

nur zur privaten Verwendung

hoch interessant

SHARP- PC-1600 Fundus

PC-1600-Beiträge aus den
Zeitschriften:
Alles für Sharp-Computer
Pocket und Laptop
La revue des Sharpentiers
Pockecom Journal

exklusiv für PCUC-Mitglieder

streng vertraulich

SHARP-PC-1600-Fundus

ISBN die allerletzte aus der Reihe

Autoren: ganz viele

Herausgeber:ich

Layout: ich, Schere, Klebstoff und mein Kopierer

Druck: irgendwo im Kopierkeller

Copyright: zum Teufel

Keine Rechte vorbehalten. Mit ausdrücklicher Genehmigung des Herausgebers ist es gestattet, diese Sammlung oder Teile daraus für private Zwecke auf fotomechanischem oder sonstigem Wege zu vervielfältigen. Für etwaige Schäden durch Anwendung der Beiträge oder Programme übernehme ich sowieso keine Haftung.

Vorwort

Hauptbestandteil dieser Beitragsammlung speziell für, den SHARP-PC-1600 sind die Beiträge aus den "Büschel-Heften" (Fischel) mit der, Idee, den riesigen Papierwust mit unnützen Reklame- und Schnick-Schnack-Flächen ein Ende zu setzen und damit wieder einen Überblick in Sachen PC-1600 zu gewinnen. Daraus ergab sich ein Ertrag von etwa 25:1; will heißen durchschnittlich jede 25.Seiten konnte ich verwenden. In gleicher Weise verfuhr ich auch mit den mir zur Verfügung stehenden französischen und japanischen Zeitschriften.

Von Layout und Qualität kann leider nicht die Rede sein. Ein Grund dafür war u.a. die zusammenkopierten Beiträge aus den erstgenannten Heften, welche ohnehin schon nicht in ihrer Qualität glänzten.

Trotzdem wollte ich unseren Club-Mitgliedern diese Information nicht vorenthalten und habe es mit einer Auflage von zunächst 8 (i.W. acht) vervielfältigt. Wer, also ausser den mir am nächsten Stehenden ebenfalls ein Exemplar wünschen Möge mir reichlich Briefmarken und eine Anforderung zukommen lassen

Viel Spaß
Peter Lawatsch

PC-1600

I N H A L T S V E R Z E I C H N I S

(Erklärungen zum Programmstartcode)
(A : Allgemein)
(D : Datenverwaltung)
(E : Elektro-/Elektronik)
(F : Finanzen)
(G : Grafik)
(I : Ingenieur-Software)
(M : Mathematik)
(P : Physik/Chemie)
(S : Spiele)
(T : Textverarbeitung)
(Z : Astronomie)
(:)
(H : Hardware)
(U : Utilities/Tools)
(X : PC- Intern)

Code : Beitrag	"Büschel"s"	
	Alles f. Sharp	Computer, und
	Pocket&Laptop	Heft/S. : Seite
:(Titelseite)	:	: 004
U :Erste Tips/Tricks (CE-150)	: 21/08	: 005
G :Grafik-Demo (Invers-Total)	: 22/25	: 005
D :PC-1600 Besonderh.(Abspeich.v.Text.)	: 23/09	: 005
H :Interrobotik	: 24/04	: 006
U : "sz" und andere Besonderheiten	: 24/14	: 008
G :Display-Kopierung	: 24/15	: 008
G :Berliner-Uhr	: 24/16	: 009
S :Lernspiel	: 24/24	: 010
G :GCURSOR / CURSOR	: 24/49	: 011
M :Sinusfunktion	: 24/48	: 011
E :Messprogramm	: 25/29	: 012
U :Hardcopy	: 25/56	: 014
M :Multiplikaons-Übung	: 25/06	: 014
U :Formatieren auf DIN A5	: 26/	: 015
S :4-gewinnt	: 27/08	: 015
T :Einfache Textverarbeitung	: 27/16	: 016
G :Verbesserung zu (24/49)	: 27/20	: 016
S :Flugsimulatur	: 27/49	: 017
X :ASCII-Code	: 28/07	: 021
M :Dezimal-hexadezimal-binär-Umwandlung	: 29/10	: 022
G :Selbstbildnis	: 30/05	: 023
G :Schriftprobe	: 30/08	: 025
M :Umrechnung Dezimal- in Dualzahlen	: 31/39	: 025

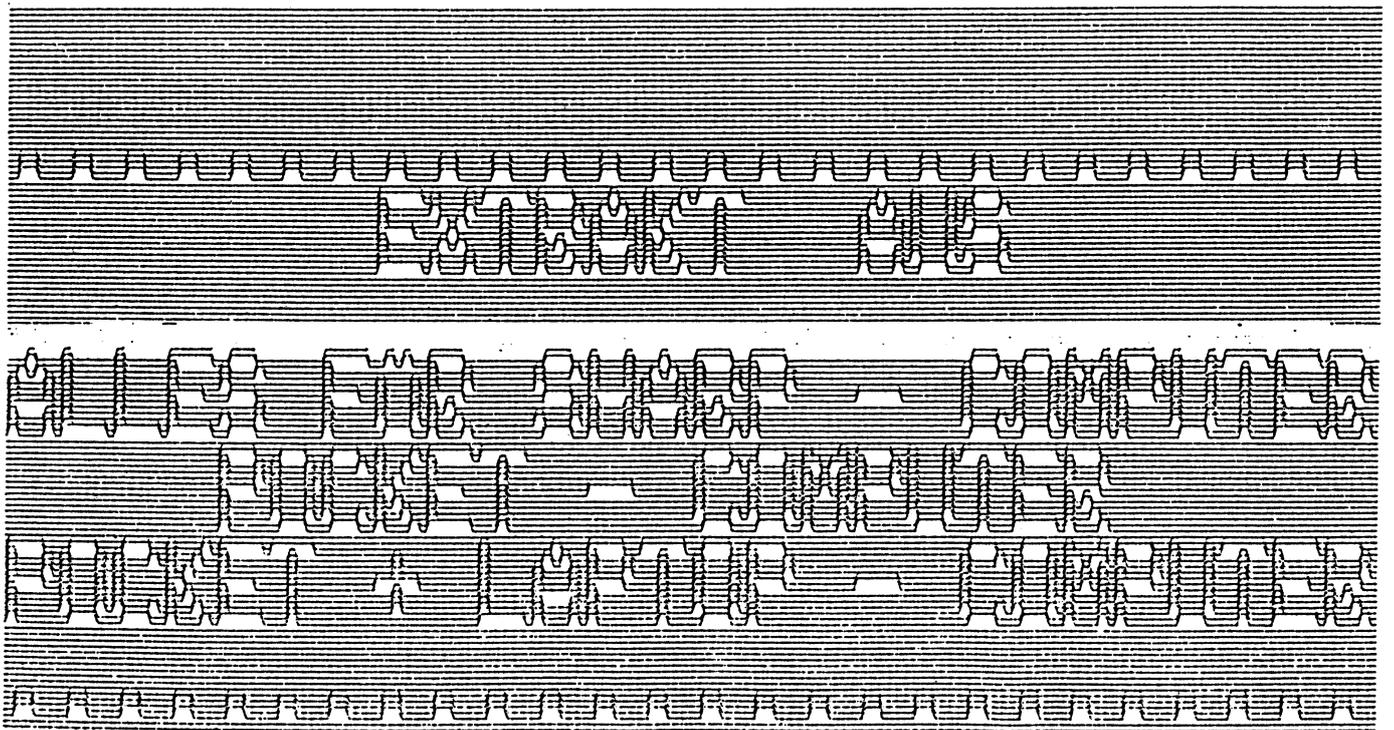
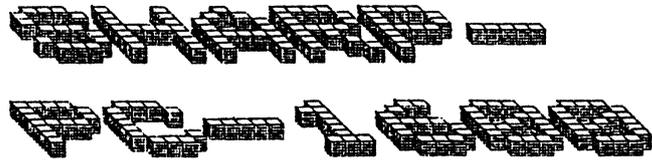
S	:Hangman	:	31/30	:	026
E	:Pegelberechnung	:	31/39	:	027
A	:Überweisungsformular	:	31/40	:	027
A	:Euroschecks	:	31/41	:	028
G	:Scanner	:	32/15	:	029
G	:Bilder auf Display (speich,Hardcopy.)	:	33/23	:	030
S	:Bayowaren-Invasion	:	34/19	:	033
U	:Diskman	:	34/21	:	034
S	:Solitaire	:	34/25	:	036
U	:Kartentrick	:	35/09	:	036
S	:Serielles Datengerät am PC-1600	:	35/08	:	037
S	:Lissajous-Figure	:	36/04	:	038
M	:Interrobotik II	:	36/05	:	038
U	:Hardcopy	:	36/26	:	042
G	:39 char/ 5 line	:	36/44	:	042
S	:Hangman	:	37/18	:	042
A	:Lotto-Tippscheinanalyse	:	37/19	:	043
M	:Zufallszahlenverteilung opt. sichtbar	:	38/06	:	046
U	:INKEY	:	38/10	:	047
U	:STATUS	:	38/10	:	047
U	:2 Programme In den PC-1600 laden	:	38/15	:	048
G	:OPTI-CHAR	:	38/46	:	048
T	:Einfache Textverarbeitung	:	38/47	:	050
U	:Monitor	:	38/48	:	050
T	:Briefeschreiben Vers.5	:	39/05	:	050
X	:Rechnerschutz	:	39/18	:	053
X	:Mode wechseln	:	39/28	:	053
X	:CE-158 Centronics	:	39/28	:	053
X	:Serielle Datenübertragung	:	39/28	:	054
X	:Keine Linput	:	39/28	:	054
G	:Zufallsbild-Grafik	:	39/33	:	054
T	:Elektronisches Taschenbuch	:	39/43	:	055
T	:Kurzbrief-Programm	:	40/26	:	056
U	:Ein Assembler ohne Assembler	:	42/09	:	057
U	:Autom.Reserve-Tasten-Belegung	:	42/17	:	058
U	:Editieren von Werten	:	44/06	:	059
M	:Tortendiagramm	:	44/06	:	059
M	:3D-Funktionsplot	:	44/09	:	060
U	:Ein Assembler ohne Assembler (Forts.)	:	46/07	:	063
M	:Kreisroutine	:	46/08	:	063
U	:Suche nach angesprochenen Zeilen	:	47/05	:	064
U	:Belegung der Funktionstasten	:	47/06	:	065
U	:DFÜ	:	48/35	:	066
A	:Zeitanzeige	:	49/21	:	070
T	:Textdatei	:	49/19	:	071
A	:Menütreiber	:	50/18	:	072
G	:Grafik-Demo	:	50/30	:	073
D	:Eingabe-Korrektur-Kontrolle	:	50/50	:	073
X	:SAVE-LOAD-Display	:	52/19	:	073
D	:Datenbaumprogramm	:	52/23	:	074
X	:Userfragen	:	52/13	:	076

X	:GCPRINT	:	53/30	:	076
A	:Weltzeituhr	:	53/39	:	077
A	:Rasterprogramm	:	54/40	:	077
D	:Anschriften-/Adress-Datei	:	54/41	:	078
M	:Funktionsdarstellung	:	55/15	:	078
A	:Cassettenhüllenbeschriftung	:	55/23	:	079
U	:Mapro-Utility	:	55/19	:	079
G	:Diagramme	:	55/21	:	080
G	:Zeichencopy	:	56/25	:	082
G	:CHR\$-Show	:	56/09	:	082
H	:Speichererweiterung	:	57/08	:	083
A	:Tauchsport-Programm	:	57/04	:	084
U	:Poker II	:	58/05	:	085
U	:Rechnerkopplung seriell mit IBM	:	58/26	:	086
X	:FOR NEXT Schleifen	:	58/44	:	087
I	:Ausdehnung von Körpern und Gasen	:	58/18	:	087
A	:Rasterprogramm	:	59/19	:	088
D	:Eingabe-Korrektur-Kontrolle	:	59/06	:	088
U	:Menü-Treiber	:	59/10	:	089
A	:QTH-Kenner	:	4-90/39	:	089
H	:CE-1600P-Akkus-leer?	:	4-90/28	:	090
A	:PC-1600 Kalender	:	5-90/36	:	091
U	:Mapro-Utility	:	6-90/22	:	092
M	:Math.Operationen auf d.Ebene v.symb.	:	6-90/40	:	093

					:franz.
					:La revue des
					:SHARPENTIERS +
					:NANP ORDINATEURS
Code	: Beitrag				: Nr.-M/J : Seite

	:(Titelseite)	:	27 -1/87	:	100
X	:Codes assembler	:	27 -1/87	:	101
X	:News	:	27 -1/87	:	101
X	:Le port d'entrée analogique	:	27 -1/87	:	102
	:(Titelseite)	:	1 -3/87	:	104
U	:Astuces pur pc-1600	:	1 -3/87	:	105
U	:Les premiers routines	:	1 -3/87	:	105
S	:Bombardier	:	1 -3/87	:	106
U	:Une ligne de 255 caracteres	:	1 -3/87	:	106
	:(Titelseite)	:	4 -9/87	:	107
M	:Traceur de courbes	:	4 -9/87	:	108
	:(Titelseite)	:	5-11/87	:	111
U	:Editeur-Ass-Dess	:	5-11/87	:	112
X	:Extensions pc-1600	:	5-11/87	:	113
U	:Trucs et astuces	:	5-11/87	:	114
	:(Titelseite)	:	6 -1/88	:	115
X	:Eprom tape	:	6 -1/88	:	116
S	:Serpent	:	6 -1/88	:	117
S	:Routines	:	6 -1/88	:	117
U	:Driver Clavier	:	6 -1/88	:	118
	:	:		:	:

		:japan.	
		:Pokecom	
		:Journal	
Code	: Beitrag	: Nr.-M/J	: Seite
	:(Titelseite)	: 4/90	: 100
U	:KFILES (Extended Directory List	: 4/90	: 101
	:	:	:



Der PC-1600 kann ja an dem alten Plotter CE-150 betrieben werden, aber nur wenn der PC-1600 im PC-1500-Modus (MODE 1) betrieben wird. Wird er im PC-1600-Modus (MODE 0) betrieben, dann kommt es bei der Druckerausgabe zu einer Fehlerausgabe, da die LPRINT-Befehle nur für den CE-1600p gelten. Es ist auch nicht möglich während des Programmablaufes von MODE 0 auf MODE 1 zu wechseln. Um die großen Möglichkeiten des 1600ers aber auch am CE 150 ausnutzen zu können, kann man die Umschaltung über einen Trick bewerkstelligen. Der 1600er bietet nämlich die Möglichkeit mittels des Befehles KBUFFs einen String in dem Tastaturspeicher abzulegen, der dann abgearbeitet wird, als ob man ihn über die Tastatur eingegeben hätte.

Am Beispiel sieht dies so aus:

```

10
20
30
40 KBUFFs-"MODE 1"+CHR$13+"GOTO 60"+CHR$13
50 END
60 LPRINT"Test"
70 KBUFFs-"MODE 0"+CHR$13+"GOTO 90"+CHR$13
80 END
90

```

```

10 ... 30      Programmablauf im 1600er Modus
40            Tastaturpuffer wird gefüllt CHR$13 ist das
              Zeichen für ENTER
50            Hier hält das Programm an, und arbeitet dann
              den Inhalt des Tastaturpuffers ab. In diesem
              Fall: MODE 1 ENTER GOTO 60 ENTER
60            Der Text wird gedruckt
70            Der Tastaturpuffer wird wieder gefüllt, um zu-
              rückzuschalten.
80            An dieser Stelle erfolgt die Rückschaltung
90            normaler Ablauf wieder im 1600er Modus

```

Da doch ein Mehraufwand entsteht, sollte man sich überlegen, ob die Druckerausgabe nicht so organisiert werden kann, daß nicht ständig zwischendurch Daten ausgedruckt werden, sondern man erst einmal sammelt, und dann ausgibt. Dies würde den Aufwand senken, da nicht ständig hin und her geschaltet werden muß.

Sven Nimsgarn
Finkenried 4d

2000 Norderstedt

Frank Dabringhausen

Der SHARP PC-1600 hat nun den alten PC 1500(A) abgelöst. Im nachfolgendem kleinem Beispielprogramm sehen Sie die Grafikfähigkeit des LCD-DISPLAYs. Ein Text wird in alle vier Richtungen nacheinander invertiert. Leider geschied dies etwas langsam. Vielleicht hat ein PC 1600-Anwender schon eine MAPRD-Routine geschrieben. Bitte an die Redaktion senden.

Listing INVERS-Total

```

10:"A" CLS:WAIT 0:CURSOR 1,0:PRINT "Alles für SHARP-Computer"
20:CURSOR 6,1:PRINT "Fischel GmbH"
30:CURSOR 1,2:PRINT "Kaiser-Friedrich-Str.54a"
40:CURSOR 5,3:PRINT "1000 Berlin 12"
50:WAIT 0
60:FOR I=0 TO 155
70:FOR J=0 TO 31
80:PSET(I,J),X
90:NEXT J:NEXT I
100:FOR I=31 TO 0 STEP-1
110:FOR J=0 TO 155
120:PSET(J,I),X
130:NEXT J:NEXT I
140:FOR I=155 TO 0 STEP-1
150:FOR J=31 TO 0 STEP-1
160:PSET(I,J),X
170:NEXT J:NEXT I
180:FOR I=0 TO 31
190:FOR J=155 TO 0 STEP-1
200:PSET(J,I),X
210:NEXT J:NEXT I
220:WAIT:PRINT
230:END

```

PC-1600-Besonderheit

Beim Abspeichern von Textvariablen auf Disketten mit PRINT darf in den Textvariablen kein Komma enthalten sein, da dieses Komma als Trennkomma zwischen Daten aufgefaßt wird. Vergleiche folgenden Programmausdruck: Vor dem Abspeichern steht in der Variablen A\$ der Text "AB,CD", nach dem Laden "AB".

```

10:A$="AB,CD"
20:PRINT A$
30:LPRINT A$
40:MAXFILES -1
50:OPEN "X:TEST.DAT"FOR OUTPUT AS #1
60:PRINT #1,A$
70:CLOSE #1
80:OPEN "X:TEST.DAT"FOR INPUT AS #1
90:A$="":INPUT #1,A$
100:PRINT A$
110:LPRINT A$
120:END

```

AB,CD
AB

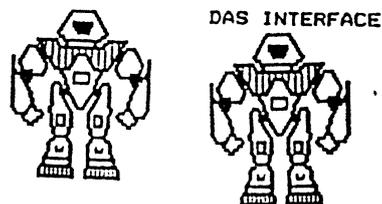
Lassen sich die beiden Sharp-Pocketcomputer FC 1500 (A) und FC 1600 als vollwertige Computer in handlicher Ausführung nicht auch als Steuerungszentrale von technischen Abläufen bzw. Modellen wie der Fischertechnik-Baukästen gebrauchen, wo doch die PCs so nützlich für viele Schreibtischaufgaben sind?

für
Fischertechnik,
Robotik

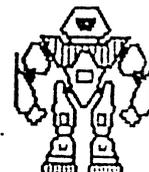
Kann denn nicht ein FC 1500/1600 auf dem Schreib- oder Demonstrationstisch auch noch eine Modellsteuerung zur Simulation technischer Funktionen und Vorgänge gleich selbst vornehmen, wenn er schon die vorhergehenden Berechnungen entsprechender Projekte übernommen hat?

und noch mehr.....!!!!!!!!!!!!!!

Diese Fragen drängen sich doch geradezu auf; und in der Tat läßt sich das oben genannte realisieren, wenn nur ein entsprechendes Interface vorhanden ist. Dieses muß allerdings die besonderen Gegebenheiten der beiden Pocketcomputer berücksichtigen: Die besonders kleine 60polige Steckverbindung, die am FC noch nicht vorhandenen Fortleitungen, welche die Tore zur Außenwelt bilden sollen, und die geforderte räumliche Beengtheit eines "Anbaus".



Jetzt gibt es dieses neue Interface. Es bringt die Möglichkeit, das vollständige Fischertechnik-Computing-System am kleinen PC zu betreiben. Die Konstruktionsbaukästen regen zum Selberbauen von verschiedenen Robotern, Automaten und Graphikgeräten an, wobei als Einsatzorte außer dem Hobbybereich auch der semiprofessionelle - insbesondere der schulische - Bereich zu sehen sind.



Bis zu 4 Motoren und/oder Elektromagneten, die über Leistungsstufen des Interfaces angesteuert und von dem normalen Fischertechnik-Netzteil mit Strom versorgt werden, verleihen dem FC "Arme und Beine". Die "Sinnesorgane" werden sozusagen durch 8 Taster und zwei Potentiometer nachgebildet. Sie geben die verschiedensten Rückmeldungen aus den Modellen an den Computer bzw. sein Programm. Die Tasten können aber auch zum Aufbau von Kommandofeldern für die abzuarbeitenden Programmenteile dienen.

Robotik

- 1.) steuern
- 2.) regeln
- 3.) messen
- 4.) vergleichen
- 5.) ausführen

Als Beispiel für die vielen Modellvorschläge der Fischertechnikbaukästen sei hier nur die Solarzellennachführanlage angesprochen. Eine Solarzelle wird mit Hilfe eines Programms frontal zu jeder Himmelsrichtung auf optimale Energieeinstrahlung hin ausgerichtet.

Wer hat Erfahrungen und kann berichten ???
Melden bei der
Fischel GmbH

Dank einer gegenüber den Home-Computer-Interfaces erweiterten und verbesserten analogen Meßwertaufnahme wird sich mit nur wenigen Zusätzen wie weiteren Solarzellen und eventuell einem Spannungswandler eine komplette Selbstnachladeeinrichtung für die Akkus des PCs realisieren lassen. Diese kann sich selbst dann auch noch überwachen und zeitbezogene Daten darüber speichern oder graphisch aufzeichnen.

Robotik

Uns interessieren alle Anwendungen im Roboterbereich. Funktioniert z.B. Ihre Heizungssteuerung mit einem SHARP-Taschen-Computer ?? Unbedingt bei der Fischel GmbH umgehend melden.

Hieran wird schon deutlich, daß mit den Fischertechnik-Modellen und erst recht durch die Erweiterungsmöglichkeiten des Interfaces, wie im folgenden aufgeführt, der Phantasie keine Grenzen gesetzt sind. Zum ersten werden außer den von den Fischertechnik-Modellen gebrauchten 8 Ein- und 8 Ausgangsleitungen noch 8 wahlweise als Ein- oder Ausgänge zu benutzende Leitungen geboten, die z.B. zum Anschluß von zusätzlichen digitalen Gebern wie Reed(Magnet)-Schaltern oder zur direkten Ansteuerung von Leuchtdioden oder Relais dienen können.

Dann sind nicht nur die zwei Analogmeßwertumwandler für die Potentiometer der Fischertechnikmodelle wie bei dem HomeComputer-Interfaces vorhanden, sondern ein 4-kanaliger hochgenauer 10 bit Analog-Digital-Wandler ist zusammen mit einer Präzisionsreferenzspannungsquelle aufgebaut. Dieser kann noch zusätzliche Meßwerte wie über Temperatur oder Lichtverhältnisse zur Auswertung liefern. Mit dem auch angebotenen Sensor ist dabei z.B. eine genaue Temperaturbestimmung in einem Bereich von 0 C - 100 C mit einer Genauigkeit von 0,25 C (sensorbedingt) möglich. Man sieht hier, daß das Interface mit einem guten Programm auch zu einer genauen universellen Meß- und Steuerstation werden kann.

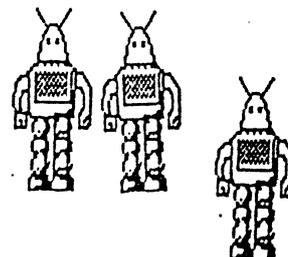
Die Netzunabhängigkeit der Pocketcomputer und dieses Interfaces sowie die Unabhängigkeit von einem Monitor machen die ganze Steuerung auf einem Tisch noch einfacher; UND DAS WIEDERUM NEUE daran: Der PC 1500/1600 wird auf einem Fahrzeug zusammen mit einem Akku SELEST FAHREN !!!!!. So können die SHARP PCs zu Gehirnen von Robotern werden. Ausführlich wird dieser für die Computersteuerung völlig neue Aspekt demnächst in einem Buch über SHARP PCs UND DIE ROBOTIK behandelt.

Und nun noch zusammengefaßt die technischen Daten:

1) Analogmeßwertaufnahme:

- A/D-Wandler mit 8 oder 10 bit Auflösung, was bei einer maximalen Eingangsspannung von 2,56 V 10mV pro Digit bzw. 2,3 mV pro Digit entspricht

- Meßrate maximal 300 Messungen pro Sekunde



- mit dem Temperatursensor ist im Bereich von 0 C bis 100 C eine Auflösung von 0,25 C möglich
- 4 Kanäle für 4 verschiedene Eingangsspannungen mit Eingangswiderständen von über 100 MOhm
- Referenzspannungsquelle mit 2,56V für bis zu 4 Potentiometer, wird hier z.B. auch für die Fischertechnikmodelle benutzt

2) Digitale Ein-,Ausgänge

- 8 Eingänge für Taster der Fischertechnikmodelle usw.
- 8 Ausgänge benutzt für die Motorsteuergruppen
- 8 frei als Ein-, Ausgänge, bzw. für Maschinenprogrammierer auch als Interrupteingänge belegbare Leitungen
- digitale Ausgänge maximal bis je 2 mA belastbar

3) Motorsteuerung

- 4 Gleichstrommotoren bzw. auch Schrittmotoren und/oder Magnetspulen für Links- Rechtsaufrichtung
- betreibbar mit Fischertechnik-Netzteil oder externem Akku bis je 350 mA (kurzzeitig bis 2 Ampere) pro Motor
- typischer Stromverbrauch der Fischertechnik-Minimotoren je ca. 90 mA (der Leistungstransistoren im Stillstand unter 1 mA)

4) Software

- alle Baugruppen über Basicbefehle (Poke../Peek..) ansprechbar, jedoch auch kurze Maschinenroutinen für die schnellere Analogmeßwertaufnahme vorhanden
- ausführliche Bedienungsanleitung mit mehreren Beispielprogrammen auch zur Temperaturmessung und mit sämtlichen Programmen der Fischertechnik-Modelle (Ergänzung zur Fischertechnik-Programmieranleitung)
- Sämtliche Programme auf C-Cassette

5) Erweiterungsmöglichkeiten

- Digital-Analog-Wandlermodul: gibt analoge Spannungen im Bereich von 0 V bis 2,5 V in Schritten von 10 mV mit 1 mA Belastung aus und ist geeignet:
 - als Funktionsgenerator auch zur Ton-/ Klangerzeugung
 - zur softwaregesteuerten Drehzahlregelung von Motoren usw.
- Sprachausgabemodul für Februar 87 vorgesehen

6) Anschlüsse, Gehäuse, und Sonstiges

- 20-poliger Stecker für die Fischertechnik-Buchsen-Verbindung
- 20-poliger Stecker für Erweiterungen über zwei Analogeingänge, 8 Portleitungen, Speisespannung, Referenzspannung und Masse
- Netzteil- oder Akkuanschluß für die Motorsteuerung
- Stromversorgung dank CMOS-Bauteilen mit nur ca. 10 mA vom PC aus
- alle ICs gesockelt
- Gehäuse beige/braun, mit seitlicher Herausführung der Stecker passend z.B. in den Aufbewahrungskasten des PC 1500/CE 150; Maße: 150mm * 60mm * 55mm (Höhe)

PREIS FÜR INTERROBOTIK inkl.14% MWST. :

Das Interface für Fischertechnik, Robotik und noch mehr...
komplett wie beschrieben **335 DM**

Digitalanalogwandlermodul mit Verbindungskabel, Stecker
sowie Funktionsgenerator-Programm **75 DM**

genauer Temperatursensor anschlussfertig mit 2 m abge-
schirmtem Kabel (dreifadrig, noch verlängerbar) **28 DM**

Ich besitze den PC-1600 und den Printer CE-1600P erst einen Tag. Obwohl ich vorerst nur wenige Vorteile der zahlreichen Möglichkeiten kennengelernt und erprobt habe, möchte ich auf Grund der Beiträge im Heft 21 sofort das Wort ergreifen. Die Kritik auf S. 6 f. ist m.E. größtenteils unberechtigt und teilweise unzutreffend. Das betrifft z.B. das angebliche Fehlen des ß im zweiten Zeichensatz.

Zwar ist dieses Sonderzeichen nicht über die KBII-Taste abrufbar. Aber wenn man eine RESERVE-Taste, mit dem Befehl "CHR\$&E;" belegt, ist das ß jederzeit mit der entsprechenden RESERVE-Taste abzurufen. Achtung: Die beiden Anführungszeichen (") gehören mit zur Belegung. Anschließend kann man wie gewohnt die nächsten Buchstaben eingeben.

Beispiel: Ich habe die X-Taste auf der Ebene I mit dem obigen Befehl belegt. Bei dem Wort „draußen“ betätige ich folgende Tasten:

d r a u ß e n

Pastor Peter Kriz
Paalende 37
2000 Hamburg 73

PC-1600-DISPLAY-Kopierung

Das große DISPLAY-Feld des PC-1600 (155x31-Punkt-Matrix) reizt dazu, DISPLAY-Grafik-Programme zu erstellen. Ich habe ein DISPLAY-Kopier-Programm erstellt, das ermöglicht, interessante DISPLAY-Grafiken jederzeit über den Printer des PC-1600 auf Papier zu kopieren.

Zur Demonstration habe ich den FISCHEL-Macker kopiert. Das Kopierprogramm selbst umfaßt die Programmzeilen 1-19.

Die Einzelheiten des Programmablaufs ergeben sich aus der Programmliste. Ich möchte nur einige Erläuterungen geben. Je größer der Unterschied zwischen dem Vergrößerungsfaktor V und dem Abbildungsmaßstab der DISPLAY-Punkte W ist, um so blasser wird die Kopie. Programmäßig ist der Faktor V=6 vorgegeben, weil bei diesem Faktor das ganze DISPLAY auf die Breitseite der Papierrolle bzw. eines DIN-A4-Blattes paßt. Wenn Sie die vorgegebenen Parameter nicht verändern möchten, braucht bei Anforderung der Parameterangaben nur die ENTER-Taste betätigt zu werden.

Nach Zeile 19 geben Sie Ihr Programm ein, dessen DISPLAY-Bild gegebenenfalls kopiert werden soll. Beachten Sie, daß nach Vollendung des DISPLAY-Bildes im Programm eine Zeile mit dem Befehl GOSUB "Abdruck" eingefügt werden muß. Kopierfähige DISPLAY-Programme müssen mit DEF "D" gestartet werden, damit das DISPLAY nicht gelöscht wird. Darum sind auch stets die Parameter-Eingaben am Anfang des Programms-Ablauf erforderlich.

Möchten Sie mehrere DISPLAYfelder eines Programms kopieren, die durch FOR-NEXT-Schleifen erzeugt werden können, ist die PROGRAMM-Zeile mit dem GOSUB "Abdruck" so zu ergänzen: GOSUB "Abdruck";GOSUB "Neues DISPLAY";CLS:GOSUB "Abdruck". Dann wird, bevor ein NEXT I ein neues DISPLAY aufbaut, das Papier um eine DISPLAY-Kopie-Breite transportiert und das alte DISPLAY gelöscht.

Möchten Sie COMPUTER-SPIEL-DISPLAYs, die sich laufend verändern, als Momentaufnahme kopieren,

D I S P L A Y - K O P I E R U N G

```

1:"D"CLEAR :PRINT "DISPLAY-Kopierung"
2:REM ***by Peter Kriz Hamburg***
3:REM ***Parametervorgaben***
4:CX=0:INPUT "GLCURSOR X ";CX
5:CY=0:INPUT "GLCURSOR Y ";CY
6:C=0:INPUT "COLOR";C
7:U=6:INPUT "Ueagrößerungs-Faktor ";U
8:W=U:PRINT "Punktgröße <=";W; " ";:INPUT W
9:IF W<1LET W=1
10:GOTO "Programm"
11:"Abdruck"GRAPH :COLOR C
12:GLCURSOR (CX,CY):SORGN
13:FOR JX=0TO 155
14:FOR JY=0TO 31
15:P=POINT (JX,JY)
16:IF PGLCURSOR (U*JX,-U*JY):RLINE -(W,0)-(0,W)
   -(-W,0)-(0,-W)-(W,W)
17:NEXT JY:NEXT JX:LLINE -(0,0),9:TEXT :RETURN
18:"Neues DISPLAY"GLCURSOR (CX,CY-31*U):SORGN :
RETURN
    
```

Auf dem DISPLAY (und im PRO-MODE in der entsprechenden Programmzeile) steht dann:

drau";CHR\$&E;"en

Entsprechend kann mit allen Sonderzeichen verfahren werden, die nicht über die KBII-Taste abrufbar aber gemäß Code-Tabelle der Bedienungsanleitung (A-4) im PC-1600-ROM vorhanden sind.

Als besonders angenehm finde ich das geräuscharme Arbeiten des Printers CE-1500P. Er arbeitet leiser als meine doppelt so teure elektronische Schreibmaschine, die schallgedämpft ist.

Als weitere Vorteile, die ich schon nach einem Tag nicht mehr missen möchte, empfinde ich die automatische Zeilennummerierung und die Tatsache, daß man jeweils eine ganze Druckzeile im DISPLAY hat.

Ich hoffe, mich nach weiterem Kennenlernen dieses großartigen Computers mit ähnlichen Empfehlungen wie die Benutzung des ß wieder melden zu können.

DISPLAY-Kopierung

U = 6 - W = 6

ist statt der „GOSUB "Abdruck"-Zeile" im Spielprogramm vor der entsprechenden NEXT-I-Zeile, die neue "Bewegungen" auf dem DISPLAY erzeugt, eine Programmzeile einzufügen, die lautet: IF ASC INKEY\$ = 13 THEN GOSUB "Abdruck". Dann können Sie durch Betätigung der ENTER-Taste Ihr Spiel in spannenden Momenten unterbrechen und das DISPLAY kopieren. Haben Sie bitte Geduld, wenn das Kopierprogramm beginnt. Da das DISPLAY vertikal Punkt für Punkt abgetastet wird, kann es einige Zeit dauern, bis der erste Punkt der Grafik erreicht und kopiert wird - besonders wenn die Grafik nur auf der rechten DISPLAY-Hälfte sein sollte.

```

19:"Programm":WAIT 0:CLS
20:PRINT " Alles für"
21:PRINT " SHARP-COMPUTER"
22:PRINT " FISCHEL hat's!"
23:PRINT "DISPLAY-Kopierung"
24:FOR I=1TO 192
25:READ X,Y
26:PSET (X,Y)
27:NEXT I
28:DATA 110,31,111,30,112,30,113,20,113,21,113,
   29,114,18,114,19,114,22,114,29
29:DATA 115,8,115,9,115,10,115,11,115,12,115,13
   ,115,14,115,15,115,16,115,17
30:DATA 115,22,115,20,116,5,116,6,116,7,116,10,
   116,13,116,14,116,16
31:DATA 116,22,116,23,116,24,116,27,117,3,117,4
   ,117,5,117,11,117,12,117,13
32:DATA 117,14,117,15,117,22,117,23,117,25,117,
   27,118,2,118,3,118,4,118,5
33:DATA 118,13,118,21,118,24,118,26,118,27,119,
   2,119,3,119,4,119,6,119,10
34:DATA 119,11,119,12,119,13,119,14,119,15,119,
   18,119,19,119,20,119,24,119,26
35:DATA 119,27,119,28,120,1,120,2,120,3,120,5,1
   20,10,120,13,120,14,120,16
36:DATA 120,23,120,26,120,28,120,29,121,1,121,2
   ,121,3,121,4,121,5,121,10
37:DATA 121,13,121,14,121,17,121,22,121,28,121,
   30,121,31,122,1,122,2,122,4
38:DATA 122,11,122,17,122,27,123,1,123,2,123,3,
   123,4,123,5,123,6,123,12
39:DATA 123,13,123,14,123,15,123,16,123,30,123,
   31,124,1,124,2,124,3
40:DATA 124,4,124,5,124,7,124,8,124,9,124,12,12
   4,28,124,29,125,3,125,4
41:DATA 125,5,125,6,125,7,125,8,125,9,125,10,12
   5,11,125,12,125,26,125,27
42:DATA 126,4,126,5,126,6,126,7,126,8,126,9,126
   ,10,126,11,126,12,126,13
43:DATA 126,18,126,24,126,25,127,6,127,7,127,8,
   127,9,127,10,127,11
44:DATA 127,12,127,13,127,14,127,15,127,16,127,
   17,127,21,127,22,127,23,128,14
45:DATA 128,15,128,16,128,17,128,18,128,19,128,
   20,129,21,129,22,130,23,131,23
46:DATA 132,24,133,24,134,25,135,25,136,26,137,
   26,137,27,138,27,138,29,138,30,138,31
47:DATA 138,27,138,30,139,29,139,30,139,31
48:GOSUB "Abdruck" :ND
    
```

Torbjörn Hovden
Konvallveien 23
N-3900 PORSGRUNN
NORWEGEN

Sehr geehrte Damen und Herren,

Ich bin verliebt! Liebe auf den ersten Blick. Sie heisst Sharp PC-1600, und wurde waehrend meines Urlaubs in Deutschland gekauft. Ich habe mich auch in Berlin verliebt, und fuer den PC-1600 habe ich beifuegte Programm entwickelt. Es handelt sich um eine Sharp Version von der bekannten Berlineruhr. Die Uhr laest sich einfach ablesen: Die Anzahl von schwarzen Flaechen in der ersten Zeile multipliziert mit fuef und mit der Anzahl der Flaechen in der zweiten Zeile addiert, gibt die Stunden. Gleicherweise kann man aus den Zeilen drei und vier die Minuten bekommen. Das Programm wird mittels RUN <ENTER> gestartet.

Aber nicht alles ist gut mit meinen PC-1600. Wenn ich PC-1600 Programme ueber den CE-158 Parallellport ausliste, kennt der CE-158 PC-1600 Befehle wie ALARMS und WAKES nicht, und meine Brother HR-15 schreibt statt diese nur . Gibt es eine Moeglichkeit dieses zu vermeiden?

Das letzte Problem habe ich mit der RAM-DISC. Wenn ich eine Text-Datei zur RAM-DISC schreibe, und sich in dieser Datei Zeilen mit Komma befinden, kann ich diese Zeilen nicht ungeteilt lesen, weil Komma als Daten-Separator fungiert. Ich moechte also ein LINPUT Befehl haben. Hat jemand eine Loesung zu meinen Problemen?

5. "GROSSE UEBERASCHUNG
10 H0=15/16:C=1:D=9/8:E=5/4:F=4/3:KF=25/18:G=3/2:A=5/3:H=15/8:C1=2:D1=9/4:E1=5/2

20 F1=8/3:K1=25/9:G1=3:A1=10/3:H1=15/4:C2=4:D2=9/2
30 CLS
40 LINE (0,0)-(24,20),,,
BF

50 FOR I=2TO 18STEP 4
60 FOR J=2TO 22STEP 4
70 PRESET(J,I)
80 NEXT J
90 NEXT I
100 FOR I=4TO 16STEP 4
110 FOR J=4TO 20STEP 4
120 PRESET(J,I)
130 NEXT J
140 NEXT I
150 FOR I=0TO 30STEP 5
160 X=0
170 IF I<20LET X=25
180 LINE (X,I)-(55,I+1),,
BF

190 NEXT I
200 RESTORE 580
210 GOSUB 540
220 LINE (58,0)-(101,31),,
BF

230 LINE (58,12)-(101,19)
R,,BF

240 LINE (70,0)-(77,31),R
BF

250 LINE (58,14)-(101,17)
BF

260 LINE (72,0)-(75,31),,
BF

270 RESTORE 690
280 GOSUB 540

290 LINE (104,0)-(155,31),,,BF

300 LINE (104,-4)-(155,29),R

310 LINE (104,-3)-(155,30),R

320 LINE (104,29)-(155,-4),R

330 LINE (104,30)-(155,-3),R

340 LINE (104,2)-(155,35),R

350 LINE (104,1)-(155,34),R

360 LINE (104,35)-(155,2),R

370 LINE (104,34)-(155,1),R

380 LINE (104,11)-(155,20),R,,BF

390 LINE (124,0)-(135,31),R,,BF
400 LINE (104,13)-(155,18),,,BF
410 LINE (126,0)-(133,31),,,BF
420 RESTORE 750
430 GOSUB 540

440 LINE (0,0)-(55,31),X,,BF:LINE (58,31)-(101,0),X,,BF:LINE (104,0)-(155,31),X,,BF

450 LINE (0,0)-(55,31),X,,BF:LINE (58,31)-(101,0),X,,BF:LINE (104,0)-(155,31),X,,BF

460 FOR I=0TO 1000:NEXT I
470 FOR I=15TO 0STEP -1
480 LINE (I,I)-(155-I,31-I),R,,B
490 NEXT I

500 FOR I=0TO 15
510 LINE (I,I)-(155-I,31-I),,,B
520 NEXT I
530 GOTO 30

540 READ X,Y
550 IF X=9RETURN
560 BEEP 1,229/X+.5,X*Y*30+.5
570 GOTO 540

580 DATA G,3,E,1,C,4,E,4,G,4,C1,8,
E1,2,D1,2,C1,4,E,4,XF,4
590 DATA G,8,G,2,G,2,E1,6,D1,2,C1,
4,H,8,A,2,H,2,C1,4,C1,4,G,4
600 DATA E,4,C,4

610 DATA G,3,E,1,C,4,E,4,G,4,C1,8,
E1,2,D1,2,C1,4,E,4,XF,4
620 DATA G,8,G,2,G,2,E1,6,D1,2,C1,
4,H,8,A,2,H,2,C1,4,C1,4,G,4
630 DATA E,4,C,4,E1,2,E1,2,E1,4,F1,
4,G1,4,G1,8,F1,2,E1,2

640 DATA D1,4,E1,4,F1,4,F1,8,F1,4,
E1,6,D1,2,C1,4,H,8,A,2,H,2
650 DATA C1,4,E,4,KF,4,G,8,G,4,C1,
4,C1,4,C1,2,H,2,A,4,A,4,A,4

10 REM BERLINERUHR TORBJØRN HOVDEN
20 CLS
30 H=VAL LEFT\$(TIMES,2)
40 M=VAL MID\$(TIMES,4,2)
50 IF H=0AND M=0LET H=24
60 A=H\5
70 B=HMOD 5
80 C=M\5
90 D=MMOD 5
100 I=1
110 IF I>ATHEN 140
120 LINE ((I-1)*22,0)-(I*22-2,6),,,BF
130 I=I+1:GOTO 110
140 LINE ((I-1)*22,0)-(88,6),R,,BF
150 I=1
160 IF I>BTHEN 190
170 LINE ((I-1)*22,8)-(I*22-2,14),,,BF
180 I=I+1:GOTO 160
190 LINE ((I-1)*22,8)-(88,14),R,,BF
200 I=1
210 IF I>CTHEN 260
220 P=&FFFF
230 IF INT(I/3)=I/3LET P=&AAAA
240 LINE ((I-1)*8,16)-(I*8-2,22),,P,BF
250 I=I+1:GOTO 210
260 LINE ((I-1)*8,16)-(88,22),R,,BF
270 I=1
280 IF I>DTHEN 310
290 LINE ((I-1)*22,24)-(I*22-2,30),,,BF
300 I=I+1:GOTO 280
310 LINE ((I-1)*22,24)-(88,30),R,,BF
320 GOTO 30

660 DATA D1,4,F1,2,E1
,2,D1,2,C1,2,C1,4,H,4,G,2,G
,2,C1,6,DT,2,E1,2,F1,2
670 DATA G1,8,C1,2,D1,2,E
1,6,F1,2,D1,4,C1,8

680 DATA 9,9
690 DATA G,6,F,2,E,4,D,4,
C,4,D,4,E,4,F,4,G,6,A,2,G,4
,F,4,E,12

700 DATA A,6,G,2,F,4,E,4,
D,4,E,4,F,4,G,4,G,6,A,2,A,4
,H,4,C1,12

710 DATA C1,6,C1,2,H,3,H,
1,A,3,A,1,G,8,E,8,A,6,A,2,G
,4,G,4,F,8,G,3,G,1

720 DATA G,4,A,4,A,4,H,4,
H,8,C1,8,C1,6,C1,2,H,4,C1,4
,D1,8,D1,3,D1,1

730 DATA D1,4,E1,4,F1,4,E
1,4,D1,8,C1,4,H,3,A,1,G,7,A
,1,A,4,H,4,C1,12

740 DATA 9,9
750 DATA C,4,C,4,D,4,H0,6
,C,2,D,4,E,4,E,4,F,4
760 DATA E,6,D,2,C,4,D,4,
C,4,H0,4,C,8

770 DATA G,4,G,4,G,4,G,6,
F,2,E,4,F,4,F,4,F,4,F,6,E,2
,D,4

780 DATA E,4,F,2,E,2,D,2,
C,2,E,6,F,2,G,4,A,2,F,2,E,4
,D,4,C,8

790 DATA G,4,G,4,G,4,G,6,
F,2,E,4,F,4,F,4,F,4,F,6,E,2
,D,4
800 DATA E,4,F,2,E,2,D,2,
C,2,E,6,F,2,G,4,A,2,F,2,E,4
,D,4,C,8
810 DATA 9,9

** spielend lernt man leichter **

Seit nunmehr fast zwei Jahren bin ich begeisterter Leser Ihrer Zeitschrift >> Alles fuer Sharp-Computer <<. Meine ersten Computer-Schritte habe ich mit dem PC-1500 A gemacht. Inzwischen besitze ich den PC-1600.

Was mir nun an Ihrer Zeitschrift aufgefallen ist, Sie haben bisher praktisch keine Beitrage fuer die naechste Anwender-Generation, unsere Kinder, veroeffentlicht (Spiele einmal ausgenommen). Als Vater von zwei Soehnen habe ich mir ueberlegt, was man daeuesen tun kann.

Da mein Grosser im naechsten Jahr eingeschult wird, und dann mit dem 1x1 in Beruehrung kommt, habe ich fuer ihn das kleine Programm geschrieben, das ich Ihnen, falls es Ihnen zusaegt, zur Veroeffentlichung zur Verfuegung stelle. Es ist fuer den PC-1600 geschrieben und wird mit RUN gestartet.

Das Programm dient zum Ueben der beiden ersten Grundrechenarten Addieren und Subtrahieren. Die Aufgabe, die es zu loesen gilt, wird auf der ersten Display Zeile angezeigt. Der Spieler hat einen Bonus von sechs Luftballons. Wird die Rechenaufgabe durch eintippen des richtigen Ergebnisses geloes, so "lacht" der Computer. Wird die Aufgabe falsch beantwortet

ist er "traurig" und ein kleines Maennchen bringt einen Luftballon zum Platzen. Wenn alle Luftballons geplatzt sind, ist das Programm zuende. Mit RUN kann es erneut gestartet werden.

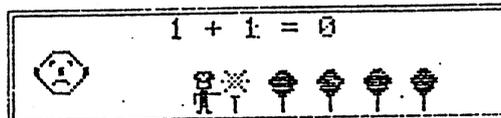
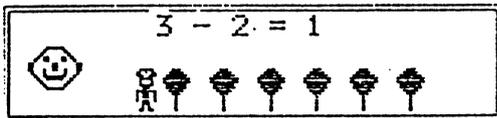
Es werden vom Programm nur Aufgaben gestellt, bei denen das Ergebnis < 10 ist. Das Ergebnis "0" habe ich vorerst nicht vorgesehen. Durch Aenderungen in den Zeilen 900 und 910 laesst sich das Programm aber auf schwierigere Aufgaben erweitern.

Ich kann Ihnen versichern, dass das Rechnen meinem Sohn grossen Spass macht und es wuerde mich freuen, wenn auch andere Schulanfaenger, nachdem ihre "alten Herren" das Programm eingetippt haben, Spass daran finden.

```

890:"RECHENAUFGABE"
900:X=INT (RND 5)
910:Y=INT (RND 4)
920:C=INT (RND 2)
930:IF C=1LET N$="+":GOTO 950
940:IF C=2LET N$="-":GOTO 970
950:A=X+Y
960:GOTO 1000
970:IF X<YLET X=Y:Y=X
980:IF X=YLET X=X+1
990:A=X-Y
1000:RETURN

```



```

10:REM *****
20:REM ** Rechenaufgaben by **
30:REM ** Hans Hölzel **
40:REM ** 6101 Fränk.-Crumbach **
50:REM ** Jahnstr. 9 **
60:REM *****
70:CLS :CLEAR :WAIT 0
80:PRINT :WAIT 80:PRINT " Rechenaufgaben":CLS
90:DIM A$(19)*20,B$(4)*36
100:A$(0)="187C7E7F7F7F7E7C18"
110:A$(1)="000001037F0301"
120:A$(2)="00000014081400000000"
130:A$(3)="000000142A1408000000"
140:A$(4)="0B412A142A142A4108"
150:A$(5)="000000000000000000"
160:A$(6)="063F6D6B6D3F06"
170:A$(7)="0F417F1F7F410F"
180:A$(8)="0F417F1F7F410E"
190:A$(9)="0F417F1F7F41010204"
200:A$(10)="0F417F1F7F41010101"
210:A$(11)="0F417F1F7F41010204"
220:A$(12)="0F417F1F7F410E0000"
230:A$(13)="0F417F1F7F410F"
240:A$(15)="7F7F7F7F"
250:A$(16)="005F5F7F755702"
260:A$(17)="001F7F581B7F42"
270:A$(18)="007F581B7F5F0202"
280:B$(0)="60601008041A19016101191A0408106060"
290:B$(1)="0103040810274C4C404D4C4C271008040301"
300:B$(2)="010304081020484445454448201008040301"
310:B$(3)="010304081024444445454444241008040301"
320:B$(4)="60601008041A09016101091A0408106060"
330:WAIT 0:GDCURSOR 5,10:GPRINT B$(0):GDCURSOR 5,17:GPRINT B$(3)
340:FOR I=50TO 125STEP 15
350:GDCURSOR I,17:GPRINT A$(0)
360:GDCURSOR I,24:GPRINT A$(1)
370:NEXT I
380:B=35
390:B=B+15
400:IF B=140BEEP 3:BEEP 1,200,250:CLS :PRINT :PRINT " * * * * E N D E * * * * "
":LND
410:REM ** GRAPHIK **
420:FOR I=B-23TO B-10

```

Seit einiger Zeit bin auch ich stolzer Besitzer eines SHARP PC-1600. Dieses neue Schlachtschiff an Computer-Markt hat meine Erwartungen vollauf erfüllt. Insbesondere durch den großzügig angelegten BASIC-Befehlssatz wird manch "größerer" Computer in den Schatten gestellt. Aber auch die komfortablen Editiermöglichkeiten und Erweiterungsanschlüsse sind durchaus nennenswert. Neue Produkte vergleicht man in der Regel immer mit seinen Vorgängern. In diesem Falle ist es der PC-1500, der dem PC-1600 nicht nur äußerlich ähnelt. Das neue Modell weißt gegenüber dem bewährten alten Modell jedoch eine Vielzahl von Verbesserungen auf, sodaß der Vergleich klar zugunsten des Neuen, nicht nur wegen der oben genannten Vorteile, ausfällt. Einen kleinen Schönheitsfehler hat der überaus gelungene PC-1600 dennoch: Die Anzeige des PC-1600 ist zwar in Bezug auf die Programmierung von Graphiken schon wegen ihrer Größe, aber auch wegen des Befehlsvorrates wesentlich geeigneter als die des PC-1500, jedoch lassen sich Zeichenketten nicht an jede beliebige Stelle des Displays setzen. Beim PC-1500 konnte eine Zeichenkette an jede beliebige Stelle der Anzeige gesetzt werden:



Gibt man diese Programmzeile in den PC-1600 ein, wird der String "SHARP" an die erste Displaystelle gesetzt, nicht aber bei Position 10. Den Befehl GPCURSOR kann man nur in Verbindung mit dem Befehl GPRINT vollwertig verwenden.



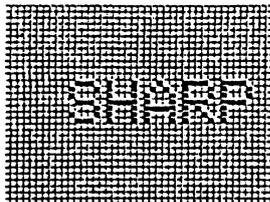
Beispiel: 10:GPCURSOR 10:GPRINT "SHARP"
Der PC-1600 besitzt zur Addressierung eines Strings lediglich den Befehl CURSOR. Mit diesem Befehl kann man jedoch nicht jede (Graphik-) Stelle in Display erreichen, sondern nur die Zeichen- bzw. Cursorpositionen.

Um aber auch eine Zeichenkette an jeder beliebigen Stelle des Displays adressieren zu können, muß man sich folgender Routine bedienen. Nachteilig ist hierbei allerdings die Verarbeitungsgeschwindigkeit.

```

Dominik A. Mueller
Hoehnerkamp 30
2000 Hamburg 56
Eingabebeispiel:
Text: SHARP
Koordinaten X: 10
Y: 11
Diese Eingaben verursachen folgenden Display-
inhalt (Ausschnitt):
1000:REM HAUPTPROGRAMM
1001:WAIT 0:CLS
1002:INPUT "Text: ";T$
1003:CURSOR 0,1
1004:INPUT "Koordinaten X: ";X
1005:CURSOR 12,2
1006:INPUT "Y: ";Y
1007:GOSUB "Z"
1008:END
2000:"Z"CLS:FOR I=1TO LEN T$
2001:A=ASC (MID$(T$,I,1))
2002:Z=0:P=(H-65)*6+44713
2003:GPCURSOR (I*6+X+Z-6),Y
2004:GPRINT PEEK P
2005:P=P+1
2006:Z=Z+1
2007:IF Z>56GOTO 2009
2008:GOTO 2003
2009:NEXT I
2010:RETURN

```



Nach Abänderung folgender drei Zeilen ist das Programm auch für die Darstellung von Fettschrift verwendbar.

```

1004:INPUT "Koordinaten X: ";X:X=X/2
2003:GPCURSOR (2*(I*6+X+Z-6)),Y
2004:GPRINT PEEK P;PEEK P

```



Peter Gargulak, Markwaldstraße 45, 6050 Offenbach/Main PC-1600

Mit diesem Programm ist man in der Lage Sinusfunktionen auf dem Display des PC-1600 darzustellen, sie addieren oder multiplizieren. STATUS 1 beträgt 2500 Bytes. Gestartet wird das Programm mit DEF A. Es erscheint ein kleines Menü.

Mit dem Menüpunkt 1 kommt man zur Funktionseingabe. Es werden insgesamt drei Funktionen abgefragt. Bei jeder Funktion wird die Amplitude, die Frequenz und die Phasenlage abgefragt.

Die Amplitude kann beliebig groß sein, sie wird immer an das Display angepaßt. Bei der Frequenz sieht es anders aus: Wird eine Frequenz größer 10 eingegeben so wird nach der Abfrage aller Funktionen eine Frage gestellt, ob die Frequenz angepaßt werden soll. Ein J wird schon vorgegeben, man muß also falls man einverstanden ist nur ENTER drücken.

Die Frequenzen werden dann so gekürzt, daß die größte Frequenz 10 Hertz beträgt.

Bei folgenden Eingaben Ampl.1 = 1 Hertz, Ampl.2 = 25 Hertz und Ampl.3 = 100 Hertz entsteht nach der Kürzung folgendes: Ampl.1 = 0.1 Hertz, Ampl.2 = 2.5 Hertz und Ampl.3 = 10 Hertz. Dies hat auch seine Nachteile wie man sieht.

Mit dem Menüpunkt 2 kann man die einzelnen Funktionen zeichnen lassen: es wird die Funktionsnummer abgefragt (1-3), gibt man aber 0 ein, so werden alle Funktionen gezeichnet.

Mit dem Menüpunkt 3 kann man die einzelnen Funktionen addieren und multiplizieren.

Man hat die Wahl ob man alle drei oder nur zwei Funktionen addieren bzw. multiplizieren will.

Wenn man nur zwei Funktionen berechnen will, wird außerdem noch abgefragt ob die Funktionen an das Display angepaßt werden sollen. Wird nur ENTER gedrückt so wird die Funktion an das Display angepaßt, wird jedoch ein N eingegeben, so wird die Funktion im richtigem Verhältnis zur der Funktion ausgegeben, die aus allen drei Funktionen zusammengerechnet worden ist.

Mit dem Menüpunkt 4 kann schließlich das Display gelöscht werden. Drückt man im Hauptmenü die Taste OFF, so wird der Computer abgeschaltet.

Ein Hinweis zum Abtippen: das Programm muß natürlich groß eingegeben werden, nur Textstellen die in Anführungszeichen stehen sollte man so eingeben wie sie im Listing sind.

Zum Schluß noch zwei Fragen: 1 wie schnell kann man den A/D - Wandler in 280 - Maschinensprache abfragen?

2 wer bietet einen A/D und D/A - Wandler an, der in einer Sekunde ca. 5000 Werte einlesen bzw. ausgeben kann?

```

10 *PLOT*b=0:b=b-f(i)+v(i):for x=0to 90:b=b+f(i):line -(x,16-sin b
+a(i)*2):next x
20 *PICK*line (0,0)-(0,31):line (90,0)-(90,31):line (90,16)-(0,16)
:return
30 *CLPC*cursor 16,0:print "
40 cursor 16,1:print "
50 cursor 16,2:print "
60 cursor 16,3:print "
70 *INP*clear idim a(3),f(3),v(3)
80 for i=1to 3:gosub "CLPC"
90 cursor 16,0:print "f Nr.1":str$ i
100 kbuff$="" :cursor 16,1:input "Ampl.1":ja(i)
110 kbuff$="" :cursor 16,2:input "Freq.1":f(i)
120 if f(i)>clet c=f(i)
130 kbuff$="" :cursor 16,3:input "Pha.1":v(i)
140 if ma(a(i))let ma=a(i)
150 next i
160 if c<=10goto 210
170 gosub "CLPC":cursor 16,0:print "Frequenz":cursor 16,1:print "a
npassen?"
180 kbuff$="" :cursor 16,2:input "(J/n)":ja$
190 if ja$<>"J"and ja$<>"N"goto 180
200 if ja$="J"let xx=c/10:for i=1to 3:f(i)=f(i)/xx:next i
210 for i=1to 3:a(i)=a(i)/ma*7.111111:f(i)=f(i)*4:next i:return
220 *D3F*kbuff$=""
230 cursor 16,1:input "2o.3f1":ja$
240 if ja$="3"return
250 if ja$<>"2"goto 230
260 kbuff$="" :cursor 16,2:input "f1":ja
270 if a<lor a>3goto 260
280 kbuff$="" :cursor 16,3:input "f2":jb
290 if b<lor b>3goto 280
300 gosub "CLPC":cursor 16,0:print "Wert an":cursor 16,1:print "Di
spl. an."
310 kbuff$="" :cursor 16,2:input "(J/n)":d$
320 if d$<>"J"and d$<>"N"goto 310else return
330 *INKE*as=inkey$:if as=""goto 330else return
340 *ADD2*p=(a(1)+a(2)+a(3))/14.0625:if d$="J"let p=(a(a)+a(b))/14
.0625
350 a1=0:a1=a1-f(a)+v(a):b1=0:b1=b1-f(b)+v(b)
360 for x=0to 90:a1=a1+f(a):b1=b1+f(b)
370 line -(x,16-(sin a1a(a)+sin b1a(b))/p):next x:return

```

Mit diesem Programm ist man in der Lage aus dem PC-1600 ein kleines Meßsystem (über den eingebauten A/D-Wandler) aufzubauen. Mit diesem Programm kann der PC-1600 so programmiert werden, daß er sich bei einer bestimmten Uhrzeit selbständig einschaltet, einen Wert über den A/D-Port einliest, den nächsten WAKE - Befehl ausrechnet und sich wieder abschaltet. Die Ausgabe der Werte erfolgt über das Display, deswegen sind nur max. 155 Werte zugelassen (leicht zu ändern).

Peter Gargulak
Markwaldstr.45
6050 Offenbach

Funktionsbeschreibung:

Gestartet wird das Programm immer nur mit DEF-A, sonst besteht Datenverlust.

Nach DEF-A erscheint ein kleines Menü:

Mit der Taste '1' kommt man in die Dateneingabe: es wird die Uhrzeit und das Datum für die erste Messung abgefragt. Danach wird die Anzahl der Daten und der Interrupt abgefragt.

Der Interrupt wird in Minuten angegeben (0-60 min.), wird jedoch eine '0' eingegeben so wird der Interrupt alle 24 Stunden ausgeführt.

Mit der Taste '2' werden schließlich alle Eingaben die unter '1' eingegeben worden sind noch einmal angezeigt.

Mit der Taste '0' läßt sich nun der Computer im Hauptmenü ausschalten.

Man kann die Daten auch "von Hand" in den Computer reinkladen.

Diese Möglichkeit besteht mit Punkt '4' im Hauptmenü:

Zuerst wird der Triggerwert abgefragt, dann die Abtastrate in Sekunden: das bedeutet -- wird zum Beispiel eine 4 eingegeben, so wird alle 4 Sekunden ein Messwert eingelesen. Wird eine '0' eingegeben, so beträgt die Abtastrate ca. 50 Messungen in der Sekunde. Mehr schafft der PC-1600 in Basic nicht.

Danach kann die Messung mit SPACE gestartet werden.

Wenn jedoch nichts geschieht, so liegt es wahrscheinlich am zu großem Triggerwert bzw. zu kleinem Eingangssignal. Deswegen kann auch mit der Taste F1 die Messung gestartet werden.

Die Taste liegt im Interrupt des PC-1600, belastet ihn daher (zeitlich) kaum.

Man kann die Daten jederzeit mit Menüpunkt '3' auswerten.

Die gemessenen Daten werden auf die Displaygröße angepaßt und gezeichnet.

Sind weniger als 32 Werte bei der Anzahl der Daten eingegeben worden, so werden die Messwerte waagrecht gezeichnet, wodurch man eine höhere Auflösung erzielt. Die Daten werden solange angezeigt, bis eine beliebige Taste gedrückt wird.

Es erscheint dann gleich der Mittelwert, der min., max.-Wert und die Differenz zwischen min und max.

Als nächstes kann man sich den "Wechselspannungsanteil" und/oder die einzelnen Daten anzeigen lassen.

```

1 GOTO "RO1"
2 GOTO "RO2"
10 "INKE" IF INKEY$="" GOTO "INKE" ELSE RETURN
20 "POF" WAKES (0)="" : POWER OFF
30 "RO1" Y=Y+1: M(Y)=AIN
40 MI=MI+IN: IF MI>59 LET MI=MI-60
50 MIS=STR$ MI: IF MI<10 LET MIS="" + MIS
60 IF Y=ANGOTO "POF"
70 WAKES (0)="" ??/?/?/?/" + MIS + " : GOTO1" + CHR$ 13
80 POWER OFF
90 "RO2" Y=Y+1: M(Y)=AIN
100 WAKES (0)="" ??/?/?/" + ST$ + " / " + MI$ + " : GOTO2" + CHR$ 13
110 IF Y=ANGOTO "POF"
120 POWER OFF
130 "A" CLS : KBUFF$=""
140 CURSOR 0,0: PRINT "1-> Zeit installieren"
150 CURSOR 0,1: PRINT "2-> Zeit anzeigen"
160 CURSOR 0,2: PRINT "3-> Daten anzeigen"
170 CURSOR 0,3: PRINT "4-> manuell aufnehmen"
180 ME=ASC INKEY$ - 48
190 IF ME<0 OR ME>40 GOTO 180
200 IF ME=0 POWER OFF
210 ON MEGOSUB "INP", "PRI", "OUT", "MAN"
220 GOTO "A"
230 "INP" CLS : CLEAR
240 CURSOR 0,0: KBUFF$="" : INPUT "Start ab Minute:"; MI
250 IF MI<0 OR MI>59 GOTO 240
260 MIS=STR$ MI: IF MI<10 LET MIS="" + MIS
270 CURSOR 0,1: KBUFF$="" : INPUT "Start ab Stunde:"; ST
280 IF ST<0 OR ST>24 GOTO 270
290 ST$=STR$ ST: IF ST<10 LET ST$="" + ST$
300 CURSOR 0,2: KBUFF$="" : INPUT "Start ab Tag:"; TA
310 IF TA<10 OR TA>31 GOTO 300
320 TAS=STR$ TA: IF TA<10 LET TAS="" + TAS
330 CURSOR 0,3: KBUFF$="" : INPUT "Start ab Monat:"; MO
340 IF MO<10 OR MO>12 GOTO 330
350 MO$=STR$ MO: IF MO<10 LET MO$="" + MO$
360 CLS : KBUFF$="" : 155 : INPUT "Anzahl der Daten:"; AN
370 IF AN<10 OR AN>155 GOTO 360
380 IF AN=1 GOTO 410
390 CURSOR 0,1: INPUT "Interrupt in Minuten:"; IN
400 IF IN<0 OR IN>60 GOTO 390
410 DIM M(AN)
420 IF IN=0 GOTO 440
430 WAKES (0)="" MO$ + " / " + TAS + " / " + ST$ + " / " + MIS + " : GOTO1" + CHR$ 13 : RETURN
440 WAKES (0)="" MO$ + " / " + TAS + " / " + ST$ + " / " + MIS + " : GOTO2" + CHR$ 13 : RETURN
450 "PRI" CLS : IF AN=0 RETURN
460 PRINT "start Tag 1"; TA
470 PRINT "start Monat 1"; MO
480 PRINT "start Stunde"; ST
490 PRINT "start Minute"; MI
500 LINE (76,0)-(76,31)
510 GOSUB "INKE"
520 CLS : PRINT "Anzahl der Messungen:"; AN
530 IF AN=1 PRINT "Interrupt nur einmal." : GOTO 570
540 IF IN=1 PRINT "Interrupt jede Minute."
550 IF IN>1 PRINT "Interrupt alle"; IN; " Minuten."
560 IF IN=0 PRINT "Interrupt alle 24 Stunden."
570 LINE (0,20)-(155,20), &5555
580 LINE (0,16)-(155,16)
590 CURSOR 0,3: PRINT "***"; TIME$; " ***"; DATE$; " ***"
600 IF INKEY$="" GOTO 590
610 RETURN
620 "OUT" CLS : KBUFF$="" : IF AN=0 RETURN
630 MA=0: MN=255: XM=0: M(0)=0: FOR I=1 TO AN: IF M<M(I) LET MA=M(I)
640 IF MN>M(I) LET MN=M(I)
650 M(0)=M(0)+M(I)
660 NEXT I: M(0)=M(0)/AN
670 IF MA>0 LET XM=255/MA
680 IF AN<2 FOR I=1 TO AN: LINE (0,1)-(M(I)/1.6451*XM,I): NEXT I: GOTO 700
690 FOR I=1 TO AN: LINE (I,31)-(I,31-M(I)/0.22*XM): NEXT I
700 GOSUB "INKE": CLS
710 PRINT "Mittelw."; M(0)
720 PRINT "Maximalwert:"; MA
730 PRINT "Minimalwert:"; MN
740 PRINT "Differenz:"; MA-MN
750 GOSUB "INKE": CLS
760 CURSOR 0,0: KBUFF$="" : PRINT "ohne Gleichspannung:" : INPUT "anzeigen (J/N):"; AS
770 IF AS<>"J" AND AS<>"N" GOTO 760
780 IF AS="" GOTO 830
790 CLS : WA=0: IF MA-MN>0 LET WA=255/(MA-MN)
800 IF AN<2 FOR I=1 TO AN: LINE (0,1)-(M(I)-MN)/1.6451*WA,I: NEXT I: GOTO 820
810 FOR I=1 TO AN: LINE (I,31)-(I,31-(M(I)-MN)/0.22*WA): NEXT I
820 GOSUB "INKE"
830 CLS : PRINT "Sollen die einzelnen:" : PRINT "Messwerte angezeigt"
840 CURSOR 0,2: KBUFF$="" : INPUT "werden (J/N):"; AS
850 IF AS<>"J" AND AS<>"N" GOTO 840
860 IF AS="" RETURN
870 CLS : FOR I=1 TO AN
880 CURSOR 0,0: PRINT "Messwert Nr.:"; I
890 CURSOR 0,1: PRINT "----->"; M(I)
900 CURSOR 0,2: PRINT "Weiter mit SPACE"
910 IF INKEY$<>CHR$ 32 GOTO 910
920 NEXT I: RETURN
930 "MAN" CLS : IF AN=0 RETURN
940 TW=0: CURSOR 0,0: KBUFF$="" : INPUT "triggern ab Wert:"; TW
950 IF TW<0 OR TW>255 GOTO 940
960 KBUFF$="" : CURSOR 0,1: INPUT "Verzögerung in Sek.:"; W
970 CURSOR 0,2: PRINT "Start mit SPACE"
980 IF INKEY$<>CHR$ 32 GOTO 980
990 CURSOR 0,2: PRINT "Messung läuft !!"
1000 WAIT 64*W
1010 KEY (1) OFF : ON KEY GOSUB "END": KEY (1) ON
1020 M(I)=AIN: IF M(I)<TW GOTO 1020
1030 KEY (1) OFF
1040 FOR I=2 TO AN: CURSOR 0,3: PRINT "----->"; I: M(I)=AIN IN EXT: WAIT 0: RETURN
1050 "END" TW=0: RETI

```

Das Programm belegt etwa 3300 Byte des Basic-Speichers, benötigt jedoch für die Daten zusätzlich noch ca. 1500 Bytes an Variablen usw.

Wichtiger Hinweis: nach Änderungen das Programm nur mit RENUM 10,10,xx neu nummerieren!

```

1:REM ** HARDCOPY FÜR DEN PC 1600 **
20:"A"CLEAR :GRAPH
20:GOSUB "LINE"
30:FOR Y=0TO 31
40:FOR X=0TO 155
50:P=POINT (X,Y)
60:IF P=1GOSUB "PRINT"
70:NEXT X
80:NEXT Y
90:TEXT :END
100:"LINE"
110:LLINE (245,5)-(720,-97),,B
120:LLINE (240,10)-(725,-102),,B
130:RETURN
140:"PRINT"
150:A=X:LLINE (A*3+250,-Y*3)-(A*3+252,-2-Y*3),,B
160:RETURN

```

Hans Hölzel
Jahnstrasse 9
6101 Fränkisch-Crumbach

Gestartet wird das Programm mit DEF A und es kann an Jedes andere Programm angehaengt werden.
Da es fuer den PC 1600 keinen Befehl gibt, der ein "mergen" von Programmen ermoeeglicht, muss es von Hand hinter das entsPrechende Programm einetippt werden.
* von einer Diskette

Peter Kriz
Paalende 37
2 Hamburg 73 **PC-1500 PC-1600**

PC-1500/PC-1600 erstellt Multiplikations-
Übungsaufgaben für Sextaner (zur Entlastung gestreßter Väter, Lehrer und Nachhilfelehrer)

Ausgelöst durch die Aufforderung in Heft 12/86, etwas für die Kinder der SHARP-User zu tun, veröffentliche ich mein Programm „Multiplikation mit unbekanntem Ziffern“.

Dieses Programm habe ich vor einigen Jahren erstellt, als mein Sohn die Sexta besuchte. Bei der Übung der Multiplikation zweier dreistelliger Faktoren wurden im Mathematikbuch Aufgaben gestellt, in denen in dreifacher Modifikation die Ziffern des Produkts und einige Ziffern einer oder beider Faktoren und/oder der drei Zwischenprodukte durch Fragezeichen ersetzt waren.

Mein Sohn fand an dieser Aufgabenstellung Gefallen und bat mich um weitere Übungsaufgaben. So entwickelte ich für meinen PC-1500 mit CE-150 (es läuft auch auf dem PC-1600, wenn in Zeile 140 und 240 der LINE-Befehl durch LLINE ersetzt wird) ein Programm, das jeweils zehn Aufgaben stellt. Außerdem werden mit 4-zeiligem Abstand zum Aufgabenteil die Ergebnisse abgedruckt. Dieser Teil wird natürlich vor Übergabe an den Schüler abgetrennt und dient zur Kontrolle. Die geraden Ziffern und die 1 werden deshalb im rechten Faktor ausgelassen, weil sich sonst mehrere Lösungsmöglichkeiten ergeben. Wer das bewußt einplanen will, braucht nur die Programmzeilen 20 und 90 entsprechend zu ändern.

```

1:"M":CLEAR :DIM A(10)
2:LPRINT "Multiplikation mit unbekanntem Ziffern"
3:LPRINT "In beiden Faktoren kommt keine Null vor."
4:LPRINT "Im rechten Faktor kommen auch keine geraden Ziffern und keine 1 vor."
5:REM ***. Sextaner-Problem ***
6:GRAPH :CSIZE 2:FOR S=1TO 10
7:IF S/2<>INT (S/2)AND S1GLCURSOR (-100,-130) :SCRN
8:IF S/2=INT (S/2)GLCURSOR (100,10):SORN
9:COLOR 3:LPRINT S;"":GLCURSOR (0,-16):SORN
10:RANDOM
15:X=RND 999:IF X<101GOTO 15
20:Y=RND 999:IF Y<333OR Y/2=INT (Y/2)GOTO 20
30:Z=X*Y:A(S)=Z
31:X$=STR$ X
32:K$=LEFT$ (X$,1):K=VAL K$
33:L$=MID$ (X$,2,1):L=VAL L$
34:M$=RIGHT$ (X$,1):M=VAL M$
40:Y$=STR$ Y
50:A$=LEFT$ (Y$,1):A=VAL A$
60:B$=MID$ (Y$,2,1):B=VAL B$
70:C$=RIGHT$ (Y$,1):C=VAL C$

```

Multiplikation mit unbekanntem Ziffern in beiden Faktoren kommt keine Null vor.

Im rechten Faktor kommen auch keine geraden Ziffern und keine 1 vor.

