

ALLES FÜR

SHARP

COMPUTER

Nr. 2/86 · DM 6,-
4. Jahrgang · öS 50, sfr 6,-

Software
Paperware
Hardware



Auswechselbare RAM-Karten

Professioneller Pocket-Computer, vielseitig einsetzbar,
System-Erweiterung möglich.

SHARP

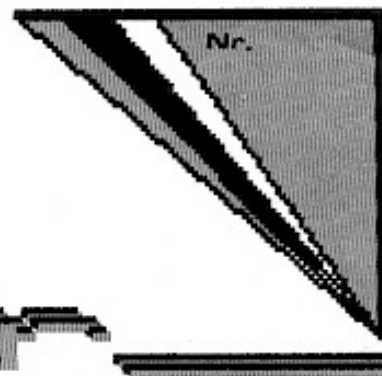


WWW.
PC-1500
.INFO

SHARP
Do not sale this PDF !!!
Durch Nachdenken vorn.

PC-1450

Alles für SHARP Computer



INHALTSVERZEICHNIS

Seite:

- 3 Pointersysteme der PC-12xx/14xx/1350
- 4 Stop-Uhr PC-1401/02
- 5 Super-Sound PC-1401
- 6 PC-1260/61 Feldvariablen, Zweizeichenspeicher, Dez.Hex-Wandler, Kleinanzeigen
- 7 PC-1350/1401 New, Retter
- 8 PC-2500 Hardcopy
- 9 PC-1421 Depotverwaltung
- 10 PC-1421 Begleitheft mit Programmbeispielen von P.Lawatsch
- 11 PC-1500 Zahlenspiegel, Aktienhandel, Tacho-Prüfer
- 12 PC-1401 Reverse-Spiel
- 13 Navigationsprogramme
- 14 Leserfragen
- 15 Schiffeversenken
- 17 PC-1500 Basis-Umwandlung
- 18 PC-1260/61 Hangmanspiel
- 19 17+4 Spiel
- 20 PC-1350 Label-Suchen
- 21-36 Software für die Entwicklung von Maschinenprogrammen
- 37 Verbessertes Hardcopy, Super-Statistik
- 38 Schachnotation
- 39-40 Tabellenprogramm mit CE-515P
- 41 Skat-Buchführung
- 42 PC-1500 Speichererweiterung, RAM-Kartenangebot
- 43 Das Textsystem Easywriter II
- 44 Buchtip
- 45-52 Großes Softwareangebot für MZ-K/A/B/700/800
- 53 Pool-Billard
- 54 MZ-700/800 Maschinensprache-Handbuch von Prof. U.Ehm
- 55 Programm zur Eingabe versch. Hexdumps, IMPRESSUM
- 56 Diatprogramm
- 57 Buchbesprechung: PC-1401/02 Maschinensprache-Handbuch von Dr.J.Stange
- 57 Anwendungshandbuch zu PC-1245/51/60/61 von I.Laue
- 58 Literatur, Bestellschein
- 59 PC-2500 Software-Angebot

Für unaufgeforderte eingesandte Beiträge übernimmt die Fischel GmbH keine Haftung und Gewährleistung. Die von der Fischel GmbH honorierte Beiträge gehen zur freien Verwendung im Eigentum der Fischel GmbH über; Eigentümer ist ebenfalls der oder die Urheber. Ausnahmen bedürfen der Schriftform; mündliche Abmachungen sind unwirksam.

Das Magazin für Soft-, Paper- u. Hardware

User-Club Deutschland

POINTER_SYSTEMS der PC-12xx/14xx/1350

Dies ist die erste Tabelle einer Serie, die die wichtigsten System-Adressen der PC's, die mit dem Mikroprozessor ESR-H ausgestattet sind, enthält.

Wir hoffen, daß wir nun eine leichtere Anpassung unserer Programme für die verschiedenen Geräte, die hier aufgeführt sind, erhalten.

Adresse & Definition	PC-1245/ 51/55	PC-1401/ 02	PC-1260/ 61	PC-1350
Start BASIC	C6E1,C6E2	46E1,46E2	66E1,66E2	6F01,6F02
Ende BASIC	C6E3,C6E4	46E3,46E4	66E3,66E4	6F03,6F04
Endadresse d. Feldvar.	C6FC,C6FD	46FC,46FD	66FC,66FD	6F07,6F08
Oper.v.WAIT	C6E5,C6E6	46E5,46E6	66E5,66E6	70B3,70B4
Start RSV-1	C6C4,C6C5	/	283E,283F	6F55,6F56
79 JP n m	C6DB	46DB	66DB	6F59
n	6CDC	46DC	66DC	6F5A
m	6CDD	46DD	66DD	6F5B
Flag WAIT n	C6D8	46D8	66D8	6F13
Flag P.=LP.	C6D7	46D7	66D7	6F12
Standard-(Z-A) variablen	C5D0-C69F	45D0-469F	6500-65CF	6C30-6D00
RSV	B800-B82F außer 1245	/	65D0-65FF	6E6F-6FFF

(Mit freundlicher Genehmigung :
Club du Sharpentiers ; 151-153 Av. Jean-Jaures
F-93307 Aubervilliers Cedex)

DURCH INFORMATION VORN 2
F I S C H E L G.M.B.H.
KAISER-FRIEDRICH-STR. 54A
1000 B E R L I N 12

```

1: "=====
2: "MIT FREUNDLICHER GENEHMIGUNG DES
3: "CLUB DU SHARPENTIER
4: "151-153. AV. JEAN-JAURES
5: "F-93307 AUBERVILLIERS CEDEX
6: "=====
7: "UEBERSETZT/UEBERARBEITET:
8: "PETER LAWATSCH
9: "=====

```

```

10: REM CHRONO 100 EME

```

```

20: REM (C) C MUNCH

```

```

30: REM & LE CLUB

```

```

100: Y=&4100,X=&80CF: FOR I=1 TO 11:

```

```

    POKE Y,PEEK (X+4),PEEK (X+3),PEEK (X+2),PEEK (X+1),PEEK X:X=X+5,Y=Y+5:
    NEXT I

```

```

4000: POKE &4000,&10,&40,&B5,&02,&00,&52,&11,&B8,&52,&11,&BE,&52,&11,&C1,&52,&11

```

```

4010: POKE &4010,&C7,&52,&11,&CA,&52,&E5,&A2,&78,&40,&B0,&E4,&8F,&67,&4B,&39,&1F

```

```

4020: POKE &4020,&67,&27,&38,&B2,&67,&15,&29,&0D,&10,&40,&D0,&02,&60,&52,&E4,&8F

```

```

4030: POKE &4030,&67,&03,&28,&09,&10,&40,&D0,&02,&00,&52,&2C,&03,&4E,&0C,&67,&10

```

```

4040: POKE &4040,&39,&27,&78,&40,&B0,&10,&40,&CA,&57,&74,&05,&52,&67,&32,&38,&05

```

```

4050: POKE &4050,&4E,&FA,&2D,&2B,&02,&00,&52,&11,&C7,&57,&74,&05,&52,&67,&32,&38

```

```

4060: POKE &4060,&05,&4E,&B1,&2D,&3C,&02,&00,&52,&11,&C1,&57,&74,&05,&52,&67,&32

```

```

4070: POKE &4070,&38,&05,&4E,&05,&2D,&4D,&02,&00,&52,&11,&BE,&57,&74,&05,&52,&67

```

```

4080: POKE &4080,&1E,&38,&05,&4E,&05,&2D,&4D,&5E,&02,&00,&52,&11,&B8,&57,&74,&05,&52

```

```

4090: POKE &4090,&67,&32,&38,&05,&4E,&00,&2D,&6F,&02,&00,&52,&11,&B5,&57,&74,&05

```

```

4100: POKE &40A0,&52,&67,&1E,&38,&05,&4E,&00,&2D,&80,&37,&00,&00,&00,&00,&00

```

```

4110: POKE &40B0,&00,&04,&85,&10,&41,&00,&18,&11,&00,&18,&11,&32,&18,&11,&00,&18

```

```

4120: POKE &40C0,&11,&00,&18,&11,&32,&18,&11,&00,&18,&11,&00,&18,&00,&27,&85,&10

```

```

4130: POKE &40D0,&60,&00,&19,&4E,&75,&37,&00,&00

```

PC-1401/02



-CHRONO- Stopp-Uhr mit 1/100 Sek.

Nachstehend ist eine Beschreibung in Maschinensprache aufgeführt betreffend der Umwandlung Ihres PC 1401/02 in einen Chronometer (Stoppuhr) mit einer Auflösung von 1/100 sek.

Nachdem Sie das Programm in Basic eingegeben haben, starten Sie es mit RUN (ENTER). Danach speichern Sie den Maschinensprache-Teil wie folgt auf Band ab:

CSAVEM "CHRONO"; &4000,&4136 (ENTER).

Sie können danach übrigens das Basic durch NEW (ENTER) löschen. Man startet dann den Chronometer mit CALL &4000 (ENTER).

Nachstehend die Tasten zur Bedienung des Chronometers

- starten
- stoppen
- Zwischenzeit
- Rückstellen auf 0
- zurück zu BASIC.

Um die Genauigkeit des Chronometers ggf. zu regeln, muß man die Werte des Parameters (WAITS) wie folgt ändern (mittels POKE)

- &4004 für Allgemeines
- &4051 für 1/100
- &4062 für 1/10
- &4073 für Sekunden
- &4084 für 10-Sekunden Abschnitte
- &4095 für Minuten
- &40A6 für 10-Minuten Abschnitte

Wenn man das letzte Ergebnis nach Umschalten in BASIC wieder haben möchte, tippen Sie CALL &4015 (ENTER). Erneutes Laden des Programmes mittels CLOADM "CHRONO"; Starten mit CALL &4000.

Üben Sie dieses Programm gut, denn es wird Ihnen sehr nützlich sein.

Viel Spaß!

SONDERANGEBOTE für Anwender

Sonderfunktionen

Hat man nach der Eingabe eines Tones Zweifel oder wurde etwas falsch eingegeben, kann der letzte Ton mit Eingabe "CORR" bei "TONHOEHE" gelöscht werden.

Das ganze Lied wird gespielt, wenn Sie bei "TONHOEHE" das Wort "PLAY" eingeben.

Einen Ausdruck aller Noten auf Papier erhalten Sie bei Eingabe von "PRINT".

Das ganze Lied wird durch "RESET" gelöscht.

PROGRAMMBESCHREIBUNG

Zelle	Beschreibung
1 - 3	Maschinenprogramm einlesen mittels POKE
10	Titel
20	Dimensionsierung der Variablen
21	Variablenbelegung (Tonhöhen/Tonlängen)
22 - 23	Datzeilen mit Werten
30	Notenzähler auf Null
40 - 130	Eingaberoutine
50	Frage nach Tonhöhe
51	Wenn PLAY dann ins Unterprogramm
52	Wenn CORR dann Zähler um 1 vermindern
54	Wenn PRINT dann ins Unterprogramm
55	Wenn RESET dann Zähler auf Null zurueck
60	Auswertung der Eingabe bei TONHOEHE
61	Falsche Eingabe, Zurueck zu Zeile 50
100 - 110	Tonlaenge
120 - 125	Ton ertönen lassen
130	Zähler um 1 vergrößern und zurueck zu Zeile 40
1000 - 1030	Abspielroutine
2000 - 2061	Printroutine
3000	Datzeilen mit Noten als Wörter

```

***** SUPER-SOUND *****
1:REM -----
2:POKE 14346,2:0:52:18
  :95:89:101:16:219:22
  :3:78:0:47:5:55
3:X=14346:Y=X+1:Z=X+11
10:PRINT "--SUPER SOUN
  D--"
20:DIM H(20),L(20),N(13
  0):DIM O(130)
21:FOR I=1 TO 20:READ H
  (I),L(I):NEXT I
22:DATA 184,7,172,11,16
  1,15,15,19,141,23,1
  32,25,123,29,115,33,
  107,37,99,41,91
23:DATA 45,85,49,80,53,
  75,57,70,61,64,65,59
  69,54,73,50,77,46,8
  1
30:I=0
40:"A" PRINT "NOTE:";I
50:INPUT "TONHOEHE:";T$
51:IF T$="PLAY" THEN
  GOTO 1000
52:IF T$="CORR" AND I>0
  THEN LET I=I-1:GOTO
  40
54:IF T$="PRINT" THEN 2
  000
55:IF T$="RESET" THEN
  LET I=0:GOTO 40
60:RESTORE 3000:V=1
61:READ W$:IF W$=T$
  THEN LET N(I)=V:GOTO
  100
62:V=V+1:IF V<21 THEN
  GOTO 61
90:IF N(I)=0 THEN BEEP
  1:PRINT "FALSCHE EIN
  GABE!";GOTO 50
100:INPUT "TONLAENGE:";T
  L

```

```

110:O(I)=TL*2+1
120:POKE Z,H(N(I)):POKE
  Y,L(N(I))
125:FOR R=1 TO O(I)-TM*2
  :CALL X:NEXT R
130:I=I+1:GOTO 40
1000:"S" FOR J=0 TO I-1
1010:POKE Z,H(N(J)):
  POKE Y,L(N(J))
1015:FOR R=1 TO O(J):
  CALL X:NEXT R
1020:NEXT J
1030:GOTO 40
2000:"Z" LPRINT "NOTE",
  "LAENGE"
2001:LPRINT "-----"
  "-----"
2005:RESTORE 3000
2010:FOR J=0 TO I-1
2020:FOR K=1 TO N(J):
  READ E$:NEXT K
2030:LPRINT E$,STR$(O
  (J)-1)/2)
2040:RESTORE 3000
2050:NEXT J
2061:LPRINT "-----"
  "-----"
  GOTO 40
3000:DATA "C1","C#1","D
  1","D#1","E1","F1",
  "F#1","G1","G#1",
  "A1","A#1","H1","C
  2"
3001:DATA "C#2","D2","D
  #2","E2","F2","F#2
  ","G2"
***** SUPER-SOUND *****

```

Thomas Jeger
Hauptstrasse 142
CH-3286 Muntelier/FR

Ich sende Ihnen schon wieder ein Programm. Diesmal handelt es sich um ein Programm zur Eingabe von Musik.
Dass der PC-1401 Töne von sich gibt, hatte ich nicht gewusst, bis ich ihr Anwendungsbuch durchlas. Dortin steht, wie man vorgehen muss. Es handelt sich um ein kleines Maschinenprogramm, das in Zeile 1 eingegeben ist. Dieses Maschinenprogramm habe ich aus dem Anwendungs- handbuch auf Seite 44/45. Das Copyright des Maschinenprogrammes sollte theoretisch bei Ihnen liegen. Der Rest des Programmes (siehe Beilage) ist ganz alleine von mir programmiert worden.

PC-1401

Zum Programm:

Es ermöglicht dem Anwender die Eingabe eines Liedes von bis zu 130 Tönen. Nach starten des Programmes mit RUN, wird man nach der Tonhöhe gefragt. Hier kann man folgende Werte eingeben (Tabelle 1):

Tabelle 1		
C1	entspricht DO	
C#1	entspricht DO#	
D1	entspricht RE	
D#1	entspricht RE#	
E1	entspricht MI	
F1	entspricht FA	
F#1	entspricht FA#	
G1	entspricht SO	
G#1	entspricht SO#	
A1	entspricht LA	
A#1	entspricht LA#	
H1	entspricht TI	
C2	entspricht DO	
C#2	entspricht DO#	
D2	entspricht RE	
D#2	entspricht RE#	
E2	entspricht MI	
F2	entspricht FA	
F#2	entspricht FA#	
G2	entspricht SO	

Für folgende Töne, muss man folgende Werte eingeben (Tabelle 2):

Tabelle 2		
Db	eingeben: Cm	
Eb	eingeben: Dm	
Fb	eingeben: Fm	
Gb	eingeben: Gm	
B	eingeben: Am	

Hat man die Eingabe erfolgreich abgeschlossen, wird man nach der Tondauer gefragt. Hier kann man individuell Zahlen zu. 0 und un- endlich eingeben. In Tabelle 3 sind Richtwerte!

Tabelle 3

Tondauer	Eingabe
16tel	1
8tel	2
4tel	4
2tel	8
Ganze	16
etc..	

Wurde auch dies erfolgreich absolviert, ertönt der Ton in der gegebenen Länge.

PC-1260/61 Liste der Feldvariablen und Zweizeichenspeicher

```

900:"PC-1261 ZUSAETZLI-
    CHE SPEICHER"
902:"I= PEEK 26364+256
    * PEEK 26365
904:Z$= CHR$ PEEK I:X=
    PEEK (I+1)
906:IF X=160 LET Z$=Z$+
    "$"
908:IF X=128 OR X=160
    LET Z$=Z$+"(*)":
    GOTO 914
910:Z$=Z$+ CHR$ (X-64*(X
    >90))
912:IF X>90 LET Z$=Z$+"$"
    "
914:PRINT STR$ I,Z$
916:I=I+4+256* PEEK (I+2
    )+ PEEK (I+3)
918:IF I=25856 PRINT
    STR$ I: END
920:GOTO 904

```

```

25533      F4$
25556      DE
25571      C$(*)
25770      B$(*)
25801      B$(*)
25856

```

Der PC-1260/61 gehört zu den Rechnern, bei denen schon ein neuer Datenspeicher entsteht, wenn man, manchmal versehentlich, zwei Buchstaben und dann ENTER tippt. Das kostet jedesmal 15 Bytes, und man hat bald keinen Überblick, welche zusätzlichen Speicher vorhanden sind. Hier ist ein Programm, das mit den Namen und Adressen der Speicher diesen Überblick schafft. Start mit DEF =. Die zusätzlichen Speicher aller Art sind im ROM unterhalb der Adresse 25856 untergebracht. Bei dem Beispiel entstanden die Speicher durch DIM B\$(0)*48, B(2), C\$(2,3), DE ENTER, F4\$ ENTER.

Der Abstand zur nächsten Adresse gibt den Bytebedarf im RAM, z.B. 25556-25533 = 23 Bytes für Textvariable und 25571-25556 = 15 Bytes für numerische Zweizeichenspeicher. Die Zweizeichen-Textvariablen nehmen stets 16 Zeichen auf, gegenüber 7 Zeichen bei den Standardvariablen A\$-Z\$.

Unerwünschte Speicher sind ohne CLEAR, was alle Speicher löscht, schwer loszuwerden. Man kann sie dagegen leicht umtaufen. Wenn man z.B. das ungeliebte DE (Adresse 25556) durch JA ersetzen will, so gibt man ein: POKE 25556, ASC"J", ASC"A" ENTER. Der Speicherinhalt bleibt dabei erhalten.

Einfacher Dez-Hex-Wandler

```

800:"DEZ>HEX"
802:"H" AREAD D:X=D:H$="
    "
804:Y= INT (X/16):Z=X-16
    *Y
806:H$= CHR$ (Z+48+(Z>9)
    *7)+H$
808:IF Y>0 LET X=Y: GOTO
    804
810:PRINT STR$ D,H$: END

```

```

168      A8
15719    3D67

```

Für die Umwandlung von dezimalen in Hexzahlen sieht man oft Routinen, die nicht so einfach sind, wie sie sein könnten. Hier ist ein besonders kurzes Programm für PC-Rechner. Es reicht für Zahlen bis 268435455 = Hex FFFFFFFF. Start mit Eingabe der Dezimalzahl und DEF H. Die Beispiele zeigen, wie dann die dezimale und die Hex-Zahl angezeigt werden. Als Unterprogramm beginnt die Routine mit 800:X=D:H\$="", Zeilen 804-808 unverändert, 810: RETURN. Die Dezimalzahl kommt vorher in Speicher D, die Hexzahl ist nach der Berechnung in H\$.

HERBERT GUDEHUS, Strandweg 53, 2 Hamburg 55

kleinanzeigen

Spiele und andere Software
für MZ-700. Eigenerst. billig!
Liste anfordern!
Christian Swoboda
Nahariyastr.20
1000 Berlin 49
Tel. (030)7461692

PC-1500: z.B. Speichererwei-
terung um 48K RAM (BASIC-RAM
+ 24K MCODE-RAM) DM 398.-
104k RAM DM 736.-
184k RAM DM1036.-
inkl.Einbau + 6 Mon. Garantie
Fa. Bajić, Wasserlooser Weg 34
2390 Flensburg
Tel.: 0461/33831 od. 26943

Do not sale this PDF !!!

Sehr geehrter Herr Fischel,

anliegend wieder zwei Beiträge vom Computer-Club Mölln für Ihre Zeitschrift.

Als Beispiel für das Programm HARDCOPY PC-2500 haben wir unsere Spiele Black Jack und 51-TOT ausgewählt. Diese Spiele laufen auf dem PC-2500 ohne Änderung. Ferner liest noch eine Hardcopy des Spiels Jack-Pot bei, weil hier das gesamte Display wiedergegeben wird.

NEW-Retter (OLD, RENEW)

PC 1350 - PC 1401

Programme, die als RENEW, OLD oder NEW-RETTTER bezeichnet werden, sind auch schon in dieser Zeitschrift erschienen. Ferner sind Programmbeispiele in verschiedenen Systemhandbüchern nachzulesen. Ist es also nötig, abermals auf dieses Thema einzugehen? Wir meinen, ja.

Alle uns bekannten Programme arbeiten in der Weise, dass zunächst das erste Byte im Programmspeicher mit 0 überschrieben wird, um die erste Zeilennummer wiederherzustellen. Dabei muss in Kauf genommen werden, dass Zeilennummern >255 sich in solche <255 ändern. Dies ist jedoch nicht zu vermeiden und wird selten vorkommen. Als nächster Schritt wird in den uns bekannten Programmen der Wert &FF als Programm-Endmarke gesucht und im Endpointer die entsprechende Adresse abgelest. Dieses Verfahren weist eine entscheidende Schwäche auf. Dazu muss man wissen, dass Zeilennummern im Programmspeicher zwei Bytes beanspruchen. Die Zeilennummer 10 wird z.B. als 00 0A abgespeichert, die Zeile 300 als 01 2C. Daraus folgt, dass der Wert &FF auch Bestandteil einer Zeilennummer sein kann, z.B. 00 FF (255), 05 FF (1535), 0A FF (2815), 0D FF (3583), u.s.w. Alle uns bekannten Programme beenden ihre Arbeit an solchen oder ähnlichen Zeilennummern. Wir haben daher ein Programm entwickelt, dass diesen Fehler nicht macht. Es ist für den PC 1350 gedacht. Des weiteren ist ein Hex-Dump für den PC 1401 beigelegt. Das Programm ist relocatibel und kann von Besitzern anderer PC's (ausser 1500) leicht umgeschrieben werden. Es müssen lediglich die Anfangs- und Endpointer eingesetzt werden. Das Programm kann nach versehentlichem NEW im Direktmodus mit POKE eingegeben werden.

```
Out 6C30 to 6C54
6C30 106F01 LD X,6F01
6C33 86 LD P,06
6C34 1A MLD (P),(X)
6C35 23 CLR A
6C36 26 LD (+D),A
6C37 1306 LD Q,06
6C39 84 LD P,04
6C3A 0A MLD (P),(Q)
6C3B 24 LD A,(+S)
6C3C 67FF CMP A,FF
6C3E 2904 JP NZ,6C3B
6C40 25 LD A,(-S)
6C41 670D CMP A,0D
6C43 3804 JP Z,6C48
6C45 24 LD A,(+S)
6C46 290C JP NZ,6C3B
6C48 25 LD A,(-S)
6C49 670D CMP A,0D
6C4B 24 LD A,(+S)
6C4C 24 LD A,(+S)
6C4D 3913 JP Z,6C3B
6C4F 106F03 LD X,6F03
6C52 84 LD P,04
6C53 1B MEX (P),(X)
6C54 37 RET
```

RUN MODE

```
Out 45D0 to 45F4
45D0 10 46 E1 86 1A 23
45D6 26 13 06 84 0A 24
45DC 67 FF 29 04 25 67
45E2 0D 38 04 24 29 0C
45E8 25 67 0D 24 24 39
45EE 13 10 46 E3 84 1B
45F4 37
```

Computer-Club Mölln
--- com ---

Horst Möller
Matth.-Cl.-Str. 73
2418 RATZBURG

Do not sale this PDF !!!

Hardcopy für PC-2500

vom Computer-Club Mölln

Inzwischen sind verschiedene Programme für Hardcopies vom Bildschirminhalt des PC 1500 veröffentlicht worden.

Diese lassen sich leicht für den PC-2500 abwandeln. Die vierzeilige Anzeige des Computers, die überdies grafikfähig ist, lässt sehr reizvolle Hardcopies entstehen. Das Programm kann leicht abgewandelt und ergänzt werden, um z. B. Teile der Graphik farblich abzusetzen. Der einzige Nachteil des Programms: Es dauert ein Weilchen, bis die Copy fertig ist.

Viel Spass beim experimentieren !!!

Listing:

```
1:"H" CLEAR
2:"h" LPRINT CHR$ 27+"b": FOR S=0 TO
  149:P= POINT (S,K): IF P LPRINT "M"
  ;S*3;",";I-K*3: LPRINT "P."
3:NEXT S:K=K+1: IF K<32 GOTO "h"
```

```

  D B K 8      8      44
  -----J-A-C-K-----P-O-T-----
  [M][M][M] Einsatz
  5 mal
  Total: 145 DM (1-9)1
```

```

  -----J-A-C-K-----P-O-T-----
  [M][M][M] Einsatz
  100
  5 Spiele *GAME OVER*
```

```

  Spieler 21 | BANK 16
```

```

  B K ?
```

```

  B
```

```

  8 10 B A
```

```

  7
```

40

DEPOHARTUNG

50 DRESS 01	
KAUFKURS=	123.50
WERT=	6206.38
VERK.KURS=	217.50
WERT=	18277.10
GEW/VERL=	3546.80
in % =	59.00

50 DRESS 3 WAR	
KAUFKURS=	107.20
WERT=	8581.73
VERK.KURS=	162.86
WERT=	8832.44
GEW/VERL=	-463.20
in % =	-5.51

50 HOECHST 70	
KAUFKURS=	140.80
WERT=	2136.84
VERK.KURS=	130.80
WERT=	6787.20
GEW/VERL=	-428.84
in % =	-6.00

100 BASF 02	
KAUFKURS=	37.80
WERT=	5824.16
VERK.KURS=	144.80
WERT=	14284.00
GEW/VERL=	4281.42
in % =	42.14

20 DI. SANK 77	
KAUFKURS=	482.80
WERT=	9721.14
VERK.KURS=	156.80
WERT=	8385.60
GEW/VERL=	-775.20
in % =	-7.83

20 SCHERING 03	
KAUFKURS=	362.80
WERT=	2330.74
VERK.KURS=	484.20
WERT=	2083.73
GEW/VERL=	644.83
in % =	8.78

35 DRESS 04	
KAUFKURS=	142.50
WERT=	5855.83
VERK.KURS=	217.50
WERT=	2588.73
GEW/VERL=	2452.50
in % =	48.51

40 KAUFLON 04	
KAUFKURS=	185.80
WERT=	4257.70
VERK.KURS=	135.80
WERT=	5261.61
GEW/VERL=	1183.51
in % =	25.52

15 HOECHST 70	
KAUFKURS=	455.80
WERT=	6319.12
VERK.KURS=	648.80
WERT=	5463.48
GEW/VERL=	2551.26
in % =	30.87

100 CONTI 04	
KAUFKURS=	71.80
WERT=	2150.85
VERK.KURS=	84.50
WERT=	8374.38
GEW/VERL=	1177.53
in % =	16.30

15 HOECHST 70	
KAUFKURS=	540.80
WERT=	8210.35
VERK.KURS=	648.80
WERT=	5463.48
GEW/VERL=	1259.85
in % =	15.33

50 BAYER 02	
KAUFKURS=	129.50
WERT=	6583.60
VERK.KURS=	147.50
WERT=	2827.81
GEW/VERL=	444.13
in % =	6.74

30 BASF 02	
KAUFKURS=	145.80
WERT=	4480.72
VERK.KURS=	144.80
WERT=	4268.68
GEW/VERL=	-149.84
in % =	-3.37

50 BASF 00 56	
KAUFKURS=	90.80
WERT=	4561.25
VERK.KURS=	134.58
WERT=	6633.21
GEW/VERL=	2871.46
in % =	45.48

50 BAYER 02 56	
KAUFKURS=	185.50
WERT=	5347.21
VERK.KURS=	142.50
WERT=	2827.81
GEW/VERL=	1688.60
in % =	31.42

FISCHER + ANWANDER

FISCHER + ANWANDER

Do not sale this PDF !!!

Sehr geehrter Herr Fischel!

Anbei ein Programm "DEPOTVERWALTUNG", welches Sie gerne veröffentlichen dürfen. Der Ausdruck kann sowohl auf dem Sharp-Plotter CE-150 als auch auf dem Brother EP 44 erfolgen; als DEMO nebeneinander sicher sehr gut geeignet. Es ist für 3 Depotteilnehmer konzipiert, eignet sich jedoch auch für einen CLUB mit mehreren Teilnehmern bei entsprechender Erweiterung. Die Gewinn-/Verlustberechnung erfolgt auf den Kaufwert bezogen, sie kann bei Bedarf natürlich auch jährlich berechnet werden, wenn Kauf- und Verkaufstag noch in das Programm eingearbeitet werden. Die tägliche Berichtigung kann leicht in den DATAS, zweiter Wert, vorgenommen werden. Die Unkosten bei Kauf und Verkauf sind bereits eingearbeitet - Gewinn und Verlust sind also Effektivwerte. Dieses Programm ist bereits seit einem Jahr auf den verschiedensten Pocketrechnern von Sharp - seit neuestem auf dem Sharp PC 1421 im Einsatz - und zwar erfolgreich, wie man sieht! Allen Lesern ebenfalls viel Erfolg mit "Depotverwaltung"

```

10 "DEPOTVERWALTUNG"
16 DIM W(16),Z(16)
20 LPRINT "DEPOTVERWALTUNG"
21 LPRINT ""
30 FOR I=1 TO 16
40 READ A$,B,C,D
45 LPRINT D;" ";A$
46 LPRINT "-----"
50 USING "#####.##":CSIZE 1
51 LPRINT "KAUFKURS=",B
52 LPRINT "WERT=",B*D+(.0135*B*D)+1
53 LPRINT "VERK.KURS=",C
54 LPRINT "WERT=",C*D-(.0135*C*D)-1
55 LPRINT "GEW/VERL=",((D*C)-(0.0135*D*C)-1)-((B*D)+(0.0135*B*D)+1)
56 E=(D*C)-(0.0135*D*C)-1
57 F=(D*B)+(0.0135*D*B)+1
58 G=(E-F)/F*100
59 LPRINT "in % = ",G
60 LPRINT "-----"
62 USING
100 W(1)=B*D+(.0135*B*D)+1
110 Z(1)=C*D-(.0135*C*D)-1
200 NEXT I
300 MG=W(1)+W(2)+W(3)+W(4)+W(5)+W(6)+W(7)+W(8)+W(9)+W(10)+W(11)+W(12)+W(13)
310 NG=Z(1)+Z(2)+Z(3)+Z(4)+Z(5)+Z(6)+Z(7)+Z(8)+Z(9)+Z(10)+Z(11)+Z(12)+Z(13)
315 LPRINT ""
320 LPRINT "KAUFWERT MARLIESE=";MG
330 LPRINT "VERKAUFWERT MARLIESE=";NG
335 LPRINT "GEWINN/VERLUST=";NG-MG
340 SG=W(14)+W(15)
350 TG=Z(14)+Z(15)
360 LPRINT ""
370 LPRINT "KAUFWERT SIGRID=";SG
380 LPRINT "VERKAUFSWERT SIGRID=";TG
385 LPRINT "GEWINN/VERLUST=";TG-SG
390 LPRINT ""
400 RG=W(16)
410 QG=Z(16)
420 LPRINT "KAUFWERT RALF=";RG
430 LPRINT "VERKAUFSWERT RALF=";QG
440 LPRINT "GEWINN/VERLUST=";QG-RG
500 END
1000 DATA "DRESD 84",133.9,217.5,50
1010 DATA "COMBK 3 WAR",55*3.05,63.0*2.5852,50
1020 DATA "HOECHST 79",140.8,136.0,50
1030 DATA "BASF 82",97.9,144.0,100
1040 DATA "DT.BANK 77",482,456.0,20
1045 DATA "SCHERING 83",362,404.7,20
1050 DATA "DRESD 84",142.5,217.5,35
1060 DATA "KAUFHOF 84",105.0,135.9,40
1070 DATA "HOECHST 75",455,640,15
1080 DATA "CONTI 84",71.0,84.9,100
1090 DATA "HOECHST 75",540.0,640.0,15
1100 DATA "BAYER 82",129.9,142.5,50
1105 DATA "BASF 82",145,144.0,30
1110 DATA "BASF 85 SG",90.0,134.5,50
1120 DATA "BAYER 82 SG",105.5,142.5,50
1130 DATA "PREUSSAG 84 RG",76.75,9.35

```

Dieter Greiner
Fritz-Schubert-Ring 10
6454 Bruchköbel

DEPOTVERWALTUNG

50 DRESD 84	50 COMBK 3 WAR
KAUFKURS= 133.90	KAUFKURS= 167.75
WERT= 6786.38	WERT= 8501.73
VERK.KURS= 217.50	VERK.KURS= 162.86
WERT= 10727.18	WERT= 8032.44
GEW/VERL= 3940.80	GEW/VERL= -469.28
in % = 58.06	in % = -5.51

KAUFWERT MARLIESE= 92089.475
VERKAUFWERT MARLIESE= 108123.0656
GEWINN/VERLUST= 16033.5906

KAUFWERT SIGRID= 9908.9625
VERKAUFSWERT SIGRID= 13661.025
GEWINN/VERLUST= 3752.0625

KAUFWERT RALF= 2696.91
VERKAUFSWERT RALF= 2619.63725
GEWINN/VERLUST= -77.27275

50 HOECHST 79

KAUFKURS= 140.80
WERT= 7136.04
VERK.KURS= 136.00
WERT= 6707.20
GEW/VERL= -428.84
in % = -6.00

100 BASF 82

KAUFKURS= 97.90
WERT= 9923.16
VERK.KURS= 144.00
WERT= 14204.60
GEW/VERL= 4281.43
in % = 43.14

20 DT.BANK 77

KAUFKURS= 482.00
WERT= 9771.14
VERK.KURS= 456.00
WERT= 8995.88
GEW/VERL= -775.26
in % = -7.93

20 SCHERING 83

KAUFKURS= 362.00
WERT= 7338.74
VERK.KURS= 404.70
WERT= 7983.73
GEW/VERL= 644.99
in % = 8.78

35 DRESD 84

KAUFKURS= 142.50
WERT= 5055.83
VERK.KURS= 217.50
WERT= 7508.73
GEW/VERL= 2452.90
in % = 48.51

40 KAUFHOF 84

KAUFKURS= 105.00
WERT= 4257.70
VERK.KURS= 135.90
WERT= 5361.61
GEW/VERL= 1103.91
in % = 25.92

15 HOECHST 75

KAUFKURS= 455.00
WERT= 6918.13
VERK.KURS= 640.00
WERT= 9469.40
GEW/VERL= 2551.26
in % = 36.87

100 CONTI 84

KAUFKURS= 71.00
WERT= 7196.85
VERK.KURS= 84.90
WERT= 8374.38
GEW/VERL= 1177.53
in % = 16.36

15 HOECHST 75

KAUFKURS= 540.00
WERT= 8210.35
VERK.KURS= 640.00
WERT= 9469.40
GEW/VERL= 1259.05
in % = 15.33

50 BAYER 82

KAUFKURS= 129.90
WERT= 6583.68
VERK.KURS= 142.50
WERT= 7027.81
GEW/VERL= 444.13
in % = 6.74

30 BASF 82

KAUFKURS= 145.00
WERT= 4409.72
VERK.KURS= 144.00
WERT= 4260.68
GEW/VERL= -149.04
in % = -3.37

50 BASF 85 SG

KAUFKURS= 90.00
WERT= 4561.75
VERK.KURS= 134.50
WERT= 6633.21
GEW/VERL= 2071.46
in % = 45.40

50 BAYER 82 SG

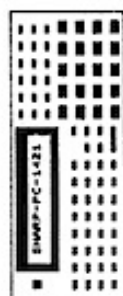
KAUFKURS= 105.50
WERT= 5347.21
VERK.KURS= 142.50
WERT= 7027.81
GEW/VERL= 1680.60
in % = 31.42

35 PREUSSAG 84 RG

KAUFKURS= 76.00
WERT= 2696.91
VERK.KURS= 75.90
WERT= 2619.63
GEW/VERL= -77.27
in % = -2.86

NEU & AKTUELL

Alles fuer
SHARP-Computer



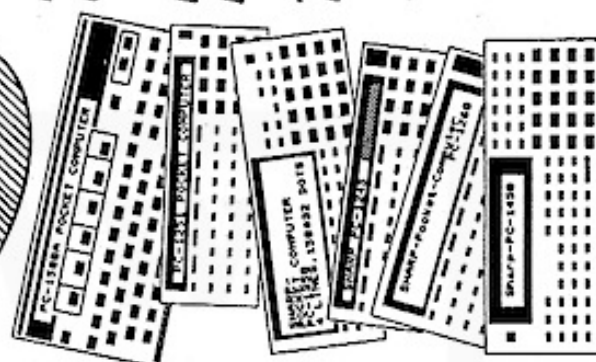
Programm
zur

teuer-Berechnung

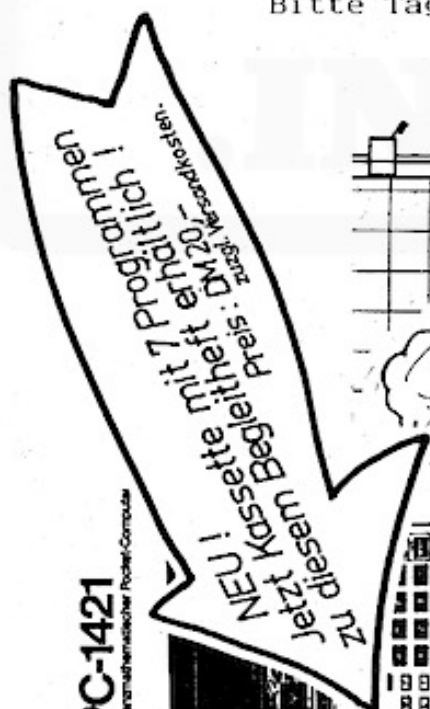
... für Lohnsteuerhilfsvereine, Steuerberater, Betriebs- u. Wirtschaftsprüfer und für alle Sharp-Pocket-Computer-Besitzer, die schon vorher wissen wollen, wieviel Steuer sie zahlen müssen.

