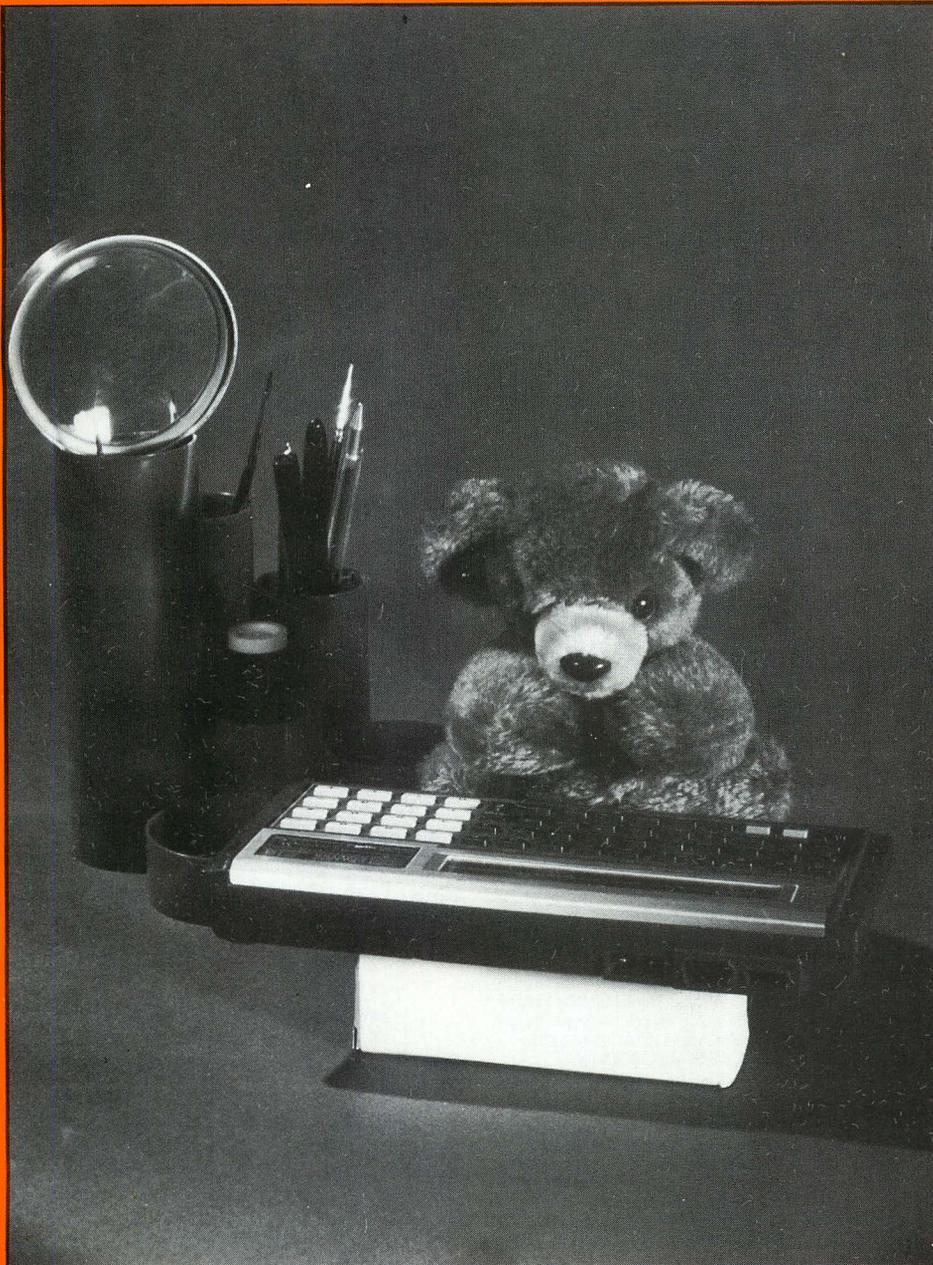


PRISMA

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

Mai/Juni 1988 Nr. 3

D 2856 F



Das ist "TEDDY", der knuddelige Text-Editor für den HP-71B! Klein aber fein und trotzdem komfortabel. Er ist sehr modern, denn er benutzt neue LEX-Files, die zum Teil noch nicht veröffentlicht wurden. Einfach toll - man muß ihn einfach "lieb" haben. Das Programm ist in der HP-71 Clubbibliothek erhältlich.

Clubnachrichten

Clubbörse
Clubreise nach USA
CCD Lexfile-ID
Mitgliederversammlung 1988

Grundlagen

Der zweite Turmbau zu Babel
(Programmiersprachen)
Vergleich v. Navigationsmodulen

MS-DOS

Referenzblätter

ATARI

Atari-Gruppe im CCD

Serie 70

"TEDDY" Text-Editor
Diskettenkatalog
LEXCHEC2
Lexfile PRINTLX
Eigenwert-Berechnung

Serie 40

E-Technik Buchbesprechung
Dreieck-Berechnungen
BARCODES des HP-41
Dreisatz
XROM-Funktionen
BANNER - Querdruckprogramm
Kurvenapproximation
Thermodrucker mit Infrarot

Serie 10

Regression mit Polynomen

Verkaufe: 2 32 KB-RAM-Module (CMT) für HP71
Stck. 250 DM, Andreas Haun, ☎ 06151/27706 (19⁰⁰-21⁰⁰)

HP-41-Zubehör zu verkaufen, gebraucht, technisch und optisch OK:

- Einsteckdrucker HP82143A, der ideale Drucker zum HP-41 400,- DM
 - X-Functions-Modul HP82180A 70,- DM
 - Doppel-X-Memory (=2 X-Memory-Module HP82181A, fachgerecht in einem Modulgehäuse zusammengebaut) 150,- DM
- Winfried Maschke, ☎ 0221/131297

Verkaufe: HP-41 CX (neu, orig. verpackt), 350,- DM
RAMBOX II (64K für HP-41, neu 10 Mon. Garantie) 800,- DM, Video-Interface (HP82163) 150,- DM, Mathe-Modul, Statistik-Modul je 40,- DM, HP-Netzteil für Akku 15,- DM, Staubschutzoverlay für HP-41 10,- DM, PRISMA 7/86 3,50 DM; K.-U. Peikert (3410), Diétr.-Bonhoeffer-Str. 5, 2800 Bremen 41, ☎ 0421/476309

Verkaufe: IL-ThinkJet HP2225B mit ca. 1000 Blatt Endlospapier 12", IL-Diskettenlaufwerk HP-9114B mit 2 Disketten, HP-75C mit I/O ROM und Graphics-Solutions auf Magnet-Karte, Plottermodul und Developmentmodul für HP 41 HP 16 C. Alle Geräte mit Original-Bedienungshandbüchern. Preise nach Gebot und Einigung. Versand frei Haus - ohne Zusatzkosten. Konrad Albers, Südring 42, 2300 Klausdorf/Schwentine, ☎ tagsüber 0431/92055 nach 18.00 Uhr 0431/79494

Hardware: HP-41 CV, Interface-Loop, IL-Laufwerk, IL-Thermodrucker, 3fach-Netzgerät, Port-Extender, Barcode-Lesestift, Time-Modul, XF-Modul, CCD-Modul, Mathe-Modul.

Zubehör: 4 Datenkassetten, Thermopapier, ca. 160 Magnetkarten, Tastaturschablonen, Barcodelabels und mehr.
Literatur: Dearing: *Calculator Tips & Routines*, Gehret: *Softwareentwicklung* . . . Gosmann: *Anwenderhandbuch*, Wickes: *Synth. Programm.* (engl. u. deutsch), PRISMA ab Jahrg. 82, *Best of PRISMA*, Programme, Handbücher.

Komplett: nur DM 1750,- (Einzelteile VHS).
Gerd M. Krause, ☎ (0521) 40014 (ich rufe zurück).

Verkaufe wegen Systemerweiterung: Grabau Video-Interface GR7 AG-IL mit 12" Monitor für DM 1000,-. Plottermodul HP 82184A für DM 100,-, HP-IL Development Module HP 00041-15043 für DM 60,-.
W. Meschede (CCD2670) abends 0911-808756.

Suche Manual für CMT-71-P01 EPROM Programmier. Kopier- und/oder Protokolle werden übernommen.
Klaus Kaiser, Mainzer Landstr. 561, 6230 Frankfurt am Main 80, ☎ 069/397852

Verkaufe HP-ThinkJet (IL) für 600,- DM, HP Plotter ColorPro (RS232) neuwertig 2300,- DM, HP Akustik Koppler / Modem 92205M (IL) 500,- DM, HP-28C mit 4K-RAM Erweiterung 260,- DM
Gregory Wesley, ☎ 0911/591462

Wegen Systemwechsel zu verkaufen: Orig. HP Software für HP 150: Personal Card File 80,- DM, Memo Maker 80,- DM, 30 3.5" Disketten Double Sided mit Software: Lotus 1-2-3, DBase II, Wordstar, Exec. Memomaker, Execudesk, Word, PCF, Condor 20-3, HPIL-Link, Graphics, Basic, Fortran u.a., HP-110 Portable für 1500,- DM zu verkaufen. Toni Lerchenfeld, (CCD 1209), ☎ 08323/7323 oder 0043/5222 86669.

Verkaufe für HP-71: Mathemodul (HP82480) und Forth-Assembler-Modul (HP82441). Jeweils 100,- DM. Jochen Haas, 5090 Leverkusen 3, ☎ 02171/46802.

Verkaufe: HP 71 B mit IL-Modul u. Ständer Tintenstrahldrucker 2225 B mit Acryl-Ständer, 2. Batterie u. ca. 1000 Bl. Endlospapier, 4 Speichererweiterungsmodule je 32 K, 1 Erw. Modul im Kartenleserschacht 32 K, Mathematikmodul, div. Bücher. Komplet 2200,- DM
W. Schneider, 5000 Köln 41, ☎ 0221/431898

HP 75C, Kartenleser DM 500,-, **EPSON FX 80** mit eingeb. IL-Schnittstelle DM 500,-, **HP 41CX** DM 500,-, Thermodrucker 82143A DM 500,-, Kartenleser 82104A DM 250,-, IL-Modul 82160A DM 100,-, Video-Interface 82163A DM 100,-, Hans Krissler, Haldestraße 7, 7901 Lonsee, ☎ 07336/5225

Verkaufe (alles für HP-41, alles Bestzustand): IL-Drucker 82162A inkl. IL-Modul DM 550,-, PORT-EXTENDER DM 75,-, Staubschutz-Overlay DM 12,-, Batteriekasten mit 4 VATRA-Akkus DM 25,-
☎ 06181/20481 (bis 19 Uhr)

HP 71 B mit IL-Modul zu verkaufen: 900,- DM, Forth/Assembler Modul 200,- DM, **Diskettenlaufwerk 9114A** mit Disketten und Netzteil 1200,- DM, Drucker **ThinkJet 2225B** mit Ständer und Papier 800,- DM, Preise V.B. Reinhard Schmidt, Heinrich Sträter Str. 1, 4600 Dortmund 1, ☎ (d) 0231/1882252, (p) 0231/730473

Suche möglichst günstig für HP41CX **IL-Modul** und **RS232-Interface**. Was gibt es z.Z. für Zugänge zu Maschinensprache auf dem 41'er? Jochen Dornauf, Buchrainweg 46, 6050 Offenbach, ☎ 069/839867

CCD-Modul für 150,- DM zu verkaufen. Literatur Wickes-Synthetische Programmierung 15,- DM. Jarett-Synthetische Programmierung leicht gemacht 15,- DM für HP-41C ☎ 0421/656439

Hallo Sammler!!! HP-10 Taschenrechner mit eingebautem Thermodrucker und Netzteil, 75,- DM; HP-41 Zubehör zum Barcodeleser 10,- DM; Daten, Dateien, Disketten (CHIP Buch, neu und ungelesen), 20 DM; CHIP Jahrgang 86 komplett und diverse 87er Hefte; Alles VHB ☎ 07131/484738

Verkaufe: HP-75C, Assembler-Handbuch + Kassette, Text75, **ThinkJet IL** (HP 222B), **HP 82163** Video-Interface, alle Einheiten mit IL, Handbücher, orig. Zubehör und Verpackung mögl. komplett abzugeben, DM 1.500,- VB Dr. Peter Reichetseder, ☎ 05943/83-140 (Büro bzw. 05921/35250 (privat, abends)

Verkaufe: HP41-Hardware: CCD-Modul (Version A) 160,- DM, Doppel-Speichererw.-Modul 40,- DM, IL-Schleife 150,- DM, IL-Video-Interf. HP28163B 200,- DM, defektes u. zerlegten HP41-C 40,- DM, ☎ 0431/323511

Programmsammlung „Kostenwesen und REFA“ für HP-41, W. Krell, Prozeptionsweg 29, 4720 Beckum.

Verkaufe wegen Systemwechsel:

HP-71B, 32K Speichererweiterung, HP-IL-Modul, HP-Mathe-Modul, HP-Forth-Assembler-Texteditor-Modul, HP-IL-Diskettenlaufwerk (HP-9114) mit zweitem Akku (9114B), HP-IL-ThinkJet, 2 Druckköpfe und Druckerständer, original HP-ThinkJet Papier (ca. 1500 Blatt), HP-71 Basic Made Easy, weitere Literatur und Programme alle Handbücher vorhanden, bei Gesamtabn. DM 3750,- VB. ☎ 040/7303157

Verkaufe HP-41 Translator PAC für HP-71, originalverpackt mit englischem Handbuch für 175 DM inkl. Porto, sowie HP-11C originalverpackt mit Handbuch für 85 DM inkl. Porto, Dennis Föh, ☎ 04721/24129

Verkaufe Hardware: HP-IL Modul für HP-41, HP2225B ThinkJet, Advanced Pac Screen mit Monitor.
Bücher: Synthetische Programmierung (WICKES), Synth Progr. made easy (GARETT), Calculator Tips and Routines (DEARING), Die Programmierbaren von HP (EK-KERT).
Suche: Handbuch zum HP-IL Converter (nur Original)
Matthias Rabe, ☎ 05204/4379

Verkaufe: HP-71B mit IL-Modul zusammen DM 750,-, HP82161A Cassettenlaufwerk mit 10 Cass. DM 150,-, HP 82162A Thermodrucker DM 400,-, HP2225B ThinkJet (neuwertig) DM 750,-,
Volker Lang (130), ☎ 07138/5349 ab 18 Uhr

Verkaufe HP125 PC, bestehend aus Monitor, Tastatur, Doppelfloppy (5 1/4") HP82901M, IB- und ser. Schnittstelle, 2 St. HP-IB-Kabel, Software, sehr ausf. Dok. VB 800,- DM,
2 Doppel-MM-f 41C 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. 06842/3041-3042) CCD 3389.

Verkaufe: HP-41 CV, X-Function, 32KB-Rambox (incl. Assembler III als Modul-File) im Kartenlesergehäuse mit Bedienungshandbüchern und Literatur (HP-41 Hilfen u. Anwendungen, Softwareentwicklung am Beispiel einer Dateiverwaltung, Optimales Programmieren mit dem HP-41, Assembler-Handbuch). Nur Komplett: DM 1100,- F. Alex, Wulfeistr. 11, 5860 Iserlohn, ☎ 02374/12003

Verkaufe: IL-ThinkJet HP2225B 800,- DM, Thermodrucker HP82143A 250,- DM, Rechner HP71B (Kartenleser), 3x4K Modulen, IL Modul sowie 41 Translator 5061-7269 900,- DM

Suche: Gerät, welches in Verbindung mit der IL-Schleife Spannung, Widerstand und Strom messen kann, und zwar am HP41. Da ich kein Telefon habe, bitte ich eine Karte zu schicken. Adresse:
Christian Grotkamp, Heidbergweg 6, 4300 Essen 15

Nachmieter gesucht:

3-Zi.Whg. mit Küche, Diele und Balkon, 66 qm; D-4750 Unna-Mühlhausen, Heerenerstr. 40; zum 1. Sept. 1988; Kaltmiete 580,- DM (einschl. Umlagen); Tel. (02303) 40504 Plümpe.

Verkaufe CMT 32 Kbyte RAM für HP-71 Frontport, 230 DM, Günther Hebekel, Tel. (06103) 81246 oder auch E-Mail GEO1.G.HEBEKERL.

Clubreise in die USA

**INTERNATIONAL CONFERENCE FOR
HAND-HELD COMPUTER USERS**

**Corvallis, Oregon/USA
vom 4. bis 6. August 1988,**

**eine Bildungsreise, die steuerlich
geltend gemacht werden kann.**

Nun ist es soweit, die Reise ist geplant und der Flug ist für eine Gruppe von maximal 16 Personen reserviert. Ein paar Plätze sind noch zu vergeben! Wer will noch mit, wer hat noch nicht? Eile ist jedenfalls geboten, und Kurzentschlüssene seien aufgefordert!

Die Reise startet am Freitag, 29.07.88, in Frankfurt mit American Airways nach Chicago und von da weiter nach Portland und Eugene, dem Zielflughafen. Die Rückreise beginnt in Eugene am Sonntag, 07.08.88, mit American Airlines über Chicago nach Frankfurt, wo die Reise am Montagmorgen, 08.08.88 endet. Die ganze Strecke wird mit American Airways geflogen.

Die nachstehend aufgeführten Reisepreise verstehen sich inklusive folgender Leistungen:

Hin- und Rückflug wie oben beschrieben, Bus-Transfer von und zum Flughafen, 9 Übernachtungen in einem Mittelklasse-Hotel in Corvallis ohne Frühstück sowie Registrierungsgebühr für den Kongress.

Der Reisepreis beträgt - sofern keine Tarifänderungen bei der Fluggesellschaft in der Zwischenzeit erfolgen - für die Übernachtung:

im Doppelzimmer DM 2.925,00
oder
im Einzelzimmer DM 3.200,00 .

Für Reiseteilnehmer, die am 29.07.88 höchstens 25 Jahre alt sind und deren 26. Geburtstag nach dem 29.07.88 stattfindet, gilt ein günstigerer Flugtarif. Sie zahlen bei Übernachtung:

im Doppelzimmer DM 2.625,00
oder
im Einzelzimmer DM 2.900,00 .

Schummeln geht nicht, denn die Geburtsdaten werden im Paß, der mit dem Flugticket vorzulegen ist, kontrolliert. Wenn der 26. Geburtstag während der Reise ist, spielt das keine Rolle.

Jeder Teilnehmer erhält eine entsprechende Rechnung, mit der der größte Teil der Reisekosten als Werbekosten steuerlich geltend gemacht werden kann.

Pro Tag sollten rund \$ 30,00 für Verpflegung gerechnet werden. Dieser Satz ermäßigt sich während der Konferenz auf die Kosten für das Frühstück von ca. \$ 8,00.

Es ist vorgesehen, daß vom 01.-03.08.88 eine Rundreise durch die Nationalparks Cascade Mountains, Crater Lake und Columbia River mit dem Bus unternommen wird, dafür muß ein Zuschlag von rund \$ 180 gerechnet werden. Somit summieren sich die Kosten in den USA auf ca. \$ 450,00 ohne Taschengeld für Einkäufe und ähnliches.

Diese Rundtour muß aber nicht unbedingt mitgemacht werden, da Corvallis und Umgebung vom Baden im Pazifik bis zum Skifahren (im Sommer!) auf den hohen Gletschern in den Rocky Mountains eine sehr breite Palette von Möglichkeiten des "Sich-Ergehens" bietet. Für Fahrten an die Küste und in die Berge benötigt man allerdings ein Mietwagen, was mit rund \$ 150,00 zu Buche schlägt. Wenn man insgesamt \$ 800,00 rechnet, wird man sehr gut auskommen.

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 Rödelheim
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 auszchnittsweise – nur mit schriftlicher Genehmigung des CCD. Eine irgendwie geartete Gewährleistung kann nicht übernommen werden.

Es empfiehlt sich, diesen Betrag je zur Hälfte in bar und Traveller-Cheques mitzunehmen.

Es sei denn, man will dort Computer und Peripherie kaufen oder Rundreisen über San Francisco und Los Angeles unternehmen, dann empfiehlt es sich, entweder die passenden Kreditkarten, wie Eurocard, Visa oder American Express, oder Reiseschecks in entsprechender Höhe bei sich zu haben.

Achtung: Eurochecks werden in den USA nur bei ganz wenigen Banken in den großen Städten eingelöst und werden sonst nirgendwo akzeptiert; deshalb

Reiseschecks mitnehmen, z.B. Amexco, die kann man überall einlösen!

Alles in allem eine touristisch interessant gestaltete Reise mit hohem Fachinformativwert, die man sich nicht entgehen lassen sollte. Die Einladung zur Konferenz ist in diesem Heft abgedruckt, so daß man sehen kann, was geboten wird.

Für Rückfragen zum Reiseprogramm und für die Anmeldeformalitäten stehe ich in den Abendstunden telefonisch unter (0201) 26 30 45 zur Verfügung.

Erich H. Klee (1170)

Inhalt

| | |
|------------------------------------|----|
| Clubnachrichten | |
| Club Börse | 2 |
| Clubreise nach USA | 2 |
| CCD Lexfile-ID | 3 |
| Impressum | 3 |
| Mitgliederversammlung 1988 | 4 |
| "Cursor" Zeitschriftenkritik | 48 |
| Grundlagen | |
| Der zweite Turmbau zu Babel | |
| Programmiersprachen 2. Teil | 7 |
| Vergleich von Navigationsmodulen | 9 |
| MS-DOS | |
| MS-DOS Referenzblätter | 25 |
| ATARI | |
| Atari-Gruppe im CCD | 5 |
| Serie 70 | |
| Literaturhinweis | 6 |
| TEDDY - der knuddelige | |
| Text-Editor | 11 |
| Diskettenkatalog | 18 |
| LEXCHECK 2 - Info über ID- und | |
| Token-Nummern | 21 |
| Lexfile: PRINTLX | 23 |
| Eigenwert-Berechnung | 24 |
| Serie 40 | |
| E-Technik Buchbesprechung | 27 |
| Dreiecks-Berechnungen | 28 |
| Buchbesprechung HP-41 Barcodes | 30 |
| Die Barcodes des HP-41 | 32 |
| Dreisatz | 34 |
| Barcodes von BEST OF PRISMA | 35 |
| Bug in der RAMBOX | 35 |
| Prüfsummen für Barcodes | 36 |
| Ermittlung von XROM-Funktionen | 36 |
| Aufruf zum Schreiben von Utilities | 37 |
| Die Hummel verstummt | |
| - summender HP-41 | 38 |
| XROMS | 38 |
| BANNER - Querdrucken | 39 |
| Kurvenapproximation nach Gauss | 41 |
| Thermodrucker mit Infrarot | 48 |
| Serie 10 | |
| Regression mit Polynomen | 42 |
| Barcodes | 45 |
| Clubadressen | 47 |

CCD Lex-File ID

Nach langem Warten ist sie nun endlich da: die clubeigene Lex-File ID #. Von Hewlett-Packard ist uns jetzt offiziell die ID # E1 zugeteilt worden.

Es ist interessanterweise die gleiche, wie sie auch die Franzosen haben. Zweimal dieselbe ID # zu vergeben, geht das überhaupt? Leider ja, aber es ist weniger schlimm als im ersten Augenblick zu vermuten war. Es wird kein Chaos mit gleichbelegten deutschen und französischen Token-Nummern geben, denn unsere französischen Clubfreunde haben nicht alle 256 Tokens erhalten. Somit sind für den CCD 48 Token-Nummern von D0 bis FF freigeblieben, die wir nun offiziell zur Programmierung von Lex-Files verwenden dürfen.

Zunächst einmal sei darauf hingewiesen, daß diese Token-Nummern NICHT zum Probieren und Experimentieren freigegeben sind, sondern ausschließlich für vom CCD offiziell festgelegte Befehle und Funktionen.

Dabei hat sogar auch Hewlett-Packard noch einige Worte mitzureden, damit doppelt vergebene Befehlsnamen vermieden werden.

Da wir aber noch weitgehend am Anfang stehen, sind alle Interessierten dazu aufgerufen, ihre Ideen und Anregungen mitzuteilen und an diesem Projekt mitzuarbeiten.:

Computerclub Deutschland e.V.
Stichwort: CCD-Lexfile
Postfach 11 04 11
D-6000 Frankfurt am Main 1

In diesem Sinne wünsche ich ein Happy Programming

Alf-Norman Tietze
verantwortlicher Redakteur

Titelfoto: Yvonne Roth, Frankfurt

Der kleine Teddy wurde freundlicherweise von Beatrix Werner, Frankfurt zur Verfügung gestellt

CCD e.V. Mitgliederversammlung 1988

Liebe Freunde,

als erste Information zur stattgefundenen Mitgliederversammlung hier den

Rechenschaftsbericht

des Vorstandes, vorgetragen von Prof. Dr. Wolfgang Fritz:

1. Begrüßung durch den Vorstand

Laut Satzung § 13 muß möglichst im 1. Quartal die ordentliche Mitgliederversammlung stattfinden. Wegen der Witterungsverhältnisse schien uns der Januar und Februar nicht gut geeignet, der März war durch die CEBIT belastet. Die jetzige Mitgliederversammlung wurde ordnungsgemäß im PRISMA Heft 1 mit Angabe der Tagesordnung rechtzeitig, nämlich mehr als vier Wochen vor dem Versammlungstermin einberufen.

2. Feststellung der Beschlußfähigkeit und andere Formalitäten

Beschlußfähigkeit wird festgestellt, da mehr als 30 Mitglieder anwesend sind.

Abstimmungen werden, falls notwendig, mit einer Stimmkarte durchgeführt, die den CCD-Mitgliedern von Alexander Wolf nach Eintrag in eine Anwesenheitsliste ausgehändigt wird.

Beiratsmitglied Alexander Wolf (ergänzt um die beiden Kassenprüfer Achim Gmein und Gerhard Link) sollen die Mandatsprüfung durchführen.

Schriftliche Anträge zur TO sind nicht eingegangen.

Die Tagesordnung wird bestätigt.

3. Bericht des Vorstandes (Vors.)

3. a) Rückblick um ein Jahr

Die letzte MV am 4.4.87. Damals wurde der aufmüpfige und besserwisserische Beirat zur Konsequenz gezwungen: Erich Klee, Werner Dworak und ich übernahmen Vorstandsverantwortung. Ebenso mußte der früher als kritischer Kassenprüfer aufgefallene Dieter Wolf, wenn auch inzwischen zum Schatzmeister arriviert, weiter die Konsequenzen tragen, die alle Besserwisser tragen sollten.

Alf-Norman Tietze kam zu unserer Vorstandsrunde als Redaktionschef von PRISMA dazu.

3. b) Sitzungen, Beiratsbeteiligung, Protokolle

Was haben wir gemacht? Die Neueintragung beim Registergericht wurde zügig durchgeführt und damit nach Jahren wieder aktualisiert.

In dem einen Jahr haben wir sechs ordentliche Vorstandssitzungen gehabt, und zwar am 2.5., 18.7., 3.10., und 13.12. und am 26.3.88 und gestern abend. Die Vorstandsmitglieder waren jeweils vollzählig beieinander. Vom Beirat, den wir grundsätzlich zu unseren Vorstandssitzungen eingeladen hatten, war Alex Wolf, die Gunst des Wohnorts nutzend, jedesmal dabei, zusätzlich meist auch Stefan Schwall, ferner Günther Schwarz, Peter Kemmerling und Martin Meyer. Die Sitzungen und vor allem ihre Ergebnisse wurden ausführlich protokolliert, die Protokolle neben den Vorstandsmitgliedern auch jedem Beiratsmitglied und natürlich dem Justitiar, Herrn Widl, zugesandt.

3. c) Aufgabenverteilung

1. Vorsitzender: Außenvertretung (Mitgliederversammlung, Koordination im Vorstand, Kontakt zum Beirat, Aufgabenverteilung an Mitglieder (insbesondere bei bezahlter Tätigkeit)

2. Vorsitzender: Vertretung des 1. Vorsitzenden und Werbung für den Club, P.R.-Aktivitäten, Messeaktivitäten, Anzeigenacquisition

Schatzmeister: Buchhaltung, Einkauf, Auftragserteilung bei der Prismaherstellung

1. Beisitzer: Betreuung der Fachgruppen (CP/M, MS-DOS, ...), Betreuung der Regionalgruppen, Mailbox

2. Beisitzer: Redaktion Prisma

3. d) Mitgliederstand

Sorge macht die negative Entwicklung des Mitgliederstandes: Zur Zeit sind es ca. 2028 Mitglieder, davon ca. 140 in der CP/M- und ca. 250 in der MS-DOS-Gruppe. In der besten Zeit gab es 2400 Mitglieder.

Wir vermuten, daß pro PRISMA 2 Mitleser dabei sind; Erich Klee, bereitet einen clubinternen Wettbewerb vor: In der Zeit bis zum Jahresende wird ein Wettbewerb zur Mitgliederwerbung ausgeschrieben. Die drei Mitglieder, die in diesem Zeitrahmen die meisten Neumitglieder (aber mindestens 3 Neumitglieder) erworben haben, erhalten Prämien. Ferner werden Preise ausgelost, dabei erhält jeder Werber soviel Lose, wie er Mitglieder erworben hat. Es soll versucht werden, Preise von Firmen (hp, Zenith usw.) zu erhalten.

Rolle des HP-41, 71, 75, Atari, CP/M und MS-DOS

Traditionsgemäß werden im CCD behandelt die HP-Rechner 41, 71, 75. Der Großteil der PRISMA-Artikel beschäftigt sich mit diesen Rechnern, die ja einmal

– ich denke an den 41er – Ausgangspunkt für die Gründung des Vereins waren.

In unserem Verein waren die ersten Nicht-HP-Rechner die Osborne-1-Rechner. Seitdem gab es eine Fachgruppe für Rechner mit CP/M-Betriebssystem, die Werner Dworak aufzog; im Jahre 1987 wechselte die Betreuung von Fritz Zick in Berlin zu Peter Cornelius Spaeth in München und läuft mit neuem Schwung mit den etwa 140 Teilnehmern weiter. Da CP/M, wenn auch ein schönes Betriebssystem, auf dem Markt immer mehr schwindet, ist allerdings ein Sinken der Teilnehmerzahl bemerkbar. Da die vorge-sehene Anzahl von ca. 10 Diskettenausgaben für 1987 nicht nachgeholt werden konnte und zudem die Roh-Disketten preiswerter geworden sind, wurde eine Rückvergütung beschlossen.

Steigende Teilnehmerzahlen bei MS-DOS: 250 Mitglieder sind es jetzt, die an der von Alexander Wolf betreuten MS-DOS-Runde teilnehmen. Dem häufig geäußerten Wunsch nach 3,5''-Disketten wurde Rechnung getragen.

Darüberhinaus wurde eine ATARI-Gruppe eingerichtet von Werner Müller in Köln, die ähnlich wie die CP/M- und die MS-DOS-Gruppe eine eigene periodisch erscheinende Informationsdiskette herausbringt. Die Teilnehmerzahl ist noch klein, es gibt bisher (nur) 13 Mitglieder. Der Vorstand hat beschlossen, diese Mitglieder zunächst weiterhin nur mit dem Grundbeitrag des CCD zu belasten, bis eine tragfähige Gruppe existiert. Die Zukunft der Atari-Gruppe scheint jedoch durch die Konkurrenz örtlicher spontaner Gruppen gefährdet.

Eine Entscheidung soll zum Jahresende fallen.

3. e) PRISMA an Bibliotheken

Seit Jahresmitte werden die wissenschaftlichen Bibliotheken in der Bundesrepublik, soweit sie auf eine Anfrage von uns geantwortet haben, mit PRISMA beliefert. Dies soll ein Beitrag zur satzungsgemäßen Förderung von Wissenschaft und Forschung auf dem Gebiet der Computertechnik sein.

3. f) Gemeinnützigkeit

Einem Mitglied in Mainz ist von seinem Finanzamt der CCD-Beitrag nicht anerkannt worden. Damaliger Stand: 1981 ist für 18 Monate dem CCD die Gemeinnützigkeit vorläufig anerkannt worden. Der CCD mußte gegenüber dem Finanzamt die Jahre 1981 – 1986 belegen. Für die Jahre vor 1984 gab es nur Belege, aber keine Buchungsunterlagen, ab 1984 waren Buchungsunterlagen vorhanden. Es mußte also nachgebucht werden.

Dieter Wolf hatte Kontakt mit dem Sachbearbeiter im Finanzamt aufgenommen. Es mußten noch ein Fülle von Restbuchungen durchgeführt werden. Erst im letzten Monat konnte endlich die mühselige Aufarbeitung der Finanzsituation seit Beginn des CCD von Dieter Wolf abgeschlossen werden. Die Unterlagen sind zur Vorlage beim Finanzamt fertiggestellt.

Ziel der Anstrengung sollte die Anerkennung der Gemeinnützigkeit des Vereins sein. Falls jedoch die Gemeinnützigkeit, die wir nie echt hatten (nur eine eineinhalbjährige vorläufige Anerkennung existierte) aufgrund restriktiven Verhaltens der Finanzbehörden nicht ohne weiteres erlangt wird, will der Vorstand keine weiteren übermäßigen Anstrengungen durchführen aus den folgenden Gründen: Viele Mitglieder sind Schüler/Studenten ohne eigenes Einkommen und können so steuerlich nichts absetzen; andere Mitglieder können die CCD-Beitragskosten als Werbungskosten absetzen.

3. g) Inhaltsverzeichnis von PRISMA

Die Erstellung des Inhaltsverzeichnisses von PRISMA, von Familie Kiefer begonnen, soll, wir hoffen von Henry Schimmer, fertiggestellt werden.