1)	2)
$\frac{221 \times 332}{66?}$	$\frac{223 \times 552}{416?}$
$\frac{727}{2229}$	$\frac{1239}{2229}$

3)	4)
$\frac{223 \times 322}{2962}$	$\frac{224 \times 592}{318?}$
$\frac{727}{2229}$	$\frac{1239}{2229}$

5)	6)
$\frac{221 \times 332}{291?}$	$\frac{226 \times 552}{458?}$
$\frac{727}{2229}$	$\frac{1239}{2229}$

Ergebnisse:

- 1: 74919
- 2: 793849
- 3: 167931
- 4: 352938
- 5: 325285
- 6: 512044

```

75:IF B=0OR C=0OR L=0OR M=0GOTO 10
80:IF B=1OR C=1GOTO 10
90:IF A/2=INT (A/2)OR B/2=INT (B/2)GOTO 10
100:D=X*A:E=X*B:F=X*C
110:I=RND 3
120:ON IGOSUB 300,400,500
130:GLCURSOR (24,-16):LPRINT Us;"*";W$
140:LINE (24,-18)-(107,-18),0
150:GLCURSOR (24,-34):LPRINT D$
160:IF D>999AND E>999GLCURSOR (36,-50)
170:IF D>999AND E<1000GLCURSOR (48,-50)
180:IF D<1000AND E<1000GLCURSOR (36,-50)
190:IF D<1000AND E>1000GLCURSOR (24,-50)
195:LPRINT D$
220:IF D<1000AND F<1000GLCURSOR (48,-66)
210:IF D>999AND F<1000GLCURSOR (60,-66)
220:IF D>999AND F>999GLCURSOR (48,-66)
230:IF D<1000AND F>999GLCURSOR (36,-66)
235:LPRINT U$
240:LINE (12,-68)-(96,-68)
250:GLCURSOR (6,-84)
260:IF Z>99999AND D>999LPRINT " ??????":GOTO 270
262:IF Z<100000AND D>999LPRINT " . ??????":GOTO 270
264:IF Z>99999LPRINT " ??????":GOTO 270
266:IF Z<100000LPRINT " ??????":GOTO 270
270:IF S=10TEXT :LF 3:GOTO 600
280:NEXT S
300:K$="?":L$="?":B$="?"
305:COLOR 0
310:U$=K$+L$+M$
320:W$=A$+B$+C$
330:IF D<1000LET D$="???"
340:IF D>999LET D$="????"
350:E$=STR$ E
360:EE$=RIGHT$ (E$,1)
370:IF F>999LET O$="???"*EE$
380:IF F<1000LET O$="???"*EE$
381:UU$=STR$ F
382:IF F>999LET F$=LEFT$ (UU$,3)
383:IF F<1000LET F$=LEFT$ (UU$,2)
390:U$=F$+"?"
395:RETURN
400:K$="?":L$="?":M$="?"
405:COLOR 1
410:U$=K$+L$+M$
415:W$=STR$ Y
420:IF D<1000LET D$="???"
430:IF D>999LET D$="????"
440:IF E<1000LET O$="???"
450:IF E>999LET O$="????"
460:U$=STR$ F
470:RETURN
500:K$="?":L$="?":C$="?"
505:COLOR 2
510:U$=K$+L$+M$
520:W$=A$+B$+C$
530:IF D<1000LET D$="???"
540:IF D>999LET D$="????"
550:F$=STR$ F
560:FF$=RIGHT$ (F$,1)

```

```

570:IF F>999LET US="???"*FF$
580:IF F<1000LET US="???"*FF$
581:EE$=STR$ E
582:IF E>999LET E$=LEFT$(EE$,3)
583:IF E<1000LET E$=LEFT$(EE$,2)
590:O$=E$+"?"
595:RETURN
600:COLOR 3:LPRINT "Ergebnisse:"*COLOR 0
610:FOR S=1TO 10
620:LPRINT S;"":A(S)
630:NEXT S
640:LF 5:END

```

Formatierung des CE-1600P auf DIN-A5-Format (hoch)

Es empfiehlt sich nicht, beim CE-1600P (Plotter des PC-1600) schmaleres als DIN-A4-Papier zu verwenden, da es sonst nicht gleichmäßig durchgezogen wird.

Um trotzdem DIN-A5-Ausdrucke im Hochformat zu erstellen, habe ich eine RESERVE-Taste mit dem Befehl: PAPER C,30, 890 und eine andere mit dem Befehl: PCONSOLE "LPT 1:", 50,2,0 belegt.

Nach Einspannen eines DIN-A4-Bogens bediene ich beide RESERVE-Tasten. Dann wird das Blatt in CSIZE 2 mit 37 Druckzeilen à 50 Zeichen bedruckt. Anschließend schneide ich (mit einer Schneidemaschine - es geht aber auch ohne) das Blatt auf DIN-A5-Hochformat zurecht.

STATUS 1

1983

P. Gargulak
Markwaldstr.45
6050 Offenbach

Es handelt sich um das bekannte Spiel "4-geinnt" für den PC-1600. Das Copyright liegt bei einem mir unbekanntem ATARI 800XL Fan, ich habe es nur für den PC-1600 umgeschrieben (die eigentlichen Routinen habe ich so belassen, wie sie der ATARI Fan geschrieben hat).

Für die, die das Spiel nicht kennen eine kleine Anleitung.

Nach dem Start des Spiels mit RUN oder DEF-A erscheint das 6x7 Kästchen große Spielfeld. Als nächstes werden die Namen der Spieler abgefragt; man kann auch gegen den Computer spielen, man muß dann entsprechend "COMPUTER" für den Namen eingeben (gibt man zweimal "Computer" ein, so spielt der Computer gegen sich selbst). Ziel des Spieles ist es in dem angezeigten Spielfeld 4 Symbole (neben des eigenen Namen) in Reihe zu bringen (horizontal, vertikal oder diagonal).

Die Eingabe: In der ersten Display-Zelle erscheint der Name, und ein Fragezeichen. Der Spieler der an der Reihe ist muß nun eine Zahl zwischen 1 und 7 eingeben. Der Computer rechnet nun ein paar Augenblicke, und zeigt dann den Zug an.

Es wird folgendermaßen angezeigt: es wurde z.B. eine 4 eingegeben, und in der 4ten Spalte befindet sich noch kein Stein, so fällt dieser bis auf den Boden des Spielfelds. Gibt nun der nächste Spieler auch 4 ein so fällt sein Stein auf den bereits vorhandenen Stein des Gegenspielers.

Dies geht so lange, bis einer der Spieler seine Steine in eine Reihe von 4 Stück bringen kann, oder das Spielfeld kein freies Feld mehr hat (unentschieden).

Das Programm belegt (mit Variablen) etwa 3000 Bytes des Basic-Speichers.

```

10 "A"CLEAR :DIM LI(7)
20 FOR I=1TO 7:LI(I)=6:NEXT I
30 DIM TA(7,6)
40 FOR I=0TO 7:FOR J=0TO 6:TA(I,J)=0:NEXT J:NEXT I
50 GO=1:CLS :GOSUB "GRID"
60 CURSOR 13,1:PRINT CHR$(7):CURSOR 13,3:PRINT " "
70 CURSOR 1,0:PRINT "1"
80 KBUFF$="":CURSOR 15,1:INPUT N1$
90 CURSOR 18,2:PRINT "gegen"
100 KBUFF$="":CURSOR 15,3:INPUT N2$
110 IF GO=1LET NA$=N1$
120 IF GO=2LET NA$=N2$
130 IF NA$="COMPUTER"GOTO 920
140 GOSUB "OUT"
150 CURSOR 16,0:PRINT " "
160 CURSOR 16,0:PRINT NA$:INPUT "I";CO
170 IF CO<1ORCO>7GOTO 160
180 IF LI(CO)=0GOTO 160
190 LI(CO)=LI(CO)+1
200 GOSUB "OUT"
210 TA(CO,LI(CO))=GO
220 WI=0
230 GOSUB 330
240 IF WI=1GOTO 220
250 IF GO=1LET GO=2:GOTO 110
260 IF CT=20CLS :PRINT "Unentschieden!":GOSUB "GRID":GOSUB "OUT":GO TO 290
270 GO=1:CT=CT+1:GOTO 110
280 CLS :GOSUB "GRID":GOSUB "OUT":PRINT NA$;" ist Sieger!"
290 CURSOR 10,1:PRINT "Noch ein Spiel?":CURSOR 13,2:PRINT " J/N ?"
300 IF INKEY$="J"GOTO "A"
310 IF INKEY$="N"CLS :CLEAR :POWER OFF
320 GOTO 300

```

```

330 X=CO
340 Y=LI(X)
350 NO=1
360 X=X-1
370 IF X=0GOTO 430
380 SQ=TA(X,Y)
390 IF SQ<>0GOTO 430
400 NO=NO+1
410 IF NO=4LET WI=1:RETURN
420 GOTO 360
430 X=CO
440 X=X+1
450 IF X=8GOTO 510
460 SQ=TA(X,Y)
470 IF SQ<>0GOTO 510
480 NO=NO+1
490 IF NO=4LET WI=1:RETURN
500 GOTO 440
510 X=CO:Y=LI(X):NO=1
520 Y=Y+1
530 IF Y=6GOTO 590
540 SQ=TA(X,Y)
550 IF SQ<>0GOTO 590
560 NO=NO+1
570 IF NO=4LET WI=1:RETURN
580 GOTO 520
590 X=CO:Y=LI(X):NO=1
600 Y=Y-1:X=X-1
610 IF X=0OR Y=6GOTO 670
620 SQ=TA(X,Y)
630 IF SQ<>0GOTO 670
640 NO=NO+1
650 IF NO=4LET WI=1:RETURN
660 GOTO 600
670 X=CO:Y=LI(X)
680 Y=Y-1:X=X+1
690 IF X=SOR Y<0GOTO 750
700 SQ=TA(X,Y)
710 IF SQ<>0GOTO 750
720 NO=NO+1
730 IF NO=4LET WI=1:RETURN
740 GOTO 680
750 X=CO:Y=LI(X):NO=1
760 Y=Y-1:X=X-1
770 IF X=0OR Y<0GOTO 830
780 SQ=TA(X,Y)
790 IF SQ<>0GOTO 830
800 NO=NO+1
810 IF NO=4LET WI=1:RETURN
820 GOTO 760
830 X=CO:Y=LI(X)
840 Y=Y+1:X=X+1
850 IF X=6OR Y=6GOTO 910
860 SQ=TA(X,Y)
870 IF SQ<>0GOTO 910
880 NO=NO+1
890 IF NO=4LET WI=1:RETURN
900 GOTO 840
910 RETURN
920 CO=1
930 IF LI(CO)=0GOTO 1000
940 LI(CO)=LI(CO)+1
950 TA(CO,LI(CO))=GO
960 GOSUB 330

```

```

970 TA(CO,LI(CO))=0
980 LI(CO)=LI(CO)+1
990 IF WI=1GOTO 190
1000 CO=CO+1
1010 IF CO<8GOTO 930
1020 ST=GO
1030 IF GO=1LET GO=2:GOTO 1050
1040 GO=1
1050 CO=1
1060 IF LI(CO)=0GOTO 1130
1070 LI(CO)=LI(CO)-1
1080 TA(CO,LI(CO))=GO
1090 GOSUB 330
1100 TA(CO,LI(CO))=0
1110 LI(CO)=LI(CO)+1
1120 IF WI=1LET GO=ST:GOTO 190

```

```

1130 CO=CO+1
1140 IF CO<8GOTO 1060
1150 GO=ST
1160 CO=CO+1
1170 IF LI(CO)=0GOTO 1160
1180 GOTO 190
1190 "OUT"
1200 FOR A=1TO 7
1210 FOR B=0TO 5
1220 IF TA(A,B)=1GOSUB "PLAYER"
1230 IF TA(A,B)=2GOSUB "COMPUTER"
1240 NEXT B
1250 NEXT A:RETURN

```

```

1260 "PLAYER"LINE (A*7,B*4+8)-(A*7+2,B*4+10),,BF:RETURN
1270 "COMPUTER"PSET (A*7+1,B*4+9):RETURN
1280 "GRID"
1290 FOR I=1TO 50STEP 7
1300 LINE (I*4,7)-(I*4,32)
1310 NEXT I
1320 FOR I=3TO 26STEP 4
1330 LINE (6,I*8)-(53,I*8)
1340 NEXT I:RETURN

```

Leopold Steininger
Margarethnerstr. 47
Enzersdorf/Fischa
2431 Kleinneusiedl
Österreich

PC-1600

Betrifft: Einfache Textverarbeitung mit 1600 + 1600P

Ich habe auf der RAM-Disk einen Briefvordruck gespeichert (Listing folgt), und immer wenn ich Zeit habe, schreibe ich ein paar Zeilen und speichere ihn sofort unter einer Namensbezeichnung ab. Mit DEF S kann ich ab Zeile Nr. 30 laufend kontrollieren, ob der Text richtig geschrieben ist, bzw. sich Einschübe an der richtigen Stelle befinden. Einige Reservetasten habe ich mit ;'', und ''+CHR\$&E1'' für B programmiert um diese Zeichen mit nur einem Tastendruck abzurufen. Wenn alles richtig ist, dann wandle ich mit DEF Z alle PRINT in LPRINT um und der Druck startet automatisch. Ich habe deshalb 52 Zeichen pro Zeile gewählt, da sich das mit zwei Anzeigezellen genau ausgeht. Den Zeilenabstand muß ich mit PRINT programmieren, da der Computer selbst LFX nicht annimmt. Mich würde interessieren, was andere Leser über mein Programm denken. Vielleicht kann jemand über den Gebrauch von SPACE\$ Auskunft geben, den der Computer zwar erkennt, ihn aber nicht verarbeiten kann.

Anschließend das Listing für den Briefvordruck.

```

10:"A":CLS :CLEAR :USING :WAIT 0:CSIZE 2
20:PCONSOLE "LPT1:",52,0,15
30:"S":WAIT :PRINT "Leopold Steininger"
40:PRINT
50:PRINT "Margarethnerstr. 47"
60:PRINT "Enzersdorf/Fischa"

```

Im Dezemberheft Ihrer Zeitschrift ALLES FÜR SHARP COMPUTER wurde ein Listing von Dominik A. Müller für den PC 1600 veröffentlicht, das in der Lage ist einen Text auf eine beliebige Stelle des Displays auszugeben. Dieses Programm war sehr gut, aber relativ langsam in der Ausgabe. Nun eine Version die in des Ausgabe des Textes schneller ist.

Im String T\$ muß der zu verarbeitende Text (max. 13 Zeichen, da meine Version nur Fettschrift ausgibt) stehen. Man kann nun die Routine "READ" aufrufen - der Text ist nun als GPRINT - Muster in Z\$(0)-Z\$(3) gespeichert. Für die Ausgabe muß in der Variable X die X - Position stehen, und in der Variable Y die Y - Position stehen. Durch den Aufruf der Routine "WRITE" wird der Text an die entsprechende Stelle des Displays ausgegeben.

```

1000 "READ"ERASE Z$(0):DIM Z$(3):80:S=0:T$=LEFT$(T$,13):FOR I=1TO
LEN T$
1010 A=ASC (MID$(T$,I,1))
1020 Z=0:P=(A-65)*6+44713
1030 X$=HEX$(PEEK P):IF LEN X$=1LET X$="0"+X$
1040 IF LEN Z$(S) SOLET Z$(S)=Z$(S)+X$+X$ELSE LET S=S+1:GOTO 1040
1050 P=P+1:Z=Z+1
1060 IF Z=5NEXT I:RETURN
1070 GOTO 1030
1080 "WRITE"GCURSOR X,Y:GPRINT "00";Z$(0);Z$(1);Z$(2);Z$(3);"00":R
ETURN

```

STATUS 1 = 280 BYTES

```

70:PRINT "2431 Kleinneusiedl"
80:PRINT "Österreich"
90:PRINT
100:PRINT " an"
110:PRINT
120:PRINT "Firma"
130:PRINT "Fischel GmbH."
140:PRINT
150:PRINT "Friedrich Kaiserstra"+CHR$ &E1+"e 54 a"
160:PRINT "1000 Berlin 12"
170:PRINT "BRD"
180:PRINT
190:PRINT "Enzersdorf, am 1986/";DATE$
200:PRINT
210:PRINT "Betrifft:"
"
220:PRINT
230:PRINT "
"
240:PRINT "
"
1000:PRINT :PRINT
1010:PRINT "
opold"
1020:GOTO 50070
50000:"Z":REM UMWANDLUNG VON PRINT IN LPRINT
50010:CLS :WAIT 0:PRINT "W A R T E N !!!"
50020:AD=49349
50030:IF PEEK AD=240AND PEEK (AD+1)=151THEN POKE (AD
+1),185
50040:AD=AD+1
50050:IF PEEK AD=13AND PEEK (AD+1)=255THEN GOTO 10
50060:GOTO 50030
50070:END

```

PC-1600

P. Gargulak
Markwaldstr. 45
6050 Offenbach

Frank Mertens
Wolfsittard 34a
4050 Mönchengladbach 5

Programm "FLUGSIMULATOR"

Das Programm von Stefan Kanarski aus Ihrer Zeitschrift "Alles für SHARP-Computer" Nr.11/86 hat mir sehr viel Spaß gemacht. Aus diesem Grunde habe ich es für meinen PC-1600 umgeschrieben. Dabei entwarf ich eine große Weltkarte, da man mit dem Plotter CE-1600P nicht mit Platz sparen muß. Die Karte habe ich vom Weltatlas durchgezeichnet. Somit liegen jetzt die eingezeichneten Städte an den richtigen Stellen. Hat man die Karte einmal Ausgedruckt und das Spiel abgebrochen, kann ein erneuter Start mit "DEF S" vorgenommen werden, ohne wiederholten Ausdruck des Weltbildes. Außerdem habe ich 3 Schwierigkeitsgrade eingebaut, die lediglich die "WAIT"-Funktion des Display-Bildes verändern, d.h. bei Schwierigkeitsgrad 1 ist die Wartezeit länger, als bei S.grad 3. Die Anzahl der Flughäfen habe ich von ursprünglich 22 auf 41 erhöht. Da die Abstände der Flugplätze teilweise sehr groß sind hat man die Möglichkeit einmal so richtig "Gas" zu geben.

Die "DATA"-Zellen, die ich geändert habe, sind gesondert ausgedruckt.

```
1:'Programm "FLUGSIMULATOR"
2:'von Stefan Kanarski
3:'Geänderte Version für den PC-1600
4:'Frank Mertens
5:'Wolfsittard 34a
6:'4050 Mönchengladbach 5
7:'
8:'STATUS 1 = 10979
9:'
10:"F"TEXT :PAPER C,2047,2047:PCONSOLE "LPT1:",255,0,0
20:CLS :WAIT 100:PRINT "   Flugsimulator (gross)"
25:CLEAR :GOSUB "GRAD"
30:RESTORE :RANDOM :DIM A$(0)*70:WAIT 0:CURSOR 2,3:PRINT "   Ausdruck de
   r Karte"
40:GRAPH :GLCURSOR (150,-50):SORGN
41:LLINE (0,0)-(650,-1140),0,1,B
42:LLINE (325,0)-(325,-1140),5,1
43:CSIZE 1:ROTATE 3:GLCURSOR (0,-1140):SORGN
44:FOR J=1TO 25:RESTORE (20000+J*100)
45:READ Z,U1,U1:GLCURSOR (U1,U1)
46:FOR I=1TO Z
47:READ U,V
48:LLINE -(U,U),0,2
49:NEXT I:NEXT J
50:RESTORE 100:FOR I=1TO 41
60:READ X,Y,S$
70:COLOR 0:LLINE (X,Y)-(X,Y+20):LLINE (X+4,Y)-(X+4,Y+20)
80:GLCURSOR (X-2,Y):LPRINT S$
90:NEXT I
400:T=1000:H=0:G=0:L=0:P=0:X=0:Y=1
410:RESTORE (100+RND 41)
420:READ XS,YS,S$
430:WAIT 125:PRINT "AUSGANGSORT:";S$
440:COLOR 3:GLCURSOR (XS+2,YS)
450:"START"RESTORE (100+RND 41)
460:READ XZ,YZ,Z$
470:IF XS=X2AND YS=Y2GOTO 450
480:WAIT 125:PRINT "Ihr Ziel ist ";Z$
490:CLS :WAIT 0
500:"STARTEN"RESTORE "STAR."
510:FOR I=1TO 4
520:L=L+G/10:IF L>75WAIT 100:PRINT "Nicht mehr rechtzeitig..":PRINT "...a
   bgehben !":GOTO "ENDE"
530:H=0:READ A$(0):GCURSOR 63,5:GPRINT A$(0)
540:GOSUB "EINGABE":IF G<16GOTO 540
550:IF H>1AND G>199LET I=4:GOTO 570
560:NEXT I:GOTO 500
570:L=0:RESTORE "AB."
580:FOR I=1TO 7
```

100	'Koordinaten
101	DATA 230,60,"HONOLULU"
102	DATA 185,200,"LOS ANGELES"
103	DATA 65,220,"FAIRBANKS"
104	DATA 245,225,"MEXIKO"
105	DATA 210,265,"NEW ORLEANS"
106	DATA 370,305,"LIMA"
107	DATA 160,285,"CHICAGO"
108	DATA 100,315,"CHURCHILL"
109	DATA 170,330,"NEW YORK"
110	DATA 290,325,"CARACAS"
111	DATA 450,365,"BUENOS AIRES"
112	DATA 375,410,"SALVADOR"
113	DATA 65,440,"GODHAVN"
114	DATA 260,495,"DAKAR"
115	DATA 90,505,"REYKJAVIK"
116	DATA 175,530,"MADRID"
117	DATA 130,545,"LONDON"
118	DATA 150,550,"PARIS"
119	DATA 100,580,"OSLO"
120	DATA 170,580,"ROM"
121	DATA 145,580,"BERLIN"
122	DATA 190,610,"ATHEN"
123	DATA 220,615,"KAIRO"
124	DATA 420,620,"JOHANNESBURG"
125	DATA 125,635,"MOSKAU"
126	DATA 295,660,"ADIS ABEBA"
127	DATA 395,680,"TANANARIVE"
128	DATA 205,680,"BAGDAD"
129	DATA 135,745,"OMSK"
130	DATA 255,765,"BOMBAY"
131	DATA 110,825,"JAKUTSK"
132	DATA 280,855,"BANKOK"
133	DATA 320,865,"SINGAPUR"
134	DATA 455,880,"PERTH"
135	DATA 240,890,"HONGKONG"
136	DATA 200,940,"OSAKA"
137	DATA 185,945,"TOKYO"
138	DATA 475,945,"MELBOURNE"
139	DATA 450,985,"SYDNEY"
140	DATA 345,985,"SUKARNAPURA"
141	DATA 620,620,"POLARSTÜTZPUNKT"

```

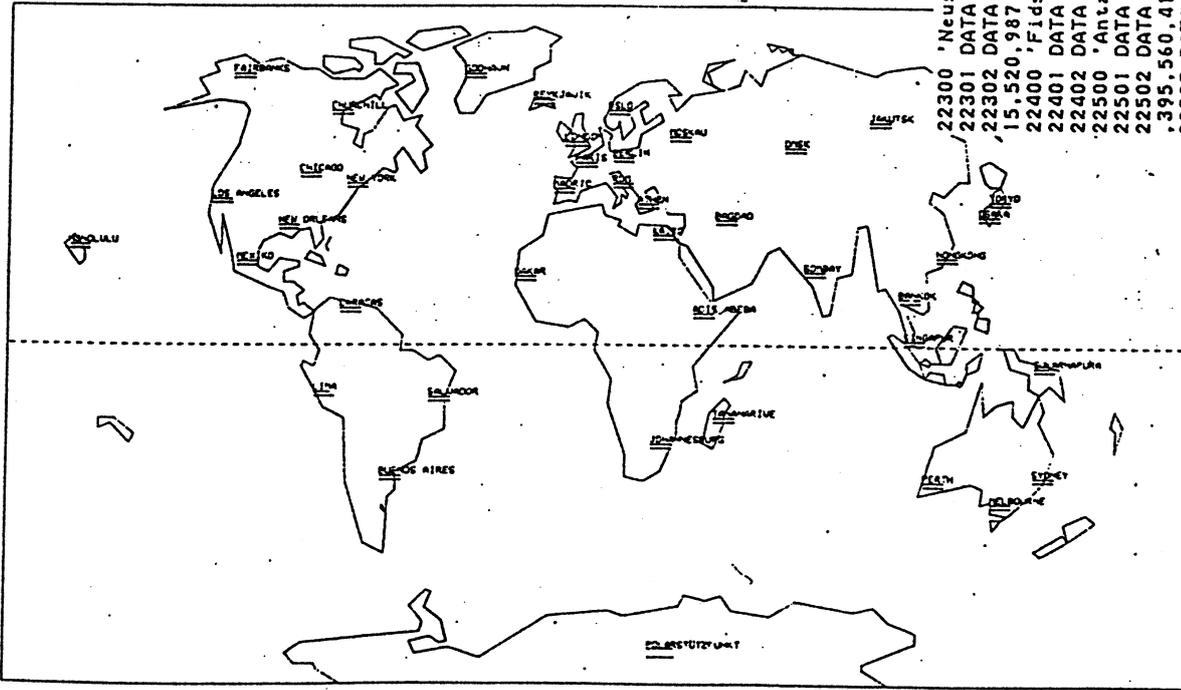
590:READ A$(0):GDCURSOR 63,5:GPRINT A$(0)
600:GOSUB "EINGABE":IF G<10R H<1WAIT 125:PRINT "Abgestürzt":GOTO "ENDE"
610:NEXT I
620:GLCURSOR (XS+2,YS+20)
630:X1=XS+2:Y1=YS+20
640:GOSUB "EINGABE":IF G<10R H<1WAIT 125:PRINT "Abgestürzt":GOTO "ENDE"
650:LLINE -(X1+(X*G/100),Y1+(Y*G/100)):X1=X1+(X*G/100):Y1=Y1+(Y*G/100)
660:IF Y1<-15LET Y1=1140:GLCURSOR (X1,Y1)
662:IF Y1>1140LET Y1=-15:GLCURSOR (X1,Y1)
665:IF X1<2LET X1=650:GLCURSOR (X1,Y1)
667:IF X1>650LET X1=2:GLCURSOR (X1,Y1)
670:IF INKEY$ ="L"GOSUB 700
680:IF E=1LET E=0:GOTO "LANDUNG"
690:GOTO 640
700:RESTORE 100
710:FOR I=1TO 41
720:READ XS,YS,S$
730:IF X1>XSAND X1<XS+4AND Y1>YSAND Y1<YS+3GOTO 750
740:NEXT I:RETURN
750:LET I=41:E=1:WAIT 125:PRINT "Landeerlaubnis erteilt!":WAIT 0:CLS :
RETURN
800:"LANDUNG"RESTORE "LAN."
810:FOR I=1TO 6
820:READ A$(0):GDCURSOR 63,5:GPRINT A$(0)
830:GOSUB "EINGABE":IF G<1AND H>1WAIT 125:PRINT "Abgestürzt":GOTO "ENDE"
840:NEXT I
850:RESTORE "STAR."
860:FOR I=1TO 4
870:L=L*G/10:IF L>75WAIT 100:PRINT "nicht mehr rechtzeitig..":PRINT "...g
elandet !":GOTO "ENDE"
880:READ A$(0):GDCURSOR 63,5:GPRINT A$(0)
890:GOSUB "EINGABE":IF G<1AND H>1WAIT 125:PRINT "Abgestürzt":GOTO "ENDE"
900:IF H<1AND G<1LET I=4:GOTO 920
910:NEXT I:GOTO 850
920:BEEP 1:WAIT 100:PRINT "Willkommen in ";S$
930:BEEP 1:IF XS=XZAND YS=YZPRINT "Ihrem Zielort":P=P+1000:GOTO 945
940:PRINT "zur Zwischenlandung":P=P-100
945:BEEP 1:PRINT "Für die Landung haben Sie":BEEP 1:PRINT 10*(30-L);" Pu
nkte bekommen"
950:P=P+10*(30-L):IF P>10000BEEP 1:PRINT "Über 10000 Punkte":GOTO "ENDE"
955:BEEP 1:PRINT "Sie haben jetzt ":BEEP 1:PRINT P;" Punkte"
960:BEEP 1:X=0:Y=1:T=1000:H=0:G=0:L=0
965:PRINT "Ihr Aufenthalt":BEEP 1:PRINT " in ";S$:BEEP 1:PRINT "Ist jetzt
beendet"
970:BEEP 1:PRINT "Auf Wiedersehen"
975:BEEP 1:PRINT "Guten Flug":IF S$<>Z$BEEP 1:PRINT "nach ";Z$
980:BEEP 1:CLS :IF S$=Z$LET XZ=XS:YZ=YS:Z$=S$:GOTO "START"
990:GOTO "STARTEN"
1000:"GRAD"
1010:INPUT "Schwierigkeitsgrad (1-3): ";W
1011:IF W<1OR W>3GOTO 1000
1020:IF W=1LET W=4
1030:IF W=2LET W=2
1040:IF W=3LET W=0
1060:RETURN
8000:"ENDE"IF P<10000LPRINT "*"
8005:WAIT 100:PRINT "Spielende"
8010:PRINT "Sie haben ":WAIT 250:PRINT P;" Punkte erreicht"
8020:GLCURSOR (0,0)
8030:"S"WAIT 0:INPUT "Noch ein Spiel? (J/N)";A$
8050:IF A$="J"GOSUB "GRAD":GOTO 400
8060:IF A$="N"WAIT 200:PRINT "Auf Wiedersehen":CLS :END
8070:GOTO 8030
9000:"EINGABE"BEEP 1,10,25:A=ASC INKEY$
9010:IF (X=0AND Y=0)OR X<-1OR X>1OR Y<-1OR Y>1LET X=0:Y=1
9011:IF A<>13GOTO 9020
9012:IF X=-1AND Y=1LET X=0:Y=1:GOTO 9020
9013:IF X=0AND Y=1LET X=1:Y=1:GOTO 9020
9014:IF X=1AND Y=1LET X=1:Y=0:GOTO 9020
9015:IF X=1AND Y=0LET X=1:Y=-1:GOTO 9020
9016:IF X=1AND Y=-1LET X=0:Y=-1:GOTO 9020
9017:IF X=0AND Y=-1LET X=-1:Y=-1:GOTO 9020
9018:IF X=-1AND Y=-1LET X=-1:Y=0:GOTO 9020
9019:IF X=-1AND Y=0LET X=-1:Y=1
9020:IF A<>32GOTO 9030
9021:IF X=0AND Y=1LET X=-1:Y=1:GOTO 9030
9022:IF X=-1AND Y=1LET X=-1:Y=0:GOTO 9030
9023:IF X=-1AND Y=0LET X=-1:Y=-1:GOTO 9030
9024:IF X=-1AND Y=-1LET X=0:Y=-1:GOTO 9030
9025:IF X=0AND Y=-1LET X=1:Y=-1:GOTO 9030
9026:IF X=1AND Y=-1LET X=1:Y=0:GOTO 9030
9027:IF X=1AND Y=0LET X=1:Y=
J 9030
9028:IF X=1AND Y=1LET X=0:Y=1:GOTO 9030
9029:IF X=0AND Y=0LET X=0:Y=1

```


21607 DATA 293,877,288,883,253,880,247,895,225,906,195,895,185,882,175,885,180,895
 21608 DATA 165,910,165,920,125,915,115,890,100,900,103,910,92,925,105,927,125,950
 21609 DATA 100,940,92,940,90,950,77,950,74,943,70,960,60,940,65,925,60,915
 21610 DATA 65,890,55,875,60,840,65,830,45,760,45,750,40,745,43,740,38,735
 21611 DATA 43,730,35,725,40,750,45,750,45,760,61,725,75,720,60,705,75,700
 21612 DATA 80,685,70,665,55,670,50,695,63,670,75,675,80,685,85,655,100,640
 21613 DATA 85,632,84,645,70,630,73,605,88,580,108,570,115,572,119,578,115,584
 21614 DATA 110,585,128,588,115,600,101,594,90,610,95,613,100,605,113,610,102,625
 21615 DATA 110,628,123,606,135,604,139,576,130,575,123,578,115,565,133,570,141,559
 21616 DATA 149,540,155,546,165,541,163,525,173,525,177,528,190,525
 21700 'Seychellen
 21701 DATA 5,360,690
 21702 DATA 355,707,340,716,341,707,354,702,360,690
 21800 'Madagaskar
 21801 DATA 7,419,670
 21802 DATA 424,680,400,690,390,690,382,696,375,691,390,676,419,670
 21900 'Indonesien+Malaysia
 21901 DATA 33,315,845
 21902 DATA 345,865,355,880,360,900,360,910,362,914,360,916,350,920,345,931,330,935
 21903 DATA 329,935,340,920,350,920,360,916,356,913,360,910,360,900,350,898,348,885
 21904 DATA 355,880,340,875,343,883,340,886,340,905,305,920,303,907,325,890,340,895
 21905 DATA 340,886,335,881,340,875,320,860,312,849,315,845
 22000 'Philippinen
 22001 DATA 18,270,910
 22002 DATA 275,925,280,925,280,928,290,925,290,930,295,930,305,927,310,940,296,941
 22003 DATA 295,930,290,930,285,935,280,928,280,931,275,925,265,925,260,915,270,910
 22100 'Japan
 22101 DATA 13,190,935
 22102 DATA 200,935,207,940,180,956,170,950,155,955,145,940,150,930,165,935,170,940
 22103 DATA 170,950,175,943,185,943,190,935
 22200 'Australien+Neuguinea
 22201 DATA 35,467,875
 22202 DATA 455,925,460,937,475,935,472,945,483,945,486,961,490,955,500,947,489,945
 22203 DATA 490,955,486,961,455,995,425,1005,396,990,385,995,370,990,361,1000,375,1015
 22204 DATA 345,1009,335,970,325,965,325,955,345,962,350,980,364,980,370,990,400,975
 22205 DATA 390,965,375,965,380,949,390,937,357,935,405,920,420,890,467,875
 22300 'Neuseeland
 22301 DATA 8,520,987
 22302 DATA 525,993,510,1020,492,1046,485,1037,495,1015,510,1020,504,1015,520,987
 22400 'Fidschi-Inseln
 22401 DATA 5,400,1060
 22402 DATA 405,1065,425,1064,400,1072,385,1064,400,1060
 22500 'Antarktis
 22501 DATA 37,650,330
 22502 DATA 635,305,630,303,630,284,620,275,610,290,600,405,590,407,575,395,560,410
 22503 DATA 562,415,577,409,590,425,610,430,611,423,605,415,610,395,621,400,615,430
 22504 DATA 625,460,585,525,580,580,585,595,575,650,565,655,570,675,565,690,580,700
 22505 DATA 585,735,570,760,580,815,575,835,590,850,600,875,610,875,630,825,635,825
 22506 DATA 650,850 Ende der 'DATA'-Zeilen
 21300 'Island
 21301 DATA 5,100,500
 21302 DATA 95,513,98,525,88,520,88,505,100,500
 21400 'England+Irland
 21401 DATA 12,120,527
 21402 DATA 130,533,119,540,135,544,145,538,135,558,120,553
 21403 DATA 115,550,103,551,104,544,119,540,110,530,120,527
 21500 'Afrika
 21501 DATA 33,268,490
 21502 DATA 303,510,308,540,300,555,313,568,330,565,358,581,381,578,430,595,451,595
 21503 DATA 455,613,445,630,410,645,398,645,390,648,378,650,368,665,345,660,293,703
 21609 DATA 100,940,92,940,90,950,77,950,74,943,70,960,60,940,65,925,60,915
 21610 DATA 65,890,55,875,60,840,65,830,45,760,45,750,40,745,43,740,38,735
 21611 DATA 43,730,35,725,40,750,45,750,45,760,61,725,75,720,60,705,75,700
 21612 DATA 80,685,70,665,55,670,50,695,63,670,75,675,80,685,85,655,100,640
 21613 DATA 85,632,84,645,70,630,73,605,88,580,108,570,115,572,119,578,115,584
 21614 DATA 110,585,128,588,115,600,101,594,90,610,95,613,100,605,113,610,102,625
 21615 DATA 110,628,123,606,135,604,139,576,130,575,123,578,115,565,133,570,141,559
 21616 DATA 149,540,155,546,165,541,163,525,173,525,177,528,190,525
 21700 'Seychellen
 21701 DATA 5,360,690
 21702 DATA 355,707,340,716,341,707,354,702,360,690
 21800 'Madagaskar
 21801 DATA 7,419,670
 21802 DATA 424,680,400,690,390,690,382,696,375,691,390,676,419,670
 21900 'Indonesien+Malaysia
 21901 DATA 33,315,845
 21902 DATA 345,865,355,880,360,900,360,910,362,914,360,916,350,920,345,931,330,935
 21903 DATA 329,935,340,920,350,920,360,916,356,913,360,910,360,900,350,898,348,885
 21904 DATA 355,880,340,875,343,883,340,886,340,905,305,920,303,907,325,890,340,895
 21905 DATA 340,886,335,881,340,875,320,860,312,849,315,845
 22000 'Philippinen
 22001 DATA 18,270,910
 22002 DATA 275,925,280,925,280,928,290,925,290,930,295,930,305,927,310,940,296,941
 22003 DATA 295,930,290,930,285,935,280,928,280,931,275,925,265,925,260,915,270,910
 22100 'Japan
 22101 DATA 13,190,935
 22102 DATA 200,935,207,940,180,956,170,950,155,955,145,940,150,930,165,935,170,940
 22103 DATA 170,950,175,943,185,943,190,935
 22200 'Australien+Neuguinea
 22201 DATA 35,467,875
 22202 DATA 455,925,460,937,475,935,472,945,483,945,486,961,490,955,500,947,489,945
 22203 DATA 490,955,486,961,455,995,425,1005,396,990,385,995,370,990,361,1000,375,1015
 22204 DATA 345,1009,335,970,325,965,325,955,345,962,350,980,364,980,370,990,400,975
 22205 DATA 390,965,375,965,380,949,390,937,357,935,405,920,420,890,467,875
 22300 'Neuseeland
 22301 DATA 8,520,987
 22302 DATA 525,993,510,1020,492,1046,485,1037,495,1015,510,1020,504,1015,520,987
 22400 'Fidschi-Inseln
 22401 DATA 5,400,1060
 22402 DATA 405,1065,425,1064,400,1072,385,1064,400,1060
 22500 'Antarktis
 22501 DATA 37,650,330
 22502 DATA 635,305,630,303,630,284,620,275,610,290,600,405,590,407,575,395,560,410
 22503 DATA 562,415,577,409,590,425,610,430,611,423,605,415,610,395,621,400,615,430
 22504 DATA 625,460,585,525,580,580,585,595,575,650,565,655,570,675,565,690,580,700
 22505 DATA 585,735,570,760,580,815,575,835,590,850,600,875,610,875,630,825,635,825
 22506 DATA 650,850 Ende der 'DATA'-Zeilen
 21300 'Island
 21301 DATA 5,100,500
 21302 DATA 95,513,98,525,88,520,88,505,100,500
 21400 'England+Irland
 21401 DATA 12,120,527
 21402 DATA 130,533,119,540,135,544,145,538,135,558,120,553
 21403 DATA 115,550,103,551,104,544,119,540,110,530,120,527
 21500 'Afrika
 21501 DATA 33,268,490
 21502 DATA 303,510,308,540,300,555,313,568,330,565,358,581,381,578,430,595,451,595
 21503 DATA 455,613,445,630,410,645,398,645,390,648,378,650,368,665,345,660,293,703
 21504 DATA 287,695,295,685,240,645,225,645,210,633,209,605,220,603,210,585,205,588
 21505 DATA 200,572,190,570,200,520,220,515,235,493,268,490
 21600 'Europa+Sowjetunion
 21601 DATA 133,190,525
 21602 DATA 186,538,188,546,177,550,169,573,183,584,194,580,188,588,180,586,184,593
 21603 DATA 180,595,160,576,158,585,184,606,190,602,199,606,195,620,177,611,172,620
 21604 DATA 199,634,199,650,211,650,226,643,210,635,226,643,230,653,230,653,270,679
 21605 DATA 282,679,265,715,235,730,230,740,268,773,296,783,294,790,259,796,255,810
 21606 DATA 210,815,208,824,271,835,269,842,290,855,323,865,323,870,310,870,284,858
 21607 DATA 293,877,288,883,253,880,247,895,225,906,195,895,185,882,175,885,180,895
 21608 DATA 165,910,165,920,125,915,115,890,100,900,103,910,92,925,105,927,125,950
 21609 DATA 100,940,92,940,90,950,77,950,74,943,70,960,60,940,65,925,60,915
 21610 DATA 65,890,55,875,60,840,65,830,45,760,45,750,40,745,43,740,38,735
 21611 DATA 43,730,35,725,40,750,45,750,45,760,61,725,75,720,60,705,75,700
 21612 DATA 80,685,70,665,55,670,50,695,63,670,75,675,80,685,85,655,100,640
 21613 DATA 85,632,84,645,70,630,73,605,88,580,108,570,115,572,119,578,115,584
 21614 DATA 110,585,128,588,115,600,101,594,90,610,95,613,100,605,113,610,102,625
 21615 DATA 110,628,123,606,135,604,139,576,130,575,123,578,115,565,133,570,141,559
 21616 DATA 149,540,155,546,165,541,163,525,173,525,177,528,190,525

21700 'Seychellen
 21701 DATA 5,360,690
 21702 DATA 355,707,340,716,341,707,354,702,360,690
 21800 'Madagaskar
 21801 DATA 7,419,670
 21802 DATA 424,680,400,690,390,690,382,696,375,691,390,676,419,670
 21900 'Indonesien+Malaysia
 21901 DATA 33,315,845
 21902 DATA 345,865,355,880,360,900,360,910,362,914,360,916,350,920,345,931,330,935
 21903 DATA 329,935,340,920,350,920,360,916,356,913,360,910,360,900,350,898,348,885
 21904 DATA 355,880,340,875,343,883,340,886,340,905,305,920,303,907,325,890,340,895
 21905 DATA 340,886,335,881,340,875,320,860,312,849,315,845
 22000 'Philippinen
 22001 DATA 18,270,910
 22002 DATA 275,925,280,925,280,928,290,925,290,930,295,930,305,927,310,940,296,941
 22003 DATA 295,930,290,930,285,935,280,928,280,931,275,925,265,925,260,915,270,910
 22100 'Japan
 22101 DATA 13,190,935
 22102 DATA 200,935,207,940,180,956,170,950,155,955,145,940,150,930,165,935,170,940
 22103 DATA 170,950,175,943,185,943,190,935
 22200 'Australien+Neuguinea
 22201 DATA 35,467,875
 22202 DATA 455,925,460,937,475,935,472,945,483,945,486,961,490,955,500,947,489,945
 22203 DATA 490,955,486,961,455,995,425,1005,396,990,385,995,370,990,361,1000,375,1015
 22204 DATA 345,1009,335,970,325,965,325,955,345,962,350,980,364,980,370,990,400,975
 22205 DATA 390,965,375,965,380,949,390,937,357,935,405,920,420,890,467,875

22300 'Neuseeland
 22301 DATA 8,520,987
 22302 DATA 525,993,510,1020,492,1046,485,1037,495,1015,510,1020,504,1015,520,987
 22400 'Fidschi-Inseln
 22401 DATA 5,400,1060
 22402 DATA 405,1065,425,1064,400,1072,385,1064,400,1060
 22500 'Antarktis
 22501 DATA 37,650,330
 22502 DATA 635,305,630,303,630,284,620,275,610,290,600,405,590,407,575,395,560,410
 22503 DATA 562,415,577,409,590,425,610,430,611,423,605,415,610,395,621,400,615,430
 22504 DATA 625,460,585,525,580,580,585,595,575,650,565,655,570,675,565,690,580,700
 22505 DATA 585,735,570,760,580,815,575,835,590,850,600,875,610,875,630,825,635,825
 22506 DATA 650,850



```

7000:REM   *** ASCII-CODE * PC-1600 + CE-1600P ***
7005:DIM A*(0):54:CLS
7010:INPUT "Schriftgrösse? (1-2) ";S:IF S=1LET Z=28ELSE IF S=2LET Z=75
7011:IF S>2THEN 7010
7012:INPUT "Rahmenfarbe? (0-3) ";RF,"Schriftfarbe? (0-3) ";SF
7013:IF RF>3OR SF>3THEN 7012
7015:GRAPH
7020:GLCURSOR (0,0):COLOR RF:CSIZE S:GOTO 7026
7022:GOTO 7060
7026:X1=50*S:Y1=-15*S:X2=375*S:Y2=-240*S
7030:LLINE (X1,Y1)-(X2,Y2),,B
7040:GLCURSOR (0,0)
7048:UX=75*S:UY=-15*S:UX=75*S:UY=-240*S
7050:LLINE (UX,UY)-(UX,UY)
7051:WX=50*S:WY=-35*S:RX=375*S:RY=-35*S
7052:LLINE (WX,WY)-(RX,RY)
7054:GLCURSOR (0,0)
7060:GLCURSOR (50*S,-14*S)
  
```

Ende der 'DATA'-Zellen

```

7070:A$(0)="Hex 0 1 2 3 4 5 6 7 8 9 A B C D E F"
7080:GLCURSOR (55*S,-30*S):LPRINT A$(0):GLCURSOR (5*S,-2):SORGN
7092:TEXT :LF 1:C=0:LCURSOR 10:CSIZE S 7330:TAB (13):LPRINT S#
7093:FOR I=1TO 10
7094:READ C
7095:TAB (9):LPRINT C
7096:NEXT I:RESTORE 7310
7112:FOR J=1TO 6
7113:READ B#
7114:TAB (10):LPRINT B#
7115:NEXT J:RESTORE 7400
7300:DATA 0,1,2,3,4,5,6,7,8,9
7310:DATA "A","B","C","D","E","F"
7311:COLOR SF
7312:LF -16
7320:FOR I=0TO 15
7325:READ S#
7335:NEXT I
7340:LF -17
7344:LCURSOR (14):N=0:T=14
7345:FOR I=0TO 254
7346:IF N=16GOSUB 7500
7350:TAB (T):LPRINT CHR# (I)
7351:N=N+1
7352:NEXT I
7354:GOTO 7505.
7400:DATA "NUL","",",",";","<","\","|","^","_","`","~","BEL","BS",
"HT","LF","UT","FF","CR","SO","SI"
7500:LF -16:N=0:T=T+3:RETURN
7505:LF 3
7510:END

```

Hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL		0	1	A	2	3	4	5	6	7	8	9	A	B	C
1		!	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2		"	#	\$	%	&	'	()	*	+	,	<	\		^
3																
4																
5																
6																
7	BEL															
8	BS															
9	HT															
A	LF															
B	UT															
C	FF															
D	CR															
E	SO															
F	SI															

REINHARD HAAK KARL MARX-STR.200 1000 BERLIN 44

DEZIMAL-HEXADEZIMAL-BINÄR UMWANDLUNGEN

Mit Hilfe dieses Programms sind Zahlenumwandlungen der 3 Zahlensysteme untereinander möglich. Starten mit DEF A. Nach dem Titel erscheint das Hauptmenue. Mit 1 kann ein Programm zur fortlaufenden POKE Anweisung und mit 2 zur PEEK Anweisung aufgerufen werden. Durch wählender Zahl 3 wird das Menü zur Umwandlung der Zahlensysteme ausgewählt. Das Ergebnis einer Umwandlung kann in eine weitere Umwandlung übernommen werden, z.B.

HEZ-DEZ Umwandlung- das Ergebnis kann z.B. in eine DEZ-BIN Umrechnung übernommen werden. Ansonsten erklärt sich das Programm selbst. Für eine Benachrichtigung bei evtl. Fehlern bzw. für Verbesserungsvorschläge bin ich dankbar.

```

10:ES=INKEY$(1):IF E<>CHR$(0) 240:CLS:PRINT "PEEK Adresse = # ";
GOTO 10ELSE CLS:RETURN
20:"A"CLS:CLEAR:DIM A$(1),X$(1): 250:IF A>65535GOTO 220
LINE (2,3)-(152,29),,B:LINE (6 260:B=PEEK A:PRINT "PEEK Wort =
,B)-(149,25),,B
30:ON ERROR GOTO 1390 270:GOSUB 300
40:CURSOR 9,1:PRINT " PC-1600 ": 280:IF E=2LET A=A+1:GOTO 240
CURSOR 7,2:PRINT "UMRECHNUNGEN": 290:IF E=1GOTO 220ELSE RETURN
:GOSUB 10" 300:LINE (0,15)-(155,15):PRINT "Neu
e Adr. >1 Weiter >2"
60:PRINT "Peek >2" 310:E=0:PRINT "Menue >3"
70:PRINT "Umrechnungen >3" 320:ES=INKEY$(1):E=VAL E$:IF E<10R
80:PRINT "ENDE >4 >?" E>3GOTO 320
90:ES=INKEY$(1):E=VAL E$:IF E<10R 330:RETURN
E>4GOTO 90 340:REM ***UMRECHNUNG***
100:X=0:ON EGOSUB 130,220,350,1410 350:CLS:W=0:PRINT " DeZ-Hex >1
110:GOTO 50 DeZ-Bin >4"
120:REM ***EINGABE*** 360:PRINT " Hex-Dez >2 Hex-Bin
130:CLS:INPUT "POKE Adresse ? # "; >5"
A 370:PRINT " Bin-Dez >3 Bin-Hex
:GOTO 160 >6"
140:IF A<0OR A>65535GOTO 130ELSE 380:PRINT " MENUE >7 ZIFFER >
GOTO 160 "
150:CLS:PRINT "POKE Adresse = # "; 390:LINE (1,-1)-(155,29),,B:LINE (
A 1,7)-(155,7):LINE (1,15)-(155,1
160:INPUT "POKE Wert ? # ";B 5):LINE (78,0)-(78,31)
170:IF B<0OR B>255GOTO 130 400:LINE (1,23)-(1,31)
180:POKE A,B:GOSUB 300 410:W#=INKEY$(1):W=VAL W$:IF W<10R
190:IF E=2LET A=A+1:GOTO 150 W>7GOTO 410
200:IF E=1GOTO 130ELSE RETURN 420:IF W=7THEN RETURN
210:REM ***AUSGABE*** 430:IF P=0GOTO 510
220:CLS:INPUT "PEEK Adresse ? # "; 440:IF (P=1AND W=2)OR (P=1AND W=5)
A GOTO 510
230:IF A<0OR A>65535GOTO 220ELSE 510:GOTO 510
GOTO 260

```

```

450:IF (P=2AND W=1)OR (P=2AND W=4)
GOTO 510
460:IF (P=3AND W=3)OR (P=3AND W=5)
GOTO 510
470:GOTO 350
480:DATA "0","0000","1","0001","2",
"0010","3","0011","4","0100","5",
"0101"
490:DATA "6","0110","7","0111","8",
"1000","9","1001","A","1010","B",
"1011"
500:DATA "C","1100","D","1101","E",
"1110","F","1111"
510:ON WGOSUB 540,640,820,960,1070,
1190
520:GOTO 350
530:REM ***DEZ-HEX***
540:CLS:N=0:A$(1)="" :M=0:E=0
550:IF X$(1)<>"LET M=VAL X$(1):
PRINT "DEZ. = #";N:GOTO 570
560:INPUT "DEZ. ? # ";N
570:IF N<0OR N>65535GOTO 540
580:A$(1)=HEX$(N)
590:PRINT "HEX. = & ";A$(1)
600:X$(1)=A$(1):P=1
610:GOSUB 1320
620:IF E=2GOTO 540ELSE RETURN
630:REM ***HEX-DEZ***
640:CLS:A$(1)="" :B$="" :C$="" :X=0:Y
=0:Z=0:A=0
650:IF X$(1)<>"LET A$(1)=X$(1):
PRINT "HEX. = & ";A$(1):GOTO 670
660:INPUT "HEX. ? & ";A$(1)
670:X=LEN A$(1):Y=X
680:IF X>4GOTO 640.
690:FOR I=1TO X
700:B$=MID$(A$(1),Y,1):B=0:Z=0.
710:READ C$,D$:IF C$(1)=B$LET Z=Z+1:
GOTO 710
720:IF I=1LET B=Z:GOTO 760
730:IF I=2LET B=Z+Z*15:GOTO 760
740:IF I=3LET B=Z+Z*15+Z*240:GOTO 7
60
750:IF I=4LET B=Z+Z*15+Z*240+Z*3040
760:RESTORE :A=A+B:Y=Y-1
770:NEXT I
780:PRINT "DEZ. = #";A:X$(1)=STR$(A
):P=2

```

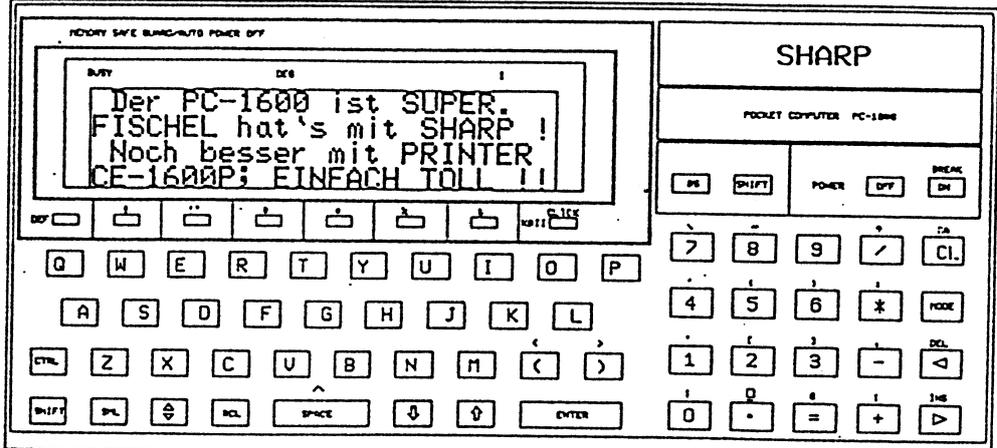
```

730:GOSUB 1320
800:IF E=2GOTO 64CEELSE RETURN
810:REM ***BIN-DEZ***
920:CLS :A*(1)="" :Z*="" :X=0 :J=0 :V=0
:A=0
930:IF X*(1)<>""LET A*(1)=X*(1):
PRINT "BIN.= % ";A*(1):GOTO 550
940:INPUT "BIN.? ";A*(1)
950:X=LEN A*(1):J=1
960:IF X>16GOTO 820
970:FOR I=1TO X
880:Z*=RIGHT$(A*(1),I):Y*=LEFT$(Z
$,1)
890:Y=VAL Y*:IF Y<0AND Y<>1LET I=X
:GOTO 820
900:A=A+Y*J:J=J*2
910:NEXT I
920:PRINT "DEZ.= #";A:X*(1)=STR$(A
):P=2
930:GOSUB 1320
940:IF E=2GOTO 820ELSE RETURN
950:REM ***DEZ-BIN***
960:CLS :A=0:A*(1)="" :B*=""
970:IF X*(1)<>""LET A=VAL X*(1):
PRINT "DEZ.= #";A:GOTO 990
980:INPUT "DEZ.? # ";A
990:IF A>65535GOTO 960
1000:A=INT(A)/2:B2=INT(A):B3=A-B2
1010:IF B3+.5=1LET A*(1)="1"+A*(1)
ELSE LET A*(1)="0"+A*(1)
1020:IF A=1GOTO 1000
1030:PRINT "BIN.= % ";A*(1):X*(1)=A*(
1):P=3
1040:GOSUB 1320
1050:IF E=2GOTO 960ELSE RETURN
1060:REM ***HEX-BIN***
1070:CLS :A*(1)="" :B*="" :C*="" :X=0
1080:IF X*(1)<>""LET A*(1)=X*(1):
PRINT "HEX.= & ";A*(1):GOTO 110
0
1090:CLS :INPUT "HEX.? & ";A*(1)
1100:X=LEN A*(1):IF X>4GOTO 1090
1110:FOR I=1TO X
1120:P=1:B*=LEFT$(A*(1),I):B*=
RIGHT$(B*,1)
1130:READ F$,D$:IF F*=B$LET B*=D$:P=
0:ELSE GOTO 1130
1140:RESTORE :C*=C*+B$:IF P=1GOTO 10
60
1150:NEXT I
1160:PRINT "BIN.= % ";C*:X*(1)=C*:P=
3:GOSUB 1320
1170:IF E=2GOTO 1070ELSE RETURN
1180:REM ***BIN-HEX***
1190:CLS :A*(1)="" :B*="" :C*="" :M*=""
:X=0:R=0
1200:IF X*(1)<>""LET A*(1)=X*(1):
PRINT "BIN.= % ";A*(1):GOTO 122
0
1210:INPUT "BIN.? ";A*(1)
1220:X=LEN A*(1):IF X>16GOTO 1210
1230:IF X<16LET A*(1)="0"+A*(1):GOTO
1220
1240:M*=LEFT$(A*(1),4):GOSUB 1300
1250:B*=LEFT$(A*(1),8):M*=RIGHT$(B
$,4):GOSUB 1300
1260:B*=RIGHT$(A*(1),8):M*=LEFT$(B
$,4):GOSUB 1300
1270:M*=RIGHT$(A*(1),4):GOSUB 1300
1280:PRINT "HEX.= & ";C*:X*(1)=C*:P=
1:GOSUB 1320
1290:IF E=2GOTO 1180ELSE RETURN
1300:READ F$,D$:IF M*=D$LET C*=C*+F$
:R=1ELSE GOTO 1300
1310:RESTORE :IF R=0GOTO 1190ELSE
RETURN
1320:REM ***WEITER-MENUE***
1330:E=0:LINE(0,15)-(15,15)
1340:PRINT "Ergebnis übernehmen_>1":
1350:PRINT "Weiter_>2 "Menu_>3"
1360:E=INKEY$(1):E=VAL E$:IF E<10R
">E)GOTO 1360
1370:IF E<>1LET X*(1)="" :P=0
1380:RETURN
1390:IF ERN =7THEN RESUME
1400:PRINT "FEHLERHAFT EINGABE":
GOSUB 10
1410:CLS :POWER OFF :END
STATUS 1 = 3873

```

Benno Sturm
Münsterscher Damm 30
2906 Wardenburg

Ich habe meinen SHARP-Computer sich selbst darstellen lassen.
--- PC-1600 mit dem CE-1600P ---



```

10:REM *** Selbstc 80:DATA 265,25,360,25 25,78,865,78,865,1 ,265,97,265,72,300
110:DATA 360,50,265,50,265 85,825,105 ,72,240,72,240,97,
20:CLEAR :CLS :GRAPH ,25,365,25,420,25, 140:DATA 825,78,865,78 ,205,97,205,72,240,
:GLCURSOR(0,-430) 420,50,365,50,385, 805,78,805,105,76 ,72
:SORGN:RESTORE : 25 5,105,765,78,805,7 200:DATA 180,72,180,97
COLOR 0 90:DATA 445,25,480,25 ,8,745,78 ,145,97,145,72,180
30:FOR I=1TO 78:READ ,480,50,445,50,445 ,72,120,72,120,97,
A,B,C,D,E,F,G,H,J, ,25 8,745,78 85,97,85,72,120,72
K:GLCURSOR(A,B) 100:DATA 505,25,600,25 ,600,50,505,50,505 150:DATA 745,105,705,1 210:DATA 60,72,60,97,2
40:LLINE -(C,D)-(E,F) ,25,645,25,685,25, 85,78,685,105 5,97,25,72,60,72,9
-(G,H)-(J,K):NEXT 685,52,645,52,645, 160:DATA 645,105,645,7 0,119,90,144,55,14
I 25 8,685,78,600,72,60 4,55,119,90,119
50:DATA 0,0,955,0,955 ,600,72 0,97,565,97,565,72 220:DATA 150,119,150,1
,430,0,430,0,0,5,1 110:DATA 705,25,745,25 ,600,72 600,72 44,115,144,115,119
7,950,17,950,425,5 ,745,52,705,52,705 ,150,119,210,119,2
,425,5,17 ,25,765,25,805,25, ,505,97,505,72,540 10,144,175,144
60:DATA 25,25,60,25,6 805,52,765,52,765, 170:DATA 540,72,540,97 ,210:DATA 175,119,210,1
0,50,25,50,25,25,8 ,25 72,480,72,480,97, 19,270,119,270,144
5,25,120,25,120,50 25 445,97,445,72,480, ,235,144,235,119,2
,85,50,85,25 120:DATA 825,25,865,25 ,72 70,119
70:DATA 145,25,180,25 ,865,52,825,52,825 ,385,97,385,72,420 240:DATA 330,119,330,1
,180,50,145,50,145 ,25,885,25,925,25, ,72,360,72,360,97, 44,295,144,295,119
,25,205,25,240,25, 925,52,885,52,885, ,325,97,325,72,360, ,330,119,390,119,3
240,50,205,50,205, 25 72 90,144,355,144
25 130: DATA 925,78,925,10 190:DATA 300,72,300,97 250:DATA 355,119,390,1
985,105,885,78,9

```

```

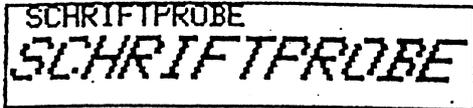
19,450,119,450,144 450:DATA 455,215,480,2 1:PCONSOLE "LPT1:" 960:LPRINT "\
,415,144,415,119,4 15,555,215,555,225 ,255,0,13
50,119 530,225,530,215,5 660:LPRINT "BUSY
260:DATA 510,119,510,1 55,215 ? CA"
44,475,144,475,119 460:DATA 680,250,680,2 DEG 970:GRAPH :GLCURSOR (0
,510,119,570,119,5 68,645,268,645,250 ,1):TEXT :CSIZE 2:
70,144,535,144 680,250,740,250,7 680,250,740,250,7 PCONSOLE "LPT1:",2
270:DATA 535,119,570,1 40,268,705,268 690:PCONSOLE "LPT1:",2 980:LPRINT " "
19,685,132,685,159 470:DATA 705,250,740,2 55,0,0 990:GRAPH :GLCURSOR (0
,645,159,645,132,6 50,865,250,865,268 700:GRAPH :GLCURSOR (0 ,12):TEXT :COLOR 0
85,132 65,250 1:PCONSOLE "LPT1:" :PCONSOLE "LPT1:",
280:DATA 745,132,745,1 480:DATA 925,250,925,2 ,255,0,55
59,705,159,705,132 68,890,268,890,250 710:LPRINT "POCKET COM 1000:LPRINT "7 8
,745,132,805,132,8 40,298,630,298 925,250,940,230,9 9 "
05,159,765,159 490:DATA 630,230,940,2 40,298,630,298 720:PCONSOLE "LPT1:",2 1010:GRAPH :GLCURSOR (0
290:DATA 765,132,805,1 32,865,132,865,159 ,825,159,825,132,8 730:PRINT " Der PC-16 1020:LPRINT "CL"
65,132 40,300 740:PRINT " FISCHEL ha 1030:GRAPH :GLCURSOR (0
300:DATA 925,132,925,1 59,685,159,885,132 500:DATA 940,352,940,4 t's mit SHARP !" 1040:LPRINT "Q W
,925,132 13,630,413,630,352 750:PRINT " Noch bess er mit PRINTER " 55,0,4
310:DATA 865,185,865,2 12,825,212,825,185 510:DATA 10,200,625,20 0,623,202,623,411, 770:GRAPH :GLCURSOR (- 1050:GRAPH :GLCURSOR (0
,865,185 12,411,12,202,623, 202 185,33):SORGN " U I O P
320:DATA 805,185,805,2 12,765,212,765,185 520:DATA 590,238,590,3 790:FOR Y=0TO 31:FOR X 1070:GRAPH :GLCURSOR (0
,805,185,745,185,7 85,25,385,25,238,5 =0TO 155:P=POINT ( :TEXT :CSIZE 1
45,212,705,212 90,238,588,240,588 ,383,27,383 800:IF P=1GOSUB "P" 1080:LPRINT " 1080:LPRINT "4 5
330:DATA 705,185,745,1 85,685,185,685,212 530:DATA 27,240,588,24 0,560,250,560,373, 55,373,55,250,560, 250 820:"L":RETURN 1090:GRAPH :GLCURSOR (0
340:DATA 615,165,615,1 90,580,190,580,165 540:LLINE (757,230)-(7 57,298):LLINE (500 840:A=X:LLINE (A*3+250 1100:LPRINT "MODE"
,615,165,555,165,5 55,190,520,190 550:LLINE (432,202)-(4 32,238):LLINE (362 850:"B":GRAPH : 1090:GRAPH :GLCURSOR (0
350:DATA 520,165,555,1 65,495,165,495,190 92,238):LLINE (222 202)-(222,238) 660:LPRINT "BREAK" 1110:GRAPH :GLCURSOR (0
,460,190,460,165,4 95,165 570:LLINE (152,202)-(1 52,238):LLINE (84, 870:GRAPH :GLCURSOR (0 ,5):TEXT :CSIZE 2
360:DATA 435,165,435,1 90,400,190,400,165 580:LLINE (80,250)-(80 345)-(535,345)-(5 35,250) 880:LPRINT "BS SH 1120:LPRINT "A S
,435,165,375,165,3 75,190,340,190 590:GLCURSOR (925,185) IFT POWER D F G H
370:DATA 340,165,375,1 65,315,165,315,190 0,3,B OFF ON" 1130:GRAPH :GLCURSOR (0 ,6):TEXT :CSIZE 1:
,280,190,280,165,3 15,165 600:GLCURSOR (0,395): 890:GRAPH :GLCURSOR (0 ,17):TEXT :CSIZE 1:
380:DATA 255,165,255,1 90,220,190,220,165 1:PCONSOLE "LPT1:",2 1:COLOR 3:PCONSOLE 1140:LPRINT "
,255,165,195,165,1 95,190,160,190 55,0,10:COLOR 0 900:LPRINT "!" [ ]
390:DATA 100,165,195,1 65,135,165,135,190 610:LPRINT "MEMORY SAF # DEL"
,100,190,100,165,1 35,165 E GUARD/AUTO POWER 1150:GRAPH :GLCURSOR (0 ,7):TEXT :CSIZE 1:
400:DATA 75,165,75,190 165,70,215,70,225, 45,225 OFF" % & PCONSOLE "LPT1:",2
410:DATA 45,215,70,215 410:DATA 45,215,70,215 910:GRAPH :GLCURSOR (0 55,0,86
,130,215,130,225,1 05,225,105,215,130 2):TEXT :CSIZE 1: 1160:LPRINT "<
,215 55,0,62:GRAPH : COLOR 0:PCONSOLE ^ >"
420:DATA 200,215,200,2 25,175,225,175,215 920:LPRINT "DEF" 1170:GRAPH :GLCURSOR (0 ,5):TEXT :CSIZE 2
,200,215,270,215,2 70,225,245,225 930:GRAPH :GLCURSOR (0 ,12):TEXT :CSIZE 1 1180:LPRINT "1 2
430:DATA 245,215,270,2 15,340,215,340,225 315,225,315,215,3 255,0,84 3 -"
40,215 650:"U":GLCURSOR (X,Y) 940:LPRINT "KBII" 1190:GRAPH :GLCURSOR (2
440:DATA 410,215,410,2 25,385,225,385,215 550:GRAPH :GLCURSOR (0 37,30):LLINE -(250
,410,215,480,215,4 660:"E":TEXT :PCONSOLE ,10):TEXT :CSIZE 1 ,25)-(250,35)-(237
80,225,455,225 "LPT1:",255,0,0 :COLOR 3:PCONSOLE ,30)
670:GRAPH :GLCURSOR (0 ,-17):TEXT :CSIZE "LPT1:",255,0,110 1200:GRAPH :GLCURSOR (0

```

```

,-10):TEXT :CSIZE 1240:GRAPH :GLCURSOR (0 1270:LPRINT "SHIFT
1:PCONSOLE "LPT1:" ,2):TEXT :CSIZE 1: SML
,255,0,5 COLOR 3:PCONSOLE " RCL SP
1210:LPRINT "CTRL" LPT1:",255,0,110 ACE"
1220:GRAPH :GLCURSOR (0 1250:LPRINT " ; 1280:GRAPH :GLCURSOR (1
,8):TEXT :CSIZE 2: - ; 0 32,8):LLINE -(127,
PCONSOLE "LPT1:",2 INS" 14)-(137,14)-(132,
55,0,8 1260:GRAPH :GLCURSOR (0 8)
1230:LPRINT "Z X , -10):TEXT :CSIZE 1290:GLCURSOR (127,18):
C U B N 1:COLOR 0:PCONSOLE LLINE -(137,18)-(1
M ( )" "LPT1:",255,0,5 32,24)-(127,18)
1300:COLOR 3:GLCURSOR (
274,37):LLINE -(27
9,42)-(284,37)
1310:COLOR 0:GLCURSOR (
373,10):LLINE -(38
8,15)-(371,15)-(37
1,23)-(375,23)
1320:GLCURSOR (375,23):
LLINE -(375,15)-(3
78,15)-(373,10)
1330:GLCURSOR (431,10):
LLINE -(431,17)-(4
28,17)-(433,23)-(4
38,17)
1340:GLCURSOR (438,17):
LLINE -(435,17)-(4
35,10)-(431,10)
1350:GRAPH :GLCURSOR (0
,3):TEXT :CSIZE 1:
PCONSOLE "LPT1:",2
55,0,98
1360:LPRINT "ENTER"
1370:GRAPH :GLCURSOR (0
,10):TEXT :CSIZE 2
:PCONSOLE "LPT1:",
255,0,55
1380:LPRINT "0
="
+"
1390:GRAPH :GLCURSOR (6
2,29):LLINE -(62,3
2)-(65,32)-(65,29)
-(62,29)
1400:GRAPH :COLOR 3:
GLCURSOR (60,19):
LLINE -(60,27)-(67
,27)-(67,19)-(60,1
9)
1410:GRAPH :COLOR 0:
GLCURSOR (238,-23)
:LLINE -(238,-13)-
(251,-18)-(238,-23
)
1420:TEXT :LF 7:
PCONSOLE "LPT1:",8
0,0,0:END

```



ES KOENNEN BIS ZU 12 ZEICHEN EINGEGEBEN WERDEN, WELCHE IN DEN OBEREN TEIL DES DISPLAYS GESCHRIEBEN WERDEN. DAS PROGRAMM GIBT DIE ZEICHEN DANN IN DOPPELTER GRÖSSE AUF DEM DISPLAY WIEDER, WAHLWEISE IN SCHRAEG- ODER NORMAL-SCHRIFT UND IN ZWEI VERSCHIEDENEN SCHRIFTTYPEN ANSCHLIESSEND ERFOLGT EINE HARDCOPY AUF DEM CE-140P ODER CE-515P. START MIT (DEF A) BEI START MIT (DEF B) BLEIBT DIE VORHERIGE EINGABE ERHALTEN UND ES KANN EINE NEUE SCHRIFTART GEWAHLT WERDEN. VIEL SPASS !!

```

1:"A" CLEAR
2:REM * EIN PROGRAMM FUER DEN PC-1350 *
3:REM * VON JUERGEN GARTINGER *
4:DIM A$(1)*13: WAIT 100
5:CLS : INPUT "BITTE ZEICHEN EINGEBEN (MAX 12)":A$(1)
6:L= LEN A$(1): IF L>12 CLS : BEEP 1: PRINT "ZU VIELE ZEICHEN": GOTO 5
7:"B" CLS : INPUT "SCHRAEGSCHRIFT (S) NORMALSCHRIFT (N) ":S$
8:CLS : INPUT "SCHRIFTTYP (1) SCHRIFTTYP (2) ":T$
10:CLS : WAIT 0: PRINT A$(1):W=0:Z=0
20:FOR Y=0 TO 6
25:W=W+0.5
30:FOR X=6 TO L*6+8
40:P= POINT (X,Y)
45:IF S$="S" LET Z=X-3-W: GOTO 50
47:Z=X-3
50:IF P=1 THEN PSET (Z*2+1,Y*2+10): PSET (Z*2+2,Y*2+11): PSET (Z*2+1,Y*2+11)
: IF T$="2" THEN GOTO 100
55:IF P=1 THEN PSET (Z*2+1,Y*2+11): PSET (Z*2+2,Y*2+10)
100:NEXT X
110:NEXT Y
115:BEEP 3: GOTO 125
120:I$= INKEY$: IF I$="" GOTO 120
125:CLOSE : OPEN : LPRINT CHR$(27)+"b":K=9
130:FOR J=0 TO 149:P= POINT (J,K)
133:IF P<>0 LPRINT "M":J*3:",":K*3: LPRINT "P."
135:NEXT J
136:K=K+1: IF K<24 THEN 130
140:BEEP 5: END

```

STATUS 1

6647

Edgar Härle

7951 Warthausen
Neherstraße 22

```

10:REM *** Umrechnung von Dezimalzahlen in Dualzahlen ***
11:REM *** von Edgar Härle ***
20:CLEAR :DIM R$(50):DIM R(50)
30:PRINT "Umrechnung von Dezimal-"
40:PRINT "zahlen in Dualzahlen ....."
50:PRINT :PRINT "Dezimalzahl = ";
60:INPUT D1
70:Z=1:D=D1
80:R(Z)=DMOD 2:R$(Z)=STR$ P(Z)
90:D=INT (D/2)
100:IF D=0THEN 130
110:Z=Z+1
120:GOTO 80
130:PRINT "Dezimalzahl =";D1
140:PRINT " Dualzahl = ";
150:FOR I=ZTO 1STEP -1
160:PRINT R$(I);
170:NEXT I
180:PRINT
190:IF Z<12THEN PRINT ""
200:INPUT " Weiter (J/N) ";A$
210:IF A$="N"THEN 220ELSE 10
220:CLS :PRINT " P R O G R A M M E N D E "
230:END

```

H A N G M A N

ein Spiel für den PC-1600

von Thomas Jeger

Bei diesem Spiel geht es darum, das vom Computer gedachte oder von einem anderen Spieler eingegebene Wort zu erraten. Man versucht alle Buchstaben, die im Wort vorkommen, zu erraten.

Tippen Sie das Programm ab und starten Sie es mit RUN. Sie werden jetzt gefragt, ob jemand das Wort eingeben will, oder ob es der Computer aus seiner Liste auswählen soll. Das Display wird gelöscht und das Spiel beginnt. Auf der ersten Zeile des Displays sehen Sie, den wievielten Versuch Sie starten. In der zweiten Zeile sehen Sie das versteckte Wort (zwischen den beiden schwarzen Punkten). In der 3. Zeile können Sie die Buchstaben eingeben und in der 4. Zeile sehen Sie die Buchstaben, welche Sie schon probiert haben. Dabei bedeuten grosse Buchstaben Versuche mit Erfolg und kleine Buchstaben Versuche, die zu einem Fehler geführt haben. Geben Sie nun Ihren ersten Versuch ein (Buchstabe und ENTER). Falls dieser Buchstabe im gesuchten Wort vorkommt, erscheint er an allen Stellen. Kommt er nicht vor, so sehen Sie je nach Fehleranzahl, einen Galgen mit einem Männchen dran. Beim achten Fehler ist es ganz gehängt und es wird Ihnen das gesuchte Wort gezeigt.

