 ;
A00A 0000    190 zy2:   defw 0      ;intern
     200 ;
A00C 0000    210 zx:    defw 0      ;augenblickliche x
A00E 0000    220 zy:    defw 0      ;augenblickliche y
     230 ;
A010 00     240 cint:  nop         ;Colour intern
     250 ;
A011 00     260 shifts: nop        ;anzahl shifts bisher
     270 ;
A012 F5     280 acint:  push af     ;count cint
A013 E5     290          push hl
A014 3A10A0 300      ld  a,(cint)
A017 3C     310      inc  a

```

Das Ergebnis hängt stark von den Variablen am Anfang des Programmes ab, die ich nun im einzelnen besprechen möchte. Ganz am Anfang steht die Variable **xadd**.

Ihr Wert bestimmt, um welchen Faktor X beim Untersuchen des Bildschirms erhöht wird. Das ist sinnvoll, weil ja mehrere horizontale Pixel in Mode 1 und 0 zusammengefasst sind.

Trotzdem kann XADD immer 1 bleiben, so daß die Hardcopy auch immer aus derselben Anzahl Punkten (und somit derselben Breite) besteht. Dann wird einfach ein- und derselbe Pixel mehrmals abgefragt, und der 'Papier-Pixel' wird dementsprechend breiter. Das Wort an Adresse +A001 bestimmt, an welche Adresse die Daten geschrieben werden sollen. An dieser Stelle müssen (je nach gewähltem XADD) maximal 640 Bytes Platz haben. Als nächstes folgt die Variable **colour**. Sie bestimmt, wieviele Farben mo-

mentan auf dem Bildschirm dargestellt werden können, und muß je nach MODE gesetzt werden.

Sehr wichtig ist die nächste Variable, **high**. Sie bestimmt, wie hoch ein einzelner Pixel auf dem Papier sein soll, und der maximale zulässige Wert ist 7. Damit ist es möglich (siehe Programm Nr.4), das Bild so auszudehnen, daß die

## Beliebige Höhe

Verzerrungen im Vergleich zum Original-Bildschirm möglichst gering werden. BUFF2 ist für uns nicht weiter interessant, diese Variable wird intern benutzt. Interessanter ist **pnum**, mit dem wir bestimmen können, wieviele Punkte horizontal überhaupt abgefragt werden sollen. Damit ist es möglich, nur einen Teil des Bildes zu bearbeiten. Schließlich folgen noch ZX und ZY. Hierher müssen die Startkoordinaten für die Bild-

## Hardcopies leicht gemacht

schirmabtastung geschrieben werden.

## Komplette Hardcopy

Um also eine komplette Hardcopy auf den Bildschirm zu bekommen, ist eine Schleife notwendig, die die Y-Koordinaten von 399 auf 0 herunterzählt, und zwar in Schritten von 14(!). 14 deshalb, weil ja vertikal immer zwei Pixel zusammengesetzt sind. Bei jedem Schleifendurchlauf müssen die notwendigen Parameter initialisiert werden, und DATA muß aufgerufen werden. Die dann im Buffer stehenden Daten können an den Drucker gesendet werden. Dieser mußte natürlich vorher auf einen Papiervorschub von 7/72 (also der Höhe von 7 Punkten) gesetzt werden, damit zwischen den einzelnen Zeilen kein Leerraum entsteht.

## Kein Leerraum

Ein Programm, das genau das tut, ist Nr. 2. Aber das wäre wohl kaum der Rede wert, wenn das alles wäre. Denn dieses Programm hat natürlich noch die altbekannte Macke, die Punktreihe ganz rechts außen zu verschlucken, da wir nur die Zahl 639 und nicht 640 über den 7-Bit-Port schicken können. Eine einfache Lösung dafür ist Programm Nr. 3: Jede Zeile wird in zwei Teilzeilen aufgespalten. Natürlich darf auch die Punktzahl der beiden Teilzeilen keine 8-Bit-Zahl enthalten, sonst hätten wir dasselbe Problem wie in Listing Nr.2 (achten Sie übrigens darauf, daß Sie eine assemblierte Version von Nr.1 auf Kassette/Diskette stehen haben, weil die Hardcopy-Programme sie nachladen).

## Programm Nr.4

Das letzte Programm, Nr. 4, zeigt eine noch bessere Ausnutzung der Möglichkeiten des MC-Teils. Das Bild wird auf die volle Blattbreite

```

A01B 3210A0 320 ld (cint),a
A01B 2103A0 330 ld hl,colour
A01E BE 340 cp (hl) ;anzahl farben
A01F 3807 350 jr c,ac1l ;
A021 2805 360 jr z,ac1l ;
A023 3E01 370 ld a,l ;Hintergrund wird nicht gedruckt
A025 3210A0 380 ld (cint),a
A02B E1 390 ac1l: pop hl
A029 F1 400 pop af
A02A C9 410 ret
420 ;
A02B E5 430 opoi: push hl ;setzt einen Punkt in C
A02C D5 440 push de
A02D CD12A0 450 call acint ;zaehle farben weiter
A030 2A0EA0 460 ld hl,(zy) ;y-Koor
A033 ED5B0CA0 470 ld de,(zx) ;x-Koor
A037 C5 480 push bc
A03B CDF0BB 490 call @bbf0 ;Test point
A03B C1 500 pop bc
A03C B7 510 or a
A03D 2809 520 jr z,opoi1 ;farbe 0 wird nie gedruckt
A03F 3D 530 dec a ;colour=colour-1
A040 2110A0 540 ld hl,cint
A043 A6 550 and (hl) ;=interne Colour
A044 2002 560 jr nz,opoi1 ;drucke nur wenn 0
A046 CBC1 570 set 0,c ;setzte bit 0 in c
A04B 2111A0 580 opoi1: ld hl,shifts
A04B 34 590 inc (hl)
A04C D1 600 pop de
A04D E1 610 pop hl
A04E C9 620 ret
630 ;
A04F 0E00 640 genbyt: ld c,0 ;wert=0
A051 3E00 650 ld a,0
A053 3211A0 660 ld (shifts),a
A056 3A04A0 670 gen0: ld a,(high) ;hoehe eines punktes (max 7)
A059 47 680 ld b,a
A05A EB01 690 gen1: rlc c ;schiebe c weiter
A05C CD2BA0 700 call opoi ;schreibe ein bit in c
A05F 3A11A0 710 ld a,(shifts)
A062 FE07 720 cp 7
A064 3002 730 jr nc,gen2
A066 10F2 740 djnz gen1 ;highpunkte
A06B 2A0EA0 750 gen2: ld hl,(zy)
A06B 110100 760 ld de,l
A06E ED52 770 sbc hl,de
A070 220EA0 780 ld (zy),hl
A073 3A11A0 790 ld a,(shifts)
A076 FE07 800 cp 7
A07B 3BDC 810 jr c,gen0 ;weniger als 7 bits
A07A C9 820 ret
830 ;
A07B 2A0BA0 840 data: ld hl,(pnua) ;anz zu konv punkte
A07E E5 850 push hl
A07F C1 860 pop bc ;anzahl in bc
A080 DD2A01A0 870 ld ix,(buffer) ;bufferpos
A084 C5 880 datl: push bc ;save count
A085 2A0EA0 890 ld hl,(zy) ;zy wird von genbyt veraendert
A08B E5 900 push hl
A089 CD4FA0 910 call genbyt ;erzeuge eine byte
A08C 220AA0 920 ld (zy2),hl ;das ist das naechste zy
A08F 79 930 ld a,c ;byte -> a
A090 DD7700 940 ld (ix),a ;byte -> buffer
A093 DD23 950 inc ix ;naechste bufpos
A095 E1 960 pop hl ;hole altes zy
A096 220EA0 970 ld (zy),hl
A099 2A00A0 980 ld hl,(add) ;x-add
A09C ED5B0CA0 990 ld de,(zx) ;x-pos
A0A0 19 1000 add hl,de
A0A1 220CA0 1010 ld (zx),hl ;neue x-pos
A0A4 C1 1020 pop bc ;hole count
A0A5 0B 1030 dec bc
A0A6 78 1040 ld a,b
A0A7 B1 1050 or c ;erzeuge naechstes byte
A0A8 20DA 1060 jr nz,datl
A0AA 2A0AA0 1070 ld hl,(zy2) ;hole naechste zy
A0AD 220EA0 1080 ld (zy),hl ;schreibe sie
A0B0 C9 1090 ret ;alles getan
1100 ;

```

Pass 2 errors: 00

Table used: #62 from 353



(jedenfalls fast) ausgedehnt, und jeder Pixel wird zwei Pixel tief auf dem Papier dargestellt. So ist die Verzerrung im Vergleich zum Originalbild minimal.

```

10 MEMORY &7FFF 'Listing Nr. 2
20 MODE 2:m=2
30 PLOT 0,0,1:DRAW 639,0:DRAW 639,399:DRAW 0,399:D
RAW 0,0:LOCATE 35,10:PRINT" T E B T !"
40 LOAD"insidi.bin" 'insidi.bin=Listing Nr.1
50 add=&A000
60 buffer=&A001
70 colour=&A003
80 high=&A004
90 points=&A005
100 pnum=&A008
110 zx=&A00C
120 zy=&A00E
130 WIDTH 255 'CRs unterdruecken
140 PRINT #8,CHR$(27)"A"CHR$(7)CHR$(27)"2"; 'Papier
rvorschub
150 '
160 FOR y=399 TO 0 STEP -14 '-(14/ph)
170 POKE zy,y-(INT(y/256)*256):POKE zy+1,INT(y/256)
)
180 POKE zx,0:POKE zx+1,0 '0
190 'poke &a012,&c9:poke &a010,255 unterdrueckt Fa
rben
200 ph=1 'Punkthoehe=1
210 pzah1=320 ' jeweils 320 Punkte drucken
220 IF m=2 THEN POKE colour,1
230 IF m=1 THEN POKE colour,3
240 IF m=0 THEN POKE colour,15
250 POKE high,ph
260 POKE points,1
270 POKE buffer,0:POKE buffer+1,&80
280 POKE pnum,128:POKE pnum+1,2 '640 Punkte
290 POKE add,1
300 '
310 CALL &A07B:ad=&8000
320 PRINT #8,CHR$(27);"L";CHR$(127);CHR$(2);
330 '
340 FOR x=0 TO 639
350 PRINT #8,CHR$(PEEK(ad));:ad=ad+1
360 NEXT x
370 PRINT #8,""
380 '
390 NEXT y

```

Übrigens werden von allen Programmen auch Farben dargestellt, und zwar durch unterschiedliche Schraffierungen. Im MC-Teil besorgt das Unterprogramm ACINT das Durchzählen der Farben, um so die Schraffierungen zu erzeugen. Wenn man keine Farbdarstellung möchte, sollte man den Aufruf von ACINT durch ein +C9 (Op-

## HARDCOPY — kein Problem

Code RET) an Adresse +A012 unterdrücken und den Wert 255 in CINT schreiben. Es gibt noch viele Möglichkeiten, wie man Listing Nr.1 nutzen kann, nicht nur als Hardcopy-Unterprogramm. Zum Beispiel könnte man Listings in der Original-Computerschrift (und somit auch alle Sonderzeichen) ausdrucken... aber in dieser Beziehung haben Sie sicher auch genug Ideen.

```

10 MEMORY &7FFF 'Listing Nr.3
20 MODE 2:m=2
30 PLOT 0,0,1:DRAW 639,0:DRAW 639,399:DRAW 0,399:D
RAW 0,0:LOCATE 35,10:PRINT" T E B T !"
40 LOAD"insidi.bin" 'insidi.bin=Listing Nr.1
50 add=&A000
60 buffer=&A001
70 colour=&A003
80 high=&A004
90 points=&A005
100 pnum=&A008
110 zx=&A00C
120 zy=&A00E
130 WIDTH 255 'CRs unterdruecken
140 PRINT #8,CHR$(27)"A"CHR$(7)CHR$(27)"2"; 'Papier
rvorschub
150 '
160 FOR y=399 TO 0 STEP -14 '-(14/ph)
170 POKE zy,y-(INT(y/256)*256):POKE zy+1,INT(y/256)
)
180 POKE zx,0:POKE zx+1,0 '0
190 'poke &a012,&c9:poke &a010,255 unterdrueckt Fa
rben
200 ph=1 'Punkthoehe=1
210 pzah1=320 ' jeweils 320 Punkte drucken
220 IF m=2 THEN POKE colour,1
230 IF m=1 THEN POKE colour,3
240 IF m=0 THEN POKE colour,15
250 POKE high,ph
260 POKE points,1
270 POKE buffer,0:POKE buffer+1,&80
280 POKE pnum,128:POKE pnum+1,2 '640 Punkte
290 POKE add,1
300 '
310 CALL &A07B:ad=&8000
320 '
330 FOR z=1 TO 2:PRINT #8,CHR$(27);"L";CHR$(64);CH
R$(1);
340 FOR x=0 TO 319
350 PRINT #8,CHR$(PEEK(ad));:ad=ad+1
360 NEXT x,z
370 PRINT #8,""
380 '
390 NEXT y

```

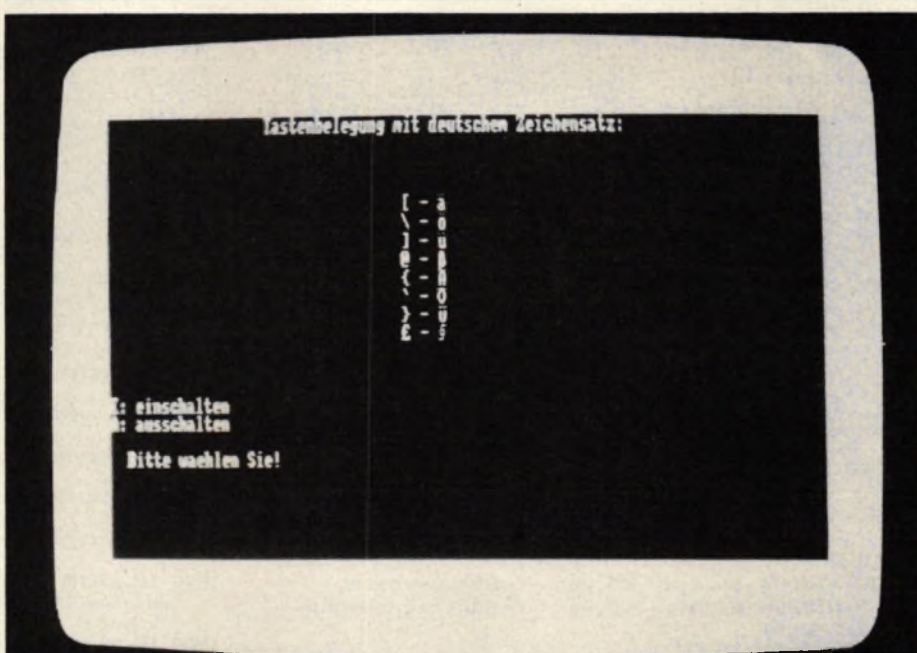
```
10 MEMORY &7FFF 'Listing Nr.4
20 MODE 1:m=1
30 PLOT 0,0,1:DRAW 639,0:DRAW 639,399:DRAW 0,399:D
RAW 0,0:LOCATE 15,1:PEN 2:PRINT" T E S T !"
40 LOAD"insid1.bin" 'insid1.bin=Listing Nr.1
50 add=&A000
60 buffer=&A001
70 colour=&A003
80 high=&A004
90 points=&A005
100 pnum=&A008
110 zx=&A00C
120 zy=&A00E
130 WIDTH 255 'CRs unterdruecken
140 PRINT #8,CHR$(27)"A"CHR$(7)CHR$(27)"2"; 'Papier
rvorschub
150 '
160 FOR y=399 TO 0 STEP -(14/2) '2 Punkte hoch!!!
170 POKE zy,y-(INT(y/256)*256):POKE zy+1,INT(y/256
)
180 POKE zx,0:POKE zx+1,0 '0
190 'poke &a012,&c9:poke &a010,255 unterdrueckt Fa
rben
200 ph=2 'P u n k t h o e h e = 2
210 pzahl=320 ' jeweils 320 Punkte drucken
220 IF m=2 THEN POKE colour,1
230 IF m=1 THEN POKE colour,3
240 IF m=0 THEN POKE colour,15
250 POKE high,ph
260 POKE points,1
270 POKE buffer,0:POKE buffer+1,&80
280 POKE pnum,128:POKE pnum+1,2 '640 Punkte
290 POKE add,1
300 '
310 CALL &A07B:ad=&8000
320 '
330 PRINT #8,CHR$(27)"Z"CHR$(0)CHR$(10); 'quadrupl
e density
340 GOSUB 380 'drucke, und
350 FOR x=0 TO 639:PRINT #8,CHR$(0);:NEXT 'fuelle
mit 0's
360 PRINT #8,"":NEXT y:END
370 '
380 FOR x=0 TO 639 'sende 1920 Punkte (640*3)
390 PRINT #8,CHR$(PEEK(ad));
400 PRINT #8,CHR$(PEEK(ad));
410 PRINT #8,CHR$(PEEK(ad));:ad=ad+1
420 NEXT x:RETURN
```

Dieses Textverarbeitungsprogramm ist bis auf die Bildschirmscrollroutinen nur in BASIC geschrieben. Nach dem Start wird als erstes nach der Textdarstellung gefragt. Es besteht die Möglichkeit, mit schwarzer Schrift auf grünem Grund zu schreiben oder umgekehrt. Danach fragt das Programm nach der maximal einzugebenden Zeilenzahl. Es ist möglich, bis zu 230 Zeilen Text zu schreiben. Schöpft man diese Möglichkeit voll aus, so geht dies jedoch etwas auf Kosten der Geschwindigkeit beim Einfügen oder Löschen von Zeilen. Als nächstes erscheint das Hauptmenü. Dort kann man zwischen folgenden sechs Möglichkeiten wählen: — Eingeben bzw. Editieren von Text — Löschen eines Textes — Ausdrucken eines Textes — Speichern oder Laden von Texten — Ein- oder Ausschalten des deutschen Zeichensatzes — Beenden des Programms. Beim Laden eines Textes wird die Cassette zunächst auf ein vorhandenes Textfile untersucht. Nach der Anzeige von Textname und Textlänge kann man entscheiden, ob geladen oder weitergesucht werden soll. Nun zum wichtigsten Punkt, dem Editieren. Nach dem Anwählen dieses Menüpunktes erscheint zunächst der Cursor in der linken oberen Bildschirmcke, sowie die Statuszeile ganz unten. Dort wird die momentane Zeilen- und Spaltenposition alle paar Sekunden angezeigt. Ebenso der momentan eingeschaltete Zeichensatz. Man kann nun seinen Text eingeben. Über CTRL «Anfangsbuchstabe» erreicht man die Ausführung der folgenden Funktionen: — Menü-Zeile anspringen — Sichern gelöschter Zeilen — Formatieren des Textes — Randbreite einstellen — Trennen der Zeile — Vereinigen von Zeilen — Worte suchen — Austauschen von Worten — Hilfe (Anzeige der Funktionen).

Zum Formatieren setzt man den Cursor auf die erste Zeile und letzte Spalte des zu formatierenden Textes und betätigt die Tasten CTRL + „F“ (Formatiere) gleichzeitig. Nach Eingabe der Zeilenanzahl werden in diese Zeilen so viele Leerzeichen eingesetzt, bis sie rechtsbündig mit der Spaltenposition abschließen.

Das Suchen und Austauschen von Worten geschieht ab Cursorposition abwärts. Beim Trennen von Zeilen wird die rechte Seite der Zeile, einschließlich der Cursorposition, zwischen die jetzige und nächste Zeile eingefügt, indem man CTRL

# Der Word- Processor II



+ „T“ (Trennen) betätigt. Das Vereinigen von Zeilen erfolgt dementsprechend mit CTRL + „V“. Wenn der Platz zu diesen Operationen nicht ausreicht, wird dieses in der Statuszeile angezeigt und die Operation nicht durchgeführt. Sollte während des Editierens der Cursor einmal für ein paar Sekunden verschwinden, so ist dies nicht weiter tragisch. Der Computer führt dann eine Sammlung von freiem Speicherplatz durch. Dieses tritt jedoch bei kürzeren Texten sehr selten auf. Jede gelöschte Zeile wird auf einen Stack (Stapel) gelegt und kann mit CTRL + „S“ (Sichern) wieder angezeigt werden. Man kann auf diese Weise bis zu neun gelöschte Zeilen retten.

Betätigt man die Taste CTRL + COPY, so wird die aktuelle Zeile ebenfalls auf den Stack gelegt, jedoch dabei nicht gelöscht. Dadurch ist es möglich, Zeilen an beliebige Textstellen zu kopieren oder zu verschieben.

Das Programm ist an einen EPSON Drucker RX-80 angepaßt, dessen Steuerzeichen stimmen jedoch weitgehend mit den meisten anderen Druckern überein. Beim Ausdrucken kann man die wichtigsten Schriftarten des Druckers auswählen.

Um ein sicheres Funktionieren zu gewährleisten, sollten Sie vor jedem Laden des Programms den Rechner zurücksetzen.

Dieses Programm wurde mit äußerster Sorgfalt erstellt, jedoch können Fehler nie ganz ausgeschlossen werden. Für eventuell entstehende Schäden oder Verluste kann keine Haftung übernommen werden.

Nachfolgend noch einige Funktionen von Wordprocessor. Bitte beachten Sie diese ebenso, wie die während des Programmlaufes erscheinenden Anweisungen, damit Ihr Text nicht verloren geht!

FTCP + THBCS

