

All and more about Sharp PC-1500 at <http://www.PC-1500.info>
1. Jahrgang HEFT 8 Berlin / Dez. 1983

SHARP

PC-1500 ZEITUNG

Taschencomputer

ORGAN DES PC-1500 USER-CLUBS DEUTSCHLAND UND DER WELTWEIT ANGESCHLOSSENEN USER-CLUBS.

Software -
Information
ZUM
SHARP PC-1500



Eglise Romane de
Bordeaux St. Clair

J.P. (Toulon/Frankreich)

INHALT

Seite:

1. Desinger	2
2. ML - Teil 4 -	7
3. CE-161 ?	13
4. Tool 3	15
5. Diskettenlaufwerk	16
6. 28k RAM Eigenbau	18
7. Basikurs	19
8. PC-1500A	20
9. KEEP II	21
10. Endlos-Cassetten	23
11. ML-Renumber	24
12. MODE "Francais"	25
13. VIP-Test	31
14. Melodie	32
15. Math.	33
16. ML-Anzeige	36
17. Bau	38
18. * Kino *	39
19. Grafik II	40
20. Diverses	42
21. Ing.-Soft	43
22. H	44
23. SHARP Corp.	45
24. Impressum	46

PC-1500 Treffen: 28.12.83

Auskunft Tel. 323 60 29

Clublokal: Hauptmann von
Köpenick; Otto-Suhr Allee

11-13 (am Ernst-Reuter
Platz) - 19:30 Uhr.

SHARP

Do not sell this PDF **Durch Nachdenken vorn.**

Volker Desinger
 Nikolaus-Gross-Straße 8
 D-4709 Bergkamen
 Telefon : 02307/80137

04. NOV. 1983

Fischel Betriebswirtschaftlicher
 Beratungs- und Programmiereidienst GmbH
 Kaiser-Friedrich-Straße 54 a

1000 Berlin 12

Zusendung der PC-1500 Zeitung 1-7

Sehr geehrter Herr Fischel,

vielen Dank für die umgehende Zusendung der PC-1500 Zeitungen 1-7. Diese Zeitungen haben mir viele neue Informationen gebracht. Beiliegend finden Sie 2 Listings von, wie ich meine, sehr nützlichen Programmen.

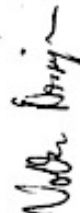
Das erste ist ein MS-Programm, ähnlich dem MS-Programm in Heft 7. Mein Programm rettet allerdings auch noch BASIC-Programme, die durch ERROR 44 nicht ganz in den Rechner geladen wurden.

Das zweite Programm listet BASIC-Programme im Querformat genauso schnell wie das LLIST-Kommando.

Das zu den beiden Programmen.

Gut an der PC-1500 Zeitung finde ich auch die Programmierung in Maschinensprache. Weiter so !!
 Was ich jedoch vermisse, sind einfache Hardwareerschaltungen (Speichererweiterungen, Schaltinterface), die man selbst nachbauen kann. Ich hoffe, in einer der nächsten Ausgaben kommt etwas ??

Mit freundlichen Grüßen



PS: Dieser Brief wurde mit dem MS-Programm "Endlostext" erstellt.

DOKUMENTATION FÜR Quer-LIST

Die Idee zu diesem Programm kam, da das BASIC-Kommando "LLIST" kein brauchbares Format hat und die BASIC-Zeilen auf dem Drucker CE-158 sehr zerstückelt ausgegeben werden und das ganze Programm schlecht überschaubar ist. Also mußte ein Programm her, das im Querformat listet. Dieses Problem wurde zuerst in BASIC gelöst, war aber viel zu langsam. Die jetzige Version benutzt zum Teil Maschinensprache und listet das Programm genauso schnell wie das LLIST-Kommando

Bei Bestellung der Programm-Kassette kann das nächste Kapitel "Herstellen einer Kassette" übersprungen werden.

HERSTELLEN EINER PROGRAMM-KASSETTE

Als erstes müssen die Maschinencodes aus dem beiliegenden Listing mit POKES oder einem Monitor-Programm ab Adresse &7658 eingegeben werden (Auf die absolut korrekte Eingabe dieser Codes ist unbedingt zu achten. Prüfen Sie die Eingabe, sonst "Absturzgefahr"). Als nächstes wird das Maschinenprogramm auf Kassette gespeichert mit dem Befehl CSAVE M "Quer-LIST (MS)".&7658,&76EE

Dann wird der Programmspeicher mit dem NEW-Befehl gelöscht. Nun muß noch der BASIC-Teil des Quer-LIST-Programmes eingegeben werden und mit dem Befehl

CSAVE "Quer-LIST (BAS)"

auf Kassette gespeichert werden.

GEBRAUCH DES PROGRAMMS Quer-LIST

Laden Sie das BASIC-Programm, das Sie auslisten möchten, in den Computer. Vergewissern Sie sich, daß STATUS 8 mindestens 455 Bytes lang ist. Das ist der Platz, den das Quer-LIST Programm einnimmt, einschließlich des Platzes für die dimensionierte Variable A\$(). Legen Sie nun die Kassette ein, auf der die beiden Teile des Quer-LIST Programms (MS+BASIC) sind. Laden Sie den Maschinenspracheteil mit CLOAD M

Nachdem das MS-Prgrn geladen wurde, laden Sie den BASIC-Teil mit MERGE

Nun geben Sie das Kommando RUN "QL" und geben auf die Frage "SIZE?" die gewünschte Schriftgröße (1 oder 2) ein. Ihr BASIC-Programm wird dann im Querformat auf dem Drucker CE-158 ausgelistet (natürlich wird das Quer-LIST Programm selbst nicht mit gelistet).

Volker Desinger
Nikolaus-Gross-Str.8
D-4789 Bergkamen

Volker Desinger
Nikolaus-Gross-Strasse 8
D-4789 Bergkamen
Telefon : 02307/80137

Hexdump "Quer-LIST"

```

38288 / 7658 : 85 2A ED 79 F4 01 88 02 85 59 AE 77 4F FD 88 CC 5myt 5P.w0H L
38384 / 7660 : 99 85 07 FD CA FD 5A 04 AE 77 4E 6A FB CD BA FD 5 0J0Z .WUJH:6
38328 / 7670 : 0A 45 28 45 2A FD 88 CD 10 48 FD 0A B5 3A 51 44 E<E+0 M 16 5r0D
38336 / 7680 : 45 87 E8 83 8D 88 07 FD 88 46 85 01 8E 24 E7' 7 6 FS $
38352 / 7698 : FB 9A 28 45 2A FD 88 48 88 4A 54 45 89 0F 8B 44 A <E+0 H0JTE9 D
38368 / 76A8 : FD CA 45 A6 89 39 85 26 89 35 46 46 85 09 91 8JEL 9 & 5FF Y
38384 / 76B8 : 05 45 89 0F 2A DF 28 A5 77 4E F9 A3 77 4F 16 91 E9 _.(Zu0ky=0
38400 / 76C8 : 07 28 16 89 02 68 88 83 88 22 1A 85 28 51 51 1 h * 5 000
38416 / 76D8 : 51 62 F3 88 03 A4 8B 83 85 28 51 FD 8A 9E 5F 44 0bu $ 5 00 _D
38432 / 76E8 : 44 44 9E 49 84 83 8B 8B 87 C8 91 53 FD 1A E4 8B DD 1 3 7H 5U d
    
```

```

1 "QL"INPUT "SIB?" :S:GRAPH :CSIZE S:ROTATE I:A:STATUS 2-STATUS I:DIK A:0:06-1:0:76/S:4:X:216
2 CALL 87659:A:D=0
3 IF X=0GLCURSOR (0,-576):SORGN :X=216
4 X=X-9:S:GLCURSOR (A):LPRINT A$(C):C+1:IF C<3:GIF A$(C)THEN 3
5 IF PEEK A$(C)FFTHEN 2
6 GLCURSOR (0,-576):TEXT

1: "QL"INPUT "SIZE?" :S:GRAPH :CSIZE S:
   ROTATE I:A:STATUS 2-STATUS I:DIK A$(C)*
   S-1)*76/5+4:X=216
2:CALL 87659,A:C=0
3: IF X=0GLCURSOR (0,-576):SORGN :X=216
4:X=X-9:S:GLCURSOR (X,0):LPRINT A$(C):C+
   1: IF C<3:GIF A$(C)THEN 3
5: IF PEEK A$(C)FFTHEN 2
6: GLCURSOR (0,-576):TEXT
    
```

1: "QL"INPUT "SIB?" :S:GRAPH :CSIZE S:ROTATE I:A:STATUS 2-STATUS I:DIK A:0:06-1:0:76/S:4:X:216
2:CALL 87659:A:D=0
3: IF X=0GLCURSOR (0,-576):SORGN :X=216
4: X=X-9:S:GLCURSOR (A):LPRINT A\$(C):C+1:IF C<3:GIF A\$(C)THEN 3
5: IF PEEK A\$(C)FFTHEN 2
6: GLCURSOR (0,-576):TEXT

EINIGE ALTERNATIVEN

Die oben beschriebene Verfahrensweise macht es Ihnen einfach, Ihre auf Kassette gesicherten BASIC-Programme im Querformat auszulisten. Das Programm selbst braucht wenig Platz im Speicher und ist in weniger als einer Minute geladen. Dadurch, daß das MS-Programm in den Stringvariablen gespeichert ist, braucht dafür nicht extra Platz reserviert zu werden.

Trotzdem, wenn das Programm in der oben beschriebenen Weise benutzt wird, kann es nicht als eine Hilfe beim Testen eines Programmes benutzt werden, das in Bearbeitung ist. Das ist deshalb so, weil das Kommando MERGE, mit dem der BASIC-Teil des QL-Programms geladen wird, das Programm, welches in Bearbeitung ist, nicht mehr editierfähig macht. Außerdem dürfen die Stringvariablen E\$-W\$ nicht benutzt werden, da diese von dem MS-Teil des QL benutzt werden (Beachten Sie, daß auch das CLEAR-Kommando den gesamten MS-Teil löscht !!).

Eine Alternative ist, den MS-Teil des Programms in etwa anders abzuspeichern. Der MS-Teil ist deshalb voll relokatable gehalten und läuft ohne Modifikation auch in einem anderen Speicherbereich (Natürlich muß dann der "CALL 87659" Ausdruck in Zeile 2 entsprechend der neuen Adresse des MS-Teiles umgeändert werden..).

Wenn der RESERVE-Speicher für nichts anderes gebraucht wird, ist dieser auch ein guter Platz für den MS-Teil.

Eine andere Möglichkeit besteht darin, den BASIC-Anfang mit dem "NEW XXXX" Ausdruck heraufzusetzen (Der MS-Teil des QL ist 159 Bytes lang).

Wenn kein MERGE-Kommando benutzt werden soll um den BASIC-Teil des QL zu laden, kann dieser auch vorher geladen werden, die 6 Zeilen können aber auch zu jeder Zeit eingetippt werden (Das Quer-LIST Programm wird dann auch mit (quer) ausgelistet).

BERND RÜTER

Programmieren in Maschinensprache Teil 4
 Bahndammstr. 65
 4955 Millie
 Tel.: 05703/672

Aufgrund der Reaktionen auf diese Serie stelle ich fest, daß ein großer Interesse nach ROM-Routinen besteht, sowie nach Speicherplätzen des System-RAM. Ich kann hier aus Platz- und Zeitgründen keine Aufstellung veröffentlichen, möchte aber doch auf Literatur verweisen: System-Handbuch der Fa. Holtkötter, zu beziehen bei Fischer, SSBH RVS-System-Handbuch bei: RVS, Parkstr. 30, 8 München 2 ROM-Unterprogramme bei: H.-G. Schlieker, Weichselstr. 5, 28 Bremen 21 Weiterhin erscheinen in dieser PC-1500-Zeitung laufend neue Adressen, und ich bin sicher, daß in der geplanten Publikation der Fa. Fischer eine Liste von Routinen aufgestellt wird.

Ich möchte heute erst einmal ein Problem darstellen, daß ich von einem Leser dieser Zeitung zugesandt erhielt und innerhalb weniger Stunden löste.

Problem: Es soll eine Zahl in Teller zerlegt werden. Diese Teller sollen aus einer vorher feststehenden Menge sein. Es soll sich folgendes Bild ergeben: $Zahl = x_1 \cdot Y_1 + x_2 \cdot Y_2 + \dots$

Dieses Problem wurde von dem Leser bereits mit einem Such-Verfahren in BASIC gelöst. Nur wird diese Routine oft benötigt und sie ist sehr langsam. Deshalb entwickelte ich ein Maschinensprache-Programm, welches das Problem durch Probieren löst. Es werden alle Teller mit (fast) allen multipliziert und dann auf Richtigkeit geprüft. Als großer Vorteil sei erwähnt, daß alle Zahlen ganze Zahlen sind und daß sie kleiner 65536 sind, also im 16Bit-Format darstellbar sind. Ich gliederte das Programm in 4 Teile:

- Initialisierung
 - Eingabe der möglichen Teller
 - Suche der Teller (1)
 - Suche nach weiteren Teller (2)
- Die Übergabe der Werte kann mit dem CALL erfolgen, gefolgt von einer Variablen.

*** VD-Software ***

RENEW In MS für PC-1500

MS-Programms zum aktivieren nicht richtig geladener oder reaktivieren irrtümlich gelöschter Basic-Programme.

