

PC-1500 ZEITUNG

Taschencomputer

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

ANWENDER

Ich kenne einen Anwender, der ein Lottorie-Programm geschrieben hat, bevor das Programm aufhörte zu rechnen, waren 13 (dreizehn) Tage vergangen!!!. Er kaufte dann einen MZ-80a bei mir, nachdem er 5 Richtige erzielt hatte.

Kennen Sie einen Anwender, der seinen PC-1500 erfolgreicher einsetzt?

Man siehe auch Seite 9a dieser Ausgabe.

PC-1500 TREFFEN: Merkgel "letzter Mittwoch im Monat" = 31.08.83

Clublokal; Auskunft Tel. 323 60 29 .



blau

grün

rot

Plot des Monats von
Thomas Stabingis, Berlin

INHALT:

=====

Seite:

1. Dr. Dorsch	1
2. Breit- und Fettschrift	5
3. Error Null	6
4. Wintersport	7
5. Utilities	8
6. Das Haus	9
7. Diverses	9a
8. Aussteller	10
9. Beitrag von M. Feussi	12
10. Hexmonitor	15
11. <u>T</u> ask- <u>V</u> erwaltungs- <u>S</u> ystem	20
12. AUSLAND	22
13. Entschuldigung	32

SHARP Büroelektronik

Erfolg mit System

PC-1500

SHARP

Durch Nachdenken vorn.

DR. ROGER DORSCH DIPLOM-PHYSIKER. KLAVIER- UND FLÜGELTECHNIK

STIMMUNG, REPARATUR UND NEUBESAITUNG, GUTACHTEN, ANKAUF, VERKAUF,
REGULIERUNG, INTONATION, RESTAURIERUNG VON POLITUR UND FURNIEREN.

Fischel Betriebswirtschaftlicher
Beratungs- und Programmierdienst GmbH
Kaiser-Friedrich-Str. 54a
1000 Berlin 12

KANTSTRASSE 38/II
1000 BERLIN 12
TELEFON 313 73 49
Postfach 6065

DATUM 14. 7. 1983

IHR AUFTRAG VOM

AUSGEFÜHRT AM

Sehr geehrter Herr Fischel!

Die deutsche Inhaltsangabe des französischen Programms PIANO -
PC-1500 sende ich Ihnen anbei wie versprochen rechtzeitig.
Falls Sie noch nähere Informationen über dieses Programm ver-
öffentlichen möchten, z. B. Flußdiagramm, Tabellen, Notendrucke,
so bitte ich um Mitteilung des Redaktionsschlusstermins.
Eine eigene Computergrafik füge ich noch hinzu. Wie diese Form
des Quintenzirkels zeigt, wäre auch schnell das Drucken chemi-
scher Strukturformeln möglich. Beispiele will ich in Kürze lie-
fern.

Mit freundlichem Gruß,

R. Dorsch



Dr. Roger Dorsch
Kantstraße 38 1000 Berlin 12
Telefon 313 73 49

Der PC-1500 als Musikinstrument, Komponierhilfe, Spieluhr
und Notendrucker

Sobald dieses aus Frankreich stammende Programm gestartet ist, fragt der PC-1500 nach dem Titel der Melodie, die Sie komponieren wollen. Dieser Titel wird am Anfang Ihres Notenblattes gedruckt. Dann wird die Zahl 1 angezeigt; es handelt sich um die Numerierung der ersten Note, die zu schreiben ist. Die zwei ersten Reihen der Tastatur dienen als "Klaviertasten". Eine Tabelle zeigt die Zuordnung der verwendbaren Töne zu den Tasten. Die in den Rechner eingegebene Note können Sie hören, sobald Sie auf ENTER drücken.

Das Programm bietet die Möglichkeit, den komponierten Part rückwärts zu lesen; dies bewirkt der nach links zeigende Pfeil. Um fortzuschreiten, braucht man nur die Taste mit dem nach rechts zeigenden Pfeil zu benutzen. Die Tasten bewirken Repetition; es genügt, sie festzuhalten, um die schon eingegebenen Noten vorbeiziehen zu lassen.

Das Programm verzeichnet neben der Numerierung jeder Note ihren Zeitwert (Viertel, Halbe, Punktierte), Erhöhungszeichen (Kreuz) und ihren Namen (do, re, mi, usw.).

Ein Drücken der Taste SPACE liefert dann das komponierte Notenblatt.

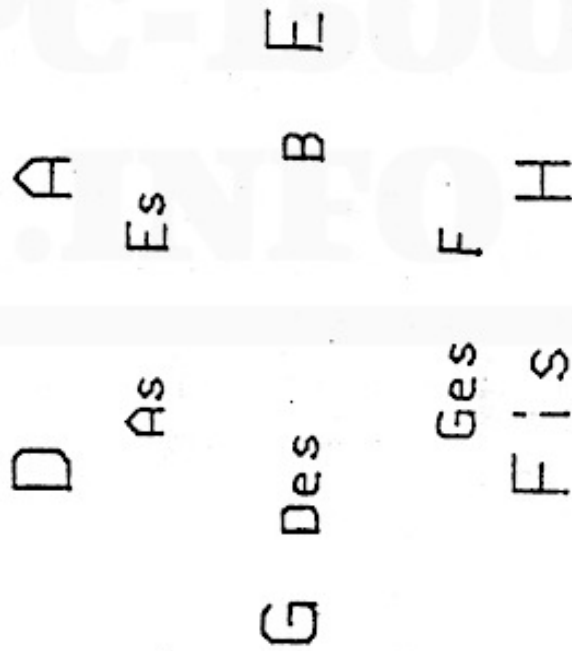
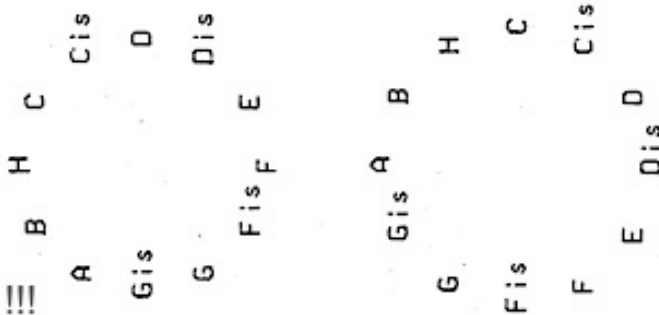
Das Programm macht nur von einem Teil der Töne Gebrauch, die der PC-1500 abgeben kann.

Dr. Roger Dorsch
Klavierstimmer

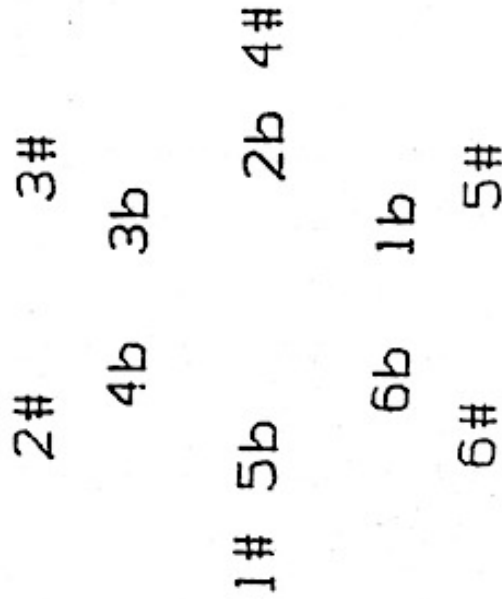
Postfach
6065

D-1000 BERLIN 12

Quintenzirkel



Erklaerung



Beispiel: Des-Dur
 hat als Vorzeichen
 5b

PC-1500 und FETTSCHRIFT

20,- DM

AUF CASSETTE

Aufgerufen wird dieses Unterprogramm durch GOSUB "FS". Dann wird die in Z\$ gespeicherte Zeichenfolge wahlweise in Fett- oder Breitschrift gedruckt. Die Wahl erfolgt durch Drücken der Tasten Ø (Fett) oder 1 (Breit). Der Zeilenwechsel erfolgt automatisch. Die vorgewählte Schriftgröße behält Gültigkeit. Nach Programmablauf befindet sich der PC-1500 im Textmode.

Soll nur eine Schriftart verwendet werden, so sind die Zeilen 10 und 15 zulöschen, die PRINT-Anweisung zu entfernen und in 5 F=1 (Fett) oder F=2 (Breit) zu setzen.

Sollen mehr als 16 Zeichen gedruckt werden, kann Z\$ durch die entsprechende Textfeldvariable ersetzt werden. Auf diese Art sind bis zu 80 Zeichen möglich.

Wenn das Programm als Hauptprogramm laufen soll, kann folgende Änderung vorgenommen werden:

```
5: "FS"AREAD Z$: C=PEEK .....
50: statt RETURN, END
```

(bei CSIZE 1 ist nur Breitschrift empfehlenswert)

Verwendete Variablen:

- Z\$, A, B, C, F, P
- Lauf-Variable
- I, J, L, M

AAAAA

AAA.

AAA

AAA

AAA.



Fischel Ber.- und Progr.-
Dienst Mo.-Fr.: 10-18.30 Uhr
Sa. 10-14; PC-1500 Club
Kaiser-Friedrich-Straße 54a
1000 Berlin 12 - Tel. 3 23 60 28

betrifft : Fehlermeldung ERROR 0

Eifrige Benutzer des PC-1500 haben wahrscheinlich schon hin und wieder Fehlermeldungen auf ihrem Display gelesen, die in der Bedienungsanleitung nicht oder nur sehr global beschrieben waren. So erhielt ich vor kurzem einen Anruf von Herrn Dr. Dorsch, der die Fehlermeldung ERROR 0 nach dem Starten seines Programmes erhielt. Auf diese Fehlermeldung hingewiesen, war ich sehr interessiert herauszufinden, wie man selbige Meldung reproduzieren könnte.

In der Bedienungsanleitung des PC-1500 auf Seite 152 wird die Fehlermeldung 0 nur sehr global erwähnt. Inzwischen habe ich herausgefunden, unter welchen Bedingungen sich das Betriebssystem mit ERROR 0 meldet.

Die fehlerhafte Applikation der Anweisungen "READ" und "DATA" verursachen den ERROR 0. Dabei ist hier besonders darauf hinzuweisen, daß bei der "DATA"-Anweisung zum einen die zwingend vorgeschriebenen Hochkommas bei Textkonstanten fehlen und zum anderen die Textkonstante eine Kombination aus Groß- und Kleinbuchstaben ist.

Folgendes Programm

```
10:RESTORE :READ
    AS
100:DATA Abc
```

erzeugt ERROR 0 IN 10.

Ich hoffe hiermit eine weitere Informationslücke geschlossen zu haben. Für weitere Informationen bzw. Ergänzungsvorschläge bin ich dankbar.

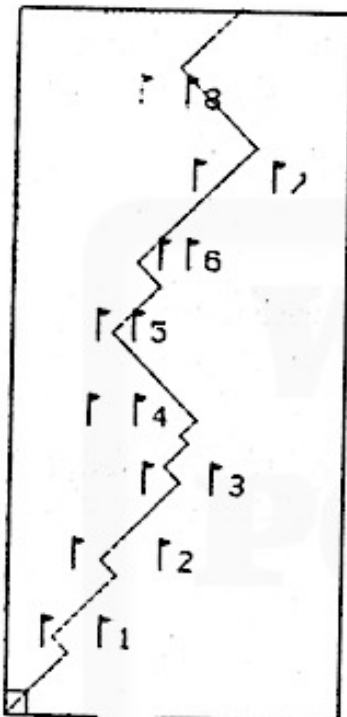
- 7 -

Wintersport :

Die Fähnchen werden zufällig generiert.
(Beibehaltung des letzten Kurses möglich)

SLALOM

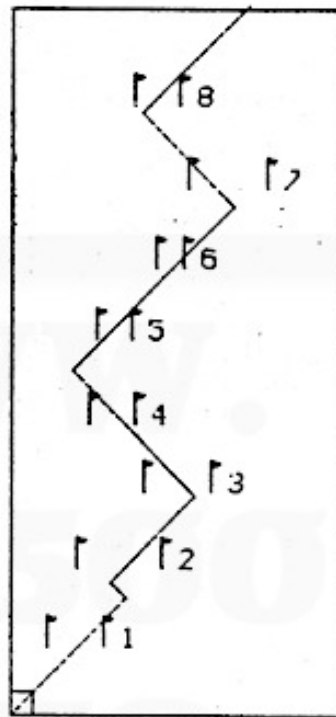
VÖLKER



*** BEWERTUNG ***
Tor-1 : 036 Pkt
Tor-2 : 008 Pkt
Tor-3 : 029 Pkt
Tor-4 : 0-u Pkt
Tor-5 : 0oW Pkt
Tor-6 : 0-u Pkt
Tor-7 : 0oW Pkt
Tor-8 : 0-u Pkt
TOTAL : 073 PKT

SLALOM

VÖLKER



*** BEWERTUNG ***
Tor-1 : 044 Pkt
Tor-2 : 008 Pkt
Tor-3 : 027 Pkt
Tor-4 : 053 Pkt
Tor-5 : 057 Pkt
Tor-6 : 068 Pkt
Tor-7 : 012 Pkt
Tor-8 : 051 Pkt
Zusatz : 060 Pkt
TOTAL : 380 PKT

Preis auf Anfrage (10,- DM Listing)

Tel. 030-323 6029

Reverse-Video für Sharp PC-1500

Mit dem folgenden kleinen Maschinensprache-Programm kann blitzartig die Anzeige des PC-1500 umgekehrt werden. Man kann dieses Programm beispielsweise für einen Crash in einem Spielprogramm verwenden. Das Programm wird folgendermaßen eingegeben (die Zahlen in Klammern beziehen sich auf den Rechner mit 8K-Modul):

1. NEW 0
2. Zeile 10:REM + 25 Doppelpunkte
3. Einpoken der Zahlen (POKE 16586 (bzw. 14538), 72, 118, 74...)

Vor dieser Zeile 10 darf keine andere Zeile stehen! Das Programm wird mit CALL 16586 (14538) aufgerufen. *Martin Erzberger*

```

EINPOKEN
DER ZAHLEN:
POKE 16586 (14538)
, 72, 118, 74, 0, 88, 11
9, 90, 0, 106, 77, 251,
181, 255, 1, 14, 251,
181, 255, 17, 30, 64
POKE 16607 (14559)
, 80, 136, 14, 154

```

```

10:REM HOJ XwZ jM
QP

```

```

15:REM BEISPIELPR
OGRAMM

```

```

20:WAIT 0:CLS :
GCURSOR 4:
PRINT "***** S
HARP PC-1500 *
*****"

```

```

30:CALL 16586
40:FOR I=1 TO 50:
NEXT I
50:GOTO 30

```

Spreizschrift-Programm

Das Programm ist als Unterprogramm ausgeführt. Das lässt sich jedoch ändern, wenn man das RETURN in Zeile 50 durch END ersetzt. Ansonsten wird das Unterprogramm folgendermassen aufgerufen:

```

.....
100 T$=-Sharp PC-1500:GOSUB D
.....
10 D:AREAD T$:WAIT 0:FOR A=1 TO LEN T$
20 B=64677+5*(ASC (MID$(T$,A,1))-33)
30 FOR C=0 TO 4:D=PEEK (B+C):GPRINT D;D:;NEXT C
40 GPRINT 0:;NEXT A
50 WAIT :GPRINT :RETURN

```

Hier erscheint der Inhalt von T\$ gespreizt in der Anzeige. Es können somit vierzehn Zeichen in der Anzeige dargestellt werden.

Bild 15: Programm für Spreizschrift


```
10: GRAPH
69: REM HAUS-RECH
    TECK
70: LINE (12, 15)-(
    160, 70), 0, 0, 8
71: LINE (11, 14)-(
    161, 71), 0, 0, 8
80: LINE (10, 10)-(
    10, 70), 9
90: SOGN
91: REM DACH
95: LINE (1, 1)-(75
    , 39)-(149, 1), 0
    , 3
96: LINE (-7, -2)-(
    157, -2)
100: LINE (-7, 0)-(7
    5, 40)-(157, 0),
    0, 3
101: REM SCHORNSTE
    IN
102: LINE (90, 21)-(
    90, 37)-(100, 37
    )-(100, 21)-(90
    , 21), 0
103: LINE (-6, 1)-(1
    56, 1), 0, 3
104: REM FIRST
105: LINE (150, 0)-(
    157, 0), 0, 1
110: LINE (157, 0)-(
    157, -5)-(-7, -5
    )-(-7, 0), 0
115: LINE (157, -4)-
    (-7, -4)
120: LINE (-7, 0)-(1
    50, 0), 0
129: REM TUER
130: LINE (75, -55)-
    (75, -22)-(95, -
    22)-(95, -55), 0
    , 2
131: LINE (74, -55)-
    (74, -21)-(96, -
    21)-(96, -55)
139: REM LINKES FE
    NSTER
140: LINE (15, -45)-
    (15, -20)-(65, -
    20)-(65, -45)-(
    15, -45), 0
142: LINE (17, -43)-
    (17, -22)-(63, -
    22)-(63, -43)-(
    17, -43)
143: LINE (40, -43)-
    (40, -22)-(42, -
    43)-(42, -22)
144: REM RECHTES F
    ENSTER
145: LINE (105, -45)
    -(105, -20)-(13
    5, -20)-(135, -4
    5)-(105, -45), 0
147: LINE (107, -43)
    -(107, -22)-(13
    3, -22)-(133, -4
    3)-(107, -43)
148: LINE (120, -43)
    -(120, -22)
150: LINE (77, -38)-
    (82, -38), 0, 2
159: REM TREPPE
160: LINE (70, -55)-
    (70, -59)-(100,
    -59)-(100, -55)
    , 0, 0
170: LINE (70, -59)-
    (65, -59)-(65, -
    63)-(105, -63)-
    (105, -59)-(100
    , -59)
180: LINE (65, -63)-
    (60, -63)-(60, -
    67)-(110, -67)-
    (110, -63)-(105
    , -63)
498: TEXT
499: LF 8
1000: LPRINT
    STATUS :
```



1072

Herr Trauer, Berlin

Sharp PC 1500 Systemhandbuch

(Technical Reference Manual)
In deutscher Sprache!

55,- DM

inkl. MwSt., Porto, Verpackung, Nachnahmegebühr
Ca. 160 Seiten starke Dokumentation in die deutsche
Sprache übersetzt

Inhalt:

- Wie kann man den PC 1500 in Maschinensprache programmieren
- Beschreibungen von Assembler-Instruktionen
- Tabelle mit allen Assembler-Befehlen
- Tabelle über Speicherbelegungen
- Schaltpläne
- Erklärung von Basic-Befehlen
- Erklärung von System-Programmen
- Aussteuerung des Display
- und vieles mehr!

Ein Muß für jeden interessierten PC 1500-Benutzer. In Buchform A4-Format gebunden. Lieferung solange Vorrat reicht (begrenzte Auflage).

Holtkötter

Das richtige Programm.

Albert-Schweitzer-Ring 9
2000 Hamburg 70, Tel. 040/66981-0

Programmieren von Taschenrechnern

Bd. 8: Lehr- und Übungsbuch für den TI-57 LCD
Bd. 9: Lehr- und Übungsbuch für den SHARP PC-1500

Ausführliche Information in unserem Sonderverzeichnis „Mikrocomputer/Taschenrechner/Programmiersprachen“.

Info-Coupon

Senden Sie mir/uns Ihr Verzeichnis „Mikrocomputer/Taschenrechner/Programmiersprachen“

Anschrift: _____

Verlag Vieweg · Postfach 5829 · 6200 Wiesbaden 1

! BASIC

for beginners - ein BASIC-Kurs
"von Anfang an"

Standard-BASIC und spezielle Anweisungen für

SHARP TANDY
&
PC-1500 PC-2

systematischer Aufbau des Kurses mit vielen
praktischen Übungen

Kursbeginn: voraussichtlich ab 9.9.83

1983 - ca. 21:00 Uhr

telefonisch bitte melden bei

Jürgen Rothe 363 65 76

Taschencomputer schafft Kontakte

Wer Interesse hat an christlichen, kirchlichen oder auch allgemein nicht-naturwissenschaftlichen Computeranwendungen, wer Lust und Ideen hat, an entsprechender Software herumzubasteln, der sei hiermit herzlich zur Kontaktaufnahme mit mir eingeladen (Werner Küstenmacher, Evangelischer Presseverband, Birkerstr. 22, 8000 München 19, Tel. (089) 184031).

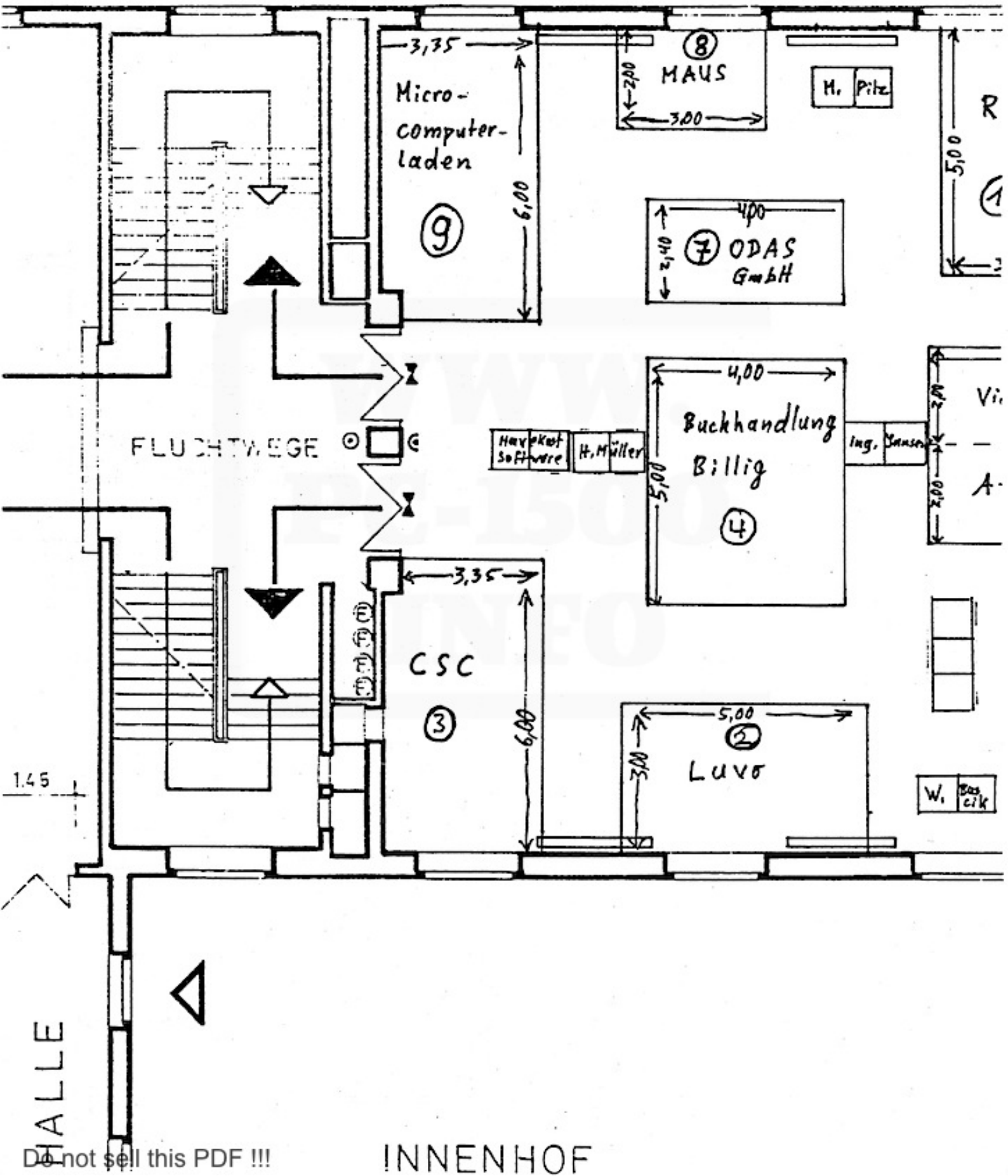
Ausstellung der 1. Computerbörse BERLIN
- ein "Bombenerfolg" -

1:100

14. J

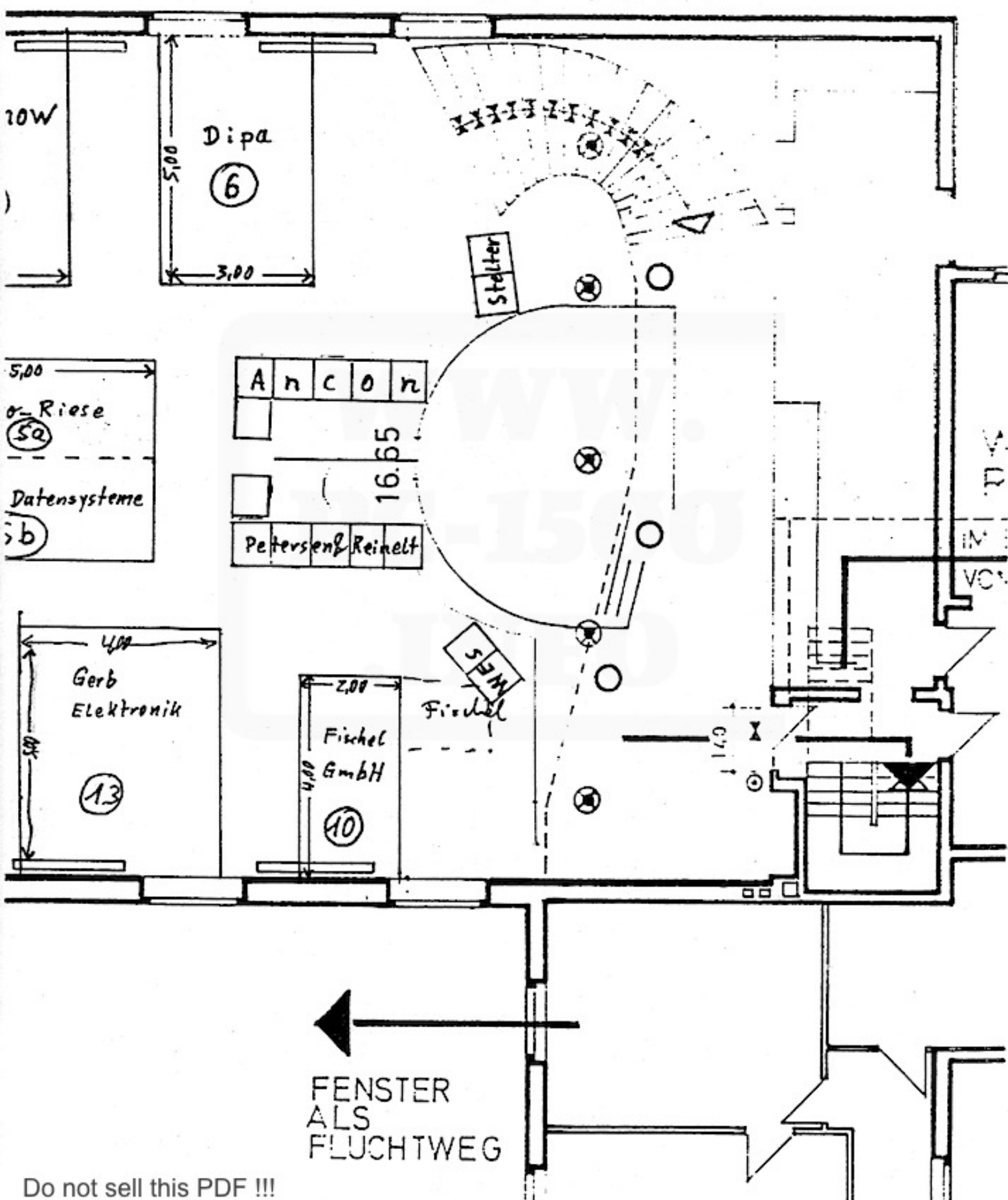
1. OBERGESCHOSS

S



Die 2. Computerbörse findet
um die Weihnachtszeit statt.
Warum, wohl?

1983



von
Marco Feussi, Schweiz

Vielleicht ist Dir beim Programmieren schon aufgefallen, dass der SHARP gewisse Fehler im Betriebssystem aufweist. SHARP hat diese Fehler nach und nach ausgemerzt. Die Original-Version des ROM-Chip trägt die Nr. A01. Die 1. Revision am ROM-Chip trägt die Nr. A03 und die letzte Revision trägt die Nr. A04. Prüfe nun welche Version Du hast.

Version	Peek & C443	Peek & C5BD
A01	56	129
A03	59	129
A04	59	74

Die meisten werden mit Peek & C443 59 und mit Peek & C5BD 129 erhalten, d.h. sie haben die Version A03.

Kleine Verbesserungen im ROM

1. Wenn man bei der Version A01 ein Programm automatisch mit ARUN starten lässt, dann wird das BUSY Signal in der Anzeige nicht aktiviert. Dies wurde bei der Version A03 korrigiert.
2. Wird bei der Version A01 und A03 nach der Formatierungsanweisung USING eine Variante angehängt, z.B. USING A, so führt das zu einem "Unfall". In der Anzeige erscheint NEW 0 : CHECK, oder ein Maschinenprogramm wird gestartet. Die Version A04 arbeitet fehlerlos.

⊕ Dieser Beitrag wurde aus seiner Club-Zeitschrift entnommen. Hierfür bedanken wir uns.

3. Gib im PRO-Mode 10 GCURSOR 152: INPUT A ein. Wen das Programm gestartet wird, so erscheint ERROR 32 IN 10 in der Anzeige. Bei der Version A01 lässt sich die Anzeige nicht mit CL löschen. Bei der Version A03 wird die Anzeige mit CL wohl gelöscht, jedoch wartet der Computer immer noch mit einem Teil von einem Fragezeichen auf eine Eingabe. Die Version A04 arbeitet korrekt.

Wichtige Verbesserungen im ROM

1. Den Gebrauch der Booleanschen Funktionen erzeugt bei der Version A01 und A03 Fehler.

z.B. NOT NOT 1 [ENTER] - 31233. Richtig wäre 1

1 OR (-1 OR 1) [ENTER] 31233. Richtig wäre -1

1 AND (-1 OR 1) [ENTER] 0. Richtig wäre -1

die Version A04 arbeitet einwandfrei.

2. Wer bei den Versionen A01 und A03 der Tangens von $100/9 = 11.11111111$ im GRAD - Modus ausrechnet, erhält 0. Z.B. TAN (100/9) [ENTER] 0. Richtig wäre 0.1763269807. Oder wenn der Sinus von 0.174532925199 (im RADIAN-Mode) in einer Rechnung vorkommt, so erhält man 0. Z.B. SIN ($\pi/18$) [ENTER] 0. Richtig wäre 0.1736481777. Wird die Zahl direkt eingegeben, z.B. SIN 0.1745329252 (= $\pi/18$) so erhält man das richtige Ergebnis. Oder wenn zuerst $A = \pi/18$ und dann SIN A gerechnet wird, erhält man das korrekte Resultat. Auf einem zur Bedienungsanleitung beigelegten Blatt wird auf diesen Fehler aufmerksam gemacht. Die Version A04 arbeitet fehlerlos.

3. Die ROM Version A04 hat 2 wichtige Aenderungen erfahren, die das SHARP Basic zu andern Basic Versionen kompatibler macht. Die 1. Aenderung betrifft das IF Statement. Zum Beispiel: 10: IF A THEN 50. Bei der Version A01 und A03 wird, wenn der Wert in A grösser als Null ist, zur Zeile 50 verzweigt. Ist A aber kleiner oder gleich Null, dann wird das Programm bei der nächsten Zeile fortgesetzt.

Bei Version A04 hingegen wird bei Zeile 10 wie folgt verfahren: Wenn der Wert in A ungleich Null ist, so wird das Programm bei Zeile 50 fortgesetzt und im andern Fall wenn A = 0 ist, wird zur nächsten Zeile verzweigt.

Die 2. Aenderung betrifft die FOR/NEXT Schleife. Folgende Beispiele sollen die Neuerung zeigen.

A. FOR X = 0 TO 3	Version A01, A03	A04
X-Wert in der Schleife	0, 1, 2, 3	0, 1, 2, 3
X-Wert nach der Schleife	3	4
B. FOR X = 0 TO 5 STEP 2		
X-Wert in der Schleife	0, 2, 4, 6	0, 2, 4
X-Wert nach der Schleife	6	6

Vorsicht bei der NOT Funktion im Zusammenhang mit dem IF Statement. 1. Beispiel soll zeigen warum.

Gib im Programm-Modus 10 IF NOT (1 = 2) BEEP 1 ein.

1 ist nicht gleich 2. Also müssen wir beim Programmablauf einen Ton hören. Nichts ist zu hören. Dies gilt nur für die Version A01 und A03, Version A04 funktioniert korrekt. Wenn man folgende Ergänzung macht, dann wird man einen Ton hören. Korrigiere die Zeile 10 wie folgt:

10 IF NOT (1 = 2) + 2 BEEP 1. Jetzt werden wir einen Ton hören. Bei IF NOT (1 = 1) BEEP 1 werden wir keinen Ton zu hören bekommen, weil 1 gleich 1 ist wird zur nächsten Zeile verzweigt.

Mit kollegialem Gruss

Mario Fueni

Laden der Programme

Die Programme werden folgendermassen in den Rechner eingelesen:

- a) Maschinenprogramme; CLOAD M <Programmname>; <Startadresse>
- b) BASIC-Programme; CLOAD <Programmname>

Eine Liste der Programme finden Sie auf der letzten Seite. Dort ist auch angegeben, wie lange jeweils der Ladevorgang dauert, damit Sie die Programme auf Cassette auffinden.

Ich empfehle Ihnen, jeweils den Zählerstand des Cassettengeräts in die Liste einzutragen. Weiter ersehen Sie in der Liste auch, in welcher Ausgabe von Mikro+Kleincomputer das jeweilige Programm beschrieben bzw. erwähnt worden ist. Sollten Sie Schwierigkeiten beim Einlesen haben, versuchen Sie es vorzugsweise einmal mit einem anderen Cassettengerät.

GEBRAUCHSANLEITUNG ZUM HEXMONITOR

Der Hexmonitor ist genau $\frac{1}{2}$ K-Byte lang (=512=200 Bytes). Da der Monitor relokatable (im Speicher frei verschiebbar) ist, kann man ihn irgendwo im RAM-Speicher plazieren:

CLOAD M"HEXMONITOR";<Startadresse>

Meistens wird man den Speicherraum für den Monitor jedoch mit NEW (siehe Anhang in Teil 1: B3/3) vor BASIC-Zugriffen schützen:

kein bzw. 4K-Modul: NEW &42C5 → CLOAD M"HEXMONITOR";&40C5

mit 8K-Modul: NEW &38C5 → CLOAD M"HEXMONITOR";&38C5

Starten des Hexmonitors

Der Einsprung erfolgt auf die erste Adresse des Monitors. Wenn der Monitor wie angegeben plaziert wurde, startet man also mit CALL &40C5 bzw. mit CALL &38C5.

Ideal ist es, wenn man diesen Befehl einer RESERVE-Taste zuordnet: **SHIFT MODE** F1:CALL &40C5

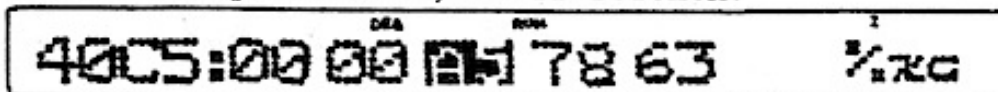
Wenn der Monitor verlassen worden ist, meldet sich der Rechner immer mit >; wenn der Monitor von einem BASIC-Programm aufgerufen worden ist, wird dieses ohne Meldung abgebrochen und kann auch mit CONT nicht mehr gestartet werden. Wenn ein BASIC-Programm vor dem Monitorkauf unterbrochen wurde, kann

All and more about Sharp PC-1500 at <http://www.PC-1500.info>
es jedoch nachher mit CONT weitergeführt werden.

Bedienung

a) Anzeigeformat

Links in der Anzeige wird die Adresse des angesprochenen Bytes angezeigt. Getrennt durch Doppelpunkt folgen fünf Bytes; das dritte ist das momentan Angesprochene, das heisst jenes mit der angezeigten Adresse. Es wird durch inverse Darstellung hervorgehoben. Danach folgen die fünf, den fünf angezeigten Bytes entsprechenden, ASCII-Zeichen:



40C5:00 00 00 78 63 /XG

Dieses Anzeigeformat hat den grossen Vorteil, dass man gute Uebersicht hat, z.B. kann man eine 4-Byte-Instruktion vollständig auf der Anzeige sehen.

b) Editierfunktionen

↓ bzw. **↑** Anzeige-"Fenster" in Richtung höhere bzw. niedriger Adresse verschieben. (Repetierfunktion eingebaut)

0..9,A..F Hex-Eingabetasten... im Inversfeld werden die letzten zwei eingetippten Ziffern angezeigt, gespeichert sind jedoch die letzten vier. (Wichtig bei M und Z !)

M Die vier zuletzt eingetippten Zeichen werden als neue Adresse genommen.

Q Das momentan angesprochene Byte wird in die nächste Speicherstelle kopiert und die Adresse erhöht. (Repetierfunktion eingebaut)

RCL Mit dieser Taste kann zwischen der momentanen und der letzten mit **M** adressierten Speicherstelle hin und her geschaltet werden. Dies ist sehr praktisch, wenn man z.B. zwei Programmstellen vergleichen will.

ENTER Die letzten zwei eingetippten Hexadezimalziffern werden als Daten in die momentan angesprochene Speicherstelle eingeschrieben und die Adresse um eins erhöht.

Z Die letzten vier Hexadezimalziffer werden als Grenz-Adresse für alle folgenden I und K Befehle aufgefasst.

All and more about Sharp PC-1500 at <http://www.PC-1500.info>
 Fügt ein freies Byte ein. Alle Bytes vom momentan angesprochenen bis zur Grenzadresse (s. **Z**) werden um eins verschoben.

I

K

Löscht das momentan angesprochene Byte und rückt alle folgenden bis zur Grenzadresse (s. **Z**) um eins nach.

Y

Es wird das nächste Byte, das den gleichen Wert hat wie das momentan Angesprochene, im Speicher gesucht. Wenn bis &FFFF kein Byte dem vorgegebenen entspricht, bricht die Suche ab. Mit RCL kann die Adresse abgerufen werden, die vor dem Druck auf Y angezeigt wurde.

ON

Dient zum Verlassen des Monitors.

Wichtig: Vor Anwendung von I und K stelle man mit **Z** die Grenzadresse so ein, dass sich der Monitor auf keinen Fall selber verschiebt. Andernfalls gibt es einen Totalabsturz mit anschließender Tastenblockierung.

c) Programme mittels Monitor starten

SHIFT G

Startet ein Programm ab der angezeigten Adresse. Das BUSY-Zeichen erscheint in der Anzeige.

ON

Unterbricht ein laufendes Programm, wenn am IRQ-Vektor und dem I-Flag nichts geändert worden ist.

R

Bringt die Adresse &7BA0 zur Anzeige. Die folgenden acht Bytes sind den Registern wie folgt zugeordnet:
 XH XL YH YL ZH ZL A CC

Beim Programmstart werden die Werte in diesen Bytes (ausser CC) in die Register übertragen. Wenn ein Anwenderprogramm mit RTS (= &9A) aufhört, werden die Registerinhalte wieder nach &7BA0 ff übertragen.

Bemerkungen

- Während langdauernden Operationen wie z.B. I, K oder Y erscheint das BUSY-Zeichen in der Anzeige.
- Der Monitor benützt den OUPUT-Buffer (&7B60-&7BAF) und dazu die Speicherstellen &7923/&7924.

Falls Sie über kein Cassetteninterface verfügen, können Sie den Hexmonitor selbstverständlich auch von Hand programmieren, was allerdings höchste Konzentration erfordert. Es empfiehlt sich hierzu zwei RESERVE-Tasten mit POKE& und die andere mit ,& zu belegen. Nachfolgend das Listing des Hexmonitors (im Beispiel von &9CF5 bis &9EF4).

```
relokatibler
Hexmonitor
-----

Hex-Dump
-----

von &9CF5 - &9EF4

&9CF5 : 05 78 03 18 03 C3 1A FD
&9CF6 : E3 F8 00 00 00 78 4F E3
&9CF7 : 78 0A 00 FD 0E E3 78 0A
&9CF8 : 00 E3 78 0F 00 FD 58 00
&9CF9 : 78 0A 00 34 FD 58 0E CE
&9CF0 : 14 FD 58 0E C3 03 30 61
&9CF1 : 00 28 0A 04 36 56 30 33
&9CF2 : 00 08 FD 0A FD 58 0E 06
&9CF3 : 00 28 FD 2A 03 28 41 00
&9CF4 : 12 FD 1A FD 58 28 56 36
&9CF5 : 0A 04 53 03 02 05 28 41
&9CF6 : 00 00 03 00 41 FD 1A 13
&9CF7 : 0E 78 00 18 78 4A 08 0E
&9CF8 : 0F 44 0A 13 48 71 4A 34
&9CF9 : 05 00 0F 41 00 0E E3 78
&9CF0 : 4E F8 0E E2 42 01 02 C0
&9CF1 : 42 E8 76 4E 01 0E E0 7D
&9CF2 : 01 20 FD C8 03 78 0A F1
&9CF3 : 23 F8 2A 03 78 0F F1 28
&9CF4 : 03 0F F3 22 0A 78 0A 04
&9CF5 : 00 F8 28 FD 0A F3 02 0A
&9CF6 : 78 0F 0E 78 0A 08 78 0A
&9CF7 : 00 FD 58 0E 41 0E 34 FD
&9CF8 : C8 FD 58 03 21 FD C0 50
&9CF9 : 00 FD 0A F7 00 03 41 00
&9CF0 : 0E 5C 53 03 FD 58 FD 0A
&9CF1 : FD CA 03 47 FD 62 FD EA
&9CF2 : FD 08 FD 5C 00 46 0A 4C
&9CF3 : 40 58 00 33 43 E3 48 FD
&9CF4 : 51 4F 58 F8 52 08 67 03
&9CF5 : 18 65 5A 7C 5E 0F FD C8
&9CF6 : 03 F8 F1 FD 0F FD 58 0E
&9CF7 : 44 FD 0A FD 0A 03 0F 00
&9CF8 : 18 07 3A 01 02 03 06 01
&9CF9 : 44 44 FD 5C 56 E3 78 0E
&9CF0 : 48 0A 54 0E 00 55 1E 3E
&9CF1 : 0C 03 78 0A 51 0A 54 0A
&9CF2 : 78 0A 03 78 0A 18 14 0E
&9CF3 : 78 0A 03 78 0F 1A 0A FD
&9CF4 : 18 03 78 0A 18 04 0E 78
&9CF5 : 0A 03 78 0A 1A 04 0E 78
&9CF6 : 08 0A 03 78 0E 0E 7A 23
&9CF7 : 05 78 0F 0A 7A 24 5A FD
&9CF8 : 58 38 38 38 38 33 36 FD
&9CF9 : CA 68 78 0A 0A 03 53 61
&9CF0 : 04 01 04 61 03 02 FD 0A
&9CF1 : 08 0A FD 01 FD 58 03 13
&9CF2 : 00 CA FD 58 FD 08 FD 58
&9CF3 : 0A 78 3F FD 0A FD 1A FD
&9CF4 : 2A FD 0A 0A 78 47 5A 0A
&9CF5 : 78 0A FD C8 FD 0A FD 58
&9CF6 : 00 08 FD 0A 0E 78 07 0A
&9CF7 : 78 48 FD 10 1E 02 34 0E
&9CF8 : 78 0A 14 0E 78 03 38 78
&9CF9 : 5A 08 3A 03 7A 22 08 03
&9CF0 : 7A 24 0A 48 43 43 04 36
&9CF1 : 13 07 04 10 03 00 43 00
&9CF2 : 5A 0E 12 FD 18 44 47 41
&9CF3 : 04 07 2A 24 53 03 04 07
&9CF4 : 7A 23 13 0F 0A 05 78 0A
&9CF5 : 00 10 44 F7 08 0E 4C 0A
&9CF6 : 13 07 4E 0A 55 08 30 E3
&9CF7 : 78 01 48 F3 46 54 0A 78
&9CF8 : 0A 14 0A 78 08 5D 5A 5A
```

Bezeichnung des Programms	BASIC	Maschinensprache	relativ	erschienen in	verg. Zeit bis zu diesem Programm	Zeilen	Adressen	Bytes	Zeit	Zähler
BEFEHLSLISTEN	x			3	0'06	90-143		374	0'37	
HEX-DUMPS	x			3	0'49	140-214		412	0'40	
HEXMONITOR		x	x	(3)	1'35		&40C5- &42C4	512	0'47	
DEZ-HEX-UMWANDL.		x		4	2'28		&40C5- &40EF	43	0'14	
AUTOR.+OFF		x	x	4	2'48		&40C5- &40D6	18	0'12	
GERAEUSCHE		x	x	4	3'07		&40C5- &40EB	39	0'14	
SPREIZSCHRIFT		x	x	4	3'26		&40C5- &40FB	55	0'15	
SPREIZSCHRIFT	x			3	3'48	10-100		146	0'22	

Erklärungen:

- Die Ziffer in der Spalte "erschienen in" gibt an, in welcher Ausgabe von Mikro+Kleincomputer im Jahrgang 83 das jeweilige Programm erklärt, aufgelistet oder erwähnt wurde.
- In der Spalte "verg. Zeit bis zu diesem Programm" steht, wieviel Zeit, (in Min. und Sek.) vom Beginn der Cassette (inkl. Vorspannband) bis zum Beginn dieses Programms auf Cassette gemessen, vergangen ist.
- Die Spalte "Zeit" gibt über die Dauer des Ladevorgangs Auskunft.
- In der Spalte "Zähler" wird der Zählerstand zu Beginn des Programms notiert, um das Programm später schnell aufzufinden.

chtung: Alle Maschinenprogramme starten bei &40C5, sind jedoch nicht unbedingt von dort aus abgespeichert worden: Vergessen Sie also beim Zurückladen in den Rechner die Startadresse nicht:d.h.

„TVS“ Task-Verwaltung-System

Als Besitzer eines PC 1500 von Sharp können Sie mit dem Befehl „MERGE“ mehrere Programme in den Rechner laden.

Wie Sie vielleicht schon bemerkt haben, ist das aber mit einigen erheblichen Nachteilen verbunden, z.B.:

- bei gleichen Labels in den verschiedenen Programmen, erreichen Sie mit Def.-Key's immer nur das erste Label
- eine Programm-Änderung ist nur beim letzten mit „MERGE“ zugeladenen Programm möglich
- dimensionierte Variablen mit gleichen Namen können in einem anderen geladenen Programm nicht noch einmal angelegt werden
- wenn in einem Programm der Befehl „CLEAR“ abgearbeitet wird, werden auch die Variablen und Datenfelder gelöscht, die von anderen Programmen belegt wurden und deren Inhalt evtl. wieder gebraucht wird
- der Befehl „NEW“ z.B. löscht **alle** mit „MERGE“ geladenen Programme
- der Befehl „RUN“ startet nur das 1. Programm, löscht aber sämtliche dimensionierte Variablen

Diese und noch einige andere Nachteile haben Sie bei der Verwendung der Software „TVS“ nicht, wenn Sie mehrere Programme gleichzeitig im PC 1500 haben wollen.

Voraussetzung ist ein 8-K-Zusatzmodul (eine individuelle Programmanpassung für größere Module ist auf Wunsch möglich).

Sie können mit TVS z.B. 3 voneinander unabhängige Daten- und Programmspeicherbereiche einrichten (Task). Jedem Bereich stehen dann 3070 Bytes für Programm und Datenbereich zur Verfügung. (Bei nur 2 generierten Bereichen sind entsprechend mehr Bytes zur Verfügung, z.B. je 4605 Bytes). Es können aber auch 3 bzw. 2 verschieden große Bereiche eingerichtet werden. Das Einrichten der Bereiche erfolgt im Dialog und kann von Ihnen ohne besondere Kenntnisse jederzeit und beliebig oft vorgenommen werden.

Wenn Sie die Task's angelegt haben, können Sie Programme (auch Programme in Maschinensprache) von der Cassette in die Task mit „CLOAD“ laden oder aus der Task auf die Cassette mit „CSAVE“ wie gewohnt save.

Wenn Sie Ihr Programm arbeiten lassen wollen, wählen Sie die Task über Reserve-Key an, in der sich das Programm befindet und starten wie üblich das Programm. Der Taskwechsel erfolgt ebenso mit Reserve-Key.

Sie können in der aktiven Task im Programm-Mode jetzt auch Ihr Programm ändern, ergänzen, listen oder mit „NEW“ ganz löschen, ohne daß die Programme und Variablen in den anderen Task's beeinflusst werden. Der CLEAR-Befehl in einem Programm löscht die Speicher nur in der aktiven Task.

Sie haben somit also 3 Computer in einem.

Das „TVS“-Programm wird auf Cassette mit ausführlicher Bedienungsanleitung geliefert.

SHARP

Intercept

**Spariamo a ufo
e astronavi
con la PC-1500**

di E. Cima

Per questo gioco non occorre la stampante supplementare CE-150 ma occorre l'espansione di memoria di 4 Kbyte.

Il gioco consiste nel distruggere 10 astronavi nemiche nel minor tempo possibile spostando un intercettore sopra una base lunare.

Dopo aver dato il RUN apparirà sul display la scritta GAME START. Sarà poi necessario attendere 30 sec. che serviranno per la preparazione della base lunare: infatti la base viene disegnata casualmente.

Dopo la completa visualizzazione della base lunare appariranno nel centro del display (nella terza riga dal basso) l'intercettore (che è rappresentato da un puntino come le astronavi invadenti). Ai lati dell'intercettore verranno visualizzati due UFO che si muovono casualmente in tutte le direzioni possibili, mentre l'intercettore si muove con i tasti 2, 4, 6, 8:

con il tasto 2 si fa scendere l'intercettore;
con il tasto 8 si fa salire l'intercettore;
con il tasto 4 si sposta l'intercettore indietro (sinistra);
con il tasto 6 si sposta l'intercettore avanti (destra);
con il tasto P si spara avanti (destra);

con il tasto / si spara indietro (sinistra).

Per poter dare al computer questi ordini occorre tenere premuto il tasto fino a quando il computer non li ha eseguiti.

Quando l'intercettore ha distrutto 10 UFO nemici, sul display viene visualizzato il tempo impiegato per portare a termine la missione.

Quando un UFO viene colpito ricompare in un punto a caso del display (qualche volta ricompare sotto il raggio appena sparato dall'intercettore: in questo caso si avrà colpito l'UFO per due volte). Quando un UFO viene colpito il computer emette 10 BEEP.

Il tiro dell'intercettore è di una lunghezza pari a 50 puntini.

Se si vuole aumentare il numero degli UFO da abbattere per vincere sarà sufficiente variare le stringhe 5012 e 5112 cambiando 10 con il numero desiderato.



```

1: PAUSE "GAME START";
5: WAIT 0: D=70: C=
16: CC=40: DD=2:
EE=2: FF=90
10: DIM A(150): KL=
TIME: TIME =0:
KJ=TIME
20: RANDOM
30: FOR X=0 TO 150:
A(X)=RND 3:
NEXT X
40: FOR X=0 TO 150
41: IF A(X)=1 LET A
(X)=64
42: IF A(X)=2 LET A
(X)=96
43: IF A(X)=3 LET A
(X)=0
44: NEXT X
45: FOR X=0 TO 150:
GCURSOR X:
GPRINT A(X):
NEXT X: BEEP 10
50: A$="": A$=
INKEY$
60: IF A$="6" LET D
=D+1: IF D>150
LET D=150
61: IF A$="4" LET D
=D-1: IF D<1 LET
D=1
62: IF A$="8" LET C
=C/2: IF C<1 LET
C=1
63: IF A$="2" LET C
=C*2: IF C>16
LET C=16
64: IF A$="P" GOSUB
5000
65: IF A$="I" GOSUB
5100
66: T=A(D): B=TOR C
67: GCURSOR D:
GPRINT B: GOSUB
100: GCURSOR R:
GPRINT A(R)
70: GOTO 150
100: IF A$="6" LET R
=D-1
101: IF A$="4" LET R
=D+1
110: RETURN
150: AA=RND 4: BB=
RND 4
151: IF AA=1 LET CC=
CC+1: IF CC>150
LET CC=150
152: IF AA=2 LET CC=
CC-1: IF CC<1
LET CC=1
153: IF AA=3 LET DD=
DD/2: IF DD<1
LET DD=1
154: IF AA=4 LET DD=
DD*2: IF DD>16
LET DD=16
156: IF BB=1 LET FF=
FF+1: IF FF>150
LET FF=15
157: IF BB=2 LET FF=
FF-1: IF FF<1
LET FF=1
158: IF BB=3 LET EE=
EE/2: IF EE<1
LET EE=1
159: IF BB=4 LET EE=
EE*2: IF EE>16
LET EE=16
200: IF AA=1 LET OO=
-1
201: IF CC<1 THEN
LET CC=1
202: IF FF<1 THEN

```

Figura 2 - Il listato BASIC.

SHARP



```

LET FF=1
209: IF AA=1 LET O=C
    C-1
210: IF AA=2 LET O=C
    C+1
220: IF BB=1 LET K=F
    F-1
230: IF BB=2 LET K=F
    F+1
231: IF CC=1 THEN
    LET CC=1
232: IF FF=1 THEN
    LET FF=1
240: T=A(CC):GG=TOR
    DD
250: T=A(FF):II=TOR
    
```

```

EE
300: G_CURSOR CC:
    G_PRINT GG:
    G_CURSOR O:
    G_PRINT A(O)
310: G_CURSOR FF:
    G_PRINT II:
    G_CURSOR K:
    G_PRINT A(K)
320: GOTO 50
5000: FOR X=0 TO 50
    : T=A(X): P=TO
    RC
5001: IF X+D>150
    GOTO 5021
5010: G_CURSOR X+D:
    G_PRINT P: IF
    D+X=CC IF C=D
    DBEEP 10, 65,
    235: AQW=AQW+
    1: CC=RND 150
5011: IF D+X=FF IF
    C=EEBEEP 10,
    56, 235: AQW=A
    QW+1: FF=RND
    150
5012: IF AQW>10
    GOTO 6000
5020: NEXT X
5021: FOR X=DTO D+
    50
5022: IF X>150
    RETURN
5023: G_CURSOR X:
    G_PRINT A(X):
    NEXT X:
    G_CURSOR D:
    G_PRINT C:
    RETURN
5100: FOR X=0 TO 50
    : Y=D-X: T=A(X
    ): P=TOR C
5101: IF Y<0 GOTO 5
    125
5110: G_CURSOR Y:
    G_PRINT P: IF
    Y=CC IF C=DD
    BEEP 10, 65, 2
    35: AQW=AQW+1
    : CC=RND 150
5111: IF Y=FF IF C=
    EEBEEP 10, 56
    , 235: AQW=AQW
    +1: FF=RND 15
    0
5112: IF AQW>10
    GOTO 6000
5120: NEXT X
5125: SD=0
5130: FOR X=DTO D+
    50: SD=SD-1:
    IF SD<0
    RETURN
5135: G_CURSOR SD:
    G_PRINT A(SD)
    : NEXT X
5140: G_CURSOR D:
    G_PRINT C:
    RETURN
6000: KO=TIME : AR=
    INT (KO/100)
    : RA=(KO-AR*1
    00)*10000
6010: SD=INT (RA/1
    0000): MS=INT
    ((RA-SD*1000
    0)/100)
6020: SE=INT (RA/1
    00): SE=RA-SE
    *100.
6030: WAIT 100:
    PRINT " HAI
    IMPIEGATO :
    "
6040: WAIT 300:
    PRINT SD; " "
    ;MS;" ";SE:
    TIME =KL+KO
6050: INPUT "UOI
    RIPETERE ";
    U$
6060: IF U$="SI"
    THEN CLEAR :
    GOTO 1
6070: END
    
```

- 1-45 Viene calcolata la generazione casuale dei puntini che formeranno la base lunare e la sua visualizzazione.
- 50-70 Viene richiesto il comando per lo spostamento dell'intercettore e l'esecuzione di questo comando.
- 100-110 Sono dei passi che servono per la cancellazione della scia che l'intercettore lascia muovendosi.
- 150-320 Vengono calcolate dal computer le posizioni casuali delle navicelle spaziali degli invasori.
- 5000-5023 Viene calcolato il tiro in avanti e anche la sua visualizzazione.
- 5100-5140 Viene calcolato il tiro indietro e anche la sua visualizzazione.
- 6000-6040 Viene calcolato il tempo impiegato.
- 6050-6070 Viene richiesto se si vuole ripetere il gioco; se si, il programma riprende dall'inizio altrimenti finisce.

Quest'anno si è formato il FORTH Interest Group Italia (FIG Italia), una associazione nata allo scopo di favorire lo scambio di esperienze e di informazioni tra utenti FORTH.

FIG Italia è affiliata alla associazione americana degli utenti FORTH, il FORTH Interest Group (FIG). L'iscrizione annuale al FIG Italia dà diritto a ricevere gratuitamente la rivista FORTH Dimensions pubblicata da FIG.

Le persone interessate possono contattare Marco Tausel - Via G. Forni, 48 - 20161 MILANO.

Figure 1 - Le principali parti in cui si suddivide il programma.

Ich kann auch kein Französisch, Du etwa
Nee!

des nombres pris en sandwich

OU

le tri rapide en Basic

Comment « trier » une liste ? Ce problème se pose souvent, et, pour ne citer que cela, classer par ordre croissant une suite finie de nombres réels (t_0, t_1, \dots, t_n) est une préoccupation si classique en informatique qu'elle constitue souvent le premier travail de programmation confié à un étudiant. Voici un exemple de tri écrit en Basic standard et programmé sur un PC 1500.

Certains langages (ceux que l'on appelle « récursifs », comme le Pascal) permettent une procédure de tri très rapide, connue sous le nom de « Quick sort » (tri rapide) ou de « Shell sort » (tri de Shell) — du nom de son inventeur —, qui est exposée dans les articles sur les méthodes de tri de L'OI n° 33 de décembre 81 et de L'OI n° 34 de janvier 82.

Le Basic n'est pas récursif ; pourtant, pour trier une liste, on peut simuler la récursivité par un programme très simple, qui constitue un réel progrès sur les méthodes venant spontanément à l'esprit, comme le « tri à bulles » ; D. Knuth a même démontré que le tri de Shell était, en moyenne, plus efficace que tout autre algorithme.

Voici un exemple. Partant de la liste :
(4, -9, 213, -7, 90, 1/2, 2),
il faut parvenir à :
(-9, -7, 1/2, 2, 4, 90, 213).

Le principe consiste à ramener le tri d'une liste de longueur n (ici $n = 7$) à celui de listes plus courtes, par exemple 4 et 3. Dans le cas énoncé, deux échanges (4 et 2 d'une part, 213 et 1/2 d'autre part) conduisent à deux sous-listes ou zones de longueur 4 et 3 :
(2, -9, 1/2, -7) et (90, 213, 4),
qui ont la propriété suivante : tout nombre de la première zone est inférieur ou égal au nombre de la seconde. Par suite, trier chacune de ces zones conduit au résultat recherché, ici :
(-9, -7, 1/2, 2) et (4, 90, 213).





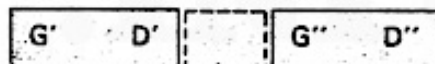
En cours de route, on opérera autant de permutations de nombres $T(K)$ qu'il le faudra pour que la règle suivante soit respectée : si K est l'indice d'un entier compris entre G (au sens large) et I (au sens strict), on a toujours : $T(K) \leq X$, alors que, pour $J < K \leq D$ on a : $T(K) \geq X$.

Quand on arrive à $I > J$, généralement par $I = J + 1$ (plus rarement $I = J + 2$), les nombres de la zone (G, J) sont tous inférieurs ou égaux à ceux de la zone (I, D) , avec, dans le cas rare où $I = J + 2$, le nombre $T(J + 1) = T(I - 1)$ pris « en sandwich » entre les deux familles, d'où les égalités : $G' = G, D' = J, G'' = I, D'' = D$.

En informatique récursive, on peut, à l'intérieur d'un programme « TRIER DE $N = A$ à $N = B$ », appeler ce programme pour réaliser, par exemple, « trier de 1 à 4 », puis « trier de 5 à 7 ». En Basic c'est impossible, mais il y a une façon, un peu subtile il est vrai, d'y remédier.

Voici l'opération standard : étant donné deux indices, G et D - avec $1 \leq G < D \leq N$ -, un algorithme permet de scinder la zone (G, D) , si elle comporte plus de deux éléments à trier, en deux zones (G', D') et (G'', D'') , plus courtes, telles que tout nombre de la première zone soit inférieur ou égal à tout nombre de la seconde ; en général, on a : $G = G' < D' < D' + 1 = G'' < D'' = D$ (schémas ci-dessous).

situé à la place d'indice $D' + 1 = G'' + 1$ est à la fois supérieur ou égal à tous ceux de (G', D') et inférieur ou égal à tous ceux de (G'', D'') , suivant le schéma :



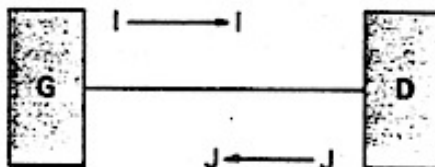
Dans le programme ci-contre, cette procédure fondamentale est décrite aux pas 60 à 120 inclus ; en notant $T(1), T(2), \dots, T(N)$ (N nombres à trier), on calcule un « pivot » :

$$X = \frac{T(G) + T(D)}{2}$$

moyenne des nombres situés à gauche et à droite de la zone (D, G) . Deux « pointeurs » I et J , partant respectivement de G et de D , partent à la rencontre l'un de l'autre jusqu'à se croiser :



Parfois, on a $G'' = D' + 2$ et non $D' + 1$, mais alors le nombre



Programme de tri rapide

TRI RAPIDE :

Copyright D.I. et l'auteur.

```

10: "A" CLEAR :
   INPUT "N=", N
20: DIM T(N), G(N/2), D(N/2)
30: FOR I=1 TO N:
   PAUSE "T("; I; "
   )=": INPUT T(I)
   : NEXT I
40: P=1: G<1>=1: D<1>=N
50: G=G<P>: D=D<P>:
   P=P-1
60: I=G: J=D: X=(T<G>
   )+T<D>)/2
70: IF T<I><X LET I
   =I+1: GOTO 70
80: IF T<J>>X LET J
   =J-1: GOTO 80
90: IF I<=J LET T=T
   <I>: T<I>=T<J>:
   T<J>=T<I>: I=I+1: J
   =J-1
100: IF I<=J THEN 70
110: IF I<D LET P=P+
   1: G<P>=I: D<P>=
   D
120: IF G<J LET D=J:
   GOTO 60
130: IF P<>0 THEN 50
140: BEEP 4
150: FOR I=1 TO N:
   PAUSE "T("; I; "
   )=": T<I>: NEXT
   I
160: END
    
```

Jeu

Explorons la grotte piégée

Vous avez été désigné pour faire partie d'une expédition scientifique sur une planète récemment découverte par les astronomes. Sur cette planète, vous avez trouvé une immense grotte que vous avez décidé d'explorer.

Ayant pénétré dans la grotte, vous remarquez que certaines commandes de votre navette spatiale ne réagissent plus normalement. Il vous est devenu impossible d'inverser le sens de marche ou même de ralentir la navette. Vous ne pouvez donc plus qu'agir sur les réacteurs qui règlent la hauteur; il s'agit maintenant de traverser cette grotte en évitant ses parois sinueuses. Il peut vous arriver aussi d'avoir à détruire des stalactites qui vous barrent le passage. Si vous n'arrivez pas à éviter une des stalactites ou une des parois, votre navette n'en sera pas pour autant détruite, elle sera simplement télétransportée en arrière. Si, par malchance, elle sera de nouveau télétransportée en arrière et ainsi de suite jusqu'au début de la grotte. Mais un essai vous en dira sûrement plus qu'un long discours!

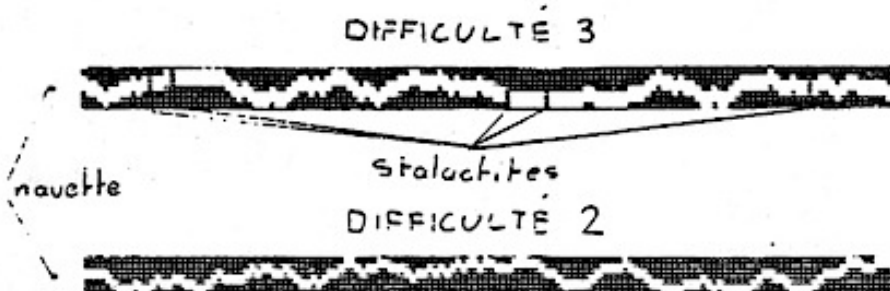
Après avoir saisi le programme et vérifié que vous n'avez pas fait de fautes de frappe, tapez RUN (enter), le programme vous demande alors la vitesse qu'aura votre navette (entre 1 et 10), 10 étant la plus rapide. Il vous demande ensuite le degré de difficulté; vous avez le choix entre:

1. Grande grotte (facile)
2. Petite grotte (difficile)
3. Grande grotte avec stalactites (très difficile)
4. Petite grotte avec stalactites (c'est de la folie!)

Si vous choisissez la difficulté 1 ou 2, il vous demandera encore si, lors des accidents, les obstacles doivent être effacés après le choc. En difficulté 3 et 4, le programme met automatiquement cette option affirmative, car il vous faudra détruire les stalactites en utilisant

votre navette comme si c'était un burin; n'ayez pas peur, elle est très solide!

Le programme dessine sur l'écran de votre Sharp une grotte aléatoire, puis positionne votre navette tout à gauche (voir croquis).



Il ne vous reste plus qu'à diriger la navette avec les touches Q pour monter et Z pour descendre. Lorsque vous aurez réussi à traverser la grotte (si, c'est possible!), votre score sera affiché

et vous pourrez recommencer en augmentant les difficultés. Et maintenant, bonne chance!

Ivano Gazzetta

L'explorateur
Programme pour Sharp PC-1500
Auteur Ivano Gazzetta, Copyright
L'Ordinateur Individuel et l'auteur

LISTING EXPLORER

```

10:REM INITIALIS
ATIONS
20:REM GRAPHISME
S NAVETTE
30:DATA 1,2,4,8,1
6,32,64:DIM H(
?):FOR I=1TO 7
?:READ H(I):
NEXT I

```

```

40:C=0:H=4:A=3
50:INPUT "Vitesse
(1-10):";Q:Q=
10-Q
60:INPUT "Difficu
lte (1-4):";P
65:IF P>2LET FL=1
:OS="O":GOTO 1
00
70:INPUT "Points
effacés (O-N):
";OS
80:REM DATAS DES
SIN GROTTES
90:REM DIFFICULT
E 1-3
100:DATA &78,&71,&
63,&47,&8F
110:REM DATA DIFF
ICULTE 2-4
120:DATA &7C,&79,&
73,&67,&4F,&1F
130:DIM G(6)
140:FOR I=1TO 5:
READ G(I):NEXT
I
150:IF P=1GOTO 170
160:FOR J=1TO 6:
READ G(I):NEXT
I
170:RANDOM
180:REM DESSIN GR
OTTE
190:WAIT 0:CLS
200:GDCURSOR 6
210:FOR I=6TO 150
220:A=A+SGN (RND (
3)-2):A=A+(A<1
)-(A>4+P)

```

```

230:GPRINT G(A);:
NEXT I
240:E1=1
250:IF FL<>1THEN 2
70
260:FOR J=1TO 5:
GDCURSOR (RND 1
44)+6:GPRINT I
27:NEXT I
270:E1=1
280:GDCURSOR 0
290:REM DEBUT
300:TIME =0:WAIT 0
310:IF ((POINT C)
AND (H(H)))=H(
H)THEN GOTO 41
0
320:GDCURSOR C:
GPRINT (POINT
C+H(H))
330:H1=H:BEEP 1,10
0,1
340:A$=INKEY$
350:IF A$='Q'AND H
>1THEN LET H=H
-1
360:IF A$='Z'AND H
<7THEN LET H=H
+1
370:GDCURSOR C:
GPRINT (POINT
C+H(H))
380:C=C+1:IF C<155
GOTO 310
390:BEEP 5:WAIT .
PRINT "Score .
",5*INT (((200

```

(Suite page XIII)

digital

apple computer

VOTRE DISTRIBUTEUR OFFICIEL

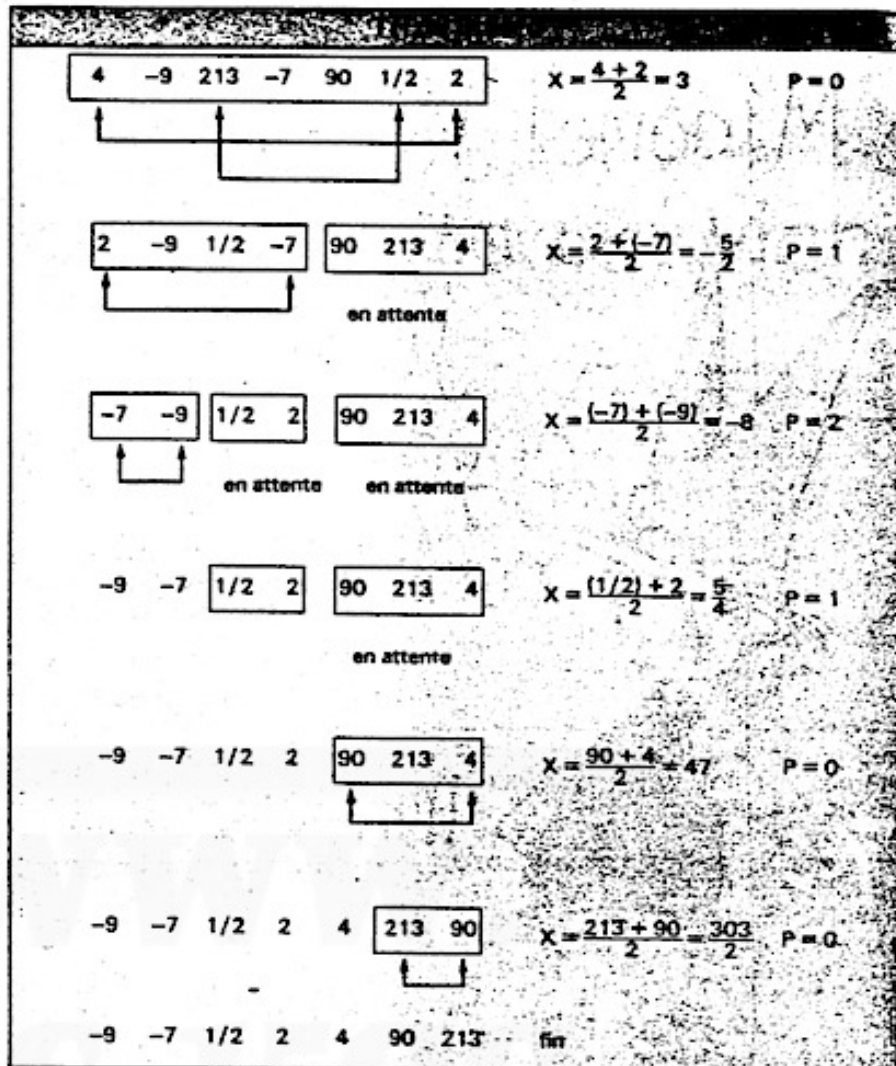
Radio **electro**

Confiance + Service
22-26 passage Malbuisson
1211 Genève 11 - ☎ 022 21 35 60

Toutefois, ceci n'a pas de raison d'être si $l = D$ (ou $G = D$), car $T(D)$ majore alors tous les $T(K)$ avec $G \leq K \leq J$, ou si $J = G$ pour une raison analogue (cf les lignes 110 et 120).

La non récursivité du Basic conduit à compliquer ce schéma en introduisant un compteur, appelé ici P , mesurant le nombre de zones actuellement encore non explorées. Dans le cas général où $G < J < l < D$, on met la zone $(G, D) = (l, D)$ en « réserve » — on ne peut pas tout faire à la fois — en lui donnant un numéro P (l'ancien numéro P augmenté de 1, cf la ligne 110) et en posant $l = G(P)$, $D = D(P)$. On explorera les zones $(G(1), D(1)), \dots, (G(P), D(P))$ par indice P décroissant — quand on en aura la possibilité — jusqu'à obtenir $P = 0$ (ligne 130), auquel cas le travail est fini. En attendant, on travaille sur la zone (G, J) , rebaptisée (G, D) (cf ligne 120), jusqu'à ce qu'elle soit triée complètement.

L'indice P peut monter, dans des cas exceptionnels, jusqu'à environ $N/2$. Par exemple, si l'on considère la suite : 0, 1, 1, 2, 2,



TRI A BULLE :

Copyright O. I.
et l'auteur.

```

10: "A" CLEAR :
   INPUT "N="; N
20: DIM T(N); M=N
30: FOR I=1 TO N:
   PAUSE "T("; I; "
   )="; INPUT T(I)
   :NEXT I
40: F=0
50: FOR I=2 TO M
60: IF T(I-1) <= T(I)
   THEN 80
70: T=T(I-1); T(I-1)
   =T(I); T(I)=T:
   F=1
80: NEXT I
90: IF M=2 THEN 110
100: IF F=1 LET M=M-1:
   GOTO 40
110: BEEP 4
120: FOR I=1 TO N:
   PAUSE "T("; I; "
   )="; T(I):NEXT I
130: END
    
```

Le tri à bulles

Le programme Basic, pour PC 1500, que voici utilise la méthode la plus simple, mais aussi la plus longue, pour trier la liste $(T(1), T(2), \dots, T(N))$. Par des comparaisons systématiques du type $T(i-1) < T(i)$? on envoie, comme des bulles qui remontent successivement à la surface de l'eau, le plus grand des $T(K)$ en place N , plus le second à la place $N-1$, etc.

L'introduction du « drapeau » $F = 0$? a pour but de faire cesser les échanges d'éléments consécutifs quand un balayage a eu lieu sans rencontrer d'inversions du type $T(i) < T(i-1)$.

La variable M , variant de N à 2, n'est pas indispensable — mettre N à la place de M à la ligne 50, supprimer la ligne 90 et simplifier les lignes 20 et 100 en conséquence —, mais elle raccourcit le tri en tenant compte des « bulles » déjà en place.

4, 4, 8, ..., $2^{n-2}, 2^{n-1}, 2^{n-1}, 2^n$ pourtant bien triée dès le départ, pour laquelle $N = 2n + 2$ et P monte jusqu'à $n = (N/2 - 1)$.

En général, P/N reste petit par rapport à $1/2$. Le tri d'une liste de longueur 2 étant évident, la procédure a une fin. Pour l'exemple du début de cet article, les différentes opérations sont regroupées dans le tableau ci-dessus.

Le tri a demandé six échanges (ou transpositions) de nombres $T(K)$ deux à deux : (4, 2), (213, 1/2), (2, -7), (-7, -9), (90, 4), (213, 90).

On peut, en insérant des instructions « LPRINT » convenables, garder une trace écrite du détail du tri. Le programme a été réalisé sur Sharp PC 1500, mais est valable sur presque n'importe quel ordinateur (quelques modifications sont nécessaires si la machine n'admet pas les tableaux autres que $A(N)$). On peut l'adapter à un tri alphabétique.

André Warusfel

Langage-machine et bruits divers

(PC- 1500)

La découverte du langage-machine du PC-1500, bien que toutes ses instructions ne nous soient pas encore connues, a fait faire — pour certaines applications — un saut qualitatif certain. L'application proposée ici en est un des nombreux exemples.

■ Sharp n'ayant pas encore publié de manuel en français sur le langage-machine du PC-1500, c'est en empruntant au microprocesseur Z 80 ses mnémoniques que vous sont donnés les nouveaux codes opératoires. Ceux-ci complètent les tables publiées dans *l'Op* n° 13 (pages 30 à 33), les neuf premiers appartenant à la première des deux tables.

Nostra culpa

■ 5A n'est pas ce que l'on croyait (*l'Op* n° 13 p. 31), le voici démasqué :

LDDE, BC 5A

Rappelons qu'un code précédé de FD est celui d'une instruction de la seconde table, que le signe # indique que l'opération porte sur la

seconde page de 64 K octets de la mémoire (potentielle) du PC-1500.

—Un mini-synthétiseur—

Le programme « Synthé » est un bon exemple de programmation en langage-machine. L'auteur en expli-

que page suivante, le principe et le fonctionnement.

La routine n'est pas relogeable ; elle doit être programmée de & 7150 à & 719A, car aux endroits marqués du signe ► vont être rangées les valeurs des différents paramètres des sons à produire. Ces adresses correspondent à celles des registres

```
--MNEMON. ---HEXA--
NOP                38
RETI               8A
SPU                A8
RPU               B8
SBRU              CF
SPU               E1
RPU               E3
RUF              F3
CPIA, (BC        F7
```

<<< 2NDE TABLE >>>

```
--MNEMON. ---HEXA--
SBC#A, (BC        01
SBC#A, (DE        11
SBC#A, (HL        21
ADC#A, (BC        03
ADC#A, (DE        13
ADC#A, (HL        23
CP#A, (BC         07
CP#A, (DE         17
CP#A, (HL         27
AND#A, (BC        09
AND#A, (DE        19
AND#A, (HL        29
OR#A, (BC         0B
OR#A, (DE         1B
OR#A, (HL         2B
SBCD#A, (BC       0C
SBCD#A, (DE       1C
SBCD#A, (HL       2C
```

```
XOR#A, (BC        0D
XOR#A, (DE        1D
XOR#A, (HL        2D
TEST#A, (BC       0F
TEST#A, (DE       1F
TEST#A, (HL       2F
RBF              4C
TEST#(BC),       4D
TEST#(DE),       5D
TEST#(HL),       6D
ADD#(BC),        4F
ADD#(DE),        5F
ADD#(HL),        6F
E1              81
ADCD#A, (BC      8C
ADCD#A, (DE      9C
ADCD#A, (HL      AC
SBC#A, (         A1
ADC#A, (         A3
CP#A, (          A7
XOR#A, (         AD
TEST#A, (        AF
LDA, F           AA
LDF, A           EC
LDA, T           BA
LDT, A           CE
HALT             B1
DJ              BE
RDP              C0
SDP              C1
SRD              D3
SLD              D7
LDHL, BC        6A
```

Mini-synthétiseur

CALL E66F génère un BEEP J, L, BC ou L, B et C sont, bien sûr, trois des registres du microprocesseur lui-même.

A l'aide des huit paramètres suivants on va définir deux types de sons A et B.

Paramètre	Localisation	Sen
répétition	& 7151	A et B
durée	& 7157	A et B
fréquence maximale 1	& 7153	B
fréquence maximale 2	& 7165	B
fréquence minimale	& 7150	A et B
fréquence 1	& 717C	A
fréquence 2	& 7182	A
fréquence 3	& 718A	A

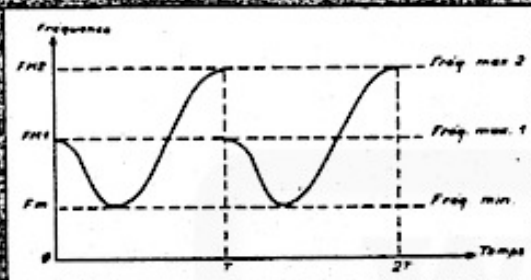


Fig. A : Programme « A »
 Dans une durée T, en partant du niveau FM1, le son va progressivement décroître jusqu'au niveau minimal Fm et remonter pour finir à FM2 et ceci autant de fois que spécifié dans le paramètre « Répétition ». Note : FM1 > Fm et FM2 > Fm

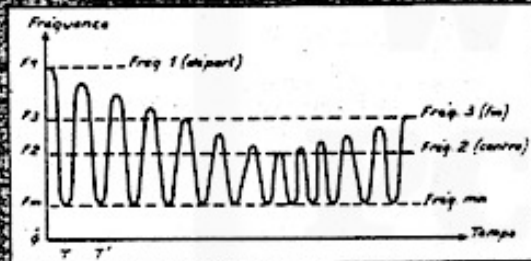


Fig. B : Programme « B »
 La période T' est répétée autant de fois que voulu grâce à « Répétition ». Note : F1 > F2 et F3 > F2.

Pour bien comprendre la signification de chaque paramètre référez-vous aux figures A et B. La localisation est l'adresse de la mémoire où est implanté le paramètre utilisé par le son. Les adresses sont situées en plein dans la routine elle-même (E71E0 à E719A).

C'est la partie basique qui demande les valeurs des paramètres et modifie en conséquence le ton.

Voici deux exemples d'application. Avec le son de type « A » (chiffres décimaux).

	Répète	F1	F2	F3	Fm	durée
démonstration	1	40	10	30	0	10
spécial	255	3	1	3	0	1

	Répète	FM1	FM2	Fm	Durée
police	15	26	20	0	25
spacial	3	1	60	0	10
laser	20	9	50	0	2

Mini synthétiseur

Sous-programme LM pour LH-5801

Auteur Elie Zagury

Copyright l'Ordinateur de poche et l'auteur

DESASS. DE
7150-719A

```

7150 68 LDH, 00
▶ 7151 00
7152 6A LDL, 00
▶ 7153 00
7154 48 LDB, 01
7155 01
7156 4A LDC, 00
▶ 7157 00
7158 BE CALL 716F
7159 71
715A 6F
715B 62 DECL
715C 6E CPL, 00
▶ 715D 00
715E 99 JRNZ- 08
715F 08
7160 BE CALL 716F
7161 71
7162 6F
7163 60 INCL
7164 6E CPL, 00
▶ 7165 00
7166 99 JRNZ- 08
7167 08
7168 FD DECH
7169 62
716A 6C CPH, 00
716B 00
716C 99 JRNZ- 1C
716D 1C
716E 9A RET
716F FD PUSHHL
7170 A8
7171 FD PUSHBC
7172 88
7173 BE CALL E66F
7174 E6
7175 6F
7176 FD POPBC
7177 0A
7178 FD POPHL
7179 2A
717A 9A RET
717B B5 LDA, 00
▶ 717C 00
717D BE CALL 718E
717E 71
717F 8E
7180 DF DECA
7181 B7 CPA, 00
    
```

Elie Zagury

Langage-machine

et bruits divers

(PC-1500)

```

▶ 7182 00
7183 99 JRNZ- 08
7184 08
7185 BE CALL 718E
7186 71
7187 8E
7188 DD INCA
7189 B7 CPA, 00
▶ 718A 00
718B 99 JRNZ- 08
718C 08
718D 9A RET
718E AE LD(), A 7153
718F 71
7190 53
7191 AE LD(), A 7165
7192 71
7193 65
7194 BE CALL 7150
7195 71
7196 50
7197 A5 LDA, C 7153
7198 71
7199 53
719A 9A RET

```

Les codes du programme

```

: POKE &7150, &68
: 0, &6A, 0, &4B, 1
: &4A, 0, &BE, &71
: &6F, &62, &6E, 0
: &99, 0, &BE, &71
: &6F, &60
: POKE &7164, &6E
: 0, &99, 0, &FD, &
62, &6C, 0, &99, &
1C, &9A, &FD, &A8
: &FD, &88, &BE, &
E6, &6F, &FD
: POKE &7177, &A,
&FD, &2A, &9A, &B
5, 0, &BE, &71, &B
E, &DF, &B7, 0, &9
9, 0, &BE, &71, &B
E, &DD, &B7
: POKE &718A, 0, &
99, 0, &9A, &AE, &
71, &53, &AE, &71
: &65, &BE, &71, &
50, &A5, &71, &53
: &9A

```

Mini synthétiseur
Programme pour PC-1500
Auteur Elie Zagury
Copyright l'Ordinateur de poche
et l'auteur

```

1: REM SYNTH+(LM
) &7150-9A
2: "A" INPUT "REPE
TE ?"; N: POKE &
7151, N
3: INPUT "FREQ. 1
?"; A, "FREQ. 2 ?
"; B, "FREQ. 3 ?"
: C
4: POKE &717C, A:
POKE &7182, B:
POKE &718A, C
5: INPUT "DUREE ?
"; D, "FREQ. min
?"; E
6: POKE &7157, D:
POKE &715D, E
7: CALL &717B:
COTO "A
10: "B" INPUT "REPE
TE ?"; N: POKE &
7151, N
11: INPUT "Fmax1 ?
"; A, "Fmax2 ?";
B, "Fmin ?"; C, "
DUREE ?"; D
12: POKE &7153, A:
POKE &7165, B:
POKE &715D, C:
POKE &7157, D
13: CALL &7150:
COTO "B

```

Petit exemple d'exécution : Le chant du plaf

```

500: "C" POKE &715D,
0
510: POKE &7151, 1
520: POKE &7153, RND
10
530: POKE &7165, RND
20
540: POKE &7157, RND
30+10
550: CALL &7150: IF
RND 3=1 GOTO 55
0
560: IF RND 15<10
GOTO 560
570: COTO 520

```



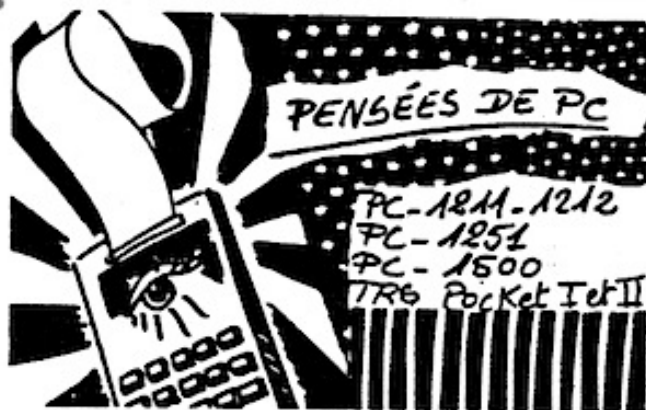
P\$ à T\$; le programme « tourne » donc quelle que soit votre version du PC-1500 (modules). Mais prenez bien garde à ne pas effacer la routine avec CLEAR, ou encore en réaffectant des contenus aux variables concernées.

Le programme est listé désassemblé : adresse hexadécimale, code hexadécimal, mnémonique. Plusieurs options sont possibles pour programmer. Si vous ne disposez que d'un PC-1500 seul, programmez les 4 lignes qui « POKE » les codes dans la mémoire. Cette séquence, exécutée avant « A » ou « B », conduira au stockage des codes aux adresses voulues.

Si vous utilisez, en revanche, un magnétophone, vous pouvez introduire les codes aux adresses précises (soit manuellement, soit à l'aide des utilitaires ENT et VER de l'Op n° 13) puis les sauvegarder de la manière suivante sur cassette : CSAVE M « SYNTH(LM) » ; & 7150, & 719A. Il suffira, à chaque fois, de relire la cassette par CLOADM pour qu'automatiquement les codes de la routine soient rangés de & 7150 à & 719A.

Enfin, si vous désirez stopper l'exécution d'un programme LM (touche BREAK inopérante) pressez All Reset au dos du PC-1500 durant 5 secondes et refusez le NEW 0 : CHECK en pressant sur la touche CL. Les programmes demeurent alors présents en mémoire.

□ JCK



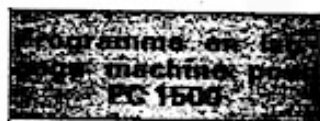
Dump sur PC 1500

Voici un programme en langage machine pour PC 1500, qui permet de se promener à bonne vitesse en mémoire en utilisant les touches ↑ et ↓.

Le langage machine offre une rapidité que le Basic ne permet pas d'atteindre.

Le programme principal est logé de & 414E à & 4182 ; le sous-programme, de & 4101 à & 4118, décode l'octet en BC et donne le résultat en DE.

Lucien Dorfman



Faire NEW&4200
Puis entrer aux

ADRESSES les
OCTETS-----

4101
85F1BE410905FB54
4109
890FB70A8704BB30
4111
1E9AB109BB401E9A

414E
680BA540D008A540
4156
D10A5870845A60F1
415E
8E41098454BE4109
4166
04F154BE41090454
415E
BE41098454BE4109
4166
04F154BE41090454
416E
BE41095A50BE4101
4176

```
FD6289019A4454BE
417E
4101BA4176
```

Puis le Progr.

```
1000:"Z"WAIT 0:
INPUT "Adres
se ler Octet
->";A
1010:B=INT (A/256
):POKE &40D0
,B;A-B*256
1020:F$=" ":
CALL &414E:
PRINT F$;E$
1030:A$=INKEY$
1040:IF ASC A$=10
LET A=A+8:
GOTO 1010
1050:IF ASC A$=11
LET A=A-8:
GOTO 1010
1060:GOTO 1030
```

Le PC 1500 n'est pas amnésique !

Il est toujours possible de retrouver un programme effacé par un NEW accidentel, et cela grâce à un petit truc fonctionnant sur PC 2 et PC 1500.

Il existe sur ces matériels un octet numéroté 255 qui sert de pointeur de fin de programme.

Quand on fait un NEW, le pointeur revient à &38C5 ou &40C5 (selon que l'on dispose du module 8 Ko ou non).

Pour retrouver la liste du programme perdu, il faut mettre l'octet 255 à une adresse lointaine, en fin de mémoire. Pour cela, on fera donc :

POKE & 7000, 255.

Mais il faudra également effacer le contenu (octet 255) des adresses &38C5 ou &40C5. On entrera donc :
POKE &38C5,0

Cela étant fait, il faut sacrifier une ligne du programme pour obtenir sa restitution. On peut choisir la première (car elle porte presque toujours le n° 10) ou la dernière (si l'on se souvient de son numéro), ou enfin une ligne ne contenant qu'une remarque. On tape donc :

< N° de ligne > < Enter >

Et voilà, c'est terminé : le programme est restitué. On comprend alors l'utilité d'un REM ou d'un END que l'on peut effacer sans perdre une ligne importante du programme.

Maurice Choucroun

Routines principales pour PC 1500

Pour pouvoir faire des programmes en langage machine réellement utiles, il faut pouvoir utiliser les entrées-sorties telles que écran, clavier, buzzer, etc. Plutôt qu'avoir à réécrire ces routines, il vaut mieux utiliser celles déjà contenues dans la MEM, dont voici les principales.

1) Pour l'écran :
CALL EE71 (ou SBR F2)
→ CLS
CALL ED5B (ou SBR 8A)
→ PRINT du caractère dont le code ASCII est dans A.
CALL EDFG (ou SBR 88)
→ G PRINT de A.

Pour les deux dernières routines, BC doit contenir le pointeur d'écran où l'on désire l'affichage ; 8 indique le quart de l'écran (74 pour le premier, 75 pour le second, 76 pour le troisième et 77 pour le quatrième), et C la position dans ce quart (de 00 à 4C).

Après un SBR 8A, BC sera remis à jour en étant augmenté de 6, sauf pour les cas spéciaux de changement de quart. De même pour SBR, où il est augmenté d'une unité.

2) Pour le clavier :
CALL E42C → INKEY\$

Si après CALL, le drapeau de retenue (« carry ») est à 1, cela signifie qu'aucune touche n'était enfoncée. S'il n'y a pas de retenue, alors une touche était enfoncée, et son code ASCII est dans l'accumulateur A.

CALL E451 (ou SBR A6)

→ test de la touche BREAK.
Après le CALL, la retenue Z sera à 1, si elle était enfoncée, et à 0 sinon.

3) Pour le buzzer :
CALL E66F → 1, L, BC.
Donc, si L contient le ton, et BC la durée, alors CALL E66F jouera cette note.

CALL E669 → BEEP1. Il s'agit ici d'un bip simple qui correspond à un CALL E66F où les registres auraient été initialisés ainsi : L = 8 et BC = 01A0.

Remarque : attention, ces sept routines modifient le contenu de certains registres. En voici la liste :

```
EE71 A,H,L
ED5B A,B,C,H,L
EDF6 A,B,C,H
E42C A,B,C,L
E451 aucun
E66F A
E669 A,B,C,L
```

Et maintenant, bonne programmation !

Christian Boyer

Exemple d'utilisation des routines

Ce sous-programme entièrement relogeable vous affichera le message pointé par HL, émettra un bip, et attendra un appui sur ENTER pour continuer.

```
FD 88 MSG: PUSH BC
FD A8 PUSH HL
F2 SBR F2
FD 2A POP HL
48 74 LD B, 74H
4A 00 LD C, 0
65 BCLE: LDH A, (HL)
87 0D CPA, ODH
88 08 JR Z, +BIP
FD A8 PUSH HL
CD 8A SBR 8A
FD 2A POP HL
9E 0D JR -BCLE
BE E6 69 BIP: CALL E669H
BE E4 2C GET: CALL E42CH
93 05 JR C, -GET
87 0D CPA, ODH
99 09 JR NZ, -GET
FD 0A POP BC
9A RET
```

```
4200 68 42 LD H,42H
6A 08 LD L,08H
BE 42 10 CALL MSG
9A RET
4208 42 4F 4E DEFM
«BONJOUR»
4A 4F 55
52
OD DEFB 0DH
```

Si vous logez ce message en 4210H, alors un CALL 4200 du programme suivant affichera BONJOUR.

Christian Boyer

-32-

Der PC-1500 spricht Basic.

Japanisch müfte
man können.



- CE-158 -

TaCoSo, Berlin