3. h) UPLE

Die UPLE-Bibliothek (für HP-Rechner) lag ungeordnet in den Geschäftsräumen. Das Vorsortieren anhand eines Katalogs übernimmt Erich Klee. Die Zusammenstellung der Programme für die einzelnen Kategorien sollen von einzelnen Zuständigen durchgeführt werden, für die zum Teil schon Namen benannt werden konnten:

- Vermessung:
 - Manfred Hammer
- Numerische Mathematik:
 - Andreas Wolperts
- Finanzen:
 - Erich Klee
- Computerwissenschaften:
 - Martin Meyer
- Geo-Wissenschaften:
 - Alf-Normen Tietze
- Physik/Chemie:
 - NN
- Sozialwissenschaften:
 - NN
- Engineering:
 - Martin Meyer

Nicht als Einzelprogramme, sondern als ein Block von Programmen einheitlicher „Kategorie“ sollte dies den Mitgliedern zur Verfügung gestellt werden. Der momentane Stand ist: Die Puzzle-Arbeit ist soweit gediehen, daß inzwischen fast alles sortiert im Rechner ist. Über weiteres wird im PRISMA nachzulesen sein.

3. i) USA-Studienreise

Der auch im PRISMA veröffentlichte Plan, eine gemeinsame Studienreise nach den USA für CCD-Mitglieder zur organisieren, ist wohl gescheitert, da sich bisher nur eine Handvoll Mitglieder näher informiert haben, bis nach Ostern jedoch mindestens 25 Interessenten hätten bekannt sein müssen.

3. j) Messen

In diesem Jahr gab es keine Messebeteiligungen, obwohl sie grundsätzlich erwünscht bleiben. Ob eine Beteiligung an der CEBIT 89 (kostengünstig oder

kostenfrei) möglich ist, wird vom 2. Vorsitzenden Erich Klee untersucht.

3. k) PRISMA-Herstellung

Ziel der Änderungen war die Verbesserung von PRISMA in Inhalt und Aufmachung und gleichzeitig die Verringerung der Kosten bei der PRISMA-Herstellung. Die Druckherstellung wurde in die Reha-Werkstatt in Rödelheim verlegt. Stefan Schwall hat ein geeignetes Programm zur Barcode-Erstellung geschrieben und dem CCD zur Verfügung gestellt. Die Setzerei Fotosatz-Studio Lind/Neumann in Neu-Isenburg berechnet nach Aufwand und ermöglicht somit dem CCD eine weitere Kostenreduzierung. Zu den Satzkosten kommen noch die Montage- und Reprokosten. Dieter Wolf hatte Kontakt mit Fachleuten aufgenommen; zwei Leute können an zwei Abenden eine Ausgabe montieren, auf Pauschalbasis. Die Redaktion wurde um einen Hersteller (Dieter Wolf) erweitert. Ohne dies neue Konzept, das auf Wunsch Dieter Wolf gerne näher erläutern wird, wären wir heute wahrscheinlich in finanziellen Schwierigkeiten.

Bei Wechsel im technischen Bereich bestand ein günstige Gelegenheit für eine Überprüfung und Verbesserung des Layouts. Wir haben die Gelegenheit genutzt. Ich hoffe, daß die neue Form von allen Mitgliedern als Verbesserung registriert wurde.

Anmerkung:

Sobald das Protokoll der MV vorliegt, wird es in PRISMA veröffentlicht werden.

ATARI-Gruppe im CCD

Im August 1987 wurde innerhalb des CCD's eine Atari-Gruppe gegründet. Sie soll, wie auch die CPM-Gruppe und MS-DOS-Gruppe, einen festen Platz im CCD erhalten. Nun nach 8 Monaten möchte ich Bericht über die bisher herausgegebenen Disketten und über noch geplante Aktivitäten berichten.

Diskette 1:

Diese Diskette entspricht der als Probediskette versendeten Diskette. (Sie ist weiterhin für 5,- bei mir zu haben . . .). Diese Diskette ist insbesondere für Neueinsteiger gedacht. Die nützlichen Dinge, die bei dem Atari nach dem Auspacken noch fehlen: eine Ram-Disk (um auch mit einem Laufwerk arbeiten zu können), ein Kommando-Interpreter (man fühlt sich dann wie auf einem MS-DOS Rechner), ein einfacher Editor (um direkt einen Beitrag für die nächsten Atari-Disketten schreiben zu können), ein ein-

faches Malprogramm, um die ersten Bilder zu erstellen, und ein Spiel zum Abwarten sind auf dieser Diskette enthalten.

Diskette 2:

Diese Diskette enthält einen CPM-80 Emulator (für die Umsteiger aus der CPM-80 Gruppe), einen Testbericht über DBASE auf dem Atari und das Spiel Reversi (für die Umsteiger von IBM-kompatiblen Rechnern). Auch ist der erste Beitrag eines Atari-Gruppen-Mitglieds über das Programm Beckertext, auf dieser Diskette zu finden.

Diskette 3:

Diese Diskette enthält sogenannte Accesorie's = Beigaben. Diese Programme können (wie Sidekick auf dem MS-DOS Rechnern, nur viel bequemer), jederzeit benutzt werden. Für ordnungsliebende Computerbesitzer gibt es ein einfaches Diskettenverwaltungsprogramm. Für

Datenübertragungsfan's wird Kermit geliefert und für Experten ein vollständiges VT100 Terminal mit eingebauter Textronic-Graphik Emulation (Uniterm). Last not least, ein noch einfacheres Malprogramm als auf Diskette 1.

Diskette 4 demonstriert die Graphikfähigkeiten des Ataris. (Moleküle).

Diskette 5:

enthält einen noch leistungsfähigeren Editor als Diskette 1, die fehlende Beschreibung zu Uniterm von Diskette 3 sowie ein sehr leistungsfähiges Zeichenprogramm mit einer Macintosh ähnlichen Benutzeroberfläche. Für Xmodem-Benutzer gibts schließlich noch Xmodem für den Atari.

Diskette 6

enthält alle Programme, die man benötigt, um ungewöhnlich formatierte Disketten

ATARI

| Rechner Basic | $x^{(2.5^{20})}$ x = nach Programm | $x^{(2.5^{20})}$ y = nach Programm | $x^{(2.5^{20})}$ z = nach Programm (oder direkt) | Bemerkung |
|--------------------------------|---------------------------------------|---------------------------------------|--|---------------------------|
| HP 41 | | | 8910.1518 | |
| HP 71 | 8909.9575 | 8910.10778 | 8910.1518 | |
| HP 9000/320 Basic 4.0 | 8910.151926 | 8910.151882 | 8910.151897 | mit Float.Point Prozessor |
| HX20 MS-Basic | 1 | 38695.188 | 1 | DEFSNG |
| | 1 | 18435.238 | 1 | DEFDBL |
| Atari ST: | | | | |
| ST-Basic | 1 | 9.97701 E8 | 1 | DEFSNG |
| ST-Basic | 1 | 9.97701 E8 | 1 | DEFDBL |
| GFA-Basic | 2.0 | | | |
| | 9177.5471 | 8910.7861 | 9070.0931 | |
| (Nur eine Genauigkeit möglich) | | | | |
| Omikron-Basic | 8910.1518500 | 8910.1518500 | 8910.152054 | |
| (Höchstmögliche Genauigkeit) | | | | |

zu formatieren und zu kopieren. Es gibt ein Programm für Ordnungsliebende, das auch die exotischen Kassettenhüllen und Ordner passgenau beschriften kann. Auch ist wieder ein schönes Spiel enthalten und es werden erste Tips zum täglichen Umgang für den Atari gegeben. (Nach 5 Disketten!!!).

Diskette 7

Beiträge von Mitgliedern der Atari-Gruppe!!!
Testberichte, Programme und Benchmarks zu Omikron.Basic.

Diskette 8:

Schon wieder Beiträge von Mitgliedern der Atari-Gruppe!!!
Fractal's = Apfelmännchen in Schwarz-weiß und Farbe . . .

An Beiträgen und Programmen ist noch viel in der Planung oder wird gerade verwirklicht. Es wird ein weitverbreiteter Programmierer kommen (Wer kennt EMACS noch nicht?), es wird über die Unterschiede von HP-Basic und Omikron-Basic berichten, praktische Tip's zur Erstellung von Serienbriefen werden kommen und vieles? mehr.

Public Domain Software gibt es für den Atari megabyteweise . . .

Die gute = brauchbare, lehrreiche oder schöne wird auch zu den CCD-Disketten durchdringen. Eine ordnende Hand wird hier noch gesucht.

Ohne den Atari-ST verkaufen zu wollen, möchte ich sagen, warum ich diesen Computer anderen vorziehe. Es gibt kaum einen zweiten Computer, der neben seinem eigenen Betriebssystem so viele verschiedene andere Computer emuliert. Auf dem Atari läuft CPM-80 Software und

MS-DOS Software ohne Hardwareänderungen (Die Geschwindigkeit ist zugegebenermaßen etwas langsam). Es gibt neue „Multiuser und Multitasking“ Betriebssysteme wie RTOS oder OS9 oder alte Betriebssysteme für den 68 000 wie CPM68K. Das Atari-eigene Betriebssystem TOS ist seiner graphischen Benutzeroberfläche GEM kommt gerade Computereinsteigern sehr entgegen. Programme kosten für den Atari in der Regel nur 1/3 als vergleichbare Programme für IBM und IBM-kompatible Rechner. Der 68 000 Prozessor ist sehr leistungsfähig. Inzwischen kann man den Atari-ST schon mit einem 68 020 ausrüsten. Na dann . . .

Nur noch eine Zahl aus der CT April 1988. Inzwischen wurden etwa 250 000 Atari-ST Computer in Deutschland verkauft . . .

Ganz zum Ende noch ein kleines Testprogramm zur Bestimmung der Genauigkeit von Omikron.Basic:

Auf den Disketten wird dieser Basic-Dialekt in Zukunft unterstützt.)

Kurze Information zu Omikron.Basic:

Es ist sehr ähnlich zu MS-Basic, GW-Basic und HP-Basic. Es unterstützt das Betriebssystem des Ataris bis in die letzte Einzelheit. Es erlaubt die Definition von Prozeduren und Funktionen (auch Rekursiv!) mit Parameterübergabe.

Es ist sehr schnell und rechnet erstaunlich genau:

Der folgende Test wurde von Peter Jochen aus Reutlingen durchgeführt.

Test der Potenz-Funktion y^x

Programm in Basic:

```

0 X=1.0000001
1 Z=X^(2.5^20)
2 Y=X
3 FOR A%=1 TO 20
4     X=X^2.5
5     Y=Y*Y*SQR(Y)
6 NEXT A%
7 PRINT „X=“;X
8 PRINT „Y=“;Y
9 PRINT „Z=“;Z
    
```

Werner Müller
Classen-Kappellmannstraße 30a
5000 Köln 41

Nachsendedienst

Computerclub Deutschland e.V.
PRISMA-Nachsendedienst
Postfach 11 04 11
D-6000 Frankfurt am Main 1

Literaturhinweis

Die in PRISMA 3/87, Seite 19 "Temperaturverlauf in einem Wärmequellennetzwerk" zu Grundlagen angekündigte Arbeit: **K. Lutz und E. Thum, Erweitertes Wärmequellennetz zur Berechnung instationärer Vorgänge** ist in 'Zeitschrift für elektrische Energietechnik' etz-Archiv Band 9, 1987, Heft 6, Seiten 199-201 erschienen.

Dr. Karlheinz Lutz
Bayernstr. 39
D-8501 Rückersdorf

Der zweite Turmbau zu Babel

von Dr. Ralf Kern (1100)

2. Teil

C schlägt alles an Effizienz

Etwas zur gleichen Zeit wie Pascal wurde eine andere Sprache „geboren“, die inzwischen eine ähnlich überragende Bedeutung gewonnen hat: C. Sie wurde, aus den Vorläufern BCPL und B hervorgegangen, von Biran W. Kernighan und Dennis M. Ritchie in den Laboratorien von Bell zur Implementierung des Betriebssystems Unix konzipiert, ist inzwischen aber über die Unix-Welt weit hinausgewachsen. Gemäß der ursprünglichen Zweckbestimmung wurde bei C auf Hardware-Nähe und auf höchste Effizienz geachtet, denn ein Betriebssystem muß schnell und kompakt sein. C-Programme brauchen nur noch wenig mehr Speicherplatz und Laufzeit als äquivalente Assembler-Programme, gegenüber denen man bei C mit der anderthalb- bis zweieinhalbfachen Laufzeit rechnet, bei Pascal mit der doppelten bis fünffachen, und noch mehr bei FORTRAN, COBOL und BASIC.

Man kann daher C als das Ergebnis eines Versuchs ansehen, zwei Fliegen mit einer Klappe zu schlagen, nämlich die Vorteile von Assembler- und Hochsprachen miteinander zu verbinden. Wie wir eingangs schon festgestellt hatten, ist Assemblerprogrammieren kompliziert, zeitraubend und fehleranfällig. Andererseits hat ein kompiliertes Hochsprachenprogramm die erwähnten Nachteile bezüglich Speicherplatz und Laufzeit. Daher wird trotz der Vorteile der problemorientierten Sprachen auch heute noch viel in Assembler programmiert, vor allem wenn es auf Hardware-Nähe oder Schnelligkeit ankommt, also etwa bei Systemsoftware und Editoren, die schließlich mit der Schreibgeschwindigkeit einer flinken Stenotypistin mithalten können müssen. So ist z.B. WordStar bis zur Version 3.4 in Assembler programmiert.

Es gibt eine Reihe von Lösungsansätzen für dieses Problem. „Von unten her“ versucht man, Assemblersprachen anzureichern, vor allem um Sprachkonstrukte zur Programmablaufsteuerung (z. B. IF – THEN – ELSE) oder zur Behandlung von Datenstrukturen. Dadurch will man an Programmierkomfort und -sicherheit von Hochsprachen herankommen. Hilfsmittel hierfür sind Präprozessoren, die assemblerfremde Anweisungen in Assemblerbefehle auflösen, oder sogenannte Makroübersetzer, die es dem Programmierer gestatten, ganze Befehlsformen in „Makrobefehlen“ zusammenzufassen und damit beim Programmieren die gewünschte höhere Abstraktionsebene zu erreichen. So erstellte Programme sind zwar genauso effizient wie reine Assemblerprogramme, bleiben aber weiterhin voll hardwareabhängig.

In der anderen Richtung, „von oben her“, wurden hierfür eigens auf Effizienz hin getrimmte Hochsprachen entworfen. Neben C sind hier PL/M, PLZ und – mit einigen Abstrichen – auch Forth zu nennen. PL/M (Programming Language for Microprocessors) wurde von der Firma Intel durch Abmagerung von PL/I geschaffen und weist einige strukturelle Ähnlichkeiten zu C auf, ist aber weniger leistungsfähig, dafür noch hardware-näher als C. Die ähnlich geratene Sprachfamilie PLZ stammt von der Firma Zilog und hat sich ebenso wie PL/M nur im Industriebereich verbreiten können.

A propos Forth: Benutzern von HP-Rechnern ist diese Sprache recht vertraut, da sie von Hewlett-Packard für Eckzeitwendungen im Labor- und Datenerfassungsbereich nachhaltig propagiert wird. Was vielleicht nicht so augenfällig ist, Forth hat offensichtlich bei der „Geburt“ der Architektur des HPH 41 C kräftig Pate gestanden, denn es finden sich dort zahlreiche Forth-Konzepte wieder, z.B. die Berechnung arithmetischer Ausdrücke in der umgekehrten polnischen Notation (Reversed Polish Notation, RPN) mittels eines Register-Stapels (Stack-Orientierung), die Erweiterung des Vokabulars fest einprogrammierter Funktionen durch eigene Programme, die Verkettung globaler Marken, der Begriff des „Compilierens“ von lokalen Sprungbefehlen usw.

Indessen ist Forth ziemlich umstritten: Die Anhänger loben Effizienz, Flexibilität durch konzeptionelle Erweiterbarkeit, ja und auch Klarheit der Sprache; die Gegner hingegen kritisieren Forth als schwer verständlich und behaupten, daß ein Forth-Programmierer nach ein paar Monaten nicht mehr in der Lage sei, seine eigenen Programme zu verstehen.

An dieser Stelle seien noch zwei weitere interessante „Exoten“ erwähnt: PEARL und APL. PEARL (Process and Experiment Automation Realtime Language) ist eine ALGOL-ähnliche Sprache, die um Konzepte für die Echtzeitverarbeitung erweitert wurde (z.B. Ereignisse, Prozesse, Zeitbedingungen, Datenstationen). PEARL wurde von der deutschen Computer-Industrie unter Beteiligung deutscher Forschungsinstitute entwickelt. Dementsprechend ist PEARL vor allem auf Prozeßrechnern und seit neuestem auf dem Atari ST verbreitet.

APL (A Programming Language) hingegen wurde 1962 von Kenneth Iverson und A. D. Falkoff bei IBM zunächst als eine Art mathematischer Schrift entwickelt und später als Programmiersprache implementiert. Sie ist vor allem auf IBM-(Groß-)Rechnern verbreitet und eig-

net sich vor allem für technisch-wissenschaftliche und betriebswirtschaftliche Probleme, da sie über besonders handliche Operatoren für die Vektor- und Matrizenrechnung verfügt. Als Beispiel für die große Flexibilität von APL sei aber auch erwähnt, daß man die Architektur des /360-Systems von IBM in APL formal beschrieben hat. Dennoch hat APL eine recht einfache Syntax und kennt z.B. nur drei Anweisungsformen und nur Funktionen mit bis zu zwei Parametern.

Neulinge schreckt APL durch die verwirrende Symbolik eher ab: Jeder Operator wird durch ein spezielles Graphik-Zeichen notiert, wodurch besondere Anforderungen an APL-fähige Bildschirme gestellt werden. Ausdrücke werden unabhängig von normalen Vorrangregeln („* und / geht vor + und –“) von rechts nach links abgearbeitet, und die Vielfalt und Leistungsfähigkeit der APL-Operatoren führt zu sehr kompakten Programmen.

Was setzt sich durch – Pascal oder C?

Es lohnt sich, die beiden derzeit wohl aktuellsten Sprachen, Pascal und C, näher zu vergleichen, da in der Fachliteratur ein fast erbitterter Streit darüber geführt wird, welche der beiden Sprachen vorzuziehen sei. Die Auseinandersetzung gipfelt in dem Verdikt eines der beiden Schöpfer von C, B. W. Kernighan, in seinem Pamphlet mit dem Titel „Warum Pascal nicht meine Lieblingssprache ist“. Ihm zufolge ist (Standard-)Pascal nicht geeignet, ernsthafte Programme zu schreiben. Trotz aller Argumente, mit denen Kernighan seine These untermauert, übergeht er dabei, daß die meisten Pascal-Compiler diese Schwächen durch (nicht-standardisierte) Spracherweiterungen beseitigt haben und daß umfangreiche, auch kommerziell genutzte Software wie Compiler, Editoren usw. erfolgreich in Pascal erstellt wurde. Darüber hinaus hat Pascal die Entwicklung zahlreicher anderer Sprachen, allen voran Ada, nachhaltig beeinflusst.

Auf den ersten Blick weisen Pascal und C einige Ähnlichkeiten auf, vor allem im Bereich der Datenstrukturen und der Ablaufsteuerung. Dennoch unterscheiden sie sich fundamental in der Grundphilosophie: C versucht, so wie eine Assemblersprache, dem Programmierer möglichst große Freiheit und Flexibilität zu geben – mit dem Nachteil relativ hoher Fehleranfälligkeit. Selbst erfahrene C-Programmierer tapen immer wieder in eine der vielen Fallen von C.

Pascal hingegen ist auf hohe Sicherheit für den Programmierer bedacht und fängt zahlreiche Fehler schon in der Übersetzungsphase ab – dafür wird der Programmierer erheblich eingeengt. Man kann dies etwas vereinfacht so zusammenfassen, daß Pascal dem Anfänger das Erlernen einer Programmiersprache durch die Schutzmechanismen erleichtert, während mit C dem versierten Software-Entwickler ein effizientes und flexibles Werkzeug in die Hand gegeben ist. Anfänger sollten besser die Finger von C lassen, denn manche C-Compiler lassen sogar gewisse Syntaxfehler durchgehen!

Diese Überlegungen lassen sich am besten durch die Fortsetzung des bei der Besprechung von Pascal begonnenen Beispiels verdeutlichen. In C muß man die Wochentage wie in ALGOL oder FORTRAN durch Zahlen codieren, wobei es sich empfiehlt, die Möglichkeit in C zu nutzen, Konstanten zu definieren:

```
#define MONTAG 1
#define DIENSTAG 2
#define MITTWOCH 3
#define DONNERSTAG 4
#define FREITAG 5
#define SAMSTAG 6
#define SONNTAG 7
```

Definition der Datentypen WOCHE und WERKWOCHE als ganze Zahlen (int):

```
typedef int WOCHE;
typedef int WERKWOCHE;
```

Nun werden die Variablen „wochentag“ und „werktag“ deklariert:

```
WOCHE wochentag;
WERKWOCHE werktag;
```

Rein formal ist die folgende Anweisung fast identisch mit der entsprechenden Pascal-Anweisung, aber es verbirgt sich dahinter eine ganz andere Bedeutung:

```
wochentag = DIENSTAG;
```

weist in C der ganzzahligen Variablen „wochentag“ die Zahl 2 zu, die hinter dem symbolischen Namen DIENSTAG steckt, während im Pascal-Beispiel der Wert DIENSTAG des Datentyps WOCHE einer Variablen desselben Typs zugewiesen wird. Indessen ist in C die unlogische Befehlszeile

```
werktag = SONNTAG;
```

vollkommen zulässig, da die Datentypen WOCHE und WERKWOCHE nur Umbenennungen des Datentyps „int“ („integer“, also ganze Zahl) sind, ohne daß Grenzen für die Wertebereiche von WOCHE bzw. WERKWOCHE angegeben werden können. Daher schöpft der C-Compiler keinen Verdacht bei dieser Anweisung, während der Pascal-Compiler sie nicht hätte durchgehen lassen. Der logische Fehler des C-Programmierers wäre also nicht verhindert worden

und würde sich erst viel später beim Programmtest oder gar erst beim praktischen Einsatz bemerkbar machen. Dann läßt er sich aber nur mit viel größerer Mühe beheben. – Unser Pascal-Beispiel mit einer Programm-Schleife lautet übrigens in C:

```
for (wochentag = MONTAG; wochentag
<= SONNTAG; wochentag++)
```

```
berechne (umsatz (Wochentag));
```

Auch bei der Behandlung von Unterprogrammen und Funktionen gibt es markante Unterschiede zwischen C und Pascal. In C wird nicht geprüft, ob Anzahl und Typ der übergebenen Parameter bei Unterprogramm-Aufruf und -definition miteinander übereinstimmen – auch darin liegt Chance und Gefahr zugleich für den Programmierer. Da Pascal-Compiler diese Überprüfung vornehmen, muß immer das komplette Programm übersetzt werden – ein großer Nachteil bei der Durchführung großer Softwareprojekte, vor allem, wenn mehrere Programmierer daran beteiligt sind –, während in C auch Programmteile (Module) separat kompiliert und erst später zum Gesamtprogramm zusammengebunden werden können. Bei Modula-2 und Ada, zwei Nachfolgern von Pascal, sind Parameterüberprüfung und separate Übersetzung auf elegante Weise miteinander vereint worden.

Hart auf den Fersen: die Nachfolger

Damit ist die Rede auf die Weiterentwicklung von C und Pascal gekommen. Zu C gibt es die „objektorientierte“ Nachfolgersprache C++, bei der C um das sogenannte Klassenkonzept erweitert wurde. C++ ist aber noch zu neu, um bisher größere Verbreitung erlangt zu haben.

Modula-2 hingegen, das wieder aus dem Hause Wirth stammt, stützt sich auf Pascal, vermeidet aber die meisten von dessen Nachteilen. Da Pascal-Experten der Umstieg auf Modula-2 besonders leicht fällt und inzwischen auch eine Reihe leistungsfähiger Compiler auf den Markt gekommen ist, kann man dieser Sprache ohne Zögern eine erfolgreiche Zukunft prognostizieren.

Dasselbe gilt für Ada, aber aus anderen Gründen: Ada wurde im Auftrag des amerikanischen Verteidigungsministeriums entwickelt und streng standardisiert. Nur noch die Sprache Ada darf in Software-Projekten dieses Ministeriums (und bald auch anderer NATO-Dienststellen) eingesetzt werden. Damit ist für Ada ein kräftig expandierender Markt vorgegeben.

Im Hinblick auf den militärischen Einsatzbereich wurde bei der Spezifikation von Ada besonderer Wert auf Sicherheitsmaßnahmen gelegt – kein Wunder, daß sich entsprechende Pascal-Konzepte in Ada wiederfinden. Gerade in diesem Punkt ist allerdings scharfe Kritik an Ada geübt worden, vor allem von C. A. R. Hoare aus Oxford, der diese Sprache vor allem ihrer Unübersichtlichkeit, ihres großen Umfangs (auch Ada tendiert zur „eierlegenden Wollmilchsau“!) und damit ihrer Kompliziertheit wegen für sehr unsicher hält.

Ada ist übrigens das Ergebnis eines vom amerikanischen Verteidigungsministerium ausgeschriebenem Wettbewerbs, den eine Arbeitsgruppe der französischen Firma Honeywell Bull unter der Leitung von Jean D. Ichbiah mit dem Ada-Entwurf gewann. Im Mikrocomputer-Bereich dürfte diese Sprache in naher Zukunft noch keine allzu große Rolle spielen, da vollständige Ada-Compiler nicht selten eine Codelänge von mehr als einem Megabyte haben und damit gar nicht auf einen Rechner vom Schlage eines IBM PC oder gar eines Apple II passen.

Ist der zweite Turmbau zu Babel nun fertig?

Noch recht neu sind die nichtprozeduralen Sprachen, bei denen der Programmierer nur noch das zu lösende Problem, jedoch nicht mehr den Lösungsweg anzugeben hat. Natürlich muß man für eine solche Sprache den Aufgabenbereich geeignet einengen; die meisten nichtprozeduralen Sprachen entstammen der Welt der Datenbanken. Als entfernte

Preiswerte Vorführgeräte und Restbestände

| | |
|------------------------------|---------|
| HP 2225 BB ThinkJet | DM |
| mit IL Interface | 890,- |
| HP 7470A Plotter DIN A4 | |
| mit IL Interface | 1.450,- |
| HP 82161AD Kassettenlaufwerk | |
| mit IL Interface | 890,- |

Neue



Produkte:

Leise Tintenstrahldrucker

| | |
|--|---------|
| HP 3630A PaintJet | |
| (330 Farben) | 2.998,- |
| HP 2276A DeskJet | |
| (bis zu 240 Zeichen pro Sekunde mit Laserqualität) | 2.128,- |

GES-COMPUTER

GESELLSCHAFT FÜR EDV UND SOFTWARE mbH
 Hartmann-Ibach-Straße 63 Steinheimer Straße 22
 6000 Frankfurt 60 6450 Hanau
 Tel.: (0 69) 46 20 41 Tel.: 0 61 81) 2 48 26
 Teletex: 6 999 356 = GES Fax: (0 61 81) 2 81 79
 Fax: (0 69) 46 20 54

Verwandte der nichtprozeduralen Sprachen kann man die dBASE-Befehlssprache ansehen: Bei einer dBASE-Sortieranweisung wie

`SORT ON NAME TO SORTDATEI`

braucht sich der Programmierer nicht um das zu verwendende Sortierverfahren zu kümmern (und damit auch nichts darüber zu wissen). Damit haben die nichtprozeduralen Sprachen die bisher letzte Abstraktionsstufe unter den Programmiersprachen und ein Höchstmaß an „Menschenfreundlichkeit“ erreicht.

Damit ist aber das babylonische Gewirr auf dem Gebiet der Programmiersprachen noch lange nicht abgeschlossen. So

gibt es hochspezialisierte Sprachen zur Anwendung in bestimmten Bereichen wie Kristallographie, Linguistik, Entwurf von integrierten Schaltungen, Graphik, um nur ein paar Beispiele zu nennen. Für Zwecke der künstlichen Intelligenz verwendet man die sogenannten „logischen“ Programmiersprachen, mit denen man besonders gut logische Ableitungen durchführen kann. Hier dominierte jahrelang der „Oldtimer“ LISP (LIST Processing Language) aus dem Jahr 1959; inzwischen ist ihm in den Sprachen Prolog (eine französische Entwicklung unter Alain Colmerauer) und Smalltalk Konkurrenz erwachsen.

Die Entwicklung auf dem Gebiet der Computersprachen setzt sich unge-

bremst fort; der Trend geht ziemlich von den ganz universellen Sprachen weg zu den eher abgemagerten Sprachen mit speziellem Einsatzprofil, und da es solcher Einsatzbereiche viele gibt, wird auch in Zukunft kräftig am zweiten Turm zu Babel gemauert werden.

Eine Randbemerkung zum Schluß: Während sonst auf dem Gebiet der Datenverarbeitung die wesentlichen Innovationen zweifellos aus den USA kommen, reden die Europäer, wie die Arbeiten von Wirth, Dijkstra, Hoare, Ichia, Colmerauer und vielen anderen zeigen, im Bereich der Programmiersprachen ein gewichtiges Wörtchen mit – vielleicht eine Folge der Vielsprachigkeit der Alten Welt.

Vergleich von Navigationsmodulen

für verschiedene Taschencomputer

Module für Astronomie und Navigation

In der letzten Zeit sind mehrere solche Module neu erschienen, so daß ein Überblick über die wichtigsten Module angebracht scheint. Zunächst ein tabellarischer Überblick über die angebotenen Funktionen in Tabelle 1.

Navigation Pac von Hewlett Packard für den HP41

Die Programme dieses Moduls wurden offensichtlich von Berufs-Nautikern und Berufs-Programmierern geschrieben. Infolgedessen haben wir hier ein in jeder Hinsicht mustergültiges Programmsystem: komplett, leicht bedienbar, mit vollständigen numerischen Beispielen zu allen vorhandenen Programmen, sehr gutem Design und mit richtiger nautisch-astronomischer Terminologie ohne Kauderwelsch. Viele interne Unterprogramme sind beschrieben, so daß diese in eigenen Programmen des Benutzers verwendet werden können. Außerdem kann man den Inhalt des Moduls ausdrucken und sich also von ihrem Aufbau und von ihrer Richtigkeit überzeugen, man kann Änderungen und Ergänzungen anbringen usw. Die zu Grunde liegende Theorie der Ephemeridenrechnung ist in einer angegebenen wissenschaftlichen Arbeit veröffentlicht. Alle für eigene astronomische Programme erforderlichen Werte und Koordinationen (Grt, δ , β , JD, GST, Δh , Az, Semidiameter SD oder Horizontal Parallax HP) sind teils als Unterprogramme teils als Datenwerte abrufbar.

Dieses ist das älteste der vorhandenen Module. Für eine Neuauflage würden wir uns wünschen:

- eine CX-Version mit der Option (Flag) der Übernahme von der Zeit und Datum für real-time Benutzung;
- die Aufnahme des Merkurs;

- Anweisungen für die Aufnahme weiterer Objekte nach Wunsch;
- Benutzung des X-Memory, z.B. für Wegmarken;
- vollständiges Koppelprogramm mit Mw, δ , BSW;
- Mitteln der Eingabewerte, Ausgleichen der Ausgabewerte bei mehr als 2 Standlinien nach Morrison zu einer Most Probable Position;
- statt der GB auf Wunsch die Einzelbeschickungen gemäß mb und $^{\circ}C$;
- für alle Himmelskörper deren Abstand von der Erde als UP;
- eine bessere Ausnutzung des Druckers; Erweiterung der Bedienungsanleitung mit Hinweisen auf die Möglichkeiten, die das Modul bietet.

Schenk-Modul von Delius & Klasing für den HP41

Der Hauptvorteil des Moduls ist es, daß durch eine komprimierte Programmierung – z.B. Weglassen der Sternnamen – Platz geschaffen wurde für eine ganze Reihe von zusätzlichen Navigations-Programmen (s. Tabelle). Der Hauptanteil des Moduls ist, daß es „Private“ ist und daß keine Unterprogramme oder Daten für eigene Benutzerprogramme angeboten werden. Durch diese unsinnige Geheimniskrämerie wird der Hauptvorteil eines Moduls, nämlich ein elektronisches Nautisches Jahrbuch zur Verfügung zu haben, wieder zunichte gemacht. Beim Einschalten des Rechners führt das Modul einen Autostart durch, Flags werden gesetzt und 29 Tastenfunktionen werden mit einer anderen Bedeutung belegt. Statt wie üblich Δh und Az zu berechnen, gibt Schenk den Leitpunkt und die Richtung der Standlinie an. Das Eintragen dieser Größen in die Seekarte kostet eher mehr Zeit als weniger; schlimmer ist aber, daß man nun

(Prüfungsaufgaben!) in ungewohnter Weise arbeiten muß. Winkel kleiner 15° werden nicht eingelesen oder ausgegeben, auch dort nicht, wo man sie üblicherweise braucht, z.B. bei der Kompaßkontrolle. Der eingegebene Rechenort wird durch viele Programme geändert und muß daher immer wieder neu eingegeben werden. Das Programm ASTRO ersetzt ihn durch den Leitpunkt, was schlicht falsch ist. Die Identifizierung eines unbekanntes Sterns erkennt Planeten nicht. Die Mitteilung der Eingabewerte ist löblicherweise vorhanden, wird aber als „Güte der Sextantmessung“ mißverstanden. Die nautische Terminologie – genormt (DIN 13312) oder nicht genormt – wird von Schenk souverän mißachtet. Entsprechend prangen auf jeder Seite oft mehrere Stilblüten. Eine Analyse des Moduls zeigte folgende wichtige, aber in der Bedienungsanleitung nicht angegebene Registerbelegungen:

01 Δh , 02 Az, 03 MJD = JD-2415020.0, 13 δ , 11 Rektaszension.