Rettet nach ERROR 44 den noch ladbaren Teil des Basic-Programms oder versehentlich mit NEW gelöschte Basic-Programme.

```

ADRESSE  HEX-CODE  OP-CODE  IBEHERRIGEN
60000  A5 78 65  LD A,(7865)  iLade A mit (&7865) (/38021) : 1. Byte Basic-Anf.
60003  88          LD XH,A      iLade XH mit A
60004  A5 78 66  LD A,(7866)  iLade A mit (&7866) (/38022) : 2. Byte Basic-Anf.
60008  8A          LD XL,A      iLade XL mit A
6000B  FD 6A      LD U,X      iLade U mit X
6000A  FD 5A      LD Y,X      iLade Y mit X
6000C  44          INC X       iIncrementiere X um 1
6000D  44          INC X       iIncrementiere X um 1
6000E  05          LD A,(X)    iLade A mit (X)
6000F  FD CA      ADD X,A     iAddiere A zu X
60011  B5 00      LD A,00     iLade A mit &00 /13 : Zeilenende-Zeichen
60013  F7          CPI A,(X)   iVergleiche A mit (X) und incrementiere X um 1
60014  98 0C      JR Z,-BC    iSprunge, wenn Zeilenende um -&0C -/12 By nach &000A
60016  B5 FF      LD A,FF     iLade A mit &FF /255 : Programmende-Zeichen
60018  1E          LD (Y),A    iLade (Y) mit A
60019  94          LD A,YH     iLade A mit YH
6001A  3E 78 67  LD (7867),A  iLade (&7867) (/38023) mit A : 1. Byte Basic-Ende
60000  14          LD A,YL     iLade A mit YL
6001E  AE 78 68  LD (7868),A  iLade (&7868) (/38024) mit A : 2. Byte Basic-Ende
60021  B5 00      LD A,00     iLade A mit &00 /0
60023  2E          LD (U),A    iLade (U) mit A
60024  9A          RET         iZurück zum aufrufenden Programm
  
```

BERND RÜTER

Rahdenstr. 65
4955 Mille
Tel.: 05703/672

Initialisierung:

Die In. ist nötig, da das Programm eine Liste aller möglichen Teiler aufbauen muß. Bei der In. werden nun die Pointer für diese Liste gesetzt. Die Liste wird der Einfachheit halber direkt hinter das BASIC-Programm gelegt. Die Folge von Befehlen, die mir diesen Wert in ein 16-Bit Register der CPU lädt habe ich im Assembler als MAKRO definiert: !LOXENDILO A, (PRGEND):LD XH, A:LD A, (PRGEND+1):LD XL, A / PRGEND=47867 wird vorher gesetzt
Dieser Makro ergibt übersetzt:

```
LD A,7867
LD XH,A
LD A,7868
LD XL,A ( wir werden diese Folge noch mehrmals verwenden )
```

Den Gleichen Makro habe ich auch für das Y-Register definiert.

Diesen letzten Makro verwende ich zuerst um den Pointer in das Y-Reg zu laden. Damit wäre die In. fast abgeschlossen, denn ich muß diesen Pointer nach im Speicher sichern, damit nach dem Rückkehrung in BASIC die Zahl nicht verloren geht. Zum Ablegen dieses Pointers verwende ich den Speicherbereich &77F0-&77FF, in dem nachher noch mehr Werte abgelegt werden. Ich stelle mir folgende Adressen auf:

```
ZUX=&77F0 (Zwischenspeicher, reserviert für das X-Reg)
ZUY=ZUX+2 (" " " " Y-Reg)
ZUW=ZUY+2 (" " " " U-Reg)
ZUWERT=ZUW+2 (" " " den zu unters. Wert)
```

Oazu auch folgende MAKRO-BIBLIOTHEK:

```
!SVEIXILO A, XH:LD (ZUX), A:LD A, XL:LD (ZUX+1), A
gleiches mit Y und U und dem 'Wert':
!SVEUYILO A, XH:LD (ZUWERT), A:LD A, XL:LD (ZUWERT+1), A
```

der Wert wird in X behandelt, da er hier vom BASIC übergeben wird, darum wird hier nochmal das X-Reg verwendet, sonst natürlich die Y- und U-Reg. Zum Wiederholen dieser Werte:

```
!GETXILO A, (ZUX):LD XH, A:LD A, (ZUX+1):LD XL, A
so auch mit Y, U, Wert
```

Nach dem Anlegen dieser Makros und der Adressen kann man jetzt richtig anfangen (mit ASSEMBLER):

```
INIT !LOYENDI:!SVEYI:RCF:RET Der Pointer-Wert wird aus der Adresse
7867 geladen und in ZUY abgelegt. Danach wird das Carry-
Flag gelüschet und es erfolgt ein Rückkehrung in BASIC,
ohne den Inhalt aus dem X-Reg zu übernehmen (Carry)
```

BERND RÜTER

Rahdenstr. 65
4955 Mille
Tel.: 05703/672

Eingabe der möglichen Teiler:

Dieses Programm kann auf die Makros aus der In. zurückgreifen, d.h. ich muß nicht alles noch einmal eingeben, denn Makros sind keine Unterprogramme, sondern nur Befehlsfolgen, die man unter einem Namen ablegt.

STORE !GETYI Der Pointer auf die Tabelle wird geladen. Er zeigt auf das erste freie Element.

LD A, XH das erste freie Element.

LDI (Y), A Das H-Byte des einzuspeichernden Wertes aus X wird übernommen und abgelegt, gleichzeitig wird der Pointer erhöht.

LD A, XL

LDI (y), A Gleich dem H-Byte. Jetzt steht die Adresse des ersten freien Tabellenplatzes wieder in Y und wird wieder abgelegt.

RCF

RET

Diese Routine legt also eine Tabelle im Speicher an, beginnend bei der in INIT festgelegten Adresse, die in ZUY steht und mit !GETYI geladen bzw mit !SVEYI wieder abgelegt wird.

Wir wollen vereinbaren, daß das BASIC-Programm zuletzt eine 0 einaspeichert mit diesem Programm, womit das Ende der Tabelle angezeigt ist durch zwei aufeinander folgende Bytes mit 0 als Inhalt.

Bei dem letzten Teil, der Suche nach Teilern, verwende ich ein Unterprogramm aus dem ROM, welches mir das Y-Reg mit dem U-Reg multipliziert und das Ergebnis in X-Reg und Y-Reg übergibt. Das Ergebnis darf maximal 16 Bit lang sein. Wird diese Länge überschritten ist nach der Rückkehr das Carry-Flag gesetzt. Die Anfangswerte in Y-Reg und U-Reg dürfen auch nur max. 16 Bit lang sein (größer geht auch nicht!). Der Aufruf ist: CALL (&50) und als MAKRO definiert.

Raddeenerstr. 65
4955 Millie
Tel.: 05703/672

Suche nach Teilern:
Diesen Teil gestalte ich einfach in 2 FOR-NEXT-Schleifen.

Zuerst sichere ich den im X-Reg übergebenen gesuchten Wert mit ISVEUERT!. Dann beginne ich die erste Schleife: !LDXEND!. Mit diesem Befehl setze ich den 1.Zähler in X Die zweite Schleife starte ich entsprechend: ISVEXI:LD Y,X. Ich sichere den alten X-Reg und lade diesen Wert nach Y-Reg, meinem 2.Zähler. Ich beginne also nicht immer von vorn, weil ich nicht alle mit allen Zahlen multiplizieren brauche, sondern nur die Hälfte.

Jetzt beginnt der eigentliche Teil der Schleife, in dem die beiden Zahlen, auf die die Schleifenzähler zeigen miteinander multipliziert werden und verglichen wird, ob es schon das Ergebnis ist.

Das geschieht folgend:
ISVEY! Das Y-Reg sichern
LDI A,(Y):LD XH,A:LD A,(Y):LD XL,A Lädt die Zahl, auf die Y-Reg zeigt in X-Reg
JR NZ,Weiter Sprung, wenn noch nicht letzte Zahl erreicht.
LD A,XH:JR Z,NEXT1 Sprung, wenn letzte Zahl (0)

WEITER LD Y,X:GETY! Geledene Zahl nach Y-Reg und X-Reg wieder restaurieren.
LDI A,(X):LD UH,A:LD A,(X):LD UL,A:ISVEU! Zahl laden, auf die

X-Reg zeigt (FOR1) nach U-Reg und absichern.
!MUL:JR C,NEXT2 Multiplizieren und wenn Ergebnis zu groß, dann nächsten Wert nehmen.

GETWERTI:LD A,YH:CP A,XH:JR NZ,NEXT2 Gesuchte Zahl nach X-Reg laden LD A,YL:CP A,XL und mit Y-Reg(Ergebnis) vergleichen. Wenn nicht JR NZ,NEXT2 gleich, dann nächsten Wert.

GETU:LD X,U:SCF:RET Gefundenen Teiler wieder laden und zurück zu BASIC
NEXT2 GETYI:INC Y:INC Y:JR TEST Der Schleifenzähler der 2.Schleife wird erhöht. Sprung zum Anfang der Schleife.

NEXT1 GETYI:INC X:INC X:LD A,(X):JR NZ,FOR2 Zähler der 1.Schleife erhöhen und 1.Hälfte des Wertes laden. Wenn nicht null, denn FOR2 eröffnen.

INC X:INC X:LD A,(X):DEC X:JR NZ,FOR2 ebenso 2.Hälfte.
ENDE LD X,0:SCF:RET Ende der Suche, 0 übergeben in BASIC

Raddeenerstr. 65
4955 Millie
Tel.: 05703/672

Programm zum Auswerten eines Assembler-Quell-Programms:

```
1: "LISTASH" CLEAR
: TEXT /CLS /
WAJT 0:CSIZE 1
:LF 3:INPUT "N
ame:";A$
2:IF A$PRINT A$
: LPRINT
3:C=1:INPUT "C";
ze "1:C=C+1
AND C<2 THEN 3
4:CLS :A=STATUS
J:J=0:PEEK (A+
1)=0:PEEK (A+
2)=3:A$=CHR$
PEEK (J+1)
6:IF L=AND A$="
"LET A$="
LPRINT RIGHT$
(" "
C$);L=5+C
8:LPRINT A$:L=L+
1:LEN A$:NEXT J
9:A=A+PEEK (A+2)
+3:IF A$EVEN
5
10:IF 5:POKE &286
7,E/256,EAND 2
55:END
```

*Mit Menge darzuladen
und dann mit
"RUN" LIST ASM" starten.*

Quell-Programm für Teiler-Suche

```
=!MULCALL (&50)
JR INIT
JR STORE
JR SUCH
JR NEXT2
INIT !LDYEND!:ISVEY
STORE !GETY!
LD A,XH:LDI (Y),
/A
LD A,XL:LDI (Y),
/A
!SVEY!:PCF:RET
SUCH !SUEWERT!
FOR1 !LDXEND!
FOR2 !SVEY!:LD Y,X
TEST !SVEY! LD A,
/(Y):LD XH,A
LD A,(Y):LD XL,A
/(X):LD XH,JEITER
LD A,XH:JR Z,NEX
T)
WEITER LD Y,X:GET
Y!
/A
LDI A,(X):LD UH,
/A:LD A,(X):LD
UL,A:ISVEU!
!MUL!:JR C,NEXT2
!GETWERTI:LD A,Y
H:CP A,XH:JR
/NZ,NEXT2
LD A,YL:CP A,XL:
/JR NZ,NEXT2
!GETU:LD X,U:SC
F:RET
NEXT2 !GETY!:INC Y
/(Y):INC Y:JR TES
T)
NEXT1 !GETX!:INC X
/(X):INC X:JR
/ND,FOR2
INC X:INC X:LD
/A,(X):DEC X:J
/R NZ,FOR2
ENDE LD X,0:SCF:RE
T)
END
```

Hier möchte ich eine Erweiterung des bereits ausgeführten Speichertestes vorstellen. Diese Version prüft einen weiteren Fehler, der oft bei Eigenentwicklungen auftritt. Dieser Fehler äußert sich, indem zwei Speicherbereiche angesprochen werden bei einer Adresse. Dieser Fehler liegt an der Decodierung. Deshalb sind die Adressen, die doppelt angesprochen werden im Abstand von 800 (2K), weil die ICs meist 2K speichern.

Diesem zweiten Teil des Programms füge ich an das Programm, darum müssen die Anfangswerte (X,Y) gesichert werden, damit sie nach dem ersten Test noch verfügbar sind. Dies geschieht durch ein PUSH Y:PUSH X (Reihenfolge!) vor dem relativen Sprung am Programmstart. Wird im ersten Test bereits ein Fehler festgestellt, so müssen diese Register wieder vom Stack geholt werden (!). Jetzt erkennt man, warum die Register in dieser Reihenfolge abgelegt wurden. Man kann mit POP Y:POP X die Werte vom Stack holen, wobei das originale Y-Reg überschrieben, danach aber wieder geladen wird. Die Werte nach einem Rücksprung aus dem ersten Teil bleiben deshalb wie zuerst beschrieben. Wird der erste Teil erfolgreich durchlaufen, so muß der zweite Teil die abgelegten Werte wieder vom Stack lesen mit POP X:POP Y. Danach beginnt der zweite Teil, der dem ersten Teil sehr ähnlich ist.

Es erfolgt ein Sprung zum Ende der Schleife, an dem auf Ende geprüft wird. Zuerst jedoch wird der Wert aus X-Reg in U-Reg übertragen für den späteren Test. Jetzt wird zum X-Reg 800 addiert mit LD A,8:RCL:ADC A,XH:LD XH,A. Jetzt wird überprüft wie bei der ersten Schleife, allerdings steht XH bereits in A. Ist das Ende noch nicht erreicht, so wird zum Anfang der ersten Schleife verzweigt. Hier wird in die Speicherstelle, auf die U-Reg zeigt 0 und in die Speicherstelle von X-Reg 255. (X-Reg ist 2K weiter wie U-Reg!). Würde jetzt ein Decodierfehler vorliegen, so überschreibt die 2. Speicherung die erste. Darum werden jetzt die Speicherinhalte wieder ausgelesen und auf ihren Wert überprüft. Stimmt dieser Wert nicht, so wird U-Reg auf 0 gesetzt und es erfolgt ein Ausprung aus dem Programm mit der Adresse in X-Reg, die die um 2K tiefere überschrieben hat.

Assembler-Listing

```

ELSE:LD A,UH:CP
  A,1:IF#2:LD
  U,80:J:ENDJF
DJC LOPTST
POP A:LDX:INC
  X
LD A,UH:AND A,1
LD UH,A
LOOPE LD A,XH:CP A
  YH:JR NC,L00
  P A,XL:JR C,L
  OP:ENDJF
;Test Teil 2
TFST2 POP X:POP Y
  JR LOOP2E
LOOP2 LD A,A:LD CU
  A:LD A,255:
  LD X),A
LD A,(U):IF#2:
  LD X),B
ELSE:LD A,(X):
  JNC A:IF#2:LD
  U,B
  ELSE

```

Rahdenerstr. 65
4955 Hille
Tel.: 05703/672

CE-161-Erfahrungsbericht ?? (siehe auch unten)

In der letzten Ausgabe der PC-1500-Zeitung wurden die Erfahrungen mit dem CE-161 berichtet. Bei dem Autor muß es sich wahrscheinlich um jemanden handeln, der das CE-161 noch nie in den Händen hatte, denn wie kann man sonst diese Fehler alle machen. (?)