```

1 REM Frank Thielen
2 REM      &
3 REM Thomas Barndt
4 REM
5 REM
10 MODE 2
20 DEFSTR s-w:DEFINT a-r,x-z
30 KEY DEF 68,1,9,10
40 KEY DEF 9,1,224,223,222
50 KEY DEF 18,1
60 scrollup=""
70 READ byte
80 WHILE byte>=0
90   scrollup=scrollup+CHR$(byte)
100  READ byte
110  WEND
120  szej="engl."
130  INK 0,0:INK 1,26
140  CLS
150  LOCATE 34,4
160  PRINT
170  LOCATE 28,7
180  PRINT "Copyright FTCP + THBCS 1985"
190  LOCATE 1,10
200  PRINT "Textdarstellung schwarz auf gruenum G
rund (S) oder gruen auf schwarzem Grund (G)"
210  *
220  *           *** Farben ***
230  *
240  s=UPPER$(INKEY$)
250  IF INSTR("SG",UPPER$(s))<2 THEN 240
260  IF s="S" THEN BORDER 23:PEN 0:PAPER 1:PEN #1
,1:PAPER #1,0:PEN #2,1:PAPER #2,0 ELSE BORDE
R 14:PEN 1:PAPER 0:PEN #1,0:PAPER #1,1:PEN
#2,0:PAPER #2,1:MID$(scrollup,4,1)=CHR$(0)
270  DATA &06,&01,&3E,255,&21,0,0,&11,23,79,&CD,&
50,&BC,&C9,-1
280  scrolldown=scrollup
290  MID$(scrolldown,2,1)=CHR$(0)
300  scrollinsert=scrolldown:scrolldelete=scrollu
p
310  CLS
320  WINDOW #1,1,80,25,25      Statuszeile
330  WINDOW #2,44,77,8,21     Funktionsmenue
340  CLS #1
350  PRINT
360  INPUT "Maximale Zeilenzahl ";maxze
370  IF maxze<24 THEN PRINT:PRINT "Dokument muss
laenger als 23 Zeilen sein":GOTO 350
380  IF maxze>230 THEN PRINT:PRINT "Dokument darf
nicht laenger als 230 Zeilen sein":GOTO 350
390  DIM text(maxze),stack(9)  Textspeicher + St
ack fuer geloeschte Zeilen
400  FOR i=0 TO maxze:text(i)=STRING$(80," "):NEX
T
410  FOR i=0 TO 9:stack(i)=STRING$(80," "):NEXT
420  GOSUB 510      Menue
430  ON INSTR("ELACBD",s) GOSUB 680,3010,3110,376
0,4280,4630
440  GOTO 420
450  *
460  *           *** Status in Statuszeile (Window #1) a
usgeben ***
470  *
480  LOCATE #1,1,1:PRINT#1,USING " Z:### S:##
Hilfe : CTRL 'H'
Zeichensatz: &";ize,isp,szej;
490  RETURN
500  *           *****
510  *           **      Hauptmenue      **
520  *           *****
530  CLS
540  CLS#1
550  PRINT"E: Eingabe oder Edieren von Text"
560  PRINT"L: Loeschen des Textes"
570  PRINT"A: Ausdrucken des Textes"
580  PRINT"C: Speichern oder Laden von Texten"
590  PRINT"D: Deutscher Zeichensatz"
600  PRINT"B: Beenden des Programms"
610  PRINT
620  PRINT"      Bitte waehlen Sie!"
630  s=UPPER$(INKEY$):IF s="" THEN 630
640  IF INSTR("ELACBD",s)<1 THEN 630
650  CLS:CLS#1
660  RETURN
670  *
680  *           *** Eingabe bzw. Edieren ***
690  *
700  EVERY 250 GOSUB 460
710  ize=1:isp=1+rand      Zeile und Spalte des C
ursor im Textarray
720  iba=1                Bildschirmanfang im Te
xtarray (vertikal)
730  icu=1                Cursorposition auf dem
Bildschirm (vertikal)
740  GOSUB 2920
750  EI
760  LOCATE isp,icu
770  PRINT CHR$(24);MID$(text(ize),isp,1);CHR$(24
);CHR$(8);
780  IF INKEY(23)<>128 THEN s=INKEY$:IF s="" THEN
780 ELSE 810
790  s=INKEY$:IF s="" THEN 790 ELSE IF ASC(s)>31
THEN 810
800  u=CHR$(ASC(s)+96):GOTO 1360
810  IF s<CHR$(32) OR s>CHR$(126) THEN PRINT MID$
(text(ize),isp,1):GOTO 830
820  PRINT s;:MID$(text(ize),isp,1)=s:isp=isp+1:G
OTO 2800      Druckbares Zeichen
830  IF s=CHR$(13) THEN isp=1+rand:ize=ize+1:icu=
icu+1:GOTO 2800      ENTER
840  IF s=CHR$(242) THEN isp=isp-1:GOTO 2800
Cursor links
850  IF s=CHR$(243) THEN isp=isp+1:GOTO 2800
Cursor rechts
860  IF s=CHR$(240) THEN ize=ize-1:icu=icu-1:GOTO
2800      Cursor aufwaerts
870  IF s=CHR$(241) THEN ize=ize+1:icu=icu+1:GOTO
2800      Cursor abwaerts
880  IF s=CHR$(224) THEN GOTO 710
COPY (HOME)
890  IF s=CHR$(223) THEN iba=maxze-23:icu=24:ize=
maxze:GOSUB 2920:GOTO 760      SHIFT + COPY (Te
xtende)
900  IF s=CHR$(9) THEN isp=isp-isp MOD 10+11:GOTO
2800      TAB
910  IF s=CHR$(10) THEN isp=isp-isp MOD 10-9:GOTO
2800      SHIFT + TAB
920  IF s<>CHR$(16) AND s<>CHR$(246) THEN 990
930  *
940  *           *** CLR oder SHIFT+Cursor links ***
950  MID$(text(ize),1)=LEFT$(text(ize),isp-1)+MID
$(text(ize),isp+1)+" "
960  LOCATE 1,icu:PRINT text(ize);:GOTO 2800
970  *
980  *           *** DEL ***
990  IF s<>CHR$(127) THEN 1050
1000 IF isp=1 THEN PRINT#1,TAB(23);CHR$(7);"*** Z
eilenanfang erreicht ***":GOTO 760
1010 MID$(text(ize),1)=LEFT$(text(ize),isp-2)+MID
$(text(ize),isp)+" "
1020 LOCATE 1,icu:PRINT text(ize);:isp=isp-1:GOTO
2800
1030 *
1040 *           *** SHIFT+Cursor rechts: Zeichen einfu
egen ***
1050 IF s<>CHR$(247) THEN 1110
1060 MID$(text(ize),1)=LEFT$(text(ize),isp-1)+" "
+MID$(text(ize),isp)
1070 LOCATE 1,icu:PRINT text(ize);:GOTO 2800
1080 *
1090 *           *** SHIFT+Cursor oben: Zeile loeschen
***
1100 *           *** CTRL + COPY: Zeile in Puffer einl
esen
1110 IF ABS(ASC(s)-233)<>11 THEN 1180
1120 MID$(stack(ipo),1)=text(ize):ipo=ipo+1:IF ip
o>9 THEN ipo=9:FOR i=0 TO 8:MID$(stack(i),1)
=stack(i+1):NEXT i

```

```

1130 IF s<>CHR$(222) THEN ize=ize+1:icu=ic+1:GOTO
2800
1140 FOR i=ize TO maxze-1:MID$(text(i),1)=text(i+
1):NEXT i:MID$(text(maxze),1)=STRING$(80," "
)
1150 MID$(scrolldelete,6,1)=CHR$(ic+1):CALL PEEK
(@scrolldelete+1)+PEEK(@scrolldelete+2)*256:
LOCATE 1,24:PRINT text(iba+23);:GOTO 760
1160 '
1170 ' *** SHIFT+Cursor unten: Zeile einfueg
en ***
1180 IF s<>CHR$(245) THEN 1250
1190 u=""
1200 IF text(maxze)<>STRING$(80," ") THEN PRINT#1
,TAB(25);CHR$(7);"*** Textspeicher voll ***"
:GOTO 760
1210 FOR i=maxze TO ize STEP -1:MID$(text(i),1)=t
ext(i-1):NEXT i:MID$(text(ize),1)=STRING$(80
," ")
1220 MID$(scrollinsert,6,1)=CHR$(ic+1):CALL PEEK
(@scrollinsert+1)+PEEK(@scrollinsert+2)*256:
IF u="s" THEN 2360 ELSE 760
1230 '
1240 ' *** CTRL + Cursor rechts ***
1250 IF s<>CHR$(251) THEN 1300
1260 MID$(text(ize),isp)=SPACE$(80-isp+1)
1270 LOCATE 1,ic+1:PRINT text(ize);:GOTO 760
1280 '
1290 ' *** CTRL + Cursor links ***
1300 IF s<>CHR$(250) THEN 1340
1310 MID$(text(ize),1,isp)=SPACE$(isp)
1320 LOCATE 1,ic+1:PRINT text(ize);:GOTO 760
1330 '
1340 GOTO 760
1350 '
1360 ' *** Funktionen ***
1370 DI
1380 IF u<>"h" THEN 1580
1390 CLS#2:GOSUB 460
1400 PRINT#2
1410 PRINT#2," Ueber 'CTRL' erreichbar:"
1420 PRINT#2
1430 PRINT#2," M: Menue"
1440 PRINT#2," Z: Zeile anwaehlen"
1450 PRINT#2," S: Sichern geloeschter Zeilen"
1460 PRINT#2," F: Formatieren des Textes"
1470 PRINT#2," R: Randbreite einstellen"
1480 PRINT#2," T: Trennen der Zeile"
1490 PRINT#2," V: Vereinigen von Zeilen"
1500 PRINT#2," W: Worte suchen"
1510 PRINT#2," A: Austauschen von Worten"
1520 PRINT#2," H: Anzeigen dieser Information"
1530 LOCATE#1,25,1
1540 PRINT#1," < Leertaste! >"
1550 CALL &BBO6:GOSUB 2920:GOTO 750
1560 '
1570 ' *** Rueckkehr zum Menue ***
1580 IF u<>"m" THEN 1620
1590 RETURN
1600 '
1610 ' *** Sprung zu Zeile ***
1620 IF u<>"z" THEN 1670
1630 GOSUB 460
1640 LOCATE #1,28,1
1650 INPUT #1," Welche Zeile ";ize:IF ize<1 OR iz
e>maxze THEN PRINT #1,"Falsche Zeile. ";:GOT
O 1640
1660 iba=MIN(maxze-23,ize):ic+1=ize-iba+1:GOSUB 29
20:GOTO 750
1670 '
1680 ' *** Randbreite einstellen ***
1690 IF u<>"r" THEN 1760
1700 GOSUB 460
1710 LOCATE #1,25,1
1720 INPUT#1," Linke Randbreite ";rand:IF rand<0
OR rand>79 THEN 1720
1730 GOTO 750
1740 '
1750 ' *** Zeile trennen ***
1760 IF u<>"t" THEN 1880
1770 IF text(maxze)<>STRING$(80," ") THEN PRINT#1
,TAB(25);CHR$(7);"*** Textspeicher voll ***"
:GOTO 750
1780 IF ize=maxze THEN 750
1790 FOR i=maxze TO ize+1 STEP -1:MID$(text(i),1)
=text(i-1):NEXT i
1800 MID$(text(ize+1),1)=STRING$(rand,"")+RIGHT$
(text(ize),81-isp)+STRING$(MAX(isp-1-rand,0)
," ")
1810 MID$(text(ize),1)=LEFT$(text(ize),isp-1)+STR
ING$(81-isp," ")
1820 IF ic+1<23 THEN MID$(scrollinsert,6,1)=CHR$(ic
+1):CALL PEEK(@scrollinsert+1)+PEEK(@scrolli
nsert+2)*256
1830 LOCATE 1,ic+1:PRINT text(ize);
1840 IF ic+1<24 THEN LOCATE 1,ic+1:PRINT text(ize
+1);
1850 GOTO 750
1860 '
1870 ' *** Zeilen zusammensetzen ***
1880 IF u<>"v" THEN 2060
1890 IF ize=maxze THEN 750
1900 IF text(ize+1)=STRING$(80," ") THEN 2030
1910 il=1
1920 WHILE MID$(text(ize+1),il,1)=" "
1930 il=il+1
1940 WEND
1950 ir=80
1960 WHILE MID$(text(ize+1),ir,1)=" "
1970 ir=ir-1
1980 WEND
1990 IF RIGHT$(text(ize),81-isp)<>STRING$(81-isp,
" ") THEN PRINT #1:PRINT #1," *** Cursor is
t nicht am Zeilenende ***";CHR$(7):FOR i=1 T
O 3000:NEXT i:GOTO 750
2000 IF LEN(RIGHT$(text(ize),81-isp))<ir-1+1 THE
N PRINT #1:PRINT #1," *** Platz hinter Curs
or nicht ausreichend ***";CHR$(7):FOR i=1 TO
3000:NEXT i:GOTO 750
2010 MID$(text(ize),1)=LEFT$(text(ize),isp-1)+MID
$(text(ize+1),il,ir-il+1)
2020 LOCATE 1,ic+1:PRINT text(ize);
2030 FOR i=ize+1 TO maxze-1:MID$(text(i),1)=text(
i+1):NEXT i:MID$(text(maxze),1)=STRING$(80,"
")
2040 MID$(scrolldelete,6,1)=CHR$(ic+1):CALL PEEK(@
scrolldelete+1)+PEEK(@scrolldelete+2)*256:LO
CATE 1,24:PRINT text(iba+23);:GOTO 750
2050 '
2060 ' *** Text Formatieren ***
2070 IF u<>"f" THEN 2320
2080 merk=ize
2090 GOSUB 460
2100 LOCATE#1,25,1
2110 INPUT#1," Wieviele Zeilen ";anze
2120 zeil=ize+anze-1
2130 IF zeil>maxze THEN zeil=maxze
2140 FOR i=ize TO zeil
2150 IF RIGHT$(text(i),80-rand)=STRING$(80-rand
," ") THEN 2290
2160 anf=rand+1
2170 WHILE MID$(text(i),anf,1)=" "
2180 anf=anf+1
2190 WEND
2200 du=2
2210 l=INSTR(anf,text(i)," ")
2220 IF RIGHT$(text(i),80-1)=STRING$(80-1," ")
THEN 2290
2230 WHILE MID$(text(i),isp,1)=" " AND l<>0
2240 MID$(text(i),1)=LEFT$(text(i),l-1)+" "+M
ID$(text(i),l)
2250 l=INSTR(l+du,text(i)," ")
2260 WEND
2270 IF MID$(text(i),isp,1)=" " THEN du=du+1:GO
TO 2210
2280 IF ic+(i-merk)<25 THEN LOCATE 1,ic+(i-me
rk):PRINT text(i);
2290 NEXT i
2300 GOTO 750
2310 '
2320 IF u<>"s" THEN 2400
2330 EI
2340 PRINT MID$(text(ize),isp,1);
2350 GOTO 1200
2360 ipo=ipo-1:ipo=MAX(0,ipo)
2370 MID$(text(ize),1)=stack(ipo):LOCATE 1,ic+PR
INT text(ize);
2380 GOTO 750
2390 '
2400 ' *** Worte suchen und austauschen
***
2410 IF u<>"w" AND u<>"a" THEN 750
2420 PRINT#1:PRINT#1," Welches Wort suchen Sie ?
";:LINE INPUT#1,such
2430 subst=""
2440 IF u<>"a" THEN 2460
2450 PRINT#1," Mit welchem Wort wollen Sie austau
schen ? ";:LINE INPUT#1,subst

```

