

PC-1500 ZEITUNG

Taschencomputer

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

PC-1500 als Sieger beim
amerikanischen Volkslauf

ERGEBNISLISTE:

Cross Country Race
final results
day: 06/18/83

1983 PC1500-Pocket-Computer
List 3, - DM

Place StNr Time
1 59 0h36:03
Name: CLOUSE
born: 61 Sex: M
Club: BRR

women
18 to 30 years
name:
RINGSMUTH
club: -
time: 0h52:26

Place StNr Time
2 468 0h36:36
Name: THURSTON
born: 52 Sex: M
Club: -

men
31 to 40 years
name:
THURSTON
club: -
time: 0h36:36

Place StNr Time
3 51 0h37:48
Name: RODGERS
born: 59 Sex: M
Club: -

name:
BRASCH
club: -
time: 0h38:51
name:
COLON
club: TCA500
time: 0h39:59

Place StNr Time
4 377 0h38:06
Name: BARTLETT
born: 69 Sex: M
Club: BRR

name:
YOUNG
club: BRR
time: 0h42:00

Place StNr Time
5 54 0h38:21
Name: COLON
born: 51 Sex: M
Club: TCA500

name:
SUENSEN
club: -
time: 0h43:29

Place StNr Time
6 465 0h38:51
Name: BRASCH
born: 52
Club:

PC-1500 Club Berlin:
Treffen am 20.7.83
19:30 Clublokal
Tel. 030/3236029

Inhalt:

	Seite
1. Hard-Copy	2
2. Ergänzung zur "Schreibmaschine"	3
3. TV-Interfase - Stand -	4
4. Passwort	4
5. Was der PC-1500 noch nicht kann!	4
6. Programmieren in Maschinensprache von Bernd Rüter	
Vorwort, Teil I und II	5
7. Aufrufe	13
8. Lesenswertes	14
9. PC-1500 Schulung: DATA, Arbeiten mit 80 Zeichenmatrikdrucker	15
10. 2. Zeichensatz	18
11. Technical Reference Manual	27
12. CE-0815	28
13. Tips und Tricks	29
14. 1. Computerbörse Berlin	30

Das Fahrrad



-- PC-1500 --

PC-1500 Plot
Peter Tschötschel, Berlin

HARD-COPY

Hierbei handelt es sich um ein Unterprogramm zur nahezu maßstabgerechten Darstellung des PC-1500-Displays samt momentanen Inhalts auf dem Plotter.

Durch das Verwenden hoher Zeilennummern wird das Zuladen von Cassette durch "MERGE" zum Hauptprogramm und ein problemloser Programmablauf ermöglicht.

Aufgerufen wird "HARD-COPY" vom Hauptprogramm durch GOSUB "H". Es erscheint zunächst zusätzlich zum bisherigen Displayinhalt die "BUSY"-Anzeige. Nun kann durch drücken von "J" oder "N" entschieden werden, ob ein Ausdrucken erfolgen soll oder nicht. Falls nicht, erfolgt der Rücksprung in das Hauptprogramm sofort, sonst erst nach Druck.

Folgende Variablen werden verwendet und müssen eventuell ausgetauscht werden: A,L,W\$ und als Laufvariablen: I,J,K

BUSY	DEG RUN
* * HARD-COPY * *	

```

65271: "H"W$=INKEY$
      : IF W$=""
      GOTO "H"
65272: IF W$(">")="J"
      RETURN
65273: GRAPH : LINE
      (40,-300)-(4
      0,0)-(90,0)-
      (90,-300):
      ROTATE 1:
      CSIZE 1
65274: GLCURSOR (78
      ,-8):LPRINT
      "BUSY":
      GLCURSOR (78
      ,-255):
      LPRINT "DEG"
      :GLCURSOR (7
      8,-315):
      LPRINT "RUN"
      :GLCURSOR (0
      ,-4):SORGN
65275: FOR K=0 TO 15
      5:A=POINT K:
      IF A=0GOTO 6
      5279
65276: FOR J=6 TO 0
      STEP -1: IF A
      <2^JGOTO 652
      78
65277: A=A-2^J:L=7-
      J:FOR I=1 TO
      3:LINE (L*4+
      I+40,-K*4)-(
      L*4+I+40,-3-
      K*4):NEXT I
65278: NEXT J
65279: NEXT K:LINE
      (40,-295)-(4
      0,-624)-(90,
      -624)-(90,-2
      95):GLCURSOR
      (78,-545):
      LPRINT "I":
      RETURN
  
```

WINFRIED FELDKAMP
DIPLOM-WIRTSCHAFTSINGENIEUR

7000 STUTTGART 80
BRÜHLBACHWEG 16
TELEFON 74 47 34

PC-1500

Zeitung, Heft 2
vom Juni 1983

Fischel GmbH
Kaiser-Friedrich-Straße 54 a
1000 Berlin 12

Programm:

>SCHREIBMASCHINE<

Sehr geehrter Herr Fischel!

FEHLER in Ihrer
Veröffentlichung
(Listings) — — —

Zunächst meinen Dank für die schnelle Zustellung der diversen Unterlagen. Ich vermisse jedoch das Textverarbeitungsprogramm mit deutschen Umlauten und Abspeicherungsmöglichkeit der eingegebenen Texte. Sollten Sie derartiges verfügbar haben, bitte ich um Zusendung.

ZEILE:

160"11" statt "10"

Das in der PC-1500-Zeitung abgedruckte Textprogramm "Schreibmaschine" enthielt einige Fehler, die nebenstehend berichtigt sind. Durch die dort aufgeführte, von mir vorgenommene Ergänzung zur Silbentrennung erhöht sich m.E. die Anwenderfreundlichkeit.

nach
350 Zeile 360
statt 60

410 T1 statt 1

Nachteil des Programms ist das schlechte Ansprechverhalten zwischen Eingabe und Druck. Eine Anwendung empfiehlt sich daher nur für Kurztex-te.

440 vor GOTO fehlt
"48"

Mit freundlichen Grüßen

Um vor Zeilenende eine Silbentrennung zu ermöglichen, wurde eine Abfrage entsprechend der Zeilenbreite ergänzt.



```
305: IF CZ<6AND L>1
      70BEEP 1,1,200
      0
```

```
308: IF CZ>=6AND L>
      160BEEP 1,1,20
      00
```

Damit wird vor Zeilenende ein akustisches Zeichen gegeben!

TV - Interface

- Stand 25.06.83 -

We also confirm your exclusive agency in Germany for our modules ABS - 200 and ABS - 201 and our TV - Interface. As soon as the TV - Interface will be ready for production we will give you the price and all information necessary for distribution.

In Sweden ABS-200 sells for DM 315,- and ABS-201 for DM 409:-

Your price will be DM 165,- for ABS - 200 and
DM 215,- for ABS - 201.

We will send you an English translation for the manuals and leaflets within the next few weeks.