Das Modul liegt in Bereich 8000 bis 83FF und wird mit NEW 0 initialisiert! Der Bereich für Reserve-Tasten wird bei 'NEW 0' automatisch mit 197 Byte reserviert. Der BASIC-Bereich geht nun bis 847FF, der geschützte Bereich (Schreibschutz) geht nur bis 83FFF, da hier das CE-161 endet.

Ich hoffe hiermit verschiedene Fragen gelöst zu haben, auch wenn ich das CE-161 nicht kenne, aber genug Erfahrung mit selbstgebauten Speichern habe.

Zusatz zu 'Programmieren in Maschinensprache 4':

Ich erhielt nach Fertigstellung des Programms von dem Leser wieder einen Brief, in dem er schreibt, daß mein Programm langsamer sein, als seines in BASIC, und zudem noch Zahlen ausglibt, die nicht vorkommen dürfen. Dazu sei gesagt, daß dieses Programm nicht systematisch den Teiler sucht, sondern durch Probieren einen erhält. Dieses Probieren kann erheblich länger dauern, als das Suchen, da bei n möglichen Teilern max. n(n+1)/2 Rechnungen durchgeführt werden können.

Ich habe dieses Beispiel auch nicht als Problemlösung sondern als Lernobjekt gewählt. Sicher kann man durch geschicktere Programmierung noch etwas Laufzeit herausholen, muß denn aber beim Aufbau der Tabelle mehr aufpassen. (kleine Zahlen zuerst)

WAS IN DER BEDIENUNGSANLEITUNG DES CE-161 NICHT STEHT

Die in der Bedienungsanleitung des CE-161 gegebenen Anweisungen zum Einbau des 10K-Moduls in den PC-1500 sind offenbar nicht vollständig. Es kann vorkommen, daß, nach Einbau des Moduls und Neudefinieren des Programmpeichers (NEW 256), der Reservspeicher nicht mehr einwandfrei zu benutzen ist. Dies äußert sich, durch verzögertes Ansprechen der Reserve-Tasten und/oder durch Füllung des gesamten Displays beim Drücken von (RCL). Sollte dieser Fall eingetreten sein, so löst er sich durch Löschen des Reservspeichers; Eingabe von NEW im Reservemode, beheben. Diese Anweisung sollte bei erstmaligem Einbau des CE-161, auf jeden Fall durchgeführt werden.

BERND RÜTER

Rekorderstr. 65
4955 Millie
Tel.: 05703/672

TOOL 3

Wer hat das TOOL 3 von TransSoft?
Ich möchte wissen, ob sich die Anschaffung lohnt und suche
deshalb Besitzer dieser Erweiterung.

Klaus Schreiner
Im Langacker 2
6750 Kaiserslautern 31

(84) EFOO Schaltet blinkenden Cursor auf der Anzeige aus.

Syntax des Aufrufes: CD 84

veränderte Register: keine
(787C) Bit 0 u. 1 werden gelöscht

(32) D071 Die 16-Bit Adresse im U-Reg wird auf den Basic-Stack
berettet. Der Pointer auf diesen Stack wird um 2 inkrementiert.

Syntax des Aufrufes: CD 32

veränderte Register:

X-Reg zeigt auf die nächst
A enthält das Low-F
ck.

(30) DC16

Der Pointer auf U-Reg
8 dekrementiert
Stack gerettet

Syntax des Aufrufes:

veränderte Register:

X-Reg zeigt auf die nächste
UL-Reg zeigt auf die nächste
UL-Reg zeigt auf die nächste
UL-Reg zeigt auf die nächste

(3A) CFFV

Das Buch 'ROM-Unterschiede' wird
aufgerufen und alle Unterprogramme
ausgeführt

Syntax des Aufrufes: CD 3A

veränderte Register:

Bit 5-7 werden gelöscht
? ?

(96) EA7B

Using-Formatierung. Der numerische Wert im AR-X
wird entsprechend dem vorgewählten USING/ Format
im AR-Y und AR-U (Vorkomastellen) sowie AR-V
und AR-W (Nachkommastellen) zur Ausgabe bereit-
gestellt.

Syntax des Aufrufes: CD 96

veränderte Register:

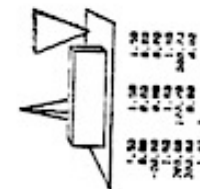
X-Reg, Y-Reg, U-Reg, A



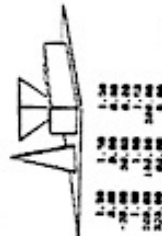
FILE: 3-4-85
OBJ: 3-4
VIB: 8000
REV: 2
STP: 2-1



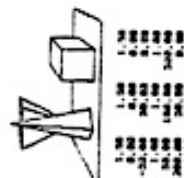
FILE: 3-4-85
OBJ: 3-4
VIB: 8000
REV: 2
STP: 2-1



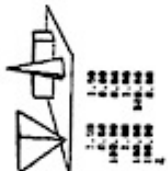
2-00 1.00 0.50
2-00 1.00 0.50
2-00 1.00 0.50
2-00 1.00 0.50
2-00 1.00 0.50



2-00 1.00 0.50
2-00 1.00 0.50
2-00 1.00 0.50
2-00 1.00 0.50
2-00 1.00 0.50



2-00 1.00 0.50
2-00 1.00 0.50
2-00 1.00 0.50
2-00 1.00 0.50
2-00 1.00 0.50



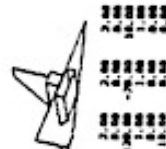
2-00 1.00 0.50
2-00 1.00 0.50
2-00 1.00 0.50
2-00 1.00 0.50
2-00 1.00 0.50



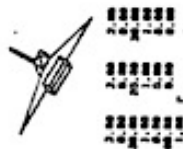
2-00 1.00 0.50
2-00 1.00 0.50
2-00 1.00 0.50
2-00 1.00 0.50
2-00 1.00 0.50



2-00 1.00 0.50
2-00 1.00 0.50
2-00 1.00 0.50
2-00 1.00 0.50
2-00 1.00 0.50



2-00 1.00 0.50
2-00 1.00 0.50
2-00 1.00 0.50
2-00 1.00 0.50
2-00 1.00 0.50



2-00 1.00 0.50
2-00 1.00 0.50
2-00 1.00 0.50
2-00 1.00 0.50
2-00 1.00 0.50

Produktinformation

Diskettenlaufwerk zum PC-1500/1500 A

Zur Verarbeitung größerer Datenmengen reicht die interne Kapazität des Rechners oft nicht aus. Beim Arbeiten mit Cassettenspeichern sind die Zugriffszeiten zulange. Außerdem ist es nur möglich sequentielle Files zu verarbeiten.

Um gezielt Informationen aus einem großem Datenbestand einzulesen bzw. zu verarbeiten kann kostengünstig ein Diskettenlaufwerk angeschlossen werden.

Mir haben das Diskettenlaufwerk VC-1541 von Commodore modifiziert, sodaß es über die serieller Schnittstelle CE-158 an den PC-1500/1500 A angeschlossen werden kann. Die Schnittstelle CE-158 verfügt außerdem noch über eine Parallelschnittstelle; u.a. ist der Anschluß eines Druckers und Datenfernübertragung möglich.

Derzeit sind folgende Möglichkeiten realisiert:

- Disketten formatieren
- Directory lesen
- Dateien abspeichern und einlesen
- Programme abspeichern (*)
- Programme und Dateien löschen

Alle Befehle zur Diskettensteuerung und Verwaltung werden in einem Programm auf Cassette mitgeliefert. Dieses Programm müssen Sie mittels HERGE mit Ihrem Programm verbinden. Einzelheiten können Sie der mitgelieferten Bedienungsanleitung entnehmen.

(*) Nach dem derzeitigen Stand ist es leider noch nicht möglich Programme vom Diskettenlaufwerk einzulesen, weil der PC-1500 den Befehl CLOAD nur als DIRECT COMMAND kennt. An einer Abhilfe wird gearbeitet. Dies ist jedoch nur ein Softwareproblem. Diskettenlaufwerke, die derzeit geliefert werden, müssen nicht umgebaut werden.

Sie müssen derzeit wie bisher Ihre Programme über Cassettenspeicher einlesen. Sie können jetzt aber mit Ihrem Programm Dateien bis zu einem Umfang von ca. 170.000 Zeichen (z.B. 1000 Adressen mit einem Informationsgehalt von jeweils 170 Buchstaben) anlegen und im direkten Zugriff bearbeiten.

Lieferumfang: Commodore Diskettenlaufwerke VC-1541
Anschlußkabel an Sharp CE-158
Dt. Bedienungsanleitung zum VC-1541/PC-1500 oder PC-1500 A

Garantie: 1/2 Jahr
Lieferung: derzeit 14 Tage Lieferzeit

Preisstellung: 1298,-,-DM incl. Mwst.

Technische Daten des VC-1540/41 Diskettenlaufwerkes

Speicherung

Totale Speicherkapazität 17488 Bytes pro Diskette
Sequentielle Files 16856 Bytes pro Diskette
Relative Files 16732 Bytes pro Diskette
65535 Records pro File
Einträge in die Directory 144 pro Diskette
Sektoren pro Spur 17 bis 21
Bytes pro Sektor 256
Spuren 35
Blöcke 683 (664 Blöcke frei)

IC's:

6502 Mikroprozessor
6522 (2) I/O Timer
Puffer: 2114 (4) 2 KRAM

Abmessungen

Höhe 97 mm
Breite 200 mm
Tiefe 374 mm

Stromversorgung

Spannung 100, 120, 220 oder 240 Volt
Frequenz 50 oder 60 Hz
Leistungsaufnahme 25 Watt

Speichermedium

Disketten Standard 5 1/4", einfache Schreibdicke

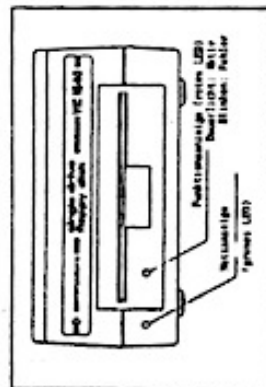


Abb. 1. FRONTPLATTE

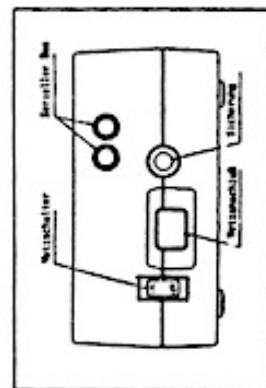


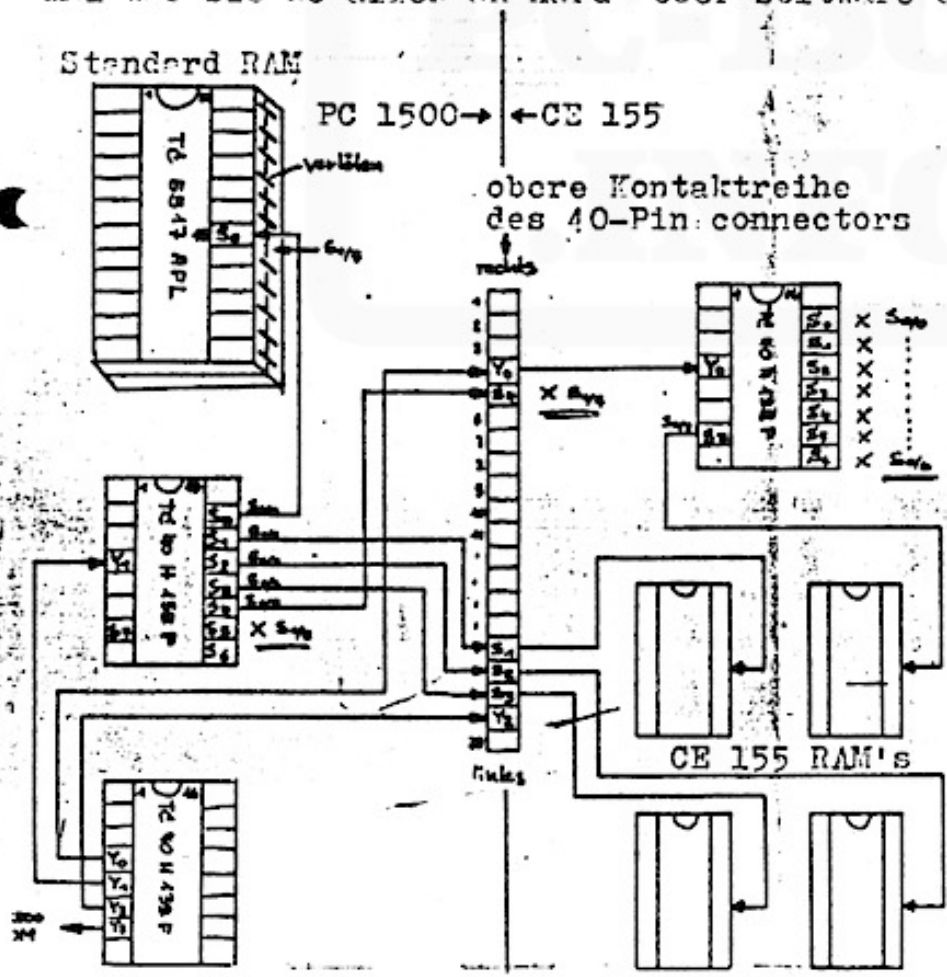
Abb. 2. RÜCKSEITE

Erweiterung des Sharp PC-1500 auf 28 KByte RAM

Im Baustein TC 40 H 139 F wird der Adreßbereich von 64K in vier Bereiche von 16K aufgeteilt Y_0, Y_1, Y_2, Y_3 . (Mit PEEK -Adresse- kann jeder dieser 65.536 Speicherplätze abgerufen, mit POKE -Adresse-, -Zahl- gefüllt werden, sofern dort RAM angeschlossen ist). Die ersten beiden 16K Bereiche Y_0, Y_1 werden dann mit den Bausteinen TC 40 H 138 F in jeweils acht 2K Bereiche unterteilt $S_{0/0}$ bis $S_{0/7}$ bzw. $S_{1/0}$ bis $S_{1/7}$.

