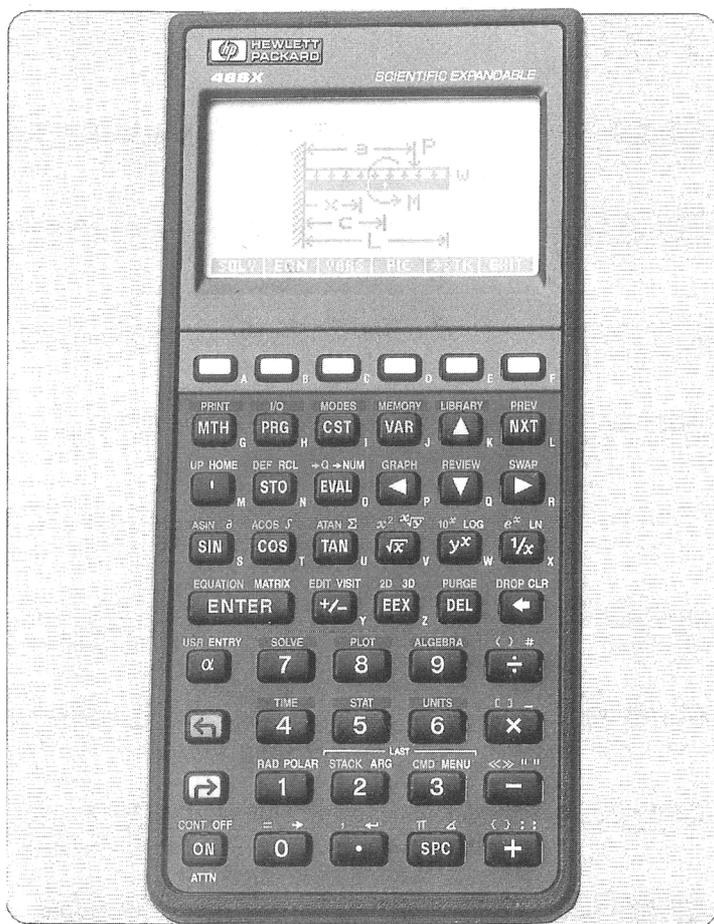


Tips und Tricks

Programmhandbuch für den

HP-48SX

Pocket Computer



ISBN 3-89374-076-7

Autorenteam

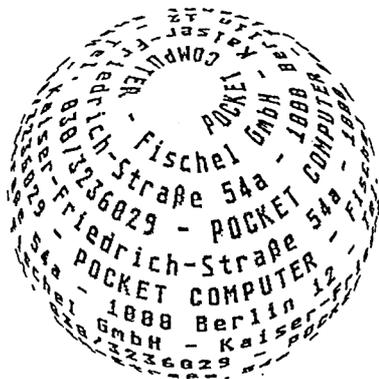
Fischel GmbH

POCKET COMPUTER

FISCHEL GmbH
Zeitschrift für Taschencomputer

durch Information vorn

Kaiser-Friedrich-Straße 54a
1000 Berlin 12
Telefon (030)3236029
HRB 19396 Amtsgericht Charlottenburg



=====
C FISCHEL GMBH

Alle Rechte vorbehalten. Ohne ausdrückliche Genehmigung des Herausgebers ist es nicht gestattet, das Buch oder teile daraus auf fotomechanischem (Fotokopie, Mikrokopie) oder sonstigem Wege zu vervielfältigen. Es kann keine Haftung für die Richtigkeit der Programme übernommen werden, obwohl sie ausgetestet wurden.

=====
POCKET COMPUTER

Fischel GmbH
Kaiser-Friedrich-Straße 54a
1000 Berlin 12 - Tel. 030/3236029

Fax: 030/324 09 28

Bankverbindung: Postgiroamt Berlin-West, Bankleitzahl 10010010, Kontonummer 461533-103

Öffnungszeiten: Montag-Freitag 10.00-16.00 Uhr, Samstag 10.00-14.00

- Inhaltsverzeichnis -

	Seite:
◦ User - Tips	3
◦ Mathematikprogrammiersammlung:	
- Grafische Darstellung von Linien "Locate"	5
- Binominalkoeffizienten	6
- Berechnung des Gaußschen Algorithmus'	7
- Berechnen von Eigenwerten einer 3*3 Matrix	10
- Aufstellen einer Funktionsgleichung	12
- Lösungen von Gleichungen nach der Newton-Methode	13
- Chain für Matrizen	14
- Automatische Nullstellensuche	15
- Kurvenanpassung	16
- Nullstellensuche	17
- n-te Wurzel einer komplexen Zahl	18
- Komplexe Wurzel auf dem HP 48SX	19
- Stern-Dreieck-Umwandlung	20
- Berechnen von Flächenträgheitsmomenten	23
◦ Spiele:	
- Würfel-Simulator	28
- Senso	29
- Wurmy	33
- Catch	38
- Morse-Code-Generator (kein Spiel)	40
- Geduldsspiel	41
- Mastermind	43
- Hangman	45
◦ Sonstiges:	
- Romnum (römische Zahlenumwandlung)	47
- Back	48
- Sumn	49
- Rename	50
- Backup-Übertragung des Speicherinhalts	51
- Telefongebührenzähler	52

• The HP 48 SX-Demo Book	55
• HP 48 SX-File manager	77
• HP 48SX - Programmer's Toolkit	79
• Casino 48	81
• The HP - 48 Handbook	82
• HP - 48 Graphics	83
• Using the HP - 48 SX	86
• HP - 48SX Programmierhandbuch	90
• Höhere Mathematik	93
• Super-Bestellschein	96
• Abonnement	98
•	
• Zubehör	99

Olaf Rauch
Mainzer Str. 26
5600 Wuppertal 1

Für HP-48-User ein paar Tips, die nicht im Handbuch beschrieben sind

Mit den Befehlen ASN und STONEYS lassen sich auch die Umschalttasten undefinieren, z.B. belegt 71.1 ASN die gelbe Taste, 71.3 die blau umgeschaltete gelbe Taste, letztere Funktion wird also (im User-Modus) durch die Tastenfolge blau-gelb aktiviert, andere Kombinationen sind auch möglich, z.B. alpha-gelb-gelb (71.5) oder alpha-gelb-blau (81.5)

Entgegen den Angaben im Handbuch kann der Dezimalpunkt (wenn das Komma eingestellt ist, das Komma) in Namen verwendet werden, ich verwende beispielsweise in einem Adressregister die Menüasten 'A...F' etc., erzeugt mit dem STO-Befehl und 'A...F' als Variablenname.

Ebenfalls abweichend von den HP-Angaben funktionieren die Flags -23 bis -26, die überlauf, Unterlauf, etc. anzeigen, nur dann, wenn man die entsprechende Fehlermeldung mit Flag -20 bis -22 abgeschaltet hat. Die Fehleranzeige bleibt gesetzt, bis man sie mit CF löscht.

Gelegentlich macht der Rechner eine Denkpause von ca. einer Sekunde, um den Speicher neu zu organisieren (von HP "garbage collection" genannt). Wer Programme mit Grafik, insbesondere Spiele, programmiert, weiß, wie lästig diese Unterbrechungen sein können. Der Befehl MEM erzwingt diesen Vorgang. Man kann also durch regelmäßiges Einstreuen der Befehle MEM DROP in ein Programm die Pausen regelmäßig gestalten, z.B. vor jedem neuen Bildaufbau oder jeder Bewegung eines Punktes auf dem Display. Je kürzer man die Abstände hält, umso kürzer fällt auch das garbage collection aus, so daß die Unterbrechungen bis zur Unmerklichkeit reduziert werden können.

Die Taste DRAW im PLOT-Menu zeichnet die Funktion mit Achsenkreuz. Möchte man das Achsenkreuz nicht im Bild haben, drückt man (ERASE -) ENTRY - DRAW - ENTER. Damit wird der Programmbefehl DRAW ausgeführt, der bekanntlich die Funktion ohne Achsen zeichnet.

Die Taste `EXPR=` aus dem `SOLVR`-Menü kann nicht nur, wie im Handbuch beschrieben, zur Kontrolle einer Solver-Lösung verwendet werden. Sie ist auch nützlich beim Auswerten einer beliebigen Formel, die man in `EQ` gespeichert hat. Man kann im `SOLVR`-Menü wie gewohnt die Variablen definieren (wenn man den Solver nicht verwendet, muß man natürlich alle Variablen definieren), mit `REVIEW` auflisten lassen und mit `EXPR=` das Ergebnis berechnen.

Im Unterschied zum Solver verarbeitet `EXPR=` nicht nur reelle Zahlen und Einheitenobjekte, sondern alle Datentypen.

Ein Beispiel: Winkel zwischen zwei Vektoren

```
<< A B DOT A ABS B ABS * / ACOS » ENTER
```

```
Im Menü SOLVE: STEQ SOLVR
```

```
Im Menü SOLVR: [ 7 4 1 J A
```

```
[ 3 -2 3 J B EXPR=
```

```
Ausgabe:
```

```
EXPR: 65.1722873232
```

Was das Handbuch sonst noch verschweigt:

`NEG (+/-)` invertiert ein grafisches Objekt, auch `PICT` als Argument funktioniert.

Wenn man trotz "Low Bat"-Warnung wittertippt, kann der Rechner plötzlich und unerwartet abstürzen und ist dann möglicherweise auch durch `Reset` mit `ON+C` nicht wieder in Gang zu bringen. Für diesen Fall befindet sich unter einem der Gummifüße ein mit "gr" beschriftetes Loch, was den `Hardware-Reset` darstellt. Dabei gehen nur der `Stack` und andere Kleinigkeiten verloren (wie beim `Software-Reset`). Bei mir hat der Computer übrigens mehrstündigen Batterieentzug nach Absturz ohne Datenverlust überstanden.

Program:LOCATE

This program locates string at certain point of HP's LCD. You must put a definite string in the level 2 of the stack and in level 1 a complex number containing x and y coordinates of place where you want your string to be displayed. Then you start the program LOCATE.

Program:

```
<< V-> SWAP 'x' STO ->
y
  << CLLCD ""
    WHILE x 1 >
      REPEAT 'x' 1
    STO- " " +
      END SWAP + 'x'
  PURGE y DISP 3
  FREEZE
  >>
>>
```

[ENTER] 'LOCATE [STO]

Checksum: #6449d
Bytes: 131

Example: Display string "HP 48SX" at x coordinate 5 and y coordinate 4:

```
2: "HP 48SX"
1: (5,4)
[VAR] LOCATE
```

Luka Stravs
Lubejeva 1
61000 Ljubljana
Yugoslavia

48SX HEWLETT PACKARD 48SX

BINOMINALKOEFFIZIENTEN

Dieses Programm erwartet aus der Ebene 1 den Exponenten eines zweigliedrigen Ausdrucks von der Form $(a+b)^n$. Nach Aufruf des Programms wird eine Liste mit den Koeffizienten sämtlicher Glieder ausgegeben.

Gruss an alle HP 48SX User

Ruedi Kuhn, Amriswilerstr.64
CH-8570 Weinfelden

Programm: BIKOE

```

< 'N' STO CLEAR N 'N'
PURGE → N
  <
    IF 'N≠0'
    THEN 0 → C
      < 1 1 N
        START 'C' INCR
        DROP 'N!/(C!*(N-C)!)'
        EVAL 1 RND
          NEXT 1
            > DROP DEPTH →LIST
        ELSE 1
        END
      >
    >
  'BIKOE STO
Bytes ≠ 47309d
      193.5
    
```

Markus Sprunck
Caracciolastr.56

Sehr geehrte Damen und Herren,

8000 München 45

anlässlich ihres Aufrufes in der Zeitschrift "Pocket + Laptop Computer" Nr.: 01/91, Programme für HP Rechner zur Veröffentlichung einzusenden, sende ich Ihnen ein Programm zur Berechnung des Gaußschen Algorithmuses ein.

Wie Ihnen sicherlich bekannt ist, kann der HP 48SX eindeutig bestimmbare lineare Gleichungssysteme lösen. Da diese in der Praxis seltener auftreten, als nicht eindeutig bestimmbare lineare Gleichungssysteme, müssen Lösungsverfahren wie z.B. das Eliminationsverfahren oder eben den Gaußschen Algorithmus verwendet werden.

ZUM PROGRAMMAUFBAU

I. Gegeben ist ein Gleichungssystem folgender Art:

$$a_{11} \times X_1 + a_{12} \times X_2 + \dots + a_{1n} \times X_n = a_1$$

$$a_{21} \times X_1 + a_{22} \times X_2 + \dots + a_{2n} \times X_n = a_2$$

$$\cdot \quad \cdot \quad \quad \quad \cdot \quad \cdot$$

$$\cdot \quad \cdot \quad \quad \quad \cdot \quad \cdot$$

$$a_{m1} \times X_1 + a_{m2} \times X_2 + \dots + a_{mn} \times X_n = a_m$$

Hieraus kann eine Matrix mit m-Reihen und n+1 Spalten gebildet werden.

$$((a_{11} \ a_{12} \ \dots \ a_{1n} \ a_1)$$

$$(a_{21} \ a_{22} \ \dots \ a_{2n} \ a_2)$$

$$\cdot \quad \cdot \quad \quad \quad \cdot \quad \cdot$$

$$\cdot \quad \cdot \quad \quad \quad \cdot \quad \cdot$$

$$\cdot \quad \cdot \quad \quad \quad \cdot \quad \cdot$$

$$(a_{m1} \ a_{m2} \ \dots \ a_{mn} \ a_m))$$

Diese Matrix wird nun unter der Variabelen 'MA.1' abgespeichert.

II. Durch Anwendung des Gaußschen Algorithmuses soll nun das gestaffelte Schema

$$a_{11} \quad a_{12} \quad \dots \quad a_{1n} \quad | \quad a_1$$

$$c_{21} \quad | \quad b_{22} \quad \dots \quad b_{2n} \quad | \quad b_2$$

$$c_{31} \quad c_{32} \quad | \quad \dots \quad b_{3n} \quad | \quad b_3$$

entstehen, mit den Bedingungen:

$$c_{mn} = - \frac{1}{b_{nn}} \left(a_{mn} + \text{skalares Produkt } m\text{-te Reihe und } n\text{-te Spalte} \right)$$

$$b_{mn} = a_{mn} + \text{skalares Produkt } m\text{-te Reihe und } n\text{-te Spalte}$$

wird $b_{nn} = 0$ oder $a_{11} = 0$ so wird die n-te Reihe übersprungen.

Tritt dieser Fall auf, so wird die Matrix 'MA.1' durch Zeilentausch entsprechend verändert. Kann durch Zeilentausch die Bedingung $b_{nn} \neq 0$ nicht erfüllt werden, so wird das Programm vorzeitig abgebrochen und die Meldung "Teste Matrix, gleiche Reihen oder Spalten?" 20 sek. angezeigt.

z.B.

1	2	10	4
2	4	6	3
2	4	5	1

Dieses Gleichungssystem ist ebenfalls zu lösen, wenn man eine der gleichen Spalten entfernt und das Programm neu startet.

(Analog bei gleichen Reihen)

Das Auftauchen gleicher Reihen oder Spalten muß aber nicht zum Abbruch führen.

III. Die Zeilen- und Spaltennummern der gerade berechneten Komponenten werden angezeigt.

Auch der Zeilentausch wird mit den Nummern der vertauschten Zeilen angezeigt.

IV. Es ist nötig die Variable 'RN' vor Beginn des Programmes auf Null zu setzen. Dies geschieht durch das Starten mit 'RUN.'.

Dieses Programm ist für den HP 48SX geschrieben. In der Anlage finden Sie bitte das Listing.

Ich habe das Programm an allen, mir bekannten Sonderfällen getestet und es sind nie Probleme aufgetaucht. Trotzdem bitte ich Sie das Programm nochmals auf etwaige Mängel zu testen, da ich nicht für Fehler und mittelbare oder unmittelbare Schäden im Zusammenhang mit den Programm hafte.

Berechnen von Eigenwerten einer 3*3 Matrix

Ein Programm für den HP 48SX.

Michael Hülsmann
 Henrichenburgerstr. 179
 4350 Recklinghausen

Das Programm erwartet in der Variablen B eine 3*3 Matrix die mit dem Matrix Writer oder direkt eingegeben wurde.
 Zur Lösung der charakteristischen Gleichung wurde das Verfahren aus dem Taschenbuch der Mathematik von Bronstein und Semendjajew (S. 132 3. Methode) angewendet.

Beispiel:
$$B = \begin{pmatrix} 35 & 88 & 116 \\ -2 & -5 & -6 \\ -8 & -20 & -27 \end{pmatrix}$$

Die Lösung wird wie folgt in dem Display des HP angezeigt:

```
3
-1 * y
-3 * y ^ 2
y1= -1
y2= 3
y3= 1
```

Ich wünsche allen Anwendern viel Freude an diesem Programm.

```
<<
'B' 1 GET 'B' 8 GET 'B' 6 GET * *
'B' 2 GET 'B' 4 GET 'B' 9 GET * * †
'B' 3 GET 'B' 7 GET 'B' 5 GET * * +
'B' 1 GET 'B' 5 GET 'B' 9 GET * * --
'B' 2 GET 'B' 7 GET 'B' 6 GET * * -
'B' 3 GET 'B' 4 GET 'B' 8 GET * *
- DUP 'X' STO ->STR CLLCD 1 DISP
'B' 1 GET 'B' 5 GET *
'B' 2 GET 'B' 4 GET * -
'B' 1 GET 'B' 9 GET * +
'B' 3 GET 'B' 7 GET * -
'B' 6 GET 'B' 8 GET * -
'B' 5 GET 'B' 9 GET * +
DUP 'Y' STO ->STR " * y" + 2 DISP

'B' 5 GET NEG 'B' 9 GET NEG 'B' 1 GET NEG +
+ DUP 'Z' STO ->STR " * y ^ 2" + 3 DISP
"y ^ 3" 4 DISP

RAD '(3*Y-SQ(Z))/3' EVAL 'p' STO
'2*Z^3/27-Z*Y/3+X' EVAL 'q' STO
'(p/3)^3+(q/2)^2' EVAL 'v' STO
```

```

IF 'v>0 OR p>0'
THEN DUP2 "KOMPLEX" 5 DISP 5 FREEZE HALT
ELSE
  IF 'v==0 AND p==0'
    THEN NEG 'Z' RCL 3 / ->STR "y123 = " SWAP +
    DISP 5 FREEZE
    ELSE
      IF 'q==0'
        THEN '√ABS(p/3)' EVAL 'U' STO
        ELSE 'SIGN(q)*√ABS(p/3)' EVAL
          'U' STO

        END 'ACOS(q/(2*U^3))' EVAL 'g'STO
        '-2*U*cos(g/3)-Z/3' EVAL
        DUP 5 RND
        ->STR "y1 = " SWAP + 5 DISP
        '-2*U*cos(g/3+2* pi/3)-Z/3'
        ->NUM DUP 5 RND
        "y2 = "SWAP + 6 DISP
        '-2*U*cos(g/3+4* pi/3)-
        Z/3' ->NUM DUP 5 RND
        "y3 = " SWAP + 7 DISP 7
        FREEZE

      END

    END

  END

  {g p q U v X Y Z} PURGE
  CLEAR
  HALT
  >>

```



Hewlett Packard HP48sx.

Aufstellen der Funktionsgleichung,
Geschrieben von Stefan Kubsch, Flensburg.

Aufgabe: Welche Gleichung hat die Funktion, die durch die Punkte P1(4/-7), P2(8/13) verläuft.

Ansatz: $y = a1 * x + a0$

Gesucht sind a1 und a0.

2 Unbekannte --> 2 Gleichungen.

I -7 = 4*a1 + a0

II 13 = 8*a1 + a0

Beide Punkte sind Punkte der gesuchten Funktion. Ihre Koordinaten müssen die Funktionsgleichung erfüllen.

Wenn man das obige Gleichungssystem auflöst, dann erhält man das folgende Ergebnis:

a1=5 a0=-27 --> $y = 5x - 27$

Bei zwei gegebenen Punkten ist es noch sehr einfach die gesuchten Lösungen zu ermitteln. Wenn aber nun z.B. 3 oder 4 oder mehr Punkte gegeben sind, so erhöht sich der Rechenaufwand doch erheblich. Hier setzt mein Programm ein. Es ermittelt aus beliebigen vielen eingegebenen Punkten, die Funktionsgleichungen sowie die dazu gehörigen Lösungen.

Eingegeben werden die Punkte als Komplexe Zahlen. Also (4,-7) ENTER (8,13) ENTER.

Nun startet man das Programm.

SOLVE löst das Gleichungssystem.

MATR zeigt die Matrix an.

VEKT zeigt den Vektor an.

EXIT beendet das Programm und löscht die vom Programm erzeugten Variablen.

Es ist darauf zu achten, daß bei der Eingabe der Komplexen Zahlen, der Stack Anfrags leer ist.

Da es sonst einen Fehler geben kann

der dann falsche Ergebnisse liefert.

Stefan Kubsch
Sauermannstr.10
2390 Flensburg

Listing:

```

<<
DEPTH 'd' STO d 1 →LIST 1 CON d d 2
→LIST 1 CON 'MA' STO 'VE' STO d DUPN
d 1 FOR i C→R SWAP DROP VE SWAP i
SWAP PUT 'VE' STO
-1 STEP 0 'N' STO d SQ 'D' STO DO
'C' STO C C→R DROP 'X' STO 1 d FOR j
MA X N ^ D SWAP PUT 'MA' STO 'N' INCR
'D' DECR NEXT d 2 * DROPN 0 'N' STO
UNTIL D 0 == END
( ( "SOLVE" ( « VE MA / VE MA 3 PICK
RSD MA / + OBJ→ DROP » ) ) ( )
( "MATR=" ( « MA » ) ) ( "VEKT=" (
« VE » ) ) ( ) ( "EXIT" ( « CONT>> ) ) )
TMENU "Select your weapon,$$ Exit
leave Programm." PROMPT
( X C D N VE MA d ) PURGE 1 MENU
>>
    
```

Achtung wichtiger Hinweis:

Statt der runden Klammern im Listing, sind die geschweiften Klammern zu nehmen. Leider kann mein Drucker diese nicht darstellen.

\$\$ bedeutet einen Zeilenumbruch.

(Shift rechts und Punkt.)

