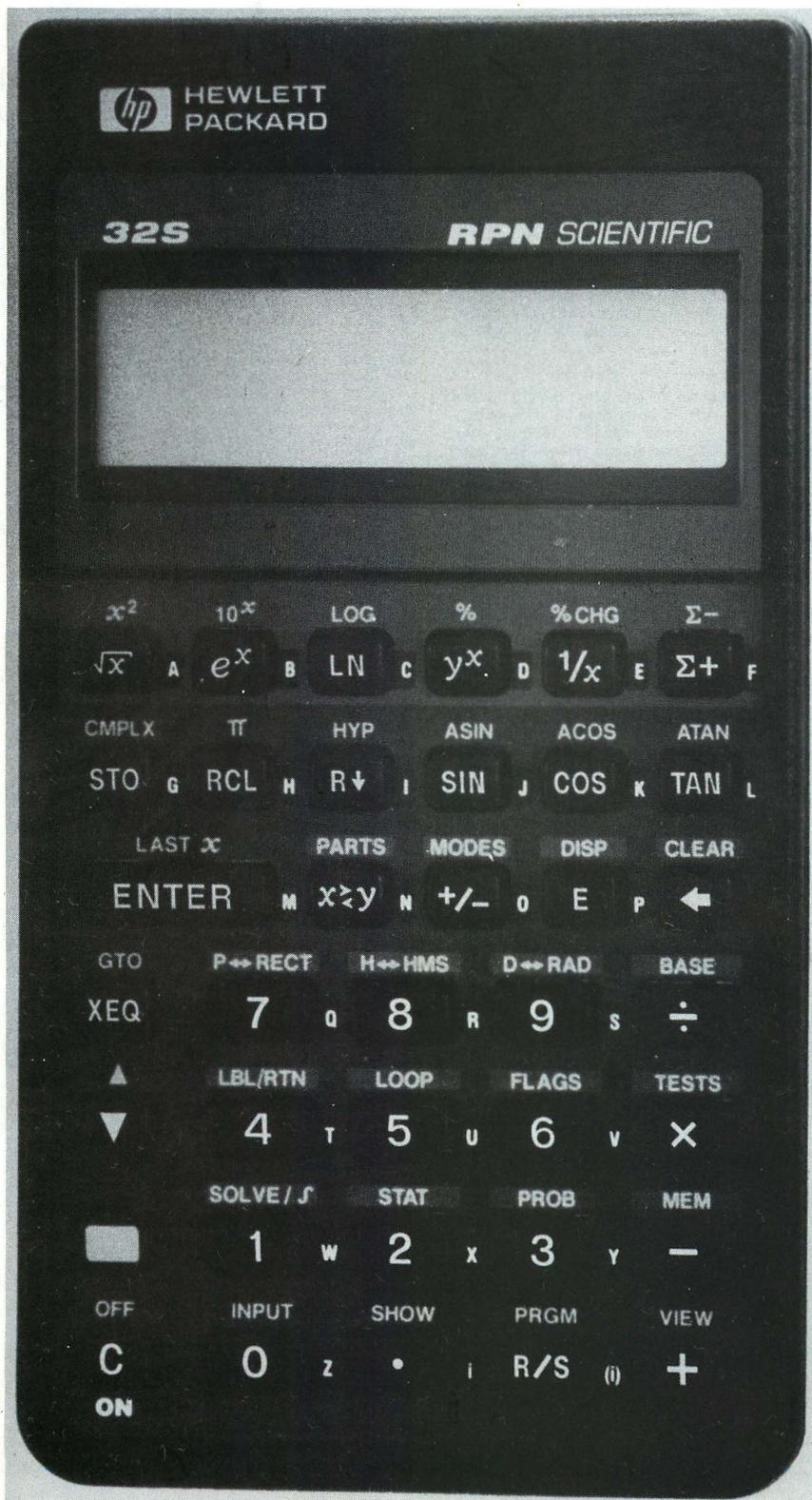


PRISMA

Computerclub Deutschland e.V. · Postfach 11 04 11 · Schwalbacher Straße 50 · D-6000 Frankfurt am Main 1

September/Oktober 1988 Nr. 5

D 2856 F



Magazin

Clubbörse
JPC-EPROM

MS-DOS

Referenzblätter

CP/M

Transform
MBASIC/HP-97 Vergleich

Serie 70

Menü-System
EPROMs brennen
PRIMLEX
MARGELEX
DISABLEX
Messx
Bug in RENUMBER

Serie 40

Speed-UP HP-41 CX
Flexibles Datenbanksystem
VIDEO Editing
NAVIS
M-Code Utilities
Kleiner Texteditor
Kleinbuchstaben
Produkt & Quotientenregel
Schreibschriftlineal
Eckausrundungen
Textzentrierung

Taschenrechner

Der HP-32S
HP-28S Special

Orig. HP 22708A **Soft Fonts** für DeskJet Drucker für 160,- DM zu verkaufen. **Toni Lerchenfeld**, (CCD 1209), ☎ 08323/7323 oder 0043/5222 86669.

Verkaufe **PORT-EXTENDER DM 50,-**, Staubschutz-Overlay DM 10,-, Batteriekasten mit 4 VARTA-Akkus DM 20,- ☎ 06181/20481 (bis 19 Uhr).

Thinkjet (IL-Interface) unbenutzt, 580,- DM, IL-IB-Interface 600,-, Plottermodul 120,- DM, HP-IL-Modul 200,-, HP 16C 150,- DM. ☎ 0721/696922

Verkaufe Schreibmaschine **Brother CE-60** mit Computer-Interface IF-50 und diverserem Zubehör, komplett DM 700,- (VB). Helmut Schaefer, Im Dammwald 20, 6382 Friedrichsdorf 1 ☎ 06172/79741.

Verkaufe **HP 71 B, IL-Modul, IL-Diskettenlaufwerk 9114, IL-ThinkJet**
H. Lunz, ☎ 0931/18383, tagsüber

Verkaufe **HP-41 CV** mit Handbuch und Software 180,- DM Hr. Kumpf ☎ 089/7203208, 8.00 – 17.00h

Suche **HP-IL Interface** Specification 82166-90017 und HP-IL Integrated Circuit 82166-90016 oder zum kopieren. Suche Mitglieder in meiner Umgebung zum Infoaustausch sowie PPC Technical Notes 9,11,13.
Heinz Schmidt, Heideweg 85, 4137 Rheurdt, ☎ 02845/6204.

Suche **Quad-Ram HP 41 c.** Berresheim Friedrich, Mendig, Schillerstr.2. ☎ Privat: 02652/3022, Büro: 02641/25535.

Verkaufe **HP-28 CD**, 1 Jahr jung, VHB 220,- DM, ATARI ST, Original Software „Leisure Suit Carry“, Grafik-Adventure VHB 45,- DM
☎ 07131/484738

Verkaufe wegen Systemwechsel: **HP-71B**, Speichererweiterung, HP-IL-Modul, HP-Mathe-Modul, HP-Forth-Assembler-Texteditor-Modul, HP-IL-Diskettenlaufwerk (HP-9114) mit zweitem Akku (9114B), HP-ThinkJet, 2Druckknöpfe und Druckerstände, original HP-ThinkJet Papier (ca. 1500 Blatt), HP-71 Basic Made Easy, weitere Literatur und Programme alle Handbücher vorhanden, und Zubehör DM 2950,-
☎ 040/7303157

Suche **Programmsammlung**: Business Statistics/Marketing/Sales (00041-90094), auch Kopie erwünscht. W. Knell, Prozeptionsweg 29, D-4720 Beckum, ☎ 02521/7819

Verkaufe **HP125 PC**, bestehend aus Monitor, Tastatur, Doppelfloppy (5 1/4") HP82901M, IB- und ser. Schnittstelle, 2 Stk. HP-IB-Kabel, Software, sehr ausf. Dok. VB 600,- DM
2 Doppel-MM f 410 zusammen 40,- DM. Wer tauscht mein IL-ThinkJet gegen Centronic-ThinkJet? Wer hilft mir bzgl. **Hardware 41CX?** Ich rufe zurück! Gysbert Hageman, Alter Weg 1, 6653 Blieskastel 2, ☎ 06842/2805. (Geschäftszeit.: 06842/3041 oder 3042) CCD 3389.

Suche für **HP 71**: IDS Bände I, II, III sowie HP-IL-Design-Spec. gebraucht oder Kopie (möglichst). Suche **HP-IL/RS232-HP-IL/GPIO-Interfaces** und HP-Mini-Digital-Kassetten. Wer kann Tips/Literaturhinweise zum Entwurf von HP71-LEX-Files geben? Günter Dörfler (2638), ☎ 089/912777 (abends).

Verkaufe: **Advanced Pac Screen** Video Interface, Version: V1, 1A für 550,- DM. Suche für HP71B Math-Modul.
☎ 02751/2114

HP-71B mit IL-Modul, Magnetkartenleser, 4K Speichererweiterung und HP-41 Translator PAC für 800,- DM zu verkaufen.
☎ 05726/281

HP-82161 A Digital Cassette Drive, 380,- DM, 17 Kassetten, z.T. mit Software für HP-41/71, je Stück 15,- DM, **HP-28 C**, 235,-DM. Alles VHB ☎ 07131/484738

Verkaufe **Original-Software**, alle **unbenutzt, neueste Versionen**: dBase III Plus, englische oder deutsche Version, VB 1000,- DM; HP AdvanceLink, englisch, VB 425,- DM; HP Executive Memomaker, englisch, VB 425,- DM; Picture Perfect, englisch, VB 600,- DM; Diagraph, englisch, VB 600,- DM Helmut Schaefer, Im Dammwald 20, 6382 Friedrichsdorf 1 ☎ 06172/79741 (nach 18.00 Uhr).

Suche **HP-34C, HP-65, HP-29C, HP-67**. Ralf Pfeifer (116), Rubensstraße 5, D-5000 Köln 50. ☎ 0221/352033.

Suche für **HP 41** funktionstüchtigen CT Port Extender 4100. J. Darge, ☎ 07253/32181.

Suche **Videointerface PACSCREEN**. Preisvorstellung ca. 400,- DM. Manfred Hammer (2743), Oranienstraße 42, 6000 Wiesbaden, ☎ 06121/375294.

Verkaufe wegen Anschaffung eines PC: **Grabau GR - 7** mit 64 kB Speicher, DM 1000,-, **Monitor NEC-902**, DM 250,-, Kass. Laufwerk 82161A mit 3 Kassetten, DM 500,-
☎ 089/2714396 abends.

Verkaufe **HP-110 LAPTOP**, mit eingebautem Modem! Drucker **HP-THINK-JET**, zusammen VB 2000,- DM oder Tausch gegen XT-Kompatiblen NONAME (Evtl. mit akzeptabler Peripherie)
☎ 089/183384 ab 18.00 Uhr.

VERKAUFE: HP-41 CV, X-Funktion, 32KB-Rambox im Kartenlesergehäuse (incl. Assembler III als Modul-File) mit Bedienungshandbüchern und Literatur (Tricks, Tips und Routinen für HP-41, Synthetisches Programmieren auf dem HP-41- leicht gemacht, erweiterte Funktionen des HP-41-leicht gemacht, Inside the HP-41, HP-41 Hilfen und Anwendungen, Softwareentwicklung am Beispiel einer Dateiverwaltung, optimales Programmieren mit dem HP-41, Assembler-Handbuch mit Dokumentation des HP-41 Betriebssystem) sowie Staubschutzoverlay. Nur komplett: DM 1100,- F. Alex, Wulfeistraße 11, 5860 Iserlohn, ☎ 02374/12003, CCD 3396.

Verkaufe wegen Systemerweiterung folgende Komponenten für **HP-41**: Bar-Code-Leser HP82153A für DM 200,-; Funktions/Doppelspeichermodul HP82180A mit HP82181A für DM 100,-; Extended I/O Modul HP82183A für DM 100,-; **HEPAX Modul** Std für DM 400,- und folgende Literatur: Plotten und Drucken auf dem Thermodrucker – Meschede für 15,-; Erweiterte Funktionen des HP41-leicht gemacht – Dalkowski für DM 10,-; Zusammenfassung der Bedienungsanleitungen HP80 – Stroinski für DM 15,-. Alle Komponenten und Bücher sind im Bestzustand. W. Meschede (CCD2670) ☎ 0911/808756 abends.

Verkaufe folgende Bücher: Alberts: HP-41 BARCODES mit dem HP-IL-System; Jarett: Synthetisches Programmieren – leicht gemacht; Jarett: Erweiterte Funktionen – leicht gemacht; Meschede: Plotten und Drucken auf dem HP-14-Thermodrucker; Wickes: Synthetische Programmierung; Kruse: optimales Programmieren; Gosmann: Anwenderhandbuch HP-41; Kane/Harper: Das HP-IL-System; Dearing: Tricks und Tips. Stück je DM 8,-, alle zusammen DM 50,-. **Weiter verkaufe ich: HP-41 CX** – DM 250,-; Thermodrucker – DM 250,-; Kartenleser – DM 150,-; Barcodeleser – DM 120,-; CCD-Modul – DM 100,-; PPC-ROM – DM 100,-; PORT-EXTENDER – DM 50,-; 300 Original-HP-Magnetkarten, unbeschriftet, 10 Stück DM 8,-. Bei geschlossener Abnahme alles noch billiger! ☎ 06181/20481 bis 19 Uhr.

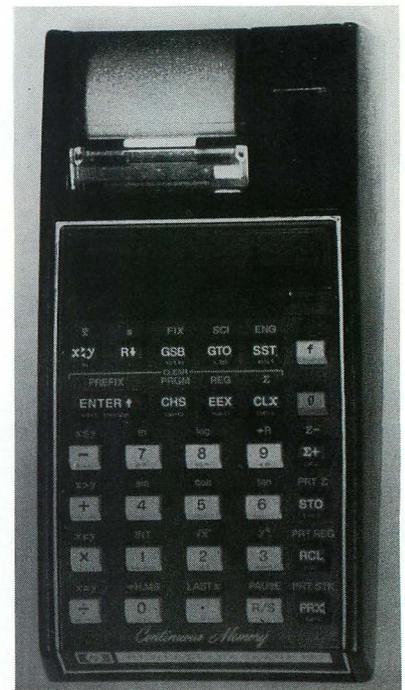
Verkaufe: HP-41 CX, Barcodeleser, Kartenleser, Battery Charger HP82066B. Alle Geräte in Originalverpackung und mit original- Handbüchern. Literatur: „Synthetische Programmierung auf dem HP-41“, „Tips, Tricks und Routinen für Taschenrechner der Serie 41“; sämtliche Ausgaben der Zeitschrift PRISMA ab 01.1982, sowie das Buch „Best of Prisma“. Nur komplett: DM 850,-. H. Ruge, Leeschamp 34, 2875 Ganderkesee/Heide II, ☎ 04221/41135.

ThinkJet HP2225B mit HP-IL Schnittstelle zu verkaufen, VB 650 DM, M. Rabe, Tel. (0521) 324474



HP-10

Auch das hat HP gebaut: Dieses Modell wurde Ende Februar 1978 gebaut. Ohne ENTER-Taste, Drucker und Gehäuse sind identisch mit dem zur gleichen Zeit gebauten technischwissenschaftlichen HP-19C.



HP-19C

Dieser Typ lief bei HP 1979 aus. Der 19C wurde zusammen mit dem HP-29C gebaut, der - bis auf den Thermodrucker - die gleichen Funktionen wie der 19C besaß und daher etwas kleiner ausfiel. Der HP-19C/29C bot 98 Programmschritte und 30 Datenregister, wobei allerdings die letzten 14 nicht im Permanent-Speicher lagen und nur indirekt zu adressieren waren. Das Modell kostete damals um 1000 DM.

Ralf Pfeifer (116)

Impressum

Titel:

PRISMA

Herausgeber:

CCD-Computerclub Deutschland e.V.

Postfach 11 04 11

Schwalbacher Straße 50

6000 Frankfurt am Main 1

Verantwortlicher Redakteur:

Alf-Norman Tietze (ant)

Redaktion:

Hans Jürgen Hübner (hjh)

Klaus Kaiser (kk)

Michael Krockner (mik)

Martin Meyer (mm)

Henry Schimmer (hs)

Dieter Wolf (dw)

Herstellung:

CCD e.V.

Manuskripte:

Manuskripte werden gerne von der Redaktion angenommen. Honorare werden in der Regel nicht gezahlt. Die Zustimmung des Verfassers zum Abdruck wird vorausgesetzt. Für alle Veröffentlichungen wird weder durch den Verein noch durch seine Mitglieder eine irgendwie geartete Garantie übernommen.

Druck und Weiterverarbeitung:

Reha Werkstatt Rödellheim

Biedenkopfer Weg 40 a, 6000 Frankfurt

Anzeigenpreise:

Es gilt unsere Anzeigenpreisliste 3 vom Juni 1987

Erscheinungsweise:

PRISMA erscheint jeden 2. Monat.

Auflage:

3000

Bezug:

PRISMA wird von allen Mitgliedern des CCD ohne Anforderung übersandt. Ein Anspruch auf eine Mindestzahl von Ausgaben besteht nicht. Der Bezugspreis ist im Mitgliedsbeitrag enthalten.

Urheberrecht:

Alle Rechte, auch Übersetzung, vorbehalten. Reproduktionen gleich welcher Art – auch ausschnittsweise – nur mit schriftlicher Genehmigung des CCD. Eine irgendwie geartete Gewährleistung kann nicht übernommen werden.

Inhalt

Neue und alte Klänge

Magazin

Clubbörse	2
Editorial	3
JPC-EPROM	4

MS-DOS

MS-DOS Referenzblätter	13
------------------------	----

CP/M

Transform	15
MBASIC/HP-97 Vergleich	17

Serie 70

HP-71 Menü-System	5
EPROMs brennen	7
Lexfile: PRIMLEX	9
Lexfile: MARGELEX	9
Messx	10
Lexfile: DISABLEX	12
Bug in RENUMBER	12

Serie 40

Speed-Up HP-41 CX	18
Flexibles Datenbanksystem	19
VIDEO Editing	26
NAVIS	27
M-Code Utilities	34
Kleiner Texteditor	36
Kleinbuchstaben in M-Code	37
Produkt & Quotientenregel	37
Schreibschriftlineal f. ThinkJet	39
Eckausrundungen	40
Textzentrierung auf dem ThinkJet	45

Taschenrechner

Der HP-32S	42
HP-28S Special	43

Clubadressen

54

Titelseite:

Der neue HP-32S: kompakt, leistungsfähig und mit UPN Eingabe-System. Er hat zahlreiche mathematische und wissenschaftliche Funktionen und ist programmierbar. Leider bietet er keinerlei Möglichkeit, die Programme extern zu speichern, weshalb er nur ein "einfacher" - aber dennoch guter - Taschenrechner bleibt.

Fangen wir am Besten gleich mit den neuen an, manch einer wird sich jetzt denken, was denn nun schon wieder.

Beim Weiterblättern wird wohl so Manchem auffallen, daß, wie aus dem Nichts, urplötzlich mitten im PRISMA richtige Bilder auftauchen. Die Redaktion hat sich nun schließlich und endlich dazu entschlossen der Kreativität freien Raum zu lassen und durch Untermalung von Bildern soetwas wie eine visuelle Vorbereitung für den dann folgenden Artikel zu inszinieren.

Mögen die Anfänge in diesem Heft noch etwas bescheiden in der Aufmachung sein, dem Ingenör ist nichts zu schwör, die Gestaltung ist immer auch ein Maß der hinter sich gebrachten Übung, das wollen wir nun denn auch in Zukunft recht heftig tun.

Sollten einige unter Euch Gefallen an solcherlei Ausgestaltung haben, es seien hiermit alle Clubmitglieder dazu aufgefordert sich an der visuellen Ausgestaltung ihrer Zeitschrift PRISMA zu beteiligen, dazu gleich die Grundregeln:

Es können Bilder (wenn möglich nur schwarz-weiß, Farbbilder haben in der Regel viel zu wenig Kontrast) jeglicher Aufmachung eingesandt werden, soweit sie irgendwie zu einem Artikel thematisch dazupassen oder im entferntesten mit PRISMA zu assoziieren sind. Bei Testberichten bieten sich ja z.B. die getesteten Geräte, dies macht einen solchen Bericht schon wesentlich plastischer für die Leser.

Ein Tip noch: wir, die Redaktion (speziell Dieter Wolf) können schwarz-weiß Vergrößerungen von Negativen machen, diese bekommt Ihr dann, wie die Bilder auch oder die Magnetkarten, nach der Veröffentlichung wieder zurück. Negative wären also das Optimum an Unterstützung, daraus läßt sich in der Regel leichter ein passendes Format und ein entsprechender Kontrast zaubern.

So, zum Schluß noch kurz ein anderes Thema: Seit Anfang des Jahres ist ja der MARYS II auf dem Weltmarkt erschienen, auf der MVV im Frühsommer wurde er ja auch vorgestellt, einen ersten Testbericht dazu hatte ich auch verfasst, seitdem ist es still um diesen (vielleicht) HP41-Nachfolger geworden, sollte er gestorben sein, bevor er so richtig das Licht der Welt erblickt hat ?

Noch zu guter Letzt: die Redaktion setzt alles daran, daß Heft 6 bis Weihnachten jedem unterm Christbaum liegt...

MM (1000)

JPC-EPROM

Das JPC Lexfile, das von den Franzosen angeboten wird, kommt immer häufiger zum Einsatz bei der Programmierung. Die zahlreichen und sehr nützlichen Befehle und Funktionen machen das HP-71 BASIC noch leistungsfähiger, weshalb viele Programmierer das JPC Lexfile gerne verwenden.

Dieses inzwischen über 26 KB große Lexfile ist ein kommerzielles Produkt mit Copyright. Deshalb ist es auch nicht in der HP-71 Clubbibliothek erhältlich.

Es wird komplett mit einem ausgezeichneten Handbuch (in Englisch) und nur in einem EPROM-Modul für den HP-71 Frontport vom PPC Paris (für Frankreich), von CMT (für USA) und von W&W in Deutschland vertrieben. Der Preis liegt bei ca. 490,- DM.

Die Bezugsadresse für Deutschland lautet:

W & W Software Products GmbH
Odenthaler Str. 214
Postfach 200970
D-5060 Bergisch Gladbach 2
Tel. (02202) 42021

Clubmitglieder sollten unter Angabe ihrer CCD-Mitgliedsnummer immer nach einem besonderen Angebot fragen.

(ant)

*Hardware · Software
Servicestation
Beratung · Zubehör*

OSBORNE
Management by Computer.

WORDLORD · Textverarbeitung · CAD-Anwendungen · Komplettsysteme

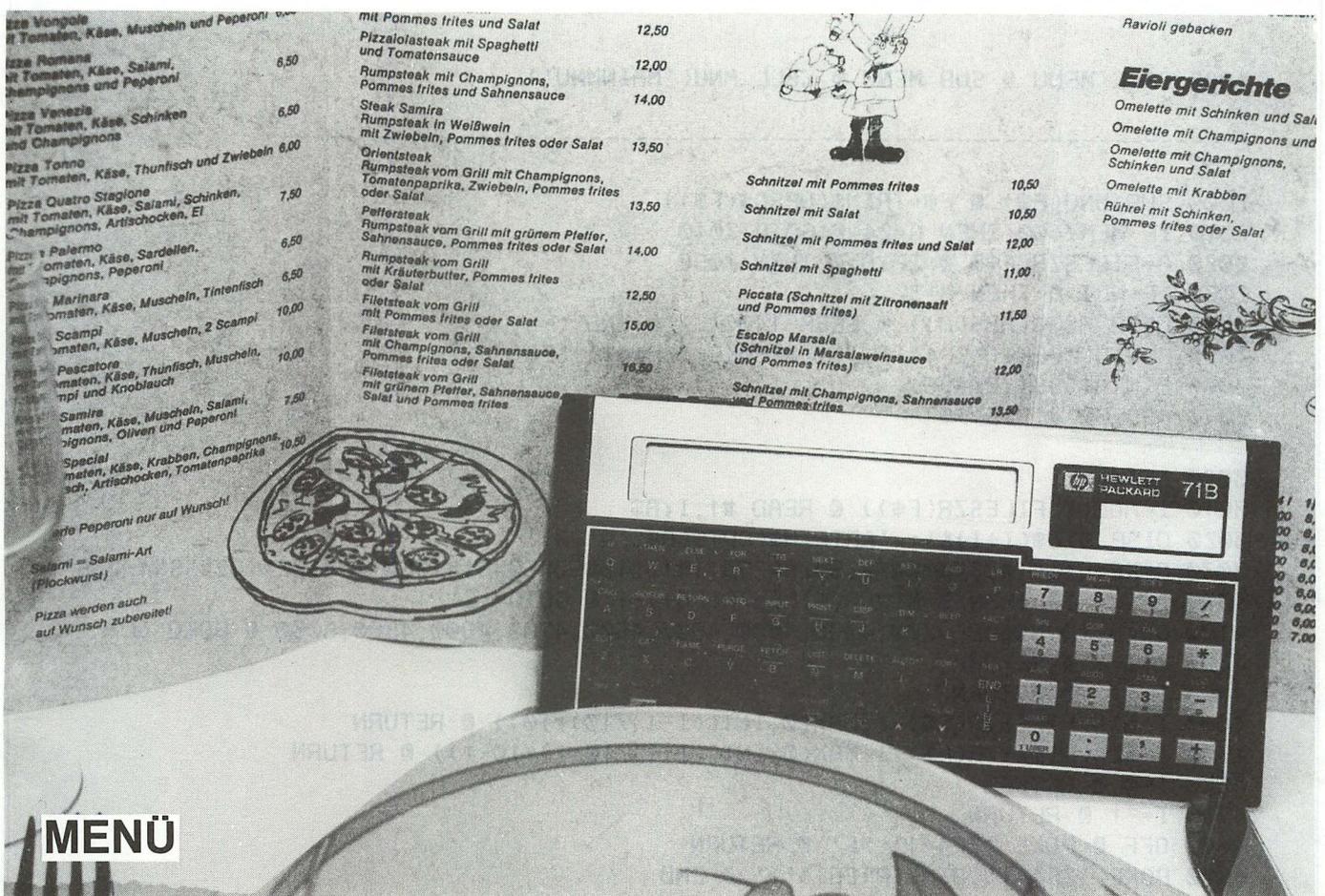
PCE **PFORTNER GMBH**
Computer-Technik · Elektronik

Branchenlösung für Klein- u. Mittelbetriebe

Postfach 1220 · 4133 Neukirchen-Vluyn
Telefon 0 28 45 / 3 22 94



Sonderpreise für CCD-Mitglieder



MENÜ

Ein Menüsystem für den HP-71

Benutzt man seinen Rechner Tag ein, Tag aus nur um immer die gleichen paar Programme, die man auch ständig im Rechner hat, laufen zu lassen (z.B. einen Wecker, um morgens aus dem Bett zu kommen, oder einen Timer zum Frühstückseier kochen), so ist es doch sehr lästig, immer CALL DINGSBUMS oder so ähnlich einzutippen oder nicht eingewiesenen Personen die Bedienung zu erklären. Einfacher ist es da doch schon, sich in einem Menü das gewünschte Programm auszusuchen und von dort aus zu starten.

Genau dafür ist MENU. Es müssen nur die gewünschten Applikationen installiert werden, und schon ist die Bedienung des Rechners kinderleicht (beschränkt auf die installierten Programme). Zum installieren der Applikationen benötigt man nur einen Text-Editor, mit dem man pro Zeile zuerst den im Menü angezeigten Text, gefolgt von einem Klammeraffen (Ⓔ) und dem Namen des Programms bzw. Unterprogramms eingibt. Leerzeilen sind nicht erlaubt. Es empfiehlt sich, die Länge des Anzeigetextes auf neunzehn Zeichen zu beschränken.

Es können an das aufzurufende Programm bis zu zwei numerische oder String-Parameter übergeben werden. MENU enthält zwei Unterprogramme: MENU und MNU. MENU ist parameterlos und ruft MNU mit dem Parameter „MAINMNU“ auf. MNU ist das eigentliche Hauptprogramm und verlangt als Parameter den Namen des Text-Files, in dem die installierten Programme stehen. MNU kann sich auch rekursiv aufrufen.

Z.B.

- erste Zeile: Programm Ⓔ PROGRAM (pi*12)
- zweite Zeile: noch ein Programm Ⓔ PROGRAM1 (123.'Text')
- dritte Zeile: Text Editor Ⓔ TEDDY
- letzte Zeile: Untermenü Ⓔ MNU ('MENU')

Befindet man sich im Menü, kann man sich mit den Cursor-Tasten (shift) „rauf“ und „runter“ durch das Menü hangeln (wobei [f][^] bzw. [f][v] in Zehnerschritten durch das Menü gehen), mit den Zifferntasten [1] bis [0] die ersten zehn Menüpunkte direkt anwählen, mit der „END LINE“-Taste eine Applikation auswählen, mit der „OFF“-Taste den Rechner ausschalten, ohne das Programm zu verlassen und mit der „ATTN“-Taste das Menü verlassen.

Für alle die sich nicht so mit dem 'rum-poken auskennen:

Steht an der Adresse „2F441“ ein Wert ungleich null, so kann ein laufendes Programm nicht durch betätigen der ATTN-Taste unterbrochen werden. Den alten Wert habe ich in einer numerischen Variable gespeichert, weil sie drei Byte weniger in Anspruch nimmt, als eine String-Variable mit einem Byte Länge und nicht dimensioniert werden muß.

An der direkt darauffolgenden Adresse befindet sich das „ATTENTION FLAG“. In dem Nibble wird abgespeichert, wie oft die ATTN-Taste gedrückt wurde. Dadurch wissen z.B. einige IL-Befehle, ob die Taste zweimal gedrückt wurde, da erst dann die Operation abgebrochen wird. Wird dort bei Beendigung eines Programms ein Wert ungleich null hin geschrieben, so wird, auch wenn zuvor etwas angezeigt wurde, die Anzeige gelöscht und der Cursor mit der vorangestellten BASIC-Eingabeaufforderung erscheint. Eine andere Möglichkeit wäre, im Programm PUT"#43" auszuführen. Allerdings erscheinen dann auf einem Monitor zwei „>“.

Folgende LEX-Files werden benötigt: CCDUTIL, STRINGLX und EDLEX.

```
1000 CALL MENU @ SUB MENU @ CALL MNU('MAINMNU')
```

```
-----
2000 SUB MNU(F$) @ F$=TRIM$(UPRC$(F$))
2010 IF MEM<400 THEN A=24 @ GOTO 2040
2020 A=FILESZR(F$) @ IF A>0 THEN 2050
2030 IF NOT A THEN A=32
2040 DISP MSG$(ABS(A)) @ BEEP 1400,.1 @ END
2050 P=HTD(PEEK$('ZF441',1)) @ POKE 'ZF441','F0'
2060 DIM A$(96),B$(96)
2070 ASSIGN #1 TO F$
```

```
3000 'L':
3010 I=MOD(I,FILESZR(F$)) @ READ #1,I;A$
3020 DISP STD$(I+1)&': '&RTRIM$(CUT$(A$,'@',0))
3030 A=KEYCTL('#39#40#41#25#26#27#11#12#13#53#50#51#106#107#162#163#99#43#38')
4000 IF A<=10 THEN I=MIN(A,FILESZR(F$))-1 @ GOTO 'L'
4100 ON A-10 GOSUB 4200,4300,4400,4500,4600,4700,4800,4900,5000 @ GOTO 'L'
4200 I=I-1 @ RETURN
4300 I=I+1 @ RETURN
4400 I=MIN(FILESZR(F$)-1,MAX(0,INT((I-1)/10)*10)) @ RETURN
4500 I=MIN(FILESZR(F$)-1,MAX(0,INT((I+1)/10+1)*10-1)) @ RETURN
4600 I=0 @ RETURN
4700 I=-1 @ RETURN
4800 OFF @ POKE 'ZF441','F' @ RETURN
4900 POKE 'ZF441',DTH$(P)[5]&'1' @ END
```

```
5010 ON ERROR GOTO 5170 @ ASSIGN #1 TO * @ POKE 'ZF441',DTH$(P)[5]&'0'
5020 A$=TRIM$(CUT$(A$,'@',1,INF))
5030 IF NOT POS(A$,'(') THEN CALL A$ @ GOTO 5180
5040 B$=TRIM$(CUT$(A$,'(',1)) @ A$=TRIM$(CUT$(A$,'(',0))
5050 IF B$[LEN(B$)]#')' THEN MESSAGE 76,32 ELSE B$[LEN(B$)]=' '
5060 A=POS('"'&'"',B$[1,1]) @ IF A THEN 5120 ELSE A=VAL('('&CUT$(B$,' ',0)&')')
5070 B$=CUT$(B$,' ',1,INF)
5080 IF NOT LEN(B$) THEN CALL A$(A) @ GOTO 5180
5090 B=POS('"'&'"',B$[1,1]) @ IF B THEN 5110
5100 B=VAL('('&B&')') @ CALL A$(A,B) @ GOTO 5180
5110 B$=CUT$(B$,'"'&'"')[B,B],1) @ CALL A$(A,B$) @ GOTO 5180
5120 C$=CUT$(B$,'"'&'"')[A,A],1) @ B$=CUT$(B$[LEN(C$)+3],',',1,INF)
5130 IF NOT LEN(B$) THEN CALL A$(C$) @ GOTO 5180
5140 A=POS('"'&'"',B$[1,1]) @ IF A THEN 5160
5150 A=VAL('('&B&')') @ CALL A$(C$,A) @ GOTO 5180
5160 B$=CUT$(B$,'"'&'"')[A,A],1) @ CALL A$(C$,B$) @ GOTO 5180
5170 BEEP 1400,.1 @ DISP ERRM$ @ WAIT 1
5180 OFF ERROR @ POKE 'ZF441','F0' @ ASSIGN #1 TO F$
5190 RETURN
6000 END SUB
```

Matthias Rabe
Teichsheide 13
4800 Bielefeld

Der HHP-EPROM-Halter ist für den Einsatz handelsüblicher EPROMs ausgelegt, die mit jedem 08/15 EPROMmer gebrannt werden können. Es ist nur darauf zu achten, daß jedes Bit invertiert wird. Sollen BASIC Files in ein EPROM gebrannt werden, so müssen Zeilenreferenzen (Sprünge etc.) aufgelöst und Unterprogramme, benutzerdefinierte Funktionen und Labels verkettet werden. Das erste kann man mit RENUMBER 1,1,1,1 erreichen, die Verkettung durch ausführen, kopieren oder zwischen TEXT und BASIC hin und her TRANSFORMieren des Programms. Sonst kann das Programm unter Umständen nicht im EPROM ausgeführt werden.