Schenk-Modul für den Sharp PC 1600

Gegenüber dem Modul für den HP41 wurde dieses Modul nochmal kräftig erweitert (s. Tabelle), insbesondere auch durch die Möglichkeit, einen Luftdruckfühler anzuschließen und dadurch den Rechner in einen Barographen zu verwandeln. Die (begrenzten) Graphikmöglichkeiten des Sharp werden auch für eine Menüsteuerung und für das Anzeigen von Standlinien benutzt. Druckerfunktionen sind keine vorgesehen. Leider hat auch dieses Modul die oben skizzierten Mängel: Leseschutz, keine Unterprogramme, ständige Ortsänderung, schwache Fehlerbehandlung, schlechte Bedienungsanleitung, Stilblüten, keine Sternnamen.

GRUNDLAGEN

| | Hewlett-Packard | Schenk I | Schenk II | Admiral von Cassens & Plath | Navset 710 Zeitnitz Software |
|---|---|--|--|--|---|
| Allgemeines | | | | | |
| Rechner | HP-41 | HP-41 | Sharp PC 1600 | Sharp 1245, 1247, 1248 1261, 1262, 1350, 2500 | HP-71 |
| Erscheinungsjahr | 1980 | 1985 | 1987 | 1985 und später | 1988 |
| Preis | DM 99,90 | DM 445,00 | DM 445,00 | DM 524,-/411,- (Kassette) | DM 800,00 |
| Bedienungsanleitung | sehr gut, knapp | schwach | schwach | gut, knapp | schlecht übersetzt, Fehler |
| Leseschutz | frei | private | private | private | private |
| Eingabe | Prompting | Prompting + Startwerte | Menus, Prompting | Prompting + Startwerte | Menus, Prompting |
| Terr. Navigation | | | | | |
| Abstand, Höhenwinkel | nein | ja | ja | ja | nein |
| Stromaufgaben | nein | teilweise | ja | teilweise | nein |
| Koppeln, Loggen, Besteckrechnung | mehrere UP für Best. Rechg. nach Großkreis, Loxodrome | ja Liste von Wegmarken | ja, Wegmarken | ja, keine Wegmarken | teilweise |
| Peilungen | nein | teilweise | ja | teilweise | wenig |
| Pothenot | nein | nein | ja | nein | nein |
| Deviationstabelle | nein | wird vorher eingegeben | wird vorher eingegeben | nein | teilweise |
| Gezeitenrechnung | nein | teilweise | ja | nein | nein |
| Großkreisrechnung | ja | teilweise | teilweise | nein | teilweise |
| Astronomie | | | | | |
| Mittelung der Beobachtungswerte | nein | ja | ja | ja | nein |
| Kulmination | nein | Zeit ohne Schiffsgeschwindigkeit keine Meridianbreite | Zeit ohne Schiffsgeschwindigkeit keine Breite | Zeit ohne Schiffsgeschwindigkeit keine Breite | ohne Schiffsgeschwindigkeit |
| Sternidentifikation | LHA, δ aus Az, h | ja | ja | ja | ja |
| Vorausberechnung | | | | | |
| Sextanteinstellung | ja, umständlich | ja | ja | ja | ja |
| Elektronischer Almanach | immerwährend ab 1582 | 1975 bis 2074 | ab 1.1.1975 | immerwährend | 1951 bis 2050 |
| Standlinien | $\Delta h, Az$ Sternname | Leitpunkt u. Richtung Sternnummer | Leitpunkt u. Richtung Sternnummer | $\Delta h, Az$ Sternnummer | Richtung und $\Delta \lambda$ Sternnamen |
| Position/Fix | nein | aus jeweils zwei Gestirnsbeobachtungen | aus jeweils zwei Gestirnsbeobachtungen | aus beliebig vielen Standlinien | aus jeweils 2 Standlinien, auch nicht-astro |
| Versegelung | nein | ja | ja | ja | ja |
| Ausgleichen der Ergebn. | nein | nein | nein | als Schwerpunkt der Fehlerfigur | nein |
| Grt, δ , GST, SD/HP/Abstand abrufbar | Unterprogramme für Grt, δ , nur RA, δ in Registern GST; SD in Register | nein | Grt, δ -Werte abrufbar, keine Unterprogramme | keine Unterprogramme | nein keine Unterprogramme |
| Theorie angegeben | ja | nein | nein | nein | nein |
| Genauigkeit angegeben | ja | nein | nein | nein | nein |

Versuche ergaben folgende „geheime“ Variabelnbelegung:

SM = Δh , AZ = Az, GR = Grt, DE = δ
AR = GST, A3 = MJD, LH = t, HG = h_b ,
L = λ_{AP} , B = φ_{AP} , AH = Ah.

Nach der Berechnung eines Himmelskörpers ist also außer einem der Werte SD/HP/GB/Abstand alles abrufbar. In dieser Hinsicht und auch bezüglich der angebotenen Funktionen ist dieses Modul also das vollständigste.

Admiral

Auch hier leider wieder Privatisierung und keine Angaben über Unterprogramme oder Variabelnbelegung. Alle anderen wichtigen Dinge sind vorhanden, insbesondere das Mitteln der Eingabewerte und das Auswerten eines Fehlerdreiecks. Drucker werden wenn vorhanden benutzt. Die Bedienungsanleitung ist knapp, aber gut; die Terminologie stimmt! Der Hauptvorteil dieser als Eprom oder als Kassette erhältlichen Programmsammlung ist der große Bereich von Rechnern, für den sie erhältlich ist, darunter auch solche unter DM 200,-.

Navset 710

Auch bei diesem Modul sind die Hauptnachteile die leidige Privatisierung und

fehlende Angaben über Unterprogramme und Variabelnbelegungen. Ansonsten scheint (mir lag nur die Bedienungsanleitung vor) das Modul gut durchdacht, flexibel und brauchbar. Die Bedienungsanleitung ist offensichtlich aus dem Englischen übersetzt, wobei dem Übersetzer leider eine ganze Reihe von Fehlern unterliefen, die vor allem auf mangelnde Kenntnis der nautischen und Computer-Terminologie zu beruhen scheinen. Die Schreibmaschinengeschriebene Anleitung ist zu umfangreich (117 Seiten), hat zahlreiche unnötige Wiederholungen, ist unübersichtlich und farblos. Die Rechneranzeigen sind englisch. Der Preis ist unrealistisch hoch. HP71 Besitzer seien auf PRISMA 3/86 hingewiesen, dem sie eine BASIC-Version des HP41 Moduls (kostenlos) entnehmen können.

Schlußbemerkung

Alle untersuchten Module können alle benötigten Himmelskörper (Sonne, Mond, 4 Planeten, ca. 60 Fixsterne) verarbeiten. Bei allen fehlt oder ist nur unvollständig vorhanden das Mitteln der Eingabewerte, das Berechnen einer wahrscheinlichsten Position unter Berücksichtigung der Azimutverteilung mit Hilfe der Ausgleichsrechnung (Morrison), die Verwendung der Einzelbeschickungen der Höhe statt GB auf Wunsch, das Berücksichtigen der

Schiffsgeschwindigkeit bei Meridianbeobachtungen, Luvgewinn/Regattarechnung und schließlich für die Amateurastronomen die Aufnahme zusätzlicher Objekte. In den nächsten PRISMA-Hefen wollen wir auf einige dieser Dinge genauer eingehen. Erstaunlich ist, daß das älteste Modul immer noch das beste ist, vor allem für Leute, die einen Computer nicht als Black-Box ansehen wollen.

Zum Schluß sei noch auf das Buch von Duffett-Smith, Astronomy with your Personal Computer, Cambridge University Press, UK, hingewiesen. Die Programme (keine Fixsterne) sind auf Diskette für IBM PCs und viele andere für £ 14,50 erhältlich. Sie sind ungeschützt; man kann hier also nach Herzenslust ändern, hinzufügen, streichen (in BASIC). Weitere Literatur: Knopp, Astronomische Navigation, Busse Verlag; H. Shufeldt & K. Newcomer, The Calculator Afloat, Naval Institut Press; G.d. Morrison, Most Probable Fix Position, Journal of the Institute of Navigation, Vol. 28, No 1, 1981, USA; W. Kumm, Ephemeridenberechnung, Ortung und Navigation 27 (1986), mehrere Teile.

Helmut Knopp
Karl-Pfitzerstraße 29
7032 Sindelfingen

Der „knuddelige“ Text-Editor

TEDDY

klein aber fein!

Einführung

TEDdy (Text-EDitor) ist als vollständiger Ersatz des HP Text-Editors gedacht. Ich benutze ihn hauptsächlich zum Erstellen von Assembler Source-Code. Deshalb ist keine Formatiermöglichkeit eingebaut, allerdings kann der Formatter des HP Text Editor Moduls aus dem Programm aufgerufen werden. Ansonsten kann TEDdy mehr als der HP Text-Editor und das Schreiben von Texten geht auch noch schneller von der Hand als mit jenem.

Man versuche nur einmal, eine einzelne Zeile in den Text einzufügen. Wird ein Video-Interface benutzt, so wird zum editieren nur eine Zeile benutzt (solange die Textlänge die darstellbare Zeilenlänge des Interfaces nicht überschreitet), sodaß man einen Ausschnitt des Textes auf dem Bildschirm anzeigen kann und während des editierens immer vor Augen hat.

TEDdy benutzt einige Befehle aus verschiedenen LEX-Files, die in der Programmbibliothek zusammen mit TEDdy angefordert werden können. Es sind dies im Einzelnen:

EDLEX
CUSTUTIL
STRINGLX
CCDUTIL
ADVPRLEX
oder das
JPC-EPROM

Das JPC-EPROM ist NICHT über die Programmbibliothek zu beziehen. Bezugsquellennachweis an anderer Stelle in diesem Heft, näheres über CCDUTIL ebenfalls in diesem und im nächsten Heft.

Starten des Programms

Es gibt drei verschiedene Möglichkeiten, das Programm zu starten:

1. [RUN] : Der Name des zu editierenden Files wird erfragt.
2. EDTEXT <NAME>[,<ZEILE>[, <SPALTE>]]
3. CALL EDITOR (<NAME>,<<ZEILE>.<SPALTE>>>)

Wenn vorhanden, dann steht der Cursor auf der optional angegebenen Position in den mit <NAME> spezifizierten Text-File. Ist kein File dieses Namens im Rechner, wird ein Text-File erzeugt. Ist ein File dieses Namens, aber mit anderem File-Typ im Speicher, wird eine Fehlermeldung ausgegeben und das Programm beendet.

Tastenzuweisungen

Ist ein KEY-File 'EDITKEYS' im Speicher, so werden diese Tastenzuweisungen aktiviert, wobei die eventuell bestehenden Zuweisungen unter dem Namen EDUKEYS (Editor User Keys) gesichert werden und die in EDIKEYS definierten zu den bestehenden dazu gemischt werden. Bei Beendigung des Programms werden die alten Zuweisungen wieder hergestellt.

Editor Funktionen

Die meisten Funktionen des Editor lassen sich direkt während der Texteingabe über Tastenkombination erreichen. Einige seltener benötigte Funktionen sind in ein Menü ausgelagert worden, um Probleme bei der Zuordnung zu vermeiden und die Tastatur nicht zu überladen (ich konnte mich schon wenige Tage, nachdem ich das Programm geschrieben hatte nicht mehr er-

innern, auf welcher Taste „MERGE“ lag, da die „M“-Taste schon mit „MOVE“ belegt war). Außerdem sind fast alle direkt erreichbaren Funktionen auch über das Menü erreichbar, falls mal irgendwelche Tastenzuordnungen vergessen wurden.

Um nicht die ganze Tastatur mit irgendwelchen Befehlen zu belegen, habe ich die [RUN]-Taste als zusätzliche Shift-Taste benutzt. Im folgenden bedeutet ein einer Taste vorangestelltes „f“, „g“ oder „^“, daß erst die „f“- oder „g“-shift-Taste bzw. die [RUN]-Taste gedrückt werden muß.

Direkt erreichbare Funktionen

Cursor-Tasten:

Mit den Cursor-tasten [<,>] kann der Cursor jeweils ein Zeichen nach links bzw. rechts bewegt werden.

Mit g[<,>] wird der Cursor auf den Zeilenanfang bzw. hinter das Zeilenende positioniert.

Mit ^[<,>] wird der Cursor auf die vorherige bzw. nächste Tabulatorposition bewegt. Das Setzen der Tabulatoren ist in der Beschreibung der Menüpunkte erklärt.

Die Cursor-Tasten [^,v] bringen die der momentanen Textzeile vorhergehende bzw. nachfolgende Textzeile in die Anzeige.

Mit ^[^,v] gelangt man in die erste bzw. hinter die letzte Textzeile.

[END LINE] Die angezeigte Zeile wird abgespeichert, wobei der alte Inhalt der Zeile im Text File überschrieben wird, falls der Ersetzungsmodus aktiv ist. (Siehe auch Menüpunkt SET OPTIONS : INSERT/REPLACE)

^[END LINE] Die angezeigte Zeile wird hinter die aktuelle Zeile im Text File eingefügt, egal ob der Ersetzungs- oder der Einführungsmodus aktiv ist.

f[END LINE] Der „UNDO“-Buffer wird in die Anzeige gebracht (! Super: Endlich eine „Rückgängig“-Funktion, Anm. d. Red.). Dort steht die Zeile, die zuletzt verloren gegangen ist. Das ist, falls eine Zeile geändert und abgespeichert wurde, der alte Inhalt der Zeile. Wurde der Inhalt einer Zeile geändert, aber nicht abgespeichert, so wird der geänderte Inhalt gerettet. Wird eine einzelne Zeile gelöscht, so wird diese in den „UNDO“-Buffer gerettet. Der Buffer hat eine Kapazität von einer Zeile.

g[END LINE] Die zuletzt eingegebene Zeile wird wieder in die Anzeige gebracht (CMD5, Command Stack).

[ATTN] Die Anzeige wird gelöscht.

f[M] Die aktuelle Zeile wird aus dem Text File gelöscht (DELETE)

[OFF] Der Rechner wird ausgeschaltet, ohne das Programm zu verlassen.

^[/] Die angezeigte Zeile wird bei der Cursorposition geteilt.

^[+] Die auf die momentane Zeile folgende wird an die angezeigte angehängt.

^[−] Die momentane Zeile wird an die ihr vorhergehende angehängt.

- ^[[] Die angezeigte Zeile wird mit der ihr vorhergehenden ausgetauscht.
- ^[)] Die angezeigte Zeile wird mit der ihr nachfolgenden ausgetauscht.
- ^[V] Die aktuelle Zeilennummer und die höchste Zeilennummer werden angezeigt (VIEW).
- ^[F] Anspringen einer beliebigen Zeile im Text File (FETCH).
- ^[S] Suchen nach einer Zeichenfolge (SEARCH).
Es werden die zu suchende Zeichenfolge und der Bereich des Textes, in dem gesucht werden soll, abgefragt. Wird die Zeichenfolge, nach der gesucht werden soll, mit [END LINE] abgeschlossen, so wird der Cursor auf die erste Stelle nach der aktuellen Zeile gesetzt, die mit der angegebenen Zeichenfolge übereinstimmt. Wird die Eingabe mit f[END LINE] abgeschlossen, so wird die Textstelle angezeigt, und eine Quittierung erwartet (Yes, No, Quit). Im Falle von [Yes] siehe oben. Falls [NO] gedrückt wurde, wird weiter gesucht. Wurde [Quit] gedrückt, erscheint wieder die alte Zeile in der Anzeige.
- ^[R] Ersetzen aller bzw. ausgewählter Vorkommnisse einer Zeichenfolge gegen eine andere (REPLACE).
Es werden die zu ersetzende und die Ersatzzeichenfolge und der Textbereich abgefragt. Wird eine der beiden Zeichenfolgen mit f[END LINE] abgeschlossen, so wird bei jedem Auftreten der zu ersetzenden Zeichenfolge gefragt, ob sie ausgetauscht werden soll (siehe SEARCH). Ansonsten wird jedes Vorkommnis ausgetauscht.
- ^[C] Kopieren eines Textblockes (COPY).
Es werden die erste und die letzte zu kopierende Zeile abgefragt und vor die aktuelle Zeile kopiert.
- ^[M] Verschieben eines Textblockes (MOVE).
Es werden die erste und die letzte zu verschiebende Zeile abgefragt, vor die aktuelle Zeile kopiert und an der alten Position gelöscht.
- ^[D] Löschen eines Textblockes (DELETE).
Es werden die erste und die letzte zu löschende Zeile abgefragt.
- ^[space] Eine Leerzeile wird eingefügt (INSERT).
- ^[RUN] Macht nichts.
- g[ON] HAUPTMENÜ

Menüs

In den Menüs wird immer nur ein Befehl angezeigt. Soll dieser ausgewählt werden, so ist [END LINE] zu drücken. Mit den Cursor Tasten [^,v] können die anderen Menüpunkte angezeigt und dann ausgewählt werden. Mit g[^,v] gelangt man zu dem ersten bzw. letzten Eintrag. Die Menüs sind durchlaufend, das heißt auf den letzten Eintrag folgt der erste und vor dem ersten ist der letzte.

Hauptmenü

Vom Hauptmenü aus kann nur eine Funktion direkt aufgerufen werden:

EXIT

Alle anderen Menüpunkte sind Untermenüs.

EDIT BLOCK. In diesem Untermenü sind alle Funktionen untergebracht, die einen in der Regel mehrere Zeilen umfassenden Textblock beeinflussen.

COPY, MOVE, DELETE, FETCH, SEARCH und REPLACE verhalten sich genau so wie die entsprechenden direkt erreichbaren Funktionen.

MERGE kopiert einen Teil eines anderen Text Files vor die aktuelle Zeile. Filename, Anfangs- und Endzeilennummer werden abgefragt.

EDIT LINE. In diesem Untermenü sind alle Funktionen untergebracht, die nur eine oder zwei Zeilen beeinflussen.

- DIVIDE LINE entspricht ^[[]
- APPEND NEXT LINE entspricht ^[+]
- APPEND TO PREV. LINE entspricht ^[-]
- SWAP PREV. LINE entspricht ^[[]
- SWAP NEXT LINE entspricht ^[)]
- INSERT DISP. LINE entspricht [END LINE]

OUTPUT. In diesem Untermenü sind die Druck- und Anzeigefunktionen untergebracht.

Mit PRINT oder LIST wird ausgewählt, ob die Ausgabe auf dem Drucker oder auf dem Display erfolgen soll.

Wird W/O LINE# ausgewählt, wird der Text ausgegeben, wie er in Textfile steht.

Wird W/LINE# ausgewählt, wird jeder Textzeile noch eine fortlaufende Zeilennummer vorangestellt.

Wird FORMAT ausgewählt, wird (wenn vorhanden) der FORMATTER der HP Text Editors aufgerufen.

In diesem Fall wird der gesamte Text ausgegeben, ansonsten erfragt das Programm die erste und letzte auszugebende Zeile.

SET OPTIONS In diesem Untermenü sind die Funktionen untergebracht, die die verschiedenen Optionen setzen. Diese Einstellungen werden in einem DATA-File EDITSTS abgespeichert und sind bei einem erneuten Aufruf des Editors wieder aktiv.

INSERT/REPLACE Schaltet zwischen (Zeilen-) Einfügungs- und Ersetzungsmodus um.
Wird durch Flag 0 angezeigt.

NO CR AT CUR ^v Wenn aktiv, dann bleibt der Cursor beim Verlassen der Zeile durch die Cursortasten in der momentanen Spaltenposition.
Wird durch Flag 1 angezeigt.

WRITE LINE AT CUR ^v Wenn aktiv, dann wird die angezeigte Textzeile auch dann in das Text File geschrieben, wenn die Zeile durch die Cursortasten verlassen wurde. Ansonsten wird der eventuell geänderte Text nur nach bestätigen der [END LINE]-Taste abgespeichert.
Wird durch Flag 2 angezeigt.

| | |
|--|--|
| <p>TRIM LINE Wenn aktiv, dann werden Leerzeichen am Ende der Zeile nicht abgespeichert. Wird durch Flag 3 angezeigt.</p> | <p>SET TAB(S) Hier können die vier Tabulatoren, die TEDdy zur Verfügung stellt, in beliebiger Reihenfolge gesetzt werden. Die aktuellen Werte werden als Vorgabewerte angezeigt.</p> |
| <p>AUTO IDENT Wenn aktiv, dann wird der Cursor nach betätigen von [END LINE] in der neuen Zeile in der Spalte positioniert, in der der vorherigen Zeile der Text anfing. Wird durch Flag 4 angezeigt.</p> | <p>STATUS Hier sind die Statusanzeige Funktionen untergebracht. Die Informationen werden angezeigt, solange eine Taste gedrückt wird.</p> |
| <p>SET FORMATTER An den FORMATTER des HP Editors müssen als Parameter die Anzahl der zu druckenden Kopien und der „fixed space character“ übergeben werden. Diese werden hier eingegeben.</p> | <p>LINE# Es werden die momentane und die höchste Zeilennummer angezeigt.</p> <p>FILE/MEMORY INFO Es werden der Filename, die augenblickliche Größe des Files in Bytes und der noch verbleibende Speicherplatz im Hauptspeicher bzw. in dem Port, in dem sich das Textfile befindet, angezeigt.</p> |
| | <p>EXIT Damit kommt man aus dem Programm raus.</p> |

TEDDY BASIC 8234

```

1 ! ***** T-Eddy *****                                HP71 Text Editor Version 1.2
2 ! Copyright (c) Matthias Rabe 87,88
3 ! Intended to be copied free for noncommercial use only !
10 CALL TEDDY @ SUB TEDDY
20 I=FILESZR('EDITSTS') @ IF I=-57 OR I=-1057 THEN 60
30 ON ERROR GOTO 50
40 ASSIGN #1 TO EDITSTS @ READ #1;F$
50 ASSIGN #1 TO * @ OFF ERROR
60 INPUT 'Edit File : ',F$;F$ @ CALL EDITOR(F$,0) @ END
70 SUB EDTEXT(F$,C$) @ C$=EDPARSE$(C$)
80 CALL EDITOR(F$,VAL(C$[1,6])+VAL(C$[7,12])/1000) @ END
100 SUB EDITOR(F$,A) @ F$=UPRC$(TRIM$(F$))
110 IF MEM<1000 THEN B=24 @ GOTO 140
120 B=FILESZR(F$) @ IF B>=0 THEN 160
130 IF B=-57 OR B=-1057 THEN CREATE TEXT F$ @ GOTO 160 ELSE IF B=-63 THEN B=59
140 BEEP 1400,.15 @ DISP MSG$(ABS(B)) @ END
150 BEEP 1400,.15 @ DISP ERRM$ @ GOTO 'Z'
160 DIM Z$[13]
170 Z$=PEEK$( '2F6D9',4)&PEEK$( '2F6E9',2)&PEEK$( '2F441',1)&PEEK$( '2F958',2)&PEEK$( '2F946',4)
180 ON ERROR GOTO 150
190 POKE '2F6D9','5' @ OPTION BASE 0 @ STD
200 DIM L$(3)[96],S$[11],C$[4] @ C$=CHR$(27)&'A'&CHR$(27)&'J'
210 INTEGER T(3),R,C,S,L,K,F @ I=FILESZR('EDITSTS')
220 IF I=-57 OR I=-1057 THEN CREATE DATA EDITSTS,0,8 @ L$(1)='00' @ GOTO 250
230 ASSIGN #1 TO EDITSTS @ READ #1;L$(0),I,T,L$(1),S$,F
240 IF L$(0)=F$ AND NOT A THEN A=I
250 POKE '2F6E9',L$(1)[1,2] @ I=FLAG(-15,FLAG(7))
260 POKE '2F441','F' @ DELAY 0,0 @ ASSIGN #1 TO F$ @ PWIDTH INF
270 C=FP(A)*1000 @ R=A-1
280 ON ERROR GOTO 290 @ PURGE EDUKEYS @ COPY KEYS TO EDUKEYS
290 ON ERROR GOTO 300 @ MERGE EDITKEYS
300 ON ERROR GOSUB 'ERR'
1000 'MLOOP':
1010 S=FILESZR(F$) @ R=MIN(MAX(0,R),S) @ L=0
1020 IF R>=S THEN L$(0)='' @ DISP C$&'[Eof]' @ GOTO 1040

```

```

1030 READ #1,R;L$(0) @ IF FLAG(0) THEN DISP C&L$(0) @ SCROLL 1 @ L$(0)=''
1040 'GETLINE':
1050 IF FLAG(3) THEN L$(L)=RTRIM$(L$(L))
1060 IF FLAG(1) OR FLAG(4) OR FLAG(5,0) THEN L$(L)=L$(L)&RPT$( ' ',MAX(0,C-1-LEN(L$(L)))
1070 IF C<0 THEN PUT '#105'
1080 C=MAX(1,MIN(LEN(L$(L))[1,95])+1,ABS(C))) @ I=MAX(1,MIN(LEN(L$(L))-21,C-10))
1090 INLINE C&L$(L),I,C,'#38#50#51#162#163#46#155#43#99#94#91#150',K,C
1100 IF FLAG(3) THEN L$(1)=RTRIM$(DISP$) ELSE L$(1)=DISP$
1110 SELECT K
1120 CASE 7 TO 12 @ C=ABS(C) @ GOSUB CUT$( 'MENU CL O U DL CS', ' ',K-7)
1130 CASE 1 @ GOSUB 'W' @ GOSUB 'W0'
1140 CASE 6 @ GOSUB 'CTL'
1150 CASE ELSE @ GOSUB 'W1' @ GOSUB CUT$( 'PL NL FL LL', ' ',K-2)
1160 END SELECT
1170 GOTO 'MLOOP'
1180 'W0': IF TRIM$(L$(1))#'' THEN L$(3)=L$(1)
1190 C=SPAN(L$(1),' ')*FLAG(4) @ R=R+1 @ RETURN
1200 'CTL':
1210 K=KEYCTL( '#46#47#48#32#35#56#42#36#37#17#31#18#16#4#14#38#99#49' )
1220 IF K>4 THEN C=ABS(C)
1230 GOTO CUT$( ' N TP TN VL M MP MM SP SN D C FE S R DIV IN O IL', ' ',K)
1240 'N':
1250 POP @ L=1 @ GOTO 'GETLINE'
1260 'MENU': DISP C&'***** Menu *****'
1270 K=FNM(0) @ ON K+1 GOTO 1370,1280,1290,1300,1310,1410,'Q'
1280 K=FNM(1) @ ON K+1 GOTO 1370,'C','M','D','FE','S','R','MF'
1290 K=FNM(2) @ ON K+1 GOTO 1370,'DIV','MP','MM','SP','SN','IN'
1300 K=FNM(3) @ ON K+1 GOTO 1370,'P','L'
1310 K=FNM(4) @ SELECT K
1320 CASE 0
1330 CASE 6 @ DISP C$; @ INPUT '# of Copies : ',STR$(MAX(1,F));A$ @ F=VAL(A$) @ DESTROY A$
1340 DISP C$; @ INPUT 'Fixed space Char : ',S$;S$
1350 CASE 7 @ A$=STR$(T(0))&','&STR$(T(1))&','&STR$(T(2))&','&STR$(T(3))
1360 DISP C$; @ INPUT 'TAB's : ',A$;T(0),T(1),T(2),T(3) @ DESTROY A$
1370 K=FNS(0,1) @ K=FNS(2,3) @ K=FNS(1,2) @ K=FNS(0,1) @ K=FNS(2,3) @ K=FNS(1,2)
1380 CASE ELSE @ K=FLAG(K-1,1-FLAG(K-1))
1390 END SELECT
1400 RETURN
1410 K=FNM(5) @ ON K+1 GOTO 1370,'VL',1420
1420 L$(0)=FILE$(F$)
1430 DISP C&CHR$(27)&'<'&L$(0)[1,9]&L$(0)[17,22];
1440 IF L$(0)[39]='' THEN DISP MEM ELSE DISP MEM(VAL(L$(0)[39]))
1450 IF KEYDOWN THEN 1450 ELSE WAIT .5 @ RETURN
1460 'W':
1470 IF R>=S THEN 1500 ELSE IF FLAG(0) THEN 1510
1480 REPLACE #1,R;L$(1) @ IF L$(0)#L$(1) AND L$(0)#'' THEN L$(2)=L$(0)
1490 RETURN
1500 RESTORE #1,S @ PRINT #1;DISP$ @ RETURN
1510 INSERT #1,R;L$(1) @ RETURN
1520 'CL': L$(1)='' @ L=1 @ C=1 @ POP @ GOTO 'GETLINE'
1530 'W1': IF R>=S AND TRIM$(L$(1))='' THEN 1550
1540 IF FLAG(2) THEN GOSUB 'W' ELSE IF L$(0)#L$(1) AND L$(1)#'' THEN L$(2)=L$(1)
1550 C=ABS(C*FLAG(1)) @ RETURN
1560 'PL': R=R-1 @ RETURN
1570 'NL': R=R+1 @ RETURN
1580 'FL': R=0 @ RETURN

```

```

1590 'LL': R=INF @ RETURN
1600 'DL': C=1 @ IF R<S THEN 1620 ELSE IF L$(1)='#' THEN L$(2)=L$(1)
1610 RETURN
1620 DELETE #1,R @ IF L$(0)='#' THEN L$(2)=L$(0)
1630 RETURN
1640 'IL': IF R<S THEN INSERT #1,R;'
1650 C=SPAN(L$(1),' ')*FLAG(4) @ RETURN
1660 'VL': DISP C#&'Line :';R+1; @ IF R=S THEN DISP '[Eof]';
1670 DISP ' of';S
1680 IF KEYDOWN THEN 1680 ELSE WAIT .5 @ L=1 @ POP @ GOTO 'GETLINE'
1690 'FE': DISP C#; @ INPUT 'Fetch Line : ',STR$(R+1);R @ R=R-1 @ C=1 @ RETURN
1700 'C':
1710 DISP C#; @ INPUT 'Copy : ',STR$(R+1)&','&STR$(R+1);A,B @ GOSUB 'AB'
1720 K=R+B-A+1 @ I=B @ IF R=S THEN 1760
1730 FOR C=0 TO B-MAX(R,A) @ READ #1,I;A$ @ INSERT #1,R;A$ @ NEXT C
1740 B=MIN(R-1,B) @ FOR I=A TO B @ READ #1,I;A$ @ INSERT #1,R;A$ @ R=R+1
1750 NEXT I @ GOTO 1770
1760 FOR I=A TO B @ READ #1,I;A$ @ RESTORE #1,FILESZR(F#) @ PRINT #1;A$ @ NEXT I
1770 R=K @ C=1 @ GOTO 'E'
1780 'AB':
1790 A=INT(MAX(A-1,0)) @ B=INT(MIN(B,S)-1)
1800 IF A>B THEN POP @ RETURN ELSE DISP C#&MSG$(240010) @ RETURN
1810 'M':
1820 DISP C#; @ INPUT 'Move : ',STR$(R+1)&','&STR$(R+1);A,B @ GOSUB 'AB'
1830 IF B>R-2 AND A<=R THEN RETURN
1840 I=A @ K=R @ IF A>R THEN R=R+B-A+1
1850 IF K<A THEN I=K @ K=B @ B=A @ A=I
1860 IF B-A>K-B THEN 1900 ELSE B=B-A+1 @ IF K=S THEN 1880
1870 FOR C=1 TO B @ READ #1,I;A$ @ INSERT #1,K;A$ @ DELETE #1,I @ GOTO 1890
1880 FOR C=1 TO B @ READ #1,I;A$ @ RESTORE #1,S @ PRINT #1;A$ @ DELETE #1,I
1890 NEXT C @ C=1 @ GOTO 'E'
1900 S=A+K-B @ I=K
1910 FOR C=A TO S @ READ #1,I;A$ @ INSERT #1,A;A$ @ DELETE #1,I+1 @ GOTO 1890
1920 'D':
1930 DISP C#; @ INPUT 'Delete : ',STR$(R+1)&','&STR$(R+1);A,B @ GOSUB 'AB'
1940 C=1 @ R=A @ FOR I=A TO B @ DELETE #1,A @ NEXT I @ RETURN
1950 'R':
1960 INLINE C#&CHR$(27)&'<Replace : ',1,1,'#38#94',I @ I=FLAG(7,I-1)
1970 L$(0)=DISP$ @ IF L$(0)='' THEN RETURN
1980 INLINE C#&CHR$(27)&'<by : ',1,1,'#38#94',I @ L$(1)=DISP$
1990 IF L$(1)=L$(0) THEN RETURN ELSE IF I-1 THEN SFLAG 7
2000 DISP C#; @ INPUT 'Line : ',STR$(R+1)&','&STR$(S);A,B @ GOSUB 'AB' @ I=A
2010 S=LEN(L$(0)) @ K=LEN(L$(1)) @ C=-K @ IF FLAG(7) THEN 2050
2020 I=SEARCH(L$(0),C+K,I,B,1)
2030 IF NOT I THEN RETURN ELSE C=FP(I)*1000 @ I=INT(I)
2040 GOSUB 2070 @ GOTO 2020
2050 GOSUB 2120 @ ON A GOTO 2060,2210,2210
2060 GOSUB 2070 @ C=-K @ SFLAG 7 @ GOTO 2050
2070 R=I @ READ #1,I;A$ @ REPLACE #1,I;A$[1,C-1]&L$(1)&A$[C+S] @ RETURN
2080 'S': INLINE C#&CHR$(27)&'<Search : ',1,1,'#38#94',I
2090 I=FLAG(7,I-1) @ L$(0)=DISP$
2100 DISP C#; @ INPUT 'Line : ',STR$(R+1)&','&STR$(S);A,B @ GOSUB 'AB'
2110 K=LEN(L$(0)) @ I=A @ IF I=R THEN C=C+1-K ELSE C=-K
2120 I=SEARCH(L$(0),C+K,I,B,1) @ C=1
2130 IF NOT I THEN DISP C#&MSG$(232) @ A=2 @ RETURN
2140 C=FP(I)*1000 @ I=INT(I)
2150 IF NOT FLAG(7) THEN R=I @ RETURN
2160 READ #1,I;A$ @ A=MAX(MIN(LEN(A$)-95,C-40),1) @ A$=A$[A,A+95] @ C=C-A+1