Mit diesen Ausgängen werden vorhandene 2K RAM Bausteine TC 5517 AF angesteuert. In der Grundversion sind an $S_{1/0}$ das Standard RAM und an $S_{1/6}$ und $S_{1/7}$ das System RAM angeschlossen. In der Erweiterung CE 155, in der sich das zweite TC 40 H 138 F befindet, werden vier weitere RAM's an $S_{0/7}, S_{1/1}, S_{1/2}$ und $S_{1/3}$ angeschlossen. Die Ausgänge $Y_0, S_{1/1}$ bis $S_{1/4}$ liegen dafür an der 40-Pin Buchse. Da bei den RAM Bausteinen-bis auf einen-alle Eingänge parallel geschlossen werden, kann man durch Auflegen und Herablößen der einzelnen Beinchen beliebig viele RAM Bausteine stapeln. Lediglich die Ansteuerung muß mittels lackiertem Kupferdraht an das Schalt-Chip angeschlossen werden. Ich hab' am CE 155 die beiden oberen RAM's um jeweils zwei, die beiden unteren um ein Chip aufgestockt. Außerdem wurde gegenüber der Kontaktleiste ein Doppelpack mit den Beinchen nach oben an das dortige RAM angelötet, die außenliegenden Beinchen wurden mit Draht an das RAM der Unterseite gelegt. Die letzte Erweiterung fand im PC 1500 statt, dort erfuhr das Standard RAM eine Huckepack Erweiterung, die Ansteuerung hierfür wurde mit Draht an das Schalt Chip auf der anderen Platine gelegt. Am Batteriefach und im Modul-fachdeckel mußte zwar etwas plastik weggenommen werden dies ist aber für einen STATUS 0 von 28.474! hinnehmbar, auch muß man nicht um 18K erweitern nur sollte man dann darauf achten, daß in der Adressierung keine Lücke entsteht. Die RAM Bausteine hab' ich von der Firma Bitronic, Dingolfinger Straße 6, 8000 München 80, ich hab' dafür nur NN 205,59 DM für neun Stück bezahlt. Viel Spaß beim Lötten und schreiben Sie mir doch mal was Sie so alles an Hard- oder Software entwickelt haben.

Dieter Härtel
Heidelberger Straße 81b
6100 Darmstadt
06151 / 315 395



Adreßraum (K)

X	$S_{0/0}$	0
X	$S_{0/1}$	200
X	$S_{0/2}$	400
X	$S_{0/3}$	1000
X	$S_{0/4}$	1200
X	$S_{0/5}$	2000
X	$S_{0/6}$	2200
X	$S_{0/7}$	3000
X	$S_{1/0}$	3200
X	$S_{1/1}$	4000
X	$S_{1/2}$	4200
X	$S_{1/3}$	5000
X	$S_{1/4}$	5200
X	$S_{1/5}$	6000
X	$S_{1/6}$	6200
X	$S_{1/7}$	7000
X	$S_{1/8}$	7200
X	$S_{1/9}$	8000

free (CE 155)
free (CE 155)
free (CE 155)
free (CE 155)
CE 155 (CE 155)
Standard
CE 155 (CE 155)
CE 155 (CE 155)
CE 155
free
System

Dieser Kurs richtet sich an PC-1500-Benutzer, die im Umgang mit der Programmiersprache BASIC noch unerfahren sind bzw. BASIC erlernen wollen. Die Abkürzung BASIC steht für Beginners All-Purpose Symbolic Instructions Code und wurde 1965 an Dartmouth College entwickelt.

Die großen Vorzüge dieser Sprache sind :

1. leichte Erlernbarkeit
2. schrittweise Programmentwicklung
3. Dialogfähigkeit

Bei der folgenden Einführung wird davon ausgegangen, daß der PC-1500 betriebsbereit ist und im Programmmodus -PRO- mit -NEW- initialisiert wurde. Es wird empfohlen mit dem Printer und Cassette Interface CE-150 zu arbeiten. Dies ist eine Empfehlung und bedeutet keine Einschränkung beim durcharbeiten dieses Kurses!

Folgende Vereinbarung soll zum besseren Verständnis beitragen :
Sämtliche erlernten Sprachkonstrukte werden zur Vertiefung in Mini-Programmen angewendet. Sollten trotzdem Unklarheiten auftauchen, so soll die Bedienungsanleitung zu Rate gezogen werden.
Anmerkung des Verfassers : Diese Einführung wurde mit dem Textprogramm THA erstellt.

....und nun wünsche ich viel Erfolg !!!

1. Aufgabe :

Es soll unser Vorname vom Rechner eingelesen werden und anschließend im Display angezeigt werden.

Zum Abspeichern des Namens benötigt man einen Speicherplatz für eine Zeichenfolge; auf den man über den Variablennamen zugreifen kann. Diese Variable nennt man Textvariable. Eine Textvariable wird immer mit einem \$-Zeichen abgeschlossen und enthält maximal 16 Zeichen. Beispiele hierfür sind >AK oder >NAME\$. Eine Textvariable kann maximal 16 Zeichen enthalten.

Zusätzlich benötigt man die Sprachkonstrukte INPUT und PRINT.

Mit INPUT weist man Variablen Werte zu, die über die Tastatur eingegeben werden. Beim Erreichen einer INPUT-Anweisung, meldet sich der Rechner mit einem ? und erwartet eine Eingabe über die Tastatur. Nachdem die Eingabe abgeschlossen ist, muß die ENTER-Taste gedrückt werden.

Mit der PRINT-Anweisung werden Zeichenfolgen und numerische Werte auf der Anzeige ausgegeben.

Sowohl die INPUT als auch die PRINT-Anweisung sind wesentlich mächtiger als hier beschrieben. Dies soll aber für das erste kleine Programm vernachlässigt werden.

Mit diesen Anweisungen sind wir bereits in der Lage das folgende Programm im Programmmodus -PRO- einzutippen. Am Anfang einer jeden Zeile muß eine Zeilennummer stehen :

```
10 INPUT NAME$
20 PRINT NAME$
REM Einlesen des Vornamens
REM Ausgabe des Namens im Display
```

Es ist nicht nötig, daß ab der REM-Anweisung der Kommentar ebenfalls eingetippt wird, da bei der Programmausführung die REM-Anweisung und der folgende Text ignoriert wird.

Jetzt können Sie in den RUN-Modus umschalten und das Programm mit RUN starten. Beim Erscheinen des Fragezeichens tippen Sie einen Text ein und drücken die ENTER-Taste, danach wird der eingegebene Text im Display angezeigt...

PC - 1 5 0 0 A

Als Erweiterung des PC-1500 und anscheinend auch als dessen Nachfolger hat SHARP jetzt den PC-1500A auf den Markt gebracht. Im folgenden soll kurz erläutert werden, was es mit diesem Gerät, außer dem deutlich höherem Preis gegenüber dem PC-1500, auf sich hat.

Rein äußerlich unterscheiden sich die beiden Rechner nur geringfügig. Beim 1500A sind die Reserve-Tasten und die ON- und OFF-Tasten etwas deutlicher von den übrigen Tasten abgehoben, und die Farbgebung des Gehäuses ist etwas dunkler ausgefallen. Es handelt sich ansonsten jedoch um exakt dieselbe Außenhaut, die auch der 1500 besitzt.

Der wesentliche Unterschied des 1500A zum 1500 liegt im Inneren des Gerätes. Es verfügt schon in der Grundausstattung über einen Basic-Programmspeicher von 5946 Bytes. Hinzu kommt als Neuheit ein extra Speicherbereich von 1 KByte, der Maschinensprachenprogrammen vorbehalten ist und in den Adressen &7C01 - &7FFF liegt. Der Hauptprogrammspeicher nimmt ohne Erweiterungsmodule den Adressbereich &40C6 - &5800 ein.

Abgesehen von diesen Speichererweiterungen läßt sich der 1500A jedoch genauso programmieren wie auch der 1500. Dies wird durch die mitgelieferte Begleitliteratur bestätigt. Lediglich an Stellen, die die Speicherkapazität betreffen, weicht die Bedienungsanleitung des 1500A von der des 1500 ab.

Alles in allem kann man sagen, daß SHARP einen entscheidenden Nachteil des 1500, nämlich die minimale Speicherkapazität in der Grundausstattung, durch die Einführung des 1500A endlich behoben hat.

Ann.: Natürliche lassen sich sämtliche Erweiterungsmodule und Peripheriegeräte des 1500 auch in Verbindung mit dem 1500A verwenden.

```

10:"START $4088,&
4AFF,&371D,&37
6F
28:";VARIABLE
30:" PRGBEG=&7865
48:" PRGEND=&7867
50:" ZWSP=&376F
58:";-----
78:"SAVE
88:"LD X,PRGBEG
;UEBERGABE H-B
YTE
98:"LD A,(X)
100:"LD (ZWSP),A
110:"LD X,PRGBEG+J
;UEBERGABE L
-BYTE
120:"LD A,(X)
130:"LD (ZWSP+J),A
140:"LD X,PRGEND
150:"LD A,(X)
160:"LD (ZWSP+2),A
170:"LD X,PRGEND+J
180:"LD A,(X)
190:"LD (ZWSP+3),A
200:"LD A,(ZWSP)
210:"LD XH,A
220:"LD A,(ZWSP+J)
230:"LD XL,A
240:"LD A,(X)
250:"LD (ZWSP+4),A
260:"RET
270:"GET
280:"LD A,(ZWSP)
290:"LD (PRGBEG),A
300:"LD A,(ZWSP+1)
310:"LD (PRGBEG+1)
,A
320:"LD A,(ZWSP+2)
330:"LD (PRGEND),A
340:"LD A,(ZWSP+3)
350:"LD (PRGEND+1)
,A
360:"LD A,(ZWSP)
370:"LD XH,A
380:"LD A,(ZWSP+1)
390:"LD XL,A
400:"LD A,(ZWSP+4)
410:"LD (X),A
420:"RET
430:"END

```

```

PROGRAM: &371D
&376E
&371D
&376E
CODE:
SYMBOL-TABLE:
SAVE &371D
GET &374A
PRGBEG &7865
PRGEND &7867
ZWSP &376F
PROGRAM-CODE:
371D: 48 78 4A 65
3721: 05 AE 37 6F
3725: 48 78 4A 66
3729: 05 AE 37 78
372D: 48 78 4A 67
3731: 05 AE 37 71
3735: 48 78 4A 68
3739: 05 AE 37 72
373D: A5 37 6F 88
3741: A5 37 78 8A
3745: 85 AE 37 73
3749: 9A A5 37 6F
374D: AE 78 65 A5
3751: 37 78 AE 78
3755: 66 A5 37 71
3759: AE 78 67 A5
375D: 37 72 AE 78
3761: 68 A5 37 6F
3765: 88 A5 37 78
3769: 8A A5 37 73
376D: 0E 9A

```

Heinz Schlottfeldt
Braunschweiger Str. 68
1000 Berlin 44
Telefon 030/6849908

Betr.: PC-1500 Zeitung, Heft 7
Programm KEEP S.27

Ich habe das Programm KEEP ausprobiert, mußte jedoch feststellen, daß es nicht ganz vollständig arbeitet. Der Autor berücksichtigt zwar die Systemadressen von je 2 Byte für Programmspeichernde (7867H) und Programmspeicheranfang (7865H), jedoch nicht die Tatsache, daß bei dem durch NEW unsichtbar gewordenem, aber noch gesicherten Programm das erste Byte auf FFH gesetzt wird. Da die ersten 2 Byte die BASIC-Zeilenummer beinhalten, erhält das mit KEEP sichtbar gemachte BASIC-Programm eine erste Zeilennummer, die über FFxH (also über65280) liegt. Diese Zeile ist nicht editierbar und führt bei Aufruf des Programms mit RUN zur Fehlermeldung.

Dies habe ich mit meinem beiliegendem Programm, bestehend aus zwei Teilen, versucht zu berücksichtigen. Es ist voll relokatable und benötigt 85 Byte Speicherplatz, wovon das Programm 81 und ein Zwischenspeicher (ZWSP) 4 Byte verbraucht.

Arbeitsweise: 1. SAVE

Aufruf hier z.B. CALL &371D
bewirkt nach fertigstellen eines BASIC-Programms die Sicherung der oben genannten Werte

2. GET
Aufruf hier z.B. CALL &374A
bewirkt nach versehentlicher Eingabe von NEW, daß die unter 1. gesicherten Werte wieder zurückgesetzt werden

Die beiden Programme wurden mit Hilfe des RVS Hexmonitors und Macro-Assemblers entwickelt. Sollten trotz aller Sorgfalt Fehler aufgetreten sein, so bitte ich um Hinweise darauf.
Mit freundlichen Grüßen

Heinz Schlottfeldt

Heinz Schlotfeldt
Braunschweiger Str. 88
1000 Berlin 44
Telefon 030/984 99 06

Betr.: Endlos - Cassetten
Ich habe Endlos - Cassetten bei:
Atzert Radio
Kleiststr. 32-33
1000 Berlin 30

- erhalten. Sie sind erhältlich mit Spieldauern von 30s, 1min, 3min und 6min. Die Preise liegen zwischen 9 und 12DM.
- Besonderheiten: 1. Nur Seite A bespielbar
- 2. Schnell vor bzw. zurückspulen nicht möglich
- 3. Keine Verwendung von Auto-Reverse-Rekordern möglich
- 4. Nur horizontaler Betrieb möglich

Ich hoffe, Ihnen mit dieser Information gedient zu haben.

Mit freundlichen Grüßen

Heinz Schlotfeldt

MASCHINENSPRACHE-RENUMBER

Dieses kurze, 29 bytes lange Maschinenprogramm ermöglicht ein extrem schnelles Umnummerieren der Programmzeilen. Sein Nachteil ist, daß Sprungbefehle (GOTO, GOSUB, THEN) nicht berücksichtigt werden. Es empfiehlt sich also, diese Befehle nur in Verbindung mit Labels (z.B. GOTO "A") zu verwenden, um sich ein nachträgliches Korrigieren von Hand zu ersparen.

