

Herausgeber
Fischel GmbH
Kaiser-Friedrich-Str. 54a
1000 Berlin 12

Berlin 1987

Alle Rechte vorbehalten. Ohne die ausdrückliche Genehmigung des Herausgebers ist es nicht gestattet, das Buch oder Teile daraus auf fotomechanischem (Foto-/Mikrokopie) oder sonstigem Wege zu vervielfältigen.

Für etwaige Schäden durch Anwendung der Anleitungen oder Programme dieses Buches übernehmen wir keine Haftung.

V o r w o r t

Dies Anwendungshandbuch möchte dem PC-1600-Einsteiger und dem PC-1500-Umsteiger helfen, die Kompakteinheit PC-1600, CE-1600P, CE-1600F und CE-1600M optimal zu nutzen.

Anfragen zum Einsatz des PC-1600 haben mich veranlaßt, den Schwerpunkt auf den Gebrauch der Diskettenstation zu setzen. Dafür habe ich den ursprünglichen Plan, Anfänger über die Vorteile strukturierten Programmierens zu unterrichten, fallen lassen. Es gibt ohnehin eine Fülle von Lehrbüchern für diese Problematik.

Mit Rücksicht auf den Umfang des Anwendungshandbuches habe ich auch darauf verzichtet, alle Befehle des PC-1600-ROM zu erläutern oder auf die Nutzung des PC-1600 als Taschenrechner einzugehen. Hier verweise ich auf die ausführliche Bedienungsanleitung zum PC-1600.

Behandelt wurden nur BASIC-Befehle - dann, wenn sie für eine bessere Nutzung weitere Erklärungen verdienten. Hier bitte ich die Computer-Freaks zu berücksichtigen, daß auf Grund von Anfragen auch Probleme behandelt wurden, deren Lösung für geübte Programmierer keine Schwierigkeiten bereiten.

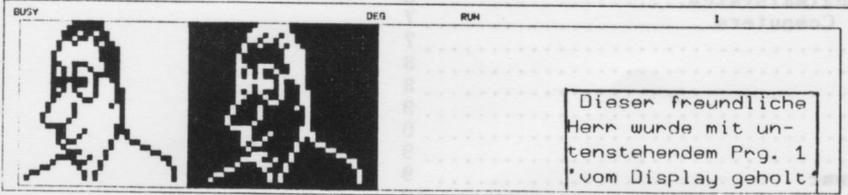
Der Zielsetzung des Buches entsprechend wurde darauf verzichtet, auf andere Peripheriegeräte oder die Maschinensprache und Maschinenprogramme näher einzugehen - ausgenommen besonderer Hinweise für PC-1500-Umsteiger.

Das Anwendungshandbuch wurde ausschließlich mit SHARP-Geräten hergestellt: Listings und Zeichnungen wurden mit dem PC-1600-System erstellt. Der Buchtext wurde mit dem PC-5000 verarbeitet und geschrieben.

Peter Kriz

Winand Klar Software
 Roßkampstraße 30
 D-3000 Hannover 81

PC-1600 Handcopy

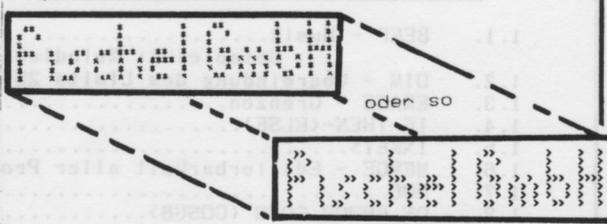


Mit folgenden kleinen Programmen kann man den Displayinhalt des PC-1600 auf dem CE-1600P kopieren.
 Das erste Programm macht dies mit Hilfe von kleinen Quadraten. Beim zweiten kann der Anwender A\$ mit beliebigen Zeichen des ASCII belegen, mit denen dann das Display dargestellt wird.
 Bei Programm 1 wird zusätzlich noch ein 'naturgetreuer' Rahmen gezeichnet.

```

1: ' Hardcopy für PC-1600 und CE-1600P
2: '
3: 'Written and Copyright 4 - 1987 by
4: 'Winand Klar Software
5: '
10:CLS :WAIT 0:FOR S=0TO 1
20:GCURSOR S*35+6,0:GPRINT "C0300C0E170B0F071BEFFCF0
    E0
30:GCURSOR S*35+6,8:GPRINT "FF593E083F5999827C04070F
    FFE0
40:GCURSOR S*35+4,16:GPRINT "0C1310F0708887402000000
    00081703F608080
50:GCURSOR S*35+1,24:GPRINT "4020201010080405060E1A6
    80460180601000000000001010202040C70
60:NEXT S
70:FOR T=32TO 67:LINE (T,0)-(T,32),X:NEXT T
95:'>> Zeile 10 - 90: Displayinhalt <<
100:GRAPH :GLCURSOR (50,-10):SURGN :LLINE (0,5)-(790,
    -155),,B:LLINE (-15,25)-(795,-160),,B:CSIZE 1
110:GLCURSOR (-5,10):LPRINT "BUSY":GLCURSOR (330,10):
    LPRINT "DEG":GLCURSOR (420,10):LPRINT "RUN":
    GLCURSOR (660,10):LPRINT "I"
120:FOR X=0TO 155:FOR Y=0TO 31
130:IF POINT (X,Y)=1GOSUB 150ELSE GLCURSOR (X*5,Y*-5)
140:NEXT Y:NEXT X:END
150:FOR Z=1TO 5:LLINE (X*5+Z,Y*-5)-(X*5+Z,Y*-5+5):
    NEXT Z:RETURN
    
```

... und so sieht das mit Textzeichen kopierte Display (anderer Inhalt als bei No 1) aus:



```

100:'Text-Hardcopy für PC-1600 und CE-1600P
110:'
120:"H"A$=CHR$ &DB:TEXT :CSIZE 1:PITCH 6,4:PCONSOLE
    LPT1:",155,2,0
130:FOR Y=0TO 31:FOR X=0TO 155
140:IF POINT (X,Y)=1LPRINT A$;ELSE LPRINT " ";
150:NEXT X:LPRINT " ":NEXT Y
    
```

PC-1600

**DER FORTSCHRITTLICHE
 TASCHEN-COMPUTER**

Inhaltsverzeichnis

Seite

I. Grundsätzliches über den PC-1600

1.	Ein Computer kann nicht bis drei zählen.....	4
2.	Das "Stellwerk" des PC-1600.....	4
2.1.	Dual(Binär)- und Hexadezimalsystem.....	6
2.2.	Die Arbeitsweise eines Computers.....	7
2.3.	Der Mikroprozessor.....	7
2.4.	Der BASIC-Interpreter.....	8
2.5.	Kompatibilität.....	8
2.6.	Der Arbeitsspeicher.....	9
2.7.	Der ASCII-Code.....	9
2.8.	Bit-Parität.....	9
2.9.	Bearbeitung der Programme.....	9
3.	Der PC-1600 kann mehr.....	10

II. Nutzung der PC-1500-Optionen

1.	ERROR 110.....	11
2.	Der Nebenprozessor LH 5803.....	11
3.	Der für den Mode 1 verfügbare Speicher.....	11
4.	Maschinenprogramme des PC-1500.....	13
5.	PC-1600+CE-150.....	14
6.	PC-1600 im PC-1500-Modus mit RAM-Disk.....	14
7.	PC-1600+CE-150/1600P+CE-158+CE-515/516P.....	15
8.	Verknüpfung von BASIC-Befehlen.....	16

III. Optimale Nutzung einiger BASIC-Befehle

1.1.	BEEP - Musik.....	16
	Beispiel 1: Melodien.....	18
1.2.	DIM - Überwindung des Limits 255.....	18
1.3.	ERASE - Grenzen.....	18
1.4.	IF-THEN-(ELSE).....	19
1.5.	INKEY\$.....	19
1.6.	MERGE - Editierbarkeit aller Programme.....	20
1.7.	MOD.....	20
1.8.	ON ERROR GOTO (GOSUB).....	20
1.9.	RENUM als Fehlerkontrolle.....	20
1.10.	WAIT 0,S.....	21
2.	Reservetastenbelegung.....	21

IV. Nutzung der LCD-Matrix

1.	Display-Auflösung.....	23
2.	Bitmuster-Schraffur.....	24
3.	Figuren und Figuren in Bewegung.....	24
3.1.	Figuren.....	24
3.2.	sich bewegende Figuren.....	25
3.3.	handgesteuerte Bewegungen.....	26
3.4.	automatische Computergrafiken.....	26
4.	Displaykopierung - Hardcopy.....	27

Beispiel 1: Display-Auflösung.....	28
Beispiel 2: Bitmuster-Schraffur.....	29
Beispiel 3: Displaykopierung.....	29
Beispiel 4: Derby mit Totalisator.....	30
Beispiel 5: Manugraf.....	32
Beispiel 6: automatische Computergrafik.....	32

V. Nutzung des CE-1600P (Drucker)

1.1.	Formatierung der Druckseiten.....	33
1.2.	Überwindung der Abschneidfunktion.....	33
1.3.	Sperrdruck, Fettdruck.....	33
2.	Grafiken mit dem CE-1600P.....	34
2.1.	Übertragung von Grafiken in Programme.....	34
2.2.	Geometrische Grafiken.....	35
	Grafische Statistik (Erklärung zu Beispiel V.3).....	35
	Biorhythmen (Erklärung zu Beispiel V.4).....	35
	Beispiel 1: Fettdruck und Sperrdruck.....	36
	Beispiel 2: Selbstporträt des PC-1600.....	36
	Beispiel 3: Grafische Statistik.....	39
	Beispiel 4: Biorhythmen.....	44

ALLES FÜR SHARP-COMPUTER

VI. Funktion und Nutzung des CE-1600F (Diskettenlaufwerk)

1.	Vorsichtsmaßnahmen.....	45
2.	Funktion des Diskettenlaufwerks.....	46
2.1.	Die Diskette.....	46
2.2.	Vorteile der Diskette gegenüber dem Tonband.....	47
2.3.	Das "Direktionszimmer" - die Directory.....	47
2.4.	Der Verwaltungsaufwand der Directory.....	47
3.	Nutzung des Diskettenlaufwerks.....	48
3.1.	Formatierung der Diskette.....	48
3.2.	Löschen+Sichern von Programmen+Dateien auf Disketten.....	49
3.3.	Programme und Dateien.....	50
3.4.	Disketten- oder Kassettennutzung.....	50
4.	Handhabung von Diskettenbefehlen.....	51
4.1.	MAXFILES.....	51
	Verwaltung großer Dateien(Erklärung zu Beispiel VI.1)	52
4.2.	#1 (OPEN, CLOSE, INPUT, OUTPUT usw.).....	52
4.3.	APPEND.....	52
4.4.	FILENAME. EXTENSION.....	52
4.5.	Schreibschutzschalter "P".....	52
4.6.	LOC.....	52
4.7.	Komma und Semikolon als Trennzeichen.....	53
4.8.	Komma und Sonderzeichen in Dateien.....	54
5.	Dateiverwaltungsprogramme.....	54
5.1.	Adressenverwaltung.....	54
5.2.	Textverarbeitung.....	56
	Beispiel 1: Verwaltung großer Dateien	58
	Beispiel 2: Byte-Umfang/Sektorenbelegung von Dateien	59
	Beispiel 3: Byte-Umfang von Programmen	59
	Beispiel 4: Trennzeichenwirkung	60
	Beispiel 5: Adressenverwaltung	61
	Beispiel 6: Textverarbeitung	65

VII. Programme

1.	39-Zeichen.....	70
2.	Symbol-Editor.....	71
3.	Briefe schreiben mit dem PC-1600.....	72
4.	Adressaufkleber schreiben mit dem PC-1600.....	76
5.	Masstab-Umrechnungen.....	77
6.	17+4.....	78
7.	Datum und Zeit.....	79
8.	Analog-Uhr.....	79
9.	Textprogramm für PC-1600+Plotter+Laufwerk.....	80
10.	Selbstlernendes Programm.....	84
11.	Zeichen definieren.....	85
12.	Hexmonitor.....	86
13.	PC-1600 Funktionsplotter.....	87
14.	Grafikprogramm Harrascho.....	88



I. Grundsätzliches über den PC-1600

I.1. Ein Computer kann nicht bis drei zählen

Der PC-1600 ist der z. Zt. leistungsstärkste Taschencomputer. Doch mußte man bisher in Gesprächen mit Computer-Freaks ausdrücklich darauf hinweisen, daß bei SHARP die Abkürzung PC nicht Personalcomputer sondern meistens Pocketcomputer bedeutet, ist dies beim PC-1600 unwichtig. In seiner Leistungsfähigkeit hält er dem Vergleich mit einem Personalcomputer stand. Das ist vor allem seinem Z-80-kompatiblen 8-Bit-CMOS-Prozessor zu verdanken. Gegenüber einem Home-Computer hat er den großen Vorteil seiner geringen Gehäuse-Abmessungen. Ohne zusätzliche Optionen läßt er sich in der Jackentasche transportieren. Als Kompakteinheit mit Printer und Diskettenlaufwerk haben Sie überall Ihr leistungsstarkes "Büro" - in der Aktentasche transportierbar - dabei.

Wenn Sie aber seine Leistungsstärke optimal nutzen wollen - und ihn nicht nur mit erworbenen oder veröffentlichten Programmen füttern und arbeiten lassen möchten -, sollten Sie als erstes Ihre eventuelle Ehrfurcht vor dem Wunderwerk an Technik eines Computers verlieren. Ein Computer ist der dümmste Mitarbeiter, den Sie beschäftigen. Ein Computer kann nicht bis drei zählen, ja noch nicht einmal bis zwei, wenn wir genau sein wollen.

Deshalb müssen Sie ihm genau "auf die Finger sehen" und jeden einzelnen Schritt vorschreiben, den er gehen soll. Dann allerdings macht er durch seine ungeheure Geschwindigkeit - die elektrischen Wellen legen bekanntlich ca 300.000 km/sec zurück - seine "Dummheit" mehr als wett. Er wird Sie wie kein anderer, noch so intelligenter Mitarbeiter optimal entlasten.

Weil Sie ihm aber alles genau vorschreiben müssen, ist es nützlich, einen groben Überblick über das Innenleben und die Arbeitsweise des Computers zu gewinnen. Ich möchte dazu in diesem Kapitel mit bildhaften Worten dazu beitragen.

Die Bilder entsprechen nicht unbedingt den genauen elektronischen Abläufen im Computer. Wenn Ihnen der Umgang mit Maschinensprache vertraut ist oder Sie sie erlernen und anwenden möchten, werden Sie die speziellen Lehr- und Handbücher benutzen und dieses Anwendungshandbuch nur für eine bessere Nutzung PC-1600-eigener BASIC-Befehle heranziehen.

Das Anwendungshandbuch ist vor allem für die PC-1600-Anwender gedacht, die neben der Bedienungsanleitung Tips und Programmbeispiele haben möchten. Das gilt besonders - aber nicht nur - für Anfänger und PC-1600-Einsteiger.

Die Bilder sind als Eselsbrücken gedacht. Sie sollen helfen, bei auftretenden Programmierproblemen, die nicht zum erwünschten Ziel oder sogar zu einer ERROR-Meldung führen, schneller zu übersehen, wo der Computer vermutlich auf den falschen Weg geschickt wurde.

I.2. Das "Stellwerk" des PC-1600

Einiges über Bits, Bytes, ROM und RAM

I.2.1. Dual(Binär)- und Hexadezimalsystem

In manchen Handbüchern findet man an dieser Stelle Informationen, die mehr oder weniger an Kurzlehrgänge über andere Zahlensysteme (Dual- und Hexadezimalsystem) erinnern. Das möchte ich aus folgenden Gründen vermeiden:

- a) Obwohl der PC-1600 wie jeder Computer mit diesen beiden Systemen arbeitet, ist es nur nötig, diese Zahlensysteme zu erlernen, wenn Sie Maschinenprogramme erstellen wollen. Dazu müssen Sie aber grundsätzlich die Maschinensprache erlernen. Dazu gibt es die entsprechenden Lehr- und Handbücher. Für den Umgang mit BASIC genügt es, nur zu wissen, warum und wie der Computer mit diesen Systemen arbeitet.

- b) Nicht zufällig hat sich im Laufe der Geschichte das Dezimalsystem durchgesetzt, weil unser biologischer "Computer" - das Gehirn - mit dem Dezimalsystem am schnellsten rechnen kann. Warum sollten Sie sich mit zeitraubenden Zahlensystemen abgeben, wenn Sie ohnehin Rechenoperationen, die über einfaches Kopfrechnen hinausgehen, besser Ihrem PC-1600 überlassen?

Ein Computer benutzt nur deshalb das Dual- oder Binärsystem (beide Begriffe sind austauschbar), weil er überhaupt nicht rechnen kann. Er läuft gleichsam mit Lichtgeschwindigkeit die Verdrahtungen seiner Hardware entlang, um an den Kreuzungen die Spannungszustände der Weichen für den weiteren Weg zu überprüfen oder zu stellen. Ist der Spannungszustand 0, wird die Abzweigung nicht beachtet, ist der Spannungszustand positiv, also 1, wird die Abzweigung genommen.

Falls es Sie beim Lesen dieser Zeilen stört, daß die Darstellung nicht den wirklichen elektronischen Abläufen entspricht, bitte ich Sie, unter l.l. nachzulesen, weshalb ich diese bildhafte Sprache benutze.

Weil der Computer nur die beiden elektrischen Statuszustände der Weichen überprüft, arbeitet er mit Bits, mit binary digits, also mit Binärzeichen oder Dualzahlen.

Der Mikroprozessor des PC-1600 kann - wie die meisten Mikroprozessoren - gleichzeitig acht Weichen stellen bzw. überprüfen. Deshalb sind in jeder Speichereinheit, den Speicherzellen, 8 Bits = 1 Byte vorhanden.

Weil die Darstellung von Zahlen im Binärsystem sehr umfangreich und damit sehr unübersichtlich und fehleranfällig sind - der Wert 255 (2^8) ergibt eine achtstellige Zahl - verwendet man stattdessen das Hexadezimalsystem.

Das Hexadezimalsystem ist nichts anderes als eine kompaktere, platz- und zeitsparende Darstellung des Binärsystems: $2^4 = 16$.

Eine weitere kompakte Bezeichnung ist Kilobyte = 2^{10} Bytes = 1024 Bytes.

Die 8 Bits einer Speicherzelle können 256 verschiedene Konstellationen annehmen. Um wieder im Hexagesimalsystem zu bleiben, werden die ersten vier Bits zum ersten Nibble und die zweiten vier zum zweiten Nibble zusammengefaßt.

Sollte Sie das verwirren, können Sie es getrost wieder vergessen. Ich nenne diese und andere Begriffe nur, weil Sie immer wieder auch in Softwareangeboten (Programmbeschreibungen) auftauchen, und Sie so wissen was gemeint ist. Weil aber unter anderem die Displaygrafik von der Bitkonstellation abhängt, möchte ich nun die 16 Konstellationen eines Nibbles zeigen. Beim Zustand 1 wird ein Punkt auf dem Display gesetzt.

0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	B
0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	i
0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	n
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	r
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	Hexa-
																dezi-
																mal
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Dezi-
																mal

Die Maschinenprogramme des ROM-Speichers umfassen zwei Bytes. Deshalb spricht man vom High Byte und Low Byte. Auch das erwähne ich nur, damit Sie etwas mit der Bezeichnung anfangen können, wenn sie in einer Programmbeschreibung begegnet.

Low Byte ist dabei das zweite Byte, das die Potenzen bis 2^7 hat; das High Byte umfaßt die Potenzen $2^8 - 2^{15}$.

Vielleicht ist Ihnen nun auch verständlich, warum 127 ($2^7 - 1$), 255 ($2^8 - 1$) und 65535 ($2^{16} - 1$) Grenzzahlen des Computers sind, die uns immer wieder begegnen.

Da wir im Gegensatz zum Computer nicht mit dem Binär- und Hexadezimalsystem arbeiten wollen, soll im Folgenden nur kurz aufgezählt werden, warum diese wenigen grundsätzlichen Informationen über Bits, Bytes und HEX-Ziffern als Eselsbrücken für die Programmierpraxis hilfreich sind:

- a) Manchmal stoßen wir beim Programmieren an Grenzen des PC-1600, die auf den ersten Blick unerklärlich sind, wenn man nicht bedenkt, daß sie mit der Binär-Arbeitsweise des Computers zusammenhängen.

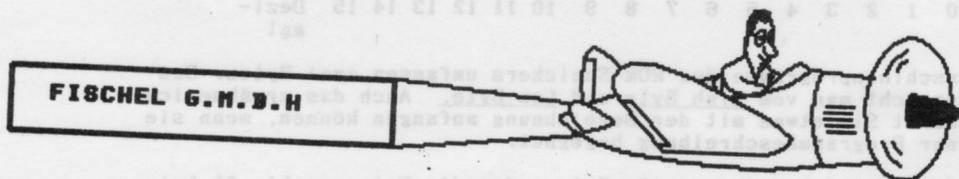
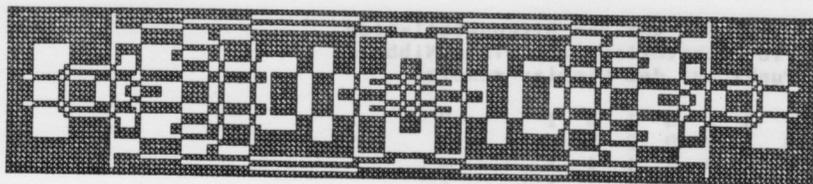
Z.B. sind Arrays nur bis zum Wert $255 = 2^8 - 1$ (da auch 0 mitzählt) dimensionierbar; bei einem höheren Wert meldet der Computer ERROR 19 = "Parameter liegt außerhalb des zulässigen Bereiches".

Die adressierbaren Druckluftkoordinaten liegen zwischen -2048 und $2047 = -(2^{11}) - (2^{11} - 1)$.

Wenn der PC-1600 bei Programmen mit einer ERROR-Meldung zeigt, daß ein zulässiger Bereich überschritten wurde, und Sie sich keines Programmierfehlers bewußt sind, empfiehlt es sich, zunächst zu überprüfen, ob durch die Bearbeitung einer bestandenen Variablen eine zulässige Zweier-Potenz für den entsprechenden Befehl überschritten wird.

Das kommt z.B. bei Dimensionierungen von Variablen vor. In einem solchen Fall begrenzt man die Dimensionierung auf 255. Dieses wiederum sollte man nur tun, wenn beim Probelauf sich erweist, daß die dimensionierte Variable den Wert von 255 überschreiten kann. Ist dieser Fall unwahrscheinlich, würde durch eine grundsätzliche Dimensionierung auf 255 unnötig RAM-Speicherraum verschenkt, was wiederum gerade der DIM-Befehl - sinnvoll eingesetzt - vermeiden soll.

- b) Das Wissen um die Hexadezimal-Arbeitsweise ist vor allem auch für die Display-Grafik nützlich. Da das große Display des PC-1600 (156x32 Punkte) zur Erstellung von Grafiken reizt, ist diesem Aufgabenbereich ein eigenes Kapitel gewidmet.



1.2.2. Die Arbeitsweise eines Computers

Da der Computer nicht rechnen, sondern nur die Weichen der Bits stellen oder überprüfen kann, ist selbst der einfachste Rechenvorgang für den PC-1600 eine Prozedur relativ langer Wege.

Dies soll nur an einer Aufgabe gezeigt werden:

Wie wird die Aufgabe 1+1 gelöst?

Sie veranlassen den Computer, "auf einem Gleis zur ersten unbenutzten Speicherzelle zu fahren und die Weichen der Bits so zu stellen, daß sie dem Code-Signal für die Ziffer 1 entsprechen. Dann werden die Weichen der nächsten Zelle so gestellt, daß der Computer in seinem Wörterbuch nachsehen muß, wie die Weichen zu stellen sind, damit eine Addition erfolgt.

Er fährt zur nächsten Zelle und stellt die Weichen der Bits so, daß sie ebenfalls den Code für die Ziffer 1 darstellen. Das Additionsprogramm seines Wörterbuches veranlaßt den PC-1600, den Speicherinhalt der genannten dritten Zelle zum Speicherinhalt der ersten Zelle zu befördern und hineinzupacken, wobei der Computer beachten soll, daß der Speicherinhalt der ersten Zelle nicht hinausgedrängt wird (was normalerweise geschieht, wenn eine Speicherzelle einen neuen Inhalt erhält), sondern zusammen mit dem Neuzugang so verknüpft wird, daß die neugestellten Weichen das Codezeichen für die Ziffer 2 ergeben.

Da Sie anschließend das Ergebnis sehen wollen, wird dem PC-1600 durch ein weiteres Programm seines Wörterbuches signalisiert, daß er das Codezeichen entweder auf dem Display oder einem Drucker darstellen soll - oder innerhalb eines Programmes für sich behalten und weiter verarbeiten soll - durch Abfahren befohlener Fahrstränge."

Allein durch seine Geschwindigkeit merken Sie nichts von den langen Wegen, die um so länger und komplizierter sind, je schwieriger der Rechenvorgang oder der Verarbeitungsbefehl ist.

Vielleicht hat man auch deshalb dem Bearbeitungsbefehl das Codewort RUN (renne! laufe!) gegeben.

1.2.3. Der Mikroprozessor das Stellwerk des Computers

In einem BASIC-Lehrbuch fand ich für das, was in der Programmier-technik im allgemeinen Flußdiagramm und in der Bedienungsanleitung des PC-1600 Syntax-Diagramm genannt wird, die Bezeichnung Fahrnetz der BASIC-Syntax.

Das inspirierte mich, den Prozessor als Stellwerk zu bezeichnen. Er ist das wichtigste Teil des Computers, die Kommandozentrale - entsprechend unserem Gehirn, das alle Nervenimpulse steuert. Ich halte es trotzdem für besser, den Prozessor nicht als Gehirn des Computers und die Verdrahtungen nicht als Nervenstränge sondern als Stellwerk bzw. Gleise zu bezeichnen. Der Computer kann nicht von sich aus tätig werden, und physikalisch gleicht der Mikroprozessor eher einem Stellwerk, und die Verdrahtungen ähneln Schienensträngen - ebenso die Programm-Syntaxstrukturen Fahr- oder Schaltplänen.

Das Bild vom Stellwerk sollte man sich geistig vor Augen halten. Mir hat es schon oft geholfen, Programmierfehler zu vermeiden oder zu überwinden oder Bearbeitungszeiten des Computers möglichst kurz zu halten - ein Vorteil, wenn die Programme sehr umfangreich sind.

1.2.4. Der BASIC-Interpreter - das Wörterbuch des Computers

Mit dem Bild des Stellwerks sollte man auch verbinden:

- a) Die Informationen über Bits und Bytes, um den Prozeß der langen Wege des PC-1600 zu verstehen.
- b) Obwohl Sie schon mit dem PC-1600 in einer Fremdsprache - englisch - sprechen, kann er auch sie nicht verstehen und im Gegensatz zu Ihnen auch nie erlernen. Bei jedem Zeichen, bei jedem Befehl muß er in seinem Wörterbuch - BASIC-Interpreter (Übersetzer) genannt - nachsehen, auch wenn das Zeichen oder der Befehl mehrmals hintereinander kommt.

Deshalb arbeitet ein Computer mit Maschinenprogrammen schneller. Der Maschinenprogrammierer hat gleichsam die Sprache des Computers erlernt und kann mit ihm ohne Dolmetscher reden. Der Computer braucht nicht mehr jedesmal zu seinem Wörterbuch zu laufen, um nachzusehen, was er auf Befehl mit der nächsten Zelle tun soll. Er kann direkt in die die zu bearbeitende Zelle springen und die Weichen und die Weichen stellen. Der entsprechende Befehl heißt folgerichtig auch jump, "springe!":

Aber bevor Sie anfangen, die Maschinsprache des Computers mit seinem unübersichtlichen Binär- und Hexadezimalsystem zu erlernen, sollten Sie bedenken, ob es Ihnen Freude macht. Mir machte es Freude, weil Mathematik zu meinen Hobbies gehört. Dennoch programmiere ich überwiegend in BASIC, da ich Geisteswissenschaftler bin und von daher wenig Programme mit rechenintensiven Routinen benötige. Aber nur rechenintensive Programme bringen bei der Bearbeitung eine spürbare Zeitersparnis, wenn sie als Maschinenprogramme (Mapro) geschrieben sind.

- c) Ein Computer ist um so besser, je größer der Wort- oder Befehlschatz in seinem Wörterbuch ist. Das ist der ROM-Speicher (read only memory = Nur-Lese-Speicher zum Nachschlagen).

Während der ROM-Speicher des PC-1600 nur 16 KBytes hat, hat der PC-1600 96 KBytes, also ein sechsmal so großes Wörterbuch wie der kleinere Bruder.

Im größeren ROM-Speicher sind offensichtlich auch die Preisunterschiede begründet, aber vor allem auch die größeren Möglichkeiten des PC-1600. Während man z.B. für ein DELETE- oder RENUMBER-Programm beim PC-1500 wertvollen Arbeitsspeicherraum abzweigen muß (meist auch verhältnismäßig viel Geld für den Erwerb solcher und anderer Software) und oft noch mehrere Tastenbetätigungen nötig sind, um die Programme anwenden zu können, genügt beim PC-1600 nur die Eingabe des entsprechenden Befehls.

Es sei in diesem Zusammenhang erwähnt, daß das Wörterbuch des Computers zum einen Zeichencodes enthält für Buchstaben, Zahlen und Sonderzeichen, zum anderen Programmcodes für die BASIC-Befehle - die ihrerseits Maschinenprogramme sind und zwei Bytes umfassen.

Wenn also jemand über den geringen Arbeitsspeicher des PC-1600 klagte, dann sollte man dabei bedenken, daß der ROM-Speicher viel wichtiger ist und eben Kompromisse und Abstriche notwendig sind, wenn es wirklich ein Pocketcomputer sein soll, der seinem Namen Ehre macht, also notfalls in der Hosentasche zu befördern ist und außerdem noch erschwinglich sein soll.

1.2.5 Kompatibilität

Wenn Computer dasselbe Wörterbuch benutzen, sind sie miteinander kompatibel, d.h. Programme können direkt zwischen ihnen ausgetauscht und sofort ohne Änderungen verarbeitet werden, da sie mit denselben Codes arbeiten. Ist dies nur teilweise der Fall, sind sie nur teilweise kompatibel. So ist der PC-1600 mit den Home- bzw. Personalcomputern kompatibel, die den weit verbreiteten Z-80-Mikroprozessor haben, mit dem PC-1500 nur aufwärtskompatibel. Der große Bruder kann zwar den kleinen Bruder hundertprozentig verstehen, hat aber selbst einen anderen und vor allem größeren Wortschatz, den der PC-1500 nicht versteht.

1.2.6. Der Arbeitsspeicher

Der Arbeitsspeicher wird auch als RAM-Speicher (random access memory = Speicher des willkürlichen Zugriffs) oder Schreib-Lese-Speicher bezeichnet. In ihm steht also nur das drin, was Sie in ihn hineinschreiben, damit es der Computer lesen und verarbeiten kann.

1.2.7. Der ASCII-Code

Eine letzte Information in diesem Zusammenhang gilt dem ASCII-Code. Er ist die Abkürzung von American Standard Code for Information Interchange = Amerikanischer Standardcode für den Informationsaustausch und wurde nicht erst für Computer entwickelt, sondern schon lange vorher, als man beispielsweise mit dem Telegraf Nachrichten übermitteln wollte.

So unterschiedlich auch immer die Wünsche der einzelnen Völker sind (wir wünschen unsere deutschen, die Skandinavier und Franzosen ihre Sonderzeichen, die Japaner, Russen, Araber und andere Völker haben ganz andere Schrift) und so sehr auch jede Computergesellschaft Wert auf ihr spezielles ROM-Wörterbuch legt, man hat sich immerhin darauf geeinigt, daß die erste Hälfte der 256 Codezeichen, die durch die Weichenstellung der acht Bits in einer Speicherzelle (2⁸) gebildet werden können, in allen Wörterbüchern identisch sind.

1.2.8. Bit-Parität

Diese 128 Zeichen sind mit sieben Bits (2⁷=128) darzustellen. Deshalb wird bei Anschluß anderer Geräte - z. B. Drucker - danach gefragt, ob eine 7-Bit- oder 8-Bit -Parität hergestellt werden soll - so auch beim SETCOM -Befehl des PC-1600.

1.2.9. Bearbeitung der Programme

Hier sollen nur wenige Informationen darüber gegeben werden, wie der PC-1600 Programme bearbeitet, um einen Überblick darüber zu vermitteln, wieviele Bytes der Computer für die Bearbeitung der Programme benötigt.

- a) Der PC-1600 braucht pro Zeilennummer zwei Bytes. Auch wenn Sie nur einstellige Zeilennummern eingeben - oder fünfstellige - benötigt er zwei Bytes. Er rechnet sie dual um und speichert sie in einem High-Byte(HB) und in einem Low-Byte(LB):

		HB	LB
Zeilennummer	1:	& 00	01
Zeilennummer	10:	& 00	0A
Zeilennummer	255:	& 00	FF
Zeilennummer	2815:	& 0A	FF
Zeilennummer	65535:	& FF	FF

Sie können also großzügig mit Zeilennummern umgehen, da Sie durch die Wahl weder Bytes sparen noch vergeuden können.

- b) Ein Byte braucht der Computer zur Berechnung der Zeilenlänge. Es ist hoffentlich für Sie inzwischen einsichtig geworden, warum der Computer dafür nur ein Byte braucht: Da eine Zeile höchstens 80 Zeichen enthalten kann, ist die Anzahl der Zeichen dual in einer Zelle zu speichern.

Diesen Merkposten richtet der Computer ein, um berechnen zu können: die Bytezahl eines Programms für die Abspeicherung; die Speicherraumreservierung für indizierte (d.h. dimensionierte) Variablen: den verfügbaren Platz, um weitere Programme in den Arbeitsspeicher laden zu können, u.a.m.

- c) Ein Byte verbraucht der ENTER-Befehl. Er ist das Signal zum Schließen einer Programmzeile und für das Öffnen der nächsten.
- d) Zwei Byte benötigt der Computer für den Beginn und den Abschluß eines Programms, um es gegebenenfalls gegen andere Programme abzugrenzen.

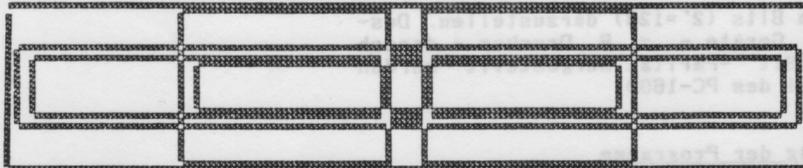
Die vier Bytes je Zeile und die zwei Bytes zur Kennzeichnung des Programms benötigen Sie auf jeden Fall.

Außerdem können Sie sich merken:

- c) Zwei Bytes (nur) kostet jeder BASIC-Befehl, auch wenn er mehr Buchstaben hat. Das hängt damit zusammen, daß jeder Befehl grundsätzlich ein ROM-Programm mit zwei Bytes Umfang ist.

Bei Platzmangel im Arbeitsspeicher werden Bytes gespart, wenn Sie mehrere Programmsätze - durch den Doppelpunkt (:) voneinander trennt - in einer Programmzeile unterbringen. Der Doppelpunkt benötigt zwar ein Byte - aber Sie sparen für jeden Programmsatz, für den Sie keine eigene Programmzeile einrichten, fünf Bytes.

Ich hoffe, daß diese grundsätzlichen Informationen über den PC-1600 Sie nicht all zu sehr ermüdet oder sogar gelangweilt hat. Sie helfen Ihnen auf jeden Fall, auf das Innen- und Eigenleben besser zu achten und weniger Frust bei Ihren Programmen zu erleben.



1.3. Der PC-1600 kann mehr, als die Bedienungsanleitung ausweist

In diesem kurzen Abschnitt werden zusätzliche Informationen zur Bedienungsanleitung gegeben. Es werden also nicht die Informationen der Anleitung erklärt, sondern nur darauf hingewiesen, wo sie Möglichkeiten verschweigt.

Unter 9-9 muß ergänzt werden:

CTRL + G : Diese Tastenkombination aktiviert oder löscht die Klick-Funktion.
Diese Tastenkombination hat damit die-
selbe Wirkung wie die CLICK -Taste.

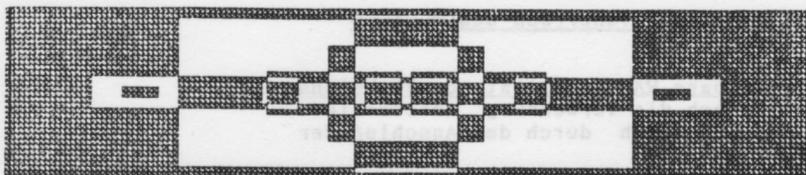
MERGE Die Information zu diesem Befehl unter 14-123 der Bedienungsanleitung ist zu-
mindest mißverständlich. Im Unter-
schied zum PC-1500 läßt das ROM-
Programm des PC-1600 durchaus zu, daß
nicht nur auf das durch MERGE zuletzt
geladene Programm zurückgegriffen wird
sondern alle Programme editiert werden
können.

Man braucht dazu nur im RUN-Modus das
Programm aufzurufen, das man editie-
ren möchte, es durch BREAK unterbre-
chen und dann anschließend in den PRO-
Modus umzuschalten. Dann kann man das
gewünschte Programm verändern.

Schließlich möchte ich noch auf einen offensichtlichen Druckfehler aufmerksam machen. Gibt man eine

größere Zeilennummer als 65279 ein, erfolgt ERROR 30

Der Hinweis unter A-14 zu ERROR 30 stimmt also nicht.



II. Nutzung der PC-1500-Optionen

und spezielle Hinweise für PC-1500-Umsteiger

Vielleicht sind Sie auf den PC-1600 aufmerksam geworden, weil Sie den PC-1500 besitzen. Bis SHARP sein neues "Schlachtschiff", den PC-1600, auf den Markt brachte, gehörte der kleinere Bruder zu den beliebtesten Pocketcomputern, weil er im Soft- und Hardware-Angebot sehr vielseitig ist und vielfach auch professionellen Ansprüchen genügt.

Wenn Sie mit dem Erwerb des PC-1600 "SHARP-Neuling" sind, sollten Sie dennoch dieses Kapitel nicht überschlagen. Zum einen lernen Sie die Vorzüge Ihres Computers noch ein bißchen besser kennen. Zum anderen regt es Sie vielleicht an, bei günstigen Angeboten von PC-1500-Hardware und Software, die immer wieder bei diesem weitverbreiteten Computer-System zu verzeichnen sind, zuzugreifen und Ihren PC-1600 zu erweitern.

II.1. ERROR 110

Beim Betreiben des PC-1600 im PC-1500-Modus kann es passieren, daß er mit der Meldung ERROR 110 aussteigt. Vielleicht gelingt es Ihnen gar nicht erst, den PC-1600 in den MODE 1 umzuschalten.

Dafür gibt es eine einfache Erklärung. Wenn Sie sie beachten, werden Sie sich künftig nicht mehr über eine ERROR-110-Meldung ärgern sondern sogar dankbar dafür sein, daß SHARP durch die Konzeption des PC-1600 die Möglichkeit geschaffen hat, die PC-1500-Optionen optimal zu nutzen - sogar besser, als es die Bedienungsanleitung und die offizielle Werbung verrät.

II.2. Der Nebenprozessor LH 5803

Um die Kompatibilität zum PC-1500 herzustellen, hat SHARP den PC-1600 außer mit dem Z-80-kompatiblen 8-Bit-CMOS-Hauptprozessor und einem Hilfsprozessor auch mit dem Nebenprozessor LH 5803 versehen.

Der Nebenprozessor LH 5803 streikt mit ERROR 110, wenn

- a) im Mode 1 Befehle gegeben werden, die der PC-1500 nicht kennt.

Wenn Sie also auf dem PC-1600 nach Umschalten in den Mode 1 nur Befehle anwenden, die Sie vom PC-1500 kennen, wird er erwartungsgemäß alle Programme, die für den kleinen Bruder geschrieben sind, ohne Unterbrechung oder gar Abbruch ausführen. Nur bei Ausführung von Maschinenprogrammen sind Besonderheiten zu beachten, die aber das eben Dargelegte nicht rückgängig machen.

- b) der Arbeitsspeicher des PC-1600 außer der Bank 0 noch weiteren Speicherraum belegt.

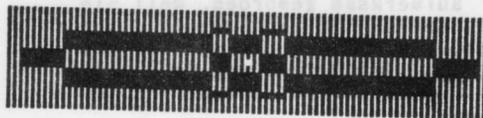
Der Nebenprozessor muß also den Arbeitsspeicher des PC-1600 ganz kontrollieren können. Ist der Arbeitsspeicher größer, ist der PC-1600 nicht in den Mode 1 umschaltbar.

11.3 Der für den Mode 1 verfügbare Speicherraum des PC-1600

Die Speicherbank 0 hat eine verfügbare RAM-Kapazität von höchstens 28218 Bytes. Sie wird nicht nur durch die Verwendung unterschiedlicher Module verringert, sondern auch durch den Anschluß der Diskettenstation.

Die verfügbare Kapazität der Bank 0 nutzen Sie vollkommen aus, wenn Sie in das Slot 1 ein CE-161-Modul mit 16-KByte installieren.

Benutzen Sie weitere Module oder nur ein CE-1600-Modul mit 32 KByte, dann müssen als RAM-Disks initialisiert werden. Sonst erscheint ERROR 110.



Die Größe des Arbeitsspeichers ist zu kontrollieren:

STATUS 0 + STATUS 1

Die Bytezahl des Arbeitsspeicher wird angezeigt.

STATUS 2

Der PC-1600 zeigt die niedrigste Speicherzelle, die in einer der acht Speicherbänke adressierbar ist. Diese Zahl ist für Sie nur wichtig, wenn Sie Maschinenprogramme verwenden.

STATUS 3

Die höchstwertige Speicherzelle in einer der 8 Speicherbänke wird angezeigt. Auch das ist nur bei Verwenden von Maschinenprogrammen wichtig.

Wichtig für die Aktivierbarkeit des PC-1500-Modus sind die beiden folgenden Eingaben.

STATUS 256

Angezeigt wird die Nummer der Speicherbank, in der die niedrigste zur freien Verfügung stehende Zelle des Arbeitsspeicher steht.

STATUS 257

ergibt die Speicherbank, in der die höchste zur freien Verfügung stehende Zelle des Arbeitsspeicher vorhanden ist.

Ist eine der beiden Zahlen unter STATUS 256 oder STATUS 257 nicht 0, kann der PC-1600 nicht im Mode 1 arbeiten. Er reagiert mit ERROR 110.

Probieren Sie die Eingaben in der angegebenen Reihenfolge unter den verschiedensten Bedingungen aus, um ein Gefühl für die Behandlung des Arbeitsspeichers und vor allem für die Festlegung seiner Größe zu bekommen. D.h. initialisieren Sie Ihre Speichermodule einmal als Arbeitsspeichererweiterung durch

INIT "S1:", "M" bzw. INIT "S2:", "M"

danach als Ram-Disk durch den Befehl

INIT "S1", "F" bzw. INIT "S2:", "F"

Außerdem sollten Sie die Module vorübergehend ausbauen und die Diskettenstation einmal anzuschließen und einmal abzukoppeln.

Sie werden beim Abfragen der Statusbereiche nicht nur die Verschiebungen des Arbeitsspeichers feststellen. Sie werden sehen, daß Module im Slot 2 grundsätzlich die Speicherbänke ab Bank 2 aufwärts ansprechen.

Sie werden auch bemerken, daß Speicherkapazität verloren geht, wenn sie Module als Ram-Disks initialisieren oder die Diskettenstation anschließen. Der PC-1600 braucht nämlich für die Verwaltung von Ram-Disk- oder Disketten-Dateien Speicherraum. Nähere Erläuterungen finden Sie unter VI. Funktion und Nutzung des CE-1600F.