Thank you for your hospitality.
We are looking forward to meet you again

Yours faithfully,
Mr. Hansen

Passwort bei PC-1500:

Dieses kleine Prgm. ermöglicht es, den PC-1500 Auszustellen und beim Anstellen mit ON wartet der Computer auf den Druck einer bestimmten Taste. Das Prgm. ist in Maschinensprache geschrieben und kann als Reserve-Tasten-Belegung abgelegt werden. Das Prgm. wird in den Speicher für Z§ eingeschrieben.

Tasten-Belegung:

POKE&77F0,&BE,&E3,&3F,&B7, (Tastenwert) ,&99,&07,&9A

und

CALL&77F0

Wenn Sie jetzt die erste Tastenbelegung ausgeführt haben, drücken Sie die zweite. Nach dem Einschalten durch ON drücken Sie nach Erscheinen der Anzeige, die von Ihnen festgelegte Taste z.B. A = 65, MODE = 15. Danach ist der Computer wieder betriebsbereit.

So schützen Sie Ihren Computer vor unberechtigtem Benutzen.

Was kann der PC-1500 noch nicht:

derzeit keine vollständige Beschreibung auf dem Markt (wie bei VC-20) .

derzeit keine große anschließbare Tastatur.

derzeit keine andere Programmiersprachen auf dem Markt.

Programmieren in Maschinensprache

Vorwort

Wir wollen hier versuchen, das Programmieren in Maschinensprache allen PC-1500-Benutzern zu erläutern.

Hilfreich bzw. notwendig wäre eine Liste der Maschinensprache-Befehle, wie sie z.B. im Systemhandbuch von RVS vorhanden ist. Wir wollen uns hier auch an diese in dem Buch aufgeführte Schreibweise der Befehle halten.

Hilfreich wäre weiterhin der HEXMONITOR von RVS, oder der Assembler. Allgemein kann man zum Programmieren in Maschinensprache sagen, daß man sehr vorsichtig sein soll, da man die entstandenen Prgm.e nicht einfach mit BREAK abbrechen kann, wie ein BASIC-Programm, denn es gibt keinen BASIC-Interpreter mehr, der sonst auf solche Eingaben reagiert.

Was mache ich, wenn ich ein Maschinenspr.-Prgm. habe, das nicht mehr zu stoppen ist ?

-Wenn Sie Ihren PC-1500 noch an den Drucker angeschlossen haben, können Sie den Rechner aus dem Drucker ziehen, und beim Einstecken in den Drucker stellt sich der Rechner aus. Nach dem Anschalten kann NEW 0 ? erscheinen, diesen Hinweis können Sie oft unbeachtet lassen, da das BASIC-Prgm. oft noch vollständig vorhanden ist. Überprüfen Sie dieses aber und sagen Sie wenn nötig doch NEW 0.

-Wenn Sie Ihren PC-1500 ohne Drucker betreiben, können Sie Ihr Programm schlecht anhalten, ohne alle Daten zu verlieren, denn Sie müssen ALL RESET drücken.

Ich hoffe, Sie sind nun für den Kampf gegen Ihren PC-1500 gerüstet.

Programmieren in Maschinensprache
Teil 1

Uns steht im PC-1500 eine 8-Bit-CPU zur Verfügung. Das bedeutet, daß wir mit 8-Bit-Werten arbeiten. Mit 8 Bit kann man 256 verschiedene Bit-Kombinationen erzeugen, die ich als Dezimalzahlen (0-255) oder als Hexadezimalzahlen (&00 - &FF) darstellen kann. (Bedienungsanleitung S. 19)

Ein 8-Bit-Wert, wir nennen es jetzt Byte ist folgend aufgebaut:

$B_7 B_6 B_5 B_4 B_3 B_2 B_1 B_0$ $B_7 = \text{MSB} = \text{höchstwertiges Bit, da es als 128 zählt.}$
 B_2 z.B. zählt nur als 4.

In der CPU des PC-1500 stehen uns mehrere solcher 8-Bit-Register zur Verfügung:
A Akkumulator, oder allgemeines Rechenwerk.

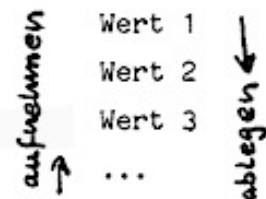
XH, XL, YH, YL, UH, UL

Diese letzten 6 Register werden paarweise zu 16-Bit-Registern zusammengefaßt, so daß man mit einem Doppelbyte 65536 Kombinationen erzeugen kann, genug um jeden Speicherplatz (64 K = 64 x 1024 = 65536) mit einer Nummer anzusprechen, zu adressieren. Man nennt diese neuen Register X, Y, oder U, und weil sie einen Speicher direkt ansprechen können auch Pointer-Register.

Zusätzlich haben wir noch andere Register, die feste Aufgaben haben:

P (16 Bit) Programmzähler, er zeigt auf die Speicherstelle, in der der nächste auszuführende Befehl steht.

S (16 Bit) Stapelzeiger, er zeigt auf die erste frei Stelle unseres Stapels. Dieser Stapel kann irgendwo in den Speicher gelegt werden und ist eine hilfreiche Einheit, denn man kann auf ihn Daten ablegen und in umgekehrter Reihenfolge wieder aufnehmen:



S zeigt auf diese Stelle → frei

T (9 Bit) Timer, Dieses Register wird mit jedem Takt weitergezählt.

F (5 Bit) Flag-Register, in diesem Register werden Zustände angezeigt, die nach Operationen auftraten.

Z Zero-Flag: Das Ergebniss einer Operation ist Null.

C Carry-Flag: Bei einer arithmetischen Operation werden zur vollständigen Darstellung des Ergebnisses mehr als die verfügbaren 8 Bit benötigt.

V Overflow-Flag: Nach einer arithmetischen Operation mit Vorzeichen-behafteten 8-Bit-Werten verläßt das Ergebniss den Bereich -127 bis 127. Vorzeichen-behaftete 8-Bit-Werte sind Werte, bei denen B_7 das Vorzeichen angibt. (1 = negativ, 0 = positiv) Diese Werte werden in einem besonderen Format geschrieben.

Dieses Format soll später besprochen werden.

- H Half-Carry-Flag: Nach einer arithmetischen Operation liegt ein Übertrag vom unteren Halbbyte (B_0-B_3) auf das obere vor.
- I Interrupt-Flag: Mit diesem Flag kann man der CPU angeben, ob es auf Unterbrechungen (Interrupts) von Uhr oder anderem reagieren soll.

Wie löse ich nun Probleme ?