Da wohl jeder EPROMmer anders angesteuert wird, ist das Programm nur als Beispiel gedacht. Es ist für den EPROG-27011 von Völkner und Conrad-Elektronik geschrieben.

Das Programm kann nur 32k EPROMs mit 12.5 V und 21 V Programmierspannung brennen. Alle zu brennenden Files müssen sich in einem 32 kByte großen Port befinden. Das Programm arbeitet etwa so schnell, wie die EPROMs gebrannt werden können. Typische Zeiten liegen zwischen sechs und sieben Minuten (je nach EPROM).

Das Unterprogramm FLTIME dient dazu, Uhrzeit und Datum aller Files eines Ports auf den gleichen Wert zu setzen. Für BURN darf das erste File im Port keinen illegalen Namen haben (Kleinbuchstaben und so), für FLTIME alle Files nicht.

Folgende LEX-Files werden benötigt: HPILROM, CCDUTIL, JPCROM, STRINGLX

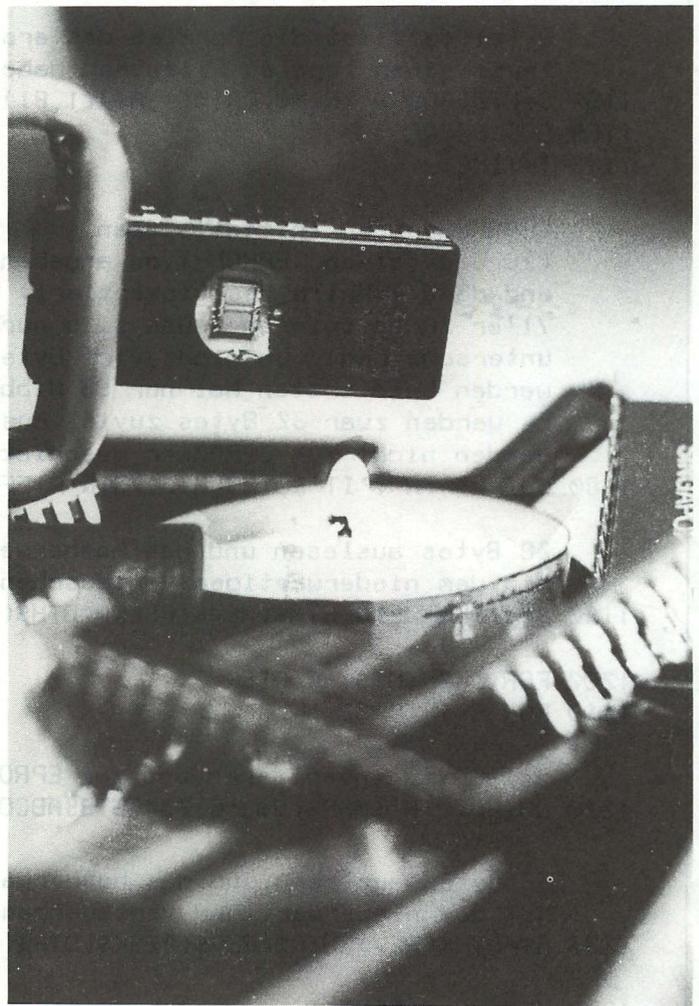
Es werden übrigens kein (CR/LF) zum EPROMmer geschickt!

Achtung!!!

Jedes BASIC-File vor dem Brennen kopieren, ausführen oder in ein Text-File und zurück transformieren um alle Labels, Subprograms und user-defined Funktionen zu verketteten. Außerdem noch RENUMBER 1,1,1,1 ausführen, um die Zeilennummer-Referenzen zu compilieren.

Ohne Kommentare wird das Programm schneller als der EPROMmer.

(und 1192 Bytes klein).



EPROMs brennen

```
100 CALL BURN
```

```
1000 SUB BURN
```

```
1010 RESTORE IO @ R=DEVADDR('RS232')
```

```
1020 IF R=-1 THEN DISP 'No RS-232' @ BEEP 1400,.1 @ END
```

```
    Schnittstelle initialisieren :
```

```
    no software protocol, observe all lines except DCD
```

```
1030 CLEAR :R @ REMOTE @ OUTPUT :R ;'C0;SL0;SL3;' @ LOCAL
```

```
1040 DIM A$(64) @ A$=' 27256 : 12.5 Volt @ 27256 : 21.0 Volt'
```

```
1050 DISP CUT$(A$,'@',A)
```

```
1060 SELECT KEYCTL('#50#51#43#38',2)
```

```
1070 CASE 1 @ A=0 @ GOTO 1050
```

```
1080 CASE 2 @ A=1 @ GOTO 1050
```

```
1090 CASE 3 @ POKE '2F442','1' @ END
```

```
1100 END SELECT
```

```
1110 ON ERROR GOTO 1120
```

```
1120 FINPUT A$,'PORT(0.00)','$PUPUUP',I
```

```
1130 IF NOT I THEN POKE '2F442','1' @ END
```

```
1140 A$=':PORT('&A$(1,11)&.'&A$(21)&')'
```

```

0 (offset) ist die Adresse des ersten Files im angegebenen Port
(muss einen legalen Filenamen haben !!)
1150 O=HTD(ADDR$(TRIM$(CAT$(1,A$)[1,8])&A$))
1160 OFF ERROR
1170 T=TIME

```

EPROM-Brenner initialisieren
(zuruecksetzen, EPROM-Type angeben, Programmieralgorithmus auswaehlen)
und dann 4 Nullbytes (invertiert) brennen, um den Device-Typ fuer den
71'er zu definieren (muss sich nur irgendwie von 'IRAM' (B30000DE)
unterscheiden). Um nicht vier Bytes zuviel zum EPROMmer zu schicken,
werden beim ersten Mal nur 56 Nibbles geschickt.
Es werden zwar 32 Bytes zuviel aus dem Speicher gelesen, aber die
werden nicht zum EPROMmer geschickt.

```
1180 OUTPUT :R ;'IT'&DTH$(A+3)[5]&'PIFFFFFFF';
```

28 Bytes auslesen und das hoeherwertige Nibbel eines Bytes
mit dem niederwertigem vertauschen

```
1190 A$=ATH$(REV$(HTA$(REV$(PEEK$(DTH$(0),56)))) @ 0=0-8
```

```
1200 FOR I=0 TO 31 @ DISP I+1
1210 FOR S=1 TO 32
```

alle Bits invertieren und zum EPROMmer schicken

```
1220 OUTPUT :R ;MAP$(A$,'0123456789ABCDEF','FEDCBA9876543210');
```

32 Bytes auslesen und das hoeherwertige Nibbel eines Bytes
mit dem niederwertigem vertauschen

```
1230 A$=ATH$(REV$(HTA$(REV$(PEEK$(DTH$(I*32+S)*64+0),64))))
```

```
1240 NEXT S @ NEXT I
```

Status aus dem EPROMmer auslesen und das Teil zuruecksetzen

```
1250 OUTPUT :R ;'S'; @ ENTER :R ;S @ OUTPUT :R ;'I';
1260 IF S THEN DISP 'ERROR !!!!'; ELSE DISP 'ok.';
1270 I=(TIME-T)/60 @ IF I<0 THEN I=I+1440
1280 BEEP 1400,.2 @ DISP INT(HMS(I/60)*10000+.5)/100;'min.'
```

```
2000 SUB FLTIME
```

- setzt Uhrzeit und Datum aller Files eines Ports auf die eingegebenen Werte
Alle Files muessen legale Filenamen haben.

```

2010 DIM A,P$,T$,D$
2020 FINPUT P$,'PORT(0.00)';'SPUPUUP',A @ IF NOT A THEN 2140
2030 P$=':PORT('&P$[1,1]&'.'&P$[2]&')'
2040 FINPUT T$,'Time ['&TIME$[1,5]&]';'6P2UP2UP',A @ IF NOT A THEN 2140
2050 FINPUT D$,'Date ['&DATE$&]';'6P2UP2UP2UP',A @ IF NOT A THEN 2140
2060 ON ERROR GOTO 2130
2070 T$=REV$(T$[1,4])&REV$(D$[1,6])
2080 FOR A=1 TO INF
2090 D$=TRIM$(CAT$(A,P$)[1,8])
2100 IF LEN(D$) THEN DISP D$ ELSE DISP 'ok.' @ END
2110 POKE DTH$(HTD(ADDR$(D$&P$))+22),T$
2120 NEXT A
2130 DISP ERRM$ @ END
2140 POKE '2F442','1' @ END

```

Matthias Rabe
Teichsheide 13
4800 Bielefeld 1

PRIMLEX

... und noch ein Lex-File

145 Bytes, HP-71

Dieser Lex-File beschäftigt sich mit Primzahlen und wendet sich an die Mathematiker unter den HP-71B Anwendern. Aber auch für weniger schwierige Aufgaben eignet sich die PRIM-Funktion, wie in den Beispiel – Programmen gezeigt wird.

Das Programm stammt von den SIG – Mitgliedern Oliver Arbey und Guy Toublanc und wurde unter anderem bereits im SIG-Journal „JPC“ Nr. 26, 35 und 37 veröffentlicht, sowie im CHHU-Chronicle V2N7p48.

Die Größe des Files ist 145 Bytes. Als Keyword ist nur eines vorhanden:

```
PRIMLEX ID=E1 SIZE=145 Bytes
57 PRIM XFN 225087 (#)
```

Die Funktion lautet PRIM (n). Dabei ist n ein beliebiges positives numerisches Argument. Es kann bis zu 2.E15 betragen. Sinnvoll ist allerdings nur eine Zahl innerhalb der 12-stelligen Rechengenauigkeit des HP-71B, also max. 999,999,999,999. Falls es nicht ganzzahlig ist, wird automatisch der gerundete Wert (4/5 Rundung) verwendet. PRIM ist eine Funktion und liefert das Argument als Ergebnis, falls es eine Primzahl ist (also nur durch sich selbst und durch 1 ohne Rest teilbar ist). Ansonsten liefert sie den kleinsten Teiler des Arguments.

Es gibt natürlich eine Menge Anwendungsmöglichkeiten für diese Funktion, z.B. Primzahlentester, Bruchrechnungsprogramme, oder auch GGT und KGV – Programme. Dafür bringe ich zwei BASIC-Listings als Beispiel.

Beispiel: Größter gemeinsamer Teiler

```
10 INPUT 'A,B ? ':A,B @ T=1
20 A1=PRIM(A) @ B1=PRIM(B)
30 IF A1=B1 THEN T=T*A1 @ A=A/A1 @ B=B/B1 @ GOTO 50
40 IF A>B THEN A=A/A1 ELSE B=B/B1
50 IF A#1 AND B#1 THEN 20
60 DISP "GGT: ";T @ END
```

Beispiel: Kleinstes gemeinsames Vielfaches:

```
10 INPUT 'A,B ? ':A,B @ V=1
20 A1=PRIM(A) @ B1=PRIM(B)
30 IF A1=B1 THEN A=A/A1 @ B=B/B1 @ V=V*A1 @ GOTO 50
40 IF A>B THEN A=A/A1 @ V=V*A1 ELSE B=B/B1 @ V=V*B1
50 IF A#1 OR B#1 THEN 20
60 DISP "KGV: ";V @ END
```

Hexdump-Listing: PRIMLEX

(Zum Abtippen benötigen ein Hex-Ladeprogramm, wie z.B. MAKEFILE aus Prisma 7/86!)

PRIMLEX	L	ID#E1	145	Bytes
0123	4567	89AB	CDEF	ck
000:	0525	94D4	C454	8502 9D
001:	802E	1061	3151	8088 E1
002:	1010	01E7	5750	0000 FF
003:	F710	0000	0000	0000 A7
004:	0710	00F7	0525	94D4 00
005:	751F	F811	8FC1	DB01 AA
006:	028F	FB6C	08F3	55C0 9A
007:	AD48	F2EA	2112	28F2 AD
008:	EA21	100A	F230	2AF7 0B
009:	7740	B667	0407	9307 0C
00A:	5307	E207	D207	6207 4A
00B:	5207	E107	7107	9107 DC
00C:	F001	12B1	A59D	1104 AA
00D:	04A1	BA1B	A1BA	F511 CE
00E:	0209	FCA0	BF10	C55F 70
00F:	B705	CFA7	00D4	80BF 28
010:	55EE	2691	C00A	FA07 8A
011:	8F22	3B1A	F68D	612F F0
012:	0			30

Michael Fiedler
Friedrichstraße 17
6070 Langen

MARGELEX

... und noch ein LEX-File

Hier ist wieder ein LEX-File, der uns hilft, den Überblick über die momentane Cursorposition im Display zu behalten. Das einzige enthaltene Befehlsword heißt MARGIN. Damit kann man eine Art „akustischen“ Tabulator auf eine beliebige Spalte setzen. Jedes Mal, wenn der Cursor über diese Position kommt, ertönt dann ein kurzer Piepston. Nützlich ist das z.B. für Texteingaben, hier könnte man das Margin etwa auf 10 Zeichen vor dem Zeilenende setzen, um rechtzeitig eine Warnung zum Beenden eines Wortes und Beginn einer neuen Zeile zu bekommen.

Das Programm stammt von Michel Martinet vom französischen SIG und wurde unter anderem bereits im SIG-Journal „JPC“ Nr. 26 veröffentlicht, sowie im CHHU-Chronicle V2N7p48. Die Größe

des Files ist 129 Bytes. Als Keyword ist nur eines vorhanden:

```
MARGELEX ID=E1 SIZE=129 Bytes
Polls!
1D MARGIN Stmt 225029
```

Der Befehl lautet MARGIN n. Dabei ist n ein beliebiger numerischer Ausdruck, der innerhalb der eingestellten Zeilenbreite (WIDTH) liegen muß. Falls er nicht ganzzahlig ist, wird automatisch der gerundete Wert (4/5 Rundung) verwendet.

Beispiel:
MARGIN 86 setzt den "Piepser" auf Spalte 86, zehn Stellen vor dem Default-Zeilenende von 96.
MARGIN 0 setzt den Piepser außer Betrieb.

MARGELEX	L	ID#E1	129	Bytes
0123	4567	89AB	CDEF	ck
000:	D414	2574	54C4	5485 BF
001:	802E	0000	2120	4088 40
002:	2E00	01ED	1D10	0000 C1
003:	F710	0000	0020	0000 BD
004:	0360	00DB	D414	2574 D4
005:	94E4	D11F	F31C	1961 C9
006:	4000	DB10	B1FE	74F2 F2
007:	14BA	E81E	789F	14B3 C8
008:	1F70	E629	6590	8FA5 5B
009:	CE01	1BD7	0016	0005 99
00A:	5000	8F87	1F08	FBC6 7D
00B:	315A	3310	69E6	131F 65
00C:	789F	214F	AE53	1080 31
00D:	E65C	C590	AEE3	1F70 D8
00E:	E6E1	498D	84A8	08D9 BD
00F:	1FB0	8DE6	A208	D394 48
010:	50			95

Michael Fiedler

Messx

46 Zeilen, 170 Bytes, HP-71, PRINTER

Als Naturwissenschaftler steht man häufig vor der Aufgabe, eine gemessene Größe, bzw. deren Fehler zu bestimmen. Das Programm „MESSX“ soll diese Aufgabe erleichtern.

- mittlerer Fehler des Mittelwerts
- relativer Fehler des Mittelwerts
- Standardabweichung
- Varianz

Der Testlauf demonstriert die Ausgabe an einem Beispiel.

Testlauf MESSX

Die in einem SDATA-File befindlichen Messwerte werden tabelliert; ist kein SDATA-File vorhanden, wird eines angelegt. Während des Ausdruckes werden die Meßwerte auf Ausreißer hin überprüft. Es wird ein zweites SDATA-File „CORRX“ mit den ausreißerfreien Werten erzeugt. Von diesen ausreißerbereinigten Werken werden dann folgende Größen berechnet:

- Anzahl der Meßwerte
- Summe der Meßwerte
- Mittelwert

Es wurde ein Ausreissertest nach Nalimoff durchgeführt. Ausreisser werden mit einer Sicherheit von 99% erkannt. Die folgende Auswertung ist ausreisserbereinigt

Anzahl der Messwerte: 10.
 Summe der Messwerte: 15.03500
 empirischer Mittelwert: 1.50350
 ± 0.00763 Ohm
 relativer Fehler: 0.50769%
 Standardabweichung: 0.02414
 Varianz: 0.00058

Auswertung		88/07/26 18:55:23
Messung	Widerstand/Ohm	
1	1.5370	
2	1.5210	
3	1.4984	
4	1.5020	
5	1.5310	
6	1.4871	
7	1.4725	
8	1.5000	
9	6.9970	Ausreisser!
10	1.5210	
11	1.4650	

```

MESSX Basic 170 Bytes
1 ! MESSX
2 ! Copyright (c) Thor Gehrman 1988
3 ! 4322 Sprockhoevel 2 Hobeuken 18
4 ! Messwertauswertung einer Variablen
5 ! Benötigt: ThinkJet
6 CALL MESSX
10 SUB MESSX @ DISP 'MESSX'
15 IMAGE 10XM2D,25XM5D.4D
20 IMAGE 10XM2D,25XM5D.4D15X"Ausreisser!"
25 DESTROY ALL @ OPTION BASE 1 @ STAT X(1)
30 INTEGER Z,N1,N2 @ DIM F$(8),B2$(10)
35 INPUT 'SDATA vorhanden? ':F$ @ IF UPRC$(F$)='N' THEN CALL SDATA
40 INPUT 'SDATA Name? ':F$ @ ASSIGN #1 TO F$ @ READ #1,0;N1 @ DIM M1(N1,2)
45 INPUT 'Beschrif. der Werte, Einheit? ':B1$,B2$ @ DISP 'Ruhe, ich arbeite!'
50 FOR Z=1 TO N1 @ READ #1,Z;M1(Z,1) @ NEXT Z
55 FOR Z=1 TO N1 @ ADD M1(Z,1) @ NEXT Z @ S1=SDEV(1) @ X1=MEAN(1)
60 PRINT TAB(30);'Auswertung';TAB(70);DATE$ @ PRINT TAB(70);TIME$ @ PRINT @ PRIN
T
65 PRINT TAB(10);'Messung';TAB(40);B1$;'/' ;B2$
70 CLSTAT @ DESTROY B1$,F$
75 READ A0,A1,A2,A3 @ INTEGER F @ F=1/(N1-1)
80 P9=EXP(A3*F^3+A2*F^2+A1*F+A0)
85 FOR Z=1 TO N1 @ M1(Z,2)=ABS(M1(Z,1)-X1)/S1*SQR(N1/(N1-1)) @ NEXT Z
90 DIM M2(N1,1) @ DESTROY A0,A1,A2,A3,S1,X1,F @ INTEGER N2 @ N2=0
95 FOR Z=1 TO N1
100 IF M1(Z,2)<P9 THEN PRINT USING 15;Z,M1(Z,1) @ N2=N2+1 @ M2(N2,1)=M1(Z,1) @ G
OTO 'E'
105 IF P9<=M1(Z,2) THEN 'AUS'
110 'AUS': PRINT USING 20;Z,M1(Z,1)
115 'E': NEXT Z
120 DIM M2(N2,1) @ DESTROY M1,I,N1 @ ASSIGN #1 TO *
125 CREATE SDATA CORRX,N2+1 @ ASSIGN #1 TO CORRX @ PRINT #1,0;N2 @ PRINT #1,1;M2
(,)
130 ASSIGN #1 TO * @ FOR Z=1 TO N2 @ ADD M2(Z,1) @ NEXT Z @ F=0
    
```

```

135 FOR Z=1 TO N2 @ F=F+(M2(Z,1)-MEAN(1))^2 @ NEXT Z @ F=SQR(F/(N2*(N2-1)))
140 DESTROY M2 @ ENG 4 @ PRINT @ PRINT
145 DIM S1$(60),S2$(25),S3$(80) @ READ S1$ @ READ S2$ @ READ S3$
150 PRINT S1$:CHR$(207);S2$ @ PRINT S3$ @ PRINT @ PRINT @ DESTROY S1$,S2$,S3$
155 FIX 0 @ PRINT 'Anzahl der Messwerte:':TAB(28);N2 @ FIX 5
160 PRINT 'Summe der Messwerte:':TAB(28);TOTAL(1)
165 PRINT 'empirischer Mittelwert:':TAB(28);MEAN(1);CHR$(254);F;B2$
170 PRINT 'relativer Fehler:':TAB(28);F/MEAN(1)*100;%'
175 PRINT 'Standartabweichung:':TAB(28);SDEV(1)
180 PRINT 'Varianz:':TAB(28);SDEV(1)^2
185 DATA 0.94718045,-0.966460915,0.23300208,0.152611296
190 DATA ' Es wurde ein Ausreissertest nach Nalimoff durchgef'
195 DATA 'hrt.Ausreisser werden mit'
200 DATA 'einer Sicherheit von 99% erkannt.Die folgende Auswertung ist ausreisse
rbereinigt'
205 DISP 'Alles erledigt' @ STD @ DESTROY ALL @ END SUB
    
```

```

SDATAX Basic 247 Bytes
5 ! Erzeugt SDATA-File fr Messwerte einer Variablen
10 CALL SDATAX @ SUB SDATAX
15 INPUT 'SDATA-Filename ? ':S$ @ INPUT 'Anzahl Werte ? ':N
20 CREATE SDATA S$,N+1 @ ASSIGN #1 TO S$ @ PRINT #1,0:N
25 FOR Z=1 TO N
30 DISP 'Messwert ':Z @ INPUT 'Wert ? ':X @ PRINT #1,Z:X
35 NEXT Z
40 ASSIGN #1 TO "" @ DESTROY ALL @ END SUB
    
```

Thor Gehrmann
4322 Sprochhövel 2 (3423)

Wir haben vieles,
manches hat nur H&G.



1040 STF/
SM 124

1.450,-

HP Vectra CS
20 MB Platte
Monitor
Herculeskarte
Tastatur
ser/par Adapter
3.298,-

HP 28 SD
398,-
HP DeskJet
1.990,-

MultiSync GS
Monitor
498,-

Amiga 500
999,-
PC 10 III
1.898,-

LQ 850
1.538,-

HANSEN & GIERATHS EDV- Vertriebsges. mbH
Münsterstr.1 5300 Bonn 1 Tel.: 0228/72908-0

Ihr Ansprechpartner bei H&G: Herr Endler

DISABLEX

... und noch ein Lexfile

Bei diesem Lex-File handelt es sich um eine besondere Spitzfindigkeit unserer französischen Freunde von SIG und beweist einen tollen Ideenreichtum gepaart mit genauen Systemkenntnissen des HP-71B.

Manchmal kommt es vor, daß man von mehreren im Computer vorhandenen Lex-Files den einen oder anderen davon zur Zeit nicht benötigt und ihn „weghaben“ will. Sei es, daß man sich davon eine Erhöhung der Rechengeschwindigkeit erhofft, oder daß man die Wirkung eines Lexfiles ausschalten will. Auch wäre es denkbar, zwei Lexfiles mit gleichen ID#- und Tokennummern zugleich im Speicher zu haben und abwechselnd den einen oder den anderen auszuprobieren. Bisher gab es dazu nur die Möglichkeit, den betreffenden File zur Gänze aus dem Speicher zu löschen. Mit DISABLEX jedoch geht es auch anders.

Das Programm stammt von Michel Martinet vom französischen SIG und wurde unter anderem bereits im SIG-Journal „JPC“ Nr. 24 veröffentlicht, sowie im CHHU-Chronicle V2N7p45. Bisher allerdings unter dem Namen DRIVELEX, den ich zum leichteren Merken in DISABLEX umbenannt habe. Die Filegröße ist 296 Bytes. Zwei Keywords sind vorhanden:

```
DISABLEX ID=E1 SIZE=296 Bytes
Messages!
Polls!
009 DISABLE Stmt 225009
0A ENABLE Stmt 225010
*
```

Die genauen Befehle lauten DISABLE s und ENABLE s. Dabei ist s ein beliebiger Stringausdruck, der den Namen des zu schaltenden Lex-Files darstellen muß. Wird nun ein Lex-File mit Hilfe des DISABLE-Befehls ausgeschaltet (z.B.: DISABLE CURLEX), so erscheint bei CAT CURLEX ab nun ein etwas geänderter Eintrag:

```
vorher: CURLEX LEX 90.....
nachher: CURLEX D-LEX 90.....
```

Es hat sich also ganz offensichtlich der Filetyp geändert! Tatsächlich ist es auch so, Michel Martinet hat einen neuen Filetyp geschaffen: D-LEX. Dabei werden nicht nur die Filetyp-Nibbles im Fileheader (Nibble Nr. 16 bis 19) geändert (LEX: 802E, D-LEX: FF00), sondern auch das Betriebssystem „erlernt“, daß es diesen neuen Filetyp gibt und benennt ihn bei CAT-Befehlen richtig. Im ausgeschalteten Zustand sind alle Schlüsselwörter des Lex-Files deaktiviert, bei Aufrufen über das Tastenfeld werden sie nicht erkannt, in BASIC Programmen stehen an den entsprechenden Stellen XWORD- und XFN-Nummern.

Durch den Aufruf z.B. von ENABLE CURLEX wird der ursprüngliche Zustand wieder hergestellt. Als Einschränkung ist anzumerken, daß das natürlich nur für Lex-Files im RAM gilt. Alle Files, die im ROM, oder in einem EPROM stehen, sind nicht veränderbar. Wenn ein D-LEX File von einem externen Medium geladen werden soll, so muß sich vorher DISABLEX im Speicher befinden, sonst wird der Ladeversuch mit „Filetype Error“ abgebrochen. Es empfiehlt sich aber ohnehin, Lex-Files nur in ihrer aktiven Form extern abzuspeichern, um derartige Komplikationen zu vermeiden.

Als sinnvolles Anwendungsbeispiel für DISABLEX habe ich schon DISABLE CURLEX angegeben, dies deshalb, da es nur so möglich ist, die VIEW-Taste wieder frei zu bekommen, falls sie vom Benutzer oder von einem anderen Programm (z.B. Texteditor) anders belegt ist (siehe eventuell erschienenen Artikel über CURLEX). Eine weitere Anwendung wäre in Verbindung mit MARGELEX oder KSPEEDLX sinnvoll.

Außerdem bringe ich heute noch zusätzlich ein kleines Lex-Programm aus der selben Quelle, Namens KBEEP. Dieses kleine Programmchen tut nichts anderes, als den HP-71B bei jedem Tastendruck kurz piepen zu lassen. Eine ganz nette Spielerei vielleicht, dennoch wird man bald von der erlösenden Möglichkeit:

```
„DISABLE KBEEP“
```

Gebrauch machen!

Hexdump-Listing: DISABLEX

(Zum Abtippen benötigen Sie ein Hex-Ladeprogramm, wie z.B. MAKEFILE aus Prisma 7/86!)

```
DISABLEX L ID#E1 296 Bytes
0123 4567 89AB CDEF ck
000: 4494 3514 24C4 5485 87
001: 802E 0000 2190 9078 BF
002: F220 01E9 0A00 0000 2B
003: F020 0E30 0860 0000 C1
004: 0BA0 00D1 10EC 000D 7F
005: D449 4351 424C 4549 AB
006: 0B54 E414 24C4 54A0 1E
007: 1FF1 0200 1204 02C4 C5
008: 5687 02C8 1105 4427 DF
009: 9667 5627 D20E AECF 8A
00A: F31D 2961 4000 D2A6 04
00B: E8A2 4000 7610 0005 CA
00C: 044D 2C45 4851 FF00 AD
00D: FF07 135A C0B4 4821 13
00E: 01E5 1003 2100 7260 DF
00F: D2A6 E8A2 C434 802E FD
010: 08A6 74D2 A6E4 A223 E5
011: 1007 F000 7630 3480 7E
012: 2E08 A2E1 D5D2 A6E8 8E
013: A691 D915 D384 08F2 B9
014: 1201 8D84 A803 1F34 3C
015: 3604 8F67 B90D 3CF8 94
016: F77F 9048 4AC2 B469 5D
```

```
017: C355 1371 086F D979 37
018: 0137 135C 2D73 51E9 DE
019: 0A02 515B 5912 2317 E9
01A: A137 8BF0 5137 58E3 AE
01B: 1938 D393 9033 101E 8B
01C: 8DA9 3903 1C34 7E17 7C
01D: 5143 2034 E300 0EA1 EE
01E: 33D6 1537 970F C137 8E
01F: 1746 79F2 0118 1351 BD
020: 7FD0 15B3 018F C9E3 F2
021: 0502 8772 2314 C8F8 C1
022: EC20 AEA8 0DF8 D624 1B
023: 508D 20F2 08D2 7130 7B
024: 8F95 7508 D003 50F C7
```

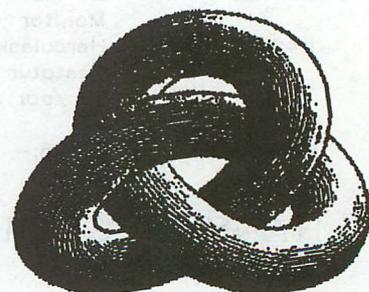
```
KBEEP L ID#E1 57 Bytes
0123 4567 89AB CDEF ck
000: B424 5454 0502 0202 AA
001: 802E 0063 1142 0178 72
002: 1500 01E0 0000 0000 24
003: FE00 0000 0800 001F 7F
004: F31B 1961 4000 DB10 87
005: B042 0348 E300 D732 A2
006: 0208 F04B E011 BD70 FD
007: 0 30
```

Michael Fiedler
Friedrichstraße 17
6070 Langen

BUG in der RENUMBER-Anweisung

Selbst in der neuen HP-71/ROM-Version 2CDCC – so teilt uns Clubmitglied Horst Debusmann mit – existiert noch der alte Fehler in der Renumber-Anweisung. Beispiel: Der Zeilenblock 300-390 soll auf 2000-2090 umnummeriert werden. Die Eingabe „RENUMBER 2000,10,300,390“ bewirkt nichts. Hewlett-Packard in Ratingen und in Frankfurt sind bereits informiert. Da der HP-71 jedoch ausläuft, ist nicht mehr mit einer Korrektur dieses Fehlers zu rechnen – wir müssen damit leben!

(ant)



Funktion:

Utility zum Vergleichen von Dateien. Im Gegensatz zu den üblichen Vergleichsprogrammen (z.B. COMP von DOS) vergleicht COMPARE nicht nur Byte für Byte, sondern erkennt auch, wenn z.B. ein Absatz eingefügt/gelöscht worden ist, etc. Dadurch wird es auch möglich Dateien unterschiedlicher Länge zu vergleichen.

Format:

[d:][p[ad]]COMPARE [d:][p[ad]]datei1 [d:][p[ad]]datei2 [/B][/W]

Hinweise:

COMPARE vergleicht die datei1 mit der datei2. Wenn Abweichungen gefunden werden, zeigt COMPARE die unterschiedlichen Bereiche bei der Dateien übereinander auf dem Bildschirm an. Sind die Dateien völlig verschieden, wird das von COMPARE gemeldet.

Wenn man keine Textdateien, sondern binäre Dateien miteinander vergleicht, sollte /B angegeben werden. Bei WordStar-Dateien sollte /W angegeben werden.

Funktion:

Mit CAPTURE kann man Screendumps auf Diskette/Platte machen, also quasi Bildschirmfotos in Dateien hinterlegen.

Format:

CAPTURE

Typ:

Intern Extern

Hinweise:

CAPTURE wird einmal angerufen, um es resident in den Speicher zu laden. Danach kann man mit <Alt-C> eine „Fotografie“ machen. Sie wird im aktuellen Verzeichnis unter dem Dateinamen SCREEN.000 abgelegt. Bei jedem weiteren „Foto“ wird die Extension um 1 inkrementiert (SCREEN.001, SCREEN.002, etc.).

Die von CAPTURE erzeugten Dateien sind kompatibel mit den Programmen SHOW, HELP und PAINT von dieser (bzw. der nächsten) Info.

Mit CAPTURE wurden z.B. die Fotos auf Info 28 von den Backupprogrammen gemacht.

Quelle:

PC-Magazine, New York

Funktion:

Residentes Programm zum Aufrufen von Hilfebildschirmen per Hot-Key.

Format:

[*d*]:[*pfad*]HELP *dateiname* [*dateiname* [...] [*m*][*hn*]/*r*]/*v*

Hinweise:

Beim ersten Aufruf installiert sich HELP resident und lädt die in der Kommandozeile angegebenen Dateien in den Speicher. Die Dateien müssen im Screenshot-Format sein. Solche Dateien werden z.B. von CAPTURE (auch auf INFO 28) und PAINT (INFO 30) erzeugt. Die Dateien SCREENS.* vom Backup-Test haben dieses Format.

Nach dem Laden läßt sich HELP mit <Alt-H> aktivieren. Dann kann man die geladenen Hilfebildschirme mit <PgDn> und <PgUp> durchgehen.

Wird HELP erneut mit einem Dateinamen aufgerufen, werden die zuerst geladenen Schirme durch die in der neuen Datei ersetzt. HELP reserviert normalerweise Platz für 4 Seiten. Mit dem Parameter /*Mn* kann das auf jeden beliebigen Wert *n* von 1-14 abgeändert werden. Mit /*Hn* wird der Hotkey auf <Alt-*n*> umgestellt. *n* kann jeder Buchstabe und jede Ziffer sein. Achtung: Für <ALT-Z> als Hotkey muß /Hy verwendet werden und umgekehrt.

/P bewirkt, das HELP sich gleich nach dem Aufruf selbst aktiviert.
/U deinstalliert HELP.

Funktion:

Editor für Bildschirmseiten, wie sie z.B. von HELP, SHOW, CAPTURE und dem neuen SCREENSAVE verwendet werden. Der Editor kann Linien zeichnen und ist voll farbfähig.

Format:

[*d*]:[*pfad*]PAINT [*d*]:[*pfad*]dateiname

Hinweise:

Wenn eine Datei angegeben ist, lädt Paint sie beim Start in den Speicher. Die Datei kann bis zu 14 Bildschirmseiten enthalten.

Mit <Pddn> und <PgUp> kann man in den Bildschirmseiten blättern.

<Ins> wechselt zwischen Überschriften und Einfügen.

 löscht das Zeichen über dem Cursor. Im Einfügen-Modus löscht <Backspace> das Zeichen links neben dem Cursor. <Home> und <End> bewegen den Cursor zu den Enden einer Zeile.

<F1> Zeigt die Belegung der Funktionstasten in der untersten Bildschirmzeile an.

<F2> Farbkarte: Auswahl zwischen 16 Vordergrund- und 8 Hintergrundfarben.

Monokarte: Auswahl zwischen Normal, Fett, Unterstrichen und Invers.