HP 48SX CHAIN FÜR MATRIZEN **HP 48SX**

Welcher HP 48SX-Statistiker kennt dieses Problem nicht ?
 Da hat man einen allmächtigen Taschenrechner, aber will man mal zwei (reelle) Matrizen für 2-Variable Statistik zusammenfügen, weil man vergessen hat diese gleich in eine Matrix zu schreiben, sucht man vergeblich nach einer solchen Funktion. Aber hier ist die Lösung des Problems :

Dieses Programm erwartet zwei Matrizen aus den Ebenen 1 und 2. Nach Aufruf des Programmes wird dann eine Matrix ausgegeben, in der die Matrix in Ebene 1 an die in Ebene 2 angehängt wurde. Dabei können die Zeilenzahlen verschieden sein, dementsprechend werden an einer Matrix Nullen hinzugefügt. Außerdem wird vorher kontrolliert ob die beiden Argumente wirklich Matrizen sind. Wer diese Kontrolle nicht benötigt kann ja die Zeilen 2-8 bis auf das "<" löschen.

Gruß an alle HP 48SX User

HACKED BY HACKER

Kristian Hacker
 Berliner Str. 53
 1000 Berlin 27

```

Programm : CHAIN
« - MT1 MT2
« MT1 TYPE
  IF 3 ≠ THEN
    "1.Arg keine Matrix" DOERR
  END
  MT2 TYPE
  IF 3 ≠ THEN
    "2.Arg keine Matrix" DOERR
  END
  MT1 SIZE OBJ→ DROP
  MT2 SIZE OBJ→ DROP
  - M1z M1s M2z M2s
  « M1z M2z MAX
  M1s M2s +
  - Mz Ms
  « MT1 Mz M1s 2 -LIST
  RDM TRN OBJ→ DROP
  MT2 Mz M2s 2 -LIST
  RDM TRN OBJ→ DROP
  Ms Mz 2 -LIST
  -ARRAY TRN
  »
»
»
'CHAIN STO
Bytes #59077d
      343
    
```

HP -15 C

Automatische Nullstellensuche

Unterprogramm zur
Eliminierung der
Nullstellen:

Anleitung zur Bedie-
nung: Die Funktion
als Unterprogramm
unter Lbl,9 abspei-
chern. Zwei Start-
werte in X- und Y-
Register laden. Pro-
gramm mit Lbl A
starten. Nachdem
alle Nullstellen
gefunden wurden oder
nach der 6. gefunde-
nen Nullstelle
bricht das Programm
ab. Die Ergebnisse
stehen in Reg 1'6.
In Reg 0 steht die
Anzahl der gefunde-
nen Nullstellen, in
Reg 8 und 9 die
Startwerte.

F. Arnecke,
Hauptstr. 32,
3387 Vienenburg 4

<u>Anzeige:</u>	<u>Eingabe:</u>	<u>Anzeige:</u>	<u>Eingabe:</u>	<u>Anzeige:</u>	<u>Eingabe:</u>
42.21.11	<u>Lbl A</u>	31	Stop	42.21.,2	<u>Lbl,2</u>
42.34	Cl. Reg.	44 7	Sto 7	45 0	Rcl 0
44 9	Sto 9	42 31	Pause	44 25	Sto 1
34	X <> Y	43. 5. 1	CF 1	33	Stack
44 8	Sto 8	1	1	36	down
43. 4. 1	SF 1	44.40. 0	1 Sto+ 0	32, 9	Enter
0. 00001	0. 00001	45 0	Rcl 0	43. 6. 1	Gosub, 9
44 0	Sto 0	44 25	Sto 1	43 32	Return
42.21.,1	<u>Lbl,1</u>	45 7	Rcl 7	42.21.,3	<u>Lbl,3</u>
43. 4. 2	SF 2	u 44 24	Sto i	34	X <> Y
45 8	Rcl 8	45 25	Rcl i	u 45 24	Rcl i
45 9	Rcl 9	43 44	Int	30	minus
42.10.,2	Solve, 2	6	6	10	geteilt
43. 5. 2	CF 2	43.30. 5	Test X=Y	42. 5.25	DSE I
43. 6. 2	F2 ?	31	Stop	22, 3	Goto, 3
		22, 1	Goto, 1	43 32	Return

```

<< -> var
<< OBJ-> { }
'LISTEX' STO { }
'LISTEX' STO -> n
<< 1 n
FOR i OBJ->
'LISTEX' STO+ 0 n 1
FOR j DUP j
^ 'LISTEX' STO+
NEXT DROP
NEXT LISTEX
OBJ-> DROP { n 1 }
->ARRY LISTEX OBJ->
DROP { n n } ->ARRY
-> b a
<< b a / b a 3
PICK RSD a / + OBJ->
DROP n 1 - 0
PURGE
FOR i n
ROLL var i ^ * -1
STEP 1 n 1
>>
FOR i +
NEXT {
LISTEX LISTEX }

```

Das Programm berechnet aus n Punkten eine Polynomfunktion vom Grade n-1, deren Funktionsgraph diese Punkte durchläuft.

Damit ist es möglich, z.B. beliebige Meßwerte zu interpolieren, die mit den eingebauten Statistikmodellen nicht angenähert werden können.

Das Programm benutzt die Matrixfunktionen und führt eine einmalige Residuumkorrektur zur Erhöhung der Genauigkeit des Ergebnisses durch.

Als Eingabe werden die Punktkoordinaten in Listenform (Stackebene 2) und der gewünschte Variablenname (Ebene 1) verlangt, bevor das Programm gestartet wird.

Beispiel:

<i>Eingabe</i>	<i>Ausgabe</i>
2: { (5,-2) (4,3) (-1,3) }	1: '-(.833333333333333* NAME^2)+(2.5*NAME+ 6.333333333333333)'
1: 'NAME'	

F. Arnecke,
Hauptstr. 32,
3387 Vienenburg 4

Prüfsumme: # 45837d

Alexander Barousch
Johann-Strauß-Gasse 30
A-1040 Wien

Nullstellenberechnung (HP-48SX)

Ich möchte Ihnen noch ein zweites Programm vorstellen, das sich auf die Berechnung der Nullstellen einer Funktion nach dem Newtonschen Näherungsverfahren stützt. Es gibt zwar eine Funktion 'ROOT' im HP, aber falls halt einmal "der Newton" mit seinen Iterationsschritten gefragt sein sollte bietet dieses Programm eine komfortable Berechnung. Es muß die Funktion in 'EQ' gespeichert sein. Das Programm leitet die Funktion ab und speichert sie in 'dEQ'. Die einzige Eingabe ist der Anfangswert der Iteration. Das Programm bricht erst ab wenn zwei Werte identisch sind. Im Stack kann man dann die Iterationswerte von oben bis unten bewundern.

```
NEWTON
← CLEAR 'X' PURGE 'EQ'
RCL 'X' EVAL → EVAL
'EQ' STO "AUSGANGSWERT"
( 'V ) INPUT OBJ → DUP DUP
'X' STO 1.50
START 'EQ' → NUM 'dEQ'
→ NUM / - DUP 3 PICK
IF ==
THEN DROP KILL, STO
END DUP DUP 'X' STO
NEXT
➤
```

Alexander Barousch
Johann-Strauß-Gasse 30
A-1040 Wien

n-te Wurzel einer komplexen Zahl (HP-48SX)

Sg Fischel Verlag.

In Ihrem Buch 'Programmierhandbuch zum HP48SX' ist auf Seite 87 ein seltsames Programm 'SQRI' abgedruckt. Ich weis zwar nicht was die ominöse Funktion 'RP' bewirken soll, aber es ist auf jedenfall keine Funktion des HP48. Daher habe ich ein ähnliches Programm zur Berechnung der n-ten Wurzel einer komplexen Zahl geschrieben, das garantiert läuft und noch dazu komfortabler ist. Es bezieht sich vor allem auf die Aufgabenstellung z.B. "Man bestimme vierte Wurzeln von -1" oder "Man bestimme die Quadratwurzeln von j. und 3-4j". Für reelle Zahlen halte ich eine solche Berechnung für nicht sinnvoll. Außerdem wird die Berechnung in Radiant ausgeführt wie es für Ingenieure wichtig ist. Die Variablen a ist der Realteil, bj der Imaginärteil der komplexen Zahl z=a+bj. n ist die Anzahl der Wurzeln die gezogen werden sollen. Im Anhang gibt es noch eine hinreichende mathematische Erklärung der verwendeten Formeln.

```
SQRN
" GLLCD 4, FIX
" a, bj, n PROMPT → a b
n ← RAD a b R+C DUP ARG
→ P ← ABS n XROOT → r
" 0 n 1 - i (p+23x#
) / n! * n! DUP COS SWAP
SIN R+C * NEXT
→
→
→
```

Komplexe Wurzel auf dem HP 48 SX

A. Linz

Das berechnen komplexer Wurzeln sollte für den 48 SX eigentlich kein Problem darstellen. Jedoch hat das interne Programm den Nachteil, daß es nur ein Ergebnis anzeigt. Bei diesem Programm werden jedoch alle Ergebnisse gezeigt und auch in einem Koordinatensystem (Re- und Im-Achse) angezeigt. Es sind im ganzen 2 Eingaben zu machen:

- eine (Komplexe) Zahl der Form (X,Z) unter -Z- ablegen
 - der Grad der Wurzel unter -N- ablegen
- dann einfach mit -Enter- zum Bild und nochmals zu den Werten .

```

KWURZ
« CLLCD ERASE
"Komplexe Wurzel" 1.00
DISP 1.50 WAIT CLLCD { {
"Z" {
« {Z' STO
» } } { "N" {
« {N' STO
» } } } TMENU HALT Z N
INV X DUP ABS SWAP ARG →
B W
« π 2.00 * →NUM N / →
S
« 0.00 N 1.00 -
FOR K B W K S * -
DUP2 COS * ROT ROT SIN *
R→C
NEXT
» N DUPN DRAX 1.00 N
START C→PX DUP PIX?
IF 0.00 ==
THEN PIXON
ELSE PIXOFF
END
NEXT GRAPH { N Z
PPAR } PURGE
»

```

Andreas Linz
 Am Lindchen 3a
 5112 Baesweiler

Stern-Dreieck-Umwandlung für den HP 48 SX

von A. Linz

Das folgende Programm ist an sich nichts neues. Das besondere liegt vielmehr in der Aufmachung, jedoch sollten schon ca. 2 K Speicher vorhanden sein. Da das Programm an sich für Kenner kein Problem darstellen sollte, wird darauf nicht weiter eingegangen. Ist es erstmal eingetippt, entfällt eigentlich auch eine Einweisung für die Bedienung, da dies vom Programm selber übernommen wird. Wenn das drücken der -Enter- Taste zuviel istum im Programm weiterzukommen, kann dies gerne mit Schleifen (WAIT XX) umgehen. Beim programmieren des Bildes sollte sehr sorgfältig gearbeitet werden, um Fehler im Bild zu vermeiden. Alle drei Programme werden in einer Ebene abgelegt. Durch andrücken DRST (Dreieck-Stern) oder STDR (Stern-Dreieck) wird das Programm BILD selbstständig aufgerufen und entsprechen dem Programm verwendet. Zum Schluß bleibt nur noch, viel Spaß zu wünschen.

```

STOR % CLLCD
" Das Programm wandelt "
1 DISP
" eine Stern in eine " 3
DISP
" Dreieckschaltung um " 5
DISP
" Bitte <ENTER> druecken "
7 DISP

DO
UNTIL KEY
END CLLCD
----> 1" 1 DISP
" " Z1
3 DISP
Z2 " Z3 Z23" 5
DISP
DISP
Z " Z 2 " 3"
3 DISP LCD+ (< # 0d # 0d
} BILD GDR +LCD
DO
UNTIL KEY
END
" Eingabe von Z1 bis Z3"
{ Z1: Z1:
Z2: " ( 1 0
Z3: ) INPUT OBJ+ +NUM 'Z3'
} INPUT +NUM 'Z2' STO +NUM
Z1 STO Z1 Z2 * Z2 Z3 *
Z3 Z1 * + + Z2 STO Z2 Z2
/ Z13 STO Z Z1 / Z23
STO Z Z3 / Z12 STO
CLLCD RE(Z12)= Z12 RE
+ 1 DISP IM(Z12)= Z12
IM + 2 DISP RE(Z13)=
Z13 RE + 3 DISP
IM(Z13)= " Z13 IM + 4
DISP RE(Z23)= " Z23 RE
+ 5 DISP IM(Z23)= Z23
IM + 6 DISP 3 FREEZE { Z
Z1 Z2 Z3 Z12 Z23 Z13
PPAR } PURGE DROP
    
```


Programm zur Berechnung von Flächenträgheitsmomenten mit dem Taschenrechner HP 48SX

Kurzanleitung:

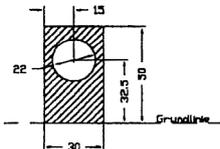
Die Programme 'PRVE'; 'VEIN'; 'UNT'; 'PAD'; 'JN'; 'KY' und 'DI' werden im Hauptverzeichnis abgespeichert. Es sind Unterprogramme die so auch von anderen Programmen aus aufgerufen werden können.

- PRVE : Gibt einen Vektor auf dem Thermodrucker aus
- VEIN : Dient zur Benutzereingabe von Vektoren
- UNT : Unterstreicht eine Zeichenkette für den Drucker
- PAD : Fügt vor eine Zeichenkette Leerzeichen ein, so daß rechtsbündiger Ausdruck ermöglicht wird.
- JN : Ja oder Nein Abfrage
- KY : Wartet auf Tastendruck
- DI : Zeigt sieben Zeichenketten im Display an

Die Programme 'IGES', 'INIT', 'DRUCK' und 'BED' stehen zusammen in einem Unterverzeichnis, z.B. 'MBAU', wie »Maschinenbau«. Vorher ist es ratsam, in diesem Verzeichnis Variablen 'IG', 'ABST', 'W' mit beliebigem Inhalt zu belegen, damit beim späteren Anwählen des Menüs sofort das Hauptprogramm 'IGES' aufgerufen werden kann.

Das Programm zeigt das errechnete Flächenträgheitsmoment, die Widerstandsmomente und den Abstand des Schwerpunktes von der Bezugslinie zwar auch an, um aber nach abgelaufenem Programm noch darauf zugreifen zu können, werden diese Daten zusätzlich in den vordefinierten Variablen gespeichert.

Erklärung der Funktionsweise anhand eines Beispielles:



Nach Aufruf des Programmes mit der Taste IGES wird zunächst nach der Anzahl der Flächen gefragt, die mit einander kombiniert sind. In diesem Fall 2 Flächen:

Anzahl der Flächen
des Körpers:
:n:2

und . Danach erfolgt eine Abfrage, ob der Körper nur aus Rechtecken und Kreisen besteht. In den meisten Fällen wird diese Frage mit JA zu beantworten sein. Wenn nicht, müssen die einzelnen Flächengrößen als Vektor eingegeben werden. (ohne Einheit, aber alle Maße in mm. Negative Flächen, also Löcher bekommen ein negatives Vorzeichen!)

Aber zurück zu unserem Beispiel:

Nur Rechtecke
und Kreise ?

[+] für Ja
[-] für Nein

. Danach wird die Breite und Höhe, bzw. der Durchmesser angegeben. (Zahl eingeben, Taste im Menüfeld drücken).

Breite und Höhe #1
oder Durchmesser [mm]
3:
2:
1:
30

Menütaste 'B', 50 Menütaste 'H', Menütaste 'CONT',
-22 Menütaste 'D', Menütaste 'CONT'.
Der Durchmesser muß negativ angegeben werden, um mit dem Wert das Loch zu erfassen. Wenn rechteckige Löcher auftreten muß und darf nur die Breite negativ eingegeben werden.

Abst. d. Einzelschwunkte
v. Bezugslin/Gesschwpkt

Taste Drücken

Nachfolgend sollen die Abstände der Einzelschwerpunkte der Flächen von der Bezugslinie oder vom Gesamtschwerpunkt angegeben werden, wenn dieser bekannt ist. . In unserem Fall ist er nicht bekannt. Deswegen wird die Grundlinie als Bezugslinie gewählt.

Als Vektor der Dim. 2 eingeben:
 :Abst. [mm]: [25 32,5]

ENTER. ACHTUNG: Wenn ein Flächenschwerpunkt unterhalb der Bezugslinie liegt, muß der Wert negativ eingegeben werden.

Ist der Gesamtschwerpunkt bekannt?

[-]. Danach gibt das Programm den Wert für den Abstand von der Bezugslinie an: ABST [mm]: 22,454.
 Diesen Wert muß man sich nicht merken oder notieren. Man kann darauf im Speicher zurückgreifen.

ENTER.

Jetzt soll man die Randfaserabstände in mm eingeben.

Randfaserabstände [mm]
 E1 E2 [] ABST [] CONT

50 ⇒ (Pfeil rechts, blaue Taste) **ABST** **-** **E1**
 ⇒ **ABST** **E2** **CONT**.

Das Ergebnis steht fest:
 Das Gesamtflächenträgheitsmoment: IG=272360.362 mm⁴
 Die Widerstandsmomente: W(1)=9887.534 mm³
 W(2)=12129.613 mm³.

Danach wird noch gefragt, ob die Werte ausgedruckt werden sollen. Wenn man keinen Drucker besitzt, kann man den vorletzten Befehl im Programm 'IGES' entfernen: ... " " DI INIT ». Die Unterprogramme 'DRUK', 'BED', 'PAD', 'UNT', 'PRVE' brauchen dann auch nicht eingegeben zu werden.

Der Ausdruck erklärt sich selbst.

Vielleicht ist die Anwendung einigermaßen klar geworden. Wichtig ist die Eingabe des negativen Vorzeichens bei:
 - negativen Abständen von der Bezugslinie
 - Löchern in Körpern (also negativen Flächen)

```
DI
< 7 DISP 6 DISP 5 DISP 4
DISP 3 DISP 2 DISP 1
DISP
DO
  UNTIL KEY
  END DROP
>
```

```
KY
<
DO
  UNTIL KEY
  END DROP
>
```

```
JN
< CLLCD 2 DISP 1 DISP
  [ + ] für Ja"
  [ - ] für Nein" 5 DISP
4 DISP
DO
  UNTIL KEY
  END 95 SAME
>
```

```
PAD
< 'o' STD →STR
  WHILE DUP SIZE 0 <
  REPEAT " " SWAP +
  END 'o' PURGE
>
```

```
UNT
< 27 CHR 251 CHR + SWAP
+ 27 CHR + 250 CHR +
>
```

```
VEIN
< "Als Vektor der Dim. "
SWAP →STR + "
eingeben:"
+ SWAP " " DUP ROT +
+ SWAP + "[ ]" + DUP 1
→LIST SWAP SIZE + INPUT
OBJ→ DTAG
>
```

```
PRVE
< DUP SIZE OBJ→ DROP → v
n
  < 1 n
  FOR n n 48 + CHR 93
  CHR + 'v(n)' EVAL →STR
  20 PAD + PR1 DROP
  NEXT
  >
>
```

```
BED
<
"=====
UNT PR1 CR
"Solien auch die vor-
"gegebenen Größen ge-
"druckt werden?"
JN
  IF 1 ==
  THEN "Flächen:" UNT
  PR1 A PRVE CR "Abst."
  IF X 3 ==
  THEN
  "v. Schwerpunkt" +
  ELSE
  "d.Schwerpunkte" + PR1
  "von der Bezugslinie"
  END ":" + PR1 s PRVE
CR "Trägheitsmomente:"
UNT PR1 I PRVE CR
"-----
PR1 CR
  END
>
```

```
DRUK
< "Ausdruck ?" "" JN
  IF 1 ==
  THEN 3 FIX -51 SF
OLDPRT BED
"Für d.ges.Körper gilt:
PR1 "Fläche:" UNT AG
→STR 15 PAD + PR1 CR
  IF X 3 ≠
  THEN "Abst. d. "
  "Schwerpunktes" UNT +
  PR1 "v.d. Bezugslin:"
  "ABST" EVAL →STR 8 PAD +
  PR1 CR
  END
"Trägheitsmoment" UNT
PR1 " I =" IG →STR 18
PAD + PR1 CR
"Randfaserabstände:" UNT
PR1 " e(1) = " 'E1' EVAL
→STR 15 PAD + PR1
" e(2) = " 'E2' EVAL →STR
```

```

15 PAD + PR1 CR
"Widerstandsmomente:"
UNT PR1 " W(1)= " 'W(1)'
EVAL →STR 15 PAD + PR1
" W(2)= " 'W(2)' EVAL
→STR 15 PAD + PR1 CR
"=====
UNT PR1 1 6
START CR
NEXT
END STD -51 CF
>
INIT
< ( 1 A n E E2 E1 X A R S
s AG h d ED d PR1PAR )
PURGE CLEAR STD 2 MENU
>
IGES
< -49 -50 CF CF
[ 0 0 ]
'W' STO
"Anzahl der Flächen
des Körpers:"
( 'n: " ( 1 0 ) V )
INPUT OBJ → DTAG 'n' STO
"Nur Rechtecke"
"und Kreise ?" JN
IF 1 ==
THEN 1 n
START 0
NEXT n →ARRY DUP 'A'
STO 'I' STO 1 n
FOR 1 0 'd' STO
"Breite und Höhe # " I
→STR +
"oder Durchmesser [mm]"
+ ( 'b=1' ( 'b h oder d (
) CONT ) ) STEQ 30 MENU
PROMPT
IF d 0 ==
THEN b h * b h 3 ^
* 12 /
ELSE d d ABS / d 2
^ * π * 4 / d 4 ^ π * 64
d d ABS / * /
END 'I(1)' STO 'A(
1)' STO
NEXT 2 MENU
ELSE "Flächenträgheitsmom." n
VEIN "I" STO 'A' STO
VEIN 'I' STO 'A' STO
END Σ(j=1,n,A(j))
FVAL 'AG' STO CLLCD

```

```

"Abst.d.Einzelschwpkte"
1 DISP
"v.Bezgslin/Gesschwpkt"
2 DISP "Taste drücken" 4
DISP KY "Abst.[mm]" n
VEIN 's' STO
"ist der Gesamtschwpkt"
"bekannt ?" JN
IF 1 ==
THEN 3 'X' STO
ELSE A s DOT 'AA' STO
AA AG / 'ABST' STO s 1 n
START ABST
NEXT n →ARRY - 's'
STO 3 FIX
"Abst.Geschwpkt-Bezglin"
"ABST [mm]:" ABST →STR +
CLLCD 3 DISP 1 DISP KY 1
'X' STO
END Σ(j=1,n,I(j)+A(j)
#S(j)^2) EVAL 'IG' STO
"Randfaserabstände [mm]"
( 'E=1' ( E1 E2 ) ABST
( ) CONT ) ) STEQ 30
MENU PROMPT IG E1 / 'W(1
)' STO IG E2 / 'W(2)'
STO 27 MENU
"Das Gesflätträmon betr."
"IG: " IG →STR +
"mm^4" +
"Die Widerstandsmom betr."
"W(1):" 'W(1)' EVAL
→STR + "mm^3" +
"W(2):" 'W(2)' EVAL
→STR + "mm^3" + " DI
DRUK INIT
>

```

Maurice Al-Khaliedy
 Ludwig-Gies-Str. 7
 5000 Köln 71

HP-48SX Würfel-Simulator

Das Programm CUBE besteht aus 3 getrennten Programmen.