Dieses in einem klaren und einfachen Basic geschriebene Programm ist auf jedem SHARP-Pocket-Computer mit mind. 3,5 kByte RAM lauffähig.

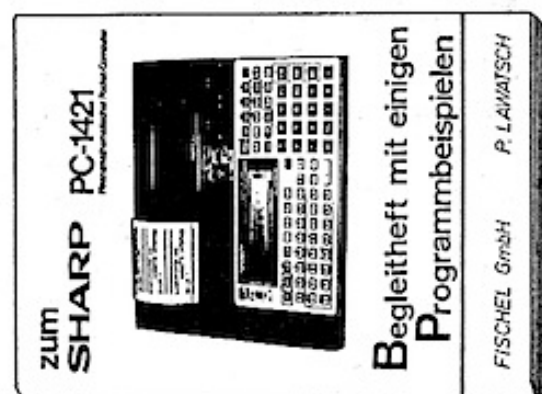
Es berücksichtigt sämtliche zur einwandfreien Berechnung erforderlichen Daten, so daß eine individuelle und unproblematische Berechnung der Steuer erfolgen kann.



Dieses Heft gibt es beim Kauf eines PC-1421;
Bitte Tagespreis anfragen.



zum
SHARP PC-1421
Personalrechner Pocket-Computer



PC-1500

ZAHLENSPIEGEL
1003193413 HAS



Da in meiner Rennsportzeitung ein Zahlenspiegel fehlt, habe ich mir einfach einen gemacht. Man kann aus dem Zahlenspiegel ersehen, wie oft eine Startnummer 1., 2. od. 3. geworden ist. Das Programm lässt sich mit kleinen Änderungen auch auf anderen Sharp-PCs gebrauchen.

BEDIENUNGSHINWEISE:

DEF A=Erreichten der DIN-Felder
DEF C=INPUT#, Variablen vom Band
DEF Z=Eingabe
DEF D=Ausdruck
DEF K=PRINT#, Variablen aufs Band

```
10: REM "ZAHLENSPIEGEL"
20: "A" CLEAR: DIM A(20,2), B(2):END
30: "C" CLEAR: DIM A(20,2), B(2):PRINT "Variablen einlesen!": INPUT A(0)
40: PRINT "Kann losgehen!"
50: "2" INPUT "Totales a. d. Pl. 1-2-3(J/N)": I$
60: IF I$="J" OR I$="N" GOTO 80
70: GOTO 50
80: IF I$="J" GOTO 320
90: IF I$="N" GOTO 150
100: FOR I=0 TO 2: WAIT 0: PRINT I+1: "ist Startnummer = ": INPUT U: WAIT
110: CLS: A(U,1)=A(U,1)+1: NEXT I
120: INPUT "Weitere Eingaben": L$: IF L$="J" GOTO "2"
130: IF L$="N" GOTO 150
140: GOTO 120
150: INPUT "Ausdruck (J/N)": L$: IF L$="J" GOTO "D"
160: IF L$="N" END
170: GOTO 150
180: END
190: LPRINT "2: COLOR 0"
200: LPRINT " * ZAHLENSPIEGEL * "
210: LPRINT " * 1 * 2 * 3 * "
220: LPRINT " * 1 * 9 * 2 * 6 * "
230: LPRINT " * 2 * 4 * 9 * 4 * "
240: LPRINT " * 3 * 9 * 9 * 7 * "
250: LPRINT " * 4 * 4 * 8 * "
260: FOR I=0 TO 19
270: AS=STR$(I+1): IF LEN AS=1 LET AS=" "+AS
280: FOR Y=0 TO 2: H$=STR$(A(I,Y)): IF LEN H$=1 LET H$=" "+H$
290: IF LEN H$=3 LET B$(Y)=H$: NEXT Y
300: IF LEN H$=2 LET B$(Y)=" "+H$: NEXT Y
310: IF LEN H$=1 LET B$(Y)=" "+H$: NEXT Y
320: IF LEN H$=0 LET B$(Y)=" "+H$: NEXT Y
330: LPRINT " * TAB 2: A$ * " : TAB 6: B$(0); " * " : B$(2); " * "
340: LPRINT " * "
350: NEXT I: IF 3: END
360: "K" PRINT "Variablen abspielen": PRINT A(0): PRINT "Fertig!": END
370: INPUT "Totales Rennen um Platz ?": K$: GOTO K$
380: "1" INPUT "ERSTER =": F(A(F,0)=A(F,0)+1)
390: INPUT "ERSTER =": F(A(F,0)=A(F,0)+1)
400: INPUT "ERSTER =": F(A(F,0)=A(F,0)+1)
410: INPUT "ERSTER =": F(A(F,0)=A(F,0)+1)
420: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
430: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
440: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
450: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
460: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
470: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
480: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
490: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
500: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
510: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
520: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
530: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
540: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
550: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
560: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
570: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
580: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
590: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
600: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
610: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
620: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
630: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
640: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
650: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
660: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
670: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
680: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
690: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
700: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
710: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
720: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
730: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
740: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
750: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
760: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
770: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
780: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
790: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
800: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
810: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
820: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
830: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
840: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
850: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
860: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
870: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
880: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
890: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
900: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
910: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
920: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
930: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
940: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
950: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
960: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
970: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
980: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
990: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
1000: INPUT "2: ERSTER =": F(A(F,0)=A(F,0)+1)
```

Do not sale this PDF !!!



SHARP®

PC-1500

WAHRE KOSTEN BEIM
HANDEL MIT AKTIEN

Beim Handel mit Aktien treten Nebenkosten auf, die wir mit 2 kleinen Programmen berechnen, und den wahren Ankaufrispreis sowie den wahren Erlös ausdrucken.

Die Nebenkosten gliedern sich in 0.25% Börsenumsatzsteuer, 0.1% Maklergebühr (min 0.50DM (Variable F in Zeile 100 und 240)), sowie 1% Provision (min. 5-8 DM (Variable E in Zeile 98 u. 230)). Die Gebühr fuer Fast-Preisauftrage ist in der Variablen D festgelegt. Die Gebühren fuer E u. D sollten Sie bei Ihrer Hausbank erfragen und einsetzen.

```
10: "A" CLEAR: INPUT "Wieviel Aktien ? " : JA
20: INPUT "Kurs ? " : B: C=JA*B
30: WAIT 0: PRINT "limitierter Auftrag (J/N)"
40: AS=INKEY$
50: IF AS="J" LET D=2.5: GOTO 80
60: IF AS="N" GOTO 30
70: D=0
80: WAIT : E=C/100: F=E/10: G=E/4
90: IF E<5 LET E=5
100: IF F<.5 LET F=.5
110: USING "#####.##"
120: PRINT "zu bezahlen=": C+E+F+G+D: "DM"
130: USING
```

```
140: END
150: "U" CLEAR: INPUT "Wieviel Aktien ? " : JA
160: INPUT "Kurs ? " : B: C=JA*B
170: WAIT 0: PRINT "limitierter Auftrag (J/N)"
180: AS=INKEY$
190: IF AS="J" LET D=2.5: GOTO 220
200: IF AS="N" GOTO 170
210: D=0
220: WAIT : E=C/100: F=E/10: G=E/4
230: IF E<5 LET E=5
240: IF F<.5 LET F=.5
250: USING "#####.##"
260: PRINT "Erlös=": C-E-F-G-D: "DM"
270: USING
280: END
```

x Copyright
x 1985 ***** 1985
x
x S H A R P
x
x Basic-Team
x

Der PC-1500 als TACHO-PRUEFER

```
9: "D" CLEAR: INPUT "MESS-STRECKE(m) = " : L
10: PRINT " "
11: INPUT " "
12: PRINT " "
13: INPUT " "
14: N1=INT (1/1000): T6=INT (1/100): N1*100
15: N2=INT (1/1000): T6=INT (1/100): N2*100
16: N3=INT (1/1000): T6=INT (1/100): N3*100
17: N4=INT (1/1000): T6=INT (1/100): N4*100
18: N5=INT (1/1000): T6=INT (1/100): N5*100
19: S1=(1/1000): M1=INT (1/100): M1*100
20: S1=(1/1000): M1=INT (1/100): M1*100
21: S1=(1/1000): M1=INT (1/100): M1*100
22: S1=(1/1000): M1=INT (1/100): M1*100
23: S1=(1/1000): M1=INT (1/100): M1*100
24: S1=(1/1000): M1=INT (1/100): M1*100
25: S1=(1/1000): M1=INT (1/100): M1*100
26: S1=(1/1000): M1=INT (1/100): M1*100
27: PRINT "Geschw. = " : J: "km/h": USING "#####.##"
```

Der PC-1500 hilft Ihnen, Ihren Tacho zu eichen. Das Programm ist so einfach, das jede weitere Erklärang unnuetz waere.

REVERSE/Spielbeschreibung.

Klaus Entling Hinter der Mühle 2
6442 Rotenburg 1

Die zufällige Anordnung der hexadezimalen Ziffernfolge

"123456789ABCDEF" wird vom Rechner ausgegeben, etwa so:

1978EACD354B6F2.

Durch geschickte Umstellung von Ziffernblöcken soll mit möglichst wenig Schritten die ursprüngliche Anordnung wiederhergestellt werden.

Wenn man nicht stumpfsinnig "drauflostippt", sondern versucht, mit möglichst wenigen Schritten die Aufgabe zu lösen, erfordert es eine Menge Überlegung!

REVERSE/Spielablauf.

1) Spielbeginn mit DEF "S" oder RUN

Es erscheint der Titel: "R-E-V-E-R-S-E". Nach ca. 2 Sek.

gibt der Rechner die verwürfelte Ziffernfolge aus, etwa:

13F9AC8D65B74E2

2) Die Umstellung der Ziffern geschieht so:

Beispiel: Angenommen, wir wollen in obiger Ziffernfolge "13F9..." die Ziffern "13" mit der "2" zusammenschließen, dann müssen wir die Folge so umordnen, daß die 14. Stelle (E) die erste, die 13. Stelle die zweite usw. wird.

Wir geben also ein:

```
21) ENTER 13F9AC8D65B74E2
22) <ENTER> E Ausgabe: E47B56D8CA9F312
    und weiter, etwa: 6 " " 65B74ED8CA9F312
    <ENTER> F " " F9AC8DE47B56312
    <ENTER> 8 " " 8CA9FDE47B56312
    <ENTER> A " " AC89FDE47B56312
```

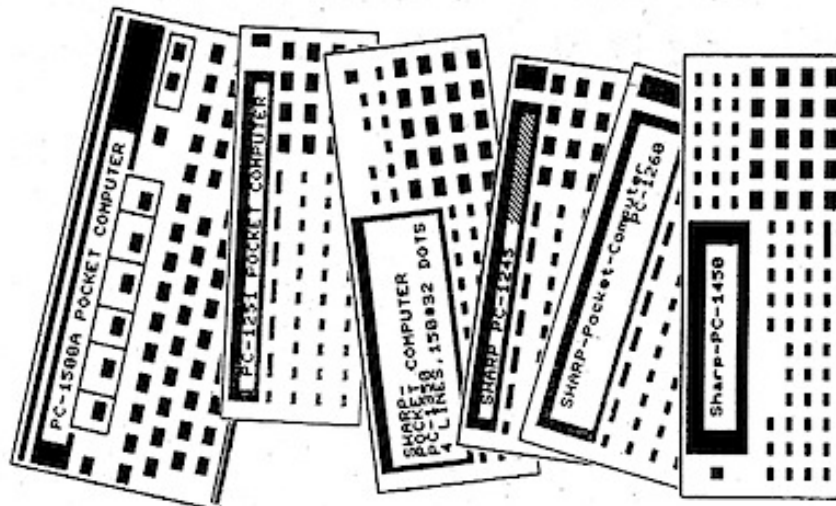
und so weiter!

Die endgültige Ziffernfolge "123456789ABCDEF" soll mit möglichst wenigen Schritten (16) erreicht werden. Es ist garnicht so einfach!

3) Ist die Lösung gefunden, gibt der Rechner das Ergebnis aus.
"SOLL 16 IST: xx" und "Kommentar"

REVERSE/Listing für PC 1401

```
8:"S"
10: CLEAR
20: DIM A$(15)*1, B$(2): VS=0
30: FOR I=1 TO 15
31: A$(I)=CHR$(48+I)
32: IF I>9 THEN LET A$(I)=CHR$(55+I)
33: NEXT I
34: WAIT 100: PRINT "R-E-V-E-R-S-E*": WAIT
35: COSUB 50
36: GOTO 100
40: WAIT 128: PRINT B$(1): PRINT "SOLL: 16
    IST: "; VS
42: IF VS<=16 THEN PRINT "SEHR GUT! PLUS"
    : 16-VS: END
44: PRINT "NA JA... MINUS": VS-16: WAIT 64:
    END
50: FOR I=1 TO 15
51: RANDOM
52: K=INT (RND 15)
53: IF A$(K)<"0" THEN LET B$(1)=B$(1)+A$
    $(K): A$(K)="0": GOTO 55
54: GOTO 52
55: NEXT I
60: RETURN
100: PRINT B$(1)
110: INPUT VS
112: IF ASC(VS)>48 AND ASC(VS)<58 OR ASC(
    VS)>64 AND ASC(VS)<71 THEN GOTO 115
113: PRINT "FEHLEINGABE": GOTO 110
115: VS=VS+1
120: FOR I=1 TO 15
130: IF MID$(B$(1), I, 1)=VS THEN LET V1=I
    : I=15
140: NEXT I
150: FOR I=V1 TO 1 STEP -1
160: B$(2)=B$(2)+MID$(B$(1), I, 1)
170: NEXT I
180: B$(2)=B$(2)+RIGHT$(B$(1), 15-V1)
190: LET B$(1)=B$(2): B$(2)=" "
200: IF B$(1)="123456789ABCDEF" THEN 40
210: GOTO 100
```



NEU & AKTUELL

Alles fuer
SHARP-Computer



Computer

IN DEN BEISPIELEN IST DAS PROGRAMM MIT FOLGENDER KOMPASS-KOMPENSATION BESCHICKT: $\mu\alpha \rightarrow 2$ / DEVIATIONSTABELLE >BR< SOWIE <DSU> DER TERMINUS >IN< BEDEUTET DAS ZENTRUM DES PERIPHERIEKREISES BEFINDET SICH ZWISCHEN ROOT U. OBJEKT.

**** ANWENDERPROGRAMM FUER POCKET-COMPUTER ****

DAS MENUE VON >NAVI-COMP 2< BIETET FOLGENDE PROGRAMME AN :

KURS (KOMPASS >> KARTE)
KURS (KARTE >> KOMPASS)
WINDBERECHNUNG
NEUER KURS (MIT DATEN d. WINDRECHNUNG)
KOPPELN
SEITENPEILUNG
HORIZONTALPEILUNG (AUCH SEKTANT)
PEILUNG DURCH VERSEGELUNG (ALLE ARTEN)
BESTECKVERSETZUNG
ABSTAND DURCH HOEHENWINKEL
ABSTAND OBJEKT UEBER KIMM

WINDRECHNUNG

EINWANDERUNG 12-26000/KM 270/1000 0

1913120

WIND 209 Grad

WIND 12.4 km 110 Grad 88 133 Grad

NEUER KURS

EINWANDERUNG 88/1000 0/100 5/1000 1/215/1011.14

KM 20 Grad/WIND 17 km/42 STB

KM 33 Grad : WIND 88 STB

BIS -18/100 3/10/100 5.14

SEKUNDEZEIT (WIND) 12/215/1011.14 1914019

HORIZONTALPEILUNG (KOMPASS)

EINWANDERUNG 133-100/45 Grad 88/1.23-80/2.08 km

RADIUS (km) 88 1.07 (km) / BC 4.08 (km)

PEILUNG DURCH VERSEGELUNG

EINWANDERUNG 133-100/45 Grad 88/1.23-80/2.08 km

PEILUNG 88/1.07 23/100 5 10/100 1/105

PEILSTRAHLE 8 137 Grad / 2.42 km

KM 110 Grad/WIND 2.08/1000 2.08 km

KM 100/100 500/100 5

AN DIESEN WENIGEN BEISPIELEN IST ZU ERSEHEN, WIE WICHTIG GUTES UND SCHNELLES NAVIGIEREN IST, UND DASS AUCH BESTE SEEMANNSCHAFT NICHTS BRINGT, WENN MAN DEN FALSCHEN KURS GEWAHLT HAT.

DIE ANLEITUNG FUER DIE BENUTZUNG VON >NAVI-COMP 2< BESTEHT AUS DEM TEIL A : OPERATION MIT PC 1500 (FUER LAIEN)

TEIL B : GENAUE ANLEITUNG FUER LISTING-ANLEITUNG

SPEICHERBEDARF FUER DAS PROGRAMM : 9150 BYTES

ENTFALLEN VORNAME, NAME-AUFLISTUNG UND DRUCKANWEISUNGEN, WIRD DER SPEICHERBEDARF EINWELTIG REDUZIERT.

PREIS (EINWANDERUNG) 149,- DM
KASSETTE FUER PC 1500 9,- DM
VERSANDSPERSEN 8,- DM
(SCHICK 001-100/1000)

alle Preise inkl. 14% MwSt



**** ANWENDERPROGRAMM FUER POCKET-COMPUTER ****

dmi NAVI-COMP 2

>NAVI-COMP 2< IST EIN NAVIGATIONSPROGRAMM, FUER UMFANGREICHE WIND- UND KURSBERECHNUNGEN UND DIE TERRESTRIISCHE NAVIGATION.

BASIS VON >NAVI-COMP 2< IST DAS PROGRAMM >INDOSAIL<, ERSTELLT FUER EINE VON MEINEM BUERO IM AUFTRAG DER HAMBURGISCHEN SCHIFFBAU-VERSUCHSANSTALT ENTWICKELTEN PRAEZISIONS WIND- U. FAHRTMESSANLAGE, FUER DAS VOM BUNDESMINISTER FUER FORSCHUNG U. ENTWICKLUNG FINANZIERTE GLEICHNARTIGE PROJEKT.

ALS BORO-ZENTRAL-RECHNER WURDE DER >SHARP PC 1500< WEGEN SEINES GUTEN PREIS- LEISTUNGSVERHAELTNISSES, SEINER GROSSEN SPEICHERKAPAZITAET (OPT.) UND SEINER GUTEN KOMMUNIKATIONSMOEGELICHKEITEN GEWAHLT.

GERAETEINTERFACE U. PROGRAMM HATTEN FOLGENDE VORGABEN:

1. EINFACHSTE BEDIENUNG
2. EINGANGSKOMPATIBEL
FUER HESSELNADEN VON 3000 U. 1840
3. AUSGANGSKOMPATIBEL
4. KEIN FESTSPEICHERPROGRAMM

DURCH EIN VOLLSTANDIGES NEUES KONZEPT WURDEN DIESE VORGABEN UEBERTROFFEN.

ES WURDE DAS MAXIMUM ERREICHT!

DER BETRIEB DER HESSELNADEN IST VOLLSTANDIG PROGRAMMIERT, AUSSER DEN KASSETTENADRESEN ALLE 24 STD.

IN MODIFIZIERTER VERSION IST DIESE ANLAGE SEIT EINIGER ZEIT UNTER DEM NAMEN >dmi NAVI-COMP< AUF DEM MARKT UND ERNUEGLICHT LEISTUNGS- UND LANGZEITSEGELN ERFOLGREICH ZU NAVIGIEREN UND DIE DATEN LANGER TOERNES AUFZUZEICHNEN, UM SIE AN LAND GRAPHISCH AUSZUKURTEN.

MEIN BUERO HAT NUN DAS PROGRAMM DIESER ANLAGE FUER EINE MANUELLE BESCHICKUNG UMGESCHRIEBEN UND STELLT ES HIER ALS >NAVI-COMP 2< VOR. BEI DIESEM PROGRAMM GALT ES HAUPTSAECHLICH, DEN FAKTOR >IRRUMK< AUF EIN MINIMUM ZU REDUZIEREN. WIE ERFOLGREICH DAS GEGLUCKT IST, ZEIGT TATSACHE, DASS ES NUR EINE FEHLERQUELLE GIBT: VERWECHSELN 88/STB :

REIHNENFOLGE DER BESCHICKUNG, RICHTIGES VORZEICHEN usw. UEBERNIMMT DER RECHNER.

IM UEBRIGEN LAUFEN ALLE PROGRAMME MENUE-GESTEUERT d.h., DER RECHNER FRAGT WAEREND DES PROGRAMMABLAUFS NACH DEN NOETIGEN EINGABEN

DURCH DIE DIREKTEN ERGEBNISSE WIRD DIE ARBEIT AM KARTENTISCH ZUM VERGEGEN, SO GEHOREN z.B. PARALLELVERSCHIEBUNG U. WINKELKONSTRUKTION DER VERGANGENHEIT AN.

SICHERHEIT DER ERGEBNISSE UND ZEITERSPARNIS LASSEN SICH LEICHT AN HAND DER ABGEDRUCKTEN PROGRAMMBEISPIELE UEBERPRUEFEN.

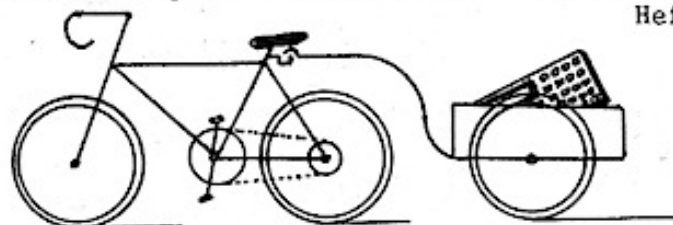
WER ALS SEGELER DIE NAVIGATION ERNST NIMMT, WIRD MIT DIESEM PROGRAMM MEHR ZEIT FUER DAS SEGELN HABEN.

WENN DIE ARBEIT AM KARTENTISCH BISHER ZU ZEITRAUBEND WAR, WIRD MIT

>NAVI-COMP 2< SPASS AM NAVIGIEREN FINDEN.

Erläuterungen zum Programm : " RADRENNEN "

Heft 10 , Seite 5



für alle PC's

Anfragen von :

Eckard Kriegesmann
Parkweg 92 a
5810 Witten

DEF V -Radabwicklung-

Mit diesem Unterprogramm wird der Weg berechnet, der bei einer Pedalumdringung zurückgelegt wird. Hierzu ist die vorhandene Übersetzung notwendig. Diese wird durch Eingabe der Anzahl der Zähne am "Kettenrad" und am "Ritzel" festgelegt.

DEF A -Durchschnittsberechnung (km/h)

DEF S -Zeitberechnung

Bei diesen Unterprogrammen ist a.u.die zurückgelegte (bzw. zurückzulegende) Entfernung (Distanz) von Bedeutung. Diese wird aus KM(ANKUNFT) - KM(ABFAHRT) = Entfernung (km) berechnet. Man kann nun den Tachometer-Stand am Start und am Ziel bzw. die auf der Tourenkarte eingetragenen km-Angaben eingeben.

PL

Wichtiger Hinweis zum Programm

" TEXTVERARBEITUNG "

Heft 11, Seite 32

PC-1401/02

In Zeile 1 dieses Programmes wird anstelle der Punkte ein zur Ansteuerung des Druckers CE-126P benötigtes Maschinen-(unter)programm gepoket (geschrieben). Die entsprechenden Codes (dezimal) dazu stehen in den Zeilen 110 und 111. Mit Hilfe dieses in Maschinensprache geschriebenen Programmes können Sonderzeichen mit dem CE-126P gedruckt werden.

Es ist zwingend erforderlich die Zeile 1 mit mindestens 24 Punkten genau einzutippen. (Auch darf davor nichts programmiert werden.)

Die Adresse &3805 beinhaltet den ersten in Zeile 1 enthaltenen Punkt. Man kann dies mit PEEK &3805 kontrollieren. Es muß 46 angezeigt werden. (46 = Code für ".")

Achtung! Beim PC-1402 kommt für &3805 &2005 infrage.

(Bei den erweiterten PC-1401 ist diese Adresse je nach RAM unterschiedlich.)

Sollte das Programm dann immer noch nicht laufen, so kann dies an den unterschiedlichen ROM-Versionen des PC-1401 liegen. Abhilfen wären:

Ändern in Zeile 110: 250 in 150
" " " 370: &A984 in &A920

Desweiteren werden u.a. diese Punkte in den nebenstehenden Büchern eingehend behandelt.

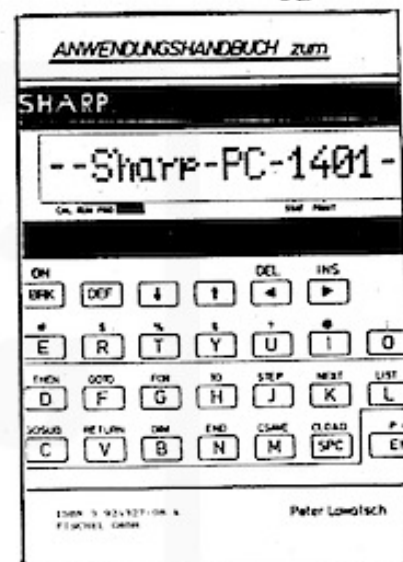
PL

Leseranfrage von
Dieter Klodt
Breiter Gang 5
4500 Osnabrück

PC-1401

...Mit großen Interesse habe ich in Ihrer neuesten Ausgabe von einem Vektor beim PC-1245 gelesen, der beim Ausschalten angesprungen werden kann. Mich interessiert nun sehr, ob ein solcher Vektor auch beim PC-1401 existiert und wo er ggf. liegt.

Dies ist leider das große Problem! -Der PC-1401 verhält sich genau wie der PC-1245. D.h. es existiert das selbe "Dilemma" wie in der Ausgabe 11/85 s.10 (oben) bereits erklärt. Etwas anderes ist uns nicht bekannt.



Für den
SHARP PC-1401
Taschencomputer
SYSTEMHANDBUCH



Fischel GmbH

Preis je Buch

39,-DM inkl. 7% MwSt.

Do not sale this PDF !!!

Spielregel:

In ein Spielfeld von 10 x 10 Quadraten zeichnen zwei Spieler, jeder für sich

- ein Schiff mit vier Kästchen
- zwei Schiffe mit je drei Kästchen
- drei Schiffe mit je zwei Kästchen

sowie, nur bei der Variante "M", vier Schiffe mit je einem Kästchen ein.

Die Schiffe können an beliebiger Stelle längs oder quer stehen, auch ganz am Rand, sie dürfen sich jedoch nicht berühren, auch nicht diagonal.

Jeder Spieler gibt anschließend abwechselnd einen "Schuß" ab, indem er das Quadrat nennt, in das er schießen will (z.B. D4). Nennt der Gegner "Wasser", ist dieser als nächster an der Reihe. Wird ein Schiff angeschossen, nennt der Gegner "Treffer"; sind alle Kästchen des Schiffes getroffen, nennt er stattdessen "Versenkt". Der Spieler, der getroffen hat, darf so lange weiterschließen, bis wieder "Wasser" gemeldet wird.

Wer als erster alle gegnerischen Schiffe versenkt hat, hat gewonnen.

Variante für den PC 1500:

Auf "RUN" oder "DEF/A" ertönt die Melodie "Frisch, Matrosen, wetzt die Messer, morgen stechen wir in See!"

Der Computer setzt seine Schiffe und fordert den Spieler auf, durch Einzeichnen auf den ausgedruckten Plan ("Ihr Spielfeld") das Gleiche zu tun. Danach jeweils "ENTER" drücken.

Der weitere Spielverlauf ergibt sich aus den Spielregeln und dem Display. Das angeschossene Schiff sendet "SOS", das versenkte verabschiedet sich mit dem "Marche funèbre" von Chopin.

Der Spielverlauf wird vom Computer in die beiden Spielfelder eingezeichnet. Der Gewinner wird mit "Hoch soll er leben!" gefeiert. Hat der Computer gewonnen, bekommt der Spieler zum Trost gesagt, wo die Schiffe stecken, nach denen er noch sucht.

Viel Spaß!

Bei Start mit "DEF/M" verläuft das Spiel ebenso, nur mit zusätzlichen vier "Einern".

Dr. Friedrich Lohmann
von-Ketteler-Str. 6
5060 Bergisch Gladbach 2

SCHIFFE VERSENKEN

IHR SPIELFELD

	W	W	W			W	10
W				U	U	W	9
	W	W	*	*		W	8
W	W	W					7
	U	*	*	W	W	*	6
	W	W	W	U			5
	W	*	*	U	W	W	4
W	W	W		W			3
		W	W	*	*	U	2
W		W					1
A	B	C	D	E	F	G	H

COMPUTER SPIELFELD

	W	W	W	*	U	W	10
W		U	W				9
	W	*	*	*	U	W	8
W		W	W				7
	W	U	W		W	W	6
W	W	*			W		5
	W	*	W	*	W	W	4
W		*		*			3
	W	W	W	*	W	*	2
W			W	W	W	U	1
A	B	C	D	E	F	G	H

Die grünen ***
hätten Sie noch
raten müssen!
Computerzüge: 46
Ihre Züge: 44



```

SCHIFFE VERSENKEN
10: "A" CLEAR : ME=0
: GOTO 25
20: "N" CLEAR : ME=1
: REM MIT EINE
RN
25: WAIT 0: PRINT "
***Schiffe ver
senken***: MU=
0: GOSUB 3000
30: WAIT 0: PRINT "
B. WARTEN, COMP
UTER SEITE!"
40: N=0+4: ME=N+0+4
: ME=0: ME=0: ME=0
: ME=0: ME=0: ME=0
50: FOR Z=4 TO 2: M
ESTEP -1
60: RANDOM : SS=RND
2: FOR I=2 TO 4
70: ZW=RND 2-1: 2X=
RND (10-22*(2U
=1)): 1: 2Y=RND
(10-22*(2U=0))
-1
80: FOR X=0 TO 0+2
M=1: 2Z2: FOR Y=
0 TO 0+(2U=0)*2
2: IF F(2X-X, 2Y
+Y) > 0: GOTO 20
90: NEXT Y: NEXT X
100: FOR X=1 TO 1+
2U=1: 2Z2: 1: 2Y=
FOR Y=1 TO 1+2
2U=0: 2Z2: 1: 2Y=
110: IF (2X+X) < 2
X+X) > 0: 2Y+Y < 0
+Y) > 0: GOTO 180
GOTO 180
120: F(2X+X, 2Y+Y)=1
130: ON 24: GOTO 160
140: IF (X=0) < Y=-1
+Y) > 2Z2: 20R X
+Y) > 2Z2: 180
150: GOTO 170
160: IF (Y=0) < X=-1
+X) > 2Z2: 20R Y
+Y) > 2Z2: 180
170: F(2X+X, 2Y+Y)=F
(2X+X, 2Y+Y)+1
180: NEXT Y
190: NEXT X
200: NEXT 1: NEXT 2Z
205: FOR X=0 TO 9:
FOR Y=0 TO 9: IF
F(X, Y)=1: LET F(
X, Y)=0
210: NEXT Y: NEXT X
215: CLS : CSIZE 2:
COLOR 3: LPRINT
"SCHIFFE VERSE
NKEN": LF 1
1045: IF YUK0GOTO
1060
1050: IF V(U, YU) > 0
: GOTO 1060
1055: GOTO 1065
1060: IF YU+1: P=10
20+YU+YU)*3
0: GOTO P
1065: M=YU: GOTO 11
60
1070: X=XO+1
1075: IF XO>GOTO
1090
1080: IF V(XO, M) > 0
: GOTO 1090
1085: GOTO 1095
1090: XO=XO-1: P=11
88-90*(XO)XU
0: GOTO P
1095: V=XO: GOTO 11
60
1100: YO=YO+1
1110: IF YO>GOTO
1140
1120: IF V(U, YO) > 0
: GOTO 1140
1130: GOTO 1150
1140: YO=YO-1: P=10
10+YO)YU)*3
0: GOTO P
1150: M=YO
1160: IF (ME=1) < (
CU+SS)*2=INT
((4+SS)/2)=
(U/2)=INT (U/
2)) > 0: LET SU
=SU+1
1170: GOTO 920
1200: COLOR 2: FOR
X=0 TO 9: FOR
Y=0 TO 9
1210: IF F(X, Y)=2
: GLCURSOR (S+
20*X, -225+20
*Y): LPRINT "
X"
1220: NEXT Y: NEXT
X
1230: GLCURSOR (0,
-220): TEXT :
LPRINT "Die
gruenen xxx"
: "hechten si
e noch":
LPRINT "rate
n messen!"
1240: LPRINT "Com
puterzeige:"
SU: LPRINT "
here Zeige:"
SF: LF 3: MU=4
0: GOSUB 3000
: END
2000: GLCURSOR (S+
20*X, 34+22*M
)
2010: LPRINT 0*
2020: RETURN
500: GOTO 410
510: IF (Q= "T")=0
: GOTO 940
520: V(U, M)=2: IF P=
0: LET P=1020
530: COLOR 3: Q= "X"
: MU=20: GOSUB 3
000: GOSUB 2000
: GLCURSOR (U, 0
): GOTO P
540: V(U, M)=3: COLOR
3: MU=30: GOSUB
3000: GOSUB 200
0
545: IF P=0: LET XU=U
: XO=U: YU=M: YO=
U
550: FOR X=XU-1 TO X
TO YO+1
555: IF (X=0) < (Y=0)
+X) > 0: 2Y=INT (Y/2)
: GOTO 920
560: IF V(X, Y) > 0
: GOTO 920
565: V(X, Y)=8: IF (M
E=1) < ((X+SS)/
2)=INT ((X+SS)/
2)) < (Y/2)=INT (
Y/2)) > 0: LET SU
=SU+1
570: NEXT Y: NEXT X
580: P=0: M=N-1: IF N
=20: GOTO 1200
590: GLCURSOR (U, 0)
: GOTO 250
1000: 2U=RND 4: P=9
80+2U*30: XU=
U: XO=U: YU=M:
YO=M: GOTO P
1010: XU=XU-1
1015: IF XU<0: GOTO
1030
1020: IF V(XU, U) > 0
: GOTO 1030
1025: GOTO 1035
1030: XU=XU+1: P=10
40+XO)XU)*3
0: GOTO P
1035: U=XU: GOTO 11
60
1040: YU=YU-1
2000
650: FOR J=(Y-P) TO
(Y-P+PP)
655: IF J=Y: GOTO 670
660: IF F(X, J)=2: LET
R= "Treffer":
E= "X": GOTO 68
0
670: NEXT J
680: F(X, Y)=1: IF E=
"X": LET M=M-1
: "U": LET M=M-1
690: LET MU=20: IF E
="U": LET MU=30
700: GOSUB 3000:
PAUSE R
710: IF E= "X": LET F
A=3
720: GLCURSOR (9+20
*X, -225-20*Y):
COLOR FA:
LPRINT E
725: IF M>0: GOTO 735
730: GLCURSOR (0, -2
0): TEXT :
COLOR 2: LPRINT
"Gewinner nach
: SF: LPRINT "Z
uegen bei": SU:
LPRINT "Comput
erzeigen!"
731: MU=40: GOSUB 30
00: LF 3: END
735: IF E= "X": GOTO
250
740: GOTO 410
750: RANDOM : SW=100
- (ME=0)*50
760: IF P>0: GOTO P
770: SX=0: SY=RND (5
U-SU)
780: FOR U=0 TO 9:
FOR M=0+ (ME=0)
: X(U+SS)/2 <
INT ((U+SS)/2)
: TO 3STEP 2: (M
E=1)
790: IF V(U, M) > 0
: GOTO 810
800: SX=SX+1: IF SX=
SY: GOTO 815
810: NEXT M: NEXT U
815: SU=SU+1
820: SU=SU+1: U=
CU+ (U-65)+
SR$ (M-1)
830: BEEP 2: WAIT 50
: PRINT "COMPU
T: SCHIESST AU
F": SU
840: WAIT 0: PRINT U
: "W(ass), T(r
e), U(n)?"
850: INPUT Q: CLS
860: IF (Q= "U") < 0
: "U": (Q= "U") < 0
: "U": (Q= "U") < 0
870: IF (Q= "U")=0
: GOTO 910
875: BEEP 1, 120, 250
: COLOR 1: GOSUB
2000
410: BEEP 1:
GLCURSOR (8+20
*X, -265): INPUT
"JHR SCHUSS <
: B.89)?": 122: S
F=SF+1
420: F=ASC (LEFT$ (
22$, 1))-65
430: Y=VAL (RIGHT$
(22$, 1))-1
440: IF LEN 22=3
: LET Y=9
450: IF (X>9) < (X<0)
+Y) > 0: 2Y=INT (Y/2)
: LET SF=SF-1:
GOTO 410
460: IF F(X, Y) < 2
: BEEP 1, 120, 250
: PAUSE "Wasser
": E= "U": FA=1
: GOTO 220
470: R= "Versenkt!"
: E= "U": FA=3
480: PP=8: PO=0: FOR
J=(Y-3) TO (Y+3
)
490: IF (J<0) < (J>9)
: =1: GOTO 535
500: IF J=Y: LET PO=P
: GOTO 535
510: IF F(X, J)=2: LET
PP=PP+1
520: IF (J=Y-1) < (F(
X, J)=0) < 2: LET P
=0
530: IF (J=Y+1) < (F(
X, J)=0) < 2: GOTO
540
535: NEXT J
540: IF PP>0: GOTO 65
0
550: FOR J=X-3 TO X+
3
560: IF (J<0) < (J>9)
: =1: GOTO 610
570: IF J=X: LET PO=P
: GOTO 610
580: IF F(X, Y) > 0: LET
PP=PP+1
590: IF (J=X-1) < (F(
J, Y)=0) < 2: LET P
=0
600: IF (J=X+1) < (F(
J, Y)=0) < 2: GOTO
610
610: NEXT J
615: IF PP=0: GOTO 69
0
620: FOR J=(X-PO) TO
(X+PO+PP)
625: IF J=X: GOTO 640
630: IF F(X, Y)=2: LET
R= "Treffer":
E= "X": GOTO 68
0
640: NEXT J: GOTO 68
0
650: FOR Y=0 TO 9
: GOTO 610
660: IF PP>0: GOTO 65
0
670: NEXT Y: NEXT X
680: BEEP 2: PAUSE "
Zeichnen Sie o
uf diesem...
390: WAIT 1: BEEP 2:
PRINT "Spiele!
an 1: Verer/CE
NTER"
395: BEEP 2: PRINT "
2 Dreier, 3 2u
eier/CENTER":
IF ME=0: GOTO 40
0
400: BEEP 2: PRINT "
X, Y)=0
410: NEXT Y: NEXT X
420: CLS : CSIZE 2:
COLOR 3: LPRINT
"SCHIFFE VERSE
NKEN": LF 1