Mit <Ctrl-Pfeiltaste> können Bildschirmbereiche mit dem eingestellten Attribut getüncht werden.

<F3> Schaltet zwischen reinem Text- und Text+Attribut-Modus.

<F4> Wählt die Linienart. Mit <Alt-Pfeiltaste> kann man Linien zeichnen.

<F5> markiert die erste Ecke eines Blockes. Jetzt muß man den Cursor zur diagonal gegenüberliegenden Ecke bewegen und erneut <F5> drücken. Danach hat man die Wahl den markierten Block entweder zu löschen oder mit der eingestellten Farbe (Attribut) zu "tünchen".

<F6> Speichert die Datei.

<F7> Beendet das Programm.

Bei eingeschaltetem <Scroll-Lock> kann man über den Ziffernblock (=Cursorblock) mit <Alt-Ziffern> direkt ASCII Werte eingeben, wie auch unter DOS. Bei ausgeschaltetem <Scroll-Lock> (das ist der Normalfall) kann man <Alt-Pfeiltaste> Linien zeichnen.

Transform

Möglichkeit des filetype TRANSFORMing in Microsoft BASIC unter CP/M

Die meisten BASIC-Dialekte verfügen nicht über den komfortablen Befehl TRANSFORM...INTO des HP-71. Dieses Sprachelement ist allerdings im Zusammenhang mit der Tatsache zu sehen, daß der HP-71 beliebig viele Dateien – begrenzt nur durch den verfügbaren Platz – im Speicher halten kann.

Das in diesem Artikel beschriebene Verfahren ist für einen EPSON PX-8 nutzbar gemacht worden, einen akkubetriebenen CP/M-Computer (Hand-Held). Er weist einige technische Besonderheiten auf, die das Prinzip der Dateitypumwandlung jedoch nicht berühren. Grundsätzliche Probleme bei der Übertragung der Methode auf andere Rechnertypen (und Betriebssysteme (?), wie z. B. MSDOS,) sollten daher nicht auftreten.

Umwandlung von BASIC in TEXT

Die Programmdatei muß mit SAVE<Dateibesreibung>,A auf eine Diskette geschrieben werden. Wozu SAVE dient, muß ich wohl nicht erklären. Der Zusatz ,A bewirkt, daß die Programmdatei im ASCII-Format, d. h. als Textdatei, abgespeichert wird. Wenn für die extension keine Angabe gemacht wird, setzt BASIC automatisch .BAS ein. Was nicht im Handbuch steht: Diese Datei kann von der Diskette z.B. mit einem Textprogramm geholt und bearbeitet werden. Eventuell muß das Format der Textdatei an das vom Textprogramm erwartete angepaßt werden.

Umwandlung von TEXT in BASIC

Für die Umwandlung eines Textes in BASIC gibt es mehrere Möglichkeiten, je nachdem man sie programmgesteuert oder von der Befehlsebene aus durchführt. Ich will hier nur auf den interessantesten Fall eingehen: Die programmgesteuerte Umwandlung vor einem Unterprogrammaufruf.

Hierzu muß der Text nach meinen Erfahrungen in einer sequentiellen Datei abgelegt worden sein. Dies kann durch das Hauptprogramm geschehen, das später den Text umwandelt und als Unterprogramm aufruft. Der Begriff Unterprogramm ist hier im herkömmlichen Sinn zu verstehen, d. h. es handelt sich um einen Teil innerhalb eines Programms, der mit GOSUB aufgerufen wird. Die Umwandlung wird durch CHAIN MERGE <Dateibesreibung>, <Zeilennummer>, ALL, DELETE<Bereich> bewirkt, wobei der umgewandelte Text in das aufrufende Programm eingefügt wird. Im gewöhnlichen Gebrauch ruft CHAIN ein auf einer Disk stehendes Programm auf und übergibt ihm (anders als RUN) Variablen aus dem aufrufenden Programm. Durch MERGE wird erreicht, daß das aufgerufene Programm in das aufrufende Programm eingefügt wird; dabei werden Programmzeilen des aufrufenden Programms durch gleich nummerierte des aufgerufenen ersetzt. Die Programmausführung wird bei <Zeilennummer> wieder aufgenommen. ALL bedeutet, daß alle Variablen des aufrufenden Programms an das aufgerufene Programm übergeben werden. Durch DELETE werden nicht benötigte Zeilen des aufrufenden Programms gelöscht.

Der Benutzer muß für eine korrekte Umwandlung sicherstellen, daß der eingegebene Text der BASIC-Syntax entspricht. Ist dies nicht der Fall, kann es zum Abbruch des Programms kommen. (Der HP-71 wandelt eine fehlerhafte Programmzeile in eine Kommentarzeile um, die mit einem ? beginnt.)

Es ist leicht zu erkennen, daß die Funktion der BASIC-Befehle hier ein wenig spitzfindig ausgenutzt wird. Über Erfahrungsberichte anderer Clubmitglieder würde ich mich freuen.

Programmbeispiel GRAPHICS

Das folgende Programm erfragt vom Benutzer eine Funktion, die anschließend im Graphik-Modus (SCREEN 3) auf dem LCD des PX-8 (480*64 Punkte) geplottet wird. Die wesentlichen Details erklären sich selbst.

```

10 'graphics, (c) Sebastian von Borries 1988
20 ON ERROR GOTO 460
30 SCREEN ,,0: CLS
40 DEF FN DD(U,UMIN,UINC) = CINT((U-UMIN)/UINC)
50 PRINT"Geben Sie auf die Abfragen hin die entsprechenden Werte ein."
60 PRINT: PRINT"Ganzrationale Funktion? (J/N) ";
70 CONIN% = 0
80 WHILE CONIN% <> 74 AND CONIN% <> 78
90     CONIN% = ASC(INPUT$(1)) AND 223
100     WEND
110 PRINT CHR$(CONIN%)
115 IF CONIN%=74 GOTO 600
120 PRINT: PRINT"Geben Sie den Funktionsterm in der Form y=f(x) als BASIC-Unterprogramm ein.,"Benutzen Sie die Zeilen 501 bis 510. Der RETURN-Befehl wird automatisch erzeugt."
130 OPEN "0", #1, "A:FUNCTION.ASC"
140 FOR L=501 TO 510
150     I$=MID$(STR$(L),2)+" "
160     PRINT I$;: LINE INPUT F$
170     PRINT #1,I$;
180     IF F$ <> "" THEN PRINT #1,F$ ELSE PRINT #1,"RETURN": GOTO 200
190     NEXT L
200 CLOSE #1

```

```

210 CHAIN MERGE "A:FUNCTION.ASC", 220, ALL
220 KILL "A:FUNCTION.ASC"
225 GANZ%=0
230 PRINT: PRINT"Wenn die Zeichnung fertiggestellt ist, sind folgende Tasten akt
iv: ";TAB(11);"Z = Zeichenbereich neu festlegen";TAB(11);"E = Programm beenden"
235 PRINT: PRINT"Zeichenbereich auf der X-Achse:"
240 INPUT"X min= ",XMIN
250 INPUT"X max= ",XMAX
260 PRINT: PRINT"Zeichenbereich auf der Y-Achse:"
270 INPUT"Y min= ",YMIN
280 INPUT"Y max= ",YMAX
290 PRINT: PRINT"Ursprung des Achsenkreuzes:"
300 INPUT"X a= ",XA
310 INPUT"Y a= ",YA
320 XINC = (XMAX-XMIN)/479
330 YINC = (YMAX-YMIN)/63
340 SCREEN 3,,0: CLS
350 LINE (FN DD(XA,XMIN,XINC),63)-STEP(0,-63)
360 LINE STEP(-2,2)-STEP(2,-2): LINE-STEP(2,2)
370 LINE (0,63 - FN DD(YA,YMIN,YINC))-STEP(479,0)
380 LINE STEP(-2,-2)-STEP(2,2): LINE-STEP(-2,2)
390 FOR I = 0 TO 479
400   X = I*XINC + XMIN
410   IF GANZ%= 0 THEN GOSUB 500 ELSE GOSUB 700
420   PSET (FN DD(X,XMIN,XINC),63 - FN DD(Y,YMIN,YINC))
430   NEXT I
431 CONIN% = 0
432 WHILE CONIN% <> 69 AND CONIN% <> 90
433   CONIN% = ASC(INPUT$(1)) AND 223
434   WEND
435 IF CONIN%=90 GOTO 230
440 SCREEN 0,0,1: END
460 IF ERL = 420 AND ERR = 6 THEN RESUME 430 ELSE ON ERROR GOTO 0
500 'Hier wird das Unterprogramm eingefügt.
511 RETURN
600 PRINT: PRINT "Grad der Funktion:"
610 INPUT"n= ",N%
620 OPTION BASE 0
630 DIM A(N%)
640 FOR I=N% TO 0 STEP -1
650   PRINT"a(";MID$(STR$(I),2);:INPUT")= ",A(I)
660   NEXT I
670 GANZ%=1
680 GOTO 230
700 Y=0
710 FOR J= N% TO 0 STEP -1
720   Y=Y*X + A(J)
730   NEXT J
740 RETURN

```

Erklärung einiger Variablen in der Reihenfolge ihres Auftretens

FN DD(U,UMIN, UINC) berechnet eine Displaykoordinate
CONIN% ist die Variable der Konsoleneingabe
XINC, YINC sind die Schrittweitenvariablen der x- und y-Achse

Die Bedeutung der anderen Variablen wird aus der Programm-
liste hinreichend deutlich.

Die Zeilen 350 . . . 380 zeichnen das Achsenkreuz auf den
Schirm. Wird kein Achsenkreuz gewünscht, sind für X a und Y
a Werte einzugeben, die außerhalb des Bereiches von X min . . . X
max. bzw. Y min . . . Y max liegen.

Sebastian von Borries (2184)
Geniner Straße 35a
2400 Lübeck 1

M-BASIC und Taschenrechner HP97 im Vergleich

HP97 ↔ M Basic Vergleich

Beim Umseigen vom guten Taschenrechner, ich beziehe mich hier auf den HP97, (die HP41 Versionen dürften ähnlich sein) auf's Rechnen mit BASIC-Programmen, treten einige Probleme zutage, welche ich als wesentliche Mängel bei der praktischen Arbeit mit MBASIC, besonders im technischen Bereich, empfinde und welche ich im folgenden darstellen möchte:

1) Das Mischen von Handrechnungen und Programm-RUN's ist in BASIC nicht möglich, wenn man nicht besondere Tricks anwendet. Dies rührt daher weil erstens der Befehl INPUT nur einzelne Werte, aber keine mathematischen Ausdrücke annimmt und zweitens auch die Funktion VAL nicht imstande ist einen als String eingegebenen math. Ausdruck zu berechnen.

Der Taschenrechner hingegen steht bei jedem Programmstopp für die Berechnung von Eingabedaten voll zur Verfügung. Erforderliche Eingabedaten können berechnet, am Druckstreifen dokumentiert, und beim Programmablauf an die Variablen (= Speicherplätze) übergeben werden.

Ganz ähnlich steht es mit der eventuell erforderlichen Weiterverarbeitung von Rechenergebnissen. Auch hier ist der Taschenrechner überlegen.

2) Die Variablenamen stehen während des RUN's weder für die Eingabe noch für die Ausgabe zur Verfügung. Dadurch hat der Eingabe-Befehl meist die Form INPUT "E";E und der Druckbefehl die Form PRINT "A" = "A. Das heißt der Variablenname muß als String wiederholt werden. (Auch dann, wenn man eigentlich keinen Kommentar zu den Ergebnissen schreiben möchte!)

Beim Taschenrechner können die Speicher-Variablen mit PRINT ERG und die Stack-Variablen mit PRINT STACK, jeweils mit ihrem Namen (= Speicherplatz) ausgedruckt werden.

4) In BASIC kann kein Ausgabeformat allgemein definiert werden. Dieses muß vielmehr bei jeder Ausgabe mit PRINT USING, recht aufwendig erzeugt werden.

Beim HP ist das Anzeigeformat durch die Tasten FIX, SCI, ENG, DSP wählbar.

3) Die Abspeicherung und das Lesen von Daten ist in BASIC mit OPEN, PRINT#, etc. verhältnismäßig aufwendig. Vor allem weil alle Variablen, welche gespeichert werden sollen einzeln aufgeführt werden müssen.

Beim HP: Tasten WRITE DATA drücken, bzw. diesen Befehl im Programm verwenden, Magnetkarte durchschieben, fertig. Zum Laden der Daten: Einfach Magnetkarte durchschieben.

5) BASIC verlangt Winkel im Bogenmaß. Eine Voreinstellung für Alt- oder Neugrad ist nicht möglich.

Zu den oben angeführten Vorteilen des HP Taschenrechners kommen natürlich noch die Vorteile die die UPN-Logik bietet: Die ENTER-Taste ersetzt nämlich zwei Klammern und das Gleichheitszeichen. Die Logik ist natürlich Geschmacksache. Persönlich finde ich es schade, daß UPN nicht zu einer höheren Sprache weiterentwickelt wurde.

Um die oben angeführten Nachteile von BASIC besser darzustellen habe ich eine Liste von zusätzlichen und geänderten Befehlen, wie ich sie mir wünschen würde zusammengestellt:

1) VEREINBARUNGSTEIL:

DEFFIX, DEFSCI, DEFENG Befehle zur Steuerung des Druckformates
DEFDSP(n) Funktion wie beim HP

DEFDEG, DEF RAD, DEFGRD zur Änderung des Argumentes bei den Winkelfunktionen

2) EINGABE:

INPUT EVAR zeigt am Schirm EVAR?, anstatt nur ? erlaubt Eingabe eines math. Ausdruckes

INPUT EVAR=Wert zeigt EVAR-Wert?, mit RTN wird der Standardwert aus dem Progr. übernommen

statt
i INPUT "EVAR (RTN=Wert)";EVAR
j IF EVAR=0 THEN:EVAR=Wert

LINPUT EVAR wie INPUT aber Echo zum Drucker: EVAR = math. Ausdruck
= Wert bzw. EVAR = Wert

3) AUSGABE:

PRINT AVAR zeigt AVAR = Wert
LPRINT AVAR druckt AVAR = Wert
BPRINT AVAR zeigt und druckt AVAR = Wert
(BOTHPRINT)

NB: Wer mit Wordstar editiert kann die Ausgabeblocks mittels Blockkommando verdoppeln, mit ^QA die PRINT's in LPRINT's verwandeln und muß dann "nur noch" die Zeilennummern ändern.

PRINTALL, LPRINTALL zeigt bzw. druckt alle Variablennamen und ihren momentanen Wert

Zu diesem Befehl wären dann noch einige Parameter wünschenswert.

4) ZUM UND VOM MASSENSPEICHER:

OPENPRINTALL "Dateiname" Speichert alle Variablen und
OPENPRINTALL DAT\$ ihren zugehörigen Wert in der genannten Datei und schließt diese ab

OPENINPUTALL DAT\$ liest alle in der Datei DAT\$ gespeicherten Variablen

So einfach könnte das sein, dutzende von FOR . . . NEXT und PRINT# wären damit überflüssig.

Alle oben angeführten Befehle sind beim HP-Taschenrechner zumindest rudimentär vorhanden. Die Vorteile von BASIC, vor allem die durch die Strings, sind hier bewußt außer acht gelassen, denn ich will ja vor allem einige der Nachteile aufzeigen.

Weitere Vorteile bietet der Taschenrechner bei den Funktionen. Hier sind sogar die meisten nicht programmierbaren Rechner dem BASIC-Interpreter überlegen. Die folgende Vergleichstabelle zeigt das deutlich:

FUNKTIONEN IM VERGLEICH

Einige weitere weniger wichtige fehlen ebenfalls.

HP 97	MBASIC
Y	
ABS	ABS
TAN-1	ATN
fehlt: X ENTER .5 + INT	CINT
COS	COS
e	EXP
fehlt	FIX
INT	INT
LN	LOG
x ENTER ABS:	SGN
SIN	SIN
V	SQR
TAN	TAN
PI	fehlt: 4*ATN(1)
RND	fehlt
LOG	fehlt
10	fehlt
>H.MS	fehlt
H.MS>	fehlt
SIN-1	fehlt
COS-1	fehlt

Alle Funktionen mit zwei Argumenten fehlen:

>P (x,y): wandelt in Polarkoordinaten

>R (x,y): wandelt in Rechteckkoordinaten

SUM+ (x,y): z.B. zum Berechnen von Schwerpunkten praktisch etc.

Mit obigem hoffe ich neben den theoretischen Betrachtungen auch einige Hilfe für's Übersetzen von der UPNotation nach BASIC und umgekehrt gegeben zu haben.

P.S.: Wer kennt sich mit der Datenübertragung zwischen MS-DOS Rechner und 0-1 aus?

Dr. Ing. Hans Torggler
I 39100 Bozen
Verdiplatz 12

HP41 Emulator

Unser schweizer Clubmitglied Armin Jakob hat in Kalifornien ein neues Softwarepaket für alle Doppelcomputerbesitzer mit der Kombination HP41 <-> MS-DOS Rechner entdeckt, daß ich an dieser Stelle kurz erwähnen möchte, für genauere Informationen wendet Euch bitte an den Besitzer selbst, die Adresse steht unter diesem Artikel, eine dreiseitige Produktbeschreibung (in Englisch) ist gegen einen SAFU (selbstadressierten Freiumschlag) in der Redaktion bei mir erhältlich.

„The Fortyone“ ist ein HP41-Emulator für MS-DOS Rechner; die Emulation als solche scheint recht gelungen, wie Armin Jakob berichtete.

Daß Synthetik nicht unterstützt wird kann man sich wohl denken. . .

Eine besondere Stärke dieses Paketes besteht in dem Austausch! von Program-

men in beiden Richtungen, ebenso wird der Datenaustausch zwischen den beiden Rechnern nach Kräften unterstützt.

Der HP41 kann somit als mobiles Datenerfassungsterminal verwendet werden, dann werden im Labor (oder der guten Stube) die gesammelten Daten in den PC übertragen und unter dem Emulator oder anderen Softwarepaketen ausgewertet, z.B. von Lotus 1-2-3. Für den Datenaustausch benötigt man allerdings die HP-IL Karte für den PC und natürlich ein HP-IL Modul im HP41, das Extendet I/O Modul und eventuell noch das X-Funktion Modul oder einen HP41CX, der enthält dieses ja schon.

Für Leute, denen das Programmieren auf der kleinen Rechnertastatur zu mühsam ist gibt es einen Crosscompiler, der ASCII-Texte in ausführbare bzw. herunterladbare Programmdateien übersetzt.

Für all diese Leistungen ist der Preis von 115,- US\$ für den Emulator und 100,- US\$ für den Compiler und die Filetransferutility doch recht angemessen.

Zusätzlich bietet die Firma STRAIGHT-FORWARD noch fertige Programmsammlungen an, die von den besseren Ausgabemöglichkeiten des PCs vollen Gebrauch machen; diese Pakete basieren auf den Standart HP-Programmsammlungen wie z.B. Statistik, höhere Mathematik, Elektrotechnik etc.

Der Preis pro Paket beträgt 45,- US\$.

Martin Meyer (1000)

Besitzer:

Armin Jakob
Obere Felsenstraße 10
CH-9000 St. Gallen
Schweiz

Speed up HP41CX

neue Ergebnisse

In dem Bericht 88.2.44 Speed Up stand daß der neue Rechner, rundes Display, bis auf das 3 fache seiner Grundgeschwindigkeit zu bringen ist. Da ich einen solchen Rechner mein Eigen nenne und der mir mit schon der 1.6 fachen Geschwindigkeit immer noch zu langsam war, begann ich mit ihm zu experimentieren.

Das Höchste, was ich aus meinen HP-41CX herausholte, lag etwa beim 3.8 bis 4 fachen der Grundrechengeschwindigkeit. Dies erreichte ich dadurch, daß ich den Kondensator ganz herausnahm, also ohne Kondensator arbeiten ließ.

Mein ADVANTAGE PAC machte diese Geschwindigkeit ohne Störung mit. Dazu benutzte ich das Programm MATRX mit Real-Zahlen kleiner 10. Der Rechner brauchte zur Lösung des linearen Gleichungssystems weniger als 2 Sekunden, oder für das SOLVE Beispiel aus dem ADVANTAGE PAC Buch etwa 3 Sekunden, ein paar Zehntel Sekunden mehr, üblich sind da ca. 12 Sekunden.

Zu meiner großen Enttäuschung arbeitete meine RSU von ERAMO nicht mehr korrekt. Bei der 3-fachen Geschwindigkeit arbeitet sie jedoch wieder ohne Feh-

ler.

Diese Geschwindigkeit habe ich jetzt beibehalten; dies geschieht jetzt mit einem 10pF Stiroflex Kondensator.

Ich habe noch einen Schalter eingebaut, womit ich auf die 2 fache Geschwindigkeit herunterschalten kann, dazu legte ich einen 22pF Kondensator parallel. Ich brauche nicht auf die Grundfrequenz zurück, da ich keinen Kartenleser habe.

Markus Eißner
Friedrich-Ebert-Str. 93
6106 Erzhausen



Flexibles Datenbanksystem

HP-41CV, 1 X-F, 1 X-M, TIME, IL, PRINTER, TAPE

Programm	Zeilen	Bytes	SIZE
BIGDAT	513	1113	102
CODE	74	161	102
CHANGE	47	105	102
MUSEL	97	182	102
LETTER	148	308	102

Flexibles Datenbanksystem

Vor einigen Jahren habe ich für den HP41 ein flexibles Datenbanksystem entwickelt. Dieses System möchte ich an dieser Stelle vorstellen.

Wer keinen Barcodeleser besitzt, der kann bei mir gegen Einsendung einer formatierten Kassette und eines frankierten Rückumschlages eine Kopie erhalten; darin enthalten ist auch eine Modulversion und die Dateibearbeitung BIGNUM, die ich in einer der nächsten PRISMA-Ausgaben vorstellen möchte.

Dateiverwaltung BIGDAT

Mit der Dateiverwaltung BIGDAT ist es möglich, kleinere Dateien auf einem HP-41CV zu erstellen und zu verwalten. Das hierzu benötigte System besteht aus folgenden Geräten und Modulen:

- HP-41CV oder HP-41C + Quad-Ram
- X-Function Modul
- X-Memory Modul
- Digitalcassettenlaufwerk
- Thermodrucker
- eventuell Time Modul

Mit dieser Systemkonfiguration können je nach Anzahl von Attributen, 700 bis 1500 Einträge verwaltet werden. Die max. Anzahl von Attributen je Eintrag beträgt zehn. Ein Adresseneintrag hätte z.B. folgende Attribute:

- Name
- Vorname
- Straße
- Wohnort
- Telefonnummer

Mit der Dateiverwaltung BIGDAT ist es nun möglich, die angegebenen Daten zu editieren und zu selektieren.

Weiterhin ist es möglich, jedem Eintrag einen Code zuzuordnen, der ein schnelles Auffinden von Einträgen ermöglicht.

Für die Ausgabe stehen drei verschiedene Arten zur Verfügung:

1. die gesamte Datei in Tabellenform
2. die Einträge untereinander
3. vier freiwählbare Attribute im Querformat (Adressenaufkleber, Karteikarten)

Der Rechner tritt mit dem Benutzer in einen Dialog. Teilweise können die Programme als Unterprogramme aufgerufen werden.

Im weiteren werden nun die einzelnen Programme in Funktion und Handhabung beschrieben.

Bei allen Programmen wird der Drucker in der Stellung MAN betrieben.

Wird eine Eingabe verlangt, können

maximal 24 Zeichen eingegeben werden. Wird diese Zahl erreicht oder überschritten, so ertönt ein akustisches Signal (siehe Bedienungs- und Programmierhandbuch HP-41C, Seite 40).

1. BIGDAT oder LBL a

Das Programm BIGDAT verwaltet die Datei:

1. Es erstellt Dateien
2. Es lädt den Rechner mit den nötigen Daten

Beim Aufruf von BIGDAT tritt nun der Rechner mit dem Benutzer in einen Dialog. Fragen des Rechners werden mit (R/S) quittiert. Das gesamte Programm ist so aufgebaut, daß Fehlbedienung keinen negativen Einfluß auf die gespeicherten Daten nimmt.

Im Folgenden werden die Fragen erläutert:

DANAME=

Hier wird der Name der zu bearbeitenden Datei eingegeben. Die angegebene Datei wird auf dem Speichermedium gesucht und die Daten, die zur Verwaltung nötig sind, in den Rechner geladen. Danach kann die Datei durch INDAT, PRDAT usw. bearbeitet werden.

Wird die angegebene Datei nicht gefunden, so wird eine Datei mit dem angegebenen Namen erstellt. Hierbei erfolgt die Frage:

ATTRIBUTE=

Hier wird der Name eines Attributes erfragt, z.B. NAME, VORNAME usw.

Dieser Aufruf wird solange wiederholt, bis er entweder ohne Eingabe mit (R/S) quittiert wird, oder die max. Anzahl von zehn Attributen erreicht ist.

Als Attribute sind alle Worte beliebiger Länge erlaubt. Eine Ausnahme bildet CODE, diesem Wort ist eine Sonderfunktion zugeordnet, die unter dem Abschnitt CODE genauer beschrieben ist.

Der weitere Ablauf ist unter INDAT erklärt, dieses Programm wird nun automatisch aufgerufen.

Das Programm BIGDAT muß vor jeder Neubenutzung der Dateiverwaltung aufgerufen werden, weil dadurch der Rechner initialisiert wird. Ohne den Aufruf kommt es zur Fehlbedienung. BIGDAT muß nicht aufgerufen werden, wenn der Rechner lediglich aus- und eingeschaltet wird.

Das Programm BIGDAT ändert die Speicheraufteilung in Size 102 um und löscht automatisch das X-Function-Modul.

2. INDAT oder LBL B

Das Programm INDAT nimmt Einträge in die Datei vor. Neue Einträge werden an den letzten angehängt. Wenn mehrere Einträge gelöscht wurden, entstehen Leerstellen. Diese können dadurch wieder gefüllt werden, daß vor Aufruf von INDAT Flag 04 gesetzt wird.

Das Programm INDAT fragt die Attribute, die der Datei zugeordnet sind nacheinander ab. Hierbei werden nur die ersten sechs Buchstaben des Attributs angezeigt.

Ist keine weitere Eingabe erwünscht, wird der Eingabemodus durch Betätigen von (R/S) ohne vorherige Eingabe abgebrochen und die Datei abgespeichert.

Wenn eine gewisse Anzahl von Einträgen eingegeben wurde, erscheint die Meldung END OF DAT. Nun wird vom Rechner die im X-Function/Memory befindliche Teildatei auf das Speichermedium übertragen. Danach wird eine neue Teildatei erstellt und die Eingabe kann fortgesetzt werden.

Das erstellen von Teildateien ist notwendig, da im X-Function/Memory nur ein Speicherplatz von ca. 2000 Byte zur Verfügung steht. Die Teildateien haben jedoch auch den Vorteil, daß man für sie kürzere Zugriffszeiten benötigt. Die Teildateien werden automatisch durchnummeriert und bei einem Ausdruck mit PRDAT oder SELECT wird die Nummer dem Dateinamen vorangestellt.

Das Programm INDAT kann im STAND-BY betrieben werden.

3. PRDAT oder LBL A

Mit PRDAT ist es möglich den gesamten Inhalt einer Datei auszudrucken. Dies geschieht in Tabellenform, d.h. alle Einträge, die einem Attribut zugeordnet sind, werden untereinander gedruckt. Es werden jeweils die Teildateien mit ihrem Namen gedruckt. Durch nebeneinanderlegen der einzelnen Attributspalten kann eine DIN A5 Tabelle erstellt werden.

Nach der Korrektur eines Eintrages ist es oft wünschenswert, einzelne Teildateien neu auszudrucken, dies ist durch setzen des Flags 04 möglich. Hierbei fragt der Rechner nach DATNR?, nach Eingabe der Nummer einer Teildatei wird diese ausgedruckt.

Ist ein Time Modul in den Rechner eingesetzt, so wird unter der Überschrift einer Teildatei auch Datum und Uhrzeit gedruckt. Dabei wird das Time Modul in den Modus DMY gesetzt.

Das Programm PRDAT kann auch als Unterprogramm aufgerufen werden. Hierzu muß vor Aufruf Flag 04 und Flag 03 gesetzt werden und im X-Register muß die Nummer der auszudruckenden Datei stehen.

Beispiel: Ausdruck der ersten Teildatei als Unterprogramm.

```

:
:
:
SF 04
SF 03
1
XEQ'PRDAT
:
:

```

4. EDDAT oder LBL E

Durch das Programm EDDAT ist es möglich, den Inhalt der Datei zu verändern bzw. zu korrigieren.

Nach Aufruf von EDDAT erscheinen folgende Fragen:

DATNR?

Der Rechner verlangt die Nummer der zu editierenden Teildatei. Diese wird dann in das X-Function/Memory geladen.

STRING=

Hier ist der String einzugeben, der verbessert werden soll. (s. Beispiel, vgl. POSFL im Benutzerhandbuch des X-Function Moduls)

INS ST=

Hier ist der String einzugeben, der eingefügt werden soll. Ist dies nicht erwünscht, so wird mit (R/S) quittiert (s. Beispiel, vgl. INSCHR im Benutzerhandbuch des X-Function Moduls).

DEL?

Der Rechner verlangt nach der Anzahl der zu löschenden Buchstaben hinter dem neuen String. Soll kein Buchstabe gelöscht werden, so wird mit (R/S) quittiert (s. Beispiel, vgl. DELCHR).

Nun wird die Datei hinter der editierten Stelle weiter nach dem angegebenen String durchsucht, wird ein solcher nicht gefunden, so erscheint die Meldung:

ST ERR

Der Edit-Modus kann nun durch die Tastenfolge (SST) (R/S) abgebrochen werden, dann wird die editierte Datei auf das Speichermedium übertragen. Wird nur (R/S) gedrückt, so kann die Teildatei nach einem anderen String durchsucht werden d.h. Fortsetzung bei der Frage STRING =.

Ist der gleiche String mehrmals vorhanden, so wird die Frage INS ST= und DEL? sooft wiederholt, bis sämtliche gleichlautenden Strings gefunden worden sind. Dann erscheint ebenfalls die Meldung ST ERR.

Bei gesetztem Flag 04 wird die Teildatei nur einmal nach dem angegebenen String durchsucht und nach der Verbesserung sofort auf das Speichermedium übertragen.

Beispiel:

In der Teildatei 3ADRESS befinden sich zwei falsche Strings: ZIEZUUNG UND MONNTAG !
Zur Verbesserung sind nun folgende Schritte notwendig:

1. Aufruf von EDDAT
2. Eingabe der Teildateinummer: DATNR? (3) (R/S)
3. Eingabe des zu suchenden Strings: STRING = IEZUUNG
4. Eingabe des einzufügenden Strings: INS ST= EIT
Der String lautet nun: ZEITIEZUUNG
5. Eingabe der zu löschenden Buchstaben: DEL? (4) (R/S)
Der verbesserte String lautet nun: ZEITUNG
6. Wenn der String IEZUUNG nicht nocheinmal in der Teildatei vorhanden ist, erscheint die Meldung: ST ERR. Wird nun die Tastenfolge (SST) (R/S) gedrückt, wird die Teildatei auf das Medium übertragen. Da aber noch der String MONNTAG zu verbessern ist wird die Meldung ST ERR nur mit (R/S) quittiert.
7. Eingabe des nächsten Strings: STRING= NNTAG
8. Die Eingabe des einzufügenden Strings wird sofort mit (R/S) quittiert, da nur der Buchstabe N gelöscht werden soll.
9. Eingabe der zu löschenden Buchstaben: DEL? (1) (R/S)

10. Kommt der String NNTAG ein zweitesmal in der Teildatei vor, erfolgt wieder die Frage INS ST =, nun wird wieder so verfahren wie unter Punkt 8. oder 4..
11. Wird nach ST ERR nun mit (SST) (R/S) quittiert, wird die vollständig editierte Datei auf das Speichermedium übertragen. Bei gesetztem Flag 04 wäre dies bereits unter Punkt 6. geschehen, ohne daß die Meldung ST ERR erschienen wäre.

Wird unter DEL? eine negative Nummer eingegeben, wird kein Buchstabe gelöscht und es erfolgt sofort ein Sprung zu der Frage STRING=. Damit ist die Möglichkeit gegeben, beim Editieren von Dateien schnell zurück zu der Frage STRING= zu gelangen.

5. SELECT oder LBL D

Durch dieses Programm ist es möglich, einzelne Einträge nach bestimmten Kriterien zu selektieren. Bei dem Aufruf SELECT wird nun die gesamte Datei nach dem nach STRING= eingegebenen String durchsucht. Wenn ein Eintrag mit dem angegebenen String gefunden wird, wird dieser in der Reihenfolge der angegebenen Attribute ausgedruckt. Bei gesetztem Flag 04 wird der erste Eintrag mit dem angegebenen String ausgedruckt und danach der Suchmodus abgebrochen.

Bei gesetztem Flag 03 ist es möglich, das Programm SELECT als Unterprogramm zu benutzen. Hierbei muß dann der zu selektierende String vor Aufruf des Programms SELECT ins ALPHA Register geschrieben werden. Das Flag 03 wird durch das Programm SELECT automatisch gelöscht.

Hierdurch ist eine Möglichkeit geschaffen, eine Datei zu selektieren, ohne daß der Benutzer anwesend sein muß, z.B. durch folgendes Programm:

```

:
'XXXX'
SF 03
XEQ'SELECT
'YYYY'
SF03
XEQ'SELECT
:

```

Die Wirkung von Flag 04 ist hierbei nicht aufgehoben.

Ist ein Time Modul vorhanden, so wird auch Datum und Uhrzeit des Selektivs mit ausgedruckt.

6. DELETE oder LBL d

Das Programm DELETE arbeitet genauso wie SELECT mit dem Unterschied, daß die selektierten Einträge gelöscht werden und die Teildateien dann neu auf das Speichermedium übertragen werden.

7. PRATT oder LBL F

Mit PRATT ist es möglich, sich die Attribute einer Datei auszudrucken.

Beispiel:

```

XEQ „PRATT“
R01=“NAME”
R02=“VORNAM”
R03=“STRASS”
R04=“WOHNOR”
R05=“TELEFO”

```

8. Optionen

Einige Programme, die nicht unbedingt zur Dateiverwaltung benötigt werden, sind so gestaltet, daß sie lediglich bei Bedarf in den Hauptspeicher des Rechners geladen werden.

Dies geschieht automatisch bei Aufruf des Programms über die lokalen Marken (Labels).

Für den Benutzer besteht kein Unterschied, ob das Programm im Rechner steht oder nicht.