11.4. Maschinenprogramme des PC-1500

Vor allem, wenn Sie Umsteiger vom PC-1500 sind, werden Sie vermutlich eine Reihe von Maschinenprogrammen besitzen, die Sie gern weiterverwenden wollen.

Alle Maschinenprogramme des PC-1500 lassen sich ohne Einschränkung in den PC-1600 laden und arbeiten fehlerfrei, wenn der Mode 1 eingeschaltet ist.

Besitzer des PC-1500A haben allerdings Folgendes zu beachten:
Da der PC-1600 wegen des großen ROM-Speichers auch den Speicherraum benötigt, der beim PC-1500A hardwaremäßig für Maschinenprogramme reserviert ist (Speicherzellen 31744-32767 =&7C00-7FFF), können sie vom PC-1600 auch im Mode 1 nicht für Maschinenprogramme des PC-1500A genutzt werden, wenn Sie für diesen Speicherraum geschrieben sind. Das heißt nicht, daß damit die Maschinenprogramme wertlos sind. Sie müssen nur Folgendes beachten:

- a) Der PC-1500A ist wie ein PC-1500 zu behandeln.
d.h.:
- b) Stellen Sie durch STATUS 2 den Beginn des Arbeitsspeichers fest.
Haben Sie kein Modul in Slot 1 (und Slot 2) wird dies normalerweise die Speicheradresse 16582 sein. Bei optimaler Nutzung des Mode 1 (RAM-Modul CE-161 in Slot 1) ist dies die Speicheradresse 198.
- c) Stellen Sie die Byte-Zahl des Maschinenprogramms fest.
Der Umfang eines Maschinenprogramms ist immer in der Programmbeschreibung angegeben.
- d) Geben Sie ein: NEW Status 2 + Mapro-Umfang.

Geben Sie entweder die Anfangsnummer von Status 2 ein, oder einfach STATUS 2.

- e) Sie laden das Maschinenprogramm mit CLOAD M (Speicheranfangsadresse des Status 2).

Beim PC-1600 ohne Speichermodul ist dies CLOADM 16582 bzw. CLOADM &40C6. Bei Verwendung des CE-161 lautet der Befehl CLOAD M 198 bzw. CLOAD M &C6.

Im Zweifelsfalle befolgen Sie die Hinweise für den PC-1500, die Maschinenprogrammen meistens beigegeben sind.

Werden diese Hinweise beachtet, werden Sie keine Probleme mit der Installation von PC-1500-Maschinenprogrammen auf dem PC-1600 haben.

11.5. PC-1600 und Drucker/Kassetten-Interface CE-150

Sie können den PC-1600 an das CE-150-Interface anschließen. Das werden Sie vor allem nutzen, wenn Sie zunächst den Computer ohne das Drucker/Kassetten-Interface CE-1600P erwerben.

Auch das CE-150 arbeitet nur problemlos mit den Befehlen für den PC-1500.

Sie können nach Laden von PC-1500-Programmen den PC-1600 in den Mode 0 umschalten. Text- und Rechenprogramme werden ausgedruckt, ohne daß Sie einen Eingriff in die Programme vornehmen müßten.

Laden Sie PC-1500-Programme von Kassetten in den PC-1600, brauchen Sie auf die Umwandlung der LINE-Befehle in LLINE nicht zu achten. Sie wird automatisch vom PC-1600 vorgenommen. (Das gleiche gilt für XCALL und XPOKE).

Bei Grafikprogrammen, die Sie von Druckvorlagen abtippen, sind die LINE-Befehle in LLINE zu ändern, weil der Line-Befehl nur für Display-Grafiken zuständig ist.

Zur Vorsicht weise ich darauf hin, daß nicht etwa Grafiken auf dem PC-1600-Display erzeugt werden, wenn Sie den LINE-Befehl eines PC-1500-Programms unverändert lassen. Für den Drucker gelten andere Parameter als für die Darstellung auf dem DISPLAY.

Die Sonderzeichen, die über die KBII-Taste verfügbar sind, können für Display-Programme auch in MODE 1 genutzt werden. Versucht man jedoch einen Ausdruck über das CE-150-Interface, erscheinen Leerzeichen auf dem Papier.

11.6. PC-1600 im PC-1500-Modus mit RAM-Disks

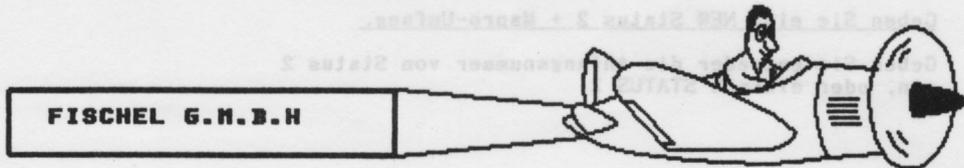
Vielleicht haben Sie zunächst einmal nur den PC-1600 erworben. Nun stehen Sie vor der Alternative, als erste Erweiterung entweder das Drucker/Kassetten-Interface oder RAM-Module zu kaufen.

Das Folgende ist als Entscheidungshilfe gedacht.

Besitzen Sie das CE-150 und vielleicht sogar noch einen Drucker, z.B. den CE-515P oder den CE-516P, ist erwägenswert, statt des CE-1600P zunächst ein oder zwei RAM-Module zu erwerben.

Zum einen ist auf der nächsten Seite zu lesen, warum es mitunter nützlicher sein kann, neben dem Drucker des PC-1600 einen anderen Drucker zu verwenden. Zum anderen sind RAM-Module wesentlich billiger zu haben als der CE-1600-Plotter.

Wollen Sie vor allem die Möglichkeit nutzen, zu gespeicherten Programmen und Dateien schnellen Zugriff zu haben, sollten Sie vielleicht zunächst RAM-Module verwenden (CE-159, CE-161 und CE-1600M).



Das Diskettenlaufwerk CE-1600F ist nur mit dem CE-1600P zu betreiben. Sie müßten also beides anschaffen, jedes der beiden Geräte zu einem Preis, der höher ist als der empfohlene Preis für das 32-KByte-Modul CE-1600M; außerdem noch die Spezialdisketten.

Da ist die Möglichkeit, Module als RAM-Disks zu nutzen, eine kostengünstige Alternative für die erste Erweiterung des PC-1600-Systems.

Auch im Mode 1 sind RAM-Module als RAM-Disks nutzbar. Bevor Sie in den Mode 1 umschalten, müssen Sie nur im Mode 0 die Module durch den Befehl

INIT "S1:", "F" bzw. INIT "S2:", "F"

als RAM-Disks initialisieren.

Bei Anschaffung von Modulen müssen Sie nur darauf achten, ob sie als RAM-Disks initialisiert werden können. Bei den Original-SHARP-Modulen ist dies beim CE-151 und CE-155 nicht der Fall.

II.7. PC-1600 + CE-150 + CE-158 + CE-515P(516P)
und
PC-1600 + CE-1600P + CE-158 + CE-515P(516P)

Da der PC-1600 mit dem CE-150 zu betreiben ist, sind über die CE-158-Schnittstellen-Einheit alle Peripheriegeräte des PC-1500 weiter nutzbar.

Daß dies aber auch für das Drucker/Kassetten-Interface CE-1600P in Verbindung mit dem CE-158 gilt, geht aus der Bedienungsanleitung zum PC-1600 und der PC-1600-System-Übersicht-Zeichnung nicht eindeutig hervor. So konnte man in Produktinformationen über den PC-1600 und bei User-Anfragen wiederholt lesen, daß z.B. der DIN-A4-Plotter CE-515P(516P) leider mit dem PC-1600 nicht verwendbar sei. Unterstützt wird dieses Fehltrium dadurch, daß in der Bedienungsanleitung unter der Aufzählung der benutzbaren PC-1500-Optionen (7-1 und 7-2) dieser Plotter fehlt.

An der linken Seite des CE-1600P befindet sich ein BUS-PORT, über den das CE-158 durch einfaches Einstecken mit dem CE-1600P verbunden werden kann.

Auf diese Weise lassen sich der CE-515P(516P) und andere Peripheriegeräte genauso problemlos betreiben, wie über das CE-150-Interface.

Wenn Sie den CE-515P bzw. CE-516P besitzen oder die Möglichkeit haben, ihn preisgünstig zu erwerben, sollten Sie ihn aus folgenden Gründen für den PC-1600 verwenden:

- a) Der CE-515P(516P) läßt sich stufenlos auf jede Papierbreite von 9.5-22 cm einstellen.

Das ist mit dem CE-1600P nicht möglich, da Papier nur in DIN-A4-Breite (bzw. im US-letter-Format) exakt im Plotter bewegt wird. Zwar können Sie mit dem PAPER-Befehl auch andere Druckbreiten einstellen (s. unter V. Nutzung des CE-1600P), aber es entsteht viel Papierabfall.

- b) Briefkarten und kartonstarke Vorlagen sind mit dem CE-515P (516P) bis zu einer Stärke von 2.8 mm zu verarbeiten.

Das kann man für die Beschriftung von Glückwunschkarten und Kunstdrucken nutzen, weil das Papier oder der Karton nicht geknickt wird.

- c) Mit dem CE-515P(516P) läßt sich auch Computer-Endlospapier in verschiedenen Formaten verarbeiten.

Dies läßt sich z.B. bei der Adressverwaltung optimal nutzen - z.B. für Listen und Adreßaufkleber.

Wenn Sie den CE-515P besitzen, weise ich Sie auf mein **DZS-Programm** hin, das Sie als Softwarekassette über die Fischel GmbH erwerben können.

Dieses BASIC-Programm erlaubt in acht Versionen je nach Speichergröße, benötigten Sonderzeichen und gewünschtem Bedienungskomfort die Nutzung der deutschen Sonderzeichen und Akzente auch für den CE-515P.

II.8. Verknüpfung von BASIC-Befehlen

Im Gegensatz zum PC-1500 sind beim PC-1600 die BASIC-Befehle jederzeit kombinierbar.

Hat man z.B. in einem Programm einen PRINT-Befehl programmiert und möchte ihn später in LPRINT oder GPRINT ändern, braucht nur ein L bzw. G eingefügt zu werden. Der PC-1600 wandelt die Befehle erst während des Programmablaufs in die jeweiligen ROM-Codes um.

Diese Programmiermöglichkeit ist vor allem nützlich, wenn man die vordefinierte Tastaturbelegung der ersten Tastenreihe des PC-1600 verwenden und erweitern will - z.B. L + DEF W = LPRINT.

III: Optimale Nutzung einiger BASIC-Befehle

Die Erläuterungen der BASIC-Befehle in der Bedienungsanleitung sollen nicht wiederholt werden. Es sollen - ausgelöst durch Anfragen - einige Befehle, soweit sie nicht in den weiteren Kapiteln behandelt werden, in ihrer Funktion näher erläutert und ergänzt werden.

III.1.1. BEEP - Musik

Als einziger BASIC-Befehl in diesem Kapitel soll BEEP in seiner Vielfalt durch ein Beispielprogramm erläutert werden. BEEP erzeugt nicht nur nervende Pieptöne, die man oft durch BEEPOFF einfach abschaltet.

Durch die drei Parameter - Anzahl, Tonhöhe, Dauer - können Sie Ihren PC-1600 veranlassen, Musik zu machen.

Wegen der Grenzzahl 255 (vgl. I.2) können "nur" 4 1/2 Oktaven gespielt werden. Vom Klang her empfehlen sich nur die Töne bis zum Parameter um 59. Das sind immerhin noch zwei Oktaven.

Die "Tonfolge" des PC-1600 beträgt - vom tiefsten Ton aufwärts:

255 - a	104 - c	39 - dis	12 - fis
242 - b	98 - cis	37 - e	11 - g
230 - h	92 - d	35 - f	10 - gis
218 - c	86 - dis	32 - fis	9 - a
206 - cis	81 - e	30 - g	8 - b
192 - d	76 - f	29 - gis	7 - h
180 - dis	71 - fis	26 - a	6 - c
169 - e	67 - g	24 - b	5 - d
159 - f	63 - gis	22 - h	3 - dis
150 - fis	59 - a	20 - c	2 - g
141 - g	55 - b	19 - cis	1 - gis
123 - gis	51 - h	18 - d	
125 - a	48 - c	16 - dis	
118 - b	45 - cis	15 - e	
111 - h	42 - d	14 - f	

Notenlänge (Relation von T und N) und Notenwert werden in DATA-Zeilen untergebracht - ebenso die Pause (N=0).

Die Notenlänge L wird von der kürzesten Note eines Musikstückes bestimmt. Bei der kürzesten Note ist L=1. Enthält also die Melodie Sechzehntel-Noten, hat eine ganze Note den Wert L=16.

Das Tempo der Melodie wird von der Variablen T bestimmt: Je größer T, umso langsamer das Tempo.

Um die Anzahl der Noten nicht mühsam abzuzählen, wird nach dem letzten Notenwert -1 eingegeben.

Um die Noteneingabe zu erleichtern, werden die Funktionstasten folgendermaßen belegt:

Reserveebene I

F1:.,104, F2:.,92, F3:.,81, F4:.,76, F5:.,118, F6:.,111,

Reserveebene II

F1:.,218, F2:.,192, F3:.,169, F4:.,159, F5:.,141, F6:.,125,

Reserveebene III

F1:.,296, F2:.,180, F3:.,150, F4:.,133, F5:.,98, F6:.,86,

Die Funktionstastenstrings lauten:

Reserveebene I

" C1 D1 E1 F1 B H"

Reserveebene II

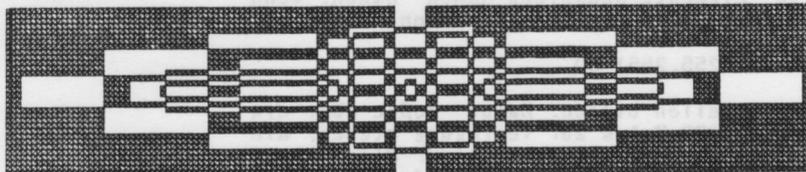
" C D E F G A"

Reserveebene III

" C# D# F# G# C1# D1#"

Beachten Sie, daß beide Kommata jeweils zu der Belegung einer Funktionstaste gehören.

Bei dieser Belegung der Reserveebene braucht in den Datazeilen jeweils nur die Zahlentaste für die Notendlänge und nach Einschalten der entsprechenden Ebene die Notentaste gedrückt zu werden.



Beispiel 3.1

Melodien

```

1:      Viva, viva la musica!
2:
3:CLS :WAIT 0:T=8000:CURSOR 3,2:PRINT "Viva, viva la musica
4:READ L,N
5:ON ERROR GOTO 9
6:IF N=0WAIT 10*L:PRINT :GOTO 8
7:BEEP 1,N,(T/N)*L
8:GOTO 4
9:END
10:
11:      Notenteil
12:
13:DATA 3,92,1,104,2,111,1,125,1,141,2,150,2,141,4,150,3,111
14:DATA 1,104,2,92,1,104,1,111,2,125,2,141,4,125
15:DATA 2,141,4,141,2,218,2,192,2,169,4,192,-1
    
```

III.1.2. DIM - Überwindung des Limits 255

Auf Seite 1 - 6 habe ich auf den Grenzwert 255 für die Dimensionierung von indizierten Variablen und Arrays hingewiesen (indiziert sind dimensionierte Variablen).

Das heißt aber nicht, daß sie nur über 256 indizierte Variablen verfügen könnten. Durch einen Trick können Sie beliebig viele indizierte Variablen verwenden.

In einem Array können alle Elemente denselben Namen haben. Wenn Sie z.B. im Textverarbeitungsprogramm definieren würden:

```
DIM Z$(255,255)*80
```

könnten Sie theoretisch 65536 Zeilen bilden. Dafür müßte aber ein Speicherraum von mindestens 542880 Bytes zur Verfügung stehen, die nicht durch Programme im Arbeitsspeicher belegt sind. Diese theoretische Möglichkeit habe ich Ihnen nur vor Augen geführt, um zu zeigen, daß selbst ein weitest ausgebauter PC-1600 die Gesamtzahl aller indizierbaren Variablen nicht nutzen kann.

III.1.3. ERASE - Grenzen

ERASE kann nicht innerhalb einer FOR-NEXT-Schleife eingesetzt werden. Der Grund ist logisch: Der PC-1600 wird mit indizierten Variablen in die FOR-NEXT-Routine geschickt. Auch wenn Sie in der Schleife keine indizierten Variablen verarbeiten, bricht der Computer mit ERROR 109 ab, wenn in der Schleife ein ERASE steht, da er nicht "weiß", ob er in einer FOR-NEXT-Routine indizierte Variablen bearbeiten soll. Eine Dimensionierung kann deshalb nur nach einer FOR-NEXT-Routine durch ERASE aufgehoben werden.

III.1.4. IF-THEN-(ELSE)

- a) Der ELSE-Befehl sollte bei einer IF-Anweisung begrenzt eingesetzt werden. Jeder Befehl kostet be-
kanntlich Bearbeitungszeit (s. 1.2.2.) und sollte
daraus vermieden werden, wenn er überflüssig ist.

Sinnvoll ist der ELSE-Befehl, wenn alternativ ein bestimmter Wert
der einen oder anderen Variablen zugewiesen werden soll - oder eine
Variable alternativ einen bestimmten Wert annehmen soll, z.B.:

```
IF A=B LET AS=C ELSE LET BS=C
IF A=B LET A=C ELSE LET A=D
```

Sinnvoll ist der ELSE-Befehl auch, wenn Variablen besonders behan-
delt werden sollen, ehe sie durch das Hauptprogramm weiter verar-
beitet werden. In solchem Fall schiebt man zwischen die Hauptpro-
grammzeilen die Nebenroutine, die mit ELSE übersprungen wird, wenn
die Sonderbehandlung entfällt.

```
100: IF A<C GOTO 250 ELSE V=W:X=Z
101: .....
249: .....
250: B=C
```

Beachten Sie, daß ohne ELSE in Zeilen Befehle nach einem Doppel-
punkt nur beachtet werden, wenn die IF-Bedingung erfüllt ist. Ist
IF mit ELSE verbunden, werden Befehle nach einem Doppelpunkt nur
ausgeführt, wenn die IF-Bedingung nicht erfüllt ist - es sei denn,
IF und ELSE weisen nur Werte an Variablen durch LET zu.

- b) Anfängern bereitet der IF-Befehl mitunter Schwier-
igkeiten bei der Formulierung von alternativen
Prüfbestimmungen.

Soll ein Befehl ausgeführt werden, wenn IF eine
positive Bedingung erfüllt, sind die Alternativen
mit OR verknüpft werden. Ist die Bedingung negativ
formuliert, ist die Verknüpfung AND. Also immer:

```
IF A=B OR A=C LET D=E
IF A<>B AND A<>C LET D=E
```

III.1.5. INKEYS

Der INKEYS-Befehl wird vor allem für Menü- und CURSOR-Steuerung
benutzt (s.a. unter IV. - VI.) Da aber einer Variablen durch
INKEYS ein Wert nur zugewiesen wird, wenn im Augenblick der Anfor-
derung eine Taste gedrückt wird - anderenfalls bleibt der alte
Wert der Variablen erhalten, also vielfach 0, -, müssen Sie Vor-
sorge treffen, daß das Programm nicht ohne Ihre rechtzeitige
Tastenbetätigung fortgesetzt wird unter Bedingungen, die Sie nicht
wollen. Das erreichen Sie durch die Routine, daß die INKEYS-Zeile
so lange angelaufen wird, bis eine Taste oder bis eine bestimmte
Taste gedrückt wird. Z.B.:

```
100: A=INKEYS: IF A<>4 AND A<>6 GOTO 100
```

Wie unterschiedlich der INKEYS-Befehl optimal genutzt werden kann,
ist aus verschiedenen Beispielprogrammen in diesem Handbuch.

Wenn Sie den Befehl für eine Steuerung von Menüs, die auf dem Dis-
play angezeigt und durch Betätigung der Funktionstasten aktiviert
werden, benutzen wollen, ist der Wert ASC INKEYS der entsprechen-
den Taste zu benutzen (s. z.B. ADR-Teilprogramm 2, Zeile 41).

Da die ASCII-Werte der Funktionstasten in der Tabelle der Bedie-
nungsanleitung auf S. A-4 nicht angegeben sind, möchte ich sie
hier wiedergeben:

ASCII-Werte der Funktionstasten

	Hex(&)	dez.	Hex(&)	dez.
SHIFT	1	1	F1(!)	11
SMALL	2	2	F2(~)	12
CTRL	3	3	F3(#)	13
KBII	4	4	F4(\$)	14
BS	5	5	F5(%)	15
←	8	8	F6(&)	16
↕	9	9	CL	18
↓	A	10	RCL	19
↕	B	11	DEF	1B
↑	C	12	MODE	1F
→	D	13	SPACE	20
ENTER		13		32
OFF	F	15		

III.1.6. MERGE - Editierbarkeit aller Programme

Der PC-1600 erlaubt, alle Programme im Arbeitsspeicher einzeln zu editieren. Die Einzelheiten werden unter I.3. erläutert.

III.1.7. MOD

Da die MOD-Funktion mehr Bearbeitungszeit benötigt als die INTER-Routine, sollte man sie nur anwenden, wenn der Rest einer Division für den weiteren Programmverlauf wichtig ist. Vergleichen Sie hierzu auch S. IV-3.

III.1.8. ON ERROR GOTO (GOSUB)

Es empfiehlt sich, diesen Befehl nur in ein Programm einzubauen, wenn Variablen einen Wert annehmen können, der nicht erlaubt ist und zum Programmabbruch führen würde, wenn der Computer nicht durch ein RESUME oder eine Rückkehr zu einem bestimmten Ausgangspunkt aufgefordert wird, das Programm ohne Beachtung des Fehlers fortzusetzen.

Ich setze ON ERROR nur in zwei Fällen ein:

- a) bei mathematischen Funktionen die erfahrungsgemäß bei Veränderung der Variablen eine verbotene Division durch Null vornehmen müßten und anschließend weiterarbeiten sollen.
- b) bei Routinen, die bei negativem Ergebnis zum Programmbeginn zurückkehren aber vorher durch ERASE eine Löschung von Dimensionierungen erfordern, da zu Programmbeginn dimensioniert wird (vergleiche z.B. das Textverarbeitungsprogramm 2-4).

Beachten Sie, daß eine ON-ERROR-GOSUB-Meldung für das ganze weitere Programm gilt - es sei denn, Sie fordern durch einen erneuten Befehl den Computer auf, ein anderes RESUME zu ziehen.

III.1.9. RENUM als Fehlerkontrolle

Der RENUM-Befehl eignet sich nicht nur zur Neunummerierung von Programmen. Ich verwende ihn zur ersten Prüfung meiner neuen Programme auf Programmierfehler, bevor ich erst beim Probelauf durch eine ERROR-Meldung auf (weitere) Syntaxfehler aufmerksam gemacht werde.

Bei der Umsetzung einer Programmidee in das reale Programm, die ich meistens ohne vorherige schriftliche Fixierung im PRO-MODE des PC-1600 vornehme, kommt es leicht vor, daß ich einen Sprungbefehl übersehe. Da ich grundsätzlich beim Programmieren zunächst den Zeilenabstand 10 wähle, nummeriere ich nach vorübergehendem oder endgültigen Programmierabschluß neu. Habe ich Sprungbefehle übersehen, die der Computer wegen der fehlenden Zieladressen nicht neu berechnen kann, macht er durch

Undefined in <Zeilennummer>

darauf aufmerksam.

III.1.10. WAIT 0,S

Der Sinn des Attributs S ist, zu verhindern, daß bei WAIT 0 die Displayanzeige aus dem Display rasend nach oben entschwindet und nicht mitgelesen werden kann.

Wird das Attribut S dem WAIT 0 hinzugefügt, wird das Programm unterbrochen, wenn die vier Displayzeilen gefüllt sind. Mit der ENTER-Taste wird dann jeweils nur die nächste Displayzeile freigegeben, während die obere Zeile aus dem Display entschwindet.

Um das dauernde ENTER-Drücken zu vermeiden, ist PRINT-Anweisungen, die jeweils ein neues Displayfeld einleiten sollen, ein CLS vorzusetzen. Dann wird das Programm erst wieder nach weiteren vier Displayzeilen bzw. nach einem INPUT-Befehl unterbrochen.

III.2. Reservetastenbelegung

Es ist empfehlenswert, den Reservespeicher möglichst vollständig mit häufig benutzten BASIC-Befehlen zu belegen. Dabei ist zu berücksichtigen, daß die Tastatur unter dem Display (!-&) und die "QWERTYUIOP"-Reihe mit 16 Befehlen vorprogrammiert sind, die mit der Betätigung der DEF-Taste und der vorprogrammierten Taste (s. Abdeckschablone) aufgerufen werden.

Folgende Reservetastenbelegung ist für mich besonders vorteilhaft.

Reserve-Ebene I

F1:: F3:, F5:";CHR\$&E1;" F6::

Reserve-Ebene II

F1:MAXFILES =1:OPEN "X: F5:FOR OUTPUT AS #1 F6:FOR INPUT AS #1

Reserve-Ebene III

F1:PAPER C@ F2:PCONSOLE "LPT1:",78,2,2@
F3:PAPER C,30,890@ F4:PCONSOLE "LPT1:",52,2,0@
F5:DATA F6:RENUM

Zur Erklärung:

Auf der Ebene I habe ich die Funktionstasten F2 und F4 nicht belegt, weil ich die Zeichen " und \$ beim Programmieren immer wieder benötige - ebenso Semikolon, Komma und Doppelpunkt, die ich so mit einem einzigen Tastendruck abrufen kann, anstatt jedesmal noch die SHIFT-Taste betätigen zu müssen.

Auf der Ebene II habe ich Befehle für den Diskettenbetrieb untergebracht. Hier sind ", # und \$ wegen ihrer häufigen Nutzung nicht anders belegt.

Auf der Ebene III sind die Druckformatparameter, DATA und RENUM programmiert, die ich ebenfalls häufig verwende.

Es ist einfacher, selten benutzte Grundbelegungen der Funktionstasten bei Bedarf durch Betätigung der SHIFT-Taste und der betreffenden Funktionstaste zu erreichen als häufig genutzte Befehle immer wieder Zeichen für Zeichen einzutasten oder mit Hilfe der SHIFT-Taste zu aktivieren.

Die Neubelegung der %-Taste in der Ebene I ermöglicht, bei Textprogrammen das β auf genau so einfache Weise wie die übrigen deutschen Sonderzeichen (Umlaute) zu verwenden. Wird das β benötigt, wird nach Aktivierung der Ebene I nur die %-Taste gedrückt. Zwar steht dann in der Programmzeile ";CHR\$&E1;" - aber im gestarteten Programm (RUN) wird das β gedruckt. Beachten Sie aber: Steht das β am Ende einer Druckzeile, sind die beiden letzten Zeichen (; und ") in der Programmzeile zu löschen.

Damit Sie sich bei Bedarf sofort über die Neubelegung der Reservetasten informieren können, sollten Sie, wie in der Bedienungsanleitung unter 9-12 beschrieben, ein Funktionstastenmenü anfertigen.

In der Reserveebene I wird eingegeben:

" ; , SZ :"

Nach Umschalten in die Reserveebene II:

"MAXF OUT IN"

Der Menüstring in der Reserveebene III:

" DIN A4 DIN A5 DAT RENUM"

Danach wird der Reservespeicherinhalt auf Kassette gesichert - mit CSAVE "Reserve normal" im Reserve-Modus.

Auf Diskette ist eine Speicherung normalerweise nicht möglich, da SAVE nicht im Reserve-Modus funktioniert.

Wenn Sie die Maschinensprache beherrschen, wäre dies durch BSAVE des Reservespeichers durchführbar, aber unnötig:

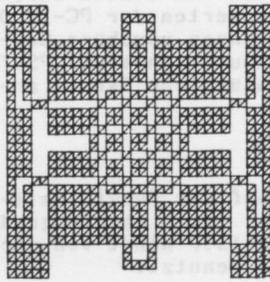
Weil der Reservespeicher einen Umfang von nur 188 Byte hat, wird ein kassettengesichertes Reservespeicherprogramm in wenigen Sekunden geladen. Schon aus diesem Grund sollten Sie die Reserve-Belegungsprogramme auf Band sichern. Außerdem gilt, was unter dem Abschnitt "Disketten- oder Kassettennutzung" (VI.3.4.) steht.

Wenn Sie verschiedene Reservetastenbelegungen benötigen, können Sie entsprechend viele Reservespeicherprogramme erstellen und auf sie schnell zurückgreifen.

Ein Reservespeicherprogramm ist z.B. zu BEEP in diesem Handbuch angegeben.

Aber denken Sie daran, daß Sie Reservespeicherprogramme nur im Reserve-Modus aufzeichnen oder laden. Sonst werden die Programme im Arbeitsspeicher zerstört.

Beim Erstellen eines Programms können Sie sich durch Drücken der RCL-Taste vergewissern, wie die Reserve-tasten belegt sind. Keine Panik, wenn dabei die aktive Programmzeile vom Display verschwindet. Nach erneutem Drücken der RCL-Taste erscheint die Programmzeile wieder. Sie wird vom Reserve-Menüstring nur vorübergehend überlagert, nicht gelöscht.



IV. Nutzung der LCD-Matrix

Die LCD-Matrix des PC-1600-Displays ist gut für Grafiken geeignet, da sie einen Umfang von 156x32 Displaypunkten hat.

Die Grafikbefehle für das Display werden in der Bedienungsanleitung ausreichend erläutert. Deshalb möchte ich in diesem Handbuch nur Hinweise geben, wie die Matrix des Display gut zu nutzen ist und hardwarebedingte Probleme zu lösen sind.

IV.1. Display-Auflösung

Das Hauptproblem, zufriedenstellende Grafiken auf dem Display zu erzeugen, liegt in der geringen Auflösung der Linien. Auflösung ist die Fähigkeit, Linien (und entsprechend andere Grafikeinzelheiten) möglichst genau wiederzugeben. Je mehr Punkte eine Matrix oder ein Raster hat, umso höher ist die Auflösung.

Hier sind nun bei einer Matrix von 156x32 Punkten unüberwindbare Grenzen gesetzt. Um so mehr gilt es, die Möglichkeiten optischer Täuschungen zu nutzen. Da das Auge verhältnismäßig träge ist, können uns Dinge optisch vorgetäuscht werden, die in Wirklichkeit nicht da sind. In Wirklichkeit gibt es z.B. keine laufenden Bilder auf der Leinwand oder dem Fernsehschirm. Nur der schnelle Wechsel der Einzelbilder täuscht Bewegung vor. Das wird z.B. auch bei LCD-Spielen genutzt, wie z.B. das Derby-Programm in diesem Handbuch zeigt.

Die geringe Auflösungsfähigkeit zeigt sich vor allem beim Zeichnen von Linien auf dem Display, die nicht senkrecht oder waagerecht verlaufen.

Lassen Sie z.B. durch LINE(0,0)-(155,31) den PC-1600 eine Diagonale über das ganze Display ziehen, entsteht eine Zickzacklinie, die zwar wegen ihrer Symmetrie optisch schön aussieht, aber eben keine Gerade ist.

Die Geradenzeichnung sieht optisch zufriedenstellender aus, wenn Sie den LINE-Befehl mit einem Bitmuster-Attribut versehen.

Probieren Sie das Programmbeispiel IV.1 zur Erprobung der Bitmuster (BM) aus - bei jeweiligem ENTER werden die Diagonalen mit und ohne Bitmuster gezeichnet. Mir gefallen die Bitmuster &5555 und &8888 am besten. Aber das ist Geschmacksache.

..... Bitmuster &8888

Bei einer Grafik, in denen die einzelnen Punkte durch mathematische Funktionen errechnet werden, hilft man sich am besten durch den PSET-Befehl. Da bei gebrochenen Koordinatenwerten der PC-1600 immer abrundet, d.h. den INTEGER-Wert bildet, können unschöne Grafikpunkte vermieden werden, indem man den Computer anweist, PSET nur auszuführen, wenn die Koordinatenwerte natürliche Zahlen annehmen:

```
IF X=INTX AND Y=INTY PSET (x,y)
```

Dann werden nur die Punkte gesetzt, die exakt auf der Funktionslinie liegen. Ob das im einzelnen sinnvoll ist, d.h. ob genügend Punkte für die Erkennbarkeit der Funktionslinie ist, hängt von der Funktion ab. Gegebenenfalls wird der MOD-Befehl genutzt:

```
1000: IF X=INTX AND Y=INTY GOTO 1003
1001: IF X<>INTX LET M=XMOD10:IF M>4 LET X=INTX+1
1002: IF Y<>INTY LET M=YMOD10:IF M>4 LET y=INTY+1
1003: PSET (X,Y)
```

In diesem Beispiel werden alle X/Y-Werte, die Brüche sind, abgerundet, wenn die erste Ziffer nach dem Komma kleiner als 5 ist, anderenfalls aufgerundet. Eine Abrundung braucht nicht programmiert zu werden, weil sie automatisch vollzogen wird.

IV.2. Bitmuster-Schraffur

Mit dem Bitmuster- in Kombination mit dem BF-Attribut können Sie Flächen auf dem Display schraffieren. Mit dem Beispiel IV.2 bietet Ihnen der Computer alle möglichen Bitmuster zu Auswahl an. Gefällt Ihnen eines der jeweils 4 Bitmuster auf dem Display, notieren Sie sich die Bitmusterzahl, die in Ziffern des Dezimalsystems über dem jeweiligen Muster angegeben ist.



IV.3. Figuren und Figuren in Bewegung

Das Display des PC-1600 reizt dazu, Displayspiele zu erdenken. Das wichtigste bei Spielen sind Figuren, die sich nach Möglichkeit auch noch bewegen sollen.

IV.3.1. Figuren

Figuren werden in Hexstrings codiert. Am besten nimmt man dazu normales kariertes Papier, das in Reihen à 8 Kästchen Höhe eingeteilt wird, da eine Displayzeile aus 156 Spalten mit jeweils 8 vertikalen Punkten besteht. In die Reihen werden die gewünschten Figuren eingezeichnet.

Um die Zeichnung in den Hexstring zu übertragen, halbieren Sie zunächst die Reihe horizontal, so daß in jeder Spalte vier Kästchen über und unter der Halbierungslinie stehen.

Je vier übereinanderstehende Kästchen ergeben das Muster für einen Wert im Hexadezimalsystem. Nehmen Sie dazu die Tabelle auf Seite 1-3 zu Hilfe. In den Binärreihen sind die Nullen jeweils leere, die Einsen ausgefüllte Karos. In den Hexstring ist zunächst immer der Hexwert der unteren Karos, dann der Wert der oberen einzutragen. Sind alle vier Karos leer, muß die 0 als Hexwert eingetragen werden. Wird sie unterschlagen, werden andere Grafikzeichen gebildet und das Programm durch eine ERROR-Meldung abgebrochen, wenn im Hexstring eine ungerade Anzahl von Zeichen steht, da die Spalte jeder Displayzeile aus zwei Hexziffern zusammengesetzt ist.

Das ganze möchte ich Ihnen am Beispiel der Pferdefigur 1 aus dem Derby-Programm Zeile 19 darstellen:

Die ausgefüllten Kästchen einer Reihe sind mit X, die leeren Kästchen mit . wiedergegeben. Die Halbierungslinie ist gestrichelt. Unter den einzelnen Spalten ist der Hexwert der Spalte, ebenfalls durch die Halbierung getrennt, angegeben.

```

.....XX..XX..
.....X..XXXX
.....XX..XX..
.....XXXXXXXXX
-----
..XXXXXXXXXX..
....X.X....X..
....X.X.....X.
...XX.....X

```

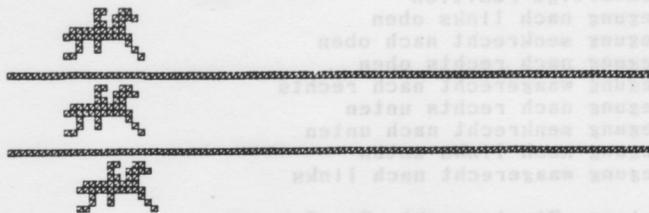
0008888FD8EF320

009971171111348

PF\$(0)=""000090987818187F1D181E1F334280"

Mit dem Befehl GPRINT"000090987818187F1D181E1F334280" wird diese Figur auf das Display gezaubert.

(GPRINT PF\$(0) funktioniert natürlich nur, wenn wie im Derby-Programm PF\$(0) als Array dimensioniert wurde).



IV.3.2. sich bewegende Figuren

Vielleicht werden Sie sich gewundert haben, daß zu Beginn des Hexstrings zwei Leerspalten (0000) stehen. Es wäre überflüssig, wenn Sie die Figur nur als Standbild "errichten" wollen. Aber schließlich soll das Pferd ja über das Display galoppieren. Sie werden sehen, warum dafür die beiden Leerspalten notwendig sind.

Bewegungen der verschiedensten Art erreichen Sie durch Veränderung der Variablen folgender Musterzeile:

```

WAIT T: FOR I=0TO B STEP SB:FOR J=0TOH STEP SH: GCUR-
SOR B,H: GPRINT PF$(0):NEXT H: NEXT B

```

B = Breite des Display H = Höhe des Display
 SB = Sprungweite SH = Sprunghöhe
 T = Zeit

Je größer die Variable T ist, um so abgehackter ist die Bewegung. Die Veränderung der Variablen B bewegt die Figur horizontal über das Display - nach rechts, wenn SB positiv, nach links, wenn SB negativ ist. Je größer SB ist, um so sprunghafter läuft die Figur. Ist SB=1, läuft sie ganz gemächlich über das Display, d.h. sie wird jeweils um eine Spalte verschoben. Wählen Sie B größer als 155, läuft die Figur aus dem Display.

Entsprechendes gilt für die Variable H, die die Figur vertikal über das Display bewegt. Werden B und H verändert, wird die Figur diagonal bewegt.

Werden einzelne Variablen nicht gebraucht, weil sie nicht verändert werden, werden die Programmschritte fortgelassen, wie im Derbyprogramm, in dem die Pferde nur waagrecht über das Display galoppieren.

Lassen Sie in dem Pferdestring nun die ersten vier Nullen weg, werden Sie feststellen, daß Reste des Pferdes auf der jeweiligen vorhergehenden Position zurückbleiben, um so mehr, je größer SB (bzw. im Derbyprogramm UU, VV und WW) ist. Stellen Sie bei Ihren Figuren diese Erscheinung fest, sollten Sie ausprobieren, wieviele Nullen - jeweils als Paar - vor oder nach dem eigentlichen Figurenstring stehen müssen, die als LösCHFunktion gelten.

Lassen Sie die Figur nur als einen einzigen Hexstring über das Display wandern, wird sie wie eine tote Figur bewegt. Um sie zum Leben zu erwecken, müssen mehrere Zeichnungen in verschiedenen Stellungen gezeichnet und in Hexstrings umgesetzt werden. Meistens genügen zwei Stellungen, wie das Derby-Programm demonstriert.

IV.3.3. handgesteuerte Bewegungen

Man kann Figuren oder den GCURSOR programmgesteuert - wie im Derby-Programm - über das Display bewegen. Sie können dies aber auch manuell tun.

Im Manugraf-Programm können Sie nachvollziehen, wie so etwas gemacht wird.

Im allgemeinen - auch bei anderen Computern, deren Tastatur ähnlich wie beim PC-1600 konzipiert ist - hat sich eingebürgert, für die manuelle Steuerung in alle acht möglichen Richtungen die Zahlentasten um die 5 herum zu nehmen. Die Bewegung wird mit der INKEYS-Funktion gesteuert. Dabei sind dann folgende Positionen mit der Betätigung der Zahlentasten verbunden:

5	(X,Y)	gegenwärtige Position
7	(X-1,Y+1)	Bewegung nach links oben
8	(X,Y+1)	Bewegung senkrecht nach oben
9	(X+1,Y+1)	Bewegung nach rechts oben
6	(X+1,Y)	Bewegung waagerecht nach rechts
3	(X+1,Y-1)	Bewegung nach rechts unten
2	(X,Y-1)	Bewegung senkrecht nach unten
1	(X-1,Y-1)	Bewegung nach links unten
4	(X-1,Y)	Bewegung waagerecht nach links

Die Bewegung erfolgt jeweils um einen Displaypunkt. Die Schrittweite kann entsprechend verändert werden, wenn die Zahl 1 geändert wird.

Zum Manugraf-Programm ist noch anzumerken:

Durch zwei gleichartige Routinen werden die Displaypunkte gesetzt oder gelöscht. Wird die Zahlentaste 5 betätigt, wird die LösCHFunktion aktiviert. Der PSET-Modus wird durch Drücken der Null-Taste reaktiviert.

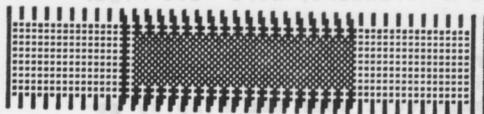
Bei Drücken der Punkt-Taste (.) wird die gegenwärtige Position des GCURSORs durch Blinken angezeigt.

IV.3.4. Automatische Computergrafiken

Auf einigen Seiten dieses Handbuches sind Grafiken abgebildet, die der PC-1600 automatisch erzeugt hat.

Das Listing finden Sie unter Beispiel IV.6 (automatische Computergrafik).

Sie erzielen unterschiedliche Wirkungen, wenn Sie z.B. das BF durch B ersetzen, das Attribut X weglassen oder nach dem zweiten Komma ein Bitmuster eintragen.



Bevor Sie nun die angefügten Display-Grafikprogramme ausprobieren - eines ist übrigens als Teilprogramm im Statistik-Programm (Beispiel V.3.) enthalten -, sollten Sie sich mit der Möglichkeit vertraut machen, Display-Grafiken auf Papier zu kopieren.

IV.4. Display-Kopierung - Hardcopy

Gelungene Grafiken oder spannende Spielsituationen sind auf Papier übertragbar, wenn Sie den CE-1600P angeschlossen haben - oder einen anderen Drucker, für den Sie gegebenenfalls die LLINE-Befehle nach dessen Bedienungsanleitung verändern müssen.

Probieren Sie das folgende Programm aus, mit dem das Fischelmännchen gezeichnet wird.

Das eigentliche Kopierprogramm ist in den Zeilen 6-20 untergebracht. Die Parameter sind immer vor der Aktivierung des Grafikprogramms einzugeben. Deshalb ist es immer mit DEF D einzuleiten, da das Kopierprogramm normalerweise durch MERGE mit dem Grafikprogramm verbunden wird.

Zunächst werden die Ausgangskordinaten für den GLCURSOR eingegeben. Sie bestimmen die Ausgangsposition des Zeichenstiftes auf dem Papier.

Danach werden die Vergrößerungsfaktoren für das ganze Display und für die einzelnen Displaypunkte eingegeben. Je größer der Unterschied zwischen dem Vergrößerungsfaktor V und dem Abbildungsmaßstab der Punkte W ist, um so blasser wird die Kopie, da Leerräume zwischen die einzelnen Punkte gesetzt werden. Der Faktor V=6 ist für DIN-A4-Breite der Kopie vorgegeben.

Ihrem Grafik-Programm setzen Sie die Zeile

```
1: "Programm"
```

voran. Sie können natürlich auch eine andere Sprungmarke verwenden. Verwenden Sie Einzelzeichen als Marke, müssen Sie sich vergewissern, daß sie nicht schon als Sprungmarke innerhalb des Programms verwendet wurden.

Vergessen Sie nicht, einen möglichen CLEAR-Befehl zu Beginn Ihres Grafikprogramms zu löschen. Sonst wird nichts kopiert, da auch die Parameter des Kopierprogramms auf Null gesetzt werden. Eventuell sind im Kopierprogramm auch andere Variablenamen zu wählen, wenn sie mit Variablen des Grafikprogramms identisch sind und eine ordnungsgemäße Kopie verhindern.

Die Zeile, die ein Display-Bild beendet, ist durch den Befehl

```
GOSUB "Abdruck"
```

zu ergänzen.