```

PC-1500: Basis-Umwandlungen

Für die Umwandlung von Zahlen mit verschiedener Basis sind schon einige Programme veröffentlicht worden (z.B. Alles für Sharp Computer 6,48; 9,30). Das hier vorgestellte Programm erlaubt die gegenseitige Umwandlung von Binär-, Hexadezimal- und Dezimalzahlen, was bei der Beschäftigung mit Assembler oder beim Programmieren ganz nützlich sein kann. Die Ausgabe der Zahlen ist auch entsprechend formatiert: 4 Stellen bei Hex-Zahlen, 8 Stellen bei Binärzahlen, wenn die Umwandlung Hex zu Binär vorgenommen wird; bei dieser Umwandlung soll die Eingabe höchstens 2 Stellen haben. Das Menue erscheint im Display in abgekürzter Form. Die Wahl erfolgt mit den jeweiligen Reservetasten. Die arithmetischen Prozeduren benutzen z.T. die ASCII-Werte der Zeichen und logische Funktionen. Das Ergebnis wird im Display angezeigt. (F. Hintze)

```

18:REM "BASIS-UM
WANDLUNG"
28:REM Umwandlung
en von Bin-,
Dex- und Hex-
Zahlen
38:REM Menuwahl
ueber die Re
servetasten
48:REM Eingabe:
Hex=Zahl hoch
stens 2 Stel
len bei HEX-->
BIN...
58:REM ...hochst
ens 4 Stellen
bei HEX-->DEZ
68:REM Ausgabe:
Hex=Zahl immer
4 Stellen, Bin
=Zahl 8 Stel
len bei HEX-->
BIN
78:REM -----
88:REM (F.HINTZE
Aus. 83F)
98:REM "*****
*****"
108:DJM BM(15):FOR
I=8TO 15:BM(1)
=2^I:NEXT I
118:WAIT 128:PRINT
"MENU..."
128:WAIT 188:PRINT
"Wahl= Res.Tas
te druecken!"
138:"MENU"WAIT 8:
PRINT " B>D B>
H D>B D>H H>B
H>D"
148:AS=INKEY$:A=
ASC A$:IF A<17
OR A>22THEN 58
158:A=A-16:ON A
GOTO "B>D", "B>
H", "D>B", "D>H"
, "H>B", "H>D"
168:REM -----
178:REM "BIN>DEZ"
188:REM -----
198:"B>D"=8:WAIT
8:INPUT "BIN, 2
AHL=";B$:PRINT
"B=";B$
208:L=LEN B$
218:FOR J=1TO L
228:U=VAL MID$(B$
, J, 1)
238:D=D+U*2^(L-1)
248:NEXT J
258:WAIT :GCursor
118:PRINT "D="
:STR$ D
268:GOTO "MENU"
278:REM -----
288:REM "BIN>HEX"
298:REM -----
308:"B>H"=8:WAIT
8:INPUT "BIN, 2
AHL=";B$:PRINT
"B=";B$
318:"J"=LEN B$<1
6LET B$="B"+B$
:GOTO "J"
328:FOR K=1TO 13
STEP 4
338:H=8:IF=MOD$(B
, K, 4)
348:FOR J=1TO 4
358:H=H+VAL MID$(
, J, 1)*2^(4-J
)
368:NEXT J
378:Y$=CHR$(H+48+
7*(H/9))
388:H$=H$+Y$
398:NEXT K
408:WAIT :GCursor
118:PRINT "H="
:H$
418:GOTO "MENU"
428:REM -----
438:REM "DEZ>BIN"
448:REM -----
458:"D>B"=8:WAIT
8:INPUT "
DEZ, ZAHL=";D:
PRINT "D=";
STR$ D
468:"2"=D/2
478:X$=STR$(D-
INT D)*2)
488:B$=X$+B$
498:D=INT D
508:IF D<8THEN "2"
"
518:WAIT :GCursor
58:PRINT "B=";
B$
528:GOTO "MENU"
538:REM -----
548:REM "DEZ>HEX"
558:REM -----
568:"D>H"=8:WAIT
8:INPUT "
DEZ, ZAHL=";D:
PRINT "D=";
STR$ D
578:GOSUB "3":GOTO
"5"
588:"3"=INT (D/25
6):GOSUB "4"
598:B=D-8*256
608:"4"=L=(BAND 15)
+48:L=L+7*(L>5
7)
618:H=(BAND 248)/1
6+48:H=H+7*(H
57)
628:H$=H$+CHR$(H+
CHR$ L
638:RETURN
648:"S"=WAIT :
GCursor 188:
PRINT "H=";H$
658:GOTO "MENU"
668:REM -----
678:REM "HEX>BIN"
688:REM -----
698:"H>B"=8:WAIT
8:INPUT "
HEX, ZAHL=";H$
708:"6"=LEN H$:IF
L<2LET H$="B"+
H$:GOTO "6"
718:IF L>2THEN "H>
B"
728:PRINT "H=";H$
738:FOR K=1TO 2
748:IF=MOD$(H$, K,
1)
758:C=(ASC I$)-48-
7*(I$="9")
768:FOR J=1TO 4
778:B$=B$+STR$ 56N
(CAND BM(1-J))
788:NEXT J
798:NEXT K
808:WAIT :GCursor
48:PRINT "B=";
B$
818:GOTO "MENU"
828:REM -----
838:REM "HEX>DEZ"
848:REM -----
858:"H>D"=8:WAIT
8:INPUT "HEX, 2
AHL=";H$
868:"7"=LEN H$:IF
L<4LET H$="B"+
H$:GOTO "7"
878:IF L>4THEN "H>
D"
888:PRINT "H=";H$
898:FOR K=1TO 4
908:IF=MOD$(H$, K,
1)
918:N=(ASC I$)-48-
7*(I$="9")
928:D=D+N*16^(4-K)
938:NEXT K
948:WAIT :GCursor
58:PRINT "D=";
STR$ D
958:GOTO "MENU"

```

BYTES

2883

+
B
u
n
d
l
i
n
g
e
n
t
e
n
n
e

PC-1350

LABEL SUCHEN

Das nebenstehende Programm hilft, eine Fehlermeldung ERROR 4 zu verhindern.

Die folgenden BASIC-Zeilen machen deutlich, welche Vorteile sich der Anwender bieten:

```
500:AS= INKEY$: IF AS=" "
  * GOTO 500
510:CALL 6032: IF AS=" "
  GOTO 500
520:GOTO AS
```

Genauso ist eine Variante mit RESTORE AS denkbar.

Das Programm liest mit den hier verwendeten Adressen am Anfang des BASIC-Bereiches der PC-1350-Grundversion. Bei Verwendung einer RAM-Card ist dieser Bereich meistens frei. In der Grundversion kann durch Hochsetzen der BASIC-Pointer ein Teil für Maschinensprache reserviert werden.

```
6032 10 6C F9 LD X,N
6035 90      LD P,10
6036 00 07   LD L,n
6038 10      LLD(P),(X)
6039 F4 C7   JSR 14C7
603B 04      LD P,04
603C 1A      MLD(P),(X)
603D 24      LD A,(+S)
603E 67 FF   CMP A,n
6040 28 07   JPNZ+r6040
6042 23      CLR A
6043 10 6C F9 LD X,N
6046 52      LD (X),A
6047 37      RET
6048 04      INC S
6049 24      LD A,(+S)
604A 08      LD P,08
604B 0B      EX A,(P)
604C 24      LD A,(+S)
604D 67 22   CMP A,n
604F 28 13   JPNZ+r6063
6051 90      LD P,10
6052 49      DEC C0
6053 24      LD A,(+S)
6054 C7      CMP (P),A
6055 28 04   JPNZ+r605A
6057 50      INC P
6058 2D 07   JP -r6052
605A 63 00   CMP (P),n
605C 28 06   JPNZ+r6063
605E 67 22   CMP A,n
6060 28 02   JPNZ+r6063
6062 37      RET
6063 C0      DEC C0
6064 23      CLR A
6065 DA      EX A,E
6066 08      LD P,08
6067 0B      EX A,(P)
6068 04      LD P,04
6069 14      ADD (P),AE
606A 2D 2E   JP -r605D
```

AS= zu suchendes Label
wird nach \$10 bis \$16
ins interne RAM kopiert

= LD X,r6F01

BASIC-Start wird ins
S-Register geladen

BASIC-Ende erreicht ?
nein

Ja, Label nicht gefunden
\$02=00, AS=---

Zeilenlücke in den Accu
und nach \$08 (C0)

nächstes Zeichen
Label ?

Label-Inhalt
mit internem RAM
vergleichen

Labels bisher identisch
Label-Ende im int. RAM ?

und im BASIC-Programm ?

Label gefunden, \$02=22

Adresse der nächsten
Programmzeile berechnen

```
6038 10 70 A7 LD X,N
603B 02      LD P,02
6034 00 07   LD L,n
6036 10      LLD(P),(X)
6037 10 6C 3F LD X,N
603A 02      LD P,02
603B 19      LEX(P),(X)
603C 37      RET
```

das vorstehende Programm
für den SHARP PC-1350
kopiert das Ergebnis der
letzten arithmetischen
Operation in den Vari-
ablenregister X.
Anwendung z.B. hinter
dem Befehl AREAD um 2
Parameter überleben zu
können.

```
500:"X" AREAD A: CALL 66
  C30: PRINT A,X
```

Durch die Tastenfolge
Zahl/ENTER/Zahl/DEF X
können Sie 2 Werte ein-
geben. Aber auch nach ei-
nem Stop durch
WAIT/PRINT lässt es sich
als AREAD-Ersatz verwen-
den. Mit den hier verwen-
deten Adressen belegt es
die Speicher Y und Z

Günther Weber
Händelstr. 8
4320 Maitingen

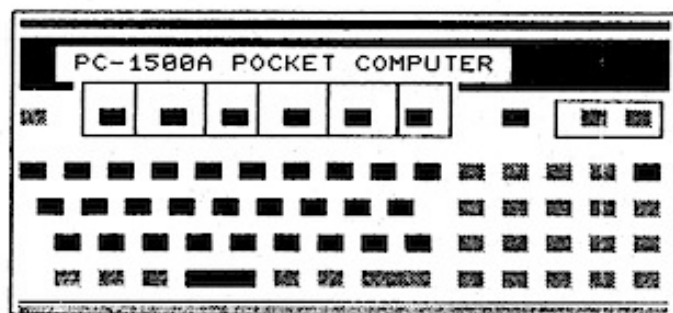
```
1200: IF GS=PK8BEEP
  3: WAIT 150:
  PRINT "Sie h
  oben "PK8;" gew
  onnen !";
  GOTO "END"
1290: IF GS=PCBEEP
  3: WAIT 150:
  PRINT "Ich h
  oben "PC8;" gew
  onnen !";
  GOTO "END"
1300: CLS: WAIT 0:
  CURSOR 5:
  PRINT CHR$(8)
  7: "Neues Sp
  iel !"; CHR$(
  8):
  GOTO "NS"
1310: FOR I=20 TO 5
  STEP -1:
  BEEP 1, I, 25:
  NEXT I
1320: WAIT 0: GOTO
  "KA"
1330: REM SVEN NIM
  SGARN
1340: REM FINKENRI
  ED 4d
1350: REM 2000 NOR
  DERSTEDT
1360: "END": WAIT 0:
  CLS: BEEP 1:
  PRINT "Noch
  ein Spiel (J
  /N) ?"
1370: "END": A=ASC
  INKEY$
1380: IF A=28: CLS:
  GOTO "ENDE"
1390: IF A=24: GOTO
  "A"
1400: GOTO "END1"
1410: "END": FOR I=
  1550 TO 30 STEP
  -1: BEEP 1, I,
  10
1420: CURSOR 1:
  PRINT "(c) S
  ven Nimsgarn
  !"; NEXT I
1430: WAIT 10: FOR
  I=1 TO 10:
  BEEP 1, 20+I,
  15: CURSOR 3
  6: PRINT "C"
1440: CURSOR 36:
  PRINT "C":
  NEXT I
1450: "END"

STATUS 1 4896
DATEN: 1747
GESAMT: 6543

920: "COMP": CLS: J=0
  SC=0: A=0: B=0:
  BEEP 1, 50, 10
  BEEP 1, 25, 1
  0
1170: IF SC=SW
  GOTO "COMPI"
1180: IF SC=AND S
  C<(SW+2): GOTO
  "COMPI"
1190: WAIT 150:
  PRINT "WAIT
  100"
1200: IF SC=21: LET
  PK=PK+1: BEEP
  1, 50, 50: BEEP
  1, 100, 50:
  BEEP 1, 50, 15
  0: PRINT "Ih
  r Spiel !";
  GOTO "NS"
1210: IF J=2AND SC
  =220: SC=21:
  LET PC=PC+1:
  BEEP 1, 100, 5
  0: BEEP 1, 50,
  50: PRINT "H
  ein Spiel !";
  GOTO "NS"
1220: IF J=5AND SC
  =20AND SC=21
  LET PC=PC+1:
  BEEP 1, 100, 5
  0: BEEP 1, 50,
  50: PRINT "H
  ein Spiel !";
  GOTO "NS"
1230: IF SC=5: LET P
  C=PC+1: BEEP
  1, 150, 50:
  BEEP 1, 100, 5
  0: BEEP 1, 50,
  50: PRINT "H
  ein Spiel !";
  GOTO "NS"
1240: IF SC=5: LET P
  C=PC+1: BEEP
  1, 150, 50:
  BEEP 1, 100, 5
  0: BEEP 1, 50,
  50: PRINT "H
  ein Spiel !";
  GOTO "NS"
1250: IF SC=5: LET P
  C=PC+1: BEEP
  1, 150, 50:
  BEEP 1, 100, 5
  0: BEEP 1, 50,
  50: PRINT "H
  ein Spiel !";
  GOTO "NS"
1260: WAIT 0: GOTO
  "KA"
1270: "NS": CLS: PC
  =STR$ PC: PK
  =STR$ PK

1130: SC=STR$ SC:
  PC=STR$ PC
1140: CURSOR 102:
  PRINT CHR$(8)
  7: "P": PC$
  PRINT CURSOR 132:
  7: SC$
  CURSOR 150:
  PRINT CHR$(8)
  7: WAIT 20:
  PRINT "WAIT
  0"

Do not sale this PDF !!!
```

SOFTWARE
für die Entwicklung von
Maschinenprogrammen
in SHARP - Maschinsprache

SHARP - PC-1500/A

POCKET-COMPUTER

HEX-EDITOR-MONITOR

- Eingabe von Hexcode oder Zeichen
- Editieren: Einfügen oder Löschen von einzelnen Bytes
- Austesten mit Anzeige und Änderungsmöglichkeit aller CPU-Register
- Einzelschritt und Haltepunkt
- Kein Speicherzusatz erforderlich

Auf Cassette mit Bedienungsanleitung
1,57 kByte, relocatibel, incl.MwSt.: DM 43,--

MACRO- ASSEMBLER-MONITOR

Wie ASSEMBLER-MONITOR, jedoch um mehrere Funktionen erweitert, z.B.:
Macros (definieren, aufrufen, listen, löschen).

- PC-1500 : mind.8 kB-Erweiterung erforderlich
- PC-1500A: mind.4 kB-Erweiterung erforderlich

Auf Cassette mit Bedienungsanleitung,
etwa 6 kByte, relocatibel,
incl.MwSt.: DM 98,--

ASSEMBLER-MONITOR

- Eingabe von Befehlsworten (Mnemonic)
- Original SHARP-Maschinsprache
- Editieren: Einfügen und Löschen von Befehlen
- Direkte oder Label-Adressierung
- Ein-Lauf-Assembler
- Minimaler Speicherplatzbedarf. Es wird kein Basic-Programm benötigt
- Prüfsummenroutine, Copyroutine
- Zwei Ausdruckroutinen für CE-150
- Nach eigenen Wünschen erweiterbar. Drei Erweiterungsvorschläge, z.B. Macros (definieren, aufrufen, listen u. löschen).
- Testlauf der entwickelten Programme mit Befehlsanzeige und Anzeige und Änderungsmöglichkeit aller CPU-Register
- Single Step und Breakpoint
- PC-1500: mind.4kB-Speichererweiterung erforderlich
- PC-1500A: keine Speichererweiterung erforderlich

Auf Cassette mit Bedienungsanleitung und
Erweiterungsvorschlägen,
4,96 kByte, relocatibel, incl.MwSt.: DM 98,--

alle Preise inkl.14%Mwst.



Do not sale this PDF !!!

Sharp



Bedienungsanleitung

zum

PC-1500/A ASSEMBLER-MONITOR

und zum

PC-1500/A MACRO-ASSEMBLER-MONITOR

© 1984 Dr. J. Stange

Version: LASMO 124

Länge: 4,96 KByte (z.B. &2000 - &33D8)

Prüfsumme: 7E0D

Version: LASMO 124E (incl. drei Erweiterungen)

Länge: 5,98 KByte (z.B. &2000 - &37E7)

Prüfsumme: 6B4F

Für etwaige Schäden, die durch unsachgemäße Anwendung der Maschinensprache entstehen, übernehmen wir keine Haftung.

Inhaltsverzeichnis

- I Einleitung
- II Übersicht über die Tastenfunktionen und Beispiele für die Eingabe von Befehlen und Labels
- III Beschreibung des Programmes
- Anhang 1: Ausdruck aller Befehle
- Anhang 2: Die VBJ-Befehle im Basic-Interpreter
- Anhang 3: Das Wichtigste zur Speichereinteilung
- Anhang 4: Erweiterungsmöglichkeiten des PRO-Mode

MoveAB

I EINLEITUNG

Der Label-Assembler-Disassembler-Editor (Insert, Delete)-Monitor (Single Step, Breakpoint) ist in einen PRO-Mode und einen RUN-Mode unterteilt. Das Programm benutzt bis auf eine Ausnahme beim VBJ-Befehl die gleiche Bezeichnung der Befehle wie die Firma Sharp im Pocket Computer PC-1500 Technical Reference Manual, Befehlsliste siehe Anhang 1. Es ist relocatable und kann somit mit CLOAD Adresse an eine beliebige Stelle des Speicherbereiches für Maschinenprogramme geladen werden. Das Programm benötigt 4,96 KByte Speicherplatz. Zusätzlich benutzt es den Speicherbereich der Textvariablen A\$ bis U\$.

Die im PRO-Mode als Befehlswoorte (Mnemonic) eingegebenen Befehle werden vom Programm unmittelbar in den Befehlscode umgewandelt und bei der Anzeige oder beim Ausdrucken vom Maschinencode wieder in die Befehlswoorte zurückübersetzt. Solange keine Label benutzt werden, entsteht bei der Eingabe unmittelbar ein lauffähiges Programm. Es können jedoch auch für alle absoluten und relativen Verzweigungen Label benutzt werden. In diesem Fall wird zuerst ein Quellprogramm erzeugt, das mit Hilfe eines Organisationsprogrammes in das lauffähige Objektprogramm umgewandelt wird. Das Quellprogramm ist kaum umfangreicher als das Objektprogramm. Es beansprucht lediglich drei zusätzliche Speicherplätze für jedes gesetzte Label. Es können auch Datenfelder in das Quellprogramm gesetzt werden.

Neben der Labelorganisationsroutine stehen zwei Ausdrucken-, eine Prüfsummen-, eine Kopier- und eine Labelsuchroutine zur Verfügung.

Das Programm ist so aufgebaut, daß der PRO-Mode das Programmende mit einem RTN-Befehl berührt. Dies gibt dem Benutzer die Möglichkeit, Erweiterungen nach eigenem Bedarf anzufügen. Im Anhang 4 sind drei Beispiele angegeben, z.B. Macros (definieren, aufrufen, listen, löschen).

I-1



II.1 ÜBERSICHT ÜBER DIE TASTENFUNKTIONEN IM PRO-MODE

CALL Programmanfangsadresse

Starten des Programmes.

Taste . (Punkt)

Beendigung des Programmes.

Taste ↓

Anzeige des nächsten Befehls.

Taste ↑

Anzeige des letzten (angezeigten) Befehls.

Taste M

Änderung der angezeigten Speicheradresse. Es ist nur Eingabe einer Hexzahl, ENTER, CL oder BREAK möglich.

Taste < (ohne SHIFT)

Änderung der Zweibyte- oder/und Einbyte-Immediate des angezeigten Befehls.

Taste CL

Nullsetzen einer Hexzahl bei der Eingabe.

Taste BREAK

Abbruch einer Eingabe.

Buchstaben-Taste (außer M)

Wird als erster Buchstabe eines einzugebenden Befehls interpretiert.

ENTER

Abschluß einer Hexzahleingabe. (Bei << stets nötig. Bei < nicht nötig, falls alle Stellen eingegeben wurden.) Abschluß einer Befehlseingabe, falls der Befehl in die angezeigte Speicheradresse eingegeben werden soll.

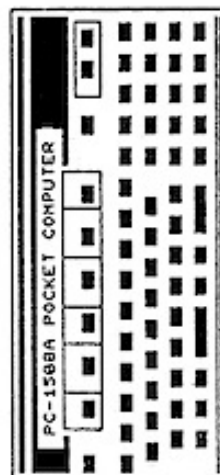
Taste INS (ohne SHIFT)

Abschluß einer Befehlseingabe, falls der Befehl auf der

Der RUN-Mode gibt die Möglichkeit, die im PRO-Mode erstellten Programme mit den Monitorfunktionen auszutesten. Neben dem Befehlswort werden die Register des Prozessors angezeigt und können geändert werden.

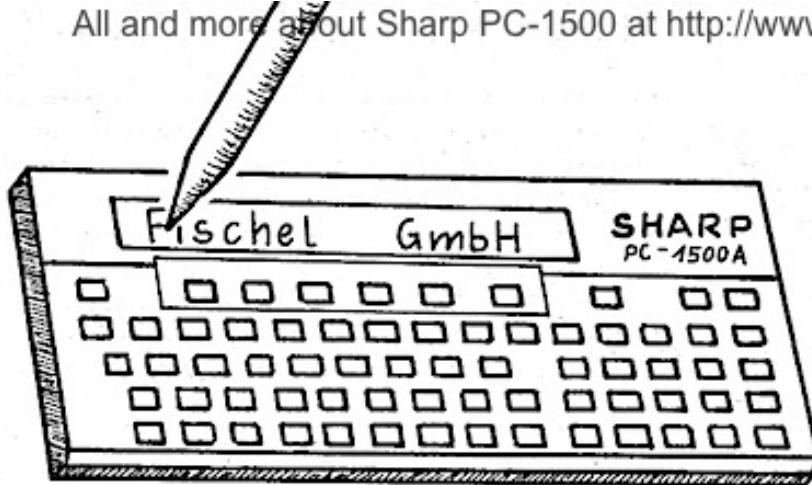
Der Single Step ist mit einem Interrupt realisiert und kann somit auch in schreibgesperrten Speicherbereichen durchgeführt werden.

Der Start eines auszutestenden Programmes mit oder ohne Breakpoint ist auf zwei Arten möglich. In der einen ist das Interruptsystem initialisiert, sodas nach der Break-Taste die Kontrolle an den Monitor zurückgeht, es sei denn, das auszutestende Programm hat die Interruptinitialisierung rückgängig gemacht. In der anderen wird das Interruptsystem nicht benutzt. Hierdurch ergibt sich die Möglichkeit, auch Befehle, die das Interruptsystem betreffen, auszutesten.



SHARP - PC-1500/A
POCKET-COMPUTER

SOFTWARE
für die Entwicklung von
Maschinenprogrammen
in SHARP - Maschinensprache



Taste SHIFT L

Labelorganisationsprogramm. Quellprogrammstart- und -endadresse und die Anfangsadresse, auf der das lauf- fähige Objektprogramm benutzt werden soll, werden ab- gefragt. Die Endadresse muß um mindestens 8 größer sein als die Anfangsadresse.

Taste /

Labelprogramm. Suchbereichsstart- und -endadresse und die vierstellige Label-Hexzahl werden abgefragt.

Print in Stellung P

Es werden die Ein-/Ausgaben von LIST1, LIST2, CHECKSUM, COPY und LABO protokolliert.

BEISPIELE FÜR DIE EINGABE VON BEFEHLEN ODER LABELN

Befehl/Label	Eingabe
LAB BB12	LAB ENTER BB12
AEX	AEX ENTER
LDA XH	IDAXH ENTER
STA X	STAX ENTER
STA *X	STA*X ENTER
LDI XH 12	LDIXH ENTER 12
BCH -12	BCH- ENTER 12
JMP 1234	JMP ENTER 1234
ADC 1234	ADC ENTER 1234
ADC *1234	ADC* ENTER 1234
ADI 1234 AB	ADI ENTER 1234AB
ADI *1234 AB	ADI* ENTER 1234AB



SOFTWARE
für die Entwicklung von
Maschinenprogrammen
in SHARP - Maschinensprache

SHARP - PC-1500/A
POCKET-COMPUTER

II-3

angezeigten Speicheradresse in die vorhandene Befehls- folge eingefügt werden soll. (Editorbereichende wird angezeigt und kann mit ENTER bestätigt oder nach SHIFT E geändert werden. Die Editorbereichsadresse muß um mindestens 8 größer sein als die angezeigte Speicheradresse.)

Taste DEL (ohne SHIFT)

Der angezeigte Befehl wird aus der vorhandenen Befehls- folge herausgenommen. (Für den Editorbereich gilt das gleiche wie bei INS.)

Taste SHIFT E

Eingabe der Editorbereichsadresse, bis zu der bei INS oder DEL alle nachfolgenden Befehle beeinflusst werden.

Taste ! (Erste Reservetaste)

Ausdrucken einer Befehlsfolge in der ersten Form: Speicheradresse, vollständiger Code, Befehlswort bei CSIZE 1 oder 2. Anfangs- und Endadresse werden abgefragt. Erstere darf nicht größer sein als die zweite.

Taste " (Zweite Reservetaste)

Ausdrucken einer Befehlsfolge in der zweiten Form: Meist nur die letzte Stelle der Adresse, Befehlswort, usw.

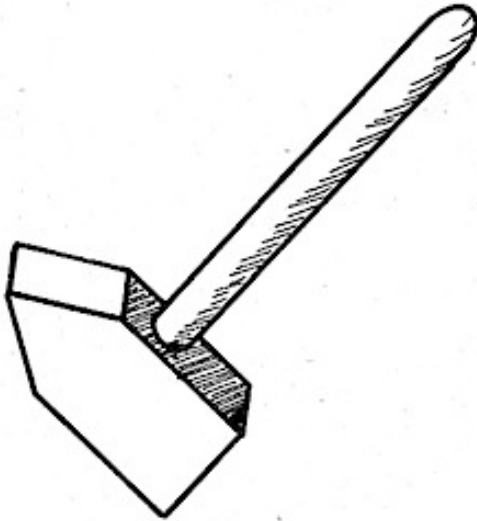
Taste SHIFT C

Kopieren eines Speicherbereiches. Quellbereichsstart- und -endadresse und Zielbereichsstartadresse werden ab- gefragt. Die Quellbereichsstartadresse darf nicht grö- ßer sein als die Endadresse und der Zielbereich darf die 64 KByte nicht überschreiten.

Taste \$ (4. Reservetaste)

Berechnung einer vierstelligen Hexprüfsumme. Anfangs- und Endadresse werden abgefragt.

II-2



Taste ! (nach *)

Listen aller Macros (Name, Speicherbereich) auf der Anzeige oder, bei Printscharter in Stellung P, auf CE 150.

Taste DEL (ohne SHIFT) (nach *)

Löschen des eingegebenen Macro aus dem Lexikon.

Taste CL (nach *)

Löschen aller Macros ab dem eingegebenen.

Taste CA (nach *) und ENTER

Löschen des gesamten Macro-Lexikon

SOFTWARE
für die Entwicklung von
Maschinenprogrammen
in SHARP - Maschinensprache

SHARP - PC-1500/A
POCKET-COMPUTER

CSIZE-Erweiterung

Taste SHIFT ! (Erste Reservetaste)

Wechsel zu CSIZE 1.

Taste SHIFT " (Zweite Reservetaste)

Wechsel zu CSIZE 2.

HEX-Erweiterung: & (#)

Taste & (#)

Anzeige von Hexcode/Zeichen aus MEO (ME1).

Taste M

Änderung der Speicheradresse.

Taste ↓ (↑)

Anzeige des nächsten (letzten) Speicherinhalt.

Taste BREAK

Rückkehr zur Befehlseingabe.

Hexaste

Einleitung der Eingabe eines Hexcode.

Taste < (ohne SHIFT)

Wechsel zur schnellen Hexeingeabe.

Taste "

Wechsel zur Zeicheneingabe.

Taste !

Ausdruck von Hexcode/Zeichen.

MACRO-Erweiterung: *

Taste : (nach *)

Definition eines Macro.

Macroname (nach *)

Aufruf eines Macro.

Do not sale this PDF !!!

Nach jedem Tastendruck schreibt der Computer in die Adressen &780A und &780B den Wert &FE1D. Während er auf den nächsten Tastendruck wartet, zählt er ca. im 0,87 s Takt von diesem Wert weiter, bis er &10000 erreicht hat. Dann schaltet er sich aus. Je höher der Wert in &780A/B ist, desto kürzer ist also die APO-Zeit:

$(\&10000 - \&FE1D) \times 0,87s = 7min$ vom System aus und

$(\&10000 - \&FFDE) \times 0,87s = 30s$ im Beispielprogramm.

Die zeitbestimmenden (im Programmlisting unterstrichenen) Werte können beliebig variiert werden.

Mein Programm schreibt nach jedem Tastendruck diesen Wert (im Beispiel &FFDE) in &780A/B und springt dann hinter den Teil im ROM, der den Systemwert &FE1D für 7min setzt.

Der Maximalwert von &10000 entspricht rechnerisch etwa 15h 47min.

Da durch das Ausschalten des Computers durch die (OFF)-Taste der Wert in &79D4 gelöscht wird, empfiehlt es sich den Rechner nur

durch APO ausschalten zu lassen, um sich das "POKE &79D4, &55" nach dem Einschalten zu sparen.

Programmlisting:

10:A= Startadresse (beliebig)

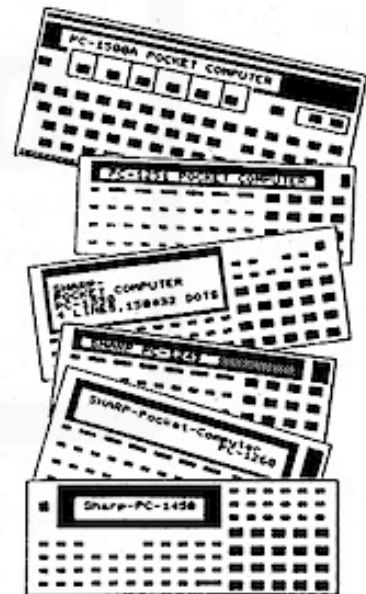
20:POKE A, &E9, &78, &0E, &FD, &48, &7B, &ED, &7B, &0E, &40, &89, &05, &4A, &09, &85, &60

30:POKE A+&10, &41, &4A, &0A, &05, &FF, &41, &85, &DE, &8A, &E2, &62

40:POKE &7B5B, A/256, A AND 255:POKE &79D4, &55

Viel Spaß beim Ausprobieren,

Torsten Crull
G.-Hauptmann-Str. 71
2000 Wedel



Anwendungs- Handbuch

zum

SHARP

Taschencomputer

Preis 49,-DM inkl. 7% MWS

FISCHEL GmbH

ISBN 3-924 327-14-9

INGO LAUE



III.1 BESCHREIBUNG DES PROGRAMMES, PRO-MODE

Anzeige der Befehle

Nach dem Start meldet sich das Programm kurz mit Name und Version. Danach erscheint in der Anzeige eine Speicheradresse als Hexzahl und das zugehörige Befehlswort inclusive eventueller Immediate. Ganz rechts steht der Code des Befehls.