Es muß lediglich darauf geachtet werden, daß nach dem Laden von BIGDAT ein END an das Programm angehängt wird, da es sonst bei Aufruf dieser Programme zu einem Überschreiben des Hauptprogrammes kommt und damit auch die Zusatzprogramme nicht mehr lauffähig sind.

Im folgenden sind nun die Zusatzprogramme mit ihrer Funktionsweise erklärt.

8.1 CODE oder LBL C

Mit CODE ist eine weitere Möglichkeit geschaffen, Einträge zu selektieren oder zu verbessern. Damit das Programm Code überhaupt benutzt werden kann, ist es notwendig, beim Erstellen einer Datei das Wort CODE als ein Attribut anzugeben. Bei der Eingabe der Daten erscheint dann dieses Attribut nicht mehr, weil es automatisch von dem Rechner mit den nötigen Daten belegt wird. Diese Daten bestehen je Eintrag aus einer sechs-stelligen Nummer, die ersten beiden Ziffern geben die Teildatei an, in der sich der Eintrag befindet, den anderen vier Stellen ist eine laufende Nummer von 1000 bis 9999 zugeordnet. Ist nun eine Datei mit dem Attribut CODE versehen, kann sie auch mit dem Programm CODE bearbeitet werden. Damit ist es möglich sehr schnell einzelne Einträge zu selektieren oder zu editieren.

Editieren mit dem Programm CODE

Zuerst wird das Programm CODE aufgerufen, die erste Frage lautet CODE?. Hier ist nun die sechs-stellige Nummer einzugeben, die dem zu editierendem Eintrag zugeordnet ist. Diese Nummer wird bei PRDAT oder SELECT mit ausgedruckt.

Die nächste Frage lautet ATTRIBUTE=, hier ist das Attribut anzugeben, dessen Inhalt verändert werden soll.

Danach erscheint INS ST=, hier wird nun der neue String verlangt, der unter dem Attribut in dem angegebenen Eintrag abgespeichert werden soll.

Beispiel:

In einer Adressendatei hat sich die Telefonnummer von X verändert. Er hat die Codenummer 111011, das Attribut lautet: TELEFON.

Es wird nun folgendermaßen vorgegangen:

1. Aufruf von Programm CODE
2. Eingabe der Codenummer: 111011
3. Eingabe des Attributes: TELEFON
4. Eingabe der neuen Telefonnummer
Wird kein String angegeben, wird ein '-?' abgespeichert.

Selektieren mit dem Programm CODE.

Zuerst ist es notwendig, Flag 04 zu setzen, dann wird nach Aufruf von CODE und Eingabe der Codenummer der Eintrag mit dem angegebenen Code ausgedruckt; so kann das Programm dazu genutzt werden Verbesserungen zu überprüfen.

Wird eine Codenummer nicht gefunden, so erscheint die Meldung CODE ERR. Bei (R/S) wird das Programm CODE wieder gestartet.

Nach dem Selektieren mit dem Programm CODE erscheint ebenfalls die Meldung CODE ERR und CODE kann dann mit (R/S) wieder gestartet werden.

8.2 LETTER oder LBL c

Das Programm LETTER ist ein Zusatzprogramm zur Dateiverwaltung BIGDAT. Mit ihm ist es möglich, Einträge aus den Dateien zu selektieren und dann in selbstgewählter Reihenfolge quer auszudrucken.

Nach dem Aufruf von LETTER erscheint zuerst die Frage ATTRIBUTE=, hier werden nun die Attribute in der Reihenfolge angegeben, wie sie später ausgedruckt werden sollen. Es müssen immer vier Attribute angegeben werden. Sind weniger als vier Attribute vorhanden, müssen entsprechend andere zweimal angegeben werden. Mehr als vier Attribute können nicht ausgedruckt werden.

Wird ein Attribut falsch eingegeben, so erscheint die Meldung ATT ERR und die Eingabe des Attributs wird neu verlangt.

Nach Eingabe der Attribute verlangt der Rechner nun den String, wonach die Datei selektiert werden soll, mit STRING=.

Die gesamte Datei wird nun nach dem angegebenen String durchsucht und die

Einträge entsprechend der zuvor eingegebenen Reihenfolge quer ausgedruckt.

Bei gelöschtem Flag 04 wird der untere, bei gesetztem Flag 04 der obere String fett gedruckt.

Ist die Reihenfolge der Attribute einmal festgelegt, kann das Querdruckprogramm auch über PRTLET aufgerufen werden. Es wird jetzt nur noch nach dem zu selektierenden String gefragt.

Durch Setzen von Flag 03 ist auch hier eine Benutzung des Programms als Unterprogramm möglich.

Der zu selektierende String muß dann vor Aufruf des Programms PRTLET im ALPHA-Register gespeichert werden.

Wird das Programm LETTER aufgerufen, überprüft der Rechner, ob die Daten für die Querschrift vorhanden sind; ist dies nicht der Fall, so wird versucht die Daten vom Speichermedium zu lesen; ist dies auch nicht möglich, so werden die Daten über den Kartenleser mit der Meldung CARD verlangt. Wenn die Daten über den Kartenleser eingelesen wurden, werden sie auf das Speichermedium übertragen.

Das Programm LETTER benötigt mit BIGDAT den gesamten Speicherplatz eines HP-41CV. Deshalb ist es notwendig alle Tastenbelegungen und alle anderen Programme zu löschen.

8.3 CHANGE oder LBL e

Durch dieses Programm ist die Möglichkeit gegeben, gleichlautende Strings durch neue zu ersetzen. Dies hat den Vorteil, daß man für immer wiederkehrende Eintragungen Abkürzungen verwenden kann, die dann mit Hilfe von CHANGE in ihre endgültige Form verändert werden.

Nach Aufruf von CHANGE erscheint folgende Eingabeaufforderung:

OLDST=
Hier wird der String eingegeben, der durch einen neuen ersetzt werden soll (max. 12 Zeichen).
Die Anzahl von Zeichen, die bei OLDST eingegeben worden sind, werden bei der Umwandlung gelöscht.

NEWST=
Hier wird der neue String eingegeben (max. 18 Zeichen).

Bei gesetztem Flag 04 erscheint noch die Aufforderung:

DATNR?
Hierbei kann eine einzelne Teildatei angegeben werden; dadurch kann man umgehen, daß jeweils die ganze Datei nach einem zu ersetzenden String durchsucht wird.

Beispiel:
In einer Adressdatei ist oft der Wohnort gleich, z.B. 5000 KOELN.

Bei der Eingabe der Datei durch INDAT, kann man den Wohnort z.B. durch%K abkürzen.

Ist man mit der Eingabe fertig, besteht nun die Möglichkeit mit CHANGE den String %K durch 5000 KOELN zu ersetzen.

OLDST= %K
NEWST= 5000 KOELN

8.4 MUSEL oder LBL b

Durch dieses Programm ist die Möglichkeit gegeben, die gesamte Datei nach logisch miteinander verknüpften Strings zu durchsuchen.

Bei gelöschtem Flag 04 wird bei Aufruf des Programms MUSEL die UND-Verknüpfung aktiviert.

Die erste Eingabeaufforderung heißt:

STRING=
Hierbei wird der erste String eingegeben,

alle weitere Eingabeaufforderungen heißen:

AND=

Wird kein String eingegeben und mit R/S quittiert, wird die Datei nach den angegebenen Strings durchsucht und die Einträge ausgedruckt, in denen sich alle angegebenen Strings befinden.

Bei gesetztem Flag 04 wird die ODER-Verknüpfung aktiviert und die Eingabeaufforderung lautet hier

STRING=
für den ersten String und

OR=
für alle weiteren.

Wird diese Aufforderung ohne Eingabe mit R/S quittiert, so wird die Datei ebenfalls nach den angegebenen Strings durchsucht und die Einträge ausgedruckt, in denen sich jeweils einer der angegebenen Strings befindet. Befinden sich zwei angegebene Strings in einem Eintrag, so wird dieser auch zweimal ausgedruckt.

9. Unterprogramme der Dateiverwaltung

Unterprogramm LBL "A"

Erstellt im X-Func/Memory eine durch die Register 15 + 20 spezifizierte Teildatei und überträgt diese vom Speichermedium in das X-Func/Memory.

Unterprogramm LBL "B"

Überträgt bei gesetztem Flag 06 die durch die Register 15 + 20 spezifizierte Teildatei aus dem X-Func/Memory auf das Speichermedium und löscht die Teildatei im X-Func/Memory. Bei gelöschtem Flag 06 wird lediglich die Datei im X-Func/Memory gelöscht.

Unterprogramm LBL „C“ und LBL "D"

Durchsucht eine durch LBL "A" erstellte Teildatei nach dem String, der in den Registern 26 + 27 als Alphastring abgelegt ist.

Zuvor muß jedoch der Pointer des Arbeitsfiles im X-Func/Memory auf 0 gesetzt werden (0 SEEKPT).

Wird der angegebene String gefunden, so wird der Pointer des Arbeitsfiles auf das erste Attribut des Eintrags gesetzt. Bei gesetztem Flag 08 wird das Programm an dieser Stelle durch ein RTN abgebrochen.

Bei gelöschtem Flag 08 wird der gesamte Eintrag ausgedruckt. Ist dabei Flag 05 gesetzt, so wird dieser Eintrag auch gelöscht und Flag 06 gesetzt (siehe LBL "B").

Bei gesetztem Flag 04 wird der Wert im Register 15 zu einem INTEGER-Wert.

Der Arbeitsfile wird solange durchsucht, bis der angegebene String nicht mehr gefunden wird.

Das Unterprogramm LBL "D" kann nur mit gesetztem Flag 08 benutzt werden.

Es setzt den Arbeitspointer des X-Func/Memory auf den Beginn des Eintrages, der durch das X-Register spezifiziert ist.

Beispiel:

```
...
"BERNHARD"
POSFL
SF 08
XEQ "D"
...
```

Unterprogramm LBL "E"

Durch dieses Unterprogramm wird die Eingabe an den vorher im Alpha-Register abgelegten String im Display angehängt und die Länge des eingegebenen Strings im X-Register übergeben.

Die Eingabe wird, durch Angabe eines numerischen Zeichens im Alpha-Register vor Aufruf des Programms LBL "E", als numerische Eingabe im FIX-Format interpretiert.

Beispiel für die Benutzung der Unterprogramme.

Durchsuchen der Teildateien 1 bis 5 auf einen einzugebenden String:

```

"STRING"
XEQ "E"
X = 0?
RTN
ASTO 26
ASHF
ASTO 27
1,005
STO 15
LBL 01
XEQ "A"
0
SEEKPT
XEQ "C"
XEQ "B"
ISG 15
GTO 01
RTN
    
```

10. LBL H ... LBL J

Den lokalen Labeln H ... J sind den Programmnamen USER 1 ... USER 4 zugeordnet.

Damit ist die Möglichkeit gegeben, daß der Benutzer selber Programme erstellt, um die Datei entsprechend seinen Anforderungen zu beeinflussen.

Diese Programme werden genauso behandelt wie die schon vorhandenen Zusatzprogramme.

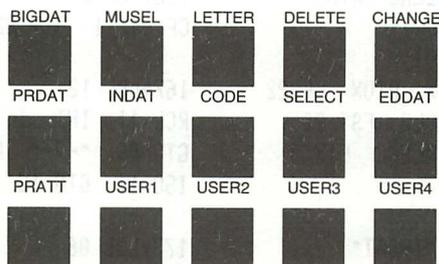
Zur Programmierung der USER-Programme seien folgende Informationen gegeben:

1. Das Programm muß einen Label USER1 ... USER4 enthalten.
2. Register, die von SIGDAT benutzt werden und ihre Bedeutung:

- Reg 00 = Flagstatus für das Unterprogramm LBL "E"
- Reg 01 → Reg 10 = Abkürzungen der Attribute
- Reg 12 = Anzahl der Attribute in der Form 1,0 mm
- Reg 15 = Momentan bearbeitete Datei
- Reg 26 + Reg 27 = Alpha-String für div. Unterprogramme
- Reg 20 = Dateiname
- Reg 21 = Codenummer
- Reg 22 = Anzahl der Dateien
- Reg 38 → Reg 101 = Zeichensatz für LETTER

12. Übersicht über die lokalen Marken

Damit mit der Dateiverwaltung BIGDAT besser gearbeitet werden kann, sind die Programme den lokalen Marken a ... e und A ... J zugeordnet. So kann der Aufruf eines Programms entweder über seinen Namen oder dem entsprechenden lokalen Label erfolgen.



11. Übersicht über die Funktion des Flag 04

	Flag 04 gesetzt	Flag 04 gelöscht
BIGDAT	Flag 04 wird gelöscht	
INDAT	Leere Records werden gesucht und durch Neueintrag gefüllt	Neueinträge werden an's Ende der Datei geladen
PRDAT	Nur die gewünschte Teildatei wird gedruckt	Die gesamte Datei wird gedruckt
EDDAT	Die angegebene Teildatei wird nur einmal durchsucht	Der Editmodus kann nur durch die Tastenfolge (SST) (R/S) beendet werden
DELETE	Der erste Eintrag mit dem angegebenen String wird gelöscht	Alle Einträge mit dem angegebenen String werden gelöscht
SELECT	Der erste Eintrag mit dem angegebenen String wird gesucht	Alle Einträge mit dem angegebenen String werden gesucht
CODE	Der Eintrag mit dem angegebenen Code wird selektiert	Der Eintrag mit dem angegebenen Code wird im angegebenen Attribut komplett erneuert
LETTER	Bei der Ausgabe wird der obere String fett gedruckt	Bei der Ausgabe wird der untere String fett gedruckt
CHANGE	Nur die gewünschte Teildatei wird bearbeitet	Die gesamte Datei wird bearbeitet
MUSEL	ODER-Verknüpfung	UND-Verknüpfung

13. Beispiele zu BIGDAT

```

1 BSP 1
09.11.83      13:48:53

TITEL      AUTOR      REG.NR.    STICHWORT
-----
AAAAAA     XXXXXX     123456     NNNNNN
BBBBBB     XXXXXX     234567     NNNNNN
CCCCCC     XX?XXX     345678     NNNNNN
DDDDDD     XXXXXX     456789     NNNNNN
EEEEEE     XXXXXX     567890     NNNNNN
FFFFFF     XXXXXX     678901     NNNNNN
GGGGGG     XXXXXX     -?         -?-
HHHHHH     XXXXXX     256983     NNNNNN
IIIIII     XXXXXX     369852     NNNNNN
JJJJJJ     XXXXXX     256987     NNNNNN
    
```

```

** SELECT **
09.11.83      13:55:47
1 BSP 1
    
```

```

JJJJJJ
XXXXXX
256987
NNNNNN

D D D D D D
X X X X X X
1 @ G + 0 0
Z Z Z Z Z Z
    
```

Beispiel für eine Buchdatei:

Der Eintrag mit dem Titel JJJJJJ wurde mit SELECT selektiert. Der Eintrag mit dem Titel AAAAAA wurde mit dem Programm LETTER erstellt. Reihenfolge der Attribute bei LETTER: TITEL, AUTOR, RFG.NR. STICHWORT.

Vor Aufruf von Letter wurde Flag 04 gesetzt.

```

01+LBL "CHANGE"
"OLDST" XEQ "E"
ASTO 26 ASHF ASTO 27
STO 14 "NEWST" XEQ "E"
ASTO 28 ASHF ASTO 29
ASHF ASTO 30 "DATNR?"
RCL 22 E3 / 1 +
FS?C 04 PROMPT STO 15
24+LBL 01
XEQ "A"
26+LBL 03
CLA ARCL 26 ARCL 27
POSFL X<0? GTO 02 CLA
ARCL 28 ARCL 29
ARCL 30 INSCHR RCL 14
DELCHR GTO 03
41+LBL 02
SF 06 XEQ "B" ISG 15
GTO 01 SF 25 END
    
```

SERIE 40

1BSP2

09.11.83 14:39:17

ANREDE	NAME	WOHNORT	STRASSE	TELEFON	CODE
HERR	B. SAALFELD	5000 KOELN 80	D.-MUELHEIMER-STR. 344	629619	#101000
FRAU	M. SAALFELD	5000 KOELN 80	D.-MUELHEIMER-STR. 344	629619	#101001
FIRMA	ABC	1000 BERLIN 41	GASSE 45	457896	#101002

HERR
B. SAALFELD
D.-MUELHEIMER-STR. 344

5000 KOELN 80

Beispiel für eine Adressendatei:

Bei der Erstellung der Datei, wurde ein Attribut mit CODE angegeben.

Der Tabellenausdruck erfolgte über PRDAT; Der Adressenausdruck über LETTER. Reihenfolge der Attribute: ANREDE, NAME, STRASSE, WOHN-ORT.

```

01+LBL "BIGDAT"          86+LBL "E"
02+LBL a                 "+=" CF 22 ANUM          FC? 04 GTO 01 ISG 15
, X<> c STO 64 X<> c      FS? 22 GTO 01 RCL 00    GTO T
102 PSIZE "ap+*"        X<> d AVIEW CLA STOP    162+LBL 01
RCL [ STO d DMY SF 25   X<> d ALENG RTN        CF 04 1 ST+ 22 GTO B
" a" ASTO 00 1.01
STO 11 "DANAME"         100+LBL 01
XEQ "E" ASTO 20 CLA     FIX IND X ATOX CF 22   167+LBL 12
ARCL 20 30 CRFLAS      PROMPT CLA FS? 22     RCL 11 INT 1 X=Y?
GETAS FC?C 25 GTO 01   ARCL X ALENG FIX 0    GTO 06 "-?-" APPREC
RTN                    RTN                    ISG 11 GTO 12

28+LBL 02
GETREC 35 POSA X=0?    111+LBL "INDAT"
GTO 20 ASTO IND 11     112+LBL B
ISG 11 GTO 02          RCL 22 STO 15 FC? 04
                        GTO T E3 / 1 +
                        STO 15

37+LBL 20
ANUM STO 22 3 AROT     122+LBL T
ANUM STO 21            PWRUP XEQ "A" FC?C 25
                        CREATE PWRDN

44+LBL 03
RCL 11 1 - INT E3      128+LBL 11
/ 1 + STO 12 CLA       SF 25 165 SEEKPT
ARCL 20 PURFL FS?C 04  FS?C 25 GTO 01 RCL 12
GTO B RTN              STO 11

60+LBL 01
SF 04 CREATE           136+LBL 05
                        "CODE" ASTO X
63+LBL 04              RCL IND 11 X=Y? GTO 20
"ATTRIBUTE" XEQ "E"    CLA ARCL IND 11
X=0? GTO 01           XEQ "E" X=0? GTO 12
ASTO IND 11 APPREC
ISG 11 GTO 04

72+LBL 01
"*" 1 STO 22 ARCL X    147+LBL 21
"+*" E3 ARCL X        APPREC ISG 11 GTO 05
STO 21 APPREC CLA     GTO 11
ARCL 20 SAVEAS GTO 03 152+LBL 01
                        "END OF DAT" CF 21
                        AVIEW SF 21 XEQ 06

245+LBL 08
FS? 31 XEQ 10 RCL 12  177+LBL 06
FRC E3 * STO 14 E5    SF 06 PWRUP XEQ "B"
/ ,9 + 1 - STO 13     CLA ARCL 20 30 CRFLAS
SF 21 DSE 16          GETAS "*" POSFL
                        DELREC ARCL 22 "+*"
                        ARCL 21 APPREC CLA
                        ARCL 20 SAVEAS PURFL
                        RTN

262+LBL 13
ADV RCL 13 1 +        198+LBL 20
STO 13 STO 11 RCL 14  RCL 15 INT 9 + E4
RCL 16 INT X=Y? RTN  * RCL 21 + "*"
CLA ARCL 20 SEEKPT    ARCL X 1 ST+ 21 E4
GETREC CF 22 ANUM     RCL 21 X=Y? E3
FS? 22 ATOX PRA       STO 21 GTO 21
ISG 16 "-----" ASTO L
ARCL L ARCL L ARCL L
PRA SF 25

291+LBL 14
RCL 11 INT CLA        304+LBL "EDDAT"
ARCL 15 ARCL 20       305+LBL E
SEEKPTA FC? 25 GTO 13 SF 06 "DATNR?" PROMPT
GETREC PRA ISG 11    STO 15 RCL 22 X<Y?
GTO 14                GTO E XEQ "A"

314+LBL X
"STRING" XEQ "E"      320+LBL 16
ASTO 26 ASHF ASTO 27 CLA ARCL 26 ARCL 27
                        POSFL X<0? GTO 09
                        "INS ST" XEQ "E" X=0?
                        INSCHR "DEL?" ,
                        PROMPT X<0? GTO X
                        X=0? DELCHR SF 25
                        RCLPT INT 1 + SEEKPT
                        FS?C 25 GTO 16

346+LBL 09
FS?C 04 GTO "B"      356+LBL "A"
"ST ERR" PROMPT FS? 51 SF 25 CLA ARCL 15
GTO "B" , SEEKPT    ARCL 20 330 CRFLAS
GTO X                GETAS RTN

365+LBL "B"
CLA ARCL 15 ARCL 20  373+LBL "DELETE"
FS?C 06 SAVEAS PURFL 374+LBL d
RTN                  SF 05

```

```

376+LBL "SELECT"
377+LBL D
FC? 03 "STRING"
FC?C 03 XEQ "E"
ASTO 26 ASHF ASTO 27
SF 12 "** SELECT"
FS? 05 "** DELETE"
"+ **" AVIEW CF 12
RCL 22 E3 / 1 +
STO 15 FS? 31 XEQ 10

400+LBL 07
1,01 STO 16 CLA
ARCL 15 ARCL 20 SF 12
PRA CF 12 FC? 07 ADV
XEQ "A" FC? 07 XEQ "C"
FS? 07 XEQ 08 XEQ "B"
ISG 15 GTO 07 CF 05
CLA ARCL 20 FS?C 07
PURFL RTN

425+LBL "C"
CLA ARCL 26 ARCL 27
POSFL X<0? RTN

432+LBL "D"
INT RCL 12 FRC E3 *
/ LASTX X<Y INT
X<Y * SEEKPT FS?C 08
RTN RCL 12 FS? 05
SF 06

450+LBL 19
GETREC PRA FS? 05
DELREC ISG X GTO 19
ADV FC?C 04 GTO "C"
RCL 15 INT .STO 15
GTO "C"

464+LBL 10
FIX 4 CLA DATE ADATE
"+ " TIME ATIME
FIX 0 PRA RTN

475+LBL "PRATT"
476+LBL F
RCL 12 PRREGX RTN

480+LBL C
"CODE" GTO 01

483+LBL e
"CHANGE" GTO 01

486+LBL c
"LETTER" GTO 01

489+LBL h
"MUSEL" GTO 01

492+LBL G
"USER1" GTO 01

495+LBL H
"USER2" GTO 01

498+LBL I
"USER3" GTO 01

501+LBL J
"USER4"

503+LBL 01
SF 25 ASTO Y XEQ IND Y
FC? 25 READP ASTO X
X=Y? XEQ IND X CLX
END

01+LBL "CODE"
"CODE?" FC?C 03 PROMPT
"#" ARCL X ASTO 26
ASHF ASTO 27 E4 /
INT 9 - STO 15
XEQ "A" FC? 25 GTO 01
"ATTRIBUTE" FC? 04
XEQ "E" X=0? SF 05
X=0? SF 04 FS? 04
XEQ "C" CF 05 X<=0?
GTO 01 ASTO 19 CF 22
ANUM STO 14 RCL 12
STO 11

37+LBL 03
RCL 19 RCL IND 11 X=Y?
GTO 02 ISG 11 GTO 03
GTO 01

45+LBL 02
RCL 11 INT 1 -
STO 19 SF 08 XEQ "C"
X<0? GTO 01 RCL 19 +
SEEKPT DELREC CLA
FS? 22 ARCL 14
"INS ST" XEQ "E" X=0?
"-?- " INSREC SF 06

68+LBL 01
XEQ "B" "CODE ERR"
AVIEW SF 25 END

01+LBL "MUSEL"
26,1 STO 11 "STRING"

05+LBL 00
XEQ "E" X=0? GTO 07
ASTO IND 11 ASHF
ISG 11 ASTO IND 11
"AND" FS? 04 "OR"
ISG 11 GTO 00

18+LBL 07
RCL 11 INT 1 - E3
/ 26 + STO 14 RCL 22
E3 / 1 + STO 15

34+LBL 01
XEQ "A" XEQ 05 XEQ "B"
ISG 15 GTO 01 CF 04
RTN

42+LBL 05
RCL 14 STO 11 FS? 04
GTO 03 SF 09

48+LBL 02
CLA ARCL IND 11 SF 08
ISG 11 ARCL IND 11
POSFL X<0? RTN
XEQ "D" FS?C 09 GTO 08
X=Y? GTO 05

62+LBL 08
ISG 11 GTO 02 XEQ 09
GTO 05

67+LBL 09
RCL 12 STO 13

70+LBL 06
GETREC PRA ISG 13
GTO 06 ADV RTN

77+LBL 03
SF 08 CLA RCL 11
ARCL IND X E +
ARCL IND X POSFL X<0?
GTO 04 XEQ "D" XEQ 09
GTO 03

91+LBL 04
SEEKPT ISG 11
ISG 11 GTO 03 END

109+LBL 02
SF 25 RCL 14 INT E3
/ + SEEKPT FS? 25
GETREC 32 FS? 25 ATOX
RCL IND X ACSPEC 14
SKPCOL RTN

127+LBL 08
SF 25 "LET-D" 0
XROM 28,14 30,101
FS? 25 XROM 28,09
FC? 25 RDTAX 64
FC? 25 XROM 28,01 RDN
0 XROM 28,14 RDN
FC? 25 XROM 28,22
RCL 41 STO 32 END

24+LBL 06
ISG 11 GTO 05
"ATT ERP" CF 21 AVIEW
SF 21 PSE GTO 04

33+LBL "PRTLET"
ASTO 26 ASHF ASTO 27

37+LBL 03
RCL 38 123 X=Y?
XEQ 08 FS?C 03 GTO 07
"STRING" XEQ "E"
ASTO 26 ASHF ASTO 27

49+LBL 07
RCL 22 E3 / 1 +
STO 15 SF 21

57+LBL 09
XEQ "A" XEQ 10 XEQ "B"
ISG 15 GTO 09 CF 04
SF 25 RTN

66+LBL 10
SF 08 XEQ "C" X<0?
RTN STO 13 ,023
STO 14

74+LBL 01
FC? 04 SF 12 RCL 13
RCL 31 + XEQ 02 CF 12
RCL 13 RCL 30 +
XEQ 02 RCL 13 RCL 29
+ XEQ 02 FS? 04 SF 12
RCL 13 RCL 28 +
XEQ 02 CF 12 ADV
ISG 14 GTO 01 RCL 13
RCL 12 FRC E3 * +
SF 25 SEEKPT GTO 10

```

Bernhard Saalfeld
Karl Höher Straße 4
5000 Köln 80

VIDEO Editing

HP-41C, IL, VIDEO-Interface HP 82163

Programm	Zeilen	Bytes	Regs.	SIZE
„VID“	40	129	19	000
„SCR“	8	25	4	000
„CLS“	20	51	8	000

Folgende drei Programme haben sich als sehr nützlich zur Steuerung des Video-Interfaces HP 82163 erwiesen.

Die zugrundeliegenden Ideen lassen sich evtl. auch auf andere IL-Controller und Video-Interfaces übertragen.

A. Bildschirm-Rollen: „VID“ und „SCR“
Konfiguration: HP 41, IL, Video-Interface 82163, ggf. TIME; mit Hilfsprogramm „SCR“ auch XF und CCD

Flags 00: Umsteuerung der Auf- und Abbewegung

51: Verlassen der Endloschleife

Register : Alpha und Stack werden gelöscht

Dieses Programm rollt schnell die 31 nutzbaren Zeilen über den Bildschirm. Zur besseren Orientierung sind die ersten bzw. letzten 15 Zeilen zusammen mit einer invers dargestellten Seitennummer gekennzeichnet. Mit Hilfe von „R/S“ läßt sich wiederholt zwischen diesen beiden Seiten wechseln. Drückt man gleich nach dem Programmstart auf „R/S“, so wird nur um eine halbe Seite verschoben.

Dennoch ist dieses Programm unterprogrammfähig. Nachdem eine Seite ganz über den Bildschirm gelaufen ist, kann

durch die Tastfolge „SST: R/S“ (nach „SST“ eine kurze Pause) mittels des SST-Flag 51 ins Hauptprogramm zurückgesprungen werden.

Vor dem Beginn des Bildschirmrollens wird noch die momentane Uhrzeit in die rechte obere Ecke des Schirmes geschrieben und der Programmlauf stoppt.

Das Video-Interface muß bei der gegebenen Programmierung das erste Gerät in der Schleife sein (Wie soll es sonst auch ohne X-I/O-Modul problemlos gefunden werden?). Die schnelle Bewegung auf dem Bildschirm wird dadurch ermöglicht, daß fertige, zusammenhängende Alphaketten durch die Schleife gesendet werden. Diese Ketten werden im Programm aus synthetischen Textzeilen gebildet. Damit das Programm unabhängig von Flag 21 (printer enable) arbeitet, werden die Alphaketten mit „OUTA“ gesendet.

Bei mir lasse ich dieses Programm als erstes im XF-Speicher laufen, um so mehr Platz im RAM zu haben. Zum Aufruf benutze ich folgendes kleines Programm, wobei die Tonfolge nach einem Unterprogramm-Ende den erfolgreichen Rücksprung signalisiert (evtl. „R/S“ mehrfach drücken):

```
LBL'SCR: CF15: 189,6: X>RTN:
PC<>RTN: TONE 0: TONE 0: END
```

Das TIME-Modul kann ggf. auch weglassen werden, indem man die Zeilen 6-12 löscht.

B. Video-Löschung: CLS“

Konfiguration: HP 41, IL-Modul, Video-Interface, XF-Modul ggf. TIME-Modul

Flag 00: wird gelöscht

Register : Alpha und Y,Z,T werden gelöscht, X bleibt erhalten

Mit dieser Routine ist es möglich den Bildschirm zu leeren, ohne den PRINT-MODE (Flags 15, 16) zu ändern. Der Cursor wird als Pfeil dargestellt und ausgeschaltet. In die rechte obere Ecke des Bildschirms wird die momentane Uhrzeit geschrieben.

Die Programmzeile 19 (CF 00) ist nur in Verbindung mit dem Programm VID sinnvoll, damit zum Bildschirm-Rollen ein definierter Anfangszustand erzeugt wird.

Man kann das TIME-Modul auch weglassen, indem man der Einfachheit halber die Programmzeilen 7,8 und 9 löscht und stattdessen „R↑“ einfügt.

Beim Editieren längerer Programme mittels Bildschirm ist es zweckmäßig, am Programmstart folgende Anweisungen einzufügen:

```
LBL A: XEQ'VID: RTN: LBL C: XEQ'CLS:
RTN: LBL E: (prgm) . . .
```

Man kann dann nach „RTN: R/S“ mit den Funktionstasten A,C oder E den Bildschirminhalt bewegen, ihn löschen, das Programm starten und natürlich auch listen.

Happy Editing

```
Ø1LBL "VID" 24 SF ØØ
Ø2 CF 15 25 GTO Ø3
Ø3 CF 16
Ø4 1 26LBL Ø2
Ø5 SELECT 27 SF 17
Ø6 -- 28 --
Ø7 FIX 2 29 OUTA
Ø8 TIME 3Ø --
Ø9 ATIME 31 OUTA
1Ø -- 32 CLA
11 OUTA 33 CF 17
12 STOP 34 CF ØØ
```

```
13LBL Ø1 35LBL Ø3
14 FS? ØØ 36 STOP
15 GTO Ø2 37 FS? 51
16 SF 17 38 RTN
17 -- 39 GTO Ø1
18 OUTA 4Ø .END.
19 --
2Ø OUTA
21 OUTA
22 CLA
23 CF 17
```

```
Ø1LBL "CLS"
Ø2 RCLFLAG
Ø3 CF 15
Ø4 1
Ø5 SELECT
Ø6 --
Ø7 FIX 2
Ø8 TIME
Ø9 ATIME
1Ø --
11 OUTA
12 R^
13 STO L
14 R^
15 STOF LAG
16 CLST
17 LASTX
18 CLA
19 CF ØØ
2Ø END
```

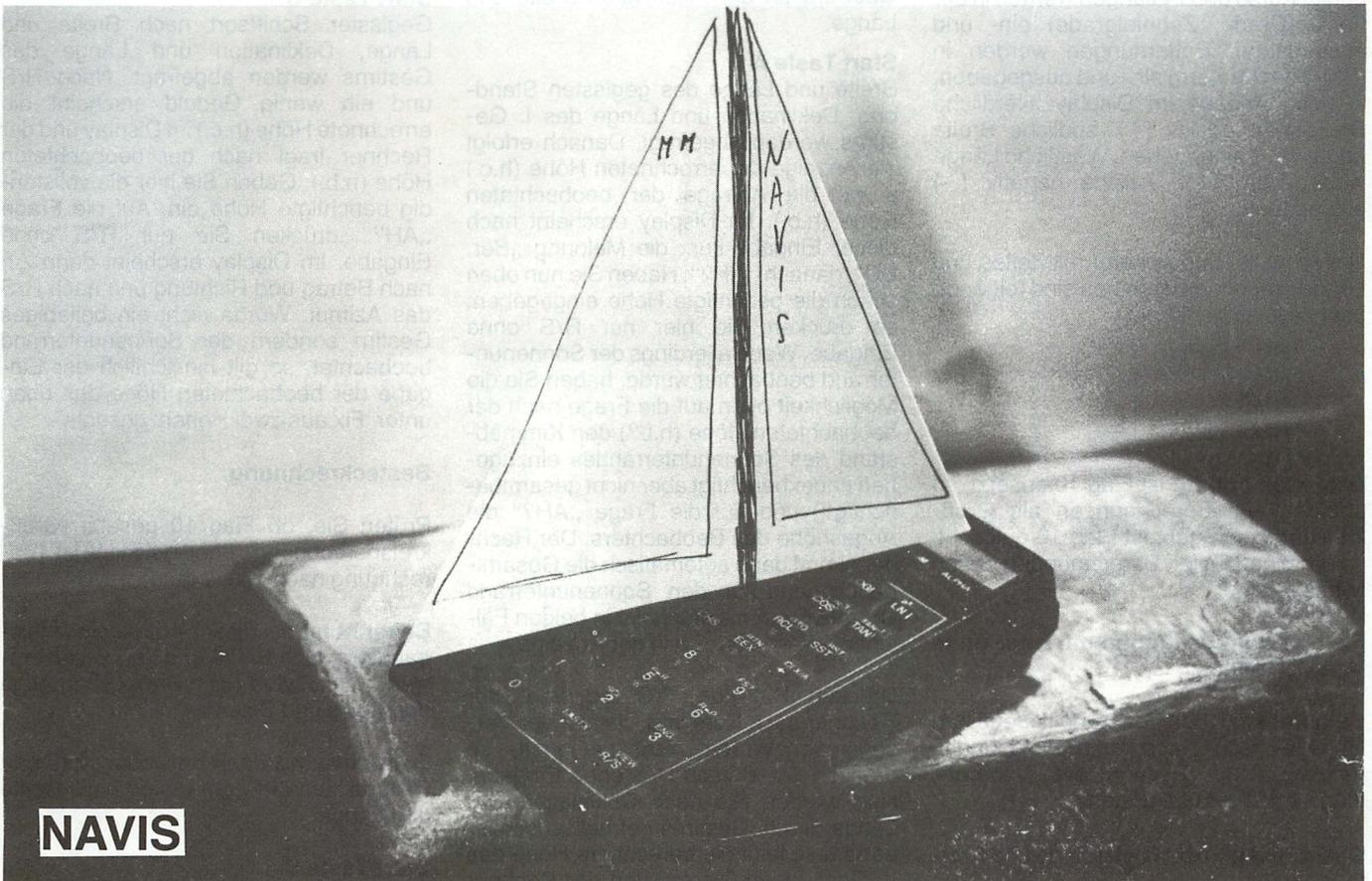
```
Ø1LBL "SCR"
Ø2 CF 15
Ø3 189,6
Ø4 X>RTN
Ø5 PC<>RTN
Ø6 TONE Ø
Ø7 TONE Ø
Ø8 END
```

CLS

```
006:FA 1B 45 1B 3C 1B
51 1B 25 1B 10
010:F3 7F 1B 48
```

```
VID
006:F4 1B 25 1B 10
010:F5 7F 1B 25 20 0F
017:FE 1B 25 5E 4F B2
1B 54 1B 25 5E 30 B1
1B 48
019:FE 1B 54 1B 54 1B
54 1B 54 1B 54 1B 54
1B 54
028:FE 1B 53 1B 53 1B
53 1B 53 1B 53 1B 53
1B 53
030:F7 7F 1B 53 1B 25
20 4F
```

Peter Hempler
Lönsring 2a
2105 Seevetal 2



NAVIS

Navigationsprogramm für Sportjachten

1070 Zeilen, 1903 Bytes, 272 Regs., SIZE 040, HP-41CV, Kartenleser (TIME)

Der Artikel über Navigationsmodule in Heft 3/88 brachte mich erneut auf die Idee, Euch mein schon 1980 auf dem HP 41 geschriebenes Navigationsprogramm für Yachten zu schicken. Ich wollte dies schon oft tun, zweifelte aber immer am allgemeinen Interesse.