1. Problem: Ich will die Anzeige mit einem Strich füllen.

In BASIC: CLS:FOR I=1TO 156:GPRINT 8;:NEXT I

In Maschinensprache können wir das Problem ähnlich lösen:

Wir beginnen mit dem Löschen der Anzeige:

Für das Löschen der Anzeige muß man ein aufwendiges Programm schreiben. Im ROM des PC-1500 steht fest an einer Stelle schon solch ein Programm. Wenn wir dieses Programm benutzen wollen, so rufen wir es auf mit CALL .. (ähnlich GOSUB). Damit am Ende des Unterprogrammes im ROM ein Rücksprung in unser Programm erfolgt, hat es den Befehl RTN (ähnlich RETURN)

Wir können nun folgendes schreiben: (in Maschinensprache)

CALL &EE71. Dieser Befehl bewirkt ein Sprung zu dem Programm, welches ab &EE71 beginnt und wieder zurückkehrt mit RTN. Diesen Befehl kann man ändern:

Im ROM des PC-1500 im Bereich &FF00 bis &FFFF stehen 128 Adressen (je 2 Byte erst high-Byte, dann low-Byte) Diese Adressen zeigen auf häufig benötigte Programme im ROM, so wie z.B. die Anzeige zu löschen. In &FFF2 und &FFF3 steht diese Adresse. Wir können nun auch schreiben CALL (&F2). Der Microprocessor nimmt nun nicht wie früher die Adresse &EE71 direkt in das P-Register, sondern holt sich den Wert aus &FFF2 und &FFF3, welche er &FFF2, zuerst, in das P-Register übernimmt, wo die Adresse zum nächsten Befehl steht.

Als nächste Einheit haben wir eine Schleife. Solch eine Schleife kann man aufbauen, indem man den Zahlenwert 156 in ein Register nimmt, und solange 1 subtrahiert, bis 0 im Register steht. Also:

LD UL,156 ;Lade direkt die Zahl 156 in das Register UL

(Anweisung)

DEC UL ;subtrahieren 1 von UL

JR NZ,-x ;springe wenn das Ergebniss aus 'DEC UL' nicht null ist (Z = 0)
x Schritte zurück (Vorzeichen) (es wird auf die (Anweisung) gespr.)

Das Programm nimmt nun die Zahl 156 in ein Register. Danach beginnt die Schleife, in der die Anweisung durchgeführt wird und anschließend das Register um 1 vermindert wird. Nach der letzten Operation, einer arithmetischen Operation, wird auch das Z(ero)-Flag gesetzt. Solange in unserem Register UL aber noch ein Wert steht wird Z auf 0 gesetzt, denn das Ergebniss ist nicht null. In der letzten Anweisung nun wird ein bedingter Sprung ausgeführt, d.h. wenn die Bedingung erfüllt ist (NZ wahr, Z nicht gesetzt) wird ein Sprung um x Schritte (x wird später

ausgerechnet) zurück ausgeführt. Dieser Sprung würde auf die Anweisung führen, damit die Schleife geschlossen ist. Wenn Bedingung nicht erfüllt ist, d.h. in unserem Fall ist im UL-Register aus 1 null geworden, so ist das Z-Flag gesetzt und der Sprung wird nicht ausgeführt. Das ist das Ende der Schleife. Diese Schleife läßt sich etwas kürzer lösen, indem man einen anderen Befehl verwendet für DEC UL und JR NZ,-x, nämlich DJC ,-x. Dieser Befehl bewirkt, daß im UL-Reg. 1 subtrahiert wird und anschließend überprüft wird, ob im UL-Reg. jetzt 255 (&FF) steht, d.h. ob von 0 auf 255 gewechselt wurde. Wurde null noch nicht unterschritten, so wird der Sprung um x Schritte nach vorn ausgeführt. Also:

```
LD UL,155 ( 155, da nicht mehr von 156 bis 1, sondern von 155 bis 0 gezählt wird)
(Anweisung)
```

```
DJC ,-x
```

Unsere Anweisung besteht aus dem Befehl 'GPRINT 8;'. Für diesen Befehl gibt es wieder ein Unterprogramm, denn zur Ansteuerung der Anzeige gehört viel Arbeit. Dieses Unterprogramm verlangt aber von uns bestimmte Angaben. Nämlich:

Wo soll ich welches Zeichen zeichnen. Das Wo teilen wir dem Prgm. mir, indem wir im X-Reg. die 'Matrixspaltenadresse' haben. Diese Adresse ist, wie alles bei der Arbeit mit der Anzeige, ein kompliziertes Gebilde. Es genügt wenn wir wissen, daß die Adresse für die erste Spalte auf der Anzeige &7400 ist. Die weiteren Adressen brauchen wir in diesem Prgm. nicht berechnen, da das von uns angesprungene Unterprogramm diese selber ausrechnet und uns im X-reg. wieder zur Verfügung stellt. Wir müssen also vor Beginn der Schleife die erste Adresse aufbauen, denn die weiteren werden für uns errechnet. Wir sagen:

```
LD XH,&74 ; Lade &74 in das XH-Register, der oberen Hälfte des X-Reg.
LD XL,&00 ;
```

Damit haben wir die erste Adresse im X-Reg. In der Schleife ist die Wo-Frage gelöst, bleibt nur noch die welches-Frage. Das UP (Unterprogramm) verlangt von uns das anzuzeigende Zeichen (GPRINT) im Akkumulator, dem Hauptrechenregister.

Wir sagen daher:

```
LD A,8 ; Lade 8 in A
CALL &EDF6 oder CALL (188)
```

Nach diesem Unterprogramm ist der Akku und das UH-Reg. zerstört. Da wir diese Reg. nicht weiter brauchen können wir diese Tatsache vernachlässigen.

Wir haben jetzt alle Befehle unseres Prgm. es erklärt und hoffentlich verstanden. Wir können jetzt das Programm schreiben:

LD XH,&74	übersetzt in CPU-Code:	&48 &74
LD XL,&00		&4A &00
CALL (&F2)		&F2
LD UL,155		&6A 155
LD A,8		&B5 8
CALL (&88)		&CD &88
DJC ,-6		&88 6
RET		&9A

Jetzt können wir erst unser x bei DJC berechnen. Wir sehen, daß die CPU 6 Schritte zurückgehen muß, um von RET (&9A) auf LD A,8 (&B5) zu kommen, denn unser P(rogrammcounter) steht nach Lesen des Befehles DJC und der Zahl 6 auf RET. Nun muß er 6 Schritte zurückspringen (Wenn Bedingung erfüllt ist).

Wir können unser 14 Byte langes Programm nun in einen Speicherplatz POKE'n und mit CALL aufrufen:

Wir wählen den Speicherraum &7650 bis &765C. In diesem Speicherraum liegt die Variable ES. Wie sagen also: POKE&7650,&48,&74,&4A,0,&F2,&6A,155,&B5,8,&CD,&88,&88,6,&9A.

Jetzt können wir unser Programm mit 'CALL &7650' starten und wir werden das Ergebniss sehen.

```
10:CALL &7650
20:GOTO 20
```

PC-1500
.INFO

Programmieren in Maschinensprache

Teil 2

Ähnlich unserem letzten Problem wollen wir jetzt wieder eine Schleife benutzen.

2. Problem: Laufschrift

Wir wollen versuchen eine Laufschrift auf dem PC-1500 zu erzeugen mit Maschinensprache.

```
In BASIC sähe es folgend aus :  1Ø WAIT Ø
                                2Ø FOR I=155TO ØSTEP -1
                                3Ø CLS :GCURSOR I:PRINT (TEXT):NEXT I
```

Wieder können wir unser Programm genauso strukturieren:

Fangen wir mit der Schleife an:

```
LD UL,155
```

Anweisung

```
DJC ,-x (Sprung zur Anweisung)
```

Die 'Anweisung' besteht aus der Ausgabe des Textes auf der Anzeige an der Stelle, die durch dem 8-Bit-Register UL angegeben wird.