```

```

2170 DISP C$&EDPARSE$(STR$(I+1))[2,6]&': '
2180 WINDOW 7 @ DISP C$&A$ @ SCROLL C @ WINDOW 1
2190 A=KEYCTL('#6#34#43#1#99') @ ON A GOTO 2200,2120,2210,2210,2220
2200 CFLAG 7 @ GOTO 2150
2210 C=1 @ GOTO 'E'
2220 GOSUB 'O' @ GOTO 2170
2230 'L':
2240 L$(0)=PEEK$('ZF794',7) @ POKE 'ZF794',PEEK$('ZF78D',7) @ GOSUB 'P'
2250 POKE 'ZF794',L$(0) @ RETURN
2260 'P': C=1
2270 K=FNM(6) @ ON K+1 GOTO 2250,2280,2290,2370
2280 CFLAG 7 @ GOTO 2300
2290 SFLAG 7
2300 DISP C$; @ INPUT 'Line : ',STR$(R+1)&','&STR$(S);A,B @ GOSUB 'AB'
2310 DISP C$&MSG$(240010) @ DISP C$[1,2];
2320 FOR I=A TO B @ READ #1,I;A$ @ IF NOT KEYCTL('#43',1) THEN 2350
2330 DISP C$[3]&'Quit ? [Y/N]' @ DISP C$[1,2];
2340 IF KEYCTL THEN 2360 ELSE DISP C$[3]&MSG$(240010) @ DISP C$[1,2];
2350 PRINT (EDPARSE$(STR$(I+1))[6-EXPONENT(S),6]&': ') [1,9*FLAG(7)]&A$ @ NEXT I
2360 CFLAG 7 @ DISP @ GOTO 'E'
2370 DISP C$&MSG$(240010) @ ASSIGN #1 TO * @ ON ERROR GOTO 2390
2380 CALL EDFORM(F$,MAX(1,F),S$) @ GOTO 2400
2390 DISP C$&'Formatter Not Found' @ BEEP 1400,.15
2400 ON ERROR GOSUB 'ERR' @ ASSIGN #1 TO F$ @ RETURN
2410 'O': OFF @ POKE 'ZF441','F' @ RETURN
2420 'DIV':
2430 IF R>=S THEN PRINT #1;L$(1)[1,C-1],L$(1)[C] @ GOTO 2460
2440 IF L$(0)#L$(1) AND L$(0)#'' THEN L$(2)=L$(0)
2450 REPLACE #1,R;L$(1)[C] @ INSERT #1,R;L$(1)[1,C-1]
2460 R=R+1 @ C=1 @ RETURN
2470 'MM':
2480 IF NOT R THEN RETURN
2490 R=R-1 @ I=R @ READ #1,R;A$ @ DELETE #1,R @ C=LEN(A$)+1
2500 REPLACE #1,R;A$&L$(1) @ GOTO 'E'
2510 'MP':
2520 IF R=S-1 THEN RETURN
2530 I=R+1 @ READ #1,I;A$ @ DELETE #1,R @ C=LEN(L$(1))+1
2540 REPLACE #1,R;L$(1)&A$ @ GOTO 'E'
2550 'MF':
2560 DISP C$; @ INPUT 'Merge File : ';L$(0)
2570 I=FILESZR(L$(0)) @ IF NOT POS(L$(0),':') AND (I=-57 OR I=-1057) THEN 2690
2580 DISP C$; @ INPUT 'Line : ';A,B
2590 A=INT(MAX(A-1,0)) @ B=INT(MIN(B,FILESZR(L$(0)))-1)
2600 IF A>B THEN RETURN ELSE DISP C$&MSG$(240010)
2610 ON ERROR GOTO 2680 @ ASSIGN #1 TO L$(0) @ ASSIGN #2 TO F$
2620 ON ERROR GOSUB 'ERR'
2630 IF R<S THEN 2660
2640 RESTORE #2,S @ FOR I=A TO B @ READ #1,I;A$ @ PRINT #2;A$ @ NEXT I
2650 R=INF @ GOTO 2670
2660 FOR I=A TO B @ READ #1,I;A$ @ INSERT #2,R+I-A;A$ @ NEXT I @ R=R+I-A
2670 ASSIGN #2 TO * @ ASSIGN #1 TO F$ @ GOTO 'E'
2680 IF ERRN=54 OR ERRN=1054 THEN 2670 ELSE I=ERRN
2690 BEEP 1400,.15 @ DISP C$&MSG$(I) @ GOTO 2670
2700 'SP': IF R=S OR R=0 THEN RETURN
2710 READ #1,R;L$(0) @ DELETE #1,R @ R=R-1 @ INSERT #1,R;L$(0) @ RETURN
2720 'SN': IF R>S-2 THEN RETURN
2730 I=R+1 @ READ #1,I;A$ @ DELETE #1,I @ INSERT #1,R;A$ @ R=I @ GOTO 'E'
2740 'IN': GOSUB 'W0'

```

```

2750 IF R>=S THEN R=S @ RESTORE #1,S @ PRINT #1;L$(1) @ RETURN
2760 INSERT #1,R;L$(1) @ RETURN
2770 'U': IF L#2 THEN L=2 ELSE L=0
2780 C=1 @ POP @ GOTO 'GETLINE'
2790 'TP': L=SGN(C+.1) @ C=ABS(C) @ IF C<=T(0) THEN C=1 @ GOTO 2860
2800 FOR K=3 TO 0 STEP -1 @ IF C>MIN(C,T(K)) THEN C=T(K) @ K=-1
2810 NEXT K @ GOTO 2860
2820 'TN': L=SGN(C+.1) @ C=ABS(C)
2830 IF C>=T(3) THEN C=MAX(T(3),LEN(L$(1))+1) @ GOTO 2860
2840 FOR K=0 TO 3 @ IF C<MAX(C,T(K)) THEN C=T(K) @ K=4
2850 NEXT K
2860 SFLAG 5 @ C=C*L @ L=1 @ GOTO 'GETLINE'
2870 'CS': IF L#3 THEN L=3 ELSE L=0
2880 C=1 @ POP @ GOTO 'GETLINE'
3000 DEF FNM(M) @ SELECT M
3010 CASE 0 @ DIM M$[51]
3020 M$='EDIT BLOCK,EDIT LINE,OUTPUT,SET OPTIONS,STATUS,EXIT'
3030 CASE 1 @ DIM M$[43]
3040 M$='COPY,MOVE,DELETE,FETCH,SEARCH,REPLACE,MERGE'
3050 CASE 2 @ DIM M$[98]
3060 M$='DIVIDE LINE,APPEND NEXT LINE,APPEND TO PREV. LINE,'
3070 M$=M$&'SWAP PREV. LINE,SWAP NEXT LINE,INSERT DISP. LINE'
3080 CASE 3 @ DIM M$[10]
3090 M$='PRINT,LIST'
3100 CASE 4 @ DIM M$[97]
3110 M$='INSERT/REPLACE,NO CR AT CUR ^v,WRITE LINE AT CUR ^v,TRIM LINE,'
3120 M$=M$&'AUTO INDENT,SET FORMATTER,SET TAB's'
3130 CASE 5 @ DIM M$[25]
3140 M$='LINE #,FILE / MEMORY INFO'
3150 CASE 6 @ DIM M$[28]
3160 M$='W/O LINE #,W/ LINE #,FORMAT'
3170 END SELECT @ M=0
3180 M=MOD(M,COUNT(M$','')+1) @ DISP C$&CUT$(M$','',M)
3190 SELECT KEYCTL('43#50#51#162#163#38#99')
3200 CASE 1 @ M=0 @ GOTO 3280
3210 CASE 2 @ M=M-1
3220 CASE 3 @ M=M+1
3230 CASE 4 @ M=0
3240 CASE 5 @ M=K-1
3250 CASE 6 @ M=M+1 @ GOTO 3280
3260 CASE 7 @ GOSUB '0'
3270 END SELECT @ GOTO 3180
3280 DESTROY M$ @ FNM=M @ END DEF
4000 DEF FNS(A,B) @ T(A)=MOD(MIN(T(A),97),97) @ T(B)=MOD(MIN(T(B),97),97)
4010 K=MIN(T(A),T(B)) @ T(B)=MAX(T(A),T(B)) @ T(A)=K @ FNS=K @ END DEF
9000 'ERR':
9010 IF NOT (ERRN=37 OR ERRN=1037) THEN BEEP 1400,.1 @ DISP C$&ERRM$ @ GOTO 'Q'
9020 ON ERROR GOSUB 'ERR' @ IF ERRL=1030 THEN 9040
9030 E=E+1 @ DIM A$[32*(E+1)] @ READ #1,I;A$ @ RETURN
9040 DISP C$&MSG$(65) @ BEEP 1400,.1
9050 I=R @ READ #1,R;A$ @ L$(0)=A#[1,96]
9060 INSERT #1,R;L$(0) @ REPLACE #1,R+1;A#[97]
9500 'E': DESTROY E,A$ @ RETURN
9900 'Q': ON ERROR GOTO 9910 @ PURGE KEYS @ RENAME EDUKEYS TO KEYS
9910 A=R+1+C/1000 @ I=FLAG(7,FLAG(-15)) @ L$(1)=PEEK$('2F6E9',2)
9920 DISP C$&'Done : '&F$
9930 ON ERROR GOTO 'Z' @ ASSIGN #1 TO EDITSTS @ PRINT #1;F$,A,T,L$(1),S$,F
9940 'Z': POKE '2F6D9',Z#[1,4] @ POKE '2F6E9',Z#[5,6] @ POKE '2F441',Z#[7,7]
9950 POKE '2F958',Z#[8,9] @ POKE '2F946',Z#[10,13] @ END SUB

```

Matthias Rabe
Muckenbrock 14
4803 Steinhagen

DISKCAT

Diskettenkatalog

Dieses SUB-Programm wird aus dem Hauptspeicher, wo es in einer Bibliothek abgelegt werden sollte, mit CALL CAT aufgerufen. Es leistet folgendes:

- **Ausdrucken** der (evtl. gekürzten) Disketten-Katalogeinträge mit den Ordnungsnummern ihres Platzes auf der Diskette
 - vollständig oder
 - in beliebig wählbaren Bereichen
- zu jedem File kann (über das Tastenfeld) eine beliebig lange Bemerkung geschrieben werden
- **Sichern** einzelner oder aller Files auf der Diskette
- **Löschen** einzelner Files von der Diskette mit der Kontrollfrage, ob der eben genannte File **wirklich** gelöscht werden soll
- **Schreiben** des so erarbeiteten Katalogs in Textfile
 - dazu Schreib - SUB-Programm PCAT siehe 5.

140 Disk.name,pruef.
 ,, ,0“ Ein Diskettenname kann eingegeben werden. 1 nach dem Komma bewirkt per Programm die Prüfung, ob dieser Name mit der Datenträgerkennung (vgl. Initialisierung) übereinstimmt. Meldungen: „Name stimmt“ oder „D.(iskette) hat ändern (keinen) Namen“

160 Bemerkungen 0 Falls Bemerkungen zu den einzelnen Files geschrieben werden sollen, hier 1 setzen

170 absp.(eichern) in File Soll der Katalog - ggf. mit Bemerkungen - in einen Textfile geschrieben werden, 1 setzen Falls kein File-Name eingegeben wurde, jetzt Textfilenamen eintragen

230 von bis 1,10 Es soll der Diskettenkatalog von der ersten bis zur zweiten Nummer (nacheinander) aufgerufen werden. Voreinstellung: die ersten 10, bei Fortsetzung (Zl. 550) wird die folgende Nummer gebracht und das gewählte Intervall beibehalten (statt 1 wird aber 10 gesetzt)

1) Geräteausstattung

- HP 71 B + HP IL
- Diskettenlaufwerk HP 9114 (auch für Magnetbandrekorder geeignet)
- ThinkJet HP 2225B (auch ohne Drucker teilweise nutzbar)

nützliche Ergänzungen:
 - vergrößerter Speicher, Bildschirm, Tastatur

300 (Filenummer und -name) angeben für Sichern oder Löschen (wenn nicht gedruckt wird)

320 auf Disk.(ette) sichern 0 Zustimmung geben durch 1

2) Allgemeines

Bei Entscheidungsfragen bedeutet 1= ja
 alle anderen Zahlen= nein
 Buchstaben und Zeichen setzen Fehler: Numeric Input

Die **Numerierung** der Files im Katalog hat nur eine **vorübergehende** Bedeutung. Jeder Lösch- und jeder Kopiervorgang ändert sie, zumindest teilweise. Für die Verwaltung der Files und den Überblick über den Katalog ist sie aber sehr hilfreich.

Manche Programmläufe belasten den Akku des Diskettenlaufwerkes stark. Überlastungen vermeiden, rechtzeitig laden!

350 wirkl.(ich auf) Disk.(ette) loeschen? 0 Diese warnende Rückfrage ist mit 1 zu belegen, wenn der eben angezeigte File auf der Diskette wirklich gelöscht werden soll

380 Bemerk(un)g.(en) Voreinst. Rest von Vorzeile Falls die Frage Zl. 160 bejaht wurde, können zu jedem File beliebig lange Bemerkungen geschrieben werden. Es wird die Filekatalognummer wiederholt und die Bemerkung bei einem passenden Zwischenraum zur nächsten Zeile gebrochen

550 fortsetzen 0 (Vgl.230) Mit 1 wird fortgesetzt

570 Summe drucken? 0 mit 1 wird die Anzahl der aufsummierten Bytes sowie die Zahl der erfaßten Files ausgegeben

3) Programmbeschreibung

Eingaben, Fragen des Programmes

| Zl. Frage | Voreinstellg. | Erläuterungen |
|---------------------------------------|---------------|---|
| 60 kurz-CAT? -6 od.-15 | 0 | Es wird gefragt, ob der Katalogeintrag (insbesondere zugunsten der Bemerkungen) gekürzt werden soll. Zugelassene Werte 0 bis 15, Empfohlen 6 oder -6 = Eintrag ohne Uhrzeit 15 (-15) Eintrag ohne Uhrzeit und ohne Datum |
| 90 Dr.(ucken) Sich.(ern) Loesch.(en)? | D | Durch Setzen von) D,S,P (=PURGE) (wahlweise Kleinbuchstaben) werden die verschiedenen Bereiche aktiviert. SEC(ure) ALL: Alle Files der Diskette werden gesichert |

600 COPY? 0 Eingabe 1 bewirkt das Kopieren des Textfiles N\$ auf die Diskette

4) LABELS

230 '!': Einspruchsmöglichkeit - solange das SUB-Programm CAT aktiv ist - mit CONT I, um neue Katalogbereiche zu wählen

600 'COPY': springt Kopiervorgang des Textfiles N\$ auf Diskette an

nicht zum Anspringen vorgesehen:

'ERR': Fehlerbehandlung und 'E': END SUB

5) Schreibprogramm PCAT Aufruf = CALL PCAT

Ab Zeile 1000 bis 1070 ist ein primitives SUB-Programm zum Schreiben des im Hauptspeicher befindlichen Textfiles N\$ mit den katalogisierten und ggf. mit Bemerkungen versehenen Files aufgeführt.

Die SUB-Programme E und N sind im allgemeinen bereits in der Bibliothek vorhanden und können dann hier gestrichen werden.

Nützliche Programmelemente

In dem Programm CAT finden sich Elemente, die sich zur Verwendung in anderen Programmen eignen:

- Durchlaufen eines Blockes in einer FOR-NEXT-Schleife und dann bei Bejahung der Frage „fortsetzen?“ bietet das Programm den anschließenden Block gleicher Größe an. Die Einsprungstelle mit dem Label „I“: ermöglicht dabei ein rasches Neuwählen eines willkürlich veränderten Bereiches. Zeilen 200 bis 240 1. Befehl und 550
- Abtrennen von Leerstellen am Ende eines Strings (hier von dem zunächst mit 8 Stellen aufgerufenen Filenamen) z.B. Zeile 270.
- Abteilen von Text an einem Zwischenraum, Zi. 400 bis 420.
- Erfassen der Bytezahl eines Files aus dem Katalogeintrag, Zi. 500.

```
DISKCAT BASIC 2741 05/11/88 10:30
```

```
10 SUB CAT @ ! DISKCAT DISKettenCATalog
15 ! Dr. Karlheinz Lutz (2696),Bayernstr. 39, D 8501 Rueckersdorf
20 ! D Drucken; S Sichern auf Diskette; P Loeschen von Diskette
30 ! SEC(ure) ALL: Sichern aller Files auf Diskette; wahlweise Kleinbuchstaben
40 ! 6: ohne Uhrzeit, 15: ohne Uhrzeit und ohne Datum
50 INTEGER A,C1,I,I0,I1,I2,I3,J,J1,J2,K,P,Q,Q1,Q9,S
60 INPUT "kurz-CAT? -6 od.-15 ", "0";I0 @ I0=ABS(I0) @ IF I0>15 THEN 60
70 DIM C$(40),C2$(40-I0),N$(8),L$(80),S$(2),T$(5) @ T$=":TAPE"
80 FOR Q=1 TO 40-I0 @ C2$=" "&C2$ @ NEXT Q
90 INPUT "Dr. Sich. Loesch. ? ", "D";E$ @ E$=UPRC$(E$) @ IF POS(E$,"D") THEN J4=1
100 IF POS(E$,"SEC") AND POS(E$,"ALL") THEN A=1
110 IF POS(E$,"S") THEN S=1
120 IF POS(E$,"P") THEN P=1
130 IF J4#1 THEN 200
140 INPUT "Disk.name,pruef. ",N$&" ", "0";N$,J @ N$=UPRC$(N$) @ IF J#1 THEN 150
142 ON ERROR GOTO 'ERR' @ DIM B9$(40),B8$(40) @ B9$=CAT$(1,":TAPE")
144 B8$=CAT$(1,":&N$) @ IF B9$=B8$ THEN DISP "Name stimmt " @ GOTO 148
146 DISP "D.hat andern (keinen) Namen"
148 DESTROY B8$,B9$ @ K$=KEY$ @ IF K$="" THEN 148
150 IF N$#" " THEN PRINT "Diskette: "; @ CALL E @ PRINT N$ @ PRINT @ CALL N
160 INPUT "Bemerkungen ? ", "0";J1 @ DIM B$(34+I0),B1$(90)
170 INPUT "absp.in File ", "0,"&N$;J2,N$ @ N$=UPRC$(N$) @ IF J2#1 THEN 200
180 ON ERROR GOSUB 'ERR' @ CREATE TEXT N$
190 ASSIGN #1 TO N$ @ IF J2=1 AND N$#" " THEN PRINT #1;"Disk.name: "&" "&N$
200 IF I1=0 THEN I1=1
210 IF I2=0 THEN I2=10
220 IF I3=0 THEN I3=9
230 'I': INPUT "von bis ",STR$(I1)&" ", "&STR$(I1+I3);I1,I2 @ I3=I2-I1
240 FOR I=I1 TO I2 @ C$=CAT$(I,T$) @ C$=C$[1,40-I0] @ Q=0 @ Q1=0
250 IF C$="" THEN 560
260 IF A#1 THEN 300
```

```

270 FOR I=1 TO 600 @ C$=CAT$(I,T$) @ N1=POS(C$," ") @ C9$=C$[1,N1-1]&T$
280 J9=0 @ IF C$[1,81]="" THEN 'E'
290 SECURE C9$ @ NEXT I @ GOTO 'E'
300 IF J4#1 OR S=1 OR P=1 THEN INPUT "",STR$(I)&," "&C$;I,C$
310 IF S#1 THEN 340
320 INPUT "auf Disk.sichern ? ","0";J @ IF J#1 THEN 340
330 N1=POS(C$," ") @ C9$=C$[1,N1-1]&T$ @ SECURE C9$
340 IF P#1 THEN 370
350 INPUT "wirkl.Disk.loesch. ? ","0";K @ IF K#1 THEN 370
360 IF P=1 AND K=1 THEN N1=POS(C$," ") @ C9$=C$[1,N1-1]&T$ @ PURGE C9$
370 DELAY 0,0 @ IF J1#1 THEN 430
380 DISP I;C$ @ LINPUT "Bemerkg. ",B1$;B1$
390 IF LEN(B1$)<35+I0 THEN B$=B1$ @ B1$=' ' @ GOTO 430
400 FOR Q9=0 TO 34+I0 @ Q=POS(B1$,' ',Q+1) @ IF Q<Q1 OR Q>34+I0 THEN 420
410 Q1=Q @ NEXT Q9
420 B$=B1$[1,Q1-1] @ B1$=B1$[Q1+1] @ J9=J9+1
430 S$="" @ IF I<100 THEN S$=" "
440 IF I<10 THEN S$=" "
450 IF J9>1 THEN C$=C2$ @ Q=0 @ Q1=0
460 L$=S$&STR$(I)&" "&C$&" "&B$
470 IF J4=1 THEN PRINT L$
480 IF J2=1 THEN PRINT #1;L$
490 J9=0 @ IF C$=C2$ THEN 510
500 C=C+VAL(C$[20,25]) @ C1=C1+1
510 IF J2#1 THEN 540
520 IF NOT ((Q OR Q1) AND B1$#') THEN 540
530 C$=C2$ @ Q=0 @ Q1=0 @ GOTO 370
540 DELAY 9,9 @ NEXT I
550 INPUT "fortsetzen ? ","0";J @ IF J=1 THEN I1=I2+1 @ I2=I1+I3 @ GOTO 220
560 IF J4#1 THEN 'COPY'
570 INPUT "Summe drucken? ","0";J @ IF J#1 THEN 'COPY'
580 PRINT @ L$=" Summe der Bytes = "&STR$(C)&" "&STR$(C1)&" Files" @ PRINT
L$
590 IF J2=1 THEN PRINT #1;L$
600 'COPY': INPUT "COPY ? ","0";J @ IF J=1 THEN COPY N$ TO T$
610 DISP "Summe der Bytes ";C;C1;"Files " @ GOTO 'E'
620 'ERR': IF ERRL=144 AND ERRN=58 THEN 146
630 IF ERRL=180 AND ERRN=59 THEN 190
640 DISP ERRL;ERRM$ @ RETURN
650 'E': END SUB
1000 SUB PCAT @ ON ERROR GOSUB 'ERR'
1010 INPUT "drucken aus TEXT-File ",N$;N$ @ DIM L$(80) @ ASSIGN #1 TO N$
1020 INPUT "ab Zeile, bis ",I1,I2
1030 FOR I=1 TO I2+1 @ READ #1;L$ @ IF I>I1 THEN PRINT L$
1040 NEXT I
1050 'ERR': IF ERRL=1020 AND ERRN=54 THEN 1070
1060 DISP ERRL;ERRM$
1070 END SUB
9998 SUB N @ PRINT CHR$(27);'&k0S';
9999 SUB E @ PRINT CHR$(27);'&k1S';

```

Dr. Karlheinz Lutz
Bayernstr. 39,
8501 Rückersdorf

LEXCHEC2

73 Zeilen, 1964 Bytes, HP-71, IL, PRINTER

... UND NOCH EIN PROGRAMM FÜR LEXFILES: LEXCHEC2

Als ich im Herbst 1987 in PRISMA das Basic-Programm LEXCHECK vorstellte, mit dem man ein Lexfile überprüfen kann auf ID-Nr. und die darin vorkommende Token, wußte ich noch nicht, daß es auch Lexfiles gibt, die **mehrere** Teile enthalten mit mehreren Tokennummer-Bereichen und diese mitunter sogar mit **unterschiedlichen** ID-Nummern. Daher berücksichtigt LEXCHECK diese Möglichkeit noch nicht und von einem damit überprüften Lexfile wird u.U. nicht der gesamte Inhalt angezeigt.

Daher habe ich nun das Programm LEXCHEC2 entworfen, das diese Beschränkung nicht mehr enthält. Außerdem habe ich das Format der Ausgabe etwas geändert, so daß nun auf weniger Platz mehr Information geliefert wird. Da dieses Ausgabeformat für Lexfileinhalte sicher aussagekräftiger und übersichtlicher ist als das bisher verwendete, werde ich es auch ab nun bei Artikeln der Lexfileserie verwenden.

Ein Beispielausdruck eines Lexfiles (z.B. EDLEX aus dem HP-Texteditor- oder Forth/Assemblermodul) sieht nun so aus:

```
EDLEX ID=F0 SIZE =2576 Bytes
  Messages!
  Polls!
01 DELETE      Stmt 240001
02 EDTEXT      Stmt 240002
03 FILESZR     XFN 240003 ($)
04 INSERT      Stmt 240004
05 REPLACE     Stmt 240005
06 SEARCH      XFN 240006 ($,#,#,#,#)
07 EDPARSE$    XFN 240007 ($)
*
EDLEX ID=52 (see:ID=F0)
02 SCROLL      Stmt 82002
03 MSG$        XFN 82003 (#)
*
```

In der Überschrift werden Filename, ID-Nr. hexadezimal und die Größe des Files dezimal angegeben. Hier ist wieder die tatsächlich im Speicher belegte Anzahl von Bytes gemeint, inclusive aller Header- und Linkbytes. Sie ist übrigens bei Lexfiles um 19 Bytes größer als die im CAT angezeigte.

Danach folgen unter Umständen die Meldungen „Polls!“ und/oder „Messages!“, falls in diesem File ein Pollhandling stattfindet, bzw. eigene Meldungstexte vorhanden sind. Anschließend folgt in jeder Zeile ein Token. Erst die Tokennummer hexadezimal, dann das Schlüsselwort. Nun wird mit XFN, Stmt oder Word angegeben, ob es sich um eine Funktion, ein Statement oder um ein ergänzendes Wort handelt (wie z.B. VARS und FLOW beim Befehl TRACE). Danach folgt die Codenummer des Tokens, wie sie auch in BASIC-Programmen erscheint, wenn ein Lexfile nach Eintippen einer seiner Schlüsselwörter entfernt wird. Sie besteht einfach aus der ID-Nr. dezimal mal 1000 plus Tokennummer dezimal.

Handelt es sich nun um ein Statement, so folgt in der Zeile dahinter im Normalfall nichts mehr. Sollte das Statement aber einer Beschränkung unterworfen sein, so wird entweder vermerkt:

„Not in PRG“, „Not from Kbd“, oder „Not after IF“. Oder mehrere dieser Meldungen.

Falls es sich um eine Funktion handelt (XFN), so wird nun durch symbolische Zeichen in Klammern die Anzahl und Art der erwarteten Parameter angegeben. Die einzelnen Parameter sind durch Kommata getrennt. Ein „#“ bezeichnet einen numerischen-, ein „\$“ einen String-Parameter. Ein „?“ bedeutet, das beide Typen erlaubt sind. Alle Parameter, die in eckigen Klammern stehen, sind optional, können also, müssen aber nicht verwendet werden. Falls das Symbol „()“ auftaucht, so ist damit eine Feld-Variable gemeint, die als solche vorher dimensioniert werden muß. Falls als Parameter „(-)“ angegeben ist, so bedeutet das eine parameterlose Funktion.

Sollte im überprüften Lexfile eine zweite Reihe von Token vorkommen (in einer Reihe müssen alle Token aufeinanderfolgende Nummern haben), so wird für die nächste Reihe wieder eine Überschrift mit der – eventuell neuen – ID-Nr. ausgegeben, allerdings diesmal ohne Filesize. Dafür steht dort eine Referenz zur ID der vorangehenden Reihe. Daran läßt sich dann im Listing erkennen, daß es sich um eine Unterreihe zu einem bereits vorhandenen Lexfile handelt.

Am Ende einer jeden Reihe wird ein „*“ ausgedruckt.

Auch dieses Programm ist übrigens in der Lage, entweder einen bestimmten Lexfile zu überprüfen, oder alle Lexfiles in einem bestimmten Port. Sowohl aktive Lexfiles werden bearbeitet, als auch mit „DISABLE“ deaktivierte (siehe Lexfile: DISABLEX). Falls das Programm mit einem Peek-Lexfile im Rechner (z.B. PEEKUTIL, DESAL, etc.) eingetippt wird, so werden auch Lexfiles mit PRIVATE Status fehlerlos bearbeitet (wie z.B. FORTHROM im Forth-Assembler Modul).

Das Programm wird einfach mit RUN gestartet und die Frage nach „File?“ mit dem Filenamem des zu überprüfenden Lexfiles beantwortet. Soll ein Port durchsucht werden, so ist stattdessen die Portnummer einzugeben, oder „0“ für Main Memory.

Vor dem Programmstart sollte ein Drucker mit PRINTER IS zugewiesen sein. Wer keinen Drucker besitzt sollte das Programm im DELAY Inf Modus starten. Die jeweils nächste Zeile wird dann durch Drücken irgendeiner Taste angezeigt.

Eine etwas erweiterte Version dieses Programms ist bei mir erhältlich, die es gestattet, die Ausgabe entweder auf die Anzeige, oder an den Drucker, oder direkt in ein Textfile zu leiten. Damit ist es dann möglich, mit Hilfe eines weiteren Programms den Textfile nach ID- und Tokennummern zu sortieren und automatisch auf Kollisionen (gleiche ID- und Tokennummern) überprüfen zu lassen.

Interessenten an diesen Programmen (LEXCHK2 und LC2SORT) wenden sich bitte per Brief oder Postkarte an mich (oder über Mailbox: M.FIEDLER und bekommen dann Listing und Anleitung zugeschickt. Auf Disk oder Magnetkarte sind diese beiden Programme sowie LEXCHEC2 bei unserer Programmbibliothek erhältlich.

Happy Lexchecking.

PROGRAMMLISTING

LEXCHEC2 BASIC 1964 05/12/88 19:33

```

10 ! ** LEXCHEC2 ** Erforscht Lexfiles
20 ! Erforderliche Lexfiles: DCMLX
40 INPUT 'Check FILE?', N$; N$
60 STD @ ON ERROR GOTO 'ERR' @ DIM Z$[80], N1$[45]
70 IF N$ > 'A' THEN 'NOK'
80 P=VAL(N$)
90 IF P THEN P$=':PORT('&STR$(P)&')' ELSE P$=':MAIN'
100 FOR I=1 TO 999 @ IF CAT$(I,P$)=' ' THEN I=INF @ GOTO 'NXT'
110 IF CAT$(I,P$) > 'a' THEN 'NXT'
120 N1$=CAT$(I,P$) @ N$=N1$[1,POS(N1$, ' ')-1]&P$
130 'NOK': A$=ADDR$(N$) @ N$=N$[1,POS(N$&':', ':')-1]
140 T$=PEEK$(HEXADD$(A$, '10'), 4)
150 IF T$# '802E' AND T$# 'FF00' THEN 'NXT'
160 A$=HEXADD$(A$, '20')
170 S=HTD(REV$(PEEK$(A$, 5)))+32 @ S=CEIL(S/2)
180 A$=HEXADD$(A$, '5')
190 'NRE': I$=REV$(PEEK$(A$, 2))
200 GOSUB 'HL'
210 A$=HEXADD$(A$, '2') @ T0=HTD(REV$(PEEK$(A$, 2)))
220 A$=HEXADD$(A$, '2') @ T9=HTD(REV$(PEEK$(A$, 2)))
230 C$=HEXADD$(A$, '2')
240 A$=HEXADD$(A$, '7') @ IF PEEK$(A$, 1)='0' THEN A$=HEXADD$(A$, '4F')
250 A$=HEXADD$(A$, '1') @ T$=HEXADD$(HEXADD$(A$, REV$(PEEK$(A$, 4))), 'FFFFFF')
260 A$=HEXADD$(A$, '4') @ IF PEEK$(A$, 4) # '0000' THEN PRINT ' Messages!'
270 A$=HEXADD$(A$, '4') @ IF PEEK$(A$, 5) # '00000' THEN PRINT ' Polls!'
280 A$=HEXADD$(A$, '5') @ C1$=A$
290 IF T9=0 OR T0>T9 THEN PRINT ' No Tokens' @ GOTO 'NTO'
300 FOR T=TO TO T9
310 X$=HEXADD$(T$, REV$(PEEK$(A$, 3)))
320 T0$=' '&STR$(HTD(I$)*1000+T) @ T0$=T0$[LEN(T0$)-5]
330 Z$=' '&DTH$(T)[4]&' '&MEM$(HEXADD$(X$, '1'), (HTD(PEEK$(X$, 1))+1)/2)
340 A$=HEXADD$(A$, '8') @ E$=PEEK$(A$, 1)
350 IF E$='F' THEN Z$[14]='XFN '&T0$ @ GOSUB 'FPA' @ GOTO 'ESL'
360 IF E$='0' THEN Z$[14]='Word '&T0$ @ GOTO 'ESL'
370 Z$[14]='Stmt '&T0$ @ GOSUB 'WRE'
380 'ESL': PRINT Z$
390 A$=HEXADD$(A$, '1')
400 NEXT T
410 'NTO': PRINT '*'
420 T$=PEEK$(C$, 5)
430 IF T$# '00000' THEN A$=HEXADD$(C$, REV$(T$)) @ I1$=I$ @ GOTO 'NRE'
440 'NXT': NEXT I
450 'EN': PUT '#43' @ END
460 'ERR': IF ERRL=130 AND ERRN=61 THEN 'NXT'
470 IF ERRL=440 AND ERRN=43 THEN 'EN' ! NUR EINZELFILE
480 BEEP @ DISP ERRM$:ERRL @ PAUSE @ END
490 'HL': Z$=N$ @ Z$[10]='ID='&I$
500 IF S THEN Z$=Z$&' SIZE='&STR$(S)&' Bytes' @ S=0 @ PRINT Z$ @ RETURN
510 Z$=Z$&' (see:ID='&I1$&')' @ PRINT Z$ @ RETURN
520 'FPA': D$=HEXADD$(C1$, DTH$((T-T0)*9))
530 E$=HEXADD$(D$, HEXADD$('1', REV$(PEEK$(HEXADD$(D$, '3'), 5))))
540 P=HTD(PEEK$(E$, 1))
550 Q=HTD(PEEK$(HEXADD$(E$, '1'), 1))
560 Z$=Z$&' ('
570 FOR I1=1 TO Q
580 IF I1>P THEN Z$=Z$&'["' @ SFLAG 0
590 IF I1#1 THEN Z$=Z$&' , '
600 E$=HEXADD$(E$, 'FFFFFF')
610 X=HTD(PEEK$(E$, 1))
620 IF X>11 THEN Z$=Z$&'?' ELSE IF X>7 THEN Z$=Z$&'#' ELSE Z$=Z$&'$'
630 IF MOD(X,4) THEN Z$=Z$&'()'
640 IF FLAG(0,0) THEN Z$=Z$&'}'
650 NEXT I1 @ IF Q=0 THEN Z$=Z$&'-'
660 Z$=Z$&' )'
670 RETURN
680 !
690 'WRE': E=HTD(E$)
700 IF E<=7 THEN Z$=Z$&' Not in Prg!"
710 IF NOT (MOD(E,8) DIV 4) THEN Z$=Z$&' Not after IF!"
720 IF NOT MOD(E,2) THEN Z$=Z$&' Not from Kbd!"
730 RETURN

```

Michael Fiedler
Friedrichstraße 17
6070 Langen

PRINTLEX

690 Bytes, IL, PRINTER, HP2225, HP82905B

... UND NOCH EIN LEXFILE: PRINTLEX

Dieser Lexfile ist gedacht für Besitzer der Hp-Drucker 82905B und Thinkjet und soll deren Ansteuerung vereinfachen und übersichtlicher machen.