```

2460 t="":diff=LEN(such)-LEN(subst)
2470 ze=ize
2480 PRINT MID$(text(ize),isp,1);
2490 PRINT#1,TAB(30);"Ich suche..."
2500 WHILE ize<maxze+1 AND t<>"b"
2510   nr=1
2520   WHILE INSTR(nr,text(ize),such)<>0 AND t<
>"b"
2530     ze=ize
2540     icu=ize-iba+1
2550     IF icu>24 THEN iba=MIN(maxze-23,ize):icu
=ize-iba+1:GOSUB 2920
2560     isp=INSTR(nr,text(ize),such)
2570     t=""
2580     LOCATE 1,icu:PRINT CHR$(24);text(ize);CH
R$(24);:FOR i=1 TO 100:NEXT i:
LOCATE 1,icu:PRINT text(ize);
2590     LOCATE isp,icu
2600     PRINT CHR$(24);MID$(text(ize),isp,1);CHR
$(24);CHR$(8);
2610     IF u="a" THEN PRINT#1," W)eitersuchen
A)ustauschen B)enden" ELSE
PRINT#1," W)eitersuchen L)oeschen B
)enden"
2620     WHILE t="" :t=LOWER$(INKEY$):WEND
2630     PRINT MID$(text(ize),isp,1);
2640     IF t<>"a" AND t<>"1" THEN 2680
2650     IF diff<0 THEN IF RIGHT$(text(ize),-diff
)<>STRING$(-diff," ") THEN
PRINT#1,TAB(23);"*** Platz reicht nicht
aus ***";CHR$(7):FOR i=0 TO 1500:
NEXT:GOTO 2570
2660     MID$(text(ize),1)=LEFT$(text(ize),isp-1)
+subst+RIGHT$(text(ize),81-isp-
LEN(such))+STRING$(ABS(diff)," ")
2670     LOCATE 1,icu:PRINTtext(ize);
2680     nr=isp+1
2690     PRINT#1,TAB(30);"Ich suche..."
2700     WEND
2710     ize=ize+1
2720   WEND
2730   IF t="" THEN PRINT#1,"** Kein ""such"" im Te
xt vorhanden **":GOTO 2760
2740   IF t<>"b" THEN PRINT#1,"** Kein weiteres ""s
uch"" gefunden **":GOTO 2760
2750   GOTO 2770
2760   FOR i=0 TO 3500:NEXT
2770   ize=ze
2780   GOTO 750
2790   .
2800   .           *** Neue Cursorposition ***
2810   .
2820   IF isp=75 THEN PRINTCHR$(7);
2830   IF isp>80 THEN isp=1+rand:ize=ize+1:icu=icu+
1
2840   IF isp<1 THEN isp=80:ize=ize-1:icu=icu-1
2850   IF isp=80 AND icu=iba+23 THEN isp=79
2860   IF ize=maxze+1 THEN ize=maxze:icu=24:PRINT#1
,TAB(22);CHR$(7);"*** Textende erreicht ***"
:GOTO 760
2870   IF ize<1 THEN ize=1:icu=1:PRINT#1,TAB(22);CH
R$(7);"*** Textanfang erreicht ***":GOTO 760
2880   IF icu>24 THEN iba=iba+1:icu=24:CALL PEEK(@s
crollup+1)+PEEK(@scrollup+2)*256:LOCATE 1,24
:PRINT text(iba+23);:GOTO 760 ' aufwaerts sc
rollen (neue Zeile erscheint unten)
2890   IF icu<1 THEN iba=iba-1:icu=1:CALL PEEK(@scr
olldown+1)+PEEK(@scrolldown+2)*256:LOCATE 1,
1:PRINT text(iba);:GOTO 760 ' abwaerts scrol
len (neue Zeile erscheint oben)
2900   GOTO 760
2910   .
2920   .           *** Textschirm wiederherstellen ***
2930   .
2940   CLS:CLS#1
2950   FOR i=iba TO MIN(iba+23,maxze)
2960     IF text(i)=STRING$(80," ") THEN PRINT ELSE
PRINT text(i);
2970   NEXT i
2980   GOSUB 460
2990   RETURN
3000   .
3010   .           *** Loeschen ***
3020   .
3030   INPUT "Sind Sie sicher, dass Sie den Text lo
eschen wollen ? (Ja eingeben) ";s
3040   IF s<>"Ja" THEN RETURN
3050   FOR i=0 TO maxze
3060     MID$(text(i),1)=STRING$(80," ")
3070   NEXT i
3080   MID$(stack(0),1)=STRING$(80," "):ipo=0
3090   RETURN
3100   .
3110   .           *** Ausdrucken ***
3120   .
3130   laenge=maxze
3140   WHILE text(laenge)=STRING$(80," ")
3150     laenge=laenge-1
3160   WEND
3170   PRINT "Bitte Drucker vorbereiten und Druckko
pf auf Anfang erstes Blatt stellen!"
PRINT
3180   PRINT
3190   PRINT "Automatischer oder manueller Vorschub
nach jedem Blatt (A/M) ?"
3200   s=UPPER$(INKEY$)
3210   IF s<>"A" AND s<>"M" THEN 3200
3220   CLS
3230   INPUT "Anzahl der Zeichen pro Zeile auf dem
Drucker ";druckbreite
3240   IF druckbreite<1 OR druckbreite>255 THEN 322
0
3250   WIDTH druckbreite
3260   CLS
3270   PRINT"Schriftarten:":PRINT
3280   PRINT CHR$(i1);" K ";CHR$(i1);" Kursiv"
3290   PRINT CHR$(i2);" S ";CHR$(i2);" Schmalschri
ft"
3300   PRINT CHR$(i3);" B ";CHR$(i3);" Breitschrif
t"
3310   PRINT CHR$(i4);" F ";CHR$(i4);" Fettdruck"
3320   PRINT CHR$(i5);" D ";CHR$(i5);" Doppeldruck
"
3330   PRINT CHR$(i6);" E ";CHR$(i6);" Eliteschrif
t"
3340   PRINT" ^ : Seitenvorschub"
3350   PRINT" - : Zeilenvorschub"
3360   PRINT" I : Initialisieren"
3370   PRINT" R : Rueckkehr zum Hauptmenue"
3380   PRINT" P : ";CHR$(24);" DRUCKEN ";CHR$(24)
3390   ts=UPPER$(INKEY$):IF INSTR(" KSBFDE^-IRP",ts
)<2 THEN 3390
3400   CLS
3410   PRINT "Der Drucker ist nicht angeschlossen o
der nicht 'ON LINE' !"
3420   ON INSTR("KSBFDE^-IRP",ts) GOTO 3430,3440,34
50,3460,3470,3480,3490,3500,3510,3520,3530
3430   PRINT#8,CHR$(27);"4";:i1=24:GOTO 3260
3440   PRINT#8,CHR$(27);CHR$(15);:i2=24:GOTO 3260
3450   PRINT#8,CHR$(27);"W";CHR$(1);:i3=24:GOTO 326
0
3460   PRINT#8,CHR$(27);"E";:i4=24:GOTO 3260
3470   PRINT#8,CHR$(27);"G";:i5=24:GOTO 3260
3480   PRINT#8,CHR$(27);"M";:i6=24:GOTO 3260
3490   PRINT#8,CHR$(12);:GOTO 3260
3500   PRINT#8,CHR$(10);:GOTO 3260
3510   PRINT#8,CHR$(27);"@";:i1=0:i2=0:i3=0:i4=0:i5
=0:i6=0:GOTO 3260
3520   RETURN
3530   PRINT #8,CHR$(27);"A";CHR$(15); ' Hier spezi
elle Steuerung fuer Drucker-Zeilenaabstand
3540   CLS
3550   druckhoehe=53 ' Anzahl der Zeilen auf einem
Blatt, abhaengig vom Zeilenabstand
3560   IF szej="deutsch" THEN PRINT #8,CHR$(27);"R"
;CHR$(2);
3570   IF szej="engl." THEN PRINT #8,CHR$(27);"R";C
HR$(0);
3580   INPUT"Anzahl der Ausdruecke: ",anzk
3590   CLS
3600   FOR j=1 TO anzk
3610     FOR i=1 TO laenge
3620       PRINT #8,LEFT$(text(i)+STRING$(175,32),d
ruckbreite);
3630       IF i MOD druckhoehe>0 OR i=0 OR i=laenge
THEN 3670
3640       IF s="A" THEN PRINT #8,CHR$(12);:GOTO 36
70
3650       PRINT"Naechstes Blatt vorbereiten; wenn
fertig, ENTER druecken":CALL &BB06
3660       CLS
3670       NEXT i
3680       IF j=anzk THEN PRINT#8:GOTO 3730
3690       IF s="A" THEN PRINT #8,CHR$(12);:GOTO 3730
3700       PRINT#8
3710       IF j<anzk THEN PRINT"Naechstes Blatt vorbe
reiten; wenn fertig, ENTER druecken":CALL
&BB06
3720       CLS
3730       NEXT j
3740       GOTO 3260
3750       .
3760       .           *** Speichern bzw. Laden ***
3770       CLS:CLS#1
3780       PRINT"S: Speichern"
3790       PRINT"L: Laden bzw. Textfiles suchen"

```

```

3800 PRINT"R: Rueckkehr zum Hauptmenue"
3810 PRINT:PRINT" Bitte waehlen Sie!"
3820 s=LOWER$(INKEY$)
3830 IF s="r" THEN RETURN
3840 IF s<>"s" THEN 4060
3850 CLS:CLS#1:PRINT#1,TAB(26);"---- TEXT SPEICHE
RN ----"
3860 IF sdatum="" THEN INPUT "Bitte geben Sie das
heutige Datum fuer die Speicherung an : ",s
datum:PRINT
3870 PRINT"Aufzeichnungsgeschwindigkeit schnell o
der langsam (s/l) ?"
3880 SPEED WRITE 1
3890 t=LOWER$(INKEY$)
3900 IF t="s" THEN POKE &B8D1,2:POKE &B8D2,23:GOT
O 3920
3910 IF t<>"l" THEN 3890
3920 PRINT:INPUT "Textname : ",sdatei
3930 OPENOUT "supertext.file"
3940 laenge=maxze
3950 WHILE text(laenge)=STRING$(80,32) AND laenge
>0
3960 laenge=laenge-1
3970 WEND
3980 PRINT #9,sdatei
3990 PRINT #9,sdatum
4000 PRINT #9,laenge
4010 FOR i=1 TO laenge
4020 PRINT#9,text(i)
4030 NEXT
4040 CLOSEOUT
4050 GOTO 3770
4060 IF s<>"l" THEN 3820
4070 CLS
4080 PRINT:PRINT "Druecken Sie bitte die PLAY-Tas
te am Recorder":PRINT
4090 CLS#1:PRINT#1,TAB(30);"Ich suche..."
4100 OPENIN "!supertext.file"
4110 INPUT #9,sfilename,sfiledatum,laenge
4120 PRINT "Gefunden: ";sfilename;", aufgezeichne
t am ";sfiledatum;", Laenge:";laenge;"Zeilen
"
4130 PRINT#1," W)eitersuchen L)aden B)leenden"
4140 t=LOWER$(INKEY$):IF t="" THEN 4140
4150 IF t="w" THEN CLOSEIN:GOTO 4090
4160 IF t="b" THEN 4230
4170 IF t<>"l" THEN 4140
4180 CLS#1:PRINT#1,TAB(26);sfilename;" wird gelad
en"
4190 IF laenge>maxze THEN ERASE text:DIM text(lae
nge):maxze=laenge:PRINT:PRINT"Die maximale Z
eilenzahl wird auf"maxze" erhoeht !"
4200 FOR i=1 TO laenge
4210 LINE INPUT#9,text(i)
4220 NEXT
4230 CLOSEIN
4240 PRINT:PRINT"Druecken Sie die STOP-Taste am R
ecorder und eine beliebige Taste"
4250 CALL &BB06
4260 GOTO 3770
4270
4280
4290
4300 INPUT "Sind Sie sicher, dass Sie das Program
m beenden wollen (dadurch kann Ihr Text v
erloren gehen!) ? (Ja eingeben) ";s
4310 IF s<>"Ja" THEN RETURN
4320 IF szej<>"engl." THEN GOSUB 4550
4330 BORDER 0:PEN 1:PAPER 0
4340 CLS
4350 END
4360
4370
4380
4390 PRINT#1,TAB(34);"Moment bitte"
4400 SYMBOL AFTER 63
4410 SYMBOL 64,28,32,56,68,56,8,112,0
4420 SYMBOL 91,90,60,102,102,126,102,102,0
4430 SYMBOL 92,186,108,198,198,198,108,56,0
4440 SYMBOL 93,102,0,102,102,102,102,60,0
4450 SYMBOL 123,72,0,120,12,124,204,118,0
4460 SYMBOL 124,36,0,60,102,102,102,60,0
4470 SYMBOL 125,68,0,102,102,102,102,62,0
4480 SYMBOL 126,56,108,108,108,102,118,108,96
4490 KEY DEF 24,1,&5E,64
4500 KEY DEF 22,1,124,92
4510 KEY DEF 19,1,125,93

```

```

4520 KEY DEF 17,1,123,91
4530 KEY DEF 26,1,126,96
4540 RETURN
4550 PRINT#1,TAB(34);"Moment bitte"
4560 SYMBOL AFTER 240
4570 KEY DEF 24,1,&5E,&A3
4580 KEY DEF 26,1,&40,&7C
4590 KEY DEF 17,1,91,123
4600 KEY DEF 19,1,93,125
4610 KEY DEF 22,1,&5C,&60
4620 RETURN
4630 PRINT" Tastenbelegung mit d
eutschem Zeichensatz:"
4640 IF szej="deutsch" THEN GOSUB 4550
4650 PRINT:PRINT:PRINT
4660 LOCATE 35,5:PRINT"[ -"
4670 LOCATE 35,6:PRINT"\ -"
4680 LOCATE 35,7:PRINT"] -"
4690 LOCATE 35,8:PRINT"@ -"
4700 LOCATE 35,9:PRINT"{ -"
4710 LOCATE 35,10:PRINT"` -"
4720 LOCATE 35,11:PRINT"} -"
4730 LOCATE 35,12:PRINT"# -"
4740 GOSUB 4360:szej="deutsch"
4750 LOCATE 39,5:PRINT"{ "
4760 LOCATE 39,6:PRINT"; "
4770 LOCATE 39,7:PRINT"} "
4780 LOCATE 39,8:PRINT"^ "
4790 LOCATE 39,9:PRINT"[ "
4800 LOCATE 39,10:PRINT"\ "
4810 LOCATE 39,11:PRINT"] "
4820 LOCATE 39,12:PRINT"@ "
4830 CLS#1
4840 PRINT:PRINT:PRINT"E: einschalten"
4850 PRINT"A: ausschalten"
4860 PRINT:PRINT" Bitte waehlen Sie!"
4870 s=UPPER$(INKEY$)
4880 IF s="A" THEN GOSUB 4550:szej="engl.":
RETURN
4890 IF s="E" THEN RETURN
4900 GOTO 4870

```

- SHIFT + Cursor rechts: Fügt vor dem Cursor ein Zeichen ein. - SHIFT + Cursor links: Löscht das Zeichen auf Cursorposition. - SHIFT + Cursor unten: Fügt über dem Cursor eine Zeile ein. -SHIFT + Cursor oben: Löscht die Zeile auf Cursorposition. - COPY: Setzt Cursor auf Textanfang. -SHIFT + COPY: Setzt Cursor auf Textende. - CTRL + Cursor links: löscht den Rest der Zeile rechts des Cursors. - CTRL + Cursor rechts: Löscht den Rest der Zeile rechts des Cursors. - CTRL + COPY: Lädt die Zeile in den Puffer. -CTRL + „S“: Holt gelöschte oder mit CTRL + COPY geladene Zeile und setzt diese zwischen den Cursor und die Zeile darüber. -CLR oder DEL: Diese beiden Tasten besitzen die gewohnten Funktionen. -TAB: Bewegt den Cursor in Zehnerschritten vorwärts — SHIFT + TAB: Bewegt den Cursor in Zehnerschritten rückwärts.

# 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.

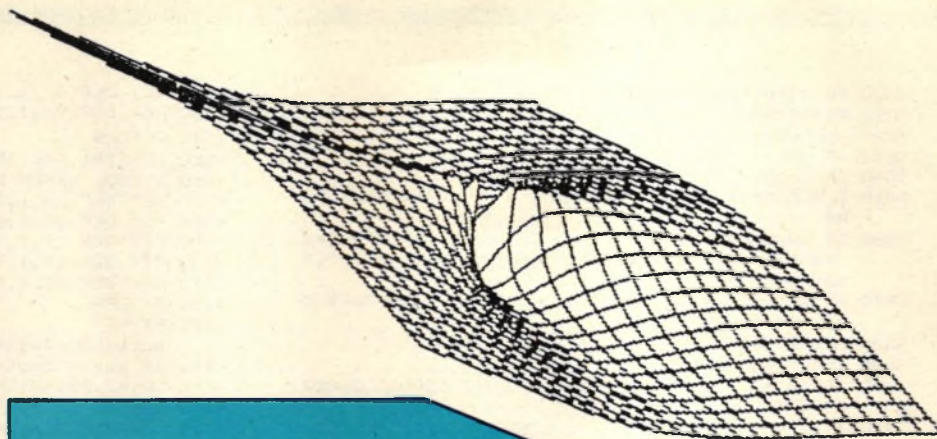
Bei der Darstellung der Funktion läuft die x-Achse nach hinten rechts, die y-Achse nach hinten links und die z-Achse nach oben. Die Funktion wird dann durch jeweils zur x- und y-Achse parallele Linienscharen dargestellt. Im Programm wird dabei für beide Richtungen die gleiche Schleife benutzt, die von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

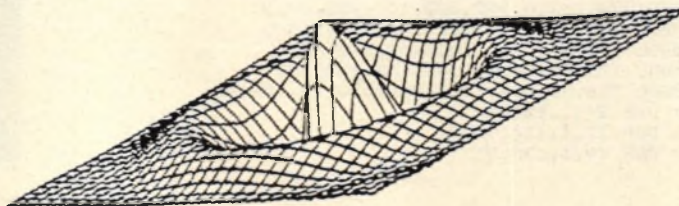
Der Bereich, in dem die Funktion geplottet wird, wird durch die Grenzen xmin und xmax, ymin und ymax, zmin und zmax angegeben. Alle außerhalb dieses Bereiches fallenden Punkte werden trotzdem gezeichnet, falls sie noch auf dem Bildschirm sichtbar sind.