Zu Beginn der Anweisung brauchen wir ein 'CLS', welches wir in Maschinensprache durch 'CALL (&F2)' (indirekt) erwirken. Diese Routine zerstört aber das U-Register (16-Bit), darum sichern wir dieses Register erst auf den Stapel mit den Befehlen:

```
PUSH U ( das ganze 16-Bit-Register )
```

```
CALL (&F2) ( Anzeige löschen )
```

```
POP U
```

Mit dem Befehl 'POP' laden wir das Register wieder vom Stapel. Das ist sehr wichtig, da der Stapel von vielen Routinen benutzt wird, und so immer wiederhergestellt werden muß. So ist unser Programm auch nur eine Routine, die den Stapel so zurücklassen muß, wie er am Anfang war !!!!!

Jetzt haben wir die Anzeige gelöscht und müssen nur (!) noch den Text schreiben.

Den Befehl GCURSOR ersetzen wir durch die Anweisung, welche den Anfang des zu schreibenden Textes in eine Speicherstelle setzt, die Matrixspaltenpointer genannt wird und die Nummer (0-155) der Spalte enthalten muß.

```
LD A,UL ( Lade den Inhalt vom UL-Register in den Akku )
```

```
LD (&7875),A ( Lade den Inhalt des Akkus in die Speicherstelle &7875 )
```

Den Befehl 'PRINT' ersetzen wir ebenfalls durch ein Unterprogramm, welches aber die Startadresse des Textes und dessen Länge braucht, d.h. die Adresse, ab der der Text gespeichert ist. Diese Adresse werden wir schon beim Aufruf des Prgm.es im X-Reg (16-Bit) und die Länge in A vorfinden. Wir werden das X-Reg in das Y-Reg übertragen, damit uns der

Wert nicht verloren geht, denn das Y-Reg ist noch nicht gebraucht. Ebenso werden wir uns den Wert aus A sichern, nämlich in das UH-Register. Also müssen wir vor der Schleife folgendes einsetzen:

```
LD Y,X ( Übertrage den Wert aus X in Y )
LD UH,A ( " " " " A in UH )
```

Wir haben, also unsere Werte, die wir brauchen, in den Registern Y und UH. Das Unterprogramm, welches wir aufrufen braucht die Startadresse des Textes im U-Reg und die Länge in A. Wir müssen also erst das U-Reg sichern, um die Werte nicht zu zerstören. Damit sichern wir gleichzeitig die Schleifenvariable und die Länge des Textes. Anschließend übertragen wir das UH-Reg mit der Länge des Textes in den Akku (A). Jetzt können wir den Inhalt aus Y nach U speichern, was jedoch nicht einfach geht, wir müssen einem Umweg über das X-Reg nehmen:

```
PUSH U ( sichere U auf den Stapel )
LD A,UH ( Lade die Text-Länge in den Akku )
LD X,Y ( Lade die Startadresse in das X-Reg )
LD U,X ( Lade die Startadresse in das U-Reg )
```

Wir haben jetzt alle Werte aufgebaut und können die Routine aufrufen:

```
CALL (&92)
```

Jetzt müssen wir unbedingt den 16-Bit-Wert wieder vom Stapel holen, den wir dort abgelegt haben:

```
POP U ( Hole einen 16-Bit-Wert vom Stapel und bringe ihn in U )
```

Wir haben jetzt das Ende der Anweisung erreicht und können das 'NEXT' programmieren mit dem Befehl DJC (siehe Anfang).

Wie erhalte ich die Startadresse und die Länge des Textes bei Aufruf des Programmes ? Bei unserem ersten Problem haben wir das Programm mit CALL aufgerufen. Jetzt rufen wir das Programm mit CALL (Adresse),(Variabl auf, wobei die Variable eine String-Variable ist.

Am Ende unseres Prgm.es wollen wir noch eine kleine Zeitverzögerung einbauen, damit man den Text lesen kann. Wir haben im UL-Reg zuletzt 255 stehen, da wir von 0 subtrahiert haben. Also laden wir in das UH-Reg 0, damit der Wert 255 im U-Reg (16-Bit steht).Anschließend rufen wir eine Routine auf mit CALL (&AC) auf, die eine Verzögerung von $U \times 1/64$ Sekunden erzeugt. Am Ende des Prgm.es setzen wir das Carry-Flag zurück, und setzen den Befehl RET. Das Carry-Flag setzen wir zurück, damit der Interpreter (BASIC) nicht den Text ab der Startadresse in X mit der Länge in A nach der String-Variablen aus dem Aufruf überträgt, da diese Werte jetzt nicht stimmen.

Fertig sieht unser Programm so aus:

LD Y,X	übersetzt in Maschinensprache	&FD &5A
LD UH,A		&2B
LD UL,155		&6A 155
PUSH U ←		&FD &AB
CALL (&F2)		&F2
POP U		&FD &2A
LD A,UL		&24
LD (&7875),A		&AE &78 &75
PUSH U		&FD &AB
LD A,UH		&A4
LD X,Y		&FD &18
LD U,X		&FD &6A
CALL (&92)		&CD &92
POP U		&FD &2A
DJC -22		&B8 22
LD UH,0		&68 0
CALL (&AC)		&CD &AC
RCF		&E9
RET		&9A

Dieses Programm kann man in den Speicher POKE'n (ab &7750).

Danach setzt man einen Text in eine String-Variable, z.B.

AS="PC-1500 SHARP" (13 Zeichen)

und ruft das Programm auf mit CALL &7750,AS.

x) Versuchen Sie doch nach jeder Anzeige des Textes eine kleine Zeitschleife einzubauen, damit man den Text besser lesen kann. Beachten Sie dabei den DJC-Wert!

Aufruf:

Wir, das ist der SFBAG (Science-Fiction-Brief- und Aktionsgemeinschaft) suchen Science-Fiction-interessierte Computer-Besitzer, die Spaß haben an der Entwicklung eines Spiel-Zykluses, der auf möglichst vielen Computern gleichzeitig laufen soll. Darum suchen wir nicht nur PC-1500-Besitzer. Bitte meldet euch bei:

Bernd Rüter, Rahdenerstr.65, 4955 Hille

Die Beantwortung der Anschriften kann etwas dauern, da wir noch grundsätzliche Angelegenheiten regeln müssen.

Ich suche PC-1500-Besitzer für einen Programm-Tausch. Umfangreiche Liste mit qualitativen Programmen vorhanden.

Bernd Rüter, Rahdenerstr. 65, 4955 Hille

Wer hat gute Unterlagen über die Schnittstelle CE-158 für den PC-1500. (gute Schaltpläne, lesbar etc.)

Bernd Rüter, Rahdenerstr.65, 4955 Hille

Ich entwickle zur Zeit ein Konverter-Programm, welches Programme aus dem PC-1211/1212/1210 in den PC-1500 überträgt, die dort weiter benutzt werden können. So wird die umständliche Arbeit des Umtippens vom Computer übernommen. Nur Maschinensprache.