Um den Drucker dazu zu veranlassen, bestimmte Sonderfunktionen auszuführen, wie z.B. Formfeed, oder einfach piepsen, sind im Normalfall bestimmte Steuerzeichen oder sogar Steuerzeichensequenzen an den Drucker zu schicken, wie sie im Handbuch angegeben sind. Dazu muß von BASIC aus oft auf die CHR\$ Funktion und die dazugehörigen Werte zurückgegriffen werden, was ein Programm leicht unübersichtlich werden läßt. Ein weiterer Nachteil ist, daß die Steuerzeichen bei der Programmstellung ständig im Handbuch nachgeschlagen werden müssen.

Um dem abzuhelpen, haben Mitglieder des französischen SIG den Lexfile PRINTLEX entwickelt und ihn in u.a. in JPC 40/87 und in CHHU Chronicle V2N7 veröffentlicht. Hier die Aufstellung im LEXCHEC2 Format:

| | |
|----------------|-------------------|
| PRINTLEX ID=E1 | SIZE=690 Bytes |
| 27 BELL | Stmt 225039 |
| 28 BOLD | Stmt 225040 |
| 29 CR | Stmt 225041 |
| 2A ESC\$ | XFN 225042 ([\$]) |
| 2B FF | Stmt 225043 |
| 2C LF | Stmt 225044 |
| 2D MODE | Stmt 225045 |
| 2E PERF | Stmt 225046 |
| 2F PL | Stmt 225047 |
| 30 UNDERLINE | Stmt 225048 |
| 31 WRAP | Stmt 225049 |
| * | |

Die Benutzung der 11 Schlüsselwörter ist denkbar einfach. 4 von ihnen senden bei ihrem Aufruf einfach eine Sequenz an den Drucker, der gerade als PRINTER IS aktiv ist:

BELL sendet Hex 07 (piepst)
 CR sendet Hex 0D (Carriage Return)
 FF sendet Hex 0D 0C(Form Feed)
 LF sendet Hex 0D 0A(Line Feed)

Hinter LF kann zusätzlich noch ein Multiplikator angegeben werden. So sendet LF 4 z.B. 4 mal Linefeed. Weitere 4 Wörter können mit ON und OFF kombiniert werden:

BOLD ON sendet Hex 0E
 (Fettdruck, oder 2. set)

BOLD OFF sendet Hex 0F
 (Fettdruck aus)

PERF ON sendet
 Hex 1B 26 6C 30 30 31 4C
 (Perforation Skip:
 neues Blatt wird rechtzeitig angefangen,
 ohne über die Perforation zu schreiben)

PERF OFF sendet
 Hex 1B 26 6C 30 30 30 4C
 (Perf. Skip aus)

UNDERLINE ON sendet
 Hex 1B 26 64 44
 (Unterstreichen)

UNDERLINE OFF sendet
 Hex 1B 26 64 40
 (Unterstreichen aus)

WRAP ON sendet
 Hex 1B 26 73 30 43
 (Zeilenumlegung im 80Z Mode)

WRAP OFF sendet
 Hex 1B 26 73 31 43
 (Zeilenumlegung aus)

Das Statement MODE kann mit den Argumenten 0-9 versehen werden und sendet an den Drucker den Code:

Hex 1B 26 6B 3a 53

wobei a das Argument 0-9 darstellt. Damit kann der Drucker in die entsprechenden Modes geschaltet werden, wobei 0 z.B. dem normalen 80-Zeichen Set entspricht, 1 dem 40-Zeichen Set, 2 dem komprimierten Set (142 Zeichen oder 132 beim HP82905B), 3 dem mittleren, usw., siehe Druckerhandbuch.

Das PL Statement kann entweder ohne Parameter verwendet werden, mit einem, oder mit 2. Es setzt die Länge der Seite: 1. Argument und die Länge des Textes auf der Seite: 2. Argument. Ohne Argument setzt PL das europäische Standardformat: 72-zeilige Blätter und 66 Zeilen für Text. Hier z.B. die gesendeten Sequenzen für PL, PL66, 54 und PL66:

PL: Hex 1B 26 6C 37 32 70 36 36
 66 31 4C

PL66, 54 Hex 1B 26 6C 30 35 34 46 1B
 26 6C 30 36 36 50

PL66 Hex 1B 26 6C 30 36 36 50

Falls nur ein Argument verwendet wird, so wird nur die Blattlänge gesetzt. Die Textlänge bleibt dabei unverändert.

Die einzige Funktion unter den Schlüsselwörtern ist ESC\$. Hier handelt es sich nicht um eine direkte Ansprache des Druckers, sondern um eine Stringfunktion, die beim Konstruieren oder Aussenden von Steuersequenzen verwendet werden kann. ESC\$ liefert einfach den Escape-Charakter CHR\$(27), der in vielen Druckerbefehlen verwendet wird. Verwendet man den optionalen Stringparameter, so wird der String an das Escapezeichen angehängt. Beispiel:

PRINT ESC\$(„k0S“) ist das selbe wie
 PRINT ESC\$&„k0S“ und entspricht
 PRINT CHR\$(27)&„k0S“

```

100 ! BEISPIELPROGRAMM
110 PRINTER IS :PRINTER
120 U$=ESC$(„,&dD)
    ! Code für Underline
130 O$=ESC$(„,&d@“)
    ! Code für Underline off
140 MODE 0 @ PL @ WRAP ON
150 LF 3
160 BOLD ON
170 PRINT TAB(33);„,Druckbeispiel“
180 LF 3
190 BOLD OFF
200 PRINT „,Hier beginnt der Text des
    Druckbeispiels.“
210 LF
220 PRINT „,Zur Demonstration der ESC$
    -Funktion wird gezeigt,“
230 PRINT „,wie mit“;
240 UNDERLINE ON @ PRINT „,zwei“;
    @ UNDERLINE OFF
250 PRINT „,verschiedenen Methoden“;
    U$; „,Underline“; O$; „,Mode“
260 PRINT „,gesetzt und gelöscht werden
    kann.“
270 FF @ BELL @ END
    
```

Hexdump-Listing PRINTLEX

(zum Abtippen benötigen Sie ein Hexdump-Ladeprogramm wie z.B. MAKE-FILE aus Prisma 7/86!)

PRINTLEX L ID#E1 690 Bytes

| | 0123 | 4567 | 89AB | CDEF | ck |
|------|------|------|------|------|----|
| 000: | 0525 | 94E4 | 45C4 | 5485 | 34 |
| 001: | 802E | 0011 | 5162 | 4078 | 9F |
| 002: | 3450 | 01E7 | 2130 | 0000 | A8 |
| 003: | F170 | 0000 | 0000 | 0000 | AD |
| 004: | 0F02 | 00DB | 0071 | 200D | 83 |
| 005: | 610D | 2200 | DD10 | 5420 | 0C |
| 006: | 0F82 | 0BB2 | 00DF | 20BC | 02 |
| 007: | 200D | 6306 | 1300 | D140 | A7 |
| 008: | 1630 | 0DC4 | 0504 | 00D3 | 71 |
| 009: | 5008 | 400D | 6602 | A400 | F8 |
| 00A: | D724 | 54C4 | C472 | 724F | D9 |
| 00B: | 4C44 | 4823 | 3425 | 9275 | 0F |
| 00C: | 4353 | 442A | 2364 | 64B2 | BD |
| 00D: | 3C46 | 4C27 | D4F4 | 4454 | 8E |
| 00E: | D270 | 5542 | 564E | 2305 | 35 |
| 00F: | C4F2 | F55E | 4445 | 425C | BB |
| 010: | 494E | 4037 | 7525 | 1405 | AF |
| 011: | 131F | F038 | F8BF | 3031 | ED |
| 012: | 5496 | 6811 | 718F | A2C2 | FC |
| 013: | 00E2 | EF1E | DDF0 | 08DB | 8D |
| 014: | 2E20 | 8D27 | 1308 | FE7A | DC |
| 015: | 2041 | F7BE | F681 | 08FE | C0 |
| 016: | 7A20 | 4FD7 | 9DF8 | D6CA | FB |
| 017: | 208D | E6A2 | 08D3 | 0350 | C3 |
| 018: | 3354 | 0218 | 18FD | FC20 | 7C |
| 019: | 14B8 | DCF2 | 508D | 0745 | C2 |
| 01A: | 0844 | 6600 | 8542 | 0048 | 68 |
| 01B: | F733 | 21F0 | 831D | 0310 | FD |
| 01C: | 48D3 | 9390 | 1311 | C414 | 16 |
| 01D: | 7D51 | 37C9 | 1080 | 7135 | 0F |
| 01E: | E6D7 | 7000 | 07D5 | 34B1 | F7 |

| | | |
|-----------------------------|-----------------------------|-----------------------------|
| 01F: 000C 906D 606D 015B A4 | 033: 6EFF 71EF F14A 8F50 65 | 047: 1361 3406 18E1 3606 1B |
| 020: 0017 0000 7DA3 4C10 D2 | 034: 450D 0570 E453 18F8 B4 | 048: 672D 61DF F4DC FF14 99 |
| 021: 00C2 0611 8061 3710 22 | 035: 71F0 8FBC 6315 FC10 5F | 049: A8F5 0450 5D17 20DB EC |
| 022: 8012 0864 9011 8060 37 | 036: 1732 F6D0 0101 783E 1C | 04A: B162 C673 2307 6363 00 |
| 023: 18D8 4A80 14A3 10E9 87 | 037: 1A01 11CC 50F6 8816 7D | 04B: 6613 C47E 3F10 117F D7 |
| 024: 6200 0133 FFFA CEFF E4 | 038: A907 1EFF AEDF F776 24 | 04C: 1331 BE95 F214 6859 81 |
| 025: 7D4F 170C 3FFF ECEF 81 | 039: 0AF2 A969 765E 0431 0C | 04D: 8BE6 1849 1318 FDF8 2E |
| 026: F73D F490 753F 1F07 0D | 03A: 03A6 A8FE 1DE0 8F12 1B | 04E: E07B 1F13 3133 3105 89 |
| 027: E2F1 E030 FFFA 9EFF 02 | 03B: DE03 A5B1 62B6 0035 98 | 04F: 7D2F 879D 0111 3164 2E |
| 028: 7400 6CAF 1F65 9F2A 80 | 03C: A728 FC2B 9015 CA13 FF | 050: 7D1F 6C2D 87CF FA0C EB |
| 029: E214 D7F0 F1D0 0140 CA | 03D: 6066 DCDC BDFF E4DF E3 | 051: FF72 2D4F 0748 C4B1 4C |
| 02A: 194E 8213 7109 1351 8D | 03E: F735 ED03 1C45 40E4 98 | 052: 6246 0477 7C4B 1624 EF |
| 02B: C120 31B1 14D8 F064 5B | 03F: 7D20 6C3E 048F 871F C3 | 053: 6440 6CFF 2FBF F77F 79 |
| 02C: A184 08D7 B181 1360 50 | 040: 08F2 EA21 044F 0AF2 73 | 054: C411 795C 5B16 2371 A7 |
| 02D: 68F8 3DB0 1371 3413 1B | 041: 3299 99F2 008D 91FB 08 | 055: 3347 A4C5 B162 3703 8A |
| 02E: 51C1 10AC 2D81 098F 6C | 042: 0AE7 D81B 178F 2367 E5 | 056: 34F |
| 02F: 261B 111A 1351 C111 01 | 043: B162 C615 C616 6815 B7 | |
| 030: 9134 1813 1B11 4C07 7A | 044: 8153 1032 2D4D 1A6A 2A | |
| 031: 1345 7A36 EFFA FDFF AC | 045: 1481 6181 10D5 DE20 72 | |
| 032: 746F 797E 1C06 EE0B 91 | 046: AEB1 4C16 1310 114C 6B | |

Michael Fiedler
Friedrichstr. 17
6070 Langen

Eigenwert-Berechnung

13 Zeilen, 494 Bytes, HP-71, MATHE

Das Programm berechnet alle komplexen Eigenwerte einer reellen Matrix A beliebiger Ordnung n.

Zunächst werden die Koeffizienten $p_i, i = 1, \dots, n$ der charakteristischen Gleichung $\det(\underline{I}s - \underline{A}) = p_0s^n + p_1s^{n-1} + \dots + p_n = 0$ rekursiv aus folgender Beziehung gebildet:

$$\underline{M} \begin{bmatrix} m_{11} \dots m_{1p} \\ \vdots \\ m_{p1} \dots m_{pp} \end{bmatrix} = \underline{I}s - \underline{A}$$

$p=n$ (Ordnung der Matrix M)

$$\det(\underline{M}) = \sum_{j=1}^p (-1)^{i+j} m_{ij} \det(\underline{M}_{ij}) \text{ wenn } p \text{ größer } 1$$

i beliebig zwischen 1 und n, im Programm $i=1$
 $\det(\underline{M}) = m_{11}$ wenn $p=1$

\underline{M}_{ij} ist eine Matrix der Ordnung $p-1$, die man aus M durch Streichen der Zeile i und Spalte j erhält.

Im Programm wird $\det(\underline{M}_{ij})$ nicht ausgewertet, wenn $m_{ij}=0$ (Zeitersparnis).

Danach werden die Nullstellen der charakteristischen Gleichung mit dem Algorithmus PROOT aus dem Mathematikmodul berechnet. Ist kein Mathematikmodul vorhanden, müssen die Nullstellen anders berechnet werden.

Nach Programmstart (RUN EW) werden die Ordnung und die Elemente von A abgefragt, die Eigenwerte berechnet und ausgegeben.

EW BASIC 494

```

10 DELAY 8 @ INTEGER N @ INPUT "EW-Berechnung: n=";N
20 OPTION BASE 0 @ REAL P(N) @ OPTION BASE 1 @ SHORT A(N,N) @ INTEGER S(N,N) @ COMPLEX L(N)
30 MAT INPUT A @ MAT DISP A; @ DELAY 0 @ DISP "... " @ MAT A=-A @ MAT S=IDN @ CALL L DET(A,S,P)
40 MAT L=PROOT(P) @ DELAY 8,8 @ MAT DISP L @ DESTROY N,A,S,P,L @ END ALL
50 SUB DET(A(,),S(,),P())
60 N=UBND(A,1) @ IF N=1 THEN P(1)=A(1,1) @ P(0)=S(1,1) @ END ELSE M=N-1
70 OPTION BASE 0 @ REAL R(M) @ OPTION BASE 1 @ SHORT B(M,M) @ INTEGER T(M,M)
80 FOR J=1 TO N @ IF A(1,J)=0 AND S(1,J)=0 THEN 130
90 FOR C=1 TO M @ FOR D=1 TO M
100 B(C,D)=A(C+1,D+(D)=J) @ T(C,D)=S(C+1,D+(D)=J) @ NEXT D @ NEXT C
110 MAT R=ZER @ CALL DET(B,T,R)
120 FOR L=0 TO N @ P(L)=P(L)-((-1)^J*((L#0)*A(1,J)*R(ABS(L-1)))+(L#N)*S(1,J)*R((L#N)*L)) @ NEXT L
130 NEXT J @ END SUB
    
```

Ralf Wanser (42)
Am Erdbeerstein 31
6240 Königstein

Funktion:

STAYDOWN erlaubt die Eingabe von zwei-Tasten Ctrl-, Alt- und Shift-Kombinationen als zwei separate Anschläge. STAYDOWN ist ein Speicherresidentes Programm.

Format:

STAYDOWN [D] [/U]

Typ:

Intern Extern

Hinweise:

Nach dem Aufruf von STAYDOWN merkt sich der Rechner jede Betätigung von Ctrl, Alt und Shift. Ein großes A kann jetzt mit Shift und A nacheinander (und nicht gleichzeitig) gedrückt eingegeben werden. Dreierkombinationen wie Ctrl-Alt-Del können so nicht eingegeben werden. Hierzu muß man erst Ctrl und dann Alt-Del drücken. Die alte Möglichkeit die Tasten gleichzeitig zu drücken funktioniert übrigens immer noch.

Hat man mal versehentlich Shift erwischt, drückt man es einfach nochmal, und der Status ist wieder gelöscht (das gilt auch für Alt und Ctrl).

Mit dem Parameter /D schaltet man STAYDOWN wieder ab, mit /U entfernt man es wieder aus dem Speicher.

Quelle:

PC-Magazine, New York

Funktion:

RECORDER ist ein Speicherresidentes Programm, das, solange es im Speicher ist, eine Statistik darüber führt, welche Dateien wie oft bearbeitet wurden.

Format:

RECORDER [n] [/R]

Typ:

Intern Extern

Hinweise:

RECORDER registriert jeden Zugriff auf Platte/Diskette. In der Tabelle ist Platz für 200 verschiedene Dateien, auf die zugegriffen wird, vorgesehen. Mit dem Parameter n kann man diese Zahl auf 2000 erweitern.

Wird RECORDER dann erneut aufgerufen, gibt es die Statistik auf dem Schirm aus. Mit dem Parameter /R kann man die Einträge in der Tabelle wieder auf Null stellen.

RECORDER kann z.B. dazu verwendet werden, um herauszufinden, welche Dateien man sinnvoll auf einen schnellen Massenspeicher (RAMDISK, Festplatte) auslagert und welche nicht.

Quelle:

PC-Magazine, New York

Funktion:

TOGGLE ermöglicht CAPS-, NUM- und SCROLL-LOCK softwaremäßig zu setzen, bzw. ihren Status anzuzeigen.

Format:

TOGGLE [+c | -c] [+n | -n] [+s | -s]

Typ:

Intern Extern

Hinweise:

Ohne Parameter aufrufen, gibt TOGGLE den Status der CAPS-, NUM- und SCROLL-LOCK Tasten aus.

Mit +c/-c kann man CAPS-LOCK, mit +n/-n NUM-LOCK und mit +s/-s SCROLL-LOCK ein-/ausschalten.

Quelle:

PC-Magazine, New York

Funktion:

Mit TOUCH kann man DOS-Datum und Zeit von Dateien ändern.

Format:

TOUCH [a:][Pfad]Datei [/D Datum] [/T Zeit]

Typ:

Intern Extern

Hinweise:

Wird TOUCH ohne Parameter aufrufen, gibt es einen Hilfbildschirm aus.

Der angegebene Dateiname darf Wildcards (*,?) enthalten.

Sind weder /D noch /T angegeben setzt TOUCH Zeit und Datum auf die aktuelle Systemzeit.

Das Datum muß tt-mm-jj oder tt/mm/jj Format haben, die Zeit hh:mm:ss Format.

Quelle:

PC-Magazine, New York

E-Technik Buchbesprechung

Steimle, Wolfgang

Der Bipolartransistor in linearen Schaltungen

Band 1: Grundlagen, Ersatzbilder, Programme (HP-41)
280 Seiten DM 48,- ISBN 3-486-25561-4

Band 2: Formeln, Schaltungsbeispiele, Programme (HP-41)
195 Seiten DM 38,- ISBN 3-486-27641-7

Oldenbourg Verlag München 1984

Zu Beginn gleich eine Feststellung: diese Bücher wurden nicht um des Programmierens willen geschrieben, sondern um einem Lernenden die Arbeitsweise und Einsatzmöglichkeiten des Bipolartransistors darzulegen. Die in den Büchern enthaltenen Programme sollen (und tun es auch) die Berechnung der vielen Parameter erleichtern.

Nun zum Inhalt:

Band 1:

1. Halbleitertechnische Grundlagen (10 S)

Es werden die Halbleitergrundlagen am Beispiel des pn-Übergangs dargestellt. Ein gewisses Grundwissen der Elektrotechnik sollte vorhanden sein. Die Aussagen sind, wie in allen Kapiteln dieser zwei Bände, durch Bilder und Kennlinien anschaulich dargestellt. Alle Gleichungen sind, im ganzen Buch, durchnummeriert, so daß übersichtliche Verweise möglich sind.

2. Entwicklung von Ersatzbildern für bipolare Transistoren (73 S)

Hier werden die Grundlagen des Bipolartransistors mit Herleitung der Differenzgleichungen und Darstellung der Kennlinien dargestellt. Die Gleichungen, die zur Berechnung der Werte und Parameter nötig sind, werden ausführlich hergeleitet und zur Festigung der Theorie werden die Kennlinien und -werte eines konkreten Bipolartransistors herangezogen.

Aufgrund des so gewonnenen Wissens werden die T- und π -Ersatzbilder für unterschiedliche Anwendungen hergeleitet und verständlich gemacht.

Zum Schluß werden noch einmal alle hergeleiteten Ersatzbilder mit den zugehörigen Gleichungen tabellarisch aufgeführt (gut zum Nachschlagen).

3. Zur Anwendung von Ersatzbildern (4 S)

Hier wird kurz auf die Anwendung der Ersatzbilder eingegangen, wie sie z.B. bei mehrstufigen Verstärkern zu verwenden sind.

4. Wechselstromanalyse der Emitterschaltung (22 S)

Die drei wichtigsten Größen Eingangs- und Ausgangsimpedanz sowie Spannungsverstärkung werden ausführlich erklärt, hergeleitet und mit Diagrammen verdeutlicht. Auch hier fehlt es nicht an einem konkreten Beispiel mit einem existierenden Transistor.

5. Wechselstromanalyse der Kollektorschaltung (31 S)

Hier werden die Größen Stromverstärkung, Spannungsverstärkung und Ausgangsimpedanz hergeleitet. Wie bei der Emitterschaltung ist auch hier alles ausführlich mit Diagrammen und konkretem Beispiel versehen.

A1. Allgemeine Hinweise (9 S)

Es werden alle verwendeten Formelzeichen mit ihrer genauen Bedeutung aufgelistet. Zusätzlich wird die verwendete Formelschreibweise dargestellt.

A2. Übertragungsfunktionen (9 S)

Die Übertragungsfunktion $F(s) = U_a/U_e$ wird in der komplexen s-Ebene, der Ortskurve und dem Bodediagramm dargestellt. Zusätzlich wird die Verwendung der HF-Tapete erläutert.

A3. Einige Besonderheiten bei aktiven Zweitoren (9 Seiten)

Hier wird auf die Möglichkeit der Annäherung einer nicht linearen Funktion in kleinen Bereichen durch eine lineare Funktion eingegangen. Der größere Teil dieses Kapitels befaßt sich aber mit kapazitiver Rückwirkung des Ausgangs- auf das Eingangssignal und deren Phasenverschiebung.

A4. Berechnung der h- und Y-Parameter eines Transistors (14 S)

In diesem Kapitel wird der Zusammenhang zwischen den bisher verwendeten Größen und den h- und Y-Parametern hergestellt.

A5. Programme für den Taschenrechner HP-41CV (84 S)

Ich möchte mich hier auf die Auflistung der Bezeichnungen der enthaltenen Programme beschränken:

save display format / restore display format / question / output data / input data / Alpha-View / X-view / Wurzeln einer Gleichung 3. Grades / Rationale Funktion 3. Grades mit imaginärem Argument / Ausgabe komplexer Zahlen / Autom. Erhöhung von Variablen / Basis-Emitter-Spannung (DC) / Analyse der Gleichstrom-Grundschialtung bei konstanter Temperatur / Analyse der erweiterten Gleichstrom-Grundschialtung / Synthese der Gleichstrom-Grundschialtung / Kurzdaten einer Emittierstufe (AC) / Emitterschaltung (AC) / Ausgangsimpedanz in Emitterschaltung (AC) / Ersatzschaltung der Ausgangsimpedanz in Emitterschaltung (AC) / Stromverstärkung in Kollektorschaltung (AC) / Eingangsimpedanz in Kollektorschaltung (AC) / Spannungsverstärkung in Kollektorschaltung (AC) / Ausgangsimpedanz in Kollektorschaltung (AC) / Schwingmöglichkeit einer Kollektorstufe (AC).

Diese umfangreiche Programmsammlung enthält immerhin 5098 Zeilen, ein Argument für sich. Leider aber mit einem kleinen Wehmutstropfen; es liegt kein einziger Strich Bar-Code bei.

Als Abschluß dieses Bandes gibt es noch 5 Seiten Literaturverzeichnis und Sachregister.

Band 2:

1. Einleitung (2 S)

Sie enthält grundsätzliche Worte zum Dimensionieren von Schaltungen.

2. Zusammenstellung der wichtigsten Ergebnisse aus Teil 1 (31 S)

Alle wichtigen Punkte aus Band 1 (ohne Programme) sind hier zusammengefaßt.

3. Dimensionierung einer Konstantstromschaltung (18 S)

Anhand eines konkreten Beispiels wird eine Konstantstromquelle dimensioniert. Das Beispiel ist leicht auf eigene Anwendungen übertragbar.

4. Zweistufiger Wechselstromverstärker (36 S)

Im vorhergehenden Abschnitt handelte es sich um eine Schaltungssynthese. D.h. eine Schaltung mußte so dimensioniert werden, daß die bestehenden Forderungen erfüllt wurden. In diesem Abschnitt wird eine Schaltungsanalyse durchgeführt. Es werden die Eigenschaften der vorgegebenen Schaltung ermittelt. Auch dieses Verfahren ist ausführlich erläutert und leicht auf andere Schaltungen anwendbar.

A1. Allgemeine Hinweise (9 S)
siehe Band 1

A2. Programme für den Taschenrechner HP-41CV (85 S)

Dies ist genau das Kapitel A5 aus Band 1. Es wurden aber die fünf Fehler beseitigt sowie einige Verbesserungen an den Programmen vorgenommen, leider auch hier ohne Bar-Code.

Es folgen noch sechs Seiten mit dem Inhaltsverzeichnis von Band 1, einem tabellarischen Listing der Programmänderungen sowie einem Literaturverzeichnis und Sachregister.

Abschließend kann man sagen: zwei gute Bücher aus der Hochschulpraxis für die Praxis, mit guten Programmen für den HP-41. Sollte jemand meinen nur mit Band 2 auskommen zu können, so sollte er doch schon einiges über die Grundlagen und Schaltungen des Bipolartransistors, einschließlich der Berechnungen, wissen.

Werner Meschede
Sorpestraße 4
5788 Siedlingshausen

Dreieck

895 Bytes, 127 Regs. SIZE 016

Dieses Programm ist eine überarbeitete Version des PRGM „TRI1“ von A. Radas

Erforderlich ist SIZE 016, PRGM-REG 127, Drucker in NORM-Modus. Bei Abwesenheit eines Druckers sind die PRA-Befehle durch PROMPT-Befehle zu ersetzen. (Zeile 285 und 410, wobei der von Zeile 410 zweckmäßig nach 389 und 353 verlegt wird!)

| Bedienung: | Tasten | Ausgabe |
|--|-----------------------|------------|
| 1. Start | XEQ TR3 oder J | DATEN |
| 2. Längen bzw. Winkel nach der 3. Eingabe | Zahl, a ... od. A ... | 3 Δ-Stücke |
| 3. Weitere Berechnungen, dann D dann Schritt 2 | | |
| 4. Wiederholung der Ergebnisse | E | wie Nr. 2 |
| 5. Fläche und Höhen | F | FI + Höhen |
| 6. Alle Stücke, auch die anfangs eingegebenen | G | Alles |

- Teilweise Verwendung von Stücken aus der vorhergehenden Berechnung aber nur (ohne Zahleneingabe) D
a ... od. A ... 3 Δ-Stücke
 - Aus den Ergebnissen die ursprünglichen Eingaben dazu auch 4. 5. und 6. e die Eingaben
 - Reihenuntersuchungen **nach** einer Berechnung Einzugeben der Code für die gewünschte Variable d VBL?
a=6 b=5 c=4 α=3 β=2 oder γ=1
- (muß zum Eingaben-Set gehören)
wird nach jedem Halt mit R/S R/S um 5% (Z.075) erhöht und im ALPHA angezeigt Auswirkung

Alles läuft auch, wenn 2 Lösungen angezeigt sind. Das PRGM erkennt auch sinnlose Eingaben als solche. Ebenso meldet es sich, wenn 2- oder 3-mal identische Werte eingegeben werden.

```

d REIHEN          37+LBL a
D ATEN IN+CALC.   6 GTO 00
S: a.. 4: A..     40+LBL b
E RGEBNIS         5 GTO 00
F LAECHE+HOEHE   43+LBL c
G ALLES           4 GTO 00
I NFO
J START          46+LBL A
                 3 GTO 00

01+LBL "TR3"     49+LBL B
02+LBL J         2 GTO 00

16 PSIZE CLKEYS "DA"
14 PASH "RE" CHS
PASH "ER" 15 PASH
"KO" CHS PASH "FH"
21 PASH "GE" 22 PASH
1.007006 STO 13

26+LBL "DA"
CLX STO 14 X<>F 3
STO 00 "DATEN" AVIEW

34+LBL 11        70+LBL "RE"
TONE 5 STOP      "VBL?" PROMPT STO 15
    
```

```

74+LBL 12
1.05 ST* IND 15
XEQ IND 15 "t = "
ARCL IND 15 AVIEW
XEQ 07 STOP GTO 12

84+LBL "KO"
FC?C 08 GTO 08 XEQ 08
RCL 13 REGSWAP

90+LBL 07
CF 00 CF 07 CLX X<>F
126 - GTO 00

98+LBL 08
CF 00 CF 07 CLX X<>F
CHS

104+LBL 00
CF 08 2 / 99 +
STO 14 SF 25 CLD
GTO IND 14 TONE 3
"EINGABE-FEHLER!"
PROMPT

117+LBL 01
"4G" RTN
    
```

120*LBL 02
"4B" RTN

123*LBL 03
"4A" RTN

126*LBL 04
"Sc" RTN

129*LBL 05
"Sb" RTN

132*LBL 06
"Sa" RTN

135*LBL 57
136*LBL 73
137*LBL 74
138*LBL 77
139*LBL 78
140*LBL 80
SF 01 XEQ 05
GTO IND 14

144*LBL 50
145*LBL 55
146*LBL 58
147*LBL 85
148*LBL 86
149*LBL 88
SF 02 XEQ 06
GTO IND 14

153*LBL 43
RCL 05 RCL 04 R-P X12
RCL 06 X12 - RCL 05
RCL 04 * ST+ X /
ACOS FC? 25 GTO 04
STO 03 SF 03

171*LBL 50
172*LBL 57
173*LBL 71
RCL 03 RCL 05 P-R
RCL 04 - CHS R-P
FC?C 03 STO 06 SF 03

184*LBL 15
STO 00 RDN STO 02 1
RCL 03 RCL 02

191*LBL 13
+ 180 - CHS X<=0?
CF 25 FC? 25 GTO 04
STO IND Y FS?C 03
GTO 16 RCL 01 SIN
RCL 06 RCL 03 SIN /
* STO 04 GTO IND 00

212*LBL 61
213*LBL 80
214*LBL 86
1 RCL 03 RCL 02
GTO 00

219*LBL 64
220*LBL 78
221*LBL 85
3 RCL 02 RCL 01
GTO 00

226*LBL 62
227*LBL 77
228*LBL 88
2 RCL 03 RCL 01

232*LBL 00
1 STO 00 RDN GTO 13
237*LBL 01
LASTX RCL 02 SIN *
STO 05 GTO 16

244*LBL 49
245*LBL 55
246*LBL 74
SF 04 XEQ 07

249*LBL 47
250*LBL 58
251*LBL 73
RCL 05 RCL 06 X<Y?
SF 08 / RCL 03 SIN *
ASIN 2 GTO 15

263*LBL 02
FC? 08 GTO 16 RCL 13
REGMOVE 180 RCL 02 -
3 GTO 15

273*LBL 03
SF 03 XEQ 16 RCL 13
REGSWAP

278*LBL 04
ADV FC? 25 "KEINE"
FS? 25 "2"
"1 LOESUNGEN" PRA
FC? 25 RTN

288*LBL 16
FS? 04 XEQ 07 FS? 01
XEQ 06 FS? 02 XEQ 05
FS?C 03 RTN CF 25
RCL 14 ST+ X 72 -
X<>F GTO 08

304*LBL 05
XEQ 06

306*LBL 06
SF 05

308*LBL 07
RCL 06 X<> 05 FS? 05
X<> 04 STO 06 RCL 03
X<> 02 FS?C 05 X<> 01
STO 03 RTN

320*LBL 17
FC? 08 GTO 00 RCL 13
REGSWAP FS?C 00 GTO 00
SF 00 "2.LSG." ADV
PRA GTO IND 00

332*LBL 00
ADV CF 07 RTN

336*LBL "GE"
SF 07

338*LBL "ER"
ADV

340*LBL 08
8 STO 00 6 STO 14

345*LBL 14
FC? IND 14 FS? 07
GTO 01 GTO 00

350*LBL 01
RCL IND 14 XEQ 01

353*LBL 00
DSE 14 GTO 14 FC? 07
GTO 17 GTO 09

359*LBL "FH"
360*LBL 09
9 FC? 07 STO 00 7.01
STO 14 RCL 06 2 /
RCL 05 RCL 01 SIN *
* LASTX X<>Y RCL 05
RCL 03 SIN * RCL 04
ST* L RDN LASTX RDN

385*LBL 18
SIGN RDN XEQ 01 LASTX
RDN ISG 14 GTO 18
GTO 17

394*LBL 01
XEQ IND 14 FS? 00 "1"
FC? 00 "1" "1" = "
E2 X<>Y? "1" RDN
E1 X<>Y? "1" RDN
ARCL X PRA RTN

412*LBL 07
"FL" RTN

415*LBL 08
"Ha" RTN

418*LBL 09
"Hb" RTN

421*LBL 10
"Hc" RTN

Dr. Hans Berghaus
An der Joch 19
5000 Köln 80

Buchbesprechung

Titel: HP41 Barcodes mit dem HP-IL-System
Autor: Konrad Albers
Verlag: Heldermann Verlag Berlin
ISBN-Nr.: 3-88538-804-9

Nach dem Abdruck meines ersten Barcodartikels schickte mir Hr. Albers ein Exemplar seines Buches mit der Bitte um Besprechung in PRISMA, ich hätte dies auch ohne seine Bitte getan, wenn ich dieses Buch schon vorher einmal in die Hände bekommen hätte, mir persönlich hat es im großen und ganzen sehr gut gefallen.