Sollen mehrere Displaybilder, die durch FOR-NEXT-Schleifen entstehen, kopiert werden, ist die Zeile mit dem GOSUB "Abdruck" so zu formulieren:

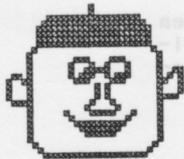
```
GOSUB"Abdruck": GOSUB "Neues DISPLAY"
```

Dann wird, ehe ein neues Bild erzeugt wird, das Papier um eine Kopiebreite transportiert.

Möchten Sie Momentaufnahmen von Spielen anfertigen, ist statt der GOSUB"Abdruck"-Zeile im Spielprogramm vor der Zeile, die neue Bewegungen auf dem Display erzeugt, einzufügen:

```
IF ASC INKEY$ = 13 GOSUB "Abdruck"
```

Dann wird durch die ENTER-Taste das Spiel unterbrochen und das Bild kopiert. Da das Display Punkt für Punkt abgetastet wird, kann bis zum Beginn der Kopie einige Zeit vergehen.



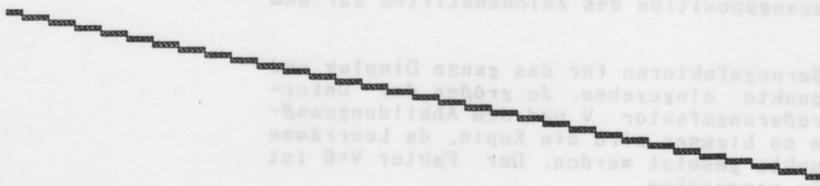
Manugraf

Beispiel I U . 1

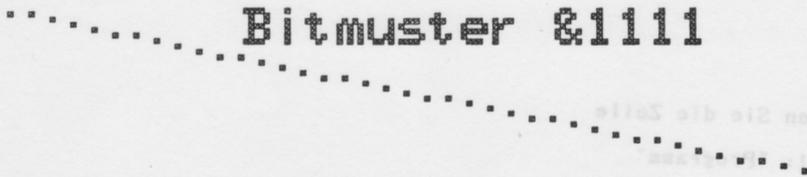
```

1:'      Display-Auflösung
2:'
3:READ BM:WAIT :CLS :LINE (0,0)-(155,31)
4:WAIT 0:CLS :CURSOR 8,0:PRINT "Bitmuster &";HEX$ (BM)
5:WAIT :LINE (0,0)-(155,31),,BM:GOTO 3
6:DATA &1111,&2222,&3333,&4444,&5555,&6666,&7777,&8888
7:DATA &9999,&AAAA,&BBBB,&CCCC,&DDDD,&EEEE,&FFFF
8:END
    
```

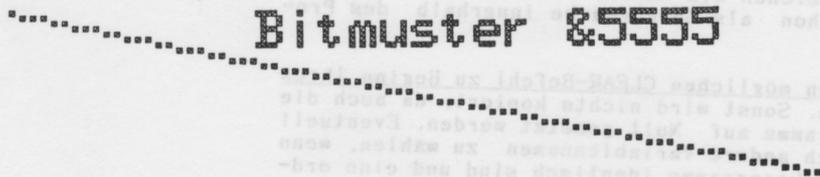
ohne Bitmuster



Bitmuster &1111



Bitmuster &5555



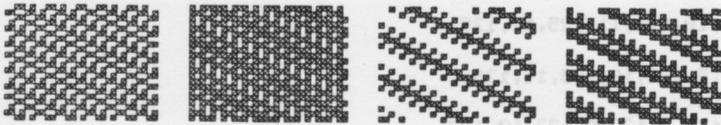
Beispiel I U . 2

```

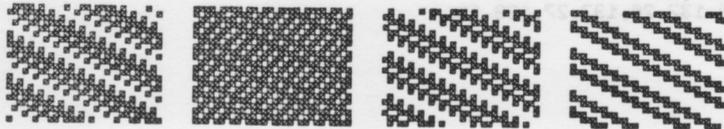
1:'      Bitmuster für Display-Schraffuren
2:'
3:CLS :WAIT 100
4:CLS :RANDOM :C=RND 65535:D=RND 65535:E=RND 65535:F=RND 65535
5:PRINT USING "#####";C;D;E;F;
6:LINE (6,10)-(35,31),,C,BF
7:LINE (42,10)-(71,31),,D,BF
8:LINE (78,10)-(107,31),,E,BF
9:LINE (114,10)-(143,31),,F,BF
10:GOTO 4:END
    
```



13523 48875 10016 33742



31251 63414 16743 57400



Beispiel I U . 3

```

1: Display-Kopierung (Hardcopy) des
2: Fischelmännchens
3:
4: Parametervorgaben für die Kopie
5:
6: CLEAR :PRINT "Display-Kopierung"
7: CLS :CX=0:PRINT "X=0":CURSOR 5,2:INPUT "Startkoordinate X ";
  CX
8: CLS :CY=0:PRINT "Y=0":CURSOR 5,2:INPUT "Startkoordinate Y ";
  CY
9: CLS :C=0:PRINT "Farbe=0":CURSOR 10,2:INPUT "Farbe ";C
10: CLS :U=6:PRINT "Faktor=6":CURSOR 2,2:INPUT "Vergrößerungs-F
  aktor ";U
11: CLS :W=U:CURSOR 5,2:PRINT "Punktgröße <=";W;" ";:INPUT W
12: IF W<1LET W=1
13: GOTO "Programm"
14: "Abdruck"GRAPH :COLOR C
15: GLCURSOR (CX,CY):SORGN
16: FOR JX=0TO 155
17: FOR JY=0TO 31
18: P=POINT (JX,JY)
19: IF PGLCURSOR (U*JX,-U*JY):RLINE -(W,0)-(0,W)-(-W,0)-(0,-W)-
  W,W)
20: NEXT JY:NEXT JX:LLINE -(0,0),9:TEXT :RETURN
21: "Programm":WAIT 0:CLS
22: PRINT " Beispiel
23: PRINT " für
24: PRINT "Displaykopierung:
25: PRINT " Fischelmännchen
26: FOR I=1TO 192
27: READ X,Y
28: PSET (X,Y)
29: NEXT I
30: DATA 110,31,111,30,112,30,113,20,113,21,113,29,114,18,114,19
  ,114,22,114,29
31: DATA 115,8,115,9,115,10,115,11,115,12,115,13,115,14,115,15,1
  15,16,115,17
32: DATA 115,22,115,28,116,6,116,7,116,10,116,13,116,14,116,16
33: DATA 116,22,116,23,116,24,116,27,117,3,117,4,117,5,117,11,11
  7,12,117,13
34: DATA 117,14,117,15,117,22,117,23,117,25,117,27,118,2,118,3,1
  18,4,118,5
35: DATA 118,13,118,21,118,24,118,26,118,27,119,2,119,3,119,4,11
  9,6,119,10
36: DATA 119,11,119,12,119,13,119,14,119,15,119,18,119,19,119,20
  ,119,24,119,26
37: DATA 119,27,119,28,120,1,120,2,120,3,120,5,120,10,120,13,120
  ,14,120,16
38: DATA 120,23,120,26,120,28,120,29,121,1,121,2,121,3,121,4,121
  ,5,121,10
39: DATA 121,13,121,14,121,17,121,22,121,28,121,30,121,31,122,1,
  122,2,122,4
40: DATA 122,11,122,17,122,27,123,1,123,2,123,3,123,4,123,5,123,
  6,123,12
41: DATA 123,13,123,14,123,15,123,16,123,30,123,31,124,1,124,2,1
  
```

24,3

DER FACHVERLAG FÜR
TASCHEN-COMPUTER !

```

42:DATA 124,4,124,5,124,7,124,8,124,9,124,12,124,28,124,29,125,
3,125,4
43:DATA 125,5,125,6,125,7,125,8,125,9,125,10,125,11,125,12,125,
26,125,27
44:DATA 126,4,126,5,126,6,126,7,126,8,126,9,126,10,126,11,126,1
2,126,13
45:DATA 126,18,126,24,126,25,127,6,127,7,127,8,127,9,127,10,127
,11
46:DATA 127,12,127,13,127,14,127,15,127,16,127,17,127,21,127,22
,127,23,128,14
47:DATA 127,15,128,16,128,17,128,18,128,19,128,20,129,21,129,22
,130,23,131,23
48:DATA 132,24,133,24,134,25,135,25,136,26,137,26,137,27,138,28
,138,29,138,30,138,31
49:DATA 138,27,138,28,139,29,139,30,139,31
50:FOR J=0TO 31:FOR K=0TO 100
51:PSET (K,J),X
52:NEXT K:NEXT J
53:GOSUB "Abdruck":END
    
```

**Beispiel
für
Displaykoeffizient:
Fischelmannchen**



Beispiel I U . 4

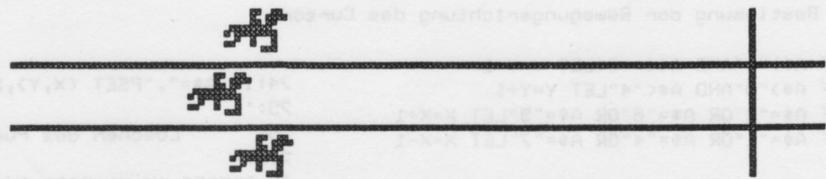
```

1:'      DERBY mit Totalisator"
2:'
3: CLEAR :CLS :WAIT 0
4:'
5:'      Namen der Spieler und Wettannahme
6:'
7: INPUT "Zahl der Mitspieler";M
8:CLS :DIM PF$(1)*30,SP$(M),T1(M),T2(M)
9:FOR I=1TO M
10:PRINT "T I P E I N G A B E"
11:INPUT "Name: ";SP$(I)
12:INPUT "Welches von 3 Pferden wird 1. ";T1(I):IF T1(I)<1OR T1
(I)>3GOTO 12
13:INPUT "Welches Pferd wird 2. ";T2(I):IF T2(I)<1OR T2(I)=T1(I
)OR T2(I)>3GOTO 13
14:NEXT I
15:WAIT 0,S
16:'
17:'      Displaygrafik zweierPferdestellungen
18:'
19:PF$(0)="000090987818187F10181E1F334280"
20:PF$(1)="0000050483818187F10989EF70302"
21:'
22:'      Strings zur Mitteilung des Einlaufs
23:'
24:S1$="Sieger: Bahn":S2$="Zweiter: Bahn"
25:'
26:'      Rennbahnzeichnung
27:'
28:CLS :LINE (0,10)-(155,10)
29:LINE (0,22)-(155,22)
30:LINE (142,0)-(142,31)
31:'
32:'      R e n n e n
33:'      mit akustischem Pferdegetrappel
34:'
35:FOR I=0TO 140
36:BEEP 1,15,10:GDCURSOR UU,0:GPRINT PF$(0)
37:BEEP 1,20,10:GDCURSOR UU,0:GPRINT PF$(1)
38:BEEP 1,15,10:GDCURSOR UU,12:GPRINT PF$(0)
39:BEEP 1,20,10:GDCURSOR UU,12:GPRINT PF$(1)
40:BEEP 1,15,10:GDCURSOR WW,24:GPRINT PF$(0)
41:BEEP 1,20,10:GDCURSOR WW,24:GPRINT PF$(1)
42:'
43:'      Zufallsbeschleunigung der Pferde
44:'      zwischen 1 und 3 Punkten bei jedem Schritt
45:'
    
```

```

46:RANDOM
47:V=RND 3:U=RND 3:W=RND 3
48:UU=UU+U:VV=UU+V:WW=WW+W
49:IF UU>=142OR VV>=142OR WW>=142GOTO 54
50:NEXT I
51:'
52:' Feststellung der Reihenfolge beim Einlauf
53:'
54:CURSOR 0,0
55:IF UU>VVAND UU>WWPRINT S1$;" 1":UU=1:GOTO 58
56:IF UU>VVAND UU>WWPRINT S1$;" 2":UU=1:GOTO 59
57:IF WW>VVAND WW>UUPRINT S1$;" 3":WW=1:GOTO 60
58:CURSOR 0,1:IF UU>WWPRINT S2$;" 2"ELSE PRINT S2$;" 3":IF UU>W
WLET UU=2ELSE LET WW=2:GOTO 64
59:CURSOR 0,1:IF UU>WWPRINT S2$;" 1"ELSE PRINT S2$;" 3":IF UU>W
WLET UU=2ELSE LET WW=2:GOTO 64
60:CURSOR 0,1:IF UU>VVPRINT S2$;" 1"ELSE PRINT S2$;" 2":IF UU>V
WLET UU=2ELSE LET VV=2
61:'
62:' Totalisatorauswertung
63:'
64:CURSOR 0,2:FOR I=1TO M
65:ON T1(I)GOTO 66,76,86
66:ON T2(I)GOTO 67,68,72
67:' nicht genutzt
68:IF UU=1AND VV=2GOTO 104
69:IF UU=1AND VV>2GOTO 105
70:IF UU>1AND VV=2GOTO 106
71:IF UU>1AND VV>2GOTO 107
72:IF UU=1AND WW=2GOTO 104
73:IF UU=1AND WW>2GOTO 105
74:IF UU>1AND WW=2GOTO 106
75:IF UU>1AND WW>2GOTO 107
76:ON T2(I)GOTO 77,81,82
77:IF VV=1AND UU=2GOTO 104
78:IF VV=1AND UU>2GOTO 105
79:IF VV>1AND UU=2GOTO 106
80:IF VV>1AND UU>2GOTO 107
81:' nicht genutzt
82:IF VV=1AND WW=2GOTO 104
83:IF VV=1AND WW>2GOTO 105
84:IF VV>1AND WW=2GOTO 106
85:IF VV>1AND WW>2GOTO 107
86:ON T2(I)GOTO 87,91,95
87:IF WW=1AND VV=2GOTO 104
88:IF WW=1AND VV>2GOTO 105
89:IF WW>1AND VV=2GOTO 106
90:IF WW>1AND VV>2GOTO 107
91:IF WW=1AND UU=2GOTO 104
92:IF WW=1AND VV>2GOTO 104
93:IF WW>1AND VV=2GOTO 106
94:IF WW>1AND VV>2GOTO 107
95:' nicht genutzt
96:'
97:' Mitteilung über die Gewinne:
98:'
99:' 100 P.: 1. + 2. Platz richtig vorausgesagt
100:' 50 P.: 1. Platz richtig vorausgesagt
101:' 10 P.: 2. Platz richtig vorausgesagt
102:' 0 P.: keinen Platz richtig vorausgesagt
103:'
104:PRINT SP$(I);" hat 100 Punkte.":NEXT I:END
105:PRINT SP$(I);" hat 50 Punkte.":NEXT I:END
106:PRINT SP$(I);" hat 10 Punkte.":NEXT I:END
107:PRINT SP$(I);" hat 0 Punkte.":NEXT I
108:END

```



Sieger: Bahn 3 
Zweiter: Bahn 1 
Peter hat 50 Punkte. 
Susanne hat 0 Punkte. 

Beispiel I U . 5

```

1:'          M A N U G R A F
2:'
3:'  handgesteuertes Display-Grafikprogramm
4:'
5:'
6:'          Urspann
7:'          und
8:'  Bestimmung der Ausgangsposition
9:'  des Grafikcursors
10:'
11:"Programm":WAIT 0
12:CLS :WAIT 100:CURSOR 10,1:PRINT "Grafik":CLS :X=78:Y=16:WAIT
   0
13:PRINT "Ausgangspunkt anders als DISPLAY-Mitte? ":INPUT "X-K
   oordinate: ";X
14:INPUT "Y-Koordinate: ";Y
15:CLS
16:PSET (X,Y)
17:'
18:'          Bearbeitungsrountinen
19:'
20:A$=""
21:'
22:'          Einsprung ins Hardcopy-Programm
23:'
24:A$=INKEY$ :IF A$="A"GOTO "Abdruck"
25:'
26:'          Einsprung zur Punktlösch-Routine
27:'
28:IF A$="5"GOTO 55ELSE GOTO 32
29:'
30:' Bestimmung der Bewegungsrichtung des Cursors
31:'
32:IF A$>"6"AND A$<="9"LET Y=Y-1
33:IF A$>"0"AND A$<"4"LET Y=Y+1
34:IF A$="3"OR A$="6"OR A$="9"LET X=X+1
35:IF A$="1"OR A$="4"OR A$="7"LET X=X-1
36:'
37:'          Bestimmung der
38:'          Display-Ränder als Cursorgrenzen
39:'
40:IF X<0LET X=0
41:IF Y<0LET Y=0
42:IF X>155LET X=155
43:IF Y>31LET Y=31
44:'
45:'          Blinken des Punktes zur Positionsanzeige
46:'
47:IF A$="."PSET (X,Y),X:GOTO 20
48:'
49:'          Fixierung des Punktes auf dem Display
50:'
51:PSET (X,Y):GOTO 20
52:'
53:'          Punkt-Lösch-Routine
54:'
55:A$=""
56:'
57:'          Einsprung in die Punkt-Fixier-Routine
58:'
59:A$=INKEY$ :IF A$="0"GOTO 20
60:'
61:' Bestimmung der Bewegung des Lösck-Cursors
62:'
63:IF A$>"6"AND A$<="9"LET Y=Y-1
64:IF A$>"0"AND A$<"4"LET Y=Y+1
65:IF A$="3"OR A$="6"OR A$="9"LET X=X+1
66:IF A$="1"OR A$="4"OR A$="7"LET X=X-1
67:IF X<0LET X=0
68:IF Y<0LET Y=0
69:IF X>155LET X=155
70:IF Y>31LET Y=31
71:'
72:'          blinkende Lösck-Cursor-Anzeige
73:'

```

Beispiel I U . 6

```

1:'          automatische Computer-Display-Grafik
2:'
3:"Programm"CLS :WAIT 60
4:PRINT "          Grafikformat":B=155:INPUT "andere Breite als 1
   55: ";B:IF B<50R B>155GOTO 4
5:H=31:INPUT "andere Höhe als 31: ";H:IF H<50R H>31GOTO 5
6:CLS
7:LINE (Z,Y)-(-Z+B,-Y+H),X,,BF
8:RANDOM :Z=RND B:Y=RND H
9:IF ASC INKEY$ =13GOSUB "Abdruck"ELSE GOTO 7

```

V. Nutzung des CE-1600P (Drucker)

Der Drucker des PC-1600-Systems ist vermutlich die erste und eine ideale Ergänzung des Computers. Trotzdem sollten Sie unter II.7 die Überlegungen nachlesen, den CE-515P(516P) zusätzlich zu nutzen, wenn Sie ihn schon besitzen oder Sie ihn günstig erwerben können.

V.1.1. Formatierung der Druckseiten

In den Drucker sollten Sie grundsätzlich nur Endlos- oder Einzelpapier in DIN-A4-Breite einführen. Schmaleres Papier wird nicht gleichmäßig durchgezogen. Das wäre schon beim Druck normaler Texte lästig. Eine Nutzung der sehr guten Grafikqualität des PC-1600P - Genauigkeit von 0,2 mm - ist bei schmalere[m] Papier unmöglich.

Trotzdem brauchen Sie auf andere Druckformate nicht zu verzichten. Es entsteht nur durch das Zurechtschneiden des bedruckten Papiers auf das gewünschte Format Papierabfall. Bei der zum DIN-A4-Format meistgenutzten Alternative DIN-A5 können Sie den Abfall dadurch verringern, daß Sie Endlospapier (z.B. Rollenpapier) verwenden. Sie brauchen dann nur noch die Breite beschneiden.

Es gibt zwar die Möglichkeit, Abfall zu vermeiden, indem durch ROTATE 1 oder ROTATE 3 die Druckrichtung des Plotters verändert wird, weil die Papierbreite des DIN-A4-Formats der Papierlänge des DIN-A5-Formats entspricht. Aber schon die Schwierigkeit, Texte beim Ausdrucken mitzulesen, steht in keinem Verhältnis zu den geringen Mehrkosten, die durch den Abfall entstehen. Außerdem ist zu beachten, daß die Druckgenauigkeit an Qualität verliert, wenn das Papier zu oft und in größerem Umfang vertikal (LF-Wirkung) hin und her transportiert wird.

Um das Drucken schnell jeweils auf DIN-A4- oder auf DIN-A5-Format umzustellen, habe ich in der Reserveebene 3 die ersten vier Funktionstasten mit den entsprechenden Befehlen programmiert, die ich bei Bedarf abrufe:

DIN-A4-Format

F1: PAPER C@ - F2: PCONSOLE"LPT1: ",78,2,2@

DIN-A5-Format

F3: PAPER C,30,890@ - F4: PCONSOLE "LPT1: ",52,2,0@

Im DIN-A4-Format werden so 56 Zeilen á 78 Zeichen mit einem linken Rand von 2 Zeichen Breite gedruckt - im DIN-A5-Format 37 Zeilen á 52 Zeichen ohne Extra-Rand.

Bei Bedarf können Sie Zeichen- und Zeilenzahl anders bestimmen.

V.1.2. Überwindung der Abschneidfunktion des CE-1600P

Um das Beschreiben der Papierwalze ohne Papier zu verhindern, was auf Dauer die Walze beschädigen und vor allem die Grafikfunktion in ihrer Genauigkeit beeinträchtigen würde, besitzt der CE-1600P eine Abschneidfunktion. Entsprechend dem PAPER-Befehl wird nach Erreichen der letzten formatierten Druckzeile der Farbstift angehoben.

Auch bei Rollen- und Endlospapier tritt die Abschneidfunktion ein. Sie können sie durch einen kleinen Trick außer Kraft setzen:

Setzen Sie in Ihre Programme eine Routine, die die ausgedruckten Zeilen mitzählt. In ADR 3, Zeile 42 können Sie sehen, wie das gemacht wird. An Stelle des STOP-Befehls geben Sie die Druckformat-Befehle für DIN-A4-Format ein. Dann wird die Abschneidfunktion für jeweils 56 Zeilen außer Kraft gesetzt.

V.1.3. Fettdruck, Sperrdruck

Fett- und Sperrdruck sind durch den PITCH-Befehl leicht zu erreichen. Das zeigt das Beispiel V.1. Probieren Sie es aus, indem Sie vor allem verschiedene PITCH- und Fettdruck-Faktoren eingeben.

Je höher die Pitch-Zahl, um so größer die Sperrdruck-Breite; je höher F, um so fetter der Druck.

V.2. Grafiken mit dem CE-1600P

Seine besonderen Qualitäten zeigt der CE-1600P als Matrixdrucker. Matrixdrucker besagt, daß er auch Grafiken verarbeiten und -farbig - darstellen kann.

Die entsprechenden Grafikbefehle werden in der Bedienungsanleitung umfassend erläutert. Wenn Ihnen in den folgenden Beispielen in dieser Hinsicht etwas unklar ist, sollten Sie die Bedienungsanleitung zu Rate ziehen. Oft genügt es, den Programmablauf im TRON-Verfahren zu verfolgen, um die Wirkung bestimmter Grafikbefehle zu verfolgen. Im übrigen habe ich versucht, die einzelnen Programmschritte innerhalb der Programme durch Kommentarzeilen zu erläutern.

Doch zuvor möchte ich einige grundsätzliche Hinweise geben. Sie können sie am Beispiel "Selbstportät des PC-1600" nachvollziehen.

V.2.1. Übertragung von Grafiken in Programme

1. Am besten besorgen Sie sich transparentes Millimeterpapier. Weil die Anzahl der Schritte des Farbstiftes in der Breite des DIN-A4-Papiers 960, in der Länge 1354 beträgt, können Sie die einzelnen Millimeterlinien in Relation zu den Schritten setzen und entsprechend beschriften.
2. Die Schnittpunkte der Millimeterlinien ergeben die X- und Y-Koordinatenwerte.
3. Sie übertragen nun die Schnittpunkte, die der Farbstift beim Linienziehen jeweils anlaufen soll, als Datawerte in ihr Programm. Dabei ist zu beachten:
 - a) Angegeben wird jeweils der X- und der Y-Wert - Sie können die Werte auch vertauschen, um zu erreichen, daß eine Breitzeichnung auf Papier im Längsformat gezeichnet wird.
 - b) Ausgangspunkt aller Linienzüge, die jeweils ohne Unterbrechung gezeichnet werden, ist zu Beginn der Datazeilen und in jeder Datazeile nach einem RESTORE-Befehl (z.B. in Beispiel V.2 die Zeilen 78, 91, 99, 107 usw.) der X/Y-Wert, der als GLCURSOR-Ausgangspunkt angesteuert wird (s. Zeilen 8+9).
 - c) Nach Beendigung ununterbrochener Linienzüge werden die notierten DATA-Werte gezählt, durch 2 geteilt (X,Y), um 1 vermindert (da GLCURSOR-Ausgangspunkt) und die ermittelte Zahl als Variable Z (Zeilen 10 + 11) nach den ersten beiden Werten einer RESTORE-Zeile (vergleiche die in b) angegebenen Zeilen) als dritte Zahl eingetragen.
 - d) Sie erleichtern sich das Feststellen von Z, indem Sie grundsätzlich in jede DATA-Zeile nur 10 X/Y-Werte (= 20 Zahlen) eintragen. Um nicht schon beim Eintragen in die DATA-Zeile jeweils mühsam nachzuzählen, habe ich mir angewöhnt, jede vorhergehende DATA-Zeile durch die folgende zu überschreiben.

Zeihen Sie mich nicht der Lüge, wenn Sie im Beispiel V.2 die DATA-Werte nachzählen. Warum dort oft mehr als 10 X/Y-Werte stehen, erfahren sie im folgenden Punkt.

- e) Bevor Sie nach dem Setzen des RESTORE-Befehles die Daten für die nächsten Linienzüge in die folgenden DATA-Zeilen eintragen, sollten Sie Ihr Grafikprogramm Probe laufen lassen.

Zu diesem Zweck baue ich für die Dauer der Erstellung einer Grafik in das Programm eine Kontrollroutine für die Korrektur einzelner Linienzüge ein, die ich nach abgeschlossenem Programm wieder lösche. Sie ist in den Zeilen 19-21. Nach jedem DATA-Schnittpunkt wird der Druck unterbrochen, das Papier nach vorne bewegt und der aktuelle DATA-Wert (Koordinaten + I-Wert) Befehl) angezeigt.

Ist der Linienzug nicht zufriedenstellend, so wird der Wert und die beabsichtigte Änderung notiert. Anschließend wird das Zeichnen durch Betätigung der ENTER-Taste fortgesetzt.

Erst nach Abbruch des Zeichnens mangels weiterer DATA-Zeilen werden die Änderungen programmiert. Dabei sollte auch die Verwendung von Dezimalbrüchen erwogen werden - ebenso die dann entsprechend in die DATA-Zeilen eingefügt werden und die den Z-Wert erhöhen.

4. Mit V geben Sie den Vergrößerungs- bzw. Verkleinerungsfaktor der jeweiligen Kopie an.

Ich hoffe, daß es Ihnen mit diesen Informationen möglich ist, Grafikprogramme zu erstellen. Ist Ihnen noch etwas unklar, können Sie mit der Überprüfungsroutine jeden einzelnen Schritt nachvollziehen. Verfolgen Sie z.B., wie ich die Einzeichnung der Tastatur gelöst habe.

V.2.2. Geometrische Grafiken

Es würde den Umfang dieses Handbuches sprengen, wenn ich auf die Lösungsmöglichkeiten aller oder auch nur zahlreicher geometrischen Probleme eingehen würde. Ich darf hier auf entsprechende Handbücher verweisen - besonders auf das ausgezeichnete Grafikhandbuch für SHARP-Computer von Klaus Schreiner im FISCHER-Verlag. Ihm verdanke ich z.B. die Lösung, negative Werte in Kreis-Diagrammen optisch gut darzustellen.

Durch zwei Beispiele, die ich speziell für den PC-1600 entwickelt habe und die vermutlich einen größeren Benutzerkreis haben werden, möchte ich zeigen, wie gut die Grafikfunktion dieses Computers für die Lösung geometrischer Probleme geeignet ist.

Ich hoffe, daß die Kommentarzeilen wieder ausreichen, Fragen - gegebenenfalls im TRON-Verfahren - zu klären. Deshalb sollen hier nur einige grundsätzliche Erläuterungen zu den beiden Programmen gegeben werden.

Beispiel V.3

Grafische Statistik

An diesem Beispiel wird besonders erkennbar, wie sich das Display gut für die Anlage von Steuermenüs eignet, die durch Betätigung der Funktionstasten die verschiedenen Unterprogramme aktivieren.

Um auch einen kleinen Arbeitsspeicher für dieses Programm zu nutzen, wird erst nach Eingabe der Höchstmenge der Posten, die verarbeitet werden sollen, dimensioniert.

Wahlweise werden die Grafiken (Balken, Kurven, Kreis) als Display oder Druck abgerufen. Der jeweils größte absolute Postenwert bestimmt das Darstellungsformat zur optimalen Ausnutzung des Displays oder Papiers.

Um auch mehr als 16 Posten (AW) darstellen zu können, wird die Grafik beim Balkendiagramm im Längsformat gedruckt, wenn sie mehr als 16 betragen. Auf diese Weise können beliebig viele Posten verarbeitet werden. Gegebenenfalls ist die Abschneidfunktion auszuschalten.

Beispiel V.4

Biorhythmen

Den Biorhythmen liegt die Theorie zu Grunde, daß positive und negative Impulse in psychisch-somatischen Körperreaktionen in durch die Geburt festgelegten Rhythmen verlaufen. Falls Sie mehr oder weniger stark Sympathie für diese These haben, sollten Sie aber zumindest dabei bedenken, daß die Rhythmen vermutlich nur Grundstimmungen verstärken. Negative Werte lassen nicht alles schief gehen, sondern werden mehr oder weniger stark entsprechend Impulse für Reaktionen geben.

Die Biorhythmtheorie sagt, daß die seelischen Impulse im Rhythmus von 28 Tagen, die geistigen im Rhythmus von 33 Tagen und die körperlichen im Rhythmus von 23 Tagen verlaufen.

- DER FACHVERLAG FÜR

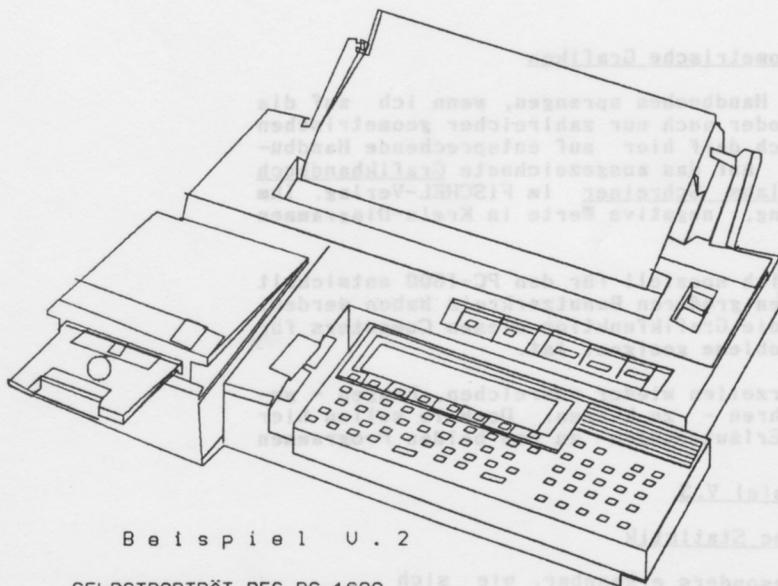
TASCHEN-COMPUTER!

Beispiel U. 1

```

1:'      Fettdruck und Sperrdruck
2:'
3: CLEAR :CLS :GRAPH :DIM T$(0)*50:INPUT "Fettdruckfaktor: ";F
4: INPUT "PITCH-Zahl(>6): ";P:IF P<7GOTO 4
5: INPUT "Schriftgrösse: ";S:CLS :PITCH P*S
6: PRINT "Text:":CURSOR 0,1:INPUT T$(0)
7: FOR I=0TO F:GLCURSOR (I,12):LPRINT T$(0):NEXT I
8: TEXT :LF 2:GOTO 3:END
    
```

PC - 1600 - Selbstporträt



Beispiel U. 2

```

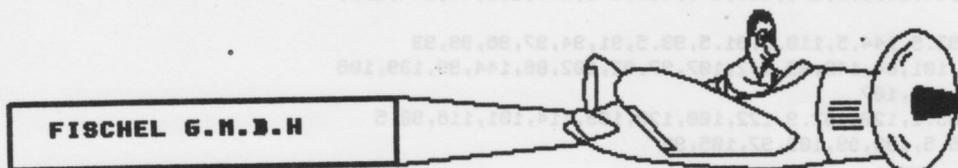
1:'      SELBSTPORTRÄT DES PC-1600
2:'
3:'      Grafik-Programm
4:'
5: GRAPH
6: PRINT "  Vergrößerungsfaktor":CURSOR 13,2:INPUT U
7: GOSUB 8:GOTO 29
8: READ X,Y
9: GLCURSOR (X*U,-Y*U)
10: READ Z
11: FOR I=1TO Z
12: READ X,Y
13: LLINE -(X*U,-Y*U),0:GOTO 25
14:'
15:'      Kontrollroutine
16:'      für Korrektur der einzelnen Striche
17:'      wird nach Erstellung der Grafik gelöscht
18:'
19: PRINT X;" ";Y,I:RLINE -(0,-100),9:A=1
20: INPUT A:IF A<>0GOTO 20
21: RLINE -(0,100),9
22:'
23:'      Fortsetzung des Grafikprogramms
24:'
25: NEXT I:RETURN
26:'
27:' Teilgrafiken des Selbstporträt des PC-1600
28:'
29: RESTORE 91:GOSUB 8
30: RESTORE 99:GOSUB 8
31: RESTORE 107:GOSUB 8
32: RESTORE 114:GOSUB 8
33: RESTORE 118:GOSUB 8
34: RESTORE 122:GOSUB 8
35: RESTORE 128:GOSUB 8
36: RESTORE 134:GOSUB 8
37: RESTORE 138:GOSUB 8
38:'
39:'      Tastatur-Grafik
40:'
41: RESTORE 143
42: READ Z
43: FOR I=1TO Z
44: READ X,Y
45: GLCURSOR (X*U,-Y*U)
46: RLINE -(U,U)-(2*U,-.8*U)-(-U,-U)-(-2*U,.8*U)
47: NEXT I
48: RESTORE 156
    
```

```

49:READ Z
50:FOR I=1TO Z
51:READ X,Y
52:GLCURSOR (X*U,-Y*U)
53:RLINE -(U,U)-(5*U,-1.7*U)-(-U,-U)-(-5*U,1.7*U)
54:NEXT I
55:''
56:''          Schraffur-Routine
57:''
58:X=152:Y=99
59:FOR I=0TO 6:LLINE ((X-I/2)*U,-(Y+I)*U)-(U*(X+27-I/2),-U*(Y+I
+9)):NEXT I
60:''
61:''          Diskettenkreis
62:''
63:GLCURSOR (34*U,-93*U):SORGN
64:R=U*3:GLCURSOR (R,R)
65:FOR W=0TO 360STEP 4
66:X=SIN W*R:Y=COS W*R:LLINE -(X,Y):NEXT W
67:TEXT :LF 15:END
68:''
69:''Koordinaten-Daten der einzelnen Grafikpunkte
70:''  Zum jeden Punkt gehören zwei Daten (X,Y).
71:''  Ausserdem folgt in jeder RESTORE-Datenzeile
72:''nach den ersten beiden Daten für den GLCURSOR
73:''das Datum für Z (Zähler für Anzahl der Linien
74:''  bis zum nächsten RESTORE).
75:''
76:''          Umriss
77:''
78:DATA 100,7,101,97,17,94,16,94,15,96,16,96,14,94,14,94,13,92,
13,83,34
79:DATA 81,34,82,32,83,32,73,30,54,51,53,57,47,55,21,79,21,80,2
4,81,24,84
80:DATA 12,96,12,97,59,115,64,110,160,142,160,139,166,141,181,1
13,181,107,180,106
81:DATA 197,75,197,63,186,61,192,41,192,39,188,38,186,40,183,39
,183,34,182,33,181,32
82:DATA 181,30,100,7,97,17,86,35,96,16,96,14,98,15,97,17,86,35,
83,43,82,47,83,43
83:DATA 81,42,75,51,78,52,81,47,81,42,81,47,78,52,166,79,171,69
,166,67,171,69
84:DATA 169,73,81,47,165,72,181,32,182,32,182,33,183,34,183,39,
180,38,180,40,182,41
85:DATA 180,40,180,38,183,39,186,40,186,43,180,60,177,59,174,61
,170,59
86:DATA 174,61,179,48,177,49,177,45,176,44,177,49,179,48,174,61
,182,41
87:DATA 186,43,180,61,186,62,184,61,190,39,184,61,186,62,187,61
88:''
89:''          PC-1600P
90:''
91:DATA 168,64,48,191,71,195,64,186,62,195,64,181,90,54,51,181,
90,194,67,194,69
92:DATA 197,63,175,102,175,105,181,107,99,81,81,105,81,106,166,
134,181,107,166,134
93:DATA 166,141,81,112,81,106,81,109,75,107,75,102,80,94,75,92,
82,84,87,86,94,77
94:DATA 175,102,175,105,99,81,94,88,94,77,87,86,87,87,88,84,91,
85,86,92,84,91,83,92
95:DATA 85,93,80,100,77,99,80,94,80,95,87,87
96:''
97:''          PC-1600
98:''
99:DATA 91,94,33,100,82,179,108,176,114,148,105,152,99,144,111,
91,94
100:DATA 91.5,93.5,144.5,110.5,91.5,93.5,91,94,97,96,99,93
101:DATA 95,92,101,84,148,99,142,107,97,92,102,86,144,99,139,106
,134,104.5,132,107
102:DATA 126,105.2,128,102.9,122,100,120,103,114,101,116,98.5
103:DATA 110,96.5,108,99,103,97,105,95
104:''
105:''          Diskette
106:''
107:DATA 24,84,34,26,82,54,91,54,92,26,83,54,92,54,94,52,93,53,9
3,40,106
108:DATA 54,91,40,104,40,106,34,104,34,102,40,104,34,102,35,100.
20,95

```

109: DATA 20, 98, 18, 100, 12, 97, 12, 96, 18, 99, 34, 105, 18, 99, 20, 98, 20, 95
 , 20, 96
 110: DATA 34, 101, 37, 98, 23, 93, 17, 98, 20, 95, 35, 100
 111: '
 112: ' CE-1600F-Teil
 113: '
 114: DATA 31, 84, 8, 27, 87, 46, 93, 49, 90, 30, 84, 38, 87, 36, 89, 39, 90, 41, 87
 115: '
 116: ' CE-1600F-Teil
 117: '
 118: DATA 24, 77, 9, 60, 90, 64, 85, 59, 83, 60, 83, 56, 88, 51, 86, 55, 81, 60, 83
 , 28, 72
 119: '
 120: ' CE-1600P-Tastaturleiste
 121: '
 122: DATA 119, 75, 28, 116, 81, 168, 97, 172, 91, 119, 75, 119, 76, 171, 92, 172
 , 91, 168, 90, 164, 96
 123: DATA 168, 90, 161, 88, 158, 93, 161, 88, 152, 85, 149, 90, 152, 85, 143, 82
 , 139, 88, 136, 87
 124: DATA 140, 81, 139, 82, 122, 77, 119.5, 82, 121.5, 78, 138, 83, 130, 80.5,
 131, 79, 128, 85
 125: '
 126: ' CE-1600F-Teil
 127: '
 128: DATA 53, 57, 18, 84, 66, 60, 90, 60, 92, 84, 66, 60, 92
 129: DATA 56, 91, 56, 96, 59, 97, 58, 99, 49, 96, 51, 94, 59, 97
 130: DATA 58, 99, 58, 100, 84, 66, 84, 67, 58, 101, 59, 115
 131: '
 132: ' Inter face
 133: '
 134: DATA 64, 110, 9, 64, 108, 59, 114, 64, 108, 64, 96, 89, 63, 64, 96, 76, 99, 7
 5, 101, 64, 98
 135: '
 136: ' Papierlöser
 137: '
 138: DATA 176, 70, 13, 182, 72, 177, 82, 171, 79, 176, 70, 176, 72, 181, 74, 179
 , 77, 178.5, 78, 174, 76
 139: DATA 172, 79, 176, 72, 175, 75, 179, 77
 140: '
 141: ' Tastatur
 142: '
 143: DATA 69, 92, 97, X+5, Y+1.5, X+5, Y+1.5, X+5, Y+1.5, X+5, Y+1.5, X+5, Y+
 1.5, X+5, Y+1.5
 144: DATA X+5, Y+1.5, X+5, Y+1.5, X+5, Y+1.5
 145: DATA 91, 100, X+5, Y+1.5, X+5, Y+1.5, X+5, Y+1.5, X+5, Y+1.5, X+5, Y+1.
 5, X+5, Y+1.5
 146: DATA X+5, Y+1.5, X+5, Y+1.5
 147: DATA 87, 102, X+5, Y+1.5, X+5, Y+1.5, X+5, Y+1.5, X+5, Y+1.5, X+5, Y+1.
 5, X+5, Y+1.5
 148: DATA X+5, Y+1.5, X+5, Y+1.5, X+5, Y+1.5
 149: DATA 84, 104, X+5, Y+1.5, X+5, Y+1.5, X+5, Y+1.5, X+16, Y+4.5, X+5, Y+1.
 5
 150: DATA 148, 109, X+5, Y+1.5, X+9, Y+4, X+5, Y+1.5
 151: DATA 145, 112, X+5, Y+1.5, X+5, Y+1.5, X+5, Y+1.5, X+5, Y+1.5
 152: DATA 143, 115, X+5, Y+1.5, X+5, Y+1.5, X+5, Y+1.5, X+5, Y+1.5
 153: DATA 141, 118, X+5, Y+1.5, X+5, Y+1.5, X+5, Y+1.5, X+5, Y+1.5
 154: DATA 139, 122, X+5, Y+1.5, X+5, Y+1.5, X+5, Y+1.5, X+5, Y+1.5, 122, 81,
 131, 83
 155: DATA 94, 94, X+5, Y+1.5, X+6, Y+2, X+6, Y+1.5, X+7, Y+2, X+6, Y+2, X+6, Y
 +2, X+7, Y+2
 156: DATA 5, 105, 111, 126, 117, 143, 87, 151, 90, 160, 92