Die Wort-Liste am Schluss des Listings kann beliebig fortgesetzt werden. Es muss aber das Wort "ENDE" gelöscht werden und ganz an den Schluss der Liste gesetzt werden.

```

10:  >> Hangman <<
11:
12:  Ein Spiel von
13:
14:  THOMAS JEGER
15:
20: CLEAR :DIM A$(3)*25
30: CLS
40: PRINT "      >> HANGMAN <<
50: CURSOR 2,2: INPUT "Eigenes Wort ? (
    J/N):";A$
60: IF LEFT$(A$,1)<"J" THEN GOTO 100
70: CLS :INPUT "=";A$(0)
80: GOTO 130
100: R=RND(100):FOR I=1 TO R:READ A$(0)
110: IF A$(0)="ENDE" THEN RESTORE :READ
    A$(0)
120: NEXT I
130: CLS
140: FOR I=1 TO LEN A$(0)
150: IF MID$(A$(0),I,1)=" " THEN LET A$
    (1)=A$(1)+" " :GOTO 170
160: A$(1)=A$(1)+"_"
170: NEXT I
180: U=1:F=0
190: CLS
200: CURSOR 0,0: PRINT "VERSUCH Nr. ";U
210: PRINT CHR$(254);A$(1);CHR$(254)
220: CURSOR 0,3: PRINT A$(2)
230: CURSOR 0,2: INPUT "=";A$:A$=LEFT$(
    A$,1)
240: IF INSTR(A$(2),A$)<>0 THEN 230
250: IF INSTR(A$(2),CHR$(ASC A$+32))<
    >0 THEN 230
260: IF A$="*" THEN "I"
270: U=U+1: IF INSTR(A$(0),A$)=0 THEN
    LET A$(2)=A$(2)+CHR$(ASC A$+32):F
    =F+1:GOTO "F"
280: A$(2)=A$(2)+A$
290: I=0
300: I=INSTR(I+1,A$(0),A$)
310: IF I=0 THEN 340
320: A$(1)=LEFT$(A$(1),I-1)+A$+RIGHT$(
    A$(1),LEN A$(1)-I)
330: GOTO 300
340: IF A$(1)=A$(0) THEN LET U=U-1:GOTO
    "E"
350: GOTO 200
400: "F"CLS
410: LINE (80,31)-(80,26):LINE -(75,26)
    :LINE -(75,21)
420: LINE -(70,21):LINE -(70,16):LINE -(
    65,16):LINE -(65,31)
430: IF F=1 THEN GOTO 600
440: LINE (65,16)-(65,0)
450: IF F=2 THEN GOTO 600
460: LINE -(90,0)
470: IF F=3 THEN GOTO 600
480: LINE (70,0)-(65,5)
490: LINE (90,0)-(90,5)
500: IF F=4 THEN GOTO 600
510: LINE (89,5)-(91,0),,BF
520: IF F=5 THEN GOTO 600
530: LINE (90,8)-(90,15)
540: IF F=6 THEN GOTO 600
550: LINE (90,10)-(94,7)
560: LINE (90,10)-(86,7)
570: IF F=7 THEN GOTO 600
580: LINE (90,15)-(94,18)
590: LINE (90,15)-(86,18)
600: FOR I=1 TO 800: NEXT I
610: CLS :PRINT "Das Wort wäre gewesen:
    ":CURSOR 0,2
620: PRINT CHR$(254);A$(0);CHR$(254)
630: IF INKEY$(1)=" " THEN 630
640: GOTO 10
680: FOR I=1 TO 500: NEXT I:GOTO 190
700: "I"CLS
720: INPUT "=";A$(3)
730: IF A$(0)=A$(3) THEN GOTO "E"
740: F=F+1:GOTO "F"
800: "E"CLS
810: PRINT "***** BRAVO ***"
820: CURSOR 0,2: PRINT " in ";U;" Versuc
    hen !"
830: GOTO 830
1000: DATA "RASENMAEHER","SCHAUKESTUHL"
    ,"BUCHBESCHWERER","POSAUNE","TISCH
    LAMPE"
1010: DATA "TELEFONMAST","TELEFONHOERER"
    ,"FERNSEHANTENNE","SCHIFF","DAMPFS
    CHIFF"
1020: DATA "ZEICHENSATZ","SHARP","INTERR
    UPT","ENDE"

```



```

10:REM *** Überweisungsformulare schreiben ***
11:REM *** von Edgar Härle ***
20:CLEAR :DIM BA$(0)*35:DIM UE$(2)*26:DIM AN$(2)*26
30:CLS :PRINT "Überweisungsformulare"
40:PRINT "Anschrift des Empfängers"
50:FOR I=1TO 2
60:PRINT I;" . Zeile (max.26 Zeichen)"
70:INPUT AN$(I)
80:NEXT I
90:PRINT :PRINT "Bank des Empfängers"
100:INPUT BA$(0)
110:INPUT "Bankleitzahl ";B$
120:INPUT "Konto-Nr. ";K$
130:PRINT :PRINT "Verwendungszweck"
140:FOR I=1TO 2
150:PRINT I;" . Zeile (max. 26 Zeichen)"
160:INPUT UE$(I)
170:NEXT I
180:PRINT :INPUT "DM-Betrag ";DM
181:L=INT (DM):R=INT ((DM-L)*100)
182:L$=STR$(L):R$=STR$(R)
183:IF R<10THEN LET DM$=L$+" ,0"+R$:GOTO 190
184:DM$=L$+" ,"+R$
190:PRINT :INPUT "Datum ";DA$
200:CLS :PRINT "Überweisungsformular in"
210:PRINT "Drucker bis z. Abrißkante"
220:PRINT "einspannen"
230:INPUT "Ausdruck (J/N) ";U$
240:IF U$="J"OR U$="j"THEN 250ELSE CLS :PRINT " E N D E":END

```

Dieses Programm ist geeignet zum Schreiben von Euro-Schecks

mit dem PC-1600 + CE - 1600 P.

Der Computer fragt den Benutzer zuerst nach folgenden Daten:

Danach sollten Sie einen Euro-Scheck in den Drucker CE-1600 P bis zur Abrißkante einspannen.

Nach der Bestätigung mit "J" erfolgt der Ausdruck.

Der Punkt zur Trennung von DM-und Pfennig-Beträgen wird beim Ausdruck automatisch durch ein Komma ersetzt.

Das Unterprogramm zur Umwandlung von ganzzahligen Zahlen in einen Text, entstammt aus dem Heft 23, Seite 43

- Alles für Sharp Computer -.

Sollten Sie den Text "Nur zur Verrechnung" nicht wünschen, so lassen Sie einfach die Zeilen 240 - 260 weg.

```

200:GLCURSOR (531,-56):LPRINT DM$
210:GLCURSOR (60,-90):CSIZE 2:LPRINT E$(0)
220:GLCURSOR (470,-90):LPRINT O$
230:GLCURSOR (470,-130):LPRINT DA$
240:GLCURSOR (60,-180):CSIZE 3:LPRINT "Nur zur Verrechnung !"
250:GLCURSOR (61,-181):LPRINT "Nur zur Verrechnung !"
260:LLINE (60,-188)-(430,-188),0
270:TEXT :CSIZE 2:LF 8
280:GOTO 90
1000:REM **** Zahlen in Worte darstellen ****
1001:DATA "ein","zehn","elf"
1002:DATA "zwei","zwanzig","zwölf"
1003:DATA "drei","dreissig","dreizehn"
1004:DATA "vier","vierzig","vierzehn"
1005:DATA "fünf","fünfzig","fünfzehn"
1006:DATA "sechs","sechszig","sechzehn"
1007:DATA "sieben","siebzig","siebzehn"
1008:DATA "acht","achtzig","achtzehn"
1009:DATA "neun","neunzig","neunzehn"
1230:DD$=STR$(INT (DM))
1240:IF LEN (DD$)<6LET DD$=LEFT$( "000000",6-LEN (DD$))+DD$
1250:FOR I=0TO 1

```

```

250:CLS :PRINT " A U S D R U C K"
260:PRINT :PRINT " Bitte warten !"
270:GRAPH :COLOR 0:CSIZE 2
280:GLCURSOR (0,-50):LPRINT AN$(1)
290:GLCURSOR (0,-72):LPRINT AN$(2)
300:GLCURSOR (500,-61):LPRINT B$
310:GLCURSOR (0,-120):LPRINT K$
320:GLCURSOR (100,-120):LPRINT BA$(0)
330:GLCURSOR (0,-175):LPRINT UE$(1)
340:GLCURSOR (0,-200):LPRINT UE$(2)
350:GLCURSOR (475,-198):CSIZE 4:LPRINT DM$
360:GLCURSOR (476,-199):LPRINT DM$
370:GLCURSOR (477,-200):LPRINT DM$
380:GLCURSOR (0,-260):CSIZE 2
390:LPRINT "10 222 000 Karl Mayer, Bahnhofstr. 00, 1000 Berlin"
400:GLCURSOR (200,-330):LPRINT DA$
410:TEXT :LF 8
420:GOTO 200

```

1. DM-Betrag?
2. Empfänger?
3. Ausstellungsort?
4. Datum?

```

10:REM **** Euro-Scheck schreiben ****
11:REM **** von Edgar Härle ****
20:CLEAR :DIM E$(0)*40:DIM D$(0)*40
30:CLS :PRINT "Euro-Scheck schreiben"
40:PRINT :PRINT "DM-Betrag ";:INPUT DM
50:PRINT :PRINT "Empfänger ";:INPUT E$(0)
60:PRINT :INPUT "Ort: ";O$
70:INPUT "Datum: ";DA$
80:CLS :PRINT "Euro-Scheck in"
90:PRINT "Drucker bis z. Abrißkante"
100:PRINT "einspannen !"
110:INPUT "Ausdruck (J/N) ";U$
120:IF U$="J"OR U$="j"THEN 130ELSE
CLS :PRINT " E N D E":END
130:CLS :PRINT " A U S D R U C K"
140:PRINT :PRINT " Bitte warten"
150:GRAPH :COLOR 0:CSIZE 2
160:GOSUB 1000
170:GLCURSOR (35,-55):LPRINT D$(0)
180:GLCURSOR (480,-55):LPRINT "DM"
181:L=INT (DM):R=INT ((DM-L)*100)
182:L$=STR$(L):R$=STR$(R)
183:IF R<10THEN LET DM$=L$+" ,0"+R$:GOTO 190
184:DM$=L$+" ,"+R$
190:GLCURSOR (530,-55):CSIZE 3:LPRINT DM$

```

```

1340: IF I=0 AND (X1$)"0" OR X2$)"0" OR X3$)"0") LET D$(0)=D$(0)+"tausend"
1350: NEXT I
1420: RETURN
1510: RESTORE (VAL (X1$)+1000): READ XX$
1520: D$(0)=D$(0)+XX$+"hundert"
1530: RETURN
1610: RESTORE (VAL (X3$)+1000): READ XX$
1620: D$(0)=D$(0)+XX$+"und"
1630: RESTORE (VAL (X2$)+1000): READ XX$: READ XX$
1640: D$(0)=D$(0)+XX$
1650: RETURN
1710: RESTORE (VAL (X2$)+1000): READ XX$: READ XX$
1720: D$(0)=D$(0)+XX$
1730: RETURN
1810: RESTORE (VAL (X3$)+1000): READ XX$: READ
      XX$: READ XX$
1820: D$(0)=D$(0)+XX$
1840: RETURN
1910: RESTORE (VAL (X3$)+1000): READ XX$
1920: D$(0)=D$(0)+XX$
1930: IF X3$="1" AND I=1 LET D$(0)=D$(0)+"s"
1940: RETURN

```

Raiffeisenbank

Öffnen Sie gegen Ihren Schenk
 dreihundertelundfünfzig
 Betrag in Buchstaben
FISCHEL GmbH, 1000 Berlin
 über Überbringer

Konto-Nr.	Betrag	351779
Ort		Warthausen
Datum		19.03.1987

Nur zur Verrechnung !

0000000231776 005023 654616

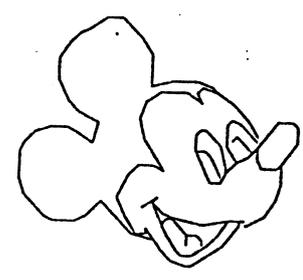
SCANNER

Thomas Jeger
 Hauptstrasse 142
 CH-3286 Murtelier
PC-1600

Endlich kann man mit dem PC-1600 Programm für folgende Konfiguration:
 Grafiken einlesen !!!!!

- Funktionsweise:**
- PC-1600
 - CE-1600P
 - CE-16COF oder RAM-Floppy
- Tippen Sie das Programm ab und starten Sie es mit RUN.
- Im Hauptmenue wählen Sie die Funktion 1 (Einlesen) an.
- Sie werden nun aufgefordert, das Original in den Drucker zu legen. Nehmen Sie also das Druckpapier aus dem CE-1600P und legen Sie das Bild, welches Sie einlesen wollen in der Drucker. Drücken Sie irgend eine Taste um ins Funktionsmenue zu gelangen. Dort angekommen, haben Sie folgende Funktionen:
- 8- Schreibkopf hinauf
 - 4- Schreibkopf links
 - 5- Punkt speichern
 - 0- Schreibkopf 50 mal links
 - Schreibkopf 50 mal runter
 - MODE- : Die Einheit Schritte in die jeweilige Richtung (Tasten 2,4,6,8) wird von 1 auf 5 gesetzt resp. umgekehrt.
 - SHIFT-: Abspeichern des eingelesenen Bildes.
 - 2- Schreibkopf hinunter
 - 6- Schreibkopf rechts
 - *- Punkt speichern (ohne Menue)
 - Schreibkopf 50 mal rechts
 - +- Schreibkopf 50 mal hinauf

Beispiele:



Um einen Punkt einzulesen gehen Sie also mit dem Schreibkopf des Druckers an die betreffende Stelle. Verschieben Sie den Druckkopf so lange, bis der kleine rote Strich (auf dem transparenten Schutz) genau auf der Punkt im Bild zeigt. Ist dies getan, drücken Sie die 5-Taste. Es erscheint ein Menue. Da es sich hier um den ersten Punkt des Bildes handelt, geben wir "02" ein. Die erste Zahl beschreibt die Farbe (C für Schwarz, 1 für blau, 2 für grün und 3 für rot). Die 2. Zahl beschreibt den weiteren Verlauf mit diesen Koordinaten. Dabei bedeutet die 0: Ziehe eine Linie von der letzten Schreibkopfposition zu den Koordinaten dieses Punktes. Die 1 bedeutet: Betrachte die letzte Schreibkopfposition und die Koordinaten dieses Punktes als Endpunkte einer Diagonale und schreibe ein Rechteck ein. Die 2 bedeutet: Setze den Schreibkopf an diese Stelle und zeichne nichts.

Sie sehen, das Prinzip ist eigentlich recht einfach und es lassen sich so recht schöne Bildchen erstellen. Nachdem Sie alle Punkte des Bildes eingelesen haben, drücken Sie die SHIFT-Taste. Sie können nun dem Bild einen Titel geben unter dem der Computer das Bild abspeichert. Sie sind jetzt wieder im Hauptmenue und können den 2. Menue-Punkt (ausdrucken) anwählen. Zuerst werden Sie aufgefordert, wiederum Druckpapier einzuspannen und danach den Titel des Bildes einzugeben. Der Drucker gibt nun das eingegeben aus.

```

10: SCANNER
11:
12: Programm von Thomas Jeger
14:
15: (c) 1987
20:CLS
30:CURSOR 0,0:PRINT "Hauptmenue"
40:CURSOR 2,2:PRINT "(1) Einlesen"
42:CURSOR 2,3:PRINT "(2) Ausdrucken"
60:A$=INKEY$
70:IF A$="1"THEN 100
72:IF A$="2"THEN 1000
80:GOTO 60

```

```

100:CLS :GRAPH
110:PRINT "Bitte spannen Sie das Original ein !"
120:CURSOR 10,3:PRINT "(TASTE)"
130:CALL &166
140:CLS
200:CLEAR :DIM X(200),Y(200),Z$(200)*2:M=1:I=1:Z$(0)="00
210:CURSOR 0,0:PRINT "X:";X;" " :CURSOR 10,0:PRINT "Y:";Y
220:PRINT "Punkt Nr.:";I
230:PRINT "(MODE) : Schnell/langsam"
240:PRINT "(SHIFT): Save"
250:A$=INKEY$
260:IF A$=CHR$ 31AND M=5THEN LET M=1:GOTO 250
270:IF A$=CHR$ 31AND M=1THEN LET M=5:GOTO 250
280:IF A$="4"THEN LET X=X-M
290:IF A$="6"THEN LET X=X+M
300:IF A$="8"THEN LET Y=Y+M
310:IF A$="2"THEN LET Y=Y-M
312:IF A$="*"THEN GOTO 378
315:IF A$="5"THEN GOTO 350
317:IF A$="-"THEN LET Y=Y+50
320:IF A$="+"THEN LET Y=Y-50
322:IF A$="0"THEN LET X=X-50
324:IF A$="."THEN LET X=X+50
330:IF A$=CHR$ 1THEN 500
340:GLCURSOR (X,Y)
345:GOTO 210
350:CLS :BEEP 1
360:PRINT " --- Auswahl ---"
370:PRINT "0+:Schwarz +0:Linie
372:PRINT "1+:Blau +1:Box
374:PRINT "2+:Grün / 3+:Rot +2:Cursor
376:CALL &166
378:CLS
380:INPUT "==" ;Z$:IF LEN Z$>20R Z$=""THEN 380
390:IF LEN Z$=1THEN LET Z$(I)=LEFT$(Z$(I-1),1)+Z$:GOTO 400
395:Z$(I)=Z$
400:X(I)=X:Y(I)=Y:I=I+1:CLS :GOTO 210
500:CLS :INPUT "Dateiname:";N$:MAXFILES =1
510:OPEN "S1:"+N$+".SCN"FOR OUTPUT AS #1
520:FOR J=1TO I:PRINT #1,X(J):PRINT #1,Y(J):PRINT #1,Z$(J)
530:BEEP 1,J,30:NEXT J
540:CLOSE #1:GOTO 10
1000:CLS :CLEAR :GRAPH
1005:DIM X(200),Y(200),Z$(200)*2
1010:PRINT "Bitte normales Papier einlegen !"
1020:CALL &166
1030:CLS
1040:INPUT "Dateiname:";N$
1050:OPEN "S1:"+N$+".SCN"FOR INPUT AS #1
1060:IF EOF (1)THEN CLOSE #1:GOTO 10
1070:INPUT #1,X:INPUT #1,Y
1090:INPUT #1,Z$
1100:F=VAL LEFT$(Z$,1)
1110:W$=RIGHT$(Z$,1)
1120:IF W$<"0"AND W$<"1"THEN GLCURSOR (X,Y):GOTO 1190
1130:IF W$="0"THEN LLINE -(X,Y),0,F:GOTO 1190
1140:IF W$="1"THEN LLINE -(X,Y),0,F,B
1190:GOTO 1060

```

Hans Hölzel
 Jahnstr. 9
 6101 Fränk.-Crumbach
 Sehr geehrter Herr Fische!;

heute möchte ich Ihnen ein Programm zur Verfügung stellen, welches es auf sehr komfortable Weise erlaubt, Bilder auf dem Display des PC-1600 zu erstellen, diese auf Diskette zu speichern und mittels eines einfachen Hardcopy Programmes zu Papier zu bringen. Gestartet wird das Programm mit RUN bzw. mit DEF A, wenn ein Bild nicht gelöscht werden soll.

Da das Programm vollkommen menügesteuert ist, ist eine umfassende Erklärung eigentlich nicht erforderlich. Trotzdem möchte ich noch einige Hinweise geben.

Nach dem Start erscheint das Hauptmenue. Darin bedeutet

ZEI	Eingabe eines neuen Bildes oder auch ändern eines Bildes
SCH	Hardcopy-Ausdruck und Berechnung des HEX-Codes
SAU	Speichern des Bildes auf Diskette
LOA	Laden des Bildes von Diskette
PRI	Darstellung des Bildes auf dem Display
END	Programmende

Nach Aufrufen von ZEI muss man zuerst die Frage beantworten, ob man ein neues Bild zeichnen will, oder ob man eines ändern will.

Bei Eingabe von "N" bekommt man den Hinweis, dass man bei Eingabe von "I" das dargestellte Bild inventieren und bei Eingabe von "L" den Zeichen-Cursor wandern lassen und damit Punkte löschen kann. Mit ENTER kommt man wieder in den Zeichen-Modus zurück und bei nochmaligem ENTER zurück zum Hauptmenue.

Die Menüpunkte SCH SAU und LOA bedürfen keiner weiteren Erklärung. Hat man über PRI ein Bild auf das Display gebracht, so kann man mittels "C" Änderungen an dem Bild vornehmen. Der Zeichen-Cursor wird mit den Ziffern-Tasten gesteuert.

Die Berechnungsroutine für den HEX-Code habe ich übrigens aus Heft 12:42 entnommen und lediglich auf das 4-Zeillendisplay umgewandelt.

```

1:REM *****
2:REM *** E D I T O R ***
3:REM *** Hans Hölzel ***
4:REM * 6101 Fr.-Crumbach *
5:REM *** Jahnstr. 9 ***
6:REM *****
7:REM *****
10:CLS :CLEAR :MAXFILES =1
15:ON ERROR GOTO 8030
20:GOSUB "DIM"

```

```

30:"A"CLS :WAIT 0:PRINT :PRINT :PRINT :PRINT " ZEI SCH SAU LOA PRI END "
40:"MENUE"AB=ASC INKEY$ -16
50:CURSOR 0,1:PRINT K$(0)
60:IF XX=1BEEP 3:FOR I=0TO 60STEP 1:NEXT I:XX=0:K$(0)=K$(7):K$(7)="" :BEEP 1:
GOTO 50
70:WAIT 0:ON ABGOTO 90,490,890,1020,1170,1340
80:GOTO "MENUE"
90:CLS :PRINT "Neues Bild: (N)          Ändern:          (C)"
100:A$=INKEY$ :IF A$<>"N"AND A$<>"C"GOTO 100
110:IF A$="N"ERASE L$(0),P$(0),K$(0),TE$(0):GOSUB "DIM":GOTO 170
120:IF A$="C"CLS :PRINT "Bild laden von Disk. (1) Bild ber. im Speicher (0) "
130:A$="" :A$=INKEY$ :IF A$=""GOTO 130
140:IF A$="0"GOTO 1170
150:IF A$<"0"OR A$>"1"BEEP 1:A$="C":CLS :GOTO 120
160:GOTO 1020
170:CLS :PRINT "I = Inventieren          "
185:PRINT "L = Löschen (wandern) des      Punktes"
190:WAIT 170:PRINT "ENTER => Menue"
200:WAIT 0:BEEP 2,60,200:BEEP 1,100,350:CLS :E=ER
220:X=0:Y=0:PSET (X,Y):P$(X,Y)="1"
230:GOSUB 260
240:BEEP 1
250:GOTO "MENUE"
260:A$=INKEY$ :IF A$=""GOTO 260
265:IF A$="L"GOSUB "WANDERN"
280:IF A$="I"GOSUB "INVERT"
290:IF A$=CHR$ 80GOTO 30
300:IF A$="1"LET X=X-1:Y=Y+1:GOSUB 420:GOTO "P"
310:IF A$="2"LET Y=Y+1:GOSUB 420:GOTO "P"
320:IF A$="3"LET X=X+1:Y=Y+1:GOSUB 420:GOTO "P"
330:IF A$="4"LET X=X-1:Y=Y-1:GOSUB 420:GOTO "P"
340:IF A$="0"OR A$="5"GOTO 260
350:IF A$="6"LET X=X+1:GOSUB 420:GOTO "P"
360:IF A$="7"LET X=X-1:Y=Y-1:GOSUB 420:GOTO "P"
370:IF A$="8"LET Y=Y-1:GOSUB 420:GOTO "P"
380:IF A$="9"LET X=X+1:Y=Y-1:GOSUB 420:GOTO "P"
400:"P"PSET (X,Y):P$(X,Y)="1":GOTO 260
420:IF X<=0LET X=0
430:IF X=ELET E=E+1:ER=E
440:IF X=155LET X=155:E=X
450:IF Y<=0LET Y=0
460:IF Y=31LET Y=31
470:RETURN
490:IF E<=0LET XX=1:K$(7)=K$(0):K$(0)=K$(5):AB=0:GOTO 30
491:CLS :PRINT "Normal          (0)
Invertiert (1)"
493:A$=INKEY$ :IF A$=""GOTO 493
494:IF A$="1"LET IN=1
495:IF A$="0"LET IN=0

500:GRAPH :SORGN :LLINE (250,0)-(718,-95),,,B:LLINE (245,5)-(723,-100),,,B
510:FOR Y=0TO 31
520:FOR X=0TO E
530:IF IN=0AND P$(X,Y)="1"GOSUB 570
532:IF IN=1AND P$(X,Y)=""LET P$(X,Y)="1":GOSUB 570:P$(X,Y)=""
540:NEXT X
550:NEXT Y
560:GLCURSOR (190,-120):GOTO 600
570:L=X
580:LLINE (L*3+250,-Y*3)-(L*3+252,-2-Y*3),,,B
590:RETURN
600:TEXT :CLS
605:IF TE$(0)GOTO 630
610:PRINT "Text : ";INPUT TE$(0):IF TE$(0)GOTO 630
620:BEEP 3:CLS :GOTO 610
630:LCURSOR 21:LPRINT "Grafik : ";TE$(0)
640:LF 1
650:A=4:B=7:A1=0:B1=3:J=0:I=0:Z=-1
660:L$(I)="" :FOR Q=JTO E
670:S=.5:SS=0
680:FOR W=A TO B
690:S=S*2
700:IF P$(Q,W)=""AND IN=1LET SS=SS+S:GOTO 710
710:IF P$(Q,W)="1"AND IN=0LET SS=SS+S
710:NEXT W
720:B$=STR$(SS):IF SS>9LET B$=CHR$(55+SS)
730:S=.5:SS=0
740:L$(I)=L$(I)+B$
750:FOR W=A1 TO B1
760:S=S*2
770:IF P$(Q,W)=""AND IN=1LET SS=SS+S:GOTO 780
771:IF P$(Q,W)="1"AND IN=0LET SS=SS+S:GOTO 780
780:NEXT W
790:B$=STR$(SS):IF SS>9LET B$=CHR$(55+SS)
800:L$(I)=L$(I)+B$
810:IF LEN L$(I)=80AND Q<ELET J=Q:I=I+1:GOTO 660
820:IF LEN L$(I)<80AND J=ENEXT Q
830:NEXT Q
840:CSIZE 1:Z=Z+1

850:FOR AA=0TO I:LCURSOR 33:LPRINT "Zefle";Z;"; ";L$(AA):NEXT AA:LF 1
860:A=A+8:B=B+8,A1=A1+8:B1=B1+8
870:IF A1=32AND B1=35BEEP 3:LF 6:CSIZE 2:GOTO 30
880:I=0:J=0:GOTO 660
890:CLS :X$="X:"
891:IF TE$(0)=""PRINT "Text : ";INPUT TE$(0):GOTO 910
892:IF TE$(0)<>"PRINT "Text : ";TE$(0);" lassen ?":PRINT "(N/...)"
893:A$=INKEY$ :IF A$=""GOTO 893
894:IF A$<>"N"GOTO 920
895:IF A$="N"LET TE$(0)="" :INPUT "Text ? ";TE$(0)
896:IF TE$(0)=""CLS :GOTO 894
910:REM SPEICHERN
920:CLS :PRINT K$(3);TE$(0)
930:OPEN X$+TE$(0)FOR OUTPUT AS #1
940:PRINT #1,E
950:FOR X=0TO E
960:FOR Y=0TO 31
970:PRINT #1,P$(X,Y)
980:NEXT Y

```

Und nun wünsche ich viel Verknügen beim Abtippen und Zeichnen




```

66:GPRINT "3058581000703080810
67:GCURSOR 62,23:GPRINT "3F4020403F0030585810007F000
87F080000003F4848330018A0A0780028
68:WAIT 150:PRINT :WAIT 0:CLS :LINE (0,0)-(155,31),,
,B:GCURSOR 35,6
69:GPRINT "3F4020403F0070007000087000304848700070000
87003048483F00000007F080077";
70:GPRINT "007F0030484870007000081000000026494932003
0484830007E09097E00003040204038";
71:GPRINT "003048487000700008100030585810
72:GCURSOR 4,18:GPRINT "7E090976003040483000FE014936
007F10106003048487000700070007000";
73:GPRINT "F048483000405028000007F08007000081000400
00022414936003E41413E000018100000";
74:GPRINT "7E41413E0000000000";
75:GPRINT "22414936003E41413E003E41413E003E41413E000
0007F00097F003048487000700008070";
76:GPRINT "00700008700030484830003640201800305858100
0700008100000036494936007F

```

```

77:WAIT 10:LINE (34,14)-(120,14):LINE (4,27)-(151,27
):GOSUB 78:GOSUB 78:PAUSE :GOSUB 11:END
78:WAIT 0:FOR Z=0TO 200:LINE (2,0)-(2,31),X:LINE (2-
45,0)-(2-45,31),X:NEXT Z:RETURN
79:END
80:IF P<10AND P<11AND P<17AND P<18AND P<19
RETURN
81:IF W=1RETURN
82:IF P>16AND P<20GOTO 85
83:IF P=10LET SP=0:X2=X:GOSUB 93:RETURN
84:IF P=11LET SP=1:X2=X:GOSUB 92:RETURN
85:IF P=17AND X<25LET K=13:GOTO 89
86:IF P=18AND X>50AND X<50LET K=37:GOTO 89
87:IF P=19AND X>50AND X<75LET K=62:GOTO 89
88:S=S-2:W=1:RETURN
89:LINE (K,28)-(X,Y):GCURSOR X,Y:WAIT 10:GPRINT "7E8
388545112517E":WAIT 0:S=S+2:GOTO 91
90:S=S-10:FOR B=50TO 100STEP 3:BEEP 1,B,75:NEXT B:
CURSOR 9:PAUSE "Treffer!":RETURN
91:CURSOR 9,1:PRINT "Treffer!":FOR B=1TO 10:BEEP 1,5
0,75:BEEP 1,75,75:NEXT B:GOTO 55
92:GCURSOR 0,24:GPRINT L$:GCURSOR 0,16:GPRINT L$:
CURSOR 0,8:GPRINT L$:GCURSOR 0,0:GPRINT "00FFFB
F80":RETURN
93:GCURSOR 0,0:GPRINT L$:GCURSOR 0,8:GPRINT L$:
CURSOR 0,16:GPRINT L$:GCURSOR 0,24:GPRINT "3EFCB
81000":RETURN

```

Thomas Jeger
Hauptstrasse 142
CH-3286 Muntelier

Diskman

Ein Diskettenverwaltungsprogramm für den PC-1600 und CE-1600F. Mit diesem Programm lassen sich einzelne Files schnell kopieren, umbenennen, schützen oder löschen. Da sich das Programm von selbst erklärt (= sehr bedienungs-freundlich) nur noch schnell das Wichtigste in Kürze. Im Hauptmenue wählen Sie die gewünschte Funktion an (Nummer, die vor der Funktion steht drücken). "Init" steht hier für "Diskette formatieren". Bei allen anderen Funktionen müssen Sie vorerst das File auswählen, welches Sie behandeln möchten. Wählen Sie also die gewünschte Funktion an. Die Namen der Files erscheinen. Unterhalb dem ersten File sieht man einen Strich. Dieser lässt sich mittels den Tasten "4" und "6" entweder nach links oder nach rechts verstellen. Wählen Sie " " an, so wird im In-haltsverzeichnis der Diskette weitergewandert, d.h. die nächsten Filenamen werden angezeigt. Haben Sie das richtige File gefunden, drücken Sie ENTER. Wollen Sie die Funktion abbrechen, drücken Sie SPACE. Achtung: Gelöschte Files werden trotzdem ange-zeigt, sind aber nicht anwählbar. Gelöschte Files erkennt man daran, dass ihr erstes Zeichen ein griechisches Rho darstellt.

PC-1600,
CE-1600F

```

5:ON ERROR GOTO 2000 Falls Sie Interesse am Programm haben, aber keine Lust es ein-
10:GOTO 190 zugeben, können Sie es für DM 39.- (inkl. 14% MwSt.) bei der
20:'INVERS 1.LINIE Fischel GmbH abgespeichert auf Diskette bestellen.
30:LINE (0,0)-(156,7),X,,BF:RETURN
40:'DATEN FUER LINIEN
51:DATA 0,15,72,15
52:DATA 78,15,150,15
53:DATA 0,23,72,23
54:DATA 78,23,150,23
55:DATA 0,31,72,31
56:DATA 78,31,150,31
60:'LOESCHEN DER 3 ZEILEN
62:CURSOR 0,1:PRINT "
"
63:CURSOR 0,2:PRINT "
"
64:CURSOR 0,3:PRINT "
":CURSOR 0,1:RETURN
70:'WARTEN AUF EINE TASTE
75:IF INKEY$(1)<>" " THEN RETURN ELSE
75
99:'FILES" EINLESEN
100:CLEAR
110:IF (INP 87AND 8)=0 THEN CURSOR 0,1
:PRINT "BITTE QUELL-DISKETTE EIN-
LEGEN !!":GOTO 110
115:K$=CHR$ 10+CHR$ 10+CHR$ 10+CHR$ 10
+CHR$ 10+CHR$ 10+CHR$ 10+CHR$ 10+
CHR$ 10+CHR$ 10:KBUFF$ =K$+K$+K$
120:FILES "X":CLS :KBUFF$ =""
130:DIM N$(20)
140:I=1:J=60375
150:FOR K=0TO 10:N$(I)=N$(I)+CHR$ PEEK
(J+K):NEXT K
160:N$(I)=LEFT$(N$(I),8)+" "+RIGHT$(
N$(I),3)
170:IF LEN N$(I)>12 THEN LET AF=I-1:
RETURN
180:I=I+1:J=J+32:GOTO 150
190:CLS :CLEAR
200:PRINT " Diskettenverwaltung":
GOSUB 20
210:PRINT "1)Kopieren 2)Rename 3
)Schutz 4)Löschen"
220:PRINT "5)Init 6)Ende
230:A$=INKEY$(1):IF A$<"0"OR A$>"6"
THEN 230
240:ON VAL A$GOTO 300,700,800,900,1000
,250
250:CLS :END
300:REM KOPIEREN
310:CLS :GOSUB 100
320:CLS :PRINT " Kopieren":
GOSUB 20:GOSUB 330:GOTO 500
330:FOR I=1TO AFSTEP 5
332:CURSOR 0,1:PRINT "
"
333:CURSOR 0,2:PRINT "
"
334:CURSOR 0,3:PRINT "
":CURSOR 0,1
340:IF AF<I+4 THEN LET K=(AF-I)MOD 5
ELSE LET K=4
350:CURSOR 0,1:FOR J=0TO K:PRINT N$(I+
J);" ";NEXT J

```

```

360:IF K=4AND K<>AF-1THEN PRINT ">>>"
370:AU=1
380:RESTORE (50+AU)
390:READ X1,Y1,X2,Y2
400:LINE (X1,Y1)-(X2,Y2)
410:AS=INKEY$(1):IF AS<>CHR$(13AND AS<>
  "<>4"AND AS<>"6"AND AS<>)" THEN 41
  0
415:LINE (X1,Y1)-(X2,Y2),X
420:IF AS="4"AND AU>1THEN LET AU=AU-1:
  GOTO 380
425:IF AS=" " THEN 190
430:IF AS=CHR$(13)THEN GOTO 470
435:IF K<>4AND K<>AF-1THEN 445
440:IF AS="6"AND AUK=K+1THEN LET AU=AU
  +1:GOTO 380
445:IF AS="6"AND AUK<K+1THEN LET AU=AU+
  1:GOTO 380
450:GOTO 380
470:IF K<>AF-1AND AU=6THEN NEXT I
471:NT$=N$(I+AU-1)
474:IF INSTR (NT$," ")=0THEN 490
475:IN=INSTR (NT$," "):NT$=LEFT$(NT$,
  IN-1)+RIGHT$(NT$,LEN NT$-IN):GOTO
  474
490:RETURN
500:GOSUB 60
510:PRINT "Medium:"
515:PRINT "1) Disk Y 2)Kassette"
520:AS=INKEY$
525:IF AS="1"THEN LET T$="Y":GOTO 550
530:IF AS="2"THEN LET T$="CAS":GOTO 5
  50
535:IF AS=" " THEN GOTO 190
540:GOTO 520
550:COPY ("X:"+NT$)TO (T$+NT$)
560:GOTO 190
700:REM RENAME
710:CLS :GOSUB 100
720:CLS :PRINT " Rename":
  GOSUB 20:GOSUB 330
730:GOSUB 60:N$=""
740:CURSOR 0,2:KBUFF$ =CHR$(13)
742:INPUT "Neuer Name:");N$
745:IF LEN N$>8OR N$=""THEN CURSOR 0,2
  :GOTO 742
750:CURSOR 0,2:INPUT "Neues Ext :");EX
  $
752:IF LEN EX<>3THEN CURSOR 0,2:GOTO
  750
755:NT$="X:"+NT$
760:NAME (NT$)AS (N$+"."+EX$)
770:GOTO 190
800:REM SCHUTZ
810:CLS :GOSUB 100
820:CLS :PRINT " Schutz":
  GOSUB 20:GOSUB 330
830:GOSUB 60
835:CURSOR 0,2:PRINT "1) setzen 2) en
  tfernen
840:AS=INKEY$
845:IF AS=" " THEN 190
850:IF AS="1"THEN LET P$="P":GOTO 870
860:IF AS="2"THEN LET P$=" ":GOTO,870
865:GOTO 840
870:SET "X:"+NT$,P$
880:GOTO 190
900:REM LOESCHEN
910:CLS :GOSUB 100
920:CLS :PRINT " Löschen":
  GOSUB 20:GOSUB 330
930:KILL "X:"+NT$
940:GOTO 190

```

```

1000:REM INIT
1020:CLS :PRINT " Initialisierung":
  GOSUB 20
1030:CURSOR 0,2:PRINT " Diskette einle
  gen !!"
1040:IF INKEY$ =" " THEN 190
1050:IF INKEY$ =CHR$(13)THEN INIT "X":
  GOTO 190
1060:GOTO 1040
2000:REM ERROR-ROUTINE
2005:KBUFF$=""
2010:IF (ERN <>151OR ERL <>760)THEN 205
  0
2020:GOSUB 60:CURSOR 0,1
2025:PRINT "Anderer Name wählen !!"
2027:GOSUB 70
2030:RESUME 730
2050:IF (ERN <>151OR ERL <>550)THEN 210
  0
2060:GOSUB 60
2070:PRINT "Datei existiert bereits !"
2080:GOSUB 70
2090:RESUME 190
2100:IF ERN <>152THEN 2150
2110:GOSUB 60
2115:PRINT "Datei nicht gefunden !"
2120:GOSUB 70
2130:RESUME 190
2150:IF ERN <>157THEN 2200
2160:GOSUB 60
2170:PRINT " Falsche Bezeichnung !
  Bitte ändern !"
2180:GOSUB 70
2190:RESUME 730
2200:IF ERN <>159THEN 2250
2210:GOSUB 60
2220:PRINT " Schutz aktiv !!"
2230:GOSUB 70
2240:RESUME 190
2250:IF ERN <>160THEN 2300
2260:GOSUB 60
2270:PRINT "keine Diskette im Laufwerk"
2280:GOSUB 70
2290:RESUME
2300:IF ERN <>161THEN 2350
2310:GOSUB 60
2320:PRINT "Diskette noch unformatiert"
2330:GOSUB 70
2340:RESUME 190
2350:IF ERN <>162THEN 2400
2360:GOSUB 60
2370:PRINT " Schreib-/Lesefehler"
2380:GOSUB 70
2390:RESUME 190
2400:IF ERN <>163THEN 2450
2410:GOSUB 60
2420:PRINT "Ursprungdiskette einlegen"
2430:GOSUB 70
2440:RESUME
2450:IF ERN <>164THEN 2500
2460:GOSUB 60
2470:PRINT " Diskette voll !!"
2480:GOSUB 70
2490:RESUME 190
2500:IF ERN <>167THEN 2550
2510:GOSUB 60
2520:PRINT " Diskette zerstört !"
2530:GOSUB 70
2540:RESUME 190
2550:IF ERN <>168THEN RESUME 190
2560:GOSUB 60
2570:PRINT " Hardwarefehler"
2580:GOSUB 70
2590:RESUME 190

```

Dirk Mährholz
Goethestr. 86
2850 Bremerhaven

PC-1600

Mit freundlichen Grüßen

Dirk Mährholz

SOLITAIRE

Von : 44
Nach : 42



SPIELANLEITUNG :

Das Spiel SOLITAIRE funktioniert auf dem Rechner wie das bekannte Logikspiel. Das Ziel ist, nur noch einen Spielstein in der Mitte des Spielfeldes zu haben. Gezogen wird durch senk- und waagerechte Hüpfen in freie Felder. Der Übersprungene Stein kann dann weggenommen werden. Zum Ansprechen der Spielsteine werden die entsprechenden Zeilen- und Spaltennummern eingegeben. Z.B. von 42 nach 44 usw.

SOLITAIRE

```
5:"S"CLS :CLEAR :WAIT 0:DEGREE :RESTORE
10:DIM Z(0,6)
15:GOCURSOR 21,3:GPRINT "CE0F0B0B0B0B0B73007EFFC3C3C3C3FF7E00FFFFC0C0C0C0C0C0
0FFFF00030303";
20:GPRINT "FFFF03030300FEFF33333333FFFE00FFFF00FFFF1B1B3B7BDF8E00FFFF0B0B0B0B0B
C3C3"
25:FOR I=0TO 4STEP 4:LINE (120+I,9)-(148-I,21),,B:LINE (140,1+I)-(128,29-I),
,,B
30:LINE (120,9+I)-(148,21-I),,B:LINE (140-I,1)-(128+I,29),,B:NEXT I
35:FOR I=0TO 31:READ A,B:Z(A,B)=1:PSET (A*4+122,B*4+3):NEXT I
40:Z(3,3)=2:Z=32
45:KBUFF#="" :CURSOR 12,2:PRINT "      " ;:CURSOR 12,3:PRINT "      " ;
50:CURSOR 5,2:PRINT "Von : ";:CURSOR 12,2:INPUT "" ;X#
55:IF X#="" :IF Y#="" :S"GOTO 45ELSE LET X#=Y#
60:GOSUB 160:W=A:U=B
65:CURSOR 5,3:PRINT "Nach : ";:CURSOR 12,3:INPUT "" ;X#:GOSUB 160:U=A:W=B
70:IF Z(U,W)<>1OR Z(T,U)<>2GOTO 45
75:IF Z((U+T)/2,(W+U)/2)<>1GOTO 55ELSE LET C=ABS (W-U)*10+ABS (U-T)
80:IF C<>2IF C<>20GOTO 45
85:PRESET (U*4+122,W*4+3):PRESET ((T+U)*2+122,(U+W)*2+3):PSET (T*4+122,U*4+3)
90:Z(U,W)=2:Z(T,U)=1:Z((U+T)/2,(W+U)/2)=2:Z=Z-1:Y#=X#
95:C=0:FOR B=0TO 6:FOR A=0TO 6:IF Z(A,B)<>1GOTO 135
100:IF B<0IF Z(A,B+1)<>1GOTO 115
105:IF B<5IF Z(A,B+2)=2GOTO 130
110:IF B>1IF Z(A,B-1)<>2GOTO 115ELSE GOTO 130
115:IF A+1<7IF Z(A+1,B)<>1GOTO 135
120:IF A+2<2IF Z(A+2,B)=2GOTO 130
125:IF A-1>0IF Z(A-1,B)<>2GOTO 135
130:A=6:B=6:C=1
135:NEXT A:NEXT B:IF C=1GOTO 45
140:LINE (21,12)-(110,30),R,,BF:IF Z>1GOTO 150
145:CURSOR 4,2:PRINT "GEWONNEN !!!":BEEP 3:GOSUB 175:GOTO 155
150:CURSOR 4,2:PRINT "VERLOREN !!!":BEEP 1:GOSUB 175
155:CLS :CURSOR 3,1:PRINT "Neues Spiel (J / ) ?":GOSUB 175:IF D#="" :J"GOTO SELSE
CLS :END
160:IF LEN X#<>2GOTO 45
165:E=VAL LEFT# (X#,1):F=VAL RIGHT# (X#,1):IF E<0OR E>7OR F<0OR F>7GOTO 45
170:A=ASC LEFT# (X#,1)-49:B=ASC RIGHT# (X#,1)-49:RETURN
175:D#="" :INKEY# (1):IF D#="" :GOTO 175ELSE RETURN
180:DATA 0,2,0,3,0,4,1,2,1,3,1,4,2,0,2,1,2,2,2,3,2,4,2,5,2,6,3,0,3,1,3,2,3,4,3
,5,3,6,4
185:DATA 0,4,1,4,2,4,3,4,4,4,5,4,6,5,2,5,3,5,4,6,2,6,3,6,4
1547 Bytes
```

Thomas Jeger
Hauptstr. 142
CH-3226 Muntelstr

KARTENTRICK

Mit diesem Programm können Sie viele Freunde beeindrucken !
Anleitung: Sie denken sich eine Karte (z.B. Herz Dame). Nun lassen Sie den Computer alle 36 Karten (von 6 bis As) aufdecken. Falls Ihre gedachte Karte erscheint, merken Sie sich die Kolonne, in der sie erschienen ist. Nach dem Aufdecken aller 36 Karten werden Sie nach der Kolonnennummer gefragt. Dieses Spiel wiederholt sich 3x. Sie müssen immer an die selbe Karte denken ! Am Schluss zeigt Ihnen der Computer, an, welche Karte Sie gedacht haben. (tj)

Jürgen Klein
Borussiostr. 25
1000 Berlin 42

Wie schließe ich ein serielles Datenendgerät an die
RS-232-Schnittstelle des PC-1600 an?

Diese Frage soll hier : konkret am Beispiel eines BROTHER EP-22
beantwortet werden. An dessen Schnittstelle stehen laut Handbuch
die Signale RD (Pin 3), DTR (Pin 20) und SG (Pin 7) zur Verfü-
gung - nicht mehr und nicht weniger, aber das langt völlig für
eine problemlose Verbindung. Ebenfalls dem Handbuch kann man ent-
nehmen, daß der Anschluß an den PC-1500 (entsprechend der 0. 9.
Reihenfolge) an SD (Pin 2), CTS (Pin 5) und SG (Pin 7) der
Schnittstelle CE-150 zu erfolgen hat. Diese soll aber gerade ein-
gespart werden, da der PC-1600 eine serielle Schnittstelle be-
sitzt. Leider entspricht deren Buchse aber keiner gängigen Norm,
so daß es nicht ganz einfach ist, deren im Handbuch beschriebene
Pins mit den EP-22 zu verbinden, obwohl alle zuletzt beschriebenen
Signale dort vorhanden sind.
Zwei Lösungen bieten sich an:

Erstens kann man ein Kabel CE-1605L kaufen und entsprechend der
üblichen Belegung eine Norm-Buchse (DB 25 female) an das freie
Kabelende anlöten. (Das geht nach der dem Kabel beiliegenden Be-
legungsbeschreibung einfach; das von Pin 9 kommende weiße Kabel
wird nicht angeschlossen, und beachtet werden muß, daß das Wort
"cyan" in der englischen Sprache "dunkelblau" bedeutet, "blue"
hingegen hier das hellblaue Kabel meint.). An diese Buchse kann
nun ein dreiadriges Kabel wie zwischen PC-1500/CE-150 und EP-22
angeschlossen werden. Der Nachteil dieses Verfahrens sind die
Kosten von etwas über hundert Mark. Für eine dreia drige Verbin-
dung ist das auch in Anbetracht der Kosten für den Spezialstecker
PC-1600-wärts ein stolzer Preis. Immerhin hat man dann ein Uni-
versalkabel.

Zweitens aber kann man auch das zur Verbindung mit den Plottern
CE-515P/516P gebaute Kabel CE-5161L kaufen, das erheblich billi-
ger ist. Den vierpoligen Stecker am Ende öffnet man und entfernt
ihn durch Ablösen oder Abschneiden von der Leitung. Das rote Ka-
bel (PIN 2 des PC-1600) kommt an Pin 3 des EP-22, das weiße an
Pin 7 und das blaue an Pin 20. Dabei bleiben die beiden im Stek-
ker verborgen gebliebenen diskreten Bauteile an ihrem Platz, also
das kleine zwischen dem roten und dem weißen und das große zwi-
schen dem weißen und die Kabelabschirmung, die nicht mit ange-
schlossen werden muß. Im DB-25-Gehäuse sollte durch reichliche
Verwendung von Isolierband aber ein Kontakt zu anderen Stellen
ebensoverhindert werden, wie eine gute Zuentslastung zu gewähr-
leisten ist.

Software wird natürlich auch gebraucht. Das EP-22 ist auf ONLINE
300 Baud zu schalten; in den PC-1600 gibt man folgendes ein:
SETCOM "COM1:",300,8,N,2

SETDEV "COM1:",P0

Die Befehle LLIST und LPRINT wirken jetzt direkt auf das
EP-22. Übrigens ist zu empfehlen, den vierpoligen Stecker zu ver-
wahren. Kauft man sich nämlich einmal einen SHARP-Plotter, kann
man sich recht einfach mit einem weiteren DB-25-female-Stecker
die ursprüngliche Verbindung erneut bauen.

PS: Natürlich kann man auch gleich ein Kabel CE-1603L kaufen -
aber das kostet auch fast hundert Mark.

```
20:DMA A+(2)*40,0+(4)*12,K+(30)
25:GOTO 200
30:FOR I=1 TO X
32:SUM NID0 (S1,1,1)+U0L NID0 (L0,1,1)
35:LINE ((1-1)*30,0)-((1-1)*30+20,21),,0
37:RESTORE "S"
43:FOR J=1 TO S:READ A:(0):READ A:(1):NEXT J
45:CURSOR (1-1)*30+9,12:PRINT A:(0)
50:CURSOR (1-1)*30+9,21:PRINT A:(1)
55:RESTORE "L"
63:FOR J=1 TO L:READ A:(NEXT J
65:CURSOR (1-1)*5+2,11:PRINT A:
70:NEXT I:RETURN
100:"S"DATA "00000000"OF070000000000,"020707C7E2
    FFE2C70702"
110:DATA "00000000"OF0FC0000000,"000E1F0CFFFC
    F0F1F000"
120:DATA "00000000"OF0C0F0F0F0F0F0E,"0001030F3FFF
    F0F030100"
130:DATA "00000000"OF0C0000000000,"0207070F3FFF
    F0F070202",""
140:"L"DATA "0","7","8","9","10","B","D",""
    K","A"
200:CLS :PRINT ")))" K=antw-ich (((((
210:PRINT :PRINT " (Taste drücken)
220:IF INKEY (1)="" THEN 220
230:RANDOMIZE (0)
240:FOR I=1 TO 30
250:K=RND (30):IF K(K)="" THEN GOTO 250
260:K(K)="A"
270:A(2)=A(2)+CHR (K+L0)
280:NEXT I:K=3
290:FOR J=1 TO 3
300:FOR F=1 TO 30STEP 3
310:A=NID0 (A(2),F,3):S=""L=""
320:FOR G=1 TO 3:G=0:G=NID0 (A,0,1)
330:K=ASC (NID0 (A,0,1))-L0
340:S=INT ((K-1)/9):L=K-(S*9):S=S-1:L=L-S:STR0 L
    :S=S+STR0 S
350:NEXT G:GOSUB 30
355:IF INKEY (1)="" THEN 355
360:NEXT F
370:CLS :CURSOR 0,1
380:INPUT "Kolonne (1-3):"J
390:B(4)=B(1)+B(2)+B(3)+B(2)+B(4)
400:A(2)=B(1)+B(2)+B(3)
405:B(4)=""B(2)=""B(3)=""
410:CLS :NEXT J
420:X=1
430:K=ASC NID0 (A(2),10,1)-L0:K=INT ((K-1)/9)
    L=L-STR0 (K-S*9):S=STR0 (S+1)
440:GOSUB 30
450:CURSOR 0,1:PRINT "Diese Karte haben":CURSOR
    0,2:PRINT "Sie sich gedacht!"
460:K=UFF0 ""
470:IF INKEY (1)="" THEN 470
480:CLS :END
```

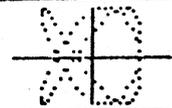
Dirk Mahrholz
Goethestr. 86
2850 Bremerhaven

PC-1600

LISSAJOUS - FIGUREN

Lissajous-Figur

$$\frac{\psi}{2} = \frac{65^\circ}{3}$$



Sie entstehen, wenn man zwei Signale verschiedener Frequenz und Phasenlage überlagert. ANWENDUNG: Mit den Oszillogrammen ist es dann möglich, eine unbekannte Frequenz zu bestimmen. Dazu muss eine der beiden Frequenzen bekannt sein.

Nach Eingabe des Phasenwinkels Phi und dem Frequenzverhältnis zeichnet der Rechner die entsprechende Lissajous-Figur aufs Display.

```
5:"L"CLS :CLEAR :WAIT 0:KBUFF#="" :DEGREE :BEEP ON
10:A=130:B=10:C=4:P#="0C10FC12120C":W#="38444030404438"
15:PRINT " ** Lissajous-Figur **"
20:GDCURSOR 4,15:GPRINT P#:CURSOR 3,2:INPUT "=" ;P
25:GDCURSOR 4,24:GPRINT W#;"0020F0":CURSOR 3,3:PRINT " : " :GDCURSOR 30,24:
GPRINT W#;"009000A0"
30:CURSOR 8,3:INPUT "=" ;W1:INPUT " : " ;W2:CLS :IF W1>30R W2>3LET C=2
35:PRINT "Lissajous-Figur":GDCURSOR 6,15:GPRINT P#:CURSOR 3,2:PRINT "=" ;P;
CHR# 248
40:CURSOR 0,3:PRINT W1;" : " ;W2:LINE (A,0)-(A,31):LINE (A-25,B)-(A+25,B)
45:FOR T=0TO 360STEP C:X=A+B*SIN (W1*T):Y=B-(B*SIN (W2*T+P)):PSET (X,Y):
NEXT T:BEEP 1:KBUFF#=""
50:D#="INKEY# :IF D#=""GOTO 50ELSE IF D#=""CLS :END ELSE GOTO 5
```

492 Bytes

NEU! INTERROBOTIK II

NEU von der HANNOVER-MESSE und von der CEBIT:

Mobile Meß- und Datenstation

und INTERFACE

für Fischertechnik, Robotik.

Lassen sich die beiden Sharp-Pocketcomputer PC 1500 (A) und PC 1600 als vollwertige Computer in handlicher Ausführung nicht auch als Steuerungszentrale von technischen Abläufen bzw. Modellen wie der Fischertechnik-Baukästen gebrauchen, wo doch die PCs so nützlich für viele Schreibtischaufgaben sind?

Kann denn nicht ein PC 1500/1600 auf dem Schreib- oder Demonstrationstisch auch noch eine Modellsteuerung zur Simulation technischer Funktionen und Vorgänge gleich selbst vornehmen, wenn er schon die vorhergehenden Berechnungen entsprechender Projekte übernommen hat?

Oder müssen noch Daten aus verschiedenen Labormessreihen mühsam einzeln in den Rechner getippt werden, wenn dieser dann innerhalb von Sekunden die Auswertung erledigt und fast in derselben Zeit die Erfassung dieser Daten gleich mitübernehmen kann?

Diese Fragen drängen sich doch geradezu auf; und in der Tat läßt sich das oben genannte realisieren, wenn nur ein entsprechendes Interface vorhanden ist. Dieses muß allerdings die besonderen Gegebenheiten der beiden Pocketcomputer berücksichtigen: Die besonders kleine 60polige Steckverbindung, die am PC noch nicht vorhandenen Portleitungen, welche die Tore zur Außenwelt bilden sollen, und die geforderte räumliche Beengtheit eines "Anbaus".

Jetzt gibt es das neue Interface INTERROBOTIK II. Es bringt nicht nur die Möglichkeit, das vollständige Fischertechnik-Computing-System am kleinen PC zu betreiben, sondern auch eine komfortable Meß- und Datenstation aufzubauen. Diese Meßstation ist vollkommen netzunabhängig und geeignet, Programme und Daten mit anderen Personalcomputern auszutauschen.

So steuert das INTERROBOTIK II die Modelle der Konstruktionsbaukästen des Fischertechnik-Computing-Systems (wie verschiedenen Roboter, Automaten und Graphikgeräte), wobei als Einsatzorte außer dem Hobbybereich auch der semiprofessionelle - insbesondere der schulische - Bereich zu sehen sind. Das 'Teach in'-Robotermodell wurde am SHARP-Stand der diesjährigen CEBIT HANNOVER MESSE ausgestellt.

Bis zu 4 Motoren und/oder Elektromagneten, die über Leistungsstufen des Interfaces angesteuert und von dem normalen Fischertechnik-Netzteil mit Strom versorgt werden, verleihen dem PC "Arme und Beine". Die "Sinnesorgane" werden sozusagen durch 8 Taster und zwei Potentiometer nachgebildet. Sie geben die verschiedensten Rückmeldungen aus den Modellen an den Computer bzw. sein Programm. Die Tasten können aber auch zum Aufbau von Kommandofeldern für die abzuarbeitenden Programmteile dienen.

Als Beispiel für die vielen Modellvorschläge der Fischertechnikbaukästen sei hier nur die Solarzellennachführanlage angesprochen. Eine Solarzelle wird mit Hilfe eines Programms frontal zu jeder Himmelsrichtung auf optimale Energieeinstrahlung hin ausgerichtet.

Hieran wird schon deutlich, daß mit den Fischertechnik-Modellen und erst recht durch die Erweiterungsmöglichkeiten des Interfaces, wie im folgenden aufgeführt, der Phantasie keine Grenzen gesetzt sind.

Zum ersten werden außer den von den Fischertechnik-Modellen gebrauchten 8 Ein- und 8 Ausgangsleitungen noch 8 wahlweise als Ein- oder Ausgänge zu benutzende Leitungen geboten, die z.B. zum Anschluß von zusätzlichen digitalen Gebern wie Reed(Magnet)-Schaltern oder zur direkten Ansteuerung von Leuchtdioden oder Relais dienen können.

An die Motorausgänge des Interfaces sind auch Schrittmotoren z.B. für Modelle, die unabhängig von dem genannten Baukastensystem aufgebaut werden sollen, unmittelbar anzuschließen. Auf der großen HANNOVER MESSE 1987 war -als Beispiel dafür- ein aus leichten Aluminiumprofilen und zwei Schrittmotoren aufgebauter Roboterarm zu sehen der an verschiedenen Raumkoordinaten Temperaturmeßwerte aufnahm. Diese wurden auf dem PC 1600 Plotter graphisch dargestellt.

Für die erwähnte Meß- und Datenstation sind nicht nur die zwei Analogmeßwertwandler für die Potentiometer der Fischertechnikmodelle wie bei den Home-Computer-Interfaces vorhanden, sondern ein 4-kanaliger hochgenauer 10 bit Analog-Digital-Wandler ist zusammen mit einer Präzisionsreferenzspannungsquelle aufgebaut. Dieser kann noch zusätzliche Meßwerte wie über Temperatur- oder Lichtverhältnisse zur Auswertung liefern. Mit dem auch angebotenen Sensor ist dabei z.B. eine genaue Temperaturbestimmung in einem Bereich von 0 C - 110 C mit einer Genauigkeit von 0,1 C (sensorbedingt) möglich (auf Wunsch Meßbereichserweiterung von -55 C bis +150 C). Mit dem Fotosensor lassen sich Helligkeitsunterschiede z.B. von Bildvorlagen wie bei Barcodelesern erfassen und auswerten.

Die Netzunabhängigkeit der Pocketcomputer und dieses Interfaces sowie die Unabhängigkeit von einem Monitor machen das Messen und Steuern auf einem Tisch noch einfacher. Zusätzlich steht zur Erweiterung der Meß- und Datenstation noch eine Centronicschnittstelle mit passender Ansteuerungssoftware für Drucker und ein fertiges Kabel dafür zur Verfügung. Diese Schnittstelle ist allerdings auch zum Datenaustausch mit anderen Computern geeignet.

Noch komfortabler zu nutzen ist allerdings das INTERROBOTIK II in Verbindung mit dem PC 1600. Hiermit läßt sich der kleine Rechner auch ohne sonstige SHARP-Peripherie zur kompletten aber äußerst kompakten Datenaufzeichnungsstation (eventuell gegen Umwelteinflüsse durch einen stabilen abschließbaren Aluminiumkoffer geschützt), einsetzen, wobei vorher mit Hilfe weniger Basicbefehle und einem seriellen Kabel von irgendeinem Computersystem mit seriellem RS-232 Anschluß das gewünschte Programm in den PC 1600 geladen wird. Zur Aufnahme von Temperaturkennlinien über beliebige lange Zeiträume und in beliebigen Zeitabständen von 1 s bis zu Tagen bzw. Monaten eignet sich z.B. ein mitgeliefertes Programm und der angebotene Sensor. Dann ist es natürlich auch möglich, später die gewonnenen Daten zur Auswertung z.B. mit Hilfe einer Fourieranalyse dem fremden Computer zu überlassen.

Und nun noch zusammengefaßt die technischen Daten:

1) Analogmeßwertaufnahme:

- A/D-Wandler mit 8 oder 10 bit Auflösung, was bei einer maximalen Eingangsspannung von 2,56 V 10mV pro Digit bzw. 2,5 mV pro Digit entspricht

- Meßrate maximal 300 Messungen pro Sekunde
- mit dem Temperatursensor ist im Bereich von 0 C bis 110 C (bzw. -55 C bis +150 C) eine Auflösung von 0,1 C möglich
- 4 Kanäle für 4 verschiedene Eingangsspannungen mit Eingangswiderständen von über 100 MOhm
- Referenzspannungsquelle mit 2,56V für bis zu 4 Potentiometer, wird hier z.B. auch für die Fischertechnikmodelle benutzt
- 2) Digitale Ein-,Ausgänge
 - 8 Eingänge für Taster der Fischertechnikmodelle usw.
 - 8 Ausgänge benutzt für die Motorsteuergruppen
 - 8 frei als Ein-, Ausgänge, bzw. für Maschinenprogrammierer auch als Interrupteingänge belegbare Leitungen
 - digitale Ausgänge maximal bis je 2 mA belastbar
- 3) Motorsteuerung
 - 4 Gleichstrommotoren bzw. auch Schrittmotoren und/oder Magnetspulen für Links- Rechtslaufrichtung
 - betreibbar mit Fischertechnik-Netzteil oder externem Akku bis je 350 mA (kurzzeitig bis 2 Ampere) pro Motor
 - typischer Stromverbrauch der Fischertechnik-Motoren je ca. 90 mA (der Leistungstransistoren im Stillstand unter 1 mA)
- 4) Software
 - alle Baugruppen über Basicbefehle (Poke./Peek..) ansprechbar, jedoch auch kurze Maschinenroutinen für die schnellere Analogmeßwertaufnahme vorhanden
 - ausführliche Bedienungsanleitung mit mehreren Beispielprogrammen auch zur Temperaturmessung und mit sämtlichen Programmen der Fischertechnik-Computing-Modelle (Ergänzung zur Fischertechnik-Programmieranleitung)
 - Sämtliche Programme auf C-Cassette
- 5) Erweiterungsmöglichkeiten
 - Digital-Analog-Wandlermodul: gibt analoge Spannungen im Bereich von 0 V bis 2,5 V in Schritten von 10 mV mit 1 mA Belastung aus und ist geeignet:
 - als Funktionsgenerator auch zur Ton-/ Klangerzeugung
 - zur Softwaregesteuerten Drehzahlregelung von Motoren usw.
 - PC-1600 -IBM PC Verbindung
 - insgesamt bis zu 4 Temperatur- u. Fotosensoren

6) Anschlüsse, Gehäuse, und Sonstiges

- 20-poliger Stecker für die Fischertechnik-Buchsen-Verbindung
- 20-poliger Stecker für Erweiterungen über zwei Analogeingänge, 8 Portleitungen, Speisespannung, Referenzspannung und Masse
- Netzteil- oder Akkuanschluß für die Motorsteuerung
- Stromversorgung dank CMOS-Bauteilen mit nur ca. 10 mA vom PC aus
- alle ICs gesockelt
- stabiles Aluminium Gehäuse, mit seitlicher Herausführung der Stecker passend z.B. in den Aufbewahrungskasten des PC 1500/CE 150
- Ausstattungsmöglichkeit: formschöner stabiler Aluminiumkoffer

PREIS FÜR INTERROBOTIK II inkl. 14% MwSt.:

Meß- und Datenstation und Interface für Fischertechnik, Robotik neu im Profi-Aluminiumgehäuse
komplett wie beschrieben **455 DM**

Digitalanalogwandlermodul mit Verbindungskabel, Stecker sowie Funktionsgenerator-Programm **75 DM**

genauer Temperatursensor anschlussfertig mit 2 m abgeschirmtem Kabel (dreifach, noch verlängerbar)

0 C bis 110 C	28 DM
-55 C bis 150 C	45 DM

Reflexkoppler (Optobaustein) auch für Barcodes geeignet
komplett anschlussfertig, mit Programm "Die Nadel im Heuhaufen"

35 DM

Schrittmotor, (bis zu zwei direkt) betreibbar am Fischertechnik Port anschlussfertig mit Maschinenprogramm

35 DM

Centronicschnittstelle für Drucker mit komfortabler Listroutine (REM-Zeilen in Breitschrift etc.) und Textprogramm für REM Zeilen einschließlich komplettem Centronicskabel

85 DM

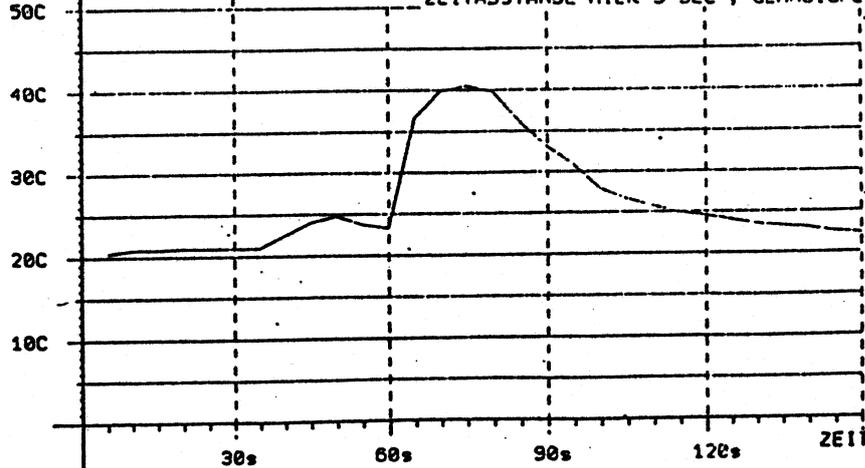
formschöner stabiler Alukoffer mit individueller Ausgestaltung nach Wunsch

auf Anfrage

serielles Kabel zur Verbindung mit IBM PCs etc.

auf Anfrage

DER SHARP PC ERSTELLT MIT HILFE DES ROBOTERS UND INTERROBOTIK II EIN DIAGRAMM FÜR DIE TEMPERATURVERÄNDERUNG BEI BEWEGUNG DES SENSORS BZW. EINTAUCHEN IN EINE FLÜSSIGKEIT.
ZEITABSTÄNDE HIER 5 SEC, GENAUIGKEIT NACH ABGLEICH 0.1 GR.C



INTERROBOTIK II-

```

1:ARUN :CLEAR :CLS :A=&B001:E=&B000:K=&B000:D=&B001:PAPEL K:XP0KE &B003,144:
GOTO 145
3:COLOR 0:TEXT :CSIZE 3:LPRINT " SHARP HARD-SOFT H.14.1.MANN"
4:COLOR 1:TEXT :CSIZE 2:LPRINT " PROGRAMM FÜR INTERROBOTIK II, SCHRITTKRATUR
GETRIEBENER
5:LPRINT " ROBOTARM UND TEMPERATURSENSOR"
6:COLOR 0:TEXT :LPRINT " DER SHARP PC ERSTELLT MIT HILFE DES ROBOTARMS UND
";
7:COLOR 3:LPRINT " INTERROBOTIK II";:COLOR 0:LPRINT " EIN"
8:LPRINT " DIAGRAMM FÜR DIE TEMPERATURVERÄNDERUNG BEI BEWEGUNG DES SENSORS
BZW."
9:LPRINT " EINTAUCHEN IN EINE FLÜSSIGKEIT."
10:LPRINT " ZEITABSTÄNDE HIER 5 SEC , GENAUIGKEIT NACH ABGLEICH 0.1 GR.C "
12:LPRINT " -----"
13:IF 20:PAPER R:GRAPH :GLCURSOR (80,0):SORGN
16:LLINE (0,-50)-(000,400),0,1:LLINE (-1,400)-(-1,-50)
17:LLINE (-30,000)-(750,000),0,1:LLINE (750,-1)-(-30,-11)
18:FOR I=1TO 00:LLINE (-7,+1*80-40)-(750,+1*00-40),0,1:LLINE (750,1*80)-(-15,
1*80),0,1
19:IF 1<GLCURSOR (-80,1*80-0):LPRINT I*10;CHR$(8);":NEXT I
20:FOR J=0TO 1:GLCURSOR (-70+J,472):CSIZE 2:LPRINT "T":NEXT J
21:FOR I=1TO 00:LLINE (-4,1*80)-(0,1*80),0,1:NEXT I
22:FOR I=1TO 02:LLINE (1*300-150,400)-(1*300-150,-15),0,1:LLINE (1*300,-15)-
(1*300,400),0,1
24:NEXT I:LLINE (750,400)-(750,-15),0,1
25:FOR I=33TO 01STEP -1:LLINE (1*25,0)-(1*25,-7),0,1:NEXT I
26:CSIZE 2:COLOR 1:FOR I=1TO 4:GLCURSOR (1*150-25,-10):LPRINT I*30;"s":NEXT I
30:FOR I=0TO 1:GLCURSOR (710+I,-30):LPRINT "ZEIT":NEXT I
31:B=2:CLS :CURSOR 0:PRINT "INIT ROBOTARM"
32:RESTORE 2000:FOR I=0TO 8:READ C:XP0KE A,C:GOSUB "C":BEEP B:NEXT I:GOTO 32
42:RESTORE 2010:FOR I=0TO 8:READ C:XP0KE A,C:GOSUB "D":BEEP B:NEXT I:GOTO 42
50:TIME =10100:Z=0:CLS
51:X=0:XX=3:REM X BEWEGUNG
52:RESTORE 2030:FOR I=0TO 8:READ C:XP0KE A,C:BEEP B:NEXT I
53:GOSUB 130:IF X=XXGOTO 61
54:GOTO 52
61:X=0:XX=0:REM Y BEWEGUNG
62:RESTORE 2020:FOR I=0TO 8:READ C:XP0KE A,C:BEEP B:NEXT I
63:GOSUB 130:IF X=XXGOTO 71
64:GOTO 52
71:X=0:XX=0:REM -Y BEWEGUNG
72:RESTORE 2000:FOR I=0TO 8:READ C:XP0KE A,C:BEEP B:NEXT I
73:GOSUB 130:IF X=XXGOTO 81
74:GOTO 52
81:X=0:XX=3:REM 2.X BEWEGUNG
82:RESTORE 2030:FOR I=0TO 8:READ C:XP0KE A,C:BEEP B:NEXT I
83:GOSUB 130:IF X=XXGOTO 91
84:GOTO 52
91:X=0:XX=0:REM Y BEWEGUNG
92:RESTORE 2020:FOR I=0TO 8:READ C:XP0KE A,C:BEEP B:NEXT I
93:GOSUB 130:IF X=XXGOTO 101
94:GOTO 52
101:X=0:XX=0:REM -Y BEWEGUNG
102:RESTORE 2000:FOR I=0TO 8:READ C:XP0KE A,C:BEEP B:NEXT I
103:GOSUB 130:IF X=XXGOTO 112
104:GOTO 52
120:"C"IF (XPEEK EAND 2)=2XP0KE A,0:GOTO 42
122:RETURN
124:"D"IF (XPEEK EAND 1)=1XP0KE A,0:GOTO 50
126:RETURN
130:XP0KE A,0:X=X+1:IF X<XX+1GOSUB 1000
132:RETURN
142:CLS :XP0KE A,0:TEXT :IF P=1LF 14
145:I=I+1:WAIT 0:CURSOR 0,1:PRINT "PLOTTER?<P>/DISPLAY?<D>"
150:IF INKEY$ ="D"GRAPH :P=0:GOTO 31
155:IF INKEY$ ="P"LET P=1:GOTO 3
160:IF 1<200GOTO 145
170:I=0:GOTO 000
600:"N"TIME =10100
605:KBUFF$ ="RUN"+CHR$(13)
610:WAKE$(0)=""01/01/00/05:RUN"+CHR$(13):XCALL RE33F:GOTO 145
1000:Z=2:1:FOR J=1TO 10:XP0KE K,2:AI=XPEEK D14-(XP0LK 0,1:TIME 132):04:00-00-A1
:NEXT J:AD=INT (AD/4)/10
1005:CURSOR 17,0:PRINT "ZEIT";Z*5;"s"
1010:CURSOR 19,1:PRINT AD;CHR$(8);"C"
1012:IF 0<AD:LLINE ((2-1)*4,41-AD)-(2*4,41-AD)
1020:IF (P=1)AND (0<AD):FOR I=0TO 1:LLINE ((2-1)*20,00+80)-(2*20,00+80),0,3:NEXT

```

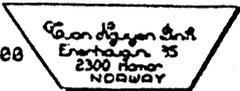
```

1022:IF TIME <10100.00106GOTO 1022
1024:TIME =10100.0005
1025:00+00:RETURN
2000:DATA 10,00,04,96,32,100,128,144,0
2010:DATA 09,00,10,02,00,004,005,001,0
2020:DATA 144,128,100,32,96,004,000,010,0
2030:DATA 001,005,004,00,02,010,000,009,0
2040:DATA 145,133,104,30,90,074,080,025,0
2050:DATA 025,000,074,90,30,104,133,145,0
3000:"M"PRINT TIME :XP0KE A,1:GOTO 3000

```

* PC-1600 39 Char/5 Line *

```
5:WAIT 0:CLS :CLEAR :DIM A$(0)*39:K=20:D=2000
6:FOR I=0TO 4:RESTORE K+I:READ A$(0)
7:IF A$(0)="END"THEN 10
8:FOR J=0TO LEN A$(0)-1:A=ASC MID$(A$(0),J+1,1):A=(A-32)*3
9:GDCURSOR J*4,I*5+1:GPRINT PEEK (D+A);PEEK (D+A+1);PEEK (D+A+2):
NEXT J
10:NEXT I
```



-TABLE GPRINT CODES OF THE SPECIAL CHARACTERS:

```
11:I*=INKEY* :IF I*=""THEN 11
12:IF I*="E"THEN 19
13:IF I*="C"THEN 15
14:GOTO 11
15:CLS :K=K+5:GOTO 6
16:I*=INKEY* :IF I*=""THEN 16
17:IF I*="E"CLS :END
18:GOTO 16
19:CLS :END
```

:	-->	"	D000 00 00 00 00 17 00 00 00 00 0A 1F 0A 15 1F 15 09 : 9C
C	-->	@	D010 04 12 0C 0B 14 00 03 00 0C 12 00 00 12 0C 0A 04 : 8E
o	-->	o	D020 0A 04 0E 04 00 14 0C 04 04 04 00 10 00 08 04 02 : 6A
R	-->	f	D030 0E 11 0E 12 1F 10 19 15 12 15 15 0A 0F 08 1C 17 : 2C
i	-->	±	D040 15 00 0E 15 09 01 10 03 1B 15 1B 12 15 0E 00 0A : F9
			D050 00 00 15 0D 04 0A 11 0A 0A 0A 11 0A 04 01 15 02 : 96
			D060 09 10 0E 1E 09 1E 1F 15 0A 0E 11 13 1F 11 0E 1F : 46
			D070 15 11 1F 05 01 0E 11 1D 1F 04 1F 11 1F 11 09 10 : 22
			D080 0F 1F 04 1B 1F 10 10 1C 0E 1F 1F 01 1E 1E 11 0F : 51
			D090 1F 05 03 1E 09 10 1F 05 1A 12 15 09 01 1F 01 1F : 12
			D0A0 10 1F 0F 10 0F 0F 1E 0F 1B 04 1B 03 1C 03 19 15 : 23
			D0B0 13 1F 11 00 02 0C 10 00 11 1F 02 01 02 08 08 08 : AE
			D0C0 03 04 00 0C 12 1E 1F 12 0C 0C 12 10 0C 12 1F 0C : F8
			D0D0 10 16 04 1E 05 10 10 0E 1F 02 1C 00 1D 10 00 10 : 07
			D0E0 1D 1F 08 14 00 1F 00 1E 1C 18 1E 02 1C 0C 12 0C : 2F
			D0F0 1E 0A 08 04 0A 1E 1E 04 02 14 1E 0A 02 0F 12 1E : FB
			D100 10 1E 0E 10 0E 0E 1C 0E 12 0C 12 10 14 0E 1A 10 : 2A
			D110 12 04 1B 11 02 05 02 11 1B 04 1F 0B 04 00 00 00 : A9
			D120 10 0F 06 1D 10 1D 1C 1E 15 1A 19 12 09 14 1D 18 : 55
			D130 19 12 08 15 1C 04 04 0A 1E 10 10 1D 1C 15 1C 10 : 4E
			D140 10 01 1C 01 02 1D 02 00 1D 02 19 14 19 18 15 18 : FF
			D150 1C 1E 15 1A 0C 10 1E 1F 15 1A 19 1A 1D 14 1D 00 : 78
			D160 19 1A 1A 11 1A 18 11 1A 15 08 05 1D 14 1D 1D 10 : 58
			D170 1D 04 1F 0A 14 1E 15 05 1E 05 0C 1F 01 14 0E 05 : 0C
			D180 08 10 1D 02 1D 00 1B 1A 01 18 12 19 1D 05 19 1D : 28
			D190 05 19 12 15 17 0E 1B 0E 18 15 00 03 01 01 01 01 : C7
			D1A0 03 0C 0A 0C 1E 05 02 12 17 12 00 00 00 00 00 00 : 85

Den Text legen Sie in die Data-Zeile von 20 ein.