Das Programm hat sich bisher gut bewährt, Fehler sind mir zur Zeit nicht mehr bekannt.

Die Programmstruktur ist ziemlich komplex. Ich war gezwungen immer wieder auf Unterroutrinen zuzugreifen, um Platz zu sparen. Dies erforderte dann teilweise seltsame Umformungen an anderer Stelle.

Hinzu kommt, daß ich Anfangs (1980) noch Neuling auf dem HP 41 war. Von synthetischer Programmierung also keine Spur.

Das Programm belegt also buchstäblich den letzten Speicherplatz des HP 41 CV.

Trotzdem finde ich es gut handhabbar und den Bedürfnissen des Freizeitkapitäns angemessen.

Das Programm „Navis“ wurde für den programmierbaren Taschenrechner HP

41 C(V) der Firma Hewlett Packard geschrieben. Die Kenntnis der grundlegenden Eigenschaften und eine gewisse Vertrautheit in der Bedienung des Rechners werden vorausgesetzt.

Das Programm löst die häufigsten in der Sportschiffahrt vorkommenden Problemstellungen:

- Fix aus zwei Peilungen mit und ohne Versegelung
- Fix aus zwei Höhen mit und ohne Versegelung
- Höhe, Δh und Azimut eines Gestirns (für die Bonne ersetzt das Programm das Jahrbuch)
- Besteckrechnung wahlweise nach Mittelbreite oder Großkreis
- Koppelkurs und Koppelort
- Distanz eines Objektes bekannter Höhe (auch wenn der Fußpunkt hinter der Kimm liegt)
- Stromrechnungen

Sämtliche Aufgaben werden im Dialogverkehr abgewickelt, d.h. der Rechner fragt die jeweils notwendige Eingabe ab und alle Ausgaben sind im Klartext kommentiert. Mit diesem Verfahren werden Eingabefehler minimiert. Außerdem ist an Bord ein „Waschzettel“ unnötig.

Programmfumfang:

- 1068 Programmschritte auf 9 Magnetkarten.
- 39 Register als Wort- und Wertespeicher auf drei Magnetkarten.
- User keys auf einer Status-Magnetkarte.

Die Magnetkarten sind beschriftet. Auf Wunsch kann ein Programmlisting mitgeliefert werden.

Notwendige Systemkonfiguration:

- HP 41 CV oder HP 41 C mit Quad-Ram.
- Magnetkartenleser HP 82104 zum Laden der Programme.

Mit der Lieferung des Programms ist keinerlei Verpflichtung, Garantie oder Haftung für Schäden, die aus der Anwendung des Programms entstehen, verbunden. Das Programm wurde während 2 Jahren in der Praxis geprüft und arbeitete zuletzt einwandfrei. Trotzdem sei darauf hingewiesen, daß ein so komplexes Programm fast niemals völlig ohne bisher noch nicht erkannte Fehler sein kann. Weiterhin möchte ich an dieser Stelle darauf hinweisen, daß eine Navigation „nur“ mit dem Rechner, ohne Überprüfung der Ergebnisse mit anderen Methoden, wegen der doch möglichen Fehlfunktion von Rechner oder Programm nicht den Regeln einer guten Seemannschaft genügt.

Programmablaufbeschreibung für die einzelnen Programme.

Vorbemerkung: Eingabe und Ausgabe erfolgen in der Form (Grad, Min., Sekunden). Nach Eingabe der ganzen Grade ist der Dezimalpunkt zu drücken. Eingabe und Ausgabe der Zeiten erfolgt in der Form (HH MM SS) ohne Dezimal-

punkt. Kurse und Peilungen werden in der Form (Grade, Zehntelgrade) ein- und ausgegeben. Entfernungen werden in (SM) oder (Metern) ein- und ausgegeben, je nach Angabe im Display. Nördliche Breite wird positiv (+), südliche Breite negativ (-) eingegeben. Westliche Länge wird positiv (+), östliche negativ (-) eingegeben.

Um schneller und sicherer mit Zeiten und Graden rechnen zu können sind folgende user keys eingegeben:

HR	in	HMS	f*
HMS	in	HR	f/
HMS	-		f-
HMS	+		f+

Großkreisooption: Ist Flag 10 gesetzt, so erfolgen alle Berechnungen als Großkreisberechnungen, ist Flag 10 gelöscht, so erfolgen alle Berechnungen nach Mittelbreite.

Fix aus zwei Peilungen mit oder ohne Versegelung

Bekannt sind: Ort des I. und II. gepeilten Objektes nach Länge und Breite, I. und II. rechtsweisende Peilung, evtl. Versegelung nach Kurs und Distanz.

Gesucht ist: Fixort nach Länge und Breite.

Start Taste F

Länge und Breite des I. und II. Objektes ("Peilort" im Display) werden abgefragt, danach die I. und II. rechtsweisende Peilung, danach Kurs und Distanz der Versegelung. Jede Eingabe ist mit R/S abzuschließen. Soll keine Versegelung berücksichtigt werden, so muß bei „Kurs = “ im Display R/S ohne Eingabe gedrückt werden.

Haben Sie nach der letzten Eingabe ein wenig Geduld. Der HP 41 ist ein Taschenrechner!

Als Ausgabe erscheint der Fixort nach Breite und (R/S) Länge. Wollen Sie nun von hier aus weternavigieren, so drücken Sie R/S ohne Eingabe. Der Rechner fragt Sie dann nach Breite und Länge des Zielortes. Haben Sie dies eingegeben so erscheinen nach dem abschließenden R/S Kurs und Distanz zum Ziel im Display. Je nach gewählter Option (Flag 10 gesetzt oder gelöscht), handelt es sich um Anfangskurs und Distanz auf dem Großkreis oder um Kurs und Distanz nach Mittelbreite.

Fix aus zwei Höhen mit oder ohne Versegelung

Bekannt sind: Gegissster Ort nach Breite und Länge; Deklination Länge (Greenwichwinkel) und beobachtete Höhe des I. und II. Gestirns im Zeitpunkt der Beobachtung. Evtl. Versegelung nach Kurs und SM.

Gesucht ist der Fixort nach Breite und Länge.

Start Taste A

Breite und Länge des gegisssten Standorts, Deklination und Länge des I. Gestirns werden abgefragt. Danach erfolgt die Anzeige der errechneten Höhe (h.c.) sowie die Abfrage der beobachteten Höhe (h.b.). Im Display erscheint nach dieser Eingabe kurz die Meldung „Ber. SU“, danach „AH?“. Haben Sie nun oben schon die berichtigte Höhe eingegeben, so drücken Sie hier nur R/S ohne Eingabe. Wenn allerdings der Sonnenunterrand beobachtet wurde, haben Sie die Möglichkeit oben auf die Frage nach der beobachteten Höhe (h.b?) den Kimmabstand des Sonnenunterrandes einzugeben (indexberichtigt aber nicht gesamtberichtigt!) und auf die Frage „AH?“ die Augeshöhe des Beobachters. Der Rechner nimmt dann automatisch die Gesamtberichtigung für den Sonnenunterrand vor. Als Ergebnis erscheint in beiden Fällen der Wert von Δh in der Anzeige.

Drücken Sie nun R/S, so fragt der Rechner nach Kurs und Distanz der Versegelung. Wurde nicht versegelt, so drücken Sie einfach R/S ohne Eingabe. Nun werden Sie nach Deklination und Länge des II. Gestirns gefragt, anschließend erscheint die berechnete Höhe des II. Gestirns im Display und der Rechner fragt nach der beobachteten Höhe „h.b.“ (siehe oben).

Als Ausgabe erscheint der Fixort nach Breite und (R/S) Länge. Wollen Sie nun von hier aus weternavigieren, so drücken Sie R/S ohne Eingabe. Der Rechner fragt Sie dann nach Breite und Länge des Zielortes. Haben Sie dies eingegeben so erscheinen nach dem abschließenden R/S Kurs und Distanz zum Ziel im Display. Je nach gewählter Option (Flag 10 gesetzt oder gelöscht), handelt es sich dabei um Anfangskurs und Distanz auf dem Großkreis oder um Kurs und Distanz nach Mittelbreite.

Besonderheiten: Ist statt eines beliebigen Gestirns die Sonne beobachtet worden, so kann bei der Abfrage der Deklination R/S ohne Eingabe gedrückt werden. Der Rechner fragt dann nach Datum und Uhrzeit (GMT bzw. UT: Format DD MM YY und HH MM SS). Ist ein TIME-Modul vorhanden so kann auch hier R/S ohne Eingabe gedrückt werden. Der Rechner übernimmt dann Datum und Uhrzeit im Moment des Tastendrucks. Vergewissern Sie sich, daß der Rechner auf GMT läuft!

Höhe, Δh und Azimut eines Gestirns

Bekannt sind: der gegissste Schiffsort nach Breite und Länge, die Deklination und Länge (GHA) des Gestirns, sowie die beobachtete Höhe.

Gesucht sind: errechnete Höhe (h.c.), h und Azimut des Gestirns.

Start Taste C

Gegissster Schiffsort nach Breite und Länge, Deklination und Länge des Gestirns werden abgefragt. Nach R/S und ein wenig Geduld erscheint die errechnete Höhe (h.c.) im Display und der Rechner fragt nach der beobachteten Höhe (h.b.). Geben Sie hier die vollständig berichtigte Höhe ein. Auf die Frage „AH?“ drücken Sie nur R/S ohne Eingabe. Im Display erscheint dann Δh nach Betrag und Richtung und nach R/S das Azimut. Wurde nicht ein beliebiges Gestirn sondern der Sonnenunterrand beobachtet, so gilt hinsichtlich der Eingabe der beobachteten Höhe das oben unter 'Fix aus zwei Höhen' gesagte.

Besteckrechnung

Prüfen Sie, ob Flag 10 gesetzt (Großkreisberechnung) oder gelöscht ist (Berechnung nach Mittelbreite).

Bekannt ist bei dieser Aufgabe der Ausgangsort nach Breite und Länge und entweder der Zielort nach Breite und Länge oder Kurs und Distanz.

Gesucht sind dann entweder Kurs und Distanz zum Ziel oder Zielort nach Breite und Länge.

Start Taste B

Der Rechner fragt nach Breite und Länge des Ausgangsortes, danach nach Breite und Länge des Zielortes. Wird dies eingegeben, so erscheint im Display der Kurs in Grad und nach R/S die Distanz in SM.

Wenn bei Abfrage des Zielortes R/S ohne Eingabe gedrückt wird, fragt der Rechner nach Kurs und Distanz und gibt dann die Breite und (nach R/S) die Länge des Zielortes aus.

Sinnvollerweise wird dieser Aufgabenteil unabhängig von der Stellung des Flag 10 immer nach Mittelbreite berechnet (eine Großkreisberechnung wäre nur für z.B. die Artillerie wichtig; und dieses Programm ist für friedliche Zwecke).

Koppelrechnung

Bekannt sind die gesegelten Kurse und Distanzen und der Ausgangsort nach Breite und Länge.

Gesucht sind resultierender Kurs und Gesamtdistanz sowie der Koppelort.

Start Taste E

Der Rechner fragt immer wieder nach Kurs und Distanz. Geben Sie hier die einzelnen Kurse und Distanzen ein. Wenn Sie dann auf die Frage „Kurs=“ R/S ohne Eingabe drücken, gibt der Rechner nach kurzer Zeit den resultierenden Kurs und (nach R/S) die Gesamtdistanz aus.

Drücken Sie nun nochmals R/S ohne Eingabe, so fragt der Rechner nach Breite und Länge des Ausgangsortes und gibt danach Breite und Länge des Koppelortes (im Display als Zielort bezeichnet) aus.

Da es sich bei diesem Programmteil um reine Vektoraddition handelt, können hiermit auch andere Kurse und Distanzen verrechnet werden (z.B. Stromvektoren; s.u.).

Distanz zu einem Objekt bekannter Höhe

Bekannt sind die Höhe eines Objektes über der Strandlinie, die Augeshöhe des Beobachters und der beobachtete Winkel über der Strandlinie oder der Kimm.

Gesucht ist die Distanz zum Fußpunkt des Objektes.

Start Taste D

Der Rechner fragt nach der Höhe des Objektes, der Augeshöhe des Betrachters (beides in Metern) und dem beobachteten Winkel (Grad.min sek.) und gibt dann die Distanz in SM im Display aus.

Stromaufgaben

Hier sind **zwei Aufgaben** zu unterscheiden: Sind Kurs und Fahrt durchs Wasser sowie Richtung und Stärke des Stromes bekannt und sind resultierender Kurs und resultierende Fahrt gesucht, so handelt es sich um einfache Vektoraddition. Sie drücken dann **Taste E** und geben bei Abfrage der „SM=“ die entsprechenden Knoten (=SM/h) ein. Das Ergebnis „....SM“ muß dann ebenso wieder als Knoten (=SM/h) gelesen werden.

Sind jedoch Kartenkurs, Stromrichtung, Stromstärke (Kn) und Fahrt durchs Wasser (Kn) bekannt und wird der zu steuernde Kurs durchs Wasser gesucht, so drücken Sie **Taste E**. Der Rechner fragt dann den Kartenkurs, die Richtung des Stroms ('Stromkurs'), die Stromstärke (SM/h) und die Fahrt durchs Wasser (SM/h) ab und gibt im Display den Kurs aus, den Sie steuern müssen (Kurs durchs Wasser).

Wird nun zusätzlich die Fahrt auf der Kartenkurslinie gewünscht, so muß mit den jetzt bekannten Daten (Kurs und Fahrt durchs Wasser, Stromrichtung und -stärke) in die Vektoraddition eingegangen werden (**Taste E**).

Umrechnung von Länge in Zeit und umgekehrt

Des Komforts wegen sind noch die beiden Programme eingebaut. Sie können damit Länge in Zeit oder umgekehrt verwandeln. Wollen Sie Länge in Zeit verwandeln, so drücken Sie **Taste I**. Der Rechner fragt nach Länge oder Längendifferenz (Grad.Min Sek) und gibt die entsprechende Zeitdifferenz aus (HH MM SS).

Drücken Sie die **Taste J**, so fragt der Rechner nach der Zeit oder Zeitdifferenz (HH MM SS) und gibt die entsprechende Länge oder Längendifferenz (Grad.Min Sek) aus.

Auf den folgenden Seiten finden sich nun einige Beispiele. Bei diesem Druckerprotokoll sind linksbündig die Abfragen, mittig die Eingaben und rechtsbündig die Tastenbefehle ausgedruckt.

Eingabe + Ausgabe

Winkel HH.MMSS
Zeit HHMMSS.
Datum DDMMYY

Labels:

- 0-15 lokale Labels
- 17 Rechenschema
- 20 Eingabe Standort und Zielort
- 21 Versiegeln
- 22 Errechnung Fixort aus Peilung/ Stern Fix
- 23 Kurs und Distanz aus zwei Orten
- 24 Zielort aus Standort + Kurs + Distanz
- 25 Einstieg in Endberechnung aus Peilung/Fix und LBL 24
- 26 Einstieg in LBL 20, wenn nur Zielort gefragt
- 30 Besteckrechnung
- 31 Fix aus zwei Sternstandlinien
- 32 Rechnung Sonnenkoordinaten

- R00= "STAND"
- R01= "ORT :"
- R02= "BREITE"
- R03= "/N+S- "
- R04= "LAENGE"
- R05= "/W+E- "
- R06= "H.c. ="
- R07= "AZIMUT"
- R08= "KURS= "
- R09= " SM= "
- R10= "ZIEL"
- R11= "*-DECL"
- R12= "*-LONG"
- R13= 39,72
- R14= -3,53
- R15= 39,72
- R16= -3,53
- R17= 81,00
- R18= 2,59
- R19= "SM HIN"
- R20= "SM WEG"
- R21= 19,66
- R22= "FIX-"
- R23= "II. "
- R24= "PEIL"
- R25= "UNG"
- R26= "H.b. ?="
- R27= 56,68
- R28= 20,63
- R29= 119,19
- R30= -8,57
- R31= -81,00
- R32= -34,00
- R33= 0,99
- R34= 196,82
- R35= 3,00
- R36= 52,28

Variabel

Variabel

USER-Tasten

- A Fix
- B Besteck
- C Hc + Az
- D Distanz
- E Koppelrechnung
- F Fix aus Peilung
- G Decl. + Az der Sonne
- H Stromrechnung
- I LIZ
- J ZIL

```
01*LBL "NAVIS"
02*LBL B
CF 02 CF 01 CF 00 CLA
"BEST." AVIEW PSE CLA
XEQ 30 RTN
```

```
13*LBL C
CF 02 CF 01 CF 00 CLA
"*-ORT" AVIEW PSE CLA
XEQ 31 RTN
```

```
24*LBL A
CLA "FIX" AVIEW PSE
CLA CF 00 CF 01 SF 02
XEQ 31 STOP XEQ 21
CLA ARCL 23 ARCL 11
CF 22 PROMPT CLA
FC?C 22 XEQ 32 FS?C 04
GTO 13 HR STO 15
ARCL 23 ARCL 12 PROMPT
HR
```

```
52*LBL 13
RCL 14 - STO 21 CLA
SF 00 XEQ 31 RCL 17
90 + CHS STO 32
SF 01 XEQ 30 RCL 15
STO 31 RCL 16 STO 21
RCL 29 STO 17 RCL 30
STO 18 SF 01 XEQ 30
RCL 15 STO 13 RCL 16
STO 14 RCL 31 STO 15
RCL 21 STO 16 RCL 17
90 + CHS STO 31
XEQ 22 RTN
```

```
91*LBL 31
```

```
92*LBL 04
FS? 00 GTO 05 FIX 4
CLA ARCL 00 ARCL 01
AVIEW PSE PSE CLA
ARCL 02 ARCL 03 PROMPT
CLA HR STO 13 FS? 02
STO 27 ARCL 04 ARCL 05
PROMPT CLA HR STO 14
FS? 02 STO 28 CF 22
ARCL 11 ARCL 03 PROMPT
CLA FC?C 22 XEQ 32
FS?C 04 GTO 13 HR
STO 15 ARCL 12 ARCL 05
PROMPT CLA HR
```

SERIE 40

```

135*LBL 13
RCL 14 - STO 21

139*LBL 05
FIX 4 RCL 13 RCL 21
CHS RCL 15

145*LBL 17
1 P-R X<>Y RDN P-R
R↑ R-P X<>Y R↑ +
X<>Y P-R X<>Y RDN
R-P R↑ X<>Y R-P RDN
FS?C 04 RTN STO 17
X<>Y 360 MOD

171*LBL 09
RCL 17 HMS ARCL 06
"-↑" ARCL X TONE 6
AVIEW PSE PSE PSE
CLA ARCL 26 PROMPT
CLA CF 22 "BER. SU:"
TONE 7 AVIEW PSE PSE
PSE "AH?" PROMPT CLA
FS?C 22 XEQ 90 FIX 1
X<>Y HMS- 1 E2 * HR
STO 18 FS? 02 STO 30
FS? 00 GTO 03 FS? 02
GTO 03 X<=0? GTO 02
ARCL X "↑" ARCL 19
AVIEW TONE 8 STOP CLA
GTO 03

221*LBL 02
ABS ARCL X "↑"
ARCL 20 AVIEW STOP
CLA

229*LBL 03
X<>Y FS? 02 STO 29
STO 17 X<>Y FS?C 00
RTN FS?C 02 RTN
ARCL 07 "↑" ARCL Y
AVIEW RTN

244*LBL 30

245*LBL 16
CLA FIX 4 CLX FS? 00
GTO 23 FS? 01 GTO 24
STO 13 STO 14 STO 15
STO 16 STO 17 STO 18
ARCL 00

260*LBL 20
ARCL 01 AVIEW PSE CLA
ARCL 02 ARCL 03 PROMPT
CLA HR FC? 06 STO 13
STO 15 ARCL 04 ARCL 05
PROMPT CLA HR FC?C 06
STO 14 STO 16 FS?C 05
RTN CLX

284*LBL 26
ARCL 10 ARCL 01 AVIEW
PSE CLA ARCL 02

ARCL 03 CF 22 PROMPT
CLA FC?C 22 GTO 06 HR
STO 15 CF 22 ARCL 04
ARCL 05 PROMPT CLA
FS? 22 GTO 04

306*LBL 06
ARCL 08 PROMPT CLA
STO 17 ARCL 09 PROMPT
CLA STO 18 GTO 24

316*LBL 04
HR STO 16

319*LBL 23
FC? 10 GTO 00
"FS10: GROSSKR." AVIEW
TONE 1 FS? 10 GTO 08

327*LBL 00
RCL 15 RCL 13 -
RCL 16 RCL 14 -
RCL 15 RCL 13 + 2 /
COS * X<>Y R-P 60 *
STO 18 X<>Y CHS
STO 17

349*LBL 18
360 MOD FS?C 00 RTN
FIX 1 ARCL 08 ARCL X
AVIEW STOP CLA
ARCL 09 ARCL Y AVIEW
STOP CLA GTO 16

366*LBL 08
RCL 13 RCL 16 RCL 14
- CHS RCL 15 SF 04
XEQ 17 90 - CHS 60
* STO 18 X<>Y STO 17
CLA GTO 18

385*LBL 24
CLX STO 15 STO 16
FIX 4 RCL 17 CHS
RCL 18 60 / P-R
ST+ 15 RCL 13 ST+ 15
RDN 2 / RCL 13 +
COS 1/X * ST+ 16
RCL 14 ST+ 16 FS?C 01
RTN

412*LBL 25
FIX 4 RCL 15 FS? 07
STO 13 HMS FS?C 04
GTO 00 ARCL 10 GTO 01

422*LBL 00
ARCL 22

424*LBL 01
ARCL 01 AVIEW PSE CLA
ARCL 02 ARCL 03 ARCL X
AVIEW STOP CLA RCL 16
FS? 07 STO 14 HMS

ARCL 04 ARCL 05 ARCL X
AVIEW STOP CLA
FS?C 07 GTO 26 RTN

448*LBL D
CLA "DIST." AVIEW PSE
FIX 2 CLA "HOEHE"
"-↑<N>" PROMPT CLA
"AH ?" PROMPT CLA -
LASTX SQRT 1,76 *
CHS "Δ? <HMS>" PROMPT
CLA HR 60 * + STO Z
X↑2 X<>Y 3,71 * +
SQRT X<>Y - "DIST="
ARCL X "↑ SM" AVIEW
CLA RTN

490*LBL E
CLA "KOPPEL" AVIEW
PSE CLA CF 00 CF 01
CF 02 FIX 1 CLX
STO 17 STO 18

503*LBL 19
CLX CF 22 CLA ARCL 08
PROMPT FC?C 22 GTO 11
CLA ENTER↑ ARCL 09
PROMPT CLA P-R ST+ 17
RDN ST+ 18 GTO 19

521*LBL 11
RCL 18 RCL 17 R-P
STO 18 RDN 360 MOD

529*LBL 12
STO 17 CLA ARCL 08
ARCL X AVIEW STOP CLA
RCL 18 ARCL 09 ARCL X
AVIEW STOP SF 05 CLA
"AUSG.-" "↑" XEQ 20
XEQ 24 RTN

549*LBL 21
CLA "VERSEG.?" AVIEW
PSE CLA ARCL 08 CF 22
PROMPT FC?C 22 RTN
STO 17 CLA ARCL 09
PROMPT STO 18 CLA
SF 01 XEQ 30 RCL 15
STO 13 RCL 16 STO 14
RTN

573*LBL F
CLA CLX ARCL 24
ARCL 25 AVIEW PSE CLA
"↑." ARCL 24 SF 05
CF 06 XEQ 20 CLA "Δ"
PROMPT 180 - CHS
STO 31 XEQ 21 CLA
SF 05 SF 06 ARCL 23
ARCL 24 XEQ 20 CLA
"Δ" PROMPT 180 - CHS
STO 32

607*LBL 22
SF 00 XEQ 30 RCL 18
RCL 17 CHS 180 -
RCL 32 - SIN RCL 32
RCL 31 - SIN / *
ABS STO 18 RCL 31 CHS
STO 17 SF 04 SF 07
GTO 24

632*LBL H
"STROM" AVIEW PSE CLA
"Ka-" ARCL 08 PROMPT
CLA "STROM" ARCL 08
PROMPT ARCL 09 PROMPT
CLA "FdW" ARCL 09
PROMPT FIX 1 / RDN -
SIN R↑ * ASIN + 360
MOD CLA ARCL 08
"↑dW" ARCL X AVIEW
STOP

667*LBL 32
SF 04

669*LBL G
"SONNE" AVIEW PSE
"DATUM?" CF 22 PROMPT
FS?C 22 GTO 01 DMY
DATE 1 E2 * INT
LASTX 1 E2 * FRC +
1 E2 *

690*LBL 01
STO 34 "GMT?" PROMPT
1 E4 / FC?C 22 TIME
CLA CF 03 HR 24 /
STO 21 X<> 29 X<> 37
X<> 29 X<> 30 X<> 38
X<> 30 XEQ 91 RTN

712*LBL 94
RCL 34 1 E4 / INT
STO 34 LASTX - CHS
1 E2 * INT STO 35
LASTX - CHS 1 E2 *
1900 + STO 36 365 *
46 - RCL 34 + 31
RCL 35 * + RCL 35 3
X>Y? GTO 95 RDN ,4 *
2,3 + INT - RCL 36
4 / INT + STO 34
RTN

761*LBL 95
RDN RDN RCL 36 1 -
4 / INT + STO 34
RTN

773*LBL 91
XEQ 94 722449 -
RCL 21 + STO 31
365,2319 - STO 18
1,60213 * STO 15

```

```
786*LBL 98
SIN ,0093838 STO 16 *
RCL 15 + 2 / TAN
1,0134 * ATAN 2 *
RCL 18 ,0000382 * +
STO 29 ,09 RCL 29 SIN
X↑ 2 * + STO 29
ENTER↑ 2 * SIN ,015
* + 131,277 + 360
MOD STO 16 ,723332
ENTER↑ 1 ENTER↑
,006783 RCL 29 COS *
- * STO 29 RCL 16
76,49 - RCL 18
,000038 * + SIN
3,3943 * STO 30
RCL 31 368,5637 -
STO 18 ,98560912 *
STO 32
```

```
854*LBL 99
SIN ,016718 STO 15
RCL 32 + 2 / TAN
1,0337 * ATAN 2 *
STO 15 102,585 +
RCL 18 ,0000382 * +
360 MOD STO 17 1
ENTER↑ ,0167 RCL 15
COS * - STO 18
STO 33 RCL 31
,98564734 * 279,306 +
STO 31 -104,55 RCL 31
SIN * 429,266 RCL 31
COS * - RCL 31 2 *
SIN 595,63 * +
RCL 31 2 * COS 2,283
* - RCL 31 3 * SIN
4,6 * + RCL 31 3 *
COS 18,7333 * +
RCL 31 4 * SIN 13,2
* - RCL 31 5 * COS
- RCL 31 5 * SIN 3
/ - RCL 31 6 * SIN
,5 * + ,231 + 240
/ 360 RCL 21 ,5 + *
+ 360 MOD STO 21
23,4412 SIN RCL 17
180 + SIN * ASIN
STO 15 RCL 21 X<> 37
X<> 29 X<> 37 X<> 38
X<> 30 X<> 38 FS? 04
RTN CLA HMS X<>Y HMS
ARCL 11 ARCL 03 ARCL X
PROMPT CLA ARCL 12
ARCL 05 ARCL Y PROMPT
RTN
1001*LBL 90
RCL T STO 34 RCL T
STO 36 RDN RDN X<>Y
HR ENTER↑ RCL Z SORT
1,8 * CHS 16,55 +
RCL 35 30 * COS ,25
* + X<>Y 1/X 60 *
```

```
- 60 / + HMS ENTER↑
ENTER↑ RCL 34 RCL 36
R↑ RTN
```

```
1040*LBL I
CLA "LIZ" CF 29 FIX 4
PROMPT HR 15 / HMS
1 E4 FIX 0 * VIEW X
RTN
```

```
1055*LBL J
CF 29 CLA "ZIL" FIX 0
PROMPT 1 E4 / FIX 4
HR 15 * HMS VIEW X
RTN END
```

Beispiele für NAVIS

```
FIX XEQ A
STANDORT :
BREITE/N+S- 56,4100 RUN
LAENGE/W+E- 20,3800 RUN
*-DECL/N+S- RUN
SONNE
DATUM? 220773,0000 RUN
GMT? 103030,0000 RUN
H.c. =41,0051
H.b.?= 40,5224 RUN
BER. SU:
AH? RUN
VERSEG.?
KURS= 70,0 RUN
SM= 36,5 RUN
II. *-DECL RUN
*-LONG 30,2500 RUN
H.c. =52,2837
H.b.?= 52,3618 RUN
BER. SU:
AH? RUN
FIX-ORT :
BREITE/N+S- 56,4914 RUN
LAENGE/W+E- 19,5817
*-ORT XEQ C
STANDORT :
BREITE/N+S- 52,4600 RUN
LAENGE/W+E- -7,2900 RUN
*-DECL/N+S- RUN
SONNE
DATUM? 00573,0000 RUN
GMT? 71143,0000 RUN
H.c. =29,2028
H.b.?= 29,1942 RUN
BER. SU:
AH? RUN
0,8 SM WEG RUN
AZIMUT 100,5 RUN
```

```
FIX XEQ A
STANDORT :
BREITE/N+S- 56,4100 RUN
LAENGE/W+E- 20,3800 RUN
*-DECL/N+S- 20,1554 RUN
*-LONG/W+E- 336,0206 RUN
H.c. =41,0114
H.b.?= 40,5224 RUN
BER. SU:
AH? RUN
VERSEG.?
KURS= 70,0 RUN
SM= 36,5 RUN
II. *-DECL 20,1406 RUN
II. *-LONG 30,2500 RUN
H.c. =52,2837
H.b.?= 52,3618 RUN
BER. SU:
AH? RUN
FIX-ORT :
BREITE/N+S- 56,4914 RUN
LAENGE/W+E- 19,5817
*-ORT XEQ C
STANDORT :
BREITE/N+S- 52,4600 RUN
LAENGE/W+E- -7,2900 RUN
*-DECL/N+S- RUN
SONNE
DATUM? 00573,0000 RUN
GMT? 71143,0000 RUN
H.c. =29,2028
H.b.?= 29,1942 RUN
BER. SU:
AH? RUN
0,8 SM WEG RUN
AZIMUT 100,5 RUN
```

SONNE
 DATUM? 20383,0 RUN
 GMT? 102500,0 RUN
 *-DECL/N+S- -7,2 RUN
 *-LONG/W+E- 333,1 RUN

FIX
 STANDORT :
 BREITE/N+S- 56,4100 RUN
 LAENGE/W+E- 20,3800 RUN
 *-DECL/N+S- 20,1554 RUN
 *-LONG/W+E- 336,0206 RUN
 H.c. =41,0114
 H.b.?=
 Ka: 40,4300 RUN
 BER. SU:
 AH? 9,0000 RUN
 VERSEG.?
 KURS= 70,0 RUN
 SM= 36,5 RUN
 II. *-DECL 20,1406 RUN
 II. *-LONG 30,2500 RUN
 H.c. =52,2037
 H.b.?=
 Ka: 52,2630 RUN

BER. SU:
 AH? 9,0000 RUN
 FS10: GROSSKR.
 FIX-ORT :
 BREITE/N+S- 56,4058 RUN
 LAENGE/W+E- 19,5759 RUN

USER KEYS:
 -11 TIME -52 DDAYS
 -12 DATE -61 HMS+
 -22 ALMNOW -62 DATE+
 -33 XYZALM -71 HR
 -41 ALMCAT -81 HMS
 -51 HMS- -84 PROMPT

XEQ G
 KOPPEL
 KURS= 79,0 RUN
 SM= 15,0 RUN
 KURS= 317,0 RUN
 SM= 25,0 RUN
 KURS= 50,0 RUN
 SM= 32,0 RUN
 KURS= 142,0 RUN
 SM= 15,0 RUN
 KURS= 216,0 RUN
 SM= 10,0 RUN
 KURS= 125,0 RUN
 SM= 22,0 RUN
 KURS= 203,0 RUN
 SM= 6,0 RUN
 KURS= 84,9 RUN
 SM= 41,4 RUN

Stromaufgaben
 XEQ E
 KOPPEL
 KURS= 225,0 RUN
 SM= 8,0 RUN
 KURS= 157,0 RUN
 SM= 2,5 RUN
 KURS= 210,5 RUN
 SM= 9,2 RUN

XEQ B
 BEST.
 STANDORT :
 BREITE/N+S- 38,1500 RUN
 LAENGE/W+E- 25,1000 RUN
 ZIELORT :
 BREITE/N+S- 45,1200 RUN
 LAENGE/W+E- 20,5500 RUN
 FS10: GROSSKR.
 KURS= 23,1 RUN
 SM= 450,2 RUN

Großkreis
 CF 10
 XEQ B
 BEST.
 STANDORT :
 BREITE/N+S- 38,1500 RUN
 LAENGE/W+E- 25,1000 RUN
 ZIELORT :
 BREITE/N+S- 45,1200 RUN
 LAENGE/W+E- 20,5500 RUN
 KURS= 24,5 RUN
 SM= 450,4 RUN

Mittelbreite
 Kurs und **SM/h**
 des Schiffes
 Kurs und **SM/h**
 des Stromes
 Resultierender
 Kurs und **SM/h**

XEQ H

STROM
Ka-KURS= 157,0 RUN gesuchter rw Ka.Kurs

STROMKURS= 112,0 RUN rw Stromrichtung

STROMKURS= SM= 2,5 RUN Stromgeschwindigkeit SM/h

FdW SM= 7,0 RUN Fahrt des Wassers SM/h

KURS= dW 171,6 gesuchter Kurs des Wassers (rw)

XEQ E

KOPPEL
KURS= 171,6 RUN Damit in die Koppelrechnung:

SM= 7,0 RUN Kurs und Fahrt des Wassers

KURS= 112,0 RUN Stromrichtung und Stärke

SM= 2,5 RUN

KURS= RUN rw Kurs + Fahrt in Grund

KURS= 157,0 RUN

SM= 8,5 RUN

Peilungen

SF 10

XEQ F

PEILUNG

I. PEILORT :

BREITE/N+S- 39,4300 RUN

LAENGE/W+E- -3,2830 RUN

Δ 261,0000 RUN

VERSEG.? KURS= RUN

II. PEILORT :

BREITE/N+S- 39,3442 RUN

LAENGE/W+E- -3,2412 RUN

Δ 214,0000 RUN

FIX-ORT :
BREITE/N+S- 39,4324 RUN

LAENGE/W+E- -3,3150 RUN

PEILUNG

I. PEILORT :

BREITE/N+S- 39,4300 RUN

LAENGE/W+E- -3,2830 RUN

Δ 261,0000 RUN

VERSEG.? KURS= RUN

II. PEILORT :

BREITE/N+S- 39,3442 RUN

LAENGE/W+E- -3,2412 RUN

Δ 214,0000 RUN

FS10: GROSSKR.
FIX-ORT :
BREITE/N+S- 39,4324 RUN

LAENGE/W+E- -3,3149 RUN

XEQ B

BEST.
STANDORT :
BREITE/N+S- 54,1500 RUN

LAENGE/W+E- -17,5500 RUN

ZIELORT :
BREITE/N+S- RUN

KURS= 35,0000 RUN

SM= 25,0000 RUN

ZIELORT :
BREITE/N+S- 54,3529 RUN

LAENGE/W+E- -18,1939 RUN

Umrechnung der Länge in Zeit

XEQ I

LIZ 16,1432 RUN

10458

Umrechnung der Zeit in Länge

XEQ J

ZIL 31233 RUN

48.0815

Distanz

XEQ D

DIST.
HOEHE <M> 60,00 RUN

AH ? 2,00 RUN

Δ? <HMS> ,42 RUN

DIST=2,64 SM

XEQ D

DIST.
HOEHE <M> 72,00 RUN

AH ? 2,00 RUN

Δ? <HMS> 1,22 RUN

DIST=1,62 SM

Ralf Neuburger
Auf der Höhe 15
2102 Hamburg 93

M-Code Utilities

Programm	Zeilen
RANDOM	97
SORT	147
FINDAD	52

Ich möchte jetzt drei Maschinenprogramme vorstellen. Die Maschinensprache auf dem HP-41 führt ja in unserem Club ein Schattendasein. Dies ist mir eigentlich sehr unverständlich, da der CCD immerhin ein Modul entwickelt hat und die Erfahrungen die dabei gemacht worden sind, sicher auch für andere interessant sein könnten.