Das Programm wird am besten an den Anfang des Programmspeichers geschrieben. Zur Schaffung des nötigen Platzes, sind folgende Befehle auszuführen:

- 1. NEW ϕ ! 2. NEW STATUS 2+29

Die Eingabe selbst kann, auf Grund der Kürze des Programmes, durch POKE-Befehle erfolgen.

Listing :

68 $\phi\phi$	LD UH, ϕ	Laden der Adresse
6A $\phi\phi$	LD UL, ϕ	des Anfanges des
A5 78 65	LD A, (&7865)	Basicprogramm-
$\phi\phi$	LD XH, A	speichers.
A5 78 66	LD A, (&7866)	Inkrementfestsetzung
$\phi\phi$	LD XL, A	(hier wurde das
B5 $\phi\phi$	LD A, &A ϕ	Inkrement mit
FD EA	ADD U, A	1 ϕ =&A ϕ gewählt,
A4	LD A, UH	es sind jedoch
41	LDI (X), A	alle Werte von
24	LD A, UL	1-255 möglich.)
41	LDI (X), A	
45	LDI A, (X)	
FD CA	ADD X, A	
B5 FF	LD A, &FF	Abfrage des Endes
$\phi\phi$	CP A, (X)	des Programmes.
99 1 ϕ	JR NZ, -&1 ϕ	
9A	RET	

Änderung des Inkrementwertes nach Eingabe des Programmes: POKE Startadr. Ma.-Pro. +13, Inkrement

Monsieur FOJUD Xavier

5 avenue Debussy
57150 CREUTZWALD

FRANCE

METZ

CREUTZWALD

SAARBRUCKEN

STRASBOURG

Sehr geehrter Herr FISCHEL,

le 18 octobre 1983

Zuerst dank ich Ihnen für die Zusendung eines Exemplar der PC 1500 Zeitschrift. Die Informationen die ich sah haben mich überzeugt und ich will ein Abonnement verlangen.

Der CE 153 ist im Verkauf in Frankreich aber es gibt noch keinen Soft.

der "Sharpkrankheit" verfallen.

Ich bin tatsächlich

le 24 octobre 1983

Laut dem Technical reference manual ist es möglich ein Autostart zu generieren mit dem dritten Befehl von CSAVE M leider nach vielen Versuche mit CLOAD M blockiert sich immer wieder die Maschine. Ist es jemand gelungen? Wenn ja würde ich gerne wissen wie. Bessen Dank an Sie für die Erläuterung.

le 26 octobre 1983

le 29 octobre 1983

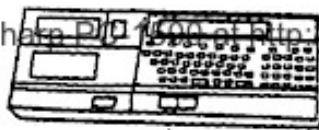
Ich muss Ihnen gestehen dass ich mich ganz gewaltig anstrenge wenn ich deutsch lese oder schreibe auch war ich angenehm überrascht plötzlich einige doch ganz gute französische Sätze in Ihrem Brief zu finden da musste ich mich in "MODE Français" umstellen .

Ich werde dem nächst Auskunft bekommen von Logis-Tick über Programme PC PLOT, GRAPH und WORD. SYBEX schlägt 2 Bücher für PC 1500 (Basic programm) das Erste Buch geht den PC 1500 an und das Zweite den Plotter. Ich glaube dass man sie in Deutschland in deutscher Sprache kaufen kann.

Wir sind ungefähr 16000 Einwohner die Hauptindustrie ist die Kohlengrube und GRUNDIG (Fernsehen). Rund um das Städtchen das ungefähr 1 km von der Grenze liegt sind grosse Waldflächen Daher der Wald im Namen. Früher waren es drei Orte (1602) CROIX (Kreuz) HOUVE und WILHELMSBRONN - NASSAU Glasindustrie und Schmiederei Giesserei Kupfergruben da war der Wald von grossen Nutzen (30000 Kubikmeter pro Jahr). 1810 wurde Creutzwald aus den drei Dörfer gegründet Heute bleibt kaum was aus dieser Vergangenheit.

Salut





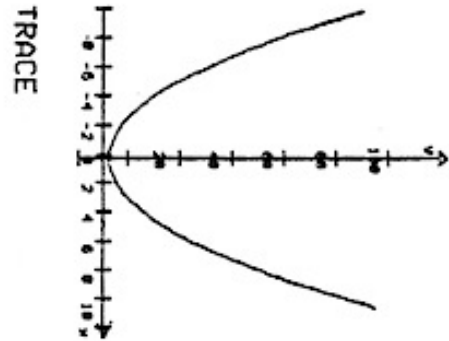
SHARP-SOFTWARE
FORT Olivier
Les Présents BORRELS
83400 HYÈRES FRANCE

1/2+1/3+1/4
13/12
BRUCHTEIL RECH.

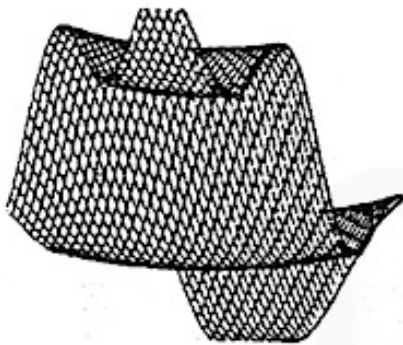
Basis : ? 2
Zahl : ? 12
1100

Basis : ? 16
Zahl : ? 127
7F

MULTI-BASE



(C) 1982



TRACE 3D
48014=128K+17112ACOS R-COS (38H)
TIME 00:12.7051
20.00/170025
20

0 5 10 15 25

30 1 7 13 19
12 18 29 5 6
4 10 11 17 28
16 27 3 9 15
8 14 20 26 2

30 1 7 14 18
12 19 28 5 6
3 10 11 17 29
16 27 4 8 15
9 13 20 26 2

30 1 8 12 19
13 17 29 5 6
4 10 11 18 27
16 28 2 9 15
7 14 20 26 3

MAGISCHE QUADRATE

SHARP

```
1;"Lissajous"BEEP ON :WAIT 200:8EEP 3:PRINT "(C) OLF0-Software 1982
10;LOCK :CLEAR :DEGREE :TEXT :CURSOR 4:PRINT "** Figure de Lissajous **
20;INPUT "1.Frequence ? ";F1
30;IF F1<=0OR F1>2E7THEN 20
40;INPUT "2.Frequence ? ";F2
50;IF F2<=0OR F2>2E7THEN 40
60;INPUT "1.Amplitude (Largeur)? ";A1
70;IF A1<20OR A1>100THEN 60
80;INPUT "2.Amplitude (Hauteur)? ";A2
```

OWNER INQUIRY

VERSUCH !
START ADD.

20480

20480: 101: 0: LDA, 000
20482: 72: 250: LDB, 0FF
20484: 100: 2: JPE, 02
FLIER BEWEGUNG
1ST AUCH PLATZ
20480: 154: RET
END: 7 BYTES

HEX	00	00	12	F8	57	22	20	7F	ROH1
DEC	20	20	20	07	87	48	40	127	1 199
HEX	31	20	20	20	20	22	80	7F	7F
DEC	80	14	18	F8	57	22	132	127	127
HEX	03	00	72	00	02	00	00	00	00
DEC	20	00	00	20	00	00	00	00	00
HEX	20	20	80	20	50	80	00	00	00
DEC	20	20	128	20	80	128	00	00	00
HEX	15	10	F8	57	22	4C	03	00	00
DEC	21	16	F8	57	22	76	3	00	00
HEX	40	00	00	20	00	00	20	00	00
DEC	64	00	00	32	00	00	32	00	00

MONITOR

ZEILE FORSCHUNG

Programme No 1		
Zeile No	Adresse	Laenge
1	14533	53
2	14580	52
3	14648	21
4	14663	63
5	14732	62
6	14771	0

CD\$*16 24500
BO\$*16 24523
BB 24546
AA 24561
24576

Total 76 BYTES

10:CAT 14533
14763 END 9812

CATALOG



UHR



OLIVIER KURFURSTER
BERLIN 28
ALFRED ORANIENBURGER
POSDAEN

SEND PROGRAM BOUNTY OF PC-1500
ALL BOUNTY INFORMATION:
SCHREIBMACHINE

ADRESVERWALTUNG

CATALOG

Multi-Use Character
NAME
CAT. LABEL
PROGRAM #

NAME	PROGRAM #	PRICE
ADRESVERWALTUNG	1	500.00
ADRESVERWALTUNG	2	500.00
ADRESVERWALTUNG	3	500.00
ADRESVERWALTUNG	4	500.00
ADRESVERWALTUNG	5	500.00

TAPEVERWALTUNG

NAME	PROGRAM #	PRICE
ADRESVERWALTUNG	1	500.00
ADRESVERWALTUNG	2	500.00
ADRESVERWALTUNG	3	500.00
ADRESVERWALTUNG	4	500.00
ADRESVERWALTUNG	5	500.00

NAME	PROGRAM #	PRICE	DATE
ADRESVERWALTUNG	1	500.00	1978
ADRESVERWALTUNG	2	500.00	1978
ADRESVERWALTUNG	3	500.00	1978
ADRESVERWALTUNG	4	500.00	1978
ADRESVERWALTUNG	5	500.00	1978

REIHE	AUSGABE	EINCOMFEN	KOSTEN	GEWINN
1978	1,250.00	2,000.00	500.00	500.00
1980	1,500.00	3,000.00	800.00	700.00
1988	1,700.00	3,500.00	400.00	1,400.00
78-88	4,450.00	8,500.00	1,700.00	2,600.00

2r

\$ @ < : 1
SATHELN

BASKET-BALL

270

(((
)>)
(((
)?)
NO ANS. IS : 2
00000
88888
00000
S=?
D.K
SUCHGANG(AUF DISP.

SKI

421 WURFEL

ALTIEN LAND

AUTORENNEN 1

* SUSPENS *

JACK-POI

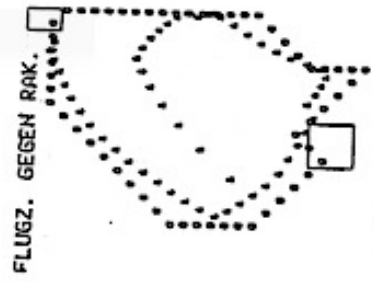
BACCARA

P->SIBEN ♦ FUNF †

10/29 L KRIEG DER STERNE

U-BOOT JAGI

SHARP



FLUGZ. GEGEN RAK.

PUNKT:
1 / 2 / 3

Rythmus:
Quadrat=8 Rund=1
Dopp ton=2 ton=3
Ton Triolett=4
Halb-ton=5
Halbton triolett=6
Zhalbton=7 Sext=8
3halbton=9

Octaves:
Baessen=B
Hoehoer=H

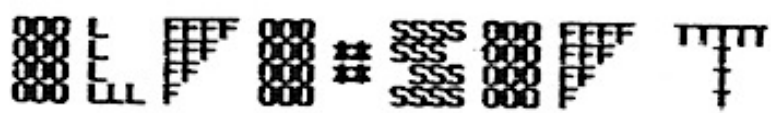
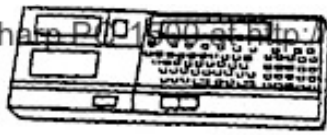
Beschaedi.:
Becarre=SPACE
Die=D
Bemol=B

Notes:
DO=D RE=R MI=M
FA=F SOL=SO LA=L
SI=SI

CODES FUER
MUSIC schreiben
mit PC-1500



LABORATION 1



RADIUS 10
ALPHA 100
BETA 300
PHI 10
WINKEL 0
SCHWITT 0

98119.202



RADIUS 22
ALPHA 10
BETA 10
PHI 20
WINKEL 45
SCHWITT 10

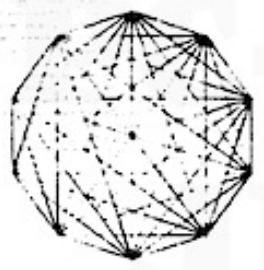
98119.1244



ASTERIX

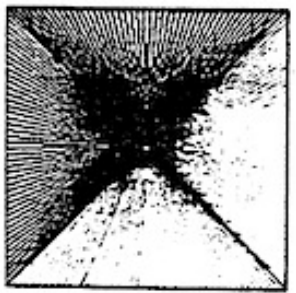
SNOOPY

POLYGONES

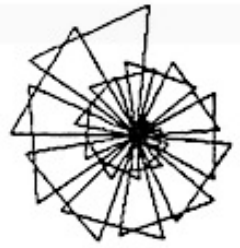


RADIUS 30
ALPHA 22
BETA 10
PHI 10
WINKEL 30
SCHWITT 10

98119.4010



UIERECK

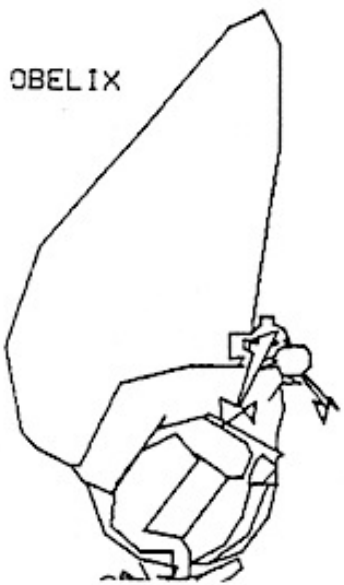


GRAPH 1

OBELIX



GRAPH 2



MZ-80B

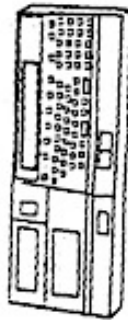
TRACE 3D : KURVE IN 3 DIMENSION 30 DM
 TRCE : 2 DIM. + KURVE STUDIUM 15 DM
 MAGISCHE QUADRATE : 5 DM
 MULTI-BASIS : BASE n (->) BASE m 5 DM
 BRUCHTEIL RECHN. : 5 DM

---GESCHAEFTLI. PGM---
 MINI-CALC : ERSTAUNLICHEN MOEGlichkeiten
 LEISTUNGSFAEHIG + EINFACH : NUR 150 DM
 ADRESSVERWALTUNG : 5 DM
 TAPEVERWALT. : AM ANFANG JEDE KAS. 10 DM
 JHR : DIGITAL + ANZEIGE. 5 DM
 SCHREIBMACHINE : 5 DM
 KALENDER : KALENDER WIRD GEPLOTTET 10 DM