1. Die PLAY-Routine (PLAY)
2. Die würfel Bilder (DATA)
3. Die würfel Koordinaten (COOR)

Die Programme sollten in einem eigenen Verzeichnis untergebracht werden. Erstellen Sie mit CHDIR ein Verzeichnis namens CUBE.

In dieses Verzeichnis kommen alle 3 Programme hinein.

Achten Sie darauf, das die GROB's richtig abgeschrieben werden.

```
PLAY
<
GROB 125 6 3B1CC1D90C414
42539D18404A298E6A090180
1551451C655554045049AA82
AA09018C9C51C80457535C1C
5CD8BB8EA409018440514904
455554045049AA82A403B1CD
5C90C80445759D14D14AAABE
640000000000000000000000
000000000000
> t
< ERASE ( # 0d # 0d )
PVIEW 1 6
FOR X PICT DATA X
GET COOR X GET SWAP GOR
PICT ( # 4d # 32d ) t
GOR 500 X * .05 BEEP
NEXT 1 WAIT ERASE (
# 0d # 0d ) PVIEW PICT
COOR RAND 7 * IP DUP
IF 0 ==
THEN 1 +
END GET DATA RAND 7
* IP DUP
IF 0 ==
THEN 1 +
END GET GOR PICT (
# 4d # 32d ) t GOR 7
FREEZE 532 .1 BEEP 659
.1 BEEP 784 .1 BEEP 1047
.1 BEEP .1 WAIT 784 .1
BEEP 1047 .8 BEEP
>
DATA
{
GROB 22 20 FFFFF11000011
000011000011000011000011
0000110000110E00110E0011
0E0011000011000011000011
00001100001100001100001F
}
```

```
FFFF1000000
GROB 22 20 FFFFF11000011
000011000011000011000011
000011000011000011000011
308311000011000011000011
00001100001100001100001F
FFFF1000000
GROB 22 20 FFFFF11000011
000011000011000011000011
3083110000110E00110E0011
0E0011000011008311008311
00831100001100001100001F
FFFF1000000
GROB 22 20 FFFFF11000011
000011000011000011000011
308311000011000011000011
00001100001100001100001F
FFFF1000000
GROB 22 20 FFFFF11000011
000011000011000011000011
3083110000110E00110E0011
0E0011000011008311008311
30831100001100001100001F
FFFF1000000
GROB 22 20 FFFFF11000011
000011000011000011000011
3E8311000011000011000011
000011000011000011000011
3E831100001100001100001F
FFFF1000000
}
COOR
{ ( # 0d # 0d ) ( # 53d
# 0d ) ( # 110d # 0d ) (
# 0d # 45d ) ( # 53d
# 45d ) ( # 110d # 45d )
}
```

Maurice AL-Khaliedy
Ludwig-Gies-Str.7
5000 Köln 71

SENSO für den HP-48 SX by Maurice AL-Khaliedy

Sinn des Spieles ist es , Kästen die auf dem Display erscheinen in der richtigen Reihenfolge zu wiederholen .

Bei diesem SENSO ist das Display in sechs Kästen eingeteilt (Bild 1) , die der Reihenfolge nach nummeriert sind .

Der Computer beginnt zufällig mit dem ersten Zug und der Spieler muß ihn nachvollziehen.

Das ganze geht solange bis der Spieler einen Fehler macht.

Das Spiel besteht aus "11 Programmblöcken" , die alle in ein Verzeichniss mit dem Namen 'SENSO' hinein kommen.

Die Programme der Reihenfolge nach:

START , REPEAT , SCORE , RND , ROTSTACK , GO , TIME ,
BLOCK , TON , COORSENSO , ZUG

Die Geschwindigkeit des Spiels wird durch den Wert , der in der Variable TIME gespeichert ist bestimmt (Test it).

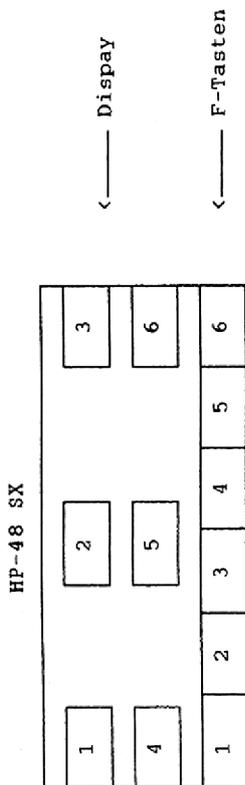


Bild 1

Die Files auf der Diskette (Format MS-DOS 720 KB) sind mit folgenden Programmen erstellt worden.

- SENSO.WRI mit Windows Write
- S-Files.WRI mit Windows Write
- S-BILD1.TXT mit Word 5.0

Auf der Diskette befindet sich auch der Source Code.
Dieser kann direkt zum HP-48SX mit dem Kabel überspielt werden.

SENSO1.BIN
SENSO1.ASC

SENSO1.PRG dieses File ist eigentlich bedeutungslos (ist nicht überspielbar).
Benötigte Ich nur zum Drucken.

```

start
<< CLLCD PICT PURGE { | "1" << 1 GO >> } | { "2" << 2 GO
>> } | { "3" << 3 GO >> } | { "4" << 4 GO >> } | { "5"
<< 5 GO >> } | { "6" << 6 GO >> } } TMENU { } 'ZUG' STO TON
PICT BLOCK COORSENSO RAND 7 *
IP DUP DUP
IF 0 ==
THEN 1 + SWAP DROP DUP
END DUP 6 ROLLD ZUG SWAP + 'ZUG' STO GET SWAP
GOR ( # Od # Od | PVIEW GET Time BEEP
3 FREEZE ROTSTACK >>
REPEAT
<< 1 ZUG SIZE FOR T TON PICT ERASE BLOCK COORSENSO ZUG T GET
DUP 5 ROLLD GET SWAP GOR GET Time BEEP NEXT rnd
>>
SCORE
" << CLLCD
"
Move(s): "
ZUG SIZE + 1 DISP 2 MENU 2 WAIT KILL
>>
rnd
<< TON ERASE PICT BLOCK COORSENSO RAND 7 *
IP DUP DUP
IF 0 ==
THEN 1 + SWAP DROP DUP
END DUP 6 ROLLD ZUG SWAP + 'ZUG' STO GET SWAP
GOR GET Time BEEP ROTSTACK >>
ROTSTACK
<< ZUG EVAL DEPTH 1 FOR W W ROLLD -1 STEP >>

```

```

GO
  << IF 2 PICK ==
    THEN TON SWAP ERASE PICT BLOCK COORSENDO 4 PICK
  GET SWAP GOR GET Time BEEP 3 FREEZE ELSE TEXT CLEAR 1000 1 BEEP SCORE
  END
  IF DEPTH 0 ==
    THEN REPEAT
  END
  >>

Time
.5

BLOCK
GROB 22 20 FFFFFFFF1FFFFFFF1FFFFFFF1FFFFFFF1FFFFFFF1
           FFFFFFFF1FFFFFFF1FFFFFFF1FFFFFFF1FFFFFFF1
           FFFFFFFF1FFFFFFF1FFFFFFF1FFFFFFF1000000

TON
{ 261.624 293.656 329.648 349.228 392.04 440 }

COORSENDO
{ { # 0d # 0d } { # 53d # 0d } { # 110d # 0d }
  { # 0d # 36d } { # 53d # 36d } { # 110d # 36d } }

ZUG
{ }

```

Matthias Sachs
 Birkenstr. 20
 W-4358 Haltern

WURMY

Ein kleiner Wurm frißt sich groß.

Ein Spiel für den HP-48SX

```

+-----+ ! selbst erklärt )! PVIEW { # 0d # 0d }
! WURMY ! ! Mit der Taste SCORE ! { # 129d # 61d }
! ! ! können Sie sich die ! BOX 1 LEVE 30 #
! Ein Geschicklich- ! ! aktuelle High-Score ! FOR i
! keitsspiel für öde ! ! Liste anzeigen las- ! DO 1 CF
! Vorlesungen. Legen ! ! sen. Viel Vergnügen. ! RAND 32 # IP DUP
! Sie ein Verzeichnis ! ! Matthias Sachs ! IF 3 <
! mit dem Namen WURM ! +-----+ THEN 1
! an, und geben Sie ! SF
! darin die nachfol- ! *****WURM***** END
! genden Programme ! * GRDF -,25 RAND 15 # IP NEG
! ein. Der jeweilige ! 32,25 XRNG -15,5 DUP
! Programmname steht ! ,25 YRNG IF -7
! immer zwischen Ster- ! DO ANLT ANL2 == 1 FS? AND
! nen eingerahmt. Zu ! HIGH THEN -1
! beachten ist noch, ! "ENTER = Spielstart" +
! daß die Zeichenkette ! 7 DISP 0 WAIT END R->C
! -> dem kleinen Pfeil ! UNTIL 51,1 DUP BELE
! über der Null-Taste ! SAME UNTIL 0
! entspricht. Zum ! END 1 'LEVE' == END C->PX
! Start gehen Sie in ! STO 0 'PUNKTE' STO PICT SWAP
! das Verzeichnis WURM ! 2 CF CASE i 6
! und wechseln von dort ! DO TEXT CLLCD LEVE # ≤
! in das CST Verzeich- ! "LEVEL " LEVE + 1 THEN P1
! nis. In diesem Ver- ! DISP "PUNKTE " END i
! zeichnis sind zwei ! PUNKTE + 3 DISP 12 LEVE # ≤
! Tasten belegt. Mit ! "TASTE" 7 DISP TAST THEN P5
! der Taste WURM star- ! ERASE { # 0d # 0d } END i
! ten Sie das Programm !
! ( welches sich dann !
    
```

```

15 LEVE # ≤
      THEN P4
      END P3
      END GOR.
      NEXT 15
LEVE # 'FUTT' STO 3
LEVE # 'SFUT' STO
1000 ,1 BEEP
      WHILE KEY
      REPEAT
      END LAUF
'LEVE' INCR 'LEVE'
STO CLEAR
      UNTIL 2 FS?
      END CLEAR
AUSW 2 CF
      ,

*****SCORE*****
      « HIGH "TASTE"
7 DISP TAST
      ,

*****ANL1*****
      « CLLCD
      " WURMY" 1
DISP
      " (C) MSA-SOFT" 3

DISP
      "Ein Wurm frisst sich"
4 DISP
      "durch. Steuerung mit"
5 DISP
      "den Pfeiltasten."
6 DISP "Taste" 7
DISP TAST CLLCD
      " WURMY" 1
DISP "VORSICHT !!!"
3 DISP
      "Ein Beruehren von"
4 DISP
      "Wand, Randlinie oder"
5 DISP
      "Wurm bedeutet ENDE."
6 DISP "Taste" 7
DISP TAST
      ,

*****ANL2*****
      « ERASE PICT (
      # 47d # 0d )
      "WURMY" 3 ->GROB
REPL PICT ( # 2d
      # 10d ) "FUTTER" 1
->GROB REPL PICT (
      # 30d # 10d ) P3
REPL PICT ( # 2d
      # 16d )

"UPERFUTTER" 1
->GROB REPL PICT (
      # 48d # 16d ) P4
REPL PICT ( # 0d
      # 22d )
      "Verlaengern den Wurm"
2 ->GROB REPL PICT (
      # 2d # 30d )
      "SCHLECHTES FUTTER"
1 ->GROB REPL PICT (
      # 72d # 30d ) P5
REPL PICT ( # 0d
      # 36d )
      "Wurm verliert ein Teil"
2 ->GROB REPL PICT (
      # 2d # 44d ) "WAND"
1 ->GROB REPL PICT (
      # 24d # 44d ) P1
REPL PICT ( # 0d
      # 53d ) "Taste" 3
->GROB REPL ( # 0d
      # 0d ) PVIEW TAST
TEXT
      ,

*****HIGH*****
      « CLLCD
      " HIGHSCORE" 1
DISP HILI LIST->

```

DROP ->STR " " SWAP	+ DUP C->R	IF fehl
+ + 2 DISP ->STR " "	IF 1 <	0 ==
SWAP + + 3 DISP	SWAP -1 > AND	THEN
->STR " " SWAP + + 4	THEN DUP	laeng 2 + ROLL PICT
DISP ->STR " " SWAP	DUP DUP BELE 'KOPO'	SWAP C->PX PO REPL
+ + 5 DISP ->STR " "	STO	END
SWAP + + 6 DISP	CASE	IF fehl
*	KOPO 2 ==	0 >
*****LAUF*****	THEN	END
« 5 5 (1;0) 0 ->	'laeng' INCR	rich
laeng fehl rich ta	'laeng' STO 'fehl'	END
« (-1;-7)	INCR 'fehl' STO	END rich
DO	PUNKTE 1 + 'PUNKTE'	'rich' STO
IF KEY	STO 'FUTT' DECR	END DUP C->R
THEN 'ta'	'FUTT' STO	IF 1 <
STO	END	SWAP -1 > AND
CASE ta	KOPO 4 ==	THEN DUP
25 ==	THEN	DUP DUP BELE 'KOPO'
THEN	laeng 3 + 'laeng'	STO
(0;1)	STO fehl 3 + 'fehl'	CASE
END	STO PUNKTE 3 +	KOPO 2 ==
ta 34 ==	'PUNKTE' STO 'SFUT'	THEN
THEN	DECR 'SFUT' STO	'laeng' INCR
(-1;0)	END	'laeng' STO 'fehl'
END	KOPO 1 ==	INCR 'fehl' STO
ta 35 ==	THEN	PUNKTE 1 + 'PUNKTE'
THEN	IF laeng 1 >	STO 'FUTT' DECR
(0;-1)	THEN laeng 2 -	'FUTT' STO
END	'laeng' STO	END
ta 36 ==	END PUNKTE 2 -	KOPO 4 ==
THEN	'PUNKTE' STO	THEN
(1;0)	END	laeng 3 + 'laeng'
END	KOPO 3 ==	STO fehl 3 + 'fehl'
rich	THEN	STO PUNKTE 3 +
END	2 SF	'PUNKTE' STO 'SFUT'
'rich' STO	END	DECR 'SFUT' STO
END rich	END	END
	C->PX PICT SWAP P2	
	REPL	

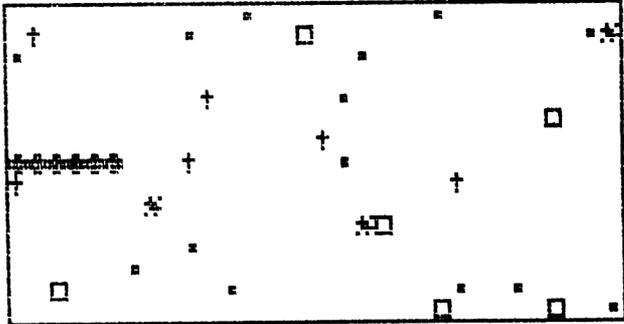
```

KOPO 1 ==          « DUP                                THEN posi
                    (,75;-75) + SWAP                    2 + 'posi' STO
                    C->PX SWAP C->PX PICT                END
IF laeng 1 >      THEN                                  UNTIL
THEN laeng 2 -    3 ROLLD SUB -> s                    PUNKTE ≥ posi 11 ==
'laeng' STO      «                                       OR
END PUNKTE 2 -   CASE s P5                                END
'PUNKTE' STO     SAME
END              THEN 1                                    IF posi 1 >
KOPO 3 ==        END s P3                                THEN TEXT
                THEN 2                                    CLLCD HILI LIST->
2 SF             END s P4                                "HIGHSCORE" 1 DISP
                END                                       "Namen eingeben" 3
                SAME                                       DISP
C->PX PICT SWAP P2 THEN 4                                "Beenden mit ENTER"
REPL             END s P0                                4 DISP "Taste" 7 /
                SAME                                       DISP TAST "Name:" {
0 ==            THEN 0                                    \6a } INPUT 13 posi -
                END 3                                    ROLLD PUNKTE 14
                END                                       posi - ROLLD 13
laeng 2 + ROLL PICT      »                               ROLL DROP 12 ROLL
SWAP C->PX PO REPL      »                               DROP ->LIST 'HILI'
                »                                       STO
                »                                       END HIGH
0 >              »                                       "Taste" 7 DISP TAST
                THEN *****CST*****
                { WURM SCORE
                }
                END
                ELSE 2 SF *****AUSW*****
                END SFUT      « 1 -> posi
                FUTT +      « TEXT CLLCD
                UNTIL 0 ==   "Erreichte Punktzahl:"
                2 FS? OR     1 DISP PUNKTE 3
                END         DISP "TASTE" 7 DISP
                »           TAST
                »           DO HILI
                »           posi GET DUP
                »           IF PUNKTE
*****BELE*****      <
                    *****GRDF*****
                    «
                    GROB 4 4 00000000
                    'PO' STO
                    GROB 4 4 60F0F060
                    'P2' STO

```

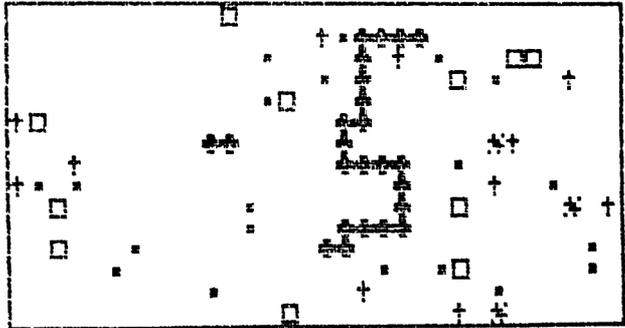
Bilder von WURMY auf dem HP-48SX

```
GROB 4 4 20702020  
'P5' STO  
GROB 4 4 00606000  
'P3' STO  
GROB 4 4 A070E050  
'P4' STO  
GROB 4 4 F09090F0  
'P1' STO  
,
```



```
*****HILI*****
```

```
( 20  
'xxx' 40  
'xxx' 60  
'xxx' 80  
'xxx' 100  
'xxx' )  
END
```



Program CATCH: This game is based on idea of Juergen Wegener in Pocket+Laptop computer. You start as small rectangle in left upper corner and should catch the other subject on the screen using cursors. Try it ain't easy.

```

3 ==
STRT
  << 99 'SCOR' STO
  { # 0h # 0h } PVIEW
  ERASE SCR 4 'ZZ'
  STO 4 'WW' STO 68
  'XX' STO 39 'YY'
  STO PICT { # 4h
  # 4h } XA GXOR PICT
  { # 72h # Dh } "99"
  1 -> GROB GXOR PICT {
  # 44h # 27h } XT 3
  DUPN GXOR
  DO
    DO GXOR XC
    CASE DUP
  1 ==
    THEN
    DROP 'YY' 6 STO-
    END DUP
  2 ==
    THEN
    DROP 5 'XX' STO+
    END DUP
    THEN
    DROP 6 'YY' STO+
    END 4
    ==
    THEN
    'XX' 5 STO-
    END
    PICT XX R->B YY R->B
    2 -> LIST XT 3 DUPN
    GXOR PICT { # 72h
    # Dh } DUP2 SCOR 1
    -> GROB GXOR 'SCOR'
    DECR 1 -> GROB GXOR
    UNTIL KEY
    END PICT
    XAC XA GXOR MOVE
    PREV2 PICT XAC XA
    GXOR
    UNTIL WW 1 -
    XX == ZZ 1 - YY ==
    AND
    END DROP DROP
    DROP 0 WAIT DROP
  >>
  MOVE <<
  LUKA STRAVS
  LUBEJEVA 1
  61000 LJUBLJANA
  SLOVENIJA
  ==
  'ZZ' 6 STO-
  END DUP 36
  ==
  'WW' STO+
  END DUP 35
  ==
  'ZZ' STO+
  THEN DROP 6
  END 34 ==
  THEN 'WW' 5
  STO-
  END
  END
  >>
  PREV
  <<
  CASE XX 3 (
  THEN 3 'XX'
  )

```


IP >>
'ATX [STO]

```

IF A4 B1 ==
A4 B2 == A4 B3 ==
A4 B4 == OR OR OR
THEN 1 'Y'
STO+
END "NUM:"
Y + " POS: " X +
+
[ENTER] to continue"
+ " INPUT DROP X
'Z' STO 0 'X' STO 0
'Y' STO
END
IF Z 4 ==
THEN
CONGRATULATIONS
YOU MADE IT IN "
W + ".TRY" +
ELSE
"YOU ARE OUT OF GAME
THE RIGHT COMBINATION
WAS "
A1 + " + A2 + "
+ A3 + " + A4 +
END " INPUT
    
```

'STRT [STO]

CTX << ATX 'A1'

STO ATX 'A2' STO

ATX 'A3' STO ATX

'A4' STO IF A1 A2

== A1 A3 == A1 A4

== A2 A3 == A2 A4

== A3 A4 == OR OR

OR OR OR THEN CTX

END

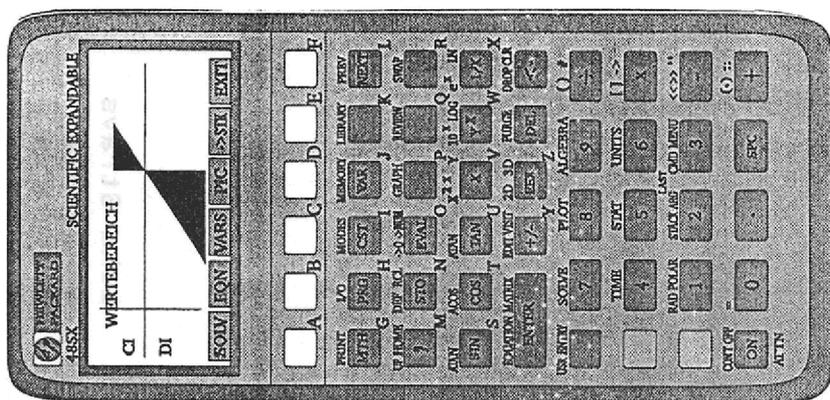
'CTX [STO]

BTX << 1 'X'

STO+ >>

'BTX [STO]

ATX << RAND 10 *



PROGRAM:HANGMAN

This game is well-known to everybody (Gangelman), here are only instructions for using it. At first you store the string with one or more words, that you would like somebody to guess, in variable SEARCH(example: "COMPUTER POCKET TELEVISION PRINTER" 'SEARCH' (STO)) and HP 48 will choose one of them for HANGMAN. Words shouldn't be longer than of 13 letters. Then you start HANGMAN with pressing STRT. You will see the picture with word HANGMAN right above. Then you press any key to continue and then you enter the letter you wish. Confirm with [ENTER] and then repeat the procedure. The listing of program contains programs: main program STRT and subprograms M1 to M6.