Bitte beachten Sie, dass die letzte Zeile mit "END" endet.

Betätigen Sie die Taste [C] : Neuer Text

Betätigen Sie die Taste [E] : ENDE

Werner Scherer
Moselufer 7
5558 Schweich

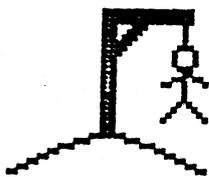
hier mein Einzeller, eine Hard-Copy-Routine für den PC-1600.

```
1:GRAPH :FOR J=0TO 31:FOR I=0TO 155:A=POINT (I,J):
GLCURSOR (3*J,-3*I):LPRINT CHR$(A*2E):NEXT I:NEXT J:
TEXT
```

Sie liefert beispielsweise folgenden Plott:

HARD - COPY
für den PC-1600
>

Die Routine ist zwar nicht schnell, aber in einer Zeile geht nicht viel.



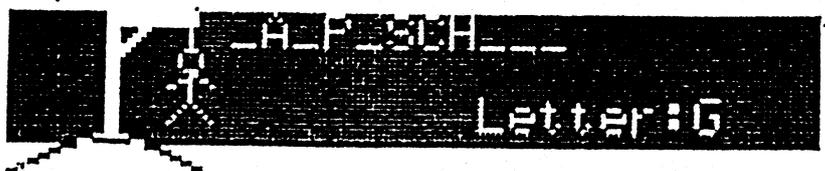
HANGMAN

Any key to start

Das Prinzip dieses Spieles ist wohl allgemein bekannt; es müssen Worte erraten werden, von denen man anfangs nur die Buchstabenanzahl kennt. Dies geschieht durch raten jeweils eines Buchstaben.

Um das Programm so zu ändern, dass Sie auch Ihre eigenen Worte, die in die DATA-Zellen ab Zeile 690 geschrieben werden müssen, erraten lassen können, müssen Sie dann nur noch das Parameter von RND in Zeile 270 (hier 17) entsprechend der Anzahl der möglichen Worte ändern.

```
10:'Hangman - Wörterraten
20:'
30:'Published 1987 by
40:'
50:'Seelig Soft
60:'Sögeler Str. 30, 3000 Hannover
70:'
```




```

10: CLEAR
20: DIM D$(0)*18
30: DIM TR(7)
40: DIM LZ(7,5)
100: CLS :WAIT 0:PRINT "* * LOTTOSCHEINANALYSE * *"
110: PRINT "-----"
120: PRINT "ZIEHUNGSZAHLENERFASSUNG=1 "
130: INPUT "TIPPSCHEINAUSWERTUNG =2";B
140: IF B<10R B>2GOTO 100
150: MAXFILES =1
160: IF B=1GOTO 810
170: IF B=2GOTO 3000
810: OPEN "X:LO-DATUM"FOR INPUT AS #1
820: INPUT #1,A$
830: CLOSE #1
840: JH=VAL (MID$(A$,1,2))
850: WO=VAL (MID$(A$,3,2))
910: OPEN "X:LO-ZAHLE"FOR APPEND AS #1
1000: CLS :WAIT 0:PRINT "Zahlenreihe für 19";STR$(JH);"/";STR$(WO);".Wo"
1005: PRINT "-----"
1010: A$="":INPUT A$:IF A$="99"LET JH=JH+1:WO=1:GOTO 1000
1015: IF A$="88"THEN CLOSE #1:GOTO 9000
1020: Z1=VAL (MID$(A$,1,2))
1030: Z2=VAL (MID$(A$,3,2))
1040: Z3=VAL (MID$(A$,5,2))
1050: Z4=VAL (MID$(A$,7,2))
1060: Z5=VAL (MID$(A$,9,2))
1070: Z6=VAL (MID$(A$,11,2))
1080: Z7=VAL (MID$(A$,13,2))
1082: IF Z1<00R Z1>49BEEP 3:PRINT "FEHLER, BITTE WIEDERHOLEN":GOTO 1000
1083: IF Z2<00R Z2>49BEEP 3:PRINT "FEHLER, BITTE WIEDERHOLEN":GOTO 1000
1084: IF Z3<00R Z3>49BEEP 3:PRINT "FEHLER, BITTE WIEDERHOLEN":GOTO 1000
1085: IF Z4<00R Z4>49BEEP 3:PRINT "FEHLER, BITTE WIEDERHOLEN":GOTO 1000
1086: IF Z5<00R Z5>49BEEP 3:PRINT "FEHLER, BITTE WIEDERHOLEN":GOTO 1000
1087: IF Z6<00R Z6>49BEEP 3:PRINT "FEHLER, BITTE WIEDERHOLEN":GOTO 1000
1088: IF Z7<00R Z7>49BEEP 3:PRINT "FEHLER, BITTE WIEDERHOLEN":GOTO 1000
1100: CLS :WAIT 0:PRINT "Zahlenreihe für 19";STR$(JH);"/";STR$(WO);".Wo"
1105: PRINT "-----"
1110: PRINT " ";Z1;Z2;Z3;Z4;Z5;Z6;Z7
1120: P$="":INPUT "RICHTIG (J/N)";P$
1130: IF P$="N"GOTO 1000
1140: D$(0)="":IF WO<10LET D$(0)=D$(0)+STR$(JH)+"0"+STR$(WO):GOTO 1160
1150: D$(0)=D$(0)+STR$(JH)+STR$(WO)
1160: D$(0)=D$(0)+A$:PRINT #1,D$(0)
1170: GOTO 1500
1500: WO=WO+1:GOTO 1000
2000: MAXFILES =1
2002: OPEN "X:LO-ZAHLE"FOR INPUT AS #1
2005: IF EOF (1)THEN CLOSE #1:GOTO 9999
2006: WAIT
2010: INPUT #1,D$(0):CLS :PRINT D$(0)
2020: GOTO 2005
3000: CLS :PRINT "TIPPSCHEINAUSWERTUNG..."
3002: PRINT "AB WELCHEM TREFFER SOLL"
3004: PRINT "GEDRUCKT WERDEN?"
3006: INPUT "BITTE (3/4/5/6) EINGEB. ";T1
3007: IF T1<3OR T1>6GOTO 3000
3009: FOR I=0TO 7
3010: CLS :WAIT 0:PRINT "Tippreihe ";STR$(I+1);" ,bitte alle 6"
3020: PRINT "Zahlen 2-stel. in Folge mit"
3030: PRINT "führenden Nullen eingeben."
3040: A$="":INPUT "Z.B. 8=08 usw.";A$
3045: IF VAL (MID$(A$,1,2))=99LET LZ(I,0)=99:GOTO 3180
3050: IF VAL (MID$(A$,1,2))<10R VAL (MID$(A$,1,2))>49GOTO 3010
3060: IF VAL (MID$(A$,3,2))<10R VAL (MID$(A$,3,2))>49GOTO 3010
3070: IF VAL (MID$(A$,5,2))<10R VAL (MID$(A$,5,2))>49GOTO 3010
3080: IF VAL (MID$(A$,7,2))<10R VAL (MID$(A$,7,2))>49GOTO 3010
3090: IF VAL (MID$(A$,9,2))<10R VAL (MID$(A$,9,2))>49GOTO 3010
3100: IF VAL (MID$(A$,11,2))<10R VAL (MID$(A$,11,2))>49GOTO 3010
3110: LZ(I,0)=VAL (MID$(A$,1,2))
3120: LZ(I,1)=VAL (MID$(A$,3,2))
3130: LZ(I,2)=VAL (MID$(A$,5,2))
3140: LZ(I,3)=VAL (MID$(A$,7,2))
3150: LZ(I,4)=VAL (MID$(A$,9,2))
3160: LZ(I,5)=VAL (MID$(A$,11,2))
3170: NEXT I

```

lassen. Dieser Idee liegt die Tatsache zugrunde, daß sich bisher noch nie eine Ziehungszahlenreihe wiederholt hat. Und wenn man der mathematischen Theorie glauben darf, wird eine derartige Sechserwiederholung auch noch sehr lange auf sich warten lassen.

Es ist doch aber möglich, daß eine vom Spieler getippte Zahlenreihe bereits in einer der ca. 1.700 vergangenen Ziehungen als "Sechser" herauskam. Die dazu notwendige mühsame Abgleicharbeit führt der PC-1600 mit dem vorhandenen Programm in ca. zwei bis drei Stunden durch. Mit etwas Erfahrung kann das Programm auch auf andere Pocketcomputer von SHARP übernommen werden.

Sollte sich also bei dem Abgleichvorgang herausstellen, daß entsprechend den vorgeannten Ausführungen bereits ein Sechser, oder auch ein Fünfer mit Zusatzzahl vorlag, so wäre vor dem Hintergrund der mathematischen Wahrscheinlichkeitstheorie die Überlegung sinnvoll, zur Gewinnoptimierung diese Zahlenreihe teilweise, oder ganz abzuändern.

Interessenten, denen es zu aufwendig erscheint, alle Lottozahlen in ihren Computer als Diskettendatei (ca. 35 Kilobytes) einzutippen, können den Datenbestand gegen Voreinsendung einer Leerdiskette und DM 39,- als Honorar für die Bearbeitung einschl. Versandkostenatz über die FISCHEL GmbH beim Autor anfordern. Wer nur einmalig seinen Lottotippschein überprüfen möchte, der kann diese Auswertung beim Autor ebenfalls gegen Voreinsendung von DM 20,- (Tippzettel beilegen) über die FISCHEL GmbH abrufen.

Geprüfte Tippreihen

	1.	5	7	16	21	24	26
2.	2	6	11	16	19	49	
3.	5	6	25	32	42	49	
4.	1	3	24	40	42	45	
5.	4	9	13	21	27	48	
6.	2	4	12	13	31	37	
7.	4	8	22	28	35	43	
8.	11	13	20	27	39	45	

```

3172:LPRINT TAB (6);"Zahlenlotto-Tippscheinanalyse über alle bisherigen Wochene
n";
3174:LPRINT TAB (65);"dziehungen"
3176:LF 2
3180:LPRINT TAB (30);"Geprüfte Tippzeihen" 3225:FOR J=0TO 5
3190:LPRINT TAB (30);"-----" 3230:LPRINT TAB ((J*3)+32);USING "###";LZ(I,J);
3200:FOR I=0TO 7 3240:NEXT J
3210:IF LZ(I,0)=99GOTO 3260 3245:LPRINT ""
3220:LPRINT TAB (30);STR# (I+1);". "; 3250:NEXT I
3260:LF 2

```

```

3264:LPRINT TAB (6);"-----"
-";
3266:LPRINT TAB (65);"-----"
3270:LPRINT TAB (6);"Ziehungs- erschienene Lottozahlen erzielte Treffer der"
;
3280:LPRINT TAB (64);"Tippzeihen"
3290:LPRINT TAB (6);"Woche, Jahr 1. 2. 3. 4. 5. 6. Zus.Z. 1. 2. 3. 4. 5. 6
.";
3300:LPRINT TAB (64);" 7. 8. "
3310:LPRINT TAB (6);"-----"
-";

```

```

3320:LPRINT TAB (65);"-----"
6000:OPEN "X:LO-ZAHLE"FOR INPUT AS #1
6010:IF EOF (1)THEN CLOSE #1:GOTO 100
6020:D#(0)="":INPUT #1,D#(0)
6030:Z1=0:Z2=0:Z3=0:Z4=0:Z5=0:Z6=0:Z7=0:JH=0:WO=0
6040:JH=VAL (MID# (D#(0),1,2))
6050:WO=VAL (MID# (D#(0),3,2))
6060:Z1=VAL (MID# (D#(0),5,2))
6070:Z2=VAL (MID# (D#(0),7,2))
6080:Z3=VAL (MID# (D#(0),9,2))
6090:Z4=VAL (MID# (D#(0),11,2))
6100:Z5=VAL (MID# (D#(0),13,2))
6110:Z6=VAL (MID# (D#(0),15,2))
6120:Z7=VAL (MID# (D#(0),17,2))
6130:FOR I=0TO 7:TR(I)=0:NEXT I

```

```

6140:FOR I=0TO 7.
6150:FOR J=0TO 5
6160:IF LZ(I,J)=21LET TR(I)=TR(I)+1:GOTO 6220
6170:IF LZ(I,J)=22LET TR(I)=TR(I)+1:GOTO 6220
6180:IF LZ(I,J)=23LET TR(I)=TR(I)+1:GOTO 6220
6190:IF LZ(I,J)=24LET TR(I)=TR(I)+1:GOTO 6220
6200:IF LZ(I,J)=25LET TR(I)=TR(I)+1:GOTO 6220
6210:IF LZ(I,J)=26LET TR(I)=TR(I)+1:GOTO 6220
6220:NEXT J
6230:IF TR(I)<>5GOTO 6270
6240:FOR J=0TO 5
6250:IF LZ(I,J)=27LET TR(I)=7:GOTO 6270
6260:NEXT J
6270:NEXT I
6280:FOR I=0TO 7
6290:IF TR(I)>(T1-1)GOTO 6320
6300:NEXT I
6310:GOTO 6010

```

```

6320:LPRINT TAB (8);USING "###";WO;;LPRINT TAB (11);"/";STR# (JH);
6330:LPRINT TAB (16);USING "###";Z1;
6340:LPRINT TAB (19);USING "###";Z2;
6350:LPRINT TAB (22);USING "###";Z3;
6360:LPRINT TAB (25);USING "###";Z4;
6370:LPRINT TAB (28);USING "###";Z5;
6380:LPRINT TAB (31);USING "###";Z6;" / ";
6390:LPRINT TAB (36);USING "###";Z7;
6400:FOR I=0TO 7
6405:IF TR(I)<T1GOTO 6420
6407:IF TR(I)=7THEN LPRINT TAB ((I*4)+43);"5mZ";:GOTO 6420
6410:LPRINT TAB ((I*4)+43);STR# (TR(I));
6420:NEXT I
6430:LPRINT ""
6440:GOTO 6010
9000:KILL "X:LO-DATUM"
9010:OPEN "X:LO-DATUM"FOR OUTPUT AS #1

```

```

9015:AS="":AS=AS+STR# (JH)+STR# (WO)
9020:PRINT #1,AS
9030:CLOSE #1
9999:END

```

Ziehungs- erschienene Lottozahlen erzielte Treffer der Tippzeihen
Woche, Jahr 1. 2. 3. 4. 5. 6. Zus.Z. 1. 2. 3. 4. 5. 6. 7. 8.

Woche, Jahr	1	2	3	4	5	6	Zus.Z.	1	2	3	4	5	6	7	8
42/56	32	10	11	25	12	42	/ 27			3					
49/56	15	9	47	10	27	21	/ 14					3			
4/57	2	39	6	9	28	49	/ 20	3							
16/57	35	44	11	39	27	45	/ 13							4	
18/57	31	41	4	2	27	22	/ 38						3		
21/57	24	35	3	45	40	1	/ 42			5mZ					
29/57	42	1	2	23	44	3	/ 47			3					
50/57	27	11	22	7	39	18	/ 46							3	
51/57	33	22	28	23	4	7	/ 9						3		
9/58	24	9	11	27	48	21	/ 42			4					
12/58	39	20	27	5	22	43	/ 17							3	
16/58	27	25	5	3	22	49	/ 7			3					
41/58	33	16	25	19	6	20	/ 2	3							
48/58	11	44	48	49	19	18	/ 37	3							
49/58	28	13	1	17	27	4	/ 46					3			
50/58	11	48	31	9	20	13	/ 1					3			
16/59	16	28	20	4	4	22	/ 30							3	
34/59	2	13	19	42	4	31	/ 4							3	

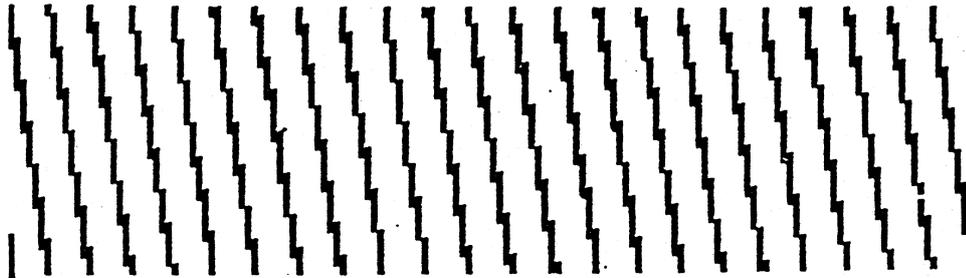
ZUFALLSZAHLENVERTEILUNG OPTISCH SICHTBAR GEMACHT

ZUFALLSZAHLENVERTEILUNG OPTISCH SICHTBAR GEMACHT (PC-1600)

Manche werden die Artikelreihe "Computer-Kurzweil" kennen, die regelmäßig in "Spektrum der Wissenschaft" erscheint. Im nun erschienenen Sonderheft findet sich im Beitrag "Wie man PI erschließt" von A. K. Dewdney ein einfaches Programm, das "zufällige" Punkte auf den Bildschirm schreibt: (Hier bereits für den PC-1600 umgeschrieben)

```
10 CLS:RANDOM 20 X=RND (156)-1 30 Y=RND (32)-1 40 PSET (X,Y) 50 GOTO 20
```

Die nun entstehenden Muster bei verschiedenen Durchläufen sind aber offensichtlich keinesfalls zufällig:



Das liegt sicherlich am rechnerinternen Algorithmus für die "Zufallszahlen". Für Würfelspiele mag er noch taugen, aber für ernsthafte Anwendungen würde man sich doch wünschen, daß nur eine möglichst gleichmäßige Verteilung in immer anderer Art bei jedem Lauf erscheint. Bevor hierfür ein simpler Trick verraten wird, noch ein paar "Verfeinerungen":

Das Programm sollte nicht ewig laufen und geBREAKt werden müssen, sondern wenn man mit einer Belegung des Displays fertig ist, aufhören; da es auch langweilig ist, immer nur einen Punkt aufleuchten und stehenbleiben zu sehen, soll noch ein "Zufallston" erzeugt werden. Schließlich sollte man das ganze Ergebnis auch schwarz auf weiß - wie oben nach Hause tragen können - hierfür eignet sich die "Hardcopy"-Routine von W. Klar aus dem Vorspann zu ?Kriz, Anwendungshandbuch, Fischel-Verlag. Alles zusammen sieht dann wie folgt aus:

```
10 "Z"CLS :RANDOM
20:FOR I=1TO 4985
30:X=RND (156)-1
40:Y=RND (32)-1
50:PSET (X,Y):BEEP 1,((Y+1)*7),120
60:NEXT I:BEEP 10,50,50:IF ASC INKEY# =12END
120:"HARDCOPY"AS=CHR# 80B:TEXT :CSIZE 1:PITCH 8,4:PCONSOLE "LPT1:",156,2,0
130:FOR Y=0TO 31:FOR X=0TO 156
140:IF POINT (X,Y)=1LPRINT AS;ELSE LPRINT " ";
150:NEXT X:LPRINT " ":NEXT Y:PCONSOLE "LPT:",0,0,0:CSIZE 2:LF 4
```

*Wobei man den Fehler aus Zeile 120 und 130 beseitigen sollte (156 durch 156 ersetzen, sonst fehlt die letzte Punktreihe)

Auf Wunsch kann man in Zeile 60 den Abbruch durch die ENTER-Taste auch als Normalfall und den Druck als Sonderfall zu jedem möglichen Zeitpunkt des Schleifenlaufs programmieren, dann ist einzufügen:

```
21 IF ASC INKEY# =13 GOTO "HARDCOPY" und Zeile 60 lautet nur NEXT I
```

Der versprochene Trick? Einfach die RND-Anweisung mit einer weiteren verschachteln! Aber bitte nicht nur in Zeile 40:

```
30 X = RND (156)-1 40 Y = RND (32)*RND (1)-1
```

verändert nämlich das Bild gar nicht wesentlich. Aber $30 X = RND (156)-1 * RND (1)-1$ liefert das gewünschte Ergebnis unabhängig von einer Verschachtelung in Zeile 40:



Wenn man es ganz kompliziert haben will, kann man natürlich auch ein

paar Verschachtelungen ineinanderbauen, nur sollte man sehr sorgfältig die Angaben zur RND-Anweisung beachten, um sicher zu gehen, daß auch nur Zahlen in einem Bereich erzeugt werden, in dem man sie wirklich will, für das Display also von 0-155 und 0-31. Eine "zufällig" falsche Zahl in einem langen Programm mit der Konsequenz eines ERRORS ist keine gute Überraschung. Eventuell deshalb ein paar Probelaufe der zufallerzeugenden Zeile durchführen!

Was bedeuten nun oben eigentlich die weißen Punkte, also die nicht "angeschalteten" Punkte des Displays? Sie sind zufällig nicht getroffen worden; wer will, kann ein Programm mit großem Speicheraufwand schreiben, das untersucht, welche Punkte zweimal und welche gar nicht in einer Serie von Läufen getroffen wurden. Wer nur zusehen will, wann in einem Lauf ein Punkt mehrmals getroffen wird, sollte den PSET-Befehl in der Form PSET (X,Y), X verwenden, wodurch er wieder gelöscht wird. Natürlich könnte man auch Effekte mit den Befehlen PRESET und POINT erreichen.

Lange RND-Anweisungen sind zudem sehr zeitintensiv. Wenn möglich, sollte man nur einmal einen Zufallswert berechnen und ihn dann in einer Variablen speichern, um ihn mehrfach zu verwenden. Den letzten Schriff erhält unser Punktmusterprogramm also durch eine Erweiterung der Zeile 20 um :A=RND (1) und Ersatz von RND (1) in den Zeilen 30 und 40, könnte man denken - dies ist gerade ein Fall, wo es nicht geht, wie die folgende Hardcopy zeigt:



Viel Spaß beim Versuch, die durchgehenden Querstreifen zu eliminieren! Und bei weiteren Versuchen mit Zufallszahlen ...

Jürgen Klein, Berlin

INKEY

Norbert von Holt.
Leibnitzstr. 4
3200 Hildesheim

```

3000:V=5:'VORGABE D. ZAHLENGROESSE+1
3010:DIM A$(7):1:'ANZAHL DER ZIFFERN EINER EINGABE
3020:CLS :USING ",*****.##"
3030:CURSOR 12,0:PRINT CHR$( &DD);"          ";CHR$( &DD):'EINGABEGRENZE
3040:WAIT 10
3050:FOR T=0TO U:'JE NACH VORGABE
3060:A$(T)=INKEY$
3070:IF A$(T)=CHR$( &D)AND T=1THEN LET T=U:GOTO 3110:'EINGABEENDE AB 1 ZIFFER
    MIT 'ENTER'
3080:IF A$(T)=CHR$( &18)THEN GOTO 3030:'EINGABE DURCH 'CL' LOESCHEN
3090:IF ASC (A$(T))<45OR ASC (A$(T))>57THEN LET A$(T)=":GOTO 3060:'NUR ZAHLEN
    UON 0-9 ZULAESSIG          4020:CLS :FT$=""          "H=3:U=1
3100:CURSOR 13+T,0:PRINT A$(T);          4030:CURSOR H,U:PRINT CHR$( &DD);FT$;CHR$( &DD)
3110:NEXT T          4040:WAIT 10
3120:FOR T=0TO U:A$=A$+A$(T):NEXT T          4050:FOR C=1TO 20
3140:G=VAL A$:A$="":'UEBERGABE          4060:B$(C)=INKEY$
3150:CURSOR 12,0:PRINT G          4070:IF B$(C)=CHR$( &D)AND C>1THEN LET C=20:GOTO 4120
4000:'          4080:IF B$(C)=CHR$( &18)THEN GOTO 4030
4010:DIM B$(20):1          4090:IF ASC (B$(C))<(&40)OR ASC (B$(C))>(&5A)THEN LET B$(C)=":GOTO 4060
          4100:CURSOR H+C,U:PRINT B$(C);
          4110:NEXT C
          4120:FOR C=1TO 20:B$=B$+B$(C):NEXT C
          4150:CURSOR H+1,U:PRINT B$
          4160:B$=""

```

STATUS

```

10:"H":CLEAR :U$="*****"
20:CURSOR 0,0:PRINT USING U$;"MEMORY :";STATUS 258
30:CURSOR 0,1:PRINT USING U$;"MEM.PRG:";STATUS 1
40:USING
50:CURSOR 0,2:PRINT "LO-ADR :";HEX$( STATUS 2-1-32768);" ";STATUS 256
60:CURSOR 0,3:PRINT "HI-ADR :";HEX$( STATUS 3-1-32768);" ";STATUS 257
70:CURSOR 17,0:PRINT "S1: 00000"
80:CURSOR 17,1:PRINT "S2:";OSKF "S2:"
90:CURSOR 17,2:PRINT " X:";OSKF "X:"
100:CURSOR 24,3:PRINT "->"
110:W$=INKEY$ :IF W$(CHR$(13))THEN 110ELSE CLS.

```

Marcel Villimann
Hergiswaldstr. 4
CH - 6012 Oberhau

2 PROGRAMME IN DEM PC-1600 EINLADEN I

Hier mit Sende ich Ihnen ein Programm zum PC - 1600 und Floppy.
Da ich Früher probleme hatte zwei programme in den computer zuladen,
schreib ich ein kleines listing.

Die anleitung zum Programm:

Man starte es mit (RUN "POKE")
danach muss man den namen eingeben, wo man in den speicher laden will.
danach erscheint eine Nummer am rechten Display hälfte.
Die zahl muss mit einem XPOKE (DISPLAY NUMMER), 255 quittiert werden.
Am schluss muss man noch XPOKE &7865, &40, &c5 eingeben.
Und so weiter bis man keinen speicher mehr hat.

ACHTUNG:

- Die Programme müssen mit Namen aufrufbar sein zb. DEF "A" ; RUN "A".
- Listen muss man das Programm zuerst starten danach "BRAKEN".
- Auf die Zeilennummer muss man nicht achten.

```
10:"POKE"CLS
20:INPUT "ANLEITUNG (J)(N) ";Z$
30:IF Z$="J"GOSUB "ANLEITUNG"
40:CLS :INPUT "WELCHES PROGRAM ? ";A$
50:A=(STATUS 2)/256:A=INT (A)
60:XPOKE &7865,A
70:B=(STATUS 2)-A*256
80:XPOKE &7866,B
90:XPOKE STATUS 2,0
100:PRINT (STATUS 2)-1
110:A$="X:"+A$:LOAD A$:END
120:"ANLEITUNG"CLS :WAIT :PRINT "ZUERST NAME DES PROGRAMM      EINGEBEN"
130:CLS :PRINT "DANACH ERSCHEINT EINE      NUMMER AM RECHTEN RAND "
140:CLS :PRINT "DIE NUMMER MUSS MIT      < XPOKE(NUMMER),255>      EINGEBEN W
ERDEN"
150:CLS :PRINT "DANACH MIT:      <XPOKE&7865,&40,&C5>      ABGESCHLOS
SEN WERDEN"
160:CLS :PRINT "MIT <LLIST> KANN MAN DIE BEFEHLE AUSDRUCKEN"
170:RETURN
180:"***XPOKE(DISPLAY NUMMER),255***"
190:"***XPOKE &7865,&40,&C5      ***"
```

PC-1600:OPTI-CHAR

Thomas Jeger - Hauptstrasse 142 - CH-3286 Muntelier - Schweiz

Dieses Programm ermöglicht die Neudefinierung der Codes &80 -
&FF. Sie können also Ihre eigenen Tasten definieren. Ein Teil
dieser Codes können Sie dann auch über die KB-II-Taste mittels
Tastendruck ansprechen. Somit löst sich auch das Problem des
Scharfen S (S). Aber: Sie können diese neue Codes nicht aus-
drucken ! Beim Versuch, diese mittels LLIST oder LPRINT auszu-
drucken, zeichnet der Plotter unverständliche Zeichen oder er
stürzt ab. In einem solchen Moment drücken Sie die Reset-Taste.
Nach einem Reset dieser Art werden die Pointer, die auf die
neue Tabelle zeigen, zurückgesetzt; man muss sie neu setzen.
Dies machen Sie wie folgt: POKE &F064,PEEK &F034,PEEK &F035,0.

Doch nun zum Programm: Nachdem Sie das Programm abgetippt, ge-
sichert und gestartet haben, fragt Sie der Computer, ob Sie
die Tabelle schon kopiert haben. Beim ersten Start ist dies
nicht der Fall und Sie drücken "N". Jetzt wird im Speicher ein
Platz reserviert, der durch Daten oder Programme nicht zer-
stört werden kann. Auch durch einen Reset (ohne CE-Taste) wird

er nicht zerstört. Einen solchen Raum reserviert man sich übrigens wie folgt: A=(Anzahl Bytes):CALL &O2DD,A. (Zeile 60). In &FO35 und &FQ34 steht die Adresse, von welcher an der Platz reserviert wurde. Nach dem Kopieren sehen Sie einen Raster auf dem Display. Oben links blinkt ein Cursor. Dieser bewegen Sie mit den Tasten 4,2,6,8 nach links, unten, rechts oder hinauf. In der Mitte des Displays sehen Sie das zu bearbeitende Zeichen (Bei Neustart, das Zeichen &R0). Dieses ändern Sie mit den Tasten + und -. Wollen Sie das Zeichen in den Raster kopieren, drücken Sie die ←-Taste. Falls Sie das Zeichen im Raster auf das richtige Zeichen kopieren wollen, drücken Sie ►. Die C-Taste löscht den Raster und die I-Taste invertiert ihn. VIEL SPASS !!

```

5:GOTO 30
6:"A"AC=AA+(AD-&80)*6:RETURN
10:"C"XC=X0+(X-1)*3:YC=Y0+(Y-1)*
  3
20:RETURN
30:CLS:PRINT "      OPTI-CHAR
  ----"
40:PRINT:PRINT "      Schon kopier
  t (J/N):"
42:A$=INKEY$(1)
45:IF A$="J"THEN 100
47:IF A$="I"THEN 90
50:IF A$=""THEN 42
60:CLS:PRINT "      Momentchen bit
  te..."
65:A=700:CALL &O2DD,A
70:AA=PEEK &F035*256+PEEK &F034
75:EE=PEEK &F064+PEEK &F065*256
80:FOR I=0TO 640:POKE AA+I,PEEK
  (EE+I):NEXT I
90:POKE &F064,PEEK &F034,PEEK &F
  035,0
100:CLS
101:AA=PEEK &F035*256+PEEK &F034
110:LINE (0,0)-(25,31),,,BF
120:LINE (2,2)-(23,29),R,,BF
130:LINE (3,3)-(22,28),,,B
140:X0=5:Y0=5:X=1:Y=1:AD=&80
150:DIM A$(6,8)
160:FOR X=1TO 6:FOR Y=1TO 8:GOSUB
  "C"
170:LINE (XC-1,YC-1)-(XC+1,YC+1),
  R,,BF
180:IF A$(X,Y)=""THEN PSET (XC,YC
  )
190:IF A$(X,Y)<>""THEN LINE (XC-1
  ,YC-1)-(XC+1,YC+1),,,BF
200:NEXT Y:NEXT X
210:X=1:Y=1
215:CURSOR 7,0:PRINT "Zeichen:";
  CHR$ AD
220:"CURSOR
230:GOSUB "C"
240:LINE (XC-1,YC-1)-(XC+1,YC+1),
  R,,BF
250:FOR I=1TO 10:A$=INKEY$:IF A$
  =""THEN NEXT IELSE LET I=99:
  NEXT I
260:IF I=100GOTO 310
270:LINE (XC-1,YC-1)-(XC+1,YC+1),
  ,,B

```

```

280:FOR I=1TO 10:A$=INKEY$:IF A$
  =""THEN NEXT IELSE LET I=99:
  NEXT I
290:IF I=100GOTO 310
300:GOTO 240
310:LINE (XC-1,YC-1)-(XC+1,YC+1),
  R,,BF
320:IF A$(X,Y)=""THEN PSET (XC,YC
  )
330:IF A$(X,Y)<>""THEN LINE (XC-1
  ,YC-1)-(XC+1,YC+1),,,BF
340:IF A$="8"AND Y>1THEN LET Y=Y-
  1
350:IF A$="2"AND Y<8THEN LET Y=Y+
  1
360:IF A$="4"AND X>1THEN LET X=X-
  1
370:IF A$="6"AND X<6THEN LET X=X+
  1
380:IF A$="5"THEN LET A$(X,Y)="1"
390:IF A$="0"THEN LET A$(X,Y)=""
400:IF A$="C"THEN ERASE A$(X,Y):GOTO
  150
410:IF A$="+"AND AD<&FFTHEN LET A
  D=AD+1:GOTO 215
420:IF A$="-"AND AD>&80THEN LET A
  D=AD-1:GOTO 215
430:IF A$=CHR$ 8THEN GOTO 1000
435:IF A$=CHR$ 12THEN GOTO 1100
440:IF A$="I"THEN GOTO 1200
999:GOTO 220
1000:" IN RASTER KOPIEREN
1005:CURSOR 7,0:PRINT "MOMENT ??
1010:GOSUB "A"
1020:FOR I=0TO 5
1030:B=PEEK (AC+I)
1040:FOR J=7TO 0STEP -1
1050:IF B<2^JTHEN LET A$(I+1,J+1)=
  ""ELSE LET A$(I+1,J+1)="1":B
  =B-2^J
1060:NEXT J:NEXT I:GOTO 160
1100:"Zurueckkopieren
1105:CURSOR 7,0:PRINT "MOMENT ??
1110:GOSUB "A"
1120:FOR I=0TO 5:B=0:FOR J=0TO 7
1130:B=B+(A$(I+1,J+1)="1")*2^J
1140:NEXT J:POKE AC+I,B:NEXT I:
  GOTO 215
1200:"INVERTIEREN
1210:FOR I=1TO 6:FOR J=1TO 8
1220:IF A$(I,J)="1"THEN LET A$(I,J
  )=""ELSE LET A$(I,J)="1"

```

Punkt setzen
 Punkt löschen

PC-1600: einfache Textverarbeitung

Sie schreiben den Text einfach in ein Programm (siehe Bsp.). Sie dürfen aber nicht mehr als 3 Displayzeilen schreiben, das es sonst eine Fehlermeldung gibt (Error 120). Am besten geht es, wenn Sie die Zeilennummern mittels AUTO bestimmen, danach ein ' eingeben und den Satz hinschreiben. Am Schluss drücken Sie den Brief mittels "LL.*"

```
s. 10:'PC-1600: einfache Textverarbeitung
    20:'
    30:'
    40:'Sie schreiben den Text einfach in etc...
```

PC-1600:Monitor

Das Programm ist frei im Speicher verschiebbar, also an keine fixe Adresse gebunden. Dennoch ein Vorschlag: NEW "SO:",300 Jetzt ab &C100 alle Daten eingeben. Aufruf des Monitors: CALL X;Y (Y: darzustellende Adresse).

Mit den Cursortasten kann man im Speicher "wandern" und mit den Zahlen und Buchstaben die Werte ändern. Es werden immer 8 Byte angezeigt. Darauf folgt eine Prüfsumme. Die Operationszeichen (=,.,+,-,/) sind mit den Buchstaben A - F belegt, was eine schnellere Eingabe der HEX-Werte erlaubt.

Mit freundlicher Genehmigung: Club des SHARPENTIERS

Rue du Temple 79

(TJ)

F-75003 Paris

```
C100:44 4D 03 0A FE 23 20 10 03 0A 06 30 F5 07 07 07 57 F1 07 02
C115:D3 31 03 0A FE 0D 20 FA 21 00 00 1E 04 08 0A FE 38 28 28 06 30
C12A:FE 0A 38 02 06 07 1D 28 08 1D 28 12 10 28 14 6F 1E 01 18 E4 07
C13F:07 07 07 85 6F 1E 02 18 DA 67 1E 03 18 D5 07 07 07 07 84 07 CD
C154:09 01 01 87 FB 3E 0D 1E 08 02 03 1D 20 F8 CD 12 01 11 00 00 CD
C169:15 01 1E 04 3E 08 F5 7C 18 1D 7D 18 1A 3E 20 CD 00 01 0E 00 06
C17E:08 7E F5 81 4F F1 18 0A 23 05 20 F5 3E 3A CD 00 01 79 F5 07 07
C193:07 07 16 01 F6 F0 27 C8 A0 CE 40 CD 00 01 15 20 03 F1 18 F0 F1
C1A8:3D F5 20 03 F1 18 0A FE 0A 28 C0 FE 09 28 BF 18 CD 78 FE 04 20
C18D:11 CD 18 01 18 00 1C 78 FE 04 20 SF 01 E0 FF 09 18 0A 3D 28 39
C1D2:3D 28 69 18 71 18 94 11 00 05 CD 15 01 3E 02 CD 1E 01 CD 06 01
C1E7:FE 0A 23 2A CD 18 01 7A F5 1C 01 08 00 09 78 FE 04 28 04 F1 57
C1FC:18 DE CD 2D 01 11 03 00 CD 15 01 1E 01 18 CC 01 F8 FF 09 1E 03
C211:F1 57 18 C7 FE 0E 20 01 C9 FE 08 20 30 CD 18 01 7A F5 1D 01 F8
C226:FF 09 78 FE FF 28 04 F1 57 18 E2 CD 30 01 11 00 00 CD 15 01 1E
C23B:02 18 99 01 F8 FF 09 1E 00 F1 57 18 CB 18 7F 18 8B 18 96 FE 0C
C250:20 0E CD 18 01 14 7A FE 15 20 04 3E 05 F5 18 90 FE 08 20 0E CD
C265:18 01 15 7A FE 04 20 C2 3E 14 F5 18 B1 06 0A FE 2F 28 2C 04 FE
C27A:2A 28 27 04 FE 2D 28 22 04 FE 2B 28 1D 04 FE 2B 28 18 04 FE 2E
C28F:28 13 D8 30 38 87 FE 0A 38 0A FE 11 38 AF D8 07 FE 10 38 A9 47
C2A4:CD 18 01 E5 D5 7A D8 05 F5 C8 3F 16 00 5F 19 F1 C8 47 28 18 7E
C2B9:E6 F0 80 77 D1 E1 D3 16 00 CD 15 01 1E 03 18 01 01 F8 FF 09 D1
C2CE:18 85 78 07 07 07 07 47 7E E8 0F 18 E0
```

Briefe schreiben mit dem PC-1600

***** Version 5 *****

Dieses Programm ist geeignet, mit dem Computer PC-1600 + CE-1600P + CE-1600F Standardbriefe zu schreiben und auf einer Diskette oder RAM-DISK unter einer Akten-Nr. abzuspeichern. Um die Briefe auf einer RAM-DISK zu speichern ist es notwendig einen RAM-Speicher in das Modulfach S2 zu stecken. Anschließend ist mit dem Befehl <INIT> das RAM-Modul S2 als RAM-DISK zu formatieren. Das Modulfach S1 sollte mit einem RAM von min. 16 KByte bestückt sein, zur Erweiterung des Arbeitsspeichers. Sie können mit diesem Programm auch Adressaufkleber in verschiedenen Größen und Farben, sowie Paketaufkleber ausdrucken. Wenn in nachfolgendem Text von einer Diskette die Rede ist, kann damit selbstverständlich auch die RAM-DISK gemeint sein. Den Datenspeicher können Sie nach dem Starten des Programmes mit DEF A oder RUN auswählen, oder im Hauptmenü auf die ENTER-Taste drücken.

Das Programm ist menügesteuert und erklärt sich damit von selbst.

STARTMÖGLICHKEITEN DES PROGRAMMES:

RUN ---> Achtung Datenverlust !!!
DEF A ---> Zuerst Datenspeicher auswählen (kein Datenverlust)
DEF S ---> Start ins Hauptmenü (kein Datenverlust)

Im Hauptmenü sind folgende Funktionen möglich:

- W = Brief bzw. Adresse neu schreiben
- T = Brief editieren
- D = Brief bzw. Adressaufkleber mit dem Drucker ausdrucken
- S = Brief bzw. Adresse auf der Diskette abspeichern
- L = Brief bzw. Adresse von der Diskette in den Computer laden
- F = Inhaltsverzeichnis der Diskette anzeigen oder ausdrucken
- K = Brief bzw. Adresse auf der Diskette löschen
- E = Programmende
- ENTER = Datenspeicher auswählen

Programmteil (W) - Brief bzw. Adresse neu schreiben -

Mit diesem Programmteil kann ein Brief (1) oder nur eine Adresse (2) neu geschrieben werden. Soll beim Schreiben eines neuen Briefes eine auf der Diskette gespeicherte Adresse übernommen werden, so ist dies unter (3) ebenfalls möglich.

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 Zeichen
2. Adresse ?	8	30 Zeichen
3. Betreff ?	8	63 Zeichen
4. Text ?	65	63 Zeichen
5. Anlage ?	8	63 Zeichen

Die jeweilige Eingabe bei der Adresse, beim Text und beim Anlagentext wird durch (E) + (ENTER) beendet.

Sie können bei der Anlage und dem Gruß unter 3 vorgegebenen Möglichkeiten auswählen, oder auch einen eigenen Text eingeben (Menüpunkt 4). Möchten Sie eine Leerzeile eingeben, so brauchen Sie nur auf die ENTER-Taste zu drücken.

Werden bei der Text-Eingabe mehr als 63 Zeichen in eine Zeile geschrieben, so wird der Rest abgeschnitten und die Zeile erscheint sofort wieder auf dem Display im Editiermodus um entsprechende Veränderungen vorzunehmen. 63 Zeichen sind auf

dem Display des PC-1600 ca. 2 x Zeilen. Ich empfehle, unter dem Anzeigedisplay eine Markierung anzubringen.

Es können alle Zeichen außer den Anführungszeichen über die Tastatur eingegeben werden, also auch Leerzeichen am Anfang einer Zeile und das Komma. Diese Zeichen werden beim Speichern auf der Diskette in andere Zeichen umcodiert bzw. beim Laden von der Diskette in ihre ursprünglichen Zeichen zurückcodiert. Das 8-Zeichen kann durch das f-Zeichen (KBI1 + []), eingegeben werden, es wird beim Ausdruck in ein 8-Zeichen umgewandelt.

Programmteil (T) - Brief editieren -

Dieser Programmteil ist gänzlich menügesteuert und erklärt sich damit von selbst. Während bei der Adresse und beim Anlagentext immer die erste Zeile zum Editieren angewählt wird, muß bei den Textzeilen die gewünschte Zeilenzahl zuerst eingegeben werden. Ist die zu editierende Zeile auf dem Display sichtbar, so können folgende Funktionen gewählt werden:

- ↑ = eine Zeile zurück
- ↓ = eine Zeile vorwärts
- + = Zeile einfügen, nach der gerade auf dem Display angezeigte Zeile
- = Zeile löschen
- x = Zeile editieren
- M = zurück zum Menü

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 (mit ↑ anwählen).

Programmteil (D) - Brief bzw. Adressaufkleber ausdrucken -

Dieser Programmteil bietet folgende Möglichkeiten:

- 1 = Brief mit oder ohne Bankverbindung drucken
- 2 = Adressaufkleber in 6 verschiedenen Größen drucken
- 3 = Adressaufkleber mit Absender drucken
- 4 = Paketaufkleber mit Absender drucken

Beim Ausdruck eines Briefes erscheint auf dem Display die Aufforderung ein DIN-A4-Blatt bis zur Abbriskante des Druckers einzuspinnen und mit (J) + (ENTER) zu bestätigen. Ist die Adressenanzahl + Textzeilenanzahl + Anlagenzeilenanzahl gleich oder größer 37, so wird der Druckvorgang gestoppt. Auf dem Display erscheint dann die Aufforderung, ein zweites A4-Blatt bis zur Abbriskante einzuspinnen und mit (J) + (ENTER) zu bestätigen.

Beim Ausdruck einer Adresse oder eines Paketaufklebers wird die Adresse des momentan in den Computer geladenen Briefes verwendet. Selbstverständlich können auch reine Adressen geschrieben, auf der Diskette gespeichert und mit dem CE-1600P gedruckt werden.

Vor der Benutzung des Programms sind folgende persönliche Daten zu programmieren:

Zelle 100 : Vorname + Nachname
Zelle 170 : Straße + Hausnummer
Zelle 180 : Wohnort
Zelle 190 : Postleitzahl
Zelle 200 : Telefon-Nr.
Zelle 210 : Bankverbindung
Zelle 220 : Jahreszahl z.B. 87

Programmteil (S) - Brief/Adresse auf der Diskette abspeichern -

Der Brief bzw. die Adresse kann auf der Diskette unter einer Akten-Nr. (beliebige Zeichenfolge von Großbuchstaben und Zahlen, jedoch max. 8 Zeichen) abgespeichert werden. 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. Ich empfehle zur Unterscheidung von Brief- und Adressdaten, als ersten Buchstaben B für einen Brief und A für eine Adresse zu verwenden.

z.B. B-01 bis B-99 = 99 Briefdateien
A-01 bis A-99 = 99 Adressdateien

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

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

Programmteil (K) - Brief/Adresse auf der Diskette löschen -

Eine Brief- bzw. Adressdatei kann unter der entsprechenden Akten-Nr. auf der Diskette gelöscht werden. Damit dies nicht unbeabsichtigt geschieht, fragt der Computer den Benutzer, ob die Datei auch wirklich gelöscht werden soll. Wird dies verneint durch drücken der Taste <N>, verzweigt das Programm sofort zum Hauptmenü. Die Datei bleibt auf der Diskette gespeichert. Durch drücken der Taste <J> = Ja, wird die Datei auf der Diskette gelöscht.

Programmteil (F) - Inhaltsverzeichnis der Diskette anzeigen -

Damit Sie die Brief- bzw. Adressdateien auf der Diskette wieder finden, besteht mit diesem Programmteil die Möglichkeit, das Inhaltsverzeichnis der Diskette auf dem Display anzuzeigen oder, mit dem Drucker auszudrucken.

Preis: 98.-DM inkl. 14% MuSt.

Dieses Programm ist geeignet, mit dem Computer PC-1600 + CE-1600P + CE-1600F Standardbriefe zu schreiben und auf einer Diskette oder RAM-DISK unter einer Akten-Nr. abzuspeichern. Das Modulfach S1 sollte mit einem RAM von min. 16 KByte bestückt sein, zur Erweiterung des Arbeitsspeichers. Im Modulfach S2 können Sie einen RAM-Speicher als RAM-DISK initialisieren. Der entsprechende Datenspeicher kann nach dem Starten des Programmes ausgewählt werden. Das Programm ist Menügesteuert und erklärt sich von selbst.

Funktionen des Hauptmenüs:

- Brief bzw. Adresse neu schreiben.
- Brief editieren
- Brief bzw. Adressaufkleber mit dem Drucker ausdrucken
- Brief bzw. Adresse auf der Diskette abspeichern
- Brief bzw. Adresse von der Diskette in den Computer laden
- Inhaltsverzeichnis der Diskette anzeigen oder ausdrucken
- Brief bzw. Adresse auf der Diskette löschen

Funktionen des Editiermenüs:

- <↑> eine Zeile zurück, <↓> eine Zeile vorwärts
- <+> Zeile einfügen, <-> Zeile löschen
- <*> Zeile editieren, <M> zurück zum Hauptmenü

Funktionen des Druckermenüs:

- Brief mit oder ohne Bankverbindung drucken
- Adressaufkleber in 6 verschiedenen Größen drucken
- Adressaufkleber mit Absender drucken
- Paketaufkleber mit Absender drucken

Beim Ausdruck einer Adresse oder eines Paketaufklebers wird die Adresse des momentan in den Computer geladenen Briefes verwendet. Selbstverständlich können auch reine Adressen geschrieben, auf der Diskette gespeichert und mit dem CE-1600P gedruckt werden.

Eine umfangreiche Programmbeschreibung wird mitgeliefert. Alles in allem, ein Super-Briefprogramm, das wohl alle PC-1600-Besitzer begeistern wird.

Programmlänge: ca. 16000 Bytes
Preis: 98,- DM (inkl. 14 % Mehrwertsteuer und Diskette)
Bestellungen an die Fischel GmbH, 1000 Berlin

PC-1600 RECHNERSCHUTZ

Thomas Spitzer, Birkenredder 4, 2000 Hamburg 67

Dieses Programm ist in der Lage lästige Fummler sicher abzuweisen. Das Programm wird beim Ausschalten mit DEF "S" aktiviert, schaltet man mit OFF aus, so ist der Schutz nicht aktiviert. Wird der Rechner eingeschaltet, wenn der Schutz aktiviert ist, so muss eine Tastenfolge eingegeben werden. Man hat ca. 2 sec. Zeit, um die nächste Taste zudrücken, drückt man in dieser Zeit keine, oder die falsche Taste, so beepet der Rechner auch schon los. Zusätzlich wird ein blinkender Text (siehe Zeile 6) angezeigt. Nach kurzer Zeit schaltet sich der Rechner selbstständig aus. Dadurch, dass das BREAK OFF schon vor dem Ausschalten steht, ist es absolut unmöglich das Programm zu unterbrechen. Die Tastenfolge ist in der DATA-Zeile 65278 einzugeben. (Tabelle der möglichen Zeichen siehe unten INKEY\$ in der Bedienungsanleitung). Das Ende der Tastenfolge wird durch ein "ENDE" markiert, so sind beliebig lange Tastenfolgen möglich. Ausserdem wird, nachdem die Zeichenfolge korrekt eingegeben wurde, eine "1" angezeigt, wenn keine Fehlversuche gemacht worden sind. Sind Fehlversuche gemacht worden, so wird die Zeit des letzten Fehlversuches angezeigt. Auf diese Weise kann festgestellt werden, ob jemand versucht hat das Gerät unerlaubterweise einzuschalten. Wem die Zeit, die man hat um die nächste Taste zu finden, falsch bemessen vorkommt, möge in Zeile 4 die 75 entsprechend ersetzen.

A-C-H-T-U-N-G !!

Bei der Ersteingabe das BREAK OFF in der Zeile 65279 weglassen. und das Programm ausprobieren, da bei einem Programmfehler oder wenn man die Tastenfolge nicht weis, man einen VOLLSTÄNDIGEN R-E-S-E-T durchführen muss !!.

Dies Programm kann selbstverständlich auch als Programmschutz verwendet werden. Dazu muss das zuschützende Programm zwischen die Zeilen 8 u. 65278 geschrieben werden: Wenn der Rechner nicht benutzt werden kann, kann das Programm im Rechner weder benutzt, noch eingesehen, noch verändert werden.

Torbjørn Hovden
Konvallveien 23
N-3900 PORSGRUNN

Sehr geehrte Damen und Herren,
Fuer den PC-1600 habe ich folgende Tips & Tricks:

MODE WECHSELN

In A.f.S.C Nr. 21 Seite 8 1986, zeigt Frank Dabringhausen wie man waehrend des Programmablaufes zwischen MODE 0 und MODE 1 wechseln kann. Diese Methode hat nach meiner Meinung folgende Nachteile:

- Innerhalb eine FOR ... NEXT Schleife oder einer Subroutine ausgefuehrt, fuert es zum ERROR 2 wenn NEXT bzw. RETURN erreicht wird.

- Die Zeilennummern die innerhalb Anfuierungs-Zeichen stehen, wird von RENUM nicht beruecksichtigt..

Unten finden Sie Herrn Dabringhausen's original Routine, und meinorschlag zur Verbesserung seiner sehr gute Idee:

```
10 REM ORIGINAL ROUTINE
20 REM Ablauf in MODE 0
30 ...
40 KBUFF$ ="MODE1"+CHR$ 13+"GOTO60"+CHR$ 13
50 END
60 REM Ablauf in MODE 1
70 ...
```

```
10 REM MEINE ROUTINE
20 REM Ablauf in MODE 0
30 ...
40 WAIT :KBUFF$ ="MODE1"+CHR$ 13+CHR$ 13:
PRINT :WAIT 0
50 REM Ablauf in MODE 1
60 ...
```

```
1:ARUN :RESTORE "S":IF S0=0:RESTORE :END
2:READ S$:IF S$="ENDE":RESTORE :PRINT S0:S0=0:BREAK ON :END
3:IF INKEY$ <>"" THEN 3
4:FOR I=0TO 75:IF INKEY$ <>"" LET I=75:NEXT I:IF INKEY$ ="S" THEN 2 ELSE 6
5:NEXT I
6:OUT 23,65:PRINT "Bitte erst Thomas fragen, ob Du mich benutzen darfst!!!!"
7:FOR I=0TO 10:OUT 80,62:FOR J=0TO 50:NEXT J:OUT 80,63:FOR J=0TO 150:NEXT J:
NEXT I:POWER OFF
65278:"S"DATA "F","I","S","C","H","E","L","E","NDE"
65279:BREAK OFF :S0=1:POWER OFF
```

CE-158 CENTRONICS SCHNITTSTELLE

Wenn man einen Drucker vom PC-1600 ueber CE-158 ansteuert, kann es Probleme geben: Der CE-158 kennt den Befehlsatz des PC-1600 nicht. Betrachten Sie folgendes Beispiel-Programm:

```
100 INPUT "DATO ODER UHRZEIT? (D/U)?" ;D$
110 IF D$="D"PRINT DATE$ ELSE PRINT TIME$
120 GOTO 100
```

Dieses Programm soll ueber CE-158 ausgelistet werden:

```
MODE 1
OPN "LPRT"
CONSOLE 0,0
LLIST
```

Die PC-1600 spezifische Befehle, wie DATE\$ und ELSE, werden durch "" ersetzt, und das Listing wird schwer verstaendlich. Die Zeilennummern in Sprungbefehlen werden im PC-1600 als Binaerkode gespeichert (immer 2 Byte), und nicht als ASCII wie beim PC-1500. Der CE-158 uebersetzt diese Binaerkodes nicht in ASCII, und statt Zeilennummern schreibt der Drucker seltsame Zeichen. Eine listing mit Brother HR-15 kann so aussehen:

```
100:INPUT "DATO ODER UHRZEIT?_(D/U)?" ;D$
110:IF D$="D"PRINT PRINT
120:GOTO d
```

Es gibt mehrere Wege dieses Problem zu vermeiden. Wenn der Drucker eine RS232C Schnittstelle besitzt, kann man ganz einfach den CE-158 in eine Schublade werfen, und die in PC-1600 enthaltene RS232C benutzen. Falls

der Drucker statt RS232C eine Centronics Schnittstelle hat, geht das nicht.

Man kan natuerlich einen RS232C/Centronics Umformer kaufen, und das Rechnerinterne Serieninterface benutzen, aber solche Wandler sind teuer. Besser waere es dann ein Maschinenkode Programm zu schreiben, das die CE-158 Parallelschnittstelle als ein PC-1600 Peripherie Port definierte. Es waere eine gute Idee, diese Programm so zu schreiben dass man die Parallelschnittstelle auf gleiche Art benutzen kann wie COM1n: oder LPTn:. Es koennte z.B. LPT3: oder COM3: heissen.

Fuer die traege gibt es noch eine Lesung, die ich hier beschreiben will. Angenommen das obige Programm ist in Programmspeicher: SAVE'n Sie das Programm mit 'SAVE"d:name.bas",A'; dabei ist d: entweder S1:, S2:, X:, Y:, oder CASS:. Damit ist das Programm als eine ASCII Datei gespeichert, d.h. alle Tokens und Zeilennummern sind als ASCII Zeichen gespeichert. Laden Sie dann das Programm mit 'LOAD*d:name.bas". Das Programm kann jetzt durch

```
MODE 1
OPN "LPRT"
CONSOLE 0,0
LLIST
```

gelistet werden. Das listing enthaelt das eigentliche Programm in REM Zeilen, und jede Zeile hat 2 Zeilennummern. Entfernt man die erste Zeilennummern und die "" Zeichen, so wird das Listing folgendermassen:

```
100 INPUT "DATO ODER UHRZEIT? (D/U)?" ; D$
110 IF D$="D" PRINT DATE$ ELSE PRINT TIME$
120 GOTO 100
```

SERIELLE DATENUEBERTRAGUNG

Wenn mann eine Datei, z.B. eine Programm ueber einen seriellen Port empfaengt, muss dieser unbedingt mit CHR\$ 26 abgeschlossen werden, da den Sharp CHR\$ 26 als EOF (End of File) Marke verwendet. Falls keine CHR\$ 26 empfaengt wird, muss die Uebertragung mittels <BREAK> unterbrochen werden. Die laetzten empfaengten Daten, die sich immer in Empfaengspuffer befinden, geht dann verloren. Dies steht nicht in meine Bedienungsanleitung.

KEINE LINPUT

"," fungiert als Datentrenner bei sequentielle Dateien. Das kann problematisch sein wenn man Textdateien speichern, die "," enthaelt. Wenn die Datei mittels INPUT & gelesen wird, verschwindet die Komma's. Sehen Sie folgende Beispiel:

BENNO STURM

Münsterscher Damm 30
2906 Wardenburg

ZUFALLSBILD-GRAPHIK
PC-1600 & CE-1600P

1) DISPLAY-Graphik,

d.h. es lassen sich zusätzlich die DISPLAY-Punkte in der Anzahl, Grösse, Farbe und den Abständen sowohl in der Breite wie auch in der Höhe vorher festlegen.

Das Zufallsbild auf dem DISPLAY lässt sich je nach Belieben anhalten und dann ausdrucken.

Während des Druckes kann mit der Farbwechsellaste die Farbe beliebig gewechselt werden, so dass ein buntes Bild entsteht.

2) Linienmuster-Graphik,

d.h. nach Eingabe der Rechteckbreite, -höhe, Linienstruktur, Anzahl Rechtecke und Farbe werden nach dem 1. Rechteck alle weiteren Zufallsrechtecke innerhalb des 1. Rechteckes gezogen.

Um dies zu vermeiden, kann man statt "," (CHR\$ 44) CHR\$ (44+128) verwenden. Dies ist durch KBII, SHIFT, J, KBII zu erreichen. In meine Textverarbeitungsprogramm habe ich eine kleine Routine, die alle CHR\$ 44 mit CHR\$ 172 ersetzen bevor speichern, und umgekehrt bei lesen. Wenn ich die Datei mittels LOAD*

laden, findet selbsverstaendlich keine CHR\$ 44/CHR\$ 172 Wandlung statt, aber das macht nichts; Meine Typenradprinter (Brother HR15) beruecksichtigt Bit 7 nicht, und fuer ihn ist es ganz egal, entweder man CHR\$ 44 oder CHR\$ 172 zu ihn schreibt. (Wenn man seine Dateien ausschliesslich mittels LOAD* laedt, gibt das Problem nicht, weil den Sharp fuer LOAD* nur CHR\$ 13 als Datentrenner versteht).

Wenn bei den Sharp eine LINPUT befehl zu Verfuegung stand, waehre es kein Problem mehr, weil bei LINPUT nur CHR\$ 13 als Datentrenner verstanden werden soll.

```
OPEN "X:PROBE.TXT" FOR OUTPUT AS E1
PRINT E1," CIAO!"
PRINT E1,"PC-1600 MI PIACI, ANCHE TI?"
CLOSE E1
```

```
DIM A$(2)*80
OPEN "X:PROBE.TXT" FOR INPUT AS E1
INPUT E1,A$(0)
INPUT E1,A$(1)
INPUT E1,A$(2)
CLOSE E1
PRINT A$(0)
CIAO!
PRINT A$(1)
PC-1600 MI PIACI
PRINT A$(2)
ANCHE TI?
```

Bemerken Sie auch in das obige Beispiel, dass die vier Leerzeichen bevor "CIAO!" verschwunden ist. Leerzeichen am anfang einer Zeile wird von INPUT & nicht beruecksichtigt. Wenn man die erste Leerzeichen, CHR\$ (32) durch CHR\$(32+128) = CHR\$(160), oder KBII, A, KBII, ersetzt, kommt auch die Anfangsleerzeichen mit. Dies ist auch kein Problem mit LOAD*.

Es ist wohl bekannt dass als Antwort eine INPUT Frage keine ""s eingegeben werden kann. Man kann dasselbe Trick hier benutzen. " ist CHR\$ 34, und man kann CHR\$ 162 benutzen. CHR\$ 162 ist durch KBII, F, KBII zu erreichen.

Wenn man eine Drucker mit IBM kompatibeles Zeichensatz besitzt, kann die oben genannte Tricks nicht verwendet werden, da Bit 7 hier beruecksichtigt wird.

```
10: Benno Sturm
20: Münsterscher Damm
   30
30: 2906 Wardenburg
40:
50: PC-1600 / CE-1600
   P / CE-1600 F
60:
70: CLS : CLEAR : WAIT
80: PRINT " Zufallsb
   ild-Graphik
   -----
90: PRINT "Display-Gra
   phik DEF D Lin
```

```

ienmuster-Graphik
DEF L"
100:CLS :END
110:'
120:'
130:'Start DISPLAY-Graphik
140:'
150:'
160:"D"CLS :CLEAR :
TEXT :WAIT 10
170:PCONSOLE "LPT1:",8
,0,0,0
180:PRINT " Grap
hikformat
-----"
190:INPUT "Ausdruck de
r Werte?(J/N):";G$
:COLOR 0:CLS
200:PCONSOLE "LPT1:",8
0,0,34
210:'
220:'Eingabe der Daten
230:'
240:B=50:INPUT "DISP.-
Br., <155 (+1):";B
:IF B>155GOTO 240
250:H=10:INPUT "DISP.-
Hö., <31 (+1):";H
:IF H>31GOTO 250
260:C=0:INPUT "Punktbr
eite(1=0.2mm):";C
:IF B>C>930GOTO 260
270:D=0:INPUT "P.-Abst
. waag. u. se.:";D
:IF B>D>930GOTO 27
0
280:E=0:INPUT "Punkthö
he :";E
:IF H>E>930R E>C
GOTO 280
290:K=0:INPUT "Farbe,
0-3 :";K
:IF K>3GOTO 290
300:CLS
310:IF G$="N"THEN "B"
320:'
330:'Ausdruck der Date
n
340:'
350:LPRINT B;H;-C;D;E;
-K:TEXT :LF 1
360:M=(980-(B*D))/24
370:LET L=M:PCONSOLE "
LPT1:",8,0,L
380:'
390:'Anzeige der DISPL
AY-Graphik
400:'
410:"B"COLOR K:LINE (Z
,Y)-(-Z+B,-Y+H),X,
,BF
420:RANDOM :Z=RND B:Y=
RND H:WAIT 80
430:IF ASC INKEY$ =13
GOSUB "Abdruck"
ELSE GOTO 410
440:'
450:'Stop nach Wunsch
mit BREAK
460:'BREAK-Anzeige mit
RCL löschen
470:'Ausdruck der DISP
LAY-Graphik mit DE
F A
480:'
490:"A"GRAPH
500:FOR Y=0TO H:FOR X=
0TO B
510:IF POINT (X,Y)=1
GOSUB 540ELSE
GLCURSOR (X,Y*-D)
520:NEXT X:NEXT Y:TEXT
530:PCONSOLE "LPT1:",8
0,0,0:CLS :LF 5:
END
540:FOR U=1TO C
550:LLINE (X*D+U,Y*-D)
-(X*D+U,Y*-D+E):
NEXT U:RETURN
560:'
570:'
580:'Start Linienmuste
r-Graphik
590:'
600:'
610:"L"CLS :CLEAR :
TEXT :WAIT 10
620:PRINT " Grap
hikformat
-----"
630:INPUT "Ausdruck de
r Werte?(J/N):";G$
:CLS :COLOR 0
640:PCONSOLE "LPT1:",8
0,0,34
650:'
660:'Eingabe der Daten
670:'
680:B=400:INPUT "AUSDR
.-Br.<900, 5=1mm:"
;B:IF B>900GOTO 68
0
690:H=400:INPUT "AUSDR
UCK-Höhe,<900 :";
;H:IF H>900GOTO 69
0
700:O=0:INPUT "Linienst
ruktur,0-9 :";O
:IF O>9GOTO 700
710:N=5:INPUT "Anzahl
Rechtecke :";N
:IF N>20GOTO 710
720:W=0:INPUT "Farbe,
0-3 :";W
:IF W>3GOTO 720
730:CLS
740:IF G$="N"THEN GOTO
830:GRAPH
750:'
760:'Ausdruck der Date
n
770:'
780:LPRINT B;H;-O;N;W:
TEXT
790:'
800:'Ausdruck der Linien
muster-Graphik
810:'Stop mit BREAK
820:'
830:PCONSOLE "LPT1:",8
0,0,0
840:LET I=H
850:GRAPH :GLCURSOR ((
960-B)/2,-I):SORGN
860:FOR L=1TO N
870:LLINE (Z,Y)-(-Z+B,
-Y+H),O,W,B
880:RANDOM :Z=RND B:Y=
RND H
890:NEXT L:TEXT :LF 5:
GRAPH :GLCURSOR (0
,0):END
900:IF ASC INKEY$ =13
GOSUB ELSE GOTO 87
0

```

10 20-5 10 3-1

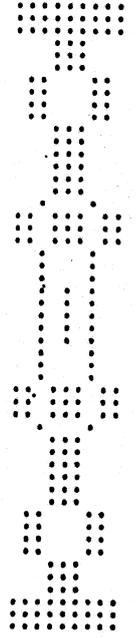
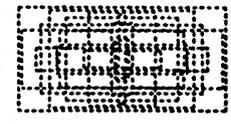
200 20 0 20 0

100 200-3 15 0

30 2-20 22 6 0

STATUS 1 2186

50 8-4 12 2-3



Programmbeschreibung für ELEKTRONISCHES TASCHENBUCH

Das Programm ist, wie der Name schon sagt ein Terminkalender b.z.w. ein Taschenbuch. Man kann damit Daten eingeben und wieder löschen.

Mit DEF A gelangt man zum Hauptmenue. Dort erscheint dann:

- 'H' für HEUTE
- 'M' für MORGEN
- 'N' für NEUES DATUM
- 'L' für LISTEN

Bei 'H' nimmt der computer das heutige Datum. Bei 'M' das morgige und bei 'N' ein Neues das extra eingegeben werden muß.

Bei 'L' werden die noch vorhandenen Datensätze gelistet.

Nach 'H', 'M' oder 'N' fragt der Computer was an diesem Datum ist (Max. 30 Zeichen). Danach gelangt man ins Hauptmenue zurück.

Bei 'L' werden die Datensätze folgendermaßen aufgelistet:
 Nummer X / AM XX.XX.XX (eingegebenes Datum) danach 'ENTER'
 XXXXXXXX (eingegebener Datensatz zu obengenannten Datum)

ocomputer, Tel.: 0
30/323 60 29"
330:
340:TEXT :COLOR 0:LF 2
350:LPRINT "Mit der Bi
tte um:

Anlage:"
360:LF 1:CSIZE 1:
PCONSOLE "LPT1:",2
55,0,46:LF -8
370:LPRINT "O Kennzn
isnahme O E
rledigung":LF 1
380:LPRINT "O Stellu
ngnahme O R
ücksprache":LF 1
390:LPRINT "O Anruf
O B
earbeitung":LF 1
400:LPRINT "O Rückga
be O P
rührung":LF -7

410:PCONSOLE "LPT1:",2
55,0,124
420:LPRINT "O Schrei
ben O Ze
ichnung":LF 1
430:LPRINT "O Rechnu
ng O An
trag":LF 1
440:LPRINT "O Kopien
O Fo
rmular"
450:CSIZE 2:PCONSOLE "
LPT1:",80,0,46:LF
1:COLOR 2:LPRINT "
Notizen":COLOR 0
460:CSIZE 1:PCONSOLE "
LPT1:",255,0,12:LF
1:COLOR 1
470:
480:LPRINT "FISCHEL Gm
bH, Kaiser-Friedri
ch-Str. 54a, D-Ber
lin 12"

490:
500:CSIZE 2:PCONSOLE "
LPT1:",80,0,46:LF
8:COLOR 2
510:LPRINT "Datum:
Unterschrift":
LF 2
520:PCONSOLE "LPT1:",8
0,0,37:COLOR 0:
CSIZE 1
530:LPRINT " <hier abt
rennen >":LF -37:
CSIZE 2
540:PCONSOLE "LPT1:",8
0,0,4:COLOR 3
550:GRAPH :LLINE (0,0)
-(900,-4),,B:
LLINE (0,-2)-(900,
-6),,B:COLOR 0
560:LLINE (0,-62)-(210
,-64),,B:LLINE (2
10,-112)-(212,-27)
,,,B
570:LLINE (500,-62)-(6
75,-64),,B:LLINE
(675,-87)-(677,-27
) ,,,B:COLOR 2
580:LLINE (500,-122)-(6
00,-124),,B:
LLINE (610,-120)-(9
00,-120)
590:LLINE (900,-150)-(6
10,-150):LLINE (5
00,-180)-(900,-180
)
600:LLINE (900,-210)-(5
00,-210):LLINE (5
00,-240)-(900,-240
)
610:LLINE (900,-270)-(5
00,-270):LLINE (5
00,-300)-(900,-300
)

Peter Vogel

4400 Münster
Seppenradeweg 10

```

*****
= PC - 1600
= EIN ASSEMBLER OHNE ASSEMBLER?
*****

```

Meine Meinung dazu ist: Ja, es geht! Und das sogar recht komfortabel! Ein Listing als Lösungsvorschlag dazu ist beigelegt. Mein Verfahren, den Basic-Editor und -Interpreter dafür zu verwenden, soll ein Denk-anstoß sein. Weitere kreative Abwandlungen und Ergänzungen sind mit Sicherheit möglich.

Mein Verfahren sieht wie folgt aus:

- Im Basic-Programmier-Modus wird das Listing geschrieben. Bei Einhalten der Abstände (s. Listing) ist das Bild auch im Sichtfenster des PC1600 übersichtlich.
- Ich habe es mir angewöhnt, jede neue Zeile durch Überschreiben der alten Zeile zu editieren. Dann weiß man immer, an welcher Stelle die Spalte, die benötigt wird, jeweils ist.
- Kommentare habe ich noch einmal durch Semikolon (;) gekennzeichnet. Das ist ein persönlicher Luxus,

der aber nicht unbedingt notwendig ist. Zusätzlich schreibe ich den gesamten Kommentar klein (Ausnahme sind nur Variablen und Abkürzungen aus der jeweiligen Programmiersprache). Das hat für mich die Lesbarkeit des Listings erhöht.