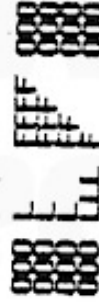
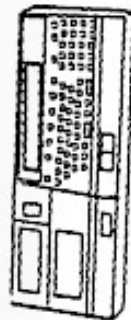
---BASIC ERWEITERUNG IN BASIC+MS ---
 QWER LISTIG : 2 GROESSE MOEGlich 15 DM
 MACRO ASSEMBLER : 280 ART 15 DM
 MONITOR : IDEAL FUER MS, VIELSEITIG 15 DM
 CATALOG : LISTET ALLE PGM VON PC 5 DM
 VARIABLE LISTING : (+ ADRESSE) 5 DM
 ZEILE AUSFORSCHEN : 5 DM
 A\$FONC : VERWENDET A\$ IN EINE FCT. 5 DM
 DELETE : 5 DM
 CSAVE VON/BIS : EINSEL ZEILEN SAVE 5 DM
 RENUMBER : MIT THEN & GOTO 10 DM
 ENT&VER : MS IN REM EINGEBEN 5 DM
 SCHUTZ : SCHUTZ PROGRAMMEN 10 DM
 ATARI SOUND : TOENE WIE BEI ATARI 5 DM
 HARD-COPY : DISPLAY GEPLOTTET 5 DM

---MS PGM = KURTZ+SCHNELL---
 LEFT : DISPLAY 1 SPALTE NACH LINKS 5 DM
 RIGHT : DISPLAY 1 SPA. NACH RECHTS 5 DM
 DEC-HEX : DECIMAL => HEXADEZIMAL 5 DM
 REN/DISP/DEL : 3 NEUEN BASIC BEFEHLEN 10 DM
 RENUMBER ~~mit THEN&GOTO~~ / DELETEN-m
 DISPLAY n VERSCHIB DAS DISP. UM n ZEICHEN
 DIRECT IN DIE FONCTION TASTE, NUR 150 DM

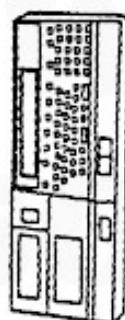
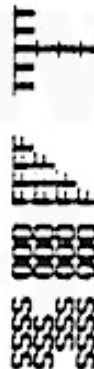
SHARP



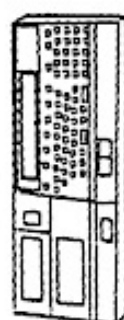
SHARP



##



SHARP



SHARP

 * PROGRAMM ANGEBOT * FUER PC-1500 *

---DISPLAY GAMES---
 SUCHGAME : REACTION SPIEL 5 DM
 KRIEG D. S. : FLUGSIMUL. IM RAUM 15 DM
 ALIEN LAND : MINI SPACE-INV. 10 DM
 JUMP BOY : NUR MIT GUTEN SEHUERM. 10 DM
 RAUBER : UNBEKANTES HAUS RAUBEN 10 DM
 AUTOREN.1 : HALB MS, SEHR SCHNELL 15 DM
 \$ SAMMELN : DENK+REACTION SPIEL 5 DM
 SKI : MIT 5 SCHWIERIGKEITSGRAD 10 DM
 1421? : WURFELSPIEL + VIEL EXTRAS 10 DM
 BASKET : WIRD IMMER SCHWIRIGER 10 DM
 ADVENTURE SPIEL : WIE EIN ECHE 10 DM
 JACK POT, : GELD BLEIBT ZU HAUSE 10 DM
 U-BOOT JAGT : OHNE RISIKO 10 DM
 BACCARA : WIE IM KASINO 10 DM

--- GAMES ---
 MUSIC SCHR. : MIT ALLEN RAFINESSEN 15 DM
 VIER IN REIHE : LANGSAM ABER GUT 10 DM
 FLUGZEUG (SIE) GEGEN RAKETE (PC) : 10 DM
 LABYRINTH 1 : NOCH IM GRIFF 10 DM
 LABYRINTH 2 : VIEL SPASS ? 15 DM
 MEMO PIANO : PC IST EIN KUENSTLER 5 DM

---GRAPH. DARSCHT. ---
 ERDE : AUS JEDEM WINKEL, JEDE GROE. 30 DM
 PLOT HILF PGM : EINFACHES ZEICHNEN 10 DM
 GRAPH 1+2 + POLYGONES : 5 DM
 FRANCE : 5 DM
 MZ-80B : 5 DM
 ASTERX + OBELIX : 15 DM
 VIERECK : 10 DM
 SNOOPY : 10 DM
 SUZI VON INTEN+VORN (OHNE ZENSUR) 20 DM

- 34 -

Andreas Donner

PC-1500 Systemssoftware

Obere Seelgasse 2

8500 Bamberg

An Firma

Bernd Fischer

Kaiser-Friedrich-Str. 54a

1000 Berlin 12

Bamberg, 18. 10. 1983

Sehr geehrter Herr Fischer,

aufgrund Ihres dankenswerten Abdrucks meines Infos "VIP" gingen bei mir zahlreiche Fragen, die Lauffähigkeit der Programme "VIP" bzw. "EDITOR" auf dem Modul SWM-22 der Fa. Kauna betreffend, ein.

Die gesamte Software des "VIP"-Pakets setzt eine fortlaufende Bytezzählung des BASIC-Speichers voraus.

Aus den mir zur Verfügung stehenden Unterlagen der Fa. Kauna geht nicht klar ersichtlich hervor, ob diese kontinuierliche Bytezzählung durch den SWM-22 gewährleistet ist. Der Arbeitsbereich des Moduls wird dort angegeben in den Bereichen 0 bis 16383 und 16432 bis 24575.

Da der RAM-Bereich des PC-1500 in der Standardversion bei 16591 beginnt, verbüßten mich diese Angaben. Mir schien hier ein Sprung um 198 Bytes vorzuliegen, von dem ich annehme, daß hier der Reserve-Bereich untergebracht wird, und daß die Lauffähigkeit von Programmen durch modulinterne Steuerung gewährleistet wird. Aus dieses Problem betreffenden Anfragen mehrerer Leser ging nun u.a. hervor, daß der Anfangsbereich des SWM-22 nicht im Bereich 0 bis 16383, sondern STA.2 ohne Programm = 198, also 197 bis 16580 (??) liegt. Somit scheint das SWM-22 die fortlaufende Bytezzählung zu garantieren.

Mit folgendem kleinen Testprogramm kann jeder Speicher auf seine Eignung auf "VIP" getestet werden:

```
10: FOR X = STATUS 2 TO STATUS 3
```

```
20: POKE X,255: IF PEEK X <> 255 PRINT "NICHT BEEIGNET"
```

```
30: POKE X,0: IF PEEK X <> 0 PRINT "NICHT BEEIGNET"
```

```
40: NEXT X: PRINT "BEEIGNET"
```

mit freundlichen Grüßen

Andreas Donner

DR. ROGER DORSCH

POSTFACH 6085

KANTSTRASSE 38/III, D-1000 BERLIN 12

TELEFON P 030 / 3 19 73 49

Der PC-1500 spielt Melodien in drei Oktaven

Die Tastatur dient zunächst zum Eingeben des Programms, welches einen Teil der Tasten so transformiert, daß sie wie bei einem Musikinstrument funktionieren. Das neue Musikinstrument besitzt einen Tonumfang von drei Oktaven.

Im Rahmen einer Folge von Programmen des Autors François Jeanvoine, in denen zur Tonerzeugung die Maschinensprache verwendet wird, bezweckt das vorliegende kurze Programm ein Abspielen von Piano-Melodien.

Die Art der Zuordnung von Noten und Tasten ist in Abb. 1 dargestellt. Den Übergang von einer der drei verfügbaren Oktaven in eine andere erreicht man durch Drücken einer der Tasten 2 oder 3.

Die Tondauer ist nach Maßgabe der Eigenschaften des PC-1500 (1... 255) variabel. Nachdem man sich vergewissert hat, daß das BASIC-Programm fehlerfrei ist, startet man mit KON. Die Befehlscodes der Maschinensprache folgen auf die Speicheradresse &7150, d.h. in P\$ Q\$, R\$, und W\$. Man achte darauf, daß diese nicht vor Inbetriebnahme des kleinen Musikinstruments gelächelt werden. Im Zweifelsfall beginne man noch einmal von vorn.

Das Musizieren wird mit DEF A begonnen. Man wählt die Tondauer zwischen 1 und 255 und drückt dann ENTER. Man breche die Musik nie mit BREAK sondern stets mit SPACE ab.

Für Freunde der Maschinensprache ist das disassemblierte Programm erhältlich. Die am Anfang stehenden Daten sind Dezimalzahlen im BASIC, die sich auf die zu erzeugenden Tonfrequenzen beziehen.

Zwei nützliche Adressen für Unterprogramme des PC-1500 sind:

CALL &B42C, d.h. ein INKEY\$ in Maschinensprache, welches den

Code der gedrückten Taste in den Speicher A zurückschreibt.

CALL &E66F erzeugt ein BREAK 1, HL,BC, wo H,L,B und C Register des Mikroprozessors sind.

Abb. 1

ERRORTABELLE

```

5: "MAT":PAUSE "M
  MATRIXBERECHNUNG
6: PAUSE "I=ZE
  ILE MAT,A U.B"
7: PAUSE "J=SPAL
  TE MAT,C"
8: PAUSE "K=SPALT
  E MAT,B"
18: CLEAR :CLS :
  WAIT 8:COLOR 0
19: LLIST 210
20: PRINT "I=
  =K="
22: CURSOR 3: INPUT
  M: CURSOR 9:
  INPUT M: CURSOR
  15: INPUT L
24: BEEP J:CLS :
  LPRINT "MATRIX
  A"
25: DIM A(M,L)
30: FOR I=1 TO M:
  FOR J=1 TO L
48: A$="G(+STR$(J
  )+",""+STR$(J
  )+)"
58: CLS :PRINT A$
68: CURSOR 11:
  INPUT A(I,J):
  LPRINT A$(A(I,
  J))
78: NEXT J:NEXT I
88: BEEP J:CLS :
  LPRINT "MATRIX
  B"
108: DIM B(M,L)
110: FOR J=1 TO L
120: FOR I=1 TO M
130: A$="b(+STR$(J
  )+",""+STR$(J
  )+)"
140: CLS :PRINT A$:
150: CURSOR 11:
  INPUT B(I,J):
  LPRINT A$(B(I,
  J))
168: NEXT J:NEXT I
178: BEEP J:CLS :
  LPRINT "MATRIX
  C"
188: DIM C(I,J)
190: FOR I=1 TO M
208: FOR J=1 TO N
218: C(I,J)=B(I,J)*
  A(I,J)
228: A$="c(+STR$(J
  )+",""+STR$(J
  )+)"
238: WAIT :CLS :
  LPRINT A$(C(I,
  J)):WAIT 8
248: NEXT J:NEXT I
258: END
  
```

Matrixberechnung
 zur Berechnung einer
 Matrix C mit J-zeilen und L-Spalten
 aus zwei Matrizen A(I,J)
 und B(I,J).
 Dabei kann die Funktion
 $C(I,J) = f(A(I,J), B(I,J))$
 jedes gewünschte Form annehmen.

Herbert Klein
 Euskirchenerstr. 764
 5040 Brühl

```

210: C(I,J)=B(I,J)*
  A(I,J)
MATRIX A
a(1,1)= 2
a(1,2)= 5
a(2,1)= 25
a(2,2)= 45
MATRIX B
b(1,1)= 30
b(1,2)= 60
b(2,1)= 45
b(2,2)= 90
MATRIX C
c(1,1)= 19
c(1,2)= 176.961524
2
c(2,1)= 156.819805
2
c(2,2)= 219.0696
  
```

```

6145: WAIT 98:
  PRINT "Basic
  -Stack ueber
  schr.":WAIT
  :PRINT "GOSU
  B o.Zerl.mat
  h.Ausdruck":
  RETURN
6146: PRINT "Zahl
  zu gross:IEI
  00/&IFFAB":
  RETURN
6147: PRINT "Daten
  art falsch
  (I+ A$)":
  RETURN
6148: PRINT "Fu,ra
  isch (Zahl d
  er Angum.":
  RETURN
6149: PRINT "DIM z
  u gross":
  RETURN
6150: PRINT "Varia
  biennamen unv
  olliast.":
  RETURN
6151: PRINT "Varia
  ble fehlt":
  RETURN
6152: PRINT "Keine
  Speichenkap
  .mehr":
  RETURN
6153: PRINT "Ausdr
  uck fuer TIM
  E falsch":
  RETURN
6156: PRINT "Instr
  .im MODE fal
  sch":RETURN
6157: PRINT "Perip
  herie nicht
  angeschl.":
  RETURN
6158: PRINT "kein
  DEF benutzen
  ":RETURN
6160: PRINT "Zeile
  nummer zu g
  ross":RETURN
6162: PRINT "GCURS
  OR zu weit r
  echts":
  RETURN
6166: PRINT "USING
  zu klein":
  RETURN
6167: PRINT "Wert
  ist groesser
  als 999":
  RETURN
6168: PRINT "Divis
  ion durch 0":
  RETURN
  
```

```

6169: PRINT "Aus
  ausdruck ist
  unzul.":
  RETURN
6170: PRINT "ungen
  spezif.Ausdr
  uck":RETURN
6172: PRINT "einzu
  ladende Date
  n zu viel":
  RETURN
6173: PRINT "Progr
  amm ist vera
  endert":
  RETURN
6174: PRINT "Bandi
  ruvo nicht le
  sbor":RETURN
6200: PRINT "Koord
  inaten ueber
  schritten":
  RETURN
6201: PRINT "Papie
  r wurde zuwe
  it bewegt":
  RETURN
6202: PRINT "TAB/L
  CURSOR Ausdr
  .falsch":
  RETURN
6203: PRINT "GRAPH
  /TEXT ueberp
  rufen!":
  RETURN
6204: PRINT "Komma
  s LINE/RLINE
  zuviel!":
  RETURN
6206: PRINT "LPRIN
  T im TEXT zu
  gross":
  RETURN
6208: PRINT "Accus
  /Schreibkopf
  pruefen!":
  RETURN
6209: PRINT "Schre
  ibkopf stand
  falsch":
  RETURN
6210: PRINT "Accus
  leer!":
  RETURN
6307: PRINT "Prog.
  /Datenfeld g
  l. Nummer":
  RETURN
6308:GOTO K
6309:GOTO K
6310:GOTO K
6311:GOTO K
6354:PRINT "nach
  I./AR. falsc
  he Daten":
  RETURN
  
```

STATUS 1
 1848

Über RUN "EX"
 und Eingabe
 des Fehlercodes
 sucht sich der
 Computer die
 passende
 Fehlermeldung

Michael Drohse

1000 Berlin 27, Oktober 1983
Tel.: (030) 433 43 58 oder
433 46 84

-36-

Zum Programm -PC 1500 Zeitung- Nr.6 Seite 32

Wie die Auflösung eines Preisdreiecks jetzt die Erklärung des Programmes und die Anknüpfung einer erweiterten Fassung.