Nach $\uparrow$ (↑) wird der nächste (letzte) Befehl angezeigt. Letzteres ist jedoch nur möglich, falls der letzte Befehl bereits angezeigt wurde. Dabei wurde seine Bytezahl registriert. Hierfür stehen 176 Ebenen zur Verfügung. (Bei Festhalten der Tasten: Schnellauf.)

Änderung der Speicheradresse

Nach der Taste M läßt sich die angezeigte Speicheradresse ändern. Neben ihr erscheint das Symbol $\langle$, das anzeigt, daß die Hexzahl von rechts geändert wird, so ist die schnelle Änderung der letzten Stelle möglich.

Die Taste CL setzt die Hexzahl auf Null und die Eingabe beginnt erneut. Abgeschlossen wird die Hexzahl durch die Taste ENTER, es sei denn, es wurden alle vier Stellen eingegeben, dann ist ENTER nicht mehr nötig. Die Taste BREAK bewirkt den Abbruch der Hexzahleingabe. Es erscheint wieder der ursprüngliche Befehl in der Anzeige.

Änderung von Immediate

Nach der Taste $\langle$ (ohne SHIFT) lassen sich die Zwei-/Byte-Immediate des angezeigten Befehls ändern.

Eingabe von Befehlen

Die Eingabe eines Buchstaben (außer M) leitet die Befehlseingabe ein. Eingegebene Befehle können in die angezeigte Speicheradresse eingeschrieben (Abschluß ENTER) oder in die vorhandene Befehlsfolge eingefügt werden (Abschluß INS). Nach drei Zeichen wird die Befehlseingabe überprüft.

III-1

II.2 ÜBERSICHT ÜBER DIE TASTENFUNKTIONEN IM RUN-MODE

Taste MODE

Wechsel vom PRO-Mode zum RUN-Mode und umgekehrt.

Taste .(Punkt)

Beendigung des Programmes.

Taste C (Change)

Wechsel von der Programmcounter/Befehlswort-Anzeige zur Register-Anzeige und umgekehrt.

Taste A, X, Y, U, T (oder P), S, P

Vorbereitung zur Änderung des Accumulators, der X-, Y-, U-Register, der Flags T, des Stackpointers und des Programmcounters. Bei der Eingabe des letzteren wird gleichzeitig der Stackpointer initialisiert.

Taste SHIFT P

Register- und Stackpointer-Initialisierung mit dem zuletzt eingegebenen Programmcounter.

Taste $\pm$

Single Step.

Taste R (RUN+I) oder Taste +

Start eines Programmes mit oder ohne Breakpoint. Rückkehr zur Monitorkontrolle mit der Taste BREAK.

Taste SHIFT R (RUN-I) oder Taste -

Start eines Programmes mit oder ohne Breakpoint ohne Benutzung des Interruptsystems.

Printschalter in Stellung P

Es werden jeweils erst der Registerstand und dann der nächste auszuführende Befehl ausgedruckt. Außerdem werden beim Start mit R (+) oder SHIFT R (-) der Programmcounter und der Breakpoint ausgedruckt.

II-6



Alles für
SHARP-Computer

PC-1245
PC-1251
PC-1260
PC-1261



Die imarbeit

EJ benutzt. (Siehe auch Anhang 2.) Der Befehl LOP ist bei Sharp auch teilweise mit LOP UL bezeichnet. Hier wurde LOP gewählt. Code, die keinen Befehl bedeuten, werden mit ihrem Code benannt und erscheinen so doppelt in der Anzeige. Code können ins Programm nicht eingegeben werden, siehe jedoch HEX-Erweiterung im Anhang.

Eingabe von Hexzahlen

Bei der Eingabe von Hexzahlen wird von rechts eingegeben, bei << muß mit ENTER abgeschlossen werden, bei < ist dies nicht nötig, wenn alle Stellen eingegeben werden. Durch CL wird die Hexzahl Null gesetzt, mit BREAK wird die Eingabe abgebrochen.

Ausdrucken von Programmen

Nach der Taste ! (1. Reservetaste) werden Anfangs- und Endadresse abgefragt und anschließend die Befehlsfolge in der ersten Form ausgedruckt. Bei CSIZE 1 erscheinen in einer Zeile die Speicheradresse, der vollständige Befehlscode und der Befehlsname mit Immediate. Ist die Endadresse kleiner als die Anfangsadresse, so wird nicht ausgedruckt.

Nach der Taste " (2. Reservetaste) wird die Befehlsfolge in der zweiten Form ausgedruckt. Der Ausdruck geschieht in komprimierter Form ohne Befehlscode, mit abgekürzten Adressen und ohne unnötige Zwischenräume.

Prüfsumme, CHECKSUM

Taste \$ (Vierte Reservetaste)

Eingabe: Anfangs- und Endadresse.

Ausgabe: Zusätzlich zu den beiden eingegebenen Adressen wird die vierstellige Hexzahl angezeigt, die durch die Addition aller Speicherinhalte des Bereiches entstanden ist.



III-3

Palls nicht akzeptiert wird, ertönt BEEP, BEEP und es erscheint wieder der ursprüngliche Befehl in der Anzeige. Anschließend wird gegebenenfalls ein #, ein eventuelles Registerargument oder ein + oder - eingegeben und die Befehlseingabe durch ENTER oder INS (ohne SHIFT) abgeschlossen, also stets vor den eventuell noch eingegebenen Zweibyte- oder/und Einbyte-Immediate. Bei INS wird die Editorbereichsadresse angezeigt, gleichzeitig ertönt ein BEEP. Sie kann nun entweder mit ENTER bestätigt oder nach SHIFT E geändert werden. Diese Editorbereichsadresse bestimmt die Speicheradresse, bis zu der der Speicherinhalt durch Verschieben der nachfolgenden Befehle beeinflusst wird. Nach dem Starten des Programmes beträgt sie stets 0008. Sie muß um mindestens 8 Plätze größer sein als die angezeigte Speicheradresse, anderenfalls wird nicht akzeptiert, BEEP, BEEP. Danach wird der Befehl eingeschrieben oder eingefügt und der nächste im Speicher befindliche Befehl angezeigt, es sei denn, der eingegebene Befehl benötigt die Angabe von ein oder zwei Immediate. In diesem Fall müssen sie durch ENTER bestätigt oder geändert werden, bevor der nächste Befehl angezeigt wird. Mit BREAK wird die Eingabe eines Befehls abgebrochen.

Das Löschen eines Befehls geschieht mit der Taste DEL (ohne SHIFT). Auch hier wird vor der Durchführung die Editorbereichsadresse angezeigt, die mit ENTER bestätigt oder nach SHIFT E geändert werden kann.

Überprüfung der Befehlseingabe

Bei der Eingabe von Befehlen werden sie überprüft und nur akzeptiert, wenn existiert. Wird die Eingabe nicht akzeptiert, so ertönt stets BEEP, BEEP und es erscheint wieder der vorher angezeigte Befehl in der Anzeige. Der Ausdruck sämtlicher Befehlsnamen ist im Anhang 1 angefügt. Die VBJ-Befehle werden hier ohne die Buchstaben

III-2

Do not sale this PDF!!!

ÜBERRASCHUNG:
PC-1500/A

mit freundlicher Genehmigung vom:
Club des SHARPENTIERS S.B.M.
151-153 av. Jean-Jaurès
F-93307 Aubervilliers Cedex

"LABEL ? TASTE" angezeigt und es ertönt BEEP, BEEP. Nach Betätigung einer Taste wird der Labelbefehl angezeigt, für den das Label nicht gefunden wurde. Nach Korrektur im Quellprogramm muß LABO neu gestartet werden.

Beschreibung

Das Quellprogramm wird mit Labeln LAB XXXX (nur LAB DDPF ist für die Kennzeichnung von Datenfeldern reserviert) und Labelbefehlen (siehe Liste) aufgebaut. Bei den absoluten Labelbefehlen, JWL (zu JMP) und SJL (zu SJP), können vierstellige Hexzahlen als Label-Nummer angegeben werden, wie bei den Labeln. Letztere stehen vor den Befehlen, auf die im Programmablauf gesprungen werden soll. Bei relativen Labelbefehlen, LOL (zu LOP), LOH + (zu BOH +), LOH - (zu BOH -) und den bedingten Verzweigungen (jeweils B durch L ersetzt), können nur zweistellige Hexzahlen als Label-Nummer eingegeben werden. Die zugehörigen Label müssen mit BB (Branch-Befehl) beginnen. Also zu LZS - 03 gehört das Label LAB BBO3.

Das Labelorganisationsprogramm LABO wandelt die Labelbe- fehle in die Maschinenbefehle um und ordnet ihnen die richtigen Sprungadressen oder relativen Sprungweiten zu. Bei den letzteren wird auch überprüft, ob die Label im Einzugsbereich liegen (davor, dahinter, nicht zu weit entfernt). Anschließend werden die Labelmarken aus dem Quellprogramm entfernt.

Fehlermöglichkeiten

Abgesehen davon, daß ein Label nicht oder nicht im Ein- zugsbereich vorhanden ist, gibt es weitere Fehlermög- lichkeiten. Da das Labelorganisationsprogramm das Quell- programm in Befehlsschritten durchsucht, kann es z.B. ein Label (oder auch einen Labelbefehl) nicht finden, das im 2. Byte eines Mehrbytebefehls beginnt. Solange das Quell- programm Befehl für Befehl vollständig über die Tastatur eingegeben wurde, kann dieser Fehler nicht auftreten. Gibt man jedoch ein Label, das zu einem noch nicht ent- wickelten Programmteil führen soll, irgendwo ein, so ist

III-5

Kopierprogramm, COPY

Taste SHIFT C

Eingabe: Quellbereichansfangsadresse - Quellbereich- endadresse -- Zielbereichansfangsadresse. Die Quellbe- reichendansadresse muß größer oder gleich der Quellbereich- ansfangsadresse sein, anderenfalls wird das Kopierpro- gramm nicht durchgeführt, BEEP, BEEP. Außerdem darf der Zielbereich die 64 KByte nicht überschreiten. Funktion: Der Quellbereich wird auf den Zielbereich ko- piert. Überdeckt der Zielbereich den Quellbereich teil- weise, so wird der letztere teilweise überschrieben. Ausgabe: Nach der Durchführung wird neben der Quellbe- reichansfangs- und -endansadresse auch die Zielbereichan- fangs- und -endansadresse angezeigt.

Labelorganisation, LABO

Taste SHIFT L

Eingabe: Quellprogrammansfangsadresse (QA) - Quellpro- grammendansadresse (QE) -- Objektprogrammansfangsadresse (OA). QE muß die Adresse des letzten Byte des letzten Befehls oder des letzten Byte eines Datenfeldes sein. Es muß QE größer oder gleich QA + 8 sein, anderenfalls wird abgebro- chen, BEEP, BEEP.

Funktion: Das im Bereich QA - QE vorhandene Quellpro- gramm, das Label und Labelbefehle enthält, wird in ein lauffähiges Objektprogramm umgewandelt, das beginnend auf der Adresse OA benutzt werden soll. Das Quellprogramm kann auch Datenfelder (siehe anschließend) enthalten.

Ausgabe: Bei fehlerfreiem Ablauf werden die Quellprogramm- anfangs- und die neue -endansadresse und die Objektprogramm- anfangs- und -endansadresse angezeigt. Das Objektprogramm steht im Bereich QA - QE und muß nun mit COPY nach OA kopiert werden.

Falls zu einem Labelbefehl das Label nicht gefunden wird oder, falls bei einem relativen Sprung das Label nicht im Einzugsbereich des Befehls gefunden wird, wird

III-4

```
10: "A" CLEAR : DIM
2*(5)*12,*(5)
*(8)**(20)*10:
2*(8)*"0000207
FE000:2*(1)*"
0000000F1000
20: 2*(2)*"0004221
F3400:2*(3)*"
0024221F1224":
2*(4)*"0014221
F2440
30: 2*(5)*"0000163
F6400:2*(8)*"
2F00000000007F
00000007020000
40: 2*(1)*"2F00000
000007F0000000
0000000:2*(2)*"
123456789ABCDEF
F: PAUSE "Gute
n Tag
50: INPUT "Dein Vo
name, bitte",
A: CURSOR 19:
WAIT 8: PRINT "
Mosent
99: WAIT 8
100: FOR B=1 TO LEN
A: FOR D=(B-1)
*6 TO D+5: GOSUB
500: NEXT D:
NEXT B: CLS
```

```

105: J=155-LEN A#*6
-38: FOR D=870
J: GPCURSOR D:
PRINT 127:
NEXT D
110: FOR B=10 LEN
A#*6: V#(0)=-7F" +
X#(B)*RIGHT# (
V#(B), 16)
B#*RIGHT# (Y#(
1), 16)
130: FOR C=15510 (8
-1)*6: JSTEP -1
140: I=8: IF C-INT (
C/2)*2LET I=1
150: GPCURSOR C:
PRINT Y#(1):
NEXT C: H#C+14
160: GPCURSOR C+2:
PRINT "8866IF
668888
170: FOR C=870 8:
GPCURSOR H-13:
PRINT POINT (
H-13)-2^C#(
POINT (H-13)-2
^C)=8)
180: GPCURSOR H-7:
PRINT POINT (
H-7)-2^C#(
POINT (H-7)-2^
C)=8): NEXT C
190: FOR C=H-710 15
: G=C-INT (C/6
): G: GPCURSOR C:
PRINT 2#(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# (H#, E+1, 1
): MID# (H#, F+1
): RETURN

```

ten. Bei der LABO-Routine werden diese Datenfelder aus-
gespart und etwaige Label oder Labelbefehle darin nicht
beachtet.

Die Datenfeldlänge kann im Prinzip zwischen 0000 und
FFFF liegen, wobei 0000 heißt: April, April, das Daten-
feld enthält kein Byte. Die Datenfeldlängen werden vom
Programm überprüft und bei unsinniger oder fehlender An-
gabe wird nicht akzeptiert, BEEP, BEEP, und nach einer
Taste das Datenfeldlabel LAB DFFF angezeigt. Folgende
Fehler können aufgetreten sein.

- 1) Es fehlt das Datenfeldlängenlabel.
- 2) Das Datenfeldende liegt hinter dem Quellprogrammende.
- 3) Das Datenfeldende reicht über 64 KByte hinaus.

Labelsuchprogramm, SLAB

Taste /

Eingabe: Suchbereichanfangs- und -endadresse und die
Labelnummer (vierstellige Hexzahl).
Funktion: Falls das gesuchte Label im Suchbereich nicht
gefunden wird, ertönt BEEP, BEEP. Ist das Label vorhan-
den, so wird es angezeigt. Gleichzeitig wird die Suchbe-
reichanfangsadresse hinter das angezeigte Label gesetzt,
so daß der Suchbereich nach erneutem Aufruf von SLAB
nach weiteren gleichen Labeln durchsucht werden kann.

Beendigung des Programmes, Ausschalten

Das Programm wird mit der Taste . (Punkt) beendet, sie
führt zum Basic-Interpreter zurück. Ist das Programm in
Funktion, so wird der Computer automatisch ausgeschaltet,
falls ca. 7 Minuten lang keine Taste betätigt wurde. Will
man den Computer ausschalten, so muß zuerst die Taste.
betätigt werden.

Die CSIZE-, HEX- und MACRO-Erweiterung werden im Anhang
im einzelnen beschrieben.

darauf zu achten, daß es in die dort vorhandene Befehls-
folge eingefügt wird und nicht in einem Mehrbytebefehl
beginnt. Am sichersten ist es, den ganzen Bereich des
Quellprogramms anfangs mit Einbytebefehlen (NOP oder FF,
aus einem leeren Speicherbereich kopiert) zu füllen. Es
können auch Daten im Quellprogramm stehen. Sie müssen
jedoch als Datenfeld gekennzeichnet werden (siehe an-
schließend), andernfalls könnten sie Label oder Labelbe-
fehle vertauschen und würden dann verändert werden.

Label und Labelbefehle

LAB 0123

JML 4567

SJL 89AB

LCL + EP

LCH - 01

LCR + 23

LCS - 45

LCS + 67

LCS - 89

LHR + AB

LHR - CD

LMS + EF

LMS - 01

LZR + 23

LZR - 45

LZS + 67

LZS - 89

LVR + AB

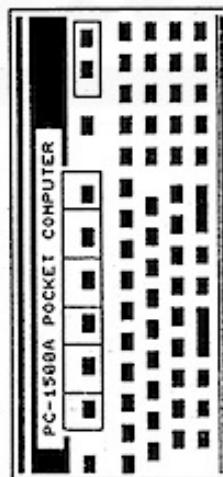
LVR - CD

LVS + EF

LVS - 01

Datenfelder

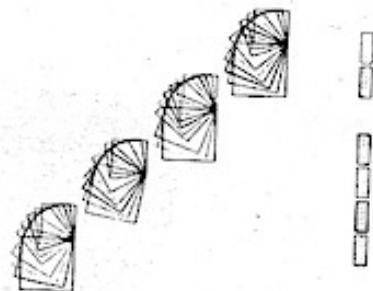
Daten im Quellprogramm könnten Label oder Labelbefehle
vertauschen und würden dann bei der LABO-Routine verändert
werden. Aus diesem Grunde müssen sie, unmittelbar vorange-
hend, durch zwei Label gekennzeichnet werden. Das erste
Label, LAB DFFF kennzeichnet den folgenden Bereich als Da-
tenfeld, das nächste, das Datenfeldlängenlabel, gibt die
Länge des Datenfeldes an. Enthält das Datenfeld z.B. drei
Byte, so wird es durch LAB DFFF, LAB 0003 gekennzeichnet.
Ein Quellprogramm kann beliebig viele Datenfelder enthal-



SHARP - PC-1500/A -
POCKET-COMPUTER

Compact and portable, the Sharp CE-140P Color Dot Printer adds a touch, or a splash of color to your graphic output. Whether you need to emphasize important data from financial calculations, or make complex experimental results easier to understand, the CE-140P can give your printout visual appeal and impact in any of 7 different colors. The CE-140P is battery powered, uses normal paper, and is designed to integrate with Sharp's popular PC-1350 and PC-1450 Pocket Computers for a "go-anywhere" computer/printer combination—ideal for the office, laboratory or field use. The CE-140P Color Dot Printer—for the computer user who wants to develop a reputation for creative solutions and colorful output.

SHARP



Monitor zum letzten eingegebenen Programmcounter zurück und initialisiert neu. Ändert man den Stackpointer mit der Taste S, so sollte nur noch mit Single Step oder Breakpoint gestartet werden.

Für den Stackbereich wurde der Basic-Stackbereich gewählt, der von 7A38 bis 7A7F reicht. Es steht also für normale Programme ausreichend Stackbereich zur Verfügung. Der Monitor selbst benötigt davon ca. 30 Plätze.

Single Step

Der Single Step (Einzelschritt-Befehlsausführung) wird mit einem Interrupt realisiert. Daher kann er auch in ROM-Bereichen durchgeführt werden. Es ist jedoch darauf zu achten, daß Single Step in zeitkritischen Programmen (z.B. Motorsteuerung im CE 150) zu Defekten führen kann.

Für den Interrupt werden interne Register des PC-1500 Input/Output-Controllers und das Interrupt-Enable Flag IE (siehe Pocket Computer PC-1500 Technical Reference Manual) benutzt. Alle Befehle, die diese Register oder das IE berühren, können daher im Single Step nicht ausgetestet werden.

RUN mit oder ohne Breakpoint

Nach der Taste R oder + (RUN+I) bzw. SHIFT R oder - (RUN-I) kann ein auszustestendes Programm ohne oder mit einem Breakpoint gestartet werden. Das Programm läuft bis zum Breakpoint in Echtzeit ab und stoppt dann.

Ein Breakpoint darf nur auf den Beginn eines Befehls und nicht auf ein weiteres Byte eines Mehrbyte-Befehls gesetzt werden. Die Eingabe/Anzeige 0000 (CL-Taste) bedeutet kein Breakpoint.

Auf den Breakpoint wird ein SJP-Befehl geschrieben und so der Rücksprung in den Monitor bewirkt. Für diesen SJP-Befehl werden drei Speicherzellen benötigt. D.h., das zu testende Programm darf nicht den nächsten oder übernäch-

III.2 BESCHREIBUNG DES PROGRAMMES, RUN-MODE

Vorsichtsmaßnahmen

Das Aussetzen zeitkritischer Programme mit Single Step oder Breakpoint ist mit Bedacht auszuführen, da dabei Bauteile (z.B. durch zu lange Strombelastung) zerstört werden können.

Um Fehlreaktionen dieses Programmes zu vermeiden, sollten Single Step und Breakpoint im Programm selbst nicht durchgeführt werden.

Register- und Programmcounter/Befehlswort-Anzeige

Die Taste C bewirkt den Wechsel von der Programmcounter/Befehlswort-Anzeige zur Register-Anzeige und umgekehrt. Die Register werden in der Reihenfolge Accumulator A, X-, Y-, U-Register, Flags T und Stackpointer angezeigt. Nach den Tasten A, X, Y, U, T (oder F), S können diese geändert werden. Der Programmcounter kann nach P geändert werden. Bei der Eingabe eines neuen Programmcounters werden gleichzeitig die Register und der Stackpointer initialisiert. Befehlswoorte können nur im PRO-Mode eingegeben werden.

Stackpointer - Ende eines zu testenden Programmes

Beim Starten des Assembler-Monitors und bei der Eingabe eines Programmcounters werden der Stackpointer und die Register initialisiert. D.h. die Register werden Null und der Stackpointer wird 7APD gesetzt. Dahinter wird eine Rückkehradresse in den Monitor geschrieben, so daß bei einem mit RUN oder mit Single Step gestarteten Programm der erste freie Return-Befehl (d.h. nicht durch SJP oder PSH kompensiert) die Rückkehr zum Monitorkontrollen bewirkt. Ein zu testendes Programm sollte daher als Unterprogramm geschrieben und mit einem RTN-Befehl abgeschlossen werden. Nach dem freien RTN kehrt der

III-8

III-9



CE-140P

Tagespreis bitte anfragen!

LISTE DER BEFEHLE

ANHANG I

ADC XL	BCR + 12	CPI YH 12
ADC YL	BCR - 12	CPI YH 12
ADC UL	BHS - 12	OCA X
ADC XH	BHS - 12	OCA X
ADC YH	BHR + 12	OCA Y
ADC UH	BHR - 12	OCA Y
ADC X	B11 A 12	OCA #X
ADC Y	B11 X 12	OCA #Y
ADC U	B11 Y 12	OCA #U
ADC 0123	B11 U 12	DCS X
ADC #X	B11 1234 56	DCS Y
ADC #X	B11 #X 12	DCS U
ADC #Y	B11 #Y 12	DCS #X
ADC #U	B11 #U 12	DCS #Y
ADC #4567	B11 #1234 55	DCS #U
AD1 A 89	B11 X	DEC A
AD1 X A8	B11 Y	DEC XL
AD1 Y CD	B11 U	DEC YL
AD1 U EF	B11 1234	DEC UL
AD1 0123 45	B11 #X	DEC XH
AD1 #X 67	B11 #Y	DEC YH
AD1 #Y 89	B11 #U	DEC UH
AD1 #U A8	B11 #1234	DEC X
AD1 #CDEF 01	BUS + 12	DEC Y
ADR X	BUS - 12	DEC U
ADR Y	BUR + 12	DRL X
ADR U	BUR - 12	DRL #X
AEX	BZS + 12	DRR X
AND X	BZS - 12	DRR #X
AND Y	BZR + 12	EAL 12
AND U	BZR - 12	EOR X
AND 1234	CDU	EOR Y
AND #X	CIN	EOR U
AND #Y	CPA XL	EOR 1234
AND #U	CPA YL	EOR #X
AND #1234	CPA UH	EOR #Y
AN1 A 12	CPA XH	EOR #U
AN1 X 12	CPA YH	HLT #1234
AN1 Y 12	CPA UH	HLT
AN1 U 12	CPA X	INC A
AN1 1234 56	CPA Y	INC XL
AN1 #X 12	CPA U	INC YL
AN1 #Y 12	CPA 1234	INC UL
AN1 #U 12	CPA #X	INC XH
AN1 #1234 56	CPA #Y	INC YH
AN0	CPA #U	INC UH
AT1	CPA #1234	INC X
ATP	CPI A 12	INC Y
ATT	CPI XL 12	INC U
BCH + 12	CPI YL 12	ITA
BCH - 12	CPI UL 12	JMP 1234
BCS + 12	CPI XH 12	LDA XL
BCS - 12		LDA YL

—

sten Speicherplatz nach der Breakpointadresse berühren, da diese verändert wurden. Nach der Rückkehr zur Monitor-kontrolle werden an dem Breakpoint wieder die ursprünglichen Befehle eingesetzt.

Für das Setzen des Breakpoints gelten folgende weitere Einschränkungen. Diese werden jedoch vom Monitorprogramm überprüft und bei Verletzung wird der RUN-Befehl nicht ausgeführt. Es erfolgt Abbruch mit BEEP, BEEP.

1) Der Breakpoint darf nicht in einem schreibgesperrten Speicherbereich liegen.

22) Ein oder zwei Plätze vor den Startpunkt (momentaner Programmcounter) darf kein Breakpoint gesetzt werden.

Bei der Taste R oder + wird RUN+I (Start mit Interrupt-Initialisierung) angezeigt. Hierbei wird der mit der Taste BREAK verbundene PB7-Interrupt initialisiert, so daß nach der BREAK-Taste die Kontrolle an den Monitor zurückgeht, es sei denn, das auszutestende Programm hat diese Interruptmöglichkeit rückgängig gemacht.

Bei der Taste SHIFT R oder - wird RUN-I (Start ohne Interruptinitialisierung) angezeigt. Gerät hierbei das auszustehende Programm außer Kontrolle, so bleibt nur noch die RESET-Taste. Der RUN-I gibt jedoch die Möglichkeit, sämtliche Befehle, also auch den PC-1500 Input/Output Controller und das IE berührende, mit einem Breakpoint auszustesten.

Die Taste SHIFT P initialisiert Register und Stackpointer mit dem zuletzt eingegebenen Programmcounter. Dies vereinfacht die Testprozedur, da hierbei meist wiederholt bei der gleichen Adresse gestartet wird.



III-10

CE-140P

1 7-Color Capability

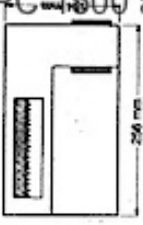
Color is one of the easiest ways to emphasize important graphic data in charts, diagrams, etc. The CE-140P provides seven colors (black, magenta, cyan, yellow, violet, green, red) to make any desired graphic information stand out for easy interpretation and understanding.

2 Uses Normal Paper

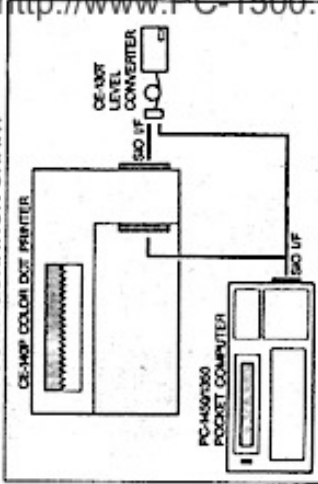
The CE-140P uses easily available, 114mm-wide normal paper. Colors come out bright and clear wherever your computer application may take you.

3 Compact Size

The CE-140P is a compact, magazine-size unit, easily portable from the laboratory to the office or classroom, or even outdoors for graphics data output while in the field. The CE-140P can also be integrated with Sharp's PC-1350 and PC-1450 Pocket Computers.



• SYSTEM CONFIGURATION CHART



*Some options may not be available in some countries.

SHARP

SHARP CORPORATION OSAKA, JAPAN
CABLE ADDRESS: LADOCET GRAMA
TELEX: LADOCET A88 JAPN

Anhang 2

Die VEJ-Befehle im Basic-Interpreter

Die VEJ-Befehle werden hier ohne die beiden Buchstaben EJ geschrieben, also VCC statt VEJ CO, usw. Die VEJ-Befehle sind für den Mikroprozessor Einbyte-Befehle. Jedoch im Zusammenhang mit dem Basic-Interpreter werden einige als 2-, 3- oder 4-Bytebefehle interpretiert. Diese werden, soweit bekannt, in diesem Programm entsprechend behandelt (siehe Liste der Befehle), d.h. sie verlangen bei der Eingabe stets auch die Angabe von Immediate. Die Funktion der für die Programmierung in Maschinensprache allgemein wichtigen Routinen VCA, VCC, VF4 und VF6 ist im folgenden beschrieben. Bezüglich der anderen sei auf die einschlägige Literatur verwiesen.

VCA,i : Rücksprung auf die auf i folgende Adresse.

Bringt Inhalt von X nach 78i, 78i+1.

Zerstört: Z, A, U.

VCC,i : Rücksprung auf die auf i folgende Adresse.

Holt Inhalt von 78i, 78i+1 nach X.

Zerstört: Z, A.

VF4,i,j : Rücksprung auf die auf j folgende Adresse.

Holt Inhalt von ij, ij+1 nach U.

Zerstört: Z, A, X.

VF6,i,j : Rücksprung auf die auf j folgende Adresse.

Bringt Inhalt von U nach ij, ij+1.

Zerstört: Z, A, X.



SOFTWARE
für die Entwicklung von
Maschinenprogrammen
in SHARP - Maschinensprache

SHARP - PC-1500/A
POCKET-COMPUTER

A2-1

LDA XL	STA Y
LDA YH	STA U
LDA YL	STA 1234
LDA XH	STA #X
LDA X	STA #Y
LDA Y	STA #U
LDA U	STA #1234
LDA 1234	STX X
LDA #X	STX Y
LDA #Y	STX U
LDA #U	STX S
LDA A 12	STX P
LDA XL 12	TIN
LDA YL 12	TIA
LDA XH 12	UCS 12
LDA YH 12	UCR 12
LDA YU 12	UC0 1234 56
LDA YU 12	UC4
LDA S 1234	UC5
LDE X	UC8 12
LDE Y	UCA 12
LDE U	UCC 12
LDE X	UCC 1234
LDX Y	UC0 1234
LDX U	UD2 1234
LDX S	UD4 12
LDX P	UD6 12
LIN X	UD8
LIN Y	UDA
LIN U	UDC
LOP 12	UDE X
NGP	UE0 Y
OFF	UE2 U
ORA X	UE4
ORA Y	UE6
ORA U	UE8
ORA 1234	UEA
ORA #X	UEC
ORA #Y	UEE
ORA #U	UE0
ORA #1234	UE2
ORI A 12	UF4 1234
ORI X 12	UF6 1234
ORI Y 12	UHJ 12
ORI U 12	UUS 12
ORI 1234 56	UZS 12
ORI #X 12	UZR 12
ORI #Y 12	UHR 12
ORI #U 12	UHS 12
ORI #1234 56	
POP A	
	POP X
	POP Y
	POP U
	PSH A
	PSH X
	PSH Y
	PSH U
	RDP
	REC
	RIE
	ROL
	ROR
	RPU
	RPU
	RTN
	RTN
	SBC XL
	SBC YL
	SBC YH
	SBC XH
	SBC YH
	SBC UH
	SBC X
	SBC Y
	SBC U
	SBC 1234
	SBC #X
	SBC #Y
	SBC #U
	SBC #1234
	SBI 12
	SDE X
	SDE Y
	SDE U
	SOP
	SEC
	SHL
	SHR
	SIE
	SIN X
	SIN Y
	SIN U
	SJP 1234
	SPU
	STA XL
	STA YL
	STA YH
	STA XH
	STA YU
	STA YH
	STA UH
	STA X

A1-2

Anhang 3

Das Wichtigste zur Speichereinteilung

Beim PC-1500 mit dem Speichermodul CE-155 (8K) stehen dem Benutzer ca. 10K RAM zur Verfügung. Mit dem Kommando NEWADRESSE wird die Adresse definiert, von der ab das Basicprogramm in den Speicher eingeschrieben wird. Bei NEWO wird die erste zur Verfügung stehende Adresse gewählt. Die Anfangs- und Endadresse des vom Basicprogramm belegten Speicherbereiches wird vom System in den Speicherplätzen &7865-68 verwaltet. Nach NEWO ergibt sich mit dem PEEK-Befehl, daß in diesen Speicherplätzen &38, &05, &38, &C5 steht. Der dem Benutzer zur Verfügung stehende RAM-Bereich beginnt also bei der oben genannten Gerätekombination bei &38C5. Nach NEW&4CC5 stehen die ersten 5K (von &38C5 bis &4CC4) für Maschinenprogramme und der Rest für Basicprogramme zur Verfügung.

Anhang 4

Erweiterungsmöglichkeiten des PRO-Mode

Das Programm LASMO 124 ist so aufgebaut, daß der PRO-Mode das Programmende mit einem Return-Befehl berührt. Dies gibt dem Benutzer die Möglichkeit, Erweiterungen nach eigenem Bedarf anzufügen, indem statt dieses letzten RTN weitere Tastenabfragen und zugehörige Funktionen angehängt werden. In der Erweiterung müssen zwei RTN auftreten. Eins bewirkt den Rücksprung nach durchgeführter Funktion, das andere, das am Ende der Erweiterung stehen muß, sorgt für den Rücksprung falls keine der in der Erweiterung abgefragten Tasten betätigt worden war. Dadurch können an eine Erweiterung weitere angehängt werden.

Bei der Ankunft an das Programmende befindet sich der Wert der Tasteneingabe in UH und die Rücksprungsadresse im Stack. Die momentane Speicheradresse wird in &777C/D gehalten.

Wird sie verändert, so sollte man vor der Rückkehr eine Null nach &7650 schreiben. Dadurch wird der Speicher für die Rückwärtsanzeige blockiert. Die Programmweiterung muß außerhalb des Programmes eingegeben und ausgetestet werden. Erst abschließend kann sie mit COPY auf das letzte RTN kopiert werden.

Beim Programm LASMO 124E sind die im folgenden angegebenen drei Erweiterungen bereits vorhanden. Am Ende befindet sich die Macro-Erweiterung. Diese läßt das Anfügen weiterer Erweiterungen nicht zu, da der Platz hinter der Macroerweiterung für das Macro-Lexikon vorgesehen ist.

Will man also LASMO 124E erweitern, so muß zuerst die Macro-Erweiterung abgehängt werden. Nach dem Anfügen eigener Erweiterungen kann die Macro-Erweiterung wieder an letzter Stelle angehängt werden. Dazu geht man folgendermaßen vor. LASMO 124E steht z.B. auf &2000 bis &37E7. Die Macro-Erweiterung beginnt dann auf &35C3 mit CPI UH 2A. Diese Macro-Erweiterung, &35C3 bis &37E7 (Prüfsumme = FF80), läßt man auf Band. Danach überschreibt man &35C3 mit dem Befehl RTN. Das Programm, &2000 bis &35C3 (Prüfsumme 6C69), ist nach den vorher gegebenen Regeln erweiterbar. Anschließend kann die Macro-Erweiterung wieder auf das letzte RTN aufkopiert werden.

Im folgenden sind nun die drei Erweiterungsvorschläge mit ausführlicher Funktionsbeschreibung und den Programm Listings angegeben. In LASMO 124E sind sie bereits vorhanden.

Anhang 4.1: CSIZE-Erweiterung

Belegung von SHIFT 1 mit CSIZE 1 und SHIFT 2 mit CSIZE 2. Listing: LDI A 01, CPI UH 21, B2S + 05, INC A, CPI UH 22, B2R + 04, STA 79F4, RTN, RTN.