Listing:

```

<< { } TMENU CLEAR
GROB 5 1 F1 'POD'
STO SEARCH OBJ->
DEPTH RAND * IP
DROPN ->STR DUP SIZE
1 - 2 SWAP SUB DUP
'SRCH' STO 'SC' STO
CLEAR "MISSED:"
'MIS' STO ERASE
PICT { # 83d # 3d }
'HANGMAN' 2 ->GROB
REPL PICT { # 47d
# 30d } "WORD:" 1
->GROB REPL "" 1
SRCH SIZE
START "0" +
NEXT 'ISK' STO

'M1' 'C' STO { # 0d
# 0d } PVIEW 1 SRCH
SIZE
FOR n PICT n 6 *
41 + R->B #44d 2
->LIST POD REPL
NEXT
DO 0 WAIT DROP
DO
"Enter the letter:"
{ } INPUT 'CRK'
STO
UNTIL CRK SIZE
1 ==
END { # 0d # 0d
} PVIEW SRCH CRK
POS
IF 0 ==

THEN PICT {
# 3d # 55d } MIS
. . + CRK + DUP
'MIS' STO 1 ->GROB
REPL C
ELSE
DO SRCH CRK
POS DUP PICT SWAP
DUP SRCH SWAP DUP
SUB 1 ->GROB SWAP 6
* 41 + R->B # 38d 2
->LIST SWAP REPL
SRCH SWAP "0" REPL
DUP 'SRCH' STO CRK
POS
UNTIL 0 ==
END
END

UNTIL SRCH ISK
SAME
END PICT { # 3d
# 5d }
"YOU GUESSED IT" 1
->GROB REPL { MIS
POD SRCH ISK SC CRK
C } PURGE 0 WAIT
DROP
>>
'STRT [STO]
<< { # 5d # 48d } {
# 16d #48d } LINE
'M2' 'C' STO
>>
'M1 [STO]

```

Luka Stravs
Lubejeva 1
61000 Ljubljana
Slovenija

NEU HP 48SX – der ausbaufähige technisch-wissenschaftliche Taschenrechner

```

Purge 0 Wait Drop
Kill
>>
'M8 [STO]

<< { # 11d # 48d } {
# 11d # 20d } LINE
'M3 'C' STO
>>
'M2 [STO]

<< { # 11d # 20d } {
# 31d # 20d } LINE
'M4 'C' STO
>>
'M3 [STO]

<< { # 11d # 25d } {
# 16d # 20d } LINE
'M5 'C' STO
>>
'M4 [STO]

<< { # 25d # 20d } {
# 25d # 26d } LINE
'M6 'C' STO
>>
'M5 [STO]

<< { # 25d # 29d }
# 3d 0 360 ARC 'M7'
'C' STO

>>
'M6 [STO]
<< { # 25d # 32d }
DUP { # 21d # 37d }
LINE { # 29d # 37d
} LINE { # 25d
# 32d } { # 25d
# 41d } LINE 'M8'
'C' STO
>>
'M7 [STO]

<< { # 25d # 41d }
DUP { # 23d # 50d }
LINE { # 27d # 50d
} LINE PICT { # 3d
# 5d }
"YOU ARE HANGED" 1
->GROB REPL 1 SC
SIZE
FOR n PICT n 6 *
41 + R->B # 38d 2
->LIST SC n n SUB 1
->GROB REPL
NEXT { CRK C ISK
MIS SC SRCH POD }

```

PROGRAM:ROMNUM

This program converts roman numerals in arabic numerals. You just put roman number in the string and start the program R->A.

LISTING:

```

<< 0 'T' 0 'C' 0 'X'
0 'I' STO STO STO
STO DUP SIZE 1 SWAP
FOR n DUP n n SUB
CASE DUP "M"
SAME THEN DROP 'T'
1000 STO+ A1 A2 A3
END DUP "D"
SAME THEN DROP 'T'
500 STO+ A1 A2 A3
END DUP "C"
SAME THEN DROP 'T'
100 STO+ 'C' 1 STO+
A2 A3
END DUP "L"
SAME THEN DROP 'T'
50 STO+ A2 A3
END DUP "X"
SAME THEN DROP 'T'
10 STO+ 'X' 1 STO+
A3
END DUP "V"
SAME THEN DROP 'T'
5 STO+ A3 END
'I' SAME THEN 'T'
1 STO+ 'I' 1 STO+
END
END
NEXT DROP T { T C
X I } PURGE
>>
'R->A [STO]
STO-
A2 <<
IF X 0 >
THEN 'T' 20
END
'A1 [STO]
>>
'A3 [STO]
STO-
A1 <<
IF I 0 >
THEN 'T' 2
END
'A2 [STO]
>>

```

EXAMPLE: You want roman numeral MCMXCI to convert in arabic numeral. You simply put MCMXCI in string and put in the first level of the HP's stack.

```

1: "MCMXCI"
[R->A]

```

PROGRAM:BACK

This program writes text in string or data in list backwards. So if you put on the stack string "ABCD" after evaluating this program you will get "DCBA" or if you put list { 1 2 3 4 } you will get { 4 3 2 1 }.

LISTING:

```

<<
  IF DUP TYPE 2 ==
  THEN **
  ELSE { }
  END SWAP DUP SIZE
1  FOR n DUP n n SUB
  ROT SWAP + SWAP -1
  STEP DROP
  >>
  'BACK [STO]

```

EXAMPLE:

```

1:  "QWERTY"
[BACK]

```

**HP 48SX –
 der ausbaufähige,
 technisch-wissenschaftliche
 Taschenrechner
 für höchste Ansprüche**

PROGRAM: SUMN

This program calculates the size of programs or directories, put in list, in bytes. You put one or more names of programs or directories in a list and then start the program SUMN. It will put the size of memory in bytes that all the programs or directories named in the list take.

LISTING:

```
<< 0 'S' STO DUP 1
SWAP SIZE 1 SWAP
START GETI BYTES
'S' STO+ DROP
NEXT DROP2 S 'S'
PURGE
>>
[ENTER]
'SUMN [STO]
```

Checksum: #43486d

Bytes : 91.5

EXAMPLE: For example, you have programs or directories called PR1, PR2, PR3 and PR4 and you would like to know how much memory of your HP 48 they take.

```
1: { PR1 PR2 PR3 PR4 }
[VAR] SUMN
```

Luka Stravs
Lubejeva 1
61000 Ljubljana
Slovenija

PROGRAM: RENAME

This program renames the program named in the stack level 2 with the name that is put in stack level 1.

LISTING:

```
<< SWAP DUP RCL SWAP
PURGE SWAP STO >>
[ENTER]
'RENAME [STO]
```

Checksum: #46587d
Bytes : 38

EXAMPLE:

For example you have the program stored in variable called PRGOLD and want to rename it in name PRGNEW. You simply put the name PRGOLD in stack level 2 and name PRGNEW in stack level 1 and start the program RENAME.

```
2: 'PRGOLD'
1: 'PRGNEW'
[VAR] RENAME
```

Maurice Al-Khaliedy
 Ludwig-Gies-Str. 7
 5000 Köln 71

Backup - Übertragung des Speicherinhalts

BACKUP ist ein Programm, welches den gesamten Speicher des HP48SX auf den PC bzw. auf einen anderen HP48SX überträgt. Bei dieser Übertragung werden alle Termine sowie die Userkeys mit kopiert. Das Programm benötigt keine Parameter. Der HP benutzt als BACKUP Name das momentane Datum. Bei der Installation verwenden Sie im MEMORY Verzeichnis RESTOR.
 Das Programm sollte im HOME Verzeichnis liegen.
 Weitere Informationen im Benutzerhandbuch ab Seite 663.

```

BACKUP
◀ STD IO "B" DATE →STR
DUP
IF SIZE 8 ==
THEN "Q" SWAP +
ELSE
END DUP 1 2 SUB SWAP
DUP 4 5 SUB SWAP 8 9 SUB
+ + + STR → SWAP →TAG
ARCHIVE
▶
    
```

```

IF DUP 13 ==
THEN DROP 12
END
IF DUP 14 ==
THEN DROP 16
END 5 Start
>
Exit
< UPDIR
>
Start
< 'Zone' STO -42 SF 0
< 'Einheiten' STO 0
< 'Preis' STO CLLCD
<
IF Zone 5 ≠
THEN DATE 2 TRNC DUP
DUP2 3 DUPN
1.05 == OR SWAP 3.1 == SWAP
OR SWAP 24.12 == OR SWAP
25.12 == OR SWAP 26.12
== OR SWAP 31.12 == OR
DATE 0 TSTR 1 1 SUB 'S'
== OR TIME IP DUP 18 1
SWAP 8 < OR OR
THEN 'B'
ELSE 'N'
END 'A'
ELSE 'A' 'Tarif' STO
END DUP 'Tarif' STO
> 'START' } } THMENU
DO -1 WAIT
UNTIL IP 11 ==
END
IF Tarif "A" ≠
THEN
IF Tarif "B" ==
THEN Bliste
ELSE NListe
END Zone GET
END
Ausland
< ( 4.42 10.67 12 16 )
THMENU CLLCD
DO CLEAR
"Zeittakt in Sek.:" 7
DISP -1 WAIT
UNTIL DUP 15 <
END IP
IF DUP 11 ==
THEN DROP 4.42
END
IF DUP 12 ==
THEN DROP 10.667
END
END ( Ende ) THMENU
Takt
>
Ende
< NALM DELALARM 2 RND
CASE Tarif "N" ==
THEN "Normaltarif" {
"FEIER"
< Feiertag 2 MENU
> } { "EXIT"
< 2 MENU
> } } THMENU
END Tarif "B" ==
THEN "Billigtarif" 2
MENU END "Preis" 2 MENU
END "DM" + →TAG
Takt
< DUP 60 / →HMS 100 /
TIME HMS+ DUP
IF 24 1
THEN 24 - DATE 1 DATE+
ELSE DATE 'T1' 4 ROLL
END SWAP 'T1' 4 ROLL
8192 ≠ 4 →LIST STOALARM
'NALM' STO 0 0 NALM T1
>
Feiertag
< DROP BListe Zone GET
NListe Zone GET / / 2
RND CEIL "Einheiten"
→TAG DUP .2277 ≠ 2 RND
"Billigtarif DM" →TAG
T1
< 'NALM' STO 2000 01
BEEP DROP2 'Einheiten'
INCR 'Einheiten' →TAG
'Preis' .2277 STO+ Preis
'Preis DM' →TAG
>
IF DUP 13 ==
THEN DROP 12
END
IF DUP 14 ==
THEN DROP 16
END 5 Start
>
Exit
< UPDIR
>
Start
< 'Zone' STO -42 SF 0
< 'Einheiten' STO 0
< 'Preis' STO CLLCD
<
IF Zone 5 ≠
THEN DATE 2 TRNC DUP
DUP2 3 DUPN
1.05 == OR SWAP 3.1 == SWAP
OR SWAP 24.12 == OR SWAP
25.12 == OR SWAP 26.12
== OR SWAP 31.12 == OR
DATE 0 TSTR 1 1 SUB 'S'
== OR TIME IP DUP 18 1
SWAP 8 < OR OR
THEN 'B'
ELSE 'N'
END 'A'
ELSE 'A' 'Tarif' STO
END DUP 'Tarif' STO
> 'START' } } THMENU
DO -1 WAIT
UNTIL IP 11 ==
END
IF Tarif "A" ≠
THEN
IF Tarif "B" ==
THEN Bliste
ELSE NListe
END Zone GET
END
Ausland
< ( 4.42 10.67 12 16 )
THMENU CLLCD
DO CLEAR
"Zeittakt in Sek.:" 7
DISP -1 WAIT
UNTIL DUP 15 <
END IP
IF DUP 11 ==
THEN DROP 4.42
END
IF DUP 12 ==
THEN DROP 10.667
END
END

```

Telefon-
Gebühren-
Zähler
 für HP-48
 Olaf Rauch
 Mainzer Str. 26
 5600 Wuppertal

{ HOME TEL }
 4:
 3:
 2:
 1: Einheiten: 1
 Preis DM: 2277
 [END] beendet das Pro-
 gram, entweder so :

{ HOME TEL }
 4:
 3:
 2: Einheiten: 3
 1: Billigtarif DM: 68
 [END] (bei Ausland nur
 'Preis DM:'), oder so :

{ HOME TEL }
 4:
 3:
 2: Einheiten: 3
 1: Normaltarif DM: 68
 [END]

Mit 'FEIER' (=Feiertag)
 wird die Anzeige nach-
 träglich auf Billig-
 tarif umgerechnet. Das
 ist nur an fünf Tagen im
 Jahr nötig, die das Pro-
 gram nicht als Feierta-
 ge erkennt. Mit 'EXIT'
 kommt man ohne diese Um-
 rechnung zurück ins An-
 fangsmenü.

Das Programm wird so
 bedient :

Im Variablenmenü 'TEL'
 drücken (oder wie man es
 sonst benannt hat).
 Es erscheint das Menü :

[END] Die dem Bereich entspre-
 chende Taste drücken.
 (Die Bezeichnung 'Regio-
 nal' ist nicht meiner
 Phantasie entsprungen,
 sondern der eines Post-
 beamten. 'Ost' sind die
 neuen Bundesländer, eine
 andere brauchbare Abkur-
 zung ist mir nicht ein-
 gefallen.)
 Hat man 'Ausland' ge-
 wählt, erscheint :

Zeittakt in Sek. :
 [END] Man muß hier den Zeit-
 takt selbst wählen, was
 in den anderen Bereichen
 automatisch geschieht.
 Dann kommt :

[END] , womit man natürlich
 das Programm startet.
 Danach sieht es so aus :

Tarif
 "N"
 Einheiten
 0
 Preis
 0
 Zone
 1
 NAME
 1
 NListe
 (360 60 21 21)
 BListe
 (720 120 42 28)

Anleitung
 Neues Verzeichnis erzeu-
 gen (CRDIR),
 Variablen in der angege-
 benen Reihenfolge ein-
 geben,
 UP drücken,
 Name des Directory ein-
 geben und BYTES ausfüh-
 ren. Es sollte eine Prof-
 summe von # 62350d/1925
 Bytes erscheinen, sonst
 hat man sich vertippt.
 (Auch Groß- und Klein-
 schreibung verändern
 die Profsumme, nicht
 jedoch der Name des
 Directory)

Anmerkungen
 Das Programm 'läuft', nicht eigentlich, sondern verwendet die Uhr des Rechners. Der Stromverbrauch ist daher auch nur halb so groß wie bei einem richtigen Programm. Man kann den Rechner bei den längeren Zeittakten auch während durch ausschalten, er wacht beim nächsten Takt wieder auf. Die Zählung kann nicht durch ATTN unterbrochen werden, man muß ENDE drücken, oder den vom Programm erzeugten Alarm aus der Alarmliste löschen.
 Die Feiertagsumrechnung kann im Bereich 'OST', evtl. eine Einheit zuviel errechnen, da sich aus der Anzahl der Einheiten die Sprechdauer nicht genau rekonstruieren läßt, in den anderen Bereichen ist die Umrechnung korrekt.
 Außerdem:
 Telefoniert man bspw. von 17:59 bis 18:05, gilt der anfangs berechnete (Normal-)zeittakt während des ganzen Gesprächs [START speichert in 'tarif' zunächst ein Programm zur Zeittaktberechnung. Beim

ersten Aufruf von 'tarif', löscht sich 'tarif' selbst und speichert 'A', 'N' oder 'B' in 'tarif'.
 Mögliche Abhilfe: Um 18:00 ENDE drücken, Bereichstaste drücken, START drücken.
 Die Einheiten des ersten Teils bleiben in den Ebenen 3 bzw. 4 stehen, am Ende zu 1: bzw. 2: addieren.
 Falls sich die Gebühren ändern: MListe und Bliste enthalten die Zeittakte in Sek. für Normal- und Billigtarif, man braucht sie nur dort zu ändern.

HP 48SX

The DEMO BOOK



September 21, 1990

Settings

Keyboard - Hidden functions

Memory Management

RAM Card

Library

DOWNLOAD a Backup file from PC

Polar Graphic

Infrared Transfer

Remote Host Commands from PC

Program edition on PC

Units Management

Backup to PC & to RAM card

Graphic printings to I/R printer,
HP PCL graphic printers &
Epson compatible graphic Printers

by Christian CHAFFARD
Version 1.0

Calculators Technical Support. Europe

Copyright © 1990 HEWLETT-PACKARD COMPANY

Durch Information vorn!

Table of Contents

Section 1

Material necessary

Section 2

Settings

2.1 Keyboard conventions	2-1
2.2 Date & Time (European)	2-1
2.3 Permanent Clock Display	2-1
2.4 Point, Comma Settings	2-2

Section 3

Keyboard

3.1 Different Alpha Locks	3-1
3.2 Hidden Keyboard functions	3-1

Section 4

Memory Management

4.1 Creation of Directories	4-1
4.2 Directories & Variables creation	4-1

Section 5

RAM Card

5.1 Variables creation in a Card	5-1
5.2 MERGE a RAM Card	5-2
5.3 PORT 0 functions	5-2
5.4 FREE a RAM card	5-3

Section 6

Library

6.1 Execution of a Library	6-1
6.2 DETACH a Library from a Directory	6-2

Section 7

LOAD a Backup file from PC

7.1 HP 48SX Setup	7-1
7.2 PC Setup	7-1
7.3 Communication test	7-2
7.4 DOWNLOAD BACKUP file from PC	7-2

Table of Contents

Section 8
POLAR Graphic

8.1 DRAW a Polar Graphic. 8-1
8.2 Create a Graphic Object file (GROB) 8-1
8.3 Display a GROB file (Graphic Object). 8-2

Section 9
INFRARED Transfers

9.1 HP 48SX I/R Files Transfers 9-1
9.2 File Transfer from a HP 28S to a HP 48SX 9-2

Section 10
PC Prg. Edition and HP 48SX commands

10.1 Execute HP 48SX commands from a PC. 10-1
10.2 HP 48SX Program edition on PC 10-2

Section 11
Units management

11.1 Units conversion 11-1

Section 12
Backup to PC & RAM card

12.1 Full Memory Backup on PC 12-1
12.2 Full memory backup on RAM card. 12-2
12.3 RESTORE a Backup file from a RAM card 12-2

Section 13
Graphic printings

13.1 Graphic on HP 82240A/B I/R printer 13-1
13.2 graphic printing on a HP PCL printer 13-1
13.3 Graphic Printing on an EPSON compatible printer. 13-3

Material necessary

SECTION

1

In order to work with this booklet, you will need the following material :

- 1 PC IBM Compatible
- 2 HP 48SX calculators
- 1 HP 82208A PC link IBM Comp.
- 1 plug converter 25 pins male / male
or 1 Triport converter (9 pins, 25 pins male / female)
- 1 HP PCL graphic printer having a RS 232 Serial Interface
(one HP Laserjet II is used in the Demo)
- 1 Epson compatible graphic printer
(one Kodak - diconix 150 plus is used in the Demo)
- 1 32 Kb RAM card (HP 82214A)

Settings

SECTION

2

2.1 KEYBOARD CONVENTIONS

To use this booklet, please use the following keyboard conventions :

Direct access will use the { } delimiters, ie {ENTER}

Shift Left will use {AMBER}

Shift Right will use {BLUE}

Alpha key will use {ALPHA}

The 6 white function keys below the screen display will use the [] delimiters, ie [MEM]

The brackets {} key will be displayed {+}

You will find below the different ways to go in Alpha Lock (Upper & Lower Case) :

Alpha Lock (Upper Case) : {ALPHA} {ALPHA}

Reset Alpha Lock (Upper & Lower) {ALPHA}

Alpha Lock (Lower Case) : {ALPHA} {ALPHA} {AMBER} {ALPHA}

CAUTION

If you are in Lower Case Alpha Lock and you depress {ALPHA} to unlock, the next time you go in Alpha Lock, you will be in Lower Case. To go back in Upper Case, go in Alpha Lock then depress {AMBER} {ALPHA}.

3.2 HIDDEN KEYBOARD FUNCTIONS

There are some very useful functions which you can access directly :

{<-} BACKSPACE performs a DROP function

{<} Left Arrow performs a GRAPH function

{UP} UP Arrow activates the Interactive Stack

{DOWN} DOWN Arrow activates the EDITOR function

{>} Right Arrow performs a SWAP function

Some keys have direct functions available through the Right Shift key {BLUE}

{BLUE} {PRINT}	Performs a PR1 function (Level 1 stack print)
{BLUE} {IO}	Places the calculator into Kermit SERVER mode
{BLUE} {MODES}	Selects the customize menu
{BLUE} {MEMORY}	Selects the storage arithmetic menu
{BLUE} {SOLVE}	Selects the SOLVR menu
{BLUE} {PLOT}	Selects the PLOTR menu
{BLUE} {ALGEBRA}	Selects the EQUATION Catalog
{BLUE} {TIME}	Selects the Alarm Catalog
{BLUE} {STAT}	Selects the statistics calculation menu
{BLUE} {UNITS}	Selects the units commands menu
{BLUE} {PREV}	Moves to first page of Current menu

2.2 Date & Time (European)

Set the following date and time : 1st September 1990 at 14:00

{AMBER} {TIME} [SET]	Accesses TIME menu
[M/D]	Set European Month / Day format
[12/24]	Set European time
1.091990 [->DAT]	Set Date
14.00 [->TIM]	Set time

2.3 PERMANENT CLOCK DISPLAY

If you want to display the Date & Time permanently on the top right of the display, depress the following keys :

{AMBER} {MODES} {NXT} [CLK]	Set the clock display (a DOT appears in the "CLK" softkey display)
-----------------------------	--

To suppress the clock display, depress the same keys (toggle function).

WARNING

When you do an "ARCHIVE" function, the clock display MUST be OFF

2.4 POINT, COMMA SETTINGS

Set the comma instead of numeric point : from 1.23 format to 1,23 format

{AMBER} {MODES} {AMBER} {PREV} [FM,]	A dot appears in the softkey display
--------------------------------------	--------------------------------------

CAUTION

When you want to enter a complex number in "Comma" format, you still need to use the "," which is {AMBER} { . }; The display will show you ";" ,ie (1;2)

Keyboard

SECTION

3

3.1 DIFFERENT ALPHA LOCKS

And two other functions :

{ALPHA} {BLUE} {::}	displays only one colon ":"
{ALPHA} {BLUE} {""}	Displays only one quote "

Memory Management

SECTION

4

4.1 CREATION OF DIRECTORIES

The HP 48SX has the same tree structure as MS-DOS Operating System. We will create the directory DIR1 in the HOME directory and DIR2 in the directory DIR1.