```

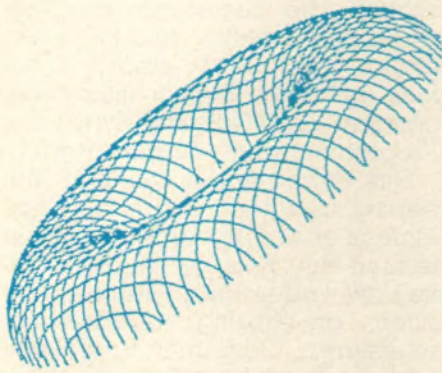
1 REM Frank Thielen
4 REM      &
5 REM Thomas Barndt
8 REM
10 MODE 2
20 BORDER 0
30 INK 1,26:PEN 1
40 INK 0,0:PAPER 0
50 MEMORY &9FFF
60 FOR i=&A000 TO &A0BF
70 READ byte:POKE i,byte:s=s+byte:NEXT
80 IF s<>23767 THEN PRINT"Fehler in datas !":END
90 CLEAR
100 DEFINT d,i,j,r,f,g,n,m
110 CLS
120 PRINT"                               3D-Plot":PRINT
    :PRINT "                               Copyright FTCP 198
    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

```

```

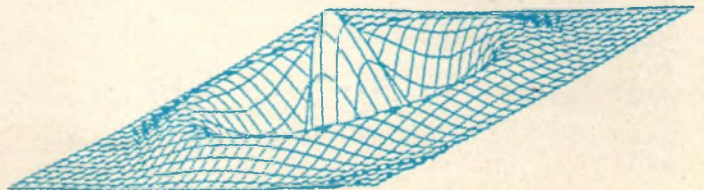
520   IF yp%>maxp(index) THEN maxp(index)=yp
     %:g=-1 ELSE IF yp%<minp(index) THEN mi
     np(index)=yp%:g=-1 ELSE g=0
530   IF j=0 THEN MOVE xp%,yp%:GOTO 560
540   IF f AND g THEN DRAW xp%,yp%
550   IF g THEN MOVE xp%,yp%
560   f=g
570   NEXT j
580 NEXT i
590 NEXT r
600 DIM rs(639)
610 FOR i=0 TO 639
620   rs(i)=PEEK(49152+i MOD 80+(i\80)*2048)
630 NEXT
640 LOCATE 1,1:PRINT"Hardcopy Bleenden Neustar
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);"@"
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,&cd,&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,&cb,&11
3260 DATA &cb,&11,&cb,&11,&c9,&00,&00,&00

```



# Hyperplot für CPC

Das Besondere an diesem Programm ist die Darstellung der Funktionsgraphen in einem Koordinatensystem mit relativ exakter Skaleneinteilung und die Eingabe von Funktionen (scheinbar) während des Programmablaufes. Außerdem können bis zu neun Funktionen vordefiniert und über Tastendruck aufgerufen werden.

Die Skaleneinteilung kann verändert werden, wodurch man Ausschnitte aus dem Graphen beliebig vergrößern kann. Der Ursprung des Koordinatensystems läßt sich beliebig, auch außerhalb des Bildschirms, verschieben. So lassen sich zum Beispiel Nullstellen ziemlich genau schätzen.

## Der Trick

Mit dem Befehl DEF FN xxx kann man im BASIC Funktionen mit Namen versehen und diese dann während des Programms mit

diesem Namen ansprechen, ohne jedesmal die Funktion neu eingeben zu müssen. Leider ist kein Befehl vorgesehen, mit dem man während des Programmlaufes Funktionen vom Benutzer einlesen kann. Die mit DEF FN definierten Funktionen können also nur durch eine Änderung des Programms neu definiert werden. In diesem Programm wird daher, wenn der Benutzer den Menüpunkt „Eingabe einer Funktion“ auswählt, eine Funktionstaste mit dem Befehl EDIT und der Zeilennummer der jeweiligen Funktion belegt. Dann wird, nachdem der Benutzer aufgefordert wurde, diese Funktions-

taste zu drücken, das Programm beendet. Wenn der Benutzer nun die Funktionstaste drückt, erscheint die Zeile mit der gewünschten Funktion und kann verändert werden. Mit einem, ebenfalls auf einer Funktionstaste liegenden, GOTO-Befehl wird das Programm dann neu gestartet.

Das Programm ist modular aufgebaut und kann dadurch relativ einfach erweitert werden. So könnte man zum Beispiel die Möglichkeit, zwei oder mehr Funktionsgraphen gleichzeitig darzustellen, realisieren. Oder man implementiert die Berechnung von Nullstellen und Extrema.(tb)

```

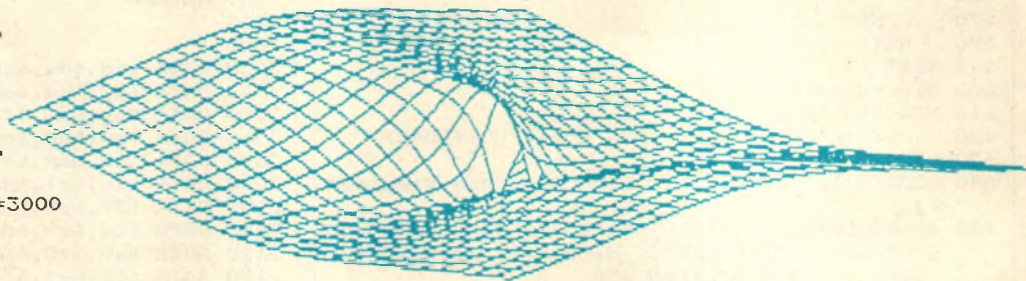
10 '(c) 1985
20 '   Thomas Barndt
50 '
60 DEFINT i
70 INK 0,0:INK 1,26:INK 2,25:INK 3,25:BORDER 8
80 MODE 0
90 LOCATE 1,10
100 PEN 2
110 PRINT TAB(6)"HYPERPLOT"
120 PEN 1
130 LOCATE 9,14
140 PRINT"by THECS"
150 LOCATE 3,24
160 PRINT CHR$(164);" 3/85"
170 FOR i=0 TO 3000
180   IF INKEY$<>" THEN i=3000
190 NEXT i
200 '
210 '
220 '   **** Funktionsnamen festlegen ****
230 '
240 CLEAR
250 DIM name$(9)
260 name$(1)="ln x"
270 name$(2)="sin x"
280 name$(3)="x^2"
290 name$(4)="x^4 (x^2 - 1)"
300 name$(5)="1/x - 3/x^3"
310 name$(6)="exp x"
320 name$(7)="!(x-1)^2 - 4!"
330 name$(8)="1/x"
340 name$(9)="tan x"
350 PEN 1:PAPER 0:BORDER 0
360 MODE 2
370 GOSUB 1320
380 x=320:y=200:mass=60
390 RAD
400 '
410 '   *****
420 '   ***  MENUE  ***
430 '   *****
440 '

```

```

450 wahl=0
460 CLS:ORIGIN 0,0
470 BORDER 10
480 LOCATE 22,2:PAPER 1:PEN 0
490 PRINT"           Waehlen Sie bitte
500 PRINT:PEN 1:PAPER 0:PRINT:PRINT

```



```

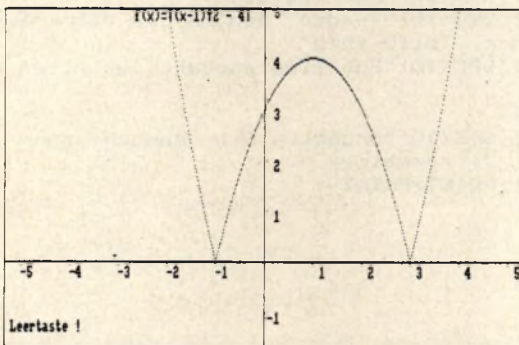
510 FOR i=1 TO 9
520   PRINT TAB(6);i". ";name$(i):PRINT
530 NEXT i
540 LOCATE 70,24:PRINTCHR$(164);" THECS"
550 LOCATE 40,6:PRINT"[P] - Funktion Plotten"
560 LOCATE 40,8:PRINT"[E] - Neue Einheit"
570 LOCATE 40,10:PRINT"[U] - Neuer Ursprung"
580 LOCATE 40,12:PRINT"[R] - Einheit+Ursprung
    normalisieren"
590 LOCATE 40,14:PRINT"[N] - Neue Funktion
    eingeben"
600 LOCATE 17,24:PRINT"Momentane Funktion: f(x)
    = "
610 LOCATE 44,24:PRINT SFC(16)
620 PEN 0:PAPER 1
630 LOCATE 44,24:PRINT" "name$(f)" "
640 PEN 1:PAPER 0
650 a$=INKEY$
660 wahl=VAL(a$)
670 IF a$="u" THEN wahl=10
680 IF a$="e" THEN wahl=11

```

```

690 IF a$="p" THEN wahl=12
700 IF a$="r" THEN 380
710 IF a$="n" THEN 2090
720 IF wahl<1 OR wahl>12 THEN 650
730 ON wahl GOSUB 1280,1320,1360,1400,1440,1480
1520,1560,1600,1860,1670,790
740 '
750 IF wahl<10 THEN 610 ELSE 430
760 '
770 '
780 '
790 '   ***   Koordinatenkreuz   ***
800 '
810 BORDER 0
820 CLS
830 PRINT TAB(20)"f(x)=";name$(f)
840 PLOT 0,y:DRAW 640,y
850 PLOT x,400:DRAW x,0

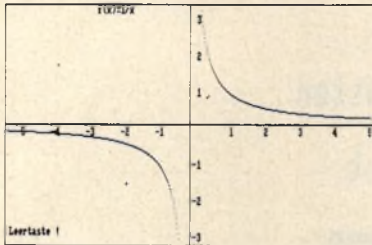
```



```

860 ORIGIN x,y
870 '
880 '
890 '   ***   SKALENEINTEILUNG   ***
900 '
910 IF mass<10 THEN 1110
920 TAG
930 FOR k=0 TO (640-x) STEP mass
940 PLOT k,-1:DRAW k,-6

```



```

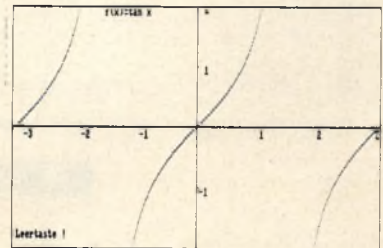
950 IF k/mass>0 AND mass>30 THEN PRINT k/mass
;
960 NEXT
970 FOR k=0 TO -x STEP -mass
980 PLOT k,-1:DRAW k,-6
990 IF k/-mass>0 AND mass>30 THEN PRINT k/mas
;
1000 NEXT
1010 FOR k=0 TO (400-y) STEP mass
1020 PLOT 1,k:DRAW 4,k
1030 IF k/mass>0 AND mass>30 THEN PRINT k/mass
;
1040 NEXT
1050 FOR k=0 TO -y STEP -mass
1060 PLOT 1,k:DRAW 4,k
1070 IF k/-mass>0 AND mass>30 THEN PRINT k/mas
;
1080 NEXT
1090 TAGOFF
1100 '
1110 '
1120 '   ***   FUNKTION PLOTTEN   ***
1130 '
1140 ON ERROR GOTO 1230
1150 FOR i=-x TO (640-x)
1160 PLOT i,FNfunktion (i/mass) * mass

```

```

1170 NEXT
1180 LOCATE 2,24:PRINT"Leertaste !"
1190 IF INKEY$<>" " THEN RETURN ELSE 1190
1200 ORIGIN 0,0
1210 ON ERROR GOTO 2310
1220 RETURN
1230 RESUME NEXT
1240 '
1250 '
1260 '   ***   Definieren der Funktionen   ***
1270 '
1280 DEF FNfunktion(x)=ln(x)
1290 f=1
1300 RETURN
1310 '
1320 DEF FNfunktion(x)=SIN(x)
1330 f=2
1340 RETURN
1350 '
1360 DEF FNfunktion(x)=x*x
1370 f=3
1380 RETURN
1390 '
1400 DEF FNfunktion(x)=x^4*(x^2-1)
1410 f=4
1420 RETURN
1430 '
1440 DEF FNfunktion(x)=1/x-3/x^3
1450 f=5
1460 RETURN
1470 '
1480 DEF FNfunktion(x)=EXP(x)
1490 f=6
1500 RETURN
1510 '

```



```

1520 DEF FNfunktion(x)=ABS((x-1)^2-4)
1530 f=7
1540 RETURN
1550 '
1560 DEF FNfunktion(x)=1/x
1570 f=8
1580 RETURN
1590 '
1600 DEF FNfunktion(x)=TAN(x)
1610 f=9
1620 RETURN
1630 '
1640 '
1650 '   ***   Festlegen der Einheit   ***
1660 '
1670 CLS
1680 MODE 1
1690 PRINT TAB(15)"f(x)=";name$(f)
1700 LOCATE 30,24:PRINT"Momentane Einheit : ";
1710 PEN 2
1720 PRINT mass;
1730 PEN 1
1740 PRINT"Punkte"
1750 LOCATE 1,6
1760 PRINT"   Geben Sie bitte die Laenge":
PRINT
1770 PRINT"   der gewuenschten Einheit in":PRIN
T
1780 PRINT"   Punkten an!":PRINT
1790 PRINT:PRINT:PRINT"   ";
1800 PEN 0:PAPER 1
1810 INPUT" ",mass
1820 PEN 1:PAPER 0
1830 MODE 2
1840 RETURN

```

```

1850 .
1860 .      *** Festlegen des Ursprungs ***

1870 .
1880 CLS
1890 MODE 1
1900 PRINT"          f(x)=";name$(f)
1910 LOCATE 30,24:PRINT"Momentaner Ursprung : ";
1920 PEN 2
1930 PRINT x;",";y
1940 PEN 1
1950 LOCATE 1,6
1960 PRINT"      Geben Sie bitte die Koordinaten":
PRINT
1970 PRINT"      des Ursprungs in folgender Form":
PRINT
1980 PRINT"      an :   x,y"
1990 PRINT:PRINT:PRINT SPC(15);
2000 PEN 0:PAPER 1
2010 INPUT" ",x,y
2020 PEN 1:PAPER 0
2030 MODE 2
2040 RETURN
2050 .
2060 .
2070 .      *** FUNKTION EINGEBEN ***
2080 .
2090 KEY 138,"goto 450"+CHR$(13)