Beispiel U. 3

```

1: GRAFISCHE STATISTIK
2:
3:   Vorprogramm und Parameter
4:
5: "S"CLS :CLEAR :WAIT 0
6:CLS :CURSOR 4,0:PRINT "Statistik-Programm":CURSOR 6,1:PRINT
   "für den PC-1600"
7:CURSOR 3,2:PRINT "(DISPLAY und Drucker)":CURSOR 6,3:PRINT "u
   on Peter Kriz":WAIT 100
8:LINE (0,0)-(155,31),X,65535,BF:CLS :WAIT 0
9:PRINT "   Anzahl der Werte":CURSOR 11,3:INPUT AW
10:DIM W(AW),BW$(AW),FW(AW),U(AW),P$(AW),Z(AW),S(AW)
11:KX=958:KY=-400:DY=31:NZ=1:XK=400:YK=-AW*56
12:'
13:'           Hauptmenü
14:'
15:CLS :CURSOR 0,1:PRINT "Wert ":PRINT "Ein      Bal      Kur Kr
   eis
16:CURSOR 0,3:PRINT "gabe      ken      ven"
17:GOSUB 218
18:ON IGOTO 22,52,132,160
19:'
20:'           Eingabe der Werte
21:'
22:CLS :WS=0:WAIT 0:PRINT "   auch negative Werte?":CURSOR 10,1
   :PRINT "(J/N)":GOSUB 232:IF J$="J"LET NZ=-1
23:IF NZ=1GOTO 25
24:KY=KY/2:XK=200:DY=15
25:CLS :CURSOR 0,3:FOR N=1TO AW
26:PRINT STR$(N);". Wert:":CURSOR 10,3:INPUT W(N)
27:IF ABS W(N)>PWLET PW=ABS W(N)
28:IF ABS W(N)<MWLET MW=ABS W(N)
29:WS=WS+ABS W(N):PRINT W(N)
30:INPUT "Postenbezeichnung: ";BW$(N)
31:INPUT "Farbe: ";FW(N):IF FW(N)<<00R FW(N)>>3GOTO 31
32:NEXT N
33:CLS :CURSOR 0,1:PRINT "Eingabe wiederholen? (J/N)":GOSUB 232
   :IF J$="J"GOTO 22
34:IF PW<MWLET PW=M
35:GOSUB 36:GOTO 43
36:FOR N=1TO AW
37:U(N)=W(N)*100/WS:U(N)=DY*W(N)/PW:Z(N)=-KY*W(N)/PW:IF WA=1LET
   Z(N)=XK*W(N)/PW
38:IF N>AWRETURN
39:NEXT N:RETURN
40:'
41:'           Achsen-Beschriftung
42:'
43:CLS :CURSOR 7,0:PRINT "Beschriftung":CURSOR 2,1:PRINT "der s
   enkrechten Achse":PRINT :INPUT KY$
44:CLS :CURSOR 7,0:PRINT "Beschriftung":CURSOR 2,1:PRINT "der w
   aagerechten Achse":PRINT :INPUT KX$
45:IF AW=14LET KX$=LEFT$(KX$,14)
46:IF AW=15LET KX$=LEFT$(KX$,9)
47:IF AW=16LET KX$=LEFT$(KX$,4)
48:CLS :CURSOR 7,0:PRINT "Überschrift":CURSOR 2,1:PRINT "des Di
   agramm-Ausdrucks":PRINT :INPUT T$:GOTO 15
49:'
50:'           Balkendiagramm - DISPLAY
51:'
52:GOSUB 227:CLS :IF DW=3GOTO 15
53:IF DW=2GOTO 67
54:B=120/(AW+(AW+1)/2):Z=B/2:GOSUB 55:GOTO 59
55:CLS :CURSOR 0,0:PRINT LEFT$(KY$,1)
56:FOR J=2TO 4:PRINT MID$(KY$,J,1):NEXT J
57:IF NZ=-1THEN CURSOR 22,2ELSE CURSOR 22,3
58:PRINT LEFT$(KX$,4):LINE (6,0)-(6,31):LINE (6,DY)-(155,DY):
   RETURN
59:N=0:FOR X=6+2TO 155STEP B+Z:N=N+1
60:LINE (X,DY)-(X+B,DY-U(N)),,65535,BF
61:IF N=AWGOTO 63
62:NEXT X
63:GOSUB 234:GOTO 52
64:'
65:'           Balkendiagramm - Druck
66:' - DER FACHVERLAG FÜR

```

```

67:SA=1:IF AW>16LET WA=1
68:IF WA<>1GOSUB 239
69:GOSUB 267:SA=0
70:Z=0:B=36:ZZ=INT (KX/AW)-60:IF WA=10R ZZ<16LET ZZ=16
71:T=14:FOR X=1TO AW:COLOR FW(X):A=Z+(X*B):S(X)=Z(X)
72:IF SF=0GOTO 78
73:S=INT ABS S(X)MOD 20
74:IF S(X)<0GOTO 77
75:IF S>9LET S(X)=INT S(X)+20-SELSE LET S(X)=INT S(X)-S
76:GOTO 78
77:IF S>9LET S(X)=INT S(X)-20+SELSE LET S(X)=INT S(X)+S
78:IF WA=1THEN LLINE (0,-A+B)-(S(X),-A),0,,ELSE LLINE (A,S(X))-
(A-B,0),0,,B
79:IF SF=0GOTO 106
80:GLCURSOR (A-B,0)
81:IF (X-1)/4=INT ((X-1)/4)LET SR=1
82:IF (X-2)/4=INT ((X-2)/4)LET SR=2
83:IF (X-3)/4=INT ((X-3)/4)LET SR=3
84:IF (X-4)/4=INT ((X-4)/4)LET SR=4
85:IF SR=10R SR=3GOTO 90
86:IF S(X)>=0GOTO 88
87:FOR HH=S(X)TO -9STEP 10:GOTO 89
88:FOR HH=0TO S(X)-9STEP 10
89:GLCURSOR (A-B+2,HH+2):GOTO 97
90:IF S(X)>=0GOTO 92
91:FOR HH=S(X)TO -18STEP 20:GOTO 93
92:FOR HH=0TO S(X)-18STEP 20
93:GLCURSOR (A-B+2,HH+4):CSIZE 2
94:*
95:*           Schraffurprogramm
96:*
97:IF SR=1RESTORE 281
98:IF SR=3RESTORE 286
99:IF SR=2RESTORE 285
100:IF SR=4RESTORE 287
101:IF SR=10R SR=3READ S1,S2,S3
102:IF SR=20R SR=4READ S1,S2,S3,S4,S5,S6,S7
103:IF SR=10R SR=3LPRINT CHR$ S1;CHR$ S2;CHR$ S3
104:IF SR=20R SR=4LPRINT CHR$ S1;CHR$ S2;CHR$ S3;CHR$ S4;CHR$ S5;
CHR$ S6;CHR$ S7
105:NEXT HH:CSIZE 1
106:IF WA<>1GOTO 110
107:IF S(X)<0GLCURSOR (25,-A+B-B/2-2)
108:IF S(X)>=0GLCURSOR (0-(LEN BW$(X)*6)-10,-A+B-B/2-2)
109:GOTO 112
110:IF S(X)<0GLCURSOR (A-B+B/2+2,25)
111:IF S(X)>=0GLCURSOR (A-B+B/2+2,0-(LEN BW$(X)*6)-10)
112:IF WA=1ROTATE 0ELSE ROTATE 3
113:LPRINT BW$(X)
114:IF WA=1ROTATE 1ELSE ROTATE 0
115:IF WA<>1GOTO 118
116:IF S(X)<0GLCURSOR (3,-A+B-B/2+18)ELSE GLCURSOR (S(X)+3,-A+B-
B/2+18)
117:GOTO 119
118:IF S(X)<0GLCURSOR (A-B+B/2-12,3)ELSE GLCURSOR (A-B+B/2-12,S(
X)+3)
119:LPRINT W(X):COLOR FW(X):XT=S(X):IF S(X)<0LET XT=0
120:GLCURSOR (0,0)
121:IF WA<>1GOTO 125
122:RLINE (XT,-A+B)-(T,-T)-(0,-B)-(-T,T):RLINE -(T,-T),9:IF S(X)
>=0RLINE -(S(X),0)-(-T,T),0
123:IF S(X)<0RLINE -(S(X),0)-(-T,T),0
124:GOTO 127
125:RLINE (A-B,XT)-(T,T)-(B,0)-(-T,-T):RLINE -(T,T),9:IF S(X)>=0
RLINE -(0,-S(X))-(-T,-T),0
126:IF S(X)<0RLINE -(0,S(X))-(-T,-T),0
127:Z=Z+ZZ:NEXT X
128:GLCURSOR (0,KY-50):SORGN :WA=0:GOTO 52
129:*
130:*           Kurvendiagramm - DISPLAY
131:*
132:GOSUB 227:IF DW=3GOTO 15
133:IF DW=2GOTO 143
134:B=132/AW:GOSUB 55
135:Z=6+B+B/2:LX=Z-B:LY=DY-U(1)
136:LINE (LX-1,LY-1)-(LX+1,LY+1),,65535,BF:GOSUB 139

```

- DER FACHVERLAG FÜR

TASCHEN-COMPUTER!

```

137:N=1:FOR X=ZTO 132STEP B:N=N+1:LINE (LX+1,LY)-(X,DY-U(N)),,21
      845
138:LX=X:LY=DY-U(N):GOSUB 139:IF N<AWNEXT XELSE GOSUB 234:GOTO
      132
139:LINE (LX-1,LY-1)-(LX+1,LY+1),,65535,BF:RETURN
140:'
141:'           Kurvendiagramm - Druck
142:'
143:WA=0:GOSUB 239
144:B=INT ((KX-12*LEN KX)/AW):GOSUB 267
145:Z=B+B/2:LX=Z-B:LY=Z(1)
146:N=1:FOR X=ZTO KX+BSTEP B:N=N+1:COLOR FW(N-1)
147:CSIZE 4:GLCURSOR (LX-8,LY-12):LPRINT CHR$ &F9:CSIZE 1
148:IF W(N-1)<0GOTO 150
149:GLCURSOR (LX-15,LY-NZ*6):LPRINT W(N-1):GOTO 151
150:GLCURSOR (LX-15,LY+NZ*6):LPRINT W(N-1):GOTO 151
151:LLINE (LX,LY)-(X,Z(N)),0,1:LX=X:LY=Z(N)
152:IF N=AWGOTO 154
153:NEXT X
154:COLOR FW(N):CSIZE 4:GLCURSOR (LX-8,LY-12):LPRINT CHR$ &F9:
      CSIZE 1
155:GLCURSOR (LX+15,LY-NZ*6)
156:LPRINT W(N):GLCURSOR (0,KY-50):SORGN :GOTO 132
157:'
158:'           Kreisdiagramm - DISPLAY
159:'
160:GOSUB 227:CLS :IF DW=3GOTO 15
161:IF DW=2GOTO 198
162:RA=15:U(0)=140:U(0)=15
163:FOR X=1TO AW:WI=ABS U(X)*3.6:P*(X)=BW*(X)
164:IF AW>6LET P*(X)=LEFT$(BW*(X),1)
165:IF AW<7AND AW>3LET P*(X)=LEFT$(BW*(X),6)
166:IF X=1LET W1=-WI/2ELSE LET W1=W2
167:W2=W1+W1
168:IF DW=2GOTO 201
169:GOSUB 170:NEXT X
170:PH=W1:U=U(0)+RA*COS PH:V=U(0)+RA*SIN PH
171:IF DW=1GCURSOR U(0),U(0)ELSE GLCURSOR (U(0),U(0))
172:IF DW=2LLINE -(U,U)
173:IF DW=1LET PH=PH+18ELSE LET PH=PH+5
174:IF PH>W2LET PH=W2
175:XD=U(0)+RA*COS PH:YD=U(0)+RA*SIN PH:U=YD
176:IF DW=2LLINE -(U,U)ELSE PSET (U,U)
177:IF PH<W2GOTO 173
178:IF DW=1LINE (XD,YD)-(U(0),U(0)),,21845ELSE LLINE -(U(0),U(0))
      ,0
179:IF DW=1AND X=AWTHEN 181
180:RETURN
181:USING "###.#":IF AW<7GOTO 188
182:FOR J=1TO 3:CURSOR 0,J-1:IF J<=AW+1PRINT P*(J);U(J):NEXT J
183:GOSUB 234:GOTO 160
184:FOR J=4TO 6:CURSOR 8,J-4:IF J<=AWTHEN 185ELSE GOSUB 234:GOTO
      160
185:PRINT P*(J);U(J):NEXT J
186:FOR J=7TO 9:CURSOR 16,J-7:IF J<=AWTHEN 187ELSE GOSUB 234:
      GOTO 160
187:PRINT P*(J);U(J):NEXT J
188:IF AW<4GOTO 193
189:FOR J=1TO 3:CURSOR 0,J-1:IF J<=AWTHEN 190ELSE GOSUB 234:GOTO
      160
190:PRINT P*(J);U(J):NEXT J
191:FOR J=4TO 6:CURSOR 12,J-4:IF J<=AWTHEN 192ELSE GOSUB 234:
      GOTO 160
192:PRINT P*(J);U(J):NEXT J
193:FOR J=7TO 9:CURSOR 0,J-1:IF J<=AWTHEN 194ELSE GOSUB 234:GOTO
      160
194:PRINT P*(J);U(J):NEXT J:GOSUB 234:GOTO 160
195:'
196:'           Kreisdiagramm - Druck
197:'
198:CLS :KR=1:GOSUB 239:RA=(ABS KY)/(2*1.1):IF NZ=-1LET RA=RA*2
199:GOSUB 267:KR=0
200:GRAPH :GLCURSOR (GX,-1.5*RA):SORGN :GOTO 163
201:U(0)=0:V(0)=0
202:IF U(X)<0LET U(0)=.1*RA*COS (W1+W1/2)
203:IF V(X)<0LET V(0)=.1*RA*SIN (W1+W1/2)
204:COLOR FW(X):GOSUB 170

```

- DER FACHVERLAG FÜR

TASCHEN-COMPUTER!

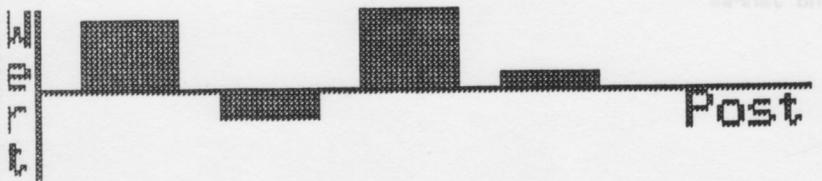
```

205:UU=U:VV=U:U=U(0)+.6*RA*COS (W1+WI/2):V=U(0)+.6*RA*SIN (W1+WI
/2)
206:CSIZE 1:GLCURSOR (U,U):RLINE -(16,-6),9
207:IF X>AWTHEN 210ELSE LPRINT BW$(X)
208:GLCURSOR (U,U):RLINE -(16,-15),9:LPRINT USING "###.#";U(X);
"%":USING
209:IF X=AWGOTO 211
210:NEXT X
211:GLCURSOR (-KX/2,-1.2*RA):SORGN :CSIZE 2
212:FOR J=1TO AW:COLOR FW(J):LPRINT BW$(J);" ";W(J):GLCURSOR (0,
-J*16):NEXT J:GOTO 160
213:'
214:'           Subroutinen
215:'
216:'           Menüsteuerung
217:'
218:WAIT :J=ASC INKEY$ :IF J<>17AND J<>19AND J<>21AND J<>22GOTO
218
219:IF J=17LET I=1
220:IF J=19LET I=2
221:IF J=21LET I=3
222:IF J=22LET I=4
223:WAIT 0:RETURN
224:'
225:'           Untermenü
226:'
227:WAIT 0:CLS :CURSOR 0,2:PRINT "Dis      Pa      Haupt
228:CURSOR 0,3:PRINT "play      pier      menü":GOSUB 218:DW=I:
RETURN
229:'
230:'           Ja-Nein-Routine
231:'
232:GOSUB 234:J$=INKEY$ :IF J$<>"J"AND J$<>"N"GOTO 232
233:RETURN
234:IF INKEY$ =""GOTO 234
235:RETURN
236:'
237:'           Formatierung der Zeichnung
238:'
239:CLS :IF SA<>1GOTO 242ELSE CURSOR 4,0:PRINT "Sollen die Balke
n
240:CURSOR 3,1:PRINT "schraffiert werden?":CURSOR 10,2:PRINT "(J
/N)"
241:GOSUB 232:CLS :IF J$="J"LET SF=1ELSE LET SF=0
242:CLS :IF KR=0GOTO 244
243:PRINT "anderer Radius als 4 cm":CURSOR 10,1:PRINT "(J/N)"
:GOTO 246
244:CURSOR 3,0:PRINT "andere Formatierung
245:PRINT "der Zeichnung als 19*8 cm";CHR$ &FD:PRINT " (Breite
*Höhe)? (J/N)"
246:GOSUB 232:IF J$="J"GOTO 249
247:KX=958:KY=-400:IF NZ=-1LET KY=KY/2
248:RETURN
249:IF KR=0THEN 253ELSE CLS :CURSOR 7,0:PRINT "Radius in cm"
250:CURSOR 12,2:INPUT RA:CB=RA:CH=RA
251:IF NZ=1GOTO 257
252:CB=2*CB:CH=2*CH:GOTO 257
253:CLS :CURSOR 7,0:PRINT "Breite in cm":CURSOR 3,1:PRINT " <=19
und >=";AW*1.2
254:CURSOR 10,2:INPUT "cm :";CB
255:IF CB>19OR CB<AW*1.2GOTO 249
256:CLS :INPUT "Höhe in cm: ";CH
257:CLS :KX=CB*50+(LEN KX$+2)*12:KY=-CH*50:IF SA=1LET KY=KY+T
258:IF NZ=-1LET KY=KY/2:IF SA=1LET KY=KY+T/2
259:'
260:'           Untermenü für Druckanordnung
261:'
262:CLS :CURSOR 4,1:PRINT "Z e i c h n u n g
263:CURSOR 0,3:PRINT "links  Mitte  rechts":GOSUB 218:DL=I:CLS
:GOTO 36
264:'
265:'           Achsen - Druck
266:'
267:IF DW<>2RETURN
268:GX=INT (KX/2)

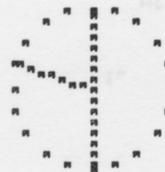
```

```

269: IF DL=2LET GX=480
270: IF DL=3LET GX=INT (959-KX/2)
271: COLOR 1:TEXT :CSIZE 2:ZL=LEN T*12
272: LCURSOR INT ((GX-ZL/2)/12):LPRINT T$: IF KR=1RETURN
273: GRAPH : IF WA<>1GOTO 278
274: GLCURSOR (941,-24):ROTATE 1:LPRINT KY$
275: IF NZ=1GLCURSOR (400,-48)ELSE GLCURSOR (600,-48)
276: SORGN
277: RLINE (0,0)-(0,YK-48),2:GOTO 280
278: GLCURSOR (GX-KX/2,-24):LPRINT KY$:GLCURSOR (GX-KX/2,KY-48):
SORGN
279: RLINE (0,0)-(KX-12*(LEN KX$+1),0),2
280: LPRINT " ";KX$:CSIZE 1:RETURN
281:'
282:'      Daten für vier Schraffurzeichen
283:'
284: DATA &B0,&B0,&B0
285: DATA &CE,&CE,&CE,&CE,&CE,&CE,&CE,&CE
286: DATA &B1,&B1,&B1
287: DATA &C5,&C5,&C5,&C5,&C5,&C5,&C5:END
    
```

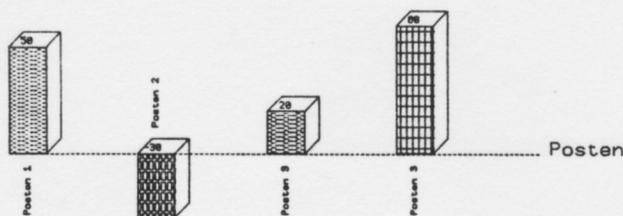


50.0
 30.0
 20.0



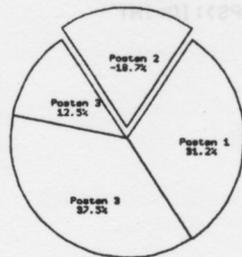
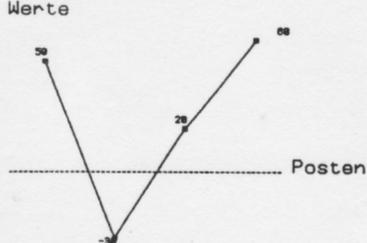
Beispiel

Werte



Beispiel

Werte



Posten 1	50
Posten 2	-30
Posten 3	20
Posten 4	60

Beispiel U.4

```

1:'          Biorhythmen-Programm
2:'
3:'  Vorspann
4:'
5:TEXT :CLS :CLEAR :CSIZE 3:CURSOR :PRINT " B i o r h y t h m
   e n"
6:COLOR 3:GRAPH :PITCH 33
7:FOR I=0TO 4:GLCURSOR (300+I,12):LPRINT "BIORHYTHMEN":NEXT I
8:'
9:'          Eingabe des Namens
10:'         und des Geburtstages
11:'
12:CLS :WAIT 0,S:TEXT :LF 2:CSIZE 2:COLOR 1:DIM N$(0)*30
13:INPUT "Name ";N$(0)
14:TAB ((80-(LEN (N$(0))+19))/2):LPRINT "für ";N$(0);" (*";
15:INPUT "Geburts-Tag ";T:INPUT "Monat ";M:INPUT "Jahr ";J:MM
   =M:JJ=J:LPRINT T;".";M;".";J;"")
16:ERASE N$(0):CLS
17:'
18:'         Eingabe des aktuellen Monats und Jahres
19:'
20:COLOR 0:LF 1:LPRINT "ab ";
21:CLS :INPUT "ab Monat ";MA:MO=MA
22:IF MA=1LPRINT "1. Januar ";
23:IF MA=2LPRINT "1. Februar ";
24:IF MA=3LPRINT "1. März ";
25:IF MA=4LPRINT "1. April ";
26:IF MA=5LPRINT "1. Mai ";
27:IF MA=6LPRINT "1. Juni ";
28:IF MA=7LPRINT "1. Juli ";
29:IF MA=8LPRINT "1. August ";
30:IF MA=9LPRINT "1. September ";
31:IF MA=10LPRINT "1. Oktober ";
32:IF MA=11LPRINT "1. November ";
33:IF MA=12LPRINT "1. Dezember ";
34:INPUT "Jahr ";JA:JO=JA:LPRINT JA
35:'
36:'  Druck des Monatsplanes
37:TAB 18:COLOR 2:LPRINT "-- physisch ";
38:COLOR 3:LPRINT "-- psychisch ";
39:COLOR 1:LPRINT "-- intellektuell"
40:GOSUB 71:GOSUB 87
41:D=TX-TG:COLOR 0:GRAPH :ROTATE 3
42:GLCURSOR (20,-150):SORGN :GLCURSOR (-2,-80):LPRINT "
   +"
43:X=I*25:LLINE (0,-100)-(-0,115)
44:FOR Z=1TO 31
45:IF Z=31AND I=30GOTO 53
46:IF Z=29AND I=28GOTO 53
47:X=Z*25
48:IF Z=1COLOR 3ELSE COLOR 0
49:LLINE (X,-100)-(X,115)
50:Y=80:W=X+10:LLINE (W-14,115)-(W-14,Y),9
51:IF Z=1COLOR 3:LPRINT Z;".";MO;".".NEXT Z
52:COLOR 0:LPRINT Z:NEXT Z
53:LLINE (X,0)-(-0,0),0
54:'
55:'  Berechnung und Druck der Biorhythmuskurven
56:'
57:PH=INT (D/23):PH=D-(23*PH):PS=INT (D/28):PS=D-(28*PS):IQ=INT
   (D/33):IQ=D-(33*IQ)
58:FOR U=1TO 3:COLOR U:E=0
59:FOR X=0TO I
60:IF U=2LET Y=SIN ((PH+X)/23*360)*80
61:IF U=3LET Y=SIN ((PS+X)/28*360)*80
62:IF U=1LET Y=SIN ((IQ+X)/33*360)*80
63:W=X*25:F=0
64:IF E=0LET F=9:E=1
65:LLINE (A,B)-(W,Y),F
66:A=W:B=Y:NEXT X:NEXT U
67:GOTO 99
68:'
69:'  Bestimmung der Anzahl der Tage pro Monat
70:'

```

```

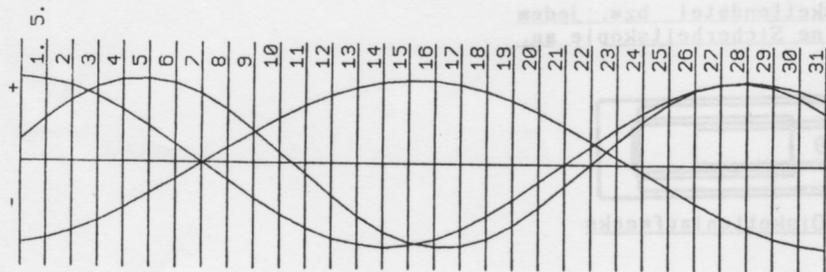
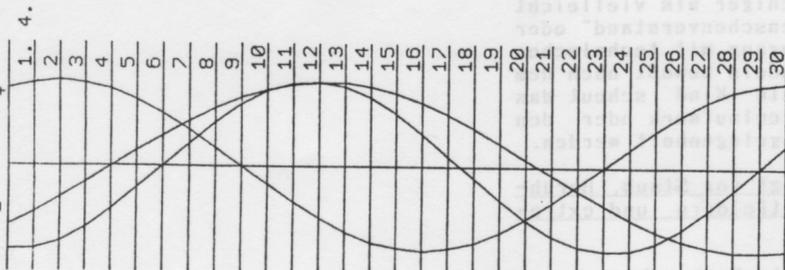
71: IF MA=2GOTO 78
72: IF MA=4OR MA=6OR MA=9OR MA=11GOTO 74
73: I=31: RETURN
74: I=30: RETURN
75: '
76: '          Schaltjahrberechnung
77: '
78: S=INT (JA/4): S=JA-S*4
79: IF S=0GOTO 80ELSE LET I=28: RETURN
80: S=INT (JA/100): S=JA-S*100
81: IF S=0GOTO 82ELSE LET I=29: RETURN
82: S=INT (JA/400): S=JA-S*400
83: IF S=0LET I=29ELSE LET I=28: RETURN
84: '
85: ' Berechnung der Tagesanzahl seit der Geburt
86: '
87: IF MA-3<0GOTO 88ELSE LET MA=MA+1: GOTO 89
88: JA=JA-1: MA=13+MA
89: TZ=INT (365.25*JA)+INT (30.6*MA)
90: TZ=TZ-INT (JA/100)+INT (JA/400): TX=TZ
91: IF M-3<0GOTO 92ELSE LET M=M+1: GOTO 93
92: J=J-1: M=13+M
93: TZ=INT (365.25*J)+INT (30.6*M)+T
94: TZ=TZ-INT (J/100)+INT (J/400): TG=TZ
95: RETURN
96: '
97: '          Fortsetzung des Programms
98: '
99: CLS : INPUT " nächster Monat? " N/J " ; Q$
100: IF Q$="N"OR Q$="n" THEN END ELSE LET MO=MO+1: MA=MO: JA=JO: J=JJ
      : M=MM: CLS : CURSOR 10,2: PRINT MO; " / " ; JA
101: GOSUB 71: GOSUB 87
102: D=TX-TG: COLOR 0: GLCURSOR (0,-250): SORGN : GOTO 42
103: END
    
```

B I O R H Y T H M E N

für Beispiel (* 12. 3. 1953)

ab 1. April 1987

-- physisch -- psychisch -- intellektuell



VI. Funktion und Nutzung des CE-1600F Diskettenlaufwerk

Die Hinweise gelten auch für die Nutzung der Speichermodule als RAM-Disks.

Ein entscheidender Grund, das PC-1600-System zu erwerben, ist die Möglichkeit, Programme und Dateien mit Disketten zu verwalten und auf sie "blitzschnell" zuzugreifen.

Für den PC-1600 hat SHARP ein eigenes Diskettenlaufwerk und spezielle Disketten entwickelt.

- DER FACHVERLAG FÜR

TASCHEN-COMPUTER!

Vielleicht mögen Sie im Blick auf die Preise größerer Diskettenlaufwerke und zugeordneter Disketten zunächst ein wenig zögern, sich das CE-1600F zuzulegen, und nach Interfaces Ausschau halten, um andere Diskettenlaufwerke zu verwenden.

Doch wenn Sie Erfahrungen mit Diskettenlaufwerken haben, werden Sie die Vorzüge des CE-1600F schnell zu schätzen wissen.

1. Das CE-1600F ist netzunabhängig und an jedem Ort einsetzbar.
2. Das CE-1600F nimmt als Teil der Kompakteinheit des PC-1600-Systems keinen zusätzlichen Platz in Anspruch.
3. Das CE-1600F arbeitet im Gegensatz zu "normalen" Diskettenlaufwerken angenehm geräuscharm. Das ist vor allem bei Dauerbetrieb ein großer Vorteil.
4. Das CE-1600F ermöglicht einen Betrieb ohne Kabelsalat.

VI.1. Vorsichtsmaßnahmen

Um aber gerade diese Option gut zu nutzen und ärgerliche Fehler zu vermeiden, ist es nützlich, sich mit der Funktion des Diskettenlaufwerks und der Eigenart der Diskettenbenutzung vertraut zu machen.

Diskettenstation und Disketten sind die empfindlichsten Teile des ganzen Systems und bedürfen besonderer Vorsichtsmaßnahmen. Das ist nicht etwa ein Fehler von SHARP. Im Gegenteil: Durch seine Spezialentwicklung sind z.B. die Disketten gegen Verbiegungsschäden besser geschützt als Normaldisketten. Aber grundsätzlich gelten auch für das CE-1600F dieselben Vorsichtsmaßnahmen wie für jedes Diskettensystem.

Ich möchte sie deshalb gleich zu Beginn nennen, damit Sie im Bedarfsfall nicht lange suchen müssen:

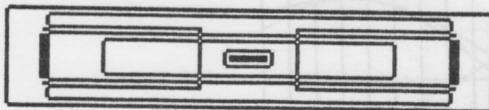
- a) Beachten Sie genau die Vorkehrungen auf S.17f. und die Vorsichtsmaßregeln auf S.21 der Bedienungsanleitung für das CE-1600F.

Verlassen Sie sich hier weniger als vielleicht sonst auf Ihren "gesunden Menschenverstand" oder auf Ihr Gefühl für den Umgang mit technischen Dingen - es sei denn, Sie handeln bewußt nach dem Grundsatz: Erst das gebrannte Kind scheut das Feuer. Das kann beim Diskettenlaufwerk oder den Disketten eine kostspielige Angelegenheit werden.

- b) Bewahren Sie Disketten geschützt vor Staub, Berührungen des Verschlusses, Magnetfeldern und extremen Temperaturen auf.

Disketten sind empfindlicher als Tonbandkassetten.

- c) Fertigen Sie von jeder Diskettendatei bzw. jedem Diskettenprogramm sofort eine Sicherheitskopie an.



VI.2. Funktion des Diskettenlaufwerks

VI.2.1. Die Diskette

Eine Diskette ist am besten mit einer Schallplatte zu vergleichen. Lassen Sie sich dabei nicht von der scheinbaren viereckigen Form der Diskette nicht verwirren. Sie wird nur durch die Schutzhülle verursacht, in der die eigentliche Diskette eingebettet ist und aus der Sie die Diskette niemals entfernen dürfen.

Der Vergleich mit einer Schallplatte hinkt dennoch. Denn während eine Schallplatte nur eine einzige Rille hat, die sich spiralförmig vom äußeren Rand zur Plattenmitte erstreckt, hat die CE-1600F-Diskette 16 konzentrische Spuren pro Seite.

Jede Spur ist wiederum in 8 Sektoren aufgeteilt. Sie sind gleichsam die Zimmer, in denen Programme und Dateien oder Teile davon einquartiert werden.

Ein Sektor umfaßt 512 Bytes. Die Daten werden gesammelt und dann in zwei internen Datensätzen je Sektor gespeichert. Sie dürfen das nicht mit Ihren Datensätzen verwechseln. Der LOC-Befehl gibt darum nur Auskunft darüber, wieviele Datensätze die Directory angelegt hat.

VI-2.2. Vorteile der Diskette gegenüber dem Tonband

Durch diese Aufteilung hat die Diskette gegenüber der Rekorder-Nutzung unüberbietbare Vorzüge.

Bei einem Tonband wird nacheinander jeweils der Bandteil mit Programmteilen belegt, der gerade am Tonkopf vorbeigeführt wird. Darum müssen Sie gegebenenfalls eine Kassette minutenlang vor- oder zurückspulen, um ein bestimmtes Programm zu laden. Außerdem müssen Sie - meist auf der Kassettenbeilage oder im Notizbuch - vermerken, bei welcher Bandzählnummer welches Programm beginnt. Sehr leicht kann bei Aufnahmen außerdem passieren, daß versehentlich Programme durch teilweise Überlappung zerstört werden - es sei denn. Sie verschenken Bandmenge durch große Sicherheitsabstände.

Die Diskettenstation gestattet den blitzschnellen Zugriff auf jede Datei bzw. jedes Programm, die auf einer Diskette gespeichert sind.

VI.2.3. Das "Direktionszimmer" - die Directory

Der sekundenschnelle Zugriff wird durch die Einrichtung einer Directory, eines "Direktionszimmers", ermöglicht.

- a) Die Directory führt intern - ohne daß Sie über alle Einzelheiten Auskunft erhalten können - genau Buch darüber, welche Sektoren mit welchen Programmen und Dateien belegt sind. Außerdem merkt sie sich Tag und Stunde der Belegung und den Umfang der verbrauchten und noch verfügbaren Speicherkapazität einer Diskette.
- b) Die Belegung der Sektoren erfolgt nicht systematisch - etwa Sektor für Sektor, Spur um Spur. Streicht der Tonkopf des Laufwerks über einen freien Sektor, wird er ihn belegen.
- c) Geben Sie den Befehl, von einer Diskette ein Programm oder eine Datei in den Arbeitsspeicher des PC-1600 zu laden, dann schlägt gleichsam der Direktor in seinem Protokollbuch nach, welche Sektoren von der aufgerufenen Datei belegt sind und sucht sie in der verzeichneten Reihenfolge der Belegung auf.



VI.2.4. Der Verwaltungsaufwand der Directory

Für die Verwaltung der Directory benötigt die Diskettenstation Speicherplatz auf der Diskette - ebenso der PC-1600 für die Kontrolle der Directory.

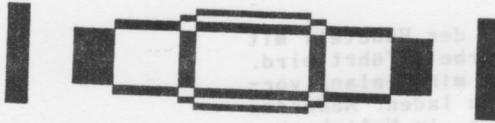
Das merken Sie, wenn Sie ein Speichermodul als RAM-Disk oder eine Diskette durch den INIT-Befehl formatieren:

- a) Eine Diskette hat für die Aufnahme von Programmen und Dateien nicht 64 Kbyte je Seite zur Verfügung, sondern höchstens 62464 Bytes. Es werden also für die Directory 3072 Bytes benötigt.
- b) Beim CE-1600M werden 2560 Bytes für die Directory abgezweigt, so daß 30208 Bytes für Programme und Dateien übrigbleiben.
- c) Beim CE-161 ist die Relation 2048 zu 14336 Bytes.
- d) Wird die Diskettenstation CE-1600F angeschlossen, geht vom Arbeitsspeicher des PC-1600 außerdem genau 1 KByte = 1024 Bytes für die Kontrolle der Diskettenstation verloren.

Deshalb werden über die Speicherkapazität einer Diskettenseite zwei Angaben gemacht: Es können höchstens 48 Programme und Dateien bis zu 62464 Bytes Gesamtumfang auf einer Diskettenseite gespeichert werden.

- a) Weil die Directory eine nicht veränderbare Größe hat, können in ihr nur Aufzeichnungen über die Anlage bis zu 48 Programmen und Dateien gemacht werden.
- b) So verschenken Sie Speicherkapazität, wenn die 48 Programme und Dateien einen geringeren Gesamtumfang als 62464 Bytes haben.

Schon deshalb ist zu überlegen, ob kleine Programme auf der Diskette oder besser auf Band gespeichert werden sollten.



VI.3. Nutzung des Diskettenlaufwerks

VI.3.1. Formatierung der Diskette

Mit dem INIT-Befehl müssen Sie eine Diskette oder ein RAM-Modul für die Diskettenverwaltung formatieren. D.h. daß auf dem Speichermedium das Direktionszimmer eingerichtet wird.

Es ist dabei Folgendes zu beachten:

- a) Mit INIT wird auf einer Diskette, die nicht mit dem Schreibschutzschalter hardwaremäßig geschützt ist, immer eine Directory neu eingerichtet.
- b) Der INIT-Befehl dient somit auch zur schnellen Löschung aller Programme und Dateien auf einer Diskette. Über die Art der Löschung s.u. VI.2.6.
- c) Gegen die ungewollte Neuformatierung einer Diskette schützen Sie sich am sichersten, wenn Sie sich angewöhnen, vor dem INIT-Befehl zu überprüfen, ob die Diskette schon formatiert ist und vielleicht wichtige Programme oder Dateien enthält.

Für die Überprüfung betätigen Sie die vordefinierte Funktionstaste DEF-%. Es erscheint auf dem Display FILES ". Sie müssen nur noch eingeben: X:" und die ENTER-Taste drücken. Dann sehen Sie, welche Programme und Dateien die Diskette enthält und ob sie überhaupt formatiert ist. Wenn nicht, erscheint ERROR 161.

Wichtige Programm- und Datei-Disketten sichern Sie am besten gegen eine ungewollte Neuformatierung der Diskette, indem Sie den Schreibschutzschalter der Diskette benutzen.

Ein softwaremäßiger Schreibschutz durch den SET-Befehl schützt die Diskette nicht vor dem INIT.

- d) Diese Vorsichtsmaßnahmen entfallen bei Ram-Disks.

Versuchen Sie, ein Ram-Disk-Modul neu zu initialisieren, erscheint ERROR 102, solange Programme oder Dateien gespeichert sind.

Ehe Sie ein Modul als Programm-Modul oder zur Erweiterung des Arbeitsspeichers nutzen wollen, müssen Sie zuerst den eventuellen softwaremäßig angebrachten Schreibschutz aller Programme und Dateien auf der RAM-Disk durch den Befehl

SET"S2:x.x", " "

beseitigen. Beachten Sie, daß zwischen den beiden Ehe Sie ein Modul als Programm-Modul oder zur Erweiterung des Arbeitsspeichers nutzen wollen, müssen Sie zuerst den eventuellen softwaremäßig angebrachten Schreibschutz aller Programme und Dateien auf der RAM-Disk durch den Befehl

SET"S2:x.x", " "

beseitigen. Beachten Sie, daß zwischen den beiden letzten Anführungszeichen ein Leerzeichen stehen muß. Ist das Modul in Slot-1 als RAM-Modul initialisiert, muß natürlich S2 durch S1 ersetzt werden. Anschließend müssen Sie jedes Programm oder jede Datei einzeln mit dem KILL-Befehl löschen.

Das ist zwar eine etwas umständliche Prozedur, die auch nur bei leerem Arbeitsspeicher vorgenommen werden kann. Aber so RAM-Disks gegen eine fahrlässige Löschung von Programmen und Dateien optimal geschützt.

UI.3.2. Löschen und Sichern von Programmen/Dateien auf Disketten

Tonbandkassetten können Sie wirklich löschen, indem Sie den Kassettenrekorder auf Aufnahme stellen und das Tonband ohne gleichzeitige Neuaufnahme am Tonkopf vorbeiführen.

Bei Disketten ist dies nicht möglich. Durch INIT oder KILL löschen Sie nur die Aufzeichnungen in der Directory. Die Programme und Dateien bleiben zunächst unverändert auf der Diskette erhalten.

Weil aber die Aufzeichnung in der Directory gelöscht sind, werden die Sektoren eines gekillten Programms oder einer neuformatierten Diskette bei Sicherung des nächsten Programms durch SAVE oder einer Datei durch OPEN FOR OUTPUT neu belegt, wenn sie vom Magnetkopf überstrichen werden.

Dasselbe gilt für den Fall, wenn ein Programm oder eine Datei auf dem Speichermedium gespeichert werden soll unter einem File-Namen, unter dem bereits ein anderes ohne Schreibschutz versehenes Programm gespeichert ist. Der PC-1600 behandelt in diesem Fall das Programm bzw. die Datei im Arbeitsspeicher wie ein Programm bzw. eine Datei, die vorher von dem betreffenden Speichermedium in den Arbeitsspeicher geladen, verändert und neu gespeichert wird.

UI.3.2. Löschen und Sichern von Programmen/Dateien auf Disketten

Tonbandkassetten können Sie wirklich löschen, indem Sie den Kassettenrekorder auf Aufnahme stellen und das Tonband ohne gleichzeitige Neuaufnahme am Tonkopf vorbeiführen.

Bei Disketten ist dies nicht möglich. Durch INIT oder KILL löschen Sie nur die Aufzeichnungen in der Directory. Die Programme und Dateien bleiben zunächst unverändert auf der Diskette erhalten.

Weil aber die Aufzeichnung in der Directory gelöscht sind, werden die Sektoren eines gekillten Programms oder einer neuformatierten Diskette bei Sicherung des nächsten Programms durch SAVE oder einer Datei durch OPEN FOR OUTPUT neu belegt, wenn sie vom Magnetkopf überstrichen werden.

Dasselbe gilt für den Fall, wenn ein Programm oder eine Datei auf dem Speichermedium gespeichert werden soll unter einem File-Namen, unter dem bereits ein anderes ohne Schreibschutz versehenes Programm gespeichert ist. Der PC-1600 behandelt in diesem Fall das Programm bzw. die Datei im Arbeitsspeicher wie ein Programm bzw. eine Datei, die vorher von dem betreffenden Speichermedium in den Arbeitsspeicher geladen, verändert und neu gespeichert wird.

Um ein versehentliches "Löschen" eines Programms oder einer Diskette zu vermeiden, sollten Sie grundsätzlich jedes Programm und jede Datei nach der Speicherung auf Diskette oder RAM-Disk mit dem SET-Befehl mit einem Schreibschutz versehen.

Ehe ich eine Diskette dem Laufwerk entnehme, gebe ich sicherheits- halber ein:

```
SET "X:*.x", "P"
```

So läßt sich eine irrtümliche Löschung verhindern.



Fischel lässt Sie nicht hängen !

VI.3.3. Programme und Dateien

Ich unterscheide in diesem Buch bewußt zwischen Disketten-Dateien und -Programmen. Sie werden nämlich unterschiedlich gehandhabt.

Programme werden auf der Diskette mit `SAVE"X:< Angabe von Filename.Extension > gespeichert` und mit `LOAD"X:< Angabe von Filenname.Extension >` in den Arbeitsspeicher geladen.

Die Handhabung entspricht also weitgehend der Kassettenbenutzung mit `CSAVE` und `CLOAD`. Allerdings muß bei `LOAD` im Unterschied zu `CLOAD` auch der Programm-Name (Filename:Extension) angegeben werden. Dafür entfällt das zeitraubende Suchen des Programmanfangs auf dem Band, und ein Programm wird im Vergleich zur Kassette von der Diskette blitzschnell geladen.

Man kann zwar mit dem `SAVE`-Befehl auch Programme auf Band speichern und mit `LOAD` wieder in den Arbeitsspeicher laden. Doch das bringt keine Vorteile, weil der Arbeitsprozeß beim Band durch die speziellen Diskettenbefehle nicht verkürzt wird. Dafür überwiegen eindeutig die Nachteile. Während mit `CSAVE` gespeicherte Programme während des Rückkladens mit dem Programm-Namen auf dem Display angezeigt werden, bleibt das Display leer, wenn mit `SAVE` auf Band gespeicherte Programme mit `LOAD` wieder abgerufen werden. Der Filename wird also nicht angezeigt.

Dateien werden mit dem `OPEN`-Programm auf der Diskette gespeichert und von ihr geladen. Hier erweist sich der große Vorteil der Diskette. Daten auf Band zu speichern, erfordert größere Sorgfalt, da alle zu speichernde Variablen beim Speicherbefehl angegeben werden müssen. Wird eine Variable vergessen, ist die ganze Datei wertlos.

Zwar ist dieser Fehler auch bei Disketten-Dateien möglich. Aber da man vor Eingabe einer Datei die Datenvariablen bestimmt, die zu benutzen sind, kommen Fehler erfahrungsgemäß sehr selten vor.

Außerdem bietet eine Disketten-Datei gegenüber einer Kassetten-Datei den einzigartigen Vorteil, daß durch Einbau entsprechender Routinen in das `OPEN`-Programm bestimmt werden kann, ob und unter welchen Bedingungen Variablen gespeichert oder gelesen werden sollen. Das können aus unterschiedlichem Anlaß jeweils andere Bedingungen für ein- und dieselbe Datei sein. Die Möglichkeiten, auf Variablen einzeln zuzugreifen, wurden z.B. bei den Adreßverwaltungs- und Textverarbeitungsprogrammen (VI.5.1. + 2.) genutzt.

Wird versucht, Dateien in den Arbeitsspeicher zu laden, obwohl die zugehörigen `BASIC`-Programme nicht im Speicher stehen, reagiert der PC-1600 mit `ERROR 30`.

VI.3.4. Disketten- oder Kassettennutzung

Die Diskette hat gegenüber der Kassette den Nachteil, daß Programme nicht zu anderen Programmen im Arbeitsspeicher geladen werden können, wie dies bei der Kassette durch `MERGE` möglich ist. Beim Laden eines Diskettenprogramms wird der Arbeitsspeicher gelöscht.