{BLUE} {HOME}	Goes to the HOME directory
{ALPHA} {ALPHA} DIR1 {ENTER}	Put DIR1 on the stack
{AMBER} {MEMORY} [CRDIR]	Creates the directory DIR1
{VAR} [DIR1]	Goes in DIR1 directory (the name appears on the top left of the display)
{ALPHA} {ALPHA} DIR2 {ENTER}	Put the name DIR2 on the stack
{AMBER} {MEMORY} [CRDIR]	Creates the directory DIR2
{VAR} [DIR2]	Goes into the directory DIR2 (DIR1 & DIR2 appear on the top left of the display)
{BLUE} {HOME}	Comes back to HOME

If you want to purge the directories DIR1 and DIR2 at the same time :

{VAR} ['] [DIR1] {AMBER} {MEMORY}	Purges DIR1 & DIR2 independently of directories contents
{AMBER} {PREV} [PGDIR]	

4.2 DIRECTORIES & VARIABLES CREATION

You can create simultaneously a directory and some variables included in it.

Example : Create the directory DIR3 containing 2 Variables A and B with the values 100 and 200.

{ALPHA} {ALPHA} DIR {SPC} A {SPC} 100 {SPC} B {SPC} 200 {ENTER}	Put the variables names and values on the stack
{ALPHA} {ALPHA} DIR3 {ENTER}	Put the directory name on the stack
{STO}	Create the directory and variables
{VAR} [DIR3] [A] [B]	Shows the result

Durch Information vorh!

RAM Card

SECTION

5

5.1 VARIABLES CREATION IN A CARD

CAUTION

Power off the HP 48SX before plugging-in the RAM card.

Plug the RAM card in Port # 1 (inner port from bottom side).

Power on the calculator.

Example : Creation of a variable ABC containing the value 123 in Port 1

```
{AMBER} {LIBRARY} [PORT1]           Accesses Port1 and checks the variables
                                       names (none if it is a new card)
```

WARNING

If you use a brand new card, you could have an error message telling the RAM card is defective. Don't take care of this message because the variable creation will initialize the card and remove the error message.

```
123 {ENTER}                           Put 123 on the stack
{BLUE} {::} 1 {>} {ALPHA} {ALPHA}     Enters the variable name and its address
ABC {ENTER}                             ":1:ABC"
{STO}                                    Creates the variable in Port1
```

To verify :

```
{AMBER} {LIBRARY} [PORT1]           ABC variable is created
[ABC]                                 Shows the value "123"
```

5.2 MERGE A RAM CARD

We will merge the 32 Kb RAM memory in Port1 with the 32 Kb main memory.

Memory values before the Merge in main and in RAM card memories :

{AMBER} {MEMORY} [MEM]	Value for example : 5767 Kbytes (available memory)
{NXT} 1 [PVARs]	"{:1: ABC} 32742 Kbytes" Gives the value of available memory in the card in Port1 (assuming it is a 32 KB) and the variable name ABC recorded in the card
1 {AMBER} {MEMORY} {AMBER} {PREV} {MERG}	Merges the RAM card with the main memory
{NXT} [MEM]	"38503 Kbytes" Shows the total available memory (Main + RAM in Port1)
{NXT} 1 [PVARs]	"SYSRAM" Shows the RAM card as part of the main memory
{AMBER} {LIBRARY} [PORT0]	Shows in RAM card (Port1) the variable "ABC" which was in RAM card (Port1). This port is a "dummy" port which is part of the main memory and is used for different purposes.
	In this case, we saved the existing variables from the RAM card in Port1.

Durch Information vorn!

5.3 PORT 0 FUNCTIONS

- Store and recall variables : Can be used to hide variables out of the normal VAR menu.
- ARCHIVE and RESTORE functions : Can be used to backup a full main memory.
- LIBRARY function : Can be used as a port to execute Library programs. This port can be accessed as Port # 1 and # 2 by using the prefix ":0:". It is part of the LIBRARY menu.

Example : Store the variable DEF with the value 500 in Port 0

500 {ENTER}	Put 500 on the stack
{BLUE} {::} 0 {>} {ALPHA} {ALPHA}	Enters the variable name and its address
DEF {ENTER}	":0:"
{STO}	Creates the variable in Port 0

5.4 FREE A RAM CARD

We will "free" from the main memory, the RAM card # 1 which was merged. At the same time we will record the variable "ABC" which was previously in the RAM card (now located in Port 0).

{AMBER} {LIBRARY} [PORT0] {AMBER} {+}	"{:0:ABC}" Put the variable and its address (Port0) on the stack
[ABC] {ENTER}	
1 {AMBER} {MEMORY} {AMBER} {PREV} {FREE}	Frees the RAM card in Port 1
{AMBER} {LIBRARY} [PORT1]	Shows the variable "ABC"
{AMBER} {MEMORY} {NXT} 1 [PVARs]	32742 Kbytes Shows the independent memory available in the RAM card # 1

Library

SECTION

6

6.1 EXECUTION OF A LIBRARY

Libraries are used to extend the main Memory in terms of programs execution. A program Library can ONLY be executed in the Library menu in Port 0, 1 or 2. A program library CANNOT be read or edited. It cannot be copied if it is located on a ROM card.

We will take the program PCLPRIN.LIB from the PC Link diskette to make some tests. (This program has been downloaded during the Restore, refer to Section 7).

{VAR} {NXT} [PCLPR] "Library 767 : HPR..." Put the program content on the stack in Library format

To execute this program which is a Graphic editor for HP PCL printers, you need to copy it in a port.

{BLUE} {::} 0 {>} {ALPHA} A {ENTER} Put the name and address, where we will copy the program, on the stack. The name "A" is meaningless.

{STO} Stores the program PCLPRINT.LIB in Port0
{AMBER} {LIBRARY} [PORT0] The library LID # 767 is recorded (the name "A" is not existing).

{BLUE} {ENTRY} [767] {ENTER} ":0: 767" Put the library name and address on the stack

To execute this program directly from the "Library" menu keys, we will need to "ATTACH" it to the current directory.

{AMBER} {MEMORY} {NXT} [ATTACH] Attaches the Library program to the "HOME" directory.

To get the program attached, power OFF and ON the HP 48SX.

{AMBER} {LIBRARY} You will find a new menu key "HPPRT" which is the Executable PCLPRINT.LIB prog (Refer to the section 14 for execution details).

{AMBER} {MEMORY} {NXT} [LIBS] Gives information about the attached Libraries, ie {HPPRT : Graphic -> PCL 767 0}
HPPRT is the Library name; "767" is the LID (Library Identification Number);
"0" is the port where the library prog. is stored; "Graphic -> PCL" is the prog. description.

6.2 DETACH A LIBRARY FROM A DIRECTORY

To detach a Library, you need NOT to execute it and NOT have a copy of it on the stack.

{BLUE} {CLR}	Clears the stack
767 {ENTER}	Put the LID # of the PCLPRINT.LIB Prog. on the stack
{AMBER} {MEMORY} {NXT} [DETACH]	Detaches the Library PCLPRINT.LIB
{AMBER} {MEMORY} {NXT} [LIBS]	"{ }" No library attached
{AMBER} {LIBRARY}	Shows the Library menu, "HPPRT" is not displayed anymore
Delete the library "767" (PCLPRINT.LIB)	
[PORT0] {AMBER} {+} [767] {ENTER}	{:0: 767} Put the Library number on the stack.
{AMBER} {PURGE}	Purges the Library from teh Port 0. PCLPRINT.LIB is still present in the VAR menu

NOTE

The ROM libraries and the UTILITIES from the PC Link diskette use an automatic "ATTACH". When they are loaded in a port, a power OFF/ power ON will attach them in the library menu.

LOAD a Backup file from PC

SECTION

7

7.1 HP 48SX SETUP

{AMBER} {IO} [SETUP]	
"IR/WIRE : WIRE"	if different, depress [IR/W]
"ASCII/BINARY : ASCII"	if different, depress [ASCII]
"BAUD : 9600"	if different, depress [BAUD] until 9600 appears
"PARITY : NONE 0"	if different, depress [PARIT]
"CHECKSUM TYPE : 3"	if different, depress [CKSM] until "3" appears
"TRANSLATE CODE : 3"	if different, depress [TRAN] until 3 appears

7.2 PC SETUP

You need to load the KERMIT protocol in your PC from the PC Link diskette. (HP 82208A for IBM Compatible PC or HP 82209A for Macintosh PC)

Check your port configuration.

Connect your cable on Port COM1.

From DOS Prompt, type C:> KERMIT

Kermit-MS> STAT {ENTER} Verifiy if speed is OK (9600 Bauds)

If speed not OK, specify it :

Kermit-MS> SET BAUD 9600 {ENTER} Set the speed to 9600 Bauds

WARNING

KERMIT Protocol works only on Port COM1 and COM2.

By default, Kermit is set to Port1 and Speed 2400 Bauds.

If you want to set to port2, Type "SET PORT COM2" at the Kermit prompt.

If you want to set automatically the speed to 9600 Bauds, add the following line in the file AUTOEXEC.BAT : "mode COM1:9600,n,8,1,p"

7.3 COMMUNICATION TEST

Set the HP 48SX in SERVER mode :

{BLUE} {IO}

On PC :

Kermit-MS> REMOTE HOST 1 {ENTER} Put 1 on the HP 48SX stack. The first stack line must be returned "1: 1"

7.4 DOWNLOAD BACKUP FILE FROM PC

{BLUE} {IO} Set the HP 48SX in Kermit Server Mode

On PC :

=====

Load your "DEMO" diskette in Drive A

In Kermit :

Kermit-MS> SEND A:/BACKDEMO Downloads the Backup file from the PC to the Calc. (about 6 Kbytes length)

The backup file is downloaded in the HP 48Sx and we will use the "RESTORE" function to create the HP 48SX variables.

{ATTN} Goes out of Server Mode ({ON} key)
{VAR} [BACK] "Backup HOMEDIR" Put the Backup file content on the stack.

{AMBER} {MEMORY} {AMBER} {PREV}

WARNING

This operation will destroy your HP 48SX main memory

[RESTO] Restores the full memory from the Backup file

POLAR Graphic

SECTION

8

Durch Information vorn!

8.1 DRAW A POLAR GRAPHIC

The HP 48SX must be downloaded as described in Section 7. We will draw a Polar Graphic from a function recorded in the PLOT menu.

[CAT]	Shows the equations recorded
{DOWN} {DOWN}	Places the cursor in front of the equation called "POL"
[PLOT] [AUTO]	Draws the graphic

8.2 CREATE A GRAPHIC OBJECT FILE (GROB)

We will store the created Polar Graphic as a variable called a GROB file (Graphic Object).

{STO} {ATTN}	Put the Graphic 131 X 64 Pixels on the stack and goes out of GRAPH display
{ALPHA} {ALPHA} POLG {ENTER}	Put the graphic variable name on the stack
{STO}	Creates the GROB variable called "POLG"

8.3 DISPLAY A GROB FILE (GRAPHIC OBJECT)

We will display the GROB variable "POLG" created before and recorded in the "VAR" menu.

{VAR} [POLG]	Put the GROB on the stack "Graphic 131 X 64"
{PRG} [DISPL] [PICT]	Put PICT name on the stack
{STO}	Stores the GROB variable in PICT
{AMBER} {GRAPH} {AMBER} {GRAPH}	Displays the graphic without showing the function keys
{ATTN}	Goes out of GRAPH

In order not to have problem of duplicate file, we will purge the GROB called "POLG"

{VAR} {'} [POLG] {AMBER} {PURGE} Purges the file "POLG"

INFRARED Transfers

SECTION

9

9.1 HP 48SX I/R FILES TRANSFERS

Transfer a Graphic Object from a HP 48SX to a second one.

HP 48SX setup (Receiver side) :
=====

{AMBER} {IO} [SETUP] [IR/W]	Shows IR/WIRE : IR (Works as a toggle) must be set to "IR"
[ASCII]	Shows ASCII/BINARY : BINARY (works as a toggle), must be set to "BINARY"

All the other parameters are meaningless

{BLUE} {IO}	Set the Calc. in Kermit server Mode. The Calc. waits until it gets any data from the HP 48SX Transmitter
-------------	--

HP 48SX Setup (Transmitter side) :
=====

- This HP 48SX needs to be downloaded according to Section 7

{AMBER} {IO} [SETUP] [IR/W] [ASCII] Set IR and BINARY parameters as above

{VAR} {'} [PRTE] {ENTER}	Put the GROB name PRTEG on the stack
{AMBER} {IO} [SEND]	Sends the GROB file to the HP 48SX Receiver.

When line activity is off, depress {ATTN} on the HP 48SX Receiver. It will go out of Server Mode.

{VAR} [PRTE] {PRG} [DSPL] [PICT]	
{STO}	Stores the GROB file in PICT
{AMBER} {GRAPH} {AMBER} {GRAPH}	Displays the Graphic
{ATTN}	Goes out of Graphic Display

9.2 FILE TRANSFER FROM A HP 28S TO A HP 48SX

To transfer from any HP calculator having the IR Interface to a HP 48SX, we need to use the program "INPRT" located on the PC Link diskette. This program is downloaded on the HP 48SX (refer to Section 7).

On the HP 28S :
=====

123456789 (ENTER)

Put the numbers on the stack

On the HP 48SX :
=====

{AMBER} {IO} [SETUP] [IR/W] [ASCII] If necessary, to set IR and BINARY in the 2 first fields

Place the HP 48SX Infrared arrow in front of the Infrared diode of the HP 28S. The distance must be not more than 1 centimeter. The HP 28S must have the cover closed on the back in order to get the correct height.

{VAR} [INPRT]

Set the HP 48SX in waiting mode

On the HP 28S :
=====

{SHIFT} {PRINT} [PR1]

sends the STRING from the HP 28S to the HP 48SX

The display on the HP 48SX should be :

2: "123456789"
1: 1

The value of "1" on Level 1 means the transmitted characters are valid. A "0" means there were some invalid characters transmitted. If some characters are missing, the flag (0 or 1) could be set to "1", which means you need to be sure that the distance between the two calc. is respected. We remind you that there is no handshake between the calc. because the HP 28S is only one way transmission.

The data received are always in a STRING. If it is a program, you need to EDIT it and adapt it.

Durch Information vorn!

PC Prg. Edition and HP 48SX commands

SECTION

10

10.1 EXECUTE HP 48SX COMMANDS FROM A PC

Refer to Section 7 to set up the PC Link and the HP 48SX connection.

We will calculate the following equation :

$$\frac{(15 \wedge 3) * XROOT(3,8)}{4 \wedge 2}$$

We will use the Kermit function "REMOTE HOST" and we will define a macro instruction called RH. In fact we will calculate from the PC keyboard without touching the HP 48SX.

From MS-DOS :

```
C:> KERMIT
Kermit-MS> DEFINE RH REMOTE HOST \%1
```

Connect the HP 48SX to the PC and set the HP 48SX in Kermit Server mode

{BLUE} {IO}

From the PC (must be in CAPS LOCK) :
=====

Kermit-MS> RH CLEAR	Clears the stack
Kermit-MS> RH 15	Put 15 on the stack
Kermit-MS> RH 3	Put 3 on the stack
Kermit-MS> RH ^	3375 (executes 15 ^ 3)
Kermit-MS> RH 8	Put 8 on the stack
Kermit-MS> RH 3	Put 3 on the stack
Kermit-MS> RH XROOT	Executes XROOT(3,8) value = 2
Kermit-MS> RH *	Multiplies 3375 by 2
Kermit-MS> RH 4	Put 4 on the stack
Kermit-MS> RH SQ	Executes the square of 4
Kermit-MS> RH /	421.875 Divides 6750 by 16

10.2 HP 48SX PROGRAM EDITION ON PC

We have a program called "WALK" recorded on the supplied diskette. We need to setup the PC Connection as described in Section 7.

Insert the diskette in drive A

C:> TYPE A:\WALK	Displays the program content written in RPL (HP 48SX language). Note the special Header at the first line and all the special characters having "/"
------------------	---

The HP 48SX must be in Server Mode : {BLUE} {IO}	Set the calc. in Server mode
---	------------------------------

On PC :
=====

Kermit-MS>SEND A:\WALK	Sends the program to the HP 48SX
Kermit-MS> REMOTE HOST WALK	Executes the program from the PC. The little guy should walk across the screen

To stop it, on the HP 48SX depress {ATTN}

NOTE

If "WALK" program already exists on the HP 48SX, the calc. will create the file name "WALK.1"

Units management

SECTION

11

11.1 UNITS CONVERSION

To illustrate HPSolve and the units conversion, we will calculate a car consumption using the following equation :

$$\text{CONS} = \text{GALON} / ((\text{END.KM} - \text{BEG.KM}) / 100)$$

CONS is the consumption in liter per 100 km
 GALON is the refill value in UK Gallons
 END.KM is the odometer value when we finish the trip
 BEG.KM is the odometer value when we start the trip

The equation has been downloaded in Section 6

1) We start a trip with an odometer value of 25000 km. When we stop to refill our tank, the odometer shows 26000 km. We refill 28 UK Gal of gas. What is the car consumption ?

{AMBER} {SOLVE} [CAT] Goes in the equations catalog
 Move, with the {UP} or {DOWN} keys, the cursor in front of the equation called "L.KM"

{SOLVR}	Solves the equation
25000 [BEG.]	Enters 25000 km in BEG.KM
26000 [END.]	Enters 26000 km in END.KM
28 [GALO]	Enters 28 UK Gallons in GALON
{AMBER} [CONS]	12.7290 l/km Calculates the car consumption in liter per 100 km.

2) With the same consumption, how many liters we will need to drive during 2000 km ?

{AMBER} {REVIEW}	Shows the different parameters
27000 [END.KM]	New value when we stop the car, the BEG.KM value will remain the same
{AMBER} [GALO]	56 UK Gal Calculates the gas consumption

{AMBER} {UNITS} [VOL] {NXT}	
{AMBER} [L]	254,581 Liters Converts 56 UK Gal in Liters

Backup to PC & RAM card

12.1 FULL MEMORY BACKUP ON PC

We will do a full memory backup on a PC. We need to setup the PC link connection as described in Section 7.

On PC (in MS-DOS) :
=====

C:\ KERMIT (ENTER) Enters in Kermit protocol
Kermit-MS> SERVER (ENTER) Set the PC in SERVER mode

On the HP 48SX :
=====

{BLUE} {CLR} Clears the stack

Verify that the I/O is set to WIRE and ASCII

{AMBER} {IO} [SETUP] [IR/W] [ASCII] Needs to have "WIRE" and "BINARY" set
{BLUE} {::} {ALPHA} {ALPHA} IO
{ALPHA} {>} {ALPHA} {ALPHA} BACKPC
{ENTER} Put ":IO: BACKPC" on the stack

{AMBER} {MEMORY} {AMBER} {PREV}

WARNING

When you do an "ARCHIVE" function, the clock display MUST be OFF

[ARCHI] Backups the full memory to the PC

At the end of transfer, {CTRL} C will exit the PC Kermit server mode

12.2 FULL MEMORY BACKUP ON RAM CARD

WARNING

Power Off the HP 48SX before you plug-in the RAM card in Port #1.

A warning message, "Invalid Card data" could appear if the RAM card is a new one. The message will disappear when some data's will be recorded on the card.

```
{BLUE} {::} 1 {>} {ALPHA} {ALPHA}
BACKRAM {ENTER}           Put ":1: BACKRAM" on the stack
{AMBER} {MEMORY} {AMBER} {PREV}
[ARCHI]                   Backups the full memory on the RAM card
{AMBER} {LIBRARY} [PORT1] Shows "BACK" as a variable
```

12.3 RESTORE A BACKUP FILE FROM A RAM CARD

In Section 7, we saw how we can RESTORE the full memory from a PC backup file. We will do it now from a RAM card backup file.

Install the RAM card containing the Backup file previously recorded.

```
{AMBER} {LIBRARY} [PORT1] {BLUE}
{ENTRY} [BACK] {ENTER}       Put ":1: BACKRAM" on the stack
{AMBER} {MEMORY} {AMBER} {PREV}
```

WARNING

This function will overwrite the main memory

```
[RESTO]                   Restores the full memory from the
                           Backupfile
```

Graphic printings

SECTION

13

13.1 GRAPHIC ON HP 82240A/B I/R PRINTER

We will print the GROB file called "TITENG"

```
{VAR} [TITEN]             Put the GROB file on the stack
                           "Graphic 131 X 64"
{PRG} [DSPL] [PICT] {STO} Stores the GROB file in PICT
```

Power ON the IR printer and place it in front of the HP 48SX.

```
34 {+/-} {BLUE} {MODES} {NXT} [CF] Set the output to I/R
```

```
4 {PRG} [OBJ] {NXT} {NXT} [CHR]
{BLUE} {PRINT}           Places the print head to the right
                           in order to avoid graphic misalignment
```

{AMBER} {GRAPH} {AMBER} {GRAPH} Displays the graphic file

Depress simultaneously the {ON}
and {PRINT} keys Prints the graphic

13.2 GRAPHIC PRINTING ON A HP PCL PRINTER

We will use a HP Laserjet II printer.

Setup the printer :

The printer must be configured in Serial mode and at 9600 bauds.
Keep the "MENU" key depressed during 10 seconds and your configuration menu will be displayed.
Depress 2 times "MENU" to get "I/O SERIAL *", if not correct, depress on "+" key and "ENTER" key to record.

Depress "MENU" again and "BAUD RATE 9600" will appear, if not correct, select the speed value with "+" or "-" key and depress "ENTER" when "9600" is selected.

Depress again "MENU" and "ROBUST XON=ON *" appears, if not correct, depress on "+" and "ENTER" to record.

Depress on "MENU" 2 times more and It will come back to "READY" state.

Cable connection :

Connect your HP 48SX RS 232 cable to a 25 pins male/male converter (Straight connections) or use the tri-port connector specified in the Technical Support newsletter N. 5 (9 pins male/ 25 pins male and female).

To print on the HP graphic printers, we will need to use the program PCLPRINT.LIB provided in the PC Link diskette. This program is already downloaded (refer to Section 7).

```
{VAR} {NXT} (if necessary) {PCLPR} "Library 767 : HPPR..." comes on the
stack
{BLUE} {::} 0 {>} {ALPHA} A {ENTER}
{STO} Stores the Library in Port0
Power Off/ Power On Attaches the library to the directory

{VAR} [PRTE] Put the Graphic Object on the stack
"Graphic 131 X 64"
{AMBER} {LIBRARY} [HPPRT] Starts the PCL graphic Prog.
[HPON] Set the graphic parameters
1 [MAG] Selects a magnitude of 1
{BLUE} {PRINT} Downloads the graphic to the printer
(Ready indicator should blink)
{ALPHA} {ALPHA} FF {ENTER} Performs a Formfeed to eject the paper
```

If you want a larger graphic, do the same procedure but set a magnitude of 4

```
{<-} Drops the Form Feed character from the
stack
4 [MAG] {BLUE} {PRINT} {ALPHA}
{ALPHA} FF {ENTER} Prints a larger graphic
```

13.3 GRAPHIC PRINTING ON AN EPSON COMPATIBLE PRINTER

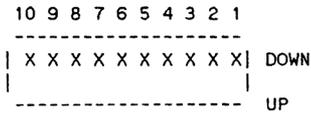
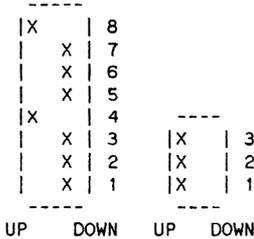
We will use the KODAK printer "DICONIX 150 Plus" to do the printing.