2100 CLS
2110 PRINT SPC(10)"Geben Sie die Nummer der Funkt
ion ein, welche Sie aendern"
2120 PRINT SPC(10)"wollen !"
2130 PRINT
2140 a$=""
2150 WHILE a$="" :a$=INKEY$:WEND
2160 a=VAL(a$)
2170 IF a<1 OR a>9 THEN 2140
2180 INPUT"      Name der neuen Funktion: ",n
ame$(a)
2190 fu=1280+(a-1)*40
2200 fu$=STR$(fu)
2210 KEY 128,CHR$(13)+"edit "+fu$+CHR$(13)
2220 CLS
2230 PRINT SPC(10)"Druecken Sie die [O] im Ziffer
nblock rechts neben"
2240 PRINT SPC(10)"der Tastatur."
2250 PRINT SPC(10)"Tragen Sie in die dann ersch
einende Zeile ihre"
2260 PRINT SPC(10)"Funktion anstatt der Alten ein
!"
2270 PRINT
2280 PRINT SPC(10)"Druecken Sie danach den [.]
im Ziffernblock."
2290 PRINT:PRINT:PRINT
2300 END
2310 CLS

```

### Wählen Sie bitte

- |                     |                                      |
|---------------------|--------------------------------------|
| 1 . $\ln x$         | [P] - Funktion Plotten               |
| 2 . $\sin x$        | [E] - Neue Einheit                   |
| 3 . $x^2$           | [U] - Neuer Ursprung                 |
| 4 . $x^4 (x^2 - 1)$ | [R] - Einheit+Ursprung normalisieren |
| 5 . $1/x - 3/x^3$   | [N] - Neue Funktion eingeben         |
| 6 . $\exp x$        |                                      |
| 7 . $ (x-1)^2 - 4 $ |                                      |
| 8 . $1/x$           |                                      |
| 9 . $\tan x$        |                                      |

Momentane Funktion:  $f(x) = \sin x$

© THBCS

# Der Grafik- Macher



Zwar haben die beiden Schneider Computer 464 und 664 eine ausgezeichnete Grafik, aber die etwas spärlichen Grafikbefehle machen es dem Benutzer nicht gerade leicht, ein vernünftiges Bild auf den Bildschirm zu bringen. Solange es sich noch um mathematisch beschreibbare Figuren handelt, geht das in BASIC ja noch gerade, aber haben Sie schon einmal versucht, z.B. ein Portrait auf den Bildschirm zu bringen? Jetzt können Sie es! Mit dem Grafikeditor sind Ihrer Phantasie keine Grenzen gesetzt. Sie können frei Hand zeichnen, Linien,

Rechtecke, Kreise zaubern Sie in Sekundenschnelle auf den Bildschirm. Flächen können automatisch ausgemalt, verschoben oder vergrößert/verkleinert werden. Sie können beliebig Text in Ihr Bild einfügen, Bilder im Speicher und auf Kassette/Diskette ablegen und sie in Ihren eigenen Programmen verwenden. Wenn Sie den Grafikeditor abgetippt (vor dem ersten Testlauf abspeichern, da Maschinensprache verwendet wird, und so das ganze Programm durch einen Fehler vernichtet werden könnte!) und alle Fehler korrigiert

haben, starten Sie ihn mit RUN. Auf dem Bildschirm baut sich die Titelf grafik auf, und Sie können wählen, in welchem Bildschirmmodus Sie arbeiten möchten. Drücken Sie einfach 0,1 oder 2. Anschließend fragt das Programm, ob Sie die Farben ändern möchten. Wenn nicht (N), dann erscheint direkt der leere Bildschirm mit dem blinkenden Cursor in der Mitte, und Sie können zeichnen. Andernfalls, wenn Sie mit J(a) geantwortet hatten, können Sie jede einzelne Farbtaste mit einer beliebigen Farbe belegen. Drücken Sie Z oder X, um die Far-

ben am Bildschirmrand zu ändern, wenn Sie ENTER drücken, wird der angezeigten Taste diese Farbe zugewiesen. Denken Sie daran, daß die erste Taste für die Farbe des Hintergrunds zuständig ist.

Wenn Sie sich im Zeichenmodus befinden, haben Sie folgende Kommandos zur Verfügung:

Die Tasten des 10er Blocks bewegen den Cursor (den kleinen blinkenden Punkt) in die entsprechende Richtung, also z.B. 7 nach links oben, 2 gerade nach unten. Die 5 hat keine Funktion. Sie können mit dem Cursor auch über bereits Gemaltes fahren, ohne daß dieses verändert wird.

Die Leertaste zeichnet in der gerade gewählten Farbe, wenn Sie zusammen mit einer Taste des Cursorblocks zusammen gedrückt wird.

L zieht eine Linie. Der Startpunkt der Linie wird festgesetzt, wenn Sie das erste Mal L drücken. Beim zweiten Druck wird die Linie von

```

10 REM
20 REM
30 REM
40 REM
50 RESTORE: a=&4200: MEMORY &61FF
60 READ b: IF b=999 THEN 100 ELSE POKE a, b: a=a+1: GOTO 60
70 DATA &21, 0, &c0, 1, &ff, &3f, &11, &40, &62, &ed, &b0, &c9
80 DATA &21, &40, &62, 1, &ff, &3f, &11, 0, &c0, &ed, &b0, &c9
90 DATA 999
100 ON BREAK GOSUB 1490
110 DATA 0, 3, 1, 12, 13, 15, 16, 9, 4, 5, 6, 11, 21, 22, 23, 24
120 DEFINT a-z: DIM farbe(16), z(980), b(80, 7): DEG: RESTORE 110: ENV 1, 100, -2, 10
130 FOR x=1 TO 15: READ farbe(x): NEXT
140 KEY 140, "ink 0, 27: ink 1, 0: border 27: mode 2: speed key 10, 2"+CHR$(13)
150 INK 0, 0: INK 1, 1: INK 2, 3: INK 3, 12: BORDER 0
160 MODE 1: farbe=1: FOR x=0 TO 640 STEP 7
170 PLOT 320, 400, 4-farbe: DRAW x, 280
180 PLOT 320, 18, farbe: DRAW x, 310
190 farbe=farbe+1: IF farbe=4 THEN farbe=1
200 NEXT: LOCATE 2, 7
210 PRINT "Graphik-Editor von Thomas M. Binzinger."
220 EVERY 20, 0 GOSUB 1580
230 LOCATE 7, 25: PRINT "Bildschirmmodus 0, 1 oder 2 ?"
240 modus=4
250 as=INKEY$
260 IF as="0" THEN modus=0: fanz=16: xadd=4: yadd=2
270 IF as="1" THEN modus=1: fanz=4: xadd=2: yadd=2
280 IF as="2" THEN modus=2: fanz=2: xadd=1: yadd=2
290 IF modus=4 THEN 230
300 LOCATE 4, 25: PRINT "Möchten Sie die Farben ändern ?"
310 as=INKEY$: IF as="n" THEN 440 ELSE IF as="" THEN 310
320 LOCATE 1, 25: PRINT STRINGS(40, 32);
330 FOR x=1 TO fanz
340 IF x>9 THEN fanz
350 LOCATE B, 25
360 PRINT I$; "X"; CHR$(48+x-add); ":add=9 ELSE I$="";
370 f=0
380 as=INKEY$
390 IF as="z" THEN f=f-1
400 IF as="x" THEN f=f+1
410 IF f<0 THEN f=27 ELSE IF f>27 THEN f=0
420 BORDER f: IF as<>CHR$(13) THEN 380 ELSE farbe(x)=f
430 NEXT
440 MODE modus: PAPER 0: PEN 1: XPD=320: YPD=200: a=REMAIN(0)
450 FOR x=1 TO 16: INK x-1, farbe(x): NEXT
460 a: farbe=1: flag=1: BORDER 0: add=1
470 xalt=-1: yalt=-1: xqalt=-1: yqalt=-1
480 xalt=0: yalt=0
490 IF INKEY(14)=0 OR (JOY(0) AND 2)=2 THEN y=-yadd
500 IF INKEY(11)=0 OR (JOY(0) AND 1)=1 THEN y=yadd
510 IF INKEY(4)=0 OR (JOY(0) AND 4)=4 THEN x=-xadd
520 IF INKEY(5)=0 OR (JOY(0) AND 8)=8 THEN x=xadd
530 IF INKEY(10)=0 THEN x=xadd: y=-yadd
540 IF INKEY(3)=0 THEN x=-xadd: y=yadd
550 IF INKEY(13)=0 THEN x=xadd: y=yadd
560 IF INKEY(18)=0 THEN xadd=xadd+1: yadd=yadd+1
570 IF INKEY(17)=0 THEN xadd=xadd+1: yadd=yadd+1
580 IF INKEY(16)=32 THEN modus=1 THEN xadd=2 ELSE IF modus=2 THEN xadd=1 ELSE xadd=4
590 IF INKEY(36)=0 THEN GOSUB 810 ' Linie ziehen
600 IF INKEY(37)=0 THEN GOSUB 850 ' Kreis
610 IF INKEY(67)=0 THEN GOSUB 930 ' Kreis gefüllt
620 IF INKEY(51)=0 THEN GOSUB 980 ' Quadrat
630 IF INKEY(67)=32 THEN GOSUB 1090 ' Text
640 IF INKEY(37)=32 THEN GOSUB 1130 ' Quadrat gefüllt
650 IF INKEY(27)=0 THEN GOSUB 1190: a: farbe=a: GOTD 500
660 IF INKEY(71)=0 THEN GOSUB 1190: fanz=a: BORDER
670 farbe(a+1): GOTD 500
680 IF INKEY(53)=0 THEN a1=fanz: fanz=16: GOSUB 1190: fanz=a1: BORDER
690 IF INKEY(69)=32 THEN GOSUB 1240
700 IF INKEY(44)=32 THEN GOSUB 1430
710 IF INKEY(60)=0 THEN GOSUB 1460
720 IF INKEY(44)=0 THEN GOSUB 1730 ' Schieben
730 IF INKEY(44)=0 THEN GOSUB 1900
740 IF INKEY(53)=32 THEN GOSUB 1950 ' Ausdrucken
750 IF INKEY(69)=0 THEN GOSUB 2160 ' musterfüllen
760 IF INKEY(61)=0 THEN GOSUB 2340 ' z=zoomen
770 IF INKEY(71)=32 THEN GOSUB 1620
780 IF x<>0 OR y<>0 THEN GOSUB 1620

```

diesem Startpunkt zur augenblicklichen Position des Cursors gezogen. Die Linie hat die augenblicklich angewählte Farbe.

K zieht einen Kreis in der augenblicklich angewählten Farbe. Der erste Druck auf K setzt den Kreis so, daß der »Kreis durch die augenblickliche Cursorposition läuft.« Wird das zweite K zusammen mit SHIFT gedrückt, wird der Kreis ausgefüllt gezeichnet.

Q zeichnet ein Quadrat in der augenblicklichen Farbe. Der erste Druck auf Q spezifiziert die obere linke, der zweite Druck die untere rechte Ecke. Das Quadrat wird ausgefüllt gezeichnet, wenn das zweite Q mit SHIFT gedrückt wird.

T drückt den Text, den Sie nach dem T eingeben, an die augenblickliche Cursorposition. Der Text hat die augenblicklich angewählte Farbe. Nach dem letzten Buchstaben muß ENTER gedrückt werden, um das Kommando zu beenden. Die CLR und DEL, überhaupt alle Kontrolltasten ergeben nur Zei-



# Für Ihren Schneider CPC

AVAILABLE  
NOW

Thomas Binzingers Grafik-Test-Adventure

# DARK POWERS

Version für Schneider 6128 in Kürze lieferbar



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 wahren 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

## Utility

chen und führen keine Funktion von DEL aus.  
Z bestimmt die Zeichenfarbe. Nach Z muß eine der Farbtasten (Modus 2:1/2, Modus 1:1-0, Shift & 1-6) gedrückt werden, um die Farbe auszuwählen. Z und SHIFT haben eine andere Bedeutung, siehe unten. F füllt eine allseitig begrenzte Fläche mit der augenblicklichen Farbe. F und SHIFT füllen die Fläche mit einem Muster. Es muß 1-6 gedrückt werden, um zu bestimmen, mit welchem A legt das Bild im Speicher ab. Es kann mit H wieder geholt werden. Dabei wird allerdings alles, was vorher auf dem Bildschirm war, überschrieben.  
SHIFT und A legt ein Bild auf Kassette/Diskette ab, SHIFT und H holen ein Bild von Kassette/Diskette auf den Bildschirm. ESC unterbricht den Zeichenmodus. Man hat die Möglichkeit, das Programm neu zu starten. Dann kann man z. B.

schneider spezial

```
790 IF INKEY(16)=0 THEN PLOT xpo,ypo,farbe:GOSUB 1710:xpo=320:ypo=200
farbe=TEST(xpo,ypo)
800 GOSUB 1600:GOTO 500
810 REM ----- Linie ziehen -----
820 GOSUB 1710
830 IF xalt<>-1 THEN DRAW xalt,yalt,afarbe:MOVE XPO,YPO:xalt=-1:GOTO 1690
840 xalt=XPOS:yalt=YPOS:PLOT XPO,YPO,afarbe:GOTO 1690
850 REM ----- Kreis zeichnen -----
860 GOSUB 1710
870 IF ymalt=-1 THEN 920 ELSE radius=SQR((XPO-xmalt)^2+(YPO-ymalt)^2)
880 PLOT radius+xmalt,ymalt,afarbe
890 FOR winkel=0 TO 360 STEP 12
900 DRAW COS(winkel)*radius+xmalt,SIN(winkel)*radius+ymalt,afarbe
910 NEXT:PLOT xmalt,ymalt,0:ymalt=-1:GOTO 1690
920 PLOT XPO,YPO,afarbe:PLOT xmalt,ymalt,0:ymalt=YPO:xmalt=XPO:GOTO 1690
930 REM ----- Quadrat -----
940 GOSUB 1710
950 IF xqalt=-1 THEN xqalt=xpo:yqalt=ypo:PLOT xpo,ypo,afarbe:GOTO 1690
960 MOVE xqalt,yqalt:DRAW xpo,yqalt,afarbe:DRAW xpo,ypo,afarbe:DRAW xqalt,
ypo,afarbe:DRAW xqalt,yqalt,afarbe
970 MOVE xpo,ypo:xqalt=-1:GOTO 1690
980 REM ----- Text -----
990 GOSUB 1710
1000 TAG:PLOT xpo,ypo,afarbe
1010 IF modus=0 THEN zadd=32 ELSE IF modus=1 THEN zadd=16 ELSE zadd=8
1020 a$=INKEY$:IF a$="" THEN 1020 ELSE b$=""
1030 FOR a=1 TO 200:b$=b$+INKEY$:NEXT
1040 a$=INKEY$:IF a$="" THEN 1040
1050 IF a$=CHR$(13) THEN PLOT xpo,ypo,farbe:TAGOFF:RETURN
1060 IF a$=CHR$(242) THEN PLOT xpo,ypo,farbe:xpo=xpo-zadd-1:PLOT xpo,ypo,
afarbe:PRINT " ";PLOT xpo,ypo,afarbe:GOTO 1040
1070 PRINT a$:IF xpo+zadd+1<641-zadd THEN xpo=xpo+zadd+1
1080 PLOT xpo,ypo,afarbe:GOTO 1040
1090 REM -----Quadrat gefuellt -----
1100 GOSUB 1710
1110 IF xqalt=-1 THEN 950
1120 FOR x=xqalt TO xpo STEP SGN(xpo-xqalt):MOVE x,yqalt:DRAW x,yqo,afarbe:
NEXT:GOTO 970
1130 REM ----- Kreis gefuellt -----
1140 GOSUB 1710
1150 IF ymalt=-1 THEN 920 ELSE radius=SQR((xpo-xmalt)^2+(ypo-ymalt)^2)
1160 FOR winkel=0 TO 360:PLOT xmalt,ymalt,afarbe
1170 DRAW COS(winkel)*radius+xmalt,SIN(winkel)*radius+ymalt,afarbe
1180 NEXT:ymalt=-1:MOVE xpo,ypo:RETURN
1190 REM ----- Farbe bestimmen -----
1200 b$="":FOR t=1 TO 100:b$=b$+INKEY$:NEXT
1210 a$=INKEY$:IF a$="" THEN 1210 ELSE a=ASC(a$)
1220 IF (a>48 AND a<58) THEN a=a-49 ELSE IF (a>32 AND a<42) THEN a=a-22
ELSE 1210
1230 IF a>=fanz THEN 1210 ELSE GOSUB 1710:GOTO 1690
1240 REM ----- Fuellen -----
1250 mf=0
1260 a=xadd:b=yadd:yadd=2:IF modus=0 THEN xadd=4 ELSE IF modus=1 THEN
xadd=2 ELSE xadd=1
1270 GOSUB 1710:PLOT xpo,ypo,0
1280 x=xpo:y=ypo
1290 ns=1:ya=-yadd
1300 GOSUB 1350:ns=0:x=xpo:y=y+yadd:IF y>400 THEN 1310 ELSE IF TEST(x,y)<>0
THEN GOSUB 1350 ELSE 1300
1310 x=xpo:y=ypo-yadd
1320 ya=yadd
1330 GOSUB 1350:x=xpo:y=y-yadd:IF y<0 THEN 1340 ELSE IF TEST(x,y)<>0
THEN GOSUB
1350 ELSE 1330
1340 xadd=a:yadd=b:GOTO 1690
1350 REM ----- Zeile -----
1360 IF TEST(x,y)=0 THEN GOSUB 2260:x=x-xadd ELSE x=xpo+xadd:GOTO 1380
1370 IF x<0 THEN 1380 ELSE 1350
1380 x=xpo+xadd
1390 IF TEST(x,y)=0 THEN GOSUB 2260:x=x+xadd ELSE 1410
1400 IF x>640 THEN 1410 ELSE 1390
1410 REM
1420 RETURN
1430 REM ----- Bild ablegen -----
1440 GOSUB 1710
1450 SPEED WRITE 1:PLOT xpo,ypo,afarbe:GOSUB 2500:SAVE name$,b,&C000,&3FFF,
0:RETURN
1460 REM ----- Bild holen -----
1470 GOSUB 1710
1480 GOSUB 2500:LOAD name$,&C000:farbe=TEST(xpo,ypo):RETURN
1490 REM ----- Break gedruickt -----
1500 ON BREAK STOP:PEN 1:PAPER 0:MODE 1:LOCATE 10,10
1510 PRINT CHR$(24)"B"CHR$(24)"asic oder "CHR$(24)"N"CHR$(24)"eustart ?"
1520 b$="":FOR a=1 TO 100:b$=b$+INKEY$:NEXT
1530 a$=INKEY$
1540 IF a$="b" THEN CLS:END
1550 IF a$="n" THEN RUN
```

die Farben ändern, und das Bild, an dem man vorher gearbeitet hatte, aus dem Speicher wieder auf den Bildschirm holen, vorausgesetzt natürlich, man hat es vorher mit A abgelegt.  
Mit S kann man einen quadratischen Bildausschnitt wie beim Q-Dazu muß das Quadrat wie beim Q-Befehl festgelegt werden (zweimal S drücken), und anschließend muß 1 oder 2 gedrückt werden, um zu bestimmen, ob nach oben oder unten verschoben werden soll. Es wird jeweils um eine Pixel-Reihe verschoben.  
Mit Z und SHIFT kann man einen Bildschirmbereich an eine andere



Stelle kopieren, bzw. vergrößern/verkleinern. Dazu muß der Bildschirmbereich, der kopiert werden soll, durch zweimaliges Drücken von SHIFT und Z festgelegt werden. Dann auf dieselbe Weise der Zielbereich. Wenn der Zielbereich genauso groß ist wie der ursprüngliche, wird nur ein Du-

plikat desselben hergestellt. Sonst wird der ursprüngliche Bereich so vergrößert oder verkleinert, daß er in den neuen Bereich hineinpaßt. So kann man z.B. einen Programmtitle in die untere Ecke des Bildschirms schreiben (mit T), und ihn dann mit dem Shift & Z (Zoom)-Kommando auf die ganze Bild-

schirmbreite (und/oder Höhe) „aufblähen“.

R ändert die Farbe des Bildschirmrandes. Es funktioniert ansonsten genauso wie das Z-Kommando (Zeichenfarbe).

P ändert die dem Hintergrund zugehörige INK. Nach P eine entsprechende Farbwahl Taste drücken.