PC-1211 muß noch vorhanden sein !!

Bernd Rüter, Rahdenerstr. 65, 4955 Hille

Ich suche PC-1500-Besitzer, die nicht wissen, was Sie mit Ihrem Computer noch alles programmieren sollen. Ich habe aus Büchern BASIC-Programme, die Sie in den PC-1500 eingeben können. Bei Interesse wenden Sie sich an:

Bernd Rüter, Rahdenerstr.65, 4955 Hille

Funktionsanalyse für PC-1500

Seite
140 CHIP

Nr. 6 — Juni 1983

Eine serielle Schnittstelle

Was ist eine RS 232 C?

Die Verbindung zwischen zwei Computern oder Computer/Peripherie ist ein Kabel mit einer Steckerbelegung nach RS232C.

Seite
234 CHIP

Nr. 6 — Juni 1983

PC-1500 CLUB DEUTSCHLAND

CLUBBEITRAG 0,- DM

PC-1500 SCHULUNG =

NAME : _____

ADRESSE : _____

AN WELCHEN TAGEN KOENNEN SIE AN EINER SCHULUNG TEILNEHMEN :

MONTAG DIENSTAG MITTWOCH DONNERSTAG FREITAG

WELCHE BASIC BEFEHLE KOENNEN SIE NOCH NICHT RICHTIG ANWENDEN :

WELCHE THEMEN MOECHTEN SIE BEHANDELT WISSEN :

Die DATA-Anweisung ermöglicht das Abspeichern von Numerischen- und von Textausdrücken innerhalb einer Programmzeile. Am Anfang der Zeile steht der DATA-Befehl; es folgen die Ausdrücke, untereinander durch Kommata getrennt. Programmzeilen, die als ersten Befehl DATA führen, werden im normalen Programmablauf übersprungen. Der Aufruf der in DATA-Zeilen gespeicherten Ausdrücke erfolgt durch die READ-Anweisung, z.B. READ A\$ für Text o. READ A für Variablen, und zwar in der Reihenfolge in der die Ausdrücke im Programm stehen.

BESCHREIBUNG DES CE-158

DAS CE-158 INTERFACE IST EINE INTELLIGENTE SCHNITTSTELLE MIT EINER PARALLEL- UND EINER SERIELLEN SCHNITTSTELLE.

A. PARALLELSCHNITTSTELLE

DIE PARALLELSCHNITTSTELLE SENDET DATEN UEBER 8 DATENLEITUNGEN UND 3 STEUERLEITUNGEN ZU ANDEREN DATENVERARBEITENDEN GERAETEN WIE ZUM BEISPIEL EINEM DRUCKER. SIE ARBEITET NUR IN EINER RICHTUNG, D. H., SIE KANN NUR DATEN SENDEN, ABER NICHT EMPFANGEN. DIE DATEN- UND STEUERLEITUNGEN ENTSPRECHEN DER CENTRONICSSCHNITTSTELLE. DIE MEISTEN DRUCKER HABEN EINE CENTRONICSSCHNITTSTELLE.

VERBINDUNGSART VON CE-158 PARALLELSCHNITTSTELLE UND CENTRONICSSCHN. :

CE-158 PARALLEL	CENTRONICS
1 STROBE	1 1 STROBE
2 DATA 1	1 2 DATA 1
3 DATA 2	1 3 DATA 2
4 DATA 3	1 4 DATA 3
5 DATA 4	1 5 DATA 4
6 DATA 5	1 6 DATA 5
7 DATA 6	1 7 DATA 6
8 DATA 7	1 8 DATA 7
9 DATA 8	1 9 DATA 8
10 BUSY	111 BUSY
11 INIT	131 INPUT PRIME NICHT ERFORDERLICH
12 , 13 KEIN ANSCHLUSS	133 , 36 KEIN ANSCHLUSS
14 - 25 MASSE (GROUND)	116 , 17 , 19 - 30 , 35 MASSE (GROUND)
	112 PE NICHT ERFORDERLICH
	113 SLCT NICHT ERFORDERLICH
	114 MASSE SS NICHT ERFORDERLICH
	115 100 KHZ NICHT ERFORDERLICH
	118 + 5 VOLT NICHT ERFORDERLICH
	132 FAULT NICHT ERFORDERLICH
	134 LINE C. PULS NICHT ERFORDERLICH

INGESCHALTET WIRD DIE PARALLELSCHNITTSTELLE MIT DEM BEFEHL
 OPN"LPRT"
 AUSGESCHALTET WIRD MIT OPN
 IM INGESCHALTETEN ZUSTAND WERDEN ALLE LPRINT , LLIST , FEED , TAB ,
 CONSOLE BEFEHLE UEBER DIE PARALLELSCHNITTSTELLE AUSGEFUEHRT.
 MIT DEM CONSOLEBEFEHL WIRD DAS DATENUEBERTRAGUNGSFORMAT FESTGELEGT.

CONSOLE n , m
n = ZEICHEN / ZEILE
m = 1 ZEILENVORSCHUB
m = 0 KEIN ZEILENVORSCHUB

DIE PROGRAMMIERUNG VON n, m IST DRUCKERABHAENGIG.

B. SERIELLE SCHNITTSTELLE

DIE SERIELLE SCHNITTSTELLE SENDET DATEN UEBER 2 DATENLEITUNGEN UND 6 STEUERLEITUNGEN ZU ANDEREN DATENVERARBEITENDEN GERAETEN WIE ZUN BEISPIEL EINEM ANDEREN PC-1500 MIT CE-158.

DIE SERIELLE SCHNITTSTELLE ARBEITET IN ZWEI RICHTUNGEN, D. H., SIE KANN DATEN SENDEN UND EMPFANGEN.

DIE DATEN- UND STEUERLEITUNGEN ENTSPRECHEN DER NORM RS-232 C. DURCH DIESE NORMUNG KOENNEN ALLE ANDEREN GERAETE DIE EINE SERIELLE SCHNITTSTELLE MIT DIESER NORMUNG HABEN ANGESCHLOSSEN WERDEN.

VERBINDUNG VON ZWEI PC-1500 MIT CE-158 :

1. PC-1500 MIT CE-158 | 2. PC-1500 MIT CE-158

2 TD		3 RD
3 RD		2 TD
4 RTS		5 CTS
5 CTS		4 RTS
6 DSR		20 DTR
20 DTR		6 DSR
7 MASSE (GROUND)		7 MASSE (GROUND)
8 BLEIBT FREI		8 BLEIBT FREI

FUER DEN BETRIEB DER RS-232 C SCHNITTSTELLE SOLLTE MAN SICH MEHRMALS DIE BEDIENUNGSANLEITUNG DURCHLESEN.

Ta Co So Tokyo / Berlin

Freigabe

C-S 2 (2. Zeichensatz)

DISTRIBUTION

Ta Co So

Um dem PC1500 weitere Moeglichkeiten zu eroeffnen, bietet SHARP die Verwendung eines zweiten Zeichensatzes an, den der Benutzer selbst definieren kann. Die Programmierung dieser Zeichentabellen ist aeusserst zeitaufwendig und nervtoetend. Wir haben daher einen Zeichensatz entworfen, der fuer die meisten Zwecke ausreichend sein sollte.