A4-2

Specifications

Printing method	Ink dot printer Method
Print columns	Max. 80 columns
Paper width	114mm, normal paper
Character size	0.3 types (0.8mm x 1.2mm to 5.0mm x 7.6mm)
Print colors	4 (black, magenta, cyan, yellow)
Print colors	7 (black, magenta, cyan, yellow, violet, green, red)
Print directions	Up, down, left, right (effective drawing range: 96mm right-left, 200mm up-down)
Min. step size	0.2mm
Power supply	Built-in rechargeable Ni-Cd battery
Dimensions	250(W) x 184(D) x 43(H) mm
Accessories	1 roll paper 1 ink cartridge set (outlin) (4 colors: black, magenta, cyan, yellow) 1 paper roll bearing 1 AC adaptor, etc.
Interface	Built-in serial IO
Indicator	Low-battery indicator
Supplies sold separately	Paper roll (EA-615P) Ink cartridge (EA-600)

*Sharp reserves the right to make design and specification changes for product improvement.

Bitte Tagespreis
anfragen.



GmbH
Kaiser - Friedrich - Strasse 54
1000 Berlin 12

durch Information!

**Erzielen Sie Vorteile mit klaren, übersichtlichen Berichten.
Verdoppeln Sie das Vergnügen am Illustrieren.
Legen Sie sich die Besonderheit des Druckens in vier
Farben zu.**



4170: LDI XH 77, 2
: LDI XL 6E, 4: UF
4 7780, 7: LDA U,
8: BII 77AA 01, C:
8ZS + 02, E: LDA
4U,
4180: INC U, J: PS
H U, 3: STA UL, 4:
INC A, 5: DEC A,
6: BZL + 02, 8: L0
J A 20, A: SIN X,
8: PSH X, D: UCC C
0, F: SJP FB90,
4192: POP X, 4: PD
P U, 6: CPI XL 72,
8: BCR - 23, A: L
DI A 08, C: STA X,
D: LDI A 20, F: S
TA Y,
41A0: UF6 7780, 3:
LDI UH 77, 5: LDI
UL 60, 7: LDI XL
1A, 9: SJP ED38, C
: LDA 79E6, F: BZS
+ 07,
41B1: UCC 04, 3: L
DI UH 02, 5: SJP D
F37, 8: LDI YL 60,
A: UCC 04, C: LDI
UH 00, E: SJP E51
0,
41C1: BCS - DC, 3:
UF4 7752, 6: LDX
U, 8: UF4 7780, 8:
LDA XH, C: CPA UH
D: BZL + 02, F:
LDA XL,
41C0: CPA UL, 1: B
CS - 78, 3: UCC 04
S: LDI UH 03, 7:
SJP DF37, A: BCH
- FS, C: RTN,

40E1: DEC U, 2: DE
C U, 3: INC U, 4:
UF6 77AE, 7: BCH -
09, 9: CPI A 40,
8: BZL + 14, 0: L0
J YL 60, F: BII 77
AA 01,
40F3: BZS + 01, 5:
INC YL, 6: UCC 00
8: SJP DD26, 8:
BCS - ED, D: BCH -
1B, F: BCH + 08,
4101: CPI A 20, 3:
BZL + 06, 5: ORI
77AA 00, 9: BCH -
FB, 8: SJP ED70, E
: BCR + 29,
4110: LDA YL, 1: L
DI YL 6A, 3: SIN Y
4: LDI A 3C, 6:
SIN Y, 7: LDI UL 0
8, 9: BII 77AA 00,
D: BZL + 03, F: S
IN Y,
4120: LDI UL 01, 2
: LDI UH 02, 4: UF
6 7758, 7: LDI YH
01, 9: LDI YL 6A,
8: UCC 00, D: LDI
A 25, F: SJP DD84,
4132: BCS - B1, 4:
STA YL, 5: UF4 77
AE, 8: LDA YL, 9:
BCH - 03, 8: CPI A
11, D: BZL - 8C,
F: LDI UH 04,
4141: LDI UL 02, 3
: UCC 05, 5: SJP D
F38, 8: INC X, 9:
LIN Y, A: BII X 50
C: BCS - CB, E:
UF6 7780,
4151: UCC 00, 3: L
DI A 2C, 5: SJP DD
04, 8: UF4 7780, B
: LDI YH 77, D: L0
I YL 60, F: UCC 00
4161: INC X, 2: SJ
P CF8D, 5: LDI A 3
A, 7: BII 77AA 01,
B: BZS + 02, D: L
DI A 23, F: SIN Y,

4041: LDA U, 2: B1
I 77AA 01, 6: BZS
+ 02, 8: LDA U, A
: LDI YL 6F, C: ST
A Y, D: LDI YL 69,
F: STA UL,
4050: UCC 00, 2: S
JP FB90, 5: BII 77
AA 10, 9: BZS + 07
8: LDI YL 72, D:
LOI A 22, F: SDE
Y,
4060: SDE Y, 1: SD
E Y, 2: LDI UH 77,
4: LDI UL 60, 6:
LOI XL 1A, 8: SJP
ED38, B: UF4 77A9,
E: DEC U, F: LDA
UH,
4070: BZL - 04, 2:
LDA 77AD, 5: BZS
+ 0E, 7: UF4 77AE,
A: INC U, B: UF6
77AE, E: LDI A 01,
4080: STA 77A9, 3:
BCH - 75, 5: LDI
A 20, 7: STA 77A9,
A: BII 77AA 10, E
: BZL + 10,
4090: SJP E42C, 3:
CPI A 0A, 5: BZS
+ 0C, 7: CPI A 08,
9: BZS + 00, B: L
DI A 40, D: STA 77
A9,
40A0: SJP E242, 3:
ANI H008 FD, 8:
CPI A 0E, A: BZL +
01, C: RTN, D: ST
A YL, E: UF4 77AE,
40B1: LDA YL, 2: B
II 77AA 10, 6: BZS
+ 17, 8: BII 77AA
01, C: BZS + 04,
E: STA U,
40C0: BCH + 01, 2:
STA U, 3: LDI A 3
0, 5: STA 77A9, 8:
LOI A 01, A: STA
77AD, D: BCH - BA,
F: CPI YL 12,
40D1: BZL + 06, 3:
ORI 77AA 10, 7: B
CH - 5B, 9: CPI YL
0A, 8: BZS + 06,
D: CPI YL 0B, F: B
ZL + 0B,

Anhang 4.2: Erweiterung für die Anzeige, Eingabe und den Ausdruck von Hexcode und Zeichen

Nach der Taste & (*) wird eine Speichersadresse aus MEO (ME1) und der zugehörige Speicherinhalt als Hexcode und als Zeichen angezeigt. Ganz rechts steht &"/, was darauf hinweist, daß sowohl Hexcode als auch Zeichen eingegeben werden können.

Nach der Taste M kann die Speicheradresse geändert werden. Nach ↓ (↑) erfolgt Anzeige der nächsten (letzten) Adresse. Bei Festhalten der Tasten: Schnellauf.

Die Taste BREAK führt zurück zur Befehlseingabe.

Die Hexcode-Eingabe wird durch eine Hextaste eingeleitet.

Danach ist nur eine weitere Hextaste, CL, BREAK oder ENTER möglich. Bei << muß die Eingabe mit ENTER abgeschlossen werden.

Mit der Taste < (ohne SHIFT) kann zur schnellen Hexeingabe mit <, d.h. ohne ENTER, umgeschaltet werden.

Die Eingabe von Zeichen erfolgt nach der Taste ". Nach dieser Taste werden alle weiteren Tasten (oder Vortaste, Taste) als Zeichen eingegeben. Nur die Taste BREAK führt zurück zur Befehlseingabe.

Editieren ist über das Einfügen oder Herausnehmen einer entsprechenden Zahl von Byte im Befehlseingabemodus möglich.

Nach der Taste ! werden je vier Hexcode und Zeichen in einer Zeile mit CSIZE 1 oder 2 ausgedruckt. Der auszudruckende Speicherbereich wird abgefragt. Ist der Printschalter auf ON, so wird die Anfangs- und Endadresse protokolliert.

ERWEITERUNG FUER
ANZEIGE, EINGABE
UND AUSDRUCK VON
HEXCODE + ZEICHEN
CHECKSUM 4000-410C
= E031
CHECKSUM 4000-407F
= 3C22
CHECKSUM 4000-40FF
= 7977
CHECKSUM 4000-417F
= B7F1
4000: LDI UL 00, 2
: CPI UH 16, 4: BZ
S + 05, 6: INC UL,
7: CPI UH 13, 9:
BZL + F4, 8: LDI U
H 40, D: UF6 77A9,
4010: LDI A 00, 2:
STA 77AD, 5: LDI
YH 77, 7: LDI YL 7
9, 9: LDI A 22, 8:
CF8D, B: LDI A 3A,
SDE Y, C: LDI A 2
F, E: SDE Y, F: L0
I A 26,

SHARP

CE-140P

Color Dot Printer

7-Color, Compact Dot-Matrix Printer Offers Versatile Graphics Capability on Normal Paper

Ist die Eingabe im Macro-Lexikon nicht vorhanden, so wird mit BEEP, BEEP abgebrochen. Anderenfalls wird das zugehörige Maschinenprogramm auf der in der Anzeige vorhandenen Speicheradresse eingeschrieben und anschließend der unmittelbar darauf folgende Speicherplatz angezeigt.

Listen der Macros: Taste !

Es werden die Macros in der Reihenfolge, in der sie im Lexikon stehen, auf dem Display angezeigt. Es wird der Name und der Speicherbereich, auf dem das zugehörige Maschinenprogramm steht, angegeben. Nach ENTER (oder einer Taste) wird zum nächsten Macro weitergeschaltet, nach ↓ durchlaufende Anzeige. Abbruch erfolgt mit der Taste BREAK. Falls der Print-Schalter eingeschaltet ist, wird ausgedruckt.

Löschen eines Macros im Lexikon: Taste DEL (ohne SHIFT) und Eingabe des zu löschenden Macro. Falls der eingegebene Name nicht im Lexikon vorhanden oder der fragliche Speicherbereich nicht beschreibbar ist, erfolgt Abbruch mit BEEP, BEEP.

Löschen der Macros ab einem Macro: Taste CL

und Eingabe des Macros von dem ab alle Macros im Lexikon gelöscht werden sollen. Falls das eingegebene Macro nicht im Lexikon vorhanden ist, oder der fragliche Speicherbereich nicht beschreibbar ist, erfolgt Abbruch mit BEEP, BEEP.

Löschen des gesamten Macro-Lexikon: Taste CA

und ENTER. Falls der fragliche Speicherbereich nicht beschreibbar ist, erfolgt Abbruch mit BEEP, BEEP.

Anhang 4.3: Macro-Erweiterung

Die Macro-Erweiterung zu LASMO 124 erlaubt Macros zu definieren, bei der Programmentwicklung aufzurufen, zu listen und zu löschen. Die Macro-Funktionen werden mit der Taste * aufgerufen, d.h. es darf keine Erweiterung gleichzeitig benutzt werden, die diese Taste verwendet. Hinter der Macro-Erweiterung muß genügend RAM-Speicherplatz zur Verfügung stehen, da das Macro-Lexikon an das Ende der Macro-Erweiterung geschrieben wird. Aus diesem Grund darf an die Macro-Erweiterung auch keine weitere Erweiterung angefügt werden.

Die Macro-Namen können eine Länge von 8 Zeichen haben. Leertasten (space) werden hierbei mitgezählt. Beim Ver gleich zwischen Macro-Aufruf und Macro-Lexikon bleiben Leertasten jedoch unberücksichtigt. Mit der Taste BREAK können Eingaben abgebrochen werden.

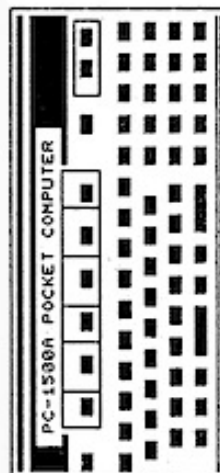
Aufruf der Macro-Funktionen: Taste *

Definition eines Macro: Taste :

Es wird der Name des Macro (bis 8 Zeichen) eingegeben und mit ENTER abgeschlossen. Falls der eingegebene Name bereits im Lexikon vorhanden ist, wird mit BEEP, BEEP abgebrochen. Nach der Namen-Eingabe wird der Speicherbereich abgefragt, auf dem das zugehörige Maschinenprogramm steht. Dieses Programm wird danach mit dem definierten Namen in das Macro-Lexikon aufgenommen. Beim Einschreiben in den Speicher wird überprüft, ob alles richtig eingeschrieben wurde. Wenn nicht, wird mit BEEP, BEEP abgebrochen. Bis zum letzten ENTER kann mit der Taste BREAK die Definition abgebrochen werden.

Einschreiben eines Macro in ein Maschinenprogramm:

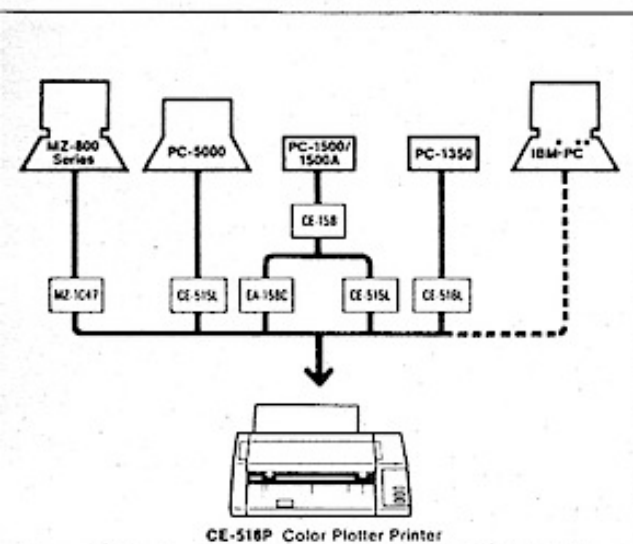
Nach dem Aufruf der Macroerweiterung mit der Taste * wird ohne weitere Vortaste der Macro-Name eingegeben und mit ENTER abgeschlossen. Beim Vergleich der Eingabe mit dem Macro-Lexikon bleiben Leertasten (space) unberücksichtigt.



SOFTWARE

für die Entwicklung von
Maschinenprogrammen
in SHARP - Maschinensprache

Examples of System Expansion



Als Beigabe übersende ich Ihnen noch ein kleines Spielprogramm, das in die Kategorie der Reaktionsspiele eingeordnet werden kann. Die Regeln sind leicht erklärt: Über den Funktionstasten schichten sich kleine Türmchen auf. Je höher ein Turm hochklettert, desto höher ist die Punktzahl dafür (1.Schicht=1 Punkt, ..., 6.Schicht=6 Punkte). Der Turm darf jedoch nicht die Deckenmarkierung berühren. Durch Drücken der jeweiligen Funktionstaste verschwindet der darüberstehende Turm; hierfür gibt es dann aber auch zwei Punkte Abzug. Das Spiel ist bei Berührung der Deckenmarkierung bzw. nach ungefähr einer Minute beendet. STATUS 1 beträgt 601.

```

5: CLEAR : WAIT 0
10: DIM C$(0)*22, A
    (6), Z(6)
15: A(1)=9: A(2)=32
    : A(3)=56: A(4)=
    80: A(5)=104: A(
    6)=127
20: T=TIME : TIME =
    0
25: FOR I=0 TO 155
30: GCURSOR I:
    GPRINT 1
35: NEXT I
40: IF TIME >.01
    GOTO 110
45: P=RND 6
50: Z(P)=Z(P)+1
55: H=H+Z(P)
60: RESTORE (120+Z
    (P))

65: READ C$(0)
70: GCURSOR A(P)
75: GPRINT C$(0)
80: IF Z(P)>66 GOTO
    110
85: A=ASC INKEY$
90: IF A>22 OR A<17
    GOTO 40
95: A=A-16: Z(A)=0:
    Y=Y+2
100: GCURSOR A(A):
    GPRINT "010101
    010101010101
    01"
105: GOTO 40
110: BEEP 5: TIME =
    TIME + T
115: WAIT 99: PRINT
    "Punkte : "; H
    -Y)
121: DATA "41414141
    4141414141414141
    122: DATA "61616161
    61616161616161
    "
123: DATA "71717171
    71717171717171
    "
124: DATA "79797979
    79797979797979
    "
125: DATA "70707070
    70707070707070
    "
126: DATA "7F7F7F7F
    7F7F7F7F7F7F7F
    "
127: DATA "7F707971
    6141617179707F
    "

Dominik A. Müller
Höhnerkamp 30
2000 Hamburg 56
    
```

```

45E2: DEC U, 3: UC
    C 0, 5: INC X, 6:
    SJP 0023, 9: UCC
    E4, 8: SJP F890, E
    : B11 W82E 20,
    45F3: B25 + 14, 5:
    LDA A808, 8: CPI
    A 44, A: B25 + 24,
    C: CPI A BE, E: B
    ZR + 89,
    4508: UCC 0, 2: L
    DJ A 2C, 4: SJP 00
    84, 7: BCH - 5F, 9
    : SJP E42C, C: CPI
    A 0A, E: BZR + 89
    ,
    4610: LDJ XH 40, 2
    : DEC X, 3: LDA XH
    : 4: BZR - 24, 6:
    BCH - 6E, 8: SJP E
    242, 8: CPI A 0A,
    D: B25 - 0F, F: CP
    1 A 8E,
    4621: BZR - 79, 3:
    RTN, 4: FF,
    
```

```

4560: REC, 1: ADC
    XL, 2: STA XL, 3:
    LDA XH, 4: ADC YH,
    5: STA XH, 6: PSH
    X, 8: UCC F9, A:
    PSH X, C: LDJ UH F
    F, E: LDJ UL 08,
    4570: UFF 7768, 3:
    UCC E6, 5: SJP CA
    F4, 8: POP U, A: P
    OP Y, C: SEC, D: L
    DA XL, E: SBC YL,
    F: STA XL,
    4580: LDA XH, 1: S
    BC YH, 2: STA XH,
    3: LIN Y, 4: STA U
    : 5: PSH X, 7: STA
    XL, 8: LIN U, 9:
    CPA XL, A: POP X,
    C: BZR - 01, E: DE
    C X, F: LDA XL,
    4590: BZR - 0F, 2:
    LDA XH, 3: BZR -
    12, 5: RTN, 6: STA
    YL, 7: UFF 777C,
    A: LIN X, B: SIN U
    , C: DEC Y, D: LDA
    XL, E: BZR - 06,
    45A0: LDA YH, 1: B
    ZR - 09, 3: STA 76
    50, 6: UFF 777C, 9
    : RTN, A: UCC F9,
    C: LDJ YL 79, E: L
    DJ A 00,
    45B0: SDE Y, 1: CP
    I YL 61, 3: BCS -
    05, 5: LDA X, 6: J
    NC A, 7: BZR + 01,
    9: RTN, A: LIN X,
    B: B25 + 03, D: S
    IN Y, E: BCH - 06,
    45C0: LDJ A 20, 2:
    SIN Y, 3: CPI YL
    69, 5: BCR - 07, 7
    : LIN X, 8: STA UH
    , 9: LIN X, A: PSH
    X, C: REC, D: ADC
    XL, E: STA XL, F:
    LDA UH,
    45D0: ADC XH, 1: S
    TA XH, 2: UCA F9,
    4: POP U, 6: UCC C
    0, 8: INC X, 9: SJ
    P 00B3, C: LDJ A 2
    D, E: SIN Y, F: UF
    4 78F9,
    
```

```

44F1: INC YL, 2: B
    ZR + 5C, 4: CPI A
    4E, 6: BZR - 1B, 8
    : DEC X, 9: LDJ YL
    68, 8: LIN Y, C:
    STA X, D: STA UL,
    E: LIN X, F: CPA U
    L,
    4500: BZR + 13, 2:
    LDA UL, 3: BZR -
    0A, 5: LDJ UL 02,
    7: UCC D6, 9: LDJ
    A 05, B: SJP FB91,
    E: INC YL, F: DEC
    YL,
    4510: ADJ X 47, 2:
    DEC YL, 3: BCR +
    07, 5: UCC F9, 7:
    LDJ A FF, 9: STA X
    : A: BCH - 3A, C:
    UFF 78F9, F: LIN U
    ,
    4520: BZR - 03, 2:
    SEC, 3: LDA YL, 4
    : SBC XL, 5: STA Y
    L, 6: LDA YH, 7: S
    BC XH, 8: STA YH,
    9: INC Y, A: LDA Y
    N, B: STA U, C: LJ
    N U, D: CPA YH, E:
    BZR - 18, 7: SEC,
    4530: LDA YL, 1: S
    TA U, 2: LIN U, 3:
    CPA YL, 4: BZR -
    21, 6: LIN X, 7: S
    TA U, 8: PSH Y, A:
    STA YL, B: LIN U,
    C: CPA YL, D: POP
    Y, F: BZR - 2C,
    4541: DEC Y, 2: LD
    A YL, 3: BZR - 0F,
    5: LDA YH, 6: BZR
    - 12, 8: LDJ A FF
    , A: STA U, B: LDA
    U, C: INC A, D: B
    ZR - 3A, F: RTN,
    4550: CPI A 4E, 2:
    B25 - 77, 4: CPI
    A 4F, 6: B25 - A6,
    8: STA UL, 9: LIN
    X, A: STA YH, B:
    LIN X, C: CPI UL 5
    4, E: BZR + 36,
    
```

```

4460: LDJ A 02, 2:
    ADC XH, 3: STA XH
    , 4: UCA F9, 6: UC
    C E4, 8: SJP FB90,
    B: LDJ YL 68, D:
    SJP E242,
    4470: CPI A 3A, 2:
    BZR + 0A, 4: LOX
    P, 6: BCH + 49, 8:
    INC X, 9: LIN X,
    A: DEC X, B: ANI X
    4E, D: LIN X, E:
    CPI A 0B,
    4480: BZR + 0A, 2:
    LDJ P, 4: BCH + 3
    B, 6: INC X, 7: LJ
    N X, 8: CPI XH 45,
    A: INC Y, B: LIN
    X, C: CPI A 19, E:
    BZR + 0A,
    4490: LOX P, 2: BC
    N + 20, 4: SDE X,
    5: CPI XH 46, 7: 0
    EC YL, 8: ADJ X 40
    : A: CPI A 1A, C:
    BZR + 1C, E: LDJ A
    43,
    44A0: SIN Y, 1: LD
    I A 41, 3: STA Y,
    4: UCC E4, 6: SJP
    FB90, 9: SJP E242,
    C: CPI A 0D, E: B
    ZR + 20,
    44B0: UCC F9, 2: L
    DJ A FF, 4: STA XH,
    5: LDA X, 6: INC
    A, 7: BZR + 24, 9:
    RTN, A: CPI A 11,
    LDA UH, C: SDE Y,
    D: LDJ A 20, F: S
    IN Y, F: BCH + 0B,
    44C1: LDJ YL 68, 3
    : INC X, 4: INC X,
    5: LDJ UL 05, 7:
    TIN, 8: LOP 03, A:
    LDJ YL 68, C: UCC
    E4, E: SJP FB90,
    44D1: SJP E242, 4:
    CPI A 0D, 6: B25
    + 10, 8: SIN Y, 9:
    CPI YL 71, B: BCR
    - 11, D: ANI #F00
    B FD,
    44E2: UCC E0, 4: S
    JP FB90, 7: RTN, 8
    : UCC E6, A: SJP F
    B90, D: STA YL, E:
    LDA 7764,
    
```

```

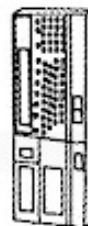
MACRO-ERWEITERUNG
CHECKSUM 4400-4624
= FF00
CHECKSUM 4400-447F
= 302C
CHECKSUM 4400-44FF
= 27A5
CHECKSUM 4400-457F
= B267

4400: CPI UH 2A, 2
: B25 + 33, 4: LDJ
UH 77, 6: LDJ UL
60, 8: LDJ XL 1A,
A: SJP E03B, D: RT
N, E: UCC F9,
4410: LDJ YL 68, 2
: LIN Y, 3: CPI A
20, 5: B25 - 05, 7
: STA UL, 8: LIN X
: 9: BZR + 01, B:
RTN, C: CPI A FF,
E: B25 - 05,
4420: CPI A 20, 2:
B25 - 0C, 4: CPA
UL, 5: B25 - 15, 7
: LIN X, 8: BZR -
03, A: LIN X, B: S
TA UL, C: LIN X, D
: REC, E: ADC XL,
F: STA XL,
4430: LDA XH, 1: A
DC UL, 2: STA XH,
3: UCA F9, 5: BCH
- 27, 7: LDJ YH 77
: 8: LDJ YL 67, B:
LDA UH, C: SDE Y,
D: LDJ A 20, F: S
IN Y,
4440: INC Y, 1: LD
I A 20, 3: SIN Y,
4: CPI YL 2A, 6: B
CR - 05, 8: LOX P,
A: LDJ A 0A, C: R
EC, D: ADC XL, E:
STA XL, F: LDJ A F
F,
4451: ADC XH, 2: S
TA XH, 3: UCA E4,
5: LDJ A 0A, 7: AD
R X, 9: UCA E6, B:
LDJ A 16, D: REC,
E: ADC XL, F: STA
XL,
    
```

PC-1500

verbessertes

HARDCOPY



```

10 : DIM A$(0)*26
20 : "H" INPUT "COLOR (0..3)=";C:COLOR C
30 : PAUSE "Bitte Text eingeben!"
40 : INPUT A$(0);WAIT 0:PRINT A$(0)
50 : GRAPH FOR S=0 TO 155:A=64:FOR I=0 TO 6
60 : IF POINT(SAND A6,CURSOR(1)*10,-S*10):GOSUB 80
70 : A=A/2:NEXT I:NEXT S:WAIT 1:TEXT:END
80 : RLINE -(0,9)-(1,0)-(0,-9)-(1,0)-(0,-9)
90 : RLINE -(0,9)-(1,0)-(0,-9)-(1,0)-(0,-9)
100 : RLINE -(1,0)-(0,9)-(1,0)-(0,-9):RETURN
110 : RLINE -(1,0)-(0,9)-(1,0)-(0,-9)

```

```

1:REM "Eishockey
Super-Statist
ik"

```

```

2:REM vom SHARP
-Basic-Team
10:"A" CLEAR:DIM
A$(30)*10,A(30
),3),B(30,3),C(
30,3),D(30,3),
S(30),K$(3)
20:PAUSE "Namen e
ingeben!"
30:WAIT 0
40:FOR I=1 TO 30:
PRINT "N";I;
":":INPUT A$(
I)
50:IF A$(I)=""CLS
:NEXT I:WAIT:
END
60:CLS:NEXT I
70:WAIT:END
80:"C"PRINT "Reco
rder bereit, E
NTER":CLEAR
90:DIM A$(30)*10,
A(30,3),B(30,3
),C(30,3),D(30
),3),S(30),K$(3
)
100:INPUT A$(X),B(
X),C(X),D(X),S
(X),A$(X)
110:PRINT "F E R T
I G !":END

```

PC-1500
EISHOCKEY
SUPER-STATISTIK

Ein schoenes Pro-
gramm fuer alle
Eishockey-Fans. Sie
koennen jetzt ihre
Lieblings-Genau
statistisch ueber-
wachen. Sie erhal-
ten einen komplet-
ten Ausdruck ueber
Skorer-Pkt., Tore,
Vorlagen u. Straf-
zeiten sowie ueber
die Anzahl der
Einsatze.



SHARP

Um die Lesbarkeit
bei HARDCOPY-
Drucken zu er-
hoechern, haben wir
das Programm von
M. STECK verbessert.

Programmstart mit
RUN, weitere Schritte
mit DEF H.

Utel Spass euer
SHARP-Basic-Team

```

120:"X"FOR I=1 TO 3
0:IF A$(I)=""
NEXT I:END
130:PRINT "N";I;
":":A$(I)
140:WAIT 0:PRINT "
eingesetzt (J/
N)":B$=INKEY$
150:IF B$="N"WAIT
:NEXT I
160:IF B$="J"WAIT
:S(I)=S(I)+1:
GOTO 180
170:GOTO 140
180:PAUSE "1. Dritt
el":INPUT "Tor
e=";T1
190:INPUT "Vorlese
n=";U1:INPUT "
Strafzeiten=";
S1
200:PAUSE "2. Dritt
el":INPUT "Tor
e=";T2
210:INPUT "Vorlese
n=";U2:INPUT "
Strafzeiten=";
S2
220:PAUSE "3. Dritt
el":INPUT "Tor
e=";T3
230:INPUT "Vorlese
n=";U3:INPUT "
Strafzeiten=";
S3
240:A(I,0)=A(I,0)+
T1+T2+T3+U1+U2
+U3
250:A(I,1)=A(I,1)+
T1+T2+T3
260:A(I,2)=A(I,2)+
U1+U2+U3
270:A(I,3)=A(I,3)+
S1+S2+S3
280:B(I,0)=B(I,0)+
T1+U1
290:B(I,1)=B(I,1)+
T1
300:B(I,2)=B(I,2)+
U1
310:B(I,3)=B(I,3)+
S1
320:C(I,0)=C(I,0)+
T2+U2
330:C(I,1)=C(I,1)+
T2
340:C(I,2)=C(I,2)+
U2
350:C(I,3)=C(I,3)+
S2
360:D(I,0)=D(I,0)+
T3+U3
370:D(I,1)=D(I,1)+
T3
380:D(I,2)=D(I,2)+
U3
390:D(I,3)=D(I,3)+
S3
400:T1=0:T2=0:T3=0
:U1=0:U2=0:U3=
0:S1=0:S2=0:S3
=0
410:NEXT I:WAIT:
END
420:"K"PAUSE "Reco
rder zur Aufna
hme":PRINT "be
reit, dann ENTE
R"
430:PRINT A$(X),B(
X),C(X),D(X),S
(X),A$(X)
440:PRINT "F E R T
I G !":END
450:"O"PAUSE "Ausd
ruck":COLOR 1:
CSIZE 1
460:LPRINT "NR";
TAB 3;"Name";
TAB 14;"Sp";
TAB 19;"S";TAB
24;"T";
461:LPRINT TAB 29;
"U";TAB 34;"St
r."
470:FOR I=1 TO 30:
IF A$(I)=""
NEXT I:END
480:IF LEN STR$ I=
1LET C$=" "+
STR$ I:GOTO 50
0
490:C$=STR$ I
500:FOR Y=0 TO 3:IF
LEN STR$ A(I,Y
)=1LET K$(Y)="
"+STR$ A(I,Y
):GOTO 530
510:IF LEN STR$ A(
I,Y)=2LET K$(Y
)=" "+STR$ A(
I,Y):GOTO 530
520:K$(Y)=STR$ A(
I,Y)
530:NEXT Y
540:S$=STR$ S(I)
550:IF LEN S$=1LET
S$=" "+S$
560:LF 1:COLOR 1
570:LPRINT C$;TAB
3;A$(I);TAB 14
;S$;TAB 19;K$(
0);TAB 23;K$(1
);
571:LPRINT TAB 28;
K$(2);TAB 33;K
$(3)
580:FOR Y=0 TO 3:IF
LEN STR$ B(I,Y
)=1LET K$(Y)="
"+STR$ B(I,Y
):GOTO 730
580:FOR Y=0 TO 3:IF
LEN STR$ D(I,Y
)=1LET K$(Y)="
"+STR$ D(I,Y
):GOTO 730
590:FOR Y=0 TO 3:IF
LEN STR$ C(I,Y
)=1LET K$(Y)="
"+STR$ C(I,Y
):GOTO 730
600:FOR Y=0 TO 3:IF
LEN STR$ K$(Y)=
1LET K$(Y)=STR$
K$(Y)
610:IF LEN STR$ D(
I,Y)=2LET K$(Y
)=" "+STR$ D(
I,Y):GOTO 730
620:K$(Y)=STR$ D(
I,Y)
630:FOR Y=0 TO 3:IF
LEN STR$ C(I,Y
)=1LET K$(Y)="
"+STR$ C(I,Y
):GOTO 730
640:FOR Y=0 TO 3:IF
LEN STR$ K$(Y)=
1LET K$(Y)=STR$
K$(Y)
650:IF LEN STR$ C(
I,Y)=2LET K$(Y
)=" "+STR$ C(
I,Y):GOTO 730
660:K$(Y)=STR$ C(
I,Y)
670:NEXT Y
680:COLOR 3
690:LPRINT TAB 3;"
2. Dritt el";TAB
2
18;K$(0);TAB 2
3;K$(1);
691:LPRINT TAB 28;
K$(2);TAB 33;K
$(3)
700:FOR Y=0 TO 3:IF
LEN STR$ D(I,Y
)=1LET K$(Y)="
"+STR$ D(I,Y
):GOTO 730
710:IF LEN STR$ D(
I,Y)=2LET K$(Y
)=" "+STR$ D(
I,Y):GOTO 730
720:K$(Y)=STR$ D(
I,Y)
730:NEXT Y
740:COLOR 0
750:LPRINT TAB 3;"
3. Dritt el";TAB
2
18;K$(0);TAB 2
3;K$(1);
751:LPRINT TAB 28;
K$(2);TAB 33;K
$(3)
760:NEXT I:END
STATUS 1
2322

```



```

180 CLS :PRINT "Abstand von oben: mm
190 CURSOR 19: INPUT R0
200 IF RL>BRLET RL=BR
210 IF R0>HRET R0=HR
220 CLS :INPUT "Spaltenzahl (vertikal) " :IS
230 INPUT "Reihenanzahl (horizontal) " :IR
235 PRINT "xxx Ausgabe xxx
240 IF ZHS+ZUS=" THEN "RMAX"
245 IF ZHS+ZUS="JJ" THEN "JJ"
246 IF ZHS+ZUS="NN" THEN "NN"
250 IF ZHS="N" THEN "ZU"
260 LR=((BR/2)-2)/.2:WR=((R0-3)/.2:GOTO "TAB"
270 "ZU"WR=((HR/2)+2)/.2:LR=((RL-10)/.2:GOTO "TAB"
275 "RMAX"LR=((BR/2)-2)/.2:WR=0
280 "JJ"LR=((BR/2)-2)/.2:WR=((HR/2)/.2:GOTO "TAB"

```

```

..5:"HARDCOPY".
..10:"A".
500:GRAPH
600:CSIZE 2:COLOR
    0:G=10:A$="o"
700:GLCURSOR (50,0)
    ):SORGN
1000:FOR I=0TO 15
    5STEP 1
1100:X=POINT 1
2003:B$="0":IF X>
    =64THEN
    GOSUB 2017
2004:C$="0":IF X>
    =32THEN
    GOSUB 2016
2005:D$="0":IF X>
    =16THEN
    GOSUB 2015
2006:E$="0":IF X>
    =8THEN GOSUB
    2014
2007:F$="0":IF X>
    =4THEN GOSUB
    2013
2008:G$="0":IF X>
    =2THEN GOSUB
    2012
2009:H$="0":IF X>
    =1THEN GOSUB
    2011
2010:GOTO 4001
2011:X=X-1:H$="1"
    :RETURN
2012:X=X-2:G$="1"
    :RETURN
2013:X=X-4:F$="1"
    :RETURN
2014:X=X-8:E$="1"
    :RETURN
2015:X=X-16:D$="1"
    :RETURN
2016:X=X-32:C$="1"
    :RETURN
2017:X=X-64:B$="1"
    :RETURN

```

```
00355 LPRINT "M";SAZ;"",":0:LPRINT "D";SAZ;"",":Y;Z=Z+1  
00360 IF Z<THEN 370  
00365 LPRINT "M";SAZ;"",":Y:LPRINT "D";SAZ;"",":0:Z=Z+1:GOTO 350  
00370 GOTO "R"  
00380 "Y1"LPRINT "D";X;"",":0:"";X;"",":S99:LPRINT "I";LPRINT "O";0;"",":Y1;"",":  
X;"",":Y1  
00390 LPRINT "D";-X;"",":0:LPRINT "I";LPRINT "O";0;"",":999:LPRINT "I"  
00400 Z=1  
00410 IF Z<THEN "H"  
00420 LPRINT "M";SA;"",":0:LPRINT "D";SA;"",":S99:LPRINT "I"  
00430 LPRINT "D";0;"",":Y1:Z=Z+1  
00440 IF Z<THEN "H"  
00450 LPRINT "M";SA;"",":Y1:LPRINT "D";SA;"",":0:LPRINT "I"  
00460 LPRINT "D";0;"",":999:LPRINT "I";Z=Z+1:GOTO 410  
00465 "H";S1=S  
00470 P=S1-Z:S1=P:IF P>1 THEN 470  
00480 IF P<1 THEN "U"  
00490 LPRINT "M";-(S-1)*SA;"",":999:LPRINT "I";LPRINT "M";Y1;"LPRINT "I";GOTO "R"
```

```

4001: IF B$="1"
      THEN GOSUB 5
      001
4002: IF C$="1"
      THEN GOSUB 5
      002
4003: IF D$="1"
      THEN GOSUB 5
      003
4004: IF E$="1"
      THEN GOSUB 5
      004
4005: IF F$="1"
      THEN GOSUB 5
      005
4006: IF G$="1"
      THEN GOSUB 5
      006
4007: IF H$="1"
      THEN GOSUB 5
      007
4200: GLCURSOR (0,
      -G):SORGN
4500:NEXT I
4700:RETURN

5001:XX=G:
      GLCURSOR (XX
      ,0):LPRINT A
      $:RETURN
5002:XX=2*G:
      GLCURSOR (XX
      ,0):LPRINT A
      $:RETURN
5003:XX=3*G:
      GLCURSOR (XX
      ,0):LPRINT A
      $:RETURN
5004:XX=4*G:
      GLCURSOR (XX
      ,0):LPRINT A
      $:RETURN
5005:XX=5*G:
      GLCURSOR (XX
      ,0):LPRINT A
      $:RETURN
5006:XX=6*G:
      GLCURSOR (XX
      ,0):LPRINT A
      $:RETURN
5007:XX=7*G:
      GLCURSOR (XX
      ,0):LPRINT A
      $:RETURN
5555:END
6000:"Z"
6005:PRINT "====
      SHARP PC-150
      0===="
6009:GOSUB "A"

```

100-103000-1-364

SHARP PC-1500(A) TABELLEN-PROGRAMM SHARP CE-515P

PC 1500

SKAT- BUCHFÜHRUNG

Wer kennt nicht beim Skat die leidige Frage "Wer schreibt an?".
Ihr PC 1500/A + CE 150 lösen dieses Problem brillant!