Printer Setup :

There are three blocks of switches located in the print mechanism. Two are located on the left side and one is near the front panel (called SW1). This switch SW1 must have the 10 switches in DOWN position.

The 2 other blocks, located on the left, one has 3 switches, all these switches must be in UP position.

The second block of 8 switches must have the switches 4 and 8 in UP position and all the other ones in DOWN position.



Durch Information vorn!

Cable connection :

Connect your HP 48SX RS 232 cable to a 25 pins male/male converter (Straight connections) or use the tri-port connector specified in the Technical Support newsletter N. 5 (9 pins male/ 25 pins male and female).

On the HP 48SX :

We will need to use a program called "EPSPRINT.LIB" provided in the PC Link diskette and already downloaded in the HP 48SX.

- | | |
|-------------------------------------|--|
| {VAR} [EPSPR] | "Library 766 : EPPR" Put the library on the stack |
| {BLUE} {::} 0 (>) {ALPHA} A {ENTER} | |
| {STO} | Stores the Library in Port0 (A is meaningless) |
| Power Off / Power On | Does an automatic attached of the library to the directory |
| {VAR} [TITEN] | Put the graphic on the stack |
| {AMBER} {LIBRARY} [EPPRT] | Starts the EPSON print program |

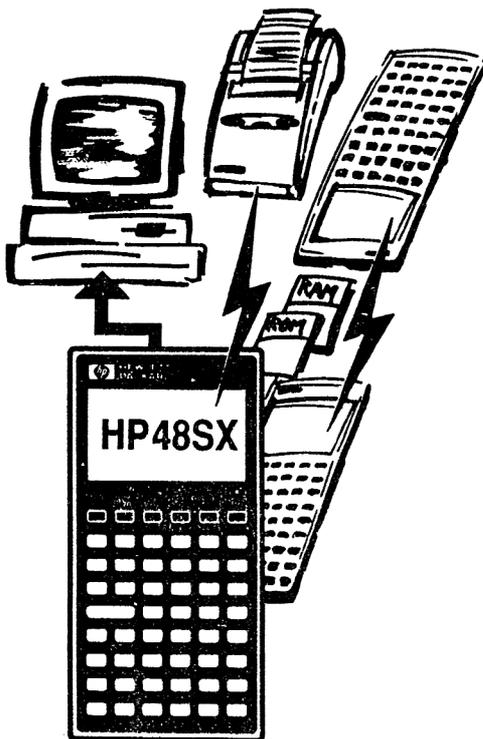
[EPON]
1 [MAG]
{BLUE} {PRINT}

Set the graphic parameters for Epson
Set the graphic magnitude to 1
Prints the graphic

If you want a larger graphic, set a magnitude of 4 in the EPSRPT program.

4 [MAG] {BLUE} {PRINT}

Prints a larger graphic



HP 48SX-TOOL

TESTBERICHT



The HP 48
File Manager

Robert Hübnex

Überblick

Ein Problem des HP 48SX ist die schlechte Qualität des Schnittstellenpakets: "HP Serial Interface Kit # 8220-8A/9A".

Dieses auf dem Public-Domain Programm KERMIT basierende Übertragungssystem besteht aus einem Schnittstellenkabel zur seriellen Datenübertragung, dem PC-Übersetzungsprogramm KERMIT, und ein paar Programmen zum Drucken, Grafikkonvertieren und HP 28 Druckdaten einlesen

Erklärung:

Public Domain : Software welche kostenfrei kopiert und weitergegeben werden kann. Der Verteiler darf nur Kopierunkosten geltend machen.

KERMIT : Programm zur Datenübertragung zwischen verschiedenen Computeranlagen.

Dieses Kermit ist hauptsächlich zur Zeichenkonvertierung entwickelt worden. Es bietet eine reine Text-Bildschirmarstellung und kommt ohne Mausbedienung aus, da viele Großcomputer über keine Grafikschnittstelle verfügen. Kermit läßt sich fernsteuern und hat auch eine Fernsteuerfunktion.

Unverständlich ist, warum HP kein eigenes, auf den HP 48 angepaßtes PC-Übertragungsprogramm auf den Markt bringt.

Ein Amerikaner namens James Donnelly hat sich dieser Problematik angenommen und ein HP 48-Tool zur besseren Daten- bzw. Programmübertragung zwischen IBM-PC und HP 48 SX entwickelt.

Den "HP 48 File Manager"

Programmbeschreibung

Das Programm wird auf einer 5,25" Diskette im Format 360 kB und mit Handbuch ausgeliefert.

Der File Manager stellt ein paar wichtige Funktionen zum schnellen Übertragen zwischen PC und HP zur Verfügung.

Dazu muß KERMIT auf dem PC gestartet und laut Anleitung konfiguriert werden. Dann wird das Programm FMLIB (File-Manager-Library) zum HP 48 übertragen und installiert.

Nun wird KERMIT als ferngesteuert (remote host) eingestellt und der File Manager auf dem HP 48 gestartet.

Ist dies alles nach Plan verlaufen, werden alle Dateien (Daten und Programme) auf dem Personal Computer am HP 48 grafisch angezeigt und können vom HP 48 aus manipuliert werden. Das ist auch mit den im HP gespeicherten Dateien möglich

Die Manipulationsmöglichkeiten sind:

- Senden von Dateien
- Verzeichnisse erstellen
- Dateiinfos ausgeben
- Verzeichnisse anzeigen
- Backups erstellen
- Den freien Speicherplatz ermitteln
- Dateien umbenennen

Die Diskette enthält folgende Programme:

- | | |
|-------------------------------------|---------------------------|
| FMLIB | Library |
| File Manager | Programm |
| README | Text |
| Informationen über den File Manager | |
| KERMIT | Programm und Hilfe |
| KERMIT.PIF | MS-Windows PIF-Datei |
| Kermit | Startdatei für MS-Windows |

Gesamtwertung Programm

- | | |
|------------------|---------------------------------------|
| Auslieferung: | 1x5,25" 360 kB
2 Programme, Hilfen |
| Speicherbedarf: | ca 10 kB |
| Installation: | Sehr gut |
| Schnittstelle: | (Kermit), Befriedigend |
| Fähigkeiten: | Gut |
| Preiswürdigkeit: | Sehr gut |

Handbuch

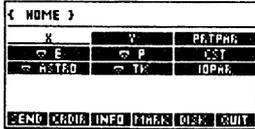
- | | |
|----------------|------------------------|
| Auslieferung: | 1xPaperback, 16 Seiten |
| Sprache: | Englisch |
| Dokumentation: | Sehr gut |
| Kaufpreis: | 69,- |

The HP 48 File Manager

Includes Disk for IBM Compatible Computers

The HP 48 File Manager has been designed to simplify the display of your data within the HP 48, and facilitate data transfer to or from an IBM compatible computer.

When File Manager begins, the current directory in the HP 48 is displayed first.



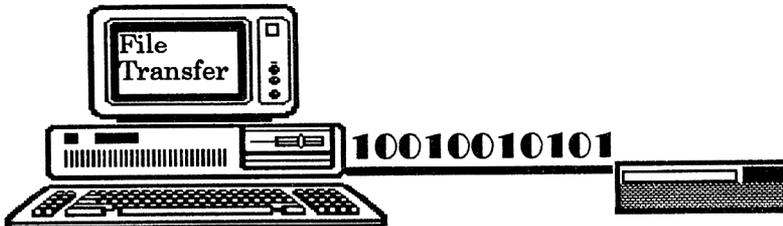
69,-DM
(incl. 7% MwSt)

The path to the current directory is shown at the top of the display. The objects stored in that directory are shown in the middle portion of the screen. Names shown with a prefix character (▽) indicate a directory.

The HP 48 arrow keys (▼▲◀▶) move the highlight from one object to another. The keystrokes [ENTER], [↑], and [HOME] traverse the directory structure of either the HP 48 or the server disk. Menu keys may be used to show the directory of the Kermit server's disk, transfer files, purge or rename files, create directories, show the available free space, or archive the memory of the HP 48.

The HP 48 File Manager is an essential tool for anyone who routinely exchanges files with a personal computer.

The HP 48 File Manager



HP 48SX-TOOL

TESTBERICHT



The HP 48
Programmer's Toolkit

Robert Hübnert

Überblick

Wer kennt nicht das leidige Problem eines Programmierers, in einem selbstgestrickten Programm immer wieder auf die "langsamen" HP 48-Basisfunktionen zurückgreifen zu müssen.

Eine Lösung wäre, den HP 48 SX auf der System-Ebene zu programmieren (Sysevals) oder das Programm in Maschinencode direkt zu programmieren.

Doch leider stellt HP die dafür benötigte Befehlsreferenz dem Endkunden nicht zur Verfügung.

So bleibt dem normalen Programmierer nur die Möglichkeit einzelne Programmmodule (= Sammlung aus Basisfunktionen) für Standardfunktionen zusammenzustellen, die in alle eigenen Programme eingebaut werden können. Die Basisfunktionen des Moduls werden dann einzeln oder in kleinen Gruppen gegen gleichwertige Basisfunktionen ausgetauscht und immer wieder auf Ihr Laufzeitverhalten überprüft.

Auf diese Weise bekommt man mit der Zeit eine Sammlung von geschwindigkeitsoptimierten, leistungsfähigen Programmmodulen zum schnellen Aufbau von Programmen.

Nun gibt es aber einen amerikanischen Anbieter, der fertige leistungsoptimierte Programmier-Tools im Maschinencode auf den Markt bringt.

Das "HP 48 Programmer's Toolkit"

Programmbeschreibung

Das HP 48 Programmer's Toolkit stellt eine Sammlung nützlicher Hilfsprogramme für die Programmentwicklung auf dem HP 48 zur Verfügung.

Das Programm wird auf einer 5,25" Diskette im Format 360 kB ausgeliefert.

Zur Datenübertragung von PC auf HP 48 wird allerdings das etwas veraltete HP Schnittstellenpaket (Kermit) benötigt.

Ein Handbuch gibt Auskunft über alles Wesentliche.

Die Diskette enthält folgende Programme:

CSCAT System-Programm

Stringkatalog : Tabellarische Übersicht über alle verfügbaren HP 48 Zeichen, mit dem Zeichencode in HEX, DEC, OCT und BIN.

Note: Gut

L.BLD System-Programm

Softkey-Designer : Malprogramm zum Zeichnen von Softkey-Symbolen.

Note: Sehr gut

FCLIB Library

Flag-Übersicht : Tabellarische Übersicht über alle verfügbaren Flags und ihren momentanen Status. Alle Flags lassen sich manipulieren.

Note: Sehr gut

TLLIB Library

74 neue Kommandos : Array, Graphics, Meta, Set, Stack, String u. a. Operationen.

Note: Gut

TBLIB Library

Titel-Tabelle : Tabellarische Auswahldarstellung von Programmfunktionen.

Note: Gut

DBLIB Library

Datensicht-Programm : Programm zum Definieren und Ändern von Datensätzen (Minidatenbank).

Note: Gut

Gesamtwertung

Programm
Auslieferung: 1 x 5,25" 360 kB
6 Programme, Beispiele
Speicherbedarf: 27661 Byte
Installation: Befriedigend
Schnittstelle: (Kermit), Befriedigend
Fähigkeiten: Gut
Preiswürdigkeit: Gut

Handbuch
Auslieferung: 1xRingbuch, 124 Seiten
Sprache: Englisch
Dokumentation: Gut
Kaufpreis: 149,-

The HP 48 Programmer's Toolkit

149,-DM (incl. 7% MwSt.)

The HP 48 Programmer's Toolkit

Getting Started	1
Character Set Catalog	5
Menu Label Builder	7
Flag Catalog	10
Data Browser	13
Title Browser	22
Tool Library	26
Command Reference	40
Object Types	116
Character Codes	117
Character Translations	119
Flags	120
Alpha Keyboard	124

The HP 48 Programmer's Toolkit

Includes Disk for IBM-Compatible Computers

The HP 48 Programmer's Toolkit is a collection of software tools designed with the programmer in mind. These tools improve program performance by combining some common, slow operations into faster internal system languages and provide additional capability in object manipulation not directly available in the HP 48.

- The Character Set Catalog is an interactive catalog that displays the HP 48 character set in all three available fonts and shows the I/O translation forms.
- The Menu Label Builder is an interactive program for building graphics objects for use in custom menus.
- The Flag Catalog is an interactive program that shows the states and descriptions of system and user flags.
- The Data Browser and Title Browser are two powerful screen-oriented user interface utilities that may be used to enhance an application:



- The Tool Library provides 74 new commands for arrays, graphics, list utilities, meta-objects, sorting, sets, stack utilities, string utilities, and more. These commands are fast and powerful. For instance, the command MREVERSE will reverse the order of 1000 objects on the stack in less than a second, and ROTATE will rotate character strings, lists, or graphics objects.

The 124 page manual provides complete descriptions of all the new catalogs, programs, and commands with examples. Additional chapters provide reference tables for object types, the character set, and flags.

Casino 48



CONTENTS

Part 1. Casino 48 Page 2

Part 2. Bank Page 4

Part 3. Blackjack..... Page 5

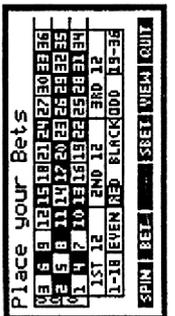
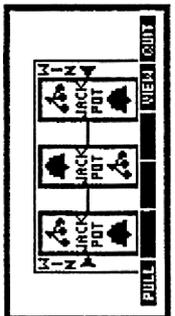
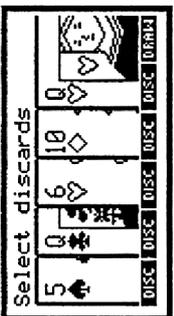
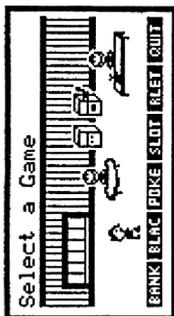
Part 4. Video Poker..... Page 9

Part 5. Slot Machine Page 12

Part 6. Roulette Page 14

Part 7. Additional Features..... Page 16

Part 8. Errors and Problems Page 18



Casino 48 is a fun-filled simulation of a casino for the Hewlett-Packard 48SX calculator. Casino 48 contains four games: Blackjack, Video Poker, Slots and Roulette. Use Casino 48 to learn to play, to improve your skills and to have fun! In the casino you walk to the table of your choice and start a new game.

Blackjack
 Over at the blackjack table you can hit, stay, double down, split pairs or get a recommendation. The recommendation key tells you your best option and the rule you need to remember. For example, if you have 18 and the dealer shows a ten then the recommendation key would say, "Stay on 17 or more when the dealer shows 7, 8, 9, 10 or Ace".

Video Poker
 At the Video Poker machine you select discards and then draw. A pair of aces or better wins!

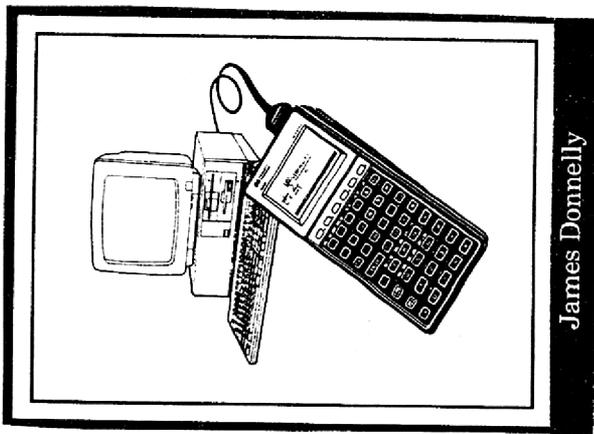
Slot Machine
 Pull the handle of the slot machine and watch the wheels spin of big payoffs.

Roulette
 In Roulette you place many bets, and then spin the wheel.
Also - You can create your own card game with commands for drawing cards on the screen.

198-DM
(incl. 7% Mwst)

All the fun and excitement of a casino for the Hewlett-Packard 48SX. Created by Preston Brown Labs.

The HP 48 Handbook



**The
HP 48 Handbook**
James Donnelly

Introduction	1
Objects, Names, and Constants	3
Memory Organization	15
Graphics	21
Statistics Data	27
Data Transfer	29
Menus	40
User Keys	43
Programming	45
HP Solve Equation Library	64
System Operations	96
Printer Control	101
Built-in Units	103
Messages	110
Menu Numbers	117
Character Codes	118
Object Types	120
Flags	121
Subject Index	125
Command Reference	139
Alpha Keyboard	195



Bücher
aus USA

69,-DM (incl. 7% MwSt)

Bestellungen bitte an:

Fischel GmbH, Kaiser-Friedrich-Str. 54a,

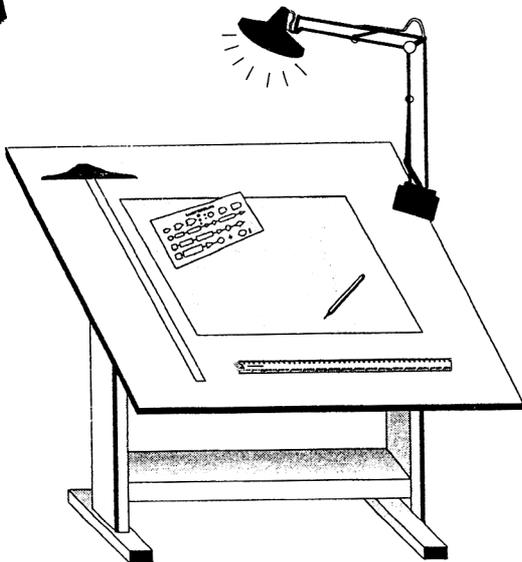
1000 Berlin 12, Fax: 030/ 324 09 28

BÜCHER AUS USA

HP 48 Graphics

The book begins with a good in-depth review of the EquationWriter, the SOLVE and the PLOT applications. Next, it guides you through the locations and uses of the 48's built-in graphic commands. Then you learn to build your own grobs and combine them into some extensive application programs. There's even a chapter that discusses transferring your HP 48 Graphics to other computers or printers.

So don't miss this insightful—and fun—excursion into the world of HP 48 Graphics. It adds a whole new dimension to your use of this powerful machine.



By R. Ray Depew

A GRAPEVINE PUBLICATION

69,-DM
(incl. 7% MwSt)

HP 48 Graphics

Here's a "must-have" if you want to use the full potential of that big display on your HP 48. Written by HP engineer Ray Depew, this book shows you how to build graphics objects ("grobs") and then use them to customize displays with diagrams, pictures and plots.

The book even starts with a great in-depth review of the built-in graphics tools of the calculator—the PLOT menu and the SOLVER—and how to use them *well*.

Then you'll "walk through" the process of building your own grobs and combine them to produce some very impressive graphics—right there on your own HP 48. Don't miss it!

CONTENTS	
1: Introduction.....	8
What This Book Is About.....	9
Plotting a Simple Function.....	10
Solving Within the Plotter.....	13
Freehand Graphics.....	14
Grobbing Around.....	16
What Next?.....	18
Notes on Using This Book.....	20
2: The EquationWriter.....	24
Preparations.....	25
Opening Remarks.....	25
So What Does It Do?.....	27
Examples.....	28
Using the EquationWriter.....	30
The Selection Environment.....	32
A Fourier Series Example.....	34
Test Your Skill.....	35
Other Things.....	38
Closing Remarks.....	39
3: The Solver.....	40
Opening Remarks.....	41
Preparations.....	41
Apples and Oranges.....	42
The Ideal Gas Law.....	45
The Time Value of Money.....	48
A Third-Degree Polynomial.....	50
Customizing the Solver.....	52
Linking Equations: Solving Several at Once.....	59
Using the Solver on Ill-Mannered Functions.....	64
Using the Solver Inside the Plotter.....	70
Programmable Use of the Solver.....	79
Review.....	81
4: What's a Grob?.....	82
Opening Remarks.....	83
A Clean Slate.....	83
What Is a Grob?.....	84
Pixel Numbers vs. User Units.....	86
"Roll Your Own" Grobs.....	89
The Hexadecimal Bitmap.....	90
The SEE Program.....	92
What Does a Grob Eat?.....	93
The Grob as Icon.....	95
Review.....	97

5: Graphics Basics98	8. Programmable Graphics Applications146
The Graphics Functions99	Introduction147
The PVIEW Bug101	Programmable Scanning Inside a Big Grob.....148
The Secrets of PPAR.....103	Generating a Stripchart164
The PLOTR Menu.....105	An Analog Voltmeter.....176
The PRG-DSFL Menu110	Plots with Two Independent Variables182
Other Graphics Commands115	A Contour-Plotting Program188
Building a Toolbox.....116	Driving a Bulldozer Around the Display.....196
Sines and Big Sines118	A Friendly Game of Checkers.....200
Review120	A Calendar Demo.....230
	More Suggestions236
6: Graphics Improvements122	
Opening Remarks123	9. Graphics Beyond the 48238
Labelling the Axes124	Printing Graphics on the Infrared Printer.....239
Connecting the Dots125	Printing Graphics on a Larger Printer240
Adding Text to Graphics126	Printing Graphics on a Pen Plotter.....258
Adding Graphics to Enhance Plot..134	Grobs and Other Computers259
Review136	Graphics Between Two 48's261
	Final Thoughts262
7: Freehand Drawing138	
How to Do It139	Appendices264
Drawing a Voltmeter Face141	A. Review of the Hexadecimal Number System...265
Review145	B. Graphics Commands and Operations272
	C. User-Named Objects282
	Index290
	Other Books and Information298

HP 48 Graphics

Here's a fascinating look at the potential of that big display on your HP 48. HP engineer Ray Depew shows you how to build graphics objects ("grobs") and use them to customize displays with diagrams, pictures, labels, titles, multiple plots, games, and menu icons.

Here's the fastest, easiest way to get up-to-speed on the incredible HP-48SX! This fascinating and friendly Course gives you jargon-free, hands-on, practical lessons on **objects, tools, menus, the Stack, writing, solving and plotting equations, using matrices and statistics, programming, using directories—and much more.**

Each lesson shows you working examples of the commands and concepts you need to learn. There are plenty of review points that summarize what you've learned—and a quiz at the end of each chapter.