- Gestartet wird das Programm im RUN-Modus. Dabei legt es über POKE-Befehle den Maschinen-Code im Speicher ab. Anschließend übernimmt ein Basic-Steuerprogramm den Start des Maschinen-Programms.
- Nach der Rückkehr aus dem Maschinenprogramm kann das Basic-Steuerprogramm weitere Bearbeitungen, Auswertungen oder den Start eines neuen Maschinenprogramms übernehmen.

Voraussetzung für einen solchen Assembler ist natürlich ein Buch über den Z80. Viel Freude beim Entdecken weiterer Möglichkeiten wünsche ich allen Lesern!

Und hier das Listing:

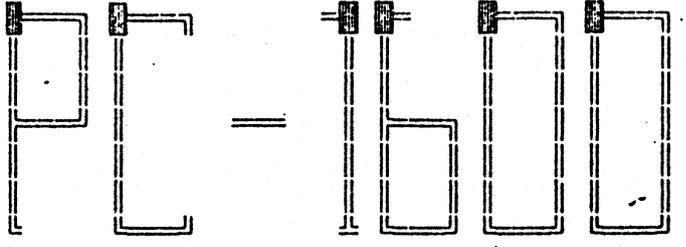
```

10 ;
11 ;
12 ;
13 S-STATUS 2-&8001:
14 P-S:
15 POKE P,&7B,&02:P-P-2: LD A,2
16 POKE P,&16,&00:P-P-2: LD D,0
17 POKE P,&5F:P-P-1: LD E,A
18 POKE P,&37:P-P-1: SCF
19 POKE P,&C9:P-P-1: RET
20 ;
21 ;
22 ;
23 ;
24 INPUT "variablen-wert:":V
25 CLS :WAIT 0
26 PRINT "Überg.-wert:&".HEXS V
27 CALL S,V
28 BEEP 10
29 PRINT "akku-wert :&".HEXS V
30 END
31 ;

```

;programmende

AUTOMATISCHE RESERVE-TASTEN-BELEGUNG



```

Der nebenstehende Shriftzug wurde mit
1000: "A" UP.RESERVE-TASTENBELEGUNG
1010: ' (VERWENDETE VAR.: I,B,E,S,P)
1020: B=2: ' BANK-NR.(siehe Tabelle?)
1030: S=&8008: 'START-ADR.(siehe Tabelle?)
1040: E=&80C4: ' END-ADR.(siehe Tabelle?)
1050: P=0: ' POKE-WERT
1060: ' INITIALISIERUNG
1070: FOR I=0TO 188: POKE #B, I+S, 0: NEXT I
1080: ' BELEGUNG DER INFO-ZEILE (3*26; .=0)
1090: DATA ..., &84, ..., &8E, ..., &94, ..., &99, ..., &81, ..., &9A, ...
1100: DATA ..., &C9, ..., &BB, ..., &BC, ..., &CB, ..., &CD, ..., &BA, ...
1110: DATA ..., &B9, ..., &CA, ..., &CB, ..., &CC, ..., &CE, ..., &7F, ...
1120: ' DATEN ZUR TASTATURBELEGUNG (MAX.110)
1130: DATA &01, &84, &02, &8E, &03, &94, &04, &99, &05, &81, &06, &9A sind, so können
1140: DATA &11, &C9, &12, &BB, &13, &BC, &14, &CB, &15, &CD, &16, &BA wir diese gezielt
1150: DATA &09, &B9, &0A, &CA, &0B, &CB, &0C, &CC, &0D, &CE, &0E, &7F mittels POKE ver-
1160: DATA -1 ändern und mit
1170: ' AUSFÜHRUNG den gewünschten Zeichen belegen.
1180: RESTORE Zum Programm:
1190: FOR I=0TO 77: READ P: IF P=0LET P=32 Dieses Programm kann als oder mit
1200: POKE #B, I+S, P: NEXT I einem AUTOSTART-Programm geladen werden.
1210: I=78 In Zeile 1020...1040 sind die der je-
1220: READ P: IF P=-1CLS :END weiligen Speicherkapazität betreffen-
1230: POKE #2, I+S, P: I=I+1: GOTO 1220 den Werte einzugeben. Diese können
der Bedienungsanleitung S.0/Anhang L
entnommen werden.
    
```

Hilfe der Reserve-Tasten erstellt. Normalerweise ist es jedoch nur möglich, diese mit den mittels der Tasten darzustellenden Zeichen zu belegen. Wissen wir aber wo und wie die Codes der Reserve-Tasten abgespeichert werden. Diesem Programm kann als oder mit einem AUTOSTART-Programm geladen werden. In Zeile 1020...1040 sind die der jeweiligen Speicherkapazität betreffenden Werte einzugeben. Diese können der Bedienungsanleitung S.0/Anhang L entnommen werden.

Der Reserve-Speicher liegt nun in den Adressen S...E auf Bank B und hat stets eine Größe von 188 Bytes. Diese sind wie folgt aufgeteilt: Die Info-Zeile der Reserveebene I wird in den 1. 26 Bytes abgespeichert, die der R.ebene II in den 2. 26 Bytes und die der R.ebene III in den 3. 26 Bytes. Leerzeichen werden dabei mit dem Code &20 (dez.32) abgespeichert. Somit sind schon 78 Bytes verbraucht und es verbleiben nur noch max.110 Bytes für die eigentliche Res.tasten-Belegung. Diese sind dann wie folgt abgespeichert:

Zuerst ein Code für die entsprechende Taste und danach die/der Code(s) des/der Zeichen. Die Tasten-Codes sind der nebenstehenden Tabelle zu entnehmen. Mit dem obenstehenden Programm wurden die Tasten wie folgt programmiert:

Taste	F1	F2	F3	F4	F5	F6
R.Ebene I	01	02	03	04	05	06
II	11	12	13	14	15	16
III	09	0A	0B	0C	0D	0E

(Angaben hexadez.!)
(Info-Zeile mit Taste RCL)

Res.Ebene I	Res.Ebene II	Res.Ebene III
ä Æ ö Ü ü U	ƒ η ∫ ℓ =	≡ √ ∫ ∫ █

Leider ist es mir allerdings nur gelungen Zeichen mit den Codes unter &E0 dort einzugeben. Dabei

wären doch gerade die Zeichen mit den griech.Buchstaben, d Grad-Zeichen etc. und vor allen das 'ß' (griech. Beta) sehr interessant für diesen Zweck. *P.L.*

-----?--Wer kennt hier eine Möglichkeit, die Ursache oder gar Abhilfe ?-?-?-?

EDITIEREN VON WERTEN

```

10 'PC1600: Editieren von Werten, d.h. Eingabe u./o. Korrektur
20 'an einem Beispiel einer 6X6 Matrix mit Ablaufsteuerung
30 'U up, T top, L left, R right, E end(des Edit). Blaettern endlos.
40 'Angeregt von: PC1600 Systemhandbuch Seite 46
50 '
60 CLS :PRINT T$;" ";W$=STR$ W:KBUFF$ =W$+LEFT$ (W1$,LEN W$)
70 INPUT "" ;W$:A$=LEFT$ (W$,1):IF INSTR ("RLEUT",A$)RETURN :ELSE LET W=VAL W$
  ,A$=""
80 RETURN
90 "Z"CLEAR :W1$=CHR$ 8+CHR$ 8,W1$=W1$+W1$+W1$+W1$,W1$=W1$+W1$:DIM X(5,5).
100 I=I+(I<0)*6-(I>5)*6,J=J+(J<0)*6-(J>5)*6
110 T$="X("+STR$ I+", "+STR$ J+)"= ",W=X(I,J):GOSUB 60:IF A$ELSE LET X(I,J)=W,I=
  I+1:GOTO 100
120 IF A$="E"THEN 130:ELSE LET I=I*(A$<>"T")-(A$="U"),J=J+(A$="R")-(A$="L"):GO
  TO 100
130 STOP
140 '
150 'CHR$ 8 ist Cursor left, davon muessen max.16 verfuegbar sein.
160 'Zeile 130 wuerde die Fortsetzung sein. Hier nur angedeutet.
170 'Die Eingabe 60-80 ist allgemeiner verwendbar, 90-130 ist ein Treiber dazu

```

PC 1600 TOWER 1600

```

10:CLS :CLEAR :WAIT 0:USING :TEXT :LF 15:
  GRAPH :CD=1000:XY=-1000:MM=0
20:GLCURSOR (150,0):SORGN
30:PRINT "*****"
40:PRINT "**** "KREIS PLOT" ****"
50:PRINT "*****"
60:PRINT "*****"
70:FOR T=1TO 500:NEXT T
80:INPUT "- Datenzahl ==>";N:IF N>400R N<1
  THEN CLS :GOTO 20
90:DIM D(N)
100:FOR I=1TO N
110:CLS :PRINT I;". ZAHL:";CURSOR 8:INPUT D(
  I):D(I)=ABS (D(I))
120:IF D(I)>>XYTHEN LET XY=D(I)
130:IF D(I)<<CDTHEN LET CD=D(I)
140:LET MM=MM+D(I)
150:NEXT I
160:CLS :INPUT "Nochmalige Eingabe?";RR$:IF
  RR$="J"THEN 10
170:CLS :COLOR 3
180:FOR EE=1TO N
190:D(EE)=(D(EE)*360)/MM
200:NEXT EE
210:GG=0:CSIZE 1
220:FOR X=1TO NSTEP 1
230:GG=GG+D(X)
240:LLINE (200,-100)-(200+(120*(1.3/1)*COS
  (GG)), -100+(120*(1/1.3)*SIN (GG)))
250:COLOR 1:LPRINT INT (((D(X)*MM)/360)*10
  00+.5)/1000:COLOR 3
260:GLCURSOR (200+(120*(1.3/1)*COS (GG)), -1
  00+(120*(1/1.3)*SIN (GG)))
270:IF GG>180THEN LLINE -(200+(120*(1.3/1)*
  COS (GG)), -140+(120*(1/1.3)*SIN (GG)))
280:NEXT X
290:FOR SS=0TO 360STEP 5
300:LLINE -(200+(120*(1.3/1)*COS (SS)), -100
  +(120*(1/1.3)*SIN (SS)))
310:IF SS=180OR SS=360THEN LLINE -(200+(120
  *(1.3/1)*COS (SS)), -140+(120*(1/1.3)*
  SIN (SS)))
320:NEXT SS
330:FOR SS=360TO 180STEP -5
340:LLINE -(200+(120*(1.3/1)*COS (SS)), -140
  +(120*(1/1.3)*SIN (SS)))
350:NEXT SS
360:TEXT :LF 5:COLOR 0:LPRINT "Höchster Wer
  t:";XY
370:LPRINT "Niedrigster Wert:";CD
380:LPRINT "Durchschnitt:";MM/N
390:LF 3:BEEP 3:PRINT "CIAO!":END
400:'
410:'(C) by MARKUS GOESE
420:'

```

Hansjörg Schlösser
Burgunder Str. 31
5 Köln-1

35 11 PLOTTER-PROGRAMME FÜR DEN IBM 360/50 + CEE-1600F

Eine Funktion $Z=f(X,Y)$ wird mit verdeckten Kanten zweifarbig gplottet. Erreichbare Auflösung 256 * 256 Linien. Speicherbedarf bei max. Auflösung ca. 19600 Bytes, d.h. mit Tricks (siehe 1600 Systemhandbuch) reicht eine 8K Erweiterung, 16K Erweiterung reichen immer. Aber: bezahlt werden muß dieser "geizige" Algorithmus mit Laufzeiten bis 36 std.!! Die Wirkung ist vorab kaum zu schätzen, deshalb immer erst einen Versuch mit 20-30 Linien durchführen.

Das Programm startet mit DEF Z.

F(Z)=? ist mit dem Term für die Funktion zu beantworten. Z.B. $Z=\text{SINSQR}(X*X+Y*Y)$. Nur ENTER erhält den vorher gültigen Term. Mehrere Anweisungen sind erlaubt. Sie müssen in 50 Stellen darstellbar sein und "irgendwie muß Z als Ergebnis von X und Y rauskommen". Die Zeile des Pgm. wird mit LL. zu Papier gebracht. Ändert sich die Zeilennr. mit dem Term (REN. z.B.) muß in Zeile 98 der Ausdruck "8"+A\$(0) angepaßt werden.

XU, XO, YU und YO sind die Unter- Obergrenzen für X und Y. Diese und folgende Fragen dürfen nicht mit ENTER alleine beantwortet werden. Es müssen Angaben gemacht werden. Aus Platzgründen gibt es keinerlei Prüfungen! Eingaben werden protokolliert und auf dem Papier geprüft.

Auflös.=? fragt nach der Anzahl der Linien. Die Laufzeit hängt etwa vom Quadrat der Auflösung ab. Unzureichender Speicher führt vor Verlust an Rechenzeit zu einem ERROR. Man starte erneut mit DEF Z.

Breite=? Anzugeben sind Plottereinheiten. Da die Grafik im Querformat entsteht, maximal 1354. 1320 sollten aber nicht überschritten werden.

H.Fakt.=? Anzugeben ist, das wievielfache der Breite die Höhe einer Linie betragen soll. Dieser Höhe ist später die perspektivische Tiefenstaffelung der Linien zuzurechnen. Sie beträgt das 0.5 fache der Breite. Es stehen außerdem in dieser Richtung nur 960 Einheiten zur Verfügung. Bei 1320 in der Breite ist max. ein Faktor von 0.2 möglich.

Persp.Fakt.? Üblicherweise geht eine perspektivische Darstellung von 0.5 facher Breite als Höhe aus. Durch Angabe eines Faktors kann darauf Einfluß genommen werden. Faktoren kleiner 1 stellen in einem größeren Betrachtungswinkel dar, Faktoren größer 1 in einem kleineren Winkel. Zwischen etwa 0.7 und 1.4 kann man experimentieren. Die Perspektive beeinflusst die Verdeckung und kann trotzdem später ohne Verlust der Rechenzeit geändert werden.

Die jetzt beginnende Rechenzeit ist beachtlich. Sie wird im Display von einer Anzeige begleitet: C1 = Extrema, K1 und K2 = Kantenlage. Dazu wird der jeweils bearbeitete Index angezeigt.

PAPERR OK(J/N)? wird mit J beantwortet, nachdem der Plotter fertig ist. Die Trommel wird dazu mit frischen Stiften doppelt bestückt, z.B. schwarz, rot, schwarz, rot (sonst muß man Zeilen 48,62 von COL. (V+1) auf COL. V ändern.). So ist aber eine gleichmäßigere Zeichnung möglich. Stiftwechsel zwischendurch sind nicht so einfach. Das Papier wird knapp am oberen Rand eingelegt. Bessere Wiederholgenauigkeit ergibt sich nach mehrfachem Verfahren des Papiers über die ganze Länge, ohne es dabei rausfallen zu lassen versteht sich. Selbstverständlich muß am Netz gplottet werden und das Papier darf nicht behindert werden. Ist beim Übergang zu einer neuen Linie eine Taste gedrückt, wird (mit entspr. Anzeige) unterbrochen. "C" setzt dann fort.

DEF X beginnt den Plottvorgang erneut. Dabei kann durch Verändern der Stand.Var. H die Zeichnung verschoben werden. Z.B. $H=H-100$, DEF X verschiebt nach rechts um $100/4 (=25)$ Einheiten. Falls abgeschnitten werden soll, kann man damit bestimmen, wo und wieviel. Man achte während des Plottens auf die Lage der ersten Linie. Sie soll unbeschnitten knapp am linken Rand liegen. Gegf. breche man ab, ändere H und starte neu mit DEF X. Vorsicht: bei DEF Z ist die Rechenzeit verloren.

DEF C erlaubt die Wiederholung der Zeichnung mit einem geänderten Persp.Faktor. Man kann geringfügig den Faktor vergrößern. Verdeckungen werden mit einem weißen Rand abgesetzt. Das ist eine Geschmacksfrage.

Das zweite Programm ist ein "feelalike, workalike", d.h. es "tut dasselbe" und braucht auch die gleichen Angaben. Es ist die "beschränkte, gefräßige, schnelle Schwester". Beschränkt weil nur etwa 90 Linien Auflösung erreichbar sind. Gefräßig, weil als Datenspeicher für 90 Linien 72900 Bytes erforderlich sind. Aber die 90 Linien sind 5 mal schneller "fertig". Ein ganz klassisches Beispiel, wie über den Algorithmus Platz und Zeit gegeneinander getauscht werden. Dieser hier brauchte für 256 Linien 576K. Eine Abweichung gibt es doch: das zweite Pgm. verwendet COL. 1 und 2 (was immer dort in der Trommel steckt). Doppelbestückung ist hier überflüssig, Auswechseln der Stifte einfacher als Programmänderung. Das erste Pgm. läßt sich leicht für PC1500/CE158/CE516 anpassen, das zweite kaum. Die Auflösung wird so klein, daß damit nur ein CE150 "zufrieden" ist.

Anm.: genau wie der CE516P bringt der CE1600P bessere Wiederholgenauigkeit bei dünnerem Papier (40g) mit etwas glatterer Oberfläche. Kann man über Druckereibetriebe beziehen. Abweichend zum 516 "laufen" satinierte Papiere schlecht. Dort liegt das Papier unter dem Stift hohl, hier auf einer Walze. Der Stift kommt zum Schwingen (hörbar!) und die Linie wird "schmuddelig". Beim ersten Pgm. zeigt sich eine unbefriedigende Genauigkeit evtl. erst nach 12 std. Plottzeit! Der 1600P führt bei sehr vielen Verfahrbewegungen in einer Zeichnung zu einer unvermeidlichen Streckung des Papiers. Es wird dann zu häufig um eine dünne Walze gerollt. Die Materialveränderung setzt der vorzüglichen Genauigkeit des Plotters Grenzen. In den beiden Pgm. steckt eine Korrektur dafür. Sie basiert auf Erfahrung. Noch besser ist aber die Vermeidung von mehr als ca. 50 Längsbewegungen über die selbe Stelle. Läßt man mit dem ersten Pgm. 1600 und 1500 gegeneinander "antreten", beginnt der 1600 mehrere std. eher mit dem Plotten. Das ist zu erwarten. Zum Schluß ist von seinem Vorsprung nicht mehr viel übrig! Der 516P ist dabei nicht so viel schneller, nur hat er einen eigenen Prozessor. Während er "malt" ist der Rechner nicht angehalten, sondern rechnet schon den nächsten Wert. Bei 65000 solcher Vorgänge summiert sich das.

PROGRAMM 1

```

2 X=B*F+XU,Y=C*P+YU:GOTO 8
4 X=B*O+XU,Y=C*F+YU:GOTO 8
6 X=B*O+XU,Y=C*P+YU
8 Z=SQR (X*X+Y*Y),Z=Z/(Z+180)
10 RETURN
12 X=INT (F/4),Y=F-X*4:GOTO 16
14 X=INT (P/4),Y=P-X*4
16 K2=1:IF Y=0RETURN :ELSE IF Y=1LET K2=4:RETURN :ELSE IF Y=2LET K2=16:RETURN :E
LSE IF Y=3LET K2=64:RETURN
18 K2=256:RETURN
20 R=INT (J*(O-P)+.5):GOSUB 6:S=INT (K*(O+P)+(Z-E)/Q+.5),K1=0:GOSUB 14:I=K2
22 L=I*3,V=INT ((ASC FS(O,X)AND L)/I+.5):RETURN
24 IF W=3LET K1=9:ELSE LET K1=0
26 LLINE -(INT ((S+U)/2+.5),INT ((R+T)/2+.5)),K1:W=V:RETURN
28 LLINE -(S,R),K1:T=R,U=S,W=V:RETURN
30 K2=3:R=1E99,S=-R,B=(XO-XU)/G,C=(YO-YU)/G:FOR O=0TO G:PRINT "C1 ";O+1:FOR P=0T
O G:GOSUB 6
32 IF Z<RLET R=Z
34 IF Z>SLET S=Z
36 NEXT P:NEXT O:ZU=R,E=R,ZO=S:Q=(S-E)/D:GOTO 66
38 "V"FOR O=0TO G:P=0:GOSUB 104:PRINT O+1:GOSUB 6:T=INT (J*O+.5),U=INT (K*O+(Z-E
)/Q+.5)
40 W=ASC FS(O,0)AND 3:LLINE -(U,T),9:N=9:FOR P=1TO G:GOSUB 20
42 IF V=3IF W=3LET K1=9:GOSUB 28:NEXT P:NEXT O:ES="":GOTO 52
44 IF V<>3IF W<>3LET K1=0:GOTO 48
46 GOSUB 24:GOTO 42
48 IF N<>VIF V>0COLOR (V-1):N=V
50 GOSUB 28:NEXT P:NEXT O:ES="":GOTO 52
52 FOR P=0TO G:O=0:GOSUB 104:PRINT P+1:T=INT (J*-P+.5):GOSUB 6:U=INT (K*P+(Z-E)/
Q+.5):GOSUB 14
54 L=K2,W=INT ((ASC FS(O,X)AND L)/(L/3)+.5):LLINE -(U,T),9:FOR O=1TO G:GOSUB 20
56 IF V=3IF W=3LET K1=9:GOSUB 28:NEXT O:NEXT P:END
58 IF V<>3IF W<>3LET K1=0:GOTO 62
60 GOSUB 24:GOTO 56
62 IF N<>VIF V<0COLOR (V+1):N=V
64 GOSUB 28:NEXT O:NEXT P:END
66 FOR O=0TO G-1:R=1E99,S=-R:PRINT "K1 ";O+1:FOR P=0TO G-O:F=O+P:GOSUB 2
68 Z=M*(F+P)+(Z-E)/Q,V=0:IF Z<=RLET R=Z,V=2,H=Z
70 IF Z>=SLET S=Z,V=1:GOTO 74
72 IF Z<=SIF Z>=RLET V=3
74 GOSUB 14:N=K2,L=ASC FS(F,X)AND (255-N*3),L=LOR (N*V):FS(F,X)=CHR$ L
76 NEXT P:NEXT O:FOR P=1TO G-1:R=1E99,S=-R:PRINT "K2 ";P:FOR O=0TO G-P:F=O+P:GOS
UB 4

```

```

73 Z=M*(O+F)+(Z-E)/Q,V=0:IF Z<=RLET R=Z,V=2
80 IF Z>=SLET S=Z,V=1:GOTO 84
82 IF Z<SIF Z>RLET V=3
84 GOSUB 12:N=K2,L=ASC FS(O,X)AND (255-N*3),L=LOR (N*V):FS(O,X)=CHRS L:NEXT O:NE
XT P
86 "X"ES="1":CLS :PRINT "PAPER OK(J/N)? ":IF INKEYS <>"J"BEEP 1:GOTO 86
88 GRAPH :PAPER C,2047,1354:LLINE -(20-INT (H/4),-677),9:SORGN :GOTO 38
90 INPUT "Breite? ";W:W=INT (W/4)*4,J=W/2:INPUT "H.Fakt.? ";D:D=INT (W*D):INPUT
"Persp.Fakt.? ";P
92 K=INT (P*J/2),M=K,J=J/G,K=K/G,M=M/A:LPRINT "W";W;" H";D;" P";P:ZO=INT ((A+3)/
4),PP=P:RETURN
94 "Z"CLEAR :DEGREE :USING :PP=0:DIM AS(0)*80:ZU=0,ZO=0,K1=9
96 INPUT "F(Z)=? ";AS(0):KBUFFS =CHRS 31+"S"+AS(0)+CHRS 13+CHRS 31+"GOTO"+CHRS 3
4+"M"+CHRS 34+CHRS 13:END
98 "M"TEXT :COLOR 0:LLIST 8:INPUT "XU=? ";XU,"XO=? ";XO,"YU=? ";YU,"YO=? ";YO,"A
uf1.=? ";A:G=A-1
100 LPRINT "XU=";XU;" XO=";XO;" YU=";YU;" YO=";YO;" Auf1.=";A
102 GOSUB 90:DIM FS(G,ZO)*1:GOTO 30
104 IF ESIF OIF (OMOD 60)=0LLINE -(0,-1),9:SORGN
106 IF INKEYS ELSE RETURN
108 IF INKEYS BEEP 1:GOTO 108:ELSE LLINE -(900,0),9:CLS :PRINT "C to continue"
110 IF INKEYS <>"C"GOTO 110:ELSE CLS :RETURN
112 "C"CLS :PRINT "P.FAKT";PP:INPUT "Persp.Fakt.? ";P:K=K/PP*P,PP=P:GOTO "X"

```

PROGRAMM 2

```

10 R=INT (J*(O-P)+.5),S=INT (K*(O+P)+F(O,P)/Q+.5),K1=0,V=VAL FS(O,P):RETURN
20 K1=(W-3)*9:LLINE -(INT ((S+U)/2+.5),INT ((R+T)/2+.5)),K1:W=V:RETURN
30 LLINE -(S,R),K1:T=R,U=S,W=V:RETURN
40 R=1E99,S=-R,B=(XO-XU)/G,C=(YO-YU)/G:FOR O=0TO G:PRINT "C1 ";O+1:FOR P=0TO G
50 X=B*O+XU,Y=C*P+YU
60 Z=SQR (X*X+Y*Y),Z=COS Z/(Z+180)
70 IF Z<RLET R=Z
80 IF Z>SLET S=Z
90 F(O,P)=Z:NEXT P:NEXT O:ZU=R,E=R,ZO=S:Q=(S-E)/D:FOR O=0TO G:FOR P=0TO G:F(O,P
)=F(O,P)-R
100 NEXT P:NEXT O:GOTO 250
110 "V"FOR O=0TO G:P=0:GOSUB 430:PRINT O+1:T=INT (J*O+.5),U=INT (K*O+F(O,P)/Q+.5
)
120 W=VAL FS(O,P):LLINE -(U,T),9:N=9:FOR P=1TO G:GOSUB 10
130 IF V=3IF W=3LET K1=9:GOSUB 30:NEXT P:NEXT O:ES="":GOTO 180
140 IF V<>3IF W<>3LET K1=0:GOTO 160
150 GOSUB 20:GOTO 133
160 IF N<>VIF VCOLOR V:N=V
170 GOSUB 30:NEXT P:NEXT O:ES=" "
180 FOR P=0TO G:O=0:GOSUB 430:PRINT P+1:T=INT (J*-P+.5),U=INT (K*P+F(P,O)/Q+.5):
W=VAL FS(P,O)
190 LLINE -(U,T),9:FOR O=1TO G:GOSUB 10
200 IF V=3IF W=3LET K1=9:GOSUB 30:NEXT O:NEXT P:END
210 IF V<>3IF W<>3LET K1=0:GOTO 230
220 GOSUB 20:GOTO 200
230 IF N<>VIF VCOLOR V:N=V
240 GOSUB 30:NEXT O:NEXT P:END
250 FOR O=0TO G-1:R=1E99,S=-R:PRINT "K1 ";O+1:FOR P=0TO G-O:F=O+P
260 Z=M*(F+P)+F(F,P)/Q:IF Z<=RLET R=Z,H=Z,FS(F,P)="2"
270 IF Z>=SLET S=Z,FS(F,P)="1":GOTO 290
280 IF Z<SIF Z>RLET FS(F,P)="3"
290 NEXT P:NEXT O:FOR P=1TO G-1:R=1E99,S=-R:PRINT "K2 ";P:FOR O=0TO G-P:F=O+P
300 Z=M*(O+F)+F(O,F)/Q:IF Z<=RLET R=Z,FS(O,F)="2"
310 IF Z>=SLET S=Z,FS(O,F)="1":GOTO 330
320 IF Z<SIF Z>RLET FS(O,F)="3"
330 NEXT O:NEXT P
340 "X"ES="1":CLS :PRINT "PAPER OK(J/N)? ":IF INKEYS <>"J"BEEP 1:GOTO 340
350 GRAPH :PAPER C,2047,1354:LLINE -(20-INT (H/4),-677),9:SORGN :GOTO 110
360 INPUT "Breite? ";W:W=INT (W/4)*4,J=W/2:INPUT "H.Fakt.? ";D:D=INT (W*D):INPUT
"Persp.Fakt.? ";P
370 K=INT (P*J/2),M=K,J=J/G,K=K/G,M=M/A:LPRINT "W";W;" H";D;" P";P:PP=P:RETURN
380 "Z"CLEAR :DEGREE :USING :PP=0:DIM AS(0)*80:ZU=0,ZO=0,K1=9
390 INPUT "F(Z)=? ";AS(0):KBUFFS =CHRS 31+"60"+AS(0)+CHRS 13+CHRS 31+"GOTO"+CHRS
34+"M"+CHRS 34+CHRS 13:END
400 "M"TEXT :COLOR 0:LLIST 60:INPUT "XU=? ";XU,"XO=? ";XO,"YU=? ";YU,"YO=? ";YO,
"Auf1.=? ";A:G=A-1
410 LPRINT "XU=";XU;" XO=";XO;" YU=";YU;" YO=";YO;" Auf1.=";A
420 GOSUB 360:DIM FS(G,G)*1,F(G,G):GOTO 40
430 IF ESIF OIF (OMOD 60)=0LLINE -(0,-1),9:SORGN
440 IF INKEYS ELSE RETURN
450 IF INKEYS BEEP 1:GOTO 450:ELSE LLINE -(900,0),9:CLS :PRINT "C to continue"
460 IF INKEYS <>"C"GOTO 460:ELSE CLS :RETURN
470 "C"CLS :PRINT "P.FAKT";PP:INPUT "Persp.Fakt.? ";P:K=K/PP*P,PP=P:GOTO "X"

```

```

*****
*                               PC - 1 6 0 0                               *
*   E I N   A S S E M B L E R   O H N E   A S S E M B L E R ?   *
*****
(Fortsetzung)

```

Speicherplätze - ihre Reservierung und Verarbeitung:

- Mit Hilfe der Programmvariablen P wird die Adresse reserviert und in einer Variablen (A1, A2, A3 usw...) abgelegt. (s. Zeile 15, 17, 19 im Programm-Listing)
- Für die Verarbeitung beim Ablegen des Maschinenprogramms müssen die Adress-Variablen in ihr H-Byte und ihr L-Byte aufgeteilt werden. Dafür wähle ich die Buchstaben H bzw. L zusammen mit der angehängten Zahl der jeweiligen Adress-Variablen. H1 und L1 entsprechen also dem H-Byte/L-Byte der Adress-Variablen A1. (s. Zeile 16, 18, 20 im Programm-Listing) (1)
- Beim Ablegen der Adresse im eigentlichen 'Assembler'-Programm werden die Variablen in der Reihenfolge L1, H1 also erst L-Byte, dann H-Byte) über POKE-Befehl abgelegt. (s. Programm-Zeilen 26, 28, 30, 31, 33 im Programm-Listing)

Und hier das Listing:

```

10 '                               ;additionsprogramm
11 '                               ;-----
12 '
13 S=STATUS 2+&8001:'           ;speicherbeginn
14 P=S:'                         ;programmzähler
15 A1=P:P=P+1:'                 ;adr.1 (A1) reservieren
16 H1=INT (A1/256):L1=A1-H1*256:' ;adr.1 H- u. L-byte (1)
17 A2=P:P=P+1:'                 ;adr.2 (A2) reservieren
18 H2=INT (A2/256):L2=A2-H2*256:' ;adr.2 H- und L-byte
19 A3=P:P=P+1:'                 ;adr.3 (A3) reservieren
20 H3=INT (A3/256):L3=A3-H3*256:' ;adr.3 H- und L-byte
21 POKE A1,0,0,0:'             ;adresse 1-3 auf 0
22 '
23 '
24 START=P:'                     ;programmstartadresse
25 POKE P,&7B:P=P+1:'           LD A,E           ;übergabewert -> A
26 POKE P,&32,L1,H1:P=P+3:'     LD (A1),A       ;übergabewert -> adr.1
27 POKE P,&3E,&01:P=P+2:'       LD A,1          ;1 -> A
28 POKE P,&32,L2,H2:P=P+3:'     LD (A2),A       ;1 -> adresse2
29 '                               ;additionsprogramm (2)
30 POKE P,&3A,L1,H1:P=P+3:'     LD A,(A1)       ;operand1 von adresse1
31 POKE P,&21,L2,H2:P=P+3:'     LD HL,A2        ;adresse2 -> HL
32 POKE P,&86:P=P+1:'           ADD A,(HL)      ;operand1 + operand2
33 POKE P,&32,L3,H3:P=P+3:'     LD (A3),A       ;ergebnis -> adresse3
34 POKE P,&C9:'                 RET             ;ende des masch.-progr.
35 '
36 '                               ;basic-steuerprogramm
37 '                               ;-----
38 CLS :WAIT 0:USING "####"
39 PRINT "zellen vorher ";PEEK# (0,A1);PEEK# (0,A2);PEEK# (0,A3)
40 INPUT "übergabe: ";V
41 CALL START,V
42 PRINT "zellen nachher";PEEK# (0,A1);PEEK# (0,A2);PEEK# (0,A3)
43 USING :END

```

Literaturhinweise zum Programm:

- (1) vergl.: Andreas Donner: 'weltere Möglichkeiten von PEEK und POKE' in: PC-1500 Programmier- und Programmierhandbuch, Hrsg.: Fischer GmbH, Berlin 1984 S.26
- (2) vergl.: Rodney Zaks - 'Programmierung des Z80' SYDEX-Verlag GmbH, Düsseldorf, 10. Auflage 1987 S.90

Hanns Jörg Schlösser
Burgunder Str.31
5. Köln-1

PC-1600 Kreisroutine

```

1 'PC1600+CE1600P
2 'Das Up zeichnet einen Kreis um X,Y mit Radius R
3 'Abweichung +-1 Plottereinheit
4 '
5 T=9:FOR C=1TO 4:B=R-1,D=0,E=R:FOR A=0TO 1:IF B<0LET E=E-1,B=B+E:E:IF B<0IF E>0LET A=0:NEXT A
6 IF C=1LLINE -(X+D,Y+E),T:T=0:ELSE IF C=2LLINE -(X+E,Y-D),0:ELSE IF C=3LLINE -(X-D,Y-E),0
7 IF C=4LLINE -(X-E,Y+D),0
8 B=B-D-D-1,D=D+1,A=E<0:NEXT A:NEXT C:RETURN

```

```

1 'PC1600
2 'Das Up malt einen Kreis in's Display
3 'Mittelpunkt X,Y Radius R
4 '
5 B=R-1,C=0,D=R:FOR A=0TO 1:IF B<0LET D=D-1,B=B+D+D
6 PSET (X+C,Y+D):PSET (X-C,Y+D):PSET (X+C,Y-D):PSET (X-C,Y-D):PSET (X+D,Y+C)
7 PSET (X-D,Y+C):PSET (X+D,Y-C):PSET (X-D,Y-C):B=B-C-C-1,C=C+1,A=C>D:NEXT A:RETURN

```

Suche nach angesprungenen Zeilen für PC-1600

Kb: 1800

Ladexzeit: 2:15

Status 1:1732

Aufgabe:

Sucht aus einem Programm alle angesprungenen Zeilen heraus.

Ablauf:

Nach dem Starten des Programmes beginnt dieses mit der Suche nach Verzweigungen, sortiert diese und druckt eine Liste aus.

Bedienung:

Wird während der Suche nach Verzweigungen der Print-Schalter des CE-150 auf "P" gestellt, so druckt der Computer jede gefundene Verzweigung aus.

Ablaufbedingungen:

Das Programm, das nach Verzweigungen durchsucht werden soll, muß das erste Programm im Hauptspeicher sein. Das Suchprogramm wird dazugemergelt.

Benützte Hardware:

- PC-1600
- Drucker CE-150

Michael Smetana
Staarg. 16
A-1140 Wien

Verwendete Variablen:

- A... Anfangsadresse des Listings
E... Anfangsadresse der GOTO-Liste
DR*... Wurde gedruckt? (Ja="J"/Nein="")
TB... Ausgabezeitpunkt: Bildschirm
TD... Ausgabezeitpunkt: Drucker
I, J... Laufvariablen für FOR...NEXT Schleifen
B, F, G, H, L, M, N, O, P... Hilfsvariablen
GE, R, S, U, X, Y, Z... Hilfsvariablen

```
10: "GO" CLEAR:CLS:
LDIT 0:USING
20: A=XPEEK R7866:
S6=XPEEK R7866:
XPEEK R7866:
XPEEK R7866+1:F
R:B=A
30: X=XPEEK A:Y=XPEEK
EK (A+1):Z=XPEEK
(A+2):I=3:IF X=N
SS GOTO 140
40: P=XPEEK (A+1)
50: IF P=13 LET A=D
+N+3:GOTO 30
60: IF P=31 GOTO 80
70: I=I+1:GOTO 40
80: L=L+1:R=XPEEK (
A+L+1):S=XPEEK (
A+L+2):I=I+4:IX
XN56+Y:Z=XN56+Y
90: PRINT USING ;"##
:CHR#";RDB:";:C
SING ;"##:";:J
N:E=E+4:IF
XPEEK R800E+1 G
OTO 40
100: IF DR*="J" GOT
O 130
110: TEXT:COLOR 0:C
SIZE 2:LPRINT "
:U#NE:;9UC999:":
:DR#NE:;J:;9UC999:":
120: LPRINT "#####
#####:LP
RINT " N: N:
: " U#NE:;9UC999:":
#####:LPRINT
#####
130: LPRINT USING "
G##:;L:;":;USIN
G "##:;:;:;":
;N:GOTO 40
140: CLS:PRINT "###
#####:PRINT "
##### E#9:;DT :;9
150: PRINT "###:;US
ING ##:;L:;:C
)N:;9UC999:;:
*):;PRINT "###
#####
#####
160: G=E:HE:E=F:IF
DR*="J" LPRINT "
#####:LF 3
170: R=XPEEK (E+2):
S=XPEEK (E+3):L=
1:E=E+4:XPOKE G,
R,S:G=G+2
180: IF E=H GOTO 25
0
190: R=XPEEK (E+2):
S=XPEEK (E+3)
200: GE=0:FOR I=0 T
O L-1:IF XPEEK (
I+I*N)R GOTO 23
```

```
210: NEXT I:IF GE=1
LET E=E+4:GOTO 1
80
220: XPOKE G,R,S:L=
L+1:G=G+2:E=E+4:
GOTO 180
230: IF XPEEK (H+I*
2+1)<>S GOTO 210
240: GE=1:I=L-1:GOT
O 210
250: CLS:PRINT "###
#####:
*****:
***** E#9:;DT :;9
260: PRINT "###:;US
ING ##:;L:;:N#
:;:;:;:;:;:;:;
*****:
*****:
*****
270: G=H:FOR I=0 TO
L-1:Y=XPEEK (G+I
*N):Y=XPEEK (G+I
*N+1):Z=XN56+Y:
FOR J=1+1 TO L-1
280: R=XPEEK (G+J*N
):S=XPEEK (G+J*N
+1):P=R*XN56+S:IF
P<O GOTO 300
290: NEXT J:NEXT I:
GOTO 310
300: XPOKE (G+J*N),
X,Y:XPOKE (G+1*N
),R,S:XR=Y:S=O=
P:GOTO 290
310: E=F:G=H:TEXT:C
OLOR 0:CSIZE 2:L
PRINT " U#NE:;
9UC999:":DR#
NE:;J:;9UC999:":
#####:L
#####:LP
RINT " N: N:
: " U#NE:;9UC999:":
#####:LPRINT
#####
330: CLS:USING "##
:;FOR I=0 TO
L-1:X=XPEEK (G+I
*N):Y=XPEEK (G+I
*N+1):Z=XN56+Y:
PRINT Z;:;:
340: LPRINT Z;:TB=1
:TD=1
350: IF E=G GOTO 42
0
360: IF X<>XPEEK (E
+2)OR Y<>XPEEK (
E+3)GOTO 410
370: R=XPEEK E:S=XP
EEK (E+1):C=R*XN
56+S
380: TB=TB+1:IF TB=
5 LET TB=1:PRINT
:
390: TD=TD+1:IF TD=
4 LET TD=1:LPRIN
T "
400: PRINT U;:LPRIN
```

```
T C;
410: E=E+4:GOTO 350
420: E=F:TEXT:LF 1:
PRINT:NEXT I:LPR
INT "#####
#####:LF 3:CL
S:E=0
```

STATUS 1 = 1732

U#	NE	9UC999:
1:	30:	140
2:	50:	30
3:	60:	80
4:	70:	40
5:	90:	40
6:	100:	130
7:	130:	40
8:	180:	250
9:	200:	230
10:	210:	180
11:	220:	180
12:	230:	210
13:	240:	210
14:	280:	300
15:	290:	310
16:	300:	290
17:	350:	420
18:	360:	410
19:	410:	350

Verzweigungen:

Zeile:angespr. von:		
30	50	
40	70	90
	130	
80	60	
130	100	
140	30	
180	210	220
210	230	240
230	200	
250	180	
290	300	
300	280	
310	290	
350	410	
410	360	
420	350	

Das Pgm belegt ca. 7700 Bytes, zu denen 1200 Bytes Variablen und 1000 Bytes Eingabepuffer kommen. Um komfortabel arbeiten zu können, sollte eine Ramdisk und Floppy vorhanden sein. Zur Not geht's auch ohne. Start: DEF Z. Dann wird ein zweizeiliges Menü angezeigt. Cursor up/down wechselt die Zeilen und die F-Tasten unter den Texten lösen die Funktionen aus.

BOX RdP EDT SET TER LST
BOX EDT WtP PRT TER END

BOX vereinbart einen Namen, um eine komplette Einstellung aller Parameter auf Floppy schreiben oder von dort lesen zu können. Das File heißt X:nnnn.BOX (nnn = Boxname). Eine solche Datei mit der Normaleinstellung erlaubt einfache Grundstellung nach DFÜ. Jede erfolgreiche Einstellung kann reproduziert werden.

RdP Read Parameter liest ein X:nnn.BOX-File. Bezugnahmen auf nicht mehr vorhandene Files (up/download) werden nicht übernommen.

Wordlen /7/8/	= 7	BOX=GENOA	
Stopp-B /1/2/	= 2	Download=	S2:GEN1.TXT
Parity /O/E/N/	= N	Append /J/N/	= N
B /300/1200/2400/=	300	Text-1=	DMARYBEL
Sh.IN-OUT /S/N/	= S	Text-2=	M01F36K58
XON-XOFF /X/N/	= X	Text-3=	A4511234802
SNDSTAT (0-63)	= 63	Text-4=	RQ00587
Timeout (5-99)	= 5	Text-5=	USID:22A57YV
Sendline (20-80)	= 80	Upload-1=	X:KOPF.GEN
EOL /0/1/2/	= 2	Upload-2=	S2:TXT.GEN
Buff.(1000-6000)	= 4096	Upload-3=	
Echo /J/N/	= N	Delay (0-500)	= 20

EDT legt die Parameter zur Änderung vor. Cursor up/down blättern, T zeigt die erste, B die letzte Zeile, Q beendet EDT. Wo genau definierte Werte zugelassen sind (im Text durch // getrennt), kann mit der 4-Taste gewechselt werden. Wahlfreie Angaben (Texte, Filenamen) können mit CL gelöscht werden. Sonst wird mit A der Wert zur Eingabe/Änderung vorgelegt. Sie wird mit ENTER abgeschlossen. Besonderheiten: SNDSTAT (Anleitung zum 1600 und Handbuch zum Koppler!): 63 wartet auf keine Pegel, sendet ggf. auch "in die Luft". Sonst tritt ein TIMEOUT-Error auf, wenn zu lange gewartet werden muß. Behandlung: (wenn nicht upload) Message, Sendezichen ignorieren und zurück zum Empfang, (bei upload) Message, Abbruch Terminal vorschlagen. Wenn nicht Abbruch, Zeile erneut versuchen. Zeitangabe unter Timeout nicht zu groß wählen. Während der Wartezeit kann der Buffer vollaufen. Mit Sendline und EOL kann der Box ein Wunsch nach max. Zeilenlänge und Endzeichen erfüllt werden. Beim Empfang muß ein CR am EOL "beteiligt" sein. Zusätzliche LF/CR haben keine Wirkung. Buffer (mind.1000) bestimmt die Größe des Empfangspuffers der Schnittstelle. Geben Sie soviel Sie können. Echo muß auf J gesetzt werden, wenn die Box die Zeichen zurücksendet. Kann man das in der Box (mit CTRL ..) abschalten, sollte das geschehen. Download: Empfangsfile zur Speicherung der Daten. Wenn irgend möglich Ramdisk eintragen. Die Datei kann später gedruckt werden. Append: wenn download-file schon besteht (auch zweiter Anlauf nach Abbruch!) setzt das Pgm J ein. Wenn J, wird "angehängt" (sonst überschrieben). Text-1/Text-5 sind wahlfreie Angaben (max. 16 Stellen), die auf Knopfdruck (F1-F5) gesendet werden. Sie sind für NUI/NUA/PSWD gedacht. Vorsicht: bis zum Menüpunkt END sind solche Angaben im Speicher ggf. auch auf der Floppy! Geraten sie in falsche Hände, kann das ein Vermögen kosten! Nach Sendung des n-ten Textes werden die Texte mit kleinerer Nummer nicht mehr gesendet. Die NUI wird so nicht der Box mitgeteilt, ist beim nächsten Versuch aber wieder da. Mehr Schutz ist nicht eingebaut.

Upload-1/-3 sind wahlfreie files, die der Box per Knopfdruck komplett gesendet werden. Während dieser Zeit werden Daten von der Box gepuffert aber nicht bearbeitet. Arbeitet die Box zwingend mit Echo, wird während des upload regelmäßig der Puffer gelöscht. Delay= hat nur Bedeutung bei upload. Hier wird die Wartezeit (Millisek.) angegeben, die nach jeder Zeile vergehen soll.

SET stellt die Schnittstelle auf die Parameter ein. Bei Beginn des Terminalmode geschieht das auch automatisch. Im Falle einer Rücksetzung auf Standard will man TER aber gerade nicht.

LST listet die Parameterdaten über CE1600P.

WtP Write Parameter speichert die Parameter auf Floppy. Tunlichst erst nach erfolgreichem Betrieb. Enthält die Floppy NUI/PSWD, gehört sie unter Verschuß.

PRT Druckt das downloadfile wahlweise über CE1600P (nicht empfehlenswert) oder seriell. Wer das lieber parallel hätte, muß geringfügig ändern. Nach dem Druck kann die Datei auf Wunsch gelöscht werden.

END löscht den Variablenspeicher und schaltet den Rechner ab. Erst jetzt sind NUI/PSWD wirklich "weg".

TER beginnt den Terminalmode, das Herzstück des ganzen Pgm. Tasten und Anzeige sind hier "unnormal", weil die Zeitprobleme sonst unüberwindlich werden. KBII wird nicht bedient, SML wird bedient aber als RESERVE III angezeigt, SHIFT wird bedient aber als RAD angezeigt, CTRL wird bedient aber als GRAD angezeigt. Die Zeichen !"#\$\$ liegen auf SHIFT (RCL) (F1-F5 sind den Texten 1-5 vorbehalten). ENTER wird als "dicker Punkt" angezeigt. Keine neue Zeile im Display. RESERVE I zeigt an, ob z.Zt. Zeichen in das downloadfile geschrieben werden. Die Anzeige der Reservesymbole erfolgt um eine Displayzeile verzögert. Zu Beginn ist download inaktiv und SML aus.

Sonderbehandlung ankommender Zeichen: CR wird ignoriert, wenn vorher nicht "richtige" Zeichen verarbeitet wurden. Andere Zeichen kleiner Space werden immer ignoriert. Von mehreren aufeinander folgenden Spaces wird nur das erste bearbeitet. Komma wird als Apostroph in's downloadfile geschrieben.

Sondertasten: OFF/ON beenden Terminal. Das muß quittiert werden. CL löscht den Puffer. Nur wenn man die Box an dieser Stelle kennt! "Hübsche" Einstiegsbilder sehen auf unserem Display sowieso furchtbar aus. MODE unterbricht das Abspeichern im downloadfile (oder hebt diese Unterbrechung wieder auf). F1-F5 senden die Texte 1-5. Ohne Wirkung, wenn der entspr. Text nicht (mehr) verfügbar. Falls die Box ohne Echo arbeitet, werden Tasten mitten zwischen den Empfangsdaten angezeigt. Oder man muß warten, bis der Puffer abgearbeitet ist.

F6 startet den upload-Betrieb, muß aber quittiert werden. Während des Sendens werden nur OFF/ON/CL bedient. Sonst werden die Zeichen gesendet wie Tasten gedrückt. Der Tastaturpuffer wird solange vorrangig abgearbeitet, bis leer.

ERROR: Timeout ist schon beschrieben. Wenn nicht upload: Disk-fehler deaktivieren download. Message. Abbruch wird vorgeschlagen. Gegf. kann weitergearbeitet werden. Bei upload: eröffnet eine Datei nicht korrekt, wird mit Message die nächste versucht. Sonst wird mit Message Terminal abgebrochen.

Im Terminalmode kann sich das Pgm wegen timeout nicht "festrennen". Außerhalb schon (bei fehlendem Drucker z.B.). BREAK, DEF X erzwingt dann Rückkehr in's Menü.

```

5 JS=ES
10 H=H+1,ES=RXD$ :ON LEN ES:GOTO 15,45:ES="Q"
15 IF ES<>A$THEN 25ELSE PRINT Z$::IF XIF RPRINT #1,B$::R=0
20 JS=ES:GOTO 45
25 IF ES<" THEN ELSE IF J$<>E$THEN 30ELSE IF E$=" THEN 10
30 PRINT E$::IF ELSE 40
35 R=1:IF E$=","PRINT #1,I$;ELSE PRINT #1,E$;
40 JS=ES:IF H<15THEN 10
45 H=0:IF PEEK N=PEEK O:GOTO 10ELSE CALL Y:C$=CHR$ PEEK Z,A=INSTR (F$,C$):IF ELSE
60
50 ON AGOSUB 135,140,150,155,275,160,170,180,180,180,180,180,225
55 GOTO 45
60 G=ASC C$:IF ELSE 70
65 IF GAND UIF G<MLET A=GAND P:IF ALET G=A,C$=CHR$ A:GOTO 105
70 IF HELSE:80
75 IF STHEN 80ELSE IF C$<"A"THEN 80ELSE IF C$<"Z"THEN 80ELSE LET G=G+T,C$=CHR$ G
80 IF DELSE 90
85 IF C$=" "LET C$=CHR$ 880:GOTO 105ELSE IF G>57IF G<MLET G=G+U,C$=CHR$ G:GOTO 10
5
90 IF ELSE 105
95 IF C$>"A"IF C$<"Z"IF MTHEN 105ELSE LET G=G+T,C$=CHR$ G:GOTO 105
100 R$(0)=C$,C$=R$(INSTR (S$(0),C$))
105 IF C$=A$PRINT :F=0ELSE LPRINT C$::F=1
110 S=0,C=0,D=0:DEGREE :IF Y$THEN 10
115 IF C$<>A$THEN 125ELSE PRINT Z$::IF XIF RPRINT #1,B$::R=0
120 GOTO 45
125 IF C$=" THEN 45ELSE PRINT C$::IF ELSE 45
130 R=1:IF C$=" "PRINT #1,I$;GOTO 45ELSE PRINT #1,C$;GOTO 45
135 C=0,D=0,S=0:IF SRADIAN :RETURN ELSE DEGREE :RETURN
140 C=0,D=0,S=0,M=0:A=PEEK V:IF MLET A=AOR 16ELSE LET A=AAND 239
145 POKE V,A:DEGREE :RETURN
150 S=0,D=0,C=0:IF CGRAD :RETURN ELSE DEGREE :RETURN
155 S=0,C=0,D=0:DEGREE :RETURN
160 IF X$=" "RETURN ELSE LET X=X=0,A=PEEK V:IF XLET A=AOR 64ELSE LET A=AAND 191
165 POKE V,A:IF RPRINT #1,B$::R=0:RETURN ELSE RETURN
170 POKE &F146,PEEK &F140,PEEK &F141:POKE &F144,PEEK &F140,PEEK &F141
175 POKE &F146,PEEK &F140,PEEK &F141:POKE &F144,PEEK &F140,PEEK &F141:RETURN
180 DEGREE :S=0,D=0,C=0,A=ASC C$-17,T$=T$(A):IF T$IF FLPRINT :LPRINT T$:P=0:ELSE L
PRINT T$
185 IF Y$ELSE PRINT Z$;T$;Z$;
190 IF A>0FOR A=A-1TO 0STEP -1:T$(A)="":NEXT A:RETURN ELSE RETURN
195 CLS :CURSOR 0,1:PRINT R$;"(J/N)?" :GOSUB 205:AJ=ASC T$:IF AJ=74OR AJ=78LET AJ=A
J=74:RETURN
200 BEEP 1,20:GOTO 195
205 KBUFF$="":CALL Y:T$=CHR$ PEEK Z:KBUFF$="":RETURN
210 ON ERROR GOTO 0:RETURN
215 ON ERROR GOTO 0:ON ERROR GOTO 270:RETURN
220 OUT &17,&41:PRINT Z$;R$;"(J/.)?" :Z$::GOSUB 205:OUT &17,0:U$="":RETURN
225 R$="UPLOAD" :GOSUB 220:IF T$<"J"RETURN ELSE GOSUB 640:L=2,I=9,S$="",W$="":FOR
I=0TO 2
230 T$=B$(I+19):IF T$=" "NEXT I:GOTO 260ELSE LET K=0,S$="":OPEN T$FOR INPUT AS #2:
FOR J=0TO 1:S$=I
235 IF EOF (2)NEXT J:GOTO 255ELSE LET J=0:INPUT #2,A$(0)
240 FOR L=1TO 1:A=0:C$=INKEY$ :IF C$=U$THEN 245ELSE LET U$=C$,A=INSTR (H$,C$)
245 ON AGOSUB 275,275,170:LPRINT A$(0):PRINT A$(0):Z$::NEXT L:GOSUB 265:IF Y$IF (X
MOD 6)=0GOSUB 170
250 NEXT J
255 CLOSE #2:S$="":NEXT I:IF Y$GOSUB 170
260 U$="",W$="":KBUFF$="":C=0,D=0,S=0:DEGREE :RETURN

```

```

DR.:GOSUB 195
500 IF AJ=0 THEN 510 ELSE SETCOM "COM1: ", 1200, 8, N, 2, N, N, SNDSTAT "COM1: ", 59, 0, OUTSTAT
"COM1: ", 0
505 PCONSOLE "COM1: ", 80, 2, SETDEV "COM1: ", PO
510 FOR K=0 TO 1: K=EOF (1): IF KNEXT K: ELSE INPUT #1, AS(0): LPRINT AS(0): NEXT K
515 CLOSE #1: SETDEV "COM1: ": R$="KILL": C$="GOSUB 195: IF AJKILL C$: LET B2$(13)="N": R
ETURN ELSE RETURN
520 "LST": LPRINT TAB 27: "BOX=" + P$: FOR I=0 TO 11: LPRINT B1$(I): TAB 19: B2$(I):
525 LPRINT TAB 30: B1$(I+12): TAB 49: B2$(I+12): NEXT I: LPRINT : LPRINT : RETURN
530 "SET": IF F$(3): RETURN ELSE OUTSTAT "COM1: ", 3: GOSUB 170: GOSUB 215: INIT "COM1: ", VA
L B2$(10)
535 SETCOM "COM1: ", VAL B2$(3): IF B2$(0)="7": SETCOM "COM1: ", 7: ELSE SETCOM "COM1: ", 8
540 IF B2$(1)="2": SETCOM "COM1: ", 2: ELSE SETCOM "COM1: ", 1
545 C$=B2$(2): IF C$="N": SETCOM "COM1: ", N: ELSE IF C$="O": SETDEV "COM1: ", O: ELSE IF C
$="E": SETCOM "COM1: ", E
550 IF B2$(5)="N": SETCOM "COM1: ", N: ELSE SETCOM "COM1: ", X
555 IF B2$(4)="N": SETCOM "COM1: ", N: ELSE SETCOM "COM1: ", S
560 SNDSTAT "COM1: ", VAL B2$(6), VAL B2$(7): PCONSOLE "COM1: ", VAL B2$(8), VAL B2$(9): F
$(3)="1": RETURN
565 FOR I=19 TO 21: C$=B2$(1): IF C$GOSUB 585: B2$(1)=C$: NEXT I: ELSE NEXT I
570 IF B2$(13)="J": LET C$=B2$(12): IF C$="" LET B2$(13)="N": RETURN ELSE GOSUB 585: IF
C$RETURN ELSE LET B2$(13)="N"
575 C$=B2$(12): IF C$="" RETURN: ELSE GOSUB 585: IF C$RETURN: ELSE LET C$=B2$(12): GOS
UB 580: B2$(12)=C$: RETURN
580 GOSUB 210: ON ERROR GOTO 590: OPEN C$FOR OUTPUT AS #1: CLOSE #1: KILL C$: GOSUB 215
: RETURN
585 GOSUB 210: ON ERROR GOTO 590: OPEN C$FOR INPUT AS #1: CLOSE #1: GOSUB 215: RETURN
590 C$="" : GOSUB 215: RETURN
595 "END": R$="END ALL": GOSUB 195: IF AJCLEAR : CALL 493: END ELSE RETURN
600 GOSUB 640: A=ERN : IF W$ELSE 625
605 IF A=1600R A=1640R A=1670R A=168ELSE 615
610 PRINT : PRINT "DISK ERROR": X=1, R=0: GOSUB 160: X$="" : GOSUB 275: GOTO 10
615 IF A=143: PRINT : PRINT "TIMEOUT": DEEP 2, 30: KBUFF$="" : U$="" : GOTO 10
620 PRINT : PRINT "ERROR": STR$ A: IN : STR$ ERL: GOSUB 275: GOTO 10
625 IF S$="" PRINT : PRINT "OPEN ERROR": T$: DEEP 2, 30: IF I<2 NEXT I: ELSE GOSUB 275: GOT
O 260
630 IF A=143: PRINT : PRINT "TIMEOUT": GOSUB 275: L=0: NEXT L
635 CLS : PRINT "FATAL ERROR": Q$="ERR. ": STR$ A: IN : STR$ ERL : FOR I=20 TO 3
5: BEEP 3, I: NEXT I: GOTO 280
640 ON ERROR GOTO 0: ON ERROR GOTO 600: RETURN
645 "TER": GOSUB 530: X$=B2$(12): IF X$IF B2$(13)="J"
X$FOR OUTPUT AS #1: B2$(13)="J"
650 Y$=CHR$(B2$(11)) + "*" + 255): FOR I=0 TO 4: T$(I)=B2$(I+14): NEXT I: DV=INT (VAL B2$
(22)/3)
655 V=64E: POKE V, PEEK VAND 143: P=31, T=32, U=64, W=128, C=0, D=0, M=0, U$="" : X=0, Q=0, R=
0, F=0
660 X$="" : GOSUB 170: CLS : PRINT "***** TERMINAL *****": SETDEV "COM1: ", PO: POWER AOFF 1
665 N=697F, O=N+1: OUTSTAT "COM1: ", 0: GOSUB 640: XBUFF$="" : BREAK OFF : GOTO 10
670 "Z": CLEAR : A$=CHR$ 13, B$=A$+CHR$ 10, G$=CHR$ &1F+CHR$ 24+CHR$ 17+CHR$ 18+CHR$ 19
+CHR$ 20+CHR$ 21+CHR$ 14+CHR$ 15
675 F$=CHR$ 1+CHR$ 2+CHR$ 3+CHR$ &1B+CHR$ 14+CHR$ 15+CHR$ &1F+CHR$ 24+CHR$ 17+CHR$
18+CHR$ 19+CHR$ 20+CHR$ 21+CHR$ 22
680 I$=CHR$ 39, H$=CHR$ 14+CHR$ 15+CHR$ 24, N$="TBAQ"+CHR$ 13+CHR$ &1B+CHR$ 10+CHR$ 2
4, O$=CHR$ 8+CHR$ 8+CHR$ 8
685 O$=O$+O$, O$=O$+O$, N$=N$+CHR$ 9, Y$=166, Z=6F083, Z$=CHR$ 254: DIM F$(3)*1, B1$(23)*
18
690 DIM B2$(23)*15, M$(1)*26: AJ=0, MO=0, DV=0: DIM A$(0)*80, T$(5), S$(0)*24, R$(24)*1

```



```
1:"A"CLS :I=Z:INPUT "NEUEINGABE(N)/ERGÄNZEN(E)? ";I$:CLS :IF I$="N"CLEAR :
DIM A$(255)*70:Z=1
2:I=I+1:CLS :PRINT I;".DATE":KBUFF$ =" ":INPUT A$(I):X=LEN A$(I):IF X>1THEN
Z:ELSE LET Z=I-1:PRINT "ENDE":END
3:"S"INPUT "NAME:";FI$:MAXFILES =1:OPEN "X:"+FI$FOR OUTPUT AS #1:PRINT #1,Z:
FOR I=1TO Z:PRINT #1,A$(I):NEXT I:CLOSE #1:END
4:"D"CLS :CLEAR :DIM A$(255)*70
5:INPUT "NAME:";FI$:MAXFILES =1:OPEN "X:"+FI$FOR INPUT AS #1:INPUT #1,Z:FOR
I=1TO Z:INPUT #1,A$(I):NEXT I:CLOSE #1:END
6:"F"WAIT :PRINT "BLÄTTEN <ENTER>":CLS :FOR I=1TO Z:WAIT 0:PRINT I;")DATEI":
WAIT :PRINT A$(I):CLS :NEXT I:WAIT 0:END
7:"G"WAIT :PRINT "DRUCKEN <ENTER>":FOR I=1TO Z:PRINT A$(I):CLS :NEXT I:WAIT
0:END
8:"H"Q=0:CLS :INPUT "SUCHBEGRIFF:";SB$:FOR I=1TO Z:N=INSTR (A$(I),SB$):IF N>
0WAIT :Q=1:PRINT A$(I):WAIT 0
9:NEXT I:PRINT "DAS WARS !!":IF Q=0THEN PRINT "NICHTS GEFUNDEN
10:END
11:"J"Q=0:CLS :INPUT "SUCHBEGRIFF:";SB$:FOR I=1TO Z:N=INSTR (A$(I),SB$):IF N>
0WAIT :Q=1:LPRINT A$(I):WAIT 0
12:NEXT I:LPRINT "DER SUCHBEGRIFF WAR",SB$:PRINT "DAS WARS !!":IF Q=0THEN
PRINT "NICHTS GEFUNDEN"
13:END
14:"K"FOR I=1TO Z:PRINT A$(I):PRINT I;")DATEI ÄNDERN (J/N)";:INPUT FR$:IF FR$
="J"CLS :INPUT A$(I):I=Z:END
15:CLS :NEXT I:END
```

Wenn Sie die Überschrift lesen, und sich dann das kurze Listing anschauen, werden sie sicher denken, daß dies ein Witz sein soll. Aber lassen Sie sich, durch eine kurze Übersicht, über die einzelnen Funktionen des Programmes, überzeugen!
Folgende Funktionen werden von diesem Programm geboten:

- | | |
|------------------------------|-------------------------------|
| 1. Daten eingeben, ergänzen. | 5. Datei drucken. |
| 2. Daten sichern. | 6. Suchen in Datei (Display). |
| 3. Daten einladen. | 7. Suchen in Datei (Drucker). |
| 4. In Datei blättern. | 8. Daten ändern. |

1. Mit DEF A gelangen Sie zur Funktion *Neueingabe und ergänzen*: Bei Neueingabe wird eine, eventuell schon im Speicher enthaltene, Datei gelöscht und Sie können eine neue Datei anlegen. Die Datei kann aus bis zu 255 Datensätzen, mit jeweils bis zu 70 Zeichen, bestehen. Durch Ändern der DDM-Anweisung, in den Zeilen 1 und 4, läßt sich das Programm auch an einen kleineren Arbeitsspeicher anpassen. Sie können beliebige Texte eingeben, die allerdings keine Anführungszeichen enthalten dürfen! Dadurch läßt sich dieses Programm auch als Textverarbeitung verwenden. Da aber auch eine Suchroutine zum Umfang dieses Programmes gehört, ist es auch hervorragend als Adressverwaltung oder allgemeines Dateiprogramm zu verwenden.
2. Mit DEF S gelangen Sie zur Funktion *Daten sichern*: Mit Hilfe dieses Unterprogrammes ist es möglich, die eingegebene Datei abzuspeichern. Geben Sie, bei der Frage nach dem Namen, den Namen ein, unter dem Sie den Text auf Diskette abspeichern wollen (z.B.: DATEI1.DAT). Es ist auch möglich, das Programm an die Verwendung einer RAM-Disk anzupassen, dazu ändern Sie, in den Zeilen 3 und 5, X in S1 oder S2 um.
3. Mit DEF D gelangen Sie zur Funktion *Daten einladen*: Nach Eingabe des Dateinamens, wird die entsprechende Datei von Diskette geladen, und kann danach, mit den anderen Funktionen des Programmes, weiterverarbeitet werden.
4. Mit DEF F gelangen Sie zur Funktion *Blättern in der Datei*: Wollen Sie zum Beispiel eine Datei, auf den Display, anschauen, können Sie dieses, mit einem Druck auf die ENTER-Taste, tun. Es wird die Nummer des Datensatzes und sein Inhalt ausgegeben.
5. Mit DEF G gelangen Sie zur Funktion *Datei drucken*: Nach einem Druck auf die ENTER-Taste, wird die gesamte Datei ausgedruckt.
6. Mit DEF H gelangen Sie zur Funktion *Suche in Datei, nach Suchbegriff (Display)*: Haben Sie zum Beispiel eine umfangreiche Adressdatei eingegeben (Ein Datensatz (70 Zeichen) reicht aus für Telefonnummer und Adresse.) und wollen nun eine bestimmte Adresse

Günter Semmelmeier
Poststr. 20
7703 Rielasingen 1

Eingabe - Korrektur - Kontrolle (Programm - Beschreibung)

Mit diesem Programm ist es möglich Daten einzugeben und zu pflegen.
Die Größe des Datenfeldes wird mit N und M bestimmt.
DIM X(M,N) Zeile 110
Nach der Eingabe aller Werte wird dieser Programm-Teil verlassen.
Die neu eingegebenen, korrigierten oder kontrollierten Werte müssen mit ENTER bestätigt werden.
Der nächste Wert wird nun angezeigt.
Erfolgt anstelle einer Zahl der Buchstabe A so schaltet sich der COMPUTER aus. Beim Einschalten, auch nach Tagen, können die Daten weiter bearbeitet werden - als wäre nichts geschehen.
Beim Buchstaben I können bestimmte Werte schnell angesprochen werden.
Bei E Ende. Viel Spass !!

PC-1600 Eingabe - Korrektur - Kontrolle bei DIM X(M,N)

```
10:'G.Semmelmeier 11/1988
20:'Poststr.20 (7703) Rielasingen
30:''
40:'PC-1600 Eingabe - Korrektur - Kontrolle bei DIM X(M,N)
50:'[A] = Aus ; bei UN weiter im Programm
60:'[E] = Ende
70:'[I] = Nr. des Datentupels
80:''
90:CLS :BEEP 1:PRINT :INPUT "Anzahl M : ";M
100:BEEP 1:INPUT "Anzahl N : ";N
110:DIM X(M,N)
120:"A":WAIT 0:W1$=CHR$ 8+CHR$ 8:W1$=W1$+W1$+W1$+W1$:W1$=W1$+W1$
130:FOR I=1TO N
140:FOR J=1TO M
150:CLS :CURSOR 5,0:PRINT "Daten [A],[E],[I]":A$="":PRINT
160:PRINT "(+STR$ J+","STR$ I+)"= ";W$=STR$ X(J,I):KBUFF$ =W$+LEFT$
(W1$,LEN W$)
170:INPUT " ";W$:A$=LEFT$ (W$,1):IF INSTR ("AEI",A$)GOTO 210
180:X(J,I)=VAL W$:A$=""
190:NEXT J
200:NEXT I:GOTO "B"
210:IF A$="A"POWER AOFF :GOTO 150
220:IF A$="E"CLS :CURSOR 10,1:PRINT "Ende":END
230:IF A$="I"CLS :INPUT "Datentupel-Nr. ";I:J=1:IF I<0OR I>NGOTO 230
ELSE GOTO 150
240:''
250:"B":WAIT :CLS :CURSOR 1,1:PRINT "Programm-Fortsetzung....."
```

Dirk Mahrholz
Goethestr. 85
2850 Bremerhaven

Programmbeschreibung:

Dieses MaPro nutzt die internen IOCS-Routinen "SAVELCD" und "LOADLCD" des PC-1600. Damit ist es bequem möglich den Displayinhalt (z.B. Graphik) schnell und platzsparend zu speichern und ebenso wieder zu laden.

Durch seine Kürze eignet sich das MaPro sehr gut als Unterprogramm bei Graphikanwendungen oder Spielen.

BEISPIEL

SALODIS

```
5:' SAVE/LOAD-DISPLAY
10:CLS :CLEAR
15:POKE &C100,&F5,&C5,&D5,&E5,&3E,&00,&32,&49
20:POKE &C108,&C1,&21,&50,&C1,&22,&4A,&C1,&3A
25:POKE &C110,&4C,&C1,&FE,&01,&C2,&24,&C1,&3A
30:POKE &C118,&49,&C1,&ED,&5B,&4A,&C1,&CD,&5D
35:POKE &C120,&01,&C3,&2E,&C1,&3A,&49,&C1,&ED
40:POKE &C128,&5B,&4A,&C1,&CD,&60,&01,&2A,&4A
45:POKE &C130,&C1,&01,&9C,&00,&09,&22,&4A,&C1
50:POKE &C138,&3A,&49,&C1,&3C,&32,&49,&C1,&FE
55:POKE &C140,&04,&C2,&0F,&C1,&E1,&01,&C1,&F1,&C9
```

423 Bytes

```
5:CLS :CLEAR :WAIT 0
10:DIM T$(0)*80
15:PRINT "SAVE/LOAD -DISPLAY":PRINT "1: SAVE DISPLAY"
20:PRINT "2: LOAD DISPLAY":PRINT "3: ENDE"
25:A$=INKEY$ :IF A$=""GOTO 25
30:IF A$="3"CLS :KBUFF$ ="" :END
35:IF A$="2"GOTO 60
40:' SAVE
45:CLS :KBUFF$ ="" :INPUT " ";T$(0)
50:IF T$(0)=""GOTO 5
55:POKE &C14C,1:CALL &C100:GOTO 5
60:' LOAD
65:POKE &C14C,2:CALL &C100:KBUFF$ =""
70:IF INKEY$ =""GOTO 70ELSE GOTO 5
```

318 Bytes

```

HEX-DUMP :
C100 : F5 C5 D5 E5 3E 00 32 49 C1 21 50 C1 22 4A C1 3A = 0787
C110 : 4C C1 FE 01 C2 24 C1 3A 49 C1 ED 5B 4A C1 CD 5D = 0874
C120 : 01 C3 2E C1 3A 49 C1 ED 5B 4A C1 CD 60 01 2A 4A = 06EC
C130 : C1 01 9C 00 09 22 4A C1 3A 49 C1 3C 32 49 C1 FE = 064E
C140 : 04 C2 0F C1 E1 D1 C1 F1 C9 00 00 00 00 00 00 = 05C3

```

&C149 Auf diesem Speicherplatz steht die Displayzeilennummer (&00...&03).

&C14A+ Hier steht die Startadresse des RAM-Bereiches, in welchem sich die
&C14B Bitmusterdaten befinden.

&C14C Eine "1" bedeutet hier SAVen des Displays, eine "2" bedeutet LOAD.

&C150- Speicherbereich der Bitmusterdaten (4*156=624 Bytes).
&C3C0

&0150+ Einsprungadressen der entsprechenden IOCS-Routinen (siehe Technical
&0160 Reference Manual).

Vor der Programmeingabe sollte mit NEW"S0:",&03C1 ein Teil des RAMs reser-
viert werden.

Jürgen Gartinger
Auf der Kuhweide 13
4600 Dortmund 30

PC-1600 DATENBAUMPROGRAMM

Kleines Expertensystem DATENBAUMPROGRAMM

Dieses Programm erstellt Ihnen eine Datei in Form eines Datenbaumes. Sie können ein solches Programm benutzen, um, anhand vorher abgefragter Daten, eine Bestimmung beliebiger Objekte, Gruppen oder Arten durchzuführen. In einem Dialog zwischen Ihnen und dem Rechner lernt dieses Programm praktisch immer neue Daten und speichert sie. Natürlich können diese Daten auch auf Diskette gespeichert und wieder von dieser geladen werden.

In einem Datensatz können bis zu 125 Objekte (Tiere, Mineralien, Pflanzen, Chemische Analysen und vieles anderes mehr.) mit ihren speziellen Eigenschaften gespeichert werden. Es können dabei bis zu 40 Zeichen für jedes Objekt und jede Eigenschaft benutzt werden.

Im Bestimmungsmodus führt das Programm mit Ihnen einen Dialog durch, bis es das wahrscheinlichste der ihm bekannten Objekte gefunden hat, indem es laufend neue Daten verlangt. Sollte für Ihre Zwecke eine Datei mit 125 Objekten nicht ausreichen, können sie beliebig viele Dateien mit jeweils maximal 125 Objekten auf einer oder mehreren Disketten anlegen. Beim Beispiel Tierdatei könnten diese Dateien zum Beispiel wie folgt aussehen: Datei 1 = Säuger; Datei 2 = Reptilien; Datei 3 = Vögel; Datei 4 = Fische, usw. Alleine von Ihrem Fleiß hängt damit die Anzahl der bestimmbar Objekte ab. Die Dateneingabe erfolgt dabei nicht in der Form von DATA-Zeilen, die Sie im PRO-Mode eingeben müssen, sondern in einem echten Dialog mit dem Rechner. Sie können also in aller Ruhe zu Hause, mit Hilfe entsprechender Literatur, die Datei erstellen, um vor Ort dann auf sie zurückgreifen zu können.

Kommen wir nun zu einem einfachen Beispiel, der Bestimmung von Tieren: Nach dem Start des Programmes mit RUN oder DEF M erscheint das folgende Menue. Wählen Sie nun den Menüpunkt (1) an, darauf folgt

Programm Neustart (1)	eine Abfrage, ob eine Kurzanleitung gewünscht wird. Anschließend müssen Sie den Oberbegriff der Datei eingeben, z.B.
Daten ergänzen (2)	Tiere. Darauf verlangt das Programm die
Daten abfragen (3)	Eingabe von zwei grundverschiedenen
Daten save/laden (4)	Arten, z.B. 1) Hecht und 2) Adler.