Dieser Zeichensatz besteht aus insgesamt 4 einzelnen Tabellen, die in einem Modul-Block zusammengefasst wurden.

Fuer den Benutzer von unmittelbarem Interesse sind allerdings nur 2 Tabellen, so dass hier nur diese beschrieben werden.

Die Zeichentabellen sind als Maschinenprogramm auf Cassette gespeichert und werden mit dem BASIC-Kommando CLOAD M mit dem Namen "C-S 2" geladen. Dabei ist darauf zu achten, dass vor dem Ladevorgang das Kommando NEW mit der Adresse &4140 gegeben wird. Dadurch wird der Platz fuer die Zeichentabellen reserviert. Die Tabellen werden ab der Adresse &38C6 geladen und sind nur auf dieser Anfangsadresse verwendbar.

Die vollstaendige Kommando-Reihenfolge fuer das Laden heisst also:

NEW &4140

CLOAD M"C-S 2";&38C6

Zur Ausfuehrung des CLOAD-Kommandos muss natuerlich der Cassetten-Recorder arbeitsbereit sein.

Wenn der Ladevorgang beendet ist, muessen die Tabellen aktiviert werden. Dies geschieht, indem Sie dem Betriebssystem des SHARP PC1500 mitteilen, dass Sie mit einem zweiten Zeichensatz arbeiten wollen. Das erreichen Sie einem POKE-Kommando in der Form POKE &7850,0,&3A00.

Jetzt koennen Sie gemuess beigefuegter Tabelle die einzelnen Zeichen verwenden, d.h. sie in die Anzeige bringen oder sich ausdrucken lassen.

Die Zeichen werden dabei mit ihrer Verschluesselung angesprochen, die Sie entweder in der Hexadezimal-Form oder in der Dezimal-Form angeben koennen. Diese Entscheidung liegt bei Ihnen.

PREPARED BY : von Sprockhoff

Dokumentation

Taschen-Computer-Software

DATE 10. 6.1983

Tokyo / Berlin

PAGE 1

Ta Co So Tokyo / Berlin

Freigabe

C-S 2 (2. Zeichensatz)

DISTRIBUTION

Ta Co So

Zeichen, die Sie vermutlich haeufiger verwenden, koennen Sie auch ueber die Tastatur ansprechen, indem Sie eine zweite Tastatur-Belegung benutzen. Dies erreichen Sie ebenfalls mit einem POKE-Kommando, und zwar in der Form POKE &764E, PEEK &764E OR 4. Als Kontrolle fuer die 2. Tastaturbelegung erscheint dann in der Status-Zeile am oberen Rand der Anzeige ein Symbol aus 2 japanischen Schriftzeichen, die in der Uebersetzung "Ka Na" bedeuten. Dies ist eine Kurzbezeichnung fuer eines der japanischen Silbenalphabete, die in Japan ueber die 2. Tastaturbelegung gesteuert werden.

Bei den Tasten, die dann eine neue Belegung gewaess der Tastatur-Tabelle in der Anlage tragen, werden die urspruenglichen Zeichen nicht mehr benutzt, sondern nur noch die Neubelegung.

Der SHARP PC1500 ist nicht in der Lage, saemtliche neudefinierten Zeichen in eine neue Tastaturbelegung umzusetzen, so dass eine Auswahl getroffen werden musste. Selbstverstaerdlich koennen Sie saemtliche Zeichen weiterhin ueber die CHR\$-Funktion verwenden, doch nur 64 Zeichen davon (und 3 Zeichen aus der "normalen" Belegung) koennen Sie dann direkt ueber die Tastatur abrufen.

Wenn Sie die Taste SML (SPALL) betaetigen, erlischt die KaNa-Anzeige und die Tastatur-Zweit-Belegung wird ausgeschaltet. Dann gelten wieder die gewoehnnten Zeichen, die auf den jeweiligen Tasten auch eingetragen sind.

TaCoSo Tokyo / Berlin hat darauf geachtet, dass die 6 RESERVE-Tasten in ihrer Grundfunktion nicht veraendert wurden. Lediglich die SHIFT-Funktionen wurden mit neuen Zeichen belegt.

Soweit es moeglich war, wurden die neuen Zeichen so verteilt, dass ein Merken der Zweit-Belegung einfach faellt. So wurde z.B. das Zeichen fuer Promille auf die Taste fuer Prozent, das Zeichen fuer den Gegenschraegstrich auf die Taste mit dem Schraegstrich und das Zeichen fuer die Summe (der griechische Buchstabe Sigma) auf die Taste mit dem S gelegt.

Doch diese anemotechnische Belegung war nicht bei allen neuen Zeichen moeglich. Daher wurden bestimmte Zeichengruppen, insbesondere grafische Zeichen, in Tasten-Gruppen zusammengefasst.

Ebenfalls aufgenommen wurden einige Zeichen aus dem

PREPARED BY : von Sprockhoff

Dokumentation

Taschen-Computer-Software

DATE 10. 6.1983

Tokyo / Berlin

PAGE 2

Ta Co So Tokyo / Berlin

| Freigabe

C-S 2 (2. Zeichensatz)

| DISTRIBUTION

| Ta Co So

Grund-Zeichen-Vorrat, die zwar in die Anzeige gebracht werden koennen, vom Plotter aber nicht ausgedruckt werden. Diese Zeichen sind ueber den 2. Zeichensatz ebenfalls anzeig- und ausdruckbar.

Jedesmal, wenn Sie Ihren SHARP PC1500 einschalten und mit dem zweiten Zeichensatz arbeiten wollen, muessen Sie diese Absicht dem Computer mitteilen. Dazu genuegt (solange die Zeichentabellen geladen sind) der Befehl POKE 87850,0. Dieses Byte wird beim Einschalten durch die Initialisierungsroutine (die unter anderem auch den Plotter einschaltet) mit dem Wert &FF (dezimal 255) ueberschrieben. Damit wird nur der Standard-Zeichensatz verwendet. Durch das Schreiben des Wertes 0 in diese Stelle wird der neue Zeichensatz zusaetzlich benutzt. Ebenfalls von Neuem muessen sie den oben beschriebenen POKE-Befehl zur Aktivierung der Tastatur-Zweit-Belegung geben, wenn Sie diese Belegung benutzen wollen und das KaNa-Zeichen nicht eingeblendet ist.



PREPARED BY : von Sprockhoff

| Dokumentation

| Taschen-Computer-Software

| DATE 10. 6.1983

| Tokyo / Berlin

| PAGE 3

Ta Co So Tokyo / Berlin

Freigabe

C-S 2 (2. Zeichensatz)

DISTRIBUTION

Ta Co So

Bitte beachten Sie:

Das Zeichen mit der Verschlüsselung &E6 ist nicht in der Anzeige darstellbar, wohl aber ausdrückbar. Wenn Sie versuchen dieses Zeichen auf der Anzeige darzustellen, wird ein nicht definiertes Zeichen angezeigt werden.