Meine Artikel zum Thema Barcodes auf dem HP41 sollten nur die technischen Grundzüge aufzeigen, um selbst Barcodes herstellen zu können, egal auf welchem System. Ich habe mich zugegebenermaßen dabei recht knapp mit Informationen für Programmierer zur Realisierung dieser Barcodes gehalten, dies hätte den Rahmen dieser Artikelserie bei weitem gesprengt, da hätte ich dann wirklich ein Buch schreiben müssen, wie das, mit dem ich mich jetzt befassen möchte.

Grundsätzliches gleich zu Anfang: dieses Buch befaßt sich, wie sein Titel schon sagt, mit den Barcodes des HP41 und dessen Herstellung mit Hilfe des HP41. Dafür hat der Autor zu den meisten Beispielen jeweils drei Varianten an Programmen geschrieben und recht gut dokumentiert, nämlich je eine Realisierung für den Thermodrucker, eine für den Thinkjet und auch sehr oft für den Plotter HP7470 mit IL-Schnittstelle und Plottermodul im HP41, er behandelt diese Geräte sogar ein wenig, soweit es eben für die Barcodeherstellung relevant ist. Dabei kann ein Anfänger sehr viel über den Umgang mit diesen Geräten lernen, wie man diese steuert und welche Features diese haben, gerade für Programmieranfänger enthält dieses Buch sehr viele Hilfen und Tips für die Realisierung von Problemen mit Hilfe eines Rechners im Verbund mit Peripheriegeräten, dies ist ja immer wieder eine schwere Hürde für viele Anfänger, wie fang ich's an oder besser wo.

Mit diesem Buch kann man fast spielend etwas über die Barcodes lernen und gleichzeitig viel Hintergrundwissen über die Fähigkeiten des HP41 und seiner Peripheriegeräte erfahren, eine fast ideale Form des Entdeckens.

Das Inhaltsverzeichnis ist sehr fein gegliedert, hier findet man recht schnell, was man sucht. Es gibt derlei zwei, eines für den Inhalt nach der Seitenreihenfolge sortiert, das zweite Inhaltsverzeichnis listet die in diesem Buch enthaltenen Programme nach Kapiteln unterteilt anhand ihrer Labels auf, zu jedem Programmnamen (Label) gibt es einige

Worte zu dessen Funktion, alles sehr übersichtlich gestaltet.

Am Ende des Buchs findet man ein Stichwortverzeichnis, das man allerdings im Gegensatz zum Inhaltsverzeichnis nur noch als total verhungert bezeichnen kann, ganze 1½ Seiten für ein Buch von über 300 Seiten Stärke sind schon sehr mager.

Ich glaube es ist am besten, wenn ich an Hand des Inhaltsverzeichnisses das Buch etwas genauer betrachte, fangen wir gleich damit an:

1. Kapitel

HP41 Barcodes und die Geräte

Hier befaßt sich der Autor mit den Vor- und Nachteilen des Barcodes für den HP41, erklärt ganz kurz die verschiedenen Barcodetypen, die zur Verfügung stehen und die er dann auch in seinem Buch behandelt, es scheint nichts zu fehlen.

Dann beginnt ein großes Kapitel mit grundsätzlichen Informationen zum Aufbau der HP41 Barcodes, an erster Stelle die physikalischen Dimensionen, dann folgt eine Erläuterung der HP 41 Byte-Tafel, wie sie wohl schon jeder gesehen hat – hier sind alle Befehle und Zeichen in einer hexadezimalen Tabelle zusammengefaßt. Die Bildung der Prüfsummen mit 4 oder 8 Bit wird kurz erläutert, darauf folgt ein kurzer Vergleich zwischen der Herstellung der Barcodes mit dem Thermodrucker, dem Thinkjet-Drucker und einem HP7470 Plotter. Es folgt die Beschreibung der Barcodeherstellung mit dem Thermodrucker, eine Erläuterung der Nebeneffekte wie die Beachtung der Papierqualität und der Plotterstiftqualität schließen sich diesem an.

Eine Betrachtung über das notwendige Zubehör für die Barcodeherstellung ist der Abschluß dieses Kapitels.

2. Kapitel

ALPHA-Einzelzeichen ALPHA- und synthetische Textzeilen Drucker Sonderzeichen

Der erste Abschnitt dieses Kapitels befaßt sich mit der Herstellung von ALPHA-Einzelzeichen, dazu gehören die Zweibyte ALPHA-Barcodes wie auch die Dreibyte ALPHA-Barcodes, die allerdings auch mehr als ein ALPHA-Zeichen pro Zeile ermöglichen.

Anhand dieser beiden Barcodetypen wird sehr ausführlich die Bildung der 4-Bit Prüfsumme wie auch der 8-Bit Prüfsumme für die beiden Barcodetypen beschrieben, Programme zur Herstellung der beiden Barcodes sind ebenfalls sehr

ausführlich dokumentiert und für jeden Programmierer interessant, auch für die, die es noch werden wollen.

Es gibt jeweils dieselben Programme für verschiedene Peripheriegerätekombinationen wie z.B. entweder für den HP41 nur mit X-Function Modul oder mit Plotter-Modul und HP7470 Plotter, hier sieht man dann sehr deutlich den wesentlich geringeren Programmieraufwand mit Hilfe des Plottermoduls, worauf der Autor immer wieder hinweist. Auf mehreren Seiten sind dann alle ALPHA-Barcodes, auf dem Plotter hergestellt, abgedruckt.

Der nächste Abschnitt befaßt sich mit der Herstellung von synthetischen Textzeilen sowie deren Anwendung z.B. für die Steuerung der Flags beim HP41 mit Hilfe der Funktion STOFLAG aus dem X-Function Modul. Anhand dieser Funktion werden die Bedeutungen der einzelnen Flags des HP41 und deren Verwendungsmöglichkeiten beim Programmieren im einzelnen untersucht, gerade für Leute, die noch nicht so tief in den Rechner eingestiegen sind, ein sehr interessantes Kapitel Interna.

Auch hier finden wir wieder mehrere Programme für die Herstellung dieser Barcodes mit verschiedenen Peripheriegerätekombinationen, sei es jetzt der normale Thermodrucker oder der Thinkjet Tintenstrahldrucker.

Der dritte und letzte Abschnitt ist den Drucker Sonderzeichen gewidmet. Dies bezieht sich natürlich nur auf den Thermodrucker des HP41, in erster Linie dabei auf den IL-Thermodrucker, aber nicht ausschließlich. Dem Autor geht es dabei vor allem um die Verkürzung von Programmen mit Hilfe der synthetischen Barcodezeilen, die Drucker Sonderzeichen oder ganze Graphiken erzeugen sollen, die normale Programmierung des Druckers mit der Funktion BLDSPEC ist ja sehr speicherplatzintensiv und langwierig, es gibt da eben auch viel elegantere Wege zu Graphiken zu kommen. Neben einigen kleinen Hilfsprogrammen für Umrechnungen von Spalteninformationen für den Drucker in Dezimalwerte und Ähnliches werden auch wieder Programme für die Herstellung dieser Barcodes angeboten, wieder dieselben Lösungen für verschiedene Peripheriegerätekombinationen, je nachdem, was dem Anwender eben zur Verfügung steht.

3. Kapitel

Zahlen und sequentielle Daten

Hier geht es allgemein um den Datenbarcode, zuerst nur um den normalen einzeiligen. Besprochen wird die Berechnung seiner 8-Bit Prüfsumme sowie der etwas ungewohnten Kodierung von nor-

malen Zahlen, wie sie auch der Rechner für seine interne Darstellung benutzt, wieder etwas Rechnerinterna für Programmierneulinge. Der sogenannte Kurzformexponent, seine Anwendung in Programmen zur Speicherplatzersparnis und Geschwindigkeitserhöhung wird hier auch noch eingehend behandelt, ebenso dessen Herstellung auf der Basis von Barcodes, dem einfachsten Eingabemedium für diese Programmierhilfe.

Ein Kurzformexponent ist z.B. E-6 statt 1E-6.

Es folgt die Behandlung des sequentiellen Barcodes, einer Sonderform des einzeiligen Datenbarcodes, der als Einlesesicherheit eine fortlaufende Zeilennummer ähnlich der des Programmbarcodes besitzt, um die richtige Einleseereihenfolge zu gewährleisten.

Wie in den anderen Kapiteln gibt es hier wieder mehrere Programme zur Herstellung dieses Barcode-Typs, je nach Rechnerkonfiguration.

4. Kapitel

Anweisungen Befehle Funktionen und Synthetisches zum Programmieren

Wie die Überschrift schon zeigt, handelt es sich hier um das mit Abstand größte Kapitel des Buches mit knapp 200 Seiten Umfang, ein ganz schöner Klotz, der allerdings wiederum in relativ viele kleine Kapitel aufgeteilt ist, zum Nachschlagen besonders wichtig.

Der erste Abschnitt befaßt sich mit der einfachsten Barcode-Funktion, dem Einbyte Barcode, der die Tastenfeldfunktionen repräsentiert. Auch hier wieder verschiedene Programme zur Herstellung dieser Barcodes.

Der zweite Abschnitt ist den Zweibyte Barcodes, hier speziell den Tastenfeldfunktionen ohne Argument, gewidmet. Hier gibt es sehr viele Varianten von Funktionen, deren Darstellung als Barcode mit allen Nebeneffekten bei der Benutzung, dieses Kapitel geht da schon sehr genau auf diese Thematik ein, eine Unzahl von praktischen Beispielen erläutern selbst etwas kompliziertere Zusammenhänge, daran kann auch ein Anfänger seine Freude haben und nebenbei einiges über den internen Befehlsaufbau des HP41 erfahren, ohne geistige Klimmzüge zu benötigen. Eine Fülle verschiedener Hilfsprogramme stehen neben den Barcodeplotprogrammen selbst, die wiederum verschiedenste Rechnerkonfigurationen berücksichtigen. In diesem Abschnitt steht allerdings nicht die Programmdokumentation im Vordergrund sondern die Vielzahl der Barcode-Möglichkeiten und deren Aufbau. Es gibt hier nämlich so einige von HP nicht dokumentierte Bar-

codefunktionen; auf deren Anwendung, Sinn und Zweck möchte ich hier nicht genauer eingehen, dies würde Bücher füllen. Als Hinweis nur zwei kurze Befehle:

```
$T+N IA _____  
eGOBEEP 77 _____
```

Mehr nicht an dieser Stelle, ihr werdet mir dies verzeihen.

Als Leckerbissen sind in diesem Kapitel eine große Anzahl synthetischer Funktionen als kurze Barcodes abgedruckt, ich habe sie nicht auf Vollständigkeit geprüft, es sind gut 100 Stück.

Da ja die XROM-Funktionen auch zu diesem Barcodekomplex zählen sind neben einer ausführlichen Behandlung dieses Typs mit samt verschiedener Herstellungsmethoden auch noch die Barcodes der Einsteckmodule

```
HP-IL  
X-Function  
TIME  
„Magnetkartenleser“  
WAND  
PLOTTER Modul  
Extended IO  
Development Modul  
CCD-Modul
```

abgedruckt, mit dem Barcodeleser hat man diese schneller eingelesen als eingetippt, gerade beim Programmieren.

Der nächste Abschnitt befaßt sich mit den Mehrbytebarcodes, darunter fallen ganze Befehlsfolgen oder Befehle mit nachfolgendem Argument wie z.B. LIST 015 oder CLP „PROGRAMM“. Auch hier wieder etliche Hilfsprogramme und Hintergrundinformationen über die Funktionsweise verschiedener Befehle und über die Programmierung nicht programmierbarer Befehle. Dann folgen wieder Programme zur Herstellung dieses Barcode-Typs mit verschiedenen Konfigurationsmöglichkeiten, was die Peripherie betrifft. Alle nicht programmierbaren (wirklich nicht) Funktionen des HP41 wie z.B. ASN oder SIZE werden einzeln und detailliert beschrieben und Barcodes für diese hergestellt, dafür gibt es allein über 20 Seiten doch recht eng bedruckte Seiten. Zum Schluß gibt es dann praktische Beispiele für diese Barcodes, da lernt man diese am besten kennen. Ein sogenanntes simuliertes synthetisches Tastenfeld als Barcodes schließt sich diesem an.

Zum Schluß sind wieder alle Programme dieses Kapitels als Barcodes abgedruckt.

5. Kapitel

Programmbarcodes und Barcodeanalyse

Wie aus der Überschrift zu erkennen ist wird hier die komplizierteste Barcodeform behandelt, ich sollte besser sagen abgehandelt, Erklärungen zum Aufbau dieses doch etwas komplizierteren Barcodes

sind fast keine vorhanden, der Autor beschränkt sich in erster Linie auf die Beschreibung der Funktionsweise seiner Programme zur Herstellung von Programmbarcodes, dies dafür etwas ausführlicher. Mir würde dies aber nicht weiterhelfen, wenn ich selbst ein Programm für die Herstellung von Programmbarcodes schreiben sollte. Mit 10 Seiten einschließlich der Programmbedienungsanleitungen ist dieser Abschnitt bei weitem zu mager für dieses umfangreiche Thema, dem Autor ist da wohl auch langsam die Lust oder die Zeit ausgegangen, so habe ich jedenfalls den Eindruck.

Mit Barcodeanalyse ist dann ein kurzes Programm zum Ausdrucken der Information einer Barcode-Reihe mittels WNDSCN bzw. WNDTST gemeint, eine kleine Utility am Rande.

6. Kapitel

Der „EAN“ und andere Barcodes

Dieses letzte Kapitel reißt kurz andere in der Industrie gängigen Barcodes an mit einer kurzen Erläuterung über deren Sinn und Zweck.

Resümé

Zu so gut wie allen Programmen sind im Buch auch die Barcodes abgedruckt, dies ist ja auch deren eigentliche Anwendung. Im großen und ganzen ist das Buch leicht verständlich geschrieben und vor allem auch Anfängern zu empfehlen.

Bis auf den Programmbarcode sind alle Barcode-Typen sehr ausführlich beschrieben, die vielen Beispielprogramme tragen das ihre zum Verständnis bei, man könnte das Buch auch als Programmsammlung für Barcodeplotterprogramme verkaufen.

Für Leute mit umgebautem Drucker ein Wermutstropfen: der Autor geht bewußt nicht auf diese Druckmöglichkeit ein, da diese nicht jedem offensteht. An dieser Stelle verweist der Autor auf den CCD als Ansprechpartner.

Ansonsten ein recht gelungenes Buch, ich wünsche allen viel Spaß damit (dies soll keine Werbung sein).

Martin Meyer (1000)
Redaktion

Hilferuf

Wie kann man Daten vom HP41CX zu einem Casio Rechner mit Bildschirm senden? Wer kann dazu Literatur nennen oder hat bereits Erfahrungen mit dieser Konfiguration?

Peter Röhl
Osterfeuerbergstraße 70
2800 Bremen 1

Die Barcodes des HP41

Barcodes, was ist das



3. Teil

Wie im letzten Heft versprochen möchte ich mich an dieser Stelle mit der Herstellung der

Datenbarcodes für den HP41

befassen, hiermit lassen sich verschiedene Daten für den HP41 als Barcode „abspeichern“ und bei Bedarf wieder in den Rechner einlesen, ich denke da an die typische Anwendung wie mobile Artikelerfassung, dies wird in der Industrie immer häufiger in der Lagerhaltung benutzt.

Eine andere typische Anwendung im Bereich der Naturwissenschaften:

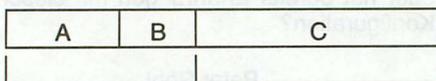
Ein Programm berechnet Sternendaten, die Konstanten der verschiedenen Himmelskörper, oft 15- und mehrstellige Zahlenmonster größerer Anzahl, befinden sich als Blatt Papier als Anlage zum Programm dabei, die Konstanten können nach Aufforderung in Sekundenbruchteilen fehlerfrei eingelesen werden, man muß sie nicht im Programm stehen haben, dies ist aus Platzgründen oft auch garnicht möglich.

Ein Schachprogramm zum Beispiel erhält die Feldeingaben des einen Spielers über ein „Barcodefeld“, ein 8*8 großes Feld mit Barcodezahlen, für jede Spielfigur steht dann daneben noch jeweils eine kleine Barcodezeile zur Verfügung, damit wird das Spiel schneller und transparenter, Fehleingaben brauchen kaum noch abgefangen werden, das spart Programmlaufzeit und Speicherplatz.

Grundsätzlich gibt es drei verschiedene Datenbarcodetypen:

1. Normale numerische Daten, d.h. Zahlen, je eine pro Zeile
2. Normale alphanumerische Daten, d.h. ASCII-Zeichen, die entweder den Inhalt des ALPHA-Registers überschreiben oder an diesen angehängt werden
3. Dasselbe wie 1. und 2., nur daß hier eine Reihenfolgenummer die richtige Lesereihenfolge bei Verwendung der Funktion WDTDTX sicherstellt, d.h. man kann z.B. 10 Zahlen nur in der richtigen Reihenfolge in 10 Speicherregister einlesen, sehr anwenderfreundlich oder auch idiotensicher, je nachdem wie man das sehen möchte

1. Normaler Datenbarcode



Kopf numerische Daten

Kopf = 12 Bit = 1½ Byte:

- A 8 Bit Prüfsumme aus B+C mit jeweils dazuaddiertem Überlauf
- B 4 Bit Typ-Indikator: 0110₂=6
- C 14½ Byte für bis zu 29 Digits, d.h. Zahlen, wie man sie auf dem HP41 eingibt, für jede Zahl ein Halbbyte oder Nibble: 0-9, „E“ und –

Für die Zahlenbarcodes gelten dabei folgende Nybblewerte:

| Digit | Nybble ₂ |
|------------|---------------------|
| 0 | 0000 |
| 1 | 0001 |
| 2 | 0010 |
| 3 | 0011 |
| 4 | 0100 |
| 5 | 0101 |
| 6 | 0110 |
| 7 | 0111 |
| 8 | 1000 |
| 9 | 1001 |
| Füller | 1010 |
| . | 1011 |
| + | 1100 |
| - | 1101 |
| E | 1110 |
| unzulässig | 1111 |

Folgende Grundregeln sind bei der Herstellung dieses Barcodetyps zu beachten:

1. Die Länge des Barcodes ist auf mindestens 3 Byte nach unten hin beschränkt, d.h. bei einer nur einstelligen Zahl müssen zwei Füller vor dieser eingebaut werden
2. Die zu kodierende Zahl muß immer „rechtsbündig“ im Barcode stehen, d.h. bei einer geraden Anzahl von Digits muß die Zahl um einen „Füller“ am Anfang der Zahl ergänzt werden, sodaß alle Barcodebytes gefüllt sind

So, genug der grauen Theorie, an Beispielen erklärt sich das Ganze wesentlich leichter:



Wie man sieht ist nur der Punkt als Dezimalpunkt zulässig, dies war ja auch schon aus der Tabelle ersichtlich. Die Zahl besteht aus 12 Digits, dabei werden natürlich alle Stellen mitgezählt, also auch der Punkt oder das Exponentialzeichen E.

Wenn wir uns jetzt den Barcode genau anschauen, so erkennen wir im 2. Byte, oberes Nybble den Typindikator 0110₂=6,

das untere Nybble ist ein Füller 1010₂=10, da die Anzahl der Digits gerade ist. Der Füller kommt immer zwischen den Typindikator und das erste Digit.

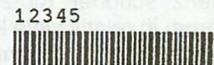
Die Werte der restlichen 12 Nybbles oder 6 Byte kann man direkt der Tabelle entnehmen, das Vorhandensein des „+“ hat wohl eher statistischen Charakter.

Das erste Byte dieses Barcodes, die Prüfsumme, errechnet sich aus der Summe der übrigen Barcodebytes mit End around Carry, das Thema hatten wir in den letzten Folgen, glaube ich, zu genüge durchgespielt, die Formel trotzdem noch einmal zum Anschauen:

$$\text{Prüfsumme} = B+C \text{ MOD } 256 + \text{INT}(B+C / 256)$$

Das MOD sorgt dafür, das die Prüfsumme nie größer als 255 wird, der 2. Term mit der INTeger-Funktion ermittelt die dazu zu addierenden Carries.

Das zweite Beispiel soll lediglich zeigen, das man auch ohne das Füll-Nybble auskommt, hier ist die Anzahl der Digits ungerade:

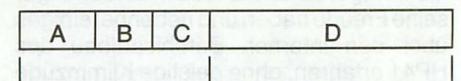


Als letztes Beispiel für diesen Barcodetyp noch eine Platzverschwendung, ein Digit in 3 Nybbles. Wie ich am Anfang schon erwähnt hatte muß der Datenbarcode immer mindestens 3 Byte haben, d.h. bei einem Digit muß ich ein ganzes überflüssiges Byte in Form von 2 Füllern zwischen den Typindikator und der Zahl einfügen, so spielt das Leben eben.

Dazu das letzte Beispiel:



2. ALPHA Barcode



Kopf ASCII-Zeichen

Kopf = 16 Bit = 2 Byte:

- A 8 Bit Prüfsumme aus B+C+D mit jeweils dazuaddiertem Überlauf
- B 4 Bit Typ-Indikator: 0111₂=7 für Überschreiben des AL-

PHA-Registers mit den ASCII-Zeichen
 $1000_2=8$ für das Anhängen des ASCII-Zeichen an den ALPHA-Register Inhalt

C 4 Bit für die Anzahl der folgenden ASCII-Zeichen

D 14 Byte ASCII-Zeichen, d.h. je 8 Bit für jedes Zeichen

Hier brauchen wir keine Übersetzungstabelle für die Buchstaben, sie werden genommen wie sie sind. Füller gibt es ebenso keine, es bleibt also kaum noch etwas zu erklären, zwei Beispiele sollen aber etwas Klarheit bringen. Grundsätzlich gibt es ja zwei verschiedene ALPHA-Barcodes, der normale, er überschreibt den Inhalt des ALPHA-Registers und den APPEND-ALPHA-Barcode, sein Text wird an den bestehenden ALPHA-Register Inhalt angehängt.

Zunächst einmal ein Beispiel für den normalen ALPHA-Barcode:

"ABCDE"


Der Typindikator, das 3.Nybble ist hier $0111_2=7$, das 4. Nybble ist 5, es repräsentiert die Anzahl der nachfolgenden Buchstaben bzw. Bytes. Die nachfolgenden Barcodebytes entsprechen direkt dem ASCII-Code des gewünschten Zeichens. Das erste Byte ist wieder die Prüfsumme des gesamten restlichen Barcodes.:

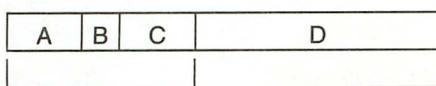
Prüfsumme =
 $B+C+D \text{ MOD } 256 + \text{INT}(B+C+D / 256)$

Noch kurz ein Beispiel zum APPEND-ALPHA-Barcode:

APPEND "TEST abcd"


Der Typindikator, das 3.Nybble ist hier $1000_2=8$, das 4. Nybble ist 9, es repräsentiert die Anzahl der nachfolgenden Buchstaben bzw. Bytes. Die Prüfsummenbildung für das erste Byte funktioniert wie oben schon erwähnt, alles also halb so kompliziert.

3. Sequentieller Datenbarcode



Kopf

Daten

Kopf = 24 Bit = 3 Byte:

A 8 Bit Prüfsumme aus B+C+D mit jeweils dazuaddiertem Überlauf

B 4 Bit Typ-Indikator:
 $1001_2=9$ für numerische Daten im selben Format wie unter 1. beschrieben

$1010_2=10$ für ASCII-Zeichen, die das ALPHA-Register überschreiben

$1011_2=11$ für ASCII-Zeichen, die an den Inhalt des ALPHA-Register angehängt werden sollen

C 12 Bit fortlaufende Nummer zum Verhindern von Einlesen in der falschen Reihenfolge

D 13 Byte für bis zu 13 ASCII-Zeichen oder bis zu 26 Digits oder Zahlen

Die Kodierung des sequentiellen Barcodes ist grundsätzlich dieselbe wie wir sie schon vom normalen Datenbarcode her kennen, neu ist dabei lediglich eine Folgenummer, die das richtige Einlesen der Daten in die Register sicherstellen soll, wie, das sehen wir gleich, dazu gleich das nächste Beispiel:

12.5 Nummer 0


8 Nummer 1


125E-88 Nummer 2


Wir tippen jetzt auf unserem HP41 .002 XEQ ALPHA WND DTX ALPHA, der Rechner fordert uns jetzt mit der Anzeige „W: SC TO 00“ zum Einlesen der Datenbarcodezeile mit der Folgenummer 0, also der ersten Zeile auf, dies tun wir. Nun erscheint „W: SC TO 01“, wir lesen die zweite Zeile ein, ebenso die dritte und letzte, danach erscheint wieder das normale Rechnerdisplay. Schaut man sich jetzt die Registerinhalte 0-2 an, so findet man hier die eingelesenen Daten, alles in Ordnung.

Der Aufbau des sequentiellen Barcodes beginnt wie immer mit der Prüfsumme, der Summe der restlichen Barcodebytes der Zeile, im Gegensatz zum Programmbarcode wird hier immer nur die Prüfsumme der entsprechenden Zeile berechnet, die der vorhergehenden Zeilen werden nicht mit berücksichtigt, das spart uns einige Programmierarbeit.

Das nächste Nybble beinhaltet den gewohnten Typindikator, er ist hier $1001_2=9$, es folgen 3 Nybbles für die Folgenummer. Diese könnten dementsprechend Zahlen bis 4095 enthalten, es gibt aber nur 319 Register, in die wir einlesen können, 2 Nybbles hätten aber nicht gereicht, die reichen nur bis 255! Das Zählen der Folgenummer erfolgt direkt binär ohne irgendeine Kodierung.

In der ersten Zeile haben wir eine Zahl mit 4 Digits, wir benötigen also keinen Füller, dafür aber in der zweiten Zeile, in

der dritten ebenso, wir müssen ja immer irgendwie die Bytes füllen, so ist das eben.

So, wenden wir uns dem nächsten Beispiel zu, jetzt ist der sequentielle ALPHA-Datenbarcode dran:

"ZEILE1" Nummer 0


"ZEILE2" Nummer 1


" " Nummer 2


"Ende" Nummer 3


Wir tippen jetzt auf unserem HP41 .003 XEQ ALPHA WND DTX ALPHA, der Rechner fordert uns jetzt mit der Anzeige „W: SC TO 00“ zum Einlesen der Datenbarcodezeile mit der Folgenummer 0, also der ersten Zeile auf, dies tun wir wieder. Die übrigen drei Zeilen lesen wir auf die gewohnte Art und Weise ein, bis wieder das normale Rechnerdisplay erscheint.

In den Registern 0-3 finden wir jetzt die eingelesenen ALPHA-Daten. Wenn wir uns den Barcode genauer ansehen, so fällt uns auf, das die Anzahl der folgenden ALPHA-Zeichen im Barcode fehlt, die holt sich der Rechner aus der Gesamtzahl der Barcodebytes, warum nicht gleich so?

Da ein Buchstabe immer ein ganzes Byte füllt gibt es hier keine Probleme mit Füllern, sogar ein String mit 0 ALPHA-Zeichen ist möglich. Jetzt werden sich Einige zu Recht fragen, wozu gibt es beim Typindikator einen für normalen und einen für APPEND-ALPHA-Barcode. Dies hat trotzdem seine Bewandtnis, die Funktion WND DTX ignoriert diese Unterscheidung natürlich. Man kann die sequentiellen Barcodezeilen auch ganz normal in das X-Register bei normalen Daten und in das ALPHA-Register bei ALPHA-Daten einlesen, hier kann man die Unterbrechung wiederum gebrauchen, eine Anwendung dafür habe ich aber bislang nicht finden können, sei's dem so.

Schauen wir uns jetzt noch ein letztes Beispiel des sequentiellen ALPHA-Barcodes an:

"1.Zeile" Nummer 5


"2.Zeile" Nummer 7


"Echo" Nummer 9


" " Nummer 10


Hier habe ich nicht mit der Folgenummer 0 angefangen, wie man sehen kann. Wir geben jetzt 5.010 XEQ ALPHA WND DTX ALPHA ein und versuchen die Zeile mit der Nummer 5 zu lesen, in der Anzeige steht ja „W: SC TO 05“. Dies beantwortet der Rechner nur mit der Fehlermeldung „W: SEQ ERR“, da die Höhe der Folgenummer nichts mit dem Zielregister zu tun hat, dies gibt einzig und allein das Argument des Befehls WND DTX an!

Mit einem Trick kann man trotzdem die Daten einlesen:

.010 XEQ ALPHA WND DTX ALPHA eingeben, die Taste SST sooft drücken, bis „W: SC TO 05“ in der Anzeige erscheint, jetzt kann man die erste Zeile einlesen; es erscheint aber nicht „W: SC TO 06“ in der Anzeige, sondern „W: SC TO 07“.

Dafür gibt es eine ganz einfache Erklärung, die da lautet: zählt die eingelesenen Buchstaben. Jaja, das sind 7, wie man nach kurzem Nachdenken merken sollte

gehen ja eigentlich nur 6 ALPHA-Zeichen in ein Speicherregister des HP41, da das 7. Byte des Speicherregisters für die Kodierung der Zeichenanzahl benutzt wird, damit kann er es ja auch von Zahlen unterscheiden.

Da also nur 6 Zeichen in ein Register passen, wo ist dann das 7. hin, in's nächste natürlich. D.h. man kann bei ALPHA-Daten gleichzeitig in mehrere Register einlesen, 13 Zeichen gehen in eine Zeile, das wären also 2 1/6 Register.

Wenn wir jetzt die nächste Zeile einlesen, die mit der Folgenummer 7, dann werden wir dasselbe Spiel noch einmal erleben, der Rechner verlangt wieder die übernächste Zeile, dafür ist aber die Zeile mit „Echo“ dann wieder richtig lang.

(Die ersten Barcodeleser mit der Version 1E beinhalteten noch keine Funktion WND DTX, sie ist erst ab der Version 1F implementiert.

Die Versionsnummer der Barcodeleser kann man mit der Funktion CAT 2 feststellen, in der Anzeige erscheint dann „- WAND x x -“, wobei xx die Versionsnummer ist).

So, das war's zu diesem Barcodetyp, es bleibt nur noch der direkt ausführbare Funktionsbarcode, hier werden ganze Befehlsfolgen in eine Barcodezeile gepackt. Das Thema werden wir uns dann zu guter Letzt das nächste Mal zu Gemüte führen.

Quelle:
Hewlett Packard
„Greating Your Own HP-41 Bar Code“
Manual, March 1981

Happy Programming,

Martin Meyer (1000)
Redaktion

Dreisatz

DREISÄTZE

3SATZ
K3SATZ

Dreisatz, proportional

A B
C D

Wo die Unbekannte (der gefragte Wert) steht ist gleich. Dafür wird 0 eingegeben.

Die Eingabe der Daten kann auf die unten angeführten acht Möglichkeiten erfolgen, die über dem Doppelstrich stehen. Die Ziffern bezeichnen die Prioritäten der Eingaben.

Natürlich kann man den Dreisatz auch als reine Proportion schreiben A:B=C:D. Die Eingaben von links nach rechts oder von rechts nach links.