Schon aus diesem Grunde ist empfehlenswert, alle Programme nicht nur auf Disketten, sondern auch auf Band zu speichern. Das hat folgende Vorteile:

- a) Sie können nach Bedarf Programme verknüpfen. Das ist besonders hilfreich, wenn bestimmte Programmteile - z.B. Grafikroutinen - immer wiederkehren.
- b) Sie haben zusätzliche Sicherheitskopien der Programme - dazu auf einem anderen Speichermedium. Das ist besonders nützlich, wenn es technische Probleme gibt, z.B. Defekte an einer Diskette oder am Laufwerk auftreten.

Ich habe auf jeder Diskette nur folgende beiden Kleinprogramme gespeichert:

- a) Das `UMFANG.PRO`-Programm (581 Bytes): Damit ist jederzeit der Byte-Umfang eines Programms für Kalkulationszwecke feststellbar, z.B. ob zur besseren Ausnutzung der Speicherkapazität einer Diskette ein Programm auf eine andere kopiert und auf der alten gelöscht werden sollte.
- b) Das `UMFANG.DAT`-Programm (844 Bytes) zur Prüfung der Umfang (Bytes und Sektoren) einer Datei.

Beide Programme sind in der Beispielsammlung enthalten.

Andere Programme mit weniger als 1 KByte Umfang speichere ich normalerweise nur auf Band. Da das eingebaute CSAVE-Programm des ROM-Speichers ein Quicksave Programm ist, werden kurze Programme auch vom Band einigermaßen schnell geladen. Da die Diskette nur für 48 Programme und Dateien Platz hat, verschenke ich somit keine Speicherkapazität der Diskette (s. auch S.VI.4).

Wenn ich den PC-1600 auf Reisen mitnehme, speichere ich benötigte Kurzprogramme vorübergehend auf der RAM-Disk, die ich übrigens immer nur für zeitweilige Speicherung von Programmen und Dateien verwende.

VI.4. Handhabung von Diskettenbefehlen

VI.4.1. MAXFILES

Der MAXFILES-Befehl erlaubt die Öffnung bis zu 15 Dateien. Zwar wird das Maximum kaum genutzt. Es kann aber hilfreich sein, wenn Sie große Mengen von Daten verwalten und im Blick auf den großen, aber dennoch begrenzten Speicherraum einer Diskette nach bestimmten Kriterien in verschiedenen Dateien unterbringen wollen.

Wenn Sie z.B. zwei CE-1600M-Module benutzen und zu RAM Disks initialisiert haben, stehen Ihnen mit einer Diskettenseite gleichzeitig 122880 Bytes=120 KByte Speicherkapazität für die Datenverarbeitung zur Verfügung.

Zwar hat Ihr Arbeitsspeicher in diesem Fall nur etwas mehr als 10 KByte. Aber da die einzelnen Daten, die ja nicht mehr als 80 Zeichen umfassen können, jeweils sofort abgespeichert werden, kann Ihr BASIC-Bearbeitungsprogramm bis zu 10 KByte umfassen, um 120 KByte Daten zu verarbeiten.

Beispiel VI.1

Verwaltung großer Dateien

Sie wollen die Mitglieder eines übermäßig großen Vereins datenmäßig verwalten. Dann gehen Sie folgendermaßen vor.

1. Sie können der Streuung der Anfangsbuchstaben gemäß 15 Einzeldateien bilden. Die Mitglieder werden entsprechend den Anfangsbuchstaben in einer der 15 Dateien aufgenommen.
2. Jeweils 5 Dateien werden auf der aktuellen Diskettenseite und den beiden RAM-Disks eingerichtet und geöffnet.
3. Nach Verarbeitung der Daten werden die Dateien der RAM-Disks auf eine zweite und dritte Diskettenseite kopiert und auf den RAM-Disks gelöscht.

Ein Adressenverwaltungsprogramm könnte dann so aussehen, wie es im Beispiel VI.1 aufgelistet ist.

Dabei ist zu beachten, daß ich hier nur Rudimente meines Adressenprogramms EINGABE.ADR verwendet habe, um die Handhabung des MAXFILES-Befehls zu demonstrieren.

Genaueren Einblick erhalten Sie deshalb bei der Erläuterung des Adressenverwaltungsprogramms, in das Sie das Rudimentprogramm mit entsprechenden Änderungen einbauen können.

Noch einige Zusatzerklärungen:

In den Zeilen 10-24 wird für je einen oder zwei Buchstaben eine Datei angelegt. Als FILENAME wurden in diesem Beispiel jeweils die betreffenden Buchstaben genommen - ohne Extensionsangabe.

In Zeile 26 wird der erste Buchstabe des Namens festgestellt, um die Daten der Person der entsprechenden Datei zuordnen zu können.

Wenn Sie Daten von einzelnen Personen bearbeiten möchten, brauchen Sie nur die Datei zu öffnen, in der die betreffende Person steht.

Sie können bei anderen Problemstellungen genauso verfahren. Beispielsweise können Kunden- und Lieferantendateien nach einzelnen Sparten angelegt werden.

VI.4.2. #1 (OPEN, CLOSE, INPUT, OUTPUT, PRINT, APPEND)

Ich möchte auf diesen - für Disketten-Erfahrene problemlosen - Befehl kurz eingehen. Vor der Anschaffung des PC-1600 hatte ich nur andere SHARP-Pocketcomputer benutzt. So war für mich der Umgang mit Disketten neu. Trotz der Erklärung der Dateinummer (# 1) in der Bedienungsanleitung konnte ich zunächst wenig mit diesem Befehl anfangen.

Das #-Zeichen mit einer nachfolgenden Ziffer (1-15) bezieht sich auf eine bestimmte Datei, für die ein Befehl gegeben wird. Am Beispiel V.1 wird das deutlich. Beim Öffnen einer Datei bestimmen Sie z.B. durch AS #3, daß Sie sie als 3. Datei (engl. as=als) behandeln wollen. Das funktioniert allerdings nur, wenn Sie durch den MAXFILES-Befehl mindestens drei Dateien öffnen können.

Die Bearbeitungsnummer gilt nur für das momentane Bearbeitungsprogramm. Sie können ein- und dieselbe Datei einmal als # 3, ein anderes Mal als # 1 oder als # 14 behandeln.

VI.4.3. APPEND

Die Bedienungsanleitung weist darauf hin, daß dieser Bearbeitungsbefehl benutzt werden muß, wenn eine bestehende Datei ergänzt werden soll. Anderenfalls wird eine Datei neu angelegt - d.h. eine alte unter diesem Namen gelöscht -, wenn Sie sie unter einem bereits existierenden Namen FOR OUTPUT öffnen.

Da aber die Dateien nur seriell verwaltet und bearbeitet werden können - d.h. daß man in eine bereits existierende Datei nicht ohne weiteres etwas hineinschreiben sondern nur anfügen kann -, benutze ich den APPEND-Befehl überhaupt nicht. Es gibt "elegantere" Lösungen, Dateien zu ändern und zu ergänzen. Das können Sie an den Beispielen ADR und TVP in diesem Handbuch sehen.

VI.4.4. Filename.Extension

Im Allgemeinen verwende ich deutsche Ausdrücke für Begriffe aus der Computertechnik, auch wenn sie in der Fachsprache meistens englische Begriffsbezeichnungen haben. Natürlich ist der englische Begriff unumgänglich, wenn er ein Programmierbefehl ist.

Aber für Filename verwende ich nie das deutsche Synonym Dateiname und für Extension nie Gruppenseparator, um deutlich zu machen, daß es Bezeichnungen im Zusammenhang mit der Diskettenverwaltung sind. Wenn in diesem Kapitel VI. trotzdem öfter von Dateinamen gesprochen wird, ist immer Filename und Extension zusammen gemeint.

Sie sollten Programme und Dateien grundsätzlich mit Filename und Extension speichern. Sie erhalten so eine bessere Übersicht über Programme und Dateien, die auf einer Diskette gespeichert sind.

Ich gebe z.B. allen Grafikprogrammen die Extension-Bezeichnung GRA. Dadurch ist es mir möglich, durch den Befehl FILES"X:*GRA jederzeit einen schnellen Überblick über alle Grafikprogramme zu gewinnen. Das ist bei der vierzeiligen Display-Anzeige übersichtlicher, als wenn ich z.B. 48 Programm- und Dateinamen über Anzeigerollen lasse und mir dabei die Grafikprogramme herauspicken muß.

Entsprechend erhalten Parallelprogramme bestimmter Routinen denselben Filename (z.B. EINGABE.ADR und EINGABE.TVP bei den Adressenverwaltungs- und Textverarbeitungsprogrammen) oder bei veränderten Dateien, die neben den alten zum späteren Vergleich gespeichert werden, eine Ordnungsnummer als Extension-Bezeichnung.

VI.4.5. Schreibschuttschalter "P"

Ich empfehle noch einmal, jede Datei sofort nach Erstellen mit einem Schreibschutz zu versehen. Versuchen Sie dann, eine 2. Datei mit einem bereits vergebenen Dateinamen zu speichern, macht der Computer durch ERROR 159 darauf aufmerksam, daß Sie entweder den Schreibschutz entfernen müssen, um die alte Datei zu überschreiben, oder daß Sie einen anderen Dateinamen wählen müssen, um die neue Datei zu speichern.

VI.4.6. LOC

Dieser Befehl stellt nicht die Anzahl Ihrer eingegebenen Datensätze fest, sondern nur die Anzahl der Datensätze, die die Directory intern aus Ihren Datensätzen macht, um sie platzsparend verwalten zu können.

Unter VI.2.1. habe ich schon darauf hingewiesen, daß in jeden Sektor einer Diskette 512 Bytes gespeichert werden können. Um diese Speicherkapazität möglichst optimal auszunutzen, speichert das Directory die Daten zunächst in einem Eingabepuffer, um sie dann in interne Datensätze zu je 256 Bytes umzuwandeln und in zwei Datensätzen pro Sektor zu speichern.

Um dadurch nicht Ihre Datensätze zu verändern, arbeitet die Directory mit den Trennzeichen Komma und Semikolon, über deren Problematik ich im nächsten Unterabschnitt informiere.

Wenn Sie Auskunft über die Zahl Ihrer Daten gewinnen wollen, müssen Sie entsprechende Zählroutinen in Ihren Programmen einbauen, wie ich dies z.B. in TVP-2 in den Zeilen 55-57 getan habe. Dort gibt Z Auskunft über die Zahl der Datensätze.

Beispiel VI.2. und VI.3 sind zwei Programme, mit denen Sie den Byte-Umfang und die interne Datensatzzahl Ihrer Dateien bzw. den Umfang Ihrer BASIC-Programme feststellen können.

VI.4.7. Komma und Semikolon als Trennzeichen

Komma und Semikolon werden von der Directory als interne Trennzeichen benutzt, um z.B. Ihre Datensätze und die der Diskettendatei-Verwaltung zu verarbeiten. Dies gilt nicht für die Verwaltung von Diskettenprogrammen, weil diese durch den SAVE-Befehl gespeichert werden - nicht mit dem OPEN-Befehl, der für Variablenspeicherung zuständig ist.

Wie unterschiedlich die Trennzeichen wirken, können Sie durch das folgende Programm feststellen.

Beispiel VI.4

Prüfen der Trennzeichenwirkung

Listing im Beispielprogramm-Anhang zu diesem Kapitel

1. Geben Sie nach dem Eintippen des Programms einige Adressen ein. Die Eingabe beenden Sie dadurch, daß Sie das Tildezeichen (~) eingeben, wenn zur erneuten Eingabe eines Vornamens aufgefordert wird.

Sie werden feststellen:

- a) Werden die PRINT#-Daten durch Komma oder Semikolon getrennt gespeichert, werden alle Daten einer Person zu einer Variablen verarbeitet. Deshalb auch die Unterschiede der INPUT#-Anweisungen in den Zeilen 19, 23 und 31. Wären die Zeilen 19 und 23 wie Zeile 31 programmiert, bricht der Computer nach einiger Zeit mit einer ERROR-Meldung ab. Denn durch die Directory-Verarbeitung enthält die Diskettendatei weniger Daten, als Sie eingegeben haben - in diesem Beispiel pro Person ein Datum anstatt vier Daten.
 - b) Der Unterschied zwischen Komma und Semikolon besteht nur in der Art, wie die Directory Ihre Daten verarbeitet - mit oder ohne Einfügung von Leerstrings. Die Länge der Leerstrings hängt von Ihrer Dimensionierung der Daten ab.
 - c) Wie der Byte-Umfang der Daten durch das Komma als Trennzeichen "aufgebläht" wird, können Sie nachprüfen, wenn Sie die drei Dateien (KOMMA, SEM, NORMAL) durch das Beispielprogramm VI.2 auf ihre Byte-Zahl überprüfen. Obwohl in alle drei Dateien dieselben Daten eingegeben wurden, ist der Umfang der Dateien durch die unterschiedliche Bearbeitung verschieden groß.
2. Ändern Sie in Zeile 19 die Dimensionierung, z.B. in DIM N\$(3)*3 und lassen Sie die Dateien durch DEF A erneut ausdrucken, um zu sehen, wie die Diskettenverwaltung reagiert - Dimensionierung der von der Directory verarbeiteten Daten oder aber der einzelnen von Ihnen eingegebenen Daten.
 3. Das Überprüfen der Trennzeichenwirkung ergibt:

Sollen Daten in der eingegebenen Form verfügbar bleiben, sind sie beim Eingeben durch PRINT# durch ";" voneinander zu trennen.

Dann wird zwar auch die Directory die Daten so verarbeiten, daß sie in die internen Datensätze zu je 256 Bytes hineinpassen, aber die Daten getrennt behandeln, so daß Sie sie als Einzeldaten wieder abrufen können. Und das allein ist sinnvoll.

VI.4.8. Komma und Sonderzeichen in Dateien

Das Semikolon bereitet in Dateien als Teil von Variablen keine Schwierigkeiten. Es wird von der Directory als solches behandelt.

Das Komma wird von der Directory ausschließlich als Trennzeichen betrachtet und behandelt, wenn es in Dateien steht. Bei Diskettenprogrammen spielt das keine Rolle, da die Problematik nur im Zusammenhang mit dem PRINT#- und INPUT#-Befehl auftritt.

Trotzdem kann das Komma auch in Textdateien verwendet werden. Es wird als Code eingegeben.

Im Textverarbeitungsprogramm (Beispiel VI.2) habe ich dafür das Apostroph vorgesehen, da es in derselben Tastenreihe wie das Komma steht und ihm äußerlich ähnlich ist. Sie können bei Bedarf auch ein anderes Codezeichen verwenden.

In TVP 3, Zeile 60 und TVP 4, Zeile 67 wird das Codezeichen beim Drucken des Textes in das Komma umgewandelt.

Entsprechend habe ich das β durch f codiert, weil das f wieder entfernt an das β erinnert.

Ähnlich können Sie benötigte Sonderzeichen, die nicht auf direktem Wege durch die Tastatur in normaler oder KBII-Funktion abrufbar sind, codieren. Die Zeilen 59 in TVP 3 und 66 in TVP 4 zeigen, wie das β decodiert wird.

Benötigen Sie Sonderzeichen, die der ASCII-Zeichensatz des PC-1600 nicht enthält, können Sie dafür Grafikprogramme entwickeln und mit GOSUB-Routinen abrufen. Ich habe sogar meine Unterschrift auf diese Weise codiert.

VI.5. Dateiverwaltungsprogramme

Abschließend möchte ich an zwei Beispielen zeigen, wie Sie Dateien anlegen und durch die Dateiverwaltung des Diskettenlaufwerks sinnvoll behandeln können. Natürlich bietet der PC-1600 nicht alle Möglichkeiten, die Text- und andere Verarbeitungsprogramme großer und teurer Bürocomputer - mit vor allem wahlweisem Zugriff auf Daten im Unterschied zum seriellen oder sequentiellen Zugriff des PC-1600 - haben, und wie ich sie z.B. zur Erstellung dieses Buches durch den PC-5000 von SHARP nutzen konnte.

Aber gerade, weil ich gewohnt bin, mit Textverarbeitung zu arbeiten, hoffe ich, mit den Verarbeitungsprogrammen in diesem Handbuch brauchbare Datenverarbeitungsmöglichkeiten für das CE-1600F entwickelt zu haben. Ich selbst verwende sie überall, wo ich mein Büro im Aktentaschenformat mit hinnehme.

VI.5.1. Adressenverwaltung

Beispiel VI.5

Das Adressenverwaltungsprogramm besteht aus vier Teilprogrammen, die bei Bedarf in den Arbeitsspeicher des PC-1600 geladen werden. Die Bytezahl, die im folgenden bei den Kommentaren zu den einzelnen Teilprogrammen angegeben sind, beziehen sich auf das eigentliche Programm, d.h. ohne die REM-Zeilen - durch das Apostroph gekennzeichnet. Die Listings stehen wiederum im Beispielprogramm-Anhang dieses Kapitels.

ADR 1 - Eingabe einer Adressendatei (1311 Bytes)

Kommentar zu den Programmzeilen

- 18: Die Dimensionierung für die einzelnen Datenstrings eines Datensatzes (Personalien einer Person) sind nach Erfahrungswerten vorgenommen. Bei Bedarf ist anders zu dimensionieren. Dabei ist empfehlenswert, für mehrere Variablen mit ungefähr gleicher Zeichenzahl indizierte Variablen gleicher Art zu verwenden. In Zeile 18 sind Familienname, Straße und Wohnort durch NS(), Titu-

lierung und Telefonnummer durch TS() und das Geburtsdatum - unterteilt in Tag, Monat, Jahr - durch GS() jeweils mit gleicher Zeichenzahl indiziert.

Variablen, die den Umfang von 16 Zeichen haben, NS in Zeile 18, brauchen nicht indiziert zu werden. Sie sollten auch als Standardvariable (1 Zeichen für den Variablennamen) deklariert werden, da für sie kein Raum im Arbeitsspeicher reserviert wird.

Wird durch Raumreservierung für indizierte Variablen der Arbeitsspeicher zu sehr belastet - z. B. bei großen Dateien - sollte man auf indizierte Variablen weitgehend verzichten und sie durch Standardvariablen ersetzen. Für sie ist im ROM-Speicher hardwaremäßig der benötigte Raum reserviert.

Geburtsdaten sollten in Tag, Monat und Jahr unterteilt werden. Dann können Geburtstagskalender oder Jahrgangslisten abgerufen werden.

- 24: In dieser und den folgenden Zeilen weisen bei der Anforderung der Dateneingabe senkrechte Begrenzungen auf dem Display auf die Dimensionierung der entsprechenden Variablen hin.
- 38: Für die Beendigung von Eingaben verwende ich immer das Tildezeichen (~), weil ich es sonst nicht benötige. Sie können es durch andere Codes ersetzen.
- 44: In dieser und den folgenden Zeilen sind die normalen Titulierungen codiert. So brauche ich bei der Anforderung des Titels z.B. nur f oder F einzugeben, um den Titel "Frau" zu erzeugen.
- 53: Ebenso habe ich eine Wohnort-Routine geschaffen, die ich zur Demonstration unverändert wiedergebe.

Die meisten meiner Adressaten wohnen im Hamburger Stadtteil Rahlstedt (2000 Hamburg 73). Durch die Eingabe eines R oder r bei Anforderung der Postleitzahl wird diese Adresse erzeugt und sofort zur Anforderung der Telefonnummer gesprungen. Diese Routine kann entfallen oder den Bedürfnissen individuell angepaßt werden.

ADR 2 - Prüfen einer Adressendatei (2897 Bytes)

Unter VI.4.3. habe ich die Gründe dafür genannt, warum ich den APPEND-Befehl nie benutze. Eine elegante Lösung, Dateien zu ergänzen und sie vor allem zu ändern, zeigt ADR 2 (und TVP 2).

Dabei folgt diese Lösung immer folgender Konzeption:

1. Eine Datei wird zur Bearbeitung geladen.
2. Alle Daten werden einzeln eingelesen, geprüft und entweder unverändert (alt) oder verändert (neu) in einer Hilfsdatei vorübergehend gespeichert. Ebenso können Daten gelöscht oder eingefügt werden.
Die Anfügung neuer Daten kann nach einer aktuellen Datenzeile beliebig oft wiederholt werden.
3. Anschließend wird die Hilfsdatei in die Zieldatei umbenannt. Da die Quellendatei schreibgeschützt ist, muß ein neuer Dateiname genommen werden - am besten durch eine andere Extension. Z.B. Quellendatei: ADRESS.ALT (oder ADRESS.1) - Zieldatei: ADRESS.NEU (oder ADRESS.2).
4. Anschließend wird die Zieldatei schreibgeschützt und die Quellendatei bei Bedarf gelöscht. Ehe der Schreibschutz der Quellendatei entfernt wird, wird zur Sicherheit zweimal gefragt, ob die Quellendatei wirklich gelöscht werden soll, damit die alten Daten nicht fahrlässig vernichtet werden.

ADR 3 - Druck von Listen und Adressen (1169 Bytes)

Dieses Programm bietet zwei Alternativen:

1. Durch Drücken der Funktionstaste ! wird die Adressendatei in Listenform ausgedruckt. Dabei wird die Titulatur (Frau.. Herr..) durch das fehlende T\$(0) in Zeile 40 unterdrückt.

Zeile 42 bewirkt, daß nach 60 Adressen pro DIN-A4-Seite der Druck zum Einführen eines neuen Blattes unterbrochen wird.

2. Die Betätigung der Funktionstaste & veranlaßt den Druck der Datei in Adressenformat mit jeweils zwei Adressen nebeneinander auf einer DIN-A4-Seite. Man kann selbstklebende Bogen einführen und so Adressenaufkleber herstellen.

ADR 4 - Suchen in Adressendateien (1896 Bytes)

Durch Eingabe von Suchdaten können Listen nach Kriterien erstellt werden, z.B. Jahrgangs- und Wohngebietslisten.

Auf Einfügen eines Sortierprogramms sollten Sie grundsätzlich bei BASIC-Programmen verzichten. Denn hierfür erweist sich der PC-1600 - wie alle Computer - als zu langsam. Um z.B. eine Liste von hundert Adressen zu sortieren, benötigt der Computer im ungünstigsten Fall mehrere Stunden!

Für Sortierprobleme sollte man darum immer nur Maschinenprogramme einsetzen. Aber die Diskettenverwaltung erspart das Sortieren vielfach. Wenn Sie z.B. ADR 2 einsetzen, können Sie mit ADR 3 schnell Suchlisten erstellen, die namensmäßig alphabetisch geordnet sind.

VI.5.2. Textverarbeitung

Beispiel VI.6

Das Textverarbeitungsprogramm entspricht in den meisten Routinen dem Adressenverwaltungsprogramm. Es bietet folgende Alternativen:

1. Wahl des Textformats in DIN-A4 oder DIN-A5.

Das Textformat ist bei Erstellung einer Textdatei festzulegen. Die Textzeilen werden als indizierte Textstrings, vom Format abhängig, gespeichert.

Im DIN-A4-Format beträgt die Zeichenzahl pro Textzeile 78 oder 3 Displayzeilen - im DIN-A5-Format 52 Zeichen pro Textzeile in 2 Displayzeilen.

Die Textstrings werden zwar auf 79 bzw. 53 Zeichen formatiert. Das eine Mehrzeichen ist für Verarbeitungsmerkmale reserviert. Verwenden Sie in einer Textzeile kein Verarbeitungsmerkmal, ist es nicht als zusätzliches Textzeichen benutzbar, da nur bis zu 52 bzw. 78 Zeichen je String gedruckt werden.

2. Zeilenbehandlung

Am Ende einer Textzeile wird durch Eingabe eines Bearbeitungszeichens nach dem letzten Textzeichen eine besondere Druckausgabe veranlaßt.

Die Programmzeilen 47-50 in TVP 3 und 45-59 überprüfen, ob das letzte Stringzeichen ein Verarbeitungsmerkmal ist. Ist das letzte Textzeichen mit einem Verarbeitungszeichen (F,P,Z) identisch, ist noch ein Leerzeichen hinzuzufügen, damit es nicht als Verarbeitungszeichen behandelt wird.

3. Fettschrift

Soll eine Textzeile fett gedruckt werden, ist als letztes Zeichen ein E einzugeben, bzw. P, wenn sie zugleich zentriert und fett gedruckt werden soll.



4. Zentrierung

Soll eine Textzeile zentriert gedruckt werden, ist das Bearbeitungszeichen ein Z bzw. ein P (s. 2.)

5. Zeilenausgleich

Soll ein Text links- und rechtsbündig zugleich gedruckt werden (Zeilenausgleich), so ist der Textstring vor ENTER manuell zu behandeln.

Zwischen die einzelnen Wörter werden an geeigneten Stellen Leerzeichen gesetzt, bis das letzte Textzeichen am rechten Rand der 2. bzw. 3. Displayzeile steht.

6. Tabulieren

Auch das Tabulieren ist manuell vorzunehmen. Die senkrechten Begrenzungslinien der Funktionstasten sind dafür als Orientierungshilfen gut geeignet. Sie ermöglichen im DIN-A4-Format bis zu 18 Tabellen - im DIN-A5-Format bis zu 12.

7. Ändern, Löschen und Einfügung von Textzeilen

Hier gilt das, was unter ADR 2 - S. VI - 29 ausgeführt ist. Die Änderung von Texten ist besonders nützlich, wenn die Bearbeitungszeichen verändert werden sollen.

Beim Textverarbeitungsprogramm ist zu beachten:

1. Komma, β , Leerzeichen am Zeilenanfang

Kommata, β und Leerzeichen am Anfang einer Textzeile werden codiert eingegeben, weil Kommata und Leerzeichen am Anfang eines Textstrings von der Directory als Trennzeichen behandelt werden. Einzelheiten dazu finden Sie unter VI.4.7. + VI.4.8. Dort wird auch die Wahl der Codezeichen begründet.

Leerzeichen am Zeilenanfang werden als Tildezeichen ~ codiert.

Die Möglichkeiten, das β - und bei Bedarf andere Sonderzeichen (s. VI.4.8.) - zu nutzen, ist eine der Vorzüge gegenüber dem Apostroph-Textprogramm.

Wie das Apostroph für die Erstellung von Textprogrammen verwendet werden kann, wird in der Bedienungsanleitung unter 14-133f. erklärt. Nach unserer Definition von Dateien und Programmen (VI.3.3) können so aber nur Textprogramme erzeugt werden. Sie gestatten zwar die problemfreie Benutzung des Kommas. Aber weil alles, was in einer Programmzeile hinter einem Apostroph steht, vom PC-1600 als Kommentar behandelt wird, können in Textprogrammen nur Zeichen verwendet werden, die über die Tasten des PC-1600 in normaler oder KBII-Funktion zugänglich sind. Das β kann z.B. nicht benutzt werden. Es entfällt auch die Möglichkeit, Verarbeitungsroutinen - Fettschrift, Zentrierung - zu verwenden.

2. Absatz

Wird in eine Zeile nur ein Leerzeichen gegeben, entsteht eine Leerzeile, also ein neuer Absatz.

3. Beendigung der Texteingabe

Die Texteingabe wird durch Eingabe des Tildezeichens in die nächste Zeile beendet.

4. Anzahl der Textzeilen

In TVP 1, Zeile 20 wird die Anzahl der möglichen Textzeilen durch M dimensioniert. Erweist sich der Arbeitsspeicher als zu klein - im DIN-A4-Format werden z.B. durch M=255 20224 Bytes für die möglichen Textstrings (Zeilen) reserviert -, ist M niedriger zu wählen.

Sie können M auch größer dimensionieren (III).
Bedenken Sie aber dabei, daß durch ein Dimensions-
Array sehr viel Platz im Arbeitsspeicher für die
Reservierung benötigt wird.

Durch Anzeige der aktuellen Zeilennummer (mit Aus-
nahme der Titelzeile) und der möglichen Zeilenzahl
auf dem Display bei TVP 1 + 2 haben sie immer die
Übersicht über die Ausnutzung der Textspeicherka-
pazität. Gegebenenfalls sollten Sie für sehr lange
Texte mehrere Textdateien erstellen - z.B. für je-
des Kapitel eine -, bevor sie umdimensionieren.

5. Druckwahl

Sie haben die Alternative, zwischen Normaldruck
(TVP 3) und Schnelldruck (TVP 4) zu wählen.

Schnelldruck ist nur relativ. Während bei TVP 3
der Text Zeichen für Zeichen gedruckt wird, weil
jedes Zeichen auf Codierung geprüft wird, werden
TVP 4 die Zeilen, die keine Bearbeitungsmerkmale
enthalten, in der normalen Geschwindigkeit des PC-
1600P gedruckt.

Dafür erfordert die Erstellung der Textdateien für
eine Nutzung von TVP 4 besondere Sorgfalt. Enthält
eine Textzeile ein Komma oder ein Sonderzeichen,
ist am Ende der Zeile eine Tilde (~) einzugeben.

Weil dies leicht vergessen wird, ist TVP 4 beson-
ders störanfällig. Ich verwende nur TVP 3 und nut-
ze die Druckzeit für andere Aufgaben.

Beispiel U I . 1

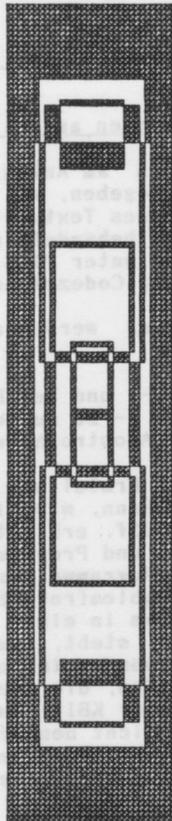
```

1:'      Verwaltung grosser Dateien
2:'
3:' Beispiel für Nutzung des MAXFILES-Befehles
4:'
5:MAXFILES =15
6:DIM N$(2)*26,T$(1)*13,G$(2)*2,H$(0)*6,P$(0)*4
7:OPEN "X:A"FOR OUTPUT AS #1:OPEN "X:B"FOR OUTPUT AS #2
8:OPEN "X:CD"FOR OUTPUT AS #3:OPEN "X:EF"FOR OUTPUT AS #4
9:OPEN "X:GH"FOR OUTPUT AS #5
10:OPEN "S1:IJ"FOR OUTPUT AS #6:OPEN "S1:KL"FOR OUTPUT AS #7
11:OPEN "S1:MN"FOR OUTPUT AS #8:OPEN "S1:OP"FOR OUTPUT AS #9
12:OPEN "S1:QR"FOR OUTPUT AS #10
13:OPEN "S2:S"FOR OUTPUT AS #11:OPEN "S2:TU"FOR OUTPUT AS #12
14:OPEN "S2:UV"FOR OUTPUT AS #13:OPEN "S2:XY"FOR OUTPUT AS #14
15:OPEN "S2:Z"FOR OUTPUT AS #15
16:CLS :PRINT "Name:"CURSOR 0,2:INPUT N$(0)
17:NL$=LEFT$(N$(0),1)
18:IF NL$="A"PRINT #1,N$(0)
19:IF NL$="B"PRINT #2,N$(0)
20:IF NL$="C"OR NL$="D"PRINT #3,N$(0)
21:IF NL$="G"OR NL$="H"PRINT #5,N$(0)
22:IF NL$="I"OR NL$="J"PRINT #6,N$(0)
23:IF NL$="K"OR NL$="L"PRINT #7,N$(0)
24:IF NL$="M"OR NL$="N"PRINT #8,N$(0)
25:IF NL$="O"OR NL$="P"PRINT #9,N$(0)
26:IF NL$="Q"OR NL$="R"PRINT #10,N$(0)
27:IF NL$="S"PRINT #11,N$(0)
28:IF NL$="T"OR NL$="U"PRINT #12,N$(0)
29:IF NL$="V"OR NL$="W"PRINT #13,N$(0)
30:IF NL$="Z"PRINT #15,N$(0)
31:IF N$(0)<>"~"GOTO 16
32:CLOSE #I:END
    
```

Beispiel U I . 2

```

1:' Byte-Umfang und Sektorenbelegung von Dateien
2:'
3:MAXFILES =1:DIM Z$(255)*80
4:CLS :WAIT 0:CURSOR 8,1:PRINT "Dateiname":CURSOR 8,3:INPUT F$
  :IF F$<>"~"GOTO 6ELSE END
5:WAIT 120:CLS :PRINT "Dateiname nicht gefunden  Eingabe w
  iederholen?":GOTO 4
6:ON ERROR GOTO 5
7:OPEN "X:"+F$FOR INPUT AS #1:N=LOF (1)
8:FOR I=0TO 155
    
```



```

9:IF EOF (1)GOTO 11
10:INPUT #1,Z$(I):NEXT I
11:D=LOC (1)
12:CLS :WAIT 0,S:CURSOR 4,1:PRINT "Display oder Druck":CURSOR 0
   ,3:PRINT "Display          Druck"
13:A=ASC INKEY$ :IF A<170R A>22GOTO 13
14:CLS :IF A=190R A=20GOTO 13
15:IF A<19GOTO 18
16:LPRINT "Die "";F$;"'-Datei hat";N;" Bytes
17:LPRINT "und";D;" interne Datensätze in";D/2;" Sektoren.":
   GOTO 20
18:PRINT "Die "";F$;"'-Datei hat";N;" Bytes
19:PRINT "und";D;" interne Datensätze in";D/2;" Sektoren."
20:CLOSE #<1>;WAIT :GOTO 4:END
    
```

Beispiel U I . 3

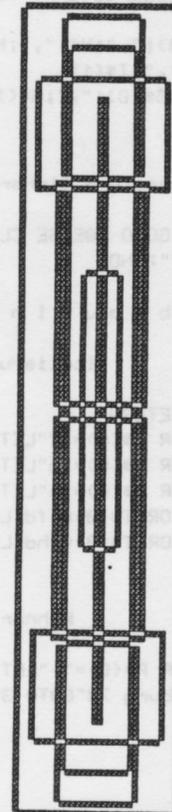
```

1:*      Byte-Umfang von BASIC-Programmen"
2:*
3:MAXFILES =1
4:CLEAR :CLS :WAIT 0:CURSOR 6,1:PRINT "Programm-Name":CURSOR 6
   ,3:INPUT F$:IF F$<>" "GOTO 6ELSE END
5:WAIT 120:CLS :PRINT "Programm nicht gefunden.  Eingabe wi
   ederholen!":40
6:ON ERROR GOTO 5
7:OPEN "X:"+F$FOR INPUT AS #1:N=LOF (1)
9:CLS :WAIT 0,S:CURSOR 4,1:PRINT "Display oder Druck":CURSOR 0
   ,3:PRINT "Display          Druck"
10:A=ASC INKEY$ :IF A<170R A>22GOTO 10
11:CLS :IF A=190R A=20GOTO 10
12:IF A<19GOTO 14
13:LPRINT "Das "";F$;"'-Programm hat";N;" Bytes.":CLOSE #1:GOTO
   4
14:WAIT :PRINT "Das "";F$;"'Programm      hat";N;" Bytes.":CLOSE
   #1:GOTO 4:END
    
```

Beispiel U I . 4

```

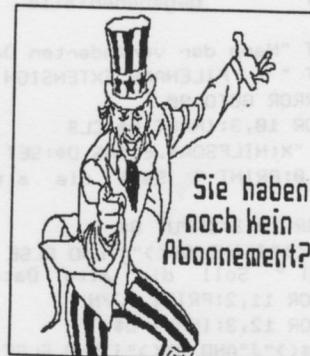
1:*Wirkung der Trennzeichen Komma und Semikolon
2:*
3:MAXFILES =3:INPUT "Zeichenanzahl je Datum ";D
4:DIM N$(3)*D
5:OPEN "X:KOMMA"FOR OUTPUT AS #1:OPEN "X:SEM"FOR OUTPUT AS #2:
   OPEN "X:NORMAL"FOR OUTPUT AS #3
6:CLS :INPUT "Vorname: ";N$(0):IF N$(0)=""GOTO 11
7:INPUT "Name: ";N$(1):INPUT "Str. ";N$(2)
8:INPUT "Ort: ";N$(3)
9:PRINT #1,N$(0),N$(1),N$(2),N$(3):PRINT #2,N$(0);N$(1);N$(2);
   N$(3)
10:PRINT #3,N$(0);",";N$(1);",";N$(2);N$(3):GOTO 6
11:CLOSE #1:CLOSE #2:CLOSE #3:ERASE N$(0)
12:"A"DIM N$(3)*80
13:OPEN "X:KOMMA"FOR INPUT AS #1:LPRINT "KOMMA:";LF 1
14:IF EOF (1)GOTO 16
15:INPUT #1,N$(0):LPRINT N$(0):GOTO 14
16:CLOSE #1:LF 2
17:OPEN "X:SEM"FOR INPUT AS #2:LPRINT "SEM:";LF 1
18:IF EOF (2)GOTO 20
19:INPUT #2,N$(0):LPRINT N$(0):GOTO 18
20:CLOSE #2:LF 2
21:OPEN "X:NORMAL"FOR INPUT AS #3:LPRINT "NORMAL,INPUT einer Ua
   riablen:";LF 1
22:IF EOF (3)GOTO 24
23:INPUT #3,N$(0):LPRINT N$(0):GOTO 22
24:CLOSE #3:LF 1
25:OPEN "X:NORMAL"FOR INPUT AS #3:LPRINT "NORMAL,INPUT aller Ua
   riablen:";LF 1
26:IF EOF (3)CLOSE #3:END
27:INPUT #3,N$(0),N$(1),N$(2),N$(3)
28:LPRINT N$(0);" ";N$(1);" ";N$(2);" ";N$(3):GOTO 26:END
    
```



```

1:'          ADRESSENVERWALTUNG
2:'
3:'          A D R
4:'
5:'          Teilprogramm 2
6:'
7:'          Prüfung einer Adressendatei
8:'          Adressen
9:'          erhalten - ändern - löschen - zufügen
10:'
11:'
12:'
13: CLEAR : WAIT 60 : CLS : CURSOR 8,0 : PRINT "A D R 2
14: PRINT " Prüfen und Bearbeiten einer Adressendatei": CLS
15:'
16:'          Vorbereitung zum Öffnen der
17:'          alten Datei (Quellendatei)
18:'          und der Hilfsdatei
19:'          und Parametervorgaben
20:'
21: MAXFILES = 2 : CLS : WAIT 0,5
22: DIM N$(2)*26, T$(1)*13, G$(2), H$(0)*6, P$(0)*4
23: PRINT " alte (Quellen) Datei": CURSOR 4,1 : PRINT "FILENAME.E
    XTENSION
24: CURSOR 10,3 : INPUT F$ : GOTO 30 : CLS
25:'
26:'          Öffnen der beiden Dateien
27:'          und Adressenbearbeitung
28:'
29: WAIT 60 : PRINT " Datei nicht gefunden": PRINT "Eingabe wiederh
    olen": CLS : WAIT 0,5 : GOTO 23
30: ON ERROR GOTO 29
31: OPEN "X:"+F$ FOR INPUT AS #1
32: OPEN "X:HILFSDAT.EI" FOR OUTPUT AS #2
33: ON ERROR GOTO 120
34: CLS : WAIT : INPUT #1, T$(0), N$, N$(0), N$(1), H$(0), P$(0), N$(2), T
    $(1), G$(0), G$(1), G$(2)
35: PRINT N$(0), " ", N$, G$(0); G$(1); G$(2), N$(1), H$(0), N$(2), T$(1)
36:'
37:' Bestimmung der einzelnen Bearbeitungsroutinen
38:'
39: WAIT 0 : PRINT "Bearbeitung der Adresse": CLS
40: CURSOR 9,1 : PRINT "Adresse": PRINT : PRINT " alt neu          zu
    weg"
41: WAIT 120 : A = ASC INKEY$ : IF A < 17 OR A > 22 OR A = 19 OR A = 20 GOTO 41
42: CLS : WAIT 0 : IF A = 18 GOTO 56
43: IF A = 21 GOTO 74
44: IF A = 22 GOTO 81
45: IF A = 17 PRINT " Adresse unverändert"
46:'
47:' Speicherung der Adresse in der Hilfsdatei
48:'
49: PRINT #2, T$(0); ", "; N$; ", "; N$(0); ", "; N$(1); ", "; H$(0); ", "; P$(0)
    ); ", ";
50: PRINT #2, N$(2); ", "; T$(1); ", "; G$(0); ", "; G$(1); ", "; G$(2): IF A =
    22 GOTO 82 ELSE GOTO 34
51:'
52:'          Subroutinen
53:'
54:'          Ändern der Adresse
55:'
56: CURSOR 7,0 : PRINT "Adresse ändern": PRINT N$, N$(0) : GOSUB 58
57: GOSUB 84 : IF Q$ <> "J" AND Q$ <> "j" GOTO 49 ELSE GOTO 74
58: PRINT "Titel: "; T$(0) : LINE (78,16)-(79,31),, , B : CURSOR 0,3 :
    INPUT T$(0) : GOSUB 108
59: CLS : PRINT "Vorname: "; N$ : LINE (102,8)-(103,31),, , B : CURSOR 0,
    2 : INPUT N$
60: CLS : PRINT "Name: "; N$(0) : CURSOR 0,2 : INPUT N$(0)
61: CLS : PRINT "Strasse: "; N$(1) : CURSOR 0,2 : INPUT N$(1)
62: CLS : PRINT "Hausnummer: "; H$(0) : LINE (42,8)-(43,31),, , B :
    CURSOR 0,2 : INPUT H$(0)
63: CLS : PRINT "Postleitzahl: "; P$(0) : LINE (30,8)-(31,31),, , B :
    CURSOR 0,2 : INPUT P$(0) : GOTO 118
64: CLS : PRINT "Wohnort: "; N$(2) : CURSOR 0,2 : INPUT N$(2)
65: CLS : PRINT "Telefon: "; T$(1) : LINE (72,8)-(73,31),, , B : CURSOR 0
    ,2 : INPUT T$(1)

```



```

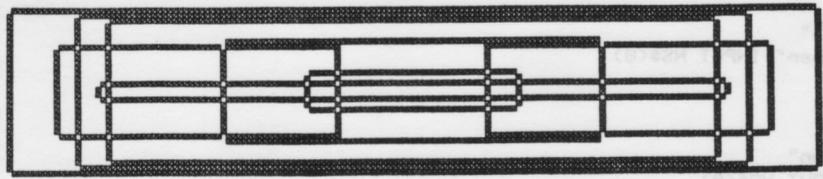
66:CLS :PRINT "Geburtsstag:";G$(0):LINE (18,8)-(19,31),,,B:
   CURSOR 0,2:INPUT G$(0)
67:CLS :PRINT "Geburtsmonat:";G$(1):LINE (18,8)-(19,31),,,B:
   CURSOR 0,2:INPUT G$(1)
68:CLS :PRINT "Geburtsjahr:";G$(2):LINE (18,8)-(19,31),,,B:
   CURSOR 0,2:INPUT G$(2)
69:RETURN
70:'
71:'      Speichern der aktuellen Adresse
72:'      und Zufügung einer neuen
73:'
74:PRINT "Adresse zufügen"
75:PRINT #2,T$(0);",";N$;",";N$(0);",";N$(1);",";H$(0);",";P$(0)
   );",";
76:PRINT #2,N$(2);",";T$(1);",";G$(0);",";G$(1);",";G$(2):GOSUB
   58
77:GOSUB 84:IF Q$<>"J"AND Q$<>"j"GOTO 49ELSE GOTO 74
78:'
79:'      Löschen der Adresse
80:'
81:PRINT "Adresse löschen"
82:GOSUB 84
83:IF Q$<>"J"AND Q$<>"j"GOSUB 34ELSE GOSUB 58:GOTO 49
84:CLS :INPUT "Adresse zufügen      J/N ";Q$:IF Q$<>"N"AND Q$<>
   "n"AND Q$<>"J"AND Q$<>"j"GOTO 84
85:CLS :RETURN
86:'
87:'      Erzeugung, Speichern und Sichern
88:'      der neuen Adressendatei (Zieldatei),
89:'      gegebenenfalls Löschen der Quelldatei
90:'
91:WAIT 60:PRINT "Umbenennung der Hilfsdatei      zur Zieldatei
92:PRINT "      gegebenenfalls      Löschen der alten Datei":
   CLS
93:PRINT "Name der veränderten Datei
94:PRINT "      FILENAME.EXTENSION
95:ON ERROR GOTO 96
96:CURSOR 10,3:INPUT D$:CLS
97:NAME "X:HILFSDAT.EI"AS D$:SET "X:"+D$,"P"
98:WAIT 0:PRINT " Soll die alte Datei      gelöscht werden?
   J/N
99:CURSOR 13,3:INPUT Q$
100:IF Q$<>"J"AND Q$<>"j"END ELSE CLS
101:PRINT " Soll die alte Datei wirklich gelöscht werden?":
   CURSOR 11,2:PRINT "J/N"
102:CURSOR 12,3:INPUT Q$
103:IF Q$<>"J"AND Q$<>"j"END ELSE CLS
104:SET "X:"+F$," ":KILL "X:"+F$
105:'
106:'      Titulierung
107:'
108:IF T$(0)=" "RETURN
109:IF T$(0)="F"OR T$(0)="f"LET T$(0)="Frau":RETURN
110:IF T$(0)="H"OR T$(0)="h"LET T$(0)="Herrn":RETURN
111:IF T$(0)="A"OR T$(0)="a"LET T$(0)="An":RETURN
112:IF T$(0)="FD"OR T$(0)="fd"LET T$(0)="Frau Dr.":RETURN
113:IF T$(0)="HD"OR T$(0)="hd"LET T$(0)="Herrn Dr."
114:RETURN
115:'
116:'      Wohnort
117:'
118:IF P$(0)="R"OR P$(0)="r"LET P$(0)="2000":N$(2)="Hamburg 73"
119:IF N$(2)="Hamburg 73"GOTO 65ELSE 64
120:CLOSE #1:CLOSE #2:GOTO 91:END

1:'      ADRESSENVERWALTUNG
2:'
3:'      A D R
4:'
5:'      Teilprogramm 3
6:'
7:'      Drucken von Namenslisten und Adressen
8:'
9:'
10:CLEAR :WAIT 60:CLS :CURSOR 8,0:PRINT "A D R 3