Doch jetzt zu den Programmen:

RANDOM – Erzeugen von Zufallszahlen in Register

Eingabe einer Zahl in das X-Register, mit folgendem Format bbb.eee, wobei bbb die Speicherregister-Anfangsadresse und eee die Speicherregister-Endadresse bezeichnet, XEQ'RANDOM.

In dem Speicherregister bbb muß sich die Startzahl für den Zufallszahlengenerator befinden. Die Zufallszahlen werden nach folgender Formel errechnet: $x = FRC(x \cdot 9821 + .211327)$

SORT – Sortieren von Registerinhalten

Eingabe einer Zahl in das X-Register, mit folgendem Format bbb.eee, wobei bbb die Speicherregister-Anfangsadresse und eee die Speicherregister-Endadresse bezeichnet, XEQ'SORT.

Die Registerinhalte werden nach der SHELL,METZNER Methode sortiert. Im Programmablauf habe ich mich streng an das Flußdiagramm von Marcel Trimborn aus PRISMA 83.9.08 gehalten. Es werden immer alle 56 BIT miteinander verglichen, deshalb können keine Zahlen sortiert werden.

FINDAD – Finden von Programmadressen

Mit dieser nichtprogrammierbaren Funktion erhält man die hexadezimale Adresse einer HP-41 Funktion.

Eingabe: XEQ'FINDAD, (ALPHA)
FINDAD (ALPHA)
Ausgabe: FOBC

```

FOB6 084 "D"
FOB7 001 "A"
FOB8 004 "D"
FOB9 00E "N"
FOBA 009 "I"
FOBB 106 "F"
FOBC 000 NOP
FOBD 278 READ 9(Q)
FOBE 158 M=C ALL
FOBF 315
FOC0 098 ?NCXQ 26C5
FOC1 2EE ?CO ALL
FOC2 381
FOC3 00A ?NCGO 02E0
FOC4 33C RCR 1
FOC5 05A C=0 M
FOC6 0FC RCR 10
FOC7 10E A=C ALL
FOC8 0E0 SLCT Q
FOC9 2DC R= 13
FOCA 0A0 SLCT P
FOCB 15C R= 6
FOCC 3F2 LSHFA P-Q
FOCD 3D4 R=R-1
FOCE 394 ?R= 0
FOCF 3EB JNC FOCC -03
FOD0 04E C=0 ALL
FOD1 130 LDI S&X
FOD2 049
FOD3 23C RCR 2
FOD4 0EE C<>B ALL
FOD5 04E C=0 ALL
FOD6 130 LDI S&X
FOD7 03A
FOD8 23C RCR 2
FOD9 15E A=A+C MS
FODA 31A ?A<C M
FODB 01F JC FODE +03
FODC 19A A=A-B M
FODD 17E A=A+1 MS
FODE 0AE A<>C ALL
    
```

```

FODF 37C RCR 12
FOE0 2FE ?CO MS
FOE1 01F JC FOE4 +03
FOE2 0AE A<>C ALL
FOE3 3B3 JNC FOD9 -0A
FOE4 0E0 SLCT Q
FOE5 09C R= 5
FOE6 04A C=0 R<
FOE7 17C RCR 6
FOE8 23E C=C+1 MS
FOE9 0E8 WRIT 3(X)
FOEA 3E0 RTN

F17E 094 "T"
F17F 012 "R"
F180 00F "0"
F181 013 "S"
F182 0FB READ 3(X)
F183 38D
F184 008 ?NCXQ 02E3
F185 1BC RCR 11
F186 10E A=C ALL
F187 378 READ 13(c)
F188 21A C=C+A M
F189 070 N=C ALL
F18A 0FB READ 3(X)
F18B 10E A=C ALL
F18C 2A0 SETDEC
F18D 356 ?AO XS
F18E 027 JC F192 +04
F18F 3FA LSHFA M
F190 1A6 A=A-1 S&X
F191 3F3 JNC F18F -02
F192 130 LDI S&X
F193 003
F194 146 A=A+C S&X
F195 0AE A<>C ALL
    
```

```

F196 260 SETHEX
F197 38D
F198 008 ?NCXQ 02E3
F199 10E A=C ALL
F19A 378 READ 13(c)
F19B 03C RCR 3
F19C 146 A=A+C S&X
F19D 130 LDI S&X
F19E 200
F19F 306 ?A<C S&X
F1A0 381
F1A1 00A ?NCGO 02E0
F1A2 0B0 C=N ALL
F1A3 03C RCR 3
F1A4 1C6 A=A-C S&X
F1A5 166 A=A+1 S&X
F1A6 0AE A<>C ALL
F1A7 268 WRIT 9(Q)
F1A8 1EE C=C+C ALL
F1A9 1EE C=C+C ALL
F1AA 1EE C=C+C ALL
F1AB 3CE RSHFC ALL
F1AC 158 M=C ALL
F1AD 0B0 C=N ALL
F1AE 046 C=0 S&X
F1AF 10E A=C ALL
F1B0 03C RCR 3
F1B1 146 A=A+C S&X
F1B2 0AE A<>C ALL
F1B3 070 N=C ALL
F1B4 0B0 C=N ALL
F1B5 10E A=C ALL
F1B6 270 RAM SLCT
F1B7 198 C=M ALL
F1B8 146 A=A+C S&X
F1B9 038 READ 0(T)
F1BA 0AE A<>C ALL
F1BB 270 RAM SLCT
F1BC 038 READ 0(T)
F1BD 30E ?A<C ALL
F1BE 19F JC F1F1 +33
    
```

F1BF 0EE C<>B ALL	F1FD 1A6 A=A-1 S&X	F23B 0AE A<>C ALL
F1C0 0AE A<>C ALL	F1FE 0B0 C=N ALL	F239 226 C=C+1 S&X
F1C1 2F0 WRITE DATA	F1FF 306 ?A<C S&X	F23A 070 N=C ALL
F1C2 198 C=M ALL	F200 353 JNC F1EA -16	F23B 03C RCR 3
F1C3 10E A=C ALL	F201 198 C=M ALL	F23C 270 RAM SLCT
F1C4 01A A=0 M	F202 05A C=0 M	F23D 038 READ 0(T)
F1C5 0B0 C=N ALL	F203 1EE C=C+C ALL	F23E 0AE A<>C ALL
F1C6 05A C=0 M	F204 1EE C=C+C ALL	F23F 2A0 SETDEC
F1C7 1BC RCR 11	F205 1EE C=C+C ALL	F240 04E C=0 ALL
F1C8 15A A=A+C M	F206 3CE RSHFC ALL	F241 35C R= 12
F1C9 0AE A<>C ALL	F207 0AE A<>C ALL	F242 250 LD&R 9
F1CA 158 M=C ALL	F208 198 C=M ALL	F243 210 LD&R 8
F1CB 198 C=M ALL	F209 046 C=0 S&X	F244 090 LD&R 2
F1CC 03C RCR 3	F20A 206 C=C+A S&X	F245 050 LD&R 1
F1CD 0A6 A<>C S&X	F20B 158 M=C ALL	F246 130 LDI S&X
F1CE 198 C=M ALL	F20C 00E A=0 ALL	F247 003
F1CF 1C6 A=A-C S&X	F20D 306 ?A<C S&X	F248 135
F1D0 0B0 C=N ALL	F20E 2EF JC F1EB -23	F249 060 ?NCXQ 184D
F1D1 03C RCR 3	F20F 3E0 RTN	F24A 0AE A<>C ALL
F1D2 306 ?A<C S&X		F24B 3FA LSHFA M
F1D3 0CF JC F1EC +19		F24C 1A6 A=A-1 S&X
F1D4 0A6 A<>C S&X		F24D 3F3 JNC F24B -02
F1D5 270 RAM SLCT	F210 08D "M"	F24E 04E C=0 ALL
F1D6 0A6 A<>C S&X	F211 00F "0"	F24F 0A0 SLCT P
F1D7 038 READ 0(T)	F212 004 "D"	F250 35C R= 12
F1D8 0AE A<>C ALL	F213 00E "N"	F251 090 LD&R 2
F1D9 32E ?A<B ALL	F214 001 "A"	F252 050 LD&R 1
F1DA 097 JC F1EC +12	F215 012 "R"	F253 050 LD&R 1
F1DB 1DB C<>M ALL	F216 0FB READ 3(X)	F254 090 LD&R 2
F1DC 03C RCR 3	F217 38D	F255 090 LD&R 2
F1DD 270 RAM SLCT	F218 008 ?NCXQ 02E3	F256 1D0 LD&R 7
F1DE 1BC RCR 11	F219 1BC RCR 11	F257 21C R= 2
F1DF 1DB C<>M ALL	F21A 10E A=C ALL	F258 250 LD&R 9
F1E0 0AE A<>C ALL	F21B 378 READ 13(c)	F259 250 LD&R 9
F1E1 2F0 WRITE DATA	F21C 21A C=C+A M	F25A 250 LD&R 9
F1E2 198 C=M ALL	F21D 070 N=C ALL	F25B 21A C=C+A M
F1E3 05A C=0 M	F21E 0FB READ 3(X)	F25C 043 JNC F264 +08
F1E4 0AE A<>C ALL	F21F 10E A=C ALL	F25D 0AE A<>C ALL
F1E5 1BC RCR 11	F220 2A0 SETDEC	F25E 35C R= 12
F1E6 15A A=A+C M	F221 356 ?A0 XS	F25F 3FA LSHFA M
F1E7 0AE A<>C ALL	F222 027 JC F226 +04	F260 1A6 A=A-1 S&X
F1E8 158 M=C ALL	F223 3FA LSHFA M	F261 342 ?A0 SR
F1E9 313 JNC F1CB -1E	F224 1A6 A=A-1 S&X	F262 3EB JNC F25F -03
F1EA 253 JNC F1B4 -36	F225 3F3 JNC F223 -02	F263 0AE A<>C ALL
F1EB 213 JNC F1AD -3E	F226 130 LDI S&X	F264 05E C=0 MS
F1EC 198 C=M ALL	F227 003	F265 2F0 WRITE DATA
F1ED 03C RCR 3	F228 146 A=A+C S&X	F266 0AE A<>C ALL
F1EE 270 RAM SLCT	F229 0AE A<>C ALL	F267 260 SETHEX
F1EF 0CE C=B ALL	F22A 260 SETHEX	F268 0B0 C=N ALL
F1F0 2F0 WRITE DATA	F22B 38D	F269 23A C=C+1 M
F1F1 0B0 C=N ALL	F22C 008 ?NCXQ 02E3	F26A 070 N=C ALL
F1F2 226 C=C+1 S&X	F22D 10E A=C ALL	F26B 0E6 C<>B S&X
F1F3 070 N=C ALL	F22E 378 READ 13(c)	F26C 03C RCR 3
F1F4 04E C=0 ALL	F22F 03C RCR 3	F26D 270 RAM SLCT
F1F5 270 RAM SLCT	F230 146 A=A+C S&X	F26E 0AE A<>C ALL
F1F6 0B0 C=N ALL	F231 130 LDI S&X	F26F 326 ?A<B S&X
F1F7 03C RCR 3	F232 200	F270 277 JC F23E -32
F1F8 0A6 A<>C S&X	F233 306 ?A<C S&X	F271 3E0 RTN
F1F9 27B READ 9(Q)	F234 381	
F1FA 146 A=A+C S&X	F235 00A ?NCGO 02E0	
F1FB 198 C=M ALL	F236 0B0 C=N ALL	
F1FC 1C6 A=A-C S&X	F237 15A A=A+C M	

Bernhard Saalfeld (3037)
Karl-Höher-Str. 4
5000 Köln 80

Kleiner Texteditor

100 Zeilen, SIZE 011, HP-41CX, RUS ERAMCO SYSTEMS, ADVANTAGE PAC

Dieses ist ein kleines Textverarbeitungs-Programm.

Es ist dafür gedacht, wichtige Texte in den Rechner einzugeben und so schnell wie möglich wieder an den Text heranzukommen. Diese Schnelligkeit brauchte ich, um in einer Prüfung sofort an die gewünschte Information heran zu kommen.

Programmbeschreibung:

Es wird in der RSU ein ASCII-File eröffnet.

Ich hatte es „PS“ genannt. Wenn es nicht „PS“ heißt, muß in Zeile 2 des Programmes der neue Name eingeschrieben werden.

In die erste Zeile des Files kommt der erste Name.

In die zweite Zeile des Files der erste Text-Abschnitt.

Für jeden neu angelegten Text wird ein neuer Name in eine Zeile, über dem Text angelegt.

Beispiel:

```
01 Name 1
02 Text
"
"
"
09 Name 2
10 Text
"
"
```

Die Namen werden in ein ASCII-File vom HP-41CX eingeschrieben. Dazu wird ein ASCII-File eröffnet mit dem Namen „EDTN“.

Es kann auch ein anderer Name sein, dann muß in Zeile 50 der neue Name eingeschrieben werden.

In dieses ASCII-File kommen nur die Namen vom Text, die sich in der RSU befinden.

Zwischen jedem Namen muß sich eine Leer-Zeile befinden. In diese Zeile kommen dann die Zeilennummern der Namen. Die schreibt sich der Rechner selbst.

In die letzte Zeile des Files kommt „EOF“, das wird benötigt damit der Rechner weiß, daß das File zu Ende ist.

Beispiel:

```
00 Name 1
02
03 Name 2
04
05 EOF
```

Wenn der Text eingegeben ist, und die Namen ebenfalls, wird das Unterprogramm „EDTC“ aufgerufen.

Das Unterprogramm sucht jeden Namen im RSU FILE und schreibt seine Position in die Leerzeile vom „EDTN“-File.

Es gibt auch eine Fehler Meldung, “ = NAM.-FEH.“ die besagt, daß sich ein Name im „EDTN“-File befindet, den es aber nicht im „PS“-File gibt.

Der falsche Name wird links an die Meldung angehängt.

Das Unterprogramm „EDTN“ zeigt die einzelnen Namen an, und wartet etwa 1.5 Sekunden auf einen Tastendruck. Wenn keine Taste gedrückt wird, wird der nächste Name angezeigt.

Die Wartezeit ist abhängig von Programmzeile 35.

Wenn eine Taste gedrückt wird, außer R/S, springt das Programm in das „PS“-File, an die Stelle, an der der Name steht. Das Springen geht sehr schnell, da der Rechner die Zeilennummer für den bestimmten Namen kennt.

Das Unterprogramm „EDTF“ wartet auf einen Namen. Wenn der Name eingegeben ist, wird die R/S-Taste gedrückt.

Der Rechner sucht dann Zeile für Zeile den „PS“-File ab. Das kann lange dauern, wenn es der letzte Name im Text ist, und das File sehr groß ist.

Ich hatte aus dem ADVANTAGE PAC den Befehl „AIP“ benutzt. Beim CCD-Modul ist es „ARCLI“.

Globale Label: 00,01,02,03,04,05,06
Speicher: 09,10

```
01#LBL "EDTF"          40 ANUM                79 GTD 00
02 "PS"                41 SRCPT                80"!=NAM.-FEH."
03 RSTAF               42 EDT                  81 AVIEW
04 IRCPT               43 GTD 02              82 GTD 02
05#LBL 01              44#LBL 00              83#LBL 00
06 "NAME?"            45 DSE 10              84 RRCPT
07 ADN                 46 GTD 04              85 E
08 STQP                47 CLD                 86 ST+ 09
09 AOFF                48 GTD 02              87 RCL 09
10 CLST                49#LBL 06              88 INT
11#LBL 03              50 "EDTN"              89 SEEKPT
12 CF 27               51 CLX                 90 RCL Z
13 E                   52 SEEKPTA            91 DELREC
14 STO Y               53 "EOF"               92 CLA
15 FINDPS              54 POSFL               93 AIP
16 GTD 00              55 E                   94 INSREC
17 "NAME FEHLER"      56 -                   95 ISG 09
18 AVIEW               57 STO 10              96 GTD 05
19 PSE                 58 CLX                 97#LBL 02
20 CLD                 59 SEEKPT              98 CLA
21 GTD 01              60 "PS"                99 CLST
22#LBL 00              61 RSTAF                100 END
23 EDT                 62 IRCPT
24 GTD 02              63 RTN
25#LBL"EDTN"           64#LBL"EDTC"
26 XEQ 06              65 XEQ 06
27 GTD 00              66 RCL 10              Anstatt AIP
28#LBL 04              67 E3                  FIX 0
29 E                   68 /                   CF 29
30 ST- 10              69 STO 09              ARCL X
31 GETREC              70#LBL 05              FIX 4
32#LBL 00              71"PS"                 SF 29
33 GETREC              72 RSTAF
34 AVIEW               73 IRCPT
35 1,5                 74 RCL 09
36 GETKEYX             75 GETREC
37 X=0?                76 E
38 GTD 00              77 STO Y
39 GETREC              78 FINDPS
```

Markus Eißner
Friedrich-Ebert-Str. 93
6106 Erzhäusen

Kleinbuchstabenumschaltung in M-Code

19 Zeilen, 19 Bytes, 3 Regs., SIZE 000, HP-41C, CCD

Die beiden Programme schalten den Kleinbuchstabenmodus des CCD-Moduls aus oder ein. Das dafür zuständige Bit liegt im Systemregister c, vorderes Bit von Byte 4 bzw. von Nibble 9.

```
LBL „KLBU“ Dieses bereits schon einmal veröffentlichte
13.4 Programm bringt leider den Stack durcheinander.
PEEKB Also habe ich kurzerhand ein kurzes Maschinen-
128 programm geschrieben, das die obige Aufgabe
XOR schneller erledigt ohne die erwähnten Nachteile.
POKEB
END
```

Das Maschinenprogramm ist verschiebbar, die Adressen sind nur beispielhaft. Die Anfangsadresse für die Adress-Liste ist BDC0.

Die Mnemonics sind in 2 Schreibweisen angegeben, damit die Anwender auf jeden Fall zurecht kommen.

Dies ist übrigens mein erster Gehversuch auf der M-Code-Ebene. Vielleicht lassen sich andere M-Code Anhänger ermutigen, ihre kurzen Routinen ebenfalls zu veröffentlichen.

Jeder fängt einmal klein an . . .

Adr. Wort Mnemonics (Variante)

```
BDBC 095 U: Letztes Zeichen des Namens (80h addiert)
BDBD 002 B
BDBE 00C L
BDBF 00B K
```

```
BDC0 04E Σ C=0 ALL C=0 ALL
BDC1 3F0 | PFAD=C PRPHSLCT
BDC2 270 | DADD=C RAM SLCT
BDC3 378 | C=REGN 13(c) READ 13(c)
BDC4 27C μ RCR 9 RCR 9
BDC5 3D8 | CST EX C<>ST
BDC6 00C L ?S=1 3 ?FSET 3
BDC7 01F _ GOC **+03 JC BDCA
BDC8 008 H S=1 3 SETF 3
BDC9 013 S GONC **+02 JNC BDCB
BDCA 004 D S=0 3 CLRf 3
BDCB 3D8 | CST EX C<>ST
BDCC 0BC < RCR 5 RCR 5
BDCE 3E0 | RTN RTN
```

Systemregister c holen

Um 9 Nibbles rechts rotieren

C gegen ST austauschen

Bit 3 in ST gesetzt?

Falls ja, springe auf BDCA

Setze Bit 3

Springe auf BDCB

Lösche Bit 3

C gegen ST zurücktauschen

Um 5 Nibbles rechts rotieren (ursprüngliche Ordnung)

und ins c-Register schreiben

Fertig

Walter Lutz
Günthersburgallee 66
6000 Frankfurt/Main 00

Produkt & Quotientenregel

169 Zeilen, 273 Bytes, 39 Regs., SIZE 2(n + m) + 9

n = Grad 1. Polynom

m = Grad 2. Polynom

Das Programm wurde entwickelt, um das Ableiten und umständliche Ausmultiplizieren von Polynomen (Funktionen der Form $f(x) = a_n x^n + a_{n-1} x^{n-1} \dots + a_1 x a_0$; n ist zugleich der Grad des Polynoms) zu vereinfachen. Das Programm war ursprünglich nur für die Produktregel konzipiert, da sich bei der Quotientenregel für den Zähler aber nur ein Vorzeichen ändert, kann das Programm auch diesen berechnen. Das Quadrieren des Nenners bleibt hier jedoch dem Anwender vorbehalten. Weiterhin ist zu beachten, daß sich aus Zähler und Nenner hin und wieder ein gemeinsamer Faktor ausklammern und kürzen läßt – das kann das Programm nicht.

Bedienung:

Zunächst die Anzahl der benötigten Datenregister wählen. Haben die beiden Polynome die Grade n und m, so ist SIZE 2(n + m) + 9 einzustellen.

1. Programm mit XEQ „QR“ (= Quotientenregel) oder XEQ „PR“ (= Produktregel) starten.
2. Jetzt fragt der Rechner nach dem Grad des ersten und zweiten Polynoms („GRAD P1 =?“ bzw. „GRAD P2 =?“). Dazu wird eine positive ganze Zahl eingegeben, und mit R/S gestartet.

3. Der Rechner beginnt mit der Abfrage der Koeffizienten. Dabei braucht der Koeffizient 0 (= nicht vorhanden) nicht eingegeben werden. Hier genügt ein einfaches R/S. Wurde ein falscher Wert eingegeben und bereits mit R/S gestartet, muß wieder bei Schritt 1 begonnen werden.
4. Sind alle Koeffizienten eingegeben, beginnt das Programm automatisch mit der Bearbeitung. Die Koeffizienten der Lösung (bei QR: des Zählers) werden dann mit dem Hinweis „d/dX“ im X-Register ausgegeben.

Wer gerne auf synthetische Befehle verzichtet, kann 62 STO d durch sein bevorzugtes Anzeigeformat (z.B. FIX 4, SF 29), 61 ARCL Y durch 61 ARCL X ersetzen, sowie 58 RCL d streichen.

Für den trivialen Fall Grad P1 = Grad P2 = 0 arbeitet das Programm nicht korrekt. Aber das sollte die Anwendbarkeit des Programms nicht ernstlich beeinträchtigen.

Produktregel:

Wenn $f(x) = u(x) \cdot v(x)$,
so ist: $f'(x) = u'(x) \cdot v(x) + u(x) \cdot v'(x)$.

Quotientenregel:

Wenn $f(x) = u(x)/v(x)$,
so ist: $f'(x) = [u'(x) \cdot v(x) - u(x) \cdot v'(x)]/v^2(x)$.

Beispiel zur Produktregel:

$$f(x) = (-2x^2 + x + 4)(x^3 + x^2 - 1)$$

$$u(x) = -2x^2 + x + 4$$

$$v(x) = x^3 + x^2 - 1$$

$$f(x) = -10x^4 - 4x^3 + 15x^2 + 12x - 1$$

Beispiel zur Quotientenregel:

$$f(x) = \frac{x^3 - 2x^2 - 13x - 10}{x^2 - 9}$$

$$u(x) = x^3 - 2x^2 - 13x - 10$$

$$v(x) = x^2 - 9$$

$$f(x) = \frac{x^4 - 14x^2 + 56x + 117}{(x^2 - 9)^2}$$

```

01*LBL "QR"      49 RCL 05
02 SF 00         50 RCL 01
03 GTO 08       51 RCL 03
04*LBL "PR"     52*LBL 00
05 CF 00        53 +
06*LBL 08       54 X(>)Y
07 CF 01        55 -
08 CLRG         56 INT
09 CLST         57 "t: Xt"
10 "GRAD P"     58 RCL d
11 ASTO 00      59 FIX 0
12 "t1 =?"     60 CF 29
13 PROMPT       61 ARCL Y
14 STO 03       62 STO d
15 CLA          63 "t ="
16 ARCL 00      64 FC?C 01
17 "t2 =?"     65 "t?"
18 PROMPT       66 CLST
19 STO 04       67 AVIEW
20 RCL 03       68 RTN
21 +            69 STO IND 01
22 6.006       70 DSE 01
23 +            71 GTO 01
24 STO 00       72 RCL 05
25 INT          73 STO 01
26 RCL X        74*LBL 02
27 E3           75 "P2"
28 /            76 RCL 06
29 +            77 RCL 02
30 RCL 03       78 RCL 04
31 +            79 XEQ 00
32 1            80 RTN
33 +            81 STO IND 02
34 STO 01       82 DSE 02
35 STO 05       83 GTO 02
36 INT          84 CLX
37 RCL X        85 STO 07
38 E3           86*LBL 03
39 /            87 RCL 06
40 +            88 STO 02
41 RCL 04       89 RCL 03
42 +            90 RCL 05
43 1            91 RCL 01
44 +            92 -
45 STO 02       93 -
46 STO 06       94 RCL IND 01
47*LBL 01       95 *
48 "P1"         96 ENTER↑
    
```

```

97 ENTER↑
98*LBL 04
99 RCL IND 02
100 *
101 ST+ IND 00
102 CLX
103 1
104 ST- 00
105 RDN
106 DSE 02
107 GTO 04
108 RCL 04
109 ST+ 00
110 DSE 01
111 GTO 03
112 RCL 06
113 STO 02
114 RCL 03
115 1
116 +
117 ST+ 00
118*LBL 05
119 RCL 05
120 STO 01
121 RCL 04
122 RCL 06
    
```

	XEQ "PR"		XEQ "QR"
GRAD P1 =?		GRAD P1 =?	
	2.00 RUN		3.00 RUN
GRAD P2 =?		GRAD P2 =?	
	3.00 RUN		2.00 RUN
P1: Xt2 =?		P1: Xt3 =?	
	-2.00 RUN		1.00 RUN
P1: Xt1 =?		P1: Xt2 =?	
	1.00 RUN		-2.00 RUN
P1: Xt0 =?		P1: Xt1 =?	
	4.00 RUN		13.00 RUN
P2: Xt3 =?		P1: Xt0 =?	
	1.00 RUN		-10.00 RUN
P2: Xt2 =?		P2: Xt2 =?	
	1.00 RUN		1.00 RUN
P2: Xt1 =?		P2: Xt1 =?	
	RUN		RUN
P2: Xt0 =?		P2: Xt0 =?	
	-1.00 RUN		-9.00 RUN
d/dX: Xt4 =		d/dX: Xt4 =	
	-10.00 ***		1.00 ***
	RUN		RUN
d/dX: Xt3 =		d/dX: Xt3 =	
	-4.00 ***		0.00 ***
	RUN		RUN
d/dX: Xt2 =		d/dX: Xt2 =	
	15.00 ***		-40.00 ***
	RUN		RUN
d/dX: Xt1 =		d/dX: Xt1 =	
	12.00 ***		56.00 ***
	RUN		RUN
d/dX: Xt0 =		d/dX: Xt0 =	
	-1.00 ***		-117.00 ***
	RUN		RUN
NONEXISTENT		NONEXISTENT	

```

123 RCL 02
124 -
125 -
126 RCL IND 02
127 *
128 FS? 00
129 CHS
130 ENTER↑
131 ENTER↑
132*LBL 06
133 RCL IND 01
134 *
135 ST+ IND 00
136 CLX
137 1
138 ST- 00
139 RDN
140 DSE 01
141 GTO 06
142 RCL 03
143 ST+ 00
144 DSE 02
145 GTO 05
146 RCL 04
147 ST+ 00
148 RCL 03
    
```

```

149 +
150 1
151 ST+ 00
152 -
153 STO 01
154 RCL 00
155 STO 02
156 CF 00
157*LBL 07
158 SF 01
159 "d/dx"
160 RCL 02
161 RCL 00
162 RCL 01
163 XEQ 00
164 RCL IND 00
165 RTN
166 DSE 00
167 GTO 07
168 CF 30
169 END
    
```

Ralf Pfeifer (116)
 Rubensstraße 5
 5000 Köln 50

Schreibschriftlineal für den THINKJET

82 Zeilen, 264 Bytes, 38 Regs., SIZE 003, HP-41C, IL, THINK-JET

Schreibschrift(Pitch) Lineal für ThinkJet-Drucker

Am ThinkJet-Drucker stehen vier Zeichenbreiten zur Verfügung:

	Zeichen/inch = pitch	Zeichen/Zeile
normal	12,0	80
expanded	6,2	48
compressed	21,3	142
expanded-compressed	10,7	71

Um sich die Arbeit bei der Formatbildung zu erleichtern wurde dieses Programm zur Herstellung eines Pitch-Lineals entwickelt.

Nach Ablauf des Programmes ist F17 von Hand aus zu löschen!