Dreisatz, verkehrt proportional

Auch hier kann das obige Programm Verwendung finden, jedoch gibt es nur vier Möglichkeiten der Eingabe (Schema unter dem Doppelstrich)

Kettendreisatz

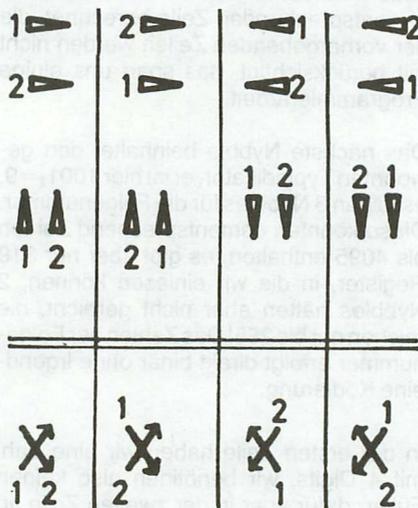
gegeben (A..B..C.. Z)....SINGLE
gesucht (a..b..c.. z).... zu berechnen

Nach Eingabe von SINGLE= wird die obere Reihe in beliebiger Reihenfolge eingegeben. Dabei wird die Anzahl der Posten gespeichert. Wenn keine weitere Eingabe mehr erfolgen soll, dann wird nach Abfrage R/S gedrückt. Eingabe der unteren Reihe.

Das Ergebnis erscheint im Display.

```
01+LBL "3SATZ"
02 "A="
03 PROMPT
04 "B="
05 PROMPT
06 "C="
07 PROMPT
08 "D="
09 PROMPT
10 X<>Y
11 X#0?
12 RDN
13 X#0?
14 RDN
15 X#0?
16 RDN
17 RDN
18 /
19 /
20 .END.
```

```
01+LBL "K3SATZ"
02 CLX
03 STO 00
04 "SINGLE="
05 PROMPT
06+LBL 00
07 CF 22
08 "GEGEBEN="
09 PROMPT
10 FC? 22
11 GTO 01
12 /
13 ISG 00
14 GTO 00
15 GTO 00
16+LBL 01
17 CF 22
18 "GESUCHT="
19 PROMPT
20 *
21 DSE 00
22 GTO 01
23 .END.
```



Martin Hochenegger
Heidelberger Landstraße 97
6100 Darmstadt 13

Bug in der RAMBOX

Ich möchte hier auf einen kürzlich entdeckten Fehler der Funktion CLLSTFL im RAMBOX-Betriebssystem hinweisen:

Arbeitet man intensiv mit der RAMBOX, tritt zwangsläufig auch der Wunsch nach Änderung bereits gespeicherter Programme auf. Dies wurde von W&W vorhergesehen und durch die Funktion CLLSTFL (Clear Last File) auch ermöglicht.

Leider enthält die Funktion CLLSTFL einen Fehler, der sich für Assembler-Programmierer fatal auswirken kann:

Jeder Aufruf von CLLSTFL kopiert den gesamten Inhalt der angesprochenen Page um ein Wort nach höheren Adressen! Dabei wird unmittelbar vor dem ROM-Header (in der Regel die erste „Funktion“ der Page) ein RTN (3E0) eingefügt. Glücklicherweise werden alle FAT-Einträge um eben dieses eine Wort

korrigiert, sodaß sich für die meisten Anwendungen keine sichtbare Veränderung ergibt.

Der Fehler tritt bei allen mir zur Verfügung stehenden Versionen des Betriebssystems auf (bis inkl. Rev 3H).

Ganz anders sieht die Sache aus, wenn in dieser Page Maschinenspracheprogramme stehen, die absolute Sprünge enthalten. Alle diese Sprungbefehle zeigen jetzt an einen falschen Ort! Daß sich dabei höchst merkwürdige Dinge ereignen können, ist sicher allen klar.

Eine Analyse der Funktion CLLSTFL zeigte, daß die Funktion viel zu „fleißig“ ist. Ein versuchsweise eingesetztes NC GO NFRPU (=Programmende) an einer Stelle, an der:

- die Adresse des neuen freien Speicherplatzes berechnet und gespeichert ist

- die letzten zwei FAT-Worte gelöscht
- und die Funktionsanzahl korrigiert sind

bewirkte tadelloses Arbeiten der Funktion bei einem Bruchteil der bisher notwendigen Zeit! Die Meldung CLEARING blitzt jetzt nur noch kurz auf, dann ist alles schon vorbei.

Warum das Betriebssystem ab dieser Stelle noch die ganze FAT „durchkaut“ und den übrigen Inhalt der Page verschiebt, ist mir schleierhaft, zumal kein effektiver Löschvorgang stattfindet. Sämtliche Informationen sind immer noch vorhanden, lediglich die Zeiger darauf wurden beseitigt.

Es wird doch wohl nicht dazu dienen, daß die Meldung CLEARING lange genug in der Anzeige steht, daß man sie auch lesen kann????

Armin Jakob
Obere Felsenstraße 10
CH-8000 St. Gallen

Die Barcodes von BEST OF PRISMA

Im nachfolgenden sind die Barcodes zu Programmen aus BEST OF PRISMA aufgelistet, die ab sofort als Barcodes in der Redaktion bei mir erhältlich sind. Die Barcodes sind nicht immer in optimaler Kombination über die Seiten verteilt, das bringen die Barcodes so mit sich. Sie wurden einmal verkleinert, die Qualität sollte aber den Ansprüchen des WAND genügen.

In der PRISMA-Ausgabe JUNI 1987 war die Liste derjenigen Programme abgedruckt, die auf der von mir zusammengestellten Kassette von BEST OF PRISMA zu finden waren; diese Kassette bildete auch die Grundlage zu der Herstellung der jetzt vorliegenden Barcodes, mehr als das habe ich nicht. Tastenzuweisungsbarcode gibt es leider nicht.

Bezug:

Brief an die Redaktion (Postfach !!) mit der Wunschliste der benötigten Seiten, je 2 Seiten kommen auf ein kopiertes Blatt, eine vorne, eine hinten, dies spart Porto und Papier.

Pro zu kopierende Seite legt bitte 40 Pf. in Briefmarken bei und 2,40 DM für das Verschicken.

So, nun aber die Liste der wünschbaren Programme:

| Progr.-name | Seite | Progr.-name | Seite |
|-------------|-------|-------------|-------|
| ABARP | 2 | MSDAT1 | * |
| ABARPLS | 1-2 | MSDAT2 | * |
| ACXR | 1 | MSMIMA | 18 |
| AD | 31 | MT111 | 20 |
| ASL | 18 | MTADD | 20 |
| AUXILY | 39 | MTAIJ | 20 |
| B3 | 10 | MTAUS | 20 |
| BARA | 3 | MTEIN | 20 |
| BB | 29-30 | MTINV | 21 |
| BK | 14 | MTMUL | 21 |
| BP | 3 | MTSHF | 20 |
| BR | 15 | MTSPU | 21 |
| BSORT | 33 | MTTRA | 21 |
| BU | 12 | NACH | 36 |

| | | | |
|---------|-------|---------|--|
| CAT | 9 | P | 13 |
| CD | 10 | P2 | 39 |
| CHANGE | 36 | PH | 39 |
| DA | 35 | PHDAT | *(1 Seite) |
| DC | 10 | PL | 23 |
| DD | 24 | PLOT | 23 |
| DEMO | 10 | PLOT2 | 28 |
| DERBY | 26-27 | PM | 22 |
| DIFFGL | 28 | POWER | 40-42 |
| DIST | 37 | POWERGK | } kein Barcode mögl., } Tastenzuweisungen |
| DREIECK | 22 | POWERNS | |
| DRST | 29 | PRF | 13 |
| EINH | 36 | PRM | 16 |
| ENTF | 37 | PT | 31 |
| EUKLID | 15 | Q5X | 4 |
| FER | 15 | Q5XDAT | 4-8 |
| FN | 39 | RDM | 17 |
| FOUR | 18 | RF | 27 |
| GAMMA | 17 | RMU | 36 |
| GLS | 19 | ROOT | 13 |
| GQ | 17 | RUKUN | 17 |
| HAUPT | 20 | SFRAGEZ | 1 |
| HP | 30 | SIGMAPS | 10 |
| IM | 30 | SNOOPY | 9 |
| INFOUR | 19 | SUM | 36 |
| KM | 37 | T | 13 |
| KMKO | 37 | T1 | 10 |
| L | 15 | T2 | 10 |
| LB | 11 | T3 | 10 |
| LINF | 21 | TANKEN | 36 |
| LOGIC | 33 | TIMER | 37 |
| LST | 34-35 | UMR | 36 |
| MANT | 1 | URLAUB | 37 |
| MK | 11-12 | VEKTOR | 14 |
| MPLOT | 25 | WECHS | 36 |
| MS | 31-32 | | |

* Die Seitennummer war bei Redaktionsschluß noch nicht bekannt.

Martin Meyer (1000)
Redaktion

Prüfsummen für Barcodes

In Prisma 87.6.37-39 beschreibt Martin Meyer, wie man die Prüfsumme bei 2-Byte Barcodes ermittelt.

Für die Berechnung der Prüfsumme gibt es zwei sehr elegante Wege, die auch in allen Barcodeplotprogrammen in BEST OF PRISMA verwendet werden. Dazu benenne ich – abweichend zu seiner Definition – zunächst die 4 Halbbytes mit den Buchstaben A-D:

- A enthält die Prüfsumme und bildet mit B das erste Byte des Barcodes
- C+D sind entsprechend das zweite Byte des Barcodes

Liegen nun die Halbbytes A-D als Dezimalzahlen vor, so ist die Prüfsumme

$$A = (B + C + D) \text{ MOD } 15$$

Als Programm:

```
B
C
+
D
+
15
MOD
```

Ist das zweite Byte des Barcodes noch nicht in die Halbbytes C und D zerlegt (es gilt: 2.Byte = C 16 + D), so gilt ebenfalls:

$$A = (B + 2.\text{Byte}) \text{ MOD } 15$$

Als Programm:

```
B
(2. Byte)
+
15
MOD
```

Dabei sind B, C und D Zahlen zwischen 0 und 15, das 2. Byte liegt zwischen 0 und 255 und aus Berechnungen ergibt sich, daß A nur zwischen 0 und 14 liegen kann, nie aber den Wert 15 (binär 1111) annehmen kann.

Ähnlich einfach ist die Berechnung der Prüfsumme bei Datenbarcodes und direkten Ausführungsbarcodes mit mehr als 2 Byte (z.B. für ein komplettes ASN) usw., also den einzeiligen Barcodes, bei denen die Prüfsumme volle 8 Bit = 1 Byte umfaßt. Hier werden wieder alle Bytes außer der Prüfsumme (zwischen 0 und 255 dezimal) in beliebiger Reihenfolge aufaddiert und dann die Summe MOD 255 bestimmt.

Als Programm:

```
.
+
255
MOD
```

Bei den Programmbarcodes kommt außerdem noch zu den Bytes in der Zeile die Prüfsumme der **letzten** Zeile hinzu; so wird verhindert, daß die Barcodes verschiedener Programme gemischt eingelesen werden können. Auch hier kann die Prüfsumme nie den Wert 255 (binär 1111 1111) annehmen.

Ralf Pfeifer
Rubensstraße 5
5000 Köln 50

XROM

Programm zur Ermittlung aller verwendeten XROM Funktionen eines Programms

124 Zeilen, 203 Bytes, 2 Regs, HP-41C

Funktionsweise:

Das zu untersuchende Programm muß im Hauptspeicher stehen und sollte gepackt sein.

Es werden alle Programmbefehle auf ihre Länge und Art untersucht. Wird ein XROM-Befehl gefunden, so wird er zusammen mit der Zeilennummer, in der er steht, ausgedruckt.

Da in zunehmendem Rahmen Module mit gleichen XROM-Nummern auf den Markt kommen, ist es technisch nicht möglich, nur aus den XROM-Nummern heraus auf das jeweils verwendete Modul zu schließen; es sei denn, der Programmierer selbst besitzt die Gnade, das Programm zu dokumentieren!

Das Programm läuft auf jedem HP-41 mit CCD-Modul (Drucker empfehlenswert, oder eventuell Flag 21 setzen)

- 01*LBL „XROM“
- 02 CLX Zeilennummern initialisieren
- 03 STO 00
- 04 SF 27 USER ein (Kleinschreibemodus ausschalten)
- 05 „PRGM? “
- 06 PMTA Programmnamen abfragen

- 07 PHD Adresse des Programmmanfangs
- 08 PPLNG Programmlänge in Bytes
- 09 CHS
- 10 A+B Adresse des Programmendes berechnen
- 11 STO 01
- 12 PHD Programmmanfang
- 13*LBL 01 **Schleifenbeginn**
- 14 ISG 00 Zeilennummer erhöhen
- 15*LBL 00 „NOP“ damit ISG keine Komplikationen macht
- 16 PEEKB Programmbyte lesen
- 17 16
- 18 – Byte aus Zeile 1? (0 bis 15)
- 19 X<0?
- 20 GTO 02 – ja, > 1-Bytebefehl
- 21 13 – nein
- 22 – Zahleneingabe? (16 bis 28)
- 23 X<0?
- 24 GTO 11 – ja, > Spezialbehandlung
- 25 CF 00 – nein, Zahleneingabe fertig, Flag 00 löschen
- 26 3
- 27 – Multibyte? (29 bis 31)
- 28 X<0?
- 29 GTO 04 – ja, > Multibyte-Behandlung
- 30 112 – nein
- 31 – Zeilen 2 bis 8? (32 bis 143)

| | | | |
|-------------|--|------------|---|
| 32 X<0? | | 85 LBL 05 | 2-Bytebefehle |
| 33 GTO 02 | - ja, > 1-Bytebefehl | 86 RDN | |
| 34 16 | - nein | 87 1 | Schleifensteuerung |
| 35 - | Zeile 9? (144 bis 159) | 88 GTO 08 | |
| 36 X<0? | | 89*LBL 06 | GLOBAL-Befehle (3. Byte gibt Länge an) |
| 37 GTO 05 | - ja, > 2-Bytebefehl | 90 RDN | |
| 38 7 | - nein | 91 A- | Adresse auf 2. Byte |
| 39 X<Y? | XROM-Befehl? (160 bis 167) | 92 GTO 09 | |
| 40 GTO 03 | - nein | 93*LBL 07 | 3-Bytebefehle |
| 41 RDN | - ja | 94 RDN | |
| 42 4 | aus erstem Byte die XROM Nr. berechnen | 95 2 | Schleifensteuerung |
| 43 * | | 96 GTO 08 | |
| 4 X<>Y | | 97*LBL 04 | Multibyte (2. Byte gibt Länge an) |
| 45 A- | Adresse des 2. Bytes | 98 RDN | |
| 46 PEEKB | 2. Funktionsbyte lesen | 99* 09 | |
| 47 64 | | 100 A- | Adresse 1 Byte tiefer |
| 48 / | welche der 4 Möglichkeiten ist es? | 101 PEEKB | Byte holen |
| 49 INT | (0..63 64..127 128..191 192..255) | 102 240 | |
| 50 ST+ Z | zur XROM Nr. addieren | 103 - | Länge berechnen (wird als Steuerzahl verwendet) |
| 51 CLA | | 104 LBL 08 | |
| 52 FIX 0 | | 105 X<>Y | Steuerzahl und Adresse vertauschen |
| 53 ARCL 00 | Zeilennummer ins ALPHA-Register | 106*LBL 10 | |
| 54 FIX 2 | | 107 A- | Adresse erniedrigen |
| 55 „I-XROM“ | Append „XROM“ | 108 DSE Y | noch mehr? |
| 56 RDN | | 109 GTO 10 | - ja |
| 57 LASTX | aus dem zweiten Byte die Funktionsnummer berechnen | 110 X<>Y | - nein, Steuerzahl und Adresse tauschen weiter |
| 58 FRC | | 111 GTO 02 | |
| 59 .64 | (Funktionsnummer = 0 bis 63) | 112*LBL 11 | Zahleingaben (mehrere 1-Bytebefehle in Zeile) |
| 60 * | | 113 FS? 00 | letztes Byte auch schon Zahl? |
| 61 RCL Z | XROM Nr. abrufen | 114 DSE 00 | - ja, > Zeilennummer wieder erniedrigen |
| 62 + | zur Funktionsnummer addieren | 115 SF 00 | Flag 00 setzen, signalisiert Zahleingabe |
| 63 ARCL X | ins ALPHA-Register laden | 116*LBL 02 | nächste Byteadresse berechnen |
| 64 AVIEW | anzeigen/drucken | 117 X<>Y | |
| 65 GTO 02 | | 118 A- | Adresse des Programmendes |
| 66*LBL 03 | hier weiter, wenn kein XROM-Befehl | 119 RCL 01 | |
| 67 - | | 120 X<>Y | noch nicht fertig? |
| 68 25 | | 121 X>Y? | - ja, > weiter in der Schleife |
| 69 - | Rest Zeile A und Zeile B? (168 bis 191) | 122 GTO 01 | fertig, Anzeige löschen |
| 70 X<0? | | 123 CLD | |
| 71 GTO 05 | - ja, > 2-Bytebefehl | 124 END | |
| 72 14 | | | |
| 73 - | GLOBAL-Befehl? (192 bis 205) | | |
| 74 X<0? | | | |
| 75 GTO 06 | - ja, > GLOBAL | | |
| 76 2 | | | |
| 77 - | letzte 2-Bytebefehle? (206 bis 207) | | |
| 78 X<0? | | | |
| 79 GTO 05 | - ja | | |
| 80 32 | | | |
| 81 - | Zeilen D und E? (208 bis 239) | | |
| 82 X<0? | | | |
| 83 GTO 07 | - ja, > 3-Bytebefehle | | |
| 84 GTO 08 | - nein, > TEXT | | |

Armin Jakob
Obere Felsenstraße 10
CH-9000 St. Gallen

Aufruf

Im vorletzten Heft hatte ich zum Schreiben von Utilities für den HP41 und seinen Peripheriegeräten aufgerufen, ebenso zum Schreiben eines Programms zur Ermittlung der von einem Programm benötigten Einsteckmodule. Für dieses Problem erreichten bislang zwei die Redaktion, die recht brauchbar erschienen.

Das kann aber nicht alles sein, eine Lösung ganz ohne das CCD-Modul wäre im Prinzip noch wünschenswert für diejenigen, denen dieses doch recht nützliche Teil eben nicht zur Verfügung steht.

Unter Utilities können sich wohl einige

nicht so recht etwas vorstellen, darum an dieser Stelle ein Beispiel:

Ich habe den HP-IL Konverter HP82166A, man kann diesem aber von außen nichts ansehen, d.h. welche Register sind wie gesetzt oder was haben bestimmte I0-Leitungen für Pegel, soweit man auf diese unbeschadet zugreifen kann. Jetzt habe ich irgendetwas mit diesem Interface programmiert, bin mir aber über dessen Status nicht ganz im klaren, ich könnte also jetzt ein kleines Programm gebrauchen, das mir die Registerinhalte entweder auf einem Drucker oder auf ein Video-Interface oder auf das Display

ausgeben, ohne den Status des Konverters dabei zu verändern!

Hier muß man sich schon ein wenig mit der IL-Schleife auseinandersetzen, mit dem Konverter sowieso.

So, damit jetzt keiner meint, dies eben sagte beziehe sich nur auf dieses Teil, damit sind alle Peripheriegeräte gemeint, die über die IL-Schleife erreichbar sind!

Also Leute, an die Arbeit, wir wollen doch mal sehen, was in unserem Club noch drinsteckt, PRISMA ist ja nicht das Sprachrohr der Redaktion sondern seiner Leser, die sollten es ja eigentlich aktiv gestalten, also ran . . .

Martin Meyer

XROMS

110 Zeilen, 281 Bytes, HP-41C

Auf die Anfrage hin in PRISMA 1/88, S.37 entstand dieses Programm, welches XROM-Codes lesbar macht. Das vorliegende Programm benötigt das CCD-Modul. Die Liste der Peripheriegeräte kann nach eigenem Bedarf angepaßt werden. Das Programm kann bestimmt noch optimiert werden, vor allem die Geschwindigkeit. Hier sind wieder mal die M-Code-Experten gefordert. Trotzdem tut das Programm das, was es soll.

An dieser Stelle möchte ich nochmal eine bitte an die M-Code-Programmierer richten, die ich bereits bei der Mitglieder-versammlung geäußert habe: Wer schreibt mal ein kurze Einführung in die Maschinensprache und gibt Beispiele dazu, die sehr prägnant sein sollten, dabei nicht zu lang und übersichtlich?

Wer kennt eine Bezugsquelle für das kommentierte Betriebssystem des HP-41 CV?

Programmbeschreibung

Zeile 01...05: Einstellung der WSIZE auf 32 und Abfrage, welches Programm untersucht werden soll.

Z. 06...13: Programmumfang und -enden feststellen

```

01*LBL "XROMS"      30 RCL 00
02 32                31 A-
03 WSIZE            32 STO 00
04 "XROM, PRGM?"   33 RCL 01
05 PMTA             34 X<Y?
06 PHD              35 GTO 40
07 STO 00           36 "READY"
08 PPLNG            37 BEEP
09 3                 38 PROMPT
10 -                 39 RTN
11 CHS              40*LBL 41
12 A+B              41 RDN
13 STO 01           42 7
14 CLX              43 AND
15 STO 02           44 4
16*LBL 40           45 *
17 RCL 00           46 RCL 00
18 PEEKB            47 A-
19 RCL X            48 STO 00
20 160              49 PEEKB
21 -                 50 64
22 8                 51 /
23 /                 52 INT
24 X<0?             53 R↑
25 GTO 42           54 +
26 INT              55 RCL 02
27 X=0?             56 X<Y
28 GTO 41           57 BS?
29*LBL 42           58 GTO 42
    
```

```

Z. 14, 15:         Flagspeicher R02 löschen.
Z. 16...28:        XROM-Code? Falls nein,
                    weiter bei LBL 42, sonst
                    LBL 41.
Z. 29...39:        Nächste Adresse bereit-
                    stellen und Rücksprung
                    auf LBL 40, wenn END
                    noch nicht erreicht ist.
                    Sonst Meldung „READY“
                    und Programmende.
Z. 40...54:        Bit 3 ausblenden, mit 4
                    multiplizieren. Nächste By-
                    te des XROM-Codes hol-
                    en, davon die vorderen
                    beiden Bits addieren.
Z. 55...58:        XROM schon gespeichert?
                    Fall ja, Rücksprung auf
                    LBL 42.
Z. 59...73:        Anzeige des Peripherie-
                    moduls und weiter bei LBL
                    42.
Z. 74...110:       Unter der XROM- bzw.
                    LBL-Nr. ist die Bezeich-
                    nung der Peripheriegeräte
                    gespeichert.
                    Die Liste kann je nach
                    Bedarf abgewandelt wer-
                    den.
    
```

Wenn XROM's mit gleicher Nummer vorliegen, muß man entweder beide (evtl. sogar mehrere) angeben oder sich für eines entscheiden.

```

59 LASTX           88 RTN
60 Sb              89*LBL 09
61 STO 02          90*LBL 11
62 LASTX           91 "FCCD-ROM"
63 "XR"           92 RTN
64 E1              93*LBL 15
65 X>Y?           94 "FPRGM-RAM A"
66 "F0"           95 RTN
67 RDN            96*LBL 25
68 ARCLI          97 "F-EXT FCH"
69 "F:"           98 RTN
70 XEQ IND X      99*LBL 26
71 TONE 7         100 "F-TIME-"
72 AVIEW          101 RTN
73 GTO 42         102*LBL 28
74*LBL 01         103 "FHP-IL"
75 "FMATH"        104 RTN
76 RTN            105*LBL 29
77*LBL 02         106 "F-PRINTER"
78 "FSTAT"        107 RTN
79 RTN            108*LBL 30
80*LBL 04         109 "F-CRD RDR"
81 "FES RSU-1A"  110 END
82 RTN
83*LBL 06
84 "FES RSU-2A"
85 RTN
86*LBL 10
87 "FESMLDL-0S"
    
```

Walter Lutz (1659)
Günthersburgallee 66
6000 Frankfurt 60

Die Hummel verstummt

Peter Röhl antwortet auf den Hilferuf von Andre Gerads aus Prisma 6/87.

Bei eingeschaltetem 41er die Löschtaste ← drücken und gedrückt halten. Dann dreimal bei gedrückter Löschtaste langsam „ON“ Taste drücken und wieder loslassen. Nach etwa einer Sekunde die Löschtaste loslassen. Dieser oben beschriebene Vorgang muß mindestens drei Sekunden dauern. Wer es zu schnell macht, kann damit rechnen, daß die Bemühungen ohne Erfolg bleiben. Der HP41 CX (beim HP41CV scheint es nicht zu funktionieren) unterbricht bei dieser Behandlung alle Operationen, ohne dabei abzustürzen. Läßt man z.B. wie im Wickes beschrieben den Tangens einer NNZ (Nicht normalisierte Zahl) berechnen, so läßt sich die Rechnung durch diese Methode abbrechen. Bei einem Packing sollte man diesen „Radikalstop“ aber vermeiden, da sonst der CAT1 durcheinander gebracht werden kann. Je nachdem, wie tief der HP41CX „in sich versunken“ ist, kann es auch schon genügen, nur einmal „ON“ zu drücken. Sobald der Rechner anhält erscheint „CLX“; ansonsten bleibt das Display leer. (Auch „USER“, „PRGM“, „0“, sind in diesem Fall nicht zu sehen) Ich benutze diese Eigenart meines HP41CX (mit der Seriennummer 2612521177) um folgenschwere Operationen jenseits des von HP vorgesehenen Betätigungsfeldes in letzter Sekunde abwenden zu können. Es funktioniert – solange noch rechtzeitig – in jedem Fall.

Tastenzuweisungen tätige ich mit folgender Routine:

```

01 LBL Tz
02 , (Komma)
03 SF25
04 X<>C
05 RCLM
06 STO IND Z
07 X<>Y
08 X<>C
09 CLA
10 FC?C25
11 AVIEW
12 END
    
```

Im Alpha Register muß dabei der CODE für die Tastenzuweisung und im X Register die Registeradresse liegen. Es sollten immer zwei Tasten zusammen belegt werden. Und natürlich müssen die Tasten vor Ausführung des Programmes „Z“ mit beliebigen Funktionen belegt sein.

Das X Register darf nicht „2“ enthalten, da sonst garantiert „MEMORY LOST“ auftaucht. Mit dem Programm können beliebige Register „vollgepackt“ werden.

Peter Röhl
Osterfeuerbergstr. 70
2800 Bremen 1

Banner

140 Zeilen, 322 Bytes, SIZE 134,
HP-41CV, IL, PRINTER

HP41 – PRGM „GROB“
Querschrift – Riesen

Hallo, Freunde!

Heute möchte ich einmal ein Programm vorlegen, das man nicht unbedingt haben muß und mit dem man auch nicht jeden Tag arbeiten wird, das aber hin und wieder ganz nützlich sein kann.

Vor einigen Jahren war die Zeitschrift des US-Clubs PPC mitunter voll von sogenannten Banner-Programmen, gemeint waren übergroße Schriftzeichen in Querschrift. Sie waren jedoch wenig ansehnlich und beschränkt auf das große Alphabet. PRISMA war da schon besser dran, und im BEST OF PRISMA wurde das Querschrift-Programm von Kötz und Maschke wiederholt, und auf das von Hagen Klemp hingewiesen.

Viel Neues kann ich also nicht bringen. Dennoch: meine Buchstaben und Zeichen sind höher, verzichten auf unnötige Zwischenräume, lassen sich positiv und negativ aufzeichnen, machen eine automatische Textwiederholung möglich,

wenn der Text nicht mehr als 24 Zeilen umfaßt, und sehen für den Anschluß eines Textes an einen vorhergehenden einen nahtlosen Übergang, also mit unverändertem Zwischenraum vor. Wiedergegeben werden können alle 127 Zeichen des ASCII-Codes, wobei – auf Anforderung – der Text entweder mittels des CCD-Moduls oder der XTOA-Funktion ins ALPHA-Register zu schreiben ist.

Benötigt wird natürlich der Drucker und wegen der auch im Programm vorkommenden XTOA- und ATOX-Funktionen das X/F-Modul. Das Programm umfaßt 322 Bytes auf 3 Karten-Spuren, benötigt SIZE 134 für die Ablage der Zeichen-Codes und 8 Spuren für die Aufzeichnung der 127 Daten-Register. Bemerkenswert ist, daß die Codes der Spaltenwerte der einzelnen Zeichen um je eine Einheit erhöht wurden; dann kommt man mit voranstehenden 0-Bytes nicht in Schwierigkeiten: es gibt gar keine.

Die Bedienung des Programms erfährt man, wenn man LBL I abrufft. Die sonstige Register-Belegung:

| | |
|--------|----------------------------|
| Reg 00 | Textlänge |
| 01 } | { enthalten die Zeichen |
| 02 } | { 31 bzw. 32 für den Druck |

03 bis 06
07

bis
133

speichern den Text.
enthalten die Zeichen-Codes.
Und zwar um 6 Einheiten gegen den Standard verschoben. Braucht der Anwender aber nicht berücksichtigen.

Die Zeit für das Ausdrucken eines einzelnen Zeichens beträgt etwa 35 sec. Die Buchstabenbreite kommt auf etwa 25 mm. Auf einem Band von 1 m Länge bringt man somit 40 Zeichen unter.

Zum Betrachten und Lesen des Negativ-Drucks ist ein größerer Augenabstand empfehlenswert.

LBL I (INFO)

- e Programm-Start
- D Druck Text positiv
- d Druck Text negativ
- B Anschluß positiv
- b Anschluß negativ
- A Wiederholung von D oder d
- I INFO

| | | | | |
|----------------------|------------|---------------|-------------|-------------|
| 01*LBL "GROB" | 32 4 | 63 6 | 94*LBL 05 | 125 CHS |
| 02*LBL I | 33 GTO 00 | 64 + | 95 ATOX | 126 2 |
| 03 "A WIEDERHOLUNG" | 34*LBL d | 65 ASTO 03 | 96 X*0? | 127 / |
| 04 PRA | 35 2 | 66 ASHF | 97 GTO 06 | 128 INT |
| 05 "B ANSCHLUSS POS" | 36 GTO 00 | 67 ASTO 04 | 98 DSE 00 | 129 X=0? |
| 06 "t." | 37*LBL D | 68 ASHF | 99 GTO 08 | 130 GTO 04 |
| 07 PRA | 38 CLX | 69 ASTO 05 | 100 SF 01 | 131 GTO 07 |
| 08 "b ANSCHL.NEG." | 39*LBL 00 | 70 ASHF | 101 GTO 02 | 132*LBL 08 |
| 09 PRA | 40 X<>F | 71 ASTO 06 | 102*LBL 06 | 133 CLA |
| 10 "D DRUCK TEXT PO" | 41 31 | 72 ASHF | 103 X<>Y | 134 ARCL 03 |
| 11 "t.s." | 42 ENTER† | 73 ARCL IND X | 104 ACCHR | 135 ARCL 04 |
| 12 PRA | 43 32 | 74 RCL 02 | 105 RDN | 136 ARCL 05 |
| 13 "d DRUCK NEG." | 44 FS?C 01 | 75 FS?C 02 | 106 E | 137 ARCL 06 |
| 14 PRA | 45 X<>Y | 76 GTO 05 | 107 - | 138 FC?C 03 |
| 15 "e START" | 46 STO 02 | 77*LBL 02 | 108 64 | 139 GTO 01 |
| 16 PRA | 47 X<>Y | 78 7 | 109*LBL 07 | 140 END |
| 17 "I INFO" | 48 STO 01 | 79 RCL 02 | 110 - | |
| 18 PRA | 49 "TEXT?" | 80 ACCHR | 111 X<0? | |
| 19 RTN | 50 AON | 81*LBL 03 | 112 SF 00 | |
| 20*LBL e | 51 PROMPT | 82 SF 12 | 113 FS? 00 | |
| 21 134 | 52 AOFF | 83 ACCHR | 114 RCL 02 | |
| 22 PSIZE | 53 GTO 00 | 84 CF 12 | 115 FC? 00 | |
| 23 "CARD" | 54*LBL A | 85 ACCHR | 116 RCL 01 | |
| 24 AVIEW | 55 SF 03 | 86 DSE Y | 117 SF 12 | |
| 25 7.133 | 56 XEQ 08 | 87 GTO 03 | 118 ACCHR | |
| 26 RDTAX | 57*LBL 00 | 88*LBL 04 | 119 CF 12 | |
| 27 RTN | 58 ALENG | 89 RCL 02 | 120 ACCHR | |
| 28*LBL b | 59 STO 00 | 90 ACCHR | 121 RDN | |
| 29 6 | 60*LBL 01 | 91 PRBUF | 122 LASTX | |
| 30 GTO 00 | 61 ATOX | 92 FS?C 01 | 123 ABS | |
| 31*LBL B | 62 XTOA | 93 RTN | 124 FS?C 00 | |

Kurvenapproximation nach Gauss

178 Zeilen, 213 Bytes, 31 Regs.,
SIZE 019

Anpassung von Meßwerten (X_i, Y_i) an eine mathematische Funktion (Zielfunktion $y=f(x)$) mit Hilfe der sog. Regressionsanalyse ist in PRISMA schon des öfteren [1], [2], [3] beschrieben worden. Daher möchte ich auf die Mathematik zur Gauss'schen Methode der kleinsten Fehlerquadrate auf obige Literaturstellen verweisen.

Das hier vorgestellte Programm hat mir jahrelang beste Dienste bei der Bestimmung von SPICE-Parametern zur Simulation von Dioden und Bipolar-Transistoren geleistet. Im Zeitalter der Workstations werden Regressionsanalysen sinnvollerweise durch Newton-Raphson-Iterationsmethoden abgelöst, da hier jede beliebige Zielfunktion ohne mathematischen Aufwand ins Programm eingebaut werden kann. Bei der Regressionsanalyse muß der Anwender die sog. Gauss'schen Normalgleichungen von Hand aufstellen und lösen. Diese Gleichungen sind im allgemeinen N nichtlineare Gleichungen für die zu ermittelnden N Parameter. Löst man diese Gleichungen von Hand in einer geschlossenen Form, und gibt die aufgelöste Form in den Rechner ein, so hat der Rechner die Lösung in wenigen Sekunden gefunden. Die Regressionsanalyse ist daher für langsamere Taschenrechner wie den HP 41C bestens geeignet, da die Lösung im Geradeaus-Verfahren gefunden wird, d.h. nicht auf iterativem Weg zustande kommt. Andererseits ist das Newton-Raphson-Verfahren bei schnellen Rechnern vorzuziehen, da der Rechner selbst die N Gleichungen löst.