Das Programm wird mit DEF "S" gestartet.
Bei der Frage nach dem Gewinner / Verlierer eines Spieles
genügt die Eingabe der ersten beiden Buchstaben des Namens.
Im übrigen erklärt sich das Programm von selbst.
Mit DEF "B" kann abgerechnet werden. (siehe Beispiel:
Abrechnung zu 1/10 Pf.)

CCM

Computer-Club Mölln

```

10: "S" CLEAR :
  PAUSE "Skat-Bu
  chführung"
15: WAIT 0: INPUT "
  Zahl der Spiel
  er? "; A
16: IF (A/4)+(A/3)
  THEN 15
17: DIM B*(3)
20: FOR I=0 TO A-1:
  PRINT "Name ("
  +STR$(I+1)+")
  = "; INPUT B*(
  I)
30: CURSOR : NEXT I
  CLS : CSIZE 1:
  COLOR 3
35: LPRINT USING "
  #####;B*(0)
  ;B*(1);B*(2);B
  *(3); " Spiel"
  :USING
36: LPRINT "=====
  =====
  ==": IF PRETURN
38: INPUT "500 + 0
  atum ?(ja=1) "
  ;Z: IF Z INPUT "
  Heutiger Tag"
  ;Y
39: COLOR 0
40: WAIT :FOR I=0
  TO A-1: IF W
  PRINT B*(1);
  GIBT ";STR$ Q;
  ". BOCK":GOTO
  50
42: IF UPRINT B*(1
  ); " GIBT ";
  STR$ Q; ". RAMS
  CH":GOTO 50
45: PRINT B*(1);
  GIBT"
50: A$="": INPUT "G
  ewinner/Verlie
  rer ? ";A$:A$=
  LEFT$(A$,2)
60: INPUT "Gewinn/
  Verlust=" ;B
65: IF Z THEN 200
70: FOR J=0 TO 3
80: IF A$=LEFT$(B
  *(J),2)LET B*(3
  +J)=B*(3+J)+B:
  GOTO 120
90: NEXT J
110: GOTO 50
120: LPRINT USING "
  #####;C:
  LPRINT USING "
  #####;D;E;F
  :LPRINT USING
  "#####;B:
  IF WLF -1:
  LPRINT TAB 35:
  "B"

```

```

122: IF WLF -1:
  LPRINT TAB 35:
  "R"
123: IF Z=0 THEN 126
124: FOR J=3 TO 6: IF
  B*(J)<=-500-Y
  BEEP 10:PRINT
  "Spielende":LF
  10:GOTO 555
125: NEXT J
126: IF W+V=0 INPUT
  "Folgt Backhau
  de ?(ja=1) ";X
  :IF X THEN 300
127: IF W THEN 305
129: IF U THEN 404
130: NEXT I
140: GRAPH :LINE (0
  ,0)-(250,0):
  TEXT :CSIZE 1:
  LPRINT :GOTO 4
  0
200: IF (A$=LEFT$(B
  *(0),2))*(B*(0)
  )LET D=D-B:E=E
  -B:F=F-B:GOTO
  230
205: IF A$=LEFT$(B
  *(0),2)LET C=C
  +B:GOTO 240
210: IF (A$=LEFT$(B
  *(1),2))*(B*(0)
  )LET C=C-B:E=E
  -B:F=F-B:GOTO
  230
215: IF A$=LEFT$(B
  *(1),2)LET D=D
  +B:GOTO 240
220: IF (A$=LEFT$(B
  *(2),2))*(B*(0)
  )LET C=C-B:D=D
  -B:F=F-B:GOTO
  230
225: IF A$=LEFT$(B
  *(2),2)LET E=E
  +B:GOTO 240
230: IF (A$=LEFT$(B
  *(3),2))*(B*(0)
  )LET C=C-B:D=D
  -B:E=E-B:GOTO
  240
235: IF A$=LEFT$(B
  *(3),2)LET F=F
  +B:GOTO 240
237: GOTO 50
238: IF A=3LET F=B
  240: GOTO 120
300: W=A:Q=1:GOTO 1
  30
305: W=W-1:D=Q+1: IF
  W=0BEEP 2:
  PRINT "Ramsch-
  Runde" :X=B:U=A
  :Q=Q-A
310: GOTO 130
404: V=V-1:D=Q+1

```

```

405: INPUT "Gränd H
  and? (ja=1) ";
  U:GOTO 410
407: U=0
410: IF ULET U=U+1:
  I=1-1:Q=Q-1:LF
  -1:LPRINT TAB
  30;"G":TAB 0
415: IF U=0BEEP 2:
  PRINT "Ramsch-
  Runde beendet"
  :Q=0
420: GOTO 130
500: "B" P=1:GOSUB 3
  6:LF 3:LPRINT
  "Abrechnung"
510: INPUT "Abr.-Mo
  dus (XX/10 P.)
  = ";D:D=1/10
520: C=INT (C*0+.5)
  :D=INT (D*0+.5)
  :E=INT (E*0+.
  5):F=INT (F*0+
  .5)
530: K=2*C-D-E: IF A
  =4LET K=K+C-F
535: L=2*D-C-E: IF A
  =4LET L=L+D-F
540: N=2*E-C-D: IF A
  =4LET N=N+E-F
545: IF A=4LET N=3*
  F-C-D-E
550: LPRINT :LPRINT
  USING "#####";
  K:LPRINT
  USING "#####";
  L:M:N
555: USING :END

```

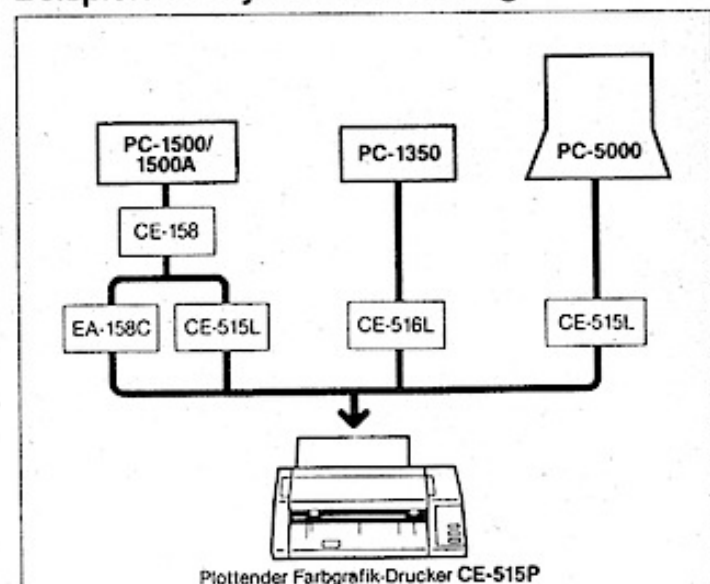
HORST	KLAUS	PETER	JUERGEN	Spiel
0	20	0	0	20
0	20	10	0	10
0	20	10	-120	-120
0	20	10	-72	480
00	20	10	-72	000
00	122	10	-72	560
00	132	110	-72	1000
-72	132	110	-72	-1520
-72	132	110	100	2400
-72	132	110	200	6 1200
-72	132	110	222	-500
-72	132	31	222	-070
-54	132	31	222	10
20	132	31	222	00
00	132	31	222	00
00	132	31	104	-10
-74	132	31	104	-100
-194	132	21	104	-1200
-194	212	21	104	000
-194	212	21	304	1200
-194	212	21	424	1700
-530	212	31	424	-3440
-530	232	31	424	6 000
-530	352	31	424	6 000
-530	302	31	424	-150
-530	302	-57	424	-000
-530	302	-110	424	-530
-010	302	-110	424	-00
-570	302	-110	424	10
-570	302	-110	402	55
-570	302	-110	325	40
-450	302	-110	325	120
-450	302	-02	325	10
-450	207	-02	325	-20
-420	207	-02	325	24

Abrechnung

-202 27 -35 101

Beispiele für Systemerweiterung

CE-515P



Plottender Farbgrafik-Drucker CE-515P

Do not sale this PDF !!!



ATTENTION ! ATTENTION



WENN DER SPEICHERPLATZ ZU KNAPP WIRD, DANN WIRD'S GEFAEHRlich !
 DESHALB HIER EIN ANGEBOT FUER ALLE PC 1350/1450/2500 ANWENDER
 EINE RAM-KARTE DIE ZU CE 202 M KOMPATIBEL IST ZUM SUPERPREIS:



16 KB RAM KARTE 249,-DM INCL.MW



***** MIT 6 MONATEN GARANTIE *****
 FALLS SIE NUN UEBERZEUGT SIND, SOLLTEN SIE SCHNELL BESTELLEN.
 DAMIT IHNEN BEIM PROGRAMMIEREN DAS LICHT (RAM) NICHT AUSGEHT!

SPEICHERERWEITERUNGEN BIS 76 KB

»»»»»»»»»»»» FUER SHARP PC 1500 UND PC 1500/A ««««««««««««
 ALLE ERWEITERUNGEN WERDEN IN DIE RECHNER EINGEBAUT.
 BESTE INDUSTRIEQUALITAET MIT ** 2 JAHREN VOLLGARANTIE **

Wie schon oben erwähnt, werden alle im folgenden beschriebenen Speichererweiterungen in den Rechner eingebaut, d.h. das Modulfach und die nach außen geführte Steckerleiste bleiben für andere Erweiterungen frei.

Selbstverständlich kann sämtliches SHARP-Zubehör wie Drucker CE 150 usw. in der gewohnten Weise weiterbenutzt werden.

Die Speicherverteilung im PC 1500 und PC 1500/A sieht folgendermaßen aus:
 (alle Angaben hexadezimal)

in der ME0-Ebene:

0000 - 6FFF gesamt = 28 KB Basic
 7C01 - 7FFF gesamt = 1 KB Maschinensprache, Variablenfelder oder Daten
 8000 - BFFF gesamt = 16 KB Maschinensprache, Variablenfelder oder Daten

in der ME1-Ebene:

0000 - 7FFF gesamt = 32 KB für Variablenfelder oder Daten

Achtung: Die 32 KB in der ME1-Ebene können mit dem Zusatzprogramm ROSY (Ram-Organisations-System) für die Speicherung von Basic-Programmen genutzt werden! Die Übertragung erfolgt blitzschnell !! Rosy kostet 60,- DM incl.Mwst.
 Bei einem Vollausbau in der ME1-Ebene erhalten Sie ROSY kostenlos !!!!!!!!

Und nun die Bestellnummern mit Preisen:

000115	Erw.um	6 KB	199,- DM	000125	Erw.um	26 KB	389,- DM
000116	"	8 KB	219,- DM	000126	"	28 KB	449,- DM
000117	"	10 KB	239,- DM	000127	"	30 KB	469,- DM
000118	"	12 KB	259,- DM	000128	"	32 KB	489,- DM
000119	"	14 KB	279,- DM	000129	"	36 KB	619,- DM
000120	"	16 KB	299,- DM	000130	"	44 KB	649,- DM
000121	"	18 KB	319,- DM	000131	"	52 KB	699,- DM
000122	"	20 KB	339,- DM	000132	"	60 KB	749,- DM
000123	"	22 KB	349,- DM	000133	"	68 KB	799,- DM
000124	"	24 KB	369,- DM				

Falls Ram-Module zum Einstecken schon vorhanden sind, können diese kostenlos in den Rechner mit eingebaut werden, d.h. wenn Sie beispielsweise für Ihren PC 1500/A einen Vollausbau auf 28 KB Basic wünschen und bereits ein 8 KB Modul besitzen, zahlen Sie effektiv nur noch 279,- DM incl.Mwst. für eine 14 KB Erweiterung, vorausgesetzt Sie schicken Ihr Modul mit dem Rechner ein. Wenn Sie den gleichen Fall mit dem PC 1500 hätten, müßten Sie noch 319,- DM incl.Mwst. berappen!

Alle Preise incl.14%Mwst

Do not sale this PDF !!!

Das Textsystem EasyWriter II Die Erfahrungen eines Anwenders

Als langjähriger WordStar-Fan ging ich zunächst sehr skeptisch an das Textsystem EasyWriter, das im Rahmen des EasyPacs von Sharp für den PC-5000 angeboten wird. Obwohl mir dieser Computer vom Konzept her sehr gut gefällt, hielt ich eine anspruchsvolle Textverarbeitung auf einem 8-Zeilen-Bildschirm doch für unmöglich. Der Einstieg fiel mir durch das gut gemachte Handbuch sehr leicht. Es bietet zu allen Funktionen Übungen, bei denen man die einzelnen Befehle schnell begreift. Für die spätere Anwendung gibt es eine kleine Broschüre, in der alles noch einmal in Kurzfassung zu finden ist.

Zuerst fällt einem auf, daß man keinen Text editieren kann, bevor man nicht einen Dateiordner eröffnet hat, in dem man seine Dokumente ablegt. Damit kann man einzelne Texte nach Obergruppen zusammenfassen, was alles übersichtlicher macht. Außerdem hat man die Möglichkeit Systemparameter wie Seitenlänge, Standardname des Dateiordners, Alternativen für den Druck etc. zu ändern und auch abzuspeichern. Das System unterstützt den eingebauten Thermodrucker übrigens vollständig, jedoch auch der Betrieb eines RS232C-Druckers ist möglich.

Jedem Text wird eine Nummer zugeordnet, unter der er immer wieder aufrufbar ist. Zusätzlich wird der Dokumentenname, das Kürzel des Autors und des Operators, sowie die Daten von Erstellung und letzter Bearbeitung gespeichert. Beim Tippen der ersten Texte merkte ich dann, daß der Bildschirm sehr gut genutzt wird. Zwei Statuszeilen und ein Lineal zeigen alle wichtigen Informationen an, die restlichen 5 Zeilen bleiben für den Text. Und das reicht wirklich, um gut arbeiten zu können. Die Cursorsteuerung ist zwar etwas eigenartig, doch man gewöhnt sich schnell an dieses gut durchdachte Prinzip. Mit den Funktionstasten kann man zwischen verschiedenen Modi wählen (Zeichen, Wort, Satz...Seite) auf die sich dann alle Funktionen wie vor, zurück, löschen und einfügen beziehen. Übrigens sind auch alle anderen Befehle auf den Funktionstasten untergebracht (3fach belegt). Die mitgelieferte Tastaturschablone gibt ständig Auskunft über die entsprechende Belegung. Auch die verschiedenen Druckarten sind hier zu finden. Über einen Tastendruck sind verschiedene Variationen aufrufbar, wobei sich die Änderung wieder auf den eingestellten Modus bezieht. Es ist also möglich eine ganze Seite mit einem Befehl fett zu drucken. Man kann einfach oder doppelt unterstreichen, verschieben, fett drucken, überschreiben und natürlich auch mehrere Arten gleichzeitig benutzen, z.B. fett, doppelt unterstrichen. Das Besondere daran ist, daß alle Druckarten auch so auf dem Bildschirm erscheinen und nicht durch Steuerzeichen ersetzt werden.

Außer den sowieso vorhandenen Sonderzeichen des PC-5000, die ja fast alle europäischen Sprachen berücksichtigen, lassen sich bei EasyWriter über Controlbefehle noch folgende weitere Zeichen abrufen, die auch ausgefallenen Sonderwünschen in der Textverarbeitung gerecht werden : © • • • * * * * * : || ¶ § _

Kopf- und Fußzeilen (für rechte und linke Seiten getrennt), Randausgleich, Zentrieren und Such- und Austauschfunktionen sind so selbstverständlich, daß ich hier auch nicht weiter darauf eingehen will. Interessanter ist dagegen die Möglichkeit für jede Zeile des Textes eine andere Gradskala mit rechtem und

linkem Rand, Tabulatoren, Dezimaltabulatoren und Schriftdicke zu wählen und unter einem Namen abzuspeichern, um sie jederzeit wieder abzurufen. Genauso lassen sich mit Namen versehene Blöcke beliebig verschieben, kopieren, löschen und sogar mit anderen Dokumenten mischen. Natürlich ist auch eine automatische Trennhilfe beim Neuformatieren vorhanden, wenn diese auch manchmal eigenartige Trennungen vorschlägt.

Natürlich bietet EasyWriter noch weitere Funktionen, die Sie aber am besten selbst einmal ausprobieren sollten. Für eventuelle Erweiterungen hat der Hersteller auf jeden Fall schon mal eine Nummer im Hauptmenü freigelassen.

Ich kann jedem dieses Gerät für die Textverarbeitung nur empfehlen, zumal man es auch anderweitig nutzen kann und es Dank des Betriebssystems MSDOS sogar IBM-kompatibel ist.

Tagespreis bitte anfragen.

--- BUCHTIP ---

oder Geschenktipp für Ehefrauen, Freundinnen, Mütter und Väter, die Ihren an Computer spielenden grossen und kleinen Kindern eine Freude machen wollen. Es ist das

Basic Lehrbuch für SHARP Computer

Das Buch ist für Anfänger und Fortgeschrittene ein unbedingtes Muß. Es beschreibt schrittweise und in einer sehr lockeren Schreibweise den Aufbau eines Computers, seine Eigenheiten und gibt danach eine ausführliche Beschreibung der wichtigsten Basic-Befehle. Viele Befehle werden auch in kleinen Programmen erklärt, so daß der Leser (Programmierer) schrittweise lernt, wie das Programm aufgebaut ist und funktioniert. Am Ende eines jeden Kapitels ist eine kleine Lernzielerfolgskontrolle, dabei kann sich jeder selbst testen, ob er es kapiert hat oder nicht. Ich möchte den Anfängern empfehlen, nach dem Studium des Buches, mal Ihre ersten Programmerversuche aus der Schublade zu holen und mit dem eben erlernten zu vergleichen (gilt auch für Fortgeschrittene). Sie werden feststellen, daß Sie jetzt viele Programme anders und vielleicht besser schreiben und gestalten werden. Die Autoren haben sich grosse Mühe gegeben und haben das Buch so geschrieben, daß niemand nach den ersten Seiten den Spaß an seinem Hobby verliert.

(SHARP-Basic-Team)

Autor: Bernhard Hartmann und Jürgen Brenner-Hartmann

Herausgeber: B. Fischel GmbH

Titel: Basic Lehrbuch für SHARP Computer

(ISBN 3-924327-09-2)

Preis 49,- DM inkl. 7% MwSt.

Weitere Funktionen:

Zugriff auf die deutschen Umlaute, Diskettenbetrieb mit FLPB-1/2 Korrigieren nach eigenen Wünschen, Abspeichern der Texte auf Cassette oder Diskette, Ablegen als ASCII-Datei zur Weiterverwendung des Textes in anderen Programmen, zahlreiche Hilfsprogramme im Bildschirmdialog. Verwendung aller gängigen Druckerinterface (LPT, SHARP, MZ-700, Schreibmaschine CE-50) zur Ausgabe auf Papier, unabhängig von der Seitenlänge einstellbarer Blattnumber, "HALT"-Funktion zum Wechseln des Papiers bei einer Schreibmaschine oder Drucker mit Einzelblatt.

SOBS 6

Amateur - Contest Programm (für KW), Abspeichern aller QSO's mit Unterscheidung aller Amateurbänder nach Contestregel, sofortige Anzeige "DOPPELT" bei gleichem Rufzeichen auf gleichem Band. In einer Sonderzeile wird das Rufzeichen, Uhrzeit, RS(T), QSO-Nr. erhalten, RS(T) und QSO-Nr. gegeben, Rapport und Datum angezeigt. Weiterhin wird vom Programm her verhindert, daß ein schon vorhandenes Rufzeichen nochmals eingegeben wird. Kontinuierliche Anzeige von Uhrzeit und Datum mit immerwährendem Kalender, eigener QSO-Nummer, Band auf welchem gerade gearbeitet wird, Rufzeichen und Rapporte. Sonderbefehle für Löschen des letzten QSO's, Suchen nach Präfixen pro Band, Bandwechsel, Ändern der QSO-Nummer, Saven aller QSO's auf Cassette. Paralleles Ausdrucken jedes gültigen QSO's als Logbeleg. Ausdruck aller QSO's nach Bändern geordnet (abwärts fallend) in einem Arbeitsgang. Laden von auf Cassette gespeicherten QSO's zur Kontrolle oder zum Gesamtausdruck. Der Ausdruck ist nur über unsere Interface LPT 2.x ... 4.x möglich. Der Bildschirm zeigt eine übersichtliche Gestaltung in Großfeldern, sodaß mit einem Blick alle wichtigen Daten erfasst werden können. Das Programm kann auch in einer Diskettenversion für unser System FLPB-1 / FLPB-2 geliefert werden.

SOBS 7

Adressenprogramm, geeignet für Rundschreiben ect..., speichert bis zu ca. 700 Adressen und zusätzlich einen Text bis zu ca. 8000 Buchstaben. Das Adressenprogramm entspricht dem Programm SOBS 4 mit Bildschirmediting. Der eingegebene Text kann auf dem Bildschirm editiert werden, sowie auf Cassette/Diskette ausgeladen oder eingelesen werden. Zusätzliche Möglichkeiten: 80 Zeichen(*), bestimmte Adressen welche unter den Suchbegriff fallen mit dem eingegebenen Text auszudrucken. Das Programm erfüllt die Aufgabe eines einfachen Schreibautomaten. Druckerbetrieb ist nur mit unseren Interface LPT 2.x ... LPT 4.x möglich, wobei in einer Sonderausführung auch als Drucker die Schreibmaschine CE-50 oder CE-60 verwendet werden kann. Bitte bei Bestellung Druckertyp angeben.

(*) = nur wenn 80-Zeichensatz eingebaut ist!

SOBS 8

Extended Maschinensprache für die gesamte SHARP Serie. Die Maschinensprache SP-2001 von SHARP erfüllt nicht alle Forderungen eines Programmierers. Besonders, da auf die sog. "geschützten Bereiche" kein Zugriff möglich ist, und das Maschinenprogramm genau in dem Speicherbereich arbeitet, in dem auch andere Programme (z.B. Basic) arbeiten. Unsere Maschinensprache hat diese Nachteile nicht. Der Arbeitsbereich liegt zwischen C000H und CFFFH, (in der EPROM Ausführung zwischen F000H und FFFFH) daher kann parallel dazu jedes andere Programm geladen werden. Das Programm enthält den normalen Befehlssatz der "MACHINE LANGUAGE SP-2001" und darüber hinaus noch folgende Verbesserungen und neue Befehle:

Produktserie MZ-80

Beschreibung der in der Preisliste angebotenen Programme:

SOBS 2

CW - Trainingsprogramm mit Zufallsgenerator für 5-er Gruppen. Einstellung der Geschwindigkeit von 40 BpM bis 125 BpM von der Tastatur aus, sowie Einstellung von Abstand zwischen den Zeichen und Wörtern, Punkt-Strich-Verhältnis und der Tonhöhe in jeweils 9 Stufen von der Tastatur aus. Zufallsgenerator für 5-er Gruppen, hierbei ist zwischen allen CW Zeichen oder Zeichen nach Vorgabe zu wählen. Einstellung der Geschwindigkeit der Ausgabe mit der laufenden CW-Zeichenausgabe. Unterbrechung der Ausgabe mit der "SPACE"-Taste. Freier Text bis zu 46000 Zeichen einbaubar. Ausgabe aller Texte (auch Zufallstexte) auf Drucker (nur über unsere Interface LPT 2.x ... 4.x oder MZ-700 Drucker).

SOBS 3

Logbuchführungs- oder Datenhaltungsprogramm. Mit Einträgen von beliebig langen Textblöcken, Nachtragen in schon bestehende Textgruppen, Überschriften und Ändern von Texten, Suchen nach Einzel- oder Sammelbegriffen. Löschen einzelner Textabschnitte mit automatischem Nachrücken der anderen Texte. Löschen des gesamten Textspeichers zum Anlegen eines neuen Kartell. Ausladen aller Daten (einschließlich des Systemprogrammes) auf Cassette. Das Programm speichert ca. 1200 Textblöcke à 40 Zeichen. Ausgabe auf Drucker im Suchformat nach dem Suchkriterium. Druckerbetrieb mit unseren Interface LPT 2.x ... LPT 4.x oder für SHARP Drucker möglich. Bitte bei der Bestellung Druckertyp und Interface angeben.

SOBS 4

Adressen- bzw. Datenhaltungsprogramm, in der Grundfunktion etwa gleich dem Programm SOBS 3, jedoch ist die Eingabe eines Textblockes auf eine Bildschirmseite pro Block begrenzt. Dafür enthält das Programm eine komfortable Bildschirm Editing Möglichkeit, sodaß in einer Funktion Texte geändert, verlängert, überschrieben oder gekürzt werden können. Speichermöglichkeit ca. 800 Adressen bei 55 Zeichen pro Adresse im Mittel. Die angegebene Speichermöglichkeit bezieht sich bei allen Programmen auf die voll ausgebaute Version des Computers. Druckerbetrieb ist mit unseren Interface LPT 2.x ... 4.x oder SHARP P3 / MZ-700 Drucker möglich. Bitte bei der Bestellung den Druckertyp und Interface angeben. In einer Sonderausführung ist dieses Programm mit einer frei einstellbaren Bildschirmmaske lieferbar.

SOBS 5

Textverarbeitungsprogramm für MZ-80A und MZ-700 (dieses Prospekt ist mit diesem Programm und einem Typenradrunder (Juki 6100) geschrieben). Eingabe bis zu ca. 8 Seiten Text, automatischer Randausgleich einstellbar, Bildschirmeditor, direktes Einschreiben mit automatischer Trennung (keine Worttrennung), Umschaltung auf 80-Zeichensatz per Programm (sofern im Computer eingebaut), Umcodierung der Tastatur auf Kleinschrift wie Schreibmaschine über Original-Tastatur, Eingabe von Steuerbefehlen an den Drucker über dezimalen Steuercode, Anpassung an gängige Drucker über Steuerzeichen Initialisierungsprogramm, Ausdruck beliebig einstellbar von 20 Zeichen pro Zeile bis 132 Zeichen pro Zeile, mit oder ohne rechten Randausgleich formatieren, der linke Rand ist variabel - Seitenlänge und Zeilenzahl sind frei einstellbar.

Automatische Rechnungsnummernzerlegung bis 4 Stellen, Rechnungsnummer wird nach jeder Rechnung um 1 erhöht. Weitere Funktionen nach Wunsch integrierbar. Reines Maschinensprache-Programm ca. 10 kByte (Cassette) bzw. ca. 14 kByte (Diskette). Das Programm wird mit Erfolg von einigen Firmen in Deutschland und der Schweiz verwendet. Bei Interesse sprechen Sie bitte mit uns.

SOBS 14

Adressenspeicher- oder Datenhaltungsprogramm wie SOBS 4, jedoch mit zusätzlicher Angabe einer Kenn-Nummer oder Sonderkennzeichen in der 1. Zeile der Adresse. Sortierung wie bei SOBS 4 oder nach den eingegebenen Sonderkennzeichen. Ausgabe der Adressen auf Etiketten, Druckerformat an alle üblichen Klebeetiketten anpaßbar. Druckerausgabe nur über unsere Interface LPT 2.x ... 4.x möglich!

SOBS 17

Adressenspeicher- und Suchprogramm. Auf Cassette und auf Diskette lieferbar. Grundaufbau etwa wie SOBS 4. Zusätzlich enthaltene Sortierfunktion für Namen (Familiennamen alphabetisch steigend) und für Postleitzahlen (aufwärts steigend). Zeigt die Anzahl der gespeicherten Adressen an, den verbleibenden Freiraum im Speicher sowie die aktuelle Sortierposition. Druckerformat frei einstellbar in Zeilenanzahl, Seitenlänge und linker Rand. Abspeichern aller Adressen auf Cassette oder Diskette. Bildschirm Editing in allen Funktionen, "Blättern" (vorwärts) in der Kartei mit gleichzeitigem Verbessern. Die Druckerausgabe kann nur über unsere Interface LPT 2.x ... 6.x oder über das Druckerport des MZ-700 erfolgen. Eine Druckerausgabe über den P3 Drucker ist nicht möglich! Auf Wunsch mit einstellbarer Bildschirmmaske und für den 80-zeilen Bildschirmbetrieb lieferbar. Ist eine Diskettenstation FLPB angeschlossen, werden Programm und alle Daten vor dem Programmende automatisch auf der Diskette abgelegt.

SOBS 18

EPROM Maschinensprache nur für den MZ-80A, das EPROM wird in den freien "USER" Sockel des MZ-80A gesteckt. Da nur 2 kByte zur Verfügung stehen, wurde diese Maschinensprache gegenüber der M-Sprache SOBS 8 etwas "abgemagert". Die Funktionen "Anzeige im Assemblerformat", "String suchen", "Druckerausgabe" und "Disk-befehle" sind nicht enthalten. Ein Breakpoint kann nur über "AST 38H" gesetzt werden.

SOBS 19

Disassembler nach Zilog-Syntax, disassembliert Befehle, Daten, Datenwörter und "SHARP ASCII" Texte im Einzelschritt oder kontinuierlich. Über Tastenbefehl kann zu jedem disassemblierten Wert ein Kommentar von Hand eingegeben werden. Zwischen den Datenzeilen können beliebig viele Kommentare geschrieben werden. Darstellung des OP-Codes nach Zilog- bzw. Mostek 2-80 Syntax. Datenbytes werden mit DEFB, Datenwörter mit DEFW und ASCII Texte mit DEFW bezeichnet. Drei Druckeroutinen sind im Programm enthalten (SHARP I/O-P3 Drucker + Interface LPT 6.1/6.2, SHARP Interface DNR und unsere gesamte Interface LPT 2.x... 4.x-Serie).

Anwendersoftware

Beim Start wird der Arbeitsspeicher nicht gelöscht. Die Bildschirmanzeige ist schneller (nur MZ-80K und MZ-700) und kann in allen Funktionen mit der "SPACE" - Taste angehalten werden. Voller Zugriff auf den gesamten 64 kByte Bereich, also auch auf Monitor, Video-RAM, Tastatur-Buffer ect...

Neue Befehle:
Anzeige eines definierten Bereichs im Assemblerformat. Löschen eines angegebenen Speicherbereichs bis 64 kByte. Löschen des Arbeitsspeichers. Umrechnung von Hex in Dezimal. Umrechnung von Dezimal in Sedezimal(Hex). Addition und Subtraktion zweier sedezimaler Zahlen. Anzeige des noch freien Speicherbereichs im Arbeitsspeicher. Texteingabe in ASCII in den angegebenen Bereich. Vergleichen von zwei Speicherblöcken bis zu je 22 kByte Länge. Anzeige von ASCII und von VIDEO Texten. Suchen nach beliebigen Zeichen in ASCII oder Hex. Verschieben der Adresse beim Laden und Speichern. Laden und Speichern von Diskette. Anzeige des Direktors der Diskette. Druckeroutine mit deutschem Blattformat (64 Zeilen, 8 Leerzeilen). Nur für Interface LPT 2.x ... LPT 4.x lieferbar. Verwendbares Floppy-System: FLPB-1 oder FLPB-2.

SOBS 10 (SA-5510 Seriell)

Seriellles Datenausgabe-Programm für den Basicinterpreter SA-5510. Dieses Programm entspricht bis auf die Druckeroutine dem originalen Basic-Interpreter SA-5510 (MZ-80A) von SHARP. Es wurden die Druckeroutinen geändert, sodaß mittels einer seriellen Schnittstelle (LPT 4.0 oder LPT 4.1) eine Ausgabe des Basic-Textes auf ein seriellles Terminal oder Modem möglich ist. Die Umcodierung auf den Standard - ASCII Code und die Einstellung des Interfaces erfolgen vom Programm aus.

SOBS 11

Rechnungsprogramm nach Ihren Sonderwünschen erstellt. Lieferbar auf Cassette und auf Diskette. In der Standard-Version sind folgende Funktionen enthalten: Speicherung von ca. 1000 Artikeln und ca. 400 Adressen (die Angabe ist relativ, da sich der Speicherbereich für Artikel und Adressen automatisch anpaßt). Für Artikel und Adressen ist ein Verwaltungsprogramm (etwa gleich dem SOBS 4) enthalten. Es können Artikel und Adressen eingegeben, geändert, gesucht, gelöscht, verlängert oder gekürzt werden. In allen Verwaltungsfunktionen ist echtes Bildschirmediting möglich. Das Hauptprogramm setzt in die zu schreibende Rechnung (oder Lieferschein, Bestellung usw...) die gewünschte gespeicherte Adresse ein. Ist diese nicht gespeichert, wird sie nachträglich eingegeben und gespeichert. Ebenso werden die geforderten Artikel eingesetzt (wenn diese nicht gespeichert sind werden sie von Hand eingegeben und abgespeichert), die Menge der Artikel wird mit dem Einzelpreis multipliziert, die Gesamtsumme, Mehrwertsteuer und Endsumme wird errechnet. Abspeichern aller Daten auf Cassette, Ausdruck der Rechnung über unser Interface LPT 4.1 auf paralleldrucker. In der Diskettenversion ist zusätzlich enthalten: Relativ unbegrenzte Länge der Artikel- und Adressdaten (nur durch die Speicherkapazität der Diskette begrenzt), Ablegen des jeweiligen Rechnungstextes auf Diskette. Ausdruck der Rechnung direkt oder von der Diskette. Wählen wieoft eine Rechnung gedruckt werden soll (1 bis 9 mal). Ausdruck aller geschriebenen Rechnungen hintereinander von Diskette. Initialisierungsprogramme für Formularestellung, Anpassung an alle gängigen Druckertypen. Einstellung der Mehrwertsteuer von 0,01% bis 99,99%. Die Rechengenauigkeit beträgt 32 Stellen dezimal. Festkomma Addition, Subtraktion und Multiplikation 2-stellig. 100er Stückpreise über "1"-Zeichen einstellbar.