Danach werden Sie aufgefordert, den Unterschied zwischen diesen beiden Arten, in Frageform, einzugeben, z.B. Schwimmt das Tier? Nun müssen Sie eingeben ob diese Eigenschaft auf den Adler zutrifft (mit J/N). Aus dieser Information baut das Programm nun den Datenbaum auf. Wenn Sie weitere Tiere, mit ihren Eigenschaften eingeben, fahren Sie einfach fort. Der Dialogbetrieb des Programmes macht es Ihnen recht einfach.

Mit dem Programmteil *Daten ergänzen*, können Sie bereits vorhandene Dateien ergänzen. Durch den Menüpunkt *Daten abfragen*, gelangen Sie in den Modus, der es Ihnen ermöglicht, die Objekte zu bestimmen. Sollte einmal das vom Programm ermittelte Ergebnis nicht stimmen, was nur auf mangelnde Informationen des Programmes zurückzuführen ist (Sie haben dann einfach zu wenig Objekte oder Daten eingegeben), haben Sie die Möglichkeit, den Fehler zu korrigieren, falls Ihr Wissensstand dazu ausreicht.

```

10:CLS :WAIT 50:PRINT :WAIT 200:PRINT "P
   C-1600 DATENBAUMPROGRAMM"
20:*****
30:*EIN PROGRAMM VON *
40:*JÜRGEN GARTINGER *
50:*AUF DER KUHWEIDE 13*
60:*4600 DORTMUND 30 *
70:*
80:*****
100:"M":WAIT 0:CLS
110:PRINT "PROGRAMM-NEUSTART ____ (1)
120:PRINT "DATEN ERGÄNZEN ____ (2)
130:PRINT "DATEN ABFRAGEN ____ (3)
140:PRINT "DATEN LADE/SAUEN ____ (4)
150:F$=INKEY$:IF F$="1"GOTO 200
160:IF F$="2"GOTO "B"
170:IF F$="3"GOTO "B"
180:IF F$="4"GOTO 920
190:GOTO 150
200:CLEAR :DIM A$(125)*40,J(125),N(125),U
   $(1)*40,X$(1)*40,OB$(1)*25
210:F$="2"
220:"A":WAIT 50:CLS :PRINT "KURZANLEITUNG
   (J/N) : "
230:X$=INKEY$:IF X$=""GOTO 230
240:IF X$="N"GOTO 320
250:WAIT 300
260:CLS :PRINT "MIT DIESEM PROGRAMM KÖNNE
   NSIE BESTIMMTEN OBJEKTEN EINE EIGEN
   SCHAFT"
270:CLS :PRINT "ZUTEILEN,DIE SIE VON ANDE
   -REN OBJEKTEN UNTERSCHIEDETIM PROGRAM
   MTEIL "
280:CLS :PRINT "DATEN ABFRAGEN WERDEN DI
   EALS DATENBAUM GESPEICHER- TEN DATEN
   ABGEFRAGT
290:CLS :PRINT "UND ES KANN AUF DIESE ART
   EINE BESTIMMUNG VON OBJEK- TEN ERFOLG
   EN,
300:CLS :PRINT "VORAUSGESETZT DASS,DAS GE
   -SUCHTE OBJEKT VORHER IM PROGRAMMTE
   IL
310:CLS :PRINT "DATEN EINGEBEN BERÜCKSICH
   TIGT WURDE.":WAIT 0
320:CLS :PRINT "OBERBEGRIFF FÜR DIE
   OBJEKTE :":KBUFF$="" :INPUT OB$(1)
330:CLS :PRINT "GEBE NUN DIE NAMEN VON
   2 GRUNDVERSCHIEDENEN ARTEN EIN"
340:INPUT "OBJEKT 1 : ";A$(2):A$(2)="*"+A
   $(2)
350:INPUT "OBJEKT 2 : ";A$(3):A$(3)="*"+A
   $(3)
360:CLS :WAIT 200:PRINT "BESCHREIBE NUN D
   EN UNTER- SCHIED ZWISCHEN":WAIT 0
370:PRINT MID$(A$(2),2,25):PRINT " und "
   :PRINT MID$(A$(3),2,25):PRINT "IN FR
   AGEFORM";
380:INPUT " ? : ";A$(1):A$(1)="/" +A$(1)
390:CLS :PRINT "DIE ANTWORT FÜR ":PRINT
   MID$(A$(2),2,25):PRINT " LAUTET (J/N
   )";
400:INPUT FF$:IF FF$="J"LET J(1)=2:LET N(
   1)=3
410:IF FF$="N"LET J(1)=3:LET N(1)=2
420:N0=3
430:"B":IF N0<3CLS :BEEP S:PAUSE "K E I N
   E DATEN VORHANDEN":GOTO "M"
440:CLS :KBUFF$="" :PRINT "WELLEN SIE
   ":PRINT OB$(1):INPUT "BESTIMMEN(J/N)
   : ";A$:IF A$="J"GOTO 450
450:GOTO "M"
460:K=1
470:GOSUB 660
480:IF T=0GOTO 520
490:WAIT 10:CLS :PRINT "NOCH EINE BESTIM
   MUNG":INPUT "(J/N) ? : ";FR$
500:IF FR$="N"GOTO "M"
510:GOTO 460
520:CLS :PRINT "GEBEN SIE DEN":PRINT OB$;
   :PRINT "NAMEN EIN ":INPUT U$(1)
530:CLS :WAIT 100:PRINT "BESCHREIBE DEN U
   NTERSCHIED ZWISCHEN"
540:CLS :WAIT 0:PRINT U$(1):PRINT " UND "
   :PRINT MID$(A$(K),2,26)
550:INPUT "IN FRAGEFORM : ";X$(1)
560:CLS :PRINT "DIE ANTWORT FÜR ":PRINT U
   $(1):PRINT " LAUTET (J/N)";
570:INPUT A$
580:IF A$(K)"N"AND A$(K)"J"GOTO 560
590:N0=N0+1:A$(N0)="/" +X$(1)
600:IF A$="J"LET J(N0)=N2+1:N(N0)=K
610:IF A$="N"LET J(N0)=K:N(N0)=N0+1
620:IF B$="J"LET J(K0)=N0
630:IF B$="N"LET N(K0)=N0
640:N0=N0+1:A$(N0)="/" +U$(1)
650:GOTO 430
660:K=1
670:IF LEFT$(A$(K),1)="/"LET K0=K:GOTO 8
   10
680:IF F$="2"GOTO 760
690:CLS :WAIT 0:PRINT "DER SUCHBEGRIFF IS
   T":PRINT :WAIT 300:PRINT MID$(A$(K)
   ,2,26);" ":T=1:WAIT 0:CLS
700:PRINT "WENN SIE 100% SICHER SIND,DASS
   DIE ANTWORT NICHT"
710:WAIT 400:PRINT "STIMMT,KÖNNEN SIE DIE
   RICHTIGE ANTWORT SPEICHERN
720:CLS :WAIT 0:INPUT "WOLLEN SIE EINE NE
   UE ANT- WORT SPEICHERN (J/N) : ";W$
730:IF W$="N"RETURN
740:IF W$<"J"GOTO 720
750:CLS :PRINT "BITTE BEANTWORTEN SIE
   FOLGENDE FRAGEN!"
760:CLS :PRINT "IST DER SUCHBEGRIFF: " :
   PRINT :PRINT MID$(A$(K),2,26)
770:INPUT "(J/N)?";A$
780:IF A$="J"LET T=1:RETURN
790:IF A$="N"LET T=0:RETURN
800:GOTO 760
810:WAIT 0:CLS :PRINT MID$(A$(K),2,30)
820:INPUT "(J/N) ? : ";B$
830:IF B$="J"GOTO 860
840:IF B$="N"GOTO 880
850:GOTO 810
860:IF J(K)0LET K=J(K):GOTO 670
870:GOTO 760
880:IF N(K)0LET K=N(K):GOTO 670
890:GOTO 760
900:CLS :WAIT 0
910:PRINT "LOAD.....(1)
920:PRINT "SAVE.....(2)
930:PRINT "INFO.....(3)
940:PRINT "BITTE TASTE !"
950:FA$=INKEY$:IF INKEY$=""GOTO 950
960:IF INKEY$="1"GOTO 1020
970:IF INKEY$="2"GOTO 1280
980:IF INKEY$="3"GOTO 1160
990:GOTO 900
1020:CLEAR :CLS :PRINT "BITTE DATEINAMEN E
   INGEBEN FILENNAMEE.EXTENSION"

```

```

1010: INPUT F1$:CLS :PRINT "DATEN WERDEN GE
LADEN:"
1020: DIM A$(120)*40, J(125), N(125), U$(1)*40
, X$(1)*40, O$(1)*25
1030: MAXFILES =1: CPEN "X:" + F1$ FOR INPUT AS
#1
1040: INPUT #1, N0
1050: FOR I=1 TO N0
1060: FOR H=1 TO 3: INPUT #1, A$(I): INPUT #1, J
(I): INPUT #1, N(I): NEXT H: NEXT I
1070: CLOSE #1: BEEP 5: CLS : PAUSE "DATEN SIN
D GELADEN": PRINT "OBERBEGRIFF ": PRINT
O$(1): GOTO "M"
1080: IF N0<3: BEEP 5: PAUSE "K E I N E
DATEN VORHANDEN:????": GOTO "
M"
1090: CLS : PRINT "BITTE DATEINAMEN EINGEBEN
FILENAME.E.EXTENSION"
1100: INPUT F1$:CLS :PRINT "DATEN WERDEN GE
SAUT !"
1110: MAXFILES =1: CPEN "X:" + F1$ FOR OUTPUT
AS #1
1120: PRINT #1, N0
1130: FOR I=1 TO N0
1140: FOR H=1 TO 3: PRINT #1, A$(I): PRINT #1, J
(I): PRINT #1, N(I): NEXT H: NEXT I
1150: CLOSE #1: BEEP 5: CLS : PAUSE "DATEN SIN
D GESAUT ": PRINT "OBERBEGRIFF ": PRINT
O$(1): GOTO "M"
1160: LPRINT "FREIE DISKETTENKAPAZITÄT = ";
DSKF "X":
1170: LPRINT "-----"
-----
1180: LFILES "X":
1190: GOTO "M"

```

Programmbeschreibung:

Dieses MaPro nutzt die IOCS-Routinen "GCRSRSET" und "PRTGSTR" des PC-1600. Damit ist es nun möglich eine fortlaufende Zeichenkette (T\$) ab der aktuellen Grafikkursorposition anzuzeigen. Die Zeichencodes (ASCII) müssen in aufeinanderfolgenden Speicherplätzen stehen (&EF20-&EF3B).

Mit frdl. Grüßen

D. Mahrholz

PS: Zum Programm "Breitschrift" auf Seite 25 im "Tips und Tricks Programmhandbuch": Bei meinem PC-1600 muss in den Zeilen 64,65 die Zahl 44650 in 44713 geändert werden.

GCPRINT

```

5:CLS :CLEAR :WAIT 0
10:A=256:CALL &02DD,A
15:POKE &EF00,&F5,&C5,&D5,&E5,&ED,&5B,&1C,&EF,&ED,&4B,&1E,&EF,&CD,&4B,&01,&11
20:POKE &EF10,&20,&EF,&3E,&40,&CD,&EE,&00,&E1,&D1,&C1,&F1,&C9
25:DIM T$(0)*27
30:"A"PRINT "PRINT-ERWEITERUNG:":PRINT :PRINT "Cursorposition:"
35:CURSOR 16,2:INPUT "x: ";X:CURSOR 16,3:INPUT "y: ";Y
40:IF X<0OR X>151OR Y<0OR Y>25GOTO 5
45:CLS :INPUT "TEXT:",T$(0)
50:IF T$(0)=""CLS :END ELSE IF LEN T$(0)>26GOTO 5
55:L=X+LEN T$(0)*5:IF L>155GOTO 5
60:CLS :POKE &EF1C,X,0,Y,0:T$(0)=T$(0)+"g"
65:FOR I=1 TO LEN T$(0):POKE (&EF20+I),ASC MID$(T$(0),I,1):NEXT I
70:CALL &EF00
75:IF INKEY$=""GOTO 75ELSE CLS :KBUFF$="" :END
523 Bytes

```

```

&014B Einsprungadresse GCRSRSET
&00EE Einsprungadresse PRTGSTR
&EF1C+ Grafikkursorposition x
&EF1D
&EF1E+ Grafikkursorposition y
&EF1F
&EF20- 27 Bytes für die ASCII-Codes der darzustellenden Zeichen.
&EF3B &(64)=Endmarke

```

Beim PC-1600 ist es möglich, Dateien mit einer Variablen (I)s (-Kunnen D.R.) zu öffnen: z.B.:

```

OPEN "X:TEST" FOR OUTPUT AS #A
INPUT
APPEND

```

Versucht man jetzt aber mit PRINT #A in die Datei zu schreiben, erscheint ERROR 27 (unbekannter Befehl). Das selbe geschieht auch bei INPUT #A. Wer kann mir sagen, wie man in eine Datei mit Variablen schreiben oder aus ihr lesen kann.

Wer weiß wie man beim PC-1600 FOR...NEXT Schleifen mit STEP kleiner 1 eingeben muß. Z.B. PC-1401 oder PC-1475 FOR ...STEP 0,1:NEXT

Holger Hoffmann

Dirk Mahrholz
Goethestr. 86
2850 Bremerhaven

PC-1600 * GCPRINT *

HEX-DUMP :

```

EF00 : F5 C5 D5 E5 ED 5B 1C EF ED 4B 1E EF CD 4B 01 11 = 0936
EF10 : 20 EF 3E 40 CD EE 00 E1 D1 C1 F1 C9 00 00 00 00 = 0775

```



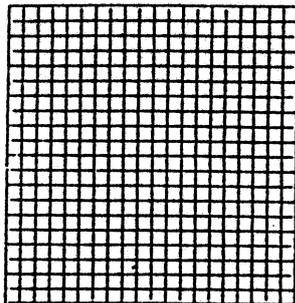
```

165:IF X=1WAIT 400:BEEP 3:PRINT "Farbe : 0 - 3":GOTO 130
170:CURSOR 0,3:PRINT "B=";STR$ B;" H=";STR$ H;" R=";STR$ RA;" F
   ";STR$ C
175:CURSOR 17,0:PRINT "Ok:(J/N)"
180:A$=INKEY$ :IF A$="N"THEN 130
185:IF A$="J"THEN 195
190:GOTO 180
195:COLOR C:GLCURSOR (0,-H):SORGN :LLINE (0,0)-(B,H),0,,B
200:D=INT ((H-1)/RA):A=0:O=0
205:FOR I=1TO D:LLINE (A,I*RA)-(O,I*RA),0:O=A:A=0:O=0:NEXT I
210:D=INT ((B-1)/RA):A=H:O=0
215:FOR I=1TO D:LLINE (I*RA,A)-(I*RA,O),0:O=A:A=0:O=0:NEXT I
220:GLCURSOR (0,0)
STATUS 1

```

755

Beispiel Rasterprogramm
Breite = 200
Höhe = 200
Raster = 10



PC-1600 PC-1600 PC-1600

Das Programm könnte in eine Anschriften- bzw. Adressdatei eingebunden werden. Es arbeitet mit allen Grossbuchstaben:(A-Z,CHR\$ 65 - CHR\$ 90) mit allen Kleinbuchstaben:(a-z,CHR\$ 97 - CHR\$ 122) und den Zeichen (. CHR\$ 46),(- CHR\$ 45),(SPACE CHR\$ 32).

Starten mit RUN. Auf die Frage Dein Name, kann eine Kombination aus den oben genannten Zeichen eingegeben werden. Das Programm berechnet (schreibt als Ausgabe) den ersten Buchstaben, nach CHR\$ 45, CHR\$ 46 und CHR\$ 32 als Grossbuchstaben.

Beispiel:Eingabe FISCHEL GMBH oder kaiser-friedrich-str.
Ausgabe Fischel Gmbh Kaiser-Friedrich-Str.

```

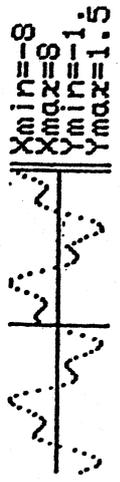
115:'cy by Klaus Nitz,Mardale 12,337 Seesen
120:'
125:CLEAR :CLS :DIM A$(0)*80,A(80)
130:PRINT "Dein Name: ":INPUT A$(0):I=LEN A$(0)
135:A(1)=ASC (LEFT$ (A$(0),1)):A=A(1)
140:IF (A(65OR A)90)AND A(>32)AND A(>46)AND (A(97OR A)122)THEN GOTO 2
   00
145:IF A(1)>91THEN LET A(1)=A(1)-32
150:FOR J=2TO I:A(J)=ASC MID$(A$(0),J,1):A=A(J)
155:IF (A(65OR A)90)AND A(>32)AND A(>46)AND A(>45)AND (A(97OR A)122)
   THEN GOTO 200
160:IF A(J)=46THEN GOTO 185
165:IF A(J)=32THEN GOTO 185
170:IF A(J)=45THEN GOTO 185
175:IF A(J)<91THEN LET A(J)=A(J)+32
180:IF A(J-1)=45OR A(J-1)=46OR A(J-1)=32AND A(J)>91THEN LET A(J)=A(
   J)-32
185:NEXT J
190:FOR J=0TO I:A$(0)=LEFT$ (A$(0),J)+CHR$ A(J):NEXT J
195:PRINT A$(0):END
200:PRINT "Eingabefehler":END

```

Martin Strösser
Parkstr.61
4018 Langenfeld 5

Programm: Funktionsdarstellung
Rechner : PC-2500/1350/60/1600
Laenge : 508 Bytes

FUNKTIONSDARSTELLUNG PC-2500/1350/60/1600



```

23:CURSOR 16,3: PRINT "Ymax="; LEFT$ (
   STR$ YX,3)
24:LINE (95,0)-(95,31): LINE (95,0)-(9
   0,31)
30:IF SGN XN(> SGN XX LINE ( ABS XN/DX
   ,0)-( ABS XN/DX,31): GOTO 40
35:LINE (( SGN XN=-1)*95,0)-(( SGN XN=
   -1)*95,31),43690
40:IF SGN YN(> SGN YX LINE (0,YX/DY-1)
   -(95,YX/DY-1); GOTO 50
45:LINE (0,( SGN YN=1)*31)-(95,( SGN Y
   N=1)*31),43690
50:FOR X=XN TO XX STEP DX
55:RADIAN :Y= SIN X+.5* COS (3*X)
60:XX=(X-XN)/DX:YY=(YX-Y)/DY-1: IF YY<
   0 OR YY>31 THEN 70
65:PSET (XX,YY)
70:NEXT X
80:WAIT : GPRINT

```

Dieses Programm ermöglicht es, den Graphen einer beliebigen Funktion in der Anzeige des Rechners darzustellen. Dazu ist zuerst in Zeile 55 die Funktionsgleichung (y=f(x)) einzugeben. Nach Start des Programms werden die Bereichsgrenzen abgefragt, woraufhin der Computer automatisch die Achsen festlegt und den Graphen zeichnet.

Viel Spass !!!!

```

10:CLEAR :CLS : WAIT 0
20:INPUT "Xmin ";XN,"Xmax ";XX,"Ymin "
   ;YN,"Ymax ";YX:X1=XX-XN:Y1=YX-YN:
   CLS :DX=X1/96
21:DY=Y1/32
22:CURSOR 16,0: PRINT "Xmin="; LEFT$ (
   STR$ XN,3): CURSOR 16,1: PRINT "Xma
   x="; LEFT$ ( STR$ XX,3): CURSOR 16,
   2: PRINT "Ymin="; LEFT$ ( STR$ YN,3
   )

```

Peter Schultz
 Alter Markt 5
 5238 Hachenburg

MAPRO

MAPRO
 MAPRO

Seite -A-	Seite -B-
<pre> 1 REMOVED 0 2 Fighting the world 3 MURDER 2, MALTEEN 0 4 "You don't remember, I'll never forget" 5 PRETTY HELPS 0 6 no one to rock 7 DID 0 8 don't talk to strangers 9 IRON MAIDEN 0 10 100 - the number of the beast 11 HEART 0 12 alone 13 GUNNEX 0 14 it's not love 15 METALLICA 0 16 she think that should not be 17 WHITE LIGHT 0 18 lady of the valley 19 CRISPEN BURY 0 20 walhalla </pre>	<pre> 1 IRON MAIDEN 0 2 run to the hills 3 METALLICA 0 4 you know the hell tolls 5 HEART 0 6 bad ominals 7 MURDER MALTEEN 0 8 fire 9 CINDERELLA 0 10 what do 11 SAMPY WOODS 0 12 when the roman falls 13 UNLEASHABLE 0 14 crying in the rain 15 REMOVED 0 16 who's on 17 ACCEPT 0 18 matinee and nite, feet as a shark 19 SAMPY WOODS 0 20 "stamen" at the sea old crossroads </pre>

>>> HEAVY METAL <<<
 HEAVY METAL

(c) Peter Schultz, Hachenburg

```

100:"C"REM CASSETTENHÜLLENBESCHRIFTUNG
110: CLEAR :LOCK
120: DIM A*(19)*38, B*(19)*38, C*(1)*39
130: FOR I=0TO 19:CLS :PRINT "SEITE A";I+1;" ZEILE"; " 38ZEICHEN";:CURSOR 13,2:
PRINT ">";
140: INPUT A*(I):NEXT I
150: FOR I=0TO 19:CLS :PRINT "SEITE B";I+1;" ZEILE"; " 38ZEICHEN";:CURSOR 13,2:
PRINT ">";
160: INPUT B*(I):NEXT I
170: CLS :PRINT "KASSETTENMITTE , 1. ZEILE(39ZEICHEN)";:CURSOR 12,2:PRINT ">";
:INPUT C*(0)
180: CLS :PRINT "KASSETTENMITTE , 2. ZEILE(39ZEICHEN)";:CURSOR 12,2:PRINT ">";
:INPUT C*(1)
190:"A"CLS :GRAPH :GLCURSOR (250,-100):SORGN
200:LLINE (0,0)-(495,-465),6,2,B
210:LLINE (10,-10)-(487,-310),0,3,B
220:LLINE (10,-330)-(487,-370),,,B
230:LLINE (0,-320)-(10,-320):LLINE (0,-380)-(10,-380)
240:LLINE (250,-310)-(250,-10),3
250:LLINE (487,-35)-(10,-35)
260:COLOR 1:GLCURSOR (78,-30):LPRINT "Seite -A-"
270:GLCURSOR (321,-30):LPRINT "Seite -B-"
280:COLOR 0:CSIZE 1
290:FOR I=0TO 19
300:GLCURSOR (15,-50-13*I):LPRINT A*(I):GLCURSOR (255,-50-13*I):LPRINT B*(I):
NEXT I
310:CSIZE 2:GLCURSOR (15,-347):LPRINT C*(0)
320:GLCURSOR (15,-367):LPRINT C*(1)
330:LLINE (495,-320)-(485,-320),0,3:LLINE (495,-380)-(485,-380)
340:ROTATE 2:GLCURSOR (480,-445):LPRINT "(c) Peter Schultz, Hachenburg"
350:GLCURSOR (0,-600):TEXT :UNLOCK
360:END
  
```

Vier farbiger Ausdruck. Dann falten (kleine Striche!) und entlang der äusseren Linie ausschneiden. Start mit DEF C Erneuter Ausdruck mit DEF A. pro Seite stehen 20 Zeilen zur Verfügung. In der Mitte sind es zwei.

Programmbeschreibung MAPRO-UTILITY:
 Dieses Programm dient zur Eingabe sowie zum Verschieben von MaPros.
 Die ersten beiden Unterprogramme sind schon in dieser oder ähnlicher Form in vorherigen Ausgaben von AfSC erschienen.
 Unterprogramm "CHANGE ADRESS" dient zum Verändern von Speicheradressen. Das heisst MaPros können mehr oder weniger frei im RAM verschoben werden. Durch die Verwendung einer ML-Routine erfolgt das Verschieben sehr schnell (ca. 4kByte pro Sekunde).
 Die Bedienung des Programms ist so einfach, dass sie keiner weiteren Erklärung bedarf.

Dirk Mahrholz
 Goethestr. 86
 2850 Bremerhaven

```

5:"M"CLS :CLEAR :WAIT 0:BEEP ON
10:A=256:CALL &020D,A
15:POKE &EF00,&F5,&C5,&D5,&E5,&2A,&16,&EF,&E0,&5B,&18,&EF,&ED,&4B,&1A,&EF,&ED
20:POKE &EF10,&80,&E1,&D1,&C1,&F1,&C9
25:GCURSOR 0,0:GPRINT "7F7F060C0C067F7F007E7F1313137F7E007F7F1B1B1B1B1F0E";
30:GPRINT "007F7F1B1B1B3B7F4E03E7F630363637F3E000808080808003F7F60606060";
35:GPRINT "7F3F000303037F7F030303007F7F007F7F606060606060007F7F000303037F7F";
40:GPRINT "030303000103067C7C060301"
45:CURSOR 0,1:PRINT "1: Hex-Load I 4: Ende"
50:PRINT "2: Hex-Load II":PRINT "3: Change Adress"
55:GOSUB 500:K=UAL Ks:IF K<10R K>4GOTO 55
60:ON KGOTO 70,165,225,65
65:CLS :KBUFF$="":END
70:CLS :CLEAR :CURSOR 7,0:PRINT "HEX-LOAD I"
75:CURSOR 0,2:KBUFF$="":INPUT "Erste Adresse : ";II:I=II
80:IF II<&C0C50R II>&F000GOTO 70
85:B=PEEK I:A=INT (B/16):B=B-A*16
90:IF A>9LET A=A+7
95:IF B>9LET B=B+7
  
```

MAPPRO-UTILITY
 1: Hex-Load I
 2: Hex-Load II
 3: Change Address
 4: Ende

```

100:As=CHR$(A+48)-CHR$(B+48):IN=INT I:Ns="":Hs="0123456789ABCDEF"
105:M=I:I=INT(I/16):M=M-I*16+1:Ns=MID$(Hs,M,1)+Ns:IF I>0GOTO 105
110:CURSOR 0,2:PRINT CHR$ 127;"Adresse: ";Ns;" (";As;")";CHR$ 127;"&"
115:CURSOR 25,2:PRINT CHR$ 127;CURSOR 22,2:INPUT B$:GOTO 125
120:GOTO 160
125:CURSOR 22,2:PRINT " ":IF B$="*"CLS:GOTO 25
130:IF B$="-"LET II=II-1:I=II:GOTO 85
135:IF B$="+"LET II=II+1:I=II:GOTO 85
140:A=ASC LEFT$(B$,1)-48:B=ASC RIGHT$(B$,1)-48
145:IF A>9LET A=A-7
150:IF B>9LET B=B-7
155:POKE II,A*16+B
160:II=II+1:I=II:GOTO 85
165:CLS:CLER:DIM A$(0)*80
170:KBUFF$="":PRINT "HEX-LOAD II":INPUT "Anzahl der Bytes : ";I
175:IF I<1OR I>255GOTO 165
180:INPUT "Anfangsadresse : ";A
185:IF A<&C0C5OR A>&F000OR &F000-A<0GOTO 165
190:Q=Q*2:G=INT(Q/80)
195:CLS:CURSOR 0,0:FOR J=0TO G:INPUT "DATA : ";A$(0):N=LEN A$(0):N=N-1
200:FOR I=1TO NSTEP 2:U$=MID$(A$(0),I,2):GOSUB 210
205:NEXT I:NEXT J:CLS:BEEP 3:CLS:GOTO 25
210:C$=LEFT$(U$,1):F=ASC C$:F=F-48:IF F>9LET F=F-7
215:X=F:X=X*16:D$=RIGHT$(U$,1):E=ASC D$:E=E-48:IF E>9LET E=E-7
220:Z=E:Y=X+Z:POKE A,Y:A=A+1:RETURN
225:CLER:A=&C0C5
230:CLS:KBUFF$="":PRINT "CHANGE ADRESS":INPUT "Anfangsadresse : ";A
235:INPUT "Endadresse : ";B:INPUT "Zieladresse : ";C
240:D=1+B-A:IF A>BOR B>&F000GOTO 225
245:G=A:H=C:POKE &EF10,&B0
250:IF A<CLET G=B:H=C+D-1:POKE &EF10,&B8
255:E=G/256:HG=INT E:LG=(E-HG)*256:E=H/256:HH=INT E:LH=(E-HH)*256
260:E=D/256:HD=INT E:LD=(E-HD)*256:POKE &EF16,LG,HG,LH,HH,LD,HD:CALL &EF00
265:CLS:BEEP 3:GOTO 25
500:K$=INKEY$:IF K$=""GOTO 500ELSE RETURN
  
```

Martin Strösser
 Parkstr. 61
 4010 Langenfeld 5

Programm: Diagramme
 Rechner: PC-2500/1350/60/1600
 Laenge: 3651 Bytes

Das Programm dient zur Erstellung von Diagrammen, wobei unter 4 Arten gewählt werden kann (Balken-, Linien-, Streifen- und Kuchendiagramm). Es können positive Daten jeglicher Art eingegeben werden; die Datenkapazität wird einzeln angegeben. Dieses Programm entstand in Anlehnung an die eingebaute Businesssoftware des PC-2500, welcher die Diagramme jedoch leider nur ueber den Plotten ausgeben kann.

Viel Spass !!!!

```

10:CLS:CLER:WAIT 0
20:PRINT "Balkendiagramm" B Linien-
    enndiagramm L Streifenndiagramm
    S
21:PRINT "Kuchendiagramm" K
30:IF INKEY$=""GOTO 30
31:IF INKEY$="B" GOTO 100
32:IF INKEY$="L" GOTO 200
33:IF INKEY$="S" GOTO 300
34:IF INKEY$="K" GOTO 400
35:GOTO 30
100:CLS:INPUT "Anzahl Spalten ";S:
    INPUT "Anzahl Reihen ";R:IF (24S+2
    )>R)138 GOTO 120
101:DIM R(R,S),Y$(2),Y(2):CLS
  
```

```

105:FOR I=1 TO S:FOR J=1 TO R
110:PRINT I;" Spalte/ ";J;" Reihe: ";
    INPUT R(J,I):IF R(J,I)>RM LET RM=R
    (J,I)
115:NEXT J:NEXT I:CLS
120:DY=24/RM:LINE (9,0)-(9,31):LINE (
    0,24)-(149,24)
130:RESTORE 190:FOR I=1 TO R:FOR J=1
    TO S
140:LINE (9+2*(I+J)+2*S*(I-1),24)-(9+2*
    (I+J)+2*S*(I-1),24-R(I,J)*DY):NEXT
    J
150:READ Z$:GOCURSOR (11+2*S*I+2*(I-5),
    31):GPRINT Z$:NEXT I
155:D=INT(9/DY)+(9/DY-INT(9/DY)).5
160:Y(1)=INT(13/DY)+(13/DY-INT(13/D
    Y)).5:Y(2)=Y(1)-D:Y(2)=Y(1)+D
165:PSET (9,20):PSET (8,11):PSET (8,2
    )
170:FOR I=0 TO 2:ON LEN STR$(Y(I))
    GOSUB 175,174:GOTO 180
174:FOR J=1 TO 2:RESTORE 199:FOR E=1
    TO VAL MID$(STR$(Y(I)),J,1)+1:
    READ Y$(J):NEXT E:NEXT J:RETURN
175:RESTORE 198:FOR E=1 TO Y(1)+1:
    READ Y$(1):NEXT E:Y$(2)="" :RETURN
180:GOCURSOR (0,22-1*I):GPRINT Y$(1);"0
    ";Y$(2):NEXT I:WAIT:GPRINT:
    END
190:DATA "91F9B1","E9A9B9","A9A9F9","39
    21F9","B9A9E9","F9A9E9","0909F9","F
    9A9F9"
191:DATA "B9A9F9","F989F9","F929F9","F9
    A9F9","F989B9","F98971","F9A9A9","F
    92909"
192:DATA "F989E9","F921F9","89F9B9","41
    8179","F96191","F98181","F911F9","F
    909F9"
193:DATA "718971","F92939","7989F9","F9
    6399","B9A9E9","09F909","F981F9","7
    38179"
  
```

```

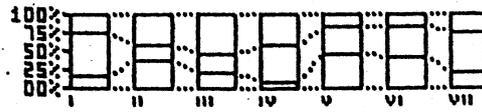
194: DATA "F941F9", "D921D9"
198: DATA "F88BF8", "92F880", "E8A883", "A8
    A8F8", "3822F8", "88A8E8", "F8A8E8", "0
    822F8"
199: DATA "F8A8F8", "89A8F8"
200: CLS : INPUT "Anzahl Spalten ( < 5 ) .
    " : S: INPUT "Anzahl Reihen ( < 24 ) "
    : R: IF R>23 OR S>4 GOTO 200
201: DIM G(R), R(R+1, S), Y(3), Y*(3): CLS
205: FOR I=1 TO S: FOR J=1 TO R
210: PRINT I: " Spalte / " ; J ; " Reihe: " :
    INPUT R(J, I): IF R(J, I)>RM LET RM=R
    (J, I)
215: NEXT J: NEXT I: CLS
220: DY=24/RM: LINE (9, 0)-(9, 31): LINE (
    0, 24)-(149, 24)
230: FOR I=1 TO S: RESTORE 190: Y1=24-R(1
    , I)*DY: FOR J=2 TO R
240: LINE ((J-2)*6+11, Y1)-((J-1)*6+11, 24
    -R(J, I)*DY): Y1=24-R(J, I)*DY
250: READ Z$: GCURSOR (11+6*(J-2), 31):
    GPRINT Z$: NEXT J: NEXT I: READ Z$:
    SCURSOR (11+6*(R-1), 31): GPRINT Z$
254: FOR I=1 TO R: PSET (6+I*6, 25): NEXT
    I
255: D= INT (9/DY)+(9/DY- INT (9/DY)).5)
260: Y(1)= INT (13/DY)+(13/DY- INT (13/D
    Y)).5): Y(0)=Y(1)-D: Y(2)=Y(1)+D
265: PSET (8, 20): PSET (8, 11): PSET (8, 2
    )
270: FOR I=0 TO 2: ON LEN STR$ Y(I)
    GOSUB 175, 174: GOTO 280
274: FOR J=1 TO 2: RESTORE 198: FOR E=1
    TO UAL MID$ ( STR$ Y(I), J, 1)+1:
    READ Y*(J): NEXT E: NEXT J: RETURN
275: RESTORE 198: FOR E=1 TO Y(I)+1:
    READ Y*(1): NEXT E: Y*(2)="": RETURN
280: GCURSOR (0, 22-1*9): GPRINT Y*(1): "0
    0": Y*(2): NEXT I: WAIT : GPRINT :
    END
300: CLS : INPUT "Anzahl Spalten ( < 8 )
    " : S: INPUT "Anzahl Reihen ( < 6 ) " :
    R: IF S>7 OR R>5 GOTO 300
301: DIM Z(R, S), G(S), R(R, S), Y(3), Y*(3):
    CLS
305: FOR I=1 TO S: FOR J=1 TO R
310: PRINT I: " Spalte / " ; J ; " Reihe: " :
    INPUT R(J, I): Z(J, I)=Z(J-1, I)+R(J, I)
    : G(I)=G(I)+R(J, I)
315: NEXT J: NEXT I: CLS
316: RESTORE 390: FOR I=1 TO S: C=-2+20*I
    : READ A: FOR J=1 TO LEN STR$ A: IF
    MID$ ( STR$ A, J, 1)="1" GCURSOR (C, 3
    1): GPRINT "E000": C=C+2
317: IF MID$ ( STR$ A, J, 1)="2" GCURSOR (
    C, 31): GPRINT "60806000": C=C+4:
    GOTO 319
319: IF MID$ ( STR$ A, J, 1)="3" GCURSOR (
    C, 31): GPRINT "A040A000": C=C+4
319: NEXT J: NEXT I: LINE (18, 2)-(18, 26)
    : FOR J=2 TO 26 STEP 24
320: FOR I=19 TO 149 STEP 20: LINE (1, J)
    -(1+10, J): LINE (1+11, J)-(1+19, J), &
    AAAA: NEXT I: NEXT J
321: FOR I=28 TO 4 STEP -6: GCURSOR (12,
    I): GPRINT "482090": NEXT I
322: FOR I=5 TO 1 STEP -1: GCURSOR (8, 6+
    I-2): IF I/2= INT (I/2) GPRINT "88A
    8E8": NEXT I: GOTO 324
323: GCURSOR (8, 6+I-2): GPRINT "F899F8":
    NEXT I

```

```

410: PRINT I: " Reihe: " : INPUT R(I): Z(I)
    =Z(I-1)+R(I): G=G+R(I)
415: NEXT I: CLS
416: CURSOR 8: PRINT "Anteil (%)":
    RESTORE 490: FOR I=1 TO R: READ A:
    CURSOR A
417: PRINT CHR$( 64+I): " " : STR$( INT (
    R(I)/G*120)+(R(I)/G*120- INT (R(I)/
    G*120)).5): NEXT I
418: CURSOR 55: PRINT "A"
419: LINE (49, 0)-(114, 7), X, EF: LINE (47,
    15)-(53, 23), X, EF
420: DEGREE : FOR I=0 TO 340 STEP 20:
    LINE (14* SIN I+25, 1+* COS I+14)-(1
    4* SIN (I+20)+25, 1+* COS (I+20)+14)
    : NEXT I
420: DY=350/G: LINE (25, 1)-(25, 29):
    LINE (29, 30)-(50, 30): PSET (30, 23):
    PSET (30, 31)
431: LINE (50, 30)-(50, 22): LINE (41, 16)-
    (41, 22): PSET (42, 17): PSET (42, 17)
    : LINE (41, 22)-(37, 26)
440: FOR I=1 TO R-1: LINE (25, 15)-(25, 1+
    * SIN (Z(I)*DY), 15+1+* COS (Z(I)*DY
    )): NEXT I: WAIT : GPRINT : END
400: DATA 33, 57, 81, 39, 62, 86, 13, 43, 67, 31
324: GCURSOR (4, 28): GPRINT "F899F8":
    GCURSOR (4, 22): GPRINT "E8A889":
    GCURSOR (4, 15): GPRINT "88A8E8":
    GCURSOR (4, 10): GPRINT "0808F9"
325: GCURSOR (4, 4): GPRINT "F899F8":
    GCURSOR (0, 4): GPRINT "92F880": FOR
    I=2 TO 26 STEP 6: PSET (17, I): PSET
    (16, I): NEXT I
330: FOR I=1 TO R-1: Y1=2+Z(I, 1)*24/G(I):
    FOR J=2 TO S: DY=24/G(J)
335: LINE ((J-2)*20+19, Y1)-((J-1, 4)*20+1
    9, Y1)
340: LINE ((J-1, 4)*20+19, Y1)-((J-1)*20+1
    9, Z(I, J)*DY+2), %AAAA: Y1=Z(I, J)*DY+2
    : NEXT J
341: LINE ((J-2)*20+19, Y1)-((J-1, 4)*20+1
    9, Y1): NEXT I
350: FOR I=1 TO S-1: LINE (19+20*I, 3)-(1
    9+20*I, 26): LINE (12+20*I, 3)-(12+20
    4, 26): NEXT I
355: LINE (12+20*S, 2)-(12+20*S, 25): IF S
    =7 LINE (149, 2)-(149, 26)
360: WAIT : GPRINT : END
390: DATA 1, 11, 111, 12, 2, 21, 211, 2111, 13, 3
    , 31, 311, 3111
400: CLS : INPUT "Anzahl Reihen ( < 11 )
    " : R: IF R>10 GOTO 320
401: DIM Z(R), R(R): CLS
405: FOR I=1 TO R

```



Betr.: Neues Produkt

PC-1600 Speichererweiterung

Sehr geehrte Damen und Herren,

für den Nachfolger des legendären SHARP PC-1500,

den SHARP PC-1600,

bieten wir Steckmodule in den Größen 32 K, 64 K, 128 K und 256 KByte an.

SHARP selbst hat nur 32 und 64 KByte Module.

Diese interessante Ergänzung ermöglicht dem SHARP PC-1600-Anwender einen maximalen Speicherausbau auf

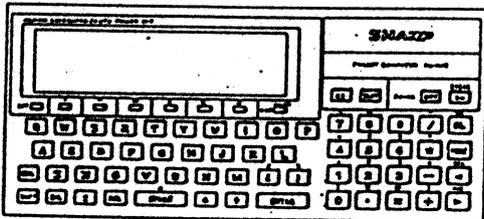
304 KByte.

Die Steckmodule sind einfach zu handhaben und werden vom SHARP PC-1600 voll verwaltet.

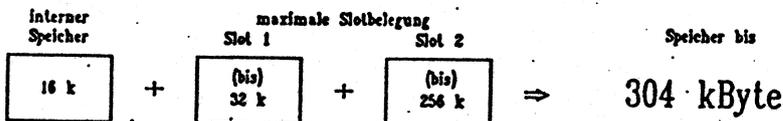
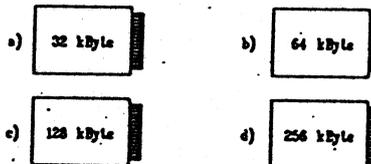
Hierdurch wird aus dem handlichen PocketComputer ein portabler Computer mit überdurchschnittlich hohen Speicherreserven.

Speichergigant

SHARP PC-1600



Speichermodule für PC-1600 im Kunststoffgehäuse



Nutzung der Module

Module	32 k	64 k	128 k	256 k
Slot	1 + 2	2	2	2
Speichererweiterung: BASIC Programm	x	32 k	32 k	32 k
RAM-Disk	x	x	x	x
32 k abzweigbar für BASIC-RAM		x	x	x

empfohlene Verkaufspreise incl. 14 % MWSt.

- | | | | |
|----|---------------|---|-----------|
| a) | 32 kByte RAM | batteriegepuffert (entspricht CE-1600M) | DM 209,-- |
| b) | 64 kByte RAM | batteriegepuffert (entspricht CE-1601M) | DM 299,-- |
| c) | 128 kByte RAM | batteriegepuffert (exklusiv bei uns) | DM 549,-- |
| d) | 256 kByte RAM | batteriegepuffert (exklusiv bei uns) | DM 899,-- |

Die Speichererweiterungen erhalten Sie bei uns:

DIVER DOWN

Ein Tauchsport-Programm für den SHARP PC-1600

Gernot Hermenau
Pelmannstraße 36
4300 Essen 1

Hallo, Freunde des Tauchsports!

Wer von Ihnen kennt nicht das Problem der Tauchzeitberechnung, welches jedem von uns schon zum Taschenrechner greifen lieb? Das muß nicht mehr sein. Tippen Sie deshalb schnell das folgende Programm in Ihren SPC, und schon sind Sie die Arbeit los.

Nun zum Programm. Nach dem Start mit RUN oder DEF A muß als erstes das Flaschenvolumen eingegeben werden. Benutzen Sie eine Doppelflasche, so addieren Sie einfach die beiden Flascheninhalte. Quittieren Sie die Eingabe und alle später folgenden mit ENTER. Danach geben Sie den Flaschendruck an. Dieser liegt normalerweise im Bereich 200 - 250 bar. Sie können aber auch höhere oder niedrigere bar-Eingaben machen. Dann müssen Sie die gewünschte Tauchtiefe festlegen. Der Grenzwert liegt hier bei 63 m. Höhere Werte nimmt der Computer nicht an. Nach einem Kontrolldurchgang mit der Aufforderung zu einer eventuellen Fehlerbeseitigung beginnt der Computer mit der Berechnung und Ausgabe. Es kann allerdings vorkommen, daß eine gewünschte Tauchtiefe nicht erreicht werden kann, da entweder die Flasche zu klein oder der Druck nicht ausreichend ist. Sollte die mit dem Luftvorrat errechnete Tauchzeit über den Nullzeit liegen, warnt der Computer mit DEKOMPRESSION. Nun müssen Sie entweder den Tauchgang auf die angegebene Nullzeit beschränken oder mit Hilfe einer Auftauchtabelle Deko-Stops in bestimmten Tiefen einhalten.

Im Programm wird von dem Luftvorrat eine 10%ige Sicherheitsreserve abgezogen (Zeile 56). Diese ist nicht zu verwechseln mit der Reserveschaltung. Ebenfalls geht das Programm von einem Luftverbrauch von 30 l/Min. aus (zu ändern in Zeile 60). Die bei der Berechnung berücksichtigte Aufstiegsgeschwindigkeit von 10 m/Min. sollte nicht überschritten werden. Wie bereits erwähnt, beinhaltet das Programm auch die Angaben der Nullzeiten, die es einem erlauben, aus einer bestimmten Tiefe gefahrlos aufzutauchen. Bei dessen Gebrauch sollte allerdings zusätzlich ein Sicherheitsstopp von 3 Min. in 3 m Tiefe eingehalten werden.

ACHTUNG: Dieses Programm ersetzt nicht die Teilnahme an einem Tauchkurs oder die Benutzung einer Deko-Tabelle. Es soll dem Taucher lediglich eine kurze Übersicht über die ihm zur Verfügung stehende Tauchzeit geben. Das Programm und die Angaben der Nullzeiten sind nur im Bereich 0 - 250 m. ü. N.N. gültig.

Das Programm habe ich auf einem PC-1600 (Mode 1) geschrieben; aber so angelegt, daß es auch auf allen anderen SPCs laufen dürfte. Es steht jedem Taucher frei, das Programm entsprechend zu erweitern. Zum besseren Verständnis habe ich eine Liste aller gebrauchten Variablen und deren Bedeutung beigelegt.

- 2: GERNOT HERMENA U
- 3: PELMANSTR. 36
- 4: 4300 ESSEN 1
- 6:

```

20:"A"CLS: CLEAR: WAIT 100: PRINT " * DIVER-DOWN * "
30: PRINT "(C) 1989 BY: PRINT " MOVIE-SOFT": CLS
40:"B" CLEAR: INPUT "FLASCHENVOL. ?": A: INPUT "FLASCHENDRUCK ?": B
50: INPUT "TAUCHTIEFE ?": C: IF C > 63 THEN 120
56: D=A*B: E=D-(D*10/100)
60: F=C/10+1: G=30: H=F*G: PRINT "KONTROLLE": CLS
70: PRINT "FLASCHENVOL. ": CLS: PRINT A: " L": CLS
80: PRINT "FLASCHENDRUCK ": CLS: PRINT B: " BAR": CLS
90: PRINT "WASSERTIEFE ": CLS: PRINT C: " M"
100: INPUT "AUFRÜHUNG(J/N)?": Z$
102: IF Z$="" THEN 100
104: IF Z$="J" THEN 40
106: IF Z$="N" THEN 110
110: I=E/H: J=C/2/10+1: K=G*J: L=C/10: M=K*L: N=E-M: O=N/H
112: IF O > 0 THEN 140
120: FOR P=1 TO 4: WAIT 20: PRINT " * * ZU TIEF * * ": PRINT "": NEXT P
130: GOTO 200
140: WAIT 100: USING "####.##": CLS: PRINT O: " MIN"
150: PRINT " TAUCHLAENGE": USING : PRINT " IN "; C: " M"
162: FOR S=1 TO 19
163: READ T, U
164: IF T >= C THEN 170
165: NEXT S
    
```

Tabelle für Oberflächenpausen
und Wiederholungstauchgänge

		Oberflächenpause (h.min.)														
Wiederholungs- erwäge	G	0	25	0	45	1	00	1	15	1	40	2	10	2	00	
	F		0	20	0	30	0	45	1	15	1	30	0	45	0	00
	E			0	10	0	15	0	25	0	25	0	25	0	40	00
	D				0	10	0	15	0	30	0	30	0	30	0	00
	C					0	10	0	25	0	30	0	30	0	00	
B						0	20	0	20	0	20	0	20	0	00	

Tiefe des Wiederholungstauchgangs (m)	Zeltzuschlag zur Grundzeit (min)														
	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
9	30	5	5	5	5	5	5	5	5	5	5	5	5	5	5
12	11	8	8	8	8	8	8	8	8	8	8	8	8	8	8
15	8	6	6	6	6	6	6	6	6	6	6	6	6	6	6
18	6	4	4	4	4	4	4	4	4	4	4	4	4	4	4
21	4	3	3	3	3	3	3	3	3	3	3	3	3	3	3
22	4	3	3	3	3	3	3	3	3	3	3	3	3	3	3
27	3	2	2	2	2	2	2	2	2	2	2	2	2	2	2
30	3	2	2	2	2	2	2	2	2	2	2	2	2	2	2
33	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
36	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1
39	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1
42	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
45	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
48	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
51	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
54	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
57	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
60	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
63	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Tiefen (m)	Nullzeit (min)	Grenztiefe (m)	Sicherheitsstopp				Wiederholung
			G	J	N	C	
9	20	120				B	
63	40	120				C	
	60	120				D	
	80	120				E	
	100	120				F	
	120	120				G	
12	30	192				B	
	45	192				C	
	60	192				D	
	75	192				E	
	90	192				F	
	105	192				G	
15	30	99				C	
	45	99				D	
	60	99				E	
	75	99				F	
	90	99				G	
18	30	65				B	
	40	65				C	
	50	65				D	
	60	65				E	
	70	65				F	
	80	65				G	
	90	65				G	
	100	65				G	
	110	65				G	
	120	65				G	
21	30	21				B	
	40	21				C	
	50	21				D	
	60	21				E	
	70	21				F	
	80	21				G	
	90	21				G	
	100	21				G	
	110	21				G	
	120	21				G	
24	30	24				C	
	40	24				D	
	50	24				E	
	60	24				F	
	70	24				G	
	80	24				G	
	90	24				G	
	100	24				G	
	110	24				G	
	120	24				G	
27	30	27				C	
	40	27				D	
	50	27				E	
	60	27				F	
	70	27				G	
	80	27				G	
	90	27				G	
	100	27				G	
	110	27				G	
	120	27				G	

Tiefen (m)	Nullzeit (min)	Grenztiefe (m)	Sicherheitsstopp				Wiederholung
			G	J	N	C	
9	10	10				C	
	20	10				D	
	30	10				E	
	40	10				F	
	50	10				G	
	60	10				G	
17	35	17				C	
	45	17				D	
	55	17				E	
	65	17				F	
	75	17				G	
	85	17				G	
	95	17				G	
	105	17				G	
	115	17				G	
	125	17				G	
30	30	30				C	
	40	30				D	
	50	30				E	
	60	30				F	
	70	30				G	
	80	30				G	
	90	30				G	
	100	30				G	
	110	30				G	
	120	30				G	
33	20	33				B	
	30	33				C	
	40	33				D	
	50	33				E	
	60	33				F	
	70	33				G	
	80	33				G	
	90	33				G	
	100	33				G	
	110	33				G	
	120	33				G	
36	20	36				B	
	30	36				C	
	40	36				D	
	50	36				E	
	60	36				F	
	70	36				G	
	80	36				G	
	90	36				G	
	100	36				G	
	110	36				G	
	120	36				G	

Austauchtabelle
Bühlmann/Hahn
0-250m ü. N.N.

```

170:RESTORE :PRINT " NULLZEIT ":"PRINT U;" MIN"
172:IF 0<=THEN 200
180:FOR U=1TO 4:WAIT 20:PRINT "*DEKOMPRESSION*"
190:PRINT """:WAIT 100:NEXT U:PRINT "ODER TAUCHZEIT":PRINT "KUERZEN !"
200:PRINT """:WAIT 50:PRINT "NEUE ":WAIT 100
210:INPUT "RECHNUNG(J/N)?";R$:IF R$=""THEN 200
220:IF R$="J"THEN 40
230:CLS :END
240:DATA 9,653
250:DATA 12,192
260:DATA 15,99
270:DATA 18,65
280:DATA 21,40
290:DATA 24,27
300:DATA 27,21
310:DATA 30,17
320:DATA 33,15
330:DATA 35,9
340:DATA 39,7
350:DATA 42,6
360:DATA 45,5
370:DATA 48,4
380:DATA 51,4
390:DATA 54,3
400:DATA 57,3
410:DATA 60,3
420:DATA 63,2

```

- A= Flaschenvolumen
- B= Flaschendruck
- C= Tauchtiefe
- D= Atemluft in l
- E= Atemluft in l abzüglich 10%
- F= Druck in Tiefe C
- G= Normalluftverbrauch in l/min
- H= Luftverbrauch in l/min in Tiefe C
- I= Atemzeit in Tiefe C
- J= Druck in C/2
- K= Luftverbrauch in l/min in C/2
- L= Aufstiegszeit aus Tiefe C in min
- M= Luftverbrauch für Aufstieg aus Tiefe C
- N= Luftvorrat für Tiefe C ohne Aufstieg
- O= Tauchlänge für Tiefe C in min

Dirk Mahrholz
Goetbestr. 86
2850 Bremerhaven



```

POKER II
Anfangsadresse : &8008
Endadresse      : &80C4
Bank-Nummer    : 1

```

```

5:CLS :CLEAR :WAIT 0:BEEP ON
10:DIM B$(0)*32,C$(0)*32:I=200
15:GOCURSOR 0,0:GPRINT "7F7F1B1B1B1F0E003E7F6363637F3E007F7F081C366341";
20:GPRINT "007F7F6B868686363007F7F101B387F4E000063637F7F63637F7F6363"
25:CURSOR 0,1:INPUT "Anfangsadresse : ";A:CURSOR 0,2:INPUT "Endadresse
";E
30:IF A<0OR A>65534OR E<=AOR E>65536CLS :GOTO 15.
35:CURSOR 0,3:INPUT "Bank-Nummer : ";B
40:IF B<0OR B>7CLS :GOTO 15
45:OUT $50,&3E
50:A$="POKE"&HEX$ A:B$="":B$(0)="" :C$(0)="" :KBUFF$=""
55:FOR J=0TO 7:B$=HEX$ (PEEK$(B,A)):IF LEN B$=1LET B$="0"+B$
60:B$(0)=B$(0)+", "&B$:A=A+1:NEXT J
65:FOR J=0TO 7:B$=HEX$ (PEEK$(B,A)):IF LEN B$=1LET B$="0"+B$
70:C$(0)=C$(0)+", "&B$:A=A+1:NEXT J
75:KBUFF$=CHR$ 31+STR$ I+A$-B$(0)+CHR$ 13+CHR$ 31+"G.85"+CHR$ 13
80:END
85:KBUFF$=CHR$ 31+"L.I"+CHR$ 13+CHR$ 12+CHR$ 3+CHR$ 12+C$(0)+CHR$ 13+CHR$ 31
+"G.95"+CHR$ 13
90:END
95:I=I+5:IF A<EGOTO 50
100:OUT $50,&3F
105:CLS :BEEP 1,100,100:KBUFF$="":END
200:POKE &XXXX,&XX,&XX
737 Bytes

```

Programmbeschreibung POKER II:
Mit diesem Programm ist es möglich, MaPros in BASIC-Programme umzuwandeln.
Das MaPro wird in die Form "200:POKE &XXXX,&XX,&XX..." verwandelt. Nach Ende
des Durchlaufs kann das Konvertierungsprogramm, d.h. die Zeilen 5...105 ge-
löscht werden.
Meiner Meinung nach, ist ein BASIC-Programm leichter handzuhaben, als ein
MaPro.
Vorsicht: Nicht verfügbare Speicherbereiche (z.B.:#1;&C000...&FFFF) dürfen
nicht angesprochen werden, da der Rechner dann abstürzt.

Tauchtiefe (m)	Flaschenvolumen (l)	Flaschendruck (bar)	Tauchtiefe (m)	Flaschenvolumen (l)	Flaschendruck (bar)
39	5	10	54	5	10
7	10	20	3	10	20
	15	30	1	10	30
	20	40	1	10	40
	25	50	1	10	50
	30	60	1	10	60
	35	70	1	10	70
	40	80	1	10	80

Tauchtiefe (m)	Flaschenvolumen (l)	Flaschendruck (bar)	Tauchtiefe (m)	Flaschenvolumen (l)	Flaschendruck (bar)
42	5	10	57	5	10
6	10	20	3	10	20
	15	30	1	10	30
	20	40	1	10	40
	25	50	1	10	50
	30	60	1	10	60
	35	70	1	10	70
	40	80	1	10	80

Tauchtiefe (m)	Flaschenvolumen (l)	Flaschendruck (bar)	Tauchtiefe (m)	Flaschenvolumen (l)	Flaschendruck (bar)
45	5	10	60	5	10
5	10	20	3	10	20
	15	30	1	10	30
	20	40	1	10	40
	25	50	1	10	50
	30	60	1	10	60
	35	70	1	10	70
	40	80	1	10	80

Tauchtiefe (m)	Flaschenvolumen (l)	Flaschendruck (bar)	Tauchtiefe (m)	Flaschenvolumen (l)	Flaschendruck (bar)
48	5	10	63	5	10
4	10	20	2	10	20
	15	30	1	10	30
	20	40	1	10	40
	25	50	1	10	50
	30	60	1	10	60
	35	70	1	10	70
	40	80	1	10	80

Tauchtiefe (m)	Flaschenvolumen (l)	Flaschendruck (bar)	Tauchtiefe (m)	Flaschenvolumen (l)	Flaschendruck (bar)
51	5	10			
4	10	20			
	15	30			
	20	40			
	25	50			
	30	60			
	35	70			
	40	80			

Aufstiegs geschwindigkeit 10 m/min

BEISPIEL:

```

200:POKE &8000,&FF,&FF,&FF,&FF,&FF,&FF,&FF,&FF,&FF,&FF,&FF,&FF,&FF,&FF,&FF,&FF,&FF,&FF,&FF,&FF
205:POKE &8010,&20,&4C,&4F,&41,&20,&53,&41,&50,&20,&46,&4C,&53,&43,&4F,&4D
210:POKE &8020,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00,&00

```

Rechnerkopplung über die serielle Schnittstelle

Alle Sharp-Taschencomputer mit der 15poligen SIO-Schnittstelle erlauben eine sehr einfache Datenverbindung mit IBM-kompatiblen Rechnern. Die Sharp-Befehle SAVE und LOAD übertragen direkt ASCII-Dateien, so daß man ohne weitere Hilfsmittel mit einem beliebigen Texteditor Sharp-Basicprogramme bearbeiten kann.

Das Datenkabel erfordert nur minimalen Hardware-Aufwand. Der Sharp-Rechner kann die RS232 Eingänge (Daten- und Handshake-Leitung) direkt treiben, während in umgekehrter Richtung Zenerdioden als Spannungsschutz eingesetzt werden müssen. Das folgende Übertragungsprogramm ist in GW-Basic geschrieben und umfaßt neben den reinen Übertragungsfunktionen auch eine übersichtliche Dateiverwaltung mit der Möglichkeit, Unterverzeichnisse zu verwenden.

```

10 CLS :KEY OFF
20 PRINT "***** Sharp-IDM Schnittstellenprogramm RS232 *****"
30 PRINT " "
40 PRINT " "
50 PRINT " "
60 PRINT " "
70 PRINT " "
80 PRINT "bitte Laufwerk wählen"
90 L$=INPUT$(1) :IF L$="a" OR L$="b" OR L$="c" GOTO 100 ELSE GOTO 90
100 L$=L$+" "
105 D$=L$+"\." :CHDIR D$
110 FILES L$
120 PRINT:PRINT
130 INPUT "welches Directory (. =zurück zum Haupt-dir).":D$
140 IF D$="" THEN DS=L$: GOTO 210
145 IF D$="" THEN LET DS=L$+"\." : GOTO 160
150 DS=L$+"\."+D$
160 CHDIR DS
170 PRINT " "
180 PRINT " "
190 FILES L$
200 PRINT " "
210 PRINT "Name der Datei ohne Typkennung:"
220 INPUT E$
230 PRINT " "
240 E$=D$+"\."+E$
250 F$=E$+".:s$er
260 ON ERROR GOTO 300
270 FILES F$
280 PRINT:PRINT " "
290 GOTO 310
300 IF ERR=53 THEN PRINT "neue Datei":E$: RESUME 310
305 CLOSE #1,2
310 PRINT " "
320 PRINT "Bitte wählen:"
330 PRINT "1 Datei neu eingeben
340 PRINT "3 Laufwerk wechseln
350 PRINT "5 ASCII-File anlegen
380 INPUT A
390 ON A GOTO 210,130,10,450,1000,1240
400 REM Übergabe mit Dateiname E$
450 KEY ON :END

```

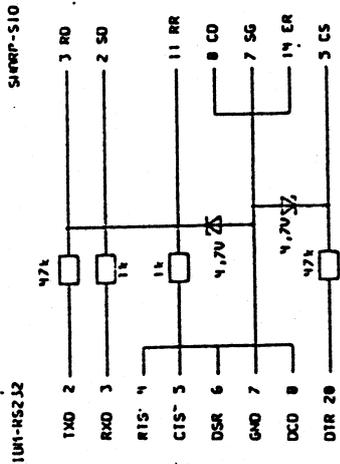
```

990 REM ***** Eingabe ASCII *****
1000 REM ASCII-File anlegen
1080 INPUT "bitte den Dateityp eingeben. z.B. .txt " :CS
1090 IF LEFT$(CS,1)<>"." THEN CS="."+CS
1100 CS=ES+CS
1110 PRINT:PRINT "Es wird die Datei " :CS;" angelegt"
1120 OPEN CS FOR OUTPUT AS #1
1130 OPEN "COM1:1200.N.0.1" FOR INPUT AS #2
1140 LINE INPUT#2, B$
1150 PRINT B$
1160 PRINT#1,B$
1170 IF LOC(2)<2 THEN FOR Y=1 TO 1000:NEXT Y: IF LOC (2)<2 THEN
FOR Y=1 TO 3000:NEXT Y: IF LOC(2)<2 GOTO 1190
1180 GOTO 1140
1190 CLOSE #2
1200 CLOSE #1
1210 INPUT "wiederholen j/n " :AS
1220 IF AS="j" GOTO 1000 ELSE GOTO 310
1230 REM ***** Ausgabe ASCII *****
1240 INPUT "Dateityp. z.B. .pgr " :CS
1255 CS=ES+CS: PRINT "Datei " :CS;" wird ausgegeben.":PRINT
1260 OPEN CS FOR INPUT AS #1
1270 REM OPEN "c:\test.txt" FOR INPUT AS #1
1280 OPEN "COM1:1200.N.0.1.CS 20000.DS 20000" FOR OUTPUT AS #2
1290 LINE INPUT#1, B$
1300 PRINT B$
1310 PRINT#2, B$
1320 IF EOF(1) GOTO 1340
1330 GOTO 1290
1340 CLOSE #1
1350 CLOSE #2
1360 INPUT "wiederholen j/n " :AS
1370 IF AS="j" GOTO 1260 ELSE GOTO 310

```

Beim Start von GW-Basic sollte man einen ausreichend großen Kommunikationspuffer vereinbaren (z.B. GWBASIC /C:32768), um bei langsamen Rechnern sicher einen Datenverlust zu vermeiden. GW-Basic hat nämlich keine automatische Unterstützung des Handshake-Protokolls in Abhängigkeit vom Füllgrad des Textpuffers. Bei 1200 Baud wurden mit diesem Programm aber keine Probleme beobachtet, da der verwendete Rechner mit 8MHz Taktrate ausreißend schnell war.

Ein weiteres Problem der seriellen Textübertragung ist im allgemeinen die Textende-Erkennung. Es wurde hier durch eine Zeitschleife gelöst. Wenn längere Zeit hindurch keine Zeichen mehr eintreffen, gilt die Übertragung als beendet



(Rechnerkopplungs- Schnittstelle)

Ausdehnung von Körpern und Gasen

Dieses Programm ist ohne Änderung lauffähig auf folgenden SHARP Computern: PC-1350, PC-1350, PC-2500 und PC-1600. Programmlänge: 1587 Bytes

Mit dem vorliegenden Programm können Sie die Ausdehnung von Stoffen und Gasen berechnen.

Das Programm erwartet folgende Eingaben: bei Körpern: Ausdehnungskoeffizienten (steht in jedem Physikbuch), die Ausgangs- und Endtemperatur, die Ausgangslänge bzw. das Ausgangsvolumen,

bei Gasen : die Ausgangs- und Endtemperatur und das Ausgangsvolumen.

Ausgegeben wird: die Temperaturdifferenz Delta t in Grad Celsius, die Längen- bzw. Volumendifferenz Delta l bzw. U, der Koeffizient, sowie die aus den eingegebenen Werten resultierenden neuen Längen- bzw. Volumenwerte.

Das Listing:

```

10:"B" CLS : WAIT 0: CURSOR 1,1: PRINT
  "Ausdehnung von Körpern": CURSOR 7,2
  : PRINT "und Gasen"
20:PSET (109,8): PSET (111,8): LINE (9,
  5)-(145,25):B: WAIT 75: LINE (7,3)-(
  147,27):B: WAIT 0
30:CLS : CLEAR : WAIT 0:A$="E888":B$="1
  0F8":A0$="Koeffiz. ":A1$="Delta t
  ="
40:A2$="Delta l ":A3$="Delta U ":A4
  $="Volumen ": USING "#####.#####
  *": GOTO 140
50:CLS : PRINT "Ausgangstemperatur t":
  GCURSOR (125,7): GPRINT B$: CURSOR 2
  1,0: PRINT "=": INPUT T1
60:IF T1<-273.15 GOTO 50
70:RETURN
80:CURSOR 0,2: PRINT "Endtemperatur....
  . t": GCURSOR (125,23): GPRINT A$:
  CURSOR 21,2: PRINT "=": INPUT T2
90:IF T2<-273.15 GOTO 80
100:RETURN
110:INPUT "Ausdehnungskoeffizient=":AK:
  IF AK=0 GOTO 110
120:RETURN
130:INPUT "Ausgangsvolumen.....=":AU:
  RETURN
140:CLS : PRINT "Möchten Sie (K)örper
  oder (G)ase berechnen?": PSET (13,
  0): PSET (15,0)
150:PSET (97,0): PSET (99,0): PRINT "B:
  te k oder G drucken.": PSET (109,16)
  : PSET (111,16)
160:ANS= INKEY$: IF ANS="" GOTO 160
170:IF ANS="K" GOTO 200
180:IF ANS="G" GOTO 340
190:GOTO 160
200:CLS : PRINT "Möchten Sie (L)änge ode
  r (U)volumen wissen?": PSET (13,0)
  : PSET (15,0)
210:PSET (97,0): PSET (99,0): PRINT "B:
  te L oder U drucken.": PSET (109,16)
  : PSET (111,16)
220:ANS= INKEY$: IF ANS="" GOTO 220
230:IF ANS="L" GOTO 260
  
```

```

240:IF ANS="U" GOTO 300
250:GOTO 220
260:CLS : GOSUB 50: GOSUB 80: GOSUB 110
  : INPUT "Ausgangslänge.....=":A
  L
270:IF AL<=0 CLS : GOSUB 430: GOTO 390
280:TD=T2-T1:EL=AL*(AK*TD)
290:CLS : PRINT A0$:AK: PRINT A1$:TD:
  PRINT A2$:EL: WAIT : PRINT "Endlaeng
  e=":AL+EL: WAIT 0: GOTO 380
300:CLS : GOSUB 50: GOSUB 80: GOSUB 110
  : GOSUB 130
310:IF AU<=0 CLS : GOSUB 430: GOTO 390
320:TD=T2-T1:EU=AU*(3*AK*TD)
330:CLS : PRINT A0$:3*AK: PRINT A1$:TD:
  PRINT A3$:EU: WAIT : PRINT A4$:AU+EU
  : WAIT 0: GOTO 380
340:CLS : GOSUB 50: GOSUB 80: GOSUB 130

350:IF AU<=0 CLS : GOSUB 430: GOTO 390
360:TD=T2-T1:EU=AU*((1/273)*TD)
370:CLS : PRINT A1$:TD: PRINT A3$:EU:
  WAIT : PRINT A4$:AU+EU: WAIT 0: GOTO
  380
380:CLS : CURSOR 1,1: PRINT "Neue Rechnu
  ng (J/N) ?"
390:ANS= INKEY$: IF ANS="" GOTO 390
400:IF ANS="J" GOTO 30
410:IF ANS="N" CLS : CLEAR : END
420:GOTO 390
430:WAIT 75: CURSOR 1,1: PRINT "Falsche
  Werteeingabe !": RETURN
  
```

Betrifft: Frage von Holger Hoffmann in Pocket Computer 6/89

FOR-NEXT Schleifen auf dem PC-1600

Beim PC-1600 ist es leider nicht möglich, FOR-NEXT-Schleifen mit einem STEP von kleiner 1 voll funktionsfähig zu programmieren. Der Laufvariablen wird immer ein Integer-Wert (also eine ganze Zahl) zugewiesen. Um diesen (für einen Computer dieser Leistungsklasse doch grossen) Mangel zu umgehen, kann man die Schleife wie folgt programmieren, so dass ein STEP von <1 möglich ist:

FOR-NEXT Schleife

umgeschrieben für PC-1600

```

FOR I=X TO Y STEP 0.25
REM Zeilen zw. Schleife
NEXT I
  
```

```

I-X
'A' REM Zeilen zw. Schleife
I-I+0.25:IF I<=Y THEN 'A'
  
```

Dateinummer mit Variablen ausdrücken

Meines Wissens gibt es keine Möglichkeit, die Dateinummer in einem PRINT# - oder INPUT#-Befehl mit einer Variablen zu bestimmen.

Th. Jeger

CE-1600P - Akkus leer ?

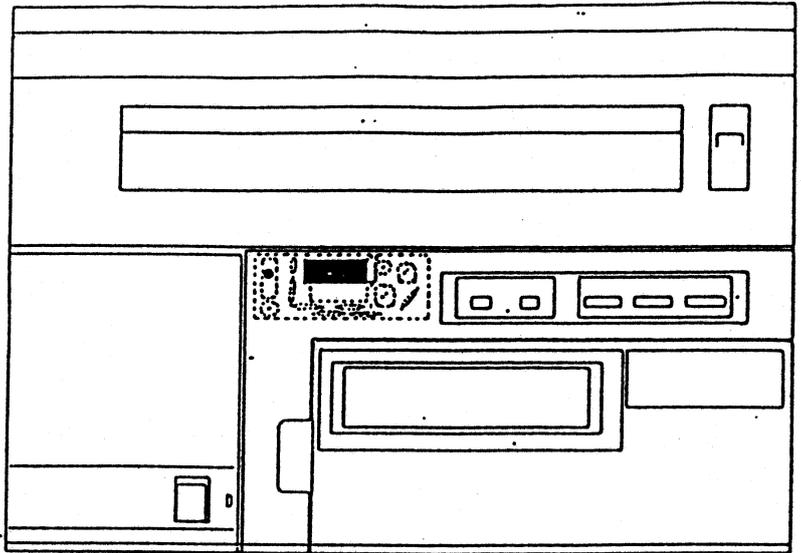
Nun, kein Problem. Netzgerät EA-160 anschließen und schon kann es weitergehen (und gleichzeitig werden dabei auch die Akkus wieder geladen).
 -Oder doch ein Problem, weil gerade kein Netzgerät dabei?
 Nun, um dieses zu vermeiden, gibt es zwei Möglichkeiten: Vor Gebrauch des 1600-Systems die Akkus ordnungsgemäß laden oder mittels Software den Ladezustand überprüfen (siehe Masch.spr.Handbuch z.PC-1600; ISBN 3-89374-001-5; S.107 und Tips+Tricks-Progr.handbuch f. PC-1600; ISBN 3-924327-85-8; S.5).

Aber Hand auf's Herz, wer macht das schon ? Den Akkus täte dies jedenfalls gut und würden es Ihnen mit einer langen Lebensdauer danken. Oft aber geht man einen einfacheren Weg (wie ich auch) und hängt den CE-1600P, wann immer möglich, ans Netz und nimmt ihn dann bei Bedarf ab und erfreut sich dann eines stets 100%igen Ladezustandes. 100%ig? -Nein, denn auf Dauer machen die Akkus diese Überlangen Ladezeiten nicht mit. Und wenn dann noch einige Tiefentladungen hinzukommen, so verkürzt sich die Lebensdauer erheblich (ich mußte die Akkus nach knapp 4 Jahren auswechseln). Zum Glück sind es handelsübliche Mignon-Akkus mit Lötflächen (16-24 DM f. 5 St.) für die man bei Sharp steckerfertig ein Vielfaches dafür bezahlen müßte.
 Um nun aber nicht wieder dieselben Fehler zu machen muß eine einfach zu bedienende und genaue Batteriespannungs-Überwachung her.
 Ein Digitalmultimeter wäre eine Möglichkeit, die eine direkte und genaue Spannungsangabe liefert. Jedoch benötigt man dazu ein weiteres Gerät mit separater Spannungsquelle. Ein Digitalvoltmeterpanel wäre zwar einbaubar, benötigt aber auch eine separate Spannungsquelle. Außerdem dürften diese Lösungen nicht gerade kostengünstig sein.

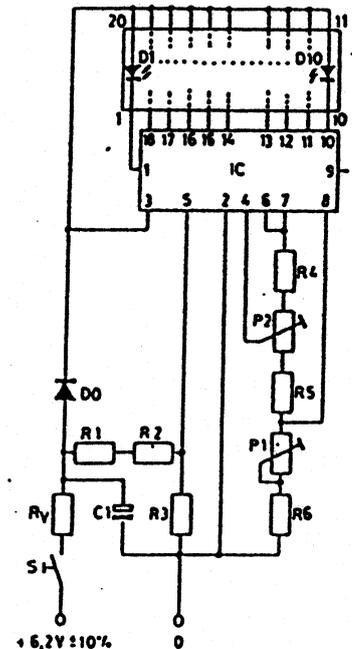
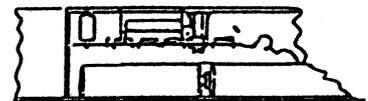
Ein Drehspulinstrument benötigt zwar keine separate Spannungsquelle, ist aber nicht genau genug ablesbar. Ein Batteriespannungs-Indikator stellt hingegen eine kostengünstige Lösung dar, die sehr klein baut und ebenfalls keine separate Spannungsquelle benötigt. Dafür zeigt es leider nur die Zustände GUT und SCHLECHT an und das macht ja schon das BATT-Symbol im PC-1600.

Ich habe nun eine Schaltung entdeckt, die relativ preiswert, klein, leicht abzugleichen, genau und leicht zu bedienen ist:

eine 10-fach-LED-Spannungs-Anzeige (Voltlupe) für den Zweigelebereich von 3,6 bis 6,8 Volt.