```

```

11:CURSOR 8,2:PRINT "Drucken von          Namenslisten und Adress
    en":CLS
12:'
13:'Eingabe der Adressendatei und Dimensionierung
14:'
15:MAXFILES =1:WAIT 0,S
16:DIM N$(2)*26,T$(1)*13,G$(2)*2,H$(0)*6,P$(0)*4
17:'
18:'          Öffnen der Datei
19:'
20:CURSOR 8,0:PRINT "Dateiname":PRINT "    FILENAME.EXTENSION":
    CURSOR 7,3:INPUT F$:CLS :GOTO 22
21:WAIT 60:PRINT "Datei nicht gefunden. Eingabe wiederholen.":
    CLS :GOTO 20
22:ON ERROR GOTO 21
23:OPEN "X:"+F$FOR INPUT AS #1
24:PAPER C:PCONSOLE "LPT1:",78,2,2
25:'
26:'          Liste oder Adressenformat
27:'
28:WAIT 0:PRINT "Liste oder Adressenformat":CURSOR 4,2:PRINT "
    < Funktionstaste >
29:PRINT "  ?      &
30:WAIT 120:A=ASC INKEY$ :IF A<>17AND A<>22GOTO 30
31:CLS :WAIT 0:IF A=17GOTO 35ELSE 46
32:'
33:'          Namensliste
34:'
35:WAIT 60:CURSOR 8,2:PRINT "Namensliste":WAIT 0:Z=0
36:GOSUB 38
37:GOTO 41
38:IF EOF (1)CLOSE #1:GOTO 54
39:INPUT #1,T$(0),N$,N$(0),N$(1),H$(0),P$(0),N$(2),T$(1),G$(0),
    G$(1),G$(2)
40:RETURN
41:LPRINT N$;" ";N$(0);" ";N$(1);" ";H$(0);" ";P$(0);" ";
42:LPRINT N$(2);" ";T$(1);" ";G$(0);" ".G$(1);" ".G$(2)
43:Z=Z+1:IF Z>1AND (Z-1)/60=INT ((Z-1)/60)STOP
44:GOTO 36
45:END
46:WAIT 60:CURSOR 6,2:PRINT "Adressenformat":WAIT 0
47:GOSUB 38
48:IF C=1LF -4:PCONSOLE "LPT1:",40,2,40
49:IF C=0PCONSOLE "LPT1:",40,2,4:LF 2
50:LPRINT T$(0):LPRINT N$;" ";N$(0):LPRINT N$(1);" ";H$(0):
    LPRINT P$(0);" ";N$(2)
51:Z=Z+1:IF Z/2=INT (Z/2)LET C=0ELSE LET C=1
52:IF Z>1AND (Z-1)/20=INT ((Z-1)/20)STOP
53:GOTO 47
54:CLS :CURSOR 7,1:INPUT "Datei-Ende          erneuter Druck? (J
    /N) ";Q$
55:IF Q$="J"OR Q$="j"GOTO 23:END
    
```



```

1:'          ADRESSENERWALTUNG
2:'
3:'          A D R
4:'
5:'          Teilprogramm 4
6:'
7:'          Suchen in Adressendateien
8:'
9:'
10:CLEAR :WAIT 60:CLS :CURSOR 8,0:PRINT "A D R 4
11:CURSOR 0,2:PRINT "Suchen in Adressendateien":CLS
12:'
13:'          Eingabe der Adressendatei
14:'
15:MAXFILES =1:WAIT 0,S
16:DIM N$(2)*26,T$(1)*13,G$(2)*2,H$(0)*6,P$(0)*4,NS$(0)*26
    
```

```

17:*
18:*           Öffnen der Datei
19:*
20:CURSOR 8,0:PRINT "Dateiname":PRINT "      FILENAME.EXTENSION":
      CURSOR 7,3:INPUT F$:CLS :GOTO 22
21:WAIT 60:PRINT "Datei nicht gefunden. Eingabe wiederholen.":
      CLS :GOTO 20
22:ON ERROR GOTO 21
23:OPEN "X:"+F$FOR INPUT AS #1
24:PAPER C:PCONSOLE "LPT1:",78,2,2
25:*
26:*           Suchen der gewünschten Daten
27:*
28:CLS :WAIT 0,S:PRINT "Suchen: 1 Name  2 Vorname":PRINT "3 St
      r. 4 Plz 5 Ort 6 Tel
29:PRINT "7 Geb-Tag 8 Monat 9 Jahr":INPUT "0 Zahlentaste + EN
      TER ";A
30:ON AGOSUB 40,45,50,55,60,65,70,75,80,86
31:IF EOF (<1)GOTO 85
32:INPUT #1,T$(0),N$,N$(0),N$(1),H$(0),P$(0),N$(2),T$(1),G$(0),
      G$(1),G$(2)
33:RETURN
34:LPRINT T$(0);" ";N$;" ";N$(0);" ";N$(1);" ";H$(0);" ";P$(0);
      " ";
35:LPRINT N$(2);" ";T$(1);" ";G$(0);".";G$(1);".";G$(2)
36:RETURN
37:*
38:*           Suchroutinen
39:*
40:CLS :PRINT "      Suche nach Namen"
41:CURSOR 5,1:PRINT "gesuchter Name":INPUT NS$(0)
42:GOSUB 31
43:IF N$(0)<>NS$(0)GOTO 42ELSE GOSUB 34
44:GOTO 42
45:CLS :PRINT "      Suche nach Vornamen"
46:CURSOR 4,1:PRINT "gesuchter Vorname":INPUT NS$(0)
47:GOSUB 31
48:IF N$(0)<>NS$(0)GOTO 47ELSE GOSUB 34
49:GOTO 47
50:CLS :PRINT "      Suche nach Strassen"
51:CURSOR 5,1:PRINT "gesuchte Strasse":INPUT NS$(0)
52:GOSUB 31
53:IF N$(1)<>NS$(0)GOTO 52ELSE GOSUB 34
54:GOTO 52
55:CLS :PRINT "Suche nach Postleitzahlen"
56:CURSOR 7,1:PRINT "gesuchte Plz":INPUT NS$(0)
57:GOSUB 31
58:IF P$(0)<>NS$(0)GOTO 57ELSE GOSUB 34
59:GOTO 57
60:CLS :PRINT "      Suche nach Wohnorten"
61:CURSOR 6,1:PRINT "gesuchter Ort":INPUT NS$(0)
62:GOTO 31
63:IF N$(2)<>NS$(0)GOTO 62ELSE GOSUB 34
64:GOTO 62
65:CLS :PRINT "      Suche nach Tel.-Nummern"
66:CURSOR 3,1:PRINT "gesuchte Tel.-Nummer":INPUT NS$(0)
67:GOSUB 31
68:IF T$(1)<>NS$(0)GOTO 67ELSE GOSUB 34
69:GOTO 67
70:CLS :PRINT "      Suche nach Geburts-Tagen"
71:CURSOR 6,1:PRINT "gesuchter Tag":INPUT NS$(0)
72:GOSUB 31
73:IF G$(0)<>NS$(0)GOTO 72ELSE GOSUB 34
74:GOTO 72
75:CLS :PRINT "Suche nach Geburts-Monaten"
76:CURSOR 5,1:PRINT "gesuchter Monat":INPUT NS$(0)
77:GOSUB 31
78:IF G$(1)<>NS$(0)GOTO 77ELSE GOSUB 34
79:GOTO 77
80:CLS :PRINT "Suche nach Geburts-Jahren"
81:CURSOR 6,1:PRINT "gesuchtes Jahr":INPUT NS$(0)
82:GOSUB 31
83:IF G$(2)<>NS$(0)GOTO 82ELSE GOSUB 34
84:GOTO 82
85:CLOSE #1:GOTO 23
86:END

```

Sei kein Frosch !
Lies "Alles für SHARP Computer"



Beispiel VI. 6

```

1: TEXTVERARBEITUNGSPROGRAMM
2:
3: T U P
4:
5: Teilprogramm 1
6:
7: Eingabe einer Textdatei
8:
9:
10:CLS :WAIT 60:CURSOR 8,0:PRINT "T U P 1":CURSOR 1,2:PRINT "E
    ingabe einer Textdatei
11:
12: Benennung der Textdatei
13: und Parametereinstellung (Druckformat)
14:
15:CLEAR :CLS :WAIT 0,S
16:MAXFILES =1
17:CURSOR 8,0:PRINT "Dateiname":PRINT " FILENAME.EXTENSION":
    CURSOR 9,3:INPUT F$:CLS
18:CURSOR 8,0:PRINT "Textformat":CURSOR 9,3:INPUT "DIN A";A:CLS
19:IF A=4LET Z=79ELSE LET Z=53
20:M=STATUS 0/Z:M=M*7:M=(STATUS 0-M)/Z:IF M>255LET M=255
21:
22: Erstellung der Textdatei
23:
24:DIM Z$(M)*Z:CURSOR 9,0:PRINT "Titelzeile":INPUT Z$(0)
25:FOR E=1TO M
26:PRINT E;"<"M;">"
27:INPUT Z$(E):IF LEN Z$(E)<=ZTHEN 29ELSE BEEP 3,200,200
28:PAUSE "Zeile zu lang":GOTO 26
29:IF Z$(E)<>" "OR E<MNEXT E
30:M=E
31:
32: Diskettenspeicherung der Textdatei
33:
34:OPEN "X:"+F$FOR OUTPUT AS #1
35:FOR A=0TO M:PRINT #1,Z$(A):NEXT A
36:CLOSE #1
37:
38: Datei mit Schreibschutz sichern
39:
40:SET "X:"+F$,"P":END

1: TEXTVERARBEITUNGSPROGRAMM
2:
3: T U P
4:
5: Teilprogramm 2
6:
7: Prüfung einer Textdatei
8: Zeilen
9: erhalten - ändern - löschen - zufügen
10:
11:
12:
13:CLEAR :WAIT 60:CLS :CURSOR 8,0:PRINT "T U P 2
14:PRINT " Prüfen und Bearbeiten einer Textdatei":CLS
15:
16: Vorbereitung zum Öffnen der
17: alten Datei (Quellendatei)
18: und der Hilfsdatei
19: und Parametervorgaben
20:
21:MAXFILES =2:CLS :WAIT 0,S
22:CURSOR 7,0:PRINT "Textformat der":PRINT " zu bearbeitenden
    Datei
23:CURSOR 12,3:INPUT "DIN A";A:CLS
24:IF A=4LET Z=79ELSE LET Z=53
25:M=255:DIM Z$(M)*Z
26:CURSOR 4,1:PRINT "FILENAME.EXTENSION
27:CURSOR 10,3:INPUT F$:GOTO 33:CLS
28:
29: Öffnen der beiden Dateien
30: und Textbearbeitung
31:

```

Der Schlüssel zu manchem Problem :



```

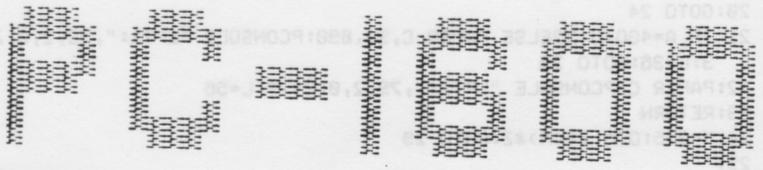
32:WAIT 60:PRINT " Datei nicht gefunden":PRINT "Eingabe wiederh
    olen":CLS :WAIT 0,S:GOTO 26
33:ON ERROR GOTO 32
34:OPEN "X:"+F$FOR INPUT AS #1
35:OPEN "X:HILFSDAT.EI"FOR OUTPUT AS #2
36:FOR J=0TO M
37:IF EOF (1)CLOSE #1:CLOSE #2:GOTO 88
38:CLS :INPUT #1,Z$(J)
39:IF J=0CURSOR 8,0:PRINT "Titelzeile":GOTO 41
40:PRINT J;". Zeile"
41:WAIT :PRINT Z$(J)
42:'
43:'Bestimmung der einzelnen Bearbeitungsrountinen
44:'
45:WAIT 0:PRINT "Bearbeitung der Zeile":CLS
46:CURSOR 10,1:PRINT "Zeile":PRINT :PRINT " alt neu
    weg"
47:WAIT 120:A=ASC INKEY$ :IF A<17OR A>22OR A=19OR A=20GOTO 47
48:CLS :WAIT 0:IF A=18GOSUB 62
49:IF A=21GOTO 68
50:IF A=22GOTO 75
51:IF A=17PRINT "Zeile unverändert"
52:'
53:' Neunumerierung der neuen und alten Zeilen
54:'
55:Z=J+N:Z$(Z)=Z$(J):PRINT #2,Z$(Z)
56:IF A<>18NEXT JELSE GOTO 81
57:'
58:'          S u b r o u t i n e n
59:'
60:'          Ändern der aktuellen Textzeile
61:'
62:CURSOR 8,0:PRINT "Zeile ändern":PRINT Z$(J)
63:INPUT Z$(J):RETURN
64:'
65:'          Speichern der aktuellen Zeile
66:'          und Zufügung einer neuen
67:'
68:PRINT "Zeile zufügen"
69:Z=Z+N:Z$(Z)=Z$(J):PRINT #2,Z$(Z)
70:Z=J+1:N=N+1:PRINT Z$(J)
71:INPUT Z$(Z):PRINT #2,Z$(Z):GOTO 81
72:'
73:'          Löschen der aktuellen Textzeile
74:'
75:PRINT "Zeile löschen"
76:N=N-1:GOTO 81
77:'
78:' Soll nach der aktuellen Zeile (noch) eine
79:' neue Zeile angefügt werden?
80:'
81:INPUT "(weitere Zeile) zufügen?      J/N ";Q$:CLS
82:IF Q$="N"AND Q$="n"NEXT JELSE GOTO 69
83:'
84:'          Erzeugung, Speichern und Sichern
85:'          der neuen Textdatei (Zieldatei),
86:'          gegebenenfalls Löschen der Quelldatei
87:'
88:WAIT 60:PRINT "Umbenennung der Hilfsdatei      zur Zieldatei
89:PRINT "          gegebenenfalls      Löschen der alten Datei":
    CLS
90:PRINT "Name der veränderten Datei
91:PRINT "      FILENAME.EXTENSION
92:ON ERROR GOTO 93
93:CURSOR 10,3:INPUT D$:CLS
94:NAME "X:HILFSDAT.EI"AS D$:SET "X:"+D$,"P"
95:WAIT 0:PRINT " Soll die alte Datei      gelöscht werden?
    J/N
96:CURSOR 13,3:INPUT Q$
97:IF Q$="N"OR Q$="n"END ELSE CLS
98:PRINT " Soll die alte Datei wirklich gelöscht werden?":
    CURSOR 11,2:PRINT "J/N"
99:CURSOR 12,3:INPUT Q$
100:IF Q$="N"OR Q$="n"END ELSE CLS
101:SET "X:"+F$," " :KILL "X:"+F$:END
    
```

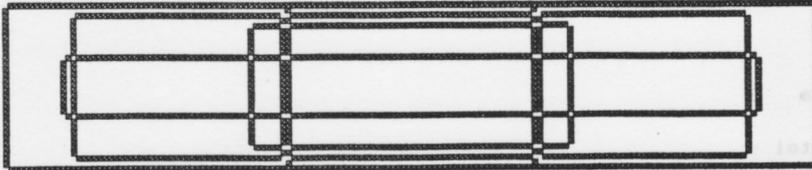


```

1: TEXTVERARBEITUNGSPROGRAMM
2:
3: T U P
4:
5: Teilprogramm 3
6:
7: Drucken einer Textdatei
8:
9:
10: Eingabe der Datei
11: und Parametervorgaben für den Drucker
12:
13: CLEAR :CLS :WAIT 60:CURSOR 8,0:PRINT "T U P 3":CURSOR 11,2:
    PRINT "Druck"
14: MAXFILES =1
15: TEXT :CLS :WAIT 0,S
16: CURSOR 8,0:PRINT "Dateiname":PRINT " FILENAME.EXTENSION":
    CURSOR 7,3:INPUT F$:CLS
17: CURSOR 8,0:PRINT "Textformat":CURSOR 9,3:INPUT "DIN A";A:CLS
18: IF A<4OR A>5GOTO 17
19: GOSUB 21
20: GOTO 24
21: IF A=4GOTO 22ELSE PAPER C,30,890:PCONSOLE "LPT1:",53,2,0:Z=5
    3:L=36:GOTO 23
22: PAPER C:PCONSOLE "LPT1:",79,2,0:Z=53:L=56
23: RETURN
24: M=255:DIM Z$(M)*Z:GOTO 29
25:
26: Einlesen der Textzeilen
27:
28: WAIT 60:PRINT "Datei nicht gefunden. Eingabe wiederholen":
    ERASE Z$(:)GOTO 15
29: ON ERROR GOTO 28
30: OPEN "X:"+F$FOR INPUT AS #1
31: FOR J=0TO M
32: INPUT #1,Z$(J):LPRINT
33:
34: Überprüfung des Textstrings auf Dateiende
35:
36: IF Z$(J)="~"CLOSE #1:END
37:
38: Überprüfung der Zeilen pro Druckseite
39:
40: IF J>0AND J/L=INT (J/L)STOP :GOSUB 21
41:
42: Überprüfung der Bearbeitungszeichen
43:
44: U=LEN Z$(J)
45: T=(Z-U-1)/2
46: R$=RIGHT$ (Z$(J),1)
47: IF R$="Z"TAB T
48: IF R$<>"P"GOTO 50
49: GRAPH :GLCURSOR (T*12,0):SORGN :GOTO 67
50: IF R$="F"GOTO 66ELSE GOSUB 55
51: NEXT J
52:
53: Drucken - Zeichen für Zeichen
54:
55: FOR B=1TO U
56: B$=MID$ (Z$(J),B,1)
57: IF B$<>R$GOTO 59
58: IF R$="F"AND B=UOR R$="P"AND B=UOR R$="Z"AND B=UOR R$="~"AND
    B=URETURN
59: IF B$="f"LPRINT CHR$ &E1;:GOTO 62
60: IF B$=","LPRINT ",":GOTO 62
61: IF B$="~"LPRINT " ";ELSE LPRINT B$;
62: NEXT B:RETURN
63:
64: Subroutine für Fettschrift
65:
66: GRAPH
67: FOR P=0TO 2
68: GLCURSOR (P,0):GOSUB 55:NEXT P:SORGN :IF B=UGOTO 69ELSE
    RETURN
69: TEXT :NEXT J:END

```





```

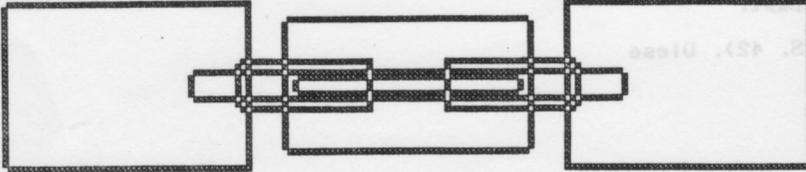
1: TEXTVERARBEITUNGSPROGRAMM
2:
3: T U P
4:
5: Teilprogramm 4
6:
7: "Schnelldruck" einer Textdatei
8:
9:
10: Eingabe der Datei
11: und Parametervorgaben für den Drucker
12:
13: CLEAR :CLS :WAIT 60:CURSOR 8,0:PRINT "T U P 4":CURSOR 8,2:
    PRINT "Schnelldruck"
14: MAXFILES =1
15: CLS :WAIT 0,S
16: CURSOR 8,0:PRINT "Dateiname":PRINT " FILENAME.EXTENSION":
    CURSOR 7,3:INPUT F$:CLS
17: CURSOR 9,0:PRINT "Textformat":CURSOR 11,3:INPUT "DIN A";A:
    CLS
18: IF A<40R A>5GOTO 17
19: GOSUB 21
20: GOTO 24
21: IF A=4GOTO 22ELSE PAPER C,30,890:PCONSOLE "LPT1:",53,2,0:Z=5
    3:L=36:GOTO 23
22: PAPER C:PCONSOLE "LPT1:",79,2,0:Z=79:L=56
23: RETURN
24: M=255:DIM Z$(M)*Z:GOTO 29
25:
26: Einlesen der Textzeilen
27:
28: WAIT 60:PRINT "Datei nicht gefunden. Eingabe wiederholen":
    CLS :ERASE Z$(:GOTO 15
29: ON ERROR GOTO 28
30: OPEN "X:"+F$FOR INPUT AS #1
31: FOR J=0TO M
32: INPUT #1,Z$(J)
33:
34: Überprüfung des Textstrings auf Dateiende
35:
36: IF Z$(J)=""CLOSE #1:END
37:
38: Überprüfung der Zeilen pro Druckseite
39:
40: IF J>0AND J/L=INT (J/L)STOP :GOSUB 21
41:
42: Überprüfung des letzten Zeilenzeichens
    auf Bearbeitungs-Charakter
43:
44:
45: R$=RIGHT$(Z$(J),1)
46: IF R$=""OR R$="F"OR R$="P"OR R$="T"OR R$="Z"GOTO 55
47:
48: Druck einer Textzeile im Ganzen
49:
50: LPRINT LEFT$(Z$(J),Z-1)
51: NEXT J
52:
53: Überprüfung der Bearbeitungszeichen
54:
55: T=Z/2-(LEN Z$(J)-2)/2
56: IF R$="Z"TAB T
57: IF R$<"P"GOTO 59
58: GRAPH :GLCURSOR (T*12,0):SORGN :GOTO 74
59: IF R$="F"GOTO 73ELSE GOSUB 64
60: TEXT :LPRINT :NEXT J
61:
62: Drucken - Zeichen für Zeichen
63:

```



```

64:FOR B=0TO LEN Z$(J)-2
65:B$=MID$(Z$(J),B+1,1)
66:IF B$="f" LPRINT CHR$ &E1;:GOTO 69
67:IF B$=" " LPRINT ", ";:GOTO 69
68:IF B$="." LPRINT " ";ELSE LPRINT B$;
69:NEXT B:RETURN
70:
71: Subroutine für Fettschrift
72:
73:GRAPH
74:FOR P=0TO 2
75:GLCURSOR (P,0):GOSUB 64:NEXT P:SORGN :IF B=LEN Z$(J)-1GOTO 7
ELSE RETURN
76:TEXT :LPRINT :NEXT J:END
    
```



Schiebemax

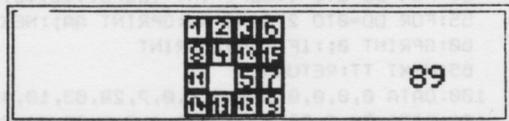
Bei diesem Spiel müssen die Zahlen 1-15 in die richtige Reihenfolge gebracht werden. Das ist Alles. Rechts wird die Anzahl der benötigten Züge angezeigt.

```

1:REM *****
2:REM *** Schiebemax **
3:REM *by Hans Hölzel**
4:PEM
5:REM *****
10:CLS :CLEAR :WAIT 0:RANDOM
20:DIM A$(17,17),O(17)
30:DIM P(9,9)
40:X=-1:Y=1:OO=0
50:A$(1)="FF818985BF8181FF"
60:A$(2)="FF81A583ABA581FF"
70:A$(3)="FF81E7C3DBFF81FF"
80:A$(4)="FF817F91F99181FF"
90:A$(5)="FF81AFCBCBB81FF"
100:A$(6)="FF81BDCBCBB381FF"
110:A$(7)="FF8183F38B8781FF"
120:A$(8)="FF81AD03D3AD81FF"
130:A$(9)="FF818DD3D3AD81FF"
140:A$(10)="FF89B08199A599FF"
150:A$(11)="FF89B08189B081FF"
160:A$(12)="FF89B081B5AD81FF"
170:A$(13)="FF89FD8105FD81FF"
180:A$(14)="FF89B0819DB991FF"
190:A$(15)="FF89FD0E1DD05F5FF"
200:A$(16)="0000000000000000"
205:PRINT :PRINT " Bitte warten"
210:Z=15:YY=0
220:YY=YY+1
230:O=RND Z:O(YY)=0
240:IF YY>2GOSUB "AA"
250:IF YY=ZCLS :GOTO 350
260:GOTO 220
270:"AA"
280:II=YY
290:II=II-1
300:IF O(II)=O(YY)LET O(YY)=0:YY=YY-1
310:IF II=1GOTO 330
320:GOTO 290
330:RETURN
340:Y=Y+1
350:X=X+1
360:GCURSOR 46+(Y*8),X*8
370:OO=OO+1
380:AA=O(OO)
390:GPRINT A$(AA)
400:P(Y,X)=AA
410:IF X=3AND Y=4GOTO 440
420:IF X=3LET X=-1:GOTO 340
430:GOTO 350
440:YY=78:XX=24:I=-1
450:GCURSOR YY,XX:GPRINT A$(16)
460:I=I+1:CURSOR 20,2:PRINT I
    
```

```

470:E$=INKEY$
480:IF ASC E$<50OR ASC E$>56GOTO 470
490:IF E$="8"GOTO "8"
500:IF E$="2"GOTO "2"
510:IF E$="6"GOTO "6"
520:IF E$="4"GOTO "4"
530:IF E$="3"OR E$="5"OR E$="7"GOTO 470
540:"2"
550:X=XX/8:Y=(YY-46)/8
560:IF X=0BEEP 3:GOTO 470
570:GCURSOR YY,XX
580:Z=P(Y,X-1):GPRINT A$(Z)
590:GCURSOR YY,XX-8:GPRINT A$(16)
600:Y=(YY-46)/8:X=XX/8
610:P(Y,X-1)=16
620:P(Y,X)=Z
630:YY=YY:XX=XX-8
640:GOTO 460
650:"8"
660:X=XX/8:Y=(YY-46)/8
670:IF X=3BEEP 3:GOTO 470
680:GCURSOR YY,XX
690:Z=P(Y,X+1):GPRINT A$(Z)
700:GCURSOR YY,XX+8:GPRINT A$(16)
710:Y=(YY-46)/8:X=XX/8
720:P(Y,X+1)=16
730:P(Y,X)=Z
740:YY=YY:XX=XX+8
750:GOTO 460
760:"6"
770:X=XX/8:Y=(YY-46)/8
780:IF Y=1BEEP 3:GOTO 470
790:GCURSOR YY,XX
900:Z=P(Y-1,X):GPRINT A$(Z)
810:GCURSOR YY-8,XX:GPRINT A$(16)
820:X=XX/8:Y=(YY-46)/8
830:P(Y,X)=16
840:P(Y,X)=Z
850:YY=YY-8:XX=XX
860:GOTO 460
870:"4"
880:X=XX/8:Y=(YY-46)/8
890:IF Y=4BEEP 3:GOTO 470
900:GCURSOR YY,XX
910:Z=P(Y+1,X):GPRINT A$(Z)
920:GCURSOR YY+8,XX:GPRINT A$(16)
930:Y=(YY-46)/8:X=XX/8
940:P(Y+1,X)=16
950:P(Y,X)=Z
960:YY=YY+8:XX=XX
970:GOTO 460
    
```



VII Programme

PC-1600

Dirk Mahrholz
Goethestr. 36
2850 Bremerhaven

Anbei lege ich noch zwei Listings für den PC-1600.
Zum Programm: "39 Zeichen": Ursprünglich hieß es
"37 Zeichen" und wurde von C. Samoticha (Heft 12,
S. 17) geschrieben. Ich habe es dann umgeschrieben
und dem Zeichensatz des PC-1600 angepaßt.
"Symbol-Editor":

Es stammt von Andre Horch (Heft 19, S. 42). Diese
Version benutzt alle Ziffern.

1,2,3,4,6,7,8,9: zur Cursorbewegung

5: zum Punkt setzen

=: zeigt den Hex-Code

o: zum Display löschen

*: zum beenden des Programmes

Ich würde mich freuen, wenn Sie die Listings ab-
drucken würden. Falls Sie an weiteren Programmen
interessiert sind, so schreiben Sie mir bitte.

39 - Zeichen

```

5:"K"CLS :CLEAR :WAIT 0:RESTORE
10:DIM Z$(0)*39
15:Z$(0)="SHARP PC-1600 / 39 Zeichen pro Zeile !!":GOSUB 40
20:GCURSOR 15,8:FOR T=32TO 64:Z$(0)=CHR$(T):GOSUB 40:NEXT T
25:GCURSOR 15,16:FOR T=65TO 96:Z$(0)=CHR$(T):GOSUB 40:NEXT T
30:GCURSOR 15,24:FOR T=97TO 126:Z$(0)=CHR$(T):GOSUB 40:NEXT T
35:A$=INKEY$ :IF A$=""GOTO 35ELSE CLS :END
40:FOR TT=1TO LEN Z$(0):ZZ=ASC (MID$(Z$(0),TT,1)):ZZ=ZZ-32
45:RESTORE 100+INT (ZZ/5):IF INT (ZZ/5)=ZZ/5GOTO 55
50:FOR DD=1TO ((ZZ/5-INT (ZZ/5))*5)*3:READ AA:NEXT DD
55:FOR DD=0TO 2:READ AA:GPRINT AA;:NEXT DD
60:GPRINT 0;:IF TT=42GPRINT
65:NEXT TT:RETURN
100:DATA 0,0,0,0,111,0,7,0,7,20,63,10,44,127,26
101:DATA 50,8,38,20,42,52,4,2,1,28,34,65,65,34,28
102:DATA 42,28,42,8,62,8,128,96,0,8,8,0,96,96
103:DATA 96,28,3,62,65,62,4,2,127,114,73,70,34,73,54
104:DATA 15,8,126,47,73,49,62,73,50,1,121,7,54,73,54
105:DATA 38,73,62,0,20,0,128,104,0,8,20,34,20,20,20
106:DATA 34,20,8,2,89,6,62,77,46,126,9,126,127,73,54
107:DATA 62,65,34,127,65,62,127,73,65,127,9,1,62,73,58
108:DATA 127,8,127,65,127,65,32,64,63,127,28,99,127,64,64
109:DATA 127,14,127,127,28,127,127,65,127,127,9,6,62,97,126
110:DATA 127,9,118,70,73,49,1,127,1,127,64,127,63,96,63
111:DATA 127,48,127,119,8,119,15,120,15,113,73,71,127,65,65
112:DATA 3,28,96,65,65,127,2,1,2,128,128,128,4,2,1
113:DATA 32,84,120,127,68,56,56,68,40,56,68,127,56,84,88
114:DATA 4,127,5,152,164,124,127,4,120,0,122,64,64,128,122
115:DATA 127,16,108,0,127,64,124,24,124,124,4,120,56,68,56
116:DATA 252,36,24,24,36,252,120,4,4,72,84,36,4,127,68
117:DATA 60,64,124,60,96,60,124,48,124,108,16,108,156,96,60
118:DATA 100,84,76,8,54,65,0,54,0,65,54,8,24,8,12

```

```

SHARP PC-1600 / 39 Zeichen pro Zeile !!
!"#$%&'()*+,-./0123456789:;<=>?@
ABCDEFGHIJKLMNPOQRSTUVWXYZ[\]^_
abcdefghijklmnopqrstuvwxyz{|}~

```



Symbol - Editor

A

007C7E36367E7C00

Symbol - Editor

```

5:"S"CLS :CLEAR :WAIT 0:BEEP ON
10:DIM P(7,7)
15:GDCURSOR 56,0:GPRINT "00266F49497B3200009CBCA0A0FC7C00787C0C380C7C7800";
20:GPRINT "433F7F44447C380000387C44447C3800000417F7F400000"
25:CURSOR 18,0:PRINT "- Editor"
30:CURSOR 7,3:PRINT ">":LINE (42,23)-(47,31),X,,BF:CURSOR 10,3:PRINT "0000000
00000000"
35:FOR Q=0TO 24STEP 3:LINE (0,Q)-(24,Q):LINE (Q,0)-(Q,24):NEXT Q
40:YY=1:XX=1
45:A=ASC (INKEY$ )
50:IF A=48GOTO 5ELSE IF A=42CLS :END
55:IF A=49GOSUB 150:Y=Y+1:X=X-1:GOTO 125
60:IF A=50GOSUB 150:Y=Y+1:GOTO 125
65:IF A=51GOSUB 150:Y=Y+1:X=X+1:GOTO 125
70:IF A=52GOSUB 150:X=X-1:GOTO 125
75:IF A=54GOSUB 150:X=X+1:GOTO 125
80:IF A=55GOSUB 150:Y=Y-1:X=X-1:GOTO 125
85:IF A=56GOSUB 150:Y=Y-1:GOTO 125
90:IF A=57GOSUB 150:Y=Y-1:X=X+1:GOTO 125
95:IF A=61GOTO 165
100:IF A<>53GOTO 115
105:BEEP 1:IF P(X,Y)=1LET P(X,Y)=0:GOTO 115
110:P(X,Y)=1
115:PRESET (XX,YY):PRESET (XX+1,YY):PRESET (XX,YY+1):PRESET (XX+1,YY+1)
120:PSET (XX,YY):PSET (XX+1,YY):PSET (XX,YY+1):PSET (XX+1,YY+1):GOTO 45
125:IF X<0LET X=7ELSE IF X>7LET X=0
130:IF Y<0LET Y=7ELSE IF Y>7LET Y=0
135:YY=Y*3+1:XX=X*3+1
140:IF Y=0LET YY=1ELSE IF X=0LET XX=1
145:PSET (XX,YY):PSET (XX+1,YY):PSET (XX,YY+1):PSET (XX+1,YY+1):GOTO 45
150:IF P(X,Y)=1GOTO 160
155:PRESET (XX,YY):PRESET (XX+1,YY):PRESET (XX,YY+1):PRESET (XX+1,YY+1)
160:RETURN
165:LL$="":FOR Q=0TO 7
170:S=.5:SS=0
175:FOR W=4TO 7:S=S*2
180:IF P(Q,W)=1LET SS=SS+S
185:NEXT W
190:B$=STR$ (SS):IF SS>9LET B$=CHR$ (55+SS)
195:S=.5:SS=0:LL$=LL$+B$
200:FOR W=0TO 3:S=S*2
205:IF P(Q,W)=1LET SS=SS+S
210:NEXT W
215:B$=STR$ (SS):IF SS>9LET B$=CHR$ (55+SS)
220:LL$=LL$+B$:NEXT Q
225:CURSOR 10,3:PRINT LL$:GDCURSOR 41,10:GPRINT LL$:GOTO 45
    
```

-- PC-1600, CE-1600P --

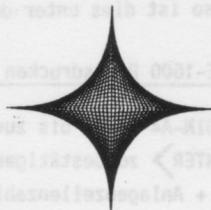
Thomas Jeger
Hauptstrasse 142

CH-3286 Muntelier

```

10:TEXT :CSIZE 2
20:LF 5
30:GRAPH
40:GLCURSOR (450,0):SORGN
50:FOR I=0TO 100STEP 4
60:LLINE (0,I)-(100-I,0)
70:NEXT I
80:FOR I=-100TO -4STEP 4
90:LLINE (0,I+100)-(I,0)
100:NEXT I
110:FOR I=-96TO 0STEP 4
120:LLINE (0,-100-I)-(I,0)
130:NEXT I
140:FOR I=4TO 96STEP 4
150:LLINE (0,-100+I)-(I,0)
160:NEXT I
    
```

Nebstehendes Programm plottet
eine solche Grafik auf Ihrem
CE-1600P.



- DER FACHVERLAG FÜR
TASCHEN-COMPUTER !

Edgar Härle 7951 Warthausen
 Neherstraße 22
 Briefe schreiben mit dem PC-1600

Dieses Programm ist geeignet, mit dem Computer PC-1600 + CE-1600 P + CE-1600 F Standardbriefe zu schreiben und auf der Diskette unter einer Akten-Nr. abzuspeichern. Nach dem Starten des Programms mit RUN oder DEF S erscheint das Hauptmenü. Dabei sind folgende Funktionen möglich:

- N = Brief neu schreiben
- T = Brief editieren
- D = Brief mit dem Drucker CE-1600 P ausdrucken
- S = Brief auf der Diskette abspeichern
- L = Brief von der Diskette in den Computer laden
- E = Programmende
- Programmteil (N) - Brief neu schreiben -

Der Computer fragt den Benutzer nach untenstehenden Angaben, wobei die jeweilige max. Zeilenzahl und Zeilenlänge zu beachten ist.

Eingabetext	max. Zeilenzahl	max. Zeilenlänge
1. Akten-Nr. ?	1	8
2. Datum ?	1	8
3. Adresse ?	6	30
4. Text ?	50	80
5. Anlage ?	4	80

Die jeweilige Eingabe bei der Adresse, beim Text und beim Anlagentext wird durch (E) + <ENTER> beendet. Der Speicherplatz des PC-1600 reicht für das Programm + Variable ohne Erweiterungs-RAM aus. Sollte jedoch für den Text die max. Zeilenzahl von 50 Zeilen nicht ausreichen, so ist im Programm die DIM-Anweisung + FOR/NEXT-Schleife entsprechend abzuändern. Dies ist jedoch nur mit Erweiterungs-RAM möglich.

Werden bei der Text-Eingabe mehr als 80 Zeichen pro Zeile eingegeben, so verzweigt das Programm in eine Fehlerbehandlungsroutine. Dabei wird der Fehler auf dem Display angezeigt und die Zeile kann anschließend neu eingegeben werden. Ich empfehle nicht mehr als 3 Zeilen des Anzeigedisplays vom PC-1600 = 78 Zeichen in eine Zeile zu schreiben.

Programmteil (T) - Brief editieren -

Dieser Programmteil ist gänzlich menügesteuert und erklärt sich damit von selbst.

Folgende Daten können geändert werden:

Akten-Nr., Datum, Adresse, Betreff, Zeilentext und Anlagentext.

Soll ein Zeilentext geändert werden, so erscheint ein weiteres Menü mit folgendem Inhalt:

- Zeilentext ändern (T)
- Zeile einfügen (+)
- Zeile löschen (-)

Der Zeilentext ist unter der entsprechenden Zeilennummer abzuändern. Beim Abzählen der Zeilennummern sind eventuelle Leerzeilen mitzuzählen. Soll eine Zeile vor die erste Zeile eingefügt werden, so ist dies unter der Zeilennummer - 0 - möglich.

Programmteil (D) - Brief mit dem Drucker CE-1600 P ausdrucken -

Soll der Brief gedruckt werden, so ist ein DIN-A4-Blatt bis zur Abrisskante des Druckers einzuspannen und mit (J) + <ENTER> zu bestätigen.

Ist die Adressenzeilenzahl + Textzeilenzahl + Anlagenzeilenzahl gleich oder größer 31, so wird der Druckvorgang gestoppt. Auf dem Display erscheint dann die Aufforderung, ein zweites DIN-A4-Blatt bis zur Abrisskante einzuspannen und mit (J) + <ENTER> zu bestätigen.

Vor der Benutzung des Programms sind folgende persönliche Daten zu

- DER FACHVERLAG FÜR

TASCHEN-COMPUTER !

programmieren:

Zeile 170 : Wohnort
 Zeile 1140 : Vorname + Nachname
 Zeile 1170 : PLZ + Wohnort
 Zeile 1180 : Straße + Nr.
 Zeile 1190 : Telefon-Nr.
 Zeile 1230 : Wohnort

Sollte der Text "Sehr geehrte Damen und Herren" anders formuliert werden, so ist dies in Zeile 1250 möglich. Sie können diese Zeile selbstverständlich auch ganz streichen und die Anrede dann jeweils in den Zeilentext einbringen.

Programmteil (S) - Brief auf der Diskette abspeichern -

Der Brief kann auf der Diskette unter einer Akten-Nr. (beliebige Zeichenfolge von Großbuchstaben und Zahlen, jedoch max. 8 Zeichen) abgespeichert werden. Da ein Komma als Datenseparator dient, wird dies bei den Textzeilen durch das &-Zeichen ersetzt. Deshalb darf in den Textzeilen kein &-Zeichen vorkommen, denn es würde beim laden von der Diskette in ein Komma umgewandelt werden.

In den Adressenzeilen und im Anlagentext darf kein Komma sein, weil das Komma hier nicht durch ein anderes Zeichen ersetzt wird. Das &-Zeichen ist hier jedoch zulässig.

Ist keine Diskette im Laufwerk, so verzweigt das Programm in eine Fehlerbehandlungsroutine. Nach dem Einlegen der Diskette und Bestätigung mit (J) + <ENTER> wird der Programmablauf fortgesetzt.

Programmteil (L)- Brief von der Diskette in den Computer laden -

Unter der entsprechenden Akten-Nr. kann ein Brief von der Diskette wieder in den Computer geladen werden.