Jeder Anwender dieses Programmes muss den Speicherbereich, in dem er das Programm ablegen will, entsprechend seiner Geräteausstattung selbst bestimmen und dann bei der Eingabe des Programmes folgende Daten des Musterprogrammes -Seite 32 umrechnen:

&I20A : das aufgerufene Unterprogramm ist n=52
&I2FD : Adressen vom Programmbeginn entfernt

&I2FB : das aufgerufene Unterprogramm ist n=54
Adressen vom Programmbeginn entfernt

Mer ohne Hexamonitor das Programm eingeben will, sollte das folgende Basic-Programm verwenden:

```
A=0 : FOR J = >Startadr.Masch.Pr.< TO >Startadr.hp + 60< :
PRINT "Adr. "; J; INPUT " : INPUT A
:POKE J,A : NEXT J
```

Bei den Adressen entsprechend &I20B, &I2FE und &I2F9 ist dann jeweils folgender Wert einzugeben:

(Startadresse + n) / 256

und bei den jeweils folgenden Adressen entsprechend &I20C, &I2FF und &I2FA der Wert

Startadr. + n - INT ((Startadr.+n)/256)*256

Erklärungen zum Programm:

Das Programm verwendet drei Sub-Routines des Systems durch direkten Aufruf und die Routinen zur Cursor-Steuerung durch ;j bzw. alle Cursor-Anweisungen.

Um Speicherplatz zu sparen und um die Verwendung von ERROR-Routinen zu erreichen, wurde die Form:

Befehl : CALL n : REM x----->

neue Zeile
PRINT

gewählt.

Dadurch werden für die Rücksprungadresse auf den Befehl nach dem REM nur 8 Bytes benötigt. Jede Änderung des Programmzählers durch das Maschinprogramm würde mehr als 8 Bytes benötigen.

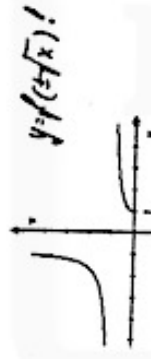
Vor dem "CALL-Befehl" muss in gleicher Zeile unbedingt ein Befehl stehen, da sonst der Pointer nicht auf das wirklich auszugebende Zeichen weist. Ein unschädlicher Versuch kann dies beweisen.

Die nächstfolgende Zeile muss mit einem PRINT-Befehl beginnen, um die mit CALL bewirkte Ausgabe im Display zu halten. Die Position der einzelnen "Ausgabe-Bausteine" kann durch

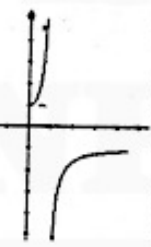
Funktionssehensbilder

```
1:"A":PAUSE "GRA
PHEN VON FUNKT
IONEN":PAUSE "
200:Y=(X)
2:INPUT "TRIGON,
FU.? JA=1: ";
0
3:IF 0=1THEN
RADIAN
4:INPUT "WURZEL
? JA=1 ";H
6:TEXT :CLS :
INPUT "FARBE X
UND Y-ACHSE
=" :D
7:INPUT "FARBE D
ES SCHAUBILDES
=" :E
8:INPUT "LAENGEN
EINHEITEN=" :L
9:INPUT "INTERVA
LLBREITE (J-20
):" :F
10:CSIZE 2:LF 5
15:GRAPH
20:LINE (0,0)-(2
16,0),0,D:LINE
(210,3)-(216,0
)-(210,-3),0
30:GLCURSOR (198,
-15)
40:CSIZE 1
50:LPRINT "x"
60:FOR J=200TO 0
STEP -20
65:LINE (J,-2)-(J
,2),0:NEXT J
66:C=0
67:IF INT (L/5)=L
/5THEN 70
68:C=C-22:USING "
#####.#"
70:GLCURSOR (120+
C,-15):LPRINT
L/5:USING
```

Leichte Handlung
des Programms des
Herrn Wolfgang
Skibler aus
"Hobby Computer" 10/83



```
200:Y=-J*((X-1)/(X+1
))
```



```
200:Y=-J*((X-1)/(X+
1))
```

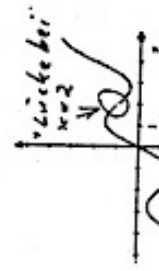
Eingabe der zu untersu-
chenden Funktion

Teil 2: Handelt es sich

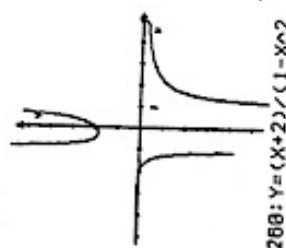
um eine Funktion mit
sin, cos, Tan oder cot?

Teil 4: Handelt es sich

um eine Funktion mit
einer Wurzel?



```
260:Y=1/2*X+SIN (R
/2)*X*(X-2)/(X
-2)
```



```
260:Y=(X+2)/(1-X^2)
```

EDGAR MAIR
 Sonnenhalde 16
 7904 Erbach-Bach
 Tel. 07305-5555

- 38 -

Herrn
 B. Fischeh
 Beratungs- u. Programmierdienst
 Kaiser-Friedrich-Strasse 54a
 1000 B e r l i n 12

22. März 1983

X Baustatik-Programmliste für PC-1500
 Ihr Schreiben vom 18.3.83

Sehr geehrter Herr Fischeh,

vielen Dank für Ihr Interesse an meinen Programmbeschreibungen. Vorab sende ich Ihnen eine Liste aller in meinem Hause erhältlichen Programme. In meinem Hause werden auch Statik-Programme für die PC 12-- Serie angeboten. Weiterhin sind in Kürze auch Programme dieser Art für den MZ-80 A zu haben.

Die angegebenen Preise verstehen sich inkl. MWST I
 Bitte weisen Sie eventuelle Interessen darauf hin, daß Sie mit der Bestellung eines Programmes auch Service und Betreuung sowie eventuelle individuelle Programmerstellung bekommen.

Für eine Kurzbeschreibung genügt es, die Programmübersicht und eventuell das Berechnungsbeispiel des Durchlaufträgers zu erwähnen. Der Durchlaufträger ist das gefragteste Programm.

Mit dem Beispiel "Balken auf elastischer Bettung" können Sie ganz skeptische Interessenten
 a) von der angebotenen Hardware und
 b) von der angebotenen Software überzeugen

sowie selbst Softwareersteller dieser Branche von diesem "Winzling" überzeugen, da soweit mir bekannt, es nur 2 Softwarehäuser in der BRD gibt, die dieses komplexe Programm mit all seinen Möglichkeiten annähernd erreichen, Hardware oder Software kosten dabei ein Vielfaches.

zu a) Beachten Sie den Ausdruck des CE 150 bei Vertikalschrift. Auf Anfrage bei der SHARP-Vertretung Stuttgart teilte man mir mit, daß eine Reklamation bezüglich dieser unsaubereren Formatierung noch nie vorgekommen sei.

Ich würde mich freuen, gleichgültig ob Sie mein Softwareangebot in Ihr Marketing miteinbeziehen, telefonisch oder schriftlich wieder von Ihnen zu hören.

Mit freundlichen Grüßen

- 37 -

Alle Cursor-Befehle bestimmt werden. Die WAIT-Anweisungen sind gleichfalls uneingeschränkt anwendbar.

Das Programm selbst wird wie folgt erläutert:

SR E0B1 setzt die aktuelle Cursor-Position eine Spalte weiter im Pointer &7B75
 SR EDEF gibt den im Accu befindlichen GPRINT entsprechenden Wert im Display aus
 SR EE4B errechnet im Register Y die Adresse der ersten Spalte der Zeichentabelle im ROM. Der ASCII-Wert des Zeichens muss im Accu stehen.

&12D7 Zwischenraum zum vorherigen Zeiche verdoppeln
 &12D0 Register X als Pointer auf aktuelles Ausgabe-Zeichen anlegen
 &12E9 Vorbereitung auf SR EE4B
 &12EA Zeilenende ?
 &12F1 Schleife für Zeichenausgabe anlegen
 &12F6 Wert für SR EDEF invertieren
 &1304 Schlusseroutine beginnt
 &130B Beginn-, Zwischenraum- und Schlusseroutine
 UP &130D Zeichenausgabe in der Schleife &12F1 bis &12FB

Wer mit dem Programm weiter spielen möchte, kann dessen graphische Möglichkeit prüfen durch Ändern des Wertes "&2F" in &12F7 und der Zeichen nach REM. Wer "rumpoked" oder "rus- next", kann auch die nicht über die Tastatur erreichbaren ASCII-Zeichen verwenden.

In der nächsten Folge wird ein Programm vorgestellt, mit dem Breitschrift, Sperrschrift und invertierte Schrift ausgegeben werden kann und alle Schriftarten gemischt möglich sind.

geschrieben mit TEXT WORKER ALPHA



- 39 -

- Kino -

Es ist so weit, sparen Sie Ihr Geld für die Kinokarte. Das für PC-1500 in der Lage ist, das Kino voll und ganz zu setzen, zeigt das folgende Programm. Also, Programm eingeben, die "n" drücken und die Vorkellung beginnt.

```

10: "a" CLEAR : DIM
   Z$(5)*12, Y$(5)
   *28, X$(20)*18:
   Z$(0) = "0000207
   F0000": Z$(1) = "
   0000663F 1000
   20: Z$(2) = "0044221
   F3400": Z$(3) = "
   0024221F 1224":
   Z$(4) = "0014221
   F2440
   30: Z$(5) = "0000163
   F6400": Y$(0) = "
   2FA0000000007F
   000E082004000
   40: Y$(1) = "7F00000
   000007F000E003
   0600000": M$ = "0
   123456789ABCDE
   F": PAUSE "Gute
   n Tag
50: INPUT "Dein Vo
   rname, bitte",
   A$: CURSOR 19:
   WAIT 0: PRINT "
   moment
   99: WAIT 0
100: FOR B=1 TO LEN
   A$: FOR D=(B-1)
   *610 D+S: GOSUB
   500: NEXT D:
   NEXT B: CLS

```

```

105: J=JSS-LEN A$*6
   -30: FOR D=0 TO
   J: CURSOR D:
   GPRINT 127:
   NEXT D
110: FOR B=1 TO LEN
   A$: Y$(0) = "7F"+
   X$(B)+RIGHT$(
   Y$(0), 16)
120: Y$(1) = "7F"+X$(
   B)+RIGHT$( Y$(
   1), 16)
130: FOR C=155 TO (B
   -1)*6+JSTEP -1
140: I=0: IF C=INT(
   C/2)*2LET I=1
150: CURSOR C:
   GPRINT Y$(I):
   NEXT C: H=C+14
160: CURSOR C+7:
   GPRINT "00061F
   660000
170: FOR C=0 TO 6:
   CURSOR H-13:
   GPRINT POINT (
   H-13)-2^C*(
   POINT (H-13))-2
   ^C>=0)
180: CURSOR H-7:
   GPRINT POINT (
   H-7)-2^C*(
   POINT (H-7))-2^
   C>=0): NEXT C
190: FOR C=H-7 TO 15
   S: G=C-INT (C/6
   )*6: CURSOR C:
   GPRINT Z$(G):
   NEXT C: NEXT B:
   WAIT : PRINT :
   END
500: E=POINT D: F=E-
   INT (E/16)*16:
   E=INT (E/16)
510: X$(B)=X$(B)+
   MID$(M$, E+1, 1
   )+MID$(M$, F+1
   , 1): RETURN

```

Erich Abels
Accumer Str. 2A
Grafschaft
2948 Schorfens 3

An

Firma Fischer GmbH

Kaiser Friedrich Str. 54 a

1000 Berlin 12

Grafschaft, den 8.11.83

Betr.: Grafikprogramm aus PC 1500 Zeitung Heft 7

Sehr geehrter Herr Fischer!

Hiermit möchte ich Ihnen eine verbesserte Version des Grafik-
programms übersenden. Das Programm tut nun auf der %
Taste eine Kreisfunktion bekommen. Der Kreis kann den
Radius von 0 - 32 mm annehmen und er kann durch Winkel-
angaben begrenzt werden z.B. ein Halbkreis von 30° - 330°



Wenn man die Farbe wechseln will, drückt man die entsprechende
Taste auf dem Ziffernfeld.
Bitte im voraus herzlichen Dank

Mit freundlichen Grüßen
Erich Abels

-42-
*** Fragen ***

Vielleicht können Sie mir helfen: Wenn ich das CEASD an den CEASD/PCASD anschließe, dann kann ich das im Reise (VORWA) befindliche Programm weder liste noch aufrufen. NEUER FÜR! Dann zu STATUS & = so! Dieses Verhalten tritt nicht auf! Wenn ich das CEASD erst nach Einschalten der PCASD anschließe, haben Sie dafür eine Erklärung?

Carole W. S. G.

Abonnent
L 0761 1 87303
(Bismarck)
YORK HILGER
305-FRITZ-STR. 12

*** kl. Anfragen ***

PC-1500 Pgm. ab 1000
Grafik bei W. Blauha, Inhaber
Gortestr. 111/2, A-1100 Wien.

Mediziner *
ENG-Auswertung durch CASIO
FX-707P + FA-2 35 DM/CC
SHARP PC-1500 + 1100 + Ph.
78 DM/CC; EPSON HS-20 143
DM/JAC; Schick an SUNSOFT
Chiffre 00 1084.