Die beschriebenen Bauteile bekommt man in gut sortierten Electronic-Fachgeschäften (Völkner, Conrad etc.) zum Preis von insg. 25 bis 30 DM. Ich habe die Schaltung auf eine Lochplatte der Größe 70 x 25 mm untergebracht und damit den Vorteil genutzt, diese in das Gehäuse des CE-1600P einzubauen. Eine schematische Anordnung sehen Sie auf den nachfolgenden Skizzen. Nachdem die Ausnehmungen für die LED-Anzeige und des versenkt einbauten Tasters sorgfältig ausgefeilt wurden, habe ich im Bereich der beiden Befestigungspunkte abgesägte Platinen-Distanzstücke mit Innengewinde unter die Oberseite des Gehäuses geklebt (Zweikomponentenkleber). Hieran konnte dann die Platine geschraubt werden. Sockelt man das IC und die LED-Anzeige, so hat man genügend Höhe für den Taster zur Verfügung. Der Taster (möglichst ein kleiner Mikroschwitch) sollte versenkt installiert sein, damit beim Zuklappen des Etuis dieser nicht betätigt wird. Sie können dafür einen Stift o.ä. benutzen.
 Den Aufbau der Schaltung entnehmen Sie dem Schaltplan. Die Anzeige benötigt ca.20 mA.



Zum Abgleich benötigt man ein einstellbares Netzteil und ein Digitalvoltmeter. Zunächst wird die Schaltung an das Netzteil angeschlossen und die Spannung auf 6,7 V eingestellt. P1 nun so verstellen, daß die 9. u. 10. LED aufleuchten. Nun 5,6 V einstellen und P2 so verstellen, daß die 1. LED gerade aufleuchtet. Nun wieder 6,7 V einstellen und wieder P1 justieren u.s.w. bis sich diese beiden Einstellungen gegeneinander nicht mehr beeinflussen.

- B : Taster (Mikroschwitch)
- Rv : 56 Ohm (ggf.47 Ohm)
- R1 : 10 kOhm
- R2 : 10 kOhm
- R3 : 10 kOhm
- R4 : 150 Ohm
- R5 : 560 Ohm
- R6 : 390 Ohm
- P1 : 250 Ohm
- P2 : 500 Ohm
- C1 : 10 uF/ 16 V
- D0 : 1N4001 o.ä.
- D1...D10 : 10-fach-LED-Anzeige im DIP-Gehäuse
- IC : LH3914

CE-1600P - Akkus leer ?

Nun, kein Problem. Netzgeraet EA-160 anschliessen und schon kann es weitergehen (und gleichzeitig werden dabei die Akkus auch wieder geladen) -Oder doch ein Problem, weil gerade kein Netzgeraet dabei?

Nun, um dieses zu vermeiden, gibt es zwei Moeglichkeiten:
Vor Gebrauch des 1600-Systems die Akkus ordnungsgemaess laden oder mittels Software den Ladezustand ueberpruefen.

Aber Hand auf's Herz, wer macht das schon? Den Akkus taete dies jedenfalls gut und wuerden es Ihnen mit einer langen Lebensdauer danken. Oft geht man aber einen einfacheren Weg (wie ich auch) und haengt den CE-1600P, wann immer moeglich, ans Netz und nimmt ihn dann bei Bedarf wieder ab und erfreut sich dann eines 100%igen Ladezustandes.

100%ig? -Nein, denn auf Dauer machen die Akkus diese ueberlangen Ladezeiten nicht mit. Und wenn dann noch einige Tiefentladungen hinzukommen, so verkuerzt sich die Lebensdauer erheblich.

Zum Glueck sind es handelsuebliche Mignon-Akkus mit Loetfahnen (16-24 DM fuer 5 St.) fuer die man steckerfertig bei Sharp ein Vielfaches dafuer bezahlen muesste.

Um nun aber nicht wieder dieselben Fehler zu machen, muss eine einfach zu bedienende und genaue Batteriespannung-Ueberwachung her.

Ein Digitalmultimeter waere eine Moeglichkeit, die eine direkte und genaue Spannungsangabe liefert. Jedoch benoetigt man dazu ein weiteres Geraet mit seperater Spannungsquelle.

Ein Digitalvoltmeterpanel waere zwar einbaubar, benoetigt aber auch eine seperate Spannungsquelle. Ausserdem duerften dies Loesungen nicht gerade kostenguenstig sein.

Ein Drehspulinstrument benoetigt zwar keine seperate Spannungsquelle, ist aber nicht genau genug ablesbar.

Ein Batteriespannung-Indikator stellt hingegen eine kostenguenstige Loesung dar, die sehr klein baut und ebenfalls keine seperate Spannungsquelle beoetigt. Dafuer zeigt es leider nur die Zustaeude GUT und SCHLECHT an und das macht ja schon das BATT-Symbol im PC-1600.

Ich habe nun eine Schaltung entdeckt, die relativ preiswert, klein, leicht abzugleichen, genau und leicht zu bedienen ist:

- eine 10-fach-LED-Band-Spannungs-Anzeige (Voltlupe)
- fuer den Anzeigebereich von 5.6 bis 6.8 Volt.

Die beschriebenen Bauteile bekommt man in gut sortierten Elektronik-Fachgeschaeften (Voelkner, Conrad etc.) zum Preis von insges. 25 bis 30 DM.

Ich habe die Schaltung auf eine Lochplatine der Groesse 70*25 mm untergebracht und damit den Vorteil genutzt, diese in das Gehaeuse des CE-1600P einzubauen. Eine schematische Anordnung sehen Sie auf den nachfolgende Skizzen.

Nachdem die Ausnehmungen fuer die LED-Band-Anzeige und des versenkt eingebauten Tasters sorgfaeltig ausgefeilt wurden, habe ich im Bereich der beiden Befestigungspunkten abgesaegte Platinen-Distanzstuecke mit Innengewinde unter die Oberseite des Gehaeuses geklebt

(Zweikomponentenkleber). Hieran konnte dann die Platine geschraubt werden. Sockelt man das IC und die LED-Band-Anzeige, so hat man genuegend Hoehe fuer den Taster zur Verfuegung. Der Taster (moeglichst ein kleiner Mikroswitch) sollte versenkt installiert sein, damit beim Zuklappen des Etais dieser nicht betaetigt und somit die Anzeige eingeschaltet wird. Sie koennen zum Einschalten einen kleinen Stift verwenden.

Den Aufbau der Schaltung entnehmen Sie dem Schaltplan.

Diese Methode klappt ganz gut, hat jedoch einen Haken:
 Denn bei Plotterbetrieb wird das System bei ca. 6.2 V abgeschaltet.
 Hinweis: Voll geladene Akkus haben $5 * 1.45 \text{ V} = 7.25 \text{ V}$
 Die Nennspannung betraegt $5 * 1.24 \text{ V} = 6.20 \text{ V}$
 Normale Entladg. auf etwa $5 * 0.9 \text{ V} = 4.5 \text{ V}$ moeglich.

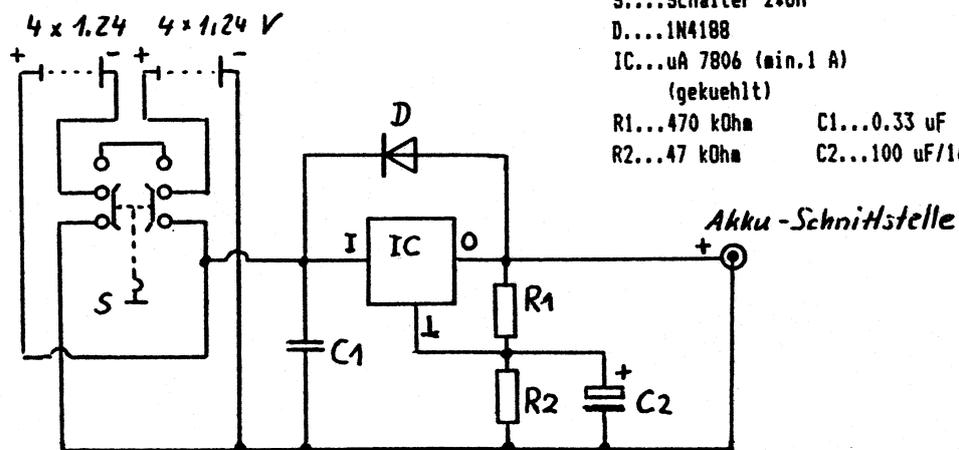
Hieran merkt man, dass der Plotterbetrieb mit Akkus kaum sinnvoll
 moeglich ist. Eine halbe bis eine Seite maximal.
 Die Zerstoerung der Akkus ist also auf den sog. Memory-Effekt sowie
 moeglicher Dauerladung zurueckzufuehren. Hier haette Sharp den Anwendern
 etwas Besseres bieten koennen (zumal das Netzgeraet ja auch nicht gerade
 billig ist).

Vielleicht gibt es jedoch noch einen Ausweg. Ich habe nachfolgend eine
 Idee aufskizziert, aber mangels Zeit noch nicht ausprobieren koennen.
 Also VORSICHT BEIM NACHBAUEN !

Ein Ergebnisbericht folgt dann in Kuerze.

Bauteileliste:

- 2 * 4 Baby-Akkus
 S....Schalter 2*UM
 D....1N4188
 IC...uA 7806 (min.1 A)
 (gekuehlt)
 R1...470 kOhm C1...0.33 uF
 R2...47 kOhm C2...100 uF/16V (Elko)



Die oben beschriebene Schaltung besteht im wesentlichen aus 2 Satz zu je
 4 Akkus (am besten Baby), einem Schalter 2*UM, einer Spannungsregelung
 sowie einer Freilaufdiode in Laderichtung.

Der Schalter schaltet die 2 * 4 Akkus entweder parallel (wie dargestellt)
 zur Aufladung und in Reihe fuer der CE-1600P-Akkubetrieb.

Ladeschaltung = 2 * 4 Akkus = 2*1200 mAh / 4*1.24 V
 Betriebsschaltg. = 1 * 8 Akkus = 1200 mAh / 8*1.24 V

Beim Ladebetrieb werden die Akkus aufgrund des hoeheren
 Netzadapter-Spannungspotentials (ca.8.4V zu ca.5V) ueber die Diode D
 geladen. Ladedauer duerfte jedoch min. 24 h betragen.

Beim Entladebetrieb, also zur Spannungsversorgung des CE-1600Ps, wird die
 Akkuspannung auf 2*5V hochgeschaltet, ueber den Positiv-Spannungsregler
 auf 6 V stabilisiert und mittels der Widerstaende R1,R2 auf ca.6.47 V
 angepasst. Der Spannungsregler arbeitet im sog. schwebenden Betrieb.

Die Ausgangsspannung berechnet sich wie folgt:

$$U_a = U_x * (1 + R_2/R_1) + I_q * R_2$$

U_x =Ausgangsspannung des ICs = 6V
 I_q =Ruhestrom des ICs = 10 mA
 gewaehlte Widerstaende R1=470 kOhm
 R2=47 Ohm
 $U_a = 6.47 \text{ V.}$

PC-1600-Kalender

'PC-1600-Kalender' ist ein Kalender-Programm für den SHARP-PC-1600 in einer ganz neuen Form. Geblieden sind nur noch der Titel und die Richtigkeit des Kalenders. Gestartet wird mit RUN.

Nach Eingabe der Jahreszahl (ab 1982) wird zunächst um ein wenig Geduld gebeten. Hiernach erscheint schließlich die Anzeige mit der Menü-Spalte in der untersten Zeile.

```

Jahr: 1991
-----
Mo Di Mi Do Fr Sa So
31 12 13 14 15 16 17 18
19 20 21 22 23 24 25 26

```

oder unten verschieben).
 (D) Das Drücken dieser Taste ermöglicht die numerische Eingabe eines Datums (Tag und Monat). Nach erfolgter Eingabe wird dann die dem Datum entsprechende Woche angezeigt.

```

Datum: Monat: 8
-----
Mo Di Mi Do Fr Sa So
31 12 13 14 15 16 17 18
Woche-Nr.: 52

```

(E) Das Drücken dieser Taste ermöglicht die Eingabe einer Woche-Nr. Die Daten dieser Woche werden hieraufhin angezeigt.

(F) Durch Drücken dieser Taste wird der komplette Jahreskalender auf dem CE-1600P ausgedruckt.
 (G) Durch Drücken dieser Taste wird das Programm beendet.

```

5 'KALENDER
6
10 CLEAR :CLS :WAIT 0:DIM M(12),MNS(12)
15 AS(O)S26,D(1),Zs(66,1)S26
20 INPUT "Jahr: ";J:IF J<1982GOTO 20

```

```

30 CLS
40 PRINT :PRINT " bitte warten...."
50
60 IF J>1976THEN LET F=1:GOTO 80
70 F=0
80 FOR Y=1TO OSTEP -1
90 H=INT ((J-Y)/100):G=(J-Y)-H*100-1
100 Z=S3+INT (1.2386)+INT (H/4)-24H/D(Y)
110 IF (J-Y)/100=INT ((J-Y)/100)AND (J-Y)
D(Y)+1
120 IF D(Y)>6THEN LET D(Y)=0
130 S=0
140 IF (G+1)/4=INT ((G+1)/4)THEN LET S=1
:IF G=1OAND ((H/4)<>INT (H/4))THE
N LET S=0
150 NEXT Y
160 IF D(1)<4AND D(0)>3THEN LET Os=" 53"
170 IF D(1)>3AND D(0)>4THEN LET Os="52":
GOTO 190
180 Os=" 1"
190 RESTORE 200:M(1)=S1:M(2)=28*S:FOR I=
3TD 12:READ M(I):NEXT I
200 DATA 31,30,31,30,31,31,30,31,30,31
210 RESTORE 230
220 FOR K=1TO 12:READ MNS(K):NEXT K
230 DATA "Januar","Februar","März","Apr
il","Mai","Juni"
240 DATA "Juli","August","September","Ok
tober","November","Dezember"
250 M=0
260 FOR K=1TO 12
300 AS(O)="":R=1
310 L=D(O)+M(K):V=L-(INT (L/7)*7)-1:IF V
=-1THEN LET V=6
320 IF V=6THEN LET R=0
330 Z=INT (L/7)
340 RESTORE (300-D(O))
350 FOR I=1TO Z
360 FOR U=1TO 7:READ N$:AS(O)=AS(O)+N$:N
EXT U
370 GOSUB 430:NEXT I
380 IF R=0GOTO 400
390 FOR I=1TO (1+V):READ N$:AS(O)=AS(O)+
N$:NEXT I:GOSUB 430:W=1
400 D(O)=V+1:IF D(O)=7THEN LET D(O)=0
410 NEXT K
420 GOTO 600
430 W=1:C=C+1:ES=STR$(W):IF W<10THEN
LET Es=" "+Es
440 IF F=1AND VAL (Os)>31THEN LET Zs(C,1)
=Os+"":AS(O)=AS(O)+W:O=O+S:"":GO
TO 480
450 IF F=1AND W>52AND (V+1)<4THEN LET Es
=" 1"
460 Zs(C,1)=Es+"":AS(O)=AS(O)+W
470
480 Zs(C,O)=" : "+MNS(K)+" "+STR$(J)+"
490 RETURN
494 DATA "
495 DATA "

```

```

496 DATA "
497 DATA "
498 DATA "
499 DATA "
500 DATA "
501 DATA "
502 DATA "
503 DATA "
504 DATA "
505 DATA "
506 DATA "
507 DATA "
508 DATA "
509 DATA "
510 DATA "
511 DATA "
512 DATA "
513 DATA "
514 DATA "
515 DATA "
516 DATA "
517 DATA "
518 DATA "
519 DATA "
520 DATA "
521 DATA "
522 DATA "
523 DATA "
524 DATA "
525 DATA "
526 DATA "
527 DATA "
528 DATA "
529 DATA "
530 DATA "
531 DATA "
532 DATA "
533 DATA "
534 DATA "
535 DATA "
536 DATA "
537 DATA "
538 DATA "
539 DATA "
540 DATA "
541 DATA "
542 DATA "
543 DATA "
544 DATA "
545 DATA "
546 DATA "
547 DATA "
548 DATA "
549 DATA "
550 DATA "
551 DATA "
552 DATA "
553 DATA "
554 DATA "
555 DATA "
556 DATA "
557 DATA "
558 DATA "
559 DATA "
560 DATA "
561 DATA "
562 DATA "
563 DATA "
564 DATA "
565 DATA "
566 DATA "
567 DATA "
568 DATA "
569 DATA "
570 DATA "
571 DATA "
572 DATA "
573 DATA "
574 DATA "
575 DATA "
576 DATA "
577 DATA "
578 DATA "
579 DATA "
580 DATA "
581 DATA "
582 DATA "
583 DATA "
584 DATA "
585 DATA "
586 DATA "
587 DATA "
588 DATA "
589 DATA "
590 DATA "
591 DATA "
592 DATA "
593 DATA "
594 DATA "
595 DATA "
596 DATA "
597 DATA "
598 DATA "
599 DATA "
600 DATA "
601 DATA "
602 DATA "
603 DATA "
604 DATA "
605 DATA "
606 DATA "
607 DATA "
608 DATA "
609 DATA "
610 DATA "
611 DATA "
612 DATA "
613 DATA "
614 DATA "
615 DATA "
616 DATA "
617 DATA "
618 DATA "
619 DATA "
620 DATA "
621 DATA "
622 DATA "
623 DATA "
624 DATA "
625 DATA "
626 DATA "
627 DATA "
628 DATA "
629 DATA "
630 DATA "
631 DATA "
632 DATA "
633 DATA "
634 DATA "
635 DATA "
636 DATA "
637 DATA "
638 DATA "
639 DATA "
640 DATA "
641 DATA "
642 DATA "
643 DATA "
644 DATA "
645 DATA "
646 DATA "
647 DATA "
648 DATA "
649 DATA "
650 DATA "
651 DATA "
652 DATA "
653 DATA "
654 DATA "
655 DATA "
656 DATA "
657 DATA "
658 DATA "
659 DATA "
660 DATA "
661 DATA "
662 DATA "
663 DATA "
664 DATA "
665 DATA "
666 DATA "
667 DATA "
668 DATA "
669 DATA "
670 DATA "
671 DATA "
672 DATA "
673 DATA "
674 DATA "
675 DATA "
676 DATA "
677 DATA "
678 DATA "
679 DATA "
680 DATA "
681 DATA "
682 DATA "
683 DATA "
684 DATA "
685 DATA "
686 DATA "
687 DATA "
688 DATA "
689 DATA "
690 DATA "
691 DATA "
692 DATA "
693 DATA "
694 DATA "
695 DATA "
696 DATA "
697 DATA "
698 DATA "
699 DATA "
700 DATA "
701 DATA "
702 DATA "
703 DATA "
704 DATA "
705 DATA "
706 DATA "
707 DATA "
708 DATA "
709 DATA "
710 DATA "
711 DATA "
712 DATA "
713 DATA "
714 DATA "
715 DATA "
716 DATA "
717 DATA "
718 DATA "
719 DATA "
720 DATA "
721 DATA "
722 DATA "
723 DATA "
724 DATA "
725 DATA "
726 DATA "
727 DATA "
728 DATA "
729 DATA "
730 DATA "
731 DATA "
732 DATA "
733 DATA "
734 DATA "
735 DATA "
736 DATA "
737 DATA "
738 DATA "
739 DATA "
740 DATA "
741 DATA "
742 DATA "
743 DATA "
744 DATA "
745 DATA "
746 DATA "
747 DATA "
748 DATA "
749 DATA "
750 DATA "
751 DATA "
752 DATA "
753 DATA "
754 DATA "
755 DATA "
756 DATA "
757 DATA "
758 DATA "
759 DATA "
760 DATA "
761 DATA "
762 DATA "
763 DATA "
764 DATA "
765 DATA "
766 DATA "
767 DATA "
768 DATA "
769 DATA "
770 DATA "
771 DATA "
772 DATA "
773 DATA "
774 DATA "
775 DATA "
776 DATA "
777 DATA "
778 DATA "
779 DATA "
780 DATA "
781 DATA "
782 DATA "
783 DATA "
784 DATA "
785 DATA "
786 DATA "
787 DATA "
788 DATA "
789 DATA "
790 DATA "
791 DATA "
792 DATA "
793 DATA "
794 DATA "
795 DATA "
796 DATA "
797 DATA "
798 DATA "
799 DATA "
800 DATA "
801 DATA "
802 DATA "
803 DATA "
804 DATA "
805 DATA "
806 DATA "
807 DATA "
808 DATA "
809 DATA "
810 DATA "
811 DATA "
812 DATA "
813 DATA "
814 DATA "
815 DATA "
816 DATA "
817 DATA "
818 DATA "
819 DATA "
820 DATA "
821 DATA "
822 DATA "
823 DATA "
824 DATA "
825 DATA "
826 DATA "
827 DATA "
828 DATA "
829 DATA "
830 DATA "
831 DATA "
832 DATA "
833 DATA "
834 DATA "
835 DATA "
836 DATA "
837 DATA "
838 DATA "
839 DATA "
840 DATA "
841 DATA "
842 DATA "
843 DATA "
844 DATA "
845 DATA "
846 DATA "
847 DATA "
848 DATA "
849 DATA "
850 DATA "
851 DATA "
852 DATA "
853 DATA "
854 DATA "
855 DATA "
856 DATA "
857 DATA "
858 DATA "
859 DATA "
860 DATA "
861 DATA "
862 DATA "
863 DATA "
864 DATA "
865 DATA "
866 DATA "
867 DATA "
868 DATA "
869 DATA "
870 DATA "
871 DATA "
872 DATA "
873 DATA "
874 DATA "
875 DATA "
876 DATA "
877 DATA "
878 DATA "
879 DATA "
880 DATA "
881 DATA "
882 DATA "
883 DATA "
884 DATA "
885 DATA "
886 DATA "
887 DATA "
888 DATA "
889 DATA "
890 DATA "
891 DATA "
892 DATA "
893 DATA "
894 DATA "
895 DATA "
896 DATA "
897 DATA "
898 DATA "
899 DATA "
900 DATA "
901 DATA "
902 DATA "
903 DATA "
904 DATA "
905 DATA "
906 DATA "
907 DATA "
908 DATA "
909 DATA "
910 DATA "
911 DATA "
912 DATA "
913 DATA "
914 DATA "
915 DATA "
916 DATA "
917 DATA "
918 DATA "
919 DATA "
920 DATA "
921 DATA "
922 DATA "
923 DATA "
924 DATA "
925 DATA "
926 DATA "
927 DATA "
928 DATA "
929 DATA "
930 DATA "
931 DATA "
932 DATA "
933 DATA "
934 DATA "
935 DATA "
936 DATA "
937 DATA "
938 DATA "
939 DATA "
940 DATA "
941 DATA "
942 DATA "
943 DATA "
944 DATA "
945 DATA "
946 DATA "
947 DATA "
948 DATA "
949 DATA "
950 DATA "
951 DATA "
952 DATA "
953 DATA "
954 DATA "
955 DATA "
956 DATA "
957 DATA "
958 DATA "
959 DATA "
960 DATA "
961 DATA "
962 DATA "
963 DATA "
964 DATA "
965 DATA "
966 DATA "
967 DATA "
968 DATA "
969 DATA "
970 DATA "
971 DATA "
972 DATA "
973 DATA "
974 DATA "
975 DATA "
976 DATA "
977 DATA "
978 DATA "
979 DATA "
980 DATA "
981 DATA "
982 DATA "
983 DATA "
984 DATA "
985 DATA "
986 DATA "
987 DATA "
988 DATA "
989 DATA "
990 DATA "
991 DATA "
992 DATA "
993 DATA "
994 DATA "
995 DATA "
996 DATA "
997 DATA "
998 DATA "
999 DATA "
1000 DATA "

```


MATHEMATISCHE OPERATIONEN AUF DER EBENE VON SYMBOLISCHEN ELEMENTEN

an Beispiel der Division zweier beliebiger Polynome (PC1600,PC1500A)

$$(2*A*B*C + A*B^2 + A*C^2 + B^2*C + B*C^2) / (A*B + A*C + B*C) = ??$$

$$(A^4 - B^4) / (A - B) = ?? \quad \text{etc.}$$

Die auf Personal Computern und auch die auf Grossrechnern verfügbaren Programme zur Verarbeitung von symbolischen Elementen sind meist in LISP geschrieben. Dies ist mit Nachteilen verbunden, der Rechenzeitbedarf und auch der Hauptspeicherbedarf sind recht hoch, die Verknüpfung mit rein numerischen Programmen zwecks sofortiger Verwertung der einzelnen neu gewonnenen mathematischen Ausdrücke (bei Ersetzung der symbolischen Variablen durch numerische Werte) in Zusammenwirken mit bereits programmierten numerischen Abläufen erweist sich als recht aufwendig. Fragestellungen, die in fertigen LISP-Programmpaketen nicht vorgesehen sind, lassen sich oft nur mit viel Mühe so umformulieren, dass diese Programmpakete eingesetzt werden können. Folglich empfiehlt es sich, insbesondere bei speziellen Fragestellungen die Verarbeitung symbolischer Matrixelemente mit einem massgeschneiderten Programm in einer mathematisch ausgerichteten Programmiersprache anzugehen (etwa in Fortran oder in Algol). Die Verknüpfung mit rein numerischen Programmen, die in der gleichen Programmiersprache geschrieben sind, ist dann problemlos. Als Nebeneffekt ergibt sich meist eine deutlich geringere Rechenzeit (Faktoren bis 100). Zur Demonstration dient die hier skizzierte Entwicklung eines BASIC-Programms, wobei BASIC zwar zu ähnlich hohen Rechenzeiten führt wie LISP, doch kann ein BASIC-Programm bereits auf einem programmierbaren Taschenrechner ausgetestet werden, um anschliessend ohne Mühe in die höhere mathematisch orientierte Programmiersprache (Fortran, Algol) übertragen zu werden. Als Problemstellung wird hier die Division zweier beliebiger Polynome gewählt, da diese Aufgabe komplex genug ist, um die wichtigsten Schritte bei der Erstellung von formelverarbeitenden Programmen anschaulich zu erläutern, während die Aufgabe andererseits so einfach gehalten ist, dass keine besonderen mathematischen Fachkenntnisse erforderlich sind.

Erster Schritt: Eine klare Festlegung der zu verarbeitenden Elemente und der gewünschten mathematischen Operationen, um die Art der Abspeicherung der zu verarbeitenden Grössen bestimmen zu können.

Polynome benötigen drei Ebenen für die Abspeicherung, Namen und Länge und Startadresse des Polynoms (level 1), Numerischen Vorfaktor und Länge und Startadresse jedes einzelnen Summanden (level 2), Adresse und Exponenten jeder einzelnen Variablen in den Produkten (level 3). Die Adresse jeder Variablen verweist auf einen Speicherplatz in der obersten Ebene (level 1). Die Division von Polynomen schliesst die Subtraktion, die Addition und einzelne Multiplikationsschritte ein. Bei diesen Operationen werden die Polynome von vorn nach hinten abgearbeitet. Zeiger brauchen folglich nur auf Folgeelemente, nicht auf rückwärts liegende Elemente zu verweisen.

Es ist fast immer sinnvoll, den Speicher blockweise zu vergeben. Werden die Blöcke zu gross gewählt, so wird Speicherplatz verschwendet, bei zu kleinen Blöcken wird durch eine zu häufige Fortschaltung der Indizes Rechenzeit verschwendet. Ein guter Kompromiss liegt meist bei einer Blocklänge, bei der die durchschnittliche Anzahl von logisch zusammenhängenden Elementen gleich der Blocklänge ist. Also werden die Blocklängen D2, D3 in unserem Beispiel auf der Ebene 2 (D2) gleich der durchschnittlichen Anzahl von Summanden je Polynom und auf der Ebene 3 (D3) gleich der durchschnittlichen Anzahl von Variablen je Produkt gewählt. Blockbelegungsvektoren (B2 und B3 in unserem Beispiel) geben Auskunft über belegte und un belegte Blöcke sowie den Verweis auf einen gegebenenfalls vorhandenen Anschlussblock.

Bei den Vorfaktoren beschränken wir uns auf ganze Zahlen (Integers), bei den Exponenten lassen wir positive ganze Zahlen und null zu. Die Länge der Ein- und Ausgabestrings kann durch maximale Wortlängen begrenzt sein (16 Bytes auf den Taschenrechner, D1) und verlangt dann die Möglichkeit beliebig vieler Folgezeilen.

Zweiter Schritt: Die Programnteile für das Löschen und für die Ausgabe der zu verarbeitenden Elemente (hier Polynome) stellen einen guten Test für die gewählte Abspeicherung (hier in drei Ebenen) und die formulierte Indexfortschaltung (hier mit Blockbelegungsvektoren für die unteren beiden Ebenen) dar. Werden diese Programnteile sehr umfangreich, so ist die Abspeicherung und die Indexfortschaltung neu zu überdenken, da anderenfalls die später zu programmierenden mathematischen Verknüpfungen viel zu lang und zu unübersichtlich werden.

Dritter Schritt: Es wird die Eingabe programmiert. Zum einen wird festgelegt, welche Eingabeformen zugelassen werden (hier: "X;" fuer das Drucken des Polynoms X und "X:=;" fuer das Loeschen des Polynoms X und "X:=.....;" fuer die Belegung des Polynoms X und "X/Y;" fuer die Division zweier Polynome, Leerzeichen werden ueberall ignoriert, die Potenzbildung wird durch einen Doppelstern "*" gekennzeichnet). Die Laenge der Namen der Polynome und der Variablen wird festgelegt (hier: genau ein Gross-Buchstabe). Insbesondere die Eingabe der einzelnen Elemente (hier: die Zuweisung des Inhalts zum Namen des Polynoms) wird fertiggestellt, da von dieser Stelle nochmals kleinere Korrekturen an der Definition der Abspeicherung und der Indexfortschaltung erzwungen werden koennen.

Vierter Schritt: Allgemeine Hilfsroutinen fuer die mathematischen Operationen werden erstellt (hier: vornehmlich Routinen zum Sichern und Wiederbeschaffen von global definierten Indizes. Die von Taschenrechnern diktierte Beschraenkung auf BASIC-Namen der maximalen Laenge von zwei Zeichen ist hierbei etwas laestig).

Fuenfter Schritt: Fuer die Elemente, die in die mathematischen Operationen Eingang finden, wird eine Normalform definiert. Und mit entsprechenden Routinen wird diese Normalform herbeigefuehrt (hier: das Sortieren der Polynome unter Zuhilfenahme der Adressen der einzelnen Variablennamen als Sortierkriterium).

Sechster Schritt: Die mathematischen Operationen werden programmiert (hier: Kopieren, Addieren, Subtrahieren, Multiplizieren und Dividieren). Dabei zeigt sich haeufig die Wichtigkeit einer klug definierten Normalform (hier: beim Dividieren braucht in jedem Teilschritt nur das erste Produkt des Zaehler-Polynoms durch das erste Produkt des Nenner-Polynoms dividiert zu werden, um den naechsten Summanden in Ergebnis-Polynom zu erhalten).

Anwendungsbeispiele:

1) Eingabe von Polynomen (U und W)
und Ausgabebefehle

U :=D +C +B +A;	Zuweisung (Definition von U)
U ;	Ausgabe: U=A+B+C+D;
U := ;	Loeschen des Inhalts von U
U :=2 *A**0 *2 *B**2 *C *D**2 -1 * C * A * A * 3**2 + 8 ;	Zuweisung (Neubelegung von U)
W :=-2*D *B * C + 3 * A * C ;	Zuweisung (Definition von W)
U ;	Ausgabe: U=-9*A**2*C+4 *B**2*C*D**2;
W ;	Ausgabe: W=3*A*C-2*B*C*D;
U / W ;	Ausgabe: U/W=-3*A-2*B*D;
U :=A**4-9*B**4 -8*A**2*B**2 -3*W*B**3 ;	Zuweisung (Neubelegung von U)
W := A - 3*B ;	Zuweisung (Neubelegung von W)
U / W ;	Ausgabe: U/W=A**3+3*A**2 *B+A*B**2;

2) Die Loesungen fuer die zwei in den Eingangszeilen gestellten
Aufgaben lauten

B + C
und

A**3 + A**2 * B + A * B**2 + B**3

Speicherbedarf fuer das Programm (Taschenrechner Sharp PC1600): 12 KB
plus Speicherbedarf fuer Variablen und die gewaehlten Feldlaengen: 7 KB

Aehnliche Programme wurden von Autor fuer die Loesung spezieller Aufgaben entwickelt, wobei Exponentialfunktionen, Gaussfunktionen und die Fehlerfunktion einbezogen wurden, waehrend bei den Operationen die Verwendung von Differentialoperatoren und Integrationsschritten Eingang fand. Auch wurden mehrfach indizierte Variablen verwendet, bei denen die Indizes jeweils zu Anteilen kommutativ, antikommutativ und nicht-kommutativ waren.

MATHEMATISCHE OPERATIONEN AUF DER EBENE VON SYMBOLISCHEN ELEMENTEN

an Beispiel der Division zweier beliebiger Polynome

```

constants and dimensions
o01: M1 = 20      o02: M2 = 10      o03: M3 = 25
o04: D1 = 16      o05: D2 = 8       o06: D3 = 4
o07: DIM I1$(2*D1)
o08: DIM N1$(M1), L1(M1), A1(M1)
o09: DIM B2(M2), V2(D2,M2), L2(D2,M2), A2(D2,M2)
o10: DIM B3(M3), A3(D3,M3), E3(D3,M3)  o11: GOTO "IT"
.....
read and erase arrays, address handling
o12: "X2"
      fill P1
o13: FOR I = 1 TO M1      o14: P1 = I
o15: IF NX$=M1$(I) RETURN
o16: NEXT I      o17: P1 = 0      o18: RETURN
o19: "X1"
      read array, on level 1
o20: NX$ = N1$(P1)      o21: LX = L1(P1)
o22: PY = A1(P1)      o23: S2 = 0      o24: RETURN
o25: "X0"
      erase array, on level 1
o26: L1(P1) = 0      o27: A1(P1) = 0      o28: RETURN
o29: "Y3"
      release space, on level 2
o30: IF MX<D2+1 THEN LET B2(P2) = -1
o31: IF MX<1 THEN LET B2(P2) = 0      o32: RETURN
o33: "Y2"
      fill S2,P2
o34: P2 = PY      o35: S2 = 0
o36: PY = B2(P2)      o37: RETURN
o38: "Y1"
      read array, on level 2
o39: IF S2=D2 OR S2=0 GOSUB "Y2"
o40: LX = LX - 1      o41: VY = V2(S2,P2)
o42: LY = L2(S2,P2)      o43: S3 = 0
o44: S3 = 0      o45: "Y0"
o46: V2(S2,P2) = 0      o47: "Y0"
o48: A2(S2,P2) = 0      o49: L2(S2,P2) = 0
o50: A2(S2,P2) = 0      o51: B2(P2) = 0
o52: RETURN
o53: "Z3"
      release space, on level 3
o54: IF MY<D3+1 THEN LET B3(P3) = -1
o55: IF MY<1 THEN LET B3(P3) = 0      o56: RETURN
o57: "Z2"
      fill S3,P3
o58: P3 = P2      o59: S3 = 0
o60: P2 = B3(P3)      o61: RETURN
o62: "Z1"
      read array, on level 3
o63: IF S3=D3 OR S3=0 GOSUB "Z2"
o64: S3 = S3 + 1      o65: LY = LY - 1
o66: AZ = A3(S3,P3)      o67: E3(S3,P3) = 0
o68: RETURN      o69: "Z0"
o70: A3(S3,P3) = 0      o71: E3(S3,P3) = 0
o72: B3(P3) = 0      o73: RETURN
.....
delete NX$, keep name
o74: "DE"
      level 1
o75: GOSUB "X2"      o76: IF P1=0 RETURN
o77: GOSUB "X1"      o78: GOSUB "X0"
o79: "DY"
      level 2 (delete all elements of the sum)
o80: IF LX=0 RETURN      o81: GOSUB "Y1"
o82: GOSUB "Y0"      o83: GOSUB "D2"      o84: GOTO "DY"
o85: "DZ"
      level 3 (delete single product)
o86: IF LY=0 RETURN      o87: GOSUB "Z1"
o89: GOSUB "Z0"      o89: GOTO "DZ"
.....
print NX$ (output), D1=16 (dimension of I/O string)
o90: "OT"
      level 1
o91: OF = 0      o92: O0$ = O0$      o93: GOSUB "X2"
o94: GOSUB "X1"      o95: GOTO "OY"
o96: "OP"
      print
o97: PRINT O0$      o98: O0$ = ""      o99: RETURN
100: "OX"
      return
101: LO = LEN(O0$) + 1
102: IF LO>D1 GOSUB "OP"
103: O0$ = O0$ + ";"
104: PRINT O0$      105: O0$ = O0$      106: RETURN
107: "OY"
      level 2, flag OF for first element
108: IF LX=0 GOTO "OX"      109: GOSUB "Y1"
110: O1$ = "-"      111: IF OF=0 THEN LET O1$ = ""
112: IF VY<0 THEN LET O1$ = "-"
113: VY = ABS(VY)      114: OF = 1
115: O2$ = STR$(VY)      116: O3$ = "-"
117: IF LY=0 THEN LET O3$ = ""
118: IF LY>0 AND VY=1 THEN LET O2$ = ""
119: IF LY>0 AND VY=-1 THEN LET O3$ = ""
120: LO = LEN(O0$) + LEN(O1$) + LEN(O2$)
121: IF LO>D1 GOSUB "OP"
122: O0$ = O0$ + O1$ + O2$
123: O1$ = O3$      124: GOSUB "OZ"      125: GOTO "OY"
126: "OZ"
      level 3

```

```

127: IF LY=0 RETURN      128: GOSUB "Z1"
129: O2$ = M1$(A2)      130: O3$ = "*" + STR$(E2)
131: IF E2=1 THEN LET O3$ = ""
132: LO = LEN(O0$) + LEN(O1$) + LEN(O2$) + LEN(O3$)
133: IF LO>D1 GOSUB "OP"
134: O0$ = O0$ + O1$ + O2$ + O3$
135: O1$ = "-"      136: GOTO "OZ"
.....
input section with I1$(32), D1=16 (dimension of I/O string)
137: "IT"
      length IL, flag JL for ";"
138: IL = 0      139: JF = 0      140: GOSUB "IX"
      exit, entry "X;" entry "X/Y;" entry "X:=...;"
141: IF I1$(1)="" PRINT "END"
142: IF I1$(1)="" END
143: IF I1$(2)="" GOTO "IA"
144: IF I1$(2)="/" AND I1$(4)="" GOTO "IB"
145: IF I1$(2)="" AND I1$(3)="" GOTO "IC"
      input error, operation not defined, return to "IT"
146: "IR"
147: I2 = 0
148: "IS"
149: I1 = I2 + 1
150: I2 = I2 + D1
151: IF I2>IL THEN LET I2 = IL      152: IN$ = ""
153: FOR I = 1 TO I2
154: IN$ = IN$ + I1$(I)      155: NEXT I
156: PRINT "IN-ERR: ";IN$      158: GOTO "IS"
157: IF I2=IL GOTO "IT"
      input error, polynomial NX$
159: "ER"
160: GOSUB "DE"
161: PRINT "IN-ERR: ";NX$      162: GOTO "IT"
      input, single line
163: "IX"
164: IN$ = ""      165: INPUT "INPUT: ";IN$.
166: PRINT "IN: ";IN$      168: IL = IL + 1
167: FOR I = 1 TO D1
169: I1$(IL) = MID$(IN$,I,1)
170: IF I1$(IL)="" THEN LET JF = 1
171: IF I1$(IL)="/" OR I1$(IL)="" THEN LET IL = IL - 1
172: NEXT I      173: IF IL=0 GOTO "IX"
174: IF JF=0 AND IL<D1 GOTO "IX"      175: RETURN
      wrong type
176: "IY"
177: IF I1$(1)<"A" OR I1$(1)>"Z" GOTO "IR"      178: RETURN
      squeeze line
179: "IZ"
180: IF IL=0 GOSUB "IX"
181: I2 = IL      182: IL = 0
183: FOR I = 1 TO I2      184: IL = IL + 1
185: I1$(IL) = I1$(I)
186: IF I1$(IL)="" OR I1$(IL)="" THEN LET IL = IL - 1
187: NEXT I      188: IF IL=0 GOSUB "IX"
189: IF JF=0 AND IL<D1 GOSUB "IX"      190: RETURN
      entry "X;"
191: "IA"
192: I1 = 1      193: GOSUB "IY"      194: NX$ = I1$(1)
195: O0$ = NX$ + "-"
196: I1 = 0      197: JF = 0
198: GOSUB "X2"      199: IF P1=0 GOTO "UN"
200: GOSUB "X1"      201: IF LX=0 GOTO "EM"
202: GOSUB "SO"      203: GOSUB "OT"      204: GOTO "IT"
      entry "Y/Z;"
205: "IB"
206: I1 = 1      207: GOSUB "IY"
208: I1 = 3      208: GOSUB "IY"
210: NY$ = I1$(1)      211: NZ$ = I1$(3)
212: O0$ = NY$ + "/" + NZ$ + "-"
213: I1 = 0      214: JF = 0      215: NX$ = NY$
216: GOSUB "X2"      217: IF P1=0 GOTO "UN"
218: GOSUB "X1"      219: IF LX=0 GOTO "EM"
220: GOSUB "SO"
222: GOSUB "X2"      223: IF P1=0 GOTO "UN"
224: GOSUB "X1"      225: IF LX=0 GOTO "EM"
226: GOSUB "SO"      227: GOTO "DY"
228: "UN"
229: PRINT NX$;" UNKNOWN"      230: GOTO "IT"
231: "EM"
232: PRINT NX$;" EMPTY"      233: GOTO "IT"
.....
fill arrays, address handling
234: "XA"
      find new space on M1
235: FOR I = 1 TO M1      238: PX = I
237: IF M1$(I)="" OR M1$(I)="" GOTO "XB"
238: NEXT I      239: PRINT "M1 FULL"
240: END
241: "XB"
242: L1(PX) = 0      243: A1(PX) = 0      244: RETURN
245: "YA"
      find new space on M2
246: FOR I = 1 TO M2      247: PY = I
248: IF A2(I)=0 GOTO "YB"      249: NEXT I
250: PRINT "M2 FULL"      251: END
252: "YB"
253: B2(PY) = -1      254: S3 = 0      255: RETURN
256: "ZA"
      find new space on M3
257: FOR I = 1 TO M3      258: PZ = I
259: IF B3(I)=0 GOTO "ZB"      260: NEXT I
261: PRINT "M3 FULL"      262: END
263: "ZB"
264: B3(PZ) = -1      265: RETURN
266: "XC"
      store new name of polynomial on M1
      or delete old polynomial of same name

```

```

267: GOSUB "X2"      268: IF P1>0 GOSUB "DE"
269: IF P1=0 GOSUB "XA"
270: IF P1=0 THEN LET P1 = PX
271: M1$(P1) = NX$  272: GOSUB "X1"      273: RETURN
274: "YC"
    IF L1(P1)=0 start first product on M2
    IF S2=D2 open next block on B2
275: GOSUB "YA"      276: IF S2=0 THEN LET A1(P1) = PY
277: IF S2=D2 THEN LET B2(P2) = PY
278: P2 = PY        279: S2 = 0      280: RETURN
281: "ZC"
    IF L2(S2,P2)=0 start new product on M3
    IF S3=D3 open next block on B3
282: GOSUB "ZA"      283: IF S3=0 THEN LET A2(S2,P2) = PZ
284: IF S3=D3 THEN LET B3(P3) = PZ
285: P3 = PZ        286: S3 = 0      287: RETURN
288: "YE"
    next element on level 2
289: IF S2=0 OR S2=D2 GOSUB "YC"
290: L1(P1) = L1(P1) + 1      291: S2 = S2 + 1
292: V2(S2,P2) = 0      293: L2(S2,P2) = 0
294: A2(S2,P2) = 0      295: S3 = 0
296: LX = 0      297: RETURN
298: "YG"
    next element on level 2, from input
299: GOSUB "YE"      300: VY = 1
301: IF I1$(1)="" THEN LET VY = -1      302: V2(S2,P2) = VY
303: IF I1$(1)="" OR I1$(1)="" THEN LET I1$(1) = ""
304: GOSUB "IZ"      305: RETURN
306: "ZE"
    next element on level 3
    write address AB and exponent EB, on level 3
307: IF S3=0 OR S3=D3 GOSUB "ZC"
308: L2(S2,P2) = L2(S2,P2) + 1      309: S3 = S3 + 1
310: A3(S3,P3) = AB      311: E3(S3,P3) = EB
312: LY = 0      313: RETURN
314: "ZG"
    next element on level 3, name NQ$ and exponent EB
    if nAneflag=1 find address of name or store, new name
315: FOR I = 1 TO M1      316: PX = I
317: IF NQ$=M1$(I) GOTO "ZH"      318: NEXT I
319: GOSUB "XA"      320: M1$(PX) = NQ$
321: "ZH"
322: AB = PX      323: GOSUB "ZE"      324: RETURN
325: "NM"
    number or name
326: UF = 0      327: VB = 0      328: GOSUB "HU"
329: AF = 0      330: GOSUB "HA"
331: IF UF+AF=0 GOTO "ER"
    look for exponent
332: EF = 0      333: EB = 0      334: GOSUB "HE"
335: IF EF=0 THEN LET EB = 1
336: IF UF=1 THEN LET V2(S2,P2) = V2(S2,P2) * VB*EB
    the character "" stands for "" (exponentiation)
337: RETURN
338: "HU"
    find number
339: IF I1$(1)<"0" OR I1$(1)>"9" RETURN
340: UF = 1      341: VB = 10 * VB + VAL(I1$(1))
342: I1$(1) = ""      343: GOSUB "IZ"      344: GOTO "NU"
345: "NA"
    find name
346: IF I1$(1)<"A" OR I1$(1)>"Z" RETURN
347: AF = 1      348: NQ$ = I1$(1)
349: IF NQ$=NX$ PRINT "RECCURENCE: ";NX$;"=";NQ$
350: IF NQ$=NX$ GOTO "ER"      351: I1$(1) = ""
352: GOSUB "IZ"      353: RETURN
354: "NE"
    find exponent
355: IF I1$(2)="" OR I1$(2)="" GOSUB "IZ"
356: IF I1$(1)="" OR I1$(2)="" THEN LET EF = 1
357: IF EF=0 RETURN
358: I1$(1) = ""      359: I1$(2) = ""      360: GOSUB "IZ"
361: IF I1$(1)<"0" OR I1$(1)>"9" GOTO "ER"
362: "NG"
363: EB = 10 * EB + VAL(I1$(1))      364: I1$(1) = ""
365: GOSUB "IZ"      366: IF I1$(1)<"0" OR I1$(1)>"9" RETURN
367: GOTO "HG"
*****
read new polynomial, entry "X:=...;"
368: "IC"
369: I1 = 1      370: GOSUB "IY"
371: IF I1<4 GOTO "IR"      372: NX$ = I1$(1)
373: I1$(1) = ""      374: I1$(2) = ""      375: I1$(3) = ""
376: GOSUB "IZ"
    on level 1 store name of polynomial
377: GOSUB "XC"      378: IF I1$(1)="" GOTO "IT"
379: "YP"
    start product on level 2 (with +,-,A to Z,0 to 9, error)
380: GOSUB "YG"
    start with number or name, if AF=1 new product (level 3)
381: GOSUB "NM"      382: IF AF=1 GOSUB "ZQ"
383: "ZP"
    continue with * + - ; or error
384: IF I1$(1)="" OR I1$(1)="" GOSUB "IZ"
385: IF I1$(1)="" GOTO "IT"
386: IF I1$(1)="" GOTO "ZQ"
387: IF I1$(1)="" OR I1$(1)="" GOTO "YP"
388: GOTO "ER"
389: "ZQ"
    next term on level 3 (starting with "A"), multiply
    continue with number or name, if AF=1 extend product
390: I1$(1) = ""      391: GOSUB "IZ"      392: GOSUB "NM"
393: IF AF=1 GOSUB "ZG"      394: GOTO "ZP"
*****
save and restore level 2 and level 3, read and write level 3
395: "UA"
    save level 2
396: U1 = P1      397: U2 = LX      398: U3 = P2
399: U4 = S2      400: U5 = PY      401: RETURN
402: "UB"
    restore level 2
403: P1 = U1      404: LX = U2      405: P2 = U3
406: S2 = U4      407: PY = U5      408: VY = V2(S2,P2)
409: LY = L2(S2,P2)      410: PZ = A2(S2,P2)
411: S3 = 0      412: RETURN
413: "UC"
    save level 2
414: U6 = P1      415: U7 = LX      416: U8 = P2
417: U9 = S2      418: Uo = PY      419: RETURN
420: "UD"
    restore level 2
421: P1 = U6      422: LX = U7      423: P2 = U8
424: S2 = U9      425: PY = Uo      426: VY = V2(S2,P2)
427: LY = L2(S2,P2)      428: PZ = A2(S2,P2)
429: S3 = 0      430: RETURN
431: "UP"
    save level 2
432: V6 = P1      433: V7 = LX      434: V8 = P2
435: V9 = S2      436: Vo = PY      437: RETURN
438: "UQ"
    restore level 2
439: P1 = V6      440: LX = V7      441: P2 = V8
442: S2 = V9      443: PY = Vo      444: VY = V2(S2,P2)
445: LY = L2(S2,P2)      446: PZ = A2(S2,P2)
447: S3 = 0      448: RETURN
449: "VA"
    save level 3
450: V1 = P2      451: VJ = S2      452: V1 = LY
453: V2 = P3      454: V3 = S3      455: V4 = PZ
456: RETURN
457: "VB"
    restore level 3
458: P2 = V1      459: S2 = VJ      460: LY = V1
461: P3 = V2      462: S3 = V3      463: PZ = V4
464: AZ = A3(S3,P3)      465: EZ = E3(S3,P3)
466: RETURN
467: "VC"
    save level 3
468: VK = P2      469: VL = S2      470: V6 = LY
471: V7 = P3      472: V8 = S3      473: V9 = PZ
474: RETURN
475: "VD"
    restore level 3
476: P2 = VK      477: S2 = VL      478: LY = V6
479: P3 = V7      480: S3 = V8      481: PZ = V9
482: AZ = A3(S3,P3)      483: EZ = E3(S3,P3)
484: RETURN
485: "VP"
    save level 3
486: VM = P2      487: VN = S2      488: V1 = LY
489: V3 = P3      490: V4 = S3      491: V5 = PZ
492: RETURN
493: "VQ"
    restore level 3
494: P2 = VM      495: S2 = VN      496: LY = V1
497: P3 = V3      498: S3 = V4      499: PZ = V5
500: AZ = A3(S3,P3)      501: EZ = E3(S3,P3)
502: RETURN
503: "RR"
    error
504: PRINT "ERROR: ";NX$
505: END
506: "RA"
    read AZ,EZ from Z(A,B) and store on AB,EB and get LB=LY
507: IF AF=0 GOSUB "UB"
508: IF AF=1 GOSUB "WB"
509: AF = 1      510: GOSUB "ZI"
    save Z(A,R)
511: GOSUB "VA"      512: AB = AZ      513: EB = EZ
514: LB = LY      515: RETURN
516: "RZ"
    read AZ,EZ from Z(C,D) and store on AD,ED and get LD=LY
517: IF CF=0 GOSUB "UD"
518: IF CF=1 GOSUB "WD"
519: CF = 1      520: GOSUB "ZI"
    save Z(C,D)
521: GOSUB "VC"      522: AD = AZ      523: ED = EZ
524: LD = LY      525: RETURN
526: "VZ"
    write AZ,EZ to Z(C,D), save Z(C,D)
527: AB = AZ      528: EB = EZ
529: IF CF=0 GOSUB "UD"
530: IF CF=1 GOSUB "WD"
531: CF = 1
532: GOSUB "ZE"      533: GOSUB "NC"      534: RETURN
*****
sort polynomial NX$
535: "SO"
    level 1 (no sorting)
536: GOSUB "X2"      537: IF P1=0 GOTO "SX"
538: GOSUB "X1"      539: IF LX=0 GOTO "SX"
    goto sort on level 3, release space on level 3
540: GOSUB "SY"
    goto sort on level 2, identical products
541: GOSUB "X1"      542: GOSUB "SP"
    goto sort on level 2, different products
543: GOSUB "X1"      544: MX = 0      545: GOSUB "SQ"
    store new length of polynomial
    and release space on level 2 and on level 3
546: GOSUB "X1"      547: L1(P1) = MX
548: IF MX=0 THEN LET A1(P1) = 0      549: GOSUB "SH"
550: "SX"
551: RETURN

```

```

.....
sort polynomial NX$ on level 3 (sort within products)
552: "SY"
    level 2, next Y(A,B), save Y(A,B)
553: IF LX=0 RETURN
554: GOSUB "Y1"
556: IF LY=0 GOTO "SY"
558: "SZ"
    level 3
559: IF LY=0 GOTO "SD"
560: GOSUB "Z1"
561: GOSUB "SA"
562: GOTO "SZ"
563: "SA"
    save Z(A,B), sort within product (sort on level 3)
564: GOSUB "WA"
565: AB = AZ
566: EB = EZ
567: IF EB=0 THEN LET AB = 0
    find largest multiplier
568: "SB"
569: IF LY=0 GOTO "SC"
571: IF EZ=0 GOTO "SB"
572: IF AB>AZ GOTO "SB"
573: A3(S3,P3) = AB
574: E3(S3,P3) = EB
575: AB = AZ
576: EB = EZ
577: IF AB>A3(S3,P3) GOTO "SB"
578: EB = EB + E3(S3,P3)
579: E3(S3,P3) = 0
580: GOTO "SB"
581: "SC"
    restore Z(A,B) and store largest multiplier
582: GOSUB "WB"
583: A3(S3,P3) = AB
584: E3(S3,P3) = EB
    check for internal errors (negative exponents)
585: IF AB>0 AND EB<0 END
586: IF AB>0 THEN LET MY = MY + 1
587: RETURN
588: "SD"
    store new length of product, release space on level 3
    level 2, restore Y(A,B)
589: GOSUB "UB"
590: L2(S2,P2) = MY
591: IF MY=0 THEN LET A2(S2,P2) = 0
592: "SE"
    level 3
593: IF LY=0 GOTO "SY"
594: GOSUB "Z1"
595: IF S3=1 GOSUB "Z3"
596: MY = MY - 1
597: GOTO "SE"
*****
combine identical products within polynomial NX$ (level 2)
598: "SP"
    first step of sort on level 2, find identical products
    sweep through all Y(A,B)
599: IF LX=0 RETURN
600: GOSUB "Y1"
601: LB = LY
602: LC = LX
    save Y(A,B) and first Y(C,D)
603: GOSUB "UA"
604: GOSUB "UC"
    first step of sort on level 2, find identical products
605: GOSUB "TA"
606: GOSUB "UB"
607: GOTO "SP"
608: "TA"
    A,B for first product and C,D for second product
    enter with Y(A,B), sweep through all Y(C,D)
609: IF LC=0 RETURN
610: GOSUB "UD"
611: GOSUB "Y1"
    save Y(C,D), restore Y(A,B), Comparison Flag CF
612: GOSUB "UC"
613: LC = LX
614: LD = LY
615: GOSUB "UB"
616: CF = 0
617: IF LB=LD GOTO "TB"
618: GOTO "TA"
619: "TB"
    compare products on level 3, enter with Y(A,B),
    read Y(C,D), sweep through all Z(A,B), Z(C,D)
620: IF LY=0 GOTO "TC"
621: GOSUB "Z1"
    save Z(A,B), read AD=AZ, ED=EZ, LD=LY from Z(C,D)
622: GOSUB "VA"
623: GOSUB "RZ"
624: GOSUB "WB"
625: IF AD=AZ AND ED=EZ GOTO "TB"
626: GOTO "TA"
627: "TC"
    combine identical products on level 2
628: GOSUB "UD"
629: VD = VY
630: V2(S2,P2) = 0
631: GOSUB "UB"
632: V2(S2,P2) = VY + VD
633: GOTO "TA"
*****
second step of sort, different products of NX$ on level 2
634: "SQ"
    sweep through all Y(A,B)
635: IF LX=0 RETURN
636: GOSUB "Y1"
637: LB = LY
638: LC = LX
    save Y(A,B) and first Y(C,D)
639: GOSUB "UA"
640: GOSUB "UC"
    attach products on level 2
641: GOSUB "TP"
642: GOSUB "UB"
643: IF ABS(VY)>0 THEN LET MX = MX + 1
644: GOTO "SQ"
645: "TP"
    A,B for first product and C,D for second product
    enter with Y(A,B), sweep through all Y(C,D)
646: IF LC=0 RETURN
647: GOSUB "UD"
648: GOSUB "Y1"
    save Y(C,D)
649: GOSUB "UC"
650: LC = LX
651: VD = VY
652: LD = LY
653: PD = PZ
654: GOSUB "UB"
655: VD = VY
656: LB = LY
657: PB = PZ
    cases: VY=0, LB,LD not equal, products not equal
658: IF VD=0 GOTO "TR"
659: IF VY=0 GOTO "TR"
660: CF = 0
661: "TR"
    compare products on level 3, enter with Y(A,B),
    read Y(C,D), sweep through all Z(A,B), Z(C,D)
    cases:
    LB,LD not equal, products not equal
662: IF LD=0 GOTO "TR"
663: IF LY=0 GOTO "TR"
664: GOSUB "Z1"
    save Z(A,B), read AD=AZ, ED=EZ, LD=LY from Z(C,D)
665: GOSUB "VA"
666: GOSUB "PZ"
667: GOSUB "WB"
668: IF AD<AZ GOTO "TP"
669: IF AD>AZ GOTO "TR"
670: IF ED<EZ GOTO "TP"
671: IF ED>EZ GOTO "TR"
672: GOTO "TQ"
    exchange different products on level 2
674: GOSUB "UD"
675: LD = LY
676: V2(S2,P2) = VY
677: L2(S2,P2) = LB
678: A2(S2,P2) = PB
679: GOSUB "UB"
680: V2(S2,P2) = VD
681: L2(S2,P2) = LD
682: A2(S2,P2) = PD
683: GOTO "TP"
*****
release space on level 2 and on level 3
684: "SR"
    level 2, save B2(P2), level 3
685: IF LX=0 GOTO "ST"
686: GOSUB "Y1"
687: BQ = B2(P2)
688: IF MX<1 GOSUB "Yo"
689: IF MX<1 GOSUB "DZ"
690: B2(P2) = RQ
691: IF S2=1 GOSUB "Y3"
692: MX = MX - 1
693: GOTO "SR"
694: "ST"
    empty is replaced by zero
695: GOSUB "X1"
696: IF LX=0 RETURN
697: GOSUB "YE"
698: GOTO "ST"
*****
copy or add polynomial NB$ to ND$ or subtract
699: "CO"
    sign SI=+1 for copy or add polynomial NB$ to ND$
700: SI = +1
701: GOSUB "CX"
702: RETURN
703: "SU"
    sign SI=-1 for subtract polynomial NB$ from ND$
704: SI = -1
705: GOSUB "CX"
706: RETURN
707: "CX"
    copy or add polynomial NB$ to ND$ or subtract
    find NB$, read first Y(A,B), save Y(A,B)
708: NX$ = NB$
709: GOSUB "X2"
710: IF P1=0 GOTO "RR"
711: GOSUB "X1"
712: IF LX=0 GOTO "RR"
713: GOSUB "Y1"
714: GOSUB "UA"
715: VB = VY
    find ND$ or create ND$
716: NX$ = ND$
717: GOSUB "X2"
718: IF P1=0 GOSUB "XC"
719: GOSUB "X1"
720: "CC"
    find last term Y(C,D) of polynomial ND$, read all Y(C,D)
721: IF LX=0 GOTO "CY"
722: GOSUB "Y1"
723: GOTO "CC"
724: "CY"
    copy elements Y(A,B) of NB$ to Y(C,D) of ND$, sign SI
725: GOSUB "YE"
726: V2(S2,P2) = VB + SI
    save Y(C,D), Copy Flag for read Z(A,B) and write Z(C,D)
727: GOSUB "UC"
728: GOSUB "UB"
729: CF = 0
730: GOSUB "CZ"
731: GOSUB "UB"
732: IF LX=0 RETURN
    read next Y(A,B)
733: GOSUB "Y1"
734: GOSUB "UA"
735: VB = VY
736: GOSUB "UD"
737: GOTO "CY"
738: "CZ"
    copy elements Z(A,B) to Z(C,D)
739: IF LY=0 RETURN
    read Z(A,B) and write AZ,EZ to Z(C,D)
740: GOSUB "Z1"
741: GOSUB "WA"
742: GOSUB "WZ"
743: GOSUB "WB"
744: GOTO "CZ"
*****
multiply polynomial NB$ by single product NP$, store on ND$
745: "HU"
    find NP$, read first Y(P,Q), save Y(P,Q), get factor SI
746: NX$ = NP$
747: GOSUB "X2"
748: IF P1=0 GOTO "RR"
749: GOSUB "X1"
750: IF LX=0 GOTO "RR"
751: GOSUB "Y1"
752: GOSUB "UP"
753: SI = VY
    find NB$, read first Y(A,B), save Y(A,B)
754: NX$ = NB$
755: GOSUB "X2"
756: IF P1=0 GOTO "RR"
757: GOSUB "X1"
758: IF LX=0 GOTO "RR"
759: GOSUB "Y1"
760: GOSUB "UA"
761: VB = VY
    write ND$
762: NX$ = ND$
763: GOSUB "XC"
764: "HV"
    copy elements Y(A,B) of NB$ to Y(C,D) of ND$
765: GOSUB "YE"
766: V2(S2,P2) = VB + SI
    save Y(C,D), Copy Flag for read Z(A,B) and write Z(C,D)
767: GOSUB "UC"
768: GOSUB "UB"
769: CF = 0
770: GOSUB "HW"
771: GOSUB "UB"
772: IF LX=0 RETURN
    read Y(A,B), restore Y(C,D)
773: GOSUB "Y1"
774: GOSUB "UA"
775: VB = VY
776: GOSUB "UD"
777: GOTO "HV"
778: "HW"
    copy elements Z(A,B) to Z(C,D)
    if LY=0 go first to "add single product", then return
779: PF = 0
780: IF LY=0 GOTO "MP"
    read Z(A,B), write AZ,EZ to Z(C,D)
781: GOSUB "Z1"
782: GOSUB "WA"
783: GOSUB "WZ"
784: GOSUB "WB"
785: GOTO "HW"
786: "MP"
    add single product, then return
    read Y(P,Q), read Z(P,Q), write AZ,EZ to Z(C,D)
787: IF PF=0 GOSUB "UQ"
788: PF = 1
789: IF LY=0 RETURN
790: GOSUB "Z1"
791: GOSUB "WP"
792: GOSUB "WZ"
793: GOSUB "WQ"
794: GOTO "MP"
*****
divide single product NB$ by single product ND$, result NP$
795: "DS"
    write NP$, save Y(P,Q)

```

```

796: NX$ = NF$
797: GOSUB "XC"
find NB$, save Y(A,B), get sign SI
800: NX$ = NF$
801: GOSUB "X2"
802: IF P1=0 GOTO "DD"
803: GOSUB "X1"
804: IF LX=0 GOTO "DD"
805: GOSUB "Y1"
806: GOSUB "UA"
807: LB = LY
808: SI = +1
809: IF VY<0 THEN LET SI = - SI
810: VB = ABS(VY)
find ND$, save Y(C,D)
811: NX$ = NC$
812: GOSUB "X2"
813: IF P1=0 GOTO "DD"
814: GOSUB "X1"
815: IF LX=0 GOTO "DD"
816: GOSUB "Y1"
817: GOSUB "UC"
818: LD = LY
819: IF VY<0 THEN LET SI = - SI
820: VD = ABS(VY)
821: IF VD=0 GOTO "DD"
merely integers, no reals, write result to Y(P,Q)
822: IF VD=INT(VB/VD)<VB GOTO "DD"
823: GOSUB "UQ"
824: V2(S2,P2) = SI * VB / VD
flags: AF read Z(A,B), CF read Z(C,D), PF read Z(P,Q)
825: AF = 0
826: CF = 0
827: PF = 0
828: "DA"
now level 3 with Z(A,B), Z(C,D), Z(P,Q) and with return
829: IF LB=0 AND LD=0 RETURN
830: IF LB=0 GOTO "DD"
read AB=AZ, EB=EZ, LB=LY from Z(A,B)
if LD=0 copy all Z(A,B) to Z(P,Q)
831: GOSUB "RA"
832: IF LD=0 GOTO "DB"
read AD=AZ, ED=EZ, LD=LY from Z(C,D)
833: GOSUB "R2"
834: "DP"
if AB>AD copy AB,EB to Z(P,Q), goto next Z(A,B)
835: IF AB<AD GOTO "DD"
836: IF AB=AD GOTO "DQ"
837: GOSUB "DC"
838: IF LB=0 GOTO "DD"
read AB=AZ, EB=EZ, LB=LY from Z(A,B)
839: GOSUB "RA"
840: GOTO "DP"
841: "DQ"
if AB=AD write AB,EB to Z(P,Q), goto next Z(A,B), Z(C,D)
842: EB = EB - ED
843: IF EB<0 GOTO "DD"
if EB=0 goto next Z(A,B), next Z(C,D)
844: IF EB=0 GOTO "DA"
if EB>0 write AB,EB to Z(P,Q), goto next Z(A,B), Z(C,D)
845: GOSUB "DC"
846: GOTO "DA"
847: "DB"
if LD=0 copy all Z(A,B) to Z(P,Q)
848: GOSUB "DC"
849: IF LB=0 GOTO "DA"
read AB=AZ, EB=EZ, LB=LY from Z(A,B)
850: GOSUB "RA"
851: GOTO "DB"
852: "DC"
write AB,EB to Z(P,Q)
853: IF PF=0 GOSUB "UQ"
854: IF PF=1 GOSUB "VQ"
855: PF = 1
856: GOSUB "ZE"
857: GOSUB "WP"
858: RETURN
859: "DD"
divide not possible, delete NP$
860: PRINT " DIV IMPOSSIBLE"
861: NX$ = NF$
862: GOSUB "DE"
863: NP$ = ""
864: RETURN
+++++
divide "polynomial NY$ / polynomial NZ$ ; "
865: "DV"
866: NC$ = "$COPY"
867: NR$ = "$RESULT"
868: NS$ = "$SCRATCH"
869: NT$ = "$TEMP"
copy polynomial NY$ to NC$
subroutine CO adds polynomial NB$ to ND$
870: NX$ = NC$
871: GOSUB "DE"
872: NB$ = NY$
873: ND$ = NC$
874: GOSUB "CO"
open polynomial NR$, save Y(C,D)
875: NX$ = NR$
876: GOSUB "XC"
877: GOSUB "YE"
878: GOSUB "UC"
879: IF NY$=NZ$ THEN LET V2(S2,P2) = 1
880: IF NY$=NZ$ GOTO "DN"
search polynomial NC$ for address P1 of name NZ$,
add product divided by NZ$ to NR$
881: NX$ = NZ$
882: GOSUB "X2"
883: AQ = P1
read Y(A,B) from NC$, flag AF for read Y(A,B)
884: NX$ = NC$
885: GOSUB "X2"
886: GOSUB "X1"
887: AF = 0
888: "DI"
read Y(A,B), save Y(A,B)
889: IF AF=1 GOSUB "UB"
890: AF = 1
891: IF LX=0 GOTO "DL"
892: GOSUB "Y1"
893: GOSUB "UA"
894: "DJ"
read all Z(A,B) and search for address of name NZ$
895: IF LY=0 GOTO "DI"
896: GOSUB "Z1"
897: IF AZ=AQ GOTO "DK"
898: GOTO "DJ"
899: "DK"
transfer product divided by NZ$ from NC$ to NR$
divide by NZ$
900: E3(S3,P3) = E3(S3,P3) - 1
901: GOSUB "UB"
copy single Y(A,B) to Y(C,D) and all Z(A,B) to Z(C,D)
902: VB = VY
903: GOSUB "UD"
904: GOSUB "YE"
905: V2(S2,P2) = VB
906: GOSUB "UC"
Copy Flag CF for read Z(A,B) and write Z(C,D)
907: CF = 0
908: GOSUB "UB"
909: GOSUB "C2"
delete product (single Y(A,B) with all Z(A,B) in NC$
and goto next Y(A,B)
910: GOSUB "UB"
911: EQ = S2(P2)
912: GOSUB "Yo"
913: GOSUB "D2"
914: S2(P2) = BQ
915: GOTO "DI"
916: "DL"
continue with divide "polynomial NY$ / polynomial NZ$ ; "
sort NC$ and NR$
917: NX$ = NC$
918: GOSUB "SO"
919: NX$ = NR$
920: GOSUB "SO"
if LX=0 and VY=0 on NC$ delete NC$, print NP$, goto "IT"

```

```

921: NX$ = NC$
922: GOSUB "X2"
923: GOSUB "X1"
924: GOSUB "Y1"
925: IF LX=0 AND VY=0 GOTO "DM"
divide single product NC$ by single product NZ$, get NS$
routine D3 divides single product NB$ by ND$, gets NP$
926: NB$ = NC$
927: ND$ = NZ$
928: NP$ = NS$
929: GOSUB "DS"
if NP$="" delete NC$, NR$, print message, return to "IT"
930: IF NP$="" OR NP$="" GOTO "DH"
multiply polynomial NZ$ by single product NS$, get NT$
subroutine HU multiplies NB$ by product NP$ and gets ND$
931: NB$ = NZ$
932: NP$ = NS$
933: ND$ = NT$
934: GOSUB "HU"
add single product on NS$ to NR$
subroutine CO adds polynomial NB$ to ND$
935: NB$ = NS$
936: ND$ = NR$
937: GOSUB "CO"
subtract polynomial NT$ from NC$
subroutine SU subtracts polynomial NB$ from ND$
938: NB$ = NT$
939: ND$ = NC$
940: GOSUB "SU"
delete NS$ and NT$
941: NX$ = NS$
942: GOSUB "DE"
943: NX$ = NT$
944: GOSUB "DE"
945: GOTO "DL"
946: "DM"
if NP$="" delete NC$, NR$, print message, return to "IT"
947: NX$ = NC$
948: GOSUB "DE"
949: NX$ = NR$
950: GOSUB "DE"
divide failed, return to input "IT"
951: PRINT " DIVIDE FAILED"
952: GOTO "IT"
953: "DN"
if LX=0 and VY=0 on NC$ delete NC$, print NR$, goto "IT"
954: NX$ = NC$
955: GOSUB "DE"
divide successful, result on NR$
956: NX$ = NR$
957: GOSUB "OT"
958: NX$ = NR$
959: GOSUB "DE"
960: GOTO "IT"
+++++

```

MATHEMATISCHE OPERATIONEN AUF DER EBENE VON SYMBOLISCHEN ELEMENTEN
an Beispiel der Division zweier beliebiger Polynome

La revue des

SHARPENTIER S

DANS CE NUMERO
GEREZ VOTRE COMPTE
LA NOUVELLE RUBRIQUE 1600
3D LABY SUR 800

ET TOUJOURS :
L'INITIATION, LES GRAPHIQUES, LE LM...

N°17 - BIMESTRIEL - MARS 86 - 30 F.
LA REVUE DES UTILISATEURS DE MICRO-ORDINATEURS ET PC/XT/COMPATIBLE SHARP

n°21 - JANVIER - FEVRIER 87

ISSN 0296-3477

CODES ASSEMBLEUR

Le PC-1600 comporte deux microprocesseurs : le SC-7852 compatible Z-80 et le LH-5803. Dans cet article, on va comparer les codes assembleur du Z-80 et du SC-7852 puis ceux du LH-

5803 et du LH-5801 du PC-1500.

D'abord le SC-7852 peut gérer huit pages mémoires de 0 à 7 et possède donc plusieurs instructions en plus par rapport au Z-80. Les codes du Z-80 se trouvant très facilement dans les livres d'informatique, je ne les donnerai pas dans cet article. Les instructions supplémentaires du SC-7852 correspondent aux fonctions RST du Z-80.

- &C7 - Code du reset du SC-7852 et enclenche la routine d'initialisation.
- &CF - Ne correspond à aucune fonction.
- &D7 - Ne correspond à aucune fonction.

&DF 010h - 01 octet faible de l'adresse de saut
0h octet fort de l'adresse de saut.

Exécute un saut inconditionnel à l'adresse 010h (JP 010h) en page 6 ou 3. Cette instruction est suivie de deux octets définissant à la fois l'adresse de saut et le numéro de page.

- Si Oh > &80 le saut s'effectue en page 6 à l'adresse 010h
- Si Oh < &80 le saut s'effectue en page 3 à l'adresse 010h.

Je rappelle que la mémoire morte programme de la page 3 va de &4000 à &7FFF, et celle de la page 6 va de &8000 à &BFFF.

&E7 a 010h :
- 01 octet faible de l'adresse de saut
- 0h octet fort de l'adresse de saut
- a numéro de la page où s'effectue le saut.

Exécute un CALL à l'adresse 010h de la page a.

Au premier RET code &C9 le programme continue son exécution dans la page où se situe le CALL.

- &EF - ne correspond à aucune fonction.
- &F7 - effectue un saut à l'adresse &F0D4 en page 0 (JP &F0D4).
- &FF - effectue un saut à l'adresse &F0D7 en page 0 (JP &F0D7).