Programm SOBS 18, Maschinensprache für den MZ-80A in EPROM für den USER-Sockel	DM 78,00
Programm SOBS 19, komfortabler Disassembler mit Kommentar, 3 Druckeroutinen	DM 68,00
Programm SOBS 31, Amateur RTTY-CW-Programm mit Interface LPT 4.0 ohne Druckeroption MZ-80A MZ-80K/MZ-700 + Gehäuse u. Netzteil	DM 478,00 DM 568,00
Programm SOBS 32, Amateur RTTY-CW-Programm mit Interface LPT 4.1 mit Druckeroption MZ-80A MZ-80K/MZ-700 + Gehäuse u. Netzteil	DM 547,00 DM 637,00
(nur MZ-80A) Amateur-RTTY-CW-Programm ohne Interface Diskettenversion + 80 Zeichen/Zeile	DM 181,00 DM 198,00
Programm SOBS 33, Morsedecoder (CW-Teil des RTTY-Program.)	DM 43,00
Programm SOBS 58, Adressenprogramm für direkten Diskettenzugriff ca. 2800 Adressen/Diskette je 128 Bytes	DM 249,00
Programm SOBS 65, Modem-Programm (Terminal) für LPT 4.x Int.DM	DM 69,00
Assembler-Editor MZ-80TU für MZ-80K und MZ-80A	DM 148,00
Assembler-Editor für MZ-700 mit deutschem Handbuch	DM 128,00
Pascal Interpreter für MZ-700 mit deutschem Handbuch	DM 128,00

Druckerinterface LPT 6.3 für SHARP MZ-80K/A/700/800

Dieses kostengünstige Druckerinterface wurde entwickelt um eine möglichst große Anzahl von verschiedenen Fabriken von Paralleldruckern am SHARP Computer anschließen zu können. Es ist zu allen SHARP Sprachen kompatibel, d.h. die von Ihnen verwendete Software muß nicht geändert werden. Im mitgelieferten Handbuch sind neben der genauen Beschreibung auch einige Tips enthalten, wie Sie moderne Drucker in ihrer Leistungsfähigkeit voll nutzen können, wenn die im Basic enthaltene Unterdrückung von Steuerzeichen abgesprochen wird. Das Interface enthält neben der Decoderlogik ein EPROM zur Umcodierung des SHARP Zeichensatzes auf Standard ASCII. Dadurch kann auch die Kleinschrift und die deutschen Umlaute ausgedruckt werden. Durch einen Satz DIL-Schalter können 8 verschiedene Zeichensätze eingestellt werden. Die Ausgangsbuchse besteht aus einer 25-poligen D-Min-Buchse.

Unterschiede MZ-80K/A/700/800:

MZ-80A: Nur als Platine mit einer 44-poligen Federleiste zum Anstecken am Systembus des MZ-80A.

MZ-80K/MZ-700/MZ-800: Platine mit Netzteil und Gehäuse. Kabel zum Systembus beim MZ-80K. Kabel zum Printerport zum MZ-700 oder MZ-800.

Technische Daten:
Daten- und Steuerleitungen TTL kompatibel.
Datenbus 8 Bit parallel.
Steuersignale: STROBE, BUSY, ACKN, PAPER-END
STROBE 1 usec. low-aktiv

alle Preise inkl. 14% MwSt.

SOBS 31/SOBS 32 Siehe Beschreibung RTTY-CW-Programm	
SOBS 33 Morsedecoder (CW-Teil unseres RTTY-CW Programmes). Decodiert CW Zeichen von Tempo 20 bis ca. Tempo 600. Zum Betrieb dieses Programmes ist ein serieller Interface (LPT 4.0 oder LPT 4.1) sowie ein RTTY-CW Converter erforderlich.	
SOBS 38 Siehe Beschreibung Floppy Betriebssystem V 3.1/84	
SOBS 58 In den Grundfunktionen etwa gleich dem Programm SOBS 17, jedoch als reines Diskettenprogramm (Direktzugriffsdatei). Es können auf einer Diskette (doppelseitig) 2800 Adressblöcke zu je 128 Bytes gespeichert werden (einsseitige Diskette ca. 1200 Adressen). Automatische Systemerkennung (erkennt MZ-80K, MZ-80A und MZ-700 selbstständig, wie auch das Diskettenformat selbstständig erkannt wird, -40 Spur einseitig, 77 Spur einseitig, 40 Spur zweiseitig). Maximale Zugriffszeit auf die letzte Adresse ca. 1,5 Sekunden. Der Suchbegriff wird sechsstellig abgelegt. Es kann nach Namen (alphabetisch steigend) oder nach Postleitzahlen (numerisch steigend) sortiert werden. Die Sortierzeit beträgt bei 2800 Einträgen ca. 10 Sekunden (Quicksort). Ausdruck aller Datenblöcke in beliebigem Format möglich. Zahlreiche automatische Hilfsprogramme im Grundprogramm installiert. Eigenes Copy - Programm zum Diskettenkopieren enthalten. Das Programm ist nur auf den Floppy-Systemen FLPB 1 oder 2 mit dem FPOS-EPROM ab Version V3.1 in der beschriebenen Geschwindigkeit lauffähig!	
Programme für SHARP MZ-700/800/MZ-80A/MZ-80K (alle Maschinensprache)	
Programm SOBS 2, Morse-Trainer mit 5-er Gruppen Gen.	Cass. DM 35,00
Programm SOBS 3, Logbuchverwaltung, Adresskartei beliebig lange Datenblöcke	Cass. DM 78,00 Disk. DM 88,00
Programm SOBS 4, komfort. Datenverwaltungsprogramm mit Bildschirmeditor	Cass. DM 118,00 Disk. DM 128,00
Programm SOBS 5, Textverarbeitung für MZ-80A/700	Cass. DM 238,00 Disk. DM 248,00
Programm SOBS 6, Amateur-Contest/Fielday (KW)	Cass. DM 158,00 Disk. DM 178,00
Programm SOBS 7, Adressprogramm mit Textspeicher für Rundschreiben ect....	Cass. DM 128,00 Disk. DM 148,00
Programm SOBS 8, Extended Maschinensprache neueste Version mit Diskettenbefehlen	Cass. DM 118,00 Disk. DM 128,00 EPROM DM 148,00
Programm SOBS 10, Serielle Datenübertragung mit LPT 4.1-Interface (Basic SA-5510 seriell)	DM 39,50
Programm SOBS 11, Rechnungsprogramm nach Kundenwunsch, Standardversion mit Hilfsprogrammen Disk.	DM 348,00
Programm SOBS 17, Adressenprogramm + Sortierprogramm, frei einstellbare Druckersteuerung beliebige Suchbegriffe usw.	Cass. DM 138,00 Disk. DM 148,00

Do not sale this PDF !!!

DURCH INFORMATION VORN

Kurzbeschreibung des Floppy-Betriebssystems und des Programmes SOBS 38

Das Betriebssystem (DOS) für unsere Floppysysteme FLPB-1 und FLPB-2 wird auf einem Epron geliefert, welches in das Interface LPT 4.1 eingesteckt wird. Es enthält die notwendigen Steuerfunktionen für die jeweilig verwendete Floppystation und das Hauptmenue SOBS 38. Das Programm ist so ausgelegt, daß max. 4 Laufwerke betrieben werden können.

Das DOS enthält folgende Programmenteile:

Formatieren einer Diskette
Eingabe des aktuellen Datums beim ersten Start.
Start durch Drücken der Taste "F" im Monitor

Lesen eines Programmes von Diskette
Schreiben eines Programmes auf Diskette

Schützen eines Programmes
Öffnen eines geschützten Programmes

Löschen eines Programmes
Ändern des angegebenen auf Diskette abgelegten Namens

Ein Programm von Diskette 1 auf Diskette 2 kopieren
Kopieren einer ganzen Diskette von Laufwerk 1 auf 2

Cassettenprogramm auf angegebene Diskette kopieren
Diskettenprogramm auf Cassette kopieren

Direktory listen und ausdrucken
Programmadresse, Länge, Spur und Sektor eines Programmes anzeigen

Angegebene Spur und Sektor lesen
Angegebene Spur und Sektor schreiben

Spur, Sektor, Datenadressen und Filenamen auf Drucker ausgeben
Direktsprung zu einer angegebenen Adresse

Rücksprung in den Monitor
Alle Befehle können im Direktmodus vom Menue aus, oder über

USR(.....) bzw. CALLH verwendet werden.
andere Formate auf Sonderwunsch erhältlich.

Format: DD 10 Sektoren à 512 Bytes bei 40 Spuren (Single Side)
DD 10 Sektoren à 512 Bytes bei 2 x 40 Spuren (DS)

alle Preise inkl. 14% MwSt.

FDS auf EPROH 2732 DT 198.-

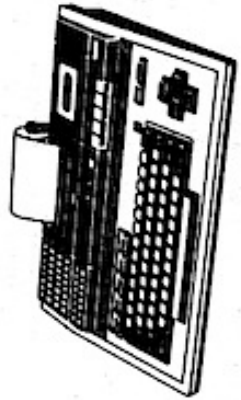
alle Preise inkl. 14% MwSt.

Geschwindigkeiten:

Formatieren SS Diskette ca. 16 Sek.
Formatieren DS Diskette ca. 35 Sek.

Übertragen von 48 Kbyte Daten ca. 3 Sek.
Kopierzeit von Disk 1 auf 2 je nach Daten-

menge ca. 20 bis 120 Sekunden



SHARP-Computer

MZ-700 / MZ-80A / MZ-80K - RTTY - CW - Programm / Kurzbeschreibung

- ASCII, BAUDOT und CW, auch während des Betriebs umschaltbar.
- Klein- und Großschreibung (nur in ASCII)
- Sichtgerätefunktion "REPEAT"
- Vorschreiben während des Empfangs (auch in CW) mit "SPLIT-SCREEN" 1/3 Bildschirm für Vorschreiben, 2/3 Bildschirm für Empfang
- oder nur Empfang mit Vollbild
- während der Sendung Darstellung des gerade ausgegebenen Textes, wie auch des eingeschriebenen oder des gerade vorgeschriebenen Textes
- Unterbrechung der laufenden Sendung und Dazwischenschieben eines neuen Textes oder totale Unterbrechung der Sendung
- Wiederholung einer Sendung (beliebig oft)
- Automatische CW-Kennung am Ende einer RTTY-Sendung (abschaltbar)
- Kommando-Eingabe über \$-Zeichen während des gesamten Betriebs
- Kontinuierliche Anzeige auf dem Bildschirm von: Code, Speicherbelegung, Betriebsart, Uhrzeit und CW-Tempo
- Die Uhrzeit wird in Echtzeit weitergezählt und auf Abruf in die Sendung eingegeben.
- Ausführliche Textmeldungen auf dem Bildschirm in der immer vorhandenen Kommando-Zeile (wird vom Scroll nicht erfasst)
- Abspeichern des Empfangssignals (ein- und ausschaltbar)
- Beliebiges Einfügen von Texten aus Textspeichern, Uhrzeit, Datum ect... über #-Zeichen, bis zu 10 Texten auch verschachtelt
- Vorlaufspeicher für Vorlaufzeit (RYRYRYRY..., CQ..., ect....)
- Vorlaufsynchrozeichen ein- und ausschaltbar (40 Zeichen "BU")
- 8 Textspeicher à 1024 Zeichen, die Speicher können auch während des Empfangs eingetragen werden. Für längere Texte lassen sich mehrere Speicher zu einem großen Speicherblock zusammenbinden
- Alle Textspeicher, FIFO oder Memory können auf Cassette/Diskette ausgeladen werden, bzw. von Cassette/Diskette gelesen werden.
- Automatische Zeilenbegrenzung für die Ausgabe, beliebig von 20 Zeichen bis 80 Zeichen pro Zeile einstellbar
- "AUTO-LINE-FEED" (Sendemodus) mit automatischer Worttrennung
- "UNSHIFT ON SPACE" - Funktion (ein- und ausschaltbar)
- Pausenfüller wird in den Schreibpausen laufend ausgegeben (BU in Baudot, NULL in ASCII, ein- und ausschaltbar!)
- Druckersteuerung für Drucker mit Parallelschnittstelle (Centronics)
- Auch 80 Zeichen - Modus bei umgerüsteten Computern möglich (MZ-80A)

Do not sale this PDF !!!

MZ-80A:	
FLPB-2/80A bestehend aus 2 Laufwerken TEAC FD-55B/35B je 500 kByte unformatiert (408 kByte formatiert), eingebaut in ein Aluminium-Gehäuse mit Netzteil, Kabel zum Controller, Controllerplatine FLPB-1 mit Befestigungsmaterial am MZ-80A, RAM-IC, Kabel zum LPT 4.1 und FDIS-EPROM, jedoch ohne Interface LPT 4.1 oder 4.2 FLPB-4/80A, wie oben, jedoch je Laufwerk 1000 kByte	DM 2.098,00 DM 2.298,00
FLPB-2/80AI wie FLPB-2/80A jedoch mit Interface LPT 4.2 FLPB-4/80AI wie FLPB-1/80A jedoch mit Interface LPT 4.2	DM 2.196,00 DM 2.396,00
MZ-800:	
FLPB-2/800 bestehend aus 2 Laufwerken TEAC FD-55B oder FD-35B, Gehäuse, Netzteil, Anschlusskabel, Controller zum Einschließen in den MZ-800, Diskbasic und Handbuch	DM 1.698,00
CP/M Betriebssystem V 2.2 für MZ-800	DM 298,00
Umrüstsatz MZ-80K/A auf 4 MHz	

Der Umrüstsatz enthält:

- 1 Monitor-EPROM
- 1 2-80A CPU (4 MHz)

Kurzbeschreibung

Für viele Anwendungen ist die 2 MHz Version des MZ-80 zu langsam, speziell bei rechenintensiven Programmen. Bei den bisher bekannten 4 MHz Umrüstungen mittels neuem Antriebstrat das Problem der nicht mehr richtig arbeitenden Sound-Routinen auf. Wir haben bei der Umrüstung einen neuen Weg gewählt: Es wurde ein neues Monitor-Programm entwickelt, welches die Tonhöhen-Einstellung dem 4 MHz Takt anpaßt, ebenfalls die Lade- und die Save-Routinen dem neuen Takt entsprechend einstellt. Das heißt, mit der Umrüstung auf 4 MHz Betrieb mittels neuem Monitor-Programm (in EPROM 4 kByte) ist am Cassettelaufwerk keinerlei Änderung vorzunehmen, und die gesamte vorhandene Software läuft fehlerfrei auf 4 MHz.

Unterschiede MZ-80K und MZ-80A

Beim MZ-80A wird das neue Monitor-Programm in einem EPROM 2732 geliefert, welches die entsprechenden Änderungen zur Taktumstellung enthält. Eine ausführliche Anleitung zum Umrüsten des Taktes von 2 auf 4 MHz liegt bei. Das Monitor-Programm entspricht sonst dem Originalprogramm des MZ-80A. Auf Wunsch kann auch hier, wie im EPROM 5025, die Tastatur neu definiert sein, so daß die deutschen Umlaute und das "p" direkt zugänglich sind.

Beim MZ-80K konnten im neuen Monitor-Programm einige Verbesserungen untergebracht werden. Als erstes wird, die Bildschirm-Routine erheblich schneller eingestellt. Durch Änderung der Umrüst-Routine von "Video-Code" auf "Sharp-ASCII-Code" wurden 256 Bytes Platz eingespart. In diesem Freiraum kann nun noch ein Maschinenprogramm untergebracht werden. In der Standard-Ausführung des neuen Monitors enthält der Freiraum folgende Unterprogramme: 1. Kopierprogramm zum Kopieren jedes Programmes auf eine neue Cassette. Auch geschützte Programme können so direkt dupliziert werden. Aufruf des Programmes über USR(3291) in Basic, oder über CALL 800H in Maschinensprache.

Typenraddrucker Juki 6100, 16 Zeichen/Sek., 4 k Buffer	DM 1.488,00
Olivetti Jet-Drucker JP-101, ** Sonderangebot **	DM 998,00
2000 Blatt Papier weiß 70g/qm 2-gefaltet mit abtrennbarer Randlochung	DM 52,00
Telefon-Modem dataphon s21d mit FTZ-Nr.	DM 298,00
Datenmonitor BMC BM 12ES 12" grün, 18 MHz + Filersch.	DM 378,00
Datenmonitor BMC BM 12EG, 12" grün, 18 MHz entspiegelt	DM 398,00
Datenmonitor BMC BM 12EN, 12" grün entsp. 20 MHz Video	DM 448,00
Ergotilt Schwenkfuß für BMC Monitore	DM 78,00
Umrüstsatz SHARP MZ-80A auf 80 Zeichen pro Zeile mit neuem EPROM, 16 MHz Quarz, Zusatz-IC und Anleitung	DM 198,00
80-Zeichenplatine für 40/80 Zeichenbetrieb mit dem MZ-700 wird nur am Systembus angesteckt, incl. Anleitung	DM 398,00
Umrüstsatz SHARP MZ-80A auf 4 MHz Betrieb, mit neuer 2-80A CPU, neuem Monitor EPROM und Anleitung	DM 88,00
Umrüstsatz SHARP MZ-80K auf 40/80 Zeichen-Betrieb (Fabr. Draht), 2 Platinen mit Einbaueinleitung	DM 298,00
Umrüstsatz SHARP MZ-80K auf 4 MHz Betrieb, mit neuer 2-80A CPU, neuem Monitor EPROM und Anleitung	DM 88,00
Deutsche ASCII-Tastatur für MZ-80K mit 10-er Zahlenblock (Graphik entfällt), einbaufertig	DM 368,00

alle preis inkl. 14% MwSt.

STANDARD - FLOPPYSYSTEME AUS UNSERER FERTIGUNG IN 5 1/4" UND IN 3 1/2"

MZ-80K: (FD-55B = 5 1/4", FD-35B = 3 1/2")	
FLPB-2/80K bestehend aus 2 Laufwerken TEAC FD-55B/35B je 500 kByte unformatiert (408 kByte formatiert), eingebaut in ein Stahlblechgehäuse, eingebautes Netzteil, eingebauter Controller FLPB-1 und Interface LPT 4.2, eingebaute Electronics-Druckerschnittstelle, Diskbasic ext. 5025 auf Diskette, Zusatzplatine zur eindeutigen Speicherdecodierung des MZ-80K	DM 2.398,00
direkt anschlussfähig am I/O-Bus des MZ-80K FLPB-4/80K, wie oben, jedoch je Laufwerk 1000 kByte	DM 2.598,00
MZ-700:	
FLPB-2/700, wie MZ-80K-System, jedoch mit neuem Gehäuse (siehe Beiblatt) in Aluminium eloxiert, S - Diskbasic (ohne Umbauarbeiten am MZ-700 direkt anschließbar!) FLPB-4/700, wie oben, jedoch je Laufwerk 1000 kByte	DM 2.498,00 DM 2.698,00

Universal-Interface LPT 4.x für SHARP MZ-80K/A/700/800

Kurzbeschreibung:

LPT 4.0
Serielle Ein-Ausgabe-Einheit mit einem USART 8251 welcher per Software auf 5, 6, 7 oder 8 Bit mit 1, 1,5 oder 2 Stopbit eingestellt wird. Die Übertragungsgeschwindigkeit ist in 8 Stufen (45,45; 50; 75; 110; 300; 600; 1200 und 2400 Baud) einstellbar. Durch Kombination der Baudratenschalter sind Geschwindigkeiten bis ca. 4800 Baud einstellbar. Weiterhin ist ein potentialfreier Relaiskontakt vorhanden, welcher vom Programm her angesprochen werden kann. Sämtliche Ein- und Ausgänge sind mit TTL-Buffern versehen.

LPT 4.1
Serielle Einheit wie LPT 4.0, jedoch zusätzlich mit einer parallelen Ein-Ausgabeeinheit. Dieses Parallelport besteht aus einer PIO (8255) mit 24 Ein-Ausgängen welche vom Programm her nach Wunsch definiert werden können. Alle 24 Anschlüsse sind an eine 25-polige D-Sub-Buchse herausgeführt. Weiterhin ist ein Socket für ein 4 kByte EPROM (2732) und die dazu notwendige Decoderlogik enthalten. Eine 44-polige Systembuchse für Erweiterungen (Floppycontroller ect.) ist ebenfalls enthalten.

LPT 4.2
Wie LPT 4.1 jedoch ohne die serielle Schnittstelle.

LPT 4.3
Nur EPROM-Socket und Decoderlogik. 44-polige Buserweiterung.

Mechanischer Aufbau:
Platine 75 x 180 mm, doppelseitige Leiterführung durchkontaktiert, 10 IC's, 1 REED-Relais, 1 25-polige D-Sub-Buchse für Parallelport, 1 5-polige DIN-Buchse für Seriellport, 1 44-polige Stiftleiste für Erweiterungen.

Lieferform:
MZ-80A: Platine wie beschrieben und abgebildet.
MZ-80K und MZ-700 oder MZ-800:
Platine in ein Gehäuse eingebaut mit Netzteil +5V und ca. 20 cm Buskabel zum jeweiligen Rechner.

I N T E R F A C E :

LPT 4.0 serielles Interface für MZ-80A (mit Gehäuse und Netzteil auch für MZ-700/80K), frei programmierbar mit USART 8251, Beschreibung siehe Datenblatt LPT. Steckplatine für MZ-80A
DM 298,00
Mit Gehäuse, Netzteil und Buskabel für MZ-700/MZ-80K DM 398,00

LPT 4.1 Universalinterface für MZ-80A (mit Gehäuse und Netzteil auch für MZ-700/80K), frei programmierbar mit PIO 8255 und USART 8251, zusätzlicher Socket für 4 kByte EPROM (z.B. unser P-DOS oder SOBS 8), 44-polige Stiftleiste zum Anschluß von Systemerweiterungen oder zum Anschluß unseres Floppycontrollers. Die Platine wird direkt auf die Hauptplatine des MZ-80A geschoben. Der AEU-Rahmen wird nicht benötigt! Steckplatine für MZ-80A
DM 369,00
Mit Gehäuse, Netzteil und Buskabel für MZ-700/MZ-80K DM 469,00

LPT 4.2 paralleles Interface für MZ-80A (mit Gehäuse und Netzteil auch für MZ-700/MZ-80K), wie LPT 4.1, jedoch nur parallele Einheit mit PIO 8255

Steckplatine für MZ-80A
DM 198,00
Mit Gehäuse, Netzteil und Buskabel für MZ-700/MZ-80K DM 298,00

LPT 6.3 Druckerinterface mit fest verdrahteter Logik für den MZ-80A (mit Gehäuse und Netzteil auch für MZ-700/800/80K), eingebautes EPROM zur Umcodierung vom SHARP Code auf Standard ASCII Code.

Beschreibung siehe Datenblatt LPT 6.3
für MZ-80A: Platine zum Aufstecken an den I/O-Anschluß für MZ-700/800/MZ-80K mit Gehäuse, Netzteil + Buskabel
DM 198,00
DM 248,00

FLPC-1 Controllerplatine mit Controller-IC NEC 765A, Anschluß von max. 4 Laufwerken 3", 3 1/2"; 5" oder 8" über DIL-Schalter sind alle gängigen Laufwerke der verschiedensten Hersteller einstellbar, 4/8 MHz Umschaltung, Zusatzsocket für 2 kByte RAM oder EPROM im Bereich D800H (nur für MZ-80A, der Socket wird beim MZ-80K und beim MZ-700 für das RAM benötigt)
DM 568,00

FDOS-EPROM zum Betrieb des Controllers FLPC-1, es muß bei der Bestellung das verwendete Laufwerk angegeben werden (Parameter-Seiten)
DM 198,00

Wichtig!

Sämtliche von uns angebotenen Interface benötigen den SHARP Erweiterungsrahmen AEU oder die I/O-Box nicht. Sie sind daher auch nicht geeignet, in original SHARP Peripherie einzubauen!

Druckerinterface LPT-700 mit eingebautem 2-80 Rechner, setzt den SHARP Code in ASCII oder Graphik um, mit Bufferspeicher DM 448,00

Peripherie, Zubehör: Preise inkl. 14% MwSt

BMC Matrix-Drucker BX-80, programmierbarer Zeichensatz Einzeladelaussteuerung, Einzelblatt und Endlospapier verwendbar, Traktoreinzug stufenlos verstellbar
DM 978,00

Drucker BX-80-S, wie BX-80 jedoch mit neuem Betriebsprogramm für MZ-700/800 Serie, eingebautes Interface für MZ-700/800 Serie, einschl. Druckerlabel zum MZ-xxx
DM 1.198,00

BMC Drucker BX-100, 120 Zeichen/Sek. sonst wie BX-80
DM 1.098,00

Drucker BX-100-S wie BX-100 jedoch mit neuem Betriebsprogramm für MZ-700/800 Serie, eingebautes Interface für MZ-700/800 Serie, einschl. Druckerlabel zum MZ-xxx
DM 1.298,00

Drucker EPSON RX-100F/T+ mit Traktor und breiter Walze
DM 1.598,00

Drucker EPSON FX-80 mit 2 kByte Buffer
DM 1.598,00

DIN A3 Farbplotter MP-1003 Sonderangebot RS-232-C E/A
DM 1.998,00

Sehr geehrte SHARP MZ-700-Freunde,

wie wäre es, wenn Sie Ihren Computer außer zum "bildschirmorientierten Programmieren" auch zum Steuern von Geräten "außerhalb seines Gehäuses" benutzen würden?

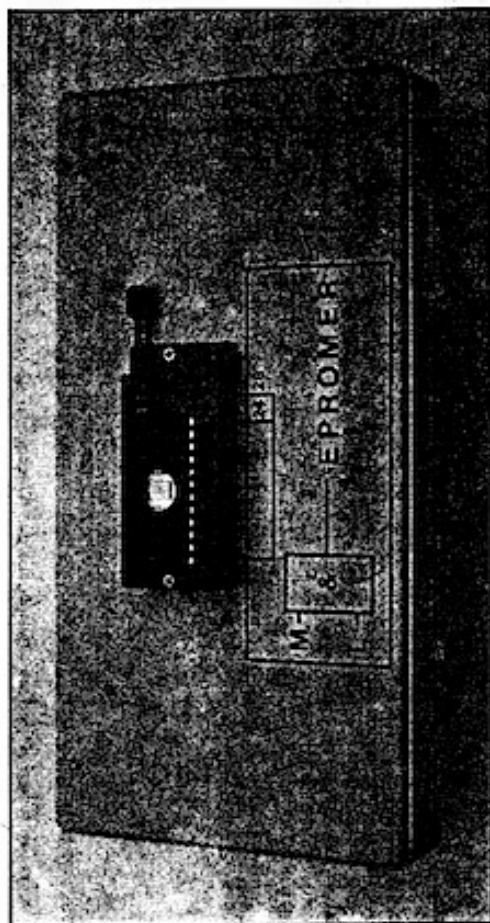
Sie könnten z. B. eine Heizungssteuerung aufbauen, eine Modelleisenbahn zum Laufen bringen, Lichteffekte in einer Diskothek steuern, mit einem Schrittmotor eine Positionierung erstellen, einen Kleinroboter zum Funktionieren bringen oder sogar ein Mikroprozessor-Entwicklungssystem aufbauen.

Um die oben beschriebenen Anwendungen zu ermöglichen, möchten wir Ihnen ein paralleles Ein-/Ausgabe-Interface vorstellen. Mit diesem Interface werden die Einsatzmöglichkeiten Ihres Computers beträchtlich erhöht. Das Interface hat 32-TTL-kompatible, auf Ein- bzw. Ausgabe programmierbare Leitungen und ist mit einer zusätzlichen Baugruppe auf 64 Leitungen erweiterbar. Zum parallelen Betrieb von z. B. Floppy und Ein-/Ausgabe-Interface ist das 50-polige Bus-System aus dem Ein-/Ausgabe-Interface herausführbar und kann mit einem geeigneten Stecker (in Kürze von uns lieferbar!) versehen werden.

Das Ein-/Ausgabe-Interface läßt sich auch in ein 64-poliges VG-Bus-System integrieren, um so Steuern und Regeln im Industriestandard (19"-Europakarten) zu ermöglichen (Preise auf Anfrage). An dieses Ein-/Ausgabe-Interface ist entweder a) Zusatzhardware der Fa. Maximini & Lichter anschließbar (siehe Prospekt) oder b) eigene Schaltungen. Der Anschluß von eigenen Schaltungen und das Programmieren des Interface ist mit Hilfe der sehr ausführlichen Bedienungsanleitung leicht möglich.

Als erstes Zubehör für das oben beschriebene Interface können wir Ihnen ein sehr aufwendiges EPROM-Programmiersgerät vorstellen. Die Bedienung ist menüorientiert, alle denkbaren Falscheingaben werden unterbunden. Mit diesem EPROM-Programmiersgerät lassen sich alle gängigen EPROM-Typen programmieren. Zusammen mit einem Assembler läßt sich so aus Ihrem Computer ein komplettes Entwicklungssystem aufbauen. Bitte entnehmen Sie weitere Einzelheiten dem beigelegten Prospekt.

-EPROM-Programmiersgerät- für Sharp MZ 700



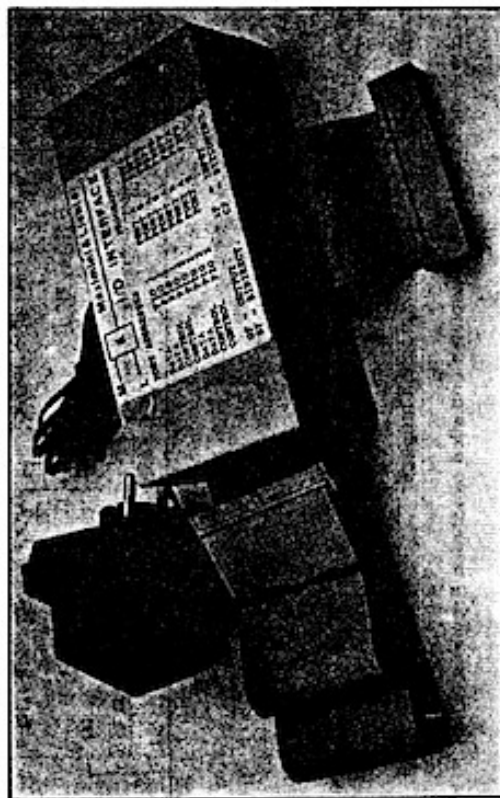
Fast jedes elektronische Gerät oder jede elektronische Anlage enthält heute EPROMs, in denen die Programme für Mikroprozessoren im Maschinencode abgespeichert sind. Außerdem müssen diese EPROM's bei Programmänderungen umprogrammiert bzw. im Servicefall überprüft werden. Dazu benötigt man ein EPROM-Programmiersgerät, eine Art modernen "Lötkolben", das es elektrische Verbindungen herstellt bzw. auflötet. Mit Hilfe des Sharp-Computers und der dazu lieferbaren Software (ASSEMBLER, MONITOR) ist es möglich, Maschinenprogramme in Assembler zu erstellen. Es fehlte bisher die Möglichkeit, diese Maschinenprogramme direkt in EPROM's abzuspeichern. Mit Hilfe des EPROM-Programmiersgerätes ist es möglich, die mit dem Assembler erstellten Maschinenprogramme in EPROM's abzulegen und diese dann in anderen Mikrocomputersystemen (z.B. Einplatinen-Steuerungscomputer) zu benutzen. Zusammen mit dem Sharp MZ 700 ist dann ein Entwicklungssystem für Mikrocomputer entstanden. Mit Hilfe des EPROM-Programmiersgerätes kann man jedoch auch Programme (z.B. Spiele) auf EPROM's abspeichern. Auf diese Programme kann man dann zugreifen, ohne über externe Speicher (Magnetband, Floppy) das Programm einladen zu müssen. Mit dem M & L-EPROM-Programmiersgerät lassen sich die folgenden EPROM-Typen programmieren:

a) INTEL (oder kompatibel):	2716 (2KByte)
	2732 (4KByte)
	2732A (4KByte)
	2764 (8KByte)
	27128 (16KByte)
	27256 (32KByte)
	2515 (2KByte)
	2532 (4KByte)
	2584 (8KByte)
b) TEXAS INSTRUMENTS (oder kompatibel):	

Software-Option:
2515 (2KByte)

- Paralleles Ein-/Ausgabe- Interface- für Sharp MZ 700

mit zahlreichen Erweiterungsmöglichkeiten



Endlich wird den MZ 700-Besitzern erstmalig die Möglichkeit geboten, mit dem parallelen Ein-/Ausgabe-Interface, den Computer auch zum Steuern und Regeln einzusetzen. Aufgrund der sehr ausführlichen Bedienungsanleitung ist die Programmierung des parallelen Interface relativ einfach durchzuführen; sie kann sowohl in BASIC als auch in Maschinensprache erfolgen.

Mit Hilfe dieses parallelen Interface ist es möglich, den Sharp MZ 700 vielfältig einzusetzen:

- a) Steuerung von z.B. Licht, Kleinstroboter, Modelleisenbahnen, Schrittmotoren, Produktionsabläufen;
- b) Meßwertaufnahme und -verarbeitung, Labordatenverarbeitung;
- c) Schnittstellennachbildungen;
- d) Programmieren von EPROM's;

Passend zu diesem Interface werden für die oben angegebenen Einsatzmöglichkeiten, Ergänzungsschaltungen angeboten bzw. vorbereitet, so daß nach und nach ein komplettes System entsteht.

Hauptbestandteil der Hardware sind zwei programmierbare Ein-/Ausgabe-Bausteine (PIO) des Typs Z80A PIO, also zum Z80A im PC genau passend.

Das Interface wird mit dem Erweiterungsanschluß des MZ 700 über ein 50-poliges Kabel verbunden. Die Stromversorgung dieser Baugruppe ist unabhängig von der Rechnerstromversorgung, so daß diese nicht überlastet werden kann.

Die 32 Port-Leitungen mit den jeweiligen Handshake- und Interrupt-Leitungen sind wieder über ein 50-poliges Kabel an eine 50-polige Buchsenleiste herausgeführt. Dadurch können die weiteren Zusatzgeräte neben und nicht hinter den Rechner platziert werden. An diese handelsübliche Buchsenleiste sind entweder vom Anwender erstellte Schaltungen oder M & L-Ergänzungen aufsteckbar.

Hardware

Die folgenden Ausstattungen des M & L-Epromers erhöhen den Bedienungskomfort zusätzlich:

- keine Belastung der Rechner-Stromversorgung;
- die für die Programmierung erforderlichen Spannungen (12,5 V, 21 V und 25 V) werden auf der Platine erzeugt;
- die Umschaltung auf die verschiedenen EPROM-Typen erfolgt softwaregesteuert (keine Umschaltung auf DIP-SWITCHES);
- Im nichtaktiven Zustand werden die Versorgungsspannungen und die Signale für den EPROM-Socket abgeschaltet; die EPROM's können so gefahrlos gewechselt werden;
- das Betriebsprogramm ist in Maschinensprache geschrieben (kein Laden von Basic);
- optische und akustische Anzeige der jeweiligen Fehlerart.

Das Steuerprogramm für das EPROM-Programmiergerät erleichtert mit Hilfe von "MENUES" und ggf. Fehlermeldungen (Ausschluß aller denkbaren Fälschungen) die Handhabung und ermöglicht so ein professionelles und rationelles Arbeiten.

Alle EPROM-spezifischen Parameter (Brennzeit, Brennspannungen, Speicherstrom) werden mit der Wahl der EPROM-Typen automatisch auf die erforderlichen Werte eingestellt.

- Bei den 5 Betriebsarten
 - A - Lesen EPROM in RAM
 - B - Leertests des EPROM
 - C - Programmieren des EPROM
 - D - Vergleichen RAM-Inhalt / EPROM-Inhalt
 - E - Duplizieren

Ist ein Zugriff sowohl auf Adressbereiche als auch auf einzelne Adressen möglich, so daß die gewählte Betriebsart auch für einzelne Adressen gilt.

Bei der Wahl der Betriebsart C - Programmieren des EPROM wird (um Bedien- und Programmierfehler weitestgehend vorzubeugen) automatisch vor der Programmierung für den gewählten Bereich ein Leertest durchgeführt. Wenn dieser Bereich leer ist (also jedes Byte = FFH), beginnt die eigentliche Programmierung, laufend auf dem Bildschirm angezeigt. Nach Beendigung der Programmierung wird automatisch ein Vergleichstest durchgeführt.

Im Vergleichsmodus "D" (nach Wahl oder automatisch nach jeder Programmierung) werden die Speicherzellen, bei denen der Vergleichstest negativ war, am Bildschirm angezeigt. In der Form: ROMADR, ROMINH, RAMADR, RAMINH. Dieser Modus besitzt damit auch die Möglichkeit EPROM-Chips miteinander zu vergleichen.

Ist die Betriebsart auf Duplizieren geschaltet, kann der Anwender EPROMs gleichen oder ungleichen Typs duplizieren (Adressbereich beachten!).