Diese Beschränkung ist durch die Ansteuerung der Tabellen fuer Ihren neuen Zeichensatz notwendig geworden. Fuer den Plotter stehen Ihnen also 128 neue Zeichen zur Verfügung, fuer die Anzeige "nur" 127.

A C H T U N G !

Einige Zeichen haben auf dem Plotter Uebersbreite und sollten deshalb nicht am Papierrand verwendet werden. Anderenfalls koennte es moeglich sein, dass weitere Zeichen nicht mehr korrekt gedruckt werden koennen.

Diese Zeichen tragen folgende Verschlüsselung:

&D0, &D1

&E6

&F8, &FA, &FB, &FE, &FF

Und nun wuenschen wir Ihnen mit den neuen Zeichen viel Spass!

PREPARED BY : von Sprockhoff

Dokumentation

Taschen-Computer-Software

DATE 10. 6. 1983

Tokyo / Berlin

PAGE 4

Ta Co So Tokyo / Berlin	Freigabe
C-S 2 (2. Zeichensatz)	DISTRIBUTION
	Ta Co So

Tabelle 1: Tastatur-Belegung

(F1)	spanisches Ausrufezeichen
(F2)	Anfuhrungszeichen unten
(F3)	Cent (amerikanische Waehrung)
(F4)	Pfund (englische Waehrung)
(F5)	Promille
(F6)	Paragraph
(OFF)	Cursorzeichen (nicht abdruckbar!)
A	Ae (Umlaut)
€	ae (Umlaut)
Q	Ce (Umlaut)
o	oe (Umlaut)
U	Ue (Umlaut)
u	ue (Umlaut)
s	ss (scharfes s; sz)
S	Summe (griechisches Sigma)
I	Integral
(x)	x quer (mathematisches Zeichen)
(y)	y quer (mathematisches Zeichen)
K	kleiner gleich (mathematisches Zeichen)
(k)	sehr klein gegen (mathematisches Zeichen)
L	groesser gleich (mathematisches Zeichen)
(L)	sehr gross gegen (mathematisches Zeichen)
B	ungefaehr (mathematisches Zeichen)
=	ungleich (mathematisches Zeichen)
(=)	entspricht (mathematisches Zeichen)
+	plusminus (mathematisches Zeichen)
(ENTER)	unendlich (mathematisches Zeichen)

Saemtliche Zahlen erzeugen in der Grundstellung ihren jeweiligen Exponenten und in der SHIFT-Stellung ihren jeweiligen Index.

E	Pfeil nach links oben
R	Pfeil nach rechts oben
D	Pfeil nach links unten
F	Pfeil nach rechts unten
(Pfeil nach unten)	Pfeil nach unten
(Pfeil nach oben)	Pfeil nach oben

PREPARED BY : von Sprockhoff	Dokumentation
Taschen-Computer-Software	DATE 10. 6.1983
Tokyo / Berlin	PAGE 5

Ta Co So Tokyo / Berlin

Freigabe

C-S 2 (2. Zeichensatz)

DISTRIBUTION

Ta Co So

(Pfeil nach links) Pfeil nach links

(Pfeil nach rechts) Pfeil nach rechts

N N Tilde (spanisches Zeichen)

(n) n Tilde (spanisches Zeichen)

(/) spanisches Fragezeichen

(c) Cent (amerikanische Waehrung)

([eckige Klammer auf

(() geschweifte Klammer auf

) eckige Klammer zu

() geschweifte Klammer zu

/ Gegenschraegstrich

(-) Hochkomma

[] senkrechte Klammer (Insert-Zeichen im PRO-Mode)

(c) cedit (franzoesisches Zeichen)

Die uebrigen Tasten tragen auch bei dieser Tastaturbelegung ihre Ihnen gewoehnten Zeichen.



PREPARED BY : von Sprockhoff

Dokumentation

Taschen-Computer-Software

DATE 10. 6. 1983

Tokyo / Berlin

PAGE 6

Tabelle 2:

Halb- byte			0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	1	2	8	9	A	B	C	D	E	F	8	9	A	B	C	D	E	F
	Á	À	Á	É	È	Ê	Ë	Ê	Ó	Ò	Ô	Å	Æ	Œ	Œ	∅	∂	∂
	á	à	á	é	è	ê	ë	ê	ó	ò	ô	å	æ	œ	œ	∅	∂	∂
	Ä	Ö	Ä	£	Ŧ	¡	»	»	»	»	≡	̄	⌈	⊖	⊖	2	3	4
	ä	ö	ä	£	Ŧ	¿	«	«	«	«	≠	̄	∖	∕	∕	7	8	9
	β	¶	β]]	↑	→	↖	↖	↖	↗	″	Σ	∫	∅	∅	2	3	4
	%	§	%	} }	←	↙	↘	↘	↘	↘	±	,	∞	5	6	7	8	9
	↔	↕	↔	↕	⇒	⇒	©	©	©	∣	≡	∕	á	à	â	æ	ä	ä
	≈	≈	≈	≈	⇐	⇐	∧	∧	∧	∣	⊕	—	•	α	α	⊠

À =>	128
Á =>	129
Â =>	130
Ã =>	131
Ä =>	132
Å =>	133
Ö =>	134
ó =>	135
Ø =>	136
À =>	137
Æ =>	138
Œ =>	139
Ū =>	140
ø =>	141
ĉ =>	142
ċ =>	143
đ =>	144
đ =>	145
ð =>	146
ē =>	147
e =>	148
ē =>	149
ó =>	150
ò =>	151
ô =>	152
ô =>	153
œ =>	154
œ =>	155
œ =>	156
ø =>	157
č =>	158
e =>	159
Ħ =>	160
Ů =>	161
Ů =>	162
ƒ =>	163
Ň =>	164
ı =>	165
› =>	166
› =>	167
š =>	168
ˆ =>	169
ċ =>	170
o =>	171

1 =>	172
2 =>	173
3 =>	174
4 =>	175
đ =>	176
đ =>	177
č =>	178
č =>	179
ñ =>	180
ı =>	181
ı =>	182
š =>	183
š =>	184
ř =>	185
ˆ =>	186
š =>	187
š =>	188
ř =>	189
š =>	190
š =>	191
β =>	192
č =>	193
[=>	194
] =>	195
† =>	196
→ =>	197
ˆ =>	198
ˆ =>	199
ˆ =>	200
Σ =>	201
ƒ =>	202
o =>	203
ı =>	204
2 =>	205
3 =>	206
4 =>	207
% =>	208
§ =>	209
< =>	210
> =>	211
↓ =>	212
← =>	213
↖ =>	214
↘ =>	215
± =>	216
' =>	217

∞ =>	218
5 =>	219
6 =>	220
7 =>	221
8 =>	222
9 =>	223
4 =>	224
† =>	225
⊕ =>	226
‡ =>	227
† =>	228
4 =>	229
⊙ =>	230
ı =>	231
≡ =>	232
' =>	233
á =>	234
â =>	235
ã =>	236
ä =>	237
å =>	238
ä =>	239
± =>	240
ˆ =>	241
ˆ =>	242
ˆ =>	243
↓ =>	244
ε =>	245
ñ =>	246
ı =>	247
⊗ =>	248
_ =>	249
... =>	250
... =>	251
ˆ =>	252
â =>	253
∏ =>	254
⊞ =>	255

Bücher

PC-1500 intern

Die Antwort auf viele Fragen von Lesern, die unsere Redaktion erhalten hat, wie beispielsweise »Ist es möglich, den PC-1500 in Maschinensprache zu programmieren?« oder »Was ist mit den Befehlen PEEK, POKE und STATUS los?« gibt das technische Handbuch zum Sharp PC-1500. Das Buch, in englischer Sprache, ist eine Erweiterung zum Benutzerhandbuch für den PC-1500.