Maurice CHOUCROUN

NEWS

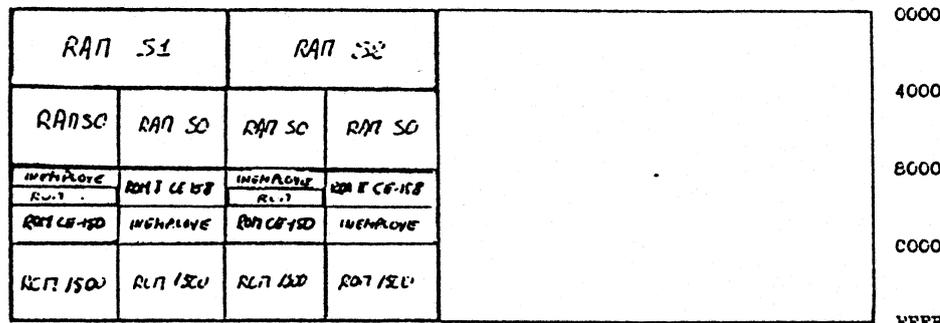
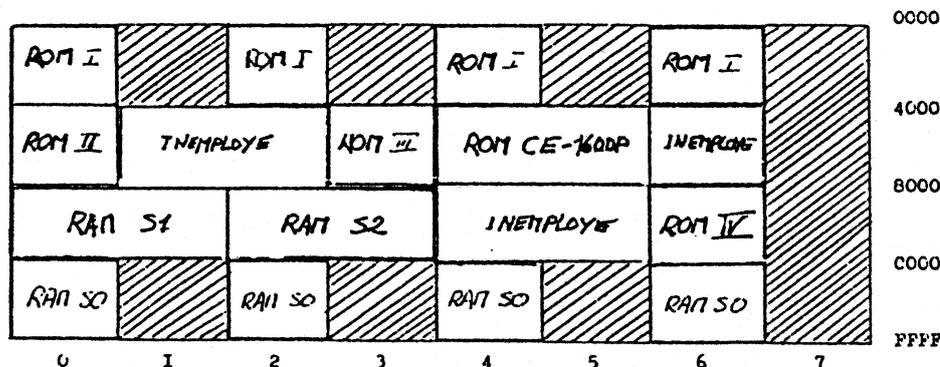
Si le manuel nous donne une carte de la mémoire du 1600, celle-ci est assez incomplète comme vous avez dû le constater vous-même.

Voici la carte telle que j'ai pu l'établir :

possesseur d'un PC-1500 depuis le mois de mai 1982, j'ai fini par tomber sous le charme du 1600 en le voyant à Paris début juillet. Après quatre années de bons et loyaux services, le PC-1500 se voyait donc remplacé par le petit nouveau. C'est pour vous parler de cela que je vous écris.

Je dirais tout d'abord que c'est une machine fantastique qui est à la fois plus puissante que le 1500, plus rapide et plus souple. Que demander de mieux ? Ma seule déception concerne la compatibilité que je trouve relative avec le 1500, mais j'y reviendrai plus tard.

Je voudrais tout d'abord vous donner quelques astuces que j'ai découvertes durant un mois d'utilisation.



là que j'ai un problème que vous pouvez sans doute résoudre : on peut compter 4 zones de ROM de 16 ko chacune dans l'aire du Z-80A, cela ne fait que 64 ko et non 96 ko comme indiqué par SHARP. Comme il n'y a pas de raisons de douter de ce que dit SHARP, la question se pose : où sont les 32 ko manquants ? Ce ne sont pas ceux de la ROM de la E-1600P puisqu'ils sont dans la périphérie et non dans l'unité centrale.

faut alors peut-être compter la ROM dressée par le LH 5803. On constate sur la carte que celle-ci est constituée de 16 ko qui constituent la ROM du 500 et de 4 ko supplémentaires. Cela nous fait que 20 ko. On arrive donc à trouver 80 ko de ROM dans le 1600. Il m'en manque 16. Les avez-vous trouvés ?

vous avouerez que lorsque j'ai découvert que la ROM adressée par le LH 5803 était celle du 1500 à quelques caractères près, j'ai trouvé ça génial : réussir à mettre un 1600 et un 1500 avec un processeur et sa ROM sous un même identique au vieux 1500, cela donne une idée de la différence de

technologie entre les deux machines. Je me suis alors dit que les logiciels que j'avais écrits et qui utilisaient des possibilités avancées du 1500 allaient pouvoir tourner sur le 1600 puisqu'en mode I c'était la ROM du 1500 qui était adressée par le LH-5803. Fort de cette certitude, je lance un programme utilisant la possibilité de détournement du driver clavier et... rien !! Histoire de vérifier je tente la manœuvre décrite dans le « Sharpentier » n° 10, et toujours rien. Me suis-je mal débrouillé où est-il impossible d'utiliser cette astuce ? Le désassemblage de la ROM à l'adresse E243 est pourtant identique à celui du 1500. Je veux bien que le clavier soit scruté au travers de la ROM spécifique au 1600, mais alors pourquoi avoir pris la peine de remettre cette routine dans la ROM ? Déçu, je charge mon logiciel de jeu préféré, écrit en collaboration avec un copain, et qui utilise pour l'affichage des messages de caractères redéfinis. C'est une horreur !!! J'ai droit aux symboles graphiques de l'IBM-PC mais pas à mes caractères. Là encore, même avec un autre logiciel utilisant cette intéressante possibilité du 1500, échec ! Ce

qui met à mal la compatibilité annoncée par SHARP ! Je ne vous surprendrai pas en vous disant que l'astuce permettant la réassignation du clavier avec ht ne fonctionne pas, etc. Je n'ai pas encore eu le temps d'essayer un logiciel basé sur la routine d'interruption publié dans le « Sharpentier » n° 6, mais j'ai peur ! Je vais chercher à faire fonctionner tout ça malgré ces premiers échecs mais je voudrais savoir si vous-même n'avez pas déjà trouvé comment faire. Il serait dommage que les logiciels les plus avancés, donc souvent les plus intéressants, ne fonctionnent pas sur le 1600.

Concernant le LM j'ai eu une autre surprise. Je charge le logiciel « PLANET FIGHTER » publié dans un « Sharpentier » et je tape : XCALL &4100. Au vu de la rapidité de déplacement de l'avion de combat, j'ai cru avoir à faire à un logiciel BASIC sur PC-1211 !!! L'intérêt de ce logiciel sur le 1500 étant sa rapidité liée au LM c'est un comble !

En fait, tous les programmes LM LH-5803 me semblent tourner nettement moins vite sur le 1600 que sur le 1500 (en particulier Planet). Cela est

étonnant car si l'on en croit la notice, le LH-5803 tourne à 1.3 Mhz comme le LH-5801. Avez-vous une explication à me donner ?

Pour terminer cette partie LM, j'ajouterais que je souhaite que SHARP publie rapidement un TRM pour le 1600.

Jérôme GAUDIN

**UN
PROGRAMME
À
DÉVELOPPER ?
CONTACTEZ
NOUS !**

PC 1600

LE PORT D'ENTRÉE ANALOGIQUE

Le PC 1600, quelle détonnante machine !!!
Le coup-ci, on a vraiment fait un effort pour avoir un micro puissant dans « la poche » chez SHARP.

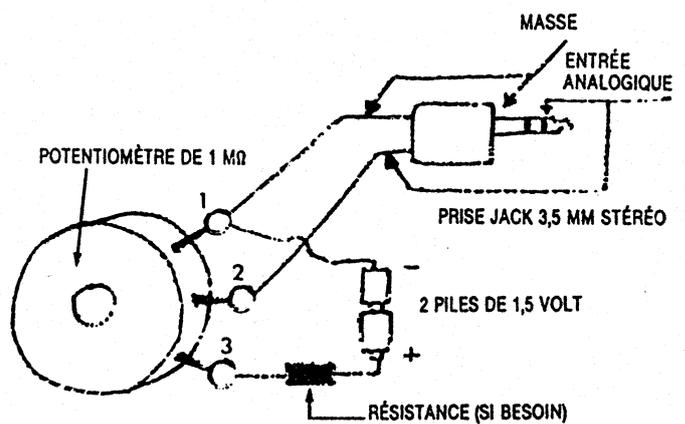
En effet, tout est prévu (d'après la documentation L.M) pour mettre dans le logement S2, 1 Mo de ROM ou 512 Ko de RAM sous forme de disque mémoire.

A quand les mémoires de 512 Ko de RAM dans un boîtier d'extension !!! Le PC 1600 possède aussi un port d'entrée analogique. La documentation est très mal faite à ce sujet. Voici un petit schéma pour s'amuser avec.

Pour vérifier le montage, s'assurer d'abord à l'aide d'un voltmètre que entre 1 et 3, il n'y a pas plus de 3 volts, sinon rajouter une résistance comme indiqué puis écrire :
10 A= AIN:P.A.:WAIT10:
CLS:G.10

Amusez-vous bien !

Olivier PROTTE

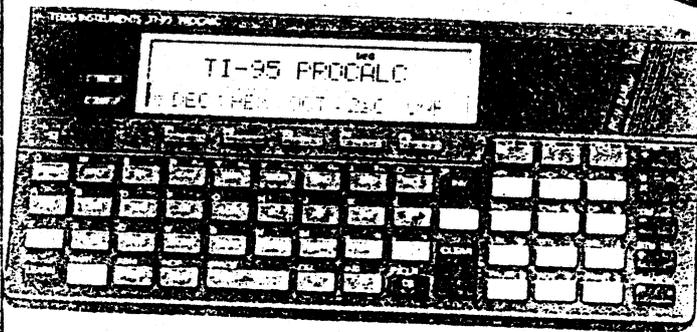


NANO- ORDINATEURS

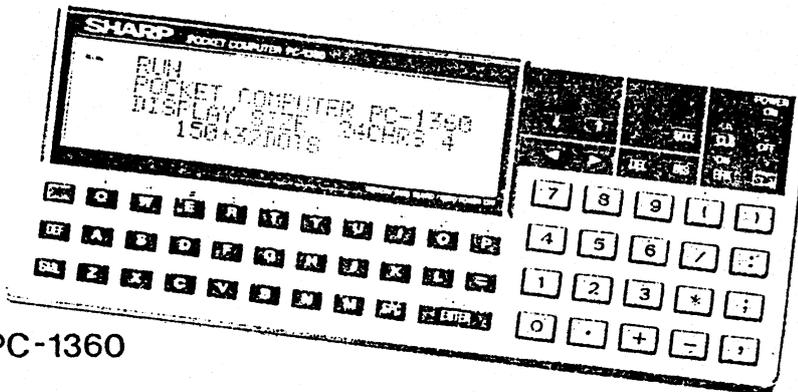
NOUVEAUTÉS 87



PB-1000



TI-95



PC-1360

30 F

ASTUCES POUR PC-1600

Une nouvelle rubrique apparaît dans le SHARPENTIER. Elle sera votre rubrique car je compte sur vous, les lecteurs pour alimenter le journal en astuces. Ainsi nous progresserons tous ensemble beaucoup plus vite dans la connaissance et dans l'utilisation de votre machine. Alors écrivez-nous...

- Bien que le PC-1600 soit doté de fonctions complémentaires SQR et PL, il est plus pratique d'utiliser un symbole unique π ou \sqrt pour ces deux fonctions. Ainsi en MODE O, le symbole [s'utilise comme \sqrt et] s'emploie comme π .
- Pour remplacer le GLCURSOR (X,Y) absent des fonctions graphiques de l'écran, on pourra faire :
 Xh Octet fort de X
 Xi Octet faible de X
 Yh Octet fort de Y
 Yi Octet faible de Y
 POKE&FCD4,Xi,Xh,Yi,Yh
 X et Y peuvent varier de -32768 à 32767
- Pour ceux qui ont une CE-150, il est possible d'imprimer les numéros de ligne après les GOTO, GOSUB et RESTORE en transformant votre programme en fichier ASCII.
 Ex. :
 SAVE "S2:XXX",A
 LOAD "S2:XXX"
 Dans l'exemple, on sauve le programme dans le module 2.
- Dans le MODE O du PC-1600, on peut simuler un appui sur la touche RCL en mode II par POKE&F127,PEEK&F127 OR 32.
- POKE&FOBA,O éteint le signal BATT
 POKE&FOBA,N avec N>O allume le signal BATT.

Maurice CHOUCROUN

LES PREMIÈRES ROUTINES

Avec l'aide d'une revue japonaise et un peu de recherche, nous avons découvert les premières routines en langage machine Z-80.

FONCTIONS	EXPLICATIONS	ADRESSES
CLS	Efface l'écran	&112
POS D,E	Met le curseur texte à la position D en abscisse et E en ordonnée	&115
PRINT A	Affiche le caractère ASCII contenu dans l'accumulateur et incrémente le curseur texte	&100
BEEP 1,A,BC	Émet un beep de hauteur A et de longueur le contenu du registre BC	&1B7
GETKEY A	Lis un caractère du buffer clavier ou se met en attente d'un caractère	&166
INKEY A	Lis un caractère du buffer clavier ou retourne 0 dans l'accumulateur	&169

Toutes ces routines modifient uniquement les registres AF et s'adressent en PAGE 0.

Maurice CHOUCROUN

BOMBARDIER

Attention, dès que vous aurez tapé ce programme, vous vous trouverez à bord d'un bombardier léger B-25 MITCHELL. Votre mission si vous l'acceptez consiste à détruire le plus de bâtiments

possible avant de vous écrasez. Vous ne pouvez lâcher qu'une seule bombe à la fois avec la touche (SPACE).

Maurice CHOUCROUN

```

10: CLS :RANDOM :
   CLEAR :DIM A$(
   0)*22
20: I=15:INPUT "N
   iveau(1-5)?" :0
   :CLS
30: L=RND 8+1:IF
   L+I>148GOTO 10
   0
40: H=RND (3*0)+1
   5:LINE (I,31)-
   (I+L,31-H),, &F
   3CF,BF:LINE (I
   -1,31)-(I-1,31
   -H)
45: LINE (I+L+1,3
   1)-(I+L+1,31-H
   ):I=I+L+3+RND
   4:GOTO 30
100: H=-3:L=0:A$(0
   )="0F090A0A0E0
   E0A0A0404"
110: L=L+2:IF L>15
   SLET H=H+1:L=0
112: GCURSOR L+1,H
   :GPRINT OR A$(
   0):LINE (L,H+3
   )-(L+1,H),R,,B
115: IF POINT (L+1
   2,H+3)+POINT (
   L+11,H+3)>0GOT
   0 230

```

```

117: IF B=1GOTO 20
   0
120: IF INKEY$ <>C
   HR$ 32BEEP 1,9
   ,20:GOTO 110
122: B=1:P=H-3:Q=L
   +5:BEEP 1
200: P=P+1:GCURSOR
   Q,P:GPRINT "4
   0A040":IF P=30
   LET B=0:GOTO 1
   10
205: IF POINT (Q,P
   +8)+POINT (Q+1
   ,P+8)+POINT (Q
   +2,P+8)=0GOTO
   110
210: IF P>26GURSO
   R Q-1,26:PRINT
   CHR$ 32:GOTO
   110
220: LINE (Q,P+3)-
   (Q+2,P+12),R,,
   BF:BEEP 1,30,5
   0:0=0:B=0:GOTO
   110
230: BEEP 2,30,50:
   WAIT :BEEP 3,3
   0,20:GCURSOR L
   +9,H:GPRINT "1
   4083E0814":END
STATUS 1= 827

```

Si n + m > 255 ou si le numéro de ligne n'existe pas, le programme ne produit aucune action.

Maurice CHOUCROUN

```

C000:3A 69 F8 LD A,(F869)
C003:F6 90 OR A,128
C005:67 LD H,A
C006:3A 6A F8 LD A,(F86A)
C009:6F LD L,A
C00A:7E LD A,(HL)
C00B:FE FF CP A,255
C00D:28 57 JR Z,&C006
C00F:7E LD A,(HL)
C010:23 INC HL
C011:5A CP A,0
C012:20 84 JR NZ,&C018
C014:7E LD A,(HL)
C015:5B CP A,E
C016:28 08 JR Z,&C020
C018:23 INC HL
C019:4E LD C,(HL)
C01A:06 00 LD B,0
C01C:09 ADD HL,BC
C01D:23 INC HL
C01E:18 EB JR,&C023
C020:23 INC HL
C021:E5 PUSH HL
C022:4E LD C,(HL)
C023:06 00 LD B,0
C025:59 LD E,C
C026:09 ADD HL,BC
C027:E5 PUSH HL
C028:23 INC HL
C029:5E LD A,(HL)
C02A:FE FF CP A,255
C02C:29 00 JP Z,&C02E
C02E:23 INC HL
C02F:23 INC HL
C030:7E LD A,(HL)
C031:47 AND A,A
C032:09 ADD A,E
C033:30 00 JR C,&C035
C035:23 INC HL
C036:01 POP DE
C037:7E LD A,(HL)
C039:12 LD (E),A
C039:FE FF CP A,255
C03B:20 04 JR Z,&C041
C03D:23 INC HL
C03E:13 INC DE
C03F:18 F5 JR,&C037
C041:3A 67 F8 LD A,(F867)
C044:F6 08 OR A,128
C046:5C CP A,H
C047:20 F4 JR NZ,&C050
C049:39 68 F8 LD A,(F859)
C04C:8D CP A,L
C04D:20 EE JR NZ,&C03D
C04F:7A LD A,D
C050:EE 80 XOR A,129
C052:22 E7 F8 LD (F8E7),A
C055:78 LD A,E
C056:32 53 F8 LD (F853),A
C058:E1 POP HL
C05A:E5 PUSH HL
C05B:06 00 LD B,0
C05D:23 INC HL
C05E:04 INC B
C05F:7E LD A,(HL)
C060:FE FF CP A,255
C062:20 F9 JR NZ,&C05D
C064:E1 POP HL
C065:7E LD (HL),B
C066:13 RET
C067:E1 POP HL
C068:E1 POP HL
C069:C9 RET
C06A:00 NOP
C06B:00 NOP

```

UNE LIGNE

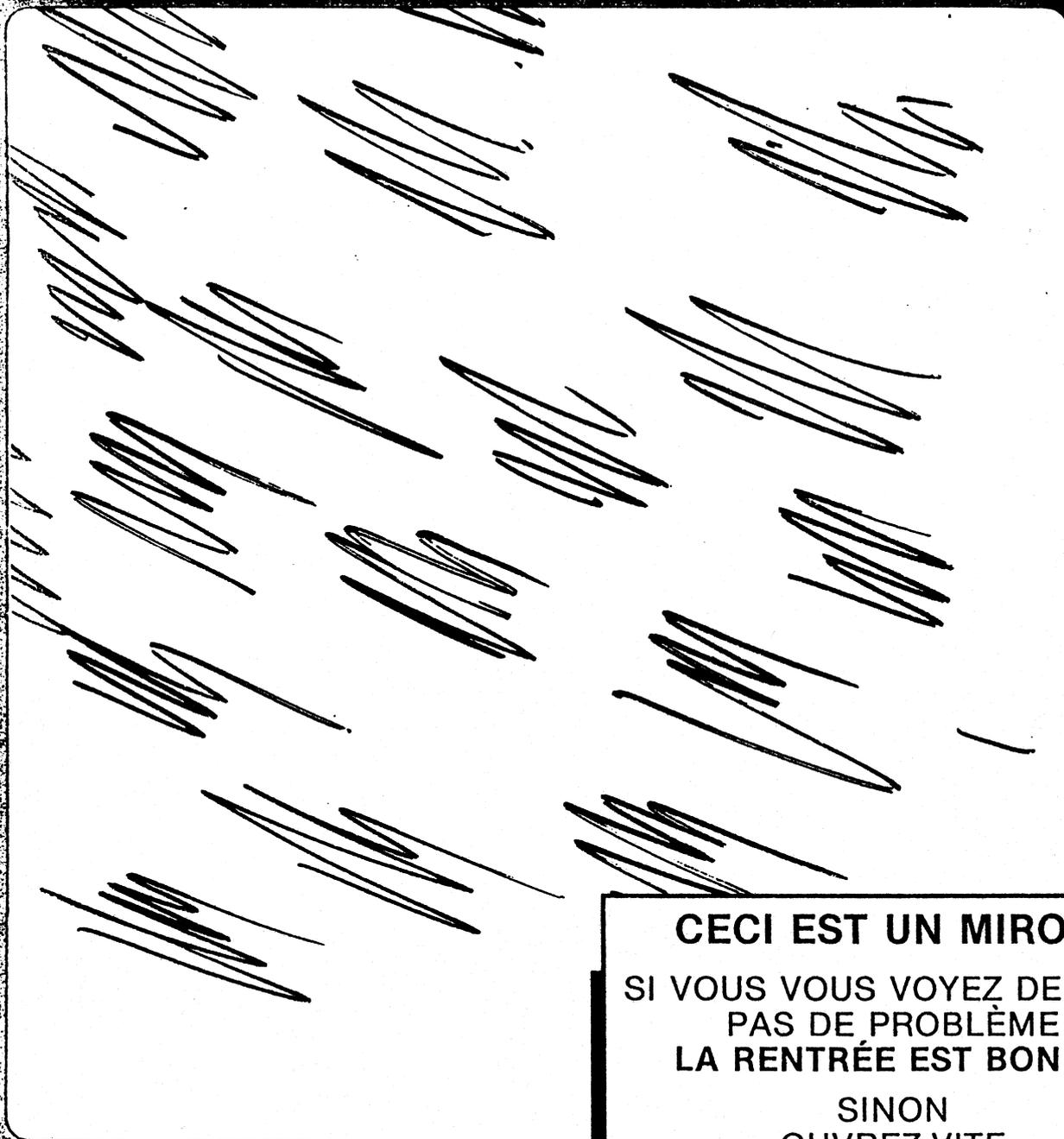
DE 255 CARACTÈRES

Le programme FUSION LIGNES permet de regrouper deux lignes pour en faire une de moins de 255 caractères.

On choisit deux lignes de n et m caractères avec n + m < 255. Puis, le numéro de la première ligne sera mis dans une variable (A par exemple) avant de faire CALL adresse, variable.

Une ligne de moins de 255 caractères sera formée, mais on ne pourra pas la corriger autrement qu'en la séparant en plusieurs lignes et en les regroupant à nouveau. Il est néanmoins possible de la lister et de l'exécuter.

NANO- ORDINATEURS



CECI EST UN MIROIR
SI VOUS VOUS VOYEZ DEDANS
PAS DE PROBLÈME
LA RENTRÉE EST BONNE
SINON
OUVREZ-VITE
CE NUMÉRO SPÉCIAL RENTRÉE
ET VOUS VERREZ...

1) ÉQUATION CARTÉSIENNE

Le PC permet de tracer 2 courbes sur une même échelle.

Si vous voulez tracer 2 courbes, entrez les fonctions F(X) et E(X).

Le PC autoprogrammera la ou les fonction(s). Appuyer sur ENTER si les fonctions étaient déjà introduites.

Vous devez donner ensuite la borne minimum et maximum de l'intervalle. L'ordinateur prend pour borne maximale le négatif de la borne minimum si vous appuyez directement sur ENTER quand le PC affiche "Xmax?".

Ceci reste valable pour l'entrée des intervalles des courbes paramétriques classiques.

Le PC vous propose ensuite une valeur pour l'infini. Le nombre 20 fonctionne généralement bien pour la plupart des courbes. Si vous l'acceptez, appuyez sur ENTER sinon entrez une autre valeur. Je rappelle que l'infini est la valeur pour laquelle tous les points dont l'ordonnée dépassant cette valeur prennent la valeur de l'infini. Ceci est très pratique pour voir des courbes de la forme exponentielle ou tout autre courbe très croissante ou décroissante. L'infini négatif et positif ont la même valeur absolue. Pour vous faire une idée de la notion d'infini, tracez les courbes des équations suivantes.

Intervalle [-180;180] F(X)=SINX
 INFINI 20 puis INFINI 0.7
 Intervalle [0;5] F(X)=EXPX
 INFINI 1E4 puis INFINI 2

ATTENTION : Ne pas confondre la borne maximum ou minimum de l'ordonnée avec l'infini. Le PC calcul tout seul ces bornes. La machine calcule ensuite les points de la courbe.

Il vous demande si vous voulez relier les points de la courbe.

Cela est utile pour les fonctions contenant des parties entières. Vous décidez aussi si vous voulez afficher les axes.

Le PC vous montre les points considérés comme minimum et maximum sur l'intervalle.

Appuyer sur ENTER pour tracer la courbe et afficher les échelles.

Sur la droite de l'écran apparaissent les coordonnées du point d'intersection des axes puis l'échelle sur les abscisses et les ordonnées.

Les axes, pour une meilleure lisibilité, sont calculés dans l'ordre de grandeur le mieux adapté pour la courbe.

La plupart du temps les axes sont [X=0] et [Y=0].

Appuyez à nouveau sur ENTER pour obtenir les valeurs des points de la courbe. Les flèches permettent de déplacer le point curseur sur l'écran. A

```

215: IF LPLINE -(I,K-A(I)*D):NEXT I:GOTO 271
225:PSET (I,K-A(I)*D)
270:NEXT I
271:O=0:IF U=1GOTO 302
272:IF B(O)=H+1LET O=O+1:GOTO 272
273:P=K-B(O)*D:LINE (O,P)-(O,P)
275:FOR I=O+1TO 100:IF B(I)=H+1GOTO 290
280:IF LPLINE -(I,K-B(I)*D):NEXT I:GOTO 302
285:PSET (I,K-B(I)*D)
290:NEXT I
300:IF O*XK=0LET M=0:GOTO 303
301:N=10^INT LOG (O-W).M=INT ((O+W)/2/N)*N-N
302:FOR I=1TO 10:M=M+N. IF M<WOR M>QNEXT I:M=M+W
303:R=K-M*D.G=M
305:IF J$="O"LINE (0,R)-(100,R)
310:IF A*B<0LET M=0:GOTO 325
320:U=10^INT LOG (B-A):M=INT ((A+B)/2/U)*U-U
322:FOR I=1TO 10:M=M+U:IF M<AOR M>BEXIT I:M=M+A
325:T=(M-A)/C
326:IF J$="O"LINE (T,0)-(T,31)
330:GOSUB 395:M=G:GOSUB 395:M=C*10:GOSUB 395:M=5/D:GOSUB 395:I=T:U=T
340:I=I+10:LINE (I,R-1)-(I,R+1):IF I<90GOTO 340
345:U=U-10:LINE (U,R-1)-(U,R+1):IF U>90GOTO 345
350:I=R:U=R
355:I=I+5:LINE (I-1,I)-(I+1,I):IF I<27GOTO 355
360:U=U-5:LINE (I-1,U)-(I+1,U):IF U>40GOTO 360
370:IF U=2LINE (O-1,P-1)-(O+1,P+1),,BF
380:IF G=0FOR I=2TO 100:A(I)=A(I)+1E4:A(I)=A(I)-1E4:IF U=2LET B(I)=B(I)+1E4:B(I)=B(I)-1E4
385:IF G=0NEXT I
387:Z=0:X=A:G=50:KBUFF$="":WAIT :CURSOR 25,3:PRINT :WAIT 0:FOR I=0TO 3:CURSOR 17,I:PRINT " ":NEXT I:GOTO 600
390:Y=H+1:E=Y:RESUME ERL +8
395:CURSOR 17,0C:OC=OC+1:IF ABS M=>PRINT " "+CHR$(44-SGN M)+CHR$ 236+" ":RETURN
396:IF M=>I:PRINT " Non def":RETURN
397:M$=STR$ M:IF ASC RIGHT$(M$,4)=69LET M$=LEFT$(M$,5)+"E"+RIGHT$(M$,3):PRINT M$:RETURN
398:IF LEN M$<SLET M$=" "+M$
399:PRINT LEFT$(M$," ",9):RETURN
400:CLS :U=100:H=20
408:IF A$="O"RESTORE 802:INPUT "P(T)=";B$:XCALL I
409:IF A$="O"GOTO 440
410:RESTORE 805:INPUT "X(T)=";B$:XCALL I:RESTORE 810:INPUT "Y(T)=";B$:XCALL I
440:INPUT "Tmin?";A:B=-A:INPUT "Tmax?";B
445:INPUT "Infini 20?";H
447:INPUT "Nb de pts (100)?";J
450:C=(B-A)/J,G=-1E99,L=T,A:ON ERROR GOTO 592
470:T=T+C:GOSUB 800:X=ABS X:Y=ABS Y:IF X>HLET X=H
472:IF X>LLET L=X
474:IF Y>HLET Y=H
476:IF Y>GLET G=Y
480:IF T<BGOTO 470
500:"B"=50/L:F=10/G:T=A:WAIT 0:CLS :ON ERROR GOTO 594
505:J$="O":INPUT "AXES (O/N)?";J$
510:M=0:CLS :GOSUB 800:X=EXX+49.5:Y=Y+Y+14.5:LINE (X,31-Y)-(X,31-Y):ON ERROR GOTO 595
520:T=T+C
521:M=X,N=Y:GOSUB 802:X=X+E+49.5+(X<0):Y=Y+14.5+(Y<0):IF X>100LET X=100
522:IF X<0LET X=0
523:IF Y<0LET Y=0
525:IF Y>31LET Y=31
530:IF ABS (X-M)>1.5IF ABS (Y-N)>1.5LET C=C/2:T=T-C:GOTO 521
535:LINE -(X,31-Y):C=C*1.3:IF T<BGOTO 520
580:IF J$="O"LINE (0,15)-(100,15):LINE (50,0)-(50,31)
582:FOR I=0TO 100STEP 5:LINE (I,14)-(I,15):NEXT I
    
```

droite, N représente la position du curseur sur l'écran. En dessous, est affichée l'abscisse puis l'ordonnée de F(X) et éventuellement E(X).

2) ÉQUATION PARAMÉTRIQUE OU POLAIRE

L'utilisation du traceur paramétrique ou polaire est équivalente.

L'équation polaire est automatiquement traduite en équation paramétrique. Après avoir entré les équations et les bornes de l'intervalle, le PC demande combien de points il doit calculer. Celui-ci vous propose 100 points.

Pour des courbes compliquées, il vaut mieux en calculer plus.

Il est inutile de mettre une variation du nombre de points inférieure à 20. En effet le programme est récurrent, il rajoutera ou supprimera éventuellement des points lorsque cela sera nécessaire. On évite ainsi les courbes hachurées par un nombre de points insuffisants ou la lenteur du calcul d'une abondance de points.

Maintenant, il ne vous reste plus qu'à maîtriser ce logiciel, ainsi qu'à découvrir la fonction réelle définie sur plus d'un point et intraquable par ce programme. Personnellement, je cherche encore...

Maurice CHOUCROUN

Taper le programme et passer en mode RUN.

Faites attention de bien mettre le même nombre de lettres D dans les lignes 2 et 3.

faites DEF V et entrez les valeurs en HEXA du programme en LM. Celui-ci sera introduit dans les lignes 2 puis 3. Pour finir, passez en mode PRO et tapez : DELETE 1000,1040.

```

585:FOR I=0TO 30STEP 5:LINE (49,!)-(51, I):NEXT I:00=0:M=L:GOSUB 395:M=G:GOSUB
395
590:M=L/10:GOSUB 395:M=G/3:GOSUB 395:KBUFF$="" :WAIT :CURSOR 25,3:PRINT :END
592:POKE &F891,255:RESUME 480
594:T=T+C:POKE &F891,255:RESUME 510
595:T=T-C:POKE &F891,255:RESUME 520
600:00=1:CURSOR 18,0:PRINT "N="+STR$ G+" " :M=X+C*G:GOSUB 395
605:M=A(G):GOSUB 395:P=K-M*D:IF U=2LET M=B(G):GOSUB 395:N=K-M*D
610:Z=Z+1:IF Z=10PSET (G,P),X:Z=0:IF INT P<INT NIF U=2PSET (G,N),X
615:S=ASC INKEY$ :IF S=0GOTO 610
622:IF U=2!F POINT (G,N)=2PSET (G,N),X
624:IF POINT (G,P)=2PSET (G,P),X
625:IF S=0LET G=G-(G>0):PSET (G,K-D+A(G)),X:IF U=2PSET (G,K-D+B(G)),X
630:IF S=12LET G=G-(G<100):PSET (G,K-A(G)+D),X:IF U=2PSET (G,K-B(G)+D),X
635:IF S=8OR S=12GOTO 620
640:IF S<>13GOTO 610
650:KBUFF$="" :WAIT :CURSOR 25,3:PRINT :END
800:IF A$="P"GOTO 805
802:P=T
804:X=P*COS T:Y=P*SIN T:RETURN
805:X=T
810:Y=T
820:RETURN
1000:"U"A=PEEK &F869*256+PEEK &F86A*256:0F:B=0
1010:FOR I=0TO &46:INPUT "Lig 2:Valeur?":G:POKE I+A,G:B=B-G
1020:NEXT I:A=A+77:IF B<>9700PRINT "ERREUR LIG 2":GOTO 1000
1030:FOR I=0TO &31:INPUT "Lig 3:Valeur?":G:POKE I+A,G
1040:NEXT I:IF B<>6397PRINT "ERREUR LIG 3":GOTO 1030

```

LIGNE 2

```

85 B0 AE 79 8B BE F9 57 48 7B 4A B0 45 B7 0D 99 05 04 DF AE 77 F0 FB B1 B0 AE 77
F1 A5 78 BE 2A A5 78 BF B9 7F 28 64 64 B5 03 61 FD A8 64 64 FD A8 FD 29 FD 5A 45
B7 0D 99 05 46 F5 05 B7 FF 99 06 F5 56 FD 2A 9E 06

```

LIGNE 3

```

FD 18 A5 77 F1 FD CA FD 89 57 43 A4 38 99 05 24 16 93 0A 15 0E FD 2A F6 78 87 48
7B 4A 90 FD 1A 56 A5 77 F1 F9 B3 03 51 54 54 F5 A5 77 F0 06 93 07 9A

```

NANO- ORDINATEURS



ma vie a mes

5 NOVEMBRE-DECEMBRE 1987 30 F

ÉDITEUR-ASS-DESS

Pour tous les programmes en LM, nous publions enfin un **ÉDITEUR-ASS-DESS**. Il fonctionne sur la version de base du PC-1600 et utilise les codes du Z-80. Comme ce logiciel est principalement fait en Basic, il ne traite que la première page du micro-processeur. En revanche il est suffisamment puissant pour palier ce défaut.

MODE D'EMPLOI :

RUN → INITIALISE LE PGM ET L'EXÉCUTE

... F A → EXÉCUTE LE PGM.

Le programme vous demande l'option que vous voulez.

L'éditeur, le désassembleur ou la liste des étiquettes.

Vous n'avez qu'à taper la lettre en parenthèse correspondante à votre choix.

ÉDITEUR

Quand le PC-1600 affiche le curseur, vous avez trois possibilités :

1) Vous pouvez taper &XXXX avec XXXX un nombre en hexadécimal ; le PC vous affiche la conversion en décimal.

2) Tapez * &XXXX ou *XXXXX et le PC transfère le curseur à l'adresse indiquée par le nombre hexa ou décimal

suivant que vous avez mis ou pas le &. D'ailleurs à chaque fois que vous mettez ce signe, le PC considérera le nombre qui suit comme hexadécimal.

3) Écrire une instruction ou une étiquette suivie d'une instruction.

Pour écrire une instruction, il faut respecter une certaine syntaxe.

Voici les syntaxes possibles pour quelques instructions. Les autres suivent le même modèle.

LD A,32 ou LD A,&20

LD HL,257 ou LD HL,&101

LD HL,&F000 ou LD HL,(61440)

LD A,(HL) et RL A et non pas

LD (HL) et RL

JR NZ,16 ou JR NZ,&10

CALL &F000

DJNZ 50 ou DJNZ,50 ou DJNZ &32

ou DJNZ,&32

JR10 ou JR,10 ou JR &0A

Tout cela est très classique. Il existe

en plus quelques instructions particulières où les mnémoniques ont été modifiées. L'utilisateur pourra toujours les remettre à son goût en changeant le tableau AS(*).

EX (SP),HL = > (SP) HL

JP nn EN PAGE 3/6 = > GTO nn

Pour adresser les autres pages, il suffit de mettre le code d'appel de la page puis d'introduire la suite de code hexa suivant ce code.

PAR EXEMPLE : [ED &102030405060

Le PC introduira en mémoire le code ED suivi par &10,&20,&30...

Les différents codes d'appel sont ED,CB,DD et FD.

L'éditeur peut bien évidemment gérer des étiquettes.

Il suffit d'introduire les étiquettes par les crochets [et].

EXEMPLE : COOO [AAA]LD A,5 définit l'étiquette AAA à l'adresse &C000.

On peut utiliser au maximum 15 étiquettes différentes imbriquées de n'importe qu'elle manière. On peut aussi redéfinir la même étiquette.

Pour utiliser une étiquette, il suffit de remplacer le paramètre d'une instruction par le crochet suivi de l'étiquette. EXEMPLE : JR [AAA ou LD HL,[AAA ou CALL [AAA ou JP [AAA ou JR NZ,[AAA...

La longueur des étiquettes est limitée à 16 caractères.

DÉSASSEMBLEUR

Fixez avant toute exécution le CSIZE de votre impression si vous utilisez l'imprimante. Prenez en général CSIZE 1 pour la CE-150 et 2 pour la CE-1600P. Je vous laisse découvrir le reste car les autres demandes du PC sont très clairement exprimées.

Le programme est assez souple d'utilisation et j'espère qu'il vous permettra de programmer en LM Z-80 aisément. N'oubliez pas de nous faire parvenir vos programmes.

Maurice CHOUCROUN

```

5: CLEAR : BLOAD "S2:DATA.LAT",PB,LE47
E: POKE $7899,$84,$7E: DIN B+10
: B+9,B,M+10: B+K+12
10: ATCLS : BUFFA "L": INPUT "Address
: ?": A
15: INPUT "(E)DIT,(D)ESS ou (E)TIC?":
G:CLS : GOTO 60
17: E:CLS
20: H="": BEEP 1,B: PRINT MEPS A:
INPUT " ?": M+10
25: FOR I=LEN M+10 TO 1 STEP -1: IF
MID$(M+10,I,1)=""LET M+RIGHT$(
M+10,LEN M+10)-I: B+LEFT$(M+
0),I-2: M+10=B+GOTO 25

```

```

22: H="": IF LEFT$(M+10,2)=""LET
C+RIGHT$(M+10,LEN M+10)-2:
GOSUB 100: B+GOTO 20
24: IF LEFT$(M+10,1)=""LET M+M+
RIGHT$(M+10,LEN M+10)-1: GOTO 2
B
25: FOR I=1 TO LEN M+10: IF MID$(M+10
),I,1=""NEXT I: B+M+10: GOTO 2
B
27: GOSUB 303
28: IF LEFT$(M,1)=""LET C+MID$(M
),2,LEN M-1: GOSUB 100: PRINT B:
GOTO 20
30: B="": FOR I=1 TO LEN M: C+MID$(M
),I,1: IF C=""OR C=""LET B+B
+MID$(M,I,1)

```


EA00:	41 20 44 00 00 00 53 55 42 20 41 20 45 00 00 00	: 59 14
EA01:	53 55 42 20 41 20 48 00 00 00 53 55 42 20 41 20	: 5F 77
EA02:	40 00 00 00 53 55 42 20 41 20 20 48 40 29 53 55	: 55 FA
EA03:	42 20 41 20 41 20 00 00 53 42 43 20 41 20 42 00	: 10 47
EA04:	20 00 53 42 43 20 41 20 43 00 00 53 42 43 20	: 65 36
EA05:	41 20 44 00 00 00 53 42 43 20 41 20 45 00 00 00	: 45 15
EA06:	53 42 43 20 41 20 48 00 00 00 53 42 43 20 41 20	: 2D 55
EA07:	41 00 00 53 42 43 20 41 20 20 48 40 29 53 42	: 44 E7
EA08:	43 20 41 20 41 00 00 00 41 4E 44 20 41 20 42 00	: 11 42
EA09:	00 00 41 4E 44 20 41 20 43 00 00 00 41 4E 44 20	: 50 35
EA0A:	41 20 44 00 00 00 41 4E 44 20 41 20 45 00 00 00	: 40 16
EA0B:	41 4E 44 20 41 20 48 00 00 00 41 4E 44 20 41 20	: 49 60
EA0C:	4C 00 00 00 41 4E 44 20 41 20 20 48 40 29 41 4E	: 3F E1
EA0D:	44 20 41 20 41 00 00 00 58 4F 52 20 41 20 42 00	: 12 06
EA0E:	00 00 58 4F 52 20 41 20 43 00 00 00 58 4F 52 20	: 65 5C
EA0F:	41 20 44 00 00 00 58 4F 52 20 41 20 45 00 00 00	: 59 24
EA10:	58 4F 52 20 41 20 49 00 00 00 58 4F 52 20 41 20	: CE F6
EA11:	4C 00 00 00 58 4F 52 20 41 20 20 48 40 29 58 57	: 65 01
EA12:	4F 52 48 20 20 20 20 42 41 53 20 00 00 00 00 00	: AE B4
EA13:	00 00 00 00 00 71 89 01 00 01 00 33 05 00 00 47	: 2A 9C
EA14:	52 41 53 48 49 43 20 42 41 53 20 00 00 00 00 00	: 19 B4
EA15:	00 00 00 00 00 4E 7C 00 00 0F 00 03 11 00 00 44	: E5 73
EA16:	4C 49 40 20 20 20 42 41 53 20 00 00 00 00 00	: 44 B4
EA17:	00 00 00 00 00 67 70 00 00 18 00 0A 00 00 00 54	: 5F 5D
EA18:	52 41 56 41 49 40 70 40 40 20 20 00 00 00 00 00	: 78 8D
EA19:	00 00 00 00 00 26 67 AF 0C 04 00 07 01 00 00 57	: 3C 1A
EA1A:	4F 52 48 20 20 20 20 40 20 20 20 00 00 00 00	: 89 8D
EA1B:	00 00 00 00 00 14 67 AF 2C 05 00 02 01 00 00 53	: 30 17
EA1C:	49 40 55 40 20 20 20 40 40 20 20 00 00 00 00 00	: E3 8D
EA1D:	00 00 00 00 00 00 59 21 00 00 00 52 00 00 00 53	: 7C 89
EA1E:	49 40 55 40 20 20 20 42 41 53 20 00 00 00 00 00	: 09 B4
EA1F:	03 00 00 00 00 58 5A 21 00 07 03 C2 00 00 00 46	: 65 1C
EA20:	4F 52 40 40 20 20 20 42 41 53 20 00 00 00 00 00	: 0C B4
EA21:	00 00 00 00 00 29 0A 21 00 00 00 F5 00 00 00 05	: 55 EE
EA22:	55 52 4F 50 45 20 20 47 41 53 20 00 00 00 00 00	: 8D B4

EA23:	00 00 00 00 00 00 61 03 21 0C 2E 00 00 01 00 00 15	: 65 08
EA24:	55 52 4F 44 41 54 20 40 40 20 20 00 00 00 00 00	: 38 8D
EA25:	00 00 00 00 00 00 03 10 22 0C 61 00 46 0A 00 00 00	: 8D A2
EA26:	43 52 41 4E 20 20 20 42 41 53 20 00 00 00 00 00	: 06 B4
EA27:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	: 02 CE
EA28:	45 52 52 45 20 50 20 42 41 53 20 00 00 00 00 00	: 00 B4
EA29:	00 00 00 00 00 00 5C 41 0C 6A 00 58 16 00 20 00	: 00 89
EA2A:	45 58 54 20 20 20 20 54 58 54 20 00 00 00 00 00	: C3 CC
EA2B:	00 00 00 00 00 00 07 58 68 0C 20 00 C5 00 00 00 00	: 1A E3
EA2C:	53 43 49 4C 4C 4F 20 42 41 53 20 00 00 00 00 00	: 28 B4
EA2D:	00 00 00 00 00 47 6A 85 0C 2F 00 00 01 00 00 00	: 80 C9
EA2E:	53 43 49 50 47 40 20 40 40 20 20 00 00 00 00 00	: 2F 8D
EA2F:	00 00 00 00 00 4E 6A 85 0C 30 00 52 00 00 00 00	: E0 8D
EA30:	52 41 43 54 49 4F 4E 4C 40 20 20 00 00 00 00 00	: 5C 8D
EA31:	00 00 00 00 00 00 00 01 21 0C 31 00 51 01 00 00 01	: 2C 00
EA32:	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	: 03 7C
EA33:	43 41 4C 4C 20 50 49 2A 00 00 50 55 53 48 20 48	: FF 48
EA34:	4C 00 00 00 41 4E 44 20 41 20 00 00 00 00 53 42	: 5F 01
EA35:	52 20 00 00 00 00 00 00 52 45 54 20 50 50 00 00	: 7D 48
EA36:	00 00 4A 50 20 48 4C 00 00 20 00 00 4A 50 20 50	: 4E 2A
EA37:	50 2A 00 00 00 00 45 58 20 44 45 2C 48 4C 00 00	: 17 69
EA38:	43 41 4C 4C 20 50 2A 00 00 45 44 28 00 00 00	: 86 B4
EA39:	00 00 00 00 58 4F 52 20 41 20 00 00 00 00 00 00	: 19 6C
EA3A:	00 00 00 00 00 00 52 45 54 20 50 00 00 00 00	: 03 50
EA3B:	00 00 4F 50 20 41 40 00 00 00 00 4A 50 20 50	: 56 04
EA3C:	2A 00 00 00 00 00 44 49 00 00 00 00 00 00 00 00	: 87 00
EA3D:	43 41 4C 4C 20 50 2A 00 00 50 55 53 48 20 41	: 86 41
EA3E:	46 00 00 4F 52 20 41 28 00 00 00 00 4A 50	: 48 C5
EA3F:	41 00 00 00 00 00 52 45 54 20 40 00 00 00 00	: 41 58
EA40:	00 00 44 20 53 50 2C 48 4C 00 00 4A 50 20 40	: 7F 58
EA41:	2A 00 00 00 00 00 45 49 00 00 00 00 00 00 00 00	: 88 00
EA42:	43 41 4C 4C 20 49 2A 00 00 46 44 28 00 00 00	: 83 B5
EA43:	F2 02 03 F0 F0 F0 F0 F0 00 00 00 00 00 00 10	: A7 00
EA44:	11 12 13 14 15 16 17 F0 19 1A 1B 1C 1D 1E 1F 20	: 7C E6

EXTENSIONS PC-1600 (3)

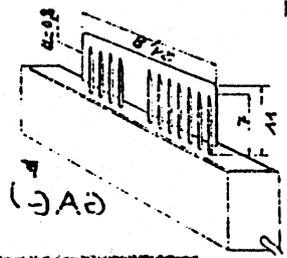
Dans ce troisième article, nous allons montrer comment fabriquer facilement un câble permettant de relier deux PC-1600 par la RS-232C.

Il faut pour cela un câble composé de trois fils électriques ou deux fils plus la masse. A chaque bout de câble, on construit un connecteur sur la RS. Ainsi un de nos lecteurs : M. Loutter a imaginé comment fabriquer facilement un connecteur quinze broches bien qu'ici trois suffisent. (Un connecteur au pas de 1,27 mm, par exemple pour la RS sur PC-1600, se réalise aisément et la recette en est simple.

Récupérer sur des cartes de surplus des picots plaqués 5 carats de 0,8*0,8*14 mm. Éliminez ceux qui présentent des raynures et qui accrochent. Soudez les aboutissants et les glisser un à un dans la prise RS. Enfilez le détrompeur-guide dans son logement puis retournez le PC-1600 dos en l'air et inclinez-le à 45 degrés. Vérifiez l'isolation et peaufinez le parallélisme, ensuite placez à cheval entre chaque broche une perle d'araldite

rapide, ne pas charger ; mais afin d'assurer une bonne liaison recommencer l'opération dès polymérisation. A noter que l'inclinaison du PC s'oppose à une remontée du polyester vers l'appareil. Laissez durcir avant d'extraire la structure obtenue. Puis surmoulez dans un boîtier de petites dimensions, ôtez les carottes puis dégraissez. Vous obtenez ainsi un nano-connecteur très honorable, parfaitement fiable et fort pratique.]]
DESSINS ET TEXTES DE M. LOUTTER
Les connecteurs terminés, on définit les fils A,B,C.
On soude à un bout du câble le fil A sur la broche correspondante à la broche N° 2 du PC. Le fil B sur la N° 3 et le fil C sur la N° 7. De l'autre côté du câble, refaites la même opération en inversant les fils A et B (A sur la 3 et B sur la 2).
Votre câble est opérationnel.

On introduit les paramètres de communication sur les machines.
SETCOM "COM1:",38400,8,E,1,X,S
SETDEV "COM1:",PO,KI
SNDSTAT "COM1:",254,2
RCVSTAT "COM1:",254,2
Pour envoyer un PGM:
SAVE "COM1:"
Pour recevoir un PGM:
LOAD "COM1:"
Vous pouvez désormais échanger des programmes, des textes ou des données entre PC-1600 (voir manuel du PC-1600 et articles précédents).



TRUCS ET ASTUCES

Vous avez eu des difficultés pour utiliser votre CE-1600P après avoir brancher votre PC-1600 sur la CE-150 ; alors faites :

- PAPER C
- POKE &F9EE,0
- POKE &F9F8,0,0
- POKE &F9F5,2

- Essayer donc de faire SHT CA en mode RUN
- A = A + 1
- POKE &F080,0
- Le PC-1600 va boucler automatiquement car le POKE remet à chaque fois le pointeur du KBUFF\$ à zéro.
- CALL 5 provoque l'AUTOSHUT OFF

- POKE &F12B,8 provoque un BEEP horaire quand le PC est allumé.
- POKE &F3C6,8 allume le s du clavier international.
- POKE &F3C6,2 allume les symboles japonais.
- POKE &F12B,0 AUTOPOWER ON.
- POKE &F12B,2 AUTOPOWER OFF

M. C.

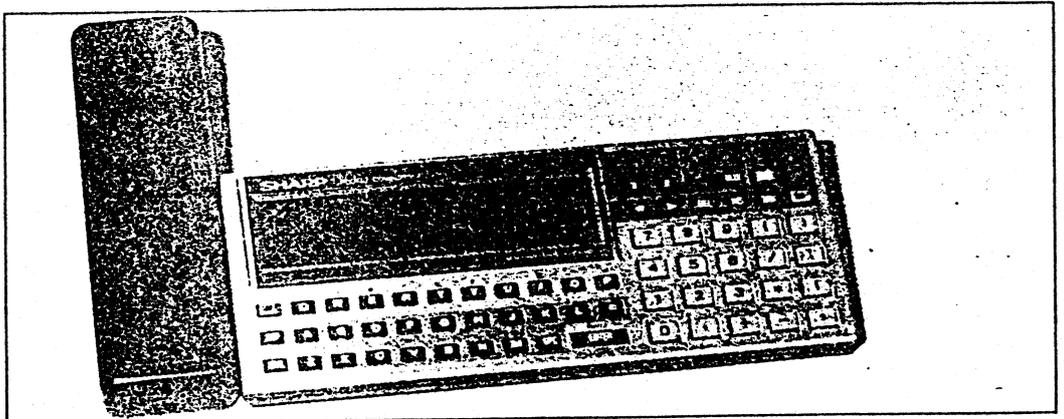
NANO- ORDINATEURS



N° 6 JANVIER-AVRIL 1988 30F

EPROM-TAPE

Il y avait longtemps que l'on n'avait pas pu tester un nouveau périphérique vraiment original pour les NANOS SHARP. C'est donc avec plaisir que l'on a pu utiliser le lecteur d'EPROM conçu par BECKER & PARTNER.



POURQUOI LIRE UNE EPROM ?

Le mérite de ce lecteur d'EPROM est de pouvoir recharger des programmes qui ont été auparavant sauvés sur ce support. Il faut tout d'abord rappeler qu'une EPROM est un circuit qui comporte de la mémoire qui est affectable uniquement par "grillage" grâce à une tension électrique bien dosée envoyée dans les pattes de la puce. Une fois l'EPROM grillée, on ne pourra l'effacer qu'avec un appareil spécial envoyant des UV à longueur d'onde précise. On voit donc que les programmes sauvés sur ce support sont indestructibles. C'est en effet un grave problème pour ce qui est applications industrielles que la perte des programmes sur cassettes qui tout en étant fiables sont fragiles.

LE MATÉRIEL

L'EPROM-TAPE se présente sous la forme d'une petite boîte noire auto-

alimentée par pile et qui inclue l'EPROM. Ce boîtier se branche sur le côté des machines sur la prise interface cassette et se met en route par un petit interrupteur. Immédiatement un petit bruit rappelant l'interface cassette se produit. En effet le contenu de l'EPROM est envoyé dans le NANO exactement comme s'il avait été sauvé sur cassette. Cette solution est très élégante puisqu'elle évite d'avoir recours à un soft pour lire le programme, l'instruction CLOAD étant suffisante. A noter que CLOAD M pourra aussi être utilisée pour lire du LM. Vous voyez donc que ce matériel permet de charger des programmes tout en oubliant que l'ERROR 8 (I/O DEVICE ERROR) a pu exister un jour.

LA CAPACITÉ

Les EPROMs utilisées peuvent être des EPROMs de 32 Ko, 64 Ko ou 128 Ko. A noter qu'il est prévu pour fonctionner avec des EPROMs de 16 mega-bits le

jour où l'on pourra en disposer facilement.

Il existe deux types d'EPROM-TAPE suivant le matériel avec lequel il doit tourner, soit donc :

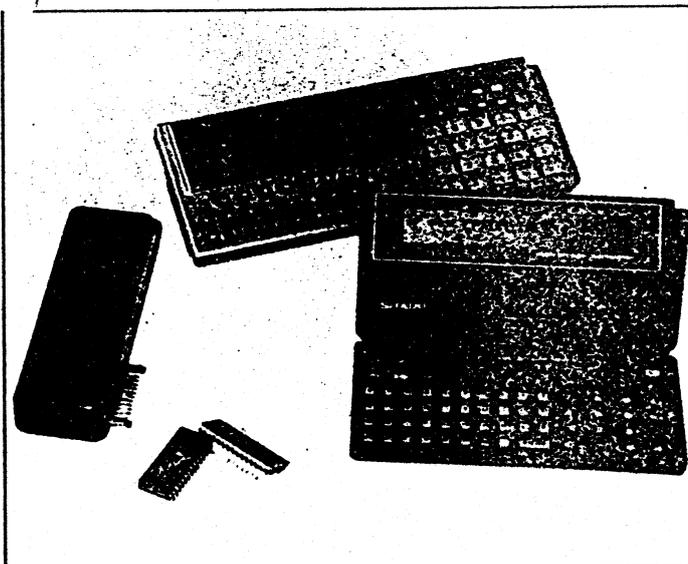
EPROM-TAPE Type A (Pour CPU 8 bits)
PC-1260/61/62, PC-1280, PC-1350/60,
PC-1401/02/03/21/50/60/75

EPROM-TAPE Type B (Pour CPU 4 bits)
PC-1245/46/46s/47/48, PC-1430.

CRÉER SON EPROM

Le format utilisé étant très spécial, il faut faire faire ses copies par le constructeur du matériel qui dispose du matériel adéquat. Cela introduit donc une lourdeur dans ce domaine. Mais ce matériel n'étant destiné qu'au marché vertical, c'est-à-dire à une utilisation professionnelle, l'avantage que procure ce matériel permet de ne pas être freiné par ce détail.

LE CLUB



INFO CLUB

TECHNICAL REFERENCE MANUAL PC-1600

Il est enfin disponible au club, cette « bible » sur la ROM et le système du PC-1600. Il est de la même trempe que celui sur PC-1500, c'est vous dire sa qualité. Vous y trouverez la description de toutes les instructions des microprocesseurs, les tables de MEV ainsi que des adresses de routines internes.

Ce petit chef-d'œuvre indispensable coûte 295 F + 20 F de port, à l'ordre du Club Nano-Ordinateurs.

SERPENT

ROUTINES

Tout le monde connaît bien sûr l'histoire de ce petit serpent affamé qui passe sa vie à engloutir des tonnes de pommes.

Or celui-ci est confronté au grave problème de la croissance : en effet chaque fois qu'une pomme est avalée, le petit serpent devient moins petit, puis grand, puis franchement très grand.

Tout cela ne serait pas grave si cette bestiole grandissante n'avait pas à se déplacer dans un espace réduit et semé d'obstacles.

Il doit évidemment jamais, à aucun instant rencontrer une partie de son corps sous peine de faire des nœuds et de s'asphyxier.

Votre rôle est donc bassement matérialiste, car vous êtes le serpent qui ne pense qu'à manger tout en évitant l'asphyxie (inévitabile vous en conviendrez !)

Pour permettre au serpent de tourner, il suffit de lui indiquer la direction à l'aide du pavé numérique. Les pommes étant représentées par des points, il suffit de foncer sur l'une d'elle pour l'avalier.

Le jeu se termine par l'agonie de la pauvre bête.

Article : Frédéric MEYER
Programme : M. C.

```

10:CLS:CLER:RANDOM:DIM B(8),C(8),A(8):PB0
12:INPUT "VERSION (0-3)?":UE,"DIFF (0-10)?":LP:CLS:S=UE+108+2E3
13:LINE (50,0)-(99,31),,B:FOR I=1TO LP+10:GOSUB S:NEXT I:BEET 1
20:WAIT 0:0=0:W=1:AC(0)="444":LINE (50,0)-(99,31),,B
22:B(2)=1,B(8)=-1,C(4)=-1,C(6)=1
30:PSET (50,2):PSET (57,2):A=50,B=2
40:C=RND 49+50:D=RND 32-1:IF POINT (C,D)GOTO 40
50:PSET (C,D):E=A,T=B,U=A+3,U=B
60:Y=VAL INKEY$:IF YLET D=B(Y),M=C(Y)
65:T=T+0:IF T<0OR T>2:LET T=T-0:GOTO 60
70:IF YLET A=C(0)=STR$ Y-A<0)ELSE LET A=C(0)=LEFT$ (A(0),1)-A=C(0)
72:L=LEN A(0)-1,R=R-1:IF R<0IF T=0BEET 1:GOSUB 300:GOTO 60
75:IF POINT (R,T)GOTO 1000
82:PSET (R,T):Y=VAL RIGHT$ (A(0),1),A(0)=LEFT$ (A(0),L),U=U-C(Y),U=U-B(Y):
PSET (U,U):GOTO 60
9: C=RND 49+50,D=RND 32-1:IF POINT (C,D)GOTO 300
210:PSET (C,D):RETURN
1200:BEET 2,50:BEET 1,52:KBUFF$="" :BEET 2,12:WAIT :GOCURSOR 102:PRINT "SCORE=":
STR$ (LEN A(0)-4):GOTO 20
2000:A=RND 51+48,B=RND 27+2:FOR J=A-2TO A-3:FOR K=B-2TO B-3:IF POINT (J,K)GOTO
2000
2025:NEXT K:NEXT J:LINE (A,B)-(A+1,B-1),,B:RETURN
2100:A=RND 50+48,B=RND 26+2:FOR J=A-2TO A+4:FOR K=B-2TO B+4:IF POINT (J,K)GOTO
2100
2110:NEXT K:NEXT J:LINE (A,B)-(A+2,B+2):RETURN
2200:A=RND 49+48,B=RND 25+2:FOR J=A-2TO A-3:FOR K=B-2TO B+5:IF POINT (J,K)GOTO
2200
2210:NEXT K:NEXT J:LINE (A,B)-(A+3,B):LINE (A,B+3)-(A+3,B+2):M=RND 2:LINE (A+M,
B-1)-(A+M,B+2):RETURN
2300:A=RND 51+49,B=RND 25+2:FOR J=A-2TO A-3:FOR K=B-2TO B+5:IF POINT (J,K)GOTO
2300
2310:NEXT K:NEXT J:M=RND 2:N=RND 2:LINE (A+M,B)-(A+M,B+3):LINE (A,B+N)-(A+3,B+N
):RETURN
2400:A=RND 51+49,B=RND 25+2:FOR J=A-2TO A+3:FOR K=B-2TO B+5:IF POINT (J,K)GOTO
2400
2410:NEXT K:NEXT J:LINE (A,B)-(A+1,B+3),,B:RETURN
2500:A=RND 51+49,B=RND 25+2:FOR J=A-2TO A+3:FOR K=B-2TO B+5:IF POINT (J,K)GOTO
2500
2510:NEXT K:NEXT J:LINE (A,B)-(A+3,B):LINE (A,B-1)-(A,B-3):RETURN
  
```

STATUS 1

1284

Fonctions	Entrées	Sorties	Page	Adr.	R. mod.
LCD ON/OFF	A=1 ON A=0 OFF	-	0	5005	A
CLS	-	-	0	5112	-
DÉCALE LCD HAUT	-	-	0	5120	-
DÉCALE LCD BAS	-	-	0	5130	-
CURSOR TEXT	D=COORD X E=COORD Y	-	0	5115	AF
CURSOR GRAPH	DE=COORD X BE=COORD Y	-	0	5148	-
PRINT CHR\$ A INC CURSOR TEXT PRINT CHAINE	A=CODE ASCII DE=ADR DE CHAINE A=CARAC D'ARRÊT	CF=1 SI FIN DE LIGNE CF=1 FIN LIGNE DE=ADR DE (A)	0	5100	AF
PSET (X,Y)	(BFB8EF)=COORD X (BFB937)=COORD Y (BFB945)=0 SET =1 RST =2 INV	-	0	5127	AF
LINE (A,B)-(C,D)	(BFB8EF)=a (BFB937)=b (BFB923)=c (BFB945)=d (BFB970)=CONFIGU (BFB99)= voir PSET	-	0	5121	AF BC DE HL
POINT (X,Y)	(BFB8EF)=COORD X (BFB937)=COORD Y	A=0 pt blanc A=1 pt éteint	0	512A	AF
GPRINT A INC CURSOR GRAPH	A=CODE GRAPHIQUE (BFB99)=0 NORMAL =1 GR =2 XOR	-	0	5154	AF
Xreg - DE	Xreg ARITHMÉTIQUE 0<Xreg <8FFFF	CF=1 ERROR	0	5247	AF BC DE HL
±Xreg - DE	- 87FFF<Xreg<87FFF	CF=ERROR	0	5241	TOUS
DE - Xreg	DE=VALEUR	-	0	5244	TOUS
STR\$ HL	DE=ADR DE CHAINE 0 HL 8FFFF	DE=ADR FIN B=NSRE CARAC	0	5253	AF BC DE HL IX
STR HL	DE=ADR DE CHAINE HL=ADR DU REG ARITH	DE=ADR FIN B=NSRE CARAC	0	5244	TOUS
OFF	-	-	0	55E0	-
AUTOSHUT OFF	-	-	0	5205	-
PEEK# (C,HL)	C=N° PAGE HL=ADR	A=VALEUR	0	5225	AF
LIT CARAC DU KBUFFER AVEC ATTENTE DU CARAC	-	A=CODE ASCII	0	5158	AF
LIT UN CARAC DU KBUFFER SANS ATTENTE	-	A=CODE ASCII CF=1 KBUFFER VIDE	0	5169	AF
INKEY\$ et CLEAR KBUFFER	-	A=CODE ASCII	0	5175	AF
TEST BREAK	-	BIT A,1=1 BREAK	0	51A,51B	-

M. C.

DRIVER CLAVIER

Bien que le PC-1600 dispose d'une gestion du clavier très puissante, il peut arriver que l'on veuille détourner le clavier pour son propre usage.

D'abord il faut savoir que la machine va à l'adresse &F0A7 à chaque fois que vous appuyez sur une touche. Il suffit donc d'y mettre un saut à une adresse supérieure à &C000 pour gérer le clavier.

De plus, il faut commencer la routine de détournement par les codes &F1 (POP AF) et la terminer par un &C9 (RET).

Sachant que le code de la touche appuyée est dans l'adresse &F083, vous pouvez commencer votre DRIVER-CLAVIER. Le revers de la médaille est que le PC exécute et affiche la touche puis l'envoie au driver. Cela signifie que le driver ne permet pas de changer une touche. On peut néanmoins faire des fonctions très utiles. Voici un exemple de petit clavier. Il produit un son différent pour chaque touche et change de son si vous dépassez 80 caractères sur une ligne. Il stocke dans W la valeur du registre arithmétique X (ANS) et effectue un AUTO-POWER-OFF quand vous pressez SHT 4.

Entrer le PGM à l'adresse &FD10 et faites POKE &F0A7,&C3,&10,&FD10.
M. CHOUCROUN

FD10:	F1	POP AF
FD11:	3A11FE	LD A, (FE11)
FD14:	FE50	CP A, E0
FD16:	300A	JR C, &FD22
FD18:	3E14	LD A, 20
FD1A:	016400	LJ BC, &00E4
FD1D:	CD8701	CALL, &01B7
FD20:	1009	JR, &FD2B
FD22:	0000	LD B, 0
FD24:	CD3901	CALL, &0139
FD27:	CB47	BIT A, 0
FD29:	201E	JR NZ, &FD49
FD2B:	3A83F0	LD A, (F083)
FD2E:	F5	PUSH AF
FD2F:	011200	LD BC, &0112
FD32:	CD8701	CALL, &01B7
FD35:	F1	POP AF
FD36:	FE60	CP A, 90
FD38:	CA0500	JP Z, &0005
FD3B:	FE57	CP A, 87
FD3D:	200A	JR NZ, &FD49
FD3F:	1100F9	LD DE, &F900
FD42:	2100FA	LD HL, &FA00
FD45:	0E00	LJ C, 0
FD47:	ED00	LDI R
FD49:	C9	RET
FD4A:	00	NOP

二成二年二日

POCKET & PORTABLE COMPUTER MAGAZINE

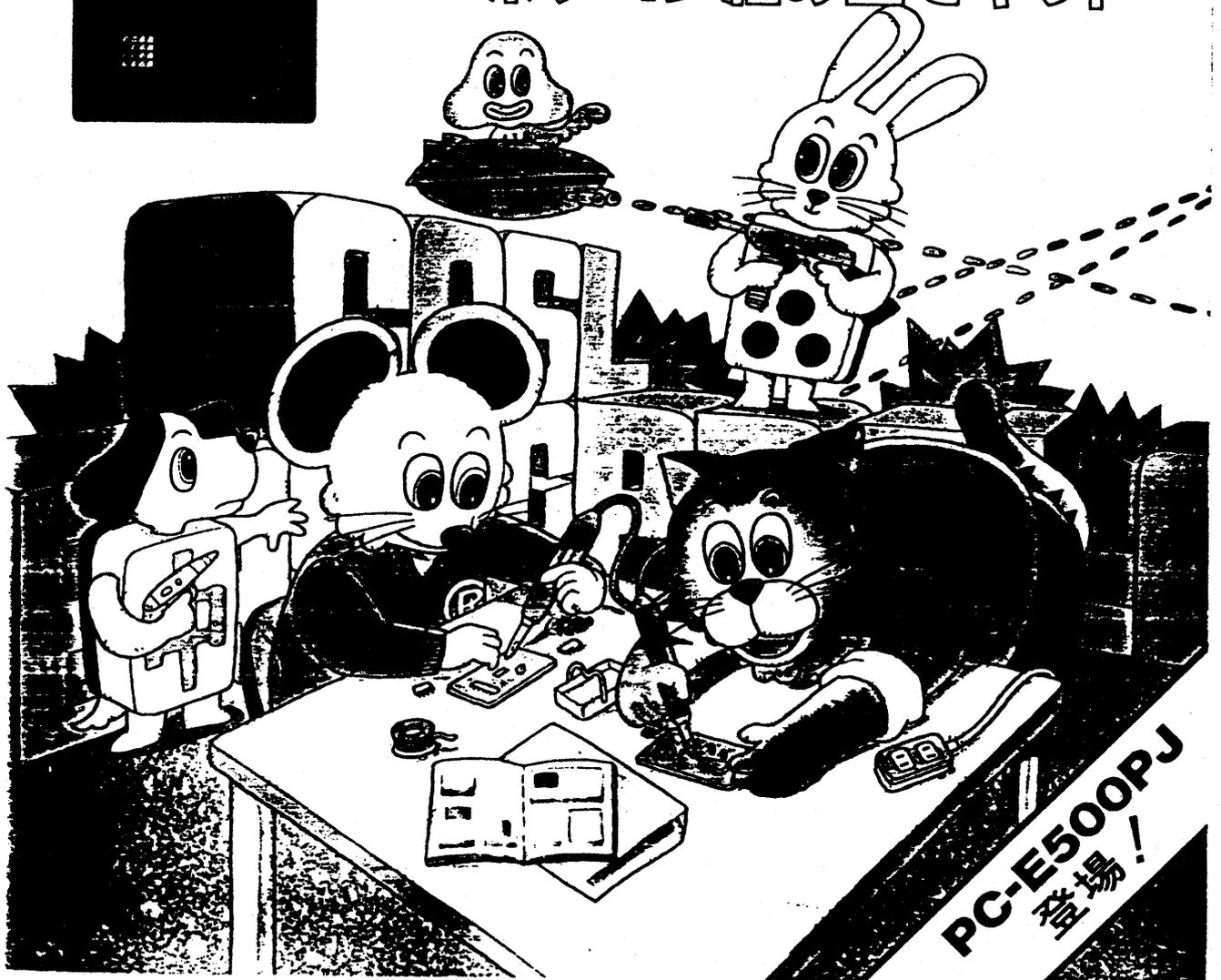
Pockecom Journal

ポケコン・ジャーナル

10増刊

'90 Number 4

AI-1000で言語入門 ポケコン組み立てキット



PC-E500PJ
登場!

エピソード

あなたが、CE-140FのIOCSを教えてください。

KFILESのPC-1280 1300シリーズおよびPC-E500版

を作ります。

□参考文献

PC-1600Kデータブック、工学社

KFILES BASICリスト

```

10 ' Extended Directory List by KATZ
    E SYSTEM TECHNIK '87
20 ' -Addr-Mch-----Op---Opr-----
    <Z80> --
30 ' F750:11 00 F6 LD DE,F650H
    ;E$--O$ Area
40 ' F753:0E 1A LD C,1AH
    ;Set DMA
50 ' F755:CD DE 01 CALL IOCS
60 ' ;*
70 ' F758:ED 5B 4E F0 LD DE,(PTRG)
    ;see pl56 F04E
80 ' F75C:0E 11 LD C,11H
    ;search first/next or read
90 ' F75E:CD DE 01 CALL IOCS
100 ' F761:32 FF F7 LD (F7FFH),A
    ;set status
110 ' F764:C9 RET
120 ' -----
130 "A"CSIZE 1,K:PCONSOLE "LPT1:",,,4:
    CLEAR
140 R$=" KFILES Ver=1.8"
150 CLS :INPUT "DEVICE (0=X: 1=S1: 2=S
    2:) ?";Y
160 W$="X":X=ASC W$
170 IF Y>0LET Y=Y+30:X=ASC "S":W$="S"
    +CHR$ Y
180 W$=W$+" ";
190 INPUT "VOL.ID ?";S$
200 MAXFILES =1
210 POKE &F750,&11,&50,&F6,&0E,&1A,&CD
    ,&DE,&01,&ED,&5B,&4E,&F0,&0E,&11,&
    CD,&DE
220 POKE &F760,&01,&32,&FF,&F7,&C9
230 A=&F650:P=&F750:Q=P+&D:H=ASC "?"
240 BU=PEEK &F04E+256*PEEK &F04F:POKE
    BU,1,X,Y,0,0,1,0,0,0
250 FOR F=1TO 48
260 POKE BU+9,H,H,H,H,H,H,H,H,H,H
270 IF F=2LET P=&F758
280 IF F>1POKE Q,&12
290 CALL P:IF PEEK &F7FF>0THEN 730
300 Y=PEEK (A+11):C$="X "
310 IF Y=32LET C$=" "
320 IF Y=33LET C$="P "
330 X=PEEK (A+18):M=XAND 31
340 N=(PEEK (A+19)AND 1)+8+INT (X/32)
350 C$=C$+RIGHT$ (" "+STR$ N,2)+"/"+RI
    GHT$ ("0"+STR$ M,2)+" "
360 X=PEEK (A+17):N=INT (X/8)
370 M=(XAND 7)+8+INT (PEEK (A+16)/32)
380 C$=C$+RIGHT$ (" "+STR$ N,2)+": "+RI
    GHT$ ("0"+STR$ M,2)+" "
390 C$=C$+RIGHT$ ("0"+HEX$ PEEK (A+1A
    ),2)
400 G=PEEK (A+1C)+256*PEEK (A+1D)
410 X=&F7E0:FOR L=0TO 7
420 U=PEEK (A+L):IF U=32THEN 440
430 POKE X,U:X=X+1:NEXT L
440 D$="":IF PEEK (A+8)<>32POKE X,46:D
    $="." :X=X+1
450 FOR K=8TO 10
460 U=PEEK (A+K):IF U=32THEN 490
470 POKE X,U:X=X+1
480 NEXT K
490 POKE X,0:IF D$="."LET D$=MID$ (Y$,
    L+1,4)
500 CLOSE :OPEN W$+Y$FOR INPUT AS #1
510 POKE Q,&14:CALL P
520 B$="A":IF PEEK A<255THEN 560
530 B$=" ":U=PEEK (A+4)
540 IF U=16LET B$="M"
550 IF U=33LET B$="B"
560 IF F>1THEN 590
570 LPRINT R$;" Vol=";W$;" (" ;S$;"
    )";TAB (68);"89/" +DATE$ + " "+TIME
    $
580 LPRINT " No Name p Date
    Time Lc Leng T Load Exec / Con
    tents"
590 LPRINT USING "###";F;:USING :LPRIN
    T " ";LEFT$ (Y$,L);TAB (13);D$;TA
    B (18);C$;
600 LPRINT USING "#####";G;" "+B$+"
    ";
610 IF B$<>"M"THEN 670
620 LPRINT RIGHT$ (" "+HEX$ PEEK (A+9
    ),3)+RIGHT$ ("0"+HEX$ PEEK (A+8),2
    );
630 LPRINT RIGHT$ (" "+HEX$ PEEK (A+1
    2),3)+RIGHT$ ("0"+HEX$ PEEK (A+11)
    ,2);
640 FOR I=16TO 27
650 LPRINT " "+RIGHT$ ("0"+HEX$ (PEEK
    (A+I)),2);
660 NEXT I:LPRINT " ":GOTO 720
670 S=A:IF B$="B"LET S=A+16
680 FOR I=0TO 45
690 T=PEEK (S+I):Z$=" ":IF T>31AND T<&
    E7LET Z$=CHR$ T
700 LPRINT Z$;
710 NEXT I:LPRINT " "
720 NEXT F
730 IF F=1LPRINT " ";W$;" ";S$;"
    = NO FILE ="
740 BEEP 2:LPRINT TAB (29);"Rem.=";DSK
    F W$
750 END

```

PJブラザ

♪ふんばあー♪

Blue KnightさんとFenegeeさんは宇宙人だったわー。くっくちがなーい。
おまけにばふたり☆おとらさんも宇宙人だったわー。はながないよ。かなびー。
げげー。誰かえろうさんもはな……。あつなんだ。誰かえろうさんは両性種だった。ほつ。
くだらなくてごめんね。

(くりアズ)