Den Ausdruck auseinanderschneiden, vorne und hinten auf zwei Streifen von Karton, Plexi- oder Acrylglas aufkleben und mit selbstklebender klarer Schutzfolie überziehen. Eventuell am Rande lochen und mit einem Schlüsselring verbinden.

```

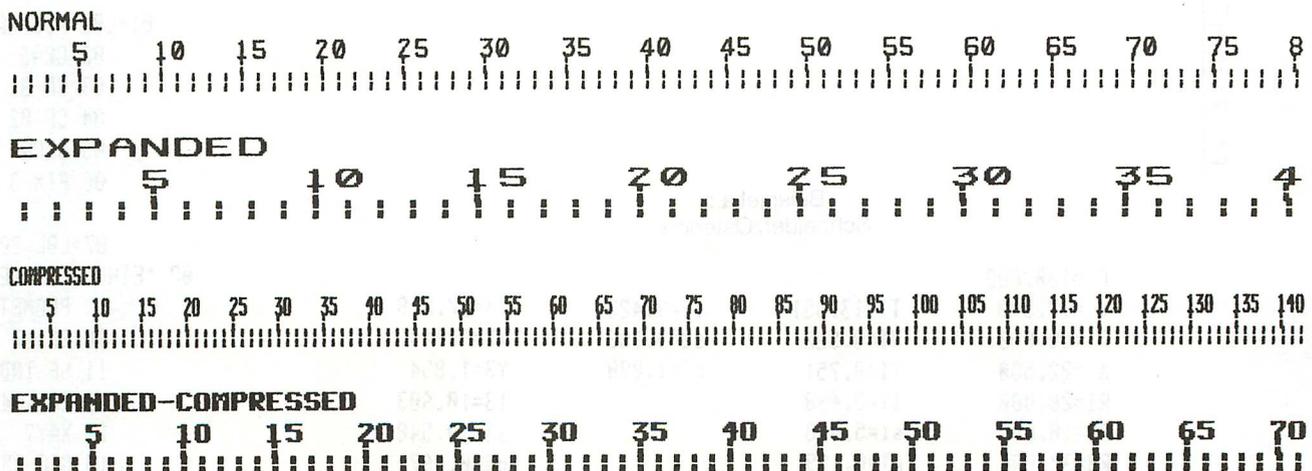
01+LBL "TJPITCH"      31 OUTA
02 FIX 0              32 "COMPRESSEDΔ"
03 SF 17              33 OUTA
04 SF 25              34 5,07105
05 CF 29              35 STO 01
06 14                 36+LBL 01
07 ACCHR              37 RCL 01
08 "E&k"              38 INT
09 ASTO 00            39 1,968
10 5,08005            40 -
11 STO 01             41 ACLX
12 16                 42 FC? 25
13 STO 02             43 XEQ 03
14 "+0SNORMALΔ"      44 RCL 01
15 OUTA               45 ACX
16 XEQ 01             46 "Δ"
17 "+1SEXPANDEDΔ"    47 OUTA
18 OUTA               48 ISG 01
19 5,04005            49 GTO 01
20 STO 01             50 "E="
21 XEQ 01             51 ACA
22 "+2SCOMPRESSEDΔ"  52 4,032
23 OUTA               53 1,124
24 SF 00              54+LBL 02
25 5,14205            55 X<>Y
26 STO 01             56 ACLX
27 28                 57 X<>Y
28 STO 02             58 ACLX
29 XEQ 01             59 DSE 02
30 "+3SEXPANDED-"    60 GTO 02
61 "ΔE="              62 OUTA
63 15                 64 STO 02
65 99,124             66 ACLX
67 43,124             68 FS?C 00
69 ACLX               70 PRBUF
71 ""                 71 ""
72 ARCL 00            72 ARCL 00
73 RTN                73 RTN
74+LBL 03              74+LBL 03
75 SF 25              75 SF 25
76 99                 76 99
77 -                  77 -
78 ACLX               78 ACLX
79 99,032             79 99,032
80 ACLX               80 ACLX
81 RDN                81 RDN
82 END                82 END
    
```

TJPITCH

```

-----
008:F3 1B 26 6B
014:F8 7F 30 53 4E 4F 52
4D 41 4C 0A 0D
017:FD 7F 31 53 45 58 50
41 4E 44 45 44 0D 0A
022:FF 7F 32 53 43 4F 4D
50 52 45 53 53 45 44 0A
0D
032:FC 43 4F 4D 50 52 45
53 53 45 44 0A 0D
046:F1 0D
050:F2 1B 3D
061:F3 0D 1B 3D
071:F1 0A
    
```

Dr. Martin Hochenegger
 Heidelberger Landstraße 97
 6100 Darmstadt 13



Konstruktion für die Eckausrundungen

335 Zeilen, 669 Bytes, 95 Regs., SIZE 027, HP-41CV, (IL), PRINTER

Nach Schneider/Osterloh und RAL-K 1976)

Folgende Forderungen werden erfüllt:

1. Berechnung 2. Arten von Schleppkurven

a. Schneider/Osterloh

$$R_1 = 2,5 \times R_2 \quad R_3 = 5,5 \times R_2$$

b. RAL-K

$$R_1 = 2,0 \times R_2 \quad R_3 = 3,0 \times R_2$$

2. Berechnung aller Bogenhauptpunkte

$$T_1, X_1, Y_1, l_1, s_1, p_1, l_2, s_2, p_2, T_3, X_3, Y_3, l_3, s_3, p_3$$

Programmbeschreibung:

1. Programmaufruf : XEQ SS/RAL

2. Die Meldung „EINGABE CODE?“ fordert zur Eingabe von

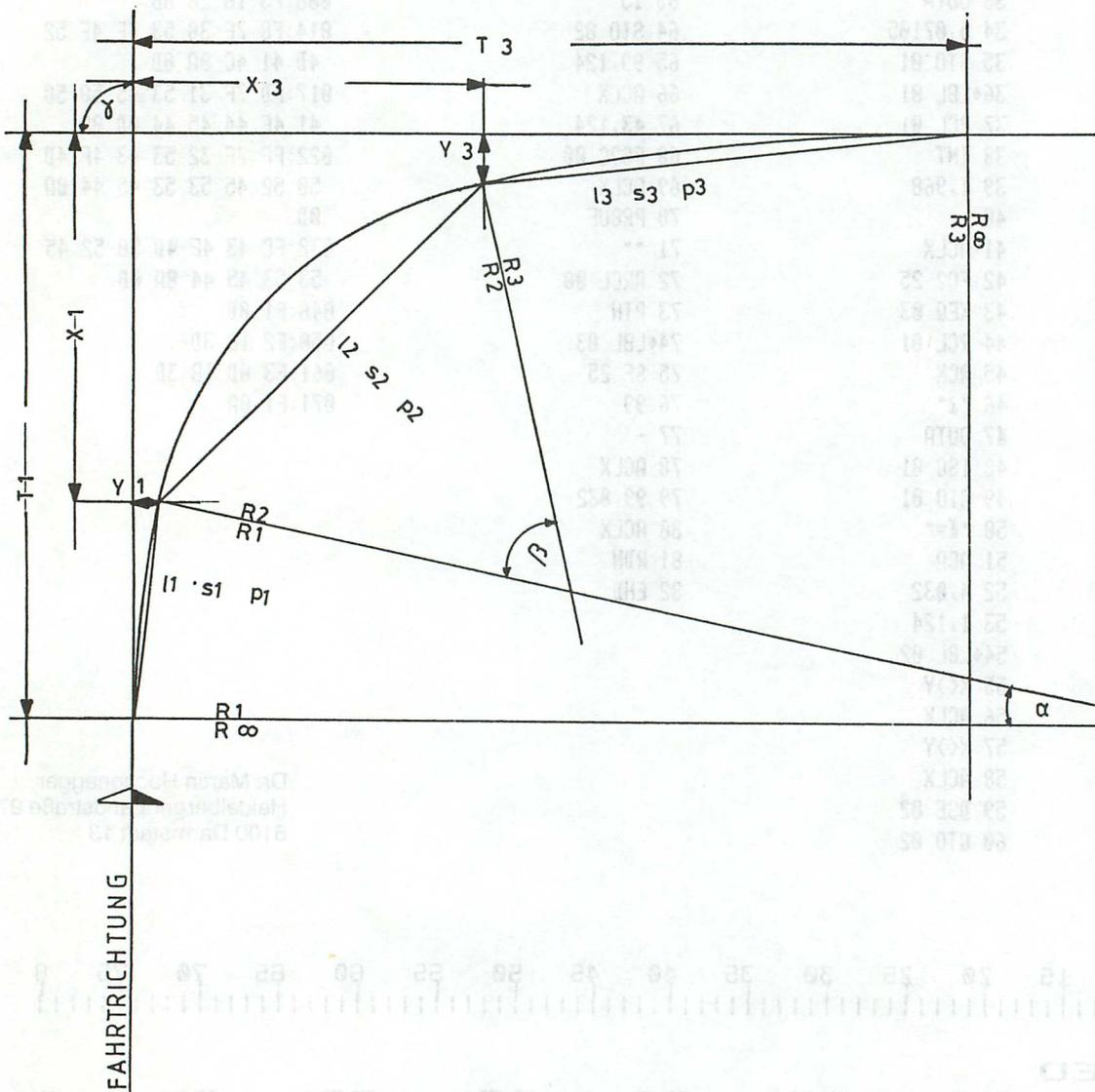
a. 1R/S für Schneider/Osterloh

b. 2R/S für RAL-K auf

c. Γ = Winkel Gamma R/S

d. R_2 = Mittleren Radius R/S

3. Nach jedem Ergebnis R/S drücken



Beispiel b.:
RAL-K

- $\Gamma = 100.000$
- $\alpha = 13.333$
- $\beta = 73.333$
- $\Delta = 26.667$
- $R_1 = 25.000$
- $R_2 = 10.000$
- $R_3 = 55.000$

- $T_1 = 14.102$
- $X_1 = 8.904$
- $Y_1 = 0.546$
- $l_1 = 5.236$
- $s_1 = 5.226$
- $p_1 = 0.137$

- $l_2 = 11.519$
- $s_2 = 10.893$
- $p_2 = 1.613$

- $T_3 = 19.684$
- $X_3 = 8.249$
- $Y_3 = 1.202$
- $l_3 = 11.519$
- $s_3 = 11.498$
- $p_3 = 0.301$

Beispiel a.:
Schneider/Osterloh

- $\Gamma = 100.000$
- $\alpha = 17.500$
- $\beta = 60.000$
- $\Delta = 22.500$
- $R_1 = 20.000$
- $R_2 = 10.000$
- $R_3 = 30.000$
- $T_1 = 13.951$
- $X_1 = 8.522$
- $Y_1 = 0.751$
- $l_1 = 5.498$
- $s_1 = 5.480$
- $p_1 = 0.189$
- $l_2 = 9.425$
- $s_2 = 9.080$
- $p_2 = 1.090$
- $T_3 = 17.298$
- $X_3 = 6.914$
- $Y_3 = 1.854$
- $l_3 = 10.603$
- $s_3 = 10.548$
- $p_3 = 0.467$

- 01*LBL "SS/RAL"
- 02 GRAD
- 03 CF 01
- 04 CF 02
- 05 CLRG
- 06 FIX 3

- 07*LBL 20
- 08 "EINGABE CODE?"
- 09 PROMPT
- 10 SF 27
- 11 SF IND X
- 12 RCL 24
- 13 X=Y?
- 14 GTO 15

15 CF IND 21	75 XEQ 03	130 RCL 15	191 -	249 COS	296*LBL I
16 X<Y	76 XEQ 04	131 X<Y	192 RCL 22	250 *	297 "12="
17 STO 24	77 XEQ 05	132 -	193 COS	251 -	298 RCL 09
	78 XEQ 06	133 STO 02	194 *	252 STO 11	299 XEQ 99
18*LBL 15	79 XEQ J	134 "R2="	195 -	253 RTN	300 "s2="
19 FS? 01	80 ADV	135 PROMPT	196 RCL 15		301 RCL 10
20 GTO 25	81 GTO 20	136 STO 03	197 TAN	254*LBL H	302 XEQ 99
21 FS? 02		137 2.5	198 /	255 "Γ ="	303 "p2="
22 GTO 26	82*LBL A	138 *	199 -	256 RCL 15	304 RCL 11
23 GTO 20	83 RCL 00	139 STO 04	200 STO 06	257 XEQ 99	305 XEQ 99
	84 STO 22	140 RCL 03	201 RTN	258 "α ="	306 RTN
	85 RCL 00	141 5.5		259 RCL 00	
24*LBL 25	86 STO 23	142 *	202*LBL 02	260 XEQ 99	307*LBL J
25 XEQ "X"		143 STO 05	203 RCL 20	261 "β ="	308 "T3="
26 XEQ A		144 RTN	204 RCL 22	262 RCL 02	309 RCL 06
27 XEQ B	87*LBL B		205 SIN	263 XEQ 99	310 XEQ 99
28 XEQ 01	88 RCL 04		206 *	264 "Δ ="	311 "X3="
29 XEQ 02	89 STO 20	145*LBL "Y"	207 -	265 RCL 01	312 RCL 07
30 XEQ 03	90 RCL 05	146 "Γ="	208 STO 07	266 XEQ 99	313 XEQ 99
31 XEQ 04	91 STO 21	147 PROMPT	209 RTN	267 "R1="	314 "Y3="
32 XEQ 05	92 RTN	148 STO 15		268 RCL 04	315 RCL 08
33 XEQ 06		149 17.5		269 XEQ 99	316 XEQ 99
34 XEQ H	93*LBL C	150 STO 00	210*LBL 03	270 "R2="	317 "13="
35 ADV	94 RCL 03	151 22.5	211 RCL 20	271 RCL 03	318 RCL 09
36 XEQ C	95 STO 20	152 STO 01	212 ENTER↑	272 XEQ 99	319 XEQ 99
37 XEQ 04	96 RCL 02	153 +	213 ENTER↑	273 "R3="	320 "s3="
38 XEQ 05	97 STO 22	154 -	214 RCL 22	274 RCL 05	321 RCL 10
39 XEQ 06	98 RTN	155 STO 02	215 COS	275 XEQ 99	322 XEQ 99
40 XEQ I		156 "R2="	216 *	276 ADV	323 "p3="
41 ADV	99*LBL D	157 PROMPT	217 -	277 "T1="	324 RCL 11
42 XEQ D	100 RCL 00	158 STO 03	218 STO 08	278 RCL 06	325 XEQ 99
43 XEQ E	101 STO 22	159 2	219 RTN	279 XEQ 99	326 RTN
44 XEQ 01		160 *		280 "X1="	
45 XEQ 02	102*LBL E	161 STO 04	220*LBL 04	281 RCL 07	327*LBL 99
46 XEQ 03	103 RCL 04	162 RCL 03	221 RCL 20	282 XEQ 99	328 ARCL X
47 XEQ 04	104 STO 21	163 3	222 RCL 22	283 "Y1="	329 FS? 21
48 XEQ 05	105 RCL 05	164 *	223 *	284 RCL 08	330 AVIEW
49 XEQ 06	106 STO 20	165 STO 05	224 200	285 XEQ 99	331 FC? 21
50 XEQ J	107 RTN	166 RTN	225 ENTER↑	286 "11="	332 PROMPT
51 ADV			226 PI	287 RCL 09	333 CF 01
52 GTO 20	108*LBL F	167*LBL 01	227 /	288 XEQ 99	334 CF 02
	109 RCL 00	168 RCL 20	228 /	289 "s1="	335 RTN
53*LBL 26	110 STO 22	169 RCL 03	229 STO 09	290 RCL 10	336 .END.
54 XEQ "Y"	111 RCL 01	170 -	230 RTN	291 XEQ 99	
55 XEQ F	112 STO 23	171 RCL 22		292 "p1="	
56 XEQ B	113 RTN	172 SIN	231*LBL 05	293 RCL 11	
57 XEQ 01		173 *	232 RCL 22	294 XEQ 99	
58 XEQ 02	114*LBL G	174 RCL 21	233 2	295 RTN	
59 XEQ 03	115 RCL 00	175 ENTER↑	234 /		
60 XEQ 04	116 STO 23	176 ENTER↑	235 SIN		
61 XEQ 05	117 RCL 01	177 RCL 03	236 RCL 20		
62 XEQ 06	118 STO 22	178 -	237 2		
63 XEQ H	119 RTN	179 RCL 23	238 *		
64 ADV		180 COS	239 *		
65 XEQ C	120*LBL "X"	181 *	240 STO 10		
66 XEQ 04	121 "Γ="	182 -	241 RTN		
67 XEQ 05	122 PROMPT	183 RCL 15			
68 XEQ 06	123 STO 15	184 SIN	242*LBL 06		
69 XEQ I	124 7.5	185 /	243 RCL 20		
70 ADV	125 /	186 +	244 ENTER↑		
71 XEQ G	126 STO 00	187 RCL 20	245 ENTER↑		
72 XEQ E	127 2	188 ENTER↑	246 RCL 22		
73 XEQ 01	128 *	189 ENTER↑	247 2		
74 XEQ 02	129 STO 01	190 RCL 03	248 /		

Klaus Müllhäuser
 Albert-Schweitzer-Str. 20
 7733 Mönchweiler

Der HP32S

In den USA habe ich zwei neue HP-Rechner entdeckt: Den HP-22S und den HP-23S. Der 32S ist wohl als Nachfolger der Serie 10 gedacht, leistungsmäßig liegt er zwischen den Modellen 11C und 15C mit einigen 16C Funktionen, und das zu Preisen zwischen 70 US\$ und 150 US\$.

Was zuerst auffällt, ist die gut zu bedienende Tastatur, und die große, sehr gut ablesbare Punktmatrix-LCD, über die der Rechner Funktionen und Programme im Klartext zeigt, ohne daß der Rechner aber über ALPHA-Funktionen verfügt. Der Rechenbereich entspricht den 28-Modellen mit 12-stelliger Genauigkeit zwischen E-499 und E+499.

Wie bei HP üblich, verschwinden in den unendlichen Weiten der 390 Bytes Programme (Eineinhalb Bytes pro Befehl sowie der Zahlen 0-99, für alle anderen Zahlen 9,5 Bytes), Daten (8 Bytes, wenn nicht =0), Statistik (48 Bytes), SOLVE (33,5 Bytes) und INTEGRATE (140 Bytes).

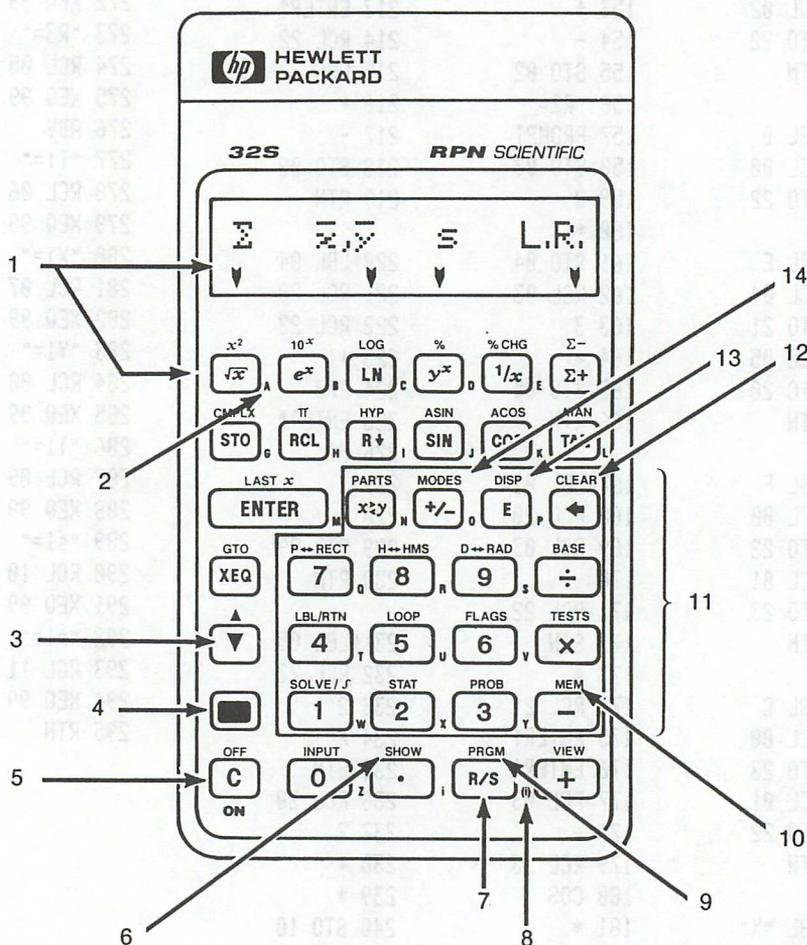
Auffällig ist auch die hohe Verarbeitungsgeschwindigkeit des 32S: Um einen Vergleich zu ermöglichen, habe ich mein Primzahlprogramm für den 41er aus BEST OF PRISMA, S. 92 an die Rechner 15C, 32S und 28S angepaßt.

Rechner HP-	15C	41CX	32S	28S
1 000 003	3:05	0:40	0:10	0:07
39 916 801	19:04	3:58	0:59	0:41
1 000 000 009	*	20:13	5:01	3:30

Zeit in min:sek.; * = Stoppuhr leider vor Testende eingeschlafen.

Alle Rechner führen die gleiche Anzahl von Tests durch, und obwohl der 28S wegen seiner Sprache das umständlichste Programm hat, ist er der schnellste.

Doch zurück zum 32S: Um viele Funktionen verfügbar zu machen, hat HP verschiedene Wege beschritten. Ich meine, daß der alte HP-67 immer noch die beste Lösung hatte: 4 Funktionen pro Taste machten den Rechner nicht gerade übersichtlich, aber jede Funktion war mit max. zwei Tastendrücken verfügbar. Die umständlichste Lösung bot der HP-41, bei dem die meisten Funktionen buchstabiert werden müssen, und der 28S bietet auch keine Besserung. In der Mitte der Bedienungsfreundlichkeit liegt der 32S. Führt man BASE aus, zeigt das Display „DEC HX OC BN“ sowie 4 Pfeile, die auf Tasten der obersten Reihe deuten. Betätigt man dann eine dieser Tasten, so wird die im Display angezeigte Funktion (z.B. HX für Hexadzimal-Modus)



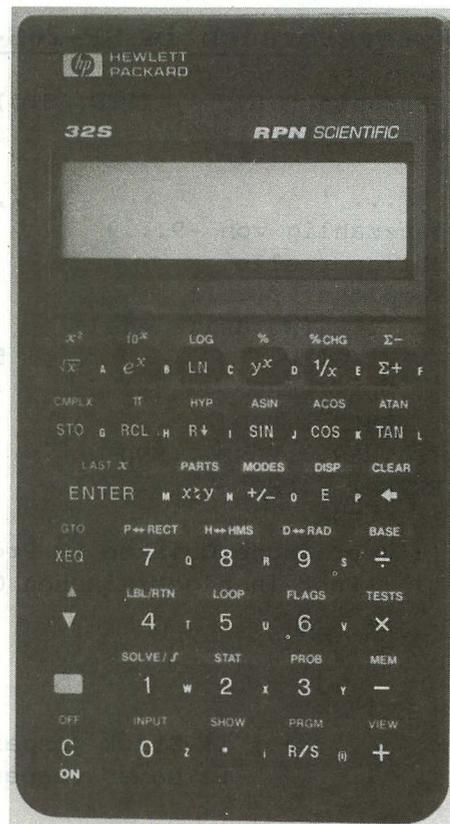
- 1. Menü und Menütasten
- 2. Buchstabetastentaste für Variable und Labels
- 3. Vorwärts/Rückwärts in Programmen und Listen
- 4. SHIFT-Taste
- 5. ON; bricht Anzeigen, Menüs und die Programmeingabe ab
- 6. Zeigt alle Dezimalstellen an
- 7. Start/Stop für Programme
- 8. für indirekte Adressierung über i
- 9. Toggelt in den Programmiermodus und wieder zurück
- 10. Benutzerspeicher; abgespeicherte Variable und Programme
- 11. Menütasten (eingerahmter Bereich)
- 12. Löscht den gesamten Speicher
- 13. Anzeigeformate
- 14. Winkelmodi, Dezimaltrennzeichen

statt der aufgedruckten (hier IN) ausgeführt, und das Menue verschwindet wieder. Manche Menues sind verschachtelt: TESTS läßt „x?y x?0“ erscheinen, auf Tastendruck verzweigt es dann zu den eigentlichen Tests (Der 32S hat davon 8 plus DSE/ISG). Ein neues Gimmick ist die Prüfsumme für Programme: Steht im Speicher z.B. LBL A so bekommt dies die Zeilennummer A01. Die folgenden Befehle werden dann mit A02, A03 . . . usw. (bis zum nächsten LBL) bekannt, und erhalten eine gemeinsame Prüfsumme, die mit der im Listing verglichen werden kann. So fallen Fehler beim Eintasten schneller auf. Alle Funktionen der Rechner 28S und 15C arbeiten auch mit komplexen Zahlen, und die Stackgröße bleibt erhalten. Der 32S jedoch erinnert mehr an das HP-41-MATH-ROM. In den 4 Stackregistern finden Real- und Imaginärteile von zwei Zahlen Platz, und außer den Grundrechenarten arbeiten nur noch SIN, COS, TAN, LN, EXP, 1/X und die allg. Potenzfunktion mit komplexen Zahlen, nicht jedoch z.B. ASIN, COSH, ATANH, ABS, x^2 , LOG . . . usw.

Das Handbuch (sehr gut: Spiralheftung) war für mich – obwohl in der englischen Version – einfach und leicht verständlich. Was nun dem Rechner noch fehlt, ist die Infrarotdiode für den Drucker.

Für den HP32S gibt es sogenannte Engineering Applikations; hier werden elektrotechnische Problemlösungen angeboten, ebenso Problemlösungen aus den Bereichen Maschinenbau, Architektur und Statistik. In Amerika werden sie für etwa 10,- US\$ angeboten.

Ralf Pfeifer
Rubenstraße 5
5000 Köln 40



HP-28S SPECIAL

Zunächst habe ich einen Steckbrief der Bytefresser zusammengestellt, da das Handbuch keine Informationen liefert. Früher nannte HP immer den geringeren Speicherplatzbedarf als Vorteil der UPN gegenüber der algebraischen Notation. Bei dem, was der HP-28 jedoch an Speicherplatz verschlingt, fällt das jedoch kaum ins Gewicht, weshalb die UPN jetzt nur noch die bessere Logik ist.

Die Tabelle gibt den Speicherplatzbedarf in Programmen an, und wenn dort steht, daß ein Programm im Stack 23 Bytes benötigt, so ist hier noch kein Befehl darin mitgerechnet. Da z.B. schon IF-THEN-Strukturen 17,5 Bytes benötigen, sollte man überlegen, ob nicht auch ein einfacher IFT-Befehl reicht. Lokale Namen kosten ebenfalls Platz: Die Definition $\rightarrow a \text{ sq} \ll$ braucht $7,5 + 4,5 + 5,5 = 17,5$ Bytes, und dann für jeden Rückruf noch einmal 4,5 bzw. 5,5 Bytes, so daß es sich manchmal lohnt scheinbar aufwendige Stackoperationen vorzuziehen. Außerdem hat jeder Name einen Inhalt (z.B. Zahlen, Matrizen, Programme), der auch noch irgendwo abgelegt werden muß, d.h. der hier berechnete Speicherbedarf ist nur aus organisatorischen Gründen erforderlich, der Inhalt wurde noch nicht mitgerechnet. Bei Listen ist der Inhalt wie im Programm zu zählen, und bei mehr als 5 Zahlen sind Vektoren günstiger als einzelne Zahlen im Programm, welche der HP-28 übrigens ohne Nullbyte trennt.

In Programmen können auch algebraische Strukturen auftauchen. Innerhalb der Begrenzungszeichen werden Befehle wie im übrigen (UPN-)Programm gezählt; So braucht 'SIN(X)' 4,5 Bytes für die Variable X, 2,5 für den Sinus und 5 für die '-Zeichen. Um den Speicherplatzbedarf eines Programms (oder andere Objekte) zu bestimmen geht man so vor: COMMAND, UNDO und LAST abschalten. MEMORY-Menue (auch das gewählte Menü beeinflusst den verfügbaren Speicherraum!) wählen, MEM, MEM, SWAP, DROP ausführen; Wenn das Programm in Ebene 1 und **zugleich** unter einem Namen gespeichert ist, wird es zwar angezeigt, aber braucht trotzdem nur 2,5 Bytes. Erst mit EDIT, ENTER, MEM, ROT, – erhält man dann den Bytebedarf des Programms **im Stack**.

Dieses Programmpaket entstand auf einem 28S, während das in PRISMA 4/88 S.37 noch auf einem 28C geschrieben wurde. Der 28S hat nicht nur neue Funktionen und einen geänderten und erweiterten UNITS-Katalog, sondern auch gewisse „Kompatibilitätsstörungen“: Wenn in Ebene 1 und 2 eine Liste steht, können sie im 28C mit „+“ verknüpft werden; der 28S ist großzügiger, bei ihm darf einer der beiden Stackinhalte auch eine Zahl oder ein algebraisches Objekt sein. Während beim 28C GET1 und PUT1 nach Erreichen des höchsten Index-Wertes automatisch zum kleinsten überge-

hen, hängt dieses intelligente Verhalten beim 28S vom Flag 46 ab, und dessen Normalzustand verhindert die dem 28C gleiche Arbeitsweise.

Hier noch ein Pflegetip: Wenn das Gelenk beim Klappen quietscht oder schwer geht, dann hilft PTFE, ein nichtleitendes, fettfreies Pulver (ähnlich wie Talkum), das in Elektronik-Bastlerläden erhältlich ist (z.B. als KONTAFLO 85, Kontakt-Chemie) und auch bei anderen Reibflächen zwischen Kunststoffen hilft (z.B. Frischluftdüsen im Auto).

PRIM ist eine Primfaktorenzerlegung, wobei **DV** das Unterprogramm zum Abdividieren und Anzeigen von Primfaktoren ist. Zunächst wird eine positive Ganzzahl in Ebene 1 eingegeben und PRIM gestartet. Der Rechner zeigt die Ausgangszahl, und dazu die Faktoren (als algebraisches Objekt) an. Nach Programmende hat dieses Objekt die Ausgangszahl in Ebene 1 überschrieben, kann jedoch mit EVAL zurückverwandelt werden. In den Programmen taucht oft der Befehl „NOT“ auf; er ersetzt die Befehle „=0“, denn immer wenn Ebene 1 eine 0 enthält, muß daraus eine 1 werden, aus allen anderen jedoch eine 0, und das macht gerade „NOT“, wobei das Programm etwa 15% schneller als mit „=0“ läuft.

OF berechnet das Osterdatum für jedes Jahr zwischen 1900 und 2100, nachdem es in Ebene 1 eingegeben wurde.

JC ist die Umkehrung zu **JDN**, das ich bereits in **PRISMA 4/88 S. 37** vorgestellt habe. Zunächst wird das julianische Datum in Ebene 1 eingegeben, das Programm hinterläßt dann in Ebene 3 den Tag, in 2 den Monat und das Jahr in Ebene 1. Die entsprechenden Formeln entstammen wieder dem **PPC-ROM Handbuch**. Ganz allgemein ist das julianische Datum eine durchlaufende Tageszählung in der Astronomie, die bereits 1581 von Justus Scalinger vorgeschlagen wurde, um die Rechenarbeit mit den üblichen Kalendern zu vermeiden. So liegt z.B. der Datumswechsel nicht um Mitternacht, sondern Mittags, d.h. wenn das Programm eine julianische Tageszahl in das Datum umrechnet, so gilt dies nur von Mittag des berechneten Tages, bis zum darauffolgenden Mittag. Heute ist der Datumswechsel um Mittag (= 12 Uhr mittlere Greenwich Zeit) bedeutungslos: Auf der ganzen Welt wird der Himmel rund um die Uhr beobachtet, mit Radioteleskopen auch bei Sonnenschein.

Die folgenden drei Programme sind eigentlich nicht als Lösung eines rekursiv gedacht, sie sollen vielmehr zeigen, daß mit dem **HP-28** auch rekursiv programmiert werden kann.

CMB macht fast genau das gleiche wie **CM** aus **PRISMA 4/88, S. 38**, jedoch in algebraischer Syntax rekursiv programmiert. Das Programm **CMB** ist kürzer als **CM**, braucht aber mehr Zeit und mehr Platz für seine Variablen aus den verschiedenen Rekursionsebenen.

NFC ist die Fakultät, und wie **FACT** zu bedienen.

ACK ist die Ackermann-Funktion und der heimliche Folterknecht der CPU, denn diese Funktion baut eine Unzahl von Rekursionsebenen auf. Werner Dworak hat sie in **PRISMA 85.7.06** zum Test eines **V30**-Prozessors eingesetzt. Anhand seiner ausführlichen Dokumentation kann man leicht erkennen, daß der **V30** etwa 10^2 bis 10^3 mal schneller ist. Zum Start von **Ackerman** gibt man in Ebene 1 und 2 ganze Zahlen ein. So führt 3 **ENTER** 2 nach ca. 29 s auf 29.

Einen Nachteil hat das Ganze aber auch: Als ich **PRIM** und **DV** programmiert habe, wollte ich in **PRIM** lokale Variable definieren. Aber nachdem **PRIM** zu **DV** verzweigt hatte, waren diese Variablen nicht mehr verfügbar, so daß die Datenübergabe jetzt im Stack erfolgt (globale Namen wären auch möglich).

KGV und **GGT** berechnen das kleinste gemeinsame Vielfache, und den größten gemeinsamen Teiler zweier ganzer Zahlen in Ebene 1 und 2. Die Reihenfolge der Zahlen ist egal.

Zum Schluß noch etwas Programmpflege: Die Programme **NPLOT** und **MPL** aus **PRISMA 4/88, S.37** lassen sich weiter verbessern, wenn man vor die **STEQ**-Befehle (je einmal vorhanden) noch ein **EVAL** eingefügt. Dadurch wird es möglich, im Stack nicht nur die Funktion abzuliegen, sondern sie in ein Programm zu verpacken, welches **PPAR** verändert. Das macht z.B. Sinn, wenn die Auflösung (Befehl **RES**) oder die Lage der Achsen (Befehl **AXES**, z.B. als Asymptoten) verändert werden soll. In diesem Programm steht am Ende wieder ein Programm oder algebraischer Ausdruck, der die eigentliche Funktion beinhaltet. Im Beispiel holt sich der Rechner aus Ebene 3 die Funktion, welche mit 1 **RES** geplottet wird. Dann findet er in Ebene 4 die erste & zweite Ableitung, deren Nullstellen an den Extrema bzw. Wendepunkten liegen.

Ralf Pfeifer
Rubensstraße 9
5000 Köln 90

Textzentrierung auf dem THINKJET

38/90 Zeilen, 94/175 Bytes, 14/25 Regs., SIZE ≥001/003, HP-41CV, IL, THINKJET

LBL „FMTJe“ für Texte ≤ 24
LBL „FMTJ“ für 24 ≥ Texte ≥ 24

Die Funktion **FMT** des **IL**-Moduls ist für den **ThinkJet**-Drucker ein Fremdwort.

Diese beiden Programme simulieren ein **FMT** am **ThinkJet**-Drucker in einer der vier Schriftarten.