```

1560 GOTO 1530
1570 END
1580 farbe=farbe+1:IF farbe>=24 THEN farbe=1
1590 INK 1, farbe:INK 2, farbe+1:INK 3, farbe+2:RETURN
1600 IF flag=1 THEN flag=0:farbe=TEST(xpo, ypo):PLOT xpo, ypo, RND*3:RETURN
1610 flag=1:PLOT xpo, ypo, farbe:RETURN
1620 PLOT xpo, ypo, farbe:xpo=xpo+x:ypoy=ypoy+y
1630 IF xpo>640 THEN xpo=0 ELSE IF xpo<0 THEN xpo=640
1640 IF ypo>400 THEN ypo=0 ELSE IF ypo<0 THEN ypo=400
1650 IF INKEY(47)<>0 AND (JOY(0) AND 16)<>16 THEN 1670
1660 PLOT xpo, ypo, afarbe
1670 x=0:y=0:farbe=TEST(xpo, ypo)
1680 RETURN
1690 REM Zeitschleife
1700 FOR t=1 TO 500:NEXT:farbe=afarbe:PLOT xpo, ypo, farbe:x=0:y=0:RETURN
1710 REM Bestaetigungston
1720 SOUND 1, 400, 50, 15, 1:RETURN
1730 REM ----- Schieben -----
1740 IF xalt=-1 THEN xalt=xpo:yalt=ypoy:GOSUB 1710:GOTO 1690
1750 b$="":FOR t=1 TO 100:b$=b$+INKEY$:NEXT
1760 a$=INKEY$:IF a$="" THEN 1760 ELSE IF a$<>"1" AND a$<>"2" THEN 1760
1770 GOSUB 1710
1780 IF a$="2" THEN 1820
1790 FOR y=ypoy+yadd TO yalt STEP yadd
1800 FOR x=xalt TO xpo STEP xadd
1810 a=TEST(x, y):PLOT x, y-yadd, a:NEXT x, y:xalt=-1:GOTO 1860
1820 FOR y=yalt TO ypo STEP -yadd
1830 FOR x=xalt TO xpo STEP xadd
1840 IF TEST(x, y)<>0 THEN PLOT x, y, afarbe
1850 NEXT x, y:xalt=-1
1860 a$=INKEY$
1870 IF a$="1" THEN 1790
1880 IF a$="2" THEN 1820
1890 GOTO 1690
1900 REM ----- Bild in internen Speicher -----
1910 GOSUB 1710:CALL &6200:FOR x=1 TO 5:MODE modus:FOR t=1 TO 50:NEXT:CALL &620C
:FOR t=1 TO 50:NEXT:NEXT:GOTO 1690
1920 REM ----- Bild aus internen Speicher -----
1930 GOSUB 1710:MODE modus:CALL &620C:GOTO 1690
1940 REM Bildschirm ausdrucken
1950 GOSUB 1710
1960 IF modus=0 THEN s=4 ELSE IF modus=1 THEN s=2 ELSE s=1
1970 PRINT #8, CHR$(27)"A"CHR$(7);
1980 y=399
1990 x=0
2000 WHILE (x<639) AND (INKEY(18)<>0)
2010 z=128:z(x+160)=0:z(x+161)=0
2020 FOR y1=y+2 TO y-12 STEP -2

```

```

2030 u=TEST(x,y,1):PLOT x,y,1:IF u=0 THEN 2070
2040 IF (u=1) OR (u>3) THEN z(x+160)=z(x+160)+z:z(x+161)=z(x+161)+z
2050 IF u=2 THEN z(x+160)=z(x+160)+z
2060 IF u=3 THEN IF flag=1 THEN z(x+160)=z(x+160)+z:flag=-1 ELSE flag=1
2070 z=z/2:NEXT y
2080 x=x+s:WEND
2090 xx=x
2100 PRINT #B,CHR$(13)CHR$(27)"L"CHR$(192)CHR$(3):
2110 FOR t=0 TO 959
2120 IF t<=xx+160 THEN PRINT #B,CHR$(INT(z(t))) ELSE PRINT #B,CHR$(0):
2130 z(t)=0:NEXT t
2140 y=y-14:IF INKEY(47)=0 THEN 2150 ELSE IF y>0 THEN 1990
2150 GOTO 1690
----- MusterFuelien -----
2160 REM
2170 bs="":FOR t=1 TO 100:bs=bs+INKEY$:NEXT t
2180 a$=INKEY$:IF a$<"1" THEN 2180 ELSE a$=VAL(a$)-1
2190 IF a=0 THEN f1=0:f2=1
2200 IF a=1 THEN f1=0:f2=2
2210 IF a=2 THEN f1=1:f2=2
2220 IF a=3 THEN f1=1:f2=3
2230 IF a=4 THEN f1=2:f2=3
2240 IF a=5 THEN f1=2:f2=3
2250 mf=1:GOSUB 1260:mf=0:RETURN
2260 REM
2270 IF flag<>1 THEN PLOT x,y,afarbe:RETURN
2280 IF ns=1 THEN 2320
2290 a=TEST(x,y,ya):IF a=f1 THEN PLOT x,y,f1 ELSE PLOT x,y,f2
2300 RETURN
2310 IF flag=1 THEN PLOT x,y,f1 ELSE PLOT x,y,f2
2320 RETURN
----- Zoomen -----
2330 REM
2340 GOSUB 1710
2350 IF xqalt0=-1 THEN xqalt0=xpo:yqalt0=yppo:PLOT xpo,ypo,afarbe:GOTO 1690
2360 IF xqalt1=-1 THEN xqalt1=xpoi:yqalt1=ypoi:MOVE xqalt0,yqalt0:DRAW xpo,yqalt0,afarbe:DRAW xqalt2,yqalt2,afarbe:DRAW xqalt0,yqalt0,afarbe:GOTO 1690
2370 IF xqalt2=-1 THEN xqalt2=xpo:yqalt2=yppo:GOTO 1690
2380 IF xqalt3=xpoi:yqalt3=ypoi:MOVE xqalt2,yqalt2:DRAW xpo,yqalt2,afarbe:DRAW xqalt0,yqalt0,afarbe:DRAW xqalt2,yqalt2,afarbe:DRAW xqalt0,yqalt0,afarbe:GOTO 1690
2390 xzerr=(xqalt3-xqalt2)/(xqalt1-xqalt0):yzerr=(yqalt3-yqalt2)/(yqalt1-yqalt0)
2400 xzerr=(xzerr+yqalt2)/xqalt0:yzerr=(yzerr+yqalt2)/yqalt0:IF modus=1 THEN xadd=2 ELSE
E xadd=1
2420 xpo=xqalt1:yppo=yqalt1:MOVE xqalt0,yqalt0 TO yqalt2-yzerr*yadd STEP -1
W xqalt0,yppo:DRAW xqalt0,yqalt0
2430 FOR x=0 TO xqalt1-xqalt0-1 STEP xadd:FOR y=0 TO yqalt0-yqalt1-1 STEP yadd
2440 f=TEST(x+xqalt0,y+yqalt0,yzerr TO yqalt2-yzerr*yadd STEP -1
2450 MOVE x*xzerr+xqalt2,y:DRAW x*xzerr+xqalt2+xzerr*xadd,y,fi:NEXT y
2460 NEXT y,x
2470 NEXT y,x
2480 xadd=a:yadd=b
2490 REM
2500 REM
2510 FOR x=0 TO 79:FOR y=0 TO 7:a=C000+x+y*2048:b(x,y)=PEEK(a):POKE a,255:NEXT
T y,x
2520 LOCATE 1,1:PRINT"Filename:.....":1=0:name$="":FOR t=1 TO 100:b$=INKEY$:
NEXT
2530 LOCATE 10,1:PRINT name$;CHR$(143);". ";
2540 a$="":WHILE a$="":a$=INKEY$:WEND
2550 IF a$=CHR$(13) THEN 2590
2560 IF a$=CHR$(127) AND 1<>0 THEN name$=LEFT$(name$,LEN(name$)-1):1=1-1:GOTO 2530
30
2570 IF 1<=8 THEN name$=name$a$+1:1=1:GOTO 2530
2580 GOTO 2530
2590 FOR x=0 TO 79:FOR y=0 TO 7:POKE &C000+x+y*2048,b(x,y):NEXT y,x
2600 IF INSTR(name$,".")=0 THEN name$=name$+".pic"
2610 RETURN

```

Der Tip

# Die Wanze im Computer

Diese Folge zeigt, wie man anderen Programmen eine 'Wanze' unterjubelt.

Macintosh- oder Atari 520-Käufer haben es gut: Sie können eine Hardcopy von jedem laufenden Programm machen, einfach so auf Knopfdruck. Und davon können Schneider-User nur träumen. Doch damit so etwas auch auf den CPCs geht, werden wir eine Wanze in den Computer schmuggeln. Sie soll vorher eingeladen werden und dann solange mit dem Benutzerprogramm im Speicher 'hocken', ohne es zu stören, bis sie durch irgendein Ereignis, z.B. einen Tastendruck, aktiviert wird. Dann soll sie die Kontrolle übernehmen.

## Screensave

In unserem konkreten Fall soll die Wanze ein Abbild des Bildschirms auf Kassette/ Diskette schreiben. Dieses Bild könnte dann z.B. von einem Hardcopy-Programm weiterverwendet werden. Aber jetzt tun sich einige Probleme auf: Wo soll das Programm stehen, wie soll es aktiviert werden? Für die Ort-Frage gibt es mehrere Lösungen: Man könnte überprüfen, in welche Adressen das Benutzerprogramm lädt, und dann die Wanze darüber oder darunter positionieren. Dann besteht aber die Gefahr, daß die Wanze durch Daten des Benutzerprogramms evtl. zerstört wird. Es gibt aber auch noch andere Möglichkeiten, das Programm 'verschwinden' zu lassen (siehe Listing). Bezüglich der Aktivierung ist man einerseits zwar völlig frei. Ob Taste, Joystick, nach einer Zeitspanne oder was auch immer: Man kann alles mögliche verwenden, um das Programm zu aktivieren. Andererseits sollte man aber die Verwendung von Firmware-Routinen, wie z.B. Tastenabfrage, vermeiden, da das Programm über Interrupt aufgerufen wird und die Firmware-Routinen ebenfalls unterbrechbar sind.

## Über Interrupt

Die Wanze wird in die Interrupt-Liste des Schneider-Betriebssystems eingebunden, und kann dann erst einmal vergessen werden. Sie fragt ab jetzt regelmäßig die Tastatur ab, und zwar direkt über den Port (Firmware-Routinen sollte man nicht unbedingt verwenden): KTEST erledigt das. Wenn nicht die entsprechende Tastenkombination (CTRL-COPY) gedrückt ist, passiert gar nichts, die Wanze gibt die Kontrolle wieder an das Betriebssystem zurück. Wenn aber die Wanze aktiviert wird, dann schlägt sie auch gleich wie mit dem Dampfhammer zu: Die Firmware-Indirections werden so geändert, daß bei deren nächstem Aufruf die SAVE-Routine angesprungen wird, auch hier können wir die SAVE-Routine nicht einfach sofort ausführen, da die Wanze ja noch in der Interruptkette abgearbeitet wird. Sobald das Anwender-Programm nun irgendeine der gepatchten Betriebssystem-Funktionen ausführen will, springt es stattdessen zu SAVE. Der Bildschirm wird abgespeichert und die alte Sprungtabelle wiederhergestellt. Sollte das Programm allerdings selbst etwas verändert haben, wird es jetzt nicht mehr korrekt laufen. Auch wird bei diesem Prozess automatisch auf Kassettenbetrieb umgeschaltet, da JUMP-RESTORE ja die Werte für die Kassettenverwaltung und nicht für das DOS einträgt. Dieses müßte wieder neu initialisiert werden, was aber nicht ganz einfach ist, da man ihm Speicher-

platz zuweisen muß. Keine einfache Aufgabe, weil die Wanze ja nicht weiß, wo das Anwenderprogramm steht. Also haben wir uns mit der einfacheren Lösung zufrieden gegeben. Ansonsten ist das Programm so simpel aufgebaut, daß man sich weitere Kommentare sparen kann. Probieren Sie es aus, und stellen Sie eigene *Wanzen-Kreationen* her. Eine Möglichkeit für die Auslösung wäre z.B. der Betriebssystem-TIME-Zähler oder eine Drucker- oder Floppy-Anmeldung.

## Nicht alle Programme

... sind unterbrechbar. Programme, die überhaupt keine Firmware-Indirections benutzen, zum Beispiel. Allerdings werden solche Programme wohl sehr selten auftreten, da ja auch schon die Standard-Ein/Ausgabe über die Firmware-Sprungtabelle läuft. Ebenfalls nicht unterbrechbar sind Programme, die direkt in die ROM-Routinen springen, ohne die Tabelle zu benutzen. Solchen hartnäckigen Fällen kann man nur bekommen, indem man das Programm selbst von der Wanze verändern läßt, aber das ist schon reichlich kompliziert. Denn dazu braucht man als Mindestausstattung schon einen Disassembler und entsprechende Maschinensprache-Kenntnisse. tmb

```

220 '
230 ' *** Hexlader ***
240 PRINT"PLEASE WAIT ..."
250 DEFSTR c
260 DEFINT i,j
270 DIM c(20)
280 c(0)="CD 97 A1 01 0C A0 21 4E A0 C3 D1 BC 20 A0 C3 44 A1 C3 4D A1"
290 c(1)="C3 56 A1 C3 6C A1 C3 7B A1 C3 8B A1 43 4C 4F 4B 4B 4F CE 43"
300 c(2)="4C 4F 43 4B 4F 46 C6 4B 4C 4F 43 4B 53 45 D4 43 4C 4F 43 4B"
310 c(3)="52 45 D3 42 45 4C 4C 5B 45 D4 42 45 4C 4C 4F 46 C6 00 61 66"
320 c(4)="00 0C 00 00 00 00 00 01 3A 52 A0 F5 3A 53 A0 47 F1 3C532 52"
330 c(5)="A0 FE 3C 20 27 3E 00 04 32 52 A0 78 32 53 A0 F5 3A 54 A0 47"
340 c(6)="F1 FE 3C 20 13 3E 00 04 32 53 A0 78 32 54 A0 FE 18 20 05 3E"
350 c(7)="00 32 54 A0 3A 57 A0 FE 00 C8 FE 02 28 5E 3A 57 A0 FE 03 CC"
360 c(8)="F9 A0 3A 86 B2 F5 3A 85 B2 F5 3A 8F B2 47 F5 3A 90 B2 F5 32"
370 c(9)="BF B2 78 32 90 B2 CD 7E BB 3E 0C 32 86 B2 3A 88 B2 32 85 B2"
380 c(10)="3A 54 A0 CD 10 A1 3E 3A CD 5A BB 3A 53 A0 CD 10 A1 3E 3A CD"
390 c(11)="5A BB 3A 52 A0 CD 10 A1 F1 32 90 B2 F1 32 8F B2 F1 32 85 B2"
400 c(12)="F1 32 86 B2 CD 7B BB C9 3A 54 A0 47 3A 56 A0 B8 C0 3A 53 A0"
410 c(13)="47 3A 55 A0 B8 C0 3E 07 CD 5A BB C9 5F F5 C5 06 00 FE 0A FA"
420 c(14)="1F A1 04 D6 DA 18 F6 F5 78 CD 2B A1 F1 CD 2B A1 C1 F1 C9 FE"
430 c(15)="0A 38 02 C6 07 C6 30 CD 5A BB C9 00 00 00 00 00 00 00 00"
440 c(16)="81 58 A0 00 3A 57 A0 F6 01 32 57 A0 C9 3A 57 A0 E6 FE 32 57"
450 c(17)="A0 C9 FE 03 C0 DD 7E 04 32 54 A0 DD 7E 02 32 53 A0 DD 7E 00"

```

In einem solchen Hex-Listing kann die Wanze versteckt sein.