So don't wait any longer to start tapping the potential of your HP-48SX. Let this clear, concise Easy Course get you started on a long and productive working relationship with HP's finest calculator. It's always a pleasant surprise when the right kind of instruction can transform a "mysterious" and powerful machine into a friendly and familiar tool—for you!



Bestellungen bitte an: Fischel GmbH, Kaiser-Friedrich-Str. 54a,
1000 Berlin 12, Fax: 030/ 324 09 28

An Easy Course in Using the HP-48SX

Here's the fastest, easiest way to get up-to-speed on the incredible HP 48SX and HP 48S! The first half of the *Easy Course* gives you a tour of the keyboard, the stack, menus, etc, and explains how to build and use each of the different kinds of *objects* in the HP 48:

- Real numbers
- Vectors
- Binary integers
- Names
- Programs
- Complex numbers
- Arrays (matrices)
- Characters and strings
- Algebraic objects
- Directories
- Units
- Flags
- Tags
- Lists

98,-DM
(incl. 7% MwSt)

The second half of the *Easy Course* teaches you how to use the various menus of hand tools and power tools to build and combine those objects. Here are some of the topics covered:

- Math
- Statistics
- Building equations
- Customizing menus
- Algebra
- Plotting
- Time calculations
- Assigning keys
- Calculus
- SOLVing equations
- Postfix programs
- Directory structures

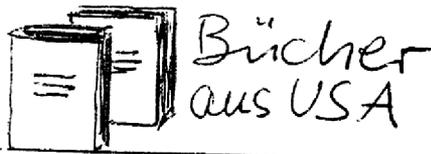
Each lesson shows you working examples of the commands and concepts you need to learn, and there are plenty of review points that summarize what you've learned. With Grapevine's *Easy Course*, you learn at your **own speed**—and soon you're calculating like a pro!

HP 48 Programming/Utilities

This book is the perfect combination of instructive examples and easy-reference, "pre-fab" program code—utility routines that you'll commonly need. After all, why reinvent the wheel over and over again when expert John Loux has done it once and for all? You'll see useful advice and routines for:

- Directory manipulation
- General object manipulation
- Program development
- Sorting/Searching
- Display formatting
- and much more

Whether or not you're experienced with the HP 48, you'll find this book to be a great collection of advice, good habits and sound programming principles and discipline that you'll appreciate.



CONTENTS

0 START HERE 8

1 YOUR 48 WORKSHOP 12

 Calculating with Tools and Objects 13

 The Big Picture: A Workshop 14

 The Display: Your Window into the Workshop 16

 The Keyboard: Access to Your Workshop 18

 The Tools in Your Workshop 21

 The Raw Materials in Your Workshop 22

 Quiz on the "Big Picture" 27

 Quiz Answers 28

**2 THE STACK AND COMMAND LINE:
YOUR WORKBENCH 30**

Typing and the Command Line 31

Simple Materials: Real Numbers 40

Postfix Notation 48

Stack Manipulations 52

Learning By Doing 59

Workbench Quiz 60

 Workbench Solutions 62

More on the HP 48SX and HP 48S

3 OBJECTS: YOUR RAW MATERIALS 66

The Fundamental Idea 67

Real Numbers 67

Units 68

Lists 74

Complex Numbers 80

Vectors 86

Arrays 92

Flags 98

Binary Integers 102

Character Strings 108

Tags 112

Names 116

Algebraic Objects 124

Postfix Programs 132

Directories 136

Objects: A Summary 142

Test Your Objectivity 143

 Objective Answers 150

4 FUNCTIONS AND EXPRESSIONS 164

Functions and Arguments 165

Some Built-In Functions 166

Symbolic Functions and Variables 178

Creating Expressions 180

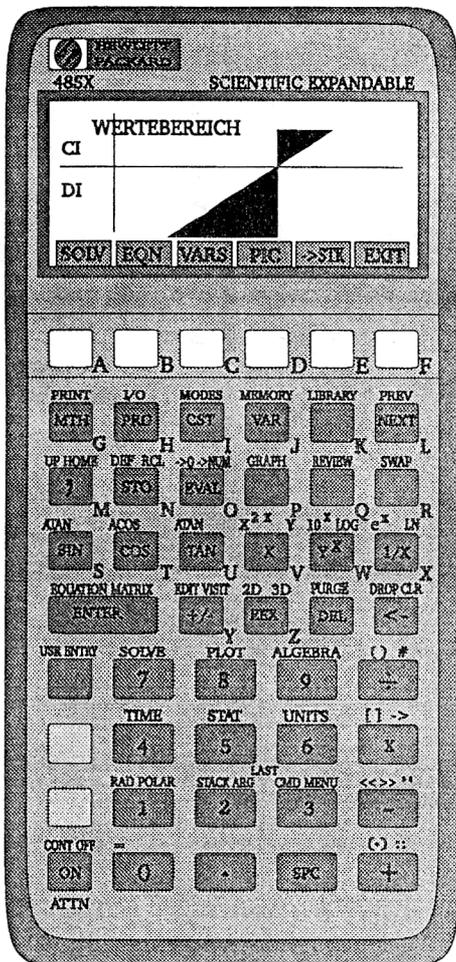
 Editing Expressions 185

 Saving Expressions 195

Using Expressions	197	6 BUILDING YOUR OWN TOOLS:	306
Evaluating Expressions	198	PROGRAMMING	
Rearranging Expressions	200	Your "Automation" Options	307
Solving Equations of Expressions	208	Local Names	310
User-Defined Functions	212	Program Design	318
Math Anxiety	216	Conditional Tests	323
Cool and Calculating	218	Branching	326
		Looping	333
		Quiz	340
		Quiz Answers	342
5 SOLVING, PLOTTING AND ANALYZING	222	7 CUSTOMIZING YOUR WORKSHOP	348
Equations, Data and Graphics	223	Labor-Saving Devices	349
Defining EQ, the Current Equation	226	Input Shortcuts	350
The SOLVR Menu	230	The LAST Commands	355
Solving Equations with SOLVE	232	Customizing Your Workspace	358
Solving Equations Involving Units	235	Directory Structure	359
Solving Equations Using PLOT	238	Custom Menus	362
Solving Two Expressions Simultaneously	245	Custom Keyboards	368
Solving Programs and User-Defined Functions	248	Custom Flag Settings	374
Multiple Equations with SOLVE and PLOT	252	Customizing the Built-In Tools	376
Solving Systems of Equations	258	Optimization: A Case Study	379
Analyzing Data: The STAT Tool	262	Custom Questions	383
Creating the Data Matrix	264	Optimum Answers	384
The STAT Menu	267		
Single-Variable Statistics	268	FOUNDATION COMPLETED	387
Two-Variable Statistics	270		
Two-Sample Statistical Tests	276		
Transforming Variables in the Data Matrix	279		
More Challenges	281		
More Solutions	289		

Hewlett Packard HP-48SX Pocket Computer

zum
Programmierhandbuch



ISBN 3-89374-065-1
Fischel GmbH

Robert Hübner

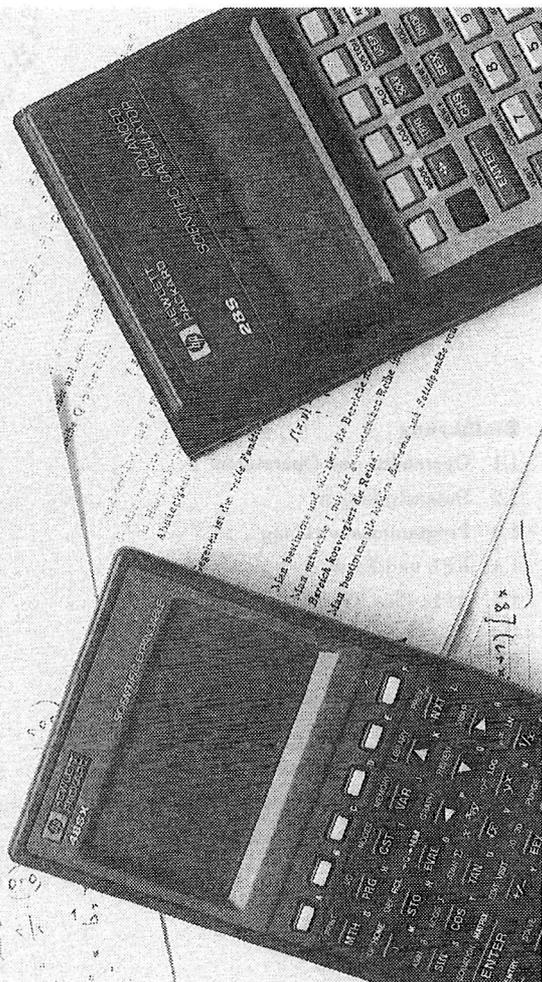
**Robert Hübner: Programmierhandbuch zum Hewlett Packard
HP-48SX Pocket Computer
ISBN 3-89374-065-1, Preis: 49,- DM (inkl. 7% MwSt.)**

Inhalt:

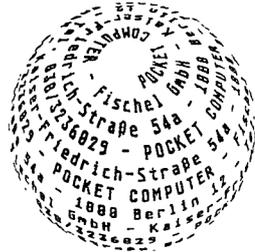
VORWORT	001
EINLEITUNG	003
KAPITEL 1	
Der HP-48SX im Überblick	007
Rechner Grundeinstellungen	007
Testen des Betriebssystems	007
Die Eingabemodi des HP-48SX	010
Tips und Übungen für Anfänger und Fortgeschrittene	015
Die wichtigsten Module des HP-48SX	015
Der terminalkalender	015
Der MatrixWriter	018
Der EquationWriter	022
Grafik auf dem HP-48SX	025
Benutzeranpassung des HP-48 Taschencomputers	028
Benutzermenüs erstellen	028
Temporäre Benutzermenüs	032
Die benutzerdefinierte	032
Benutzerfunktionen	036
Funktionsanalyse	038
Plottypen	045
Darstellung von Funktionen mit Angabe ihres Wertebereiches	046
Darstellung statistischer Daten	051
Balkendiagramme	052
Die Speicherorganisation des HP-48SX	053
Verwenden von ROM-Erweiterungskarten	057
Kapitel 2	
Die Programmierung des HP-48SX	059
Der Aufbau von Programmen	059
Grundregeln der Programmierung	059
Programmablauf	061
Übergabe von Objekten an Programme	061
Variablenamen	063
Programmdialoge unter Verwendung von Strings	064
Der INPUT-Befehl	068
Aufbereitung der Ausgabedaten	070
Temporäre Menüs zum Eingeben von Daten und Starten von Unterprogrammen	071
Programmschleifen und bedingte Verzweigungen	072
Symbolische Argumente in Programmen	077
Unterprogramme	079
Programmprozeduren	080
Lokale Variablen	081
Der richtige Aufbau eines Programms	082

Kapitel 3		
HP-48SX Programmbibliothek		085
Bibliothek Inhaltsverzeichnis		086
Mathematik-Bibliothek		086
SQRI	n-te Wurzel aus reeller und imaginärer Zahl	087
HORS	Horner-Schema für Reell, imaginär und Parameter	088
GAUSS	Gauss-Jordan-Form einer Gleichung (s. symb.)	091
DIV	Divergenz eines Vektorfeldes (auch symbolisch)	092
root	Automatische Nullstellensuche	093
Fourier	Fourier-Ordnung	094
Utilitäts Bibliothek		095
Repeat	Definierte Wiederholung von Programmen	095
ZFASS	Vollständiges Zusammenfassung	095
Oraphtik		100
Demo	Oraphtikdemo	100
Julia-Menge	Fraktale Grafik 1	101
Martin-Menge	Fraktale Grafik 2	102
Coast-Menge	Fraktale Grafik 3	104
Kapitel 4		
Ergänzungen		105
Der HP-Debugger		105
Fehlererkennung und Fehlerbehandlung		106
Menüorganisation		108
Die wichtigsten Zusatzmodule im HP-48SX Schnittstellenpaket		110
Das Dateiübertragungsprogramm KERMIT		110
Das Graphikübersetzungsprogramm GOBZTIF		115
Das HP-Druckbefehl-Übertragungsprogramm (z.B. für HP 28-Druck)		117
Weitere Schnittstellenprogramme		117
Der Befehl WSLOG	Undokumentierter HP-Befehl !!	118
Kapitel 5		
Tabellen und Listen , Literaturhinweis		121
		SEITE
	Flags	122
	Menüs	123
	Meldungen numerisch	124
	Meldungen alphabetisch	129
	Kermit-Befehle	136
	Einheiten alphabetisch	137
	Einheiten nach Menüs	144
	Literaturverzeichnis	151

**D. Ciesinger/ S. Ebeling: Programmsammlung zur
höheren Mathematik für die HP-28 und HP-48
Taschencomputer, ISBN 3-89374-078-3,
Preis: 49,- DM**



**Programmsammlung zur
höheren Mathematik für die
Hewlett Packard HP-28 und HP-48
Taschencomputer**



ISBN 3-89374-078-3
Fischel GmbH

D.Ciesinger/ S.Ebeling

Inhaltsverzeichnis

1 Einführung	1
1.1 Operanden und Operatoren	1
1.2 Datenstrukturen	2
1.3 Programmentwicklung	3
1.4 RPL und Forth	4
1.5 RPL: Eine Kritik	7
2 Konventionen	9
2.1 Unterschiede zwischen HP-28 und HP-48	9
2.2 Dokumentation	13
2.2.1 Stackdiagramme	15
2.2.2 Algorithmisches Deutsch	16
2.3 Verzeichnisbaum (Directories)	17
3 Hilfsprogramme	19
3.1 Verzeichniswechsel	19
3.2 Stackoperationen	21
3.3 ASSOCIATIVE und CASE	21
3.4 Ein-/Ausgabe	23
3.5 Debugger und Profiler	26
3.6 Sortieren	29

3.7	Operationen auf Listen	33
4	Lineare Algebra	39
4.1	Vereinbarungen	39
4.1.1	Typdeklaration	39
4.1.2	Beispiel	40
4.1.3	Hintergrund	40
4.2	Hilfsprogramme	40
4.3	Vektorrechnung	44
4.4	Matrizenrechnung	47
4.5	Determinante	52
4.6	Eigenwertberechnung	55
4.7	Inverse	59
4.8	Gauß-Jordan-Verfahren	60
5	Differentiation	67
5.1	Gradient	67
5.2	Rotation	68
5.3	Divergenz	69
5.4	Jacobi-Matrix	70
5.5	Hesse-Matrix	70
5.6	Wronski-Determinante	72
5.7	Anfangswertprobleme	74
6	Polynome	85
6.1	Vereinbarungen	85
6.1.1	Typdeklaration	85
6.1.2	Beispiel	85
6.1.3	Hintergrund	86
6.2	Hilfsprogramme	86
6.3	Hornerschema	87
6.4	Nullstellen	90
6.5	Multiplikation	96
6.6	Division	97
7	Graphische Anwendungen	99
7.1	Abschnittsweise definierte Funktionen	99
7.2	Newton-Interpolation	100
7.3	Splines	103
7.4	Diskrete Gaußsche Fehlerquadratur	107
7.5	FFT (Fast Fourier Transform)	110
8	Höhere Magie	119

Sharp PC -1402/02/03/21/50/75

PC-1100/50/1245/46S/48/60/61/80

- PC-1403 Systemhandbuch 39,-
- PC-1403 Anwendungshandbuch 49,-
- PC-1403 Maschinensprachehandbuch 49,-
- Die besten Programme für den PC-1403 49,-
- PC-1401/02/21/03 Tips + Tricks Programmierhandbuch 49,-
- PC-1401/02/21 Maschinenspracheprogrammiersammlung 49,-
- PC-1450 Anwendungshandbuch 49,-
- PC-1100 Anwendungshandbuch 39,-
- PC-1150/1245S/48/70 Anwendungshandbuch 49,-
- PC-1280/81 Maschinensprachehandbuch 49,-
- PC-1280/1475 Anwendungshandbuch 49,-
- PC-1280/1475 Systemhandbuch 49,-
- PC-1280/1475 Maschinensprachehandbuch 49,-
- PC-1280/1475 Tips + Tricks Programmierhandbuch 49,-

Der Sharp in Deiner Hand

Sharp PC-E220/500

- PC-E220 Anwendungshandbuch 48,-
- PC-E220 Basic Programmierhandbuch 48,-
- PC-E220 Maschinensprachehandbuch 48,-
- PC-E500 Systemhandbuch 48,-
- PC-E500 Maschinensprachehandbuch 48,-
- PC-E500 Anwendungshandbuch 48,-
- PC-E500 Tips + Tricks Programmierhandbuch 48,-
- Die besten Programme für den PC-E500 48,-
- PC-E500 Basic Programmierhandbuch 48,-
- PC-E500/QC-7000 Hardwarehandbuch 48,-

Z88

- 288-Cambridge Anwendungshandbuch 48,-
- Rechnerkopplung mit Sharp u. Casio Pocket Computer 48,-
- Umsetzungshandbuch für Sharp Taschencomputer 48,-
- Hardware-Entwicklung für Sharp Pocket Computer 48,-
- Entw. intelligenter Hardware mit Sharp Taschencomputer 48,-
- Maschinensprachehandbuch für Sharp Taschencomputer 48,-
- Einf. in die Maschinensprache für Sharp Taschencomputer 48,-
- Datenerfassungshandbuch f. Sharp Taschencomputer 48,-
- Datenerfassungshandbuch für Sharp Taschencomputer 48,-
- CAD- und Grafikprog. für Sharp Taschencomputer 48,-
- Hacker-Handbuch für Sharp Computer 48,-
- Computerlexikon u. Recordhandbuch für Sharp Comp. 48,-
- Schönschrift und Textverarbeitung für Sharp Computer 48,-
- Lehr- und Übungshandbuch für Sharp Taschencomputer 48,-
- Basic Lehrbuch für Sharp Computer 48,-
- Grundhandbuch für Sharp Computer 48,-

Wirtschaftswissenschaften

- Hauswirtschaft mit Pocket Computern 48,-
- Kaufmännische Progr. für Sharp Taschencomputer 48,-
- Betriebswirtschaftl. mit Sharp Taschencomputern 48,-

Personal Computer

- Sharp MZ-700/800 Maschinensprachehandbuch 49,-
- PC-FAX Telefaxen mit dem Personal Computer 49,-
- BTX mit Personal Computer 48,-
- EDV im Buchhandel 150,-
- EDV für Juristen 98,-
- EDV für Verkäufer 98,-
- EDV für Vereine 98,-

Gesamtpreis DM:

Datum, Unterschrift:

Verbindlichkeiten: Vorauszase: 4,- DM,
Nachnahme (nicht ins Ausland): 6,50 DM

Verrechnungsscheck über DM _____ (inkl. 4,- DM Versandkosten) liegt bei.
Probierfrist der Zeitschrift "Pocket + Laptop Computer" gegen Einzahlung von DM 6,- in Briefmarken (Ausland DM 7,- Eurocheck oder internationale Postwertzeichen).

Allgemeiner Deutscher Computer Club

**Pocket + Laptop Computer
Fischel GmbH**

Kaiser-Friedrich-Str. 54a
1000 Berlin 12

Tel.: 030/ 323 60 29

Fax: 030/ 324 09 28

Mo-Fr 10-18.00, Sa 10-14.00



ADCC

An alle Auslandskunden:

Wenn Sie bei uns bestellen, so fügen Sie bitte einen Vorauscheck bei (Eurocheck inkl. Versandkosten). Das gilt vor allem, wenn Sie in Österreich oder der Schweiz wohnen. Es werden keine Nachnahme- sendungen ins Ausland geschickt!

Bankverbindung:

Postgiroamt Berlin West,

BLZ 100 100 10

Konto-Nr.: 461 533-103

Lieferanschrift:

POCKET + LAPTOP COMPUTER

Zeitschrift für mobile Daten Systeme

Abonnement:

Wenn Sie den regelmäßigen Bezug dieser Zeitschrift per Post wünschen, senden Sie den Bestellschein noch heute ab.

Ja, ich möchte Abonnent und Club - Mitglied im ADCC werden!

- O Sie werden automatisch Mitglied des ADCC (Allgemeiner Deutscher Computer Club)
- O Ich abonniere die Zeitschrift "Pocket + Laptop Computer" von der Ausgabe _____ an (Preis für 12 Hefte: Inland DM 72,-, Ausland DM 84,-, Luftpostzuschlag für Übersee nach Kosten und Aufwand).
- O Das Abonnement verlängert sich um 12 Hefte zu den dann jeweils gültigen Bedingungen, wenn es nicht zwei Monate vor Ablauf schriftlich gekündigt wird.

Der Gesamtbetrag von _____ DM

- O liegt bar bei
 - O liegt als Verrechnungsscheck bei (schnellste Erledigung)
 - O wurde am _____ auf das Postgirokonto der Fischel GmbH, Kto.-Nr.: 461533-103, BLZ 100 100 10, Postgiroamt Berlin überwiesen. (Bearbeitung nach Zahlungseingang)
 - O liegt (nur kleinere Beträge) in Briefmarken oder internationalen Antwortscheinen bei.
- (Alle Preise incl. 7% MwSt.)

Name, Vorname: _____

Straße, Nr.: _____

Plz, Ort: _____

1. Datum, Unterschrift: _____

Mir ist bekannt, daß ich diese Bestellung innerhalb von 8 Tagen bei der Bestelladresse widerrufen kann. Zur Wahrung der Frist genügt die rechtzeitige Absendung (Datum des Poststempels). Ich bestätige dies durch meine zweite Unterschrift:

2. Datum, Unterschrift: _____

Bitte einsenden an:
Fischel GmbH
 Kaiser-Friedrich-Str. 54a
 W - 1000 Berlin 12
 (Fax: 030/ 324 09 28)

Abonnieren Sie jetzt! Es lohnt sich!

Sichern Sie sich jetzt Ihr "Pocket + Laptop Computer" Abonnement! Wenn Sie zusätzlich ein Buch (siehe Super - Bestellschein) bestellen, erhalten Sie aus unserem Sortiment ein Buch Ihrer Wahl gratis dazu!

$$1 + 1 = 3 !!$$

Neuabonnenten:

Dieses Angebot gilt für Neuabonnenten nur, wenn Sie den nebenstehenden Abo - Bestellschein verwenden (linke Spalte). Senden Sie ihn zusammen mit dem ausgefüllten Super - Bestellschein an:

Fischel GmbH
 Kaiser-Friedrich-Str. 54a
 W-1000 Berlin 12.

Sie abonnieren + bestellen ein Buch = Sie erhalten ein Buch Ihrer Wahl gratis dazu!

Ja, ich mache von Ihrem Angebot Gebrauch und abonniere die Zeitschrift "Pocket + Laptop Computer". Der ausgefüllte Abo - Bestellschein liegt bei.