Ist der Text in **FMTJe** (**FMT** am **ThinkJet**, einfach) zu lange, so geht das Programm an den Anfang zurück.

```
01*LBL "FMTJ"      15 GTO 05
02*LBL 00          16*LBL 03
03 "NECX"         17 "f25"
04 PNTK           18 142
05 "E&k"          19 GTO 05
06 ASTO 00        20*LBL 04
07 GTO IND X      21 "f35"
08*LBL 01         22 71
09 "f05"         23*LBL 05
10 00             24 ACA
11 GTO 05         25 STO 01
12*LBL 02         26 STO 02
13 "f15"         27 CLA
14 00             28 PNTA
```

```
29 24             58 RTN
30 ALENG          59 ALENG
31 ST- 01         60 ST- 01
32 X=Y?          61 E
33 XEQ 07         62 2
34*LBL 06         63 XROM 25,61
35 RCL 01         64 GTO 11
36 2              65 APPCHR
37 -              66 RCL Z
38 INT           67 24
39 .032          68 X=Y?
40 +              69 GTO 08
41 ACLX          70 RTN
42 FS?C 00       71*LBL 09
43 XEQ 09        72 CLX
44 OUTA          73 SEEKPT
45 GTO 11        74*LBL 10
46*LBL 07        75 GETREC
47 SF 00         76 FC? 17
48 RCL 02        77 RTN
49 6              78 OUTA
50 -              79 GTO 10
51 CRFLAS        80*LBL 11
52 APPREC        81 CLA
53*LBL 08        82 ARCL 00
54 CF 23         83 "f05"
55 CLA           84 ACA
56 PNTA          85 X<> c
57 FC? 23        86 STO 64
```

```
87 X<> c          21 71
88 FS?C 00       22*LBL 05
89 GTO 00        23 ACA
90 END           24 CLA
                25 PNTA
                26 ALENG
005:F3 18 26 6B 27 -
                28 2
                29 /
01*LBL "FMTJe"  30 INT
02 "NECX"       31 3.2 E-2
03 PNTK         32 +
04 "E&k"        33 ACLX
05 ASTO 00      34 OUTA
06 GTO IND X    35 CLA
07*LBL 01       36 ARCL 00
08 "f05"        37 "f05"
09 0 E1         38 END
10 GTO 05
11*LBL 02
12 "f15"        004:F3 18 26 6B
13 4 E1
14 GTO 05
15*LBL 03
16 "f25"
17 142
18 GTO 05
19*LBL 04
20 "f35"
```

Dr. Martin Hochenegger
Heidelberger Landstraße 97
6100 Darmstadt 13

BARCODES

Text: 53 54 41 4E 44 Nr. 0

NAVIS



-3.530494305 Nr. 16



Text: 4F 52 54 20 3A Nr. 1



81.00000000 Nr. 17



Text: 42 52 45 49 54 45 Nr. 2



2.593017623 Nr. 18



Text: 2F 4E 2B 53 2D 20 Nr. 3



Text: 53 4D 20 48 49 4E Nr. 19



Text: 4C 41 45 4E 47 45 Nr. 4



Text: 53 4D 20 57 45 47 Nr. 20



Text: 2F 57 2B 45 2D 20 Nr. 5



19.65939214 Nr. 21



Text: 48 2E 63 2E 20 3D Nr. 6



Text: 46 49 58 2D Nr. 22



Text: 41 5A 49 4D 55 54 Nr. 7



Text: 49 49 2E 20 Nr. 23



Text: 4B 55 52 53 3D 20 Nr. 8



Text: 50 45 49 4C Nr. 24



Text: 20 53 4D 3D 20 Nr. 9



Text: 55 4E 47 Nr. 25



Text: 5A 49 45 4C Nr. 10



Text: 48 2E 62 2E 3F 3D Nr. 26



Text: 2A 2D 44 45 43 4C Nr. 11



56.68333333 Nr. 27



Text: 2A 2D 4C 4F 4E 47 Nr. 12



20.63333333 Nr. 28



39.72342729 Nr. 13



119.1883807 Nr. 29



-3.530494305 Nr. 14



-8.571058000 Nr. 30



39.72342729 Nr. 15



-81.00000000 Nr. 31



-34.00000000 Nr. 32



0.991308249 Nr. 33



196.8202464 Nr. 34



3.000000000 Nr. 35



52.28365271 Nr. 36



0.722636595 Nr. 37



2.789748729 Nr. 38



0.000000000 Nr. 39



BIGDAT

123.00000000 Nr. 0



Text: 20 20 20 40 82 8 Nr. 1



Text: 10 41 2 4 4 4 Nr. 2



Text: 20 Nr. 3



Text: 22 4A A3 8A A4 88 Nr. 4



Text: 20 40 8F E2 4 8 Nr. 5



Text: 10 41 C3 80 0 0 Nr. 6



Text: F E0 0 0 Nr. 7



Text: 70 E0 0 0 0 0 Nr. 8



Text: 4 10 41 4 10 40 Nr. 9



Text: 75 14 59 34 51 5C Nr. 10



Text: F8 40 81 2 6 8 Nr. 11



Text: FC 8 27 90 20 BE Nr. 12



Text: FA C 7 4 10 7F Nr. 13



Text: 40 87 F2 45 C 10 Nr. 14



Text: FA C 8 F C0 FF Nr. 15



Text: F2 14 17 E0 41 7C Nr. 16



Text: 10 20 41 4 10 FF Nr. 17



Text: FA C 17 D0 60 BE Nr. 18



Text: 30 82 F D0 60 BE Nr. 19



Text: 30 60 0 0 6 C Nr. 20



Text: 10 40 C0 0 6 C Nr. 21



Text: 80 C0 60 21 8C 60 Nr. 22



BARCODES

Text: 3 F8 0 1F C0 0 Nr. 23



Text: C 63 8 C 6 3 Nr. 24



Text: 20 0 86 10 20 BE Nr. 25



Text: 78 F DA B7 60 BE Nr. 26



Text: 6 F F8 30 51 1C Nr. 27



Text: FE C 17 F0 60 BF Nr. 28



Text: FA 8 10 20 60 BE Nr. 29



Text: 7D 14 28 50 91 1F Nr. 30



Text: FC 8 13 E0 40 FF Nr. 31



Text: 4 8 13 E0 40 FF Nr. 32



Text: 7B F 10 20 60 BE Nr. 33



Text: 6 C 1F F0 60 C1 Nr. 34



Text: F8 40 81 2 4 3E Nr. 35



Text: 38 89 12 4 8 7C Nr. 36



Text: 5 9 11 E4 50 C1 Nr. 37



Text: FC 8 10 20 40 81 Nr. 38



Text: 6 C 19 35 71 C1 Nr. 39



Text: 7 D 19 31 61 C1 Nr. 40



Text: 71 14 18 30 51 1C Nr. 41



Text: 4 8 17 F0 60 BF Nr. 42



Text: 71 15 18 30 51 1C Nr. 43



Text: 5 9 17 F0 60 BF Nr. 44



Text: FE 4 7 C0 40 FE Nr. 45



Text: 20 40 81 2 4 7F Nr. 46



Text: FA C 18 30 60 C1 Nr. 47



Text: 20 A1 44 48 A0 C1 Nr. 48



Text: 8A AC 99 30 60 C1 Nr. 49



Text: 5 11 41 5 11 41 Nr. 50



Text: 20 40 81 5 11 41 Nr. 51



Text: FC 10 41 4 10 7F Nr. 52



Text: F8 10 20 40 81 3E Nr. 53



Text: 1 1 1 1 1 1 Nr. 54



Text: F9 2 4 8 10 3E Nr. 55



Text: 20 40 89 2A 8E 8 Nr. 56



Text: FC 0 0 0 0 0 Nr. 57



Text: 2 4 3E Nr. 58



Text: FA F E8 F 80 0 Nr. 59



Text: FE C 18 6F 40 81 Nr. 60



Text: FA 8 10 2F 80 0 Nr. 61



Text: FA C 1C 37 A0 40 Nr. 62



Text: F8 B F8 2F 80 0 Nr. 63



FMTJE (Hochenegger) 1615

Computerclub Deutschland e.V.

Zeile 1 (1- 2) CCD-Barcodes



Zeile 2 (2- 8) CCD-Barcodes



Zeile 3 (8- 12) CCD-Barcodes



Zeile 4 (12- 17) CCD-Barcodes



Zeile 5 (17- 24) CCD-Barcodes



Zeile 6 (24- 31) CCD-Barcodes



Zeile 7 (31- 37) CCD-Barcodes



Zeile 8 (37- 38) CCD-Barcodes



FMTJ (Hochenegger) 1616

Computerclub Deutschland e.V.

Zeile 1 (1- 3) CCD-Barcodes



Zeile 2 (3- 9) CCD-Barcodes



Zeile 3 (9- 14) CCD-Barcodes



Zeile 4 (14- 19) CCD-Barcodes



Zeile 5 (19- 27) CCD-Barcodes



Zeile 6 (27- 34) CCD-Barcodes



Zeile 7 (34- 42) CCD-Barcodes



Zeile 8 (42- 50) CCD-Barcodes



Zeile 9 (50- 58) CCD-Barcodes



Zeile 10 (58- 65) CCD-Barcodes



Zeile 11 (65- 74) CCD-Barcodes



Zeile 12 (74- 82) CCD-Barcodes



Zeile 13 (82- 88) CCD-Barcodes



Zeile 14 (88- 90) CCD-Barcodes



TJPITCH (Hochenegger) 1617

Computerclub Deutschland e.V.

Zeile 1 (1- 2) CCD-Barcodes



Zeile 2 (2- 8) CCD-Barcodes



Zeile 3 (8- 12) CCD-Barcodes



Zeile 4 (12- 14) CCD-Barcodes



Zeile 5 (14- 17) CCD-Barcodes



Zeile 6 (17- 19) CCD-Barcodes



Zeile 7 (19- 22) CCD-Barcodes



Zeile 8 (22- 24) CCD-Barcodes



Zeile 9 (24- 29) CCD-Barcodes



Zeile 10 (29- 30) CCD-Barcodes



Zeile 11 (30- 32) CCD-Barcodes



Zeile 12 (32- 35) CCD-Barcodes



BARCODES

Zeile 13 (35- 42) CCD-Barcodes



Zeile 14 (42- 49) CCD-Barcodes



Zeile 15 (49- 53) CCD-Barcodes



Zeile 16 (53- 59) CCD-Barcodes



Zeile 17 (59- 65) CCD-Barcodes



Zeile 18 (65- 68) CCD-Barcodes



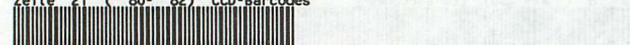
Zeile 19 (68- 75) CCD-Barcodes



Zeile 20 (75- 80) CCD-Barcodes



Zeile 21 (80- 82) CCD-Barcodes



Zeile 19 (87- 95) CCD-Barcodes



Zeile 20 (95- 104) CCD-Barcodes



Zeile 21 (104- 112) CCD-Barcodes



Zeile 22 (112- 120) CCD-Barcodes



Zeile 23 (120- 127) CCD-Barcodes



Zeile 24 (127- 137) CCD-Barcodes



Zeile 25 (137- 145) CCD-Barcodes



Zeile 26 (145- 150) CCD-Barcodes



Zeile 27 (150- 157) CCD-Barcodes



Zeile 28 (157- 169) CCD-Barcodes



Zeile 29 (169- 179) CCD-Barcodes



Zeile 30 (179- 191) CCD-Barcodes



Zeile 31 (191- 203) CCD-Barcodes



Zeile 32 (203- 213) CCD-Barcodes



Zeile 33 (213- 223) CCD-Barcodes



Zeile 34 (223- 233) CCD-Barcodes



Zeile 35 (233- 244) CCD-Barcodes



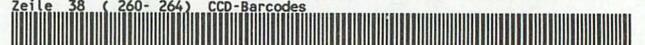
Zeile 36 (244- 255) CCD-Barcodes



Zeile 37 (255- 260) CCD-Barcodes



Zeile 38 (260- 264) CCD-Barcodes



Zeile 39 (264- 269) CCD-Barcodes



Zeile 40 (269- 274) CCD-Barcodes



Zeile 41 (274- 280) CCD-Barcodes



Zeile 42 (280- 285) CCD-Barcodes



Zeile 43 (285- 289) CCD-Barcodes



Zeile 44 (289- 294) CCD-Barcodes



Zeile 45 (294- 300) CCD-Barcodes



Zeile 46 (300- 305) CCD-Barcodes



Zeile 47 (305- 311) CCD-Barcodes



RAL (Müllhäuser) 1618

Computerclub Deutschland e.V.

Zeile 1 (1- 3) CCD-Barcodes



Zeile 2 (3- 8) CCD-Barcodes



Zeile 3 (8- 11) CCD-Barcodes



Zeile 4 (11- 18) CCD-Barcodes



Zeile 5 (18- 23) CCD-Barcodes



Zeile 6 (23- 28) CCD-Barcodes



Zeile 7 (28- 32) CCD-Barcodes



Zeile 8 (32- 37) CCD-Barcodes



Zeile 9 (37- 42) CCD-Barcodes



Zeile 10 (42- 47) CCD-Barcodes



Zeile 11 (47- 52) CCD-Barcodes



Zeile 12 (52- 56) CCD-Barcodes



Zeile 13 (56- 61) CCD-Barcodes



Zeile 14 (61- 66) CCD-Barcodes



Zeile 15 (66- 71) CCD-Barcodes



Zeile 16 (71- 75) CCD-Barcodes



Zeile 17 (75- 79) CCD-Barcodes



Zeile 18 (79- 87) CCD-Barcodes



Zeile 48 (311- 316) CCD-Barcodes



Zeile 49 (316- 320) CCD-Barcodes



Zeile 50 (320- 325) CCD-Barcodes



Zeile 51 (325- 333) CCD-Barcodes



Zeile 52 (333- 336) CCD-Barcodes



QR (Pfeifer) 1619

Computerclub Deutschland e.V.

Zeile 1 (1- 4) CCD-Barcodes



Zeile 2 (4- 10) CCD-Barcodes



Zeile 3 (10- 13) CCD-Barcodes



Zeile 4 (13- 20) CCD-Barcodes



Zeile 5 (20- 27) CCD-Barcodes



Zeile 6 (27- 38) CCD-Barcodes



Zeile 7 (38- 49) CCD-Barcodes



Zeile 8 (49- 57) CCD-Barcodes



Zeile 9 (57- 63) CCD-Barcodes



Zeile 10 (63- 70) CCD-Barcodes



Zeile 11 (70- 79) CCD-Barcodes



Zeile 12 (79- 88) CCD-Barcodes



Zeile 13 (88- 99) CCD-Barcodes



Zeile 14 (99- 108) CCD-Barcodes



Zeile 15 (108- 117) CCD-Barcodes



Zeile 16 (117- 128) CCD-Barcodes



Zeile 17 (128- 138) CCD-Barcodes



Zeile 18 (138- 146) CCD-Barcodes



Zeile 19 (146- 156) CCD-Barcodes



Zeile 20 (156- 163) CCD-Barcodes



Zeile 21 (163- 169) CCD-Barcodes



VID (Hempler) 1620

Computerclub Deutschland e.V.

Zeile 1 (1- 5) CCD-Barcodes



Zeile 2 (5- 10) CCD-Barcodes



Zeile 3 (10- 15) CCD-Barcodes



Zeile 4 (15- 17) CCD-Barcodes



Zeile 5 (17- 19) CCD-Barcodes



Zeile 6 (19- 22) CCD-Barcodes



Zeile 7 (22- 28) CCD-Barcodes



Zeile 8 (28- 29) CCD-Barcodes



Zeile 9 (29- 33) CCD-Barcodes



Zeile 10 (33- 40) CCD-Barcodes



CLS (Hempler) 1622

Computerclub Deutschland e.V.

Zeile 1 (1- 5) CCD-Barcodes



Zeile 2 (5- 7) CCD-Barcodes



Zeile 3 (7- 13) CCD-Barcodes



Zeile 4 (13- 20) CCD-Barcodes



Anmerkung

Sollten einige Clubmitglieder Probleme haben mit ihren Barcodelesern die im PRISMA abgedruckten Barcodes zu lesen, so hilft schon einfach ein kopieren derselben, dabei vergrößert sich nämlich der Abstand zwischen den einzelnen Balken.

Martin Meyer (1000)
Redaktion

SCR (Hempler) 1621

Computerclub Deutschland e.V.

Zeile 1 (1- 3) CCD-Barcodes



Zeile 2 (3- 8) CCD-Barcodes



CODE (Saalfeld) 1623

Computerclub Deutschland e.V.

Zeile 1 (1- 2) CCD-Barcodes



Zeile 2 (2- 9) CCD-Barcodes



Zeile 3 (9- 18) CCD-Barcodes



Zeile 4 (18- 20) CCD-Barcodes



Zeile 5 (20- 27) CCD-Barcodes



Zeile 6 (27- 34) CCD-Barcodes



Zeile 7 (34- 43) CCD-Barcodes



Zeile 8 (43- 52) CCD-Barcodes



Zeile 9 (52- 59) CCD-Barcodes



Zeile 10 (59- 63) CCD-Barcodes



Zeile 11 (63- 69) CCD-Barcodes



Zeile 12 (69- 72) CCD-Barcodes



Zeile 13 (72- 73) CCD-Barcodes

**MUSEL (Saalfeld) 1625**

Computerclub Deutschland e.V.

Zeile 1 (1- 2) CCD-Barcodes



Zeile 2 (2- 7) CCD-Barcodes



Zeile 3 (7- 13) CCD-Barcodes



Zeile 4 (13- 21) CCD-Barcodes



Zeile 5 (21- 30) CCD-Barcodes



Zeile 6 (30- 37) CCD-Barcodes



Zeile 7 (37- 46) CCD-Barcodes



Zeile 8 (46- 53) CCD-Barcodes



Zeile 9 (53- 61) CCD-Barcodes



Zeile 10 (61- 68) CCD-Barcodes



Zeile 11 (68- 77) CCD-Barcodes



Zeile 12 (77- 86) CCD-Barcodes



Zeile 13 (86- 93) CCD-Barcodes



Zeile 14 (93- 97) CCD-Barcodes

**CHANGE (Saalfeld) 1624**

Computerclub Deutschland e.V.

Zeile 1 (1- 2) CCD-Barcodes



Zeile 2 (2- 8) CCD-Barcodes



Zeile 3 (8- 12) CCD-Barcodes



Zeile 4 (12- 17) CCD-Barcodes



Zeile 5 (17- 26) CCD-Barcodes



Zeile 6 (26- 34) CCD-Barcodes



Zeile 7 (34- 42) CCD-Barcodes



Zeile 8 (42- 47) CCD-Barcodes

**LETTER (Saalfeld) 1626**

Computerclub Deutschland e.V.

Zeile 1 (1- 2) CCD-Barcodes



Zeile 2 (2- 7) CCD-Barcodes



Zeile 3 (7- 12) CCD-Barcodes



Zeile 4 (12- 21) CCD-Barcodes



Zeile 5 (21- 27) CCD-Barcodes



Zeile 6 (27- 33) CCD-Barcodes



Zeile 7 (33- 36) CCD-Barcodes



Zeile 8 (36- 42) CCD-Barcodes



Zeile 9 (42- 46) CCD-Barcodes



Zeile 10 (46- 55) CCD-Barcodes



Zeile 11 (55- 61) CCD-Barcodes



Zeile 12 (61- 68) CCD-Barcodes



Zeile 13 (68- 76) CCD-Barcodes



Zeile 14 (76- 83) CCD-Barcodes



Zeile 15 (83- 90) CCD-Barcodes



Zeile 16 (90- 98) CCD-Barcodes



Zeile 17 (98- 107) CCD-Barcodes



Zeile 18 (107- 116) CCD-Barcodes



Zeile 19 (116- 122) CCD-Barcodes



Zeile 20 (122- 129) CCD-Barcodes



Zeile 21 (129- 133) CCD-Barcodes



Zeile 22 (133- 139) CCD-Barcodes



Zeile 23 (139- 147) CCD-Barcodes



Zeile 24 (147- 148) CCD-Barcodes



Zeile 22 (165- 173) CCD-Barcodes



Zeile 23 (173- 182) CCD-Barcodes



Zeile 24 (182- 187) CCD-Barcodes



Zeile 25 (187- 194) CCD-Barcodes



Zeile 26 (194- 201) CCD-Barcodes



Zeile 27 (201- 209) CCD-Barcodes



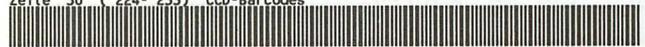
Zeile 28 (209- 215) CCD-Barcodes



Zeile 29 (215- 224) CCD-Barcodes



Zeile 30 (224- 233) CCD-Barcodes



Zeile 31 (233- 240) CCD-Barcodes



Zeile 32 (240- 249) CCD-Barcodes



Zeile 33 (249- 256) CCD-Barcodes



Zeile 34 (256- 263) CCD-Barcodes



Zeile 35 (263- 273) CCD-Barcodes



Zeile 36 (273- 281) CCD-Barcodes



Zeile 37 (281- 290) CCD-Barcodes



Zeile 38 (290- 299) CCD-Barcodes



Zeile 39 (299- 307) CCD-Barcodes



Zeile 40 (307- 315) CCD-Barcodes



Zeile 41 (315- 322) CCD-Barcodes



Zeile 42 (322- 322) CCD-Barcodes



Zeile 43 (322- 331) CCD-Barcodes



Zeile 44 (331- 343) CCD-Barcodes



Zeile 45 (343- 351) CCD-Barcodes



Zeile 46 (351- 360) CCD-Barcodes



Zeile 47 (360- 368) CCD-Barcodes



Zeile 48 (368- 377) CCD-Barcodes



Zeile 49 (377- 385) CCD-Barcodes



Zeile 50 (385- 393) CCD-Barcodes



NAVIS (Neuburger) 1627

Computerclub Deutschland e.V.

Zeile 1 (1- 3) CCD-Barcodes



Zeile 2 (3- 9) CCD-Barcodes



Zeile 3 (9- 16) CCD-Barcodes



Zeile 4 (16- 22) CCD-Barcodes



Zeile 5 (22- 30) CCD-Barcodes



Zeile 6 (30- 37) CCD-Barcodes



Zeile 7 (37- 44) CCD-Barcodes



Zeile 8 (44- 53) CCD-Barcodes



Zeile 9 (53- 60) CCD-Barcodes



Zeile 10 (60- 68) CCD-Barcodes



Zeile 11 (68- 74) CCD-Barcodes



Zeile 12 (74- 82) CCD-Barcodes



Zeile 13 (82- 89) CCD-Barcodes



Zeile 14 (89- 97) CCD-Barcodes



Zeile 15 (97- 107) CCD-Barcodes



Zeile 16 (107- 116) CCD-Barcodes



Zeile 17 (116- 124) CCD-Barcodes



Zeile 18 (124- 131) CCD-Barcodes



Zeile 19 (131- 141) CCD-Barcodes



Zeile 20 (141- 152) CCD-Barcodes



Zeile 21 (152- 165) CCD-Barcodes



CLUBADRESSEN

Zeile 51 (393- 403) CCD-Barcodes



Zeile 52 (403- 412) CCD-Barcodes



Zeile 53 (412- 420) CCD-Barcodes



Zeile 54 (420- 429) CCD-Barcodes



Zeile 55 (429- 438) CCD-Barcodes



Zeile 56 (438- 446) CCD-Barcodes



Zeile 57 (446- 452) CCD-Barcodes



Zeile 58 (452- 456) CCD-Barcodes



Zeile 59 (456- 463) CCD-Barcodes



Zeile 60 (463- 468) CCD-Barcodes



Zeile 61 (468- 476) CCD-Barcodes



Zeile 62 (476- 484) CCD-Barcodes



Zeile 63 (484- 489) CCD-Barcodes



Zeile 64 (489- 495) CCD-Barcodes



Zeile 65 (495- 502) CCD-Barcodes



Zeile 66 (502- 510) CCD-Barcodes



Zeile 67 (510- 520) CCD-Barcodes



Zeile 68 (520- 527) CCD-Barcodes



Zeile 69 (527- 536) CCD-Barcodes



Zeile 70 (536- 544) CCD-Barcodes



Zeile 71 (544- 547) CCD-Barcodes



Zeile 72 (547- 551) CCD-Barcodes



Zeile 73 (551- 560) CCD-Barcodes



Zeile 74 (560- 568) CCD-Barcodes



Clubadressen:

1. Vorsitzender

Prof. Dr. Wolfgang Fritz (125)
Kronenstraße 34
7500 Karlsruhe
GEO1: W.FRITZ

2. Vorsitzender

Erich H. Klee (1170)
Ruhrallee 8
4300 Essen 1
GEO1: E.H.KLEE

Schatzmeister Mitgliederverwaltung

Dieter Wolf (1734)
Pützerstraße 29
6000 Frankfurt 90
☎ 069 / 765912
GEO1: D.WOLF

1. Beisitzer

Werner Dworak (607)
Allewind 51
7900 Ulm
☎ 07304 / 3274
GEO1: W.DWORAK

2. Beisitzer Geowissenschaften

Alf-Norman Tietze (1909)
Thudichumstraße 14
6000 Frankfurt 90
☎ 069 / 7893995
GEO1: A.N.TIETZE

PRISMA-Nachsendedienst

CCD e.V.
Postfach 11 04 11
6000 Frankfurt 1
☎ 069 / 765912

Programm-Bibliothek HP-41

Beirat

Martin Meyer (1000)
Robert-Stolz-Straße 5
6232 Bad Soden 1

Programm-Bibliothek HP-71

Henry Schimmer (786)
Homburger Landstraße 63
6000 Frankfurt 50

Serie 80 Service

Klaus Kaiser (1661)
Mainzer Landstraße 561
6230 Frankfurt am Main 80
☎ 069 / 397852

Beirat

MS-DOS Service

Alexander Wolf (3303)
Pützerstraße 29
6000 Frankfurt 90
☎ 069 / 765912

Hardware 41

Winfried Maschke (413)
Ursulakloster 4
5000 Köln 1
☎ 0221 / 131297

Grabau GR7 Interface

Holger von Stillfried (2641)
Alsterkrugchaussee 212
2000 Hamburg 60
☎ 040 / 5116346

CP/M-80 Service

Peter-C. Spaeth
Michaeliburgstraße 4
8000 München 80

E-Technik

Werner Meschede (2670)
Sorpestraße 4
5788 Siedlingshausen

Mathematik

Andreas Wolpers (349)
Steinstraße 15
7500 Karlsruhe

Vermessungswesen

Ulrich Kulle (2719)
Schnuckentritt 14
3000 Hannover 51
☎ 0511 / 6042728

Regionalgruppe Berlin

Jörg Warmuth (79)
Wartburgstraße 17
1000 Berlin 62

Regionalgruppe Hamburg

Alfred Czaya (2225)
An der Bahn 1
2061 Sülfeld
☎ 040 / 433668 (Mo.-Do. abends)

Horst Ziegler (1361)

Schüslerweg 18 b
2100 Hamburg 90
☎ 040 / 7905672

Beirat

Regionalgruppe Karlsruhe

Stefan Schwall (1695)
Rappenwörtstraße 42
7500 Karlsruhe 21
☎ 0721 / 576756
GEO1:S.SCHWALL

Regionalgruppe Köln

Frank Ortman (1089)
Okerstraße 24
5090 Leverkusen 1

Regionalgruppe München

Victor Lecoq (2246)
Seumestraße 8
8000 München 70
☎ 089 / 789379

Regionalgruppe Rhein-Main

Andreas Eschmann (2289)
Lahnstraße 2
6096 Raunheim
☎ 61442 / 46642

Beirat

Peter Kemmerling (2466)
Danzigerstraße 17
4030 Ratingen

Beirat

Ulrich Schwaderlap (438)
An den Berken 34
5840 Schwerte 6

Beirat

Günther Schwarz (2658)
Bodelschwinghstraße 34
3408 Duderstadt 1

Atari Service

Werner Müller
Classen-Kappelmannstr. 30 a
5000 Köln 41

Zeile 75 (568- 577) CCD-Barcodes

 Zeile 76 (577- 584) CCD-Barcodes

 Zeile 77 (584- 592) CCD-Barcodes

 Zeile 78 (592- 598) CCD-Barcodes

 Zeile 79 (598- 606) CCD-Barcodes

 Zeile 80 (606- 612) CCD-Barcodes

 Zeile 81 (612- 620) CCD-Barcodes

 Zeile 82 (620- 629) CCD-Barcodes

 Zeile 83 (629- 633) CCD-Barcodes

 Zeile 84 (633- 641) CCD-Barcodes

 Zeile 85 (641- 647) CCD-Barcodes

 Zeile 86 (647- 656) CCD-Barcodes

 Zeile 87 (656- 663) CCD-Barcodes

 Zeile 88 (663- 670) CCD-Barcodes

 Zeile 89 (670- 673) CCD-Barcodes

 Zeile 90 (673- 680) CCD-Barcodes

 Zeile 91 (680- 688) CCD-Barcodes

 Zeile 92 (688- 694) CCD-Barcodes

 Zeile 93 (694- 703) CCD-Barcodes

 Zeile 94 (703- 709) CCD-Barcodes

 Zeile 95 (709- 716) CCD-Barcodes

 Zeile 96 (716- 725) CCD-Barcodes

 Zeile 97 (725- 732) CCD-Barcodes

 Zeile 98 (732- 740) CCD-Barcodes

 Zeile 99 (740- 748) CCD-Barcodes

 Zeile 100 (748- 758) CCD-Barcodes

 Zeile 101 (758- 768) CCD-Barcodes

 Zeile 102 (768- 775) CCD-Barcodes

 Zeile 103 (775- 780) CCD-Barcodes


Zeile 104 (780- 783) CCD-Barcodes

 Zeile 105 (783- 788) CCD-Barcodes

 Zeile 106 (788- 796) CCD-Barcodes

 Zeile 107 (796- 802) CCD-Barcodes

 Zeile 108 (802- 808) CCD-Barcodes

 Zeile 109 (808- 817) CCD-Barcodes

 Zeile 110 (817- 822) CCD-Barcodes

 Zeile 111 (822- 828) CCD-Barcodes

 Zeile 112 (828- 834) CCD-Barcodes

 Zeile 113 (834- 840) CCD-Barcodes

 Zeile 114 (840- 844) CCD-Barcodes

 Zeile 115 (844- 848) CCD-Barcodes

 Zeile 116 (848- 851) CCD-Barcodes

 Zeile 117 (851- 856) CCD-Barcodes

 Zeile 118 (856- 863) CCD-Barcodes

 Zeile 119 (863- 869) CCD-Barcodes

 Zeile 120 (869- 874) CCD-Barcodes

 Zeile 121 (874- 880) CCD-Barcodes

 Zeile 122 (880- 888) CCD-Barcodes

 Zeile 123 (888- 890) CCD-Barcodes

 Zeile 124 (890- 894) CCD-Barcodes

 Zeile 125 (894- 900) CCD-Barcodes

 Zeile 126 (900- 907) CCD-Barcodes

 Zeile 127 (907- 915) CCD-Barcodes

 Zeile 128 (915- 924) CCD-Barcodes

 Zeile 129 (924- 930) CCD-Barcodes

 Zeile 130 (930- 939) CCD-Barcodes

 Zeile 131 (939- 950) CCD-Barcodes

 Zeile 132 (950- 958) CCD-Barcodes


Zeile 133 (958- 966) CCD-Barcodes



Zeile 134 (966- 970) CCD-Barcodes



Zeile 135 (970- 979) CCD-Barcodes



Zeile 136 (979- 985) CCD-Barcodes



Zeile 137 (985- 995) CCD-Barcodes



Zeile 138 (995-1003) CCD-Barcodes



Zeile 139 (1003-1012) CCD-Barcodes



Zeile 140 (1012-1018) CCD-Barcodes



Zeile 141 (1018-1027) CCD-Barcodes



Zeile 142 (1027-1037) CCD-Barcodes



Zeile 143 (1037-1044) CCD-Barcodes



Zeile 144 (1044-1053) CCD-Barcodes



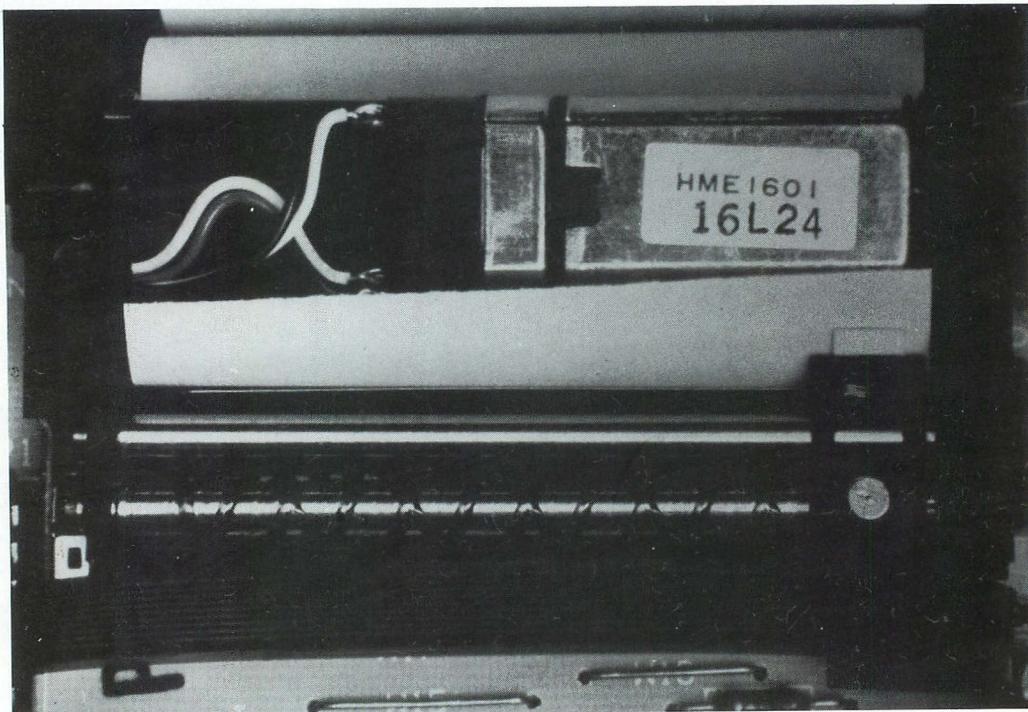
Zeile 145 (1053-1059) CCD-Barcodes



Zeile 146 (1059-1068) CCD-Barcodes



Zeile 147 (1068-1070) CCD-Barcodes



Blick auf das Druckwerk des HP-82240A Infrarotdruckers. Ganz oben sieht man noch die Papierrolle, darunter den Elektromotor (mit dem Aufkleber 16L24). Dieser Motor dreht sich nur in einer Richtung, dennoch bewegt sich der Druckkopf (darunter, rechts) auf dem Papier hin und her. Diesen mechanischen Trick bewerkstelligt die zerfurchte Welle (unten) auf der der Druckkopf sitzt. Die Funktion dieser Welle kann man sich so vorstellen: Man zieht mit einer Nadel auf einer Wachskerze gerade von links nach rechts und wieder zurück. Auf der Kerze bleibt ein gerader Kratzer zurück. Wenn sich nun die Kerze unter der Nadel in eine Richtung dreht und die Nadel die

gleiche Bewegung nocheinmal ausführt, so ist auf der Kerze genau die gleiche verschlungene Linie entstanden, wie auf der Antriebswelle des Druckkopfes.

Der helle Kreis auf dem Druckkopf ist eine Art drehbare Schraubenzieherklinge, die in die "Furchen" der Welle hineindrückt. Sobald sich die Welle dreht, wird der Druckkopf über die Schraubenzieherklinge hin- und herbewegt.

Ralf Pfeifer (116)