Die &-Zeichen in den Textzeilen werden durch das Komma ersetzt. Ist keine Diskette im Laufwerk oder wurde die Datei unter der Akten-Nr. nicht gefunden, so verzweigt das Programm in die Fehlerbehandlungsroutine. Nach der Behebung des Fehlers kann ohne Neustart weitergearbeitet werden.

```

10:"S":REM * Standardbrief schreiben mit PC-1600+CE-1600P+CE-1600F ***
15:REM *** Copyright by Edgar Härlø, 7951 Warthausen, Neherstr. 22 ***
20:CLS :ON ERROR GOTO 1800
30:PRINT "** BRIEF NEU SCHREIBEN (N)"
40:PRINT "EDITIEREN (T), DRUCKEN (D)"
50:PRINT "DISKETTE SAVE (S) LOAD (L)"
60:INPUT "ENDE (E) ";F$
70:IF F$="N"OR F$="n"THEN 140
80:IF F$="T"OR F$="t"THEN 390
90:IF F$="D"OR F$="d"THEN 1050
100:IF F$="E"OR F$="e"THEN 2080
110:IF F$="S"OR F$="s"THEN 1400
120:IF F$="L"OR F$="l"THEN 1600
130:CLS :PAUSE "Eingabe nicht möglich !!!":GOTO 30
140:CLS :REM Brief neu schreiben
150:CLEAR :DIM A$(7)*30:DIM Z$(51)*80:DIM B$(0)*80:DIM L$(5)*80:DIM T$(1)*80
160:INPUT "Akten Nr. ";X1$:PRINT
170:INPUT "Berlin, den ";D$:CLS
180:PRINT "Eing. wird mit -E- beendet"
190:PRINT "Schreiben geht an Adresse"
200:FOR I=1TO 6
210:INPUT A$(I):A=A
220:IF A$(I)="E"OR A$(I)="e"THEN LET A=I-1:I=7
230:NEXT I:CLS
240:PRINT "Betrifft:"
250:INPUT B$(0):CLS
260:FOR I=1TO 50
270:PRINT I;". Zeile eingeben"
280:INPUT Z$(I):Z=I
290:IF Z$(I)="E"THEN LET Z=I-1:I=51
300:IF Z$(I)="e"THEN LET Z=I-1:I=51
310:NEXT I:CLS
320:FOR I=1TO 4
330:PRINT "Anlage: (;I;". Zeile)"
340:INPUT L$(I):L=I
    
```

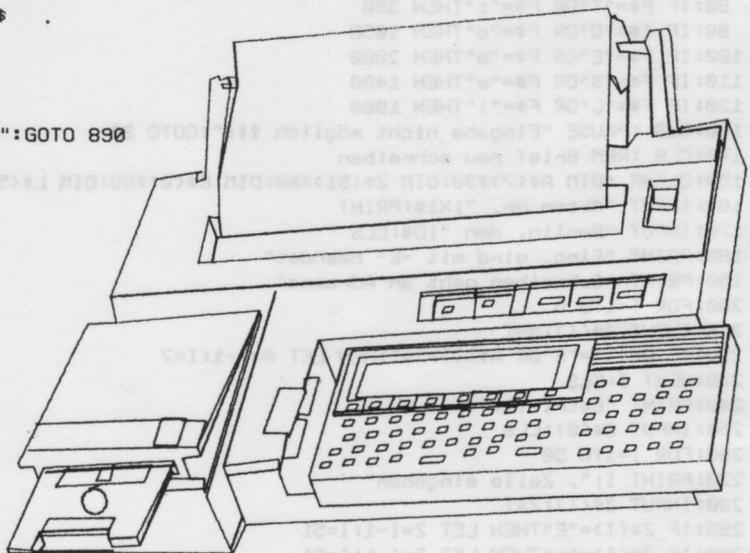
- DER FACHVERLAG FÜR

TASCHEN-COMPUTER !

```

350: IF L$(I)="E" THEN LET L=I-1:I=5
360: IF L$(I)="e" THEN LET L=I-1:I=5
370: NEXT I
380: GOTO "S"
390: CLS :REM Brief editieren
400: PRINT "alte Akten Nr. ";X1$
410: INPUT "neue Akten Nr. ";X1$
420: PRINT "altes Datum: ";D$
430: INPUT "neues Datum: ";D$
440: CLS :PRINT "Adresse(A), Betreff änd.(B)".
450: PRINT "Zeilentext ändern (T)"
460: PRINT "Anlagentext ändern (L)"
470: INPUT "Hauptmenü (M) ";F$
480: IF F$="A" OR F$="a" THEN 540
490: IF F$="B" OR F$="b" THEN 610
500: IF F$="T" OR F$="t" THEN 650
510: IF F$="L" OR F$="l" THEN 960
520: IF F$="M" OR F$="m" THEN "S"
530: CLS :PAUSE "Eingabe nicht möglich !!!":GOTO 440
540: CLS :PRINT "neue Adresse schreiben"
550: PRINT "Eing. wird mit -E- beendet"
560: FOR I=1 TO 6
570: INPUT A$(I):A=I
580: IF A$(I)="E" OR A$(I)="e" THEN LET A=I-1:I=7
590: NEXT I
600: GOTO 440
610: CLS :PRINT "Betreff ändern"
620: PRINT B$(0)
630: INPUT B$(0)
640: GOTO 440
650: CLS :PRINT "Zeilentext ändern (T)"
660: PRINT "Zeile einfügen (+)"
670: PRINT "Zeile löschen (-)"
680: INPUT "Hauptmenü (M) ";F$
690: IF F$="T" OR F$="t" THEN 740
700: IF F$="M" OR F$="m" THEN "S"
710: IF F$="+" THEN 780
720: IF F$="-" THEN 880
730: CLS :PAUSE "Eingabe nicht möglich !!!":GOTO 650
740: CLS :INPUT "Änderung der Zeile Nr. ";N
750: PRINT Z$(N)
760: INPUT Z$(N)
770: GOTO 650
780: CLS :PRINT "Textzeile einfügen nach"
790: INPUT "der Zeile Nr. ";N
800: T$(0)=Z$(N+1):Z=Z+1
805: PRINT
810: INPUT Z$(N+1)
820: FOR I=N+2 TO Z
830: T$(1)=Z$(I):Z$(I)=T$(0):T$(0)=T$(1)
840: NEXT I
850: CLS :INPUT "weitere Zeilen einf.(J/N)";F$
860: IF F$="J" OR F$="j" THEN 870 ELSE 650
870: N=N+1:PRINT "Zeile Nr. ";N:GOTO 800
880: CLS :PRINT "Textzeile löschen"
890: INPUT "Zeile Nr. ";N
900: IF N>Z THEN PRINT "Zeile nicht vorhanden":GOTO 890
910: Z=Z-1:IF N<(Z-1) THEN Z$(N)="" :GOTO 650
920: FOR I=NTO Z
930: Z$(I)=Z$(I+1)
940: NEXT I
950: GOTO 650
960: CLS :PRINT "Anlagentext neu schreiben"
970: PRINT "Eing. wird mit -E- beendet"
980: FOR I=1 TO 4
990: PRINT "Anlage: (;I;".Zeile)"
1000: INPUT L$(I):L=I
1010: IF L$(I)="E" THEN LET L=I-1:I=5
1020: IF L$(I)="e" THEN LET L=I-1:I=5
1030: NEXT I
1040: GOTO 440
1050: REM Brief drucken
1060: CLS :PRINT "DIN-A4 Blatt bis zur"
1070: PRINT "Abrisskante in den Drucker"
1080: PRINT "einspannen ....."
1090: INPUT "und mit (J) bestätigen ";F$
1100: IF F$="J" OR F$="j" THEN 1110 ELSE 1060
1110: TEXT :LF 1:GRAPH :X=0:Y=0:CSIZE 4
1120: FOR I=1 TO 4
1130: GLCURSOR (X,Y)
1140: LPRINT "FISCHEL GmbH"
1150: X=X+1:Y=Y-1
1160: NEXT I:CSIZE 2
1170: GLCURSOR (660,12):LPRINT "1000 Berlin"
1180: GLCURSOR (660,-12):LPRINT "Kaiser Friedrich Str.54a"
1190: GLCURSOR (660,-36):LPRINT "Tel.030/3236029"
1200: GLCURSOR (660,-84):LPRINT "Akten Nr. ";X1$
1210: TEXT :LF 2
1220: FOR I=1 TO A:LPRINT A$(I):NEXT I:LF 1
1230: LPRINT TAB (46);"Berlin, den ";D$:LF 2
1240: LPRINT B$(0):LF 3
1250: LPRINT "Sehr geehrte Damen und Herren,"
1260: LF 2:F$="N"
1270: FOR I=1 TO Z
1280: IF F$="J" OR F$="j" OR (A+I+L)<=31 THEN 1340
1290: LF 20
1300: CLS :PRINT "zweites Blatt bis zur"
1310: PRINT "Abrisskante einspannen"
1320: INPUT "und mit (J) bestätigen ";F$
1330: IF F$="J" OR F$="j" THEN 1335 ELSE 1300
1335: CSIZE 1:LPRINT TAB (120);"Blatt 2":CSIZE 2:LF 3
1340: LPRINT Z$(I)
1350: NEXT I:LF 3
1360: LPRINT "Mit freundlichen Grü";CHR$(225);"en"
1370: LF 4:LPRINT "Anlage:":LF -1:LPRINT " "
1380: FOR I=1 TO L:LPRINT L$(I):NEXT I:LF 20
1390: GOTO "S"
1400: REM Brief auf der Diskette abspeichern
1410: PRINT "Brief auf der Diskette":PRINT "abspeichern !"
1420: PRINT "Akten Nr. ";X1$
1430: INPUT "Akten Nr. ";X1$
1440: MAXFILES =1
1450: OPEN "X:"+X1$ FOR OUTPUT AS #1
1460: PRINT #1,A;";";Z;";";L;";";D$;";";B$(0)
1470: FOR I=1 TO A:PRINT #1,A$(I):NEXT I
1480: FOR I=1 TO Z
1490: P1=INSTR (Z$(I),","):P2=LEN (Z$(I)):P3=P2-P1
1510: IF P1=0 THEN 1550
1520: T$(0)=LEFT$ (Z$(I),P1-1)
1530: T$(1)=RIGHT$ (Z$(I),P3)
1540: Z$(I)=T$(0)+"&"+T$(1):GOTO 1500
1550: PRINT #1,Z$(I)
1560: NEXT I
1570: FOR I=1 TO L:PRINT #1,L$(I):NEXT I
1580: CLOSE #1
1590: GOTO 1640
1600: REM Brief von der Diskette in den Computer laden

```



```

1610: CLEAR : DIM A$(?) * 30 : DIM Z$(51) * 80 : DIM B$(0) * 80 : DIM L$(5) * 80 : DIM T$(1) * 80
1620: PRINT "Brief von der Diskette": PRINT "in den Computer einlesen !": PRINT
1630: INPUT "Akten Nr. "; X1$
1640: MAXFILES = 1
1650: OPEN "X:" * X1$ FOR INPUT AS #1
1660: INPUT #1, A, Z, L, D$, B$(0)
1670: FOR I=1 TO A: INPUT #1, A$(I): NEXT I
1680: FOR I=1 TO Z
1690: INPUT #1, Z$(I)
1710: P1=INSTR (Z$(I), "&"): P2=LEN (Z$(I)): P3=P2-P1
1720: IF P1=0 THEN 1760
1730: T$(0)=LEFT$ (Z$(I), P1-1)
1740: T$(1)=RIGHT$ (Z$(I), P3)
1750: Z$(I)=T$(0)+", "+T$(1): GOTO 1710
1760: NEXT I
1770: FOR I=1 TO L: INPUT #1, L$(I): NEXT I
1780: CLOSE #1
1790: GOTO "S"

1800: REM Fehlerbehandlungsroutine
1810: IF ERN = 160 THEN GOTO 1870
1820: IF ERN = 15 THEN 1910
1830: IF ERN = 152 THEN GOTO 1950
1840: IF ERN >= 70 AND ERN <= 80 THEN GOTO 1980
1850: IF ERN >= 150 AND ERN <= 168 THEN GOTO 2020
1860: GOTO 2060
1870: CLS : PRINT "keine Diskette im Laufwerk"
1880: PRINT "bitte Diskette einsetzen"
1890: INPUT "und Taste (J) drücken "; F$
1900: IF F$="J" OR F$="j" THEN RESUME ELSE 1870
1910: CLS : PAUSE "Zeile länger als "
1920: PAUSE "80 Zeichen"
1930: PAUSE "bitte neu eingeben"
1940: RESUME ERL -10
1950: CLS : PAUSE "Datei nicht gefunden"
1960: PAUSE "bitte überprüfen"
1970: RESUME 1600
1980: CLS : PRINT "Drucker-Fehler"
1990: PRINT "in Zeile "; ERL
2000: PRINT "Fehler-Code "; ERN
2010: PRINT "bitte Fehler beheben": STOP
2020: CLS : PRINT "Datei-Fehler"
2030: PRINT "in Zeile "; ERL
2040: PRINT "Fehler-Code "; ERN
2050: PRINT "bitte Fehler beheben": STOP
2060: CLS : PRINT "Fehler in Zeile"; ERL
2070: PRINT "Fehler-Code "; ERN: STOP
2080: CLS : PRINT "*** E N D ***": END

```

FISCHEL GmbH

Edgar Härle
Neherstrasse 22

Muster

7951 Warthausen

Sehr geehrte Damen und Herren,

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD

EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG

Mit freundlichen Grüßen

Anlage:
111111
222222

1000 Berlin
Kaiser Friedrich Str.54a
Tel.030/3236029

Akten Nr. 1000.

Berlin, den 03.04.87

Adressaufkleber schreiben mit PC-1600

Nach dem Schreiben eines Briefes ist meist noch ein Adressaufkleber zu schreiben. Dies ist mit dem Ergänzungsprogramm möglich. Es ist geeignet, mit dem Computer PC-1600 + CE-1600 P + CE-1600 F Adressaufkleber in 6 verschiedenen Größen zu drucken. Damit die Adresse nicht doppelt geschrieben werden muß, kann mit diesem Programm die Adresse eines Briefes unter der entsprechenden Akten-Nr. von der Diskette in den Computer geladen werden. Selbstverständlich können auch beliebige Adressen per Tastatur neu geschrieben werden.

Nach dem Ausdruck ist die Adresse auszuschneiden und mit einem Klebestift auf den Brief oder das Paket zu kleben.

Das Programm wird gestartet mit RUN oder DEF S. Es erscheint dann ein Menü mit folgenden Funktionen:

- (1) Adresse neu schreiben
- (2) Adresse von der Diskette einlesen
- (3) Adresse ausdrucken
- (4) Programmende

Firma
Fischel GmbH
Kaiser Friedrich Str.54a

1000 Berlin 12

```

10:"S":REM *** Adressaufkleber schreiben *****
20:REM ***** mit PC-1600+CE-1600P+CE-1600F ***
30:REM ***** von Edgar Härle *****
40:CLS :PRINT "Adressaufkleber ..."
50:PRINT "neu schreiben (1)"
60:PRINT "von Diskette einlesen(2)"
70:INPUT "drucken (3) Ende (4)";R
80:IF R<1OR R>4THEN "S"
90:ON RGOTO "N","B","D","E"
100:"N":CLEAR :DIM A$(7)*30
110:CLS :PRINT "Adressaufkleber schreiben"
120:PRINT "Eing. wird mit -E- beendet"
130:FOR I=1TO 6
140:INPUT A$(I):A=I
150:IF A$(I)="E"OR A$(I)="e"THEN LET A=I-1:I=7
160:NEXT I:GOTO "S"
170:"B":CLEAR :DIM A$(7)*30:DIM B$(0)*80
180:CLS :PRINT "Briefadresse von der"
190:PRINT "Diskette in den Computer"
200:PRINT "einlesen unter der"
210:INPUT "Akten Nr. ";X1$
220:MAXFILES =1
230:OPEN "X:"+X1$FOR INPUT AS #1
240:INPUT #1,A,Z,L,D$,B$(0)
250:FOR I=1TO A:INPUT #1,A$(I):NEXT I
260:CLOSE #1:GOTO "S"
270:"D":CLS :PRINT "Adressaufkleber ausdrucken":PRINT
280:INPUT "Schriftgröße (1 bis 6) ";G
290:INPUT "Farbe (0 bis 3) ";F
300:IF G<1OR G>6OR F<0OR F>3THEN "D"
310:M=0:N=0
320:FOR I=1TO A
330:M=LEN A$(I)
340:IF M>NTHEN LET N=M
350:NEXT I
360:LF 1:GRAPH :CSIZE G:X=G*10:Y=0:COLOR F
370:GR=G-1:IF G=1THEN LET GR=1
380:FOR I=1TO A
390:FOR P=1TO GR
400:GLCURSOR (X,Y):LPRINT A$(I)
410:X=X+1:Y=Y-1
420:NEXT P
430:X=G*10:Y=Y-(12*G)
440:NEXT I
450:LLINE (0,13*G+5)-(N*6*G+19*G,Y+G),0,F,B
460:GLCURSOR (0,Y-100)
470:TEXT :COLOR 0:CSIZE 2:GOTO "S"
480:"E":PRINT "..... Programmende ....."
490:PRINT "Starten ohne Datenverlust"
500:PRINT "mit DEF S":END
    
```

Firma
Fischel GmbH
Kaiser Friedrich Str.54a

1000 Berlin 12



Hch. Haller
Im Pächterried 13
CH-8105 Watt-Regensdorf

PC-1600

Masstab - Umrechnungen

Das folgende kleine Programm erlaubt auf komfortable Weise, massstäbliche Vergrößerungen oder Verkleinerungen umzurechnen. Angehörige technischer Berufe, welche täglich technische Zeichnungen zu erstellen oder zu lesen haben, aber auch Leute, die hin und wieder eine Landkarte in die Hand nehmen, können das Programm sicher gut gebrauchen. Es sind immer die letzten beiden Masseingaben auf dem Display noch sichtbar, auch bei einem Wechsel von "Vergrößerung" auf "Verkleinerung" und umgekehrt. Es sind Umrechnungsfaktoren 1...1Mio erlaubt. Eingaben können auch in Exponential-Schreibweise gemacht werden. Die umgerechneten Zahlen werden in normaler Form angezeigt wenn der Wert zwischen 1E-6 und 1E6 liegt, andernfalls in Exponential-Schreibweise.

Rechner: PC 1600 (ohne Peripherie)

Speicherbedarf: 1150 Bytes.

Bedienung: Das Programm ist selbsterklärend.

Es kann mit "RUN" oder "DEF M" gestartet werden. Dann kann der Umrechnungsfaktor eingegeben werden. Ist er i.O., können fortlaufend Zahlen zur Umrechnung eingetippt werden. Zum Umschalten von "Vergrößerung" auf "Verkleinerung" oder umgekehrt drückt man einfach auf die <ENTER>-Taste, ohne Eingabe einer Zahl. Durch zweimaliges Drücken der <ENTER>-Taste kann ein neuer Umrechnungsfaktor vorgegeben werden. Ausstieg aus dem Programm durch Drücken der <BREAK>-Taste.

Programm-Beschreibung:

17-21 Faktor-Eingabe und Verifikation.

23-30 Aufforderung zur Korrektur.

32-47 Berechnung und Anzeige.

49-52 Masseingabe und Umschaltung von "Vergrößerung" auf "Verkleinerung" und umgekehrt, bzw. Umschaltung auf neue Faktoreingabe.

54-57 Unterprogramm zum Umschalten auf Exponential-Schreibweise.

```

10 '-----
11 'Hch.Haller
12 'Im Paechterried 13
13 'CH-8105 Watt(Schweiz)
14 '-----
15 'MASSSTAB-UMRECHNUNGEN
    (** MST.BAS **)
16 '-----
17 "M":CLEAR :CLS :WAIT 0:
    M$="G":Z=0
18 PRINT " MASSSTAB-UMREC
    HNUNGEN"
19 PRINT "-----
    -----"
20 CURSOR 6,3:INPUT "Fakto
    r?: ";FAKT
21 IF FAKT<=1E6AND FAKT>=1
    AND LEN STR$(FAKT)<=7T
    HEN "M1"
48 '-----
49 "INPUT"CURSOR 0,3:INPUT
    "Mass: ";X3
50 IF X3=X2THEN IF Z=1GOTO
    "M"ELSE IF M$="G"LET M
    $="K":Z=1:GOTO "M1"ELSE
    LET M$="G":Z=1:GOTO "M
    1"
51 X1=X2:X2=X3:Z=0:GOTO "M
    1"
52 END
53 '-----
54 "E" 'EXP-SCHREIBWEISE
55 IF E>=1E6OR E<1E-6THEN
    USING "#.####^"
56 IF E=0THEN USING
57 RETURN

```

```

22 '-----
23 CLS :BEEP 1
24 CURSOR 1,2:PRINT "Max.7
    Zeichen, 1...1Mio!"
25 CURSOR 9,3:PRINT "<ENTE
    R>"
26 L=12-(LEN STR$(FAKT)/2
    )
27 "BLINK"CURSOR L,0:WAIT
    30:PRINT FAKT
28 CURSOR 0,0:WAIT 2:PRINT
    "
    "(26 Blanks)
29 E$=INKEY$(1):IF E$=CHR
    $(&OD)GOTO "M"
30 GOTO "BLINK"
31 '-----
32 "M1"CLS

```

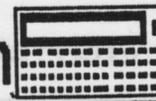
```

33 IF M$="G"THEN PRINT "Ve
    rgrsserung";FAKT;"-fac
    h":MST=FAKT
34 IF M$="K"THEN PRINT "Ve
    rkleinerung";FAKT;"-fac
    h":MST=1/FAKT
35 Y1=X1*MST:Y2=X2*MST
36 CURSOR 0,1
37 E=X1:GOSUB "E"
38 PRINT X1:USING
39 CURSOR 12,1
40 E=Y1:GOSUB "E"
41 PRINT ">>";Y1:USING
42 CURSOR 0,2
43 E=X2:GOSUB "E"
44 PRINT X2:USING
45 CURSOR 12,2
46 E=Y2:GOSUB "E"
47 PRINT ">>";Y2:USING

```



durch Information vorn



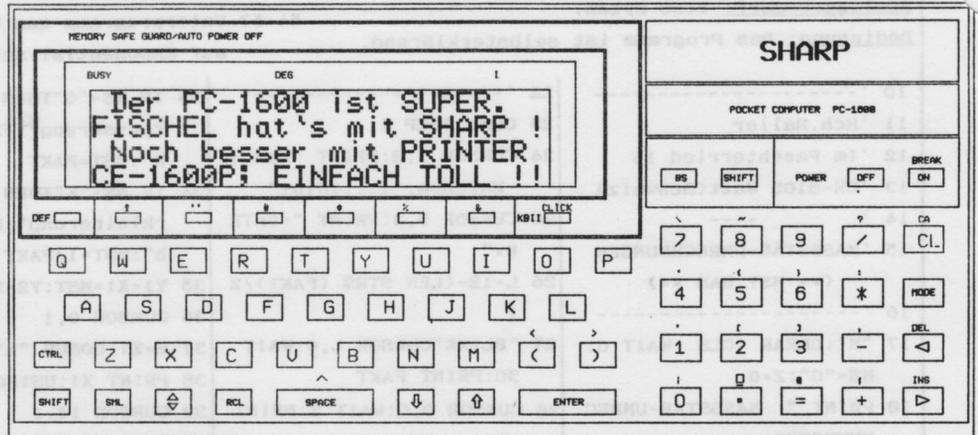
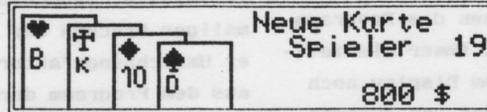
17+4

Die Spielregeln setze ich als bekannt voraus. Gestartet wird, wie die anderen Programme auch, mit RUN. Danach wird eine Spielkarte gezeigt. Durch Beantworten der Frage "Neue Karte" mit J oder N bekommt man die nächste Karte, oder die Bank, spricht der PC-1600, spielt weiter. Zum Schluss wird der Gewinner angezeigt. Hinter "Spieler" bzw. "Bank" wird die Summe der gezogenen Werte angezeigt. Jede Spielrunde wird mit 1000\$ begonnen. Pro Runde gewinnt oder verliert man 100\$. Hat man kein Geld mehr ist das Spiel aus.

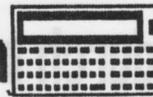
```

1:REM *****
2:REM ** 17+4 **
3:REM ** by **
4:REM **Hans Hölzel**
5:REM *****
10:CLS :CLEAR :WAIT 0:RANDOM
20:D=1000
30:DIM K$(32)*13,W$(32)*2,X$(75),KK$(32)
40:X$(55)="01390503"
50:X$(56)="36494936"
60:X$(57)="2649493E"
70:X$(49)="02017F003E41413E"
80:X$(66)="7F494936"
90:X$(68)="3F21211E"
100:X$(75)="1F040A11"
110:X$(65)="3E09093E"
120:K$(1)="Karo 7 7"
130:K$(2)="Karo 8 8"
140:K$(3)="Karo 9 9"
150:K$(4)="Karo 10 10"
160:K$(5)="Karo Bube 2"
170:K$(6)="Karo Dame 3"
180:K$(7)="Karo König 4"
190:K$(8)="Karo Ass 11"
200:K$(9)="Herz 7 7"
210:K$(10)="Herz 8 8"
220:K$(11)="Herz 9 9"
230:K$(12)="Herz 10 10"
240:K$(13)="Herz Bube 2"
250:K$(14)="Herz Dame 3"
260:K$(15)="Herz König 4"
270:K$(16)="Herz Ass 11"
280:K$(17)="Pik 7 7"
290:K$(18)="Pik 8 8"
300:K$(19)="Pik 9 9"
310:K$(20)="Pik 10 10"
320:K$(21)="Pik Bube 2"
330:K$(22)="Pik Dame 3"
340:K$(23)="Pik König 4"
350:K$(24)="Pik Ass 11"
360:K$(25)="Kreuz 7 7"
370:K$(26)="Kreuz 8 8"
380:K$(27)="Kreuz 9 9"
390:K$(28)="Kreuz 10 10"
400:K$(29)="Kreuz Bube 2"
410:K$(30)="Kreuz Dame 3"
420:K$(31)="Kreuz König 4"
430:K$(32)="Kreuz Ass 11"
440:X=3:Y=3:A=1:AA=2:B=25:BB=1
450:LINE (A,32)-(B,BB),,,B
460:I=INT (RND 32)
470:Z=VAL RIGHT$ (K$(I),2):WT=WT+Z
480:IF K$(I)=""GOTO 460
490:O=ASC MID$ (K$(I),7,1)
500:K$=LEFT$ (K$(I),11):KK$(I)=K$(I):K$(I)=""
510:IF LEFT$ (K$,4)="Karo"LET B$="081C3E7F3E1C08":GOTO 550
520:IF LEFT$ (K$,4)="Herz"LET B$="060F1F3E1F0F06":GOTO 550
530:IF LEFT$ (K$,4)="Pik"LET B$="183CBEFFBE3C18":GOTO 550
540:IF LEFT$ (K$,4)="Kreuz"LET B$="0E04457F45040E"
550:GOCURSOR X,Y:GPRINT B$
560:GOCURSOR X+1,Y+9:GPRINT X$(O)
570:IF K=1LET S$=" Bank "
580:IF K=0LET S$="Spieler "
590:CURSOR 15,1:PRINT S$:WT
600:IF K=1WAIT 100
610:CURSOR 18,3:PRINT D;" $"
620:WAIT 0

```



PC-1600



```

630:IF K=1GOTO 680
640:IF WT=21BEEP 3:WAIT 100:CURSOR 5,2:PRINT "du hast GEWONNEN...":D=D+100:
GOTO 770
650:IF WT>21BEEP 3:WAIT 100:CURSOR 0,2:PRINT "du hast leider verloren":D=D-100
:GOTO 770
660:CURSOR 13,0:INPUT "Neue Karte ";A$:IF A$="J"GOTO 730
670:CLS :K=1:Z=0:WK=WT:WT=0:GOTO 440
680:IF WT<WKAND K=1GOTO 730:GOTO 460
690:IF WT>21AND K=1BEEP 3:WAIT 100:CURSOR 5,2:PRINT "DU hast gewonnen":D=D+100
:GOTO 770
700:IF WT>WKBEEP 3:WAIT 0:CLS :PRINT " Die Bank hat mit":PRINT " ";WT
;" ";WK
710:WAIT 100:PRINT"" gewonnen":D=D-100:GOTO 770
720:IF K=1GOTO 730
730:A=A+15:AA=AA+3:B=B+15:BB=BB+3
740:LINE (A,32)-(B,BB),,,B:LINE (A+9,AA)-(B-15,32),X
750:X=X+15:Y=Y+3
760:GOTO 460
770:IF D<=0CLS :WAIT 0:PRINT :PRINT "Du bist jetzt pleite":END
780:CLS :WAIT 0:PRINT "Du hast jetzt";D;" $":PRINT
790:INPUT "Noch ein Spiel ? (J/N) ";N$
800:IF N$="N"CLS :END
810:WAIT 0:CLS :FOR I=0TO 32
820:IF LEN KK$(I)>0LET K$(I)=KK$(I)
830:NEXT I
840:CLS :K=0:Z=0:D=0:WT=0:WK=0:GOTO 40

```

Datum und Zeit

Durch Beantworten der einzelnen Fragen und nach entscheidenden Eingaben ist der PC-1600 wieder "up to date". Die Eingaben muessen zweistellig erfolgen z.B. 03 fuer Maerz oder 09 fuer 9 Minuten.

```

10:REM *****
20:REM **Datum und Zeit**
30:REM **by Hans Hölzel**
40:REM
50:REM *****
60:CLS :CLEAR :WAIT 0
70:INPUT "MONAT : ";MO$
80:IF MO$=""BEEP 1:GOSUB 220:GOTO 70
90:INPUT "TAG : ";TG$
100:IF TG$=""BEEP 1:GOSUB 220:GOTO 90
110:DATE$ =MO$+"/"+TG$
120:INPUT "STUNDE : ";S$
130:IF S$=""BEEP 1:GOSUB 220:GOTO 120
140:INPUT "MINUTE : ";MI$
150:IF MI$=""BEEP 1:GOSUB 220:GOTO 140
160:INPUT "SEKUNDE : ";SE$
170:IF SE$=""BEEP 1:GOSUB 220:GOTO 160
180:TIME$ =S$+"."+MI$+"."+SE$
190:CLS :BEEP 3
200:PRINT DATE$ ,TIME$
210:END
220:CLS :PRINT :WAIT 80:PRINT "Eingabe fehlt !":WAIT 0:CLS :RETURN

```

MONAT	:	03
TAG	:	24
STUNDE	:	15
MINUTE	:	

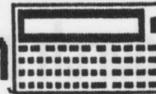
PC1600: ANALOG-UHR von Thomas Jeger
Hauptstrasse 142
CH-3286 Muntelier/FR

Dieses Programm zeigt Ihnen die Uhrzeit analog an, d.h. es wird eine Grafik erzeugt, auf der eine Uhr mit 2 Zeigern zu sehen ist.

```

1:ARUN
5:CLS
10:S=VAL LEFT$(TIME$,2)
11:M=VAL MID$(TIME$,4,2)
-1
30:CLS
31:FOR I=-90TO 269STEP 30:
PSET (78+COS I*16,16+SI
N I*16):NEXT I
35:GOTO 60
40:S=VAL LEFT$(TIME$,2)
42:M=VAL MID$(TIME$,4,2)
44:K=VAL RIGHT$(TIME$,2)
45:CURSOR 0,0:PRINT TIME$
50:IF K<>59THEN GOTO 40
60:IF S>12THEN LET S=S-12
65:N=M*6-90
70:LINE (78,16)-(78+COS (N
)*14,16+SIN (N)*14),R
72:N=N+6
74:LINE (78,16)-(78+COS (N
)*14,16+SIN (N)*14)
80:L=(S*60+M)/2-90
85:LINE (78,16)-(78+COS (L
)*8,16+SIN (L)*8),R
90:L=(S*60+M+1)/2-90
95:LINE (78,16)-(78+COS (L
)*8,16+SIN (L)*8)
100:GOTO 40

```



Michael Quittenden
 Pariser Str.20
 4 Düsseldorf 11

Textprogramm mit Drum und Dran für PC-1600+Plotter+Laufwerk.

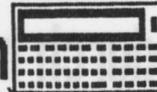
Eigentlich sollte der PC-1600 für den Basic-EMUF (MC) über U24 als Terminal dienen. Die ersten Versuche waren unbefriedigend. Das Endprodukt sieht so aus: PC-1600 als Tastatur, Elektor Vidio-Interface + Monitor als Anzeige, Basic-EMUF als Hirn für stationäre Messaufgaben. Das hieraus entstandene Textprogramm ist komfortable, einfach in der Bedienung und sehr leicht zu ergänzen. Es können, auch getrennt vom Hirn, für den EMUF Basiczeilen erstellt, gesichert und via U24 übertragen werden. Programm-Merkmale: Eingabe von bis zu 70 Zeilen @ 80 Zeichen mit Input-Befehl, also volle Editierfähigkeit. (&22)"muss als (&27)' eingegeben werden. Nach drücken von Enter beep es ca. 3 Sekunden. Ein nochmaliges Drücken während dieser Zeit führt zum Menü. Mit "↑" wird der Text nicht angenommen. Die Zeile kann neu erstellt werden. Ausgabe der Zeilen über Display mit Zeilenangabe, über Plotter oder RS 232. Korrektur der Zeilen, wobei in 40'er Gruppen wegen zu kurzem KBUFFs gearbeitet werden muss. Zeilen können nach Eingabe des Dateinamens unter Extension ".TEX" gesichert werden. Statt des Namens ein "?", bringt Files zur Anzeige. Vor dem absichern werden, "und" Space umcodiert. "SP" am Zeilenanfang geht sonst verloren und ", " beendet die Zeile. Die Umcodierung ist ein Z-80 MP, daher auch sehr schnell. Geladene Zeilen können an schon vorhandenen angehängt werden. Nach Angabe des Dateinamens, können bestimmte ".TEX" Files gelöscht werden. Dieser Text wurde mit meinem Programm geschrieben und auf Diskette gesichert.

Für Progr. ca. 4000 Bytes und für Vari. ca. 6000 Bytes.

```

10:REM      Vers.11.3.87
20:CLEAR :CLS :WAIT 100:PRINT "**** TEXTUERARBEITUNG ****"
   ":T=T+1
30:POKE &F650,&F5,&E5,&C5,&D5,&EB,&01,&50,&00,&3E,&00,&ED
   ,&B1
40:POKE &F65C,&E2,&67,&F6,&3E,&00,&2B,&77,&23,&C3,&58,&F6
   ,&D1,&C1,&E1,&F1,&37,&C9
50:IF T=1DIM A$(1)*80,TE$(70)*80:ZE=1
60:"A"ON ERROR GOTO 1090
70:CLS :INPUT "TEXTEINGABE ? (J/N) ";A$:IF A$="J"GOTO 180
80:CLS :INPUT "TEXTANZEIGE ? (J/N) ";A$:IF A$="J"GOTO 340
90:CLS :INPUT "TEXTAUSGABE ? (J/N) ";A$:IF A$="J"GOTO 390
100:CLS :INPUT "TEXT AENDERN ? (J/N) ";A$:IF A$="J"GOTO 51
   0
110:CLS :INPUT "TEXT SPEICHERN ? (J/N) ";A$:IF A$="J"GOTO
   610
120:CLS :INPUT "TEXT LADEN ? (J/N) ";A$:IF A$="J"GOTO 760
130:CLS :INPUT "FILES SEHEN (J/N) ";A$:IF A$="J"GOTO 1150
140:CLS :INPUT "DISK LOESCHEN (J/N) ";A$:IF A$="J"GOTO 102
   0
150:CLS :INPUT "ENDE (J/N) ";A$:IF A$="J"END
160:GOTO 70
170:REM *TEXT EINGABE*
180:WAIT 10:PRINT "MOMENTANE ZEILEN-NR. ";ZE
190:INPUT "AB ZEILENNUMMER : ";ZE
200:IF ZE<10R ZE>70THEN 180
210:CLS :WAIT 0
220:CORSOR 2,3:PRINT "<< Zeile ";ZE
230:CORSOR 0,0:INPUT TE$(ZE);:A=0:A$=""
240:FOR I=1TO 100:NEXT I
250:FOR I=1TO 10
260:XCALL &E42C:IF A=0GOTO 270ELSE LET I=10:GOTO 280
270:A$=INKEY$:A=ASC A$
280:BEEP 1:NEXT I
290:IF A=&0DTHEN LET ZE=ZE+1:GOTO 70
300:IF A=&0BGOTO 210
310:IF ZE=70WAIT 120:CLS :PRINT "Ich bin voll !":GOTO 70
320:ZE=ZE+1:GOTO 210
330:REM *AUSGABE Über DISPLAY*
340:FOR K=1TO ZE-1
350:CLS :WAIT 0
360:CORSOR 2,3:PRINT "<< Zeile ";K:WAIT
370:CORSOR 0,0:PRINT TE$(K):BEEP 1:NEXT K:GOTO 70
380:REM *AUSGABE Über DRUCKER*
390:CLS :INPUT "Plotter oder RS-232 (1/2) ";X:IF X=2GOTO 9

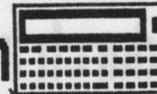
```



```

40
400:TEXT :PAPER C,90,1300
410:INPUT "CSIZE 1-9 ";S:INPUT "COLOR 0-3 ";C
420:CSIZE S:COLOR C
430:PCONSOLE "LPT1:",60,0,0
440:FOR K=1TO ZE-1
450:POKE &F659,&27:POKE &F660,&22:CALL &F650,TE$(K)
460:LPRINT TE$(K)
470:POKE &F659,&22:POKE &F660,&27:CALL &F650,TE$(K)
480:NEXT K
490:GOTO 70
500:REM *TEXT UERBESSERN*
510:CLS :INPUT "WELCHE ZEILE ? ";K
520:IF K=0THEN 70
530:IF K>ZEWAIT 100:PRINT "Zeile NIUD":GOTO 80
540:A$(0)=MID$(TE$(K),1,40)
550:KBUFF$ =A$(0):INPUT " ";A$(0);:IF LEN TE$(K)<40THEN LET
TE$(K)=A$(0):GOTO 70
560:A$(1)=MID$(TE$(K),41,80)
570:KBUFF$ =A$(1):INPUT " ";A$(1);
580:M=LEN A$(0):N=LEN A$(1):IF M+N>80GOTO 590ELSE GOTO 600
590:CLS :WAIT 100:PRINT "ZUVIELE ZEICHEN!":WAIT 0:CLS :
GOTO 540
600:TE$(K)=A$(0)+A$(1):GOTO 80
610:REM *TEXT SPEICHERN*
620:CLOSE :WAIT 0:CLS
630:INPUT "Dateiname = ";NA$:IF NA$=""?"FILES "X:*.TEX":
INPUT NA$
640:IF LEN (NA$)>8LET NA$=LEFT$(NA$,8)
650:NA$="X:"+NA$+".TEX"
660:CLS :PRINT :PRINT "Soll ";NA$;:INPUT "
gespeiche
rt werden (J/N) ";A$:IF A$="N"GOTO 70
670:CLS :WAIT 0:PRINT "TEXT WIRD GESPEICHERT ALS ";N$
680:GOSUB 1180
690:MAXFILES =1:OPEN NA$FOR OUTPUT AS #1
700:PRINT #1,ZE
710:FOR I=1TO ZE:PRINT #1,TE$(I):NEXT I
720:CLOSE #1
730:GOSUB 1230
740:GOTO 70
750:REM *TEXT LADEN*
760:WAIT 0:PRINT "kÖnnen Texte im Rechner":INPUT "gelöscht
werden ? (J/N) ";W$
770:IF W$="N"GOTO 820
780:CLEAR :DIM A$(1)*80,TE$(70)*80:ZE=1
790:POKE &F650,&F5,&E5,&C5,&D5,&E5,&01,&50,&00,&3E,&00,&E
D
,&B1
800:POKE &F65C,&E2,&67,&F6,&3E,&00,&2B,&77,&23,&C3,&58,&F6
,&D1,&C1,&E1,&F1,&37,&C9
810:CLOSE :WAIT 0:CLS
820:CLS :INPUT "Dateiname = ";NA$:IF NA$=""?"FILES "X:*.TEX
":INPUT NA$
830:IF LEN (NA$)>8LET NA$=LEFT$(NA$,8)
840:NA$="X:"+NA$+".TEX"
850:CLS :PRINT :PRINT "Soll ";NA$;:INPUT "
gelade
n werden (J/N) ";A$:IF A$="N"GOTO 70
860:WAIT 0:CLS :PRINT " TEXTE WERDEN GELADEN ALS ";N
A$
870:MAXFILES =1:OPEN NA$FOR INPUT AS #1
880:IF EOF (1)THEN 70
890:INPUT #1,X:X=X+ZE-1
900:FOR I=ZETO X:INPUT #1,TE$(I):NEXT I:CLOSE #1
910:ZE=X:GOSUB 1230
920:GOTO 80
930:REM *AUSDRUCK UEBER RS232 SCHNITTSTELLE*
940:SETDEV "COM1:",PO:SETCOM "COM1:",1200,8,N,1,X,S
950:FOR K=1TO ZE-1
960:POKE &F659,&27:POKE &F660,&22:CALL &F650,TE$(K)
970:LPRINT TE$(K)
980:POKE &F659,&22:POKE &F660,&27:CALL &F650,TE$(K)
990:NEXT K
1000:SETDEV "COM1:":GOTO 70
1010:REM *DISK LÖSCHEN*
1020:WAIT 0:CLS :INPUT "Dateiname = ";NA$:IF NA$=""?"FILES "
X:*.TEX":INPUT NA$
1030:IF LEN (NA$)>8LET NA$=LEFT$(NA$,8)
1040:NA$="X:"+NA$+".TEX"
1050:CLS :PRINT "Soll ";NA$;:INPUT "
gelöscht wer

```



```

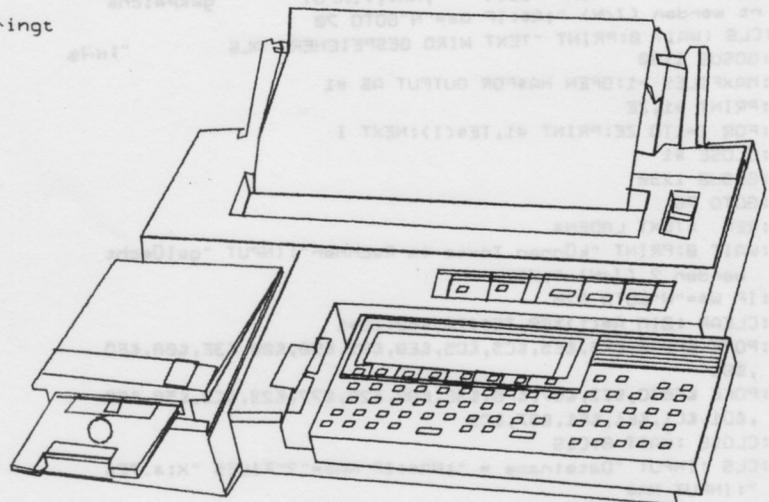
den (J/N) ";A$:IF A$="J"GOTO 1070
1050:GOTO 70
1070:CLOSE :KILL NA$:GOTO 70
1080:REM *FEHLER-BEHANDLUNG*
1090:WAIT :CLS
1100:FOR I=1TO 30:BEEP 1,I,160:NEXT I
1110:IF ERN =152THEN PRINT " DATEI NICHT GEFUNDEN!":RESUME
120
1120:IF ERN =155THEN PRINT " LAUFWERK NIUD !":RESUME 80
1130:CLS :WAIT :PRINT "FEHLER ";ERN ;" in Zeile ";ERL
1140:RESUME 70
1150:CLS :FILES "X:*.TEX":CURSOR 0,3:INPUT "nochmal (J/N) "
;A$:IF A$="J"GOTO 1150
1160:GOTO 70
1170:'Zeichen SP- u. ,-<
1180:POKE &F659,&20:POKE &F660,&5E
1190:FOR I=1TO ZE-1:CALL &F650,TE$(I):NEXT I
1200:POKE &F659,&2C:POKE &F660,&3C
1210:FOR I=1TO ZE-1:CALL &F650,TE$(I):NEXT I:RETURN
1220:'Zeichen ~-SP u. <-,
1230:POKE &F659,&5E:POKE &F660,&20
1240:FOR I=1TO ZE-1:CALL &F650,TE$(I):NEXT I
1250:POKE &F659,&3C:POKE &F660,&2C
1260:FOR I=1TO ZE-1:CALL &F650,TE$(I):NEXT I
1270:RETURN
1280:POKE &F8D0,&ED,&53,&E0,&F8,&37,&C9:INPUT "UARI. TE$(?)
";I:CALL &F8D0,TE$(I):WAIT
1290:B=PEEK &F8E1*256+PEEK &F8E0:PRINT HEX$ B:CLS :WAIT :
INPUT "ADR ";U
1300:PRINT HEX$ U:" ";HEX$ PEEK U:U=U+1:GOTO 1300
MASCH.PRO. für "TEXT"
Adr.F650 =E$-F$ CALL &F650,TE$(?) bringt

```

```

Anfangsadr.von TE$ nach REG. DE.
F650 F5 PUSH AF
F651 E5 PUSH HL
F652 C5 PUSH BC
F653 D5 PUSH DE
F654 EB EX DE,HL
F655 015000 LD BC,0050
F658 3E?? LD A,??
F65A EDB1 CPIR
F65C E267F6 JP PD,F667
F65F 3E?? LD A,??
F661 2B DEC HL
F662 77 LD(HL),A
F663 23 INC HL
F664 C358F6 JP F658
F667 D1 POP DE
F668 C1 POP BC
F669 E1 POP HL
F66A F1 POP AF
F66B 37 SCF
F66C C9 RET

```



F.Kraft, Gordenstr.17 6086 Riedstadt 5

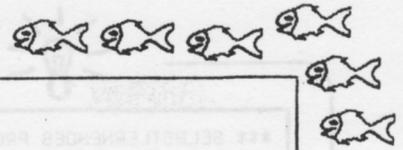
Betr.: Information über PC-1600

Sehr geehrte Damen und Herren,

seit einer Zeit bin ich Besitzer den PC-1600 mit Drucker und Diskettenstation, den ich Privat und teilweise im Betrieb benutze. Nun bereiten mir einige Programmbefehle Schwierigkeiten, für die ich keine Erklärung bzw. Ausweg finden kann.