Vom Menü her wird jeder Schritt angezeigt, z.B. Quell-EPROM einsetzen, EPROM-Typ wechseln, usw.

Ein Verschieben von Speicherblöcken ist möglich.

Der EPROM-Inhalt kann auf Kassette oder Diskette zur Sicherheit abgespeichert werden.

Eigenschaften:

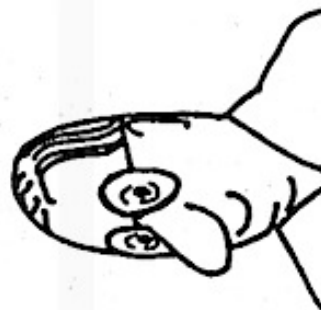
- einfaches Aufstecken auf programmierbare Ein-/Ausgabekarte
- keine externe Versorgungsspannungen, da Stromversorgung auf Ein-/Ausgabekarte
- automatisches Erzeugen der Programmierspannung (12,5V 21V 25V)
- Umschaltung der EPROM-Typen softwaremäßig
- hochwertiger 28-poliger Nullkraftstecker (Textool)
- komfortable Bediensoftware mit Menüsteuerung

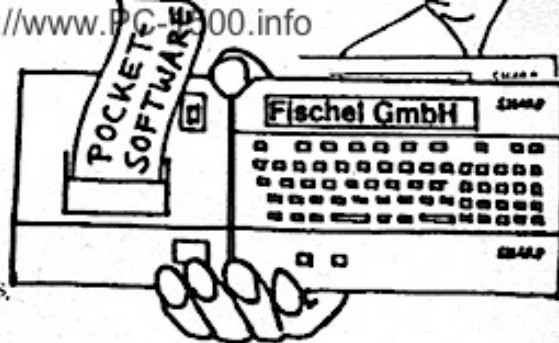
Preise inkl. 14% MwSt.

Platinenversion - DM 179,- Incl. Software auf Cassette

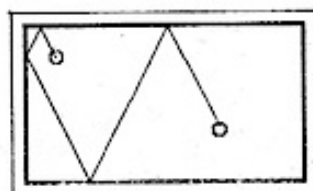
Gehäuseversion - DM 205,- Incl. Software auf Cassette

Unmöglichkeit und Änderungen vorbehalten.
Lieferung zu unseren Verkaufs- und Zulieferungsstellen.

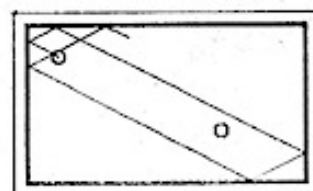




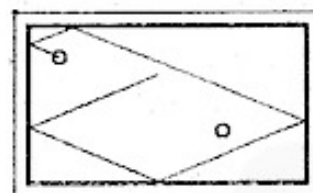
POOL-BILLARD



RICHTUNG: 63 Grad $\backslash$
KRAFT: 4 Einheiten
Sie haben 380 Punkte erreicht :



RICHTUNG: 22 Grad $\backslash$
KRAFT: 5 Einheiten
Sie haben 8 Punkte erreicht :

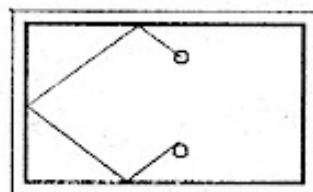


RICHTUNG: 22 Grad $\backslash$
KRAFT: 5 Einheiten
Sie haben 8 Punkte erreicht :

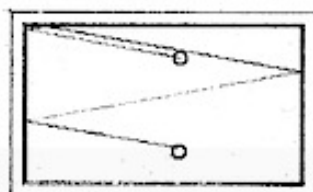
PC-1500/A POOL-BILLARD

Preis 50,-DM inkl. 14% MWS.

POOL-BILLARD

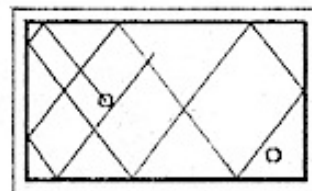


RICHTUNG: 36 Grad $\backslash$
KRAFT: 18 Einheiten
Sie haben 120 Punkte erreicht :

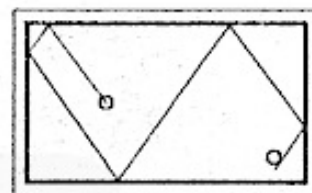


RICHTUNG: 18 Grad $\backslash$
KRAFT: 13 Einheiten
Sie haben 125 Punkte erreicht :

POOL-BILLARD

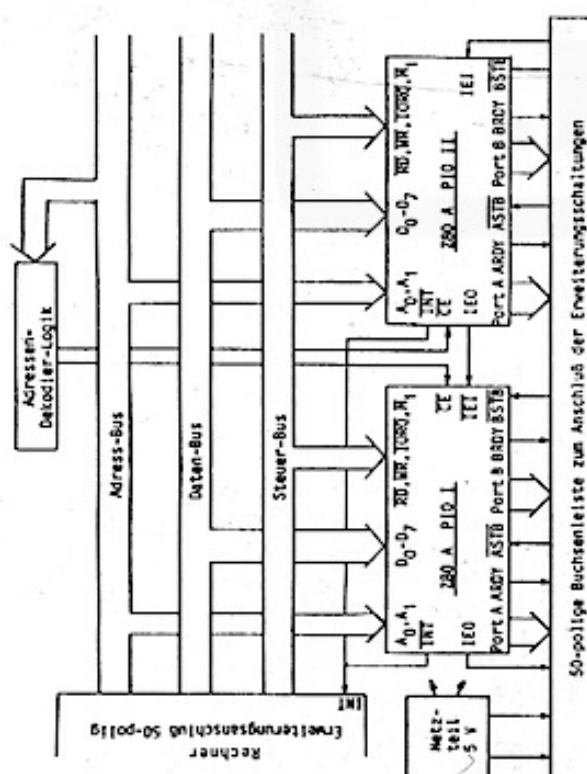


RICHTUNG: 52 Grad $\backslash$
KRAFT: 8 Einheiten
Sie haben 8 Punkte erreicht :



RICHTUNG: 54 Grad $\backslash$
KRAFT: 7 Einheiten
Sie haben 220 Punkte erreicht :

M i k r o c o m p u t e r



Eigenschaften:

- 32 programmierbare Ein-/Ausgabelineien (TTL-Level)
- Programmieren der Karten in allen Sprachen möglich
- ausführliche Beschreibung mit Beispielen und allen benötigten Programmhilfen
- keine Belastung der Rechner- Spannungsversorgung, da eigenes Netzteil
- vielfältige Ergänzungsmöglichkeiten und Ausbaumöglichkeiten zu einem professionellen Steuer- und Entwicklungssystem

Preise: (inkl. MWST)

Platinenversion: DM 198,-

Gehäuseversion: DM 218,-

Für das M & L-Parallel-Interface werden folgende Ergänzungsschaltungen angeboten bzw. sind in Vorbereitung:

1. EPROM- Programmiergerät für die 9 verbreitetsten EPROM-Typen
2. Simulationsgerät mit Schaltern und Leuchtdioden zur Simulation von Prozessen;
3. Analog-Eingabe 0...10 V mit 8-Bit-Auflösung;
4. Analog-Ausgabe 0...10 V mit 8-Bit-Auflösung;
5. Digital-Eingabe mit Relais 24 V / 2,5 A;
6. Digital-Eingabe über Optokoppler;
7. Schrittmotoren-Modul;

Unterstützung und Anleitung von Fachleuten
Lieferung zu unseren Verkaufs- und Zulieferungsstellen.

Do not sale this PDF !!!

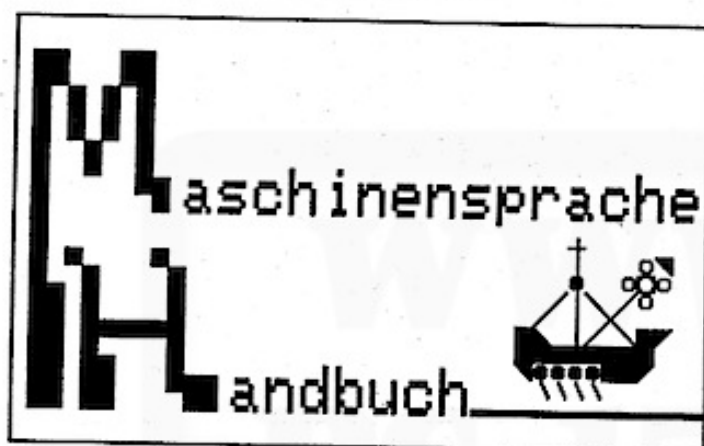
"Alles für Sharp-Computer" ist eine von der Firma SHARP-Electronics (Europe) GmbH völlig unabhängige Zeitschrift! Urheber und Herausgeber ist allein die Fischel GmbH, Berlin.

Neue Produkte

Preis 49,-DM inkl. 7% MwSt.

SHARP MZ-700 MZ-800 Serie

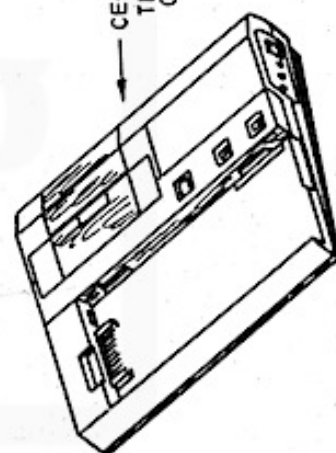
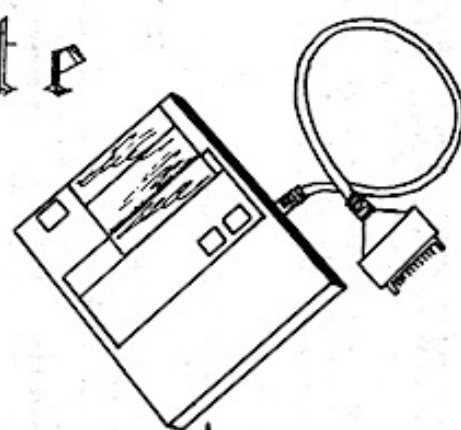
Personal Computer



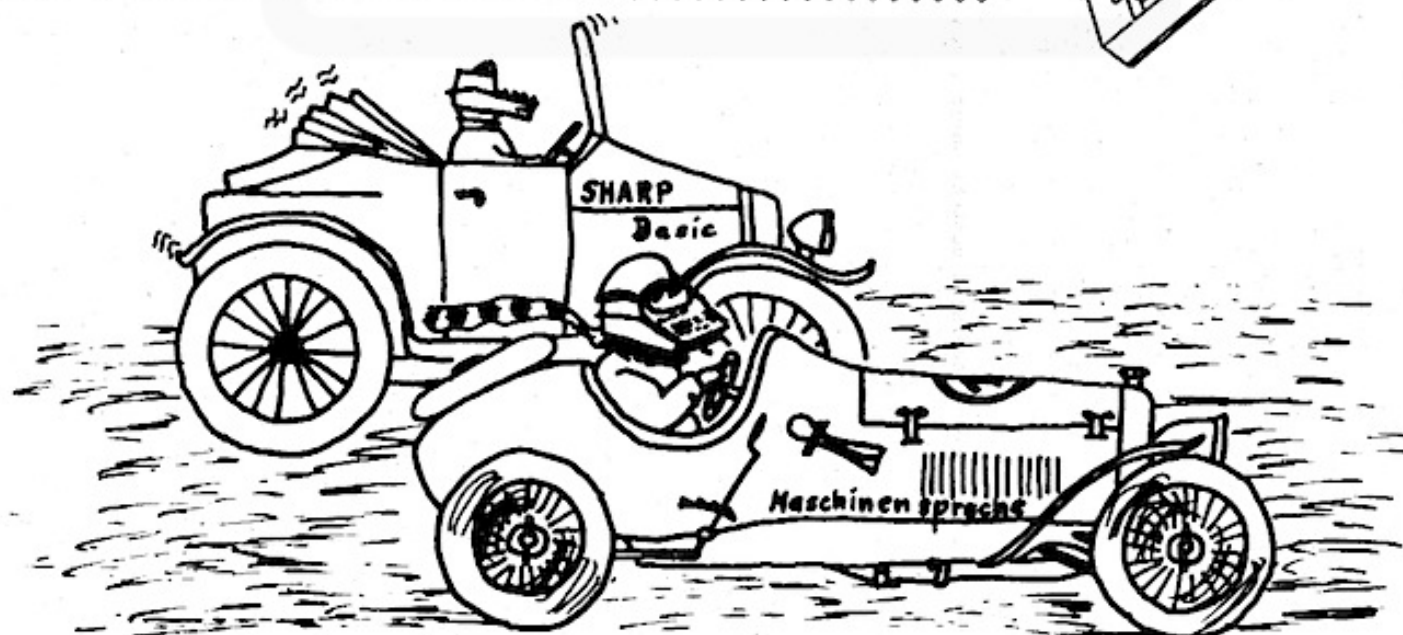
ISBN 3-924327-07-6

FISCHEL GmbH

Prof. U. Ehm



SHARP
CE-129P + CE-126P
Thermoprinter and
Cassette-Interface



schneller mit Maschinensprache

Do not sale this PDF !!!

Michael Dumont

 Blankenburgstr. 16
 Haldern
 4242 Rees 3,

Sehr geehrter Herr Fischel!

Das nebenstehende Prg errechnet aus den eingegebenen Zutaten und deren Mengen die Gesamtkalorien bzw. -joules.

BESCHREIBUNG: Um nun möglichst viele Zutaten unterzubringen, und bedingt durch die Speicherkapazität des PC 1251, ist das eigentliche Prg klein gehalten worden. Die Eingabetextlänge von F% (1) habe ich auf 17 Zeichen festgelegt; läßt sich aber bis 28 Zeichen erweitern. Das DATA-Feld ist mit X=8 und Y=8 also auf 64 Feldern begrenzt. Durch eine höhere Speicherkapazität o. Verwendung eines größeren Rechners läßt sich die DATA-Menge erheblich vergrößern. Außerdem kann man dann das eigentliche Programm wesentlich komfortabler gestalten. - Weiterhin habe ich auf Eingabefehlerkontrollen u. v. a. verzichtet.

Um die Lesedauer des Rechners gering zu halten, setze ich bei gefundener Zutat in der DATA-Zeile mit der Zeile 90 I und J gleich dem Endwert von X und Y, so daß der Rechner nicht immer die gesamte Datei durchliest.

ANWENDUNG: Bei der Eingabe der Zutat ist der Text so einzugeben, wie er in der DATA-Zeile steht, allerdings ohne "Gänsefüßchen". Bei der Mengenabfrage wird die Gewichtsmenge der Zutat in Gramm eingegeben. Wird bei der Frage "ALLES ?" mit "J" geantwortet, werden die Gesamtkalorien bzw. -joules angezeigt. Ansonsten ist immer mit der ENTER-Taste weiterzudrücken, und die nächste Zutat wird abgefragt.

HINWEIS: Die Zahlenwerte in den DATA-Zeilen, die sich jeweils hinter der entsprechenden Zutat befindet, sind einer Broschüre der BEK entnommen. Für die Richtigkeit der Kalorieneinheiten pro Gramm übernehme ich keine Gewähr.

Änderungen: Die Zusammenstellung der Zutaten ist willkürlich gewählt, und kann jederzeit geändert werden. Bei Vergrößerung o. Verkleinerung des Programmes bitte auch die Feldgröße (X und Y) entsprechend ändern.

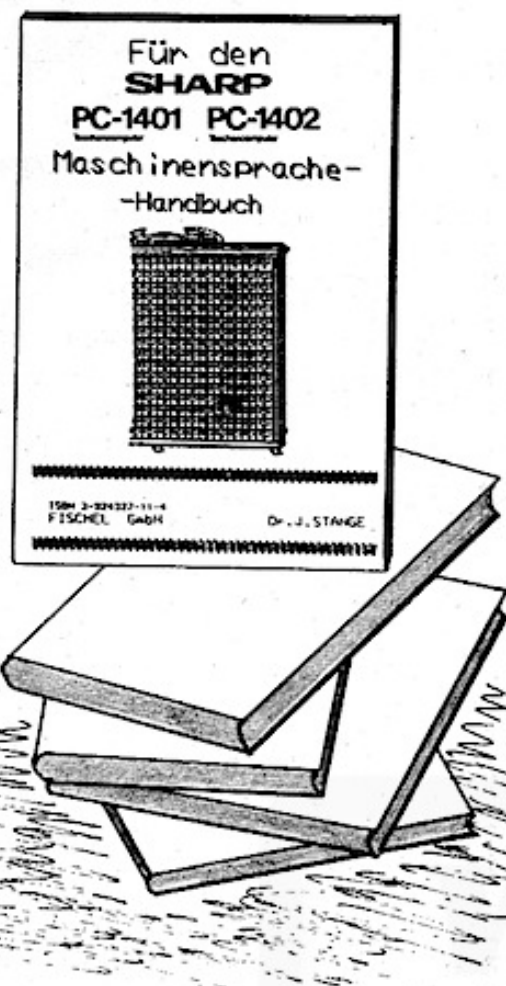
Diätprogramm mit SHARP PC

M. Dumont, Blankenburgstr. 16, 4242 Rees 3,

```

1: "D": CLEAR
5: N=8: Y=8: G=0: V=0
10: DIM B$(X,Y), C(X,Y), F
    F$(1)=17
30: INPUT "ZUTAT:"; F$(1)
40: INPUT "MENGE:"; M
50: FOR I=1 TO X
60: FOR J=1 TO Y
70: READ B$(I,J), C(I,J)
90: IF F$(1)=B$(I,J)
    THEN LET V=M*C(I,J):
    G=G+V: G=G*4.2: I=X: J=Y
110: NEXT J
120: NEXT I
130: INPUT "ALLES ?"; A$
140: IF A$="J" THEN 160
150: RESTORE : GOTO 30
160: PRINT "GESAMTKALORIE
    N: "G
170: PRINT "GESAMTJOULES:
    "G*4
500: DATA "SCHWEINE MAGER
    ", 1.43, "SCHWEINE FET
    T", 3.89, "RINDFLEISCH
    ", 2.38
501: DATA "KALBFLEISCH",
    .93, "HAMELFLEISCH", 2
    .46, "REHFLEISCH", .93
502: DATA "HASE", .99, "BRA
    THUN", 1.87, "ZUNGE",
    1.67, "KINDERLEBER", 1
    .31
503: DATA "SCHINKEN GEKOC
    HT", 2.74, "SCHINKEN R
    OH", 3.44, "SCHINKEN G
    ER", 3.44
504: DATA "SPECK DURCHW.",
    16.05, "SPECK FEIT", 7
    .7, "BRATWURST", 3.75,
    "BLUTWURST", 4.63
505: DATA "LEBERWURST MAG
    ER", 2.68, "LEBERWURST
    ", 4.4, "DOSENHUEHNERSTU
    CH", 2.39
506: DATA "FRANKE. WUERST
    CHEN", 2.5, "SERVELATE
    R WURST", 4.54, "LEBERKA
    ESE", 2.71
507: DATA "FLEISCHWURST",
    3.24, "NETTWURST", 5.3
    , "SCHWARTENHAGEN", 2.
    48, "HALL", 2.09
508: DATA "WEILBUTT", .88,
    "HERINGSFILET", 2.22,
    "ROTBARSCH", .55, "FOR
    ELLE", .52
509: DATA "HECHT", .49, "KA
    RPFEN", .65, "BUCKELIN
    G GER.", 1.46, "HATJES
    HERING", 2.85
510: DATA "OELSARDINEN", 2
    .4, "THUNFISCH IN OEL
    ", 3.04, "TRINKMILCH",
    .66
511: DATA "BUTTERMILCH",
    .36, "SAHNE", 1.27, "EIS
    CREME", 2.05, "HARTKAE
    SE", 2.79
512: DATA "MAGERQUARK", .8
    3, "VOLLK. JOGHURT",
    .74, "MARGARINE", 7.61
513: DATA "OLIVENOEL", 9.2
    7, "BUTTER", 7.75, "KLA
    RE SUPPE", 1.4, "LINSE
    NSUPPE", 1.6
514: DATA "GEBUND. SUPPE",
    1.5, "FLEISCHSUPPE",
    .17, "VOLLREIS", 3.71
    , "HAFFERFLOCKEN", 4.02
515: DATA "CORNFLOCKES", 3.
    88, "NUDELN", 3.9, "WEI
    ZENMEHL", 3.7, "ROGGEN
    BROT", 2.53
516: DATA "BROETCHEN", 2.7
    8, "KNAECKEBROT", 3.8,
    "ZWIEBACK", 4.03, "KAR
    TOFFELN", .85
517: DATA "POMMES FRITES",
    2.7, "ZUCKER", 3.94,
    "HONIG", 3.05
  
```

Do not sale this PDF !!!



Buchbesprechung :

" FÜR DEN SHARP PC-1401/02 MASCHINENSPRACHE-HANDBUCH " von Dr.J.Stange

Preis 49,-DM inkl.7%Mwst.

ISBN 3-924 327-11 -4

Erneut präsentiert sich ein Buch aus der bereits umfangreichen Serie der Maschinensprache-Handbücher. Diesmal für den Sharp-Pocket-Computer PC-1401/02.

Daß auch diese Computer in Maschinensprache programmiert werden können, dürfte mittlerweile den meisten PC-14...-Anwendern bekannt sein. -Desgleichen die Vorzüge der Maschinensprache.

Dieses im handlichen DIN-A5-Format herausgegebene Buch bietet den Schlüssel dazu.

Nach einer allgemeinen Einleitung, sowie der Erörterung der im Computer verwendeten Zahlensysteme und der Beschreibung des Mikroprozessors werden die verschiedenen Befehle ausführlich behandelt. Es folgt eine tabellarische Kurzbeschreibung dieser Befehle.

Die im Buch aufgeführten Programmbeispiele sind eingehend beschrieben und vermitteln dem Anwender einen praktischen Einstieg ins "Eingemachte".

Bei der Systemuntersuchung, der Beschreibung der Tasten-, Anzeige- und Ausdruckroutinen weist der Autor stets auf die Unterschiede zwischen der neuen ROM-Version des PC-1401/02 und der alten ROM-Version des PC-1401 hin.

Im Anhang befinden sich schließlich das Befehlslisting des Kern-ROM von &0000 bis &1FFF sowie ein umfangreiches Stichwortverzeichnis.

PL

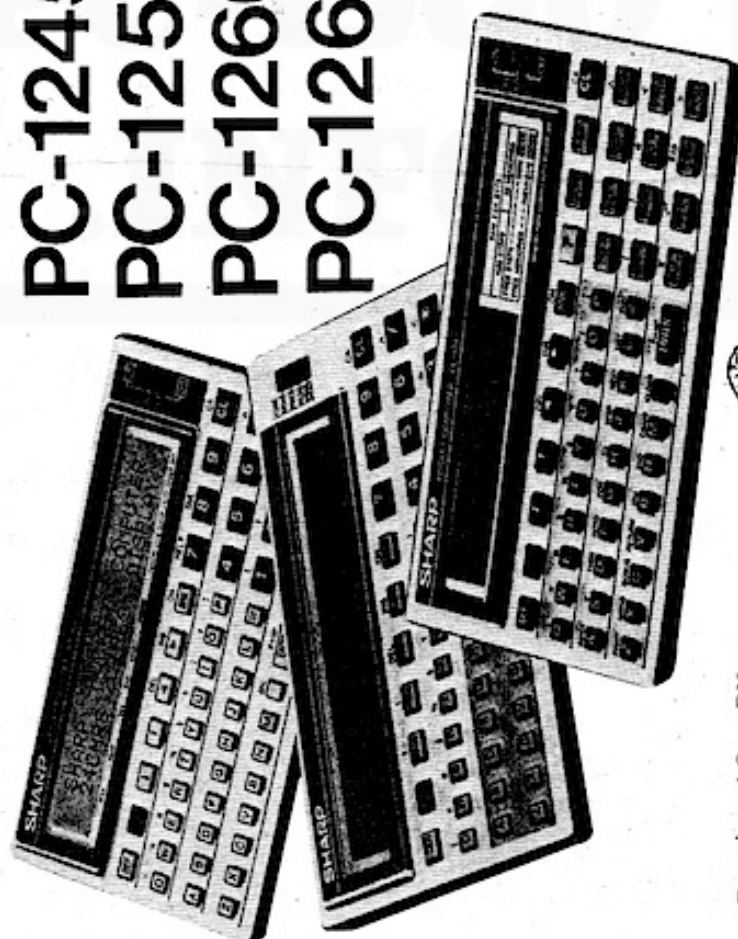
INGO LAUE

-Anwendungs-Handbuch-

zum **SHARP**

Taschencomputer

PC-1245
PC-1251
PC-1260
PC-1261



Preis 49,-DM
inkl.7%Mwst.



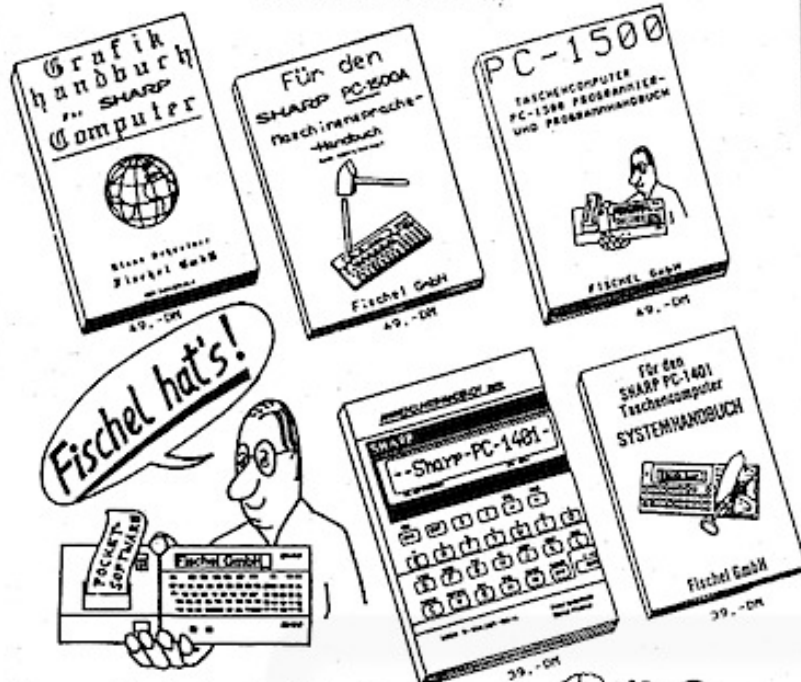
INGO LAUE

FISCHEL GmbH

ISBN 3-924 327-14-9

Die Unternehmensberatung für SHARP-Computer

Alle Preise incl. 7% MwSt.



User-Club Deutschland
DURCH INFORMATION VORN

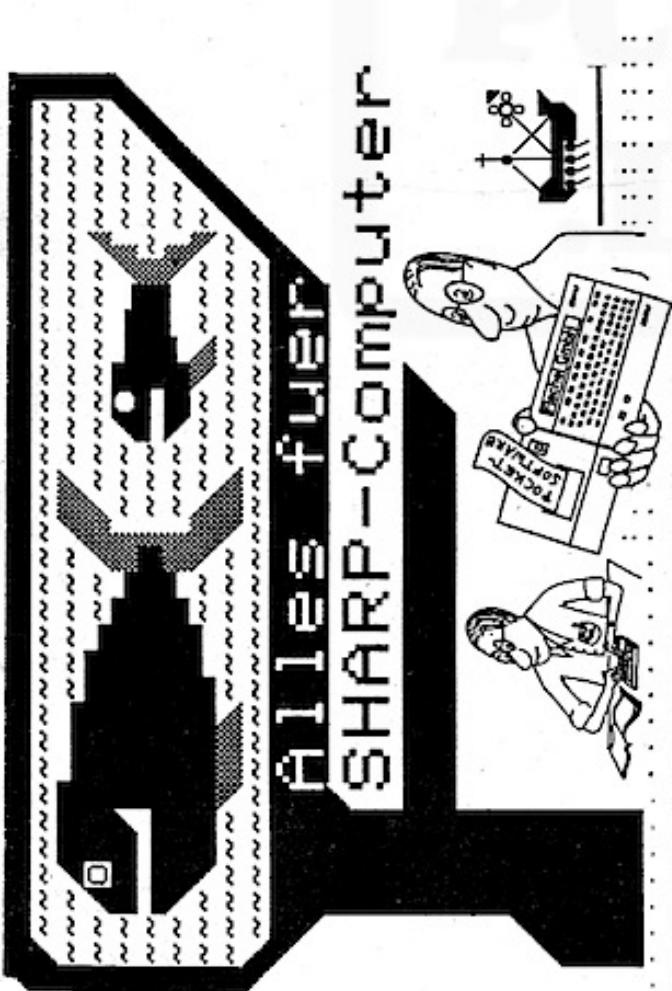
DIPL.-WIR. R. FISCHEL-KAISER-FRIEDRICH-STR. 54A D-1000 BERLIN 12 TEL. (30) 123 86 29



ALLE VERGANGENEN HEFTE SIND NOCH LIEFERBAR !!
Bitte siehe Bestellzettel für ein Abonnement. 49,- DM pro Heft

FISCHEL GMBH, KAISER-FRIEDRICH-STR. 54a, 1000 BERLIN 12
Tel. 030/3236029

SHARP



Bestellschein Bitte vollständig und lesbar ausfüllen,
unterschriften und einreichen an Fischel GmbH,
Kaiser-Friedrich-Str. 54a, D-1000 Berlin 12

- ☐ Ich abonniere die Zeitschrift "Alles für Sharp Computer" von der nächsten erreichbaren Ausgabe an (Preis pro Jahr 72 DM, Ausland 84 DM, Luftpostzuschlag 12 DM).
- ☐ Das Abonnement verlängert sich um ein Jahr zu den dann jeweils gültigen Bedingungen, wenn es nicht 2 Monate vor Ablauf schriftlich gekündigt wird.
- ☐ Ich bestelle folgende schon erschienene Exemplare von "Alles für Sharp Computer" (Stückpreis 6 DM, Ausland 7 DM):
Heft Nr. 1 Alle Preise incl. 7 % MwSt.

Der Gesamtbetrag von DM

- ☐ liegt bar bei
- ☐ liegt als Verrechnungsscheck bei (schnellste Erledigung)
- ☐ wurde am auf das Postgirokonto der Fischel GmbH, Kontonr. 461533-103, BLZ 10010010, Postgiroamt Berlin Überwiesen (Bearbeitung nach Zahlungseingang)
- ☐ liegt (nur bei kleineren Beträgen) in Briefmarken oder internationalen Antwortscheinen bei.

Name, Vorname
Straße
PLZ/Ort
Datum, Unterschrift
Mir ist bekannt, daß ich diese Bestellung innerhalb von 8 Tagen bei der Bestelladresse widerrufen kann. Zur Wahrung der Frist genügt die rechtzeitige Absendung. Ich bestätige dies durch meine zweite Unterschrift.
Datum, Unterschrift

SHARP PC-2500

Taschencomputer

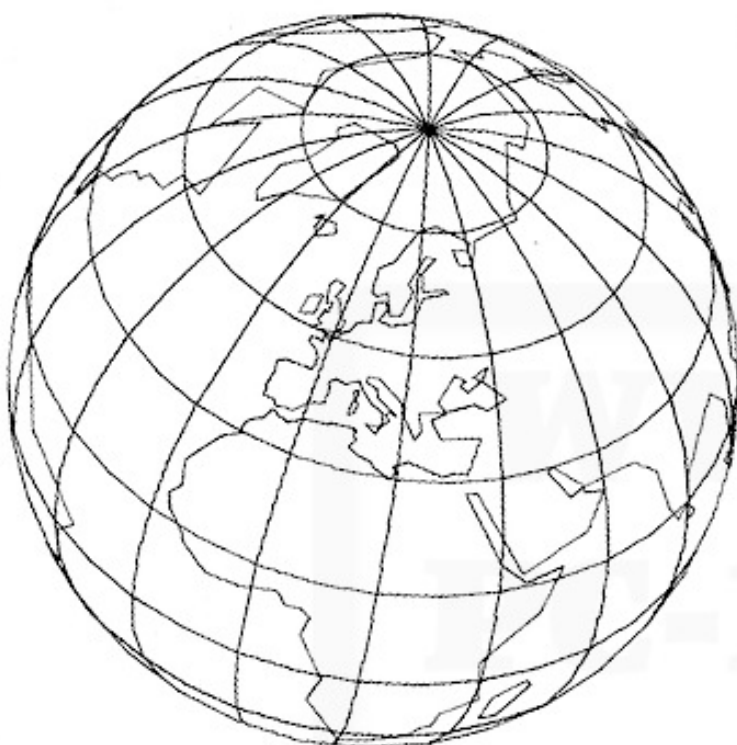
Eingebaute Software macht den PC-2500 zu einem leistungsfähigen Hilfsmittel für den kaufmännischen Einsatz unterwegs.



SHARP PC-2500 SOFTWARE



Demonstrationsdruck des Schreibmaschinenprogrammes.



**WELTKUGEL
SCHREIBMASCHINE**

BREIT—und FETTSCHRIFT

**BREIT—
und
FETTSCHRIFT**

HARDCOPY DER ANZEIGE

JAHRESKALENDER

JAHRESKALENDER
für das Jahr 1986

ANZEIGE BEI DER EINGABE

CASSETTE 39,-DM / LISTING 29,-DM

Bestellungen bitte an die FISCHEL GmbH, Kaiser-Friedrich-Straße 54a, 1000 Berlin 12

Preis pro Programm inkl. 14% MwSt. (Siehe auch Heft 13)

Beispielausdruck mit dem Programm
"Schreibmaschine" ! 0000000000000000

Betr.: Programm 'SCHREIBMASCHINE'

Sehr geehrte(r) Interessent(in),
ich möchte Sie mit diesem Demonstrations-
Ausdruck auf die Möglichkeiten des Pro-
gramms 'SCHREIBMASCHINE' für den SHARP PC-
2500 hinweisen.

Sie können Texte mit deutschen Umlauten
in allen vier Farben schreiben.

-- 000000 --

Der Farbwechsel ist auch innerhalb einer
Zeile möglich.

Ihre Texte können in 40 oder 80 Zeichen
Pro Zeile geschrieben werden.

Beim Erreichen des 30. bzw. 70. Zeichens
ertönt ein akustisches Signal zum leicht-
eren trennen der Wörter am Zeilenende.

In der Anzeige wird die Schriftfarbe und
die Anzahl der Zeichen Pro Zeile darge-
stellt.

HARDCOPY DER ANZEIGE:

**BLAU
GRÜN
ROT**

**SCHREIBMASCHINE
(C) C. Samoticha**

40/80 Zeichen

Sharp Microcomputer
..... Fischel GmbH
Kaiser-Friedrich-Str. 54 a
D - 1000 Berlin 12
..... Tel. 030 / 323 60 29
Mo - Fr 10 - 18.00, Sa - 14 h

Geschenk-Ideen

SHARP

Do not sale this CD !!!

Alles für Sharp Computer Heft 14 Seite 59

Ärger beim Eingeben . . . ?

Neu: Sonderzeichen werden durch Klartext ersetzt!

Um Ihnen die Eingabe unserer Listings zu erleichtern, werden künftig die gesamten C-64-Listings ohne Steuerzeichen abgedruckt. (gilt nicht für diese Ausgabe!!)

Die gesamten Steuerzeichen werden in unseren LISTINGS durch Klartext, welcher die Taste kennzeichnet, ersetzt. Eine Tabelle der Tastenfunktionen finden Sie in jedem Tronic-Magazin.



**... erhältlich
überall im
Zeitschriften-
handel!**

Checksummer für Commodore 64 und Schneider

Checksummer C 1.0 ist ein Prüfsummen-Programm, das die Eingabe von Programm-Listings zum wahren Vergnügen macht. Tippfehler werden schon während der Eingabe der einzelnen Programmzeilen erkannt. Dieses System zusammen mit einem neuen Druckverfahren gewährleisten, daß unsere Listings fehlerfrei eingegeben werden. (Weitere Systeme sind in Vorbereitung).

- Erleichterung
- Fehlerfreiheit
- zweispaltiger Druck
- Absolute Zeilenkontrolle

Checksummer

ab sofort in allen

Tronic-Magazinen!