Top Programs for
2DM in stamps from M. Kluecker,
Heisenweg 3, 8012 Dettenheim,
PC-1500

PC-1500 Pgm. ab 1000
Grafik bei W. Blauha, Inhaber
Gortestr. 111/2, A-1100 Wien.

PC-1500 BASIC MA Sp. 3 BASIC-
Prg. gegen 10 DM im Brief. De-
anscher 30 DM H. Sabert, Hol-
torfstr. Weg 5, 20633 Siala, Info
gegen RU!

SUPER-Hausmonitor ***
für PC-1500, M. Cook, Im Fabel-Plan,
DM 15 bei O. Lorkamp, Postfach,
4630 Bochum 7.

Wirdisch handliche und gut dokum-
mentierte PC-1500 Software:
1. Uhr-Wecker-Timer-Terminaler
2. Überwachung regeln, Zahlungen,
3. 50 DM Schmitt, Aussenbergstr.
28, 8000 München 81.

Hier gibt's Programme:

Ephemeriden mit dem Sharp PC-1500

Das Programm berechnet die Ephemeriden (Positionen) von Sonne, Planeten und periodischen Kometen. Es benötigt die 8-KByte-Erweiterung und den Printer/Plotter.

Seite 196 D-AP

Nr. 11 - November 1983

```

510:SORGN :
GLCURSOR (R,0)
: f-b
520:FOR I=100 Q
STEP 6:X=R*LOS
I=Y-R*5IN I:IF
I=ALINE -(X,Y)
: 9
530:LINE -(X,Y),f:
NEXT I:GOTO 30
600:CURSOR /:PRINT
"STRICH 0-3 ?"
:LEINT (OSC
INKEYS -18):IF
C>=8AND EX12
POKE $3BE0,E:
RETURN
610:GOTO 600
700:CLS:PRINT "0"
"PEEK $3BE0:"
<216 "":GPRINT
"/F00/":
CURSOR J3
710:INPUT "X-MERF:"
"J:CLS:INPUT
"Y-MERF:"M:
GOTO (Z-16)>X10
8
810:GOTO 5,"Z1810
1000/200/20010
700/2515111
820:GOTO 2/,"J3512
1000/200/210
800/200/210
800/200
830:GOTO 50,"Z511
100/200/210
01000/200/210
0010512100/25
111
835:DATA 78,"Z001
800/200381113
800/200100/25
45111
810:DATA 101,"Z10
600/200/2100/20
51100/2001851
21
850:DATA 125,"1851
2400/200100/20
14000/2003011
4400/210/2
510:*****
* GRAFIK *
*****
10:"A"RESTORE :
CLEAR:GRAPH :
WAIT 0:DIM A$(
5)*50,(0,5)
20:FOR J=0 TO 5:
READ A(J),G*(I
):NEXT J
30:CLS:FOR I=0 TO
5:CURSOR A(I)
:GPRINT A$(J):
NEXT I
40:Z=ASC INKEY$:
IF Z=1:CLS :
TEXT:END
45:IF <52AND Z=4
THEN COLOR (Z
-48)
50:IF >21AND <2
3:CLS:GOSUB (Z
-16)*100:GOTO
30
50:GOTO 40
100:R=JNE -(L,H):
GOTO 30
200:SORGN :
GLCURSOR (L,H)
:INPUT "RADIUS
":IN:CSIZE
:IN:CSIZE N:
INPUT "TEXT:"
:AS:LPRINT AS:
GLCURSOR (0,0)
:GOTO 30
300:SORGN :
GLCURSOR (L,H)
:INPUT "RADIUS
0-12mm:":R:
SREKUNG:":I
STEP 2:FOR I=0
TO 2PI
(RR-RR)*R:
RLINE -(Y,EX)
:NEXT I:NEXT E
:GOTO 30
400:SORGN :
GLCURSOR (L,H)
:GOTO 30
500:INPUT "RADIUS
0-20mm:":R,"H
INKEI 0-360:U
ON "IM,"BIS":
0:R=55:R=9:
SORGN :
GLCURSOR (L-R,
H)

```

Kombinatorik auf dem PC-1500

T.A. 0251 / 791609

Uwe Kleem, Königsves 19, 4400 Münster

Do not sell this PDF!!!

- U1: Iterative Spiegellinienberechnung nach Manning/Strickler bei belieb. Gerinneform (Querschnittspolygonzug)
- U2: Rohrnetzberechnung (H. Cross) - Eingabe bei Knotenkoordinaten, Stranglänge, 41.-
- U3: Ausgabebelastung, Ringdefinition, Fließrichtungen, 41.-
- U4: Ausgabe: Strangdurchflüsse, Druckhöhenverluste
- U5: Normalabflusstiefe in beliebigen Gerinne (z.B. Vorland) bei bekannter 55.-
- U6: Abflussmenge, Gerinneformeingabe als Polygonzug
- U7: Durchlauftrager (c.a. 1-15 Felder, 1-30 Lasten), Gleich-, Einzel-, Trapez-, 24.-
- U8: Teilgleichlast und Einzelmoment als Lastart! Ausgabe: Stützmomente, 24.-
- U9: Querkräfte (Q links/Rechts), Schnittgrößen (M/Q), Auf Wunsch zeichnet 24.-
- U10: der Plotter feldweise Momenten- und Querkraftlinie.
- U11: Querschnittswerte: berechnet Fläche, Trägheitsmomente, Xs, Ys, Flächenträgheitsmomente, Zentrifugalmoment, Schwerachsenrichtung und zeichnet 64.-
- U12: das Profil zum Abschluss!
- U13: Durchbiegung und Schnittgrößen von Rechteckplatten unter 32.-
- U14: Teilflächenebelastung (Navierlagerung)
- U15: Bewehrungsnachweis nach DIN 4112 und/oder DAST-Richtlinie 012 33.-
- U16: fuer unversteifte Platten.
- U17: Bemessung von Stahlbetonquerschnitten nach DIN 1045 39.-
- U18: Bemessung von allgemeinen ebenen Fachwerken (Speicherintensiv!) 30.-
- U19: Bemessung von Rechteckquerschnitten, deren Zugkraftaufnahme zu vernachlässigen ist (z.B. Mauerwerk, Fundament...)
- U20: Gradientenberechnung (RRL-LI 1973). Eingabe von <70 Tangentenscheitel- 19.-
- U21: Zuerst auf alle Kleinpunkte zu, der 1.- und letzten Station wie $x(x_0), y(y_0), S(x), H(x), T(x), f(x), AA(x), AE(x), \alpha, \beta$ -Station! 42.-
- U22: Massenmittlung nach Gauss/Elling mit Kontrolle der x/y-Eingabedaten und ubersichtliches Ausdrucken (Hauptpunkte, Kleinpunkte)
- U23: Klotoidenabsteckung (Hauptpunkte, Kleinpunkte) 25.-
- U24: Kreisbogenabsteckung (Kleinpunkte in 2-Varianten) 20.-
- U25: Geometrisches Nivellement mit Fehlerausgleichsrechnung 15.-
- U26: Polygonzuoberrechnung (Kleinpunkte in 2-Varianten) 20.-
- U27: ein-/beidseitiger Richtungsabschluss, berechnet alle gaengigen Varianten! 20.-
- U28: beiden Anschlussrichtungen (z.B. Verknuetungen), Ringpolygon-Fehler, 40.-
- U29: M1: Liste i Eingaben von Ausdrucken, sortieren, einordnen und finden 15.-
- U30: M2: Lineares Gleichungssystem 10.-
- U31: M3: Lineare Optimierung nach der Simplexmethode (<) 10.-
- U32: M4: Matrixoperationen: Addition, Subtraktion, Skalarprodukt, A*B 25.-
- U33: M5: Funktionsanalyse i Kurvendiskussion und Zeichnung. 35.-
- U34: ST1: Chi-Quadrat-Verteilungstest mit Klassenzusammenfassung 50.-
- U35: ST2: Lineare Mehrfachregression (multiple Linearregression) 15.-
- U36: ST3: (Student)-t-Verteilungstest 20.-
- U37: ST4: Mittelwert, Varianz, Standardabweichung 15.-
- U38: ST5: Permutationen, Kombinationen, Fakultaat 15.-
- U39: ST6: Empirische Verteilung (Klassenzuordnung-Haeufigkeit) 10.-
- U40: U1: Laboruhr, stoert nach Ueberge Arbeitszyklen mit Kommentar 10.-
- U41: U2: Kartei, erfasst, erweitert, sucht (Stichwort), loescht, druckt aus 10.-
- U42: U3: Morseausgabe, ueber BEEP+REM, uebersetzt eine Eingabezeile 20.-
- U43: U4: 3-D-Plot, zeichnet und dreht Koerper (Eckpunkteingabe!) 20.-
- U44: U5: Briefeditor fuer CE-158/RS-232C, schreibt und aendert! 15.-
- U45: U6: Koenig, versuchen Sie, ein Koenigreich zu regieren - intelligent! 20.-
- U46: E1: Konstantstromquelle, Stiebschaltung, unstabiles Netzteil 20.-
- U47: E2: Emitterschaltung, Kollektorschaltung eines Transistors 20.-
- U48: E3: IC 555 - Rechteckgeneratorschaltungsberechnung (Frequenzverh.) 15.-
- U49: E4: Dimensionierung einer mit Zenerdiode stabilisierten Spannung 10.-
- U50: E5: RC/LC-Hochpass/Tiefpass-Berechnung 10.-
- U51: E6: Lautsprecherweichenberechnung (2-/3-Wegeweiche) 20.-

Hinweis: alle Programme sind fuer den PC-1500, CE-150, CE-155 geschrieben. Hinweis! und getestet worden. Preis entspricht in etwa Programmbytes/100. Bei Fragen zum kompletten Speicherbedarf gebe ich gerne teleph. Auskunft

UWE KLEEM
KONIGSVEG 19
4400 MÜNSTER
TEL. 0251-791609

mit freundlichen Grüßen

ORGAN DES PC-1500 USER-CLUBS
SHARP-Vertragshändler Fiechel GmbH
Kaiser-Friedrich Str. 54a
D-1000 Berlin 12

Budapest, den 3.11.1983

Liebe Leute von Fiechel GmbH!

Besten Dank für die PC-1500 Probelet, die mir sehr gefällt. In der Zeitung habe ich über Ihre Plan gelesen. (Programmier und Programmhdbuch den PC-1500)

Ich bin Kfz-Ing, und ich beschäftige mich mit dem Verkehrsunfall-rekonstruktion. Ich habe viele Programme über Unfallanalysen, zB.: Oberholen, Motometer-Auswertung, Bremsprogramm Zusammenstosen, Kurvendynamik, Folgeproblemen, Gengbreite in Kurven, Kraftfahrzeugdynamik, Fußgängerunfällen, Weg-Zeit Diagrammen, usw. Die Länge der Programmen sehr verechidane (1.4-8 kB)

Wenn Sie glauben, könne ich für Sie die Programmen senden. Als Honorarium möchte ich (bitte zahlen Sie zu) ein Freiexp. von diesem PC-1500 Buche, und eine regelmäßige Sendung der PC-1500 Zeitungen noch.

(mit Hefte 1-6 können Sie mir auch eine grosse Freude bereitet.)

Ich würde mich freuen, schnell von Ihnen zu hören,

mit freundlichen Grüßen:

A. Melegh

Dr-Ing-MELEGH GÁBOR

/Ich bitte um Entschuldigung, dass ich nicht besser in deutsch schreiben kann. Ich lerne./

DR. MELEGH GÁBOR

H - 1111 Budapest

Kandhy Folyas Ut 22

Ungarn



SHARP CORPORATION
INDUSTRIAL INSTRUMENTS GROUP
YAMATOKOORYAMA, NARA 639-11, JAPAN

CABLE ADDRESS: LABOMETSANKU OSAKA
TELEX: (LABOMET J63428 J63429)SNK
HEAD OFFICE: ABEHO, OSAKA 649 JAPAN

YOUR REF. NO.
OUR REF. NO.

Dipl.-Kfm. Bernd Fischel
Kaiser-Friedrich-Str. 54a
D-1000 Berlin 12
WEST GERMANY

Dear Mr. Bernd Fischel

We thank you very much for your letter dated Aug.28th '83 and your patronizing our products.
Judging from your letter, you may be talking about SHARP Pocket Computer User's Club in Japan that has the contact address as follows:

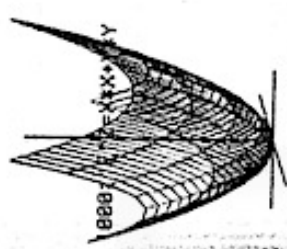
Pocket Computer Group
Personal Equipment Division
SHARP CORP.
492 HINOSHO-CHO YAMATOKOORYAMA, NARA 639-11, JAPAN

Should you have any further questions, please do not hesitate to contact us.
Sincerely yours

Akira Tanimoto
A. Tanimoto
Manager
Product Planning Dept.
Personal Equipment Div.

Leser des PC-1500 Club

*We are
PC-1500 Club Germany and
M2-80 USER-Group Bonn*



JJ Taschencomputer-Software
Jens Jürgens
Rohrteichstr. 66
D-4800 Bielefeld 1

SHARP
Personal, Micro-, Billing-
COMPUTER
Fischel Ber.- und Progr.-
Dienst St. 10-14: PC-1500 Club
Kaiser-Friedrich-Str. 54a
1000 Berlin 12, Tel. 3 22 60 29

Alles für **SHARP** Computer

- PC-1500 Programmier- und Programmhilfsmittel
- MZ-80K Soft- u. Hardware-Erweiterungen
- SHARP-Zeitung für MZ-80 A / B / K und 700 (Probeheft 6,- DM)
- PC-1500-Zeitung

Dipl.-Kfm. Bernd Fischel
Kaiser-Friedrich-Str. 54a
1000 Berlin 12
Telefon 323 60 29

Fischel hats!

IMPRESSUM

HERAUSGEBER: SHARP - COMPUTER
Fischel-GmbH
Kaiser-Friedrich-Str. 54a
1000 BERLIN 12
Tel.: 030 / 323 60 29

REDAKTION UND GESTALTUNG:
Der Redaktionsschub des
PC-1500-USER-CLUBS DEUTSCHLAND
ERSCHINUNGSWEISE: monatlich
BESTELLUNGSZ: über den Herausgeber (s.o.)



CE-150
>> PLOTTER-FANT <<