In der Elektrotechnik hat man es häufig mit der Bestimmung des Gleichstromarbeitspunktes von Dioden mit in Serie geschalteten Ohmschen Widerständen zu tun. Diese Serienschaltung gehorcht folgender Strom-Spannungs-Beziehung:

$$U = U_T \cdot \ln\left(\frac{I}{I_s} + 1\right) + I \cdot R_s \quad (1)$$

Hierin bedeuten:

$U_T = k \cdot T/q$ die sog. Temperaturspannung

I_s der Sperrsättigungsstrom der Diode

R_s der Bahnwiderstand der Diode.

Mathematisch haben wir die Normalgleichungen nur für die angenäherte Form:

$$Y = A \cdot \ln \frac{X}{B} + C \cdot X \quad (2)$$

der Diodenfunktion gelöst und nach den zu ermittelnden Parametern A, B, C aufgelöst. Daher sind nur Meßwerte mit $X > 0$ für die Regressionsanalyse brauchbar.

Bedienungsanleitung.

1. Programm eintippen und auf Zeile 01 setzen durch „GTO .001“
2. SIZE = 019 eingeben und alle Register mit „CLRG“ löschen.
3. Nacheinander die Meßwerte in der Reihenfolge:
 $U_i \uparrow I_i$ R/S bzw.
 $Y_i \uparrow X_i$ R/S eingeben.
Hierbei wird in den Registern Reg07 bis Reg16 aufsummiert. Als Statistik-Block werden Reg11 bis Reg16 benutzt.
4. Nachdem alle Meßpunkte eingegeben sind, wird mit XEQ 01 die Parameterberechnung gestartet. Zunächst wird der Sperrsättigungsstrom B angezeigt und mit R/S zu den Parametern A (= Temperaturspannung) und C (= Bahnwiderstand) weitergeschaltet.
5. Mit XEQ 02 können die Diodenspannungen auch für Werte, die nicht mit den Meßwerten identisch sind, berechnet werden; dazu ist X bzw. I zuvor ins X-Register zu bringen.

Hinweise:

Alle Ströme werden in den gleichen Einheiten z.B. [mA] eingegeben. Dann erhält man den Sperrsättigungsstrom ebenfalls in [mA]. Im Reg 00 steht die Größe LN(B), in Reg 01 die Größe A, in Reg 17 die Größe C. In Reg 16 finden Sie die Anzahl der Meßpunkte. Falls bei der Meßdaten-Eingabe Fehler entstanden sind, muß beginnend mit Punkt 2. der Bedienungsanleitung neu gestartet werden.

Beispiel:

Eine Meßkurve ist in der Tabelle für $n=8$ Meßwerte gegeben.

Hieraus ergibt sich:

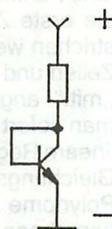
$$B = 1.467E-13 \text{ mA}$$

$$A = 0.0258 \text{ Volt}$$

$$C = 7.3E-3 \text{ kOhm}$$

Für $I_c = 0.5 \text{ mA}$ ergibt sich eine Diodenspannung in Durchlassrichtung von 0.749 Volt.

| X_i | Y_i | i |
|---------|----------|-----|
| I_c | U_{eb} | i |
| .001 mA | .585 V | 1 |
| .003 mA | .613 V | 2 |
| .01 mA | .645 V | 3 |
| .03 mA | .673 V | 4 |
| .100 mA | .705 V | 5 |
| .3 mA | .734 V | 6 |
| 1.00 mA | .771 V | 7 |
| 2.00 mA | .796 V | 8 |



```

01LBL "REGD"
02LBL 00
03 STD 05
04 X<>Y
05 STD 06
06 X<>Y
07 +
08 RCL 05
09 LN
10 ST+ 07
11 X^2
12 ST+ 08
13 RCL 05
14 ENTER^
15 LN
16 *
17 ST+ 09
18 RCL 05
19 LN
20 RCL 06
21 *
22 ST+ 10
23 STOP
24 GTO 00
25LBL 01
26 RCL 11
27 RCL 08
28 *
29 RCL 07
30 RCL 09
31 *
32 -
33 STD 04
34 RCL 11
35 RCL 15
36 *
37 RCL 12
38 RCL 13
39 *
40 -
41 RCL 04
42 *
43 STD 04
44 RCL 11
45 RCL 10
46 *
47 RCL 13
48 RCL 09
49 *
50 -
51 STD 02
52 RCL 11
53 RCL 09
54 *
55 RCL 12
56 RCL 07
57 *
    
```

| | | | |
|-----------|------------|------------|------------|
| 58 - | 91 RCL 15 | 124 X^2 | 157 STD 17 |
| 59 RCL 02 | 92 * | 125 * | 158 "C=" |
| 60 * | 93 RCL 12 | 126 - | 159 ARCL X |
| 61 RCL 04 | 94 RCL 13 | 127 RCL 16 | 160 PROMPT |
| 62 X<>Y | 95 * | 128 RCL 00 | 161LBL 02 |
| 63 - | 96 - | 129 * | 162 STD 18 |
| 64 STD 03 | 97 RCL 02 | 130 RCL 12 | 163 RCL 00 |
| 65 RCL 16 | 98 * | 131 * | 164 E^X |
| 66 RCL 12 | 99 RCL 04 | 132 + | 165 / |
| 67 * | 100 X<>Y | 133 RCL 12 | 166 1 |
| 68 RCL 11 | 101 - | 134 RCL 07 | 167 + |
| 69 X^2 | 102 RCL 03 | 135 * | 168 LN |
| 70 - | 103 X<>Y | 136 - | 169 RCL 01 |
| 71 STD 04 | 104 / | 137 RCL 04 | 170 * |
| 72 RCL 11 | 105 STD 00 | 138 X<>Y | 171 RCL 18 |
| 73 RCL 10 | 106 SCI 2 | 139 / | 172 RCL 17 |
| 74 * | 107 E^X | 140 STD 01 | 173 * |
| 75 RCL 13 | 108 "B=" | 141 "A=" | 174 + |
| 76 RCL 09 | 109 ARCL X | 142 ARCL X | 175 "U=" |
| 77 * | 110 PROMPT | 143 PROMPT | 176 ARCL X |
| 78 - | 111 RCL 11 | 144 RCL 01 | 177 PROMPT |
| 79 RCL 04 | 112 RCL 15 | 145 RCL 16 | 178 END |
| 80 * | 113 * | 146 * | |
| 81 STD 04 | 114 RCL 13 | 147 RCL 00 | |
| 82 RCL 16 | 115 RCL 12 | 148 * | |
| 83 RCL 09 | 116 * | 149 RCL 13 | |
| 84 * | 117 - | 150 + | |
| 85 RCL 11 | 118 STD 04 | 151 RCL 01 | |
| 86 RCL 07 | 119 RCL 11 | 152 RCL 07 | |
| 87 * | 120 RCL 09 | 153 * | |
| 88 - | 121 * | 154 - | |
| 89 STD 02 | 122 RCL 00 | 155 RCL 11 | |
| 90 RCL 11 | 123 RCL 11 | 156 / | |

Literatur:

- [1] C. Rimek, PRISMA 86.2.18
- [2] D. Lochner, PRISMA 86.6.29
- [3] B. Saalfeld, PRISMA 86.8.28

Peter Jochen
Heilbronner Straße 240
7410 Reutlingen

Regression mit Polynomen

HP-15C/HP-28C

Bei allen Programmierbaren von HP gibt es entweder die lineare Regression als festverdrahtete Funktion oder in der Programmsammlung. Mit ihr kann man lineare Funktionen (Geraden) rechnerisch ermitteln, wenn genügend (min. 2) Punkte gegeben sind, wenn z.B. eine Feder in die Länge gezogen wird, liegt oft ein linearer Zusammenhang vor. Zeichnet man die Kräfte und die daraus folgenden Federlängen in ein Koordinatensystem ein, so liegen die Punkte **ungefähr** auf einer Geraden. Mit der linearen Regression läßt sich nun eine Gerade rechnerisch bestimmen, die möglichst nahe an den Punkten vorbeigeht, oder sie sogar berührt. Anschaulich kann man sich das so vorstellen: Die Datenpunkte sind Nägel, die durch Gummibänder mit einem Stab (der Geraden) verbunden sind. Die Gummibänder ziehen nun den Stab in die günstigste Lage. Allerdings reicht eine lineare Funktion nicht immer aus: Bei einem Balken, der an einer Seite fest eingespannt ist, und auf dem eine Last nach außen wandert (z.B. Schwimmer auf dem Sprungbrett), folgt die Durchbiegung unter der Last einer

Funktion 3. Grades. Da die Gleichung und die darin vorkommenden Größen bekannt sind, kann man aus der Entfernung zwischen Last und Einspannung einerseits und der Durchbiegung andererseits das Elastizitätsmodul für den Werkstoff, aus dem der Balken ist, bestimmen.

Unter 1.) findet sich als Beispiel ein Polynom 3. Grades. Um die zunächst unbekanntenen Koeffizienten (a) zu finden, stellt man ein lineares Gleichungssystem auf. Bei diesem fällt der einfache Aufbau (symmetrisch zur Hauptdiagonalen) auf. Um das entsprechende Gleichungssystem für eine Funktion 2. Grades zu erhalten, muß die erste Zeile und die erste Spalte gestrichen werden. Streicht man die ersten Zeilen und Spalten und wendet die unter „mit:“ angegebenen Formeln an, sieht man sofort das Gleichungssystem für die lineare Regression. Analog kann man das Gleichungssystem auch erweitern, um Polynome 4. oder höheren Grades zu berechnen.

Neben der Regression gibt es in der Mathematik noch die Interpolation. Wenn

man das Gleichungssystem für ein Regressionspolynom m-ten Grades aufstellt, und höchstens m+1 Datenpaare zur Berechnung heranzieht, kommt man mit beiden Verfahren zum gleichen Ergebnis. Werden dagegen mehr als m+1 Datenpunkte verwendet, versagt das Interpolationsverfahren, während die Regression eine möglichst gute Näherung findet – mit den Programmen ist es also auch möglich Interpolationspolynome zu erstellen. Im übrigen bieten die Verfahren immer die einfachste Lösung an: Gibt man eine Funktion 4. Grades vor, und führt die Berechnung mit den unter 3.) gegebenen Datenpaaren durch, so ist der Koeffizient von x – im Rahmen der Rechengenauigkeit – Null, wie auch der von x². Eine Restriktion der Standardfunktion L.R. entfällt nebenbei auch: Für m = 1 (lineare Regression) reicht schon ein Datenpaar aus, das Ergebnis ist dann eine konstante Funktion (zur x-Achse parallele Gerade).

Für die Programmierung wurden die unter 2.) benutzten Formeln verwendet. Das die Anordnung etwas anders (transpo-

nierte Koeffizientenmatrix und auf den Kopf gestellter Spaltenvektor) ausfällt als unter 1.) beschrieben, dient nur einer vereinfachten Programmierung. Leider weiß ich nicht mehr, welches Mitglied der Kölner Ortsgruppe mir vor Jahren die Formeln überlassen, und auch ein entsprechendes HP-41-Programm (ohne CCD-Modul) entwickelt hat. Es ist jedoch zu beachten, daß für die meisten HP-Rechner 0° nicht 1 sondern ERROR ist, und deshalb für den Fall $x = 0$ bei der Dateneingabe vorgesorgt werden muß.

Die Programme: Für ein Polynom m-ten Grades muß eine Matrix vom Grad $(m+1) \times (m+2)$ incl. Spaltenvektor erzeugt werden, die aber nur $3m+2$ verschiedene Elemente enthält. Um die Eingabe zu beschleunigen, werden nur diese $3m+2$ Elemente berechnet. Erst wenn zu einem x der y -Wert gesucht wird, stellt der Rechner die Koeffizientenmatrix auf – zusätzlicher Vorteil: Noch bevor alle Datenpaare eingegeben wurden kann zwischendurch schon eine Regression durchgeführt werden. Eine Entfernung falscher Datenpaare ist nicht vorgesehen, weil wegen der hohen Potenzen von x schon eine Größenordnung Unterschied zu großen Rundungsfehlern führt. Bei Falscheingaben muß also ganz von vorne

begonnen werden! Der HP-15C kann höchstens Regressionspolynome 4. Grades berechnen, beim HP-28C hängt das davon ab, wieviele Stellen die Elemente der Koeffizientenmatrix haben; außerdem führt der 28C zusätzlich zur Lösung des Gleichungssystems eine Ein-Schritt-Residuum-Korrektur durch (S. 112 Ref.-Handbuch). Da diese viel zusätzlichen Platz und etwas Zeit benötigt, aber die Verbesserungen nur relativ klein sind, kann sie entfernt werden, wenn die unterstrichenen Befehle im Listing gelöscht werden.

Bedienung HP-15C (in Klammern: HP-28C):

1. Grad des gewünschten Polynoms eingeben, GSB A (INIT). Falls zu wenig Platz im Speicher ist, muß dieser mit 3 DIM (i) (LAST-, UNDO- und COMMAND-Stack abschalten, mit PURGE das USER-Menü kürzen) geschaffen werden.
2. Datenpaar wie bei der Standardfunktion $\Sigma+$ eingeben: y ENTER x , GSB B (Σ). Diesen Schritt für alle Datenpaare wiederholen. Im X-Register (Ebene 1) wird die Nummer des Datenpaares zurückgegeben.

3. Berechnung eines Schätzwertes: x eingeben, GSB C ($X \rightarrow Y$). In X (Ebene 1) wird der berechnete Schätzwert y zurückgegeben. Die Schritte 2. und 3. können beliebig hintereinander ausgeführt werden.

4. Abfragen der Koeffizienten: Dieser Schritt darf nur ausgeführt werden, wenn unmittelbar vorher Schritt 3. ausgeführt wurde. HP-15C: GSB D, in einer Pause wird die Nummer des Koeffizienten angezeigt, dann hält der Rechner mit dem Koeffizienten in X an. Fortsetzung mit R/S.

HP-28C: $AI \rightarrow$ füllt die Stackebenen mit Strings, die die Koeffizienten enthalten. Falls ein Polynom 10. oder höheren Grades angezeigt wird, werden die Nummern der Koeffizienten über 9 etwas merkwürdig angezeigt. Die Koeffizienten selbst, und das Programm sind OK.

Ralf Pfeifer
Rubenstraße 5
5000 Köln 50

| | | | | | | |
|--------------|----------------------|------------------|----------------------|--------------|-------------------|-----------------|
| 000- | HP-15C | 032-42. 5.25 | f DSE I | 064- | 45 2 | RCL 2 |
| 001-42.21.11 | f LBL A | 033-44.40.12 | STO + B | 065- | 45 2 | RCL 2 |
| 002-42.16. 0 | f MATRIX \emptyset | 034- | 34 X<>Y | 066- | 45 2 | RCL 2 |
| 003-42.16. 1 | f MATRIX 1 | 035-45.40.13 | RCL + C | 067- | 0 | \emptyset |
| 004- | 1 1 | 036 ^u | 44 13 USER STO C | 068-42.21. 3 | f LBL 3 | |
| 005- | 40 + | 037- | 22 0 GTO \emptyset | 069- | 20 | x |
| 006- | 36 ENTER \uparrow | 038- | 45 13 RCL C | 070-45.40.15 | RCL + E | |
| 007-42.23.11 | f DIM A | 039- | 31 R/S | 071-42. 5. 0 | f DSE \emptyset | |
| 008- | 1 1 | 040-42.21.13 | f LBL C | 072- | 22 3 | GTO 3 |
| 009-42.23.12 | f DIM B | 041- | 44 2 STO 2 | 073- | 31 | R/S |
| 010- | 30 - | 042-42.16. 1 | f MATRIX 1 | 074-42.21.14 | f LBL D | |
| 011- | 40 + | 043-43. 6. 0 | g F? \emptyset | 075-45.23.12 | RCL DIM B | |
| 012- | 1 1 | 044- | 22 2 GTO 2 | 076- | 44 1 | STO 1 |
| 013-42.23.13 | f DIM C | 045-43. 4. 0 | g SF \emptyset | 077- | 34 | X<>Y |
| 014- | 31 R/S | 046-42.21. 1 | f LBL 1 | 078- | 44 0 | STO \emptyset |
| 015-42.21.12 | f LBL B | 047- | 45 0 RCL \emptyset | 079-42.21. 4 | f LBL 4 | |
| 016-43. 5. 0 | g CF \emptyset | 048-45.40. 1 | RCL + 1 | 080- | 45 0 | RCL \emptyset |
| 017- | 44 2 STO 2 | 049- | 1 1 | 081-45.30. 1 | RCL - 1 | |
| 018- | 34 X<>Y | 050- | 30 - | 082- | 42 31 | f PSE |
| 019- | 44 3 STO 3 | 051- | 43 36 g LST X | 083- | 45 15 | RCL E |
| 020-45.23.12 | RCL DIM B | 052-45.43.13 | RCL g C | 084- | 31 | R/S |
| 021- | 40 + | 053 ^u | 44 11 USER STO A | 085-42. 5. 0 | f DSE \emptyset | |
| 022- | 44 25 STO I | 054- | 22 1 GTO 1 | 086- | 22 4 | GTO 4 |
| 023-42.21. 0 | f LBL \emptyset | 055-42.26.15 | f RESULT E | 087- | 43 32 | g RTN |
| 024- | 45 2 RCL 2 | 056-45.16.12 | RCL MATRIX B | | | |
| 025- | 45 0 RCL \emptyset | 057-45.16.11 | RCL MATRIX A | | | |
| 026- | 45 1 RCL 1 | 058- | 10 \div | | | |
| 027-43.30. 6 | g TEST 6 | 059-42.21. 2 | f LBL 2 | | | |
| 028- | 30 - | 060-45.23.12 | RCL DIM B | | | |
| 029- | 14 y^x | 061- | 44 1 STO 1 | | | |
| 030- | 36 ENTER \uparrow | 062- | 34 X<>Y | | | |
| 031-45.20. 3 | RCL x 3 | 063- | 44 0 STO \emptyset | | | |

1. Berechnung eines Regressionspolynoms 3. Grades:

$$a_3 \cdot x^3 + a_2 \cdot x^2 + a_1 \cdot x + a_0 = f_3(x)$$

Die Koeffizienten erhält man aus folgendem linearen Gleichungssystem:

$$\begin{cases} a_3 \cdot \sum x_i^6 + a_2 \cdot \sum x_i^5 + a_1 \cdot \sum x_i^4 + a_0 \cdot \sum x_i^3 = \sum y_i \cdot x_i^3 \\ a_3 \cdot \sum x_i^5 + a_2 \cdot \sum x_i^4 + a_1 \cdot \sum x_i^3 + a_0 \cdot \sum x_i^2 = \sum y_i \cdot x_i^2 \\ a_3 \cdot \sum x_i^4 + a_2 \cdot \sum x_i^3 + a_1 \cdot \sum x_i^2 + a_0 \cdot \sum x_i^1 = \sum y_i \cdot x_i^1 \\ a_3 \cdot \sum x_i^3 + a_2 \cdot \sum x_i^2 + a_1 \cdot \sum x_i^1 + a_0 \cdot \sum x_i^0 = \sum y_i \cdot x_i^0 \end{cases}$$

mit :

n = Anzahl der ein-
gegebenen Datenpaare

$$\begin{aligned} a_0 \cdot \sum x_i^1 &= a_0 \cdot \sum x_i; & a_0 \cdot \sum x_i^0 &= a_0 \cdot n; \\ a_1 \cdot \sum x_i^1 &= a_1 \cdot \sum x_i; & \sum y_i \cdot x_i^0 &= \sum y_i; \end{aligned}$$

2. Allgemeine Formeln zum Aufbau einer Matrix im Speicher

$$\begin{aligned} M_{i,j} &:= M_{i,j} + x^{(i+j-2)} & i &= \text{Zeilenindex} \\ B_i &:= B_i + y \cdot x^{(i-1)} & j &= \text{Spaltenindex} \\ & & x,y &= \text{aktuelles Datenpaar} \end{aligned}$$

$M_{i,j}$ = Element aus der Koeffizientenmatrix **M** des Gleichungssystems

B_i = Element aus dem Spaltenvektor **B** des Gleichungssystems

Löst man das Gleichungssystem $\mathbf{M}\mathbf{A} = \mathbf{B}$ nach **A** auf, so erhält man $\mathbf{A} = \mathbf{M}^{-1} \cdot \mathbf{B}$. Die Elemente des Lösungsvektors **A** sind die gesuchten Koeffizienten

$$a_0, a_1, \dots \text{ mit: } a_i := A_{i-1}$$

3. Beispiel. Aus folgenden Datenpaaren (x|y): (-1|-10), (0|-3), (1|4), (2|41), (5|632) ermitteln die Programme das Regressionspolynom $f(x) = 5x^3 + 2x - 3$. An der Stelle $x = -3$ wird der Wert $y = -144$ interpoliert. Der Koeffizient von x^2 wird **nicht** exakt als \emptyset berechnet.

HP-28C: Regression mit Polynomen

```

« N 1 FOR i 'L' i GET →STR "X^" + i 47 + CHR + -1
STEP »
« → x « IF 1 DUP FC?C ≠ THEN B Ø N 1 - FOR i 1 N FOR
j 'A' i j + GET NEXT NEXT N N 2 →LIST →ARRY → a « a /
B a 3 PICK RSD a / + » ARRY→ LIST→ DROP →LIST 'L' STO
END Ø N 1 FOR i x X 'L' i GET + -1 STEP » » 'X→Y' STO
« 1 SF → y x « 1 N DUP + 1 - FOR i 'A' i DUP2 GET x i
1 - ^ + PUT NEXT 1 N FOR i 'B' i 1 →LIST DUP2 GET x i 1
- ^ y X + PUT NEXT 'A' 1 GET » » 'Σ' STO
« 1 + 'N' STO N 1 →LIST 1 N N + START Ø NEXT N DUP + 1
- →LIST 'A' STO CON 'B' STO » 'INIT' STO
    
```

Hinweis : x = Variable, X = Multiplikation.

XROM (Jakob) 1603

Computerclub Deutschland e.V.

Zeile 1 (1- 5) CCD-Barcodes

 Zeile 2 (5- 9) CCD-Barcodes

 Zeile 3 (9- 17) CCD-Barcodes

 Zeile 4 (17- 26) CCD-Barcodes

 Zeile 5 (26- 34) CCD-Barcodes

 Zeile 6 (34- 45) CCD-Barcodes

 Zeile 7 (45- 53) CCD-Barcodes

 Zeile 8 (53- 57) CCD-Barcodes

 Zeile 9 (57- 65) CCD-Barcodes

 Zeile 10 (65- 75) CCD-Barcodes

 Zeile 11 (75- 84) CCD-Barcodes

 Zeile 12 (84- 93) CCD-Barcodes

 Zeile 13 (93- 102) CCD-Barcodes

 Zeile 14 (102- 111) CCD-Barcodes

 Zeile 15 (111- 119) CCD-Barcodes

 Zeile 16 (119- 124) CCD-Barcodes


K3SATZ (Hochenegger) 1602

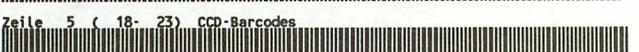
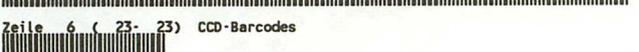
Computerclub Deutschland e.V.

Zeile 1 (1- 4) CCD-Barcodes

 Zeile 2 (4- 8) CCD-Barcodes

 Zeile 3 (8- 12) CCD-Barcodes

 Zeile 4 (12- 18) CCD-Barcodes

 Zeile 5 (18- 23) CCD-Barcodes

 Zeile 6 (23- 23) CCD-Barcodes


XROMS (Lutz) 1604

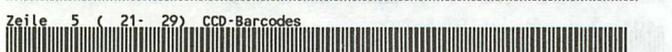
Computerclub Deutschland e.V.

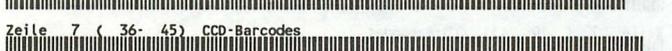
Zeile 1 (1- 3) CCD-Barcodes

 Zeile 2 (3- 4) CCD-Barcodes

 Zeile 3 (4- 13) CCD-Barcodes

 Zeile 4 (13- 21) CCD-Barcodes

 Zeile 5 (21- 29) CCD-Barcodes

 Zeile 6 (29- 36) CCD-Barcodes

 Zeile 7 (36- 45) CCD-Barcodes

 Zeile 8 (45- 55) CCD-Barcodes

 Zeile 9 (55- 63) CCD-Barcodes

 Zeile 10 (63- 69) CCD-Barcodes

 Zeile 11 (69- 75) CCD-Barcodes

 Zeile 12 (75- 81) CCD-Barcodes

 Zeile 13 (81- 84) CCD-Barcodes

 Zeile 14 (84- 87) CCD-Barcodes

 Zeile 15 (87- 90) CCD-Barcodes

 Zeile 16 (90- 94) CCD-Barcodes

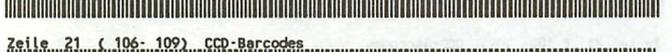
 Zeile 17 (94- 96) CCD-Barcodes

 Zeile 18 (96- 99) CCD-Barcodes

 Zeile 19 (99- 103) CCD-Barcodes

 Zeile 20 (103- 106) CCD-Barcodes

 Zeile 21 (106- 109) CCD-Barcodes

 Zeile 22 (109- 110) CCD-Barcodes


GROB (Berghaus) 1605

Computerclub Deutschland e.V.

Zeile 1 (1- 3) CCD-Barcodes
Zeile 2 (3- 4) CCD-Barcodes
Zeile 3 (4- 5) CCD-Barcodes
Zeile 4 (5- 8) CCD-Barcodes
Zeile 5 (8- 10) CCD-Barcodes
Zeile 6 (10- 10) CCD-Barcodes
Zeile 7 (10- 13) CCD-Barcodes
Zeile 8 (13- 15) CCD-Barcodes
Zeile 9 (15- 17) CCD-Barcodes
Zeile 10 (17- 23) CCD-Barcodes
Zeile 11 (23- 28) CCD-Barcodes
Zeile 12 (28- 36) CCD-Barcodes
Zeile 13 (36- 44) CCD-Barcodes
Zeile 14 (44- 51) CCD-Barcodes
Zeile 15 (51- 58) CCD-Barcodes
Zeile 16 (58- 67) CCD-Barcodes
Zeile 17 (67- 76) CCD-Barcodes
Zeile 18 (76- 84) CCD-Barcodes
Zeile 19 (84- 92) CCD-Barcodes
Zeile 20 (92- 100) CCD-Barcodes
Zeile 21 (100- 109) CCD-Barcodes
Zeile 22 (109- 118) CCD-Barcodes
Zeile 23 (118- 127) CCD-Barcodes
Zeile 24 (127- 136) CCD-Barcodes
Zeile 25 (136- 140) CCD-Barcodes

REGD (Jochen) 1606

Computerclub Deutschland e.V.

Zeile 1 (1- 6) CCD-Barcodes
Zeile 2 (6- 17) CCD-Barcodes
Zeile 3 (17- 27) CCD-Barcodes
Zeile 4 (27- 40) CCD-Barcodes
Zeile 5 (40- 53) CCD-Barcodes
Zeile 6 (53- 65) CCD-Barcodes
Zeile 7 (65- 78) CCD-Barcodes
Zeile 8 (78- 90) CCD-Barcodes
Zeile 9 (90- 103) CCD-Barcodes
Zeile 10 (103- 112) CCD-Barcodes
Zeile 11 (112- 125) CCD-Barcodes
Zeile 12 (125- 137) CCD-Barcodes
Zeile 13 (137- 146) CCD-Barcodes
Zeile 14 (146- 158) CCD-Barcodes
Zeile 15 (158- 167) CCD-Barcodes
Zeile 16 (167- 176) CCD-Barcodes
Zeile 17 (176- 178) CCD-Barcodes

INTERNATIONAL CONFERENCE FOR HAND-HELD COMPUTER USERS

CORVALLIS, OREGON August 4, 5, 6 1988

Welcome to the World Center of Hand-Held Computing!

Your conference committee (Sherman Lowell, Conference Chairman; Jeremy Smith, Program Chairman; Ron Brooks; Eric Gakstatter; Brian Walsh, HPX User Club Liaison) cordially invites all users of hand-held computers and programmable calculators to attend a three day meeting and exhibit fair in Corvallis, Oregon.

The Conference theme is "The Past, Present, and Future of the Hand-Held Computing". It is our hope to have members of the original design teams tell us the challenges that were met and overcome in the development of various hand-held calculators. There will also be several sessions for contributed papers. Local companies will be hosting various activities, such as; tours, luncheons, and a western-style barbecue.

The Conference meets in the LaSells Stewart Center on the Oregon State University campus beginning at 9 a.m., Thursday, August 4th. You may register by mail (check or money order payable to "Handheld Computing" 301 NE Byron Pl., Corvallis, OR 97330) or at the beginning of the conference. The cost per participant is \$85.00 (waived for speakers). It includes 3 luncheons, the barbecue, refreshments during the sessions, and a copy of the conference proceedings.

Exhibitor tables (8' x 5' with electrical service provided) are available in protected exhibit rooms at a fee of \$200.00. (Space and cost may be shared, if desired.) Please contact Eric Gakstatter. Tel. (503) 752-5456, as soon as possible, if you desire space.

We invite you to submit papers for presentation at the Conference on any aspect of handheld computing. The published version of papers will not be restricted in length, but the oral version should be limited to 20 minutes. Time will be allotted for discussion. We can accept papers in a variety of ways:
Via E-mail at

BitNet: Jeremy@OrState
DASnet: Jeremy@DDB1JS.DAS.NET
GEnie: Jeremy.Smith
FidoNet: 152/202 or 152/203
UseNet: smithj@k9.cs.OrSt.EDU

Via text files on magnetic media (include data on how files were prepared).

Via printed copy for reproduction on 8.5" x 11" paper (on A4 size paper include 2.5cm margins sides and top, and a 5cm margin on the bottom).

Send magnetic media or legible, ready-to-copy papers as soon as possible to Jeremy Smith at 301 NE Byron Place, Corvallis Oregon 97330 USA. Papers received by June 30th and accepted will be published in the conference proceedings; later arrivals may not make it.

Lodging during the Conference is available in 3 classes of accommodation: (a) Dormitory single or double rooms with bathroom facilities shared on floor; (b) Dormitory single or double rooms with facilities shared between two rooms; or (c) Motel rooms. Dormitory space is close to the conference center. (No alcoholic beverages are permitted on campus). Conferees arriving in Corvallis on Wednesday, August 3rd, are invited to meet at a no-host gathering in the Nendels Inn Cocktail lounge 7-10 p.m.

During the Conference the Weyerhaeuser Room of the LaSells Stewart Center will be used for disk and document exchange, and copying of disks, tapes, and barcode. The equipment can be used by registrants without charge. We will try to make user club publications available also. Desk space will be provided for user clubs represented at the Conference.

In order that we can anticipate more accurately the needs of conferees, please return the enclosed form as soon as possible. We cannot possibly ensure that all interested users of hand-held computers and programmable calculators have the most up-to-date information about the conference, your committee asks that you pass it on.

Finally, we would like to remind you that Corvallis is in the midst of some of the most scenic vacation areas in the United States and it would be a shame if you leave Oregon after the Conference without having had the opportunity to sample its treasures. We look forward to seeing you here!

Sincerely,

Sherman Lowell,
conference chairman

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)
Schnuckentrift 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)
Schülerweg 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 Ortmann (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

Postvertriebsstück
Gebühr bezahlt

D 2856 F

CCD – Computerclub Deutschland e.V.
Schwalbacherstraße 50
D-6000 Frankfurt am Main 1

CCD

ISSN 0176-8735

PRISMA

mai/juni 1988 Nr. 3

„Cursor“ Zeitschriftenkritik

CURSOR, eine neue
Computerzeitschrift?

Neugierig geworden durch die Ankündigung in der letzten PRISMA, bestellte ich für DM 5 das Probeheft der Zeitschrift CURSOR.

Allerdings war ich nach Erhalt des Exemplars sehr enttäuscht.

Diese „neue Zeitschrift auf dem hart umkämpften Computerzeitschriftenmarkt“ (Wortlaut CURSOR Editorial) werte ich persönlich als reines Werbeprospekt der Fa. W&W Software Products. Zwar sind auf ca. 35 Seiten Hochglanzpapier die Produkte anschaulich und informativ dar-

gestellt, jedoch sollte man sein Geld besser für andere Informationen ausgeben.

Kurzes Fazit (ich will nicht noch mehr wertvollen Platz unserer PRISMA für eine Katalogbeschreibung verschwenden):

Wer eine ausführliche W&W-Zubehörliste braucht, womöglich noch im Abonnement, soll aufmerksam PRISMA studieren bzw. den persönlichen Erfahrungsaustausch mit CCD-Mitgliedern suchen.

Siegfried Zeidler
Bruhweg 40
7100 Heilbronn

Thermodrucker mit Infrarot-Modul

Der Thermodrucker 82240A der neuen Serie 20 ist nun auch mit dem Infrarot-Modul 82242AB für den HP-41 zu verwenden. Der Drucker kann allerdings weniger als der HP-IL Thermodrucker.

Alf-Norman Tietze

*Hardware · Software
Servicestation
Beratung · Zubehör*

WORDLORD · Textverarbeitung · CAD-Anwendungen · Komplettsysteme

OSBORNE
Management by Computer.

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