```

Pass 1 errors: 00
10 ; diese Version unterscheidet sich in zwei Punkten von der in
20 ; Text beschriebenen Version:
30 ;
40 ; es wird eine Betriebssystem-Routine zur Tastaturabfrage
50 ; verwendet, das ist bequemer wenn auch unsaubere Program-
60 ; mierung
70 ;
80 ; die Diskette wird doch neu initialisiert, nachdem die Sprung
90 ; Tabelle neu hergestellt wurde (siehe SAVE)
100 ;
110 ; die Schneider-Wanze
120 ; speichert Bildschirm bei CTRL-COPY
130 ;
A000 140 org $a000 ;irgendwo, init wird nur einmal gebraucht
150 ;
A000 21A7BC 160 init: ld hl,$bc97 ;Adresse SOUND_RESET
A003 0A6B 170 ld hl,$1 ;ll soundroutinen durch $c1
A005 3AC9 180 li: ld (hl),$c1 ;RET, ist alles nur moelig,
A007 23 190 inc hl ;wenn Program in Soundpuffer
A008 23 200 inc hl ;steht, um zu verhindern
A009 23 210 inc hl ;das es ueberschrieben wird
A00A 10F9 220 djnz $1 ;jetzt sind alle sound-routin ka
A00C 21A8A 230 ld hl,$block ;jetzt in Interrupt-Kette
A00F EDE3BC 240 call $bc93 ;einbinden: KL_ADD_FAST_TICKER
A012 C9 250 ret ; Wanze hat sich eingenistet....
260 ;
B61A 270 org $b61a
280 ; Wanze sitzt in Soundpuffer (464!!!)
290 ; Jede Adresse ist moelig, wenn nie das
300 ; Anwenderprogramm nicht ueberlappt
310 ;
B61A 0000 320 block: defw 0 ;Interruptblock
B61C 0000 330 defw 0 ;fuor das Betriebssystem
B61E 00 340 defb 0
B61F 81 350 defb $B1
B620 25B6 360 defw $test ;adresse interrupt_routi
B622 00 370 defb 0
B623 0000 380 defw 0
390 ;
B625 00 400 $test: nop ;an diese Stelle kommt ein RET
B626 F5 410 push af ;save register...
B627 C5 420 push bc
B628 3E09 430 ld a,$9 ;copy taste
B62A CD1E8B 440 call $bb1e ;hw_test_key
B62D 2004 450 jr nz,$found
B62F C1 460 ke: pop bc
B630 F1 470 pop af
B631 FB 480 ei
B632 C9 490 ret
B633 F3 500 found: di ;zur Sicherheit
B634 C879 510 bit 7,c ;CTRL gedrueckt?
B636 28F7 520 jr z,$ke ;kein RET
B638 3EC9 530 ld a,$c9 ;RET
B63A 3225B6 540 ld ($test),a ;nachmaligen Aufruf verhindern
B63D 06BC 550 ld b,$B8 ;alle Betriebssystemspruege
B63F 0D2100B 560 ld ix,$b600
B643 215CB6 570 ld hl,$ave
B644 0B3400C3 580 li: ld (ix),$c3 ;JP
B646 0D7501 590 ld (ix+1),1
B648 0D7402 600 ld (ix+2),h
B650 8023 610 inc ix
B652 8023 620 inc ix
B654 8023 630 inc ix
B656 10EE 640 djnz $2
B658 C1 650 pop bc
B659 F1 660 pop af
B65A FB 670 ei
B65C C9 680 ret ;zurueck zum Anwenderpr
B65D 490 ;
B65E CB37B0 700 save: call $bc37 ;jump restore
B65F 114000 710 ld de,$40 ;rebase
B662 210C8C 720 ld hl,$c000 ;grantos
B665 C81BC 730 call $bc8c ;KL_ROM_WALK, init Disc
B668 0641 740 ld b,$4 ;laenge dat_name
B66A 2189A 750 ld hl,$ave
B66D 110C00 760 ld de,0 ;kein buffer moelig
B670 EBCBC 770 call $bc8c ;cas_out_open
B673 2100C0 780 ld hl,$c000 ;screen
B676 11FF3F 790 ld de,$3fff ;16k
B679 010000 800 ld bc,0 ;kein start
B67C 3E02 810 ld a,$2 ;binaer
B67E CD9BBC 820 call $bc9B ;cas_out_direct
B681 CDBFBC 830 call $bc8f ;cas_out_close
B684 AF 840 xor a ;a=0
B685 3225B6 850 ld ($test),a ;erneutes aufrufen moelig
B688 C9 860 ret ;zurueck zum prograe
870 ;
B689 4249C44 880 name: defw "BILD"
B68D 00 890 zzz: nop
Pass 2 errors: 00
Table used: 119 from 365

```

# Label Drucker

**Kennen Sie den 'WIMP-Satz'? Dieser Satz (Wo-Ist-Mein-Programm) besagt, daß ein gesuchtes Programm immer auf der letzten Diskette ist, auf der man nachsieht, gleichgültig, in welcher Reihenfolge man die einzelnen Disketten überprüft.**

Das hat teilweise zwar auch ganz angenehme Begleiterscheinungen, z.B. daß man immer wieder Programme auf seinen Disketten entdeckt, die man schon total vergessen hatte, kann aber auf die Dauer doch auf die Nerven gehen.

Für alle CPC 464/664 - User haben wir deshalb eine Lösung entwickelt: Den Label-Drucker, mit

dem man den Disketteninhalt, den Namen der Diskette und ihre Nummer zu Papier bringt, und zwar gleich in der richtigen Größe zum 'Hinter-die-Hülle-klemmen'.

Das Programm wurde für einen Star-SG 10 Drucker geschrieben, sollte daher auch auf den neueren Epson-Druckern laufen. Es dürfte

aber auch nicht schwer sein, mit Hilfe der Kommentare im Programm und dem eigenen Drucker-Handbuch die entsprechenden Programmzeilen an den eigenen Drucker anzupassen. Nun, zur Bedienung gibt es auch nicht viel zu sagen, deshalb: Viel Spaß beim Ausschneiden!

```

10 REM *****
20 REM **      Disketten Labler      **
30 REM ** von Thomas M.Binzinger **
40 REM *****
50 REM
60 INK'0,13:INK 1,0:BORDER 13:MODE 2
70 OPENOUT "d":MEMORY HIMEM-1:ba=HIMEM+2048:CLOSEO
UT
80 DIM s$(65),sa$(65),sb$(65)
90 PRINT"Label Drucker von Thomas M.Binzinger, 198
6 by COMPUTER TEAM!":PRINT
100 INPUT "Bitte legen Sie die Diskette mit Seite
A ein, druecken Sie ENTER ",n$
110 GOSUB 450 'Inhalt in s$(64)
120 x=1:WHILE s$(x)<>" " AND x<>64:sa$(x)=s$(x):x=x
+1:WEND
130 FOR y=x TO 64:sa$(y)=SPACES(12):NEXT
140 INPUT "Bitte legen Sie die Diskette mit Seite
B ein, druecken Sie ENTER ",n$
150 GOSUB 450
160 x=1:WHILE s$(x)<>" " AND x<>64:sb$(x)=s$(x):x=x
+1:WEND
170 FOR y=x TO 64:sb$(y)=SPACES(12):NEXT
180 CLS #1:PRINT"Diskette komplett gelesen."
190 INPUT "Bitte geben Sie nun eine zwei-stellige
Diskettennummer ein: ",dn$
200 IF LEN(dn$)<>2 THEN PRINT"Bitte ZWEI Stellen!"
:GOTO 190
210 PRINT "Bitte geben Sie den Diskettentitel ein
:"STRING$(10,".")STRING$(10,B);
220 INPUT "",dts$:IF LEN(dts$)>10 THEN PRINT"Zu lang
!":GOTO 210

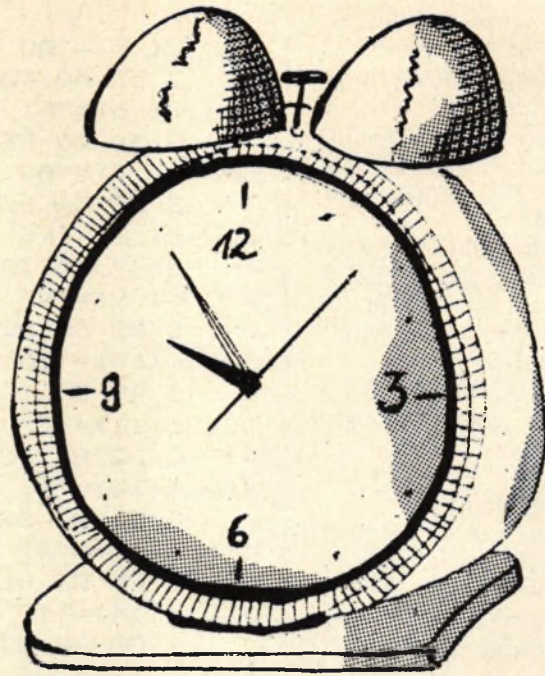
```

## CPM-Master 00

Seite A:	Seite B:
.MEY	.PRN
AMSDDS.COM	ASH.COM
BOOTGEN.COM	CHKDISC.COM
CLOAD.COM	COPYDISC.COM
CSAVE.COM	DDT.COM
DISCHK.COM	DISCCOPY.COM
DUMP.ASH	DUMP.COM
ED.COM	EX1.BAS
EX2.BAS	FILECOPY.COM
FORMAT.COM	GEBE.TXT
LOAD.COM	MOVCPH.COM
PIP.COM	ROINTIME.DEM
SETUP.COM	STAT.COM
SUBMIT.COM	SYSGEN.COM
XSUB.COM	
	AMSDDS.COM LOGO.COM
	OSPIRALE.LOG SETUP.COM
	STERN.LOG

```
230 INPUT "Bitte druecken Sie ENTER wenn der Druck
er bereit ist. ",n$
240 WIDTH 255
250 PRINT#8,CHR$(27)"A"CHR$(8)CHR$(27)"2"; 'Line s
pacing auf 8
260 PRINT #8,CHR$(27)"7"; 'fuer STAR SG-10 in IBM-
Mode
270 PRINT #8,CHR$(15); 'Komprimierte Schrift ein
280 r$=CHR$(124) 'senkrechter Strich
290 leer$=r$+STRING$(53,32)+r$
300 IF LEN(dt$)<10 THEN dt$=dt$+STRING$(10-LEN(dt$
)," ")
310 PRINT#8,CHR$(27)"G"; 'Doppeldruck ein
320 exa$=CHR$(18)+CHR$(27)+"W1" 'Breitschrift-ein
Code
330 exo$=CHR$(27)+"W0"+CHR$(15) 'Breitschrift-aus
Code
340 PRINT#8,r$;STRING$(53,"-");r$
350 PRINT #8,r$;" ";exa$;dt$;" ";dn$;exo$;" ";
r$
360 PRINT#8,r$;STRING$(53,"-");r$
370 PRINT#8,r$;"Seite A:"STRING$(18," ");r$;" Seit
e B:";STRING$(17," ")r$
380 IF sa$(61)=SPACE$(12) AND sb$(61)=SPACE$(12) T
HEN lzahl=62:PRINT #8,r$;STRING$(26,32);r$;STRING$
(26,32);r$ ELSE lzahl=64
390 FOR x=1 TO lzahl STEP 2
400 PRINT#8,r$;sa$(x)" "sa$(x+1)" "r$" "sb$(x)" "s
b$(x+1);r$
410 NEXT x
420 PRINT#8,r$;STRING$(53,"-");r$
430 CLS:GOTO 90
440 END
450 WINDOW #1,1,80,10,25
460 WINDOW SWAP 0,1:CLS:CAT
470 c=1000
480 ad=ba:WHILE c<>0 AND (PEEK(ad)<32 OR PEEK(ad)>
90):ad=ad+1:c=c-1:WEND
490 IF c=0 THEN 610
500 p=ad:z=1:WHILE PEEK(p)<>0
510 s$(z)="":x=p
520 FOR x=p TO p+10
530 s$(z)=s$(z)+CHR$(PEEK(x)):POKE x,0:NEXT
540 p=p+14
550 s$(z)=LEFT$(s$(z),8)+"."+RIGHT$(s$(z),3)
560 a=INSTR(s$(z)," "):IF a=0 THEN 600
570 IF a>INSTR(s$(z),".") THEN 600
580 s$(z)=LEFT$(s$(z),a-1)+RIGHT$(s$(z),LEN(s$(z))-a)+" "
590 GOTO 560
600 z=z+1:WEND
610 WINDOW SWAP 0,1
620 s$(z)="":RETURN
```

# Die Uhr des CPC



Ist es Ihnen auch schon einmal passiert, daß Sie am Computer gearbeitet und darüber die Zeit vergessen haben? Mit Hilfe dieses Programms können Sie nun den Rechner jederzeit nach der Uhrzeit fragen oder sich auf wichtige Termine aufmerksam machen lassen.

Nach dem Start dieses Programms stehen Ihnen die folgenden sechs zusätzlichen Befehle zur Verfügung:

:CLOCKSET, (Stunden),(Minuten),(Sekunden)

Die implementierte Uhr wird eingestellt.

:CLOCKON

Die Uhr wird am oberen Bildschirmrand angezeigt.

:CLOCKOFF

Die Uhr wird nicht angezeigt, läuft jedoch weiter.

:CLOCKRES

Die Uhr wird nicht mehr ausgerufen und kann auch durch die entsprechenden Befehle nicht mehr aktiviert werden.

:BELLSET, (Stunden),(Minuten)

Der Wecker wird eingestellt und aktiviert.

:BelloFF

Der Wecker wird ausgeschaltet.

Dabei sind einige Dinge zu beachten:

Der Doppelpunkt vor dem Befehl stellt das Zeichen über dem „Klammeraffen“ dar.

Werden bei einem Befehlsaufruf zu viele oder zu wenige Parameter angegeben, so wird dieser nicht ausgeführt.

Das Programm sollte nach dem Abtippen zunächst aufgezeichnet

```

10 MEMORY &9FFF:GOSUB 240:CALL &A000
20 :CLOCKOFF
30 MODE 2
40 PRINT
50 PRINT TAB(22);CHR$(24);" Digital Background
   Clock ";CHR$(24)
60 PRINT
70 PRINTTAB(30);"by THBCS"
80 PRINT:PRINT
90 INPUT"Time (HH:MM:SS)? ",t$
100 dopp1=INSTR(t$,":"):dopp2=INSTR(dopp1+1,t$,"
   :")
110 IF dopp1*dopp2=0 THEN 90
120 std=VAL(LEFT$(t$,dopp1-1))
130 IF std>23 OR std<0 THEN 90
140 minuten=VAL(MID$(t$,dopp1+1,dopp2-dopp1-1))
150 IF minuten>59 OR minuten<0 THEN 90
160 sec=VAL(RIGHT$(t$,LEN(t$)-dopp2))
170 IF sec>59 OR sec<0 THEN 90
180 :CLOCKSET,std,minuten,sec
190 :CLOCKON
200 CLS
210 NEW
220 '
230 '
240 PRINT"PLEASE WAIT ..."
250 DEFSTR c
260 DEFINT i,j
270 DIM c(20)
280 c(0)="CD 97 A1 01 0C A0 21 4E A0 C3 D1 BC 2
   0 A0 C3 44 A1 C3 4D A1"
290 c(1)="C3 56 A1 C3 6C A1 C3 73 A1 C3 8B A1 4
   3 4C 4F 43 4B 4F CE 43"
300 c(2)="4C 4F 43 4B 4F 46 C6 43 4C 4F 43 4B 5
   3 45 D4 43 4C 4F 43 4B"
310 c(3)="52 45 D3 42 45 4C 4C 53 45 D4 42 45 4
   C 4C 4F 46 C6 00 61 66"
320 c(4)="00 0C 00 00 00 00 00 01 3A 52 A0 F5 3
   A 53 A0 47 F1 3C 32 52"

```

\*\*\* Hexlader \*\*\*

werden, um bei einem möglichen Absturz des Rechners einen Verlust desselben (des Programms natürlich) zu vermeiden.

Der Befehl NEW sollte erst dann in das Programm eingefügt werden, wenn dieses vollkommen korrekt abgetippt ist.

Wenn die eingestellte Weckzeit erreicht ist, dann klingelt der Rechner für die Dauer einer Minute, wenn nicht vorher der Befehl BELLOF eingegeben wird.

Der Befehl :CLOCKRES sollte dann angewandt werden, wenn man den Speicherbereich ab &A000 ändern will, da sonst der Rechner abstürzen kann.

Die BASIC-Funktion TIME wird durch die Uhr nicht beeinflusst. Jedoch werden beide Uhren für die Dauer von Cassettenoperationen angehalten, da diese Vorrang haben. tb



```

330 c( 5) ="A0 FE 3C 20 27 3E 00 04 32 52 A0 78 3
      2 53 A0 F5 3A 54 A0 47"
340 c( 6) ="F1 FE 3C 20 13 3E 00 04 32 53 A0 78 3
      2 54 A0 FE 18 20 05 3E"
350 c( 7) ="00 32 54 A0 3A 57 A0 FE 00 C8 FE 02 2
      8 5E 3A 57 A0 FE 03 CC"
360 c( 8) ="F8 A0 3A 86 B2 F5 3A 85 B2 F5 3A 8F B
      2 47 F5 3A 90 B2 F5 32"
370 c( 9) ="8F B2 78 32 90 B2 CD 7E BB 3E 0C 32 8
      6 B2 3A 88 B2 32 85 B2"
380 c(10) ="3A 54 A0 CD 10 A1 3E 3A CD 5A BB 3A 5
      3 A0 CD 10 A1 3E 3A CD"
390 c(11) ="5A BB 3A 52 A0 CD 10 A1 F1 32 90 B2 F
      1 32 8F B2 F1 32 85 B2"
400 c(12) ="F1 32 86 B2 CD 7B BB C9 3A 54 A0 47 3
      A 56 A0 B8 C0 3A 53 A0"
410 c(13) ="47 3A 55 A0 B8 C0 3E 07 CD 5A BB C9 5
      F F5 C5 06 00 FE 0A FA"
420 c(14) ="1F A1 04 D6 0A 18 F6 F5 78 CD 2B A1 F
      1 CD 2B A1 C1 F1 C9 FE"
430 c(15) ="0A 38 02 C6 07 C6 30 CD 5A BB C9 00 0
      0 00 00 00 00 00 00"

440 c(16) ="81 58 A0 00 3A 57 A0 F6 01 32 57 A0 C
      9 3A 57 A0 E6 FE 32 57"
450 c(17) ="A0 C9 FE 03 C0 DD 7E 04 32 54 A0 DD 7
      E 02 32 53 A0 DD 7E 00"
460 c(18) ="32 52 A0 C9 21 37 A1 CD EC BC C9 FE 0
      2 C0 DD 7E 02 32 56 A0"
470 c(19) ="DD 7E 00 32 55 A0 3A 57 A0 F6 02 32 5
      7 A0 C9 FE 00 C0 3A 57"
480 c(20) ="A0 E6 01 32 57 A0 C9 21 37 A1 11 32 0
      0 01 32 00 CD E9 BC C9"

490 .
500 DATA 2602,2588,1736,1846,1363,1934,1756,2209
      ,3007,2500,2294
510 DATA 2786,2673,2559,3003,1202,2353,2444,2665
      ,2284,2083