Es geht intensiv auf den Mikroprozessor LH 5801 und den LH 5810/LH 5811 I/O Port Controller ein. Dabei erklärt es den Aufbau von Registern und Flipflops sowie die Belegung der CPU-Ports anschaulich an Schaltplänen und Tabellen. Außerdem stellt es den Ablauf logischer Instruktionen, von Addition und Subtraktion, in den einzelnen Registern verständlich dar. Assembler-Instruktionen, wie Vergleichs-, Such- oder Verzweigungsbefehle, werden beschrieben und an Beispielen verdeutlicht. Eine Tabelle faßt alle Assembler-Befehle übersichtlich zusammen. Damit ist es möglich, den PC-1500 nicht nur in Basic, sondern auch in Maschinensprache zu programmieren.

Ein Kapitel befaßt sich mit der internen Struktur und den Funktionen des Controllers. In einem weiteren Kapitel wird ausschließlich die Hardware anhand von Schaltplänen und Tabellen über die Speicherbelegung erläutert. Zusätzlich beschäftigt sich das Manual mit PC-1500-Software. Es erklärt die Basic-Befehle NEW, STATUS, PEEK, POKE, CALL, CSAVE M und CLOAD M sowie Systemunterprogramme zu arithmetischen und String-Operationen, zum Vergleichen und Suchen von Daten, für das Ansteuern des Display und den Anschluß eines Druckers oder Kassettensrecorders.

Am Ende des Buches findet der Leser Beispielprogramme zur Maschinensprache. Das Buch wird von Sharp selbst vertrieben und kostet zirka 50 Mark.

Info: Sharp Pocket Computer PC-1500, Technical Reference Manual, 1983, Sharp Corporation, Sharp Electronics (Europe), Sonninstr. 3, 2000 Hamburg 1, Tel. 040-237751

SHARP

**POCKET COMPUTER PC-1500
TECHNICAL REFERENCE MANUAL**

SHARP CORPORATION

D a s C E - 0 8 1 5

(Ein Produkt der Fa. TaCoSo)

Sie haben sicherlich schon bemerkt, daß die Papierrollen, die von SHARP für den CE-150 verkauft werden, recht teuer und dazu schnell verbraucht sind.

Dies ist ein ärgerlicher Umstand, dem wir jetzt abgeholfen haben:

Das CE-0815 besteht aus einem Kupferdraht vom 5 mm² Durchmesser und etwa 30 cm Länge.

Dieser Draht wird nun zu einem Ständer gebogen, auf dessen Achse man handelsübliche Papierrollen für Tischrechner stecken kann.

Der Deckel des Plotters im CE-150 kann nun so aufgelegt werden, daß Sie sowohl Papier von der großen Rolle einziehen als auch weiterhin die Abreißkante verwenden können.

Dadurch ist es möglich, auch ellenlange Listings im CSIZE 2-Format zu drucken, ohne daß mitten in der Arbeit wieder einmal das Papier zu Ende gegangen ist und der Druck eventuell sogar wiederholt werden muß.

Weitere Informationen über TacoSo bzw. Fischel GmbH.

Ta Co So Berlin
(Taschen-Computer-Software)

Tips und Tricks von TaCoSo

Für viele CE-150-Besitzer ist es sicherlich eine Quelle manchen Ärgers gewesen, wenn sich das Papier bei einem Ausdruck fein säuberlich um den Plotter ringelt und sich - manchmal nur mit Knicken - nur schwer wieder in geordnete Bahnen bringen läßt.

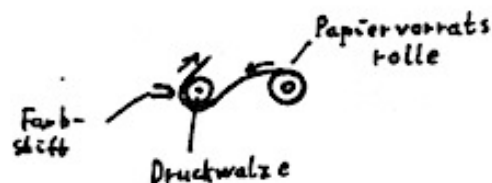
Hier gibt es nun eine relativ einfache Möglichkeit, wenigstens einen Teil dieser "Roll"-Neigung des Papiers aufzufangen:

Wenn Sie eine neue Rolle Papier einlegen, sei es im Papierfach des CE-150 oder auf dem CE-0815, legen Sie die Rolle so ein, daß Sie den Papieranfang statt von unten jetzt von oben von der Rolle ziehen.

Jetzt ist es zwar etwas schwieriger, entgegen der Aufwickelrichtung der Rolle das Papier in die Mechanik einzuführen, doch wird diese zusätzliche Mühe belohnt, wenn Sie bei längeren Ausdrucken eine spürbar geringere Neigung des Papier feststellen können, sich selbst aufzurollen, als vorher.

Die nachfolgende Skizze mag diesen Gedanken etwas verdeutlichen:

Plotter - Seitenansicht



Stand 10
ist zugesichert



1. Computerbörse Berlins

am Samstag, den 23. Juli 1983, von 10-18 Uhr
in 1000 Berlin 12, Hardenbergstr. 34
(Alte TU-Mensa)

Mietvertrag über Ausstellungsfläche

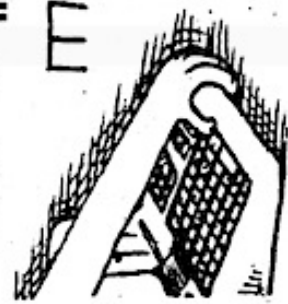
zwischen dem Veranstalter und

Fischel Betriebswirtschaftlicher
Beratungs- und Programmierdienst GmbH
Kaiser-Friedrich-Straße 54 a
1000 Berlin 12 - Tel. 323 60 29

ENVOUZZIHH!

Die 1. Computerbörse Berlins
in der alten TU-Mensa

MICROCOMPUTER
PERIPHERIE
PROGRAMME



Ob Sie nun tauschen, kaufen, verkaufen
oder einfach nur schauen, es wird
Ihr Gewinn.

Wann? 23. Juli

Hardenbergstr. 34
in der alten TU-Mensa
(Nähe Bhf. Zoo)

Wo?

Rufen Sie an!

weitere Infos
unter:
030/6926178

Berlin, den 8.6.83

B. Fischel
(Mieter)

Fischel Betriebswirtschaftlicher
Beratungs- und Programmierdienst GmbH
Kaiser-Friedrich-Straße 54 a
1000 Berlin 12 - Tel. 323 60 29