Ich muss gestehen, dass ich kein Programmierer-Profi, eher ein Anfänger bin. Ich würde Ihnen sehr dankbar wenn Sie mir ein Paar-Tips oder die folgende Fragen erklären würden.

1. Warum wird bei etwas grösseren Programmen, der Befehl-ERASE- nicht ausgeführt (ERROR 109)? Festgestellt habe ich, dass der Fehler nur dann auftritt, wenn vor ERASE Befehl mit FOR-NEXT-Schleife gearbeitet wird. Wird die FOR-NEXT-Schleife durch Zählschleife (z.B. X=0...X=X+1:GOTO usw.) ersetzt, läuft das Programm fehlerfrei nur etwas langsamer.
2. Warum ist die Schreibweise-RIGHT\$(KS\$(0),6)-falsch (ERROR 19), bevor wurde mit -DIM KS\$(0)*22 - dimensioniert?
3. Ist möglich mit einer eindimensionalen Feldvariable über 255 hinaus Daten aufzubauen, da sonst meine Sortierprogramme nur bis 254 (lauf.Nr.) laufen?



Leopold Steininger

PC-1600

Margarethnerstr. 47
Enzersdorf/Fischa
2431 Kleinneusiedl
Österreich

Betrifft: Alles für SHARP-Comuter 9/86

In oben erwähnter Zeitschrift schreiben Sie auf Seite 6 vom Fehlen des scharfen "ß". Ich verwende dieses Zeichen selbst sehr selten. Es ist aber möglich dafür das griechische "BETA" CHR\$ HEX E1 = DEZ 225 zu verwenden.

Ich habe die Reservespeicher folgendermaßen programmiert:
1. Reihe für PROGRAM-MODUS = ; " : \$, &

Das sind die wichtigsten Zeichen, die ich beim Programme schreiben ständig verwende und somit nur mit 1 Tastendruck zu erreichen sind. Für die wichtigsten BASIC-Wörter sind ohnehin Abkürzungen möglich.

2. Reihe für RUN-Modus = CLS PASS MEM frei LOAD-S1 TIME\$
3. Reihe für DATEIEN = SAVE-X,S1 LOAD-X,S1 DISKF.S1 FILES-S1

Ich habe auf der RAM-Disk einen Briefvordruck gespeichert und schreibe, sooft ich Zeit habe, einige Zeilen.

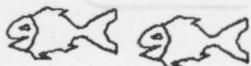
Steininger Leopold

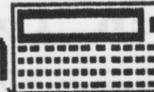
Hier das Programm-Listing; vielleicht könnte ich einiges etwas einfacher erreichen?

```

10:"A":CLS :CLEAR :USING :WAIT 0:CSIZE 2
20:LPRINT " Leopold Steininger"
30:LF 1
40:LPRINT " Margarethnerstr. 47"
50:LPRINT " Enzersdorf/Fischa"
60:LPRINT " 2431 Kleinneusiedl"
70:LPRINT " Österreich"
80:LF 2
90:LPRINT "      an"
100:LF 2
110:LPRINT " Firma"
115:LPRINT " Fischel GmbH."
120:LF 1
130:LPRINT " Friedrich Kaiserstra"+CHR$ &E1+"e 54 a"
140:LPRINT " 1000 Berlin 12"
145:LPRINT " BRD"
160:LF 2
170:LPRINT " Enzersdorf, am 1986/";DATE$
180:LF 2
190:LPRINT " Betrifft: Alles für SHARP-Comuter 9/86"
200:LF 1
205:LPRINT " In oben erwähnter Zeitschrift schreiben Sie auf Seite 6"
210:LPRINT " vom Fehlen des scharfen "+CHR$ 34+CHR$ &E1+CHR$ 34+". Ich verwen
de dieses"
220:LPRINT " Zeichen selbst sehr selten. Es ist aber möglich dafür"
230:LPRINT " das griechische "+CHR$ 34+"BETA"+CHR$ 34+" CHR$ HEX E1 = DEZ 225
zu verwenden."
240:LPRINT " Ich habe die Reservespeicher folgenderma"+CHR$ &E1+"en programmi
ert:"
245:LPRINT " 1.Reihe für PROGRAM-MODUS = ; "+CHR$ 34+" : $ , &"
248:LPRINT " Das sind die wichtigsten Zeichen, die ich beim Programme"
250:LPRINT " schreiben ständig verwende und somit nur mit 1 Tastendruck"
252:LPRINT " zu erreichen sind. Für die wichtigsten BASIC-Wörter sind"
254:LPRINT " ohnehin Abkürzungen möglich."
256:LPRINT " 2.Reihe für RUN-Modus = CLS PASS MEM frei LOAD-S1 TIME$"
258:LPRINT " 3.Reihe für DATEIEN = SAVE-X,S1 LOAD-X,S1 DISKF.S1 FILES-S1"
260:LPRINT " Ich habe auf der RAM-Disk einen Briefvordruck gespeichert"
262:LPRINT " und schreibe, sooft ich Zeit habe, einige Zeilen."
270:LF 2
280:LPRINT "
Steininger Leopold"
282:LF 2
284:LPRINT " Hier das Programm-Listing; vielleicht könnte ich "
286:LPRINT " einiges etwas einfacher erreichen?"
290:END
    
```

PC-1600





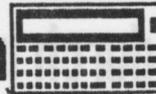
*** SELBSTLERNENDES PROGRAMM ***
*** GESCHRIEBEN VON H.RICHTER ***
*** TULPENWEG 6 4593 HALEN ***
*** MAXIMAL 45 ZEICHEN EINGABE ***

DIESES PROGRAMM ERWIETERT SICH SEBSTSTÄNDIG
DIES GESCHIED INDEM ES KÜNSTLICH DATAZEILEN ERZEUGT
ES KANN DESHALB FÜR VIELE ZWECKE BENUTZT WERDEN
ZUM BEISPIEL ALS SELBSTERWEITERTES LEXIKON ODER ALS WÖRTERBUCH
ODER ALS FRAGE- UND ANTWORT SPIEL
DIESES KURZE PROGRAMM WURDE AUF EINEM PC-1600 GESCHRIEBEN

```
3:REM
4:REM
5:REM
6:REM
7:H=1
8:WAIT :CLS :PRINT " Selbstlernendes Programm"
10:DIM B$(1)*45,C$(1)*45,D$(1)*45,E$(1)*45
24:RESTORE
25:READ E$(1):H=H+1
26:IF E$(1)="-1"LET H=H-4:H=H*10:GOTO 28
27:GOTO 25
28:I=H+210
30:WAIT :CLS :INPUT "Frage mich ";B$(1)
40:RESTORE
50:READ D$(1),E$(1)
60:IF B$(1)=D$(1)CLS :GOTO 90
70:IF D$(1)="-1"AND E$(1)="-1"RESTORE :GOTO 150
80:GOTO 50
90:LE=LEN (E$(1)):WAIT 5
100:FOR J=1TO LE
110:R$=MID$(E$(1),J,1)
120:BEEP 1:PRINT R$;
130:NEXT J
140:WAIT :CURSOR 0,0:PRINT E$(1):CLS :GOTO 30
150:I=I+10:A$=""
160:A$=STR$(I)
170:KBUFF$ =CHR$(31)+A$+"DATA"+CHR$(34)+B$(1)+CHR$(34)+CHR$(13)+CHR$(31)+
" GOTO100"+CHR$(13)
180:CLS :INPUT "Ich habe keine Antwort,gib sie mir: ";C$(1):I=I+10:A$=STR$(I)
190:KBUFF$ =CHR$(31)+A$+"DATA"+CHR$(34)+C$(1)+CHR$(34)+CHR$(13)+CHR$(31)+
" GOTO30"+CHR$(13)
200:DATA "WER BIST DU"
210:DATA "EIN SHARP PC-1600"
30000:DATA "-1"
30010:DATA "-1"
```

MIT DIESEM PROGRAMM KANN MAN SELBSTDEFINIERTER ZEICHEN ERSTELLEN. BEI BEENDIGUNG DER EINGABEN WIRD DAS DEFINIERTER ZEICHEN ,SOWIE DER DAZUGEHÖRIGER GPRINT BEFEHL IN HEXADEZIMAL AUSGEGEBEN.ZUR EINGABE: BEI DER ABFRAGE WIEVIEL SPALTEN MUSS MANN DIE PUNKTLÄNGE IN Y-RICHTUNG EINGEBEN (MAX.12).DANN WIRD BEI DER ABFRAGE BITCODE DER CODE IN BINÄRER FORM EINGEGEBEN (ZB.10011011).EINE 1 FÜR EINEN GESETZTEN-UND EINE 0 FÜR KEINEN GESETZTEN PUNKT.DEN CODE VON UNTEN NACH OBEN EINGEBEN.DIESES PROGRAMM IST AUF EINEM PC-1600 GESCHRIEBEN! MIT DEF C KANN MAN DENN GPRINTBEFEHL AUF DRUCKER AUSGEBEN

```
1000:REM ** H.RICHTER 1987 **
1001:REM ** ZEICHEN DEFINIEREN ****
1002:REM ** MIT GPRINT AUSGABE IN HEX. ***
1010:"B":CLEAR :CLS :Z1=1
1020:DIM VE$(1)*(26)
1030:INPUT "Wieviel Spalten <-->          MAX.12 ";LE:CLS
1040:CURSOR 0,3:PRINT "Bitcode bis zu 8 Zeichen"
1050:FOR B=1TO LE
1060:CURSOR 5,1:PRINT "          "
1065:CURSOR 0,0:PRINT "CODE VON U.NACH O.EINGEBEN"
1070:CURSOR 0,2:PRINT Z1
1080:CURSOR 0,1:INPUT "BITCODE ";A1$
1090:LA=LEN (A1$)
1100:IF LA>8THEN BEEP 2:GOTO 1060
1110:FOR I=0TO LA
1120:IF MID$(A1$,LA-I+1,1)="1"LET I1=2^(I-1):I2=I2+I1
1130:IF MID$(A1$,LA-I+1,1)="2"CLS :PRINT "FALSCHER ZAHLE (>1) ! ":GOTO 1000
0
1140:NEXT I
1150:@$(B)=STR$(I2):I2=0
1160:Z1=Z1+1
1170:NEXT B
1180:CLS
```



```

1190:FOR J1=1TO LE
1200:@(J1+13)=VAL @(J1)
1210:@$(J1)=HEX$( @(J1+13))
1220:IF @$(J1)=""LET @$(J1)="00"
1230:IF MID$( @$(J1),2,1)<>""GOTO 1250
1240:IF @$(J1)>="1"OR @$(J1)<="9"LET @$(J1)="0"+@$(J1)
1250:NEXT J1
1260:PRINT "GPRINT";CHR$( &22);A$;B$;C$;D$;E$;F$;G$;H$;I$;J$;K$;L$;CHR$( &
22)
1270:VE$(1)=A$+B$+C$+D$+E$+F$+G$+H$+I$+J$+K$+L$
1280:CURSOR 0,2:PRINT "Definiertes Z.->"
1290:GCURSOR 110,16:GPRINT VE$(1)
1300:LINE (109,15)-(109+LE+1,24),,,B
1310:LINE (107,13)-(107+LE+5,26),,,B
1320:END
1330:"C"LPRI " GPRINT";CHR$( &22);A$;B$;C$;D$;E$;F$;G$;H$;I$;J$;K$;L$;
CHR$( &22)

800:REM *****
810:REM ***PROGRAMM*****
820:REM ***ZEICHEN DEFINIEREN*****
830:REM ***FÜR PC-1600*****
840:REM *****
900:REM BEI DER ABFRAGE WIEVIEL SPALTEN, ANZAHL DER PUNKTE IN Y-RICHTUNG
910:REM DES ZU DEFINIERENDEN ZEICHENS (MAXIMAL 12 PUNKTE LANG), EINGEBEN
920:REM BITCODE EINGABE VON UNTEN NACH OBEN (X-RICHTUNG) IN BINÄRE EINGABE
930:REM MAXIMAL 8 BITS LANG ZB. 10110100
940:REM DEF(C)=AUSDRUCK FÜR DRUCKER IN HEXADEZIMALER FORM
1000:REM ** H. RICHTER 1987 **
1009:"B":CLEAR :CLS :Z1=1
1020:DIM VE$(1)*(26)
1030:INPUT "Wieviel Spalten <-->          MAX.12 ";LE:CLS
1040:CURSOR 0,3:PRINT "Bitcode bis zu 8 Zeichen"
1050:FOR B=1TO LE
1060:CURSOR 5,0:PRINT "          "
1070:CURSOR 0,1:PRINT Z1
1080:CURSOR 0,0:INPUT "BITCODE ";A1$
1090:LA=LEN (A1$)
1100:IF LA>8THEN BEEP 2:GOTO ,1060
1110:FOR I=0TO LA
1120:IF MID$( A1$,LA-I+1,1)=""LET I1=2^(I-1):I2=I2+I1
1130:IF MID$( A1$,LA-I+1,1)>="2"CLS :PRINT "FALSCH ZAHL (>1) ! ":GOTO 1000
1140:NEXT I
1150:@$(B)=STR$( I2):I2=0
1160:Z1=Z1+1
1170:NEXT B
1180:CLS
1190:FOR J1=1TO LE
1200:@(J1+13)=VAL @(J1)
1210:@$(J1)=HEX$( @(J1+13))
1220:IF @$(J1)=""LET @$(J1)="00"
1230:IF MID$( @$(J1),2,1)<>""GOTO 1250
1240:IF @$(J1)>="1"OR @$(J1)<="9"LET @$(J1)="0"+@$(J1)
1250:NEXT J1
1260:PRINT "GPRINT";CHR$( &22);A$;B$;C$;D$;E$;F$;G$;H$;I$;J$;K$;L$;CHR$( &22)
1270:VE$(1)=A$+B$+C$+D$+E$+F$+G$+H$+I$+J$+K$+L$
1280:CURSOR 0,2:PRINT "Definiertes Z.->"
1290:GCURSOR 110,16:GPRINT VE$(1)
1300:LINE (109,15)-(109+LE+1,24),,,B
1310:LINE (107,13)-(107+LE+5,26),,,B
1320:END
1330:"C"LPRI " GPRINT";CHR$( &22);A$;B$;C$;D$;E$;F$;G$;H$;I$;J$;K$;L$;CHR$(
22)

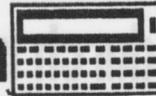
```



Endlich klappt meine Wertpapierverwaltung. Das neue Buch sagt auch Ihnen wie es geht !



durch Information vorn



PC-1600: Hexmonitor

HEXMONITOR ist ein Programm, das das Ändern von Speicherinhalten erlaubt und diese anzeigt. Was Sie früher mittels POKE und PEEK mühsam eingaben und auslasen, übernimmt ab jetzt dieses Programm. Nach Start des Programm und Eingabe der Anfangsadresse, erscheint (z.B.) folgendes Bild:

PC-1600

C10C:65 73 73 65 "esse"	Thomas Jeger
C110:3A 22 3B 41 ":",A"	Hauptstrasse 142
C114:4E 0D 00 28 "N ("	CH-3286 Muntelier
C118:04 F1 B3 30 " +/0"	
..etc.	

Die vier ersten Zeichen zeigen die momentane Adresse (in Hexadezimaler Form). Danach folgen 4 Hex-Zahlen. Dies sind die Speicherinhalte der Speicherzellen. Darauf folgend kommt eine Zeichenkette von 4 Buchstaben. Jeder dieser Buchstaben ist ein CHR\$-Code des dazugehörigen Speicherinhaltes.

Man hat natürlich noch zusätzliche Funktionen. Ein Druck auf die \downarrow Taste bewirkt, dass die Adresse um 4 erhöht und die 4 nächsten Speicherzellen angezeigt werden. Die Taste \uparrow hat die gleiche Funktion in umgekehrter Richtung. Drücken Sie die ENTER-Taste, können Sie die Werte abändern. Es erscheint vor der Zeichenkette der Cursor, mit dem Sie durch die ganze Zeile wandern können. Sie dürfen sogar die Adresse ändern. Sie sollten jedoch nur Zahlen im HEX-Format eingeben, das sonst andere Werte als gewünscht hineingeschrieben werden. Bei Druck auf die ENTER-Taste werden dann die neuen Werte in die gewünschten Adressen geschrieben.

Mit der A -Taste ändern Sie das Darstellungsformat. Es wird auf das ASCII-Format umgeschaltet. Statt den HEX-Werten von nur 4 Bytes erhalten Sie nun die CHR\$-Codes von 8 Bytes und diese kann man ebenfalls ändern: Man drückt auch hier RETURN, der Cursor erscheint und man ändert nach Wunsch ab.

```

5:DIM X$(5)*5,X(5)
10:CLS
20:PRINT " -- HEX-MONITOR --"
30:PRINT :INPUT "Anfangsadresse:";AN
40:WAIT 0
50:IF AN<0 THEN LET AN=&FFFF
52:IF AN>&FFFFFF THEN LET AN=0
54:K$=RIGHT$("000"+HEX$ AN,4)+": "
55:IF SL=1 THEN GOTO 400
60:FOR I=0 TO 3
70:K$=K$+RIGHT$("0"+HEX$(PEEK(I+AN)),2)+" "
80:NEXT I
90:PRINT K$;" ";CHR$ 34;
100:FOR I=0 TO 3:IF PEEK(I+AN)=0 THEN PRINT " ";:GOTO 107
105:PRINT CHR$ PEEK(I+AN);
107:NEXT I
110:PRINT CHR$ 34
120:A$=INKEY$(1)
130:IF A$=CHR$ 13 AND SL=1 THEN 500
132:IF A$=CHR$ 13 AND SL=0 THEN 200
135:IF A$=CHR$ 10 AND SL=1 THEN LET AN=AN+4
138:IF A$=CHR$ 10 THEN LET AN=AN+4:GOTO 50
140:IF A$=CHR$ 11 AND SL=1 THEN LET AN=AN-4
142:IF A$=CHR$ 11 THEN LET AN=AN-4:GOTO 50
145:IF A$="A" AND SL=0 THEN LET SL=1:GOTO 50
150:IF A$="A" AND SL=1 THEN LET SL=0:GOTO 50
199:GOTO 120
200:CURSOR 0,3:KBUFF$ =K$:INPUT K$:GOSUB 210:GOTO 300
210:X$=LEFT$(K$,4)
220:FOR I=0 TO 3:X=ASC MID$(X$,I+1,1):IF X>57 THEN LET X$=LEFT$(X$,I)+CHR$(X-7)+RIGHT$(X$,3-I)
230:NEXT I
240:X=0:FOR I=0 TO 3:X=X+(ASC MID$(X$,I+1,1)-48)*2^(4*(3-I)):NEXT I
250:FOR I=0 TO 3:X$(I)=MID$(K$,6+I*3,2):NEXT I
255:IF SL=1 THEN RETURN
260:FOR I=0 TO 3
270:FOR J=1 TO 2:Y$=MID$(X$(I),J,1)
272:IF Y$>"9" THEN LET Y$=CHR$(ASC Y$-7):X$(I)=LEFT$(X$(I),J-1)+Y$+RIGHT$(X$(I),2-J)
275:NEXT J
280:X(I)=16*(ASC(LEFT$(X$(I),1))-48)+ASC(RIGHT$(X$(I),1))-48
290:NEXT I:RETURN
300:POKE X,X(0),X(1),X(2),X(3):CURSOR 0,3:GOTO 50
400:K$=K$+" "
410:FOR I=0 TO 7:K$=K$+CHR$ PEEK(AN+I)
415:IF PEEK(AN+I)=0 THEN LET K$=K$+" "
417:NEXT I
420:K$=K$+" "
430:PRINT K$:GOTO 120
500:KBUFF$ =K$
510:CURSOR 0,3:INPUT K$
520:GOSUB 210
530:FOR I=0 TO 7
540:POKE X+I,ASC MID$(K$,7+I,1):NEXT I
550:GOTO 55

```

PC-1600

Thomas Jeger
Hauptstrasse 142

PC-1600

CH-3286 Muntelier/FR FUNKTIONSPLOTTER

Dieses Programm zeichnet Ihnen irgend eine Funktion auf das Display des PC-1600. Nachdem Sie das Programm abgetippt haben, starten Sie es mit "RUN". Der Computer fragt Sie jetzt nach den Grenzen des Ausschnitts, den Sie auf Ihrem Display haben wollen. Er beginnt mit der Frage nach "XMIN:". Hier geben Sie die untere Grenze auf der X-Achse ein. Die nächste Eingabe, ist die obere Grenze auf der X-Achse. Danach erfolgt die gleiche Prozedur für die Y-Achse. Werden die Eingaben als richtig erkannt, können Sie die Funktion eingeben. Diese muss die Variable X beinhalten. Eine mögliche Funktion kann wie folgt aussehen:

$$Y=0.3*X^4-1.2*X^3+2*X^2-1$$

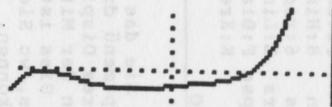
Diese Eingabe müssen Sie mit ENTER abschliessen. Jetzt programmiert sich der Computer selbst, d.h. er generiert eine Programmzeile (Zeile 170). Nun wird die Funktion gezeichnet. Die Linie auf der rechten Hälfte des Displays, ist eine Abgrenzung. Diese ist notwendig, um rechts daneben noch Text darstellen zu können. Wenn keine Funktion zu sehen ist, und sich der Computer mit "X-INPUT" meldet, haben Sie den Bereich der Funktion falsch gewählt. Stoppen Sie das Programm mit der Break-Taste und starten Sie es erneut mit "RUN".

Währenddem das Programm die Funktion auf das Display zeichnet, könne Sie die ENTER-Taste drücken. Es werden Ihnen die Koordinaten des momentan zeichnenden Punktes gezeigt. Nachdem die Funktion im gewählten Bereich fertig gezeichnet wurde, erscheint "X-INPUT". Sie können nun einen X-Wert (der aber in dem von Ihnen gewählten Bereich ist) eingeben. Das Programm rechnet den Y-Wert aus und setzt einen blinkenden Cursor an diese Stelle im Display. Wenn Sie nun die oder -Taste drücken, wandert der Cursor hinauf/hinunter. Wollen Sie einen weiteren X-Wert eingeben, drücken Sie die SPACE-Taste.

Hardcopy: Das Programm beinhaltet ausserdem eine Interruptgesteuerte Hardcopy-routine. Ein Druck auf die F1 Taste und der momentane Displayinhalt wird auf dem angeschlossenen CE-1600P ausgegeben.

Wollen Sie die Eingabe der Funktion überspringen, d.h. Sie wollen die bereits eingetragene Funktion beibehalten, können Sie mit DEF+A diese Eingabe überspringen.

```
5:ON KEY GOSUB 2000
8:KEY (1)ON
10:CLS :CLEAR :DIM A$(0)*80
15:ON ERROR GOTO 1000
20:INPUT "XMIN:";X0
30:INPUT "XMAX:";X1
40:INPUT "YMIN:";Y0
50:INPUT "YMAX:";Y1
60:X2=106/(X1-X0)
70:Y2=32/(Y1-Y0)
80:CLS
```



Hardcopy bei Druck auf F1

```
100:PRINT "Geben Sie die Funktion ein"
110:KBUFF$ ="Y=":INPUT A$(0)
120:KBUFF$ =CHR$ 31+"170"+A$(0)+CHR$ 13+CHR$ 31+CHR$ 27+"A"
130:END
140:"A"CLS
142:LINE (0,32+(Y0*Y2))-(<106,32+(Y0*Y2)),, &1111
143:IF (-X0*X2)>106THEN GOTO 146
145:LINE (-X0*X2,0)-(-X0*X2,32),, &1111
146:LINE (106,0)-(<106,32)
147:LINE (-17,16)-(<-17,16)
150:FOR I=0TO 106
160:X=(I+X0*X2)/X2:GOSUB 170:GOTO 180
170:Y=
172:RETURN
180:LINE -(I,32-Y*Y2+Y0*Y2)
184:IF INKEY$ (<CHR$ 13THEN 190
185:CUSOR 18,0:PRINT " " "":CURSOR 18,0:PRINT INT (X*1000)/1000
187:CUSOR 18,1:PRINT " " "":CURSOR 18,1:PRINT INT (Y*1000)/1000
190:NEXT I
200:CUSOR 18,0:PRINT "X-Input:"
202:CUSOR 18,1:PRINT " " "":CURSOR 18,1:INPUT X
210:IF X<X0OR X>X1THEN GOTO 200
220:GOSUB 170
225:CUSOR 18,2:PRINT " " "":CURSOR 18,2:PRINT INT (Y*1000)/1000
230:I=X2*(X-X0)
240:PSET (I,32-Y*Y2+Y0*Y2),X
245:FOR K=1TO 5
248:I#=INKEY$
250:IF I#="" THEN LET K=99
252:IF I#=CHR$ 10THEN LET K=9
253:IF I#=CHR$ 11THEN LET K=10
255:NEXT K
257:IF K(<6)THEN 270
260:GOTO 240
270:PSET (I,32-Y*Y2+Y0*Y2)
275:IF K=100THEN 200
278:IF K=10THEN LET X=X-1/X2
279:IF K=11THEN LET X=X+1/X2
280:CUSOR 18,1:PRINT " " "":CURSOR 18,1:PRINT INT (X*1000)/1000
290:GOTO 210
999:END
1000:'FEHLERBEHANDLUNG
1010:IF ERN =38AND ERN <100THEN RESUME 10
1020:CLS :PRINT "Bereich falsch gewählt !":FOR I=1TO 1000:NEXT I:RESUME 10
2000:GRAPH
2010:FOR Y=31TO 0STEP -1
2020:FOR X=0TO 155
2030:IF POINT (X,31-Y)=1THEN LLINE (X*3,Y*3)-(X*3+3,Y*3+3),,B
2040:NEXT X:NEXT Y:RETI
```

STATUS 1

1274



HARASCHO

HARASCHO

Inhalt

Seite 2

HARASCHO

Ein Grafikprogramm von Thomas Jeger fuer den SHARP PC-1600 und CE-1600

Anleitung

- Installierung und Start des Programmes 3
- Cursorfunktionen 4
- Linien, Rechtecke und ausgefüllte Rechtecke .. 5
- Kreise und Ellipsen 6
- Invertierung des Displays, Display löschen .. 7
- Abspeichern, aufladen von Bildern 8
- Ausdrucken von Bildern 9
- Modifikationen am Programm 10

Ein Grafikprogramm für den SHARP PC-1600 mit CE-1600P und CE-1600F oder RAM-Disk als Optionen.

(c) by FISCHEL GmbH
Kaiser-Friedrichstrasse 54a
D-1000 Berlin 12

Alle Rechte vorbehalten. Ohne unsere ausdrückliche Genehmigung des Herausgebers ist es nicht gestattet, die Anleitung oder Teile daraus auf fotomechanischem (Foto-/Mikrokopie) oder sonstigem Wege zu vervielfältigen. Gleiches gilt für das mitgelieferte Programm. Für etwaige Schäden durch Anwendung des Programmes übernehmen wir keine Haftung.



HARASCHO ist SPITZE

- Ⓢ Punkt erscheinen lassen
- ⓪ Punkt löschen

HARASCHO Installation und Start Seite 3

HARASCHO Linien, Rechtecke (gefüllte) Seite 5

Tippen Sie das Listing ab (sofern Sie die Listing-Version bevorzugt haben), und speichern Sie es auf eine Kassette oder Diskette ab.

Legen Sie die Kassette in den Rekorder, und tippen Sie "CLOAD" (Abschliessen mit ENTER) in den Rechner. Das Programm wird in den Arbeitsspeicher geladen.

Start des Programmes:

Den ersten Start muss unbedingt mit RUN (+ENTER) durchgeführt werden, da im Programm Variablen definiert werden. Zudem werden die Variablen belegt. Dieser Vorgang dauert bis zu einer Minute. Dies ist notwendig, damit man später (vor allem bei der Kreis-/Ellipsenfunktion) nicht so lange warten muss. Jeder weitere Start kann mit DEF+Space (Leerzeichen) erfolgen. So wird die lange Wartezeit am Anfang übersprungen und direkt das Hauptmenü angesprochen.

Das Hauptmenü sehen Sie nur bei jedem Start des Programm. Einmal mit zeichnen begonnen, kann man es nicht mehr konsultieren!

Folgende Funktionen werden angezeigt: (Mit den dafür notwendigen Abkürzungen)

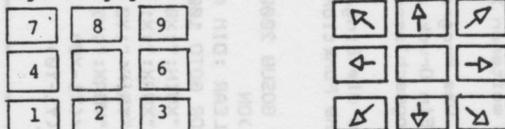
- | | | |
|------------|------------|--------------------------|
| F1:SAVE | F2:LOAD | F3:PRINT |
| 2 :Unten | 8:Hinauf | 5 :Punkt setzen |
| 4 :links | 6:rechts | 0 :Punkt löschen |
| X :Invers | L:Linien | B :Box (Rechtecke) |
| E :Ellipse | F:Blackbox | C :CLS (Löschen Display) |
| Q :Ende | K:Kreis | ENTER:Punkt markieren. |

HARASCHO Cursorfunktionen Seite 4

Nachdem Sie das Programm gestartet haben und das Hauptmenü durchgeblättert haben, sehen Sie ein leeres Display mit einem flatterenden Punkt in der Mitte. Dieser Punkt ist der Cursor. Dies ist die aktuelle Stelle des Displays, wo Sie die diversen Funktionen ausführen können.

Die Steuerung des Cursors:

Mit folgenden Tasten können Sie den Cursor steuern. Neben sind die jeweiligen Richtungen angegeben:



Die Tasten 5 und 0 dienen dazu, um einen Punkt erscheinen zu lassen oder aber um einen Punkt zu löschen. Dabei gilt:

Eine wichtige Funktionstaste dieses Programm ist die ENTER-Taste. Mit ihr kann man Punkte "markieren".

Linien:

Gehen Sie mit dem Cursor auf den Anfangspunkt der zu zeichnenden Linie. Drücken Sie ENTER (ein Beep ertönt). Nun gehen Sie mit dem Cursor auf den Endpunkt der Linie und drücken die **L**-Taste. Jetzt wird die Linie gezeichnet. Der Anfangspunkt der Linie ist immer noch gespeichert, Sie können also, wenn mehrere Linien von einem Punkt ausgehen, nur an den Endpunkt der nächsten Linie gehen und wieder die **L**-Taste drücken.

Rechtecke, gefüllte Rechtecke:

Die Verhaltensweise ist hier genau dieselbe. Nur, dass Sie einfach auf den Anfangspunkt einer Diagonale gehen müssen, ENTER drücken und danach auf den Endpunkt der Diagonale gehen und die **B**-Taste (B wie Box <engl. für Rechteck>) drücken müssen.

Ist ein ausgefülltes Rechteck erwünscht, so müssen Sie statt der **B**-Taste die **F**-Taste drücken.



BRD + DDR

HARASCHO Kreise, Ellipsen Seite 6

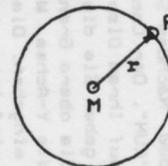
Als kleine Sensation dieses Programmes, können die Funktionen "Kreis" und "Ellipse" angesehen werden. Diese erlauben, wie der Name schon sagt, das Zeichnen von Kreisen oder Ellipsen.

Kreise:

Gehen Sie zuerst mit dem Cursor zum Mittelpunkt des zu zeichnenden Kreises und drücken Sie die ENTER-Taste. Nun können Sie mit dem Cursor in irgend eine Richtung fahren: Wenn Sie jetzt die **K**-Taste drücken, wird ein Kreis gezeichnet, der durch die momentane Cursorposition geht.

Skizze:

- M = Mittelpunkt
- P = Cursorposition
- r = Radius (wird nicht gezeichnet)



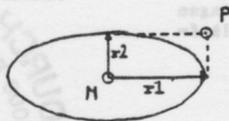
HARASCHO ist SPITZE

Ellipse:

Zuerst gehen Sie, wie beim Kreis, mit dem Cursor zum Mittelpunkt der Ellipse und drücken ENTER. Für den Radius in X-Richtung, wandern Sie in X-Richtung (also mit den Tasten 4 oder 6) bis der gewünschte Radius erreicht ist. Ebenso für den Radius in Y-Richtung. Wenn Sie die Radien 'anvisiert' haben, drücken Sie die **(E)**-Taste und die Ellipse wird gezeichnet.

Skizze:

M = Mittelpunkt
P = Cursorposition
r1 = Radius in X-Richtung
r2 = Radius in Y-Richtung



Bei Wahl eines zu grossen Radius entstehen Lücken im Kreis (in der Ellipse), die Sie "von Hand" korrigieren müssen.

HARASCHO Invertierung / Löschen Seite

Mit der **(X)**-Taste wird das ganze Display invertiert. D.h. die Punkte die gesetzt waren, werden gelöscht und die Punkte die gelöscht waren, werden gesetzt. Kommen Sie irrtümlich auf diese Taste, versetzt ein erneuter Druck auf diese Taste das Display in den ursprünglichen Zustand. Diese Gegebenheit können wir uns zu Nutze machen! Angenommen Sie haben eine Linie gezeichnet, die Sie eigentlich gar nicht an diesem Platz haben wollten. Drücken Sie nun die **(X)**-Taste und das Display wird invertiert. Ein erneuter Druck auf die **(L)**-Taste und die Linie wird noch einmal gezeichnet. Wenn Sie nun wieder die **(X)**-Taste drücken, erscheint das Bild wieder im ursprünglichen Zustand (die Linie wurde gelöscht). Diese "Korrektur"-Funktion lässt sich auf alle Zeichnungsfunktionen (also "Kreise", "Ellipse", "Rechtecke", "gefüllte Rechtecke" und "Linien") anwenden.

Löschen des Displayinhaltes:

Ein Druck auf die **(C)**-Taste ("C" steht für "CLEAR" <engl. für "Löschen">) und der gesamte Displayinhalt wird gelöscht. Es gibt keine Funktion, die diese Funktion rückgängig macht, seien Sie sich also im Klaren, dass Sie das Bild zuvor ausgedruckt oder abgespeichert haben müssen.



BERLIN ist eine Reise wert!!!

HARASCHO Abspeichern, aufladen Seite 8

Eine sehr wichtige Funktion des Programmes, ist die Abspeicherungsfunktion. Mit ihr lässt sich ein Bild entweder auf Diskette oder aber auf RAM-Disk abspeichern. Bei der Version, die Sie auf Kassette bekommen, ist das Programm auf RAM-Disk eingestellt. Verwenden Sie ein Diskettenlaufwerk, so ändern Sie bitte diese Zeilen wie folgt:

```
10000:BEEP 1:OPEN "X:"+NA$FOR OUTPUT AS #1
11000:BEEP 1:OPEN "X:"+NA$FOR INPUT AS #1
```

Was sich ändert, ist lediglich das "X:" statt des "S1:".

Beide Funktionen, das Abspeichern und das Aufladen von Bildern, sind interruptgesteuert, d.h. man kann egal welche Funktion gerade in Arbeit ist, diese Tasten drücken und das Bild wird abgespeichert resp. eingeladen. Nach der F1 oder F2 Funktion wird wieder aus der Interruptroutine herausgesprungen und die (vor dem Einsprung) behandelte Routine weiter bearbeitet. Am Anfang einer dieser beiden Funktionen ertönt ein Beep und am Schluss einer solchen ertönt ein Beep. Somit wissen Sie, ob Sie die Taste richtig gedrückt haben und, ob das Bild schon fertig gespeichert / aufgeladen wurde.

Beim Abspeichern ist es wichtig, dass Sie vor diesem Vorgang mit dem Cursor auf eine leere

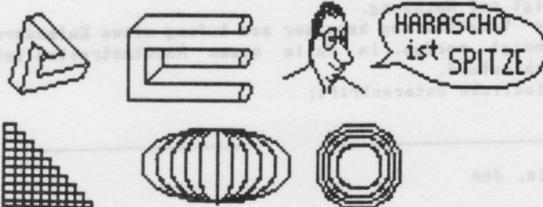
Stelle gehen (auf einen gelöschten Punkt). Dies ist notwendig, damit der Cursor nicht mit abgespeichert wird.

Es ist leider nicht möglich, mehr als ein Bild abzuspeichern. Ebenso ist es nicht möglich auf Kassette abzuspeichern.

HARASCHO Ausdrucken Seite 9

Ein Druck auf die F3-Taste (Das ist die Taste, die unterhalb dem Display mit # angeschrieben ist), wird das Bild über den angeschlossenen CE-1600 P ausgedruckt. Auch diese Funktion ist interruptgesteuert, d.h. auch hier wird die aktuelle Cursorposition gelöscht, damit der Cursor nicht mit ausgedruckt wird. Gehen Sie deshalb mit dem Cursor auf eine freie Position, damit das zu ausdrückende Bild nicht verunstaltet wird.

Beispielausdrucke:



Der Ausdruck kann bei Druck auf irgend eine Taste gestoppt werden. Das Programm springt dann wieder in den Editiermodus.

HARASCHO Modifikationen Seite 10

Das Programm ist nicht geschützt, damit Sie ohne Probleme Sicherheitskopien sowie Änderungen am Programm machen können. Doch sei an dieser Stelle darauf hingewiesen, daß das Weitergeben von Sicherheitskopien oder modifizierten Programmen gesetzwidrig ist und stark bestraft werden kann! Dieses Programm untersteht den geltenden Gesetzen über Software. Jedoch dürfen für den Eigengebrauch Modifikationen am Programm vorgenommen werden.

Ab Speichern von mehreren Bildern:

In Zeile 140 steht der Name, unter welchem ein Bild auf Diskette / RAM-Disk abgespeichert resp. aufgeladen wird. Ändern Sie hier den Namen, können Sie also noch andere Bilder abspeichern.

Einfügen von zusätzlichen Funktionen:

Nach der Zeile 290 steht Ihnen noch genügend Platz für die Programmierung von eigenen Funktionen zur Verfügung (z.B. Teilinvertierung des Displays, ausgefüllte Kreise, Abspeichern auf Kassette, Ausdrucken auf Matrixdrucker über RS-232 Schnittstelle, etc...).

Ändern der Grösse des Ausdrucks auf CE-1600P:

In Zeile 12000 steht der Wert für die Grösse der beim Ausdruck entstehenden kleinen Quadrate. Eingestellt ist hier "4". Beim Ändern der Variablen G, ändert sich auch die Grösse des Ausdrucks.

Zu bestellen bei: FISCHEL GmbH
Kaiser-Friedrichstr. 54a
D-1000 Berlin 12

Das Programm wird auf M-Kassette (Preis: DM 49,- inkl. 14% MwSt.)
oder als Listing (Preis: DM 39,- inkl. 14% MwSt.)
geliefert.



HARASCHO ist SPITZE

Sehr geehrter SHARP-Software-Lieferant!

A) ALLGEMEINES

Für eine Zusammenarbeit gibt es drei Möglichkeiten:

- a) Fachbeiträge werden honoriert.
- b) Anzeigenschaltung
- c) Senden Sie uns die beiliegende Provisionsvereinbarung unterschrieben zurück. Wir werden dann Ihre Werbeprospekte und Programmbeschreibungen kostenlos in unserer Zeitschrift "Alles für SHARP-Computer" abdrucken und die eingehenden Bestellungen an Sie weiterleiten. Diese Form ist die bei weiten vorteilhafteste für beide Seiten.

B) VEREINBARUNG

Bestellungen für Software von _____ dürfen von Bernd Fischel entgegengenommen werden. Diese Bestellungen müssen an _____ weitergeleitet werden. Für die Entgegennahme von Bestellungen und deren Weiterleitung erhält Bernd Fischel je Bestellung ein Brutto-Entgelt (die gesetzliche MWST ist bereits darin enthalten) in Höhe von % (_____ Prozent) des Bruttoverkaufspreises (= VK incl. MWST). Die Abrechnung der Provision erfolgt auf Rechnung. Diese Vereinbarung kann nur zum Anfang eines Kalenderquartals gekündigt werden. Im Falle eines Rechtsstreites ist Berlin Gerichtsstand.

verbindliche Unterschrift:

Berlin, den _____

C) ANZEIGEN IN "ALLES FÜR SHARP COMPUTER"

Sehr geehrter Anzeigen-Interessent, die erste Ausgabe unserer neuen Monatszeitschrift "Alles für Sharp Computer" ist planmäßig im Januar 1985 in einer Auflage von 26.000 Stück bundesweit erschienen. Nutzen Sie diese einmalige Chance, tausend von Sharp-Anwendern direkt zu erreichen und geben Sie eine Anzeige auf. Sie wissen ja, Werbung ist wichtig. Dazu haben Sie drei Möglichkeiten:

1.) Produktanzeigen

Im Moment gilt unsere Anzeigenpreisliste vom Dez. 1984, die auch über die möglichen Formate Auskunft gibt. Preisnachlässe auf die dort genannten Preise sind nach persönlicher Rücksprache in bestimmten Fällen möglich. Weitere Informationen auf Anfrage.

2.) Kleinanzeigen

Wenn Sie eine Kleinanzeige aufgeben wollen, senden Sie uns bitte den unten folgenden Abschnitt mit einem Scheck über den zu leistenden Betrag zu, sonst wird die Kleinanzeige nicht bearbeitet. Wir behalten uns das Recht vor, Kleinanzeigen ohne Angaben von Gründen abzulehnen. Eingereichte Schecks werden dann natürlich umgehend zurückerstattet.

Bitte ausfüllen, ausschneiden und an die Fischel GmbH senden.

Bitte veröffentlichen Sie in der nächsten erreichbaren Ausgabe von "Alles für Sharp Computer" den folgenden Text:

Bitte nur 30 Buchstaben pro Zeile einschließlich Satzzeichen und Wortzwischenräumen eintragen (ein Strich entspricht einem Zeichen). Jede angefangene Zeile kostet 6 DM incl. 7% Mwst. Der Gesamtbetrag liegt als Scheck bei.

Name: _____
 Straße, Nr.: _____
 PLZ, Ort: _____
 Telefon (für Rückfragen): _____
 Datum, Unterschrift: _____

D) RICHTLINIEN

Ein an "Alles für Sharp Computer" zur Veröffentlichung eingereichter Beitrag muß in jedem Fall folgenden Richtlinien entsprechen:

- nicht mehr als 3 bis 4 Seiten Umfang
- druckgerechter Aufbau (siehe dazu als Beispiel die Beiträge von Heft 2, 3 usw.)
- die Seiten dürfen nicht zu viele Leerräume aufweisen
- mindestens 1,5 cm Rand oben und unten und 2 cm rechts und links, aber nicht wesentlich mehr
- der Text muß mit einem frischen Farbband auf weißem Papier stehen, farbiger Hintergrund ist ungeeignet, desgleichen kariertes, gelochtes oder geklamertes Papier.
- eventuelle Kopien dürfen nicht schwach erscheinen, sondern müssen kontrastreich sein. Soweit möglich, unbedingt Originale einreichen



- DER FACHVERLAG FÜR
 TASCHEN-COMPUTER!

PC-1600

PC-1600

Anwendungshandbuch zum Sharp PC - 1600 Taschencomputer system

ISBN: 3-924327-55-6

PC-1600