520 .
530 RESTORE 500
550 adr=&A000
560 FOR i=0 TO 20
570   c(i)="00"+c(i)
580   sum=0:READ pruefsumme
590   FOR j=1 TO 20
600     wert=VAL("&"+MID$(c(i),3*j,2))
610     POKE adr,wert
620     adr=adr+1
630     sum=sum+wert
640   NEXT j
645   IF sum<>pruefsumme THEN PRINT"Fehler in c(
      ";i;")":END
660 NEXT i
680 RETURN

```



# Maussteuerung mit dem CPC 464

**Diese Befehlsenerweiterung für den Schneider CPC 464 ermöglicht die Simulation einer Maus. Da die Maus für die Schneidercomputer zur Zeit in Deutschland noch nicht erhältlich ist, wurde der Joystick als Ersatz gewählt.**

Wenn Sie Listing 1 fehlerfrei abgetippt und gestartet haben, wird auf Cassette oder Diskette die Datei MAUS.BIN erzeugt. Dieses Programm können Sie mit MEMORY & 9FFF LOAD "MAUS.BIN" CALL & A000 starten.

Ab diesem Zeitpunkt steht Ihnen der Befehl :MAUS,x,y zur Verfügung. Der Doppelpunkt vor dem Befehl stellt in diesem Fall das Zeichen über dem Klammeraffen dar.

Nach dem Aufruf des Befehls wird auf dem Bildschirm, an dem durch x und y definierten Punkt, ein invertierender Pfeil dargestellt. Diesen können Sie mit dem Joystick beliebig über den Bildschirm bewegen. Dadurch wird automatisch auch der Grafikkursor, welcher sich an der Spitze des Pfeils befindet, bewegt.

Nach dem Drücken des Feuerknopfes wird die Kontrolle wieder an das Programm übergeben; bzw. der Computer meldet sich im Direktmodus eingeben haben. Die Koordinaten des Grafikkursors stehen dann in XPOS und YPOS zur Verfügung.

Da der Pfeil sich invertierend über den Monitor bewegt, wird das Bild durch die Bewegungen nicht zerstört. Man kann diese Erweiterung zum Beispiel einsetzen, um anhand der Grafikkursorposition festzustellen, auf welchen Punkt eines Menues der Pfeil zeigt, wodurch der Computer sich dann durch Laien leichter bedienen ließe. Der neue Befehl ließe sich auch in einem komfortablen Malprogramm einsetzen; ähnlich wie beim Apple Macintosh. (tb)

```

10 'Listing 1
20 '
30 'Hexlader fuer Maussteuerbefehl
40 '(nur fuer CPC 464 !!!)
50 '
60 '(c) by THBCS
70 MEMORY &9FFF
80 PRINT"PLEASE WAIT ..."
90 DEFSTR c
100 DEFINT i,j
110 DIM c(11)
120 c( 0)="01 09 A0 21 13 A0 C3 D1 BC OE A0 C3 1
7 A0 4D 41 55 D3 00 00"
130 DATA 1964
140 c( 1)="00 00 00 FE 02 C0 DD 56 03 DD 5E 02 D
D 66 01 DD 6E 00 CD 63"
150 DATA 2034
160 c( 2)="A0 D5 E5 CD 18 BB CD 24 BB E1 D1 CD 6
3 A0 F5 E6 01 FE 01 20"
170 DATA 3107
180 c( 3)="02 23 23 F1 F5 E6 02 FE 02 20 02 2B 2
B F1 F5 E6 04 FE 04 20"
190 DATA 2176
200 c( 4)="01 1B F1 F5 E6 08 FE 08 20 01 13 F1 E
6 10 FE 10 C8 18 C3 F5"
210 DATA 2487
220 c( 5)="D5 E5 CD D6 A0 13 CD D6 A0 13 CD D6 A
0 13 CD D6 A0 13 CD D6"
230 DATA 3253
240 c( 6)="A0 13 CD D6 A0 13 CD D6 A0 2B 2B 1B 1
B 1B CD D6 A0 1B CD D6"
250 DATA 2548
260 c( 7)="A0 1B CD D6 A0 1B CD D6 A0 2B 2B CD D
6 A0 13 CD D6 A0 13 13"
270 DATA 2673
280 c( 8)="CD D6 A0 13 CD D6 A0 2B 2B CD D6 A0 1
3 CD D6 A0 1B 1B 1B"
290 DATA 2548
300 c( 9)="CD D6 A0 1B CD D6 A0 2B 2B 13 13 13 1
3 13 CD D6 A0 13 CD D6"
310 DATA 2383
320 c(10)="A0 2B 2B CD D6 A0 13 CD D6 A0 E1 D1 F
1 C9 CD C0 BB CD F0 BB"
330 DATA 3515
340 c(11)="2F CD DE BB CD C6 BB CD EA BB CD C6 B
B C9 00 00 00 00 00"
350 DATA 2668
360 '
370 RESTORE
380 adr=&A000
390 FOR i=0 TO 11
400   c(i)="00"+c(i)
410   sum=0:READ pruefsumme
420   FOR j=1 TO 20
430     wert=VAL("&"+MID$(c(i),3*j,2))
440     POKE adr,wert
450     adr=adr+1
460     sum=sum+wert
470   NEXT j
480   IF sum<>pruefsumme THEN PRINT"Fehler in c(
";i;)"":END
490 NEXT i
500 SAVE"maus.bin",b,&A000,&EA,&A000
510 PRINT"MAUS.BIN ist abgespeichert"

```

# Atari-Maus am Schneider CPC



Im Gegensatz zu vielen anderen Computern liefern die Schneiderrechner am Pin 7 des Joystickports keine 5-Volt Spannung. Dies ist jedoch mit einer simplen Blockbatterie sehr einfach und preiswert zu beheben.

Man verbindet einfach die Anschlüsse einer männlichen und einer weiblichen Joystickbuchse parallel miteinander. Mit Ausnahme des Pins 7. Dieser wird an der Joystick- (also der männlichen) Buchse mit dem Pluspol der Batterie verbunden. Den Minuspol der Batterie verbindet man mit Pin 8 einer der beiden Buchsen.

Wenn man dieses selbstgebaute Adapter zwischen Joystick und Rechner schaltet, funktioniert nun das Dauerfeuer der Quickshot II-Joysticks. Da diese oft sehr billig angeboten werden, nehme ich an, daß sie auch oft benutzt werden.

## Atari-Maus am Schneider

Wenn man nun die Pinbelegung der Atari ST-Maus betrachtet, fällt auf, daß auch diese eine 5-Volt Spannung erwartet. Leider ist die Abfrage in BASIC, mit JOY(0), etwas zu langsam, um eine realistische Maussteuerung durchzuführen. In Maschinensprache sollte dies jedoch möglich sein. (tb)

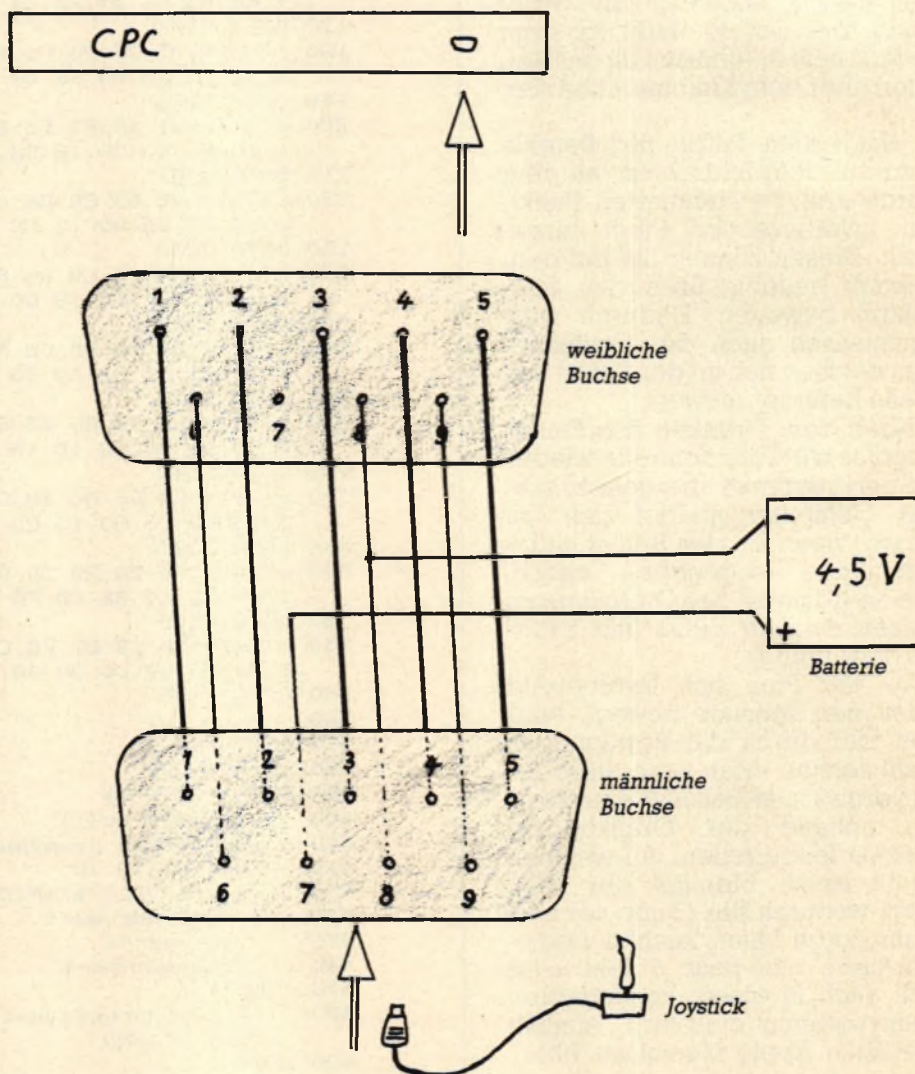
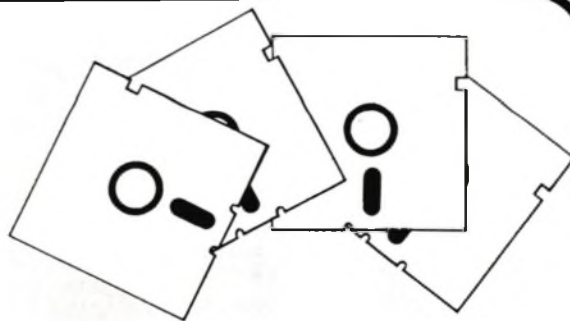


Bild 1: Verdrahtung der Buchsen

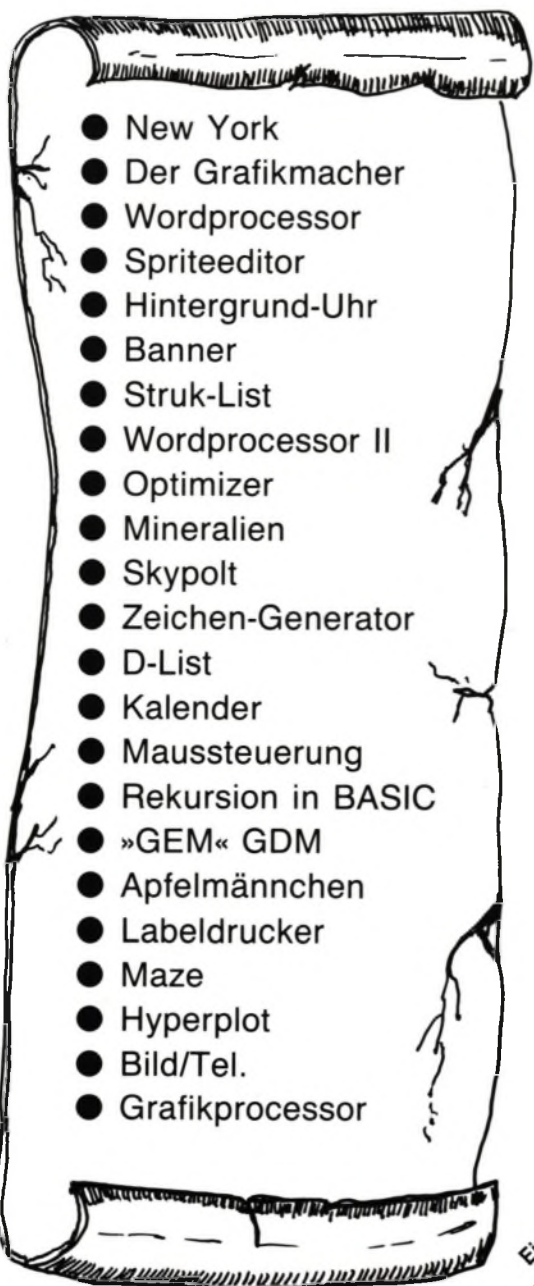
# Die Software



## zum Schneider Sonderheft

Folgende Listings finden Sie in unserem Software-Service. Sparen Sie sich das eintippen.

### Die Themen:



Über  
200 Kilobyte

für nur  
49.— Diskette (3"-Disk)  
45.— Cassette

23 Superprogramme

für nur  
49.— Diskette (3"-Disk)  
45.— Cassette

**Sofort bestellen**

Einsenden an: **SOFTWARE TEAM, Joachim Günster, Mühlenstraße 12, 5431 Boden**

**Ja, Ihr Angebot hat mich überzeugt und ich bestelle:**

Softwarepaket I (Disk) 49.—/Stk.     Softwarepaket I (Cassette) 45.—/Stk.

Name: \_\_\_\_\_ Vorname: \_\_\_\_\_  
PLZ/Ort: \_\_\_\_\_ Straße: \_\_\_\_\_

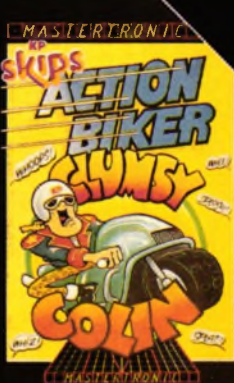
Versandwunsch bitte angeben:  
 Bargeld liegt bei     per Nachnahme  
 Verrechnungsscheck beigefügt  
Bei Versand per NN werden DM 5.—  
Versandkosten pauschal  
erhoben.

# MASTERTRONIC

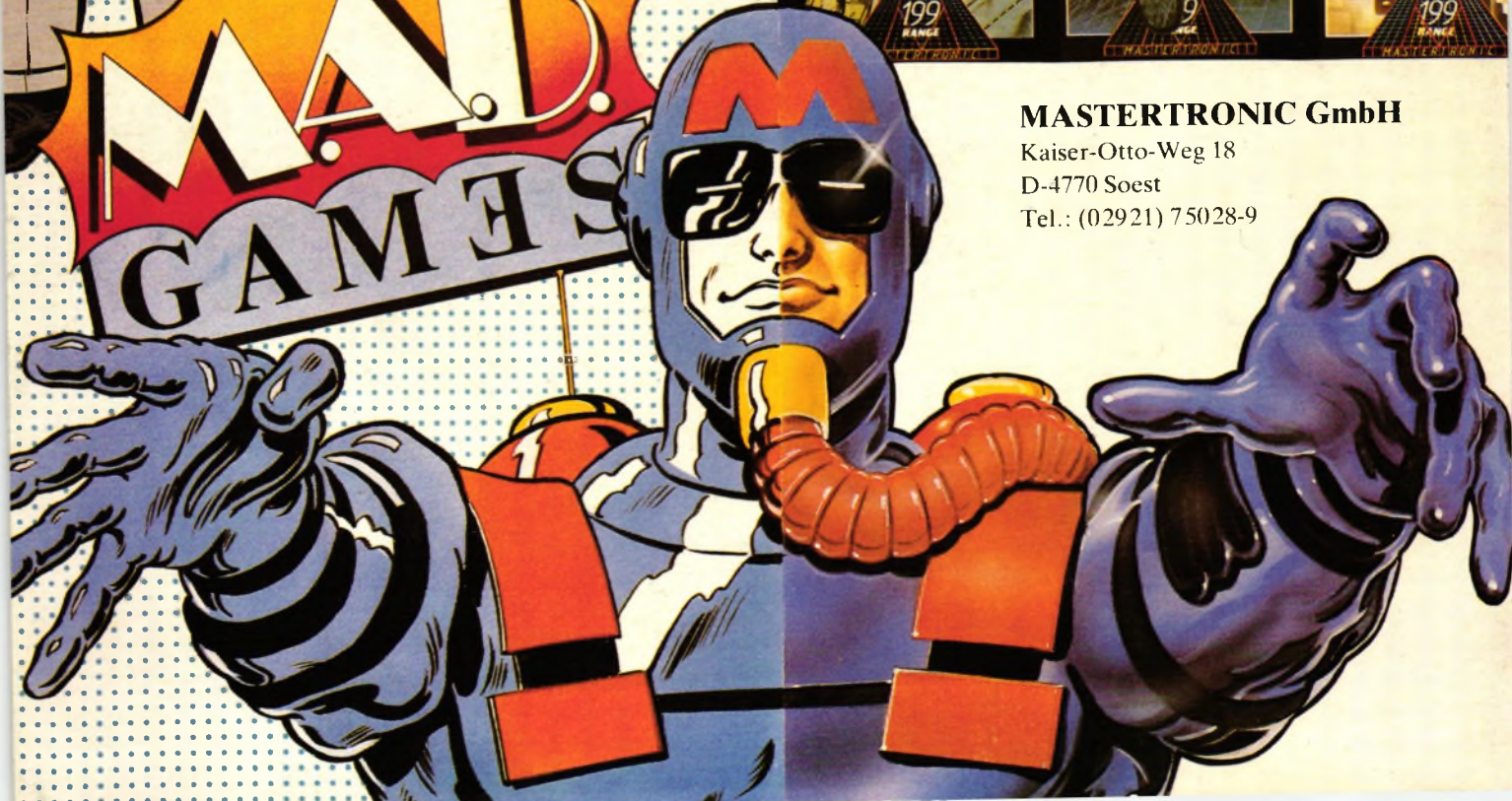
NUR VON  
MASTERTRONIC  
BEKOMMT MAN SOLCHE  
SUPER-SPIELE FÜR  
SO WENIG GELD

ODER?

## MAD GAMES



## MAD GAMES



**MASTERTRONIC GmbH**  
Kaiser-Otto-Weg 18  
D-4770 Soest  
Tel.: (02921) 75028-9