Abonnement - Verlängerung:

Selbstverständlich gilt dieses Angebot auch für Leser, die Ihr Abonnement verlängern wollen, aber nur, wenn die Verlängerung mit diesem Bestellschein erfolgt:

Name, Vorname: _____

Straße, Nr.: _____

Plz, Ort: _____

Ich habe folgenden RechnerTyp: _____

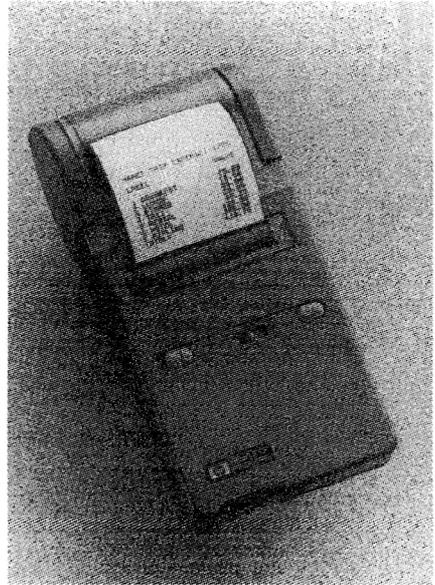
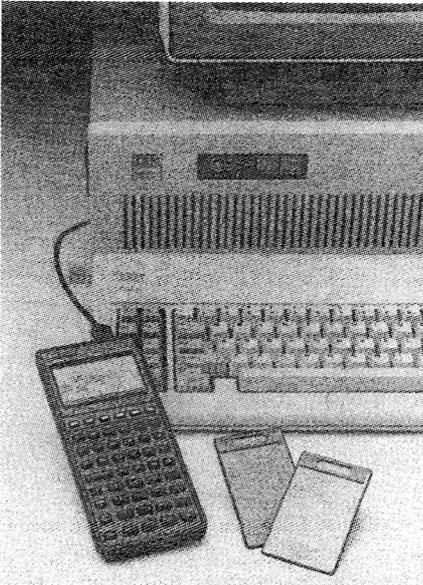
Den ausgefüllten Super - Bestellschein für Bücher habe ich beigelegt. Als Gratis - Buch schicken Sie mir bitte:

Ich habe einen Scheck über DM _____ beigelegt:

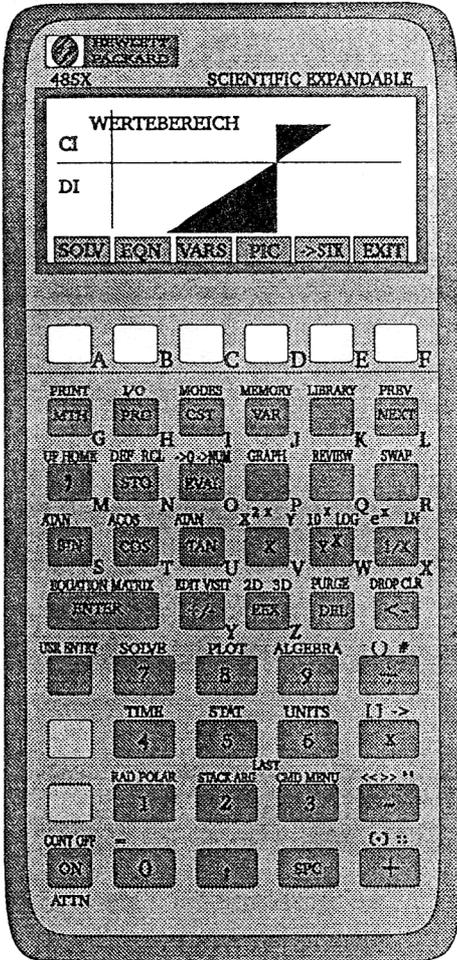
Ich habe den Betrag von DM _____ auf Ihr Konto Nr.: 461533-103, Postgiroamt Berlin, BLZ 100 100 10 überwiesen.

Die Bearbeitung erfolgt nur, wenn die Zahlungsart angegeben ist!

Für alle, die dieses Angebot noch nicht verstanden haben: Sie abonnieren diese Zeitschrift und bestellen ein Buch, dann können Sie ein weiteres Buch Ihrer Wahl gratis aus unserem Sortiment erhalten. Für den Preis des Abonnements und eines Buches erhalten Sie das Abonnement und zwei Bücher! (1 + 1 = 3)



Der HP 48SX verfügt über eine serielle Schnittstelle, die einen problemlosen Anschluß an IBM-kompatible und Apple Macintosh PCs ermöglicht. Der HP 48SX kann über eine bidirektionale Infrarot-Schnittstelle Programme und Daten an einen weiteren HP 48SX übertragen oder in Verbindung mit dem Infrarot Thermodrucker HP 82240B verwendet werden.



Baustatik-
Programme
für HP-285C
HP-48SX
gesucht!

An:



Fischel GmbH
Kaiser-Friedrich-Str. 54a
1000 Berlin 12

Datenübertragung vom und zum HP 48SX

Grundsätzlich gibt es zwei Wege, um seine Daten vom und zum HP 48SX zu bringen.

1. Datenübertragung zwischen zwei HP 48SX über den Infrarotanschluß
2. Datenübertragung zwischen dem HP 48SX und einem Computer über den seriellen Anschluß

Als Erstinformation sind beide Möglichkeiten im Benutzerhandbuch des HP 48SX ab Seite 663 beschrieben (Kapitel 33).

Der nachfolgende Beitrag beschäftigt sich hauptsächlich mit der seriellen Datenübertragung zwischen dem HP 48SX und einem IBM- oder IBM kompatiblen PC.

Dazu benötigt man das "Serial Interface Kit" von HP für den entsprechenden Computertyp. Darin befinden sich die Übertragungs-Software und das serielle Schnittstellenkabel.

Allgemeine Vorbereitungen:

Alfred Krempf
 Preinsbacherstr. 46
 A-3300 Amstetten

PC: cd \hp48 CR (Verzeichnis-Vorschlag)
 kermit.exe CR Übertragungsprogramm

Kermit - Schnittstelle einrichten: (Vorschläge)
 Kommandodatei "hp48" Aufruf im Kermit: take hp48

Dateinhalt "hp48" : set BAUD 9600
 set PARITY even
 set PORT 1
 define RH REMOTE HOST \%1
 define ST STATUS

Kermit-Befehle: SEND Senden SERVER Server-Modus
 RECEIVE Empfangen SET Setzen
 STATUS Statusanzeige RH Remote Befehle absetzen
 PUSH nach DOS Q Quit
 FINISH Server-Ende ? Hilfe-Bildschirm

HP48SX: HP48SX - Schnittstelle einrichten: (LSHFT = Left-Shift, RSHFT = Right-Shift)
 [LSHFT] [IO] [SETUP] [.....]
 [IR/W] wire Kabelverbindung
 [ASCII] binary Binär, bei Grafik & Bearbeitung ASCII
 [BAUD] 9600 9600 Baud
 [PARIT] even 2 Gerade Parität
 [CKSM] 3 Standard
 [TRAN] 3 Translation 3: CR/LF -> LF, 128 - 255

Translation: 0 Keine Übersetzung
 1 CR/LF -> LF
 2 CR/LF -> LF, Zeichen 128 - 159
 3 CR/LF -> LF, Zeichen 128 - 255

IOPAR: {baud parity receive-pacing transm-pacing checksum trans-code}

[LSHFT] [I/O] [OPENI] Seriellen Anschluß des HP48SX aktivieren.
 [LSHFT] [I/O] [CLOSE] Seriellen Anschluß des HP48SX deaktivieren.
 (Weniger Batterieverbrauch)

Vorbereitung zum Transfer HP28S -> HP48SX:

INPRT vom PC zum HP48SX übertragen. (INPRT wird mit der seriellen Schnittstelle mitgeliefert)
 LED des HP28S zwischen E und T des HEWLETT Logos am HP48SX mit max 1,3 cm Abstand.

Nach der Übertragung: Ebene 1: String
 Ebene 2: 0 -> Fehlerhaft oder 1 -> In Ordnung

Bei 1 sind trotzdem noch fehlende Bytes möglich. String auf LF durchsuchen (jeweils nach 24 Zeichen).

HP48SX <-> PC

HP48SX als Empfänger	PC als Sender
Eingabe am HP48SX:	Eingabe am PC: (Kermit ist eingabebereit)
[LSHFT] [I/O] [RECV] oder 'var' [RECNI]	send (senden; Frage nach 'dateiname')
PC als Empfänger	HP48SX als Sender
Eingabe am PC: (Kermit ist eingabebereit)	Eingabe am HP48SX:
receive CR (empfangen)	'variable' CR; [LSHFT] [I/O] [SEND]
HP48SX als Server	PC als Eingabegerät
Eingabe am HP48SX:	Eingabe am PC: (Kermit ist eingabebereit)
[LSHFT] [I/O] [SERV]	remote host CR
PC als Server	HP48SX als Eingabegerät
Eingabe am PC: (Kermit ist eingabebereit)	Eingabe am HP48SX:
server CR	[LSHFT] [I/O] [KGET]

Backup - Restore

Backup	Eingabe am:	Beschreibung:
[RSHFT] [MODES] [NXT] [RCLF] 'flags' [STO]	HP48SX	Flags vorher sichern
:IO:Name (Datei am PC; z.B.: B911231)	HP48SX	In den Stack legen
receive CR	PC	Kermit ist eingabebereit
[LSHFT] [MEMORY] [NXT] [NXT] [ARCHI]	HP48SX	ARCHIVE
Restore	Eingabe am:	Beschreibung:
'flags' [RSHFT] [MODES] [NXT] [STOF]	HP48SX	Flags wieder setzen
[LSHFT] [I/O] [RECV]	HP48SX	Empfangsbereit
senden CR (senden; Frage nach 'Dateiname')	PC	Kermit eingabebereit
'Name' [RSHFT] [RCL]	HP48SX	Backup Datei aufrufen
[LSHFT] [MEMORY] [NXT] [NXT] [RESTO]	HP48SX	Backup in den Speicher legen

HP28S -> HP48SX

HP28S:	HP48SX:
Eingabe am HP28S:	Eingabe am HP48SX:
[SHFT] [PRINT] [.....]	Vorher starten (Nach 10s ohne Daten OFF)
'variable' [PRVAR]	INPRT (Lieferung mit d. Schnittstelle)
{STACK} [PRST]	
{LCD} [PRLCD]	
Protokoll [TRAC]	

Grafik - Übertragung

Am HP48SX:	Am PC:
Grafik erstellen (z.B. mit PLOT)	GROB2TIF < grob > tiff CR (umwandeln)
[PICT] [RCL] oder [STO] aus Plot (-> GROB)	Bearbeiten (z.B. mit GWS)
GROB in eine Variable speichern	
Variable in ASCII an den PC senden	

! WARNUNG !

AN ALLE HP 48SX-USERS

ACHTUNG: Beim Editieren von Graphikobjekten ist äusserste Vorsicht geboten. Wird einer der beiden Parameter, die die Grösse des Objektes angeben, aus Versehen weggelassen, kann der Rechner einen totalen Systemabsturz erleiden. Dabei gehen alle gespeicherten Daten verloren. Es erscheint zwar die Meldung "Try to recover memory", doch die Daten sind gelöscht! (Bsp.: GROB 4 6 305030503000; die 4 oder die 6 dürfen auf keinen Fall gelöscht werden.)

J.P.B

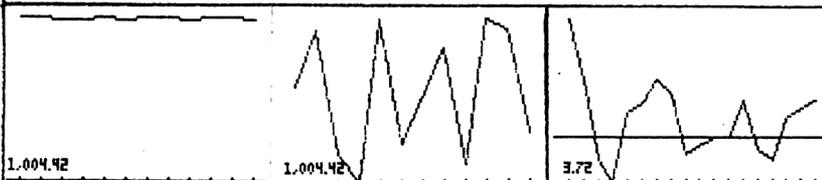
Drei Beispiele zum Senden und Archivieren:

PRESF	Flags restaurieren « RCLF -> f « EVAL f STOF » »	Unterprogramm
ARCH	Backup Sendeaufruf (DATE: 31.121991) « « STD DATE .01 * ->STR -> d « "B" d 8 9 SUB + d 4 5 SUB + d 2 3 SUB + "IO" ->TAG ARCHIVE » » PRESF »	@ ".ddmmjjjj" @ B Binär @ jj Jahr @ mm Monat @ dd Tag @ IO: "Bjjmdd"
SSET	Sende Parameter (ASCII, BIN) « -> m « -35 CF -34 CF 'IO' ->TAG PATH HOME 9600 2 0 0 3 m 6 ->LIST 'IOPAR' STO EVAL SEND » »	Unterprogramm @ 3 oder 0 = Umsetzungscod @ E/A = seriell, Drucker = IR @ Pfad in Stack legen @ "IOPAR" in HOME speichern @ Pfad wieder zurück schalten @ Datei senden
SASCII	Datei Sendeaufruf ASCII « « -35 CF 3 SSET » PRESF »	Ebene 1: Dateiname @ -35 CF = ASCII @ Flags restaurieren
SBIN	Datei Sendeaufruf binär « « -35 SF 0 SSET » PRESF »	Ebene 1: Dateiname @ -35 SF = Binär @ Flags restaurieren

ERKLÄRUNGEN

Dieses Programm ist als Ergänzung zu den bereits im Rechner integrierten Statistik-Grafikfunktionen anzusehen. Es erstellt ein Liniendiagramm von Statistikdaten. Dazu muss eine einspaltige Matrix mit den darzustellenden Daten in die Ebenen eingegeben und STAT gestartet werden. **!ACHTUNG!:** Eventuell gespeicherte Statistikdaten in Σ DAT werden überschrieben und gehen verloren!

Sind alle Werte in der Matrix positiv, fragt das Programm, ob die untere Grenze des anzuzeigende Bereichs Null (Eingabe: 0) oder der kleinste Wert der Matrix (Eingabe: 1) sein soll. Letzteres ist v.a. bei grossen, dicht zusammenliegenden Werten (z.B. 1005, 1011, 998, 995...) sinnvoll. Falls es Werte kleiner Null hat, wird bei der Nullkoordinate eine Linie gezeichnet. Am unteren Bildschirmrand erscheint eine Skala, für jeden Wert einen Punkt. Unten links wird das arithmetische Mittel angezeigt.



Hier sind zweimal die gleichen Werte dargestellt, doch beim 1. Bsp. ist die untere Grenze 0 und beim 2. ist sie = dem kl. Wert.

In diesem Bsp. sind auch einige Werte kleiner als 0 (unterhalb der Linie).

**STATISTIK-
PROGRAMM
FÜR HP 48**

J.P.812

von: J.P. Bergamin
Weststrasse 19
7000 CHUR

ERLÄUTERUNGEN

Dieses Programm stellt das Kalenderblatt eines beliebigen Monats dar. Dazu gibt man einfach den gewünschten Monat in der Form M.JJJJ (z.B. 3.1992) in die Ebene 1 und startet das Program KAL. Es erscheint die Monatsübersicht sowie ein Menü `↑ ↓ PRINT EXIT SHOW REP`. Die Meutüasten haben folgende Funktionen: Der Pfeil nach oben schreitet um einen Monat zurück, der Pfeil nach unten einen weiter. Mit "PRINT" wird der angezeigte Monat ausgedruckt (Abb. 1) und mit "Exit" gelangt man direkt zum Variablenmenü. Haben sie in diesem Monat Termine gespeichert, werden diese durch einen Pfeil gekennzeichnet und können mit "SHOW" betrachtet werden (Abb. 2). Gibt es mehr als 3 Termine in einem Monat, wird mit "SHOW" nur noch das Datum und die Uhrzeit angezeigt. Im Menüfeld ganz rechts steht das Datum des angezeigten Monats.

3.1992							Abb.1	In diesem Beispiel hätte man am 5./9. und am 12. März 92 einen Termin, angedeutet durch die Markierungen.
Mo	Di	Mi	Do	Fr	Sa	So		
			4	5	6	7	8	
9	10	11	12	13	14	15		
16	17	18	19	20	21	22		
23	24	25	26	27	28	29		
30	31							
THU 05.03.92				20:00:00			Abb.2	Mit "SHOW" werden die Termine so am Bildschirm dargestellt.
- SITZUNG								
MON 09.03.92				14:00:00				
- ARZT								
THU 12.03.92				09:00:00				
- BESPRECHUNG								

KALENDER FÜR
DEN HP48

von: J.P. Bergamin
Weststrasse 19
CH-7000 CHUR

KALENDER FÜR DEN HP48	
von: J.P. Bergamin Weststrasse 19 CH-7000 CHUR	
<pre> Prüfsumme: # 52316d KAL < DUP 100 / 1 + DUP 18.101582 DDFYS ABS 7 MOD SWAP DUP .01 + IF DUP 1.13 > THEN 1.9999 - END DAYS LASTARG SWAP IFERR FINDALARM THEN DROP 0 END SWAP IFERR FINDALARM THEN DROP 0 END DUP2 IF → A E < IF A 0 == DUP THEN 0 ELSE A E == END OR > THEN () 0 ELSE IF DUP 0 == THEN OVER DO SWAP DROP DUP RCLALARM 1 GET 1 + FINDALARM UNTIL DUP 0 == END DROP ELSE 1 - END DUP2 DUP2 1 + SWAP - 3 ROLLD FOR A A RCLALARM 1 GET IP SWAP NEXT →LIST LASTARG END 4 ROLLD 5 ROLL 0 1 → L M X Z < 4 ROLL " Mo Di Mi Do Fr Sa So " 10 CHR + SWAP WHILE DUP 0 > REPEAT SWAP " " + 'X' INCR DROP SWAP 1 - END DROP DO L Z IF POS THEN 134 CHR ELSE END Z DUP IF 10 < THEN " " SWAP + END + + 'X' INCR IF 7 MOD 0 == THEN 10 CHR + END 'Z' INCR </pre>	<pre> UNTIL M > END 5 PICK DUP FP 100 ≠ FP SWAP IP + 2 FIX →STR STD ("↑" < 4 DROPN 1 - IF DUP 1 < THEN 11.9999 + END KAL >) ("↓" < 4 DROPN 1 + IF DUP 13 > THEN 11.9999 - END KAL >) ("PRINT" < CR 5 PICK " SWAP + PR1 DROP CR PR1 CR DUP CLLCD 1 DISP 3 FREEZE >) ("EXIT" < 4 DROPN 2 MENU >) ("SHOW" < 4 PICK IF 0 == THEN " ***** # Keine Termine # #in diesem Monat!# ***** ELSE 3 PICK 3 PICK DUP2 SWAP - 3 ROLLD OBJ→ DROP2 3 ROLLD TSTR 3 PICK IF 3 < THEN 10 CHR + " - " ROT + ELSE SWAP DROP END 10 CHR + SWAP NEXT 1 SWAP IF DUP 0 ≤ THEN DROP2 ELSE START + NEXT END END CLLCD 1 DISP 3 FREEZE 0 WAIT DROP DUP CLLCD 1 DISP 3 FREEZE >) 6 PICK 1 →LIST 6 →LIST MENU DROP DUP CLLCD 1 DISP 3 FREEZE > </pre>

4-GEWINNT auf dem HP-48SX

Das Ziel dieses alten Spieles wird wohl jedem klar sein : 4 eigene Steine nebeneinander horizontal vertikal oder diagonal. Zu Beginn des Spiels geben wir die Computer spielen will, 2, falls man mit einem menschlichen Partner spielen will. Wählt man das Spiel gegen den Computer, so erfolgt die Eingabe, ob der Spieler oder der Computer anfangen soll (0 , dann startet der Spieler, 1 , so beginnt der Computer).

Um einen Stein zu setzen. wählt man eine Kolonne 1 bis 7. Die Kolonne 1 ist die ganz links, die 4. ist in der Mitte, die 7. ganz rechts. Hat man eine Kolonne gewählt und eine Taste zwischen 1 und 7 gedrückt, so berechnet der Computer seinen Zug. Dies wird mit Zahlen links und rechts des Spielfeldes angezeigt. Die Höhe der Zahlen gibt an, für welche Spalte der Computer gerade einen Zug berechnet. Die linken Zahlen betreffen seine Züge, die rechten die des Spielers. (Suchen des besten Ortes : zum eigenen Nutzen oder Schaden des anderen). Spielt man zu zweit, so gibt der Computer nach dem ersten Zug die Eingabe sofort frei für den ersten Zug des 2.Spielers. Nach jedem Zug klärt der Computer ab, ob ein Spieler bereits vier in einer Richtung hat und beendet in einem solchen Fall das Spiel mit entsprechender Meldung.

FOUR EXIT D.Glasser 1135
UPDIR >> Karl Jaspers-Allee 21
CH 4052 BASEL

```
PLAY << "Spieler" -> Q
<< W Q + "" INPUT OBJ-> 1 MAX 2 MIN 'A' STO Q DUP 'A' + 'M' STO 'B' + 'N' STO ( ) 'L' STO 1 DUP 'J'
STO 7 DUP 'H' STO SQ FOR X 0 NEXT 7 ->LIST 'B' STO 6 7 * ->LIST 'P' STO SG IF A 2 < THEN "Wer
beginnt ?" 'HP > 0: INPUT OBJ-> ELSE 0 END "E" STO ERASE PICT ( # 41d # 11d ) FE REPL ( # 42d # 3d
) "VIER GEWINNT" R IF E 1 > THEN RAND 7 * 1 + IP 'F' STO 2 'S' STO PL GE END DO 1 'S' STO 55 A 3
* - R-2B # 57d 2 ->LIST A 2 < "IHR ZUG" - M IFTE R IN PL GE F DUP 1 - H MIN 1 MAX 'H' STO 1 + J
MAX 7 MIN 'J' STO L DUP F POS DUP IF 0 > D 6 < AND THEN 0 PUT 'L' STO END CLEAR IF C 6 7 * <
THEN 2 'S' STO ( # 49d # 57d ) A 2 < "COMPUTER" N IFTE R SG IF A 1 > THEN IN ELSE CO END PL GE
END UNTIL C 6 7 * > END IF A 2 < THEN S 2 < "GEWONNEN" - VERLOREN" IFTE ELSE S 2 < "A" -
B' IFTE - GEWINNT" -> END C 7 SQ S "UNENTSCHEDEN" IF ( # 40d # 57d ) SWAP R ( G H J L P W
A B C D E F M N O S T PPAR ) PURGE KEY KEY STD LT CLEAR >>>
CO << F 'N' STO H J FOR V V 'F' STO 6 7 FOR U U SF 1B 91 U B - * + R->B # 30d 2 ->LIST V R IF L F
POS 1 < THEN P FS U 5 - PUT 'P' STO GE 'P' STO END U CF NEXT NEXT W 'F' STO IF F 0 > THEN IF B F
```


Tips und Tricks

Programmhandbuch für den

HP-48SX

Pocket Computer



ISBN 3-89374-076-7

Autorenteam

